Categorizzazione automatica di immagini mediante algoritmi neurali.

Sara Colantonio, Ovidio Salvetti

Abstract**:** - Il problema dell'elaborazione delle immagini, mirata all'interpretazione e alla classificazione del contenuto delle stesse, ha attirato l'attenzione dei ricercatori dai primi tempi della nascita e diffusione dei calcolatori: tra le prime applicazioni studiate si può annoverare il riconoscimento di caratteri alfabetici stampati o manoscritti per facilitare le operazione di ingresso nel calcolatore e successiva conversione in documenti di natura elettronica. Con il progredire della tecnologia dei sistemi di calcolo, la categorizzazione di immagini ha trovato applicazioni sempre più vaste, riguardando discipline di nuova generazione come l'*Image* e *Scene Analysis*, l'*Image Understanding*, l'*Object Recognition* e la *Computer Vision*, con applicazioni del tutto generali sia in ambiti scientifici che umanistici. A livello più generale, il problema rientra nell'ambito del *Pattern Recognition* (PR), la disciplina che si occupa degli approcci applicativi mirati al riconoscimento e alla classificazione automatici dell'entità di interesse di un fenomeno in osservazione.
Obiettivo di questo lavoro è l'analisi dei problemi connessi alla categorizzazione di immagini, quali la costruzione di un sistema di riconoscimento, la rappresentazione dei *pattern*, la selezione ed estrazione delle *feature* e la definizione di un metodo di riconoscimento con particolare riferimento all'applicazione delle Reti Neurali Artificiali. In particolare, vengono analizzate e comparate diverse soluzioni, individuando come soluzione ottimale un sistema neurale gerarchico per l'elaborazione e l'interpretazione delle immagini, costituito da moduli *Error Back-Propagation* e *Self-Organizing Map* strutturati in due livelli successivi.

Key-Words**:** - *Image Classification, Pattern Recognition, Image Analysis, Neural Networks, Error Back-Propagation, Self-Organizing Maps, Hierarchical Neural Network, Features Evaluation and Selection.*

Introduzione.

Il riconoscimento, la descrizione e la classificazione automatici delle strutture contenute nelle immagini costituiscono una questione di fondamentale importanza in un vasto insieme di discipline scientifiche ed ingegneristiche che richiedono l'acquisizione, il trattamento e la trasmissione di informazione sotto forma visuale ovvero di immagini. Le tecniche di produzione di immagini digitali (*digital imaging*) sono in continua evoluzione non solo sotto l'aspetto dell'affinamento tecnologico (che consente di ottenere immagini sempre più precise e dettagliate), ma anche dal punto di vista concettuale: le immagini ricavate, infatti, soprattutto grazie all'intervento della elaborazione elettronica, perdono gran parte del loro carattere puramente "iconografico", per acquisire invece un sempre maggiore significato funzionale, con un contenuto di informazioni tale da richiedere l'ausilio del calcolatore per essere correttamente interpretato. Esempi di applicazioni in tal senso sono:

- il riconoscimento di caratteri;
- l'analisi delle immagini biomediche (*Medical Imaging*);
- l'automazione industriale;
- la robotica;
- la cartografia;
- il telerilevamento;
- la modellazione ambientale;
- la simulazione e il controllo di mobilità nei trasporti;
- la biometrica
- la conservazione dei beni culturali
- il riconoscimento di immagini radar in ambito militare.

Il contesto teorico nel quale si collocano le questioni introdotte è quello del *Pattern Recognition* (PR) o *Classification* combinato all'*Elaborazione* ed all'*Analisi dell'Immagine* (*Digital Image Processing and Analysis*). La prima disciplina si occupa del processo di riduzione dell'informazione che consiste nell'attribuzione di pattern visuali o logici ad una classe di appartenenza, sulla base delle caratteristiche e delle relazioni tra essi [1; 2]. L'*Image Processing* si occupa delle tecniche e delle metodologie volte all'acquisizione e all'elaborazione numerica dei dati, la ricostruzione, il miglioramento ed il restauro delle immagini [2; 3; 4]. Infine, l'*Image Analysis* si occupa del processo di elaborazione delle immagini volto all'estrazione di informazioni significative necessarie per la descrizione delle strutture in esse contenute. Tali informazioni risultano indispensabili all'interno di un sistema di PR che prevede l'ulteriore elaborazione dei dati così ottenuti in un successivo processo di interpretazione e classificazione.

In merito, poi, ai metodi che permettano tale interpretazione, le *Reti Neurali Artificiali* (*Artificial Neural Network*, ANN o semplicemente NN), nate dall'idea di simulare il comportamento del sistema nervoso biologico, si prestano come ottime alternative alle altre tecniche di PR, grazie alle caratteristiche funzionali che le rendono estremamente attraenti e capaci di manipolare fenomeni molto complessi, in modo relativamente semplice.

Nelle sezioni che seguono vengono discussi gli aspetti teorici-applicativi della realizzazione di un sistema per la categorizzazione automatica dei dati immagine, viene analizzata l'applicazione delle Reti Neurali al problema considerato e sono analizzate e comparate diverse architetture neuronali, soprattutto gerarchiche, individuandone quella che risulta essere la migliore in termini di adattablità e affidabilità.

1. Pattern Recognition e Image Analysis.

Un problema di Pattern Recognition può essere formalizzato nel modo seguente [6; 7; 8; 9]:

dati
- *uno spazio di misurazione* M *sul fenomeno in osservazione;*
- *un insieme di significati, detto spazio di interpretazione,* Ω

individuare
- *uno spazio di rappresentazione* S_P *dei pattern* P*;*
- *una funzione di decisione* Ψ *che associ le rappresentazioni di* S_P *allo spazio dei concetti* Ω*:*

$$\Psi: S_P \rightarrow \Omega \qquad (1)$$

L'individuazione della funzione Ψ corrisponde nella pratica a partizionare lo spazio delle misurazioni in regioni non sovrapposte, ciascuna delle quali rappresenti un concetto, ovvero una classe. Suddetta partizione è raramente lineare o realizzabile con funzioni semplici (funzioni lineari o quadratiche), solitamente, invece, per poter ottenere una corretta separazione dei pattern appartenenti a classi diverse è necessario ricorrere a funzioni altamente non lineari.

Generalmente, i sistemi di riconoscimento sono *adattivi*, ovvero richiedono una fase di *addestramento* (*training*), durante la quale viene presentato al classificatore un insieme di pattern rappresentativo del problema, il cosiddetto *insieme di addestramento* (*training set*), opportunamente scelto per permettere al sistema di *apprendere*, ovvero di acquisire la conoscenza relativa alle caratteristiche che

distinguono le varie classi. In particolare, il processo di addestramento può avvenire secondo due modalità diverse, in riferimento alla presenza o meno della conoscenza preventiva dei significati e delle classi di appartenenza dei pattern. Nel caso in cui, sia possibile stabilire quali siano le classi e a quale di esse appartenga ciascun pattern di *training*, si parla di apprendimento supervisionato e il partizionamento dello spazio S_P è guidato dalle etichette associate ai pattern. Nel caso in cui non sia disponibile una conoscenza iniziale sul numero di classi e, di conseguenza i pattern di *training* vengano forniti al classificatore senza una etichettatura, si parla di apprendimento non-supervisionato e la partizione avviene in base alle caratteriste degli esempi, ovvero tenendo conto di come questi si aggreghino "spontaneamente", formando degli insiemi aventi proprietà simili. In quest'ultimo caso, si parla di *clustering*, definendo *cluster* i gruppi di pattern simili.

La costruzione di un sistema di PR richiede essenzialmente la messa a punto di tre aspetti:

1. l'acquisizione e la pre-elaborazione dei dati;
2. la rappresentazione dei pattern;
3. la definizione ed il perfezionamento di una funzione di decisione per il riconoscimento degli stessi.

Tali aspetti si traducono in una serie di fasi successive attraverso le quali si evolve il processo di progettazione di un sistema di PR [1; 8; 10]: dopo aver deciso le modalità di raccolta dei dati, è necessario eseguire una fase di pre-elaborazione degli stessi per renderli adatti alle analisi successive; quindi, selezionare un opportuno modello di rappresentazione dei pattern, anche in riferimento al metodo di riconoscimento adottato e approntato nella fase successiva. Esistono, infatti, diversi approcci al PR e ciascuno di essi richiede una specifica modalità di rappresentazione dei dati attraverso descrittori, caratteristiche o *feature* oppure primitive.

Nel caso in cui i dati siano immagini, il processo di sviluppo di un sistema di PR è generalmente istanziato con operazioni specifiche che riguardano la formazione delle immagini, la loro elaborazione, l'estrazione delle strutture da riconoscere o classificare e la rappresentazione di quest'ultime, operazioni che ricadono, come anticipato, nell'ambito dell'*Elaborazione* ed dell'*Analisi dell'Immagine*. In particolare, tale processo può essere schematizzato come mostra il diagramma in Figura 1, nella quale sono riportati anche le operazioni eseguite per ciascuna fase e i risultati corrispondenti.

Le prime due fasi sono legate al processo di elaborazione delle immagini *in stricto sensu*, ovvero alla digitalizzazione dei dati analogici e all'ottimizzazione delle immagini così ottenute; le due fasi successive appartengono al processo di analisi delle immagini digitali, ovvero dell'*Image Analysis*: l'estrazione delle strutture di interesse, attraverso un processo che prende il nome di *segmentazione* e la definizione e la stima di un insieme di descrittori o di caratteristiche significativi utilizzati per la rappresentazione dei *pattern*. L'ultima fase, infine, consiste nella definizione di un metodo di classificazione che sfrutti quanto ottenuto dalle fasi precedenti, traducendo l'analisi quantitativa in informazioni qualitative. E' possibile prevedere, inoltre, una fase di post-elaborazione in cui una qualche forma di conoscenza a priori del problema affrontato permetta di migliorare l'intero processo di classificazione [10].

2. Acquisizione delle immagini.

Le immagini digitali vengono opportunamente acquisite attraverso un processo di *digitalizzazione* applicato ad immagini o segnali analogici che restituisce un'immagine numerica su un supporto accessibile da parte di un dispositivo di calcolo [4]. Tale processo si articola in due fasi:

- *campionamento;*
- *quantizzazione* ;

durante la fase di campionamento, il segnale analogico viene misurato ad intervalli

regolari, assumendo che il segnale in uscita mantenga costante il valore misurato fino all'intervallo successivo. Ad ogni valore campionato, viene associato, durante la fase di

piuttosto di una regione rettangolare coincidente con una cella della griglia, per cui il valore ad esso associato rappresenta l'intensità media di un'intera regione.

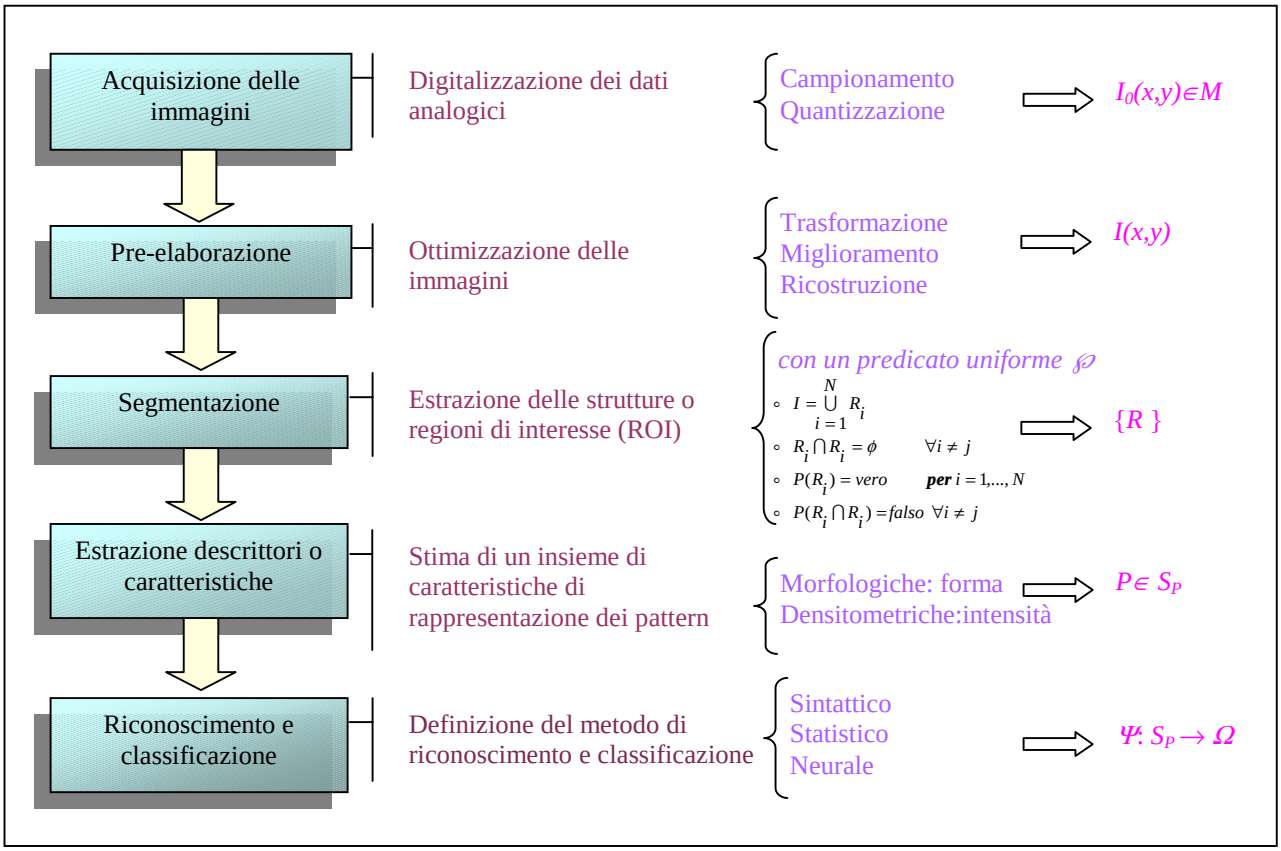

Figura 1. *Diagramma di sviluppo di un sistema di riconoscimento e classificazione di immagini con rispettive operazioni e risultati corrispondenti.*

quantizzazione, l'elemento più vicino tra quelli che appartengono ad un insieme prestabilito e finito di valori.

L'immagine ottenuta può essere rappresentata come una funzione bi-dimensionale, rappresentata come una matrice

$$I(x,y)$$

i cui elementi corrispondono alle di unità base, i *pixel* o *picture element*, identificati dalla rispettiva posizione (x,y) e dalle relative proprietà in termini di colore ed intensità. Ogni pixel rappresenta l'intensità della corrispondente posizione della griglia di campionamento dell'immagine. In realtà, non si tratta di un solo punto dell'immagine, ma

Elementi caratteristici di un'immagine sono la risoluzione di acquisizione o visualizzazione, detta *risoluzione spaziale*, pari alla densità misurata in *pixel per inch*, e il tipo, *bianco/nero*, *scala di grigi* o *colori*, che identifica la quantità di informazione associata ad ogni pixel, dal processo di quantizzazione. Le immagini dei primi due tipi vengono definite *monocromatiche* e sono rappresentate da funzioni $I(x,y)$ a valori scalari: binari nel caso di immagini in bianco/nero, compresi in un intervallo $[0,G_{max}\text{-}1]$ nel caso di immagini su scala di grigi a G_{max} valori di intensità (o di grigio). In particolare, in quest'ultimo caso, il valore G_{max} corrisponde solitamente ad una potenza di due:

$$G_{max} = 2^L$$

dove L è il numero di bit usato per codificare ciascun pixel e ne stabilisce la *profondità*.

Nel caso di immagini a colori, la funzione $I(x,y)$ non è scalare, ma vettoriale: ogni elemento è un vettore di tre componenti:

$$I(x,y)=[\ I_1(x,y),\ I_2(x,y),\ I_3(x,y)]$$

ciascuno dei quali rappresenta un *canale* e corrisponde ad una delle componenti dello spazio dei colori. La considerazione di immagini di questo tipo esula dagli obiettivi di questo lavoro, per cui di seguito si farà sempre riferimento ad immagini mono-cromatiche.

3. Pre-elaborazione.

I processi di formazione e acquisizione dell'immagine non distinguono fra informazione utile o inutile, fra informazione rumorosa o significativa, veicolando i dati tutti in un'unica struttura. La presenza di rumore, tuttavia, comporta un fenomeno di degradazione dell'immagine, al quale è utile ed opportuno porre rimedio attraverso una fase di pre-elaborazione dell'immagine che permetta di migliorare la qualità dei dati in ingresso, prima di procedere nella fase di estrazione dell'informazione. Le operazioni usate a tal scopo sono tipiche dell'*Elaborazione delle Immagini* e consistono, essenzialmente, in procedimenti di riduzione del rumore e ricostruzione. Il risultato così ottenuto rimane un'immagine delle stesse dimensioni e con le stesse caratteristiche di quella iniziale.

E' possibile individuare tre classi diverse di elaborazioni di questo tipo [3; 4; 5], che corrispondono alle operazioni di:

- trasformazione;
- miglioramento (*enhancement*)
- ricostruzione (*restoration*).

Tali classi non sono completamente disgiunte l'una dall'altra, soprattutto per quanto concerne le trasformazioni.

Trasformazione.
La teoria delle trasformazioni bidimensionali gioca un ruolo particolare all'interno dell'elaborazione delle immagini: le varie trasformate sono spesso utilizzate come basi di partenza per la costruzione di algoritmi che si occupino delle operazioni di miglioramento e ricostruzione. La trasformazione in assoluto più studiata ed utilizzata è la *Trasformata di Fourier*, questo grazie alle proprietà di cui essa gode [4].

Miglioramento.
L'obiettivo delle operazioni appartenenti a questa classe è quello di migliorare alcune proprietà dell'immagine per agevolarne le successive elaborazioni di analisi o visualizzazione. Il risultato prodotto non altera né aumenta il contenuto informativo delle immagini, ma ne enfatizza solamente alcune caratteristiche. Gli algoritmi di image enhancement sono generalmente interattivi e dipendenti dall'applicazione. Gli esempi tipici sono rappresentati dall'aumento dei contrasti o dei bordi (*edge*), il filtraggio del rumore, l'assottigliamento (*sharpering*) o l'ingrandimento (*magnifying*) dei contorni.

Ricostruzione.
In maniera più o meno accentuata le immagini acquisite sono sempre soggette ad errori, i quali possono considerevolmente diminuire le prestazioni di un classificatore. Il termine errore indica tutte quelle situazioni in cui l'insieme delle misure raccolte sono falsate da disturbi introdotti dai meccanismi di acquisizione, oppure i casi in cui il rumore sia presente alla sorgente. L'obiettivo delle tecniche di ricostruzione è, dunque, quello di eliminare o minimizzare tali degradazioni, generalmente attraverso l'applicazione di filtri.
Mentre le tecniche di miglioramento della qualità dell'immagine si possono descrivere in modo qualitativo, senza ricorrere all'uso degli strumenti matematici, le metodologie per il ripristino della qualità dell'immagine, non possono essere spiegate senza ricorrere alla formalizzazione. Generalmente si può dire che il restauro di immagini prevede due fasi: l'analisi delle fonti di deterioramento e la

compensazione delle degradazioni che ne derivano, attraverso l'utilizzo di particolari algoritmi. I contesti di applicazione più significativi di queste tecniche sono l'elaborazione di immagini spaziali, di immagini biomediche o da telerilevamento

Dal punto di vista della computazione, le tecniche di elaborazione possono essere raggruppate anche in base della regione dell'immagine sulla quale vanno ad operare e vengono, secondo tale criterio, distinte in puntuali, locali o globali. Le tecniche puntuali, dette anche manipolazioni, producono semplicemente un cambiamento nella luminosità delle scale di grigio. Le tecniche locali operano invece su piccole aree, concentrando l'elaborazione al punto correntemente analizzato. Le tecniche di questa classe hanno lo scopo di ridurre il rumore e aumentare il contrasto. Le elaborazioni globali, che richiedono l'uso di tutta l'immagine in ingresso, si possono considerare come evoluzioni matematiche più sofisticate ed elaborate delle tecniche precedenti (ne è un esempio la trasformata di Fourier).

4. Segmentazione.

Il processo di *segmentazione* di un'immagine rappresenta un passo fondamentale nell'ambito dell'*Image Analysis* e del *Pattern Recognition*, giacché permette di estrarre quelle che sono le strutture di interesse da riconoscere e classificare. Si tratta, dunque, di una fase critica nell'analisi dell'immagine: la precisione e la qualità del risultato possono influenzare molto pesantemente le elaborazioni successive.

L'obiettivo della segmentazione è la scomposizione di un'immagine in parti o ROI distinte, che siano significative rispetto all'applicazione e risultino omogenea rispetto ad una data caratteristica, se prese singolarmente, e disomogenee se unite ad una delle regioni adiacenti.

La segmentazione può essere effettuata manualmente da un operatore, sfruttando per esempio degli strumenti di selezione di aree, con il vantaggio dell'automatica valutazione del risultato ottenuto. Tuttavia, un simile approccio è estremamente svantaggioso in termini di tempo e risorse richieste, soprattutto nel caso in cui l'immagine contenga un gran numero di oggetti significativi, per ognuno di quali risulterebbe, quindi, necessaria l'estrazione manuale del contorno. E', pertanto, auspicabile poter disporre di metodi automatici, o quanto meno semi-automatici, che facilitino l'operazione di segmentazione, sulla base di tecniche di analisi delle immagini e di corrispondenti criteri di applicazione, eventualmente sviluppati per la specifica tipologia di immagini trattate. Un approccio formale in tal senso prevede la seguente definizione [11]:

Definizione. *Segmentazione.*
La segmentazione di un'immagine I con un predicato uniforme P è la partizione di I in sottoinsiemi R_1; … ;R_N non vuoti e disgiunti tali che:

$$\begin{aligned} &\circ\ I = \bigcup_{i=1}^{N} R_i \\ &\circ\ R_i \cap R_i = \phi \qquad \forall i \neq j \\ &\circ\ P(R_i) = vero \qquad \text{per } i = 1,\ldots,N \\ &\circ\ P(R_i \cap R_i) = falso \quad \forall i \neq j \end{aligned} \tag{2}$$

Inoltre R_i e R_j sono adiacenti e il predicato uniforme P assegna valore vero o falso in relazione alle proprietà di luminosità dei punti di X.

I sottoinsiemi R_i corrispondono sostanzialmente a parti generiche in cui viene suddivisa l'immagine, mentre ciò che si desidera ottenere sono delle regioni che siano insiemi connessi di pixel, ovvero costituite da pixel adiacenti. Formalmente, è possibile considerare la seguente definizione di *connettività* di una regione [12]:

Definizione. *Connettività*
Una regione R è detta connessa se:

- *ogni coppia di pixel (x_a,y_a), (x_b,y_b) appartenente ad R può essere connessa attraverso un cammino*
- *(x_a,y_a),…, (x_{i-1},y_{i-1}), (x_i,y_i), (x_{i+1},y_{i+1}),…, (x_b,y_b)*

- *ogni pixel (x_i,y_i) è adiacente ai pixel precedente (x_{i-1},y_{i-1}) e successivo (x_{i+1},y_{i+1}) lungo il cammino considerato.*

La relazione di adiacenza tra pixel è legata a quella di *vicinato* (*neighbourhood*) che può essere distinta in due casi possibili:

Definizione. *Vicinato*
S*i parla di relazione di vicinato di ordine 4 (4-neighbourhood o 4-connettività) nel caso in cui si considerino connessi al pixel (x,y) i pixel ad esso adiacenti in verticale e in orizzontale:*

$$N_4((x,y)) = \{(x-1,y),(x+1,y),(x,y-1),(x,y+1)\} \quad (3)$$

si parla di relazione di vicinato di ordine 8 (8-neighbourhood o 8-connettività) nel caso in cui si considerino connessi al pixel (x,y) i pixel ad esso adiacenti in verticale, in orizzontale e in diagonale

$$N_8((x,y)) = N_4((x,y)) \cup \begin{Bmatrix}(x-1,y-1),(x+1,y+1),\\(x+1,y-1),(x-1,y+1)\end{Bmatrix} \quad (4)$$

Varie sono le tecniche che a partire dalle nozioni appena introdotte sono state sviluppate e proposte in letteratura. La maggior parte di esse si basa sulle proprietà del contenuto dell'immagine, come intensità, colore o geometria, per poter identificare delle ROI omogenee e i contorni delle stesse [11; 13; 14]. In particolare, le tecniche maggiormente diffuse possono essere ricondotte a due approcci principali:

- *approccio basato sulle regioni;*
- *approccio basato sui contorni*.

i quali a loro volta si basano su due diverse proprietà dei livelli di grigio dell'immagine[1] :

- la *somiglianza* e
- la *discontinuità;*

Nel primo caso il processo di segmentazione procede riconoscendo le regioni sulla base delle loro caratteristiche di similitudine, gli esempi tipici sono la segmentazione per *sogliatura* dell'istogramma dei livelli di grigio dell'immagine (*Histogram Thresholding* o semplicemente *Thresholding*) e i metodi di *split & merge* e di *accrescimento delle regioni* (*region growing)*; nel secondo le immagini vengono suddivise in base alle variazioni dei toni di grigio, l'esempio tipico è quello del metodo di *individuazione dei bordi* (*edge detection*). Due metodi di natura diversa possono, tuttavia, coesistere in quanto dai contorni è possibile ottenere le regioni e viceversa.

In aggiunta alle tecniche sopra menzionate esistono numerose altre algoritmi di segmentazione, ciascuno dei quali è dotato di caratteristiche peculiari che lo rendono più o meno vantaggioso per una specifica applicazione. La scelta, quindi, può essere diversa e spesso viene fatta adattando opportunamente una delle tecniche esistenti o delineando *ex novo* un algoritmo *ad hoc*, in modo da soddisfare al meglio le esigenze richieste dall'applicazione specifica. In tal caso, è utile tener conto dei criteri che un buon metodo di segmentazione dovrebbe soddisfare [13], ovvero:

- le regioni devono essere il più possibile omogenee
- i confini delle regioni devono essere compatibili con le variazioni della misura di similarità adottata.
- aree percettivamente uniformi non dovrebbero essere divise in più parti. In particolare questo si applica a regioni con ombreggiatura graduale e a regioni con tessitura.
- piccoli dettagli, se ben definiti in forma e contrasto, non dovrebbero essere fusi con le regioni confinanti.

Tuttavia, valutare la bontà del risultato di una segmentazione non è cosa facile, essendo solitamente una questione soggettiva. Vari criteri sono stati proposti in letteratura per tale valutazione, il più semplice dei quali prevede il calcolo della percentuale di pixel

[1] delle tonalità di colore nel caso di immagini a colori.

segmentati erroneamente dal metodo utilizzato [15]. Misure più complesse sono rappresentate dal risultato del processo di classificazione complessivo ottenuto dopo la segmentazione [16], criteri di uniformità e contrasto delle regioni o dei bordi ottenuti [17], misure in merito alla forma ottenuta dalla segmentazione [18] o, ancora, il confronto tra i valori di alcune caratteristiche misurate su un'immagine idealmente segmentata e quella effettivamente ottenuta dal processo applicato [19; 20].

In realtà, il problema della segmentazione nelle immagini richiede l'emulazione della percezione psicologica e quindi non può avere una soluzione analitica, ma, al contrario, qualunque algoritmo matematico formulato a riguardo richiede l'utilizzo di una qualche euristica, basata sulla semantica o sulla descrizione della classe di immagini in considerazione [15]. In alcuni casi, è opportuno andare oltre l'uso di euristiche e introdurre della conoscenza *a priori* riguardo l'immagine, in una tale eventualità il processo di segmentazione procede di pari passo a quello di riconoscimento e comprensione dell'immagine stessa.

4.1. Metodo *Thresholding*.

La segmentazione per *sogliatura dell'istogramma* (*Histogram Thresholding*), detta anche *binarizzazione*, è uno dei metodi maggiormente conosciuti ed utilizzati, grazie alla semplicità concettuale e computazionale che lo caratterizza e al fatto che permetta sempre di individuare regioni chiuse con contorni connessi.

Si tratta di una tecnica che sfrutta le proprietà di similitudine locale dei pixel dell'immagine con l'obiettivo di determinare un particolare livello di grigio *T*, la *soglia di binarizzazione*, appunto, che permetta, sulla base di semplici controlli sulle tonalità di grigio, la suddivisione dell'immagine in regioni significative. Più precisamente, considerata una coppia di valori di grigio (g_0,g_1), una volta stabilito il valore della soglia *T*, il risultato della segmentazione per binarizzazione di un'immagine $I(x,y)$ è un'immagine binaria $I_b(x,y)$ tale che sia soddisfatta la condizione (5):

$$I_b(x,y) = \begin{cases} g_0 & \text{se } I(x,y) < T \\ g_1 & \text{se } I(x,y) \geq T \end{cases} \qquad (5)$$

Il caso più semplice di applicazione di tale metodo è quello in cui l'immagine considerata contenga un unico oggetto avente intensità omogenea, diversa da quella dello sfondo. In tal caso è facile determinare la soglia *T* in modo immediato dall'analisi dell'andamento del cosiddetto *istogramma dei toni di grigio* dell'immagine, una funzione discreta H[2] dei valori $g_k \in [0, G_{max} - 1]$ nel modo seguente:

$$H(g_k) = n_k/(n*m) \qquad (6)$$

dove n_k è il numero di pixel che hanno valore g_k, n ed m sono le dimensioni dell'immagine, da cui $n*m$ è il numero totale di pixel della stessa.

In particolare, nel caso suddetto, l'istogramma presenta un andamento bimodale, come quello mostrato in Figura 2, per cui è facile determinare la soglia come il punto di minimo tra i due picchi corrispondenti allo sfondo e all'oggetto contenuto nell'immagine. Questo approccio basato sull'andamento dell'istogramma è detto *metodo della moda* e la soglia determinata nel modo descritto è detta *globale* [13]. Si tratta del metodo più semplice, che assicura la probabilità minima di classificare i punti dell'oggetto come sfondo e viceversa, in quanto il numero di pixel con valori che cadono nella "valle" tra due picchi è ridotto.

Il metodo può essere esteso al caso in cui l'istogramma dell'immagine contenga un numero maggiore di picchi, selezionando, in questa eventualità, un insieme di soglie, ciascuna corrispondente ad una valle tra due picchi consecutivi. Un importante inconveniente presentato, tuttavia, da tale metodo è la perdita delle informazioni spaziali dell'immagine, dovuta alla mancata

[2] L'istogramma rappresenta una misura *a posteriori* della distribuzione dei livelli di grigio all'interno dell'immagine, utile perché fornisce una descrizione globale della cosiddetta apparenza dell'immagine.

considerazione delle relazioni di vicinanza dei pixel.

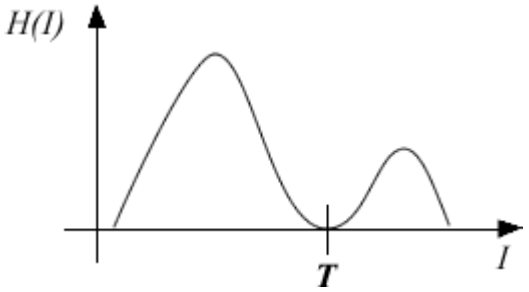

Figura 2. *Istogramma con andamento bimodale.*

D'altro canto, lo stesso metodo non può essere applicato nel caso in cui l'istogramma presenti delle ampie e piatte valli o una molteplicità caotica di picchi. Inoltre, l'analisi del solo istogramma non tiene conto delle informazioni spaziali dell'immagine, trascurando le condizioni di vicinanza dei pixel. In tal caso, è necessario ricorrere a metodi diversi per la determinazione della soglia di binarizzazione. Soluzioni a questo problema sono presenti numerose in letteratura, sulle quali nutrite panoramiche sono quelle di Fu *et al.* [13], Sahoo *et al*. [18] e Sankur *et al*. [21].

4.2. Metodi *Region growing* e *Split & Merge*.

Si tratta di due metodi di segmentazione basati su criteri di similarità per l'estrazione di regioni omogenee, che permettono di considerare sia le informazioni sui valori di grigio sia dettagli spaziali dell'immagine. Un'ulteriore proprietà comune è nel modo incrementale di procedere, richiedendo ad ogni passo il soddisfacimento di un certo criterio di omogeneità Λ.

Il metodo di *accrescimento delle regioni* è basato sull'espansione di queste mediante la progressiva fusione di pixel o regioni adiacenti [22; 23]. Anche in questo caso sono previste due fasi: la fase di inizializzazione e la fase di accrescimento.

La prima fase consiste nello scegliere un adeguato insieme iniziale di regioni elementari (al limite un pixel) detti *semi* (*seed*), dalle quali ha inizio la fase di accrescimento. Le regioni scelte come semi sono caratterizzate da un valore alto del criterio di omogeneità Λ usato. Durante la fase di accrescimento, in maniera iterativa, le coppie di regioni contigue R_i e R_j, che soddisfino come unione la condizione di omogeneità, $\Lambda(R_i \cup R_j)$, vengono aggregate. La scelta della funzione di omogeneità è spesso fatta *ad hoc*, ovvero sulla base delle caratteristiche spaziali e statistiche delle immagini trattate. Può essere rappresentata da una semplice funzione matematica oppure da una complessa combinazione di funzioni di costo e regole euristiche. Alcune regole proposte consistono nell'unire le regioni per le quali la differenza del livello medio di grigio disti meno di una determinata soglia, oppure nel considerare le regioni con contrasto inferiore ad una certa soglia, contare il numero di pixel ω del contorno tra queste e unire le regioni per cui valga

$$\frac{\omega}{P_m} > \alpha$$

dove P_m è la lunghezza del perimetro della regione più piccola e α è un valore di soglia, o, ancora, nell'utilizzare lo stesso approccio appena descritto sostituendo al perimetro la lunghezza del confine tra le due regioni.

I vantaggi presentati da questo metodo sono rappresentati da una maggiore qualità rispetto a quello per sogliatura e allo *split&merge* e dalla possibilità di adattamento a qualsiasi tipo di immagine, con l'uso di opportune regole di omogeneità. Gli svantaggi sono, invece, un elevato onere computazionale e la forte sensibilità alla fase di inizializzazione.

Le tecniche basate sulla suddivisione, *splitting*, e aggregazione, *merging* eseguono la segmentazione precedendo in due fasi successive [4; 24]:

1. fase di *split*: viene applicato un processo *top-down* che suddivide l'intera immagine in regioni elementari;
2. fase di *merge*: viene eseguito un processo *bottom-up* che permette di raggruppare le regioni elementari in regioni più complesse.

La prima fase consiste in una procedura ricorsiva che crea e scandisce una struttura ad albero quaternario (*quadtree*): partendo da un insieme di regioni quadrate di dimensione $(2n)^2$, corrispondenti a blocchi quadrati dell'immagine, si procede valutando per ogni blocco R_i la condizione di omogeneità $\Lambda(R_i)$. Nel caso in cui la condizione non sia verificata e n sia maggiore di uno, il blocco corrente viene suddiviso in quattro sotto-blocchi, di dimensioni n^2, ai quali viene applicata ricorsivamente la stessa procedura (Figura 3).

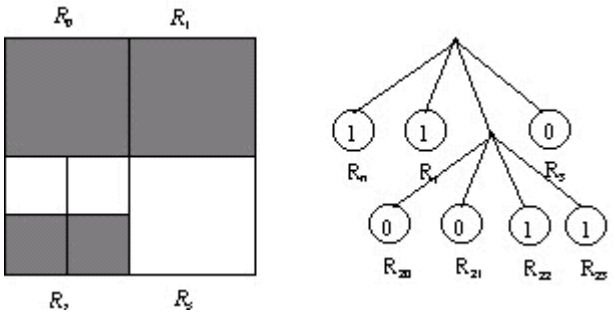

Figura 3. *Esempio di due iterazioni della fase di split: i quadranti in grigio sono quelli che non soddisfano il criterio di omogeneità e andranno, quindi, suddivisi. In particolare, viene mostrato un ulteriore passo ricorsivo sul quadrante R_2, suddiviso in 4 nodi figli, costruendo 4 quadranti ulteriori*

La fase di *split* ha lo svantaggio di perdere di vista l'omogeneità tra nodi appartenenti a diversi livelli di risoluzione oppure a diversi rami, dando origine ad un insieme molto numeroso di regioni piccole. La fase successiva, permette di correggere questa "sovra-segmentazione" effettuando una progressiva fusione delle regioni confinanti che rispettino il criterio di omogeneità. Varie misure sono state proposte per questo criterio, ne sono esempi la differenza tra il pixel di intensità massima e quello di intensità minima, la varianza del quadrante o altre misure statistiche più complesse.

Il metodo descritto presenta, tuttavia, lo svantaggio di produrre ROI troppo quadrate, quindi spigolose, inconveniente che potrebbe essere eliminato aumentando la risoluzione spaziale massima, ovvero la dimensione minima dei blocchi, ma ciò aumenterebbe ulteriormente l'onere computazionale, che si presenta già discretamente pesante.

4.3. Metodo *Edge Detection*.

Il metodo di segmentazione basato sull'individuazione dei bordi (*edge*) delle regioni da estrarre sfrutta le variazioni di intensità dell'immagine, ovvero si basa su un criterio di discontinuità tra pixel adiacenti. Il procedimento sfrutta degli opportuni operatori che trasformano le immagini originali in immagini binarie, nelle quali i pixel a valore non nullo identificano i punti dell'immagine originale con grandi variazioni nei toni di grigio. Si tratta di operatori che eseguono una differenziazione spaziale dell'immagine seguita da un'operazione di sogliatura per determinare quali punti siano da considerarsi effettivamente come di contorno. E' utile, pertanto, introdurre i concetti di derivate spaziali dell'immagine, definite in orizzontale e in verticale, rispettivamente, dalle espressioni (7) e (8):

$$d_x = \frac{\partial I(x,y)}{\partial x} \tag{7}$$

$$d_y = \frac{\partial I(x,y)}{\partial y} \tag{8}$$

Vari operatori sono stati proposti per l'impiego di queste informazioni spaziali, classificabili in operatori di primo ordine o di secondo in base all'ordine della derivata che utilizzano. Alcuni di questi vengono riportati di seguito.

E' utile osservare subito, però, che la segmentazione basata sull'individuazione dei bordi presenta come inconveniente la necessità di dover ricostruire i contorni delle regioni delle quali siano stati, per l'appunto, estratti estratto i bordi e ciò rappresenta un onere non sempre affrontabile.

Operatori gradiente.

Il gradiente di un'immagine è definito dall'espressione seguente:

$$\nabla I(x,y) = \left[\frac{\delta I(x,y)}{\delta x}, \frac{\delta I(x,y)}{\delta y} \right]^t \tag{9}$$

ed è possibile valutarne il modulo G considerando i due gradienti, orizzontale G_R e verticale G_C, secondo la seguente espressione [4]:

$$G[I(x,y)] \approx |G_R(x,y)| + |G_C(x,y)| \quad (10)$$

Usare, quindi, il gradiente per determinare i bordi di una regione equivale all'applicazione della seguente condizione, avendo selezionato un valore L da usare per considerare gli elementi di bordo:

$$G[I(x,y)] = \begin{cases} 1 \ \textbf{\textit{se}} \ |G_R(x,y)| + |G_C(x,y)| > L \\ 0 \ \textbf{\textit{altrimenti}} \end{cases} \quad (11)$$

La scelta della soglia L è estremamente importante, se è troppo bassa vengono considerati punti di bordo anche quelli che in realtà non lo sono, viceversa, se è troppo elevata si rischia di non rilevare alcuni bordi. Il problema è trovare un compromesso che limiti il numero degli errori.
Variazioni della tecnica di base appena enunciata utilizzano metodi diversi per il calcolo del gradiente, approssimandone il valore attraverso l'applicazione di cosiddette *maschere* o *operatori di gradiente* o, ancora, *operatori di bordo*, rappresentate da una coppia matrici di 2x2 o 3×3 pixel, una per ciascuna dimensione. I casi più semplici sono ottenuti con le maschere mostrate in Figura 4, il secondo dei quali è detto gradiente di *Roberts*, e corrispondono alle seguenti approssimazioni:

$$G[I(x,y)] \approx |I(x,y) - I(x+1,y)| + |I(x,y) - I(x,y+1)| \quad (12)$$

$$G[f(x,y)] \approx |f(x,y) - f(x+1,y+1)| + |f(x+1,y) - f(x,y+1)| \quad (13)$$

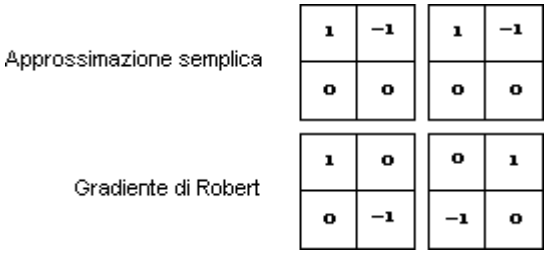

Figura 4. *Operatori di contorno 2x2.*

Nel caso di maschere 3×3, gli operatori sono più complessi e vengono definiti usando la notazione mostrata in Figura 5.

A_0	A_1	A_2
A_7	$I(x,y)$	A_3
A_6	A_5	A_4

Figura 5. *Convenzione usata nella numerazione degli elementi di un operatore di bordo.*

L'applicazione di una maschera consiste nel calcolo locale delle due componenti del gradiente secondo le espressioni seguenti:

$$G_R(x,y) = \frac{1}{k+2}\left[\begin{matrix}(A_2 + kA_3 + A_4) - \\ (A_0 + kA_7 + A_6)\end{matrix}\right] \quad (14)$$

$$G_R(x,y) = \frac{1}{k+2}\left[\begin{matrix}(A_0 + kA_1 + A_2) - \\ (A_6 + kA_5 + A_4)\end{matrix}\right] \quad (15)$$

Gli operatori 3x3 maggiormente utilizzati sono quelli di *Prewitt*, di *Sobel* e di *Frei-Chen* detto anche *isotropo*, mostrati in Figura 6, che utilizzano per il parametro *k*, usato per pesare il contributo dei pixel, i valore 1, 2 e $\sqrt{2}$, rispettivamente.

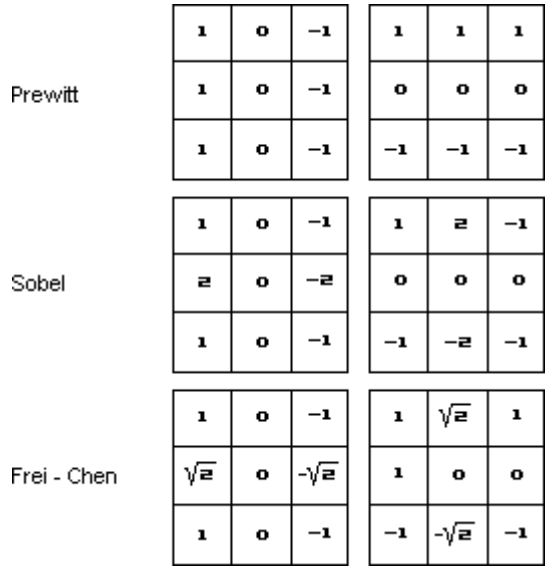

Figura 6. *I tre operatori di contorno maggiormente utilizzati.*

5. Caratteristiche e descrittori.

Dopo aver segmentato l'immagine in regioni distinte in modo da estrarre le strutture di interesse in essa contenute, è necessario selezionare un modello di rappresentazione di tali strutture. Generalmente detto modello è costituito da un insieme di *caratteristiche* o *descrittori* delle entità del problema affrontato. In particolare, con il termine *feature* può essere indicata qualsiasi tipo di misura estraibile da un'osservazione, ma è buona norma distinguere le generiche misurazioni dalle caratteristiche: mentre le prime sono semplici osservazione, le seconde sono propriamente costruite con opportuni operatori ed in tal senso, si parla di *estrazione delle caratteristiche*. L'obiettivo di questa operazione è il conseguimento di un maggiore potere discriminante delle informazioni sfruttate e, al contempo, una diminuzione dei dati utilizzati dal classificatore. Le *feature* possono essere numeriche o simboliche, ma costituiscono, generalmente, entità ad alto livello, con l'intenzione di concentrare l'informazione utile in un insieme ristretto di dati. C'è tuttavia da tener conto dello sforzo computazionale e teorico richiesto dalla riduzione dei dati e dell'eventuale aumento della probabilità di perdere informazioni. In tal senso, la corretta selezione delle variabili utilizzate è di fondamentale importanza, tanto più che esiste un ulteriore problema di cui tener conto nella fase di opportuna riduzione della dimensionalità: il cosiddetto *curse of dimensionality*, definito come il fenomeno per cui l'aumento della dimensione dello spazio delle caratteristiche comporti inizialmente un miglioramento dell'accuratezza ma porti rapidamente ad una dispersione ovvero radezza dei *pattern*, quindi ad una scorretta rappresentazione delle funzioni di densità dei dati e, di conseguenza, ad un peggioramento delle prestazioni del sistema. Il controllo di questo fenomeno richiede l'adozione di metodi opportunamente definiti, come verrà illustrato nel seguito di questa sezione.

In sostanza, le proprietà di cui devono godere le *feature* selezionate sono [25]:

- *potere discriminante*: oltre a descrivere correttamente il *pattern* in esame, le *feature* devono assicurare una sufficiente discriminazione delle varie classi di appartenenza del problema affrontato;
- *complessità computazionale ridotta*: è importante che il sistema impieghi un tempo ragionevole per calcolare l'insieme di variabili scelte. La ragionevolezza fa riferimento, in genere, ad un buon rapporto tra tempo e prestazione.

Un'altra peculiarità spesso desiderabile, soprattutto nei sistemi di *Object Recognition*, è l'*invarianza*, ovvero la capacità di un caratteristica di rappresentare l'informazione contenuta in un'immagine, indipendentemente dalla posizione che l'oggetto assume in essa. Infine, nella realizzazione di un sistema che sfrutti l'insieme di caratteristiche selezionate, un punto importante di cui tener conto è rappresentato dalla *flessibilità* del sistema stesso, ovvero dalla possibilità di adattarlo ai cambiamenti del problema affrontato. Di tale esigenza è stato tenuto particolare conto nella definizione dell'architettura neurale usata nell'approccio definito in questo lavoro di tesi.

Solitamente, tuttavia, la scelta viene fatta sulla base delle peculiarità del problema affrontato: dopo la segmentazione, è possibile

estrarre una rappresentazione delle regioni o dei bordi o la rappresentazione simbolica, ad un livello più alto, della struttura considerata. Varie rappresentazioni di questo genere sono presenti in letteratura [3; 4; 5; 26; 27] e, anche in questo caso, si tratta di tecniche *problem-oriented*, giacché ciascuna di esse presenta delle peculiarità che la rendono adatta ad un particolare problema.

In generale, le *feature* che possono essere usate per la descrizione del contenuto delle immagini possono essere raggruppate in due categorie principali:

- caratteristiche *morfologiche*: un insieme di grandezze che descrivono la forma (*shape*) delle strutture contenute nell'immagine e, quindi, richiedono la determinazione dei bordi o contorni di queste;
- caratteristiche *densitometriche* (o di ampiezza): un insieme di variabili che tengono conto delle proprietà di intensità, luminosità e colore delle immagini e delle strutture in esse contenute.

Un'altra classe individuabile è, poi, quella costituita dalle caratteristiche della *tessitura* (*texture*) dell'immagine, definita in termini di raggruppamenti di pixel che con la loro disposizione spaziale per il loro motivo uniforme non sono sufficientemente rappresentati dal solo attributo colore dell'immagine.

Caratteristiche Morfologiche.

Le regioni estratte attraverso il processo di segmentazione possono essere valutate in termini di grandezze che ne descrivano la forma, la dimensione, la posizione e l'orientamento. Tali caratteristiche vengono in tal senso indicate anche con il termine di *descrittori* (*shape descriptor*) e richiedono la determinazione dei contorni delle regioni da valutare.

La ricerca e la rappresentazione dei bordi è uno dei problemi più complessi connessi all'elaborazione e analisi delle immagini. Il procedimento deve essere necessariamente preceduto da una fase di segmentazione, che determini quali aree dell'immagine possono contenere i contorni di oggetti. Partendo da ciò, viene effettuata un'analisi di dette le aree, per scartare i punti che presumibilmente non appartengono a qualche contorno. Infine, si procede ad un'analisi più dettagliata per estrarre dall'immagine una rappresentazione dei pixel rimasti, sfruttata dagli operatori per il calcolo dei descrittori.

Uno dei più semplici descrittori di forma è la lunghezza del contorno della regione, ovvero il perimetro della stessa, e può essere ottenuta in maniera grossolana contando i pixel lungo il contorno. In alternativa, è possibile utilizzare la catena di codici (*chain code*) usata per rappresentare un contorno considerandolo una sequenza di segmenti connessi di lunghezza costante. Tipicamente, tale rappresentazione è basata sulla connettività. La catena viene, quindi, generata seguendo il contorno dell'oggetto, generalmente in senso orario, ed assegnando una direzione ai segmenti che congiungono ogni coppia di pixel. Utilizzando tale rappresentazione, la lunghezza è data esattamente dal numero di componenti verticali e orizzontali più il numero di componenti diagonali moltiplicato $\sqrt{2}$.

Un'altra caratteristica geometrica è l'area, calcolabile grossolanamente come il numero dei pixel contenute nella regione considerata. Un metodo più sofisticato consiste nel considerare la regione come un poligono e quindi calcolarne l'area sommando le aree di piccoli triangoli determinati a partire da un punto generico (x_0,y_0) scelto all'interno della regione (Figura 7):

$$dA = x_2y_1 - \frac{1}{2}x_1y_1 - \frac{1}{2}x_2y_2 - \frac{1}{2}(x_2-x_1)(y_1-y_2) \quad (16)$$

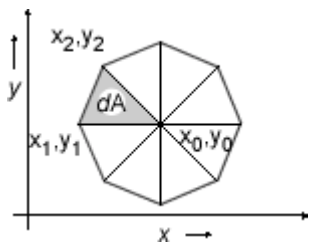

Figura 7. *Illustrazione del metodo di calcolo dell'area di un poligono.*

Dalla somma delle aree dei piccoli segmenti è possibile ottenere l'area totale del poligono:

$$A = \frac{1}{2}\sum_{i=0}^{r-1}(x_i y_{i+1} - x_{i+1} y_i) \qquad (17)$$

Ampiezza e lunghezza sono altre informazioni morfologiche che possono essere determinate valutando il massimo e il minimo delle coordinate dei punti della regione. Un'alternativa a questo metodo prevede l'individuazione, attraverso una procedura iterativa [5], del cosiddetto *rettangolo di inclusione minimo* (*Minimum Enclosing Rectangle*, MER), le dimensioni del quale corrispondono a quelle della regione considerata. Il MER permette di calcolare anche un altro descrittore di forma, la cosiddetto *rettangolarità* della ROI, definita dall'espressione seguente:

$$Rec = \frac{A}{A_{MER}} \qquad (18)$$

dove A è l'area della regione e A_{MER} quella del MER. Tale grandezza assume valore massimo pari ad 1 nel caso di regioni rettangolari.
Un descrittore dello stesso genere del precedente è il cosiddetto *shape factor* o circolarità della ROI e consiste nel seguente rapporto:

$$SF = \frac{P^2}{A} \qquad (19)$$

E' possibile normalizzare questo rapporto per il valore dello *shape fact*or di un cerchio, ovvero 4π, ottenendo, in tal modo, un indice della circolarità della regione considerata.
Altre caratteristiche morfologiche possono essere calcolate attraverso i cosiddetti *momenti*, variabili tipiche della statistica il cui uso è stato esteso all'elaborazione di immagini, giacché queste possono essere interpretate come una funzione di distribuzione in un piano. Formalmente, la definizione generale di momento cartesiano di ordine $p + q$ di una funzione distribuzione $f(x,y)$ è data dalla seguente:

$$m_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dxdy \qquad (20)$$

che nel caso discreto considerando un'immagine *I nxm* diventa:

$$m_{p,q} \overset{def}{=} \sum_{y=0}^{m}\sum_{x=i}^{n} x^p y^q I(x,y) \qquad (21)$$

L'attrattiva nell'uso dei momenti per la rappresentazione degli oggetti è dovuta al teorema di unicità di Papoulis. In esso si asserisce che, per una funzione $f(x,y)$ a tratti continua, con elementi diversi da zero solo per una regione finita del piano (x,y), esistono i momenti di tutti gli ordini. Per caratterizzare tutte le informazioni contenute in un'immagine sono richiesti potenzialmente un numero infinito di momenti, l'idea, invece, nella pratica è quella di selezionarne un insieme ridotto ma significativo di in grado di descrivere con sufficiente accuratezza gli oggetti di interesse. Il più comune è il momento di ordine zero dato dalla seguente espressione:

$$m_{0,0} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x,y)\,dxdy \qquad (22)$$

il quale rappresenta la massa o volume totale della funzione distribuzione. Se l'immagine è stata binarizzata, il momento di ordine zero rappresenta l'area totale dell'oggetto in essa rappresentato. I due momenti primi m_{10}, m_{01} permettono di localizzare il centro di massa o centroide $C_{de} = (c_x, c_y)$ della ROI, individuato dai seguenti due rapporti:

$$c_x = \frac{m_{1,0}}{m_{0,0}} \quad c_y = \frac{m_{0,1}}{m_{0,0}} \qquad (23)$$

da distinguersi dal baricentro, nel caso in cui l'applicazione non riguardi un'immagine binaria. Se l'immagine era stata precedentemente segmentata, allora il centro di massa è il punto in cui si potrebbe concentrare tutta la massa dell'oggetto senza modificare i momenti del prim'ordine. Risulta

utile, quindi, il suo utilizzo come punto di riferimento per descrivere la posizione dell'oggetto nell'immagine. Se un'immagine $I(x,y)$ è traslata in modo tale che il centro di massa venga a coincidere con l'origine (0,0), allora i momenti calcolati con la
nuova immagine sono detti momenti centrali e sono riferiti con la notazione μ_{pq}. Formalmente, vale la seguente definizione:

$$\mu_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q I(x-c_x, y-c_y)dxdy \quad (24)$$

L'importanza dei momenti centrali risiede nella loro invarianza alle traslazioni dell'oggetto nell'immagine. I momenti del secondo ordine μ_{20}, μ_{11}, μ_{02} sono conosciuti come *momenti d'inerzia* e permettono di determinare alcune utili *feature* come gli *Assi Principali* e la *best-fit elipse*, ovvero l'ellisse che ha gli stessi assi principali della ROI. I primi possono essere descritti come la coppia di assi lungo i quali si ha il minimo e il massimo momento del second'ordine. L'orientamento degli assi principali è dato dalla seguente espressione:

$$\phi = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right) \quad (25)$$

in cui ϕ rappresenta l'angolo di inclinazione dell'asse principale alla asse delle ascisse, compreso nell'intervallo $[-\pi/4; \pi/4]$.
La *best-fit elipse* è un disco ellittico costante con massa e momenti del second'ordine uguali all'originale. La lunghezza dei semiassi minore L_{min} e maggiore L_{max} di tale ellisse sono dati dall'espressioni seguenti:

$$L_{\max} = \left(\frac{2\left[\mu_{2,0}+\mu_{0,2}+\sqrt{(\mu_{2,0}-\mu_{0,2})^2+4\mu_{1,1}^2}\right]}{\mu_{0,0}}\right)^{1/2} \quad (26)$$

$$L_{\min} = \left(\frac{2\left[\mu_{2,0}+\mu_{0,2}-\sqrt{(\mu_{2,0}-\mu_{0,2})^2+4\mu_{1,1}^2}\right]}{\mu_{0,0}}\right)^{1/2} \quad (27)$$

I momenti di ordine maggiore permettono di definire caratteristiche dell'immagine che riguardano la distribuzione dei valori di grigio all'interno della stessa e pertanto appartengono alla classe di *feature* trattate nel paragrafo successivo.

Caratteristiche densitometriche.

Si tratta di un insieme di variabili definite interpretando l'immagine come una funzione di distribuzione delle tonalità di grigio. Il loro valore può essere determinato facilmente sfruttando l'istogramma dell'immagine, posto $g \in [0, G_{max}\text{-}1]$ e p_g il valore corrispondente nell'istogramma calcolato secondo la (6), ovvero come rapporto tra numero di pixel con tonalità g e il numero totale di pixel dell'immagine.

Le caratteristiche densitometriche maggiormente utilizzate sono la media μ, corrispondente al momento di ordine primo e la varianza σ, momento di secondo ordine:

$$\mu = \sum_{g=1}^{G_{\max}-1} g \cdot p_g = \frac{1}{A}\sum_{(x,y)\in ROI} I(x,y) \quad (28)$$

$$\sigma^2 = \sum_{g=1}^{G_{\max}-1}(g-\mu)\cdot p_g = \frac{1}{A-1}\sum_{(x,y)\in ROI}(I(x,y)-\mu)^2 \quad (29)$$

Con i momenti centrali di ordine maggiore è possibile ottenere l'asimmetria o *skewness* γ, legata al momento di terzo ordine, che descrive il grado di deviazione della distribuzione dalla simmetria della media e la *kurtosis* β, ottenuta dal momento di quarto ordine, che misura quanto sia acuto il picco della distribuzione dell'istogramma. Le espressione per il calcolo di queste caratteristiche sono riportate di seguito:

$$\gamma = \frac{\mu_3}{\mu_2^{3/2}} \quad (30)$$

$$\beta = \frac{\mu_4}{\mu_2^2} - 3 \quad (31)$$

Altre due grandezze utili per la descrizione del contenuto di intensità dell'immagine sono

l'energia S_E e l'entropia S_P, definite rispettivamente dalle due espressioni che seguono:

$$S_E = \sum_{g=1}^{G_{\max}} p_g^2 \qquad (32)$$

$$S_T = -\sum_{g=1}^{G_{\max}-1} p_g \log_2 p_g \qquad (33)$$

6. Estrazione e selezione delle caratteristiche.

Come introdotto nella sezione precedente, la scelta delle caratteristiche usate per la rappresentazione dei *pattern* rappresenta un problema di fondamentale importanza.

Le prestazioni di un sistema di PR sono, infatti, strettamente legate al numero di *feature* selezionate, al numero di esempi utilizzati nella fase di addestramento (dimensione del *training set*) del classificatore e alla complessità di quest'ultimo. Uno dei problemi da affrontare in questo senso riguarda il fenomeno del *curse of dimensionality* (*maledizione della dimensionalità*) che comporta il cosiddetto *peaking phenomenon*, anche noto come fenomeno di Hughes, secondo cui l'aggiunta di nuove *feature* può portare ad una degradazione delle prestazioni del sistema nel caso in cui l'insieme di training sia limitato [10; 28; 29]. Tutti i sistemi di classificazione comunemente utilizzati, incluse quelli basati su architetture neurali, possono soffrire di questo fenomeno e, purtroppo, non è possibile stabilire un'esatta misura di correlazione tra la probabilità di errore nella classificazione, la dimensione del training set, il numero di *feature* e il numero di parametri liberi usati dal sistema [30]. Un possibile metodo per affrontare il problema è quello di usare un insieme di addestramento sufficientemente elevato: secondo alcuni tale dimensione dovrebbe risultare almeno dieci volte la dimensione dello spazio delle *feature* ed aumentare con l'aumento della complessità del sistema [29].

Un secondo approccio riguarda, invece, la dimensione dello spazio delle caratteristiche, ovvero la riduzione del numero di *feature* usate per la loro rappresentazione o descrizione. Una soluzione di questo genere, basata sulla considerazione del minor numero possibile di variabili considerate, risulta complessivamente auspicabile sia per una maggiore accuratezza della classificazione che per problemi di costo computazionale: un insieme limitato di *feature* salienti semplifica sia la rappresentazione dei *pattern* sia il sistema sviluppato su tale rappresentazione. D'altro canto, la riduzione deve essere effettuata in modo da non causare una diminuzione del potere discriminante delle caratteristiche e, quindi, un peggioramento delle prestazioni del sistema.

I metodi sviluppati per la messa in pratica di questa soluzione possono essere distinti in *estrazione* e *selezione* delle caratteristiche. Spesso le due denominazioni vengono usate in maniera intercambiabile in letteratura, ma si tratta, in realtà, di due approcci diversi: la selezione consiste nella scelta del sottoinsieme ottimale dell'insieme di *feature* iniziale, mentre l'estrazione consiste nella generazione di nuove caratteristiche attraverso trasformazioni o combinazioni di quelle iniziali.

Frequentemente, l'applicazione di un metodo di estrazione precede quella di un metodo di selezione: inizialmente vengono estratte le *feature* dall'insieme di osservazioni effettuate (usando ad esempio l'analisi delle componenti principali) e successivamente quelle che risultino avere un ridotto potere discriminante vengono eliminate attraverso un processo di selezione.

I metodi di selezione comportano una riduzione del costo computazionale, grazie al minor numero di feature da determinare, inoltre le caratteristiche selezionate mantengono il loro significato fisico originale e il risultato della selezione può fornire informazioni circa il processo di generazione dei pattern. D'altro canto, le caratteristiche ottenute da un processo di estrazione possono fornire un maggior grado di discriminazione rispetto a quello del sottoinsieme ottimale, ma ad esse non corrisponde un chiaro significato fisico.

6.1. Metodi di estrazione.

I metodi di estrazione delle caratteristiche (*feature extraction*), attraverso una trasformazione lineare o non lineare, determinano all'interno dello spazio originale delle *feature* di dimensione *d* un appropriato sottospazio di dimensionalità *m*, con ovviamente $m \leq d$.

Le trasformazioni lineari, come l'Analisi delle Componenti Principali (Principal Componente Analysis, PCA), l'Analisi delle Componenti Indipendenti (Indipendent Component Analysis, ICA), l'Analisi Discriminante Lineare (Linear Discriminant Analysis, LDA), sono stati ampiamente utilizzate nell'ambito del PR per l'estrazione delle feature e la riduzione della dimensionalità dei dati.

La PCA detta anche *trasformata di Hotelling* o *espansione di Karhunen-Loève* [31; 32] è indubbiamente il metodo più conosciuto ed applicato [27; 33; 34; 35; 36; 37]. Si tratta di una tecnica di analisi basata sulle proprietà statistiche dei *pattern*; proposta inizialmente per l'analisi di covarianza, è stata applicata a vari problemi, quali la trasformazione di variabili correlate in variabili scorrelate, la ricerca di una combinazione lineare con massima varianza, la riduzione dei dati, l'estrazione delle *feature*. L'applicazione della PCA consiste nella determinazione degli *m* autovalori massimi della matrice di covarianza, di dimensioni $d \times d$, degli *n pattern* considerati. La trasformazione lineare applicata è definita nel modo seguente:

$$Y = XH \qquad (34)$$

dove

X è la matrice $n \times d$ dei *pattern* considerati;

Y è la matrice $n \times m$ risultante dalla trasformazione;

H è la matrice $d \times m$ che realizza la trasformazione lineare ed è costituita dagli autovettori corrispondenti agli *m* autovalori massimi.

Gli autovettori relativi agli autovalori massimi vengono definiti *componenti principali* e rappresentano la direzione di massima variazione dei *pattern*, mentre gli autovalori corrispondenti rappresentano la varianza degli stessi. La trasformazione implementata è una semplice rotazione dello spazio di rappresentazione dei *pattern* che allinei gli assi lungo la direzione di massima varianza.

Selezionando gli autovettori con autovalori massimi, la PCA utilizza le *feature* maggiormente espressive, ortogonali tra loro, approssimando i dati attraverso un sottospazio lineare che soddisfi il criterio di riduzione dell'errore quadratico medio [38]. Tuttavia, la scelta delle componenti principali, pur assicurando perdita di informazione minima, non sempre assicura la considerazione delle *feature* con maggiore potere descrittivo. Un esempio di quanto appena affermato, per un semplice caso di spazio di rappresentazione bidimensionale e problema con due classi di separazione è riportato in Figura 8, nella quale è evidente che la componente principale ϕ_1, pur avendo il maggior potere descrittivo, assicuri un grado minore di discriminazione delle classi.

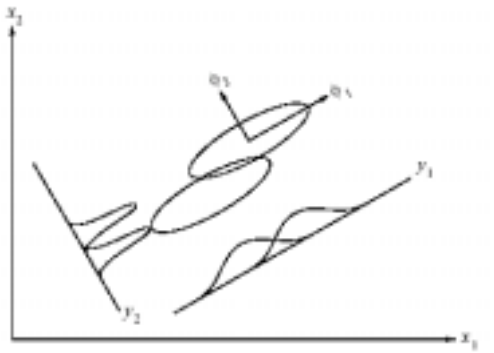

Figura 8. *Esempio di applicazione della PCA: la feature* ϕ_2 *ha un potere discriminante mentre la componente* ϕ_1 *ha un maggior potere descrittivo.*

Mentre la PCA è un metodo di estrazione non-supervisionato, nella *Analisi Discriminante Lineare* [38] si sfrutta l'informazione relativa alla classificazione da effettuare per selezionare le *feature* con maggior potere discriminante. La separazione all'interno delle classi viene enfatizzata sostituendo alla matrice di covarianza usata nella PCA una misura generale di separabilità come il criterio di Fisher, che consiste nella

considerazione degli autovalori della seguente matrice:

$$S_I^{-1} S_E$$

ottenuta come il prodotto delle due matrici:

S_I^{-1} inversa della matrice di dispersione *intra-classe*

S_E matrice di dispersione *extra-classi*;

con l'obiettivo di minimizzare la varianza intra-classi ed aumentare quella extra-classi. La trasformazione applicata in questo caso, quindi, determina una proiezione dei vettori dei *pattern* tale che quelli appartenenti alla stessa classe sia vicini, mentre le medie delle classi ottenute siano il più lontano possibile.

Nella versione di base la LDA assume che i dati abbiano distribuzioni Gaussiane, se ciò non avviene si rischia di perdere informazioni in merito alla struttura dei *pattern*.

Un metodo lineare maggiormente appropriato per distribuzioni non Gaussiane è quello dell'*Analisi delle Componenti Indipendenti* [39] sviluppato ed applicato inizialmente nell'ambito dell'elaborazione dei segnali [40; 41] ma esteso anche in altri campi, quali l'analisi delle immagini [42] e il riconoscimento di volti [42; 43]. Il metodo consiste nell'applicazione di una trasformazione lineare che minimizzi il grado di dipendenza delle nuove *feature* ottenute, formalmente si tratta della seguente espressione:

$$Y = XW \tag{35}$$

dove

X è la matrice $n \times d$ dei *pattern* considerati;

Y è la matrice $n \times m$ risultante dalla trasformazione;

W è la matrice $d \times m$ che realizza la trasformazione lineare tale che le colonne di Y siano il più indipendenti possibili secondo una qualche misura di indipendenza.

Purtroppo non esiste un'espressione esplicita per la trasformazione W e le tecniche proposte per l'applicazione dell'ICA consistono in algoritmi iterativi che applicano un qualche criterio di ricerca. Detto criterio dipende dalla misura di indipendenza selezionata che può essere *l'informazione mutuale*, l'*entropia* o la *negentropia* [44], grandezze che verranno discusse in maggiore dettaglio nella parte successiva.

6.2. Metodi di selezione.

La scelta delle *feature* usate nella rappresentazione dei *pattern* è quasi sempre basata sulle potere che queste hanno nel descrivere gli stessi, tuttavia, come evidenziato nell'esempio relativo alla PCA, il potere descrittivo di una *feature* non sempre corrisponde a quello di discriminazione della stessa (Figura 3.22). Questo problema può essere affrontato applicando un metodo per la *selezione delle caratteristiche*.

Formalmente, il problema della selezione delle *feature* può essere definito nel modo seguente [45]:

siano

l'insieme di cardinalità d *delle feature iniziali* F_i *che assumono valori* f_i,

m *il numero di feature desiderate;*

J *un criterio di selezione che indichi la bontà dell'insieme di feature selezionate*

l'obiettivo di un metodo di selezione è quello la determinazione di un sottoinsieme Γ *di* Φ*, di cardinalità* m*, tale da massimizzare* J(F), ovvero

$$\Gamma = \max_{Z \subseteq \Phi, |Z|=d} J(Z) \tag{36}$$

La scelta più semplice per J è

$$J = 1 - P_e \tag{37}$$

dove P_e rappresenta l'errore di classificazione. In tal caso, le procedure di selezione delle caratteristiche sono strettamente dipendenti dal tipo di classificatore utilizzato e dalla dimensione dell'insieme di addestramento di questo.

Ovviamente, il modo più immediato per la selezione consiste in una ricerca esaustiva, valutando del criterio J per tutti i possibili sottoinsiemi F, il che richiederebbe un calcolo combinatorio (vi sarebbero $\binom{d}{m}$ possibili casi) e, quindi, inapplicabile nella pratica. Per tale motivo sono stati proposti in letteratura vari metodi per effettuare questa ricerca.

Complessivamente, dunque, un metodo di selezione delle caratteristiche richiede la scelta di [8; 45; 46; 47]:

- una *strategia di ricerca* per selezionare le *feature* candidate;
- un *criterio di bontà* per valutare le *feature* candidate.

Per quanto riguarda il primo punto, varie sono le strategie proposte in letteratura e possono essere raggruppate in tre approcci principali:

1. algoritmi *esponenziali*;
2. algoritmi *sequenziali*;
3. algoritmi *randomizzati*.

Algoritmi esponezionali.
Valutano un numero si sottoinsiemi di caratte-
ristiche che crescono esponenzialmente con la dimensione dello spazio di ricerca. Ne sono esempi:

- la ricerca esaustiva, garantisce la soluzione ottima ma, come già riportato, è inapplicabile perché esponenziale;
- l'algoritmo *branch-and-bound*: introdotto da Fukunaga *et al.* [38] garantisce la soluzione ottima nel caso in cui sia valida la condizione di monotonia del criterio di bontà utilizzato, ovvero se vale la seguente condizione:

$$J(A \cup B) \geq J(A), \quad \forall A, B \subseteq \Phi \qquad (38)$$

  I sottoinsiemi di *feature* vengono trattati come nodi di un albero, ordinati secondo la relazione di inclusione e ottenuti eliminando una caratteristica per nodo fino ad un minimo di *m* elementi. Partendo dalla radice, costituita dall'intero insieme *F*, vengono rimosse man mano le *feature* secondo una strategia di ricerca in *profondità* (*depth-first*): quando si raggiunge una foglia il valore ottimo corrente (il *bound*) viene aggiornato, quindi, procedendo nella ricerca, i nodi che presentano un valore minore della funzione di bontà rispetto all'ottimo corrente non vengono esplorati, giacché le loro foglie non contengono una soluzione migliore. La richiesta della monotonia di *J* e la complessità esponenziale, nel caso pessimo, limitano l'applicazione di questo metodo nella pratica;
- la *beam search*: anche in questo caso [48] i sottoinsiemi di *feature* vengono trattati come nodi di un grafo, ma il criterio di esplorazione è il *best-first*. Ad ogni passo, vengono valutati i nodi singolarmente e i migliori vengono mantenuti in una coda o *fascio* (*beam*). Al passo successivo, i nuovi nodi vengono inseriti nel fascio nella posizione corretta in base al valore di bontà ad essi associato. Si tratta tuttavia di un metodo applicato molto raramente [48];

Algoritmi sequenziali.
Si tratta di metodi di tipo *greedy*, sub-ottimi, che procedono nella ricerca aggiungendo o eliminando sequenzialmente una o più *feature*. Presentano l'inconveniente di incorrere in minimi locali. In questa classe rientrano i seguenti metodi:

- la selezione sequenziale *in avanti* (*Sequential Forward Selection*, SFS): a partire dall'insieme vuoto, si seleziona, ad ogni passo, tra le *feature* candidate quella che assicura il valore del criterio di bontà maggiore quando combinata con le altre caratteristiche già selezionate. Si tratta di un metodo computazionalmente attraente, che funziona meglio nel caso in cui l'insieme ottimo sia di dimensioni ridotte, giacché nel caso in cui questo sia prossimo all'intero Φ il numero di casi esaminati è alto. Lo svantaggio principale che, invece, presenta è l'incapacità di eliminare le *feature* selezionate, anche se diventano irrilevanti dopo l'aggiunta delle nuove;
- la selezione sequenziale *all'indietro* (*Sequential Backward Selection*, SBS): a partire dall'intero insieme di *feature*, si elimina, ad ogni passo, la *feature* che assicura la minore riduzione del valore di bontà del nuovo insieme ottenuto (è possibile che la rimozione aumenti il valore di *J*, nel caso di criterio non monotono). Il metodo richiede un

maggiore sforzo computazionale rispetto al precedente e funziona al meglio nel caso in cui l'insieme ottimale sia di grandi dimensioni. Il principale svantaggio di questo metodo è l'impossibilità di rivalutare l'utilità di una *feature* una volta che estasia stata eliminata;

- l'algoritmo *Plus-l-Minus-r* (LRS): si tratta di una generalizzazione dei due metodi precedenti. Se *l* è maggiore di *r*, si parte dall'insieme vuoto, nel caso contrario si parte dall'intero Φ e ad ogni passo si aggiungono e rimuovono, rispettivamente, *l* ed *r* *feature* secondo gli stessi criteri di SFS ed SBF. Tale metodo cerca di compensare le limitazioni di SFS e SBF, offrendo una certa capacità di rivalutazione delle *feature* scartate, ma presenta come inconveniente l'assenza di un modo per stabilire i valori migliori di *l* ed *r*;
- l'algoritmo di *selezione in avanti o all'indietro fluttuante* (*Sequential Forward Floating Selection*, SFFS e *Sequential Backrward Floating Selection*, SBFS): si tratta di un'estensione del metodo LRS che non fissa i valori di *l* ed *r* [49]. SFFS parte dall'insieme vuoto e, dopo ogni passo di selezione in avanti, esegue un certo numero di passi all'indietro fintanto si abbia un aumento del valore del criterio di bontà, mentre SBFS procede esattamente nel modo opposto;

Algoritmi randomizzati.
Eseguono una ricerca casuale (*random*) per evitare i minimi locali. Esempi rappresentativi di questa classe sono:

- la generazione *random* con selezione sequenziale: introduce una certa casualità nei metodi SFS e SBS, generando ad ogni passo in maniera random un sottinsieme di *feature* al quale applicare uno di tali algoritmi;
- gli *algoritmi genetici*: introdotti nell'ambito della selezione delle *feature* da Siedlecki *et al.* [50], rappresentano il generico sottoinsieme di caratteristiche come una stringa binaria (*cromosoma*) di dimensione *d*, avente un 1 per le *feature* contenute nel sottoinsieme e uno 0 nel caso opposto. Viene, quindi, applicato il processo di evoluzione, attraverso le tipiche operazioni di selezione, mutazione e riproduzione, fino ad ottenere la popolazione ottimale.

Dei metodi sopra elencati, l'unico ottimo è il *branch-and-bound*, mentre tutti gli altri sono sub-ottimi. Alcuni studi [45; 51; 52] hanno confrontato tali metodi su vari casi applicativi, come la classificazione delle immagini radar e la diagnosi di un sistema, comparandone il tasso di errore e il tempo di esecuzione. La conclusione generale è che il metodo SFFS assicura prestazioni migliori, simili a quelle del metodo *branch-and-bound*, richiedendo una quantità inferiore di risorse computazionale.

Per quanto riguarda la selezione del criterio di bontà utilizzato per valutare le *feature* candidate, esistono due approcci diversi:

- i *filtri*: si tratta di funzioni che valutano i sottoinsiemi di *feature* sulla base del loro contenuto, della distanza tra le classi da riconoscere, delle dipendenze statistiche tra le variabili o di misure ricavate dalla Teoria dell'Informazione;
- i *wrapper* (letteralmente *involucri*): il criterio di bontà è rappresentato da un qualche algoritmo di classificazione che viene applicato ai dati rappresentati con il sottoinsieme di *feature* considerate, per valutarne l'accuratezza predittiva in termini di tasso di classificazioni corrette su un insieme di test.

Il primo approccio risulta più diffuso nella pratica in quanto non richiede la necessità dell'addestramento di un classificatore. I vantaggi offerti dai metodi basati sui *wrapper* sono maggiore accuratezza, misurata in termini di errore di classificazione inferiore grazie alla stretta interazione tra classificatore e insieme dei dati, e maggiore capacità di generalizzazione, dovuta al processo di *cross-validation* usato nella fase di selezione che

permette di evitare situazioni di *overfitting*[3]. Gli svantaggi sono, invece, la lentezza nell'esecuzione, dovuta alla necessità di addestramento del classificatore per ogni insieme di *feature* selezionato, e la mancanza di generalità della soluzione che risulta strettamente correlata al classificatore utilizzato. I vantaggi dei metodi basati sull'uso di filtri sono, in tal senso, esattamente opposti agli svantaggi dei metodi del primo tipo, ovvero velocità di esecuzione e generalità, assicurata dal fatto che la soluzione sia ottenuta indipendentemente dal classificatore e, quindi, risulti soddisfacente e valida per una famiglia di sistemi di classificazione. Uno svantaggio di questi metodi è la tendenza a selezionare insiemi di *feature* di dimensioni elevate, soprattutto nel caso in cui le misure utilizzate siano monotone.

In letteratura sono presenti metodi che adottano come filtri vari tipi di funzioni, quali:

- misure di distanza o di separabilità;
- misure di correlazione.

Le prime sono *metriche* che permettono di valutare la separabilità delle classi da riconoscere, ne sono esempi la distanza *Euclidea* o di *Mahalanobis* misurate tra le classi o gli autovalori della matrice $S_I^{-1}S_E$, secondo l'analisi discriminante;

Le altre sono rappresentate da misure, lineari o non lineari, che permettono di selezionare le *feature* che abbiano il maggior grado di correlazione e, quindi, potere predittivo delle classi e siano non correllate tra loro. Considerata la variabile di classificazione *C*, un esempio di misura lineare è di un sottoinsieme di caratteristiche è rappresentato dal coefficiente di correlazione definito nel modo seguente [8]:

$$\frac{\sum_{i=1}^{m}\rho_{ic}}{\sum_{i=1}^{m}\sum_{j=i+1}^{m}\rho_{ij}} \qquad (38)$$

dove

ρ_{iy} è il coefficiente di correlazione tra le *i*-esima *feature*, F_i, e l'etichetta *c* della classificazione *C*;

ρ_{ij} è il coefficiente di correlazione tra la *i*-esima e la *j*-esima *feature* dell'insieme considerato.

La semplice correlazione, tuttavia, permette di misurare solamente le dipendenze lineari, una misura più efficiente e significativa, in tal senso, è l'*informazione mutuale* o *reciproca* (*Mutual Information*, MI), una grandezza non lineare che prende origine dalla Teoria dell'Informazione [53].

Più precisamente, considerate due variabili casuali (v.c.) *X* e *Y*, l'informazione mutuale *MI*(*X*,*Y*) è definita dall'espressione seguente:

$$MI(X,Y) = H(X) - H(X \mid Y)^{4} \qquad (39)$$

dove *H* rappresenta l'*entropia* associata ad una v.c. e misura l'incertezza ad essa associata. Per una variabile continua, l'entropia è definita come

$$H(X) = -\int p(x)\log_2 p(x)dx \qquad (40)$$

mentre quella di una v.c. discreta, la precedente diventa:

$$H(X) = -\sum_{x} p(x)\log_2 p(x) \qquad (41)$$

in entrambi i casi $p(X)$ rappresenta la probabilità marginale della distribuzione della v.c. *X*. Come anticipato, l'entropia fornisce una misura dell'incertezza associata alla v.c.

[3] L'*overfitting* è un fenomeno per cui un classificatore adattivo si specializza eccessivamente sui dati di apprendimento, dimostrando una ridotta capacità di generalizzazione su dati estranei a tale insieme.

[4] Tale formula è equivalente alla seguente $MI(X,Y) = H(X) + H(Y) - H(X,Y)$, dove $H(X,Y)$ indica l'entropia congiunta delle due v.c., calcolata come

$$H(X,Y) = \sum_{x}\sum_{y} p(x,y)\log_2 p(x,y).$$

Vale, infatti, l'uguaglianza $H(X,Y) = H(Y) - H(X \mid Y)$.

X: nel caso in cui uno dei valori *x* sia quasi certo ($p(\bar{x})=1$, $p(x)=0\ \forall x\neq\bar{x}$), $H(X)$ assume valore nullo, se i valori di X sono tutti equiprobabili ($p(x)=1/|X|,\ \forall x$), l'entropia assume valore massimo, pari a $\log_2|X|$. Un'ulteriore definizione riguarda l'entropia *condizionata* o *relativa*, calcolabile nel modo seguente:

$$H(X \mid Y) = -\sum_{y} p(y) \sum_{x} p(x) \log_2 p(x \mid y) \qquad (42)$$

che stabilisce l'incertezza *a posteriori* di *X* dopo aver osservato *Y*.
Utilizzando, quindi, la (42) con la formula per il calcolo della probabilità condizionata, è possibile ricavare l'espressione per la definizione della MI in funzione delle probabilità marginali e congiunte delle due v.c. considerate, ovvero la (39) diventa:

$$MI(X,Y) = \sum_{x} \sum_{y} p(x,y) \log_2 \frac{p(x,y)}{p(x)\,p(y)} \qquad (43)$$

dove, per l'appunto, $p(x,y)$ rappresenta il valore della probabilità congiunta delle due variabili. Dalla (39) si deduce che la MI misuri quanto l'incertezza relativa alla v.c. *X* sia ridotta in seguito all'osservazione di *Y*: se le due variabili sono indipendenti, la loro informazione reciproca è nulla, giacché l'osservazione di *Y* non riduce l'incertezza su *X*, viceversa se *X* e *Y* coincidono, allora $MI(X,X) = H(X)$, da cui l'interpretazione dell'entropia come informazione associata alla variabile *X*. Questo dimostra, anche, che in realtà la MI non costituisce una metrica, poiché non soddisfa tutte le proprietà richieste in tal senso. Una proprietà che la caratterizza è invece la simmetria: $MI(X,Y)$ coincide con $MI(Y,X)$.
Poiché la MI non richiede nessuna assunzione riguardo la natura della relazione tra due variabili, la sua validità ha carattere generale e, in tal senso, viene essa viene spesso interpretata come una generalizzazione del coefficiente di correlazione lineare. Ciò avviene anche nell'ambito dell'analisi discriminante lineare, come introdotto in precedenza.

Il concetto di informazione mutuale può essere facilmente esteso al caso di più variabili casuali: applicando la regola della catena, l'informazione mutuale *congiunta* (*Joint Mutual Information*, JMI) tra un insieme di *N* v.c. X_i e una variabile *Y* è definita nel modo seguente:

$$JMI(X_1,..,X_N;Y) = \sum_{i=1}^{N} MI(X_i, Y \mid X_{i-1},..,X_1) \qquad (44)$$

Quest'ultima misura può essere utilizzata come criterio di bontà in un algoritmo per la selezione delle *feature*: in tal caso, il filtro applicato è rappresentato dal valore della JMI di un insieme di caratteristiche Γ costituito da *d feature* F_i, considerate come variabili casuali che assumono valori f_i nello spazio di rappresentazione dei *pattern*, e la variabile *C* che rappresenti la classificazione:

$$J(\Gamma) = JMI(F_1, F_2, \ldots, F_d; C) \qquad (45)$$

utilizzando per JMI l'espressione (44). La MI congiunta, infatti, fornisce una misura migliore di bontà rispetto alla MI calcolata semplicemente per una singola *feature* e la variabile di classificazione, in quanto tiene conto della totalità delle *feature* considerate nell'insieme corrente, mentre nel caso in cui si usasse solo la MI si potrebbe ottenere un insieme significativo ma ridondante.
Il calcolo di una simile misura richiede, tuttavia, la determinazione della probabilità congiunte dell'insieme di variabili Γ e la *C*, il che in spazi di dimensioni elevate è un problema mal-posto (*ill-posed*). Una valida soluzione a questa situazione è quella proposta da Battiti [54], nella quale viene adottata un'euristica per l'insieme di *feature* e la classificazione definita nel modo seguente:

$$J(\Gamma, C) = \sum_{i=1}^{d} MI(F_i, C) - \tau \sum_{i=1}^{d} \sum_{j=i+1}^{d} MI(F_i, F_j) \qquad (46)$$

ovvero prevede il calcolo della somma della MI tra ogni singola *feature* e la classificazione *C* alla quale viene sottratta la MI che caratterizza ciascuna delle *feature* selezionate in modo da pesare anche il grado

di correlazione tra esse e, quindi, escludere *feature* ridondanti. In particolare, Battiti propone un opportuno algoritmo per la selezione delle caratteristiche, basato su una strategia di ricerca di tipo SFS, che ad ogni passo valuta per ciascuna delle *feature* F_i candidate il seguente criterio:

$$J(F_i, C) = MI(F_i, C) - \tau \sum_{k=1}^{d} MI(F_i, F_k) \quad (47)$$

dove F_k sono le caratteristiche già inserite nell'insieme da selezionare. Il parametro τ pesa le due informazioni ed assume valori compresi nell'intervallo [0.5,1].
Un problema che comunque permane in questo approccio è la stima della probabilità congiunta delle due variabili correntemente considerate. In realtà il calcolo della MI costituisce un problema fondamentale nell'ambito dalla Teoria dell'Informazione. In letteratura è possibile trovare diverse soluzioni a tale problema proposte sia per la stima diretta della MI sia per il calcolo della densità congiunta [55; 56; 57], ma la tecnica maggiormente utilizzata [58] consiste nel cosiddetto approccio *diretto* [59; 60] che prevede il calcolo della seconda grandezza attraverso la suddivisione dello spazio di ciascuna variabile in un certo numero di intervalli l'uso dello spazio bidimensionale così discretizzato per calcolare i valori della densità. Il numero di segmenti (*bin*) può essere determinato in maniera indipendente dai dati disponibili [58] oppure utilizzando le caratteristiche di distribuzione di questi [60; 61].

Come misura per la selezione delle *feature* la MI è stata applicata a vari problemi di PR, quali, ad esempio, la diagnosi dell'embolia polmonare in immagini ottenute con la tecnica di scintigrafia nucleare [60], il riconoscimento del parlato [61; 62] e la classificazione di testi.

7. Metodi di riconoscimento.

Il processo di riconoscimento, come accennato, può essere realizzato secondo approcci distinti dei quali i quattro principali sono [1; 8; 30]:

- il *template matching*;
- l'approccio *statistico*;
- l'approccio *sintattico* o *strutturale*;
- l'approccio *neurale*.

Tali metodi non sono necessariamente indipendenti e, a volte, lo stesso metodo può avere diverse interpretazioni [30]. Inoltre, vi sono delle caratteristiche comuni alle varie metodologie elencate, come la presenza di una sorgente di pattern stocastica e l'obiettivo dell'individuazione di un *mapping* dallo spazio delle misure allo spazio dei significati, aderente alla realtà. In generale, quindi, qualunque sia il metodo utilizzato, esso si riconduce sempre all'approccio statistico e con esso se ne confrontano i risultati per poterne giudicare la qualità. Infine, sono stati fatti alcuni tentativi per l'unificazione di metodi diversi in sistemi che vengono definiti di conseguenza ibridi [63].
Di seguito viene riportata una breve descrizione dei primi tre approcci, per poi descrivere con maggiore livello di dettaglio quello neurale. Ne vengono, inoltre, riassunte le peculiarità nella Tabella I in riferimento alle soluzioni adottate da ciascuno di essi per le questioni cruciali dello sviluppo di un sistema di PR, ovvero il tipo di rappresentazione, la funzione di decisione ed il criterio di apprendimento.

Tabella I. *Quattro approcci al Pattern Recognition e rispettivi modelli descrittivi*

Approccio	**Rappresenta-zione**	**Funzione di decisione**	**Criterio tipico**
Template Matching	Campioni, pixel curve	Correlazione misure di distanza	Errore di classifica-zione
Statistico	Caratteristiche	Funzione discriminante	Errore di classifica-zione
Sintattico	Primitive	Regole e grammatiche	Errore di Accettabi-lità
Reti Neurali	Campioni, pixel, curve, caratteristiche	Algoritmo neurale	Errore di classifica-zione

Template Matching.
Il *Template Matching* è uno dei primi e concettualmente più semplici approcci proposti e consiste nello stabilire una misura di similarità tra due entità (punti, curve o forme) dello stesso tipo. Viene mantenuto un *template* o *prototipo* (di solito una forma bidimensionale) per il pattern da riconoscere, con il quale viene fatto il confronto per l'entità correntemente analizzata, considerando tutte le possibili pose (traslazione e rotazione) e riduzioni di scala. La misura di similarità utilizzata per detto confronto, spesso di correlazione, può essere ottimizzata sulla base dell'insieme di *training* disponibile e, solitamente, lo stesso *template* viene appreso dai dati. Il processo di *matching* è, in generale, molto costoso e ciò ha posto gravi limitazioni in passato all'applicazione di tale approccio nella pratica, alleviati attualmente dalla disponibilità di processori più potenti.
Il *template* considerato nel tecnica di base è rigido il che comporta degli inconvenienti nel caso di eventuali distorsioni dei pattern o, se si tratta di strutture contenute nelle immagini, cambiamento della visuale o, ancora, grande varianza dei pattern all'interno di una stessa classe. Questi svantaggi possono essere risolti utilizzando *template deformabili*, utili nei casi in cui non sia possibile modellare in maniera semplice e diretta le deformazioni dei pattern.
Esempi di applicazioni di questo metodo sono l'*object recognition* [64; 65] e il riconoscimento di caratteri manoscritti [66].

Approccio statistico.
Nell'approccio statistico, detto anche *decision theoretic*, le tecniche di decisione utilizzate per il riconoscimento sono intrinsecamente statistiche, in riferimento alle distribuzioni dei *pattern*, controllate da leggi probabilistiche. Le entità da riconoscere o classificare sono rappresentate da vettori di *caratteristiche* (*feature*) che costituiscono gli elementi dello spazio di rappresentazione, da cui deriva la denominazione di sistemi di riconoscimento di tipo *feature-based*, ovvero basati sulle caratteristiche. L'obiettivo nella selezione di tali grandezze è quello di avere i vettori corrispondenti a *pattern* tutti della stessa classe in zone compatte e disgiunte dalle altre. In tal senso, la bontà della rappresentazione è misurata in merito a quanto i *pattern* siano ben separati.
Dato, quindi, un insieme di entità, il procedimento consiste in due passi successivi: la stima delle funzioni di densità di distribuzione dei *pattern* nello spazio di rappresentazione e la suddivisione di tale spazio in regioni separate, delimitate dai cosiddetti *decision boundaries*, ciascuna delle quali corrisponda ad una delle classi presenti. In particolare, le funzioni di densità possono esser specificate direttamente, se note a priori oppure apprese durante la fase di addestramento. Nel primo caso, ci si riduce ad un problema statistico di *test dell'ipotesi*, mentre nel secondo si procede alla determinazione di una funzione discriminante secondo due possibili paradigmi: se si ha una certa conoscenza del problema che permetta di stabilire il tipo di distribuzione delle classi, è possibile utilizzare un algoritmo di classificazione parametrico per il quale la fase di apprendimento consiste nello stabilire i parametri associati alle distribuzioni stesse, in modo da avere il miglior *matching* possibile, il metodo più usato è quello basato sulla legge di Bayes; nel caso in cui ciò non sia possibile è necessario ricorrere ai cosiddetti classificatori *non-parametrici* , che stimano le densità direttamente dai dati di *training*. Il più conosciuto tra questi è il *K-nearest-Neighbor*, per la semplicità concettuale e le buone prestazioni assicurate.
L'approccio statistico è sicuramente tra quelli maggiormente utilizzatati e, come già riportato, costituisce il punto di confronto degli altri approcci possibili.

Approccio sintattico.
L'approccio sintattico (o strutturale) è indicato per i casi in cui siano coinvolti *pattern* molto complessi, tali da renderne poco rilevante la descrizione in termini di caratteristiche e maggiormente utile, al contrario, una visione prospettica come composizione gerarchica di elementi costituiti

da sotto-elementi via via più semplici (scendendo nella gerarchia di costruzione). La relazione tra gli elementi di base di questa gerarchia, chiamati *primitive* o *builidng blocks*, descrive il *pattern* nella sua interezza. In questo senso, viene fatta un'analogia tra una simile struttura dei *pattern* e la sintassi dei linguaggi: i *pattern* sono visti come frasi appartenenti ad un linguaggio e le primitive come lettere dell'alfabeto relativo. Le frasi vengano, poi, formate secondo la grammatica del linguaggio. Nell'ambito di una tale interpretazione, un'ampia collezione di *pattern* complessi può essere descritta da un numero ridotto di primitive e regole grammaticali. Tali regole vengono apprese dal sistema di riconoscimento durante la fase di addestramento, direttamente dagli esempi dell'insieme di *training*.
Da questo punto di vista, il paradigma sintattico è interessante in quanto, oltre alla classificazione, offre una descrizione di come i *pattern* siano costruiti a partire da elementi di base.
I possibili casi di applicazione riguardano i *pattern* che abbiano una struttura che possa essere catturata da regole, come i tracciati di ecocardiogramma, le immagini con tessitura[5] (*texture*) e l'analisi di contorni di forme [1]. L'implementazione dell'approccio sintattico, tuttavia, può causare dei problemi nella segmentazione di *pattern* rumorosi e nell'inferenza della grammatica.

8. Le Reti Neurali come metodi di riconoscimento.

Le *reti neurali* sono particolarmente diffuse nell'ambito del PR grazie alle proprietà funzionali che le caratterizzano rendendole particolarmente adatte a tale compito.

La principale differenza tra le ANN e gli altri approcci al PR è data dalla loro capacità di apprendere relazioni ingresso-uscita molto complesse, usando una procedura di apprendimento iterativa, che non richiede la determinazione di un modello sottostante [8; 9; 67].
Il successo delle reti neurali nei compiti di PR è dato dalla bassa dipendenza dalla conoscenza specifica del dominio, cosa che invece richiedono gli approcci basati su modelli o su regole, e dalla disponibilità di efficienti algoritmi di apprendimento.
I sistemi neurali, inoltre, costituiscono metodi non lineari per l'estrazione delle caratteristiche di descrizione dei *pattern*, eseguita da un strato intermedio di nodi che elabora una rappresentazione interna dei vettori di ingresso. Ciò rende i modelli connessionisti particolarmente adatti non solo alla classificazione, ma anche alla selezione ed estrazione delle *feature* di descrizione dei *pattern* [30; 68], dimostrandone ulteriormente la valenza applicativa nei compiti di PR.

Negli ultimi anni, è stato dimostrato che molti dei modelli neurali maggiormente conosciuti siano implicitamente equivalenti ai classici metodi statistici di PR [69; 70]. Nonostante tale similarità, le ANN si pongono nell'ambito del PR come una metodologia che offre vantaggi unici, come un approccio univoco all'estrazione delle *feature* e alla classificazione e procedure flessibili per la ricerca di una buona soluzione non-lineare.

Una rete neurale può essere definita come un sistema di elaborazione adattivo costituito da un insieme di unità elementari parallele, non lineari e altamente interconnesse. I modelli neurali sono raffigurabili come reti di grafi pesati e diretti, in cui i nodi sono i neuroni e gli archi sono le cosiddette *connessioni simpatiche*, che "trasportano" l'uscita del nodo antecedente verso l'ingresso di quello successivo. Le peculiarità che caratterizzano tali sistemi sono la capacità di apprendimento e generalizzazione, l'adattabilità, la robustezza ovvero tolleranza al rumore, l'elaborazione e la memorizzazione della conoscenza distribuite. Ciascuna unità, chiamata *neurone* o *processore* o, ancora, *nodo*, ha un piccolo potere computazionale, ma la combinazione di molti elementi siffatti consente di ottenere una notevole potenza di calcolo e uno strumento matematico di calcolo flessibile ed

[5] La tessitura di un'immagine corrisponde a piccoli raggruppamenti di pixel con motivi uniforme, come verrà specificato nella sezione successiva appositamente delineata per la caratterizzazione delle immagini (§3.2.3).

adattabile. Grazie al considerevole numero di parametri liberi che contengono (i pesi sinaptici), le reti neurali sono in grado di svolgere un ampio spettro di compiti [6]. Tanto più che le variazioni dei pesi necessarie affinché la rete apprenda e acquisisca conoscenza dall'esperienza è automatico e segue un processo di ottimizzazione matematica che richiede il minimo intervento da parte di un agente esterno. Ciò avviene durante il processo di addestramento della ANN, che segue lo schema tipico della messa a punto di un sistema di riconoscimento e classificazione, ovvero si articola in due fasi: allenamento e test. Nella prima fase, viene utilizzato l'insieme di addestramento per permettere il sistema di apprendere, quindi nella fase successiva si valuta il risultato così ottenuto su un insieme di test, indipendente dall'insieme usato in precedenza, verificando la validità della funzione di decisione raggiunta.

Formalmente, l'elemento di base di un ANN è il cosiddetto neurone artificiale [32; 71], una semplice unità di elaborazione, u_i, il cui funzionamento può essere schematizzato come mostra la Figura 9 e consiste nel calcolo del cosiddetto *ingresso netto*, *net*, come somma pesata dei valori ricevuti in ingresso con i pesi relativi alla quale viene sottratta una soglia θ, e nell'applicazione di una funzione *f*, detta *di attivazione*, a tale quantità.

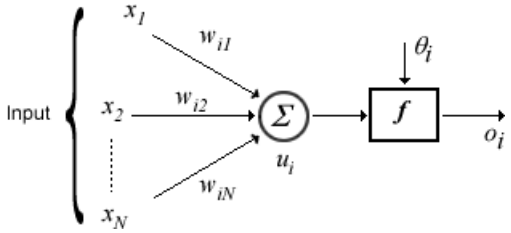

Figura 9. *Schematizzazione di un neurone artificiale.*

Il risultato rappresenta il grado di eccitazione del nodo al quale viene applicata *F* una funzione di uscita che determina il risultato finale dell'elaborazione. Nel caso più semplice, *f* è rappresentata dalla funzione a lineare, mentre funzioni più complesse generalmente utilizzate sono la funzione gradino, la logistica (o sigmoide) e la tangente iperbolica. La funzione di uscita è in genere la funzione identità. L'espressione risultante è, dunque, la seguente:

$$o_i = F_i(a_i) \qquad (48)$$

dove a_i è il risultato della funzione di applicazione calcolato come segue

$$a_i(t) = f_i(net_i) = f_i(\sum_{j=1}^{N} w_{ij} x_j - \theta_i) \quad (49)$$

indicando con *N* il numero delle connessioni in ingresso al nodo u_i. Senza perdere in generalità, è possibile considerare la soglia come un ingresso aggiuntivo lungo una connessione alla quale sia associato il valore -1.

L'insieme dei nodi interconnessi secondo la tipologia di un grafo orientato costituisce l'architettura della ANN, organizzata in modo strutturato, ovvero in un certo numero di strati o livelli, che procedono dall'ingresso verso l'uscita, passando per livelli intermedi che vengono definiti nascosti (*hidden*). In tal senso, le connessioni tra i vari nodi possono essere di tre tipi: in avanti (*feed-forward*) tra uno strato e il successivo, laterali all'interno dello stesso strato o all'indietro (*feedback*) da uno strato superiore ad uno inferiore (Figura 10). Le reti con connessioni solo del primo tipo vengono dette *feed-forward*, mentre quelle che prevedono anche connessioni del terzo tipo sono dette *ricorrenti*.

I valori dei pesi lungo le connessioni costituiscono la conoscenza incamerata dalla rete durante il processo di addestramento o allenamento, che può avvenire secondo due modalità principali, ovvero in modo supervisionato o non-supervisionato[6], attraverso un algoritmo appositamente definito per la topologia della rete in esame e corrispondente ad una particolare regola di apprendimento (*learning rule*).

[6] Ad esse va aggiunto l'apprendimento con rinforzo, come specificato in Appendice B, considerato generalmente secondario agli approcci .

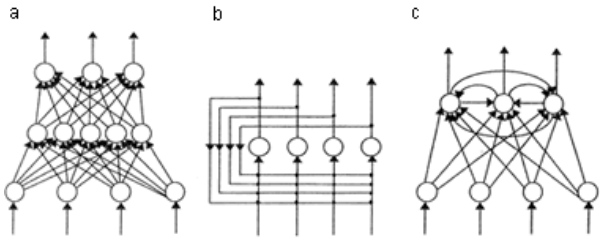

Figura 10. *Tre diversi tipi di connessioni e architetture: (a) rete feedforward con connessioni in avanti e con tre strati ingresso-nascosto-uscita, (b) rete monostrato con connessioni laterali (b) rete con connessioni ricorrenti.*

Tale distinzione fornisce un primo criterio per una tassonomia delle ANN usate nell'ambito del PR, al quale è possibile aggiungere la diversa architettura della rete. Una caratterizzazione secondo questi due criteri dei modelli neuronali maggiormente adottati [72] è riportata in Tabella II e riguarda le reti *feed-forward* multistrato (MLFF) con algoritmo di apprendimento *Back-Propagation* (EBP) [73], le reti *Radial Basis Function* (con *funzioni di attivazione a base radiale*, RBF) [74], le reti *Learning Vector Quantization* (LVQ) [75], le *Mappe Auto-Organizzanti* (*Self-Organizing Map*, SOM) [76], le reti basate sull'*Adaptive Resonance Theory* (ART) [77], le reti probabilistiche (PNN) [78] e le reti ricorrenti di Hopfield [79].

Un'ulteriore distinzione per le reti non ricorrenti riguarda il modo in cui viene appresa la classificazione: è possibile parlare di ANN

- *basate sulle caratteristiche* (*feature-based*) nel caso di reti che apprendano un *mapping* tra vettori di ingresso e vettori di uscita (EBP);
- *basate su prototipi* (*prototipe-based*) nel caso di reti che astraggano dall'insieme dei vettori in ingresso dei prototipi, utilizzati come termini di confronto durante la classificazione (LVQ, SOM,ART).

Le reti RBF sono un ibrido tra i due tipi, costituite da un primo livello basato su prototipi e un secondo livello di tipo EBP, basato su caratteristiche.

Tabella II. *Caratterizzazione delle ANN maggiormente utilizzate nell'ambito del PR.*

Architecture	Schema di connessione	Modalità di addestramento	Learning rule	Algoritmi di apprendimento
MLP	Feedforward	Supervisionato	Correzione dell'errore	Back-propagation
RBF	Feedforward	Supervisionato e non supervisionato	Correzione dell'errore e competitivo	RBF
LVQ	Connessione Laterale	Supervisionato o non supervisionato	Competitivo	LVQ1, RPCL, FSCL
SOM	Connessione Laterale	Non supervisionato	Competitivo	Kohonen's SOM
ART	Connessione Laterale	Non supervisionato	Competitivo	ART1, ART2
PNN	Feedforward	--	--	--
Hopfield	Ricorrente	Non supervisionato	Correzione dell'errore	Hebbian rule

8.1. Le reti Error Back-Propagation.

Le reti *Error Back-Propagation* (EBP) costituiscono indubbiamente la classe di modelli neuronali più popolare e maggiormente diffusa a livello applicativo [32; 71; 73]. Come anticipato, la denominazione EBP fa riferimento all'algoritmo di apprendimento utilizzato in una strategia di addestramento supervisionato, mentre l'architettura corrisponde a quella di una rete *feed-forward multistrato* (*MultiLayer FeedForward*, MLFF), costituita da uno strato di ingresso, uno strato di uscita ed eventualmente altri strati *nascosti* (Figura 11).

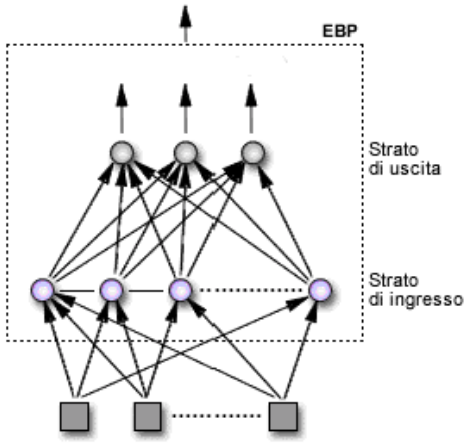

Figura 11. *Architettura di un modulo EBP.*

L'algoritmo di *Back-Propagation* rappresenta un'estensione alle MLFF della

regola delta, introdotta da Widrow e Hoff [80], per le reti *feed-forward mono-strato*, come generalizzazione, a sua volta, della regola di apprendimento del *Perceptron* di Rosenblatt [81] per le reti denominate *Adaptive Linear Neuron* (ADALINE). In tal senso, l'EBP viene anche definita *regola delta generalizzata* e le reti allenate mediante tale algoritmo vengono chiamate *Multilayer Perceptron* (MLP).

Come la *regola delta*, l'algoritmo di EBP è definito per unità di output con funzioni di attivazione differenzialibili ed è basato sulla minimizzazione, attraverso un procedimento di discesa del gradiente, di una funzione errore, calcolata come differenza tra l'uscita effettiva di ciascuna unità e l'output desiderato (tale procedura equivale alla *Least Mean Square*, LMS). In particolare, la regola EBP si distingue dalla regola del *Perceprton* giacché permette l'allenamento di reti neurali con un numero di strati arbitrari ed è questa possibilità, unitamente alla semplicità di funzionamento, la generalità di applicazione e la potenza di calcolo, che ne ha fatto il cavallo di battaglia del connessionismo, permettendo di superare il limite delle prime ANN su problemi come lo XOR.

L'idea alla base del funzionamento dell'algoritmo è quella di retro-propagare alle unità interne l'errore calcolato a livello delle unità di uscita, introducendo connessioni di *feed-back* ideali che permettano di aggiornare, attraverso un processo ricorsivo, i pesi sulle connessione interne in funzione di quanto l'uscita delle unità corrispondenti incida sull'uscita globale della rete.

La derivazione completa della regola EBP è riportata in Appendice B, insieme ad un'analisi della convergenza dell'algoritmo; di seguito ne viene riportato l'aspetto conclusivo nel caso di una rete con due soli strati, ingresso e uscita.

Siano

$\{(\mathbf{x}_p, t_p) | p = 1,..., P\}$ *training set* dei vettori di *pattern* $\mathbf{x}_p$;

x_{pj} l'elemento *j*-esimo del *p*-esimo vettore di ingresso, j=1,…,*m*;

t_{pi} l'uscita desiderata per l'unità *i*-esima, *i*=1,…,*n*, in corrispondenza del *pattern p*-esimo;

w_{ij} il peso lungo la connessione in ingresso al nodo *i*-esimo proveniente dalla componente *j*-esima del vettore di ingresso nel caso di unità di ingresso oppure dall'unità di ingresso *j*-esima nel caso di nodo di output;

o_{pi} l'uscita dell'unita *j*-esima, j=1,…,*n*, calcolata applicando la funzione di attivazione limitata e semilineare, ovvero non decrescente e differenziabile, all'ingresso netto come:

$$o_{pi} = f_i(net_{pi}) = f_i(\sum_{i=0}^{m} w_{ij} x_{pj})$$

per le unità di ingresso;

$$o_{pi} = f_i(net_{pi}) = f_i(\sum_{i=0}^{h} w_{ij} o_{pj})$$

per le unità di uscita

E la funzione errore totale quadratico da minimizzare, calcolata come:

$$E = \sum_{p=1}^{P} E_p \qquad (50)$$

dove E_p è l'errore commesso sul singolo *pattern* $\mathbf{x}_p$ calcolato nel modo seguente:

$$E_p = \frac{1}{2} \sum_{i=1}^{n} (t_{pi} - o_{pi})^2 \qquad (51)$$

La regola EBP prevede il seguente aggiornamento dei pesi dopo ogni passo di elaborazione dell'intero training set, ovvero ogni *epoca*:

$$\Delta_p w_{ij} = \eta \delta_{pi} o_{pj} \qquad (52)$$

dove

$$\delta_{pi} = (t_{pi} - o_{pi}) f_i'(net_{pi}) \qquad (53)$$

se i è un'unità di uscita

$$\delta_{pi} = f_i'(net_{pi}) \sum_k \delta_{pk} \cdot w_{ki} \qquad (54)$$

se i è un'unità di ingresso

ed η è il cosiddetto parametro o tasso di apprendimento (*learning rate*).

Contrariamente alla regola delta, l'algoritmo EBP non assicura la convergenza e presenta l'inconveniente di incorrere in minimi locali. Inoltre, si tratta di un algoritmo molto lento, soprattutto nel caso in cui l'allenamento riguardi una rete con un numero elevato di

unità nascoste. Sulla base di queste considerazioni sono state introdotte numerose varianti dell'EBP, che agiscono sull'algoritmo a vari livelli con l'obiettivo di velocizzarne e migliorarne la convergenza.

Le reti MLFF in generale e, di conseguenza, le EBP godono di importanti proprietà che le rendono un potente strumento matematico ed applicativo.

Sostanzialmente, la caratteristica fondamentale delle reti EBP è rappresentata dalla presenza di uno strato di ingresso ed eventualmente di ulteriori strati nascosti che realizzano una ricodifica interna del vettore di input. Tale codifica consente la trasformazione di un qualsiasi vettore *n*-dimensionale in un nuovo vettore *m*-dimensionale corrispondente, con *m* pari al numero dei neuroni dello strato di uscita [82; 83]. In particolare, nell'ambito del PR, l'elaborazione di un singolo neurone corrisponde all'individuazione di un iperpiano nello spazio degli input, per cui la combinazione di più nodi in un'architettura con almeno lo strato di ingresso distinto da quello di uscita (vale a dire in una rete multistrato) assicura la delineazione di regioni complesse, che permettono alla rete di apprendere e riconoscere le classi in cui risultino suddivisi i *pattern* [9; 67; 84]. In questa visione, l'approccio neurale alla classificazione è simile a quello statistico, in quanto gli iperpiani individuati da ciascun nodo corrispondono ai quelli che vengono denominati *decision boundaries* nel secondo approccio e l'elaborazione di ciascun nodo di uscita corrisponde ad una funzione discriminante non lineare che distingue una determinata classe da tutte le altre. Un'illustrazione di ciò è data in Figura 12 [84] nella quale vengono mostrate le regioni distinte individuate, in uno spazio bidimensionale, da architetture *feed-forward* di complessità diversa, per un problema di classificazione binaria, partendo da quella più semplice costituita da un unico strato.

In particolare, rimanendo nel contesto di un confronto tra i due approcci al PR, è possibile dimostrare che una rete EBP addestrata nella modalità sopra descritta,

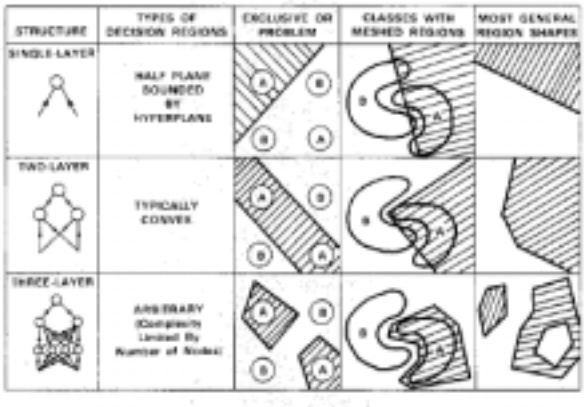

Figura 12. *Regioni di decisione in uno spazio bidimensionale e per problemi di classificazione binari individuate da diverse architetture feed-forward addestrate con l'algoritmo EBP [84].*

ovvero minimizzando l'errore quadratico, nell'ambito di un problema di classificazione, fornisce una stima diretta delle *probabilità a posteriori* di Bayes, [85; 86; 87; 88; 89] definite come le proprietà di avere una certa classe c_i avendo osservato il *pattern* **x**, ovvero

$$P(c_i|\mathbf{x})$$

Ciò, secondo Lippmann [67], offre numerosi vantaggi nello studio e nell'applicazione delle ANN, in quanto permette di combinare le uscite di più reti per ottenere una funzione di decisione ad un più alto livello, permette di compensare le differenze tra le probabilità delle classi dei *pattern* negli insiemi di addestramento e di test, permette di usare le uscite della rete per minimizzare una funzione di rischio alternativo e, infine, suggerisce una misura alternative per la valutazione delle prestazioni della rete stessa.

Altri studi hanno dimostrato l'equivalenza tra reti EBP e approccio statistico alla classificazione [70] e ciò ha portato alla visione di tali ANN come semplici alternative ai metodi statistici classici.

In realtà, le reti neurali offrono numerosi vantaggi rispetto ai metodi suddetti, trattandosi di sistemi non lineari, non parametrici e liberi da modelli, in grado di trattare dati non stazionari e con distribuzioni non Gaussiane [90]. Vari studi condotti per confrontare i due approcci hanno evidenziato

la superiorità delle reti EBP e, in generale, delle ANN, in termini di potere sia teorico [91; 92; 93] sia, sotto certe condizioni, applicativo [94; 95; 96; 97].

D'altro canto, considerando le ANN solamente all'interno del contesto statistico si perdono di vista alcune delle peculiarità che le caratterizzano, quali le capacità di apprendere e generalizzare, che le rendono uno strumento notevolmente potente e dalle svariate applicazioni.

Considerando la rete come un operatore funzionale, le prime indagini in merito alle capacità delle ANN hanno riguardato la proprietà di approssimazione delle MLFF [84]. Gli studi a tal proposito hanno preso spunto dal teorema di *superposizione* di Kolmogorov, per la rappresentazione delle funzioni multivariate attraverso la somma di funzioni monovariate. Precisamente, il teorema può essere enunciato come segue [98]:

Teorema di Kolmogorov.
Per ogni funzione f continua in n (n≥2) variabili nel dominio [0,1], $f:[0,1]^n \to \Re$, *esistono n(2n+1) funzioni univariate continue e monotone crescenti in [0,1] a partire dalle quali f può essere ricostruite secondo l'equazione:*

$$f(x_1, x_2, ...x_n) = \sum_{p=1}^{2n+1} \phi_p \left(\sum_{q=1}^{n} \psi_{pq}(x_q) \right) \quad (55)$$

□

Le funzioni ψ_{pq} sono universali per la dimensione *n* stabilita ed indipendenti da *f*, dalla quale, al contrario, dipendono le funzioni ϕ_p. Solitamente entrambi i tipi di funzioni sono complesse e altamente irregolari, quindi difficilmente determinabili.

L'estensione di tale teorema alle reti neurali è stata studiata da vari autori, nel 1988 Cybenko enunciò un teorema di approssimazione universale direttamente applicabile alle reti multistrato [32]:

Teorema di Cybenko.
Sia φ una funzione continua monotona, non decrescente, non costante e limitata. Dati una qualunque funzione f multivariata, definita nello spazio delle funzioni continue su $[0,1]^n$ *e un valore ε>0, esiste un insieme di costanti reali* α_i, b_i, w_{ij}, m *con i=1,…,m e j=1,…,n, tale che si possa definire la funzione F secondo l'espressione seguente:*

$$F(x_1, x_2, ..., x_n) = \sum_{i=1}^{m} \alpha_i \varphi \left(\sum_{j=1}^{n} (w_{ij} x_j + b_i) \right) \quad (56)$$

Detta funzione F è un'approssimante della funzione f, secondo la condizione seguente:

$$\forall (x_1, x_2, ..., x_n) \in [0,1]^n :$$
$$\left| F(x_1, x_2, ..., x_n) - f(x_1, x_2, ..., x_n) \right| < \varepsilon$$

□

Tale teorema è direttamente applicabile alle reti MLFF con uno strato di ingresso non lineare e uno strato di output lineare, con le seguenti assunzioni:

- $(x_1, x_2, ..., x_n)$ vettore di ingresso;
- strato di ingresso formato da *i*=1,…,*m* unità non lineari
- funzione di attivazione non lineare delle unità di ingresso che rispetti le condizioni del teorema, il caso tipico è quello della funzione logistica;
- w_{ij} e b_i pesi in ingresso alle unità di ingresso;
- strato di uscita come combinazione lineare delle unità di ingresso con coefficienti α_1, α_2, …, α_m.

Il teorema di approssimazione universale è un principio importante che dal punto di vista teorico fornisce uno strumento matematico necessario per motivare l'utilizzo delle reti MLFF, con il solo strato di ingresso, come *approssimatori* di qualunque funzione. Tuttavia, esso postula solamente l'esistenza di una rete in grado di effettuare l'approssimazione, ed ha pertanto, limitati risvolti pratici, in quanto non definisce come costruire la rete suddetta.

Altre proprietà fondamentali delle reti neurali sono

- la capacità di apprendimento, che consiste nell'apprendere in maniera adattiva dai dati di addestramento;
- la capacità di generalizzazione, che corrisponde alla capacità di rispondere correttamente in corrispondenza di *pattern* non visti nella fase di training, il che può corrispondere alla ricostruzione dei *pattern* corretti nel caso in cui siano parziali o affetti da rumore, al riconoscimento o alla classificazione dei *pattern* non visti in precedenza o, ancora, nella previsione degli esiti futuri in funzione di quanto visto nel passato.

Caratteristica fondamentale delle EBP, come di tutte le altre reti neurali, è quella di modificare i propri parametri liberi in maniera adattiva acquisendo conoscenza dai dati e, quindi, apprendendo un determinato compito, come la classificazione o l'approssimazione di una funzione. Tuttavia, alla proprietà di approssimazione universale non corrisponde necessariamente la capacità di apprendimento: è sì vero che una rete può approssimare qualsiasi funzione con l'accuratezza desiderata, ma non è detto che la rete riesca ad apprendere tale approssimazione. In tal senso, è stata sviluppata una specifica teoria che riguarda la capacità di apprendimento e rientra sostanzialmente nell'ambito della disciplina nota come *Apprendimento Automatico* [71]. In tal contesto, la capacità di apprendimento (*learnability*) di una rete MLFF è definita come la capacità di un algoritmo di addestramento di trovare i valori dei parametri del modello, ovvero dei pesi, per fornire l'accuratezza desiderata nel processo di approssimazione della funzione considerata. Detta capacità è influenzata da varie caratteristiche, quali l'architettura della rete, l'algoritmo di apprendimento stesso e l'insieme di addestramento. In particolare, mentre la capacità di generalizzare è legata all'informazione contenuta nel *training set*, la *learnability* è connessa maggiormente alla tipologia di rete e alla complessità computazionale del problema. Purtroppo, gli unici risultati teorici ottenuti a riguardo, che sono anche sostanzialmente legati alla misura di generalizzazione della rete, sono dei limiti superiori alle grandezze coinvolte [99]: indicato con P il numero di esempi di training, con n il numero di neuroni dello strato di ingresso, con W il numero dei pesi e con ε l'errore di generalizzazione desiderato, vale la seguente condizione

$$P \geq O\left(\frac{W}{\varepsilon}\log_2\frac{n}{\varepsilon}\right)$$

Di conseguenza, può essere dimostrato che una rete MLFF con le caratteristiche specificate non può apprendere e generalizzare bene nel caso in cui le siano forniti in fase di addestramento un numero di esempi inferiori al rapporto W/ε:

$$P \geq \Omega\left(\frac{W}{\varepsilon}\right)$$

ad esempio, nel caso in cui si voglia avere un livello di accuratezza del 90%, ovvero con ε pari a 0.1, si devono utilizzare un numero di esempi maggiore di dieci volte il numero dei pesi della rete.

Altro questione legata all'apprendimento è il grado di adattabilità del sistema neurale che influenza la stabilità dello stesso: le due proprietà possono essere infatti inversamente proporzionali, nel senso che un'elevata plasticità può andare a scapito della stabilità. Ciò che ci si aspetta da un sistema neurale, infatti, è che si adatti in risposta ad eventi significativi e rimanga contemporaneamente stabile nel caso di eventi poco significativi (*stability-to-plasticity dilemma*) [32]. E', invece, possibile che un sistema molto flessibile che abbia incamerato già un certo grado di conoscenza durante l'allenamento perda una certa quantità di questa informazione al momento della presentazione di un nuovo *pattern* di *training*. Questo fenomeno prende il nome di *interferenza catastrofica* ed è stato esplorato in vari studi [77; 101; 102], che hanno evidenziato come i fattori determinanti possano essere l'ordine di presentazione dei *pattern*, una sovrapposizione intrinseca degli stessi e il tipo di procedura seguita durante l'apprendimento.

Per quanto riguarda la proprietà di generalizzazione di una rete EBP, essa consiste nella capacità della stessa di estrapolare informazione anche da un numero ridotto di *pattern* di addestramento. Si tratta di una caratteristica comune anche ad altri sistemi adattivi usati nell'ambito del PR, ma ciò che distingue le ANN da questi è il numero elevato di parametri che esse utilizzano. Tuttavia, la capacità di generalizzazione della rete può essere compromessa nel caso in cui questa si specializzi troppo sui dati di addestramento, tale fenomeno, definito *overfitting* o *sovraddestramento*, corrisponde ad un valore molto basso dell'errore sull'insieme di training e, al contrario, ad un valore elevato dello stesso sull'insieme di test. Vari metodi sono stati proposti per risolvere tale inconveniente [103; 104; 105], uno di questi è il cosiddetto *early-stopping* che consiste nell'utilizzo in fase di addestramento di un insieme detto di validazione sul quale testare la capacità di generalizzazione della rete: contemporaneamente all'errore di training viene calcolato l'errore su tale insieme e se questo aumenta per un certo numero di epoche successive, si interrompe l'allenamento. In realtà, esistono delle controindicazioni a tale tecnica in quanto non necessariamente l'andamento dell'errore sull'insieme di validazione è monotono e può, pertanto, accadere che ad un certo incremento segua un decremento. Un'interessante osservazione è quella riportata da Dietterich [106], secondo la quale il miglioramento dell'algoritmo di apprendimento non aumenta la capacità di generalizzazione della rete.

La potenza e la semplicità di calcolo unitamente alla duttilità applicativa fanno delle reti EBP una delle classi di ANN maggiormente utilizzate nella pratica. Le architetture EBP generalmente utilizzate non prevedono strati nascosti, grazie essenzialmente alle capacità discusse nel paragrafo precedente di tale reti.

Il campo di applicazione è vasto e comprende la soluzione ai *task* fondamentali di varie discipline come la *Bioinformatica*, per la deduzione della struttura secondaria delle proteine [107]; l'elaborazione dei segnali, per l'eliminazione del rumore [9; 108]; le previsione delle serie temporali, in particolare per le previsioni meteorologiche [109] e finanziarie [110].

Nell'ambito del PR, una delle applicazioni più frequenti riguarda il riconoscimento del parlato [111] e dei caratteri manoscritti [112; 113]. Al primo caso appartiene la famosa NETTALK, un sistema in grado di realizzare la conversione tra testo e parlato riconoscendo un migliaio delle parole più comuni della lingua inglese. Il sistema è costituito da una rete EBP con sette gruppi di nodi di ingresso, ciascuno dei quali in grado di codificare un gruppo di lettre o simboli speciali.

Nell'ambito della classificazione di immagini, un campo di particolare diffusione è quello della *Medical Imaging*: le reti EBP sono state applicate a vari livelli di elaborazione, ovvero sia a livello della formazione delle immagini, come mostrato da Kerr e Barlett [114] nel caso della SPECT (*Single Photon Emission Computed Tomography*) e da Yan e Mao [115] per l'eliminazione di artefatti nel caso della risonanza magnetica (MRI); sia a livello intermedio, ovvero per la segmentazione delle immagini, come nel caso della MRI per la determinazioni della soglia di binarizzazione sulla base dell'istogramma delle immagini [116]; sia ad alto livello, per la classificazione, come nel caso della classificazione di lesioni nella MRI [117], nel caso della rilevazione microcalcificazioni nelle immagini mammografiche, attraverso un insieme di feature sia spaziali che spettrali [118] o, ancora, nel caso dell'individuazione di lesioni maligne al seno, attraverso un insieme di feature di tessitura dell'ultrasonogramma.

8.2. Le Self-Organizing Map.

Le *Self-Organizing Maps* o *Mappe auto-Organizzanti* (SOM), dette anche S*elf-Organizing Features Maps*, costituiscono un'importante classe di reti neurali in grado di apprendere in maniera autonoma, adattandosi, secondo regole di plasticità sinaptica, in risposta a stimoli provenienti dall'esterno. Il

risultato di tale adattamento consiste in una rete che mappa i *pattern* di ingresso in *pattern* di uscita, in modo *topologicamente coerente*, preservandone l'ordine e rispecchiandone la distribuzione di probabilità.

Lo sviluppo delle SOM è stato ispirato dalla osservazione della organizzazione topologica del cervello umano, in riferimento al tipo di stimolazione sensoriali: differenti stimoli percettivi, quali stimoli di natura tattile, acustica o visiva, vengono presentati in maniera topologicamente ordinata a differenti mappe computazionali del cervello, ciascuna delle quali ha una ben precisa localizzazione nella corteccia cerebrale e costituisce un blocco base nell'infrastruttura di processazione dell'informazione del sistema nervoso. Tale organizzazione è stata tradotta da Kohonen nel *principio di formazione delle mappe topografiche*, secondo il quale "*la localizzazione spaziale di un neurone di output in una mappa topografica corrisponde ad un particolare dominio o caratteristica dei dati provenienti dallo spazio degli ingressi*" [75]. Questo principio ha fornito la motivazione neurobiologica per lo sviluppo di due diversi modelli di mappe auto-organizzanti: il primo dovuto a Willshaw e von der Malsburg per spiegare il problema della corrispondenza dei segnali provenienti dalla retina, alla mappa visiva della corteccia cerebrale, il secondo introdotto da Kohonen [76] senza l'obiettivo di spiegare dettagli neurobiologici. Quest'ultimo risulta essere più generale del precedente, in quanto è in grado di effettuare una compressione dei dati, come ad esempio una riduzione dimensionale dei dati di input. Più precisamente, il modello di Kohonen appartiene alla classe degli algoritmi di *vector-coding*: offre un *mapping* topologico che colloca un numero fissato di vettori (*code words*) in uno spazio di input sovradimensionale, permettendo, in tal modo, la compressione dei dati. Questo motivo, interpretabile anche come capacità di estrarre informazioni dai dati e comprendere come questi si raggruppino spontaneamente in *cluster*, ha fatto sì che il modello di Kohonen ricevesse maggiore attenzione in letteratura.

La definizione di detto modello neurale prevede l'organizzazione delle unità come nodi di un'unica griglia o *lattice*, completamente connessi al vettore di ingresso e tra loro con connessioni laterali, in grado di apprendere in maniera non-supervisionata, attraverso un processo di *competizione*, *cooperazione* ed *adattamento*, a trasformare un *pattern* di input di dimensione arbitraria in una mappa mono o bidimensionale, assicurando che input vicini siano *mappati* in output vicini. Un esempio di architettura bidimensionale è mostrato in Figura 13.

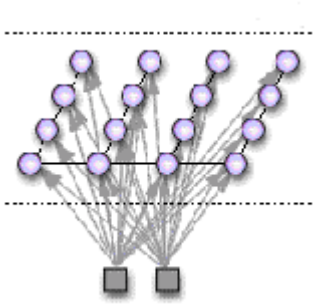

Figura 13. *Esempio di SOM bidimensionale*

La fase di addestramento di una SOM corrisponde al processo di formazione della mappa topologica durante il quale per ogni vettore di ingresso $\mathbf{x}_p$[7] i nodi della rete competono sulla base della distanza tra $\mathbf{x}_p$ e il vettore dei pesi $\mathbf{w}_i$ ad essi associati, lungo le connessioni in ingresso. Il nodo vincente *k*, detto *best match unit* (BMU), è quello che si trova a distanza inferiore e, quindi, meglio approssima la proiezione geometrica del vettore in ingresso nella rete. Il vettore dei $\mathbf{w}_k$ viene, dunque, modificato in modo da specializzare ulteriormente il nodo sull'ingresso appena processato.

Per mantenere la coerenza topologica con lo spazio degli stimoli, è prevista una fase di cooperazione in cui vengono modificati i pesi dei neuroni presenti in una zona prefissata nell'intorno del neurone vincente, detta *di vicinanza topologica*. In tal modo, è un'intera porzione della mappa che si sposta ad approssimare l'ingresso ed è ciò che distingue

[7] $\mathbf{x}_p \in \{(\mathbf{x}_p, t_p) | p = 1,..., P\}$ secondo la stessa notazione introdotta per le EBP

le SOM dalle altre reti semplicemente competitive.
Complessivamente, la regola di aggiornamento prevista è la seguente:

$$\boldsymbol{w}_j(t+1) = \begin{cases} \boldsymbol{w}_j(t) + \eta(t) h_{kj} \left[\boldsymbol{x}_i - \boldsymbol{w}_j(t) \right] & se\ j \in N_k(t) \\ \boldsymbol{w}_j(t) & se\ j \notin N_k(t) \end{cases} \quad (57)$$

dove

$\mathbf{w}_j$ è il vettore dei pesi all'iterazione t del generico nodo j;

$\eta(t)$ è il parametro di apprendimento (*learning rate*) che viene fatto decrescere durante il processo di addestramento in funzione dell'iterazione corrente e del numero massimo di passi, secondo la funzione esponenziale inversa.

$N_k(t)$ è l'insieme dei neuroni che si trovano nella zona di vicinanza topologica del nodo k di raggio $\rho(t)$, ovvero:

$$N_k(t) = \left\{ j \mid d_{kj} < \rho(t) \right\} \quad (58)$$

con $d_{kj} = \|k - j\|$ misura della distanza tra i due nodi. Anche il raggio $\rho(t)$ viene fatto decrescere nel tempo secondo un procedimento di decadimento esponenziale nel numero di passi di allenamento;

h_{kj} è, infine, una funzione che tiene conto delle distanze all'interno della zona di eccitazione, influenzando la modifica dei pesi in maniera inversamente proporzionale alla distanza tra i nodi.

Il risultato dell'algoritmo di apprendimento delle SOM basato su tale procedimento può essere controllato visivamente proiettando i nodi della griglia nello spazio degli input attraverso i valori dei vettori dei pesi. Un esempio relativo al caso di vettori di input bidimensionali uniformemente distribuiti nell'intervallo [-1;1] in entrambe le coordinate è mostrato in Figura 14: lo stadio finale viene raggiunto dalla rete dopo una fase di ulteriore adattamento dei vettori dei pesi, eseguita per meglio sintonizzarli sui vettori di input.

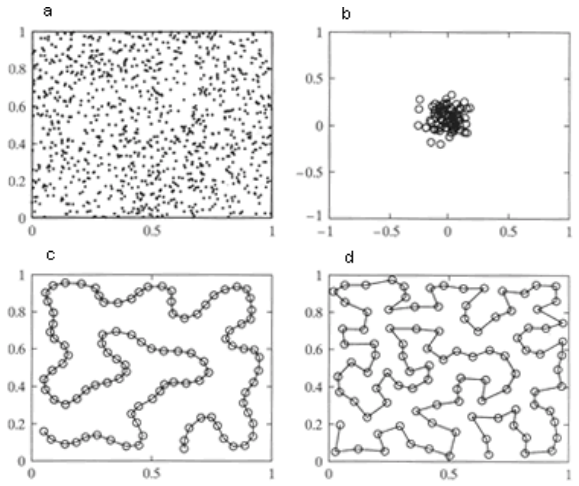

Figura 14. *Processo di formazione di una SOM unidimensionale in uno spazio degli stimoli bidimensionale [32].*

Come evidente dalla descrizione dell'algoritmo, la fase di addestramento delle SOM non prevede la minimizzazione di nessuna funzione costo esplicita. E', però, possibile valutare due diverse misure di bontà della rete una volta che questa sia stata allenata, ma anche durante l'allenamento della stessa [119; 120]:

- l'errore di *quantizzazione* definito come la distanza media tra ciascun vettore di ingresso e il BMU corrispondente:

$$E_q = \frac{1}{P} \sum_{p=1}^{P} \left\| \mathbf{x}_p - \mathbf{w}_{\mathbf{k}}(t) \right\| \quad (59)$$

- l'errore *topologico* (o *topografico*), definito come la proporzione di tutti i vettori di ingresso per i quali il primo e il secondo neurone vincente non sono vicini tra loro:

$$E_t = \frac{1}{P} \sum_{p=1}^{P} \zeta(\mathbf{x}_p) \quad (60)$$

dove $\zeta(\mathbf{x}_p)$=1 se i due BMU di $\mathbf{x}_p$ non sono vicini e 0 altrimenti.

Dell'algoritmo di base sono state proposte diverse varianti che agiscono sia sul tipo di funzioni distanza e di vicinanza topologica utilizzate nella regola di apprendimento, sia sull'architettura della rete. Le più interessanti appartengono al secondo gruppo e sono la SOM *gerarchica* (*Hierarchical* SOM,

HSOM) costituita da più strati di mappe bidimensionali e la cosiddetta SOM *temporale* (*Temporal* SOM, TSOM) in grado di gestire sequenze di vettori di ingresso.

Le SOM godono di importanti proprietà, ciascuna delle quali presenta notevoli vantaggi nell'ambito delle applicazioni pratiche [32; 75; 121].

Formalmente, si consideri uno spazio continuo dei dati di input X, la cui topologia risulti definita da relazioni metriche tra i vettori $\mathbf{x} \in X$, e sia A lo spazio discreto di output, definito topologicamente dalla disposizione di un insieme di neuroni all'interno di un lattice. Sia Φ una trasformazione non lineare, chiamata *Feature Map*, che effettui il *mapping* tra lo spazio di input X e lo spazio di output A, ossia :

$$\Phi : X \to A \tag{61}$$

La trasformazione Φ può essere interpretata come un'astrazione della funzione che definisca la posizione del neurone vincente in corrispondenza di un determinato *pattern* **x** in ingresso. Nel momento in cui venga fornito un vettori di input **x**, la SOM identifica il neurone vincente $i(\mathbf{x})$ nello spazio di output A in accordo alla trasformazione Φ. Il vettore peso $\mathbf{w}_i$ di tale neurone può essere considerato come un puntatore al neurone $i(\mathbf{x})$ nello spazio di input X, in altre parole, gli elementi del vettore $\mathbf{w_i}$ possono essere visti come le coordinate dell'immagine del neurone $i(\mathbf{x})$ proiettato nello spazio degli input. Con queste premesse, è possibile discutere le principali proprietà della mappa Φ:

i. Approssimazione dello spazio di input.
La trasformazione Φ, *rappresentata come l'insieme dei pesi sinaptici* $\{w_j\}$ *dello spazio di output* A, *offre una buona rappresentazione dello spazio di input* X.
La SOM, infatti, è in grado di memorizzare una grande quantità di vettori di ingresso, determinando un insieme ridotto di *prototipi* che li rappresentino. Questo concetto è alla base della teoria della *vector quantization* che riguarda la compressione dei dati e, in generale, la riduzione delle loro dimensioni. In tal senso, è possibile dimostrare che il processo di apprendimento della SOM coincide con un algoritmo di quantizzazione vettoriale e la funzione di vicinanza topologica definisce una funzione di densità.

ii. Ordinamento Topologico.
La trasformazione Φ *effettuata dall'algoritmo è ordinata topologicamente, nel senso che la localizzazione spaziale di un neurone all'interno del lattice corrisponde ad un particolare dominio o caratteristica dei* patterns *di input.*

Tale proprietà deriva come conseguenza diretta dalla regola di aggiornamento dei pesi che forza il vettore $\mathbf{w_i}$ del nodo vincente a muoversi nella direzione del vettore di ingresso. E' possibile in tal senso rappresentare gli elementi dello spazio di output A all'interno dello spazio degli input, ottenendo in tal modo il partizionamento di tale spazio in regioni di attivazione di ciascun neurone del lattice. Questa proprietà rende evidente, quindi, la possibilità di impiegare una SOM come un efficiente algoritmo di *clustering*.

iii. Corrispondenza delle densità.
La trasformazione Φ *riflette le variazioni statistiche della distribuzione degli input, ossia regioni nello spazio degli input* X, *dalle quali si sono scelti vettori esempio* **x** *con un'alta probabilità di occorrenza, corrispondono ad ampi domini dello spazio di output* A, *rispetto a quelle regioni in* X, *da cui sono stati scelti vettori con una bassa probabilità di occorrenza.*

E' possibile dimostrare che considerando la funzione f_X (x) di densità della distribuzione dei vettori dello spazio di input e il fattore di ingrandimento definito come il numero di neuroni in un piccolo volume dello spazio degli input, si vede che questo non è proporzionale alla densità.

iv. Estrazione delle caratteristiche.

Presi dei dati da uno spazio di input con una distribuzione non lineare, una Self-Organizing Map è in grado di estrarre un insieme delle migliori caratteristiche per l'approssimazione di tale distribuzione.

Questa proprietà è una conclusione naturale delle proprietà 1 e 2 e dimostra come le SOM possano essere applicate con successo nella fase di estrazione delle *feature* in un problema di PR. In particolare, le SOM sono equiparabili all'analisi delle componenti principali (*Principal Component Analysis*) con la vantaggiosa proprietà di dare buone prestazioni anche nel caso in cui la distribuzione dei vettori sia non lineare ed affetta da rumore [32].

Nell'ambito del PR, le SOM rientrano, come già osservato, nella classe di reti neurali basate su prototipi [9; 72], in particolare, grazie alle prime tre delle proprietà elencate, esse costituiscono degli ottimi algoritmi di *clustering*, caratterizzati dal vantaggio di non richiedere nessun informazione a priori riguardo alla distribuzione dei dati di ingresso e dei cluster da stabilire. Ciò pone queste reti ad un livello superiore rispetto agli altri algoritmi di *clustering* statistici parametrici [122]. L'elaborazione implementata in una SOM è, infatti, molto simile a quella dell'algoritmo delle *k-medie* (*k-means*), tuttavia mentre per quest'ultimo è necessario stabilire il numero di *cluster* da individuare, nel caso delle SOM è possibile selezionare un numero arbitrario di nodi (centroide dei *cluster* in questa visione), giacché il numero di insiemi può essere poi dedotto attraverso una visualizzazione opportuna dei risultati ottenuti. Un modo molto utile per effettuare questa operazione è rappresentato dalla cosiddetta *Unified distance matrix* (U-*matrix*) [123; 124] definita calcolando la distanza di ogni nodo della rete dai propri vicini e rappresentandola graficamente come una terza dimensione, in genere costituita dalle tonalità di grigio, nell'immagine mostrata. Utilizzando tale approccio, dati vicini corrispondono a valori bassi di intensità e i *cluster* identificati dalla rete sono separati dalle zone ad intensità maggiore.

Le SOM più comunemente usate nelle applicazioni pratiche sono quelle mono o bidimensionali, poiché la visualizzazione nello spazio 2D è più semplice ed immediata, inoltre la maggior parte delle implementazioni software o degli ambienti di sviluppo per applicazioni con reti di questo tipo, supportano solo mappe mono o bi-dimensionali.

Le applicazioni di questo particolare tipo di reti non supervisionate sono molteplici e spaziano dal campo della *Vector Qantization* a quelli di *Data Compression*, *Image Processing* ed *Image Undestanding* e trattano problemi che vanno da quelli di ottimizzazione [71], controllo di robot, a problemi di riconoscimento di caratteri [125], riconoscimento del linguaggio parlato [126], a problemi di riconoscimento e classificazione [127; 128; 129; 130] e comparazione di immagini [131; 132]. Gli articoli presenti nella letteratura scientifica in merito sono dell'ordine delle migliaia, di seguito ne vengono riportati gli esempi più significativi e pertinenti con il lavoro di tesi.

Nell'ambito del PR, uno dei risultati più entusiastici fu raggiunto già a partire dai primi anni del 1980 da Teuvo Kohonen e colleghi, alla *University of Technology* di Helsinki, con lo sviluppo della "*phonetic typewriter*", un sistema di riconoscimento in grado di registrare e trascrivere un linguaggio parlato in un testo ortograficamente corretto [133]. Kohonen ed il suo gruppo ebbero un notevole successo per linguaggi fonetici come il Finlandese o il Giapponese. Non è molto chiaro, quanto facilmente i loro risultati possano essere stati estesi a linguaggi meno fonetici come l'Inglese, il Russo o il Cinese, dove un ruolo fondamentale è svolto anche dall'intonazione della voce. A tale scopo è stata usata una rete LVQ (*Learning Vector Quantization*), per effettuare la corrispondenza tra i fonemi e le classi fonetiche di riferimento. I vettori *pattern* di input sono stati ottenuti dallo spettro del suono del linguaggio parlato e successivamente classificati in una delle 26 classi fonetiche del linguaggio Finlandese. Da tale classificazione risulta una sequenza di simboli che devono essere fusi in

segmenti, ciascuno dei quali rappresenta un fonema specifico del linguaggio parlato.

Un'applicazione sempre nell'ambito del riconoscimento di fonemi e linguaggio è quella proposta da James e Mikkulainen [134], che propongono l'impiego di una SOM temporale in grado di apprendere e riconoscere sequenze arbitrarie di valori binari e reali. Tale rete, denominata dagli autori SARDNET, è stata applicata al riconoscimento di fonemi della lingua Inglese, descritti da vettori di cinque *feature* secondo l'*International Phonetic Alphabet*, e ha dimostrato di assicurare ottime prestazioni anche con un numero contenuto di epoche di allenamento.

Le proprietà di *clustering* delle SOM sono state sfruttate in un'interessante applicazione è quella proposta nel lavoro di tesi [135], per il raggruppamento di punti uniformemente distribuiti all'interno di sette volumi, che rappresentano grossolanamente una sagoma umana. Detti volumi corrispondo ciascuno ad una zona caratteristica del corpo, essenzialmente il busto, il collo, la testa e gli arti superiori ed inferiori, etichettati diversamente. Alla rete vengono dati in input dei punti scelti nello spazio tridimensionale e questa è in grado di organizzarli spazialmente nello spazio bidimensionale in modo che punti vicini risultino appartenenti allo stesso volume e quindi alla stessa zona del corpo.

Ulteriore applicazione basata su queste proprietà delle SOM è quella sviluppata recentemente sempre dall'equipe di ricerca di Kohonen per il *text-mining* [136; 137]. La SOM sviluppata, denominata WEBSOM, permette di organizzare in modo automatico grandi quantità di documenti di testo in uno spazio bidimensionale rappresentato dal lattice della rete stessa. Il procedimento prevede la trasformazione dei dati in vettori di parole, quindi, la riduzione degli stessi attraverso un processo di estrazione delle caratteristiche basata sulla PCA. Una volta addestrata, la rete va a costituire una "piattaforma" sulla quale documenti similari appaiono vicini l'un l'altro e che può essere usata, associando a ciascuna zona un'etichetta che identifichi i gruppi di documenti, per effettuare ricerche visuali nella collezione di documenti presa in considerazione.

Nell'ambito delle applicazioni dell'*Image Analysis* alla *Medical Imaging*, Seiffert *et al.* [138] hanno applicato una SOM per l'individuazione di zone tumorali nelle immagini del cervello ottenute come *slice* della Tomografia Computerizzata. Tali immagini sono state rappresentate attraverso un insieme di *feature* di tessitura, successivamente ridotto attraverso un processo di selezione, utilizzando come criterio di bontà una misura delle tonalità di grigio. I vettori di descrizione così ottenuti sono stati utilizzati per addestrare una SOM di dimensioni 30x30, impiegata nelle processo di classificazione effettivo associando ad ogni neurone l'etichetta relativa alla classe di appartenenza di questo sulla base dei risultati dell'allenamento.

8.3 Le reti neurali gerarchiche.

Lo sviluppo di *reti neurali gerarchiche* (*Hierarchical Neural Network*, HNN) costituisce un ambito di ricerca molto battuto negli ultimi anni [139; 140; 141; 142; 143]. L'idea alla base dell'impiego di architetture neurali ottenute combinando un insieme di ANN è quella di utilizzare la classica strategia del *divide-et-impera*, suddividendo un problema complesso in un numero di sotto-problemi di minore difficoltà computazionale, risolvendo quest'ultimi e combinando, quindi, le soluzioni ottenute. Nell'ambito di un apprendimento supervisionato, si parla di suddivisione del compito da apprendere tra un certo numero di *esperti*, ciascuno specializzato su un sotto-problema, e il sistema risultante viene denominato *comitato di esperti* (o macchine, *Committee Machine*) [32]. Se questi esperti lavorano in maniera indipendente l'uno dall'altro per poi essere integrati da un'opportuna unità elaboratrice che non comunica loro nemmeno il risultato finale, le ANN corrispondenti vengono riferite come *moduli* e si parla di *rete modulare* [144]. D'altro canto, la combinazione di reti neuronali in strutture gerarchiche è un problema molto simile a quello affrontato nell'ambito del PR che si occupa della combinazione di più

classificatori (*Classifier Combination*) [30; 145]

Le HMM posso essere distinte in due categorie principali comprendenti, rispettivamente:

- strutture statiche;
- strutture dinamiche;

Le prime sono reti in cui il risultato di più esperti è combinato secondo meccanismi che non coinvolgono il vettore in ingresso: ne sono esempi l'*Ensemble Averaging* [32], in cui l'uscita di diversi elementi neuronali sono combinati linearmente per produrre un unico risultato finale e il *Boosting* [146; 147; 148], in cui un algoritmo di apprendimento debole viene convertito in uno nuovo in grado di raggiungere l'accuratezza desiderata.

Il secondo tipo di struttura corrisponde a reti in cui il segnale di ingresso è direttamente utilizzato nel meccanismo di integrazione dei risultati dei vari esperti: ne sono esempi, la *Mixture of Experts* (Figura 3.35a) nella quale le uscite delle reti costituenti vengono combinate attraverso un rete detta di *gating* e la *Misture of Esperts* gerarchica (*Hierarchical Mixture of Esperts*), costituita da un insieme di reti di tipo *Mixture of Esperts* organizzate secondo una gerarchia ad albero [149; 150].

In generale, la scelta della combinazione di più reti, soprattutto nell'ambito del PR, è motivata da vari fattori, quali l'esistenza di diversi reti già disponibili sviluppate in contesti diversi e con differenti modelli di rappresentazione o descrizione del problema affrontato; la disponibilità di più insiemi di addestramento collezionati in tempi o ambienti diversi, che eventualmente usano diverse caratteristiche di descrizione; la possibilità di combinare diverse reti che abbiano prestazioni diverse a livello locale, nell'ambito della classificazione ciò equivale a classificatori che danno ottimi risultati su regioni limitate dello spazio dei *pattern*; la possibilità di combinare reti che danno risultati diversi in corrispondenza di differenti valori dei parametri di apprendimento.

Complessivamente, dunque, nell'applicazione di una HNN si possono avere a disposizione diversi insiemi di caratteristiche di descrizione dei dati, diversi insiemi di training e diversi sistemi neurali. L'idea è, quindi, quella di combinare queste ultime per migliorare la prestazione complessiva del sistema gerarchico così ottenuto. Ciò che è necessario, quindi, stabilire è la modalità con la quale vengano combinati gli esperti. In letteratura sono state proposte molte soluzioni a tale questione. Generalmente esse prevedono un insieme di esperti individuali che interagiscono e vengono invocati sulla base dello schema di combinazione stabilito. Gli schemi proposti in tal senso possono essere raggruppati in riferimento all'architettura neurale e corrispondono a tre gruppi principali:

- parallelo;
- sequenziale;
- gerarchico.

Nel primo caso, tutti gli esperti individuali vengono invocati separatamente e i risultati forniti da ciascuno di essi vengono combinati da un modulo definito *combiner* come avviene nel caso della HNN *Ensemble Averaging* mostrata in Figura 3.34. Nella versione *pesata* (*gated*) di questo schema, i risultati vengono innanzi tutto pesati da un'unità di sbarramento e, quindi, combinati (in questa tipologia rientra l'architettura *Mixture of Export* di Figura 3.35(a)). Nello schema sequenziale, le reti individuali vengono invocate in cascata, l'una di all'altra. Per motivi computazionali, l'esperto meno costoso anche se meno efficiente viene eseguito per primo, seguito da quelli via via più efficienti. Nello schema gerarchico, infine, le reti singole vengono combinate secondo un'architettura ad albero, come nel caso della *Mixture of Export* gerarchica. Il vantaggio di un'architettura di questo tipo è rappresentato da maggiori efficienza e accuratezza dovute alla possibilità di esplorare il potere discriminante di insiemi di *feature* diverse. Infatti, un'architettura di questo genere si presta al meglio e dovrebbe essere impiegata per considerare insiemi di *feature* diverse piuttosto che diversi insiemi di addestramento [30]. Per quanto riguarda, quindi, il modulo *combiner*, le soluzioni possibili sono nuovamente molteplici e

possono consistere in un processo di *voting* (il risultato è valido solo se condiviso dalla metà più uno degli esperti), di mediazione o somma, oppure nel cosiddetto *Borda count* che prevede di assegnare a ciascun risultato possibile la somma dei risultati equivalenti forniti dalle reti individuali.

I risvolti applicativi delle reti neurali gerarchiche sono di carattere generale e sono da ricondursi a quelli delle reti singole che le costituiscono, giacché il loro impiego mira essenzialmente a migliorare le prestazioni di queste ultime. Si va, quindi, dalla robotica [151] all'elaborazione dei segnali [152], dal riconoscimento dei caratteri manoscritti [153; 154] alle previsioni in campo industriale [155], dal controllo del traffico [156] al *clustering* di geni [157]. A differenziare le varie applicazioni sono le reti che costituiscono la HNN e la tecnica adottata per la combinazione di queste.

Nell'ambito del PR, un'interessante applicazione è quella proposta da Kang *et al.* nell'ambito dell'OCR, rappresentata dalla combinazione di una SOM e di un insieme di reti EBP [158]. L'architettura delineata prevede un primo livello costituito da una SOM che riceve in ingresso una griglia di rappresentazione del carattere e ne stabilisce il *cluster* di appartenenza, in relazione a quelli associati a ciascun carattere dell'alfabeto precedentemente appresi dalla rete. Il *cluster* in questione contiene l'insieme di vettori di ingresso simili, per cui al passo successivo della classificazione viene utilizzata una rete EBP per distinguere tra questi, riconoscendo quello corretto. Tale rete viene addestrata con quello che gli autori chiamano nodo di *falso allarme*, che equivale all'aggiunta di un ulteriore nodo di uscita che stia a rappresentare il caso in cui il *cluster* riconosciuto dalla SOM non sia corretto. La rete gerarchica così addestrata è stata utilizzata per il riconoscimento dei caratteri della lingua coreana e di quella cinese, con una SOM di 30x30 nodi iniziali, poi ridotti con un ulteriore processo di *clustering* a 19x19 e con un totale di 361 reti EBP, con uno strato interno di unità nascoste.

Nell'ambito di PR ed IA applicati alla *Medical Imaging*, Vladutu *et al*. hanno proposto una rete neurale gerarchica costituita da una SOM combinata con una rete MLFF, per l'individuazione dell'ischemia del miocardio attraverso l'analisi dell'andamento dell'elettrocardiogramma [159]. L'idea sfruttata è quella di combinare le due topologie di reti come esperti locali, utilizzando la SOM nelle regioni dello spazio degli input laddove le classi da riconoscere siano linearmente separabili e, quindi, facilmente apprendibili dalla rete e di utilizzare una rete con addestramento supervisionato nel caso di regioni in cui le classi si sovrappongano, in modo da "forzare" la costruzione di *decision boundaries* complessi. La combinazione delle due reti consiste nell'addestramento iniziale della SOM finché il numero di input classificati in maniera ambigua non scenda al di sotto di una certa soglia, condizione a partire dalla quale viene fatto partire l'allenamento della rete MLFF. Gli autori hanno confrontato le prestazioni di due possibili architetture per la rete supervisionata: la RBF e la SVM, paragonando i risultati di entrambe anche con quello di una semplice SOM. La rete migliore è risultata essere quella combinata con la SVM.

Ulteriori applicazioni nello stesso ambito sono quelle proposte da Sadja *et al.* [160] e da Keem *et al.* [161] per il riconoscimento di microcalcificazioni in immagini ottenute dalla mammografia digitale e delle lesioni rilevate dall'arteriografia coronaria, rispettivamente. Nel primo caso, viene proposta un'architettura costituita da due reti EBP multistrato senza unità nascoste collegate in modo gerarchico, ovvero l'uscita della prima viene mandata in ingresso alla seconda e le due reti vengono allenate separatamente. Nel secondo caso, l'architettura è di tipo parallelo costituita da due reti allenate separatamente su due insiemi di caratteristiche diverse, distinte in densitometriche e in geometriche.

Infine, Di Bona *et al.* [162] hanno delineato ed applicato una HNN costituita da un insieme di moduli SOM e una rete EBP, per la caratterizzazione di zone cerebrali a diversa densità in immagini volumetriche ottenute attraverso la tomografia computerizzata e la risonanza magnetica.

L'approccio proposto prevede un'organizzazione delle reti in due livelli distinti: il primo è costituito da un insieme di moduli SOM, uno per ogni caratteristica utilizzata per la descrizione dei *voxel*, tali moduli eseguono una prima classificazione grossolana che viene poi raffinata a livello superiore da una rete EBP che prende in input i risultati dello strato precedente per restituire l'uscita finale dell'intera HNN. L'architettura di detto sistema si presenta come mostrato in Figura 15.

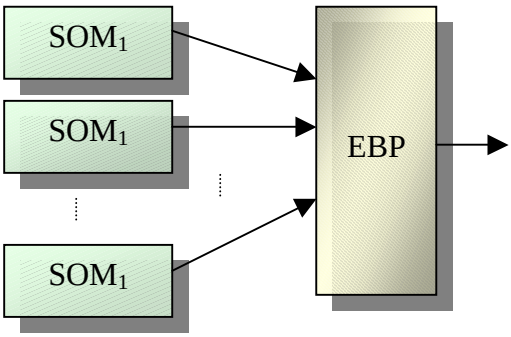

Figura 15. *Architettura del sistema gerarchico costituito da un primo livello di moduli SOM e un secondo livello con un unico modulo EBP.*

Dall'analisi dei problemi connessi alla progettazione e allo sviluppo di un sistema per il PR e *Image Analysis* per l'interpretazione e la classificazione di dati immagine, si evince come, in un approccio alla categorizzazione di tipo *feature-based*, la scelta delle caratteristiche rappresenta il passo critico nella costruzione del sistema, poiché in genere non è possibile stabilire a priori, con certezza, quale sia l'insieme ottimo delle misure disponibili, ma al contrario si procede molto spesso nella selezione delle caratteristiche sulla base del loro significato intuitivo. Inoltre, nel caso in cui si adotti per la classificazione un paradigma neurale costituito da un'unica ANN, l'eventuale modifica nella descrizione del problema, ovvero dell'insieme di caratteristiche selezionate, comporterebbe la necessità di ristrutturare ed allenare nuovamente l'intero sistema neurale.

Sulla base delle osservazioni precedenti, il sistema neurale gerarchico delineato nel modo suddetto risulta offrire numerosi vantaggi nell'ambito del problema della categorizzazione delle immagini: permette di sfruttare i vantaggi della combinazione gerarchica di più moduli neuronali introdotti nella parte iniziale della sezione e offre un elevato grado di scalabilità al modello adottato.

Si tratta, inoltre, di un approccio del tutto generale che può essere applicato a problemi di categorizzazione di dati immagine di qualsiasi tipo e particolarmente adatto a casi complessi in cui il compito abbia natura multivariata, in riferimento alle grandezze coinvolte.

9. Conclusioni.

In questo lavoro è stato affrontato il problema della categorizzazione delle strutture contenute in immagini analizzando tutte le questioni connesse alla realizzazione e alla progettazione di un sistema di riconoscimento. In particolare, sono stati affrontati gli aspetti che rientrano nell'ambito dell'*Image Processing* e *Analysis* e del *Pattern Recognition*, per quanto riguarda l'acquisizione e il miglioramento delle immagini, l'estrazione da queste delle strutture da classificare, la rappresentazione di tali strutture attraverso un insieme di caratteristiche e descrittori e la selezione delle variabili con maggior potere discriminante delle varie classi di appartenenza. Infine, sono stati analizzati i vari metodi di riconoscimento, con una panoramica sugli approcci possibili e una descrizione approfondita dell'applicazioni delle Reti Neurali. In tal senso, sono state discusse due delle architetture più utilizzate nell'ambito del PR, ovvero le reti *Error Back-Propagation* e le *Self-Organizing Map*. Inoltre, sono state analizzate le composizioni gerarchiche di più moduli neurali, individuando un'architettura che risulta godere di numerosi vantaggi in termini di affidabilità e scalabilità nei problemi di riconoscimento ed interpretazione delle immagini.

Riferimenti.

[1] Robert Schalkoff. ***"Pattern Recognition. Statistical, structural and neural approaches"***. John Wiley & Sons, Inc., 1992.

[2] Britannica Online. Enclyclopaedia Britannica on the Internet, 2003 www.eb.com

[3] Anil K. Jain. ***"Fundamentals of digital image processing"***. 1989

[4] William K. Pratt. ***"Digital image processing"***. Wiley Interscience, second edition, 1991.

[5] Kenneth R. Castleman. ***"Digital image processing"***. Prentice Hall, 1996.

[6] Schurmann. "***Pattern Classification: A Unified View of Statistical and Neural Approaches"***. New York: John Wiley & Sons, 1996.

[7] R.O. Duda and P. B. Hart. ***"Pattern Classification and Scene Analysis"*** New York: John Wiley & Sons, 1973.

[8] B. D. Ripley. ***"Pattern Recognition and Neural Networks"***. Cambridge University Press, 1996.

[9] Richard P. Lippmann. "***Pattern Classification Using Neural Networks***". IEEE Communication Magazine, pp. 47-63, November 1989

[10] J. Lampinen, J. Laaksonen, E. Oja. ***"Neural Network Systems, Techniques and Applications in Pattern Recognition"***. Report B1, Laboratory of Computational Engineering, Helsinki University of Technology, 1997.

[11] N. R. Pal, S. K. Pal. ***"A review on image segmentation techniques"***. Pattern Recognition, vol. 26, pp. 1277-1294, 1993.

[12] A. Rosenfiled. ***"Connectivity in digital pictures"*** . Journal of the ACM, vol. 17, pp. 146-160, 1970

[13] K.S. Fu, J.K. Mui. ***"A survey of image segmentation"***. Pattern Recognition, vol. 13, n.1, pp. 3-16, 1981.

[14] R.M. Haralick , L.G. Shapiro. ***"Survey, image segmentation techniques"***. Computer Vision, Graphics and Image Processing, vol. 29, pp. 100-132, 1985.

[15] G. J. Awcock, R. Thomas. ***"Applied image processing"***. McGraw-Hill, 1996

[16] Weszka, A. Rosenfeld. ***"Threshold evaluation techniques"***. IEEE Transactions on Systems, Man and and Cybernetics, vol. 8, pp. 622-629, 1978.

[17] M.D.Lavine, A.M. Nazif. ***"An experimental rule based system for testing low level segmentation techniques"***. Multicomputers and Image Processing Algorithms and Programs, Academic Press, pp. 149-160, 1982.

[18] P.K. Sahoo, S. Soltani, A.K.C. Wong, Y.C. Chen. ***"A survey of thresholding techniques"***. Computer Vision, Graphics and Image Processing, vol. 41, pp. 233-260, 1988.

[19] Y.J. Zhang, J.J. Gerbrands. ***"Segmentation evaluation using ultimate measurement accuracy"***. Proc. SPIE vol. 1657, Image Processing Algorithms and Techniques III, pp. 449-460, 1992.

[20] Guoqiang Peter Zhang . ***"Neural Networks for Classification: A Survey"***. IEEE Transaction Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 30, n. 4, November 2000.

[21] B. Sankur, M. Sezgin. ***"Image Thresholding Techniques".*** A Survey over Categories. Pattern Recognition, 2001. (under review). http://www.busim.ee.boun.edu.tr/~sankur/SankurFolder/MVA_21.doc

[22] J. C. Tilton. ***"Image segmentation by iterative parallel region growing and splitting"***. Proc. 1989 International Geoscience and Remote Sensing Symposium, pp. 2235-2238, Vancouver, Canada, 1989.

[23] Rolf Adams, Leanne Bischof. ***"Seeded region growing"***. IEEE Trans. on PAMI, vol. 16, n.. 6, pp. 641 –647, June 1994.

[24] S. L. Horowitz, T. Pavlidis. ***"Picture Segmentation by a Directed Split and Merge Procedure"***. CMetImAly77, pp. 101-111, 1977.

[25] Jeffrey Wood. ***"Invariant pattern recognition: a review"***. Pattern Recognition, vol. 29 n. 1, pp. 1-17, 1996.

[26] Karsten Rodenacker, Ewert Bengtsson. ***"A feature set for cytometry on digitized microscopic images"***. Annalytical Cellular Pathology, vol. 25, pp. 1-36, 2003.

[27] Dirk Michaelis, Matthias Fröhlich, Hans Werner Strube. ***"Selection and combination of acoustic features for the description of pathologic voices"***. Acoustical Society of America, vol. 103, n. 3, 1998.

[28] S.J. Raudys and V. Pikelis. ***"On dimensionality, sample size, classification error and complexity of classification algorithms in pattern recognition"***. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 2, pp. 243-251, 1980

[29] A.K.Jain and B. Chanrasekaran. ***"Dimensionality and sample size consideration in pattern recognition practice"***. in Hanbook of Statistics, P.R. Krishnaiah and L.N. Kanal eds., vol. 2, pp. 835-855, North-Holland, Amsterdam, 1982

[30] Anil K. Jain, Robert P.W. Duin, Jianchang Mao. "***Statistical Pattern Recognition: A review***". IEEE Transcactions on Pattern Analysis and Machine Intelligence, vol.22,n.1, Gennaio 2000

[31] I. T. Jollife. ***"Principal Component Analysis"***. Springer-Verlag, New York, 1986.

[32] Simon Haykin. ***"Neural networks a comprehensive foundation".*** Prentice hall, second edition, 1999**.**

[33] Saied Sanei, Tracey K.M. Lee. ***"Cell Recognition Based On Pca And Bayesian Classification"***. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003.

[34] K.Y. Teung, W. L. Ruzzo. ***"Principal component analysis for clustering gene expression data"***. Bioinformatics, vol. 17, n. 9, pp. 763-774, 2001.

[35] George Tzanetakis, Georg Essl, Perry Cook. "***Automatic musical genre classification of audio signal***". Proc. Int. Symposium on Music Inform. Retriev. (ISMIR), pp. 205--210, October 2001

[36] Cristina Conde , Antonio Ruiz, Enrique Cabello. ***"PCA vs Low Resolution Images in Face Verification"***. IEEE Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03), 2003.

[37] Mykola Pechenizkiy, Seppo Puuronen, Alexey Tsymbal. ***"Feature Extraction for Classification in Knowledge Discovery Systems"***. Proc. 7th Int. Conf. on Knowledge-Based Intelligent Information & Engineering Systems KES'2003, University of Oxford, United Kingdom, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 2773, pp. 526-532, 2003.

[38] Fukunaga.***" Introdution to Statistical Pattern Recognition"*** . Second Ed., New York Academic Press, 1990

[39] P. Comon. ***"Independent component analysis, a new concept?"***. Signal Processing, vol. 36, n. 3, pp. 287-314, 1994

[40] M. Bartlett, S. Makeig, A. J. Bell, T-P Jung, T. J. Sejnowski. ***"Independent Component Analysis of EEG Data"***. Society for Neuroscience Abstracts, vol. 21, p. 437, 1995.

[41] Jung T-P, Makeig S, Lee T-W, McKeown M.J., Brown G., Bell, A.J. T. J. Sejnowski. ***"Independent Component Analysis of Biomedical Signals"***. The 2nd Int'l Workshop on Indeppendent Component Analysis and Signal Separation, pp. 633-44, 2000.

[42] A. J. Bell, T. J. Sejnowski, ***"The Independent Components' of Natural Scenes are Edge Filters"***. Vision Research, vol. 37, n. 23, pp. 3327-3338, 1997.

[43] K. Baek, B. A. Draper, J. R. Beveridge, K. She, ***"PCA vs ICA: A comparison on the FERET data set"***. presented at Joint Conference on Information Sciences, Durham, N.C., 2002

[44] A. Hyvärinen, E. Oja. ***"Independent Component Analysis: Algorithms and Applications"***. Neural Networks, vol. 13 n. (4-5), pp. 411-430, 2000.

[45] A. Jain, D. Zongker. ***"Feature selection: Evaluation, Application, and Small Sample Performance"***. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19 n. 2, pp. 153-158, 1997.

[46] A. L. Blum, P. Langley. ***"Selection of relevant features and examples in machine learning"***. Artificial Intelligence, vol.97, pp. 245--271, 1997.

[47] Luis Carlos Molina, Lluís Belanche, Àngela Nebot. ***"Feature Selection Algorithms: A Survey and Experimental Evaluation"***. IEEE International Conference on Data Mining pp. 306-313, 2002.

[48] Puneet Gupta, David Doermann, Daniel De Menthon. ***"Beam Search for Feature Selection in Automatic SVM Defect Classification"***. IEEE, vol. 2, pp. 212-215, 2002.

[49] P.Pudil, J. Novovicova, J.Kittler. ***"Floating search methods in feature selection".*** Pattern Recognition Letters, vol. 15, n.11, pp. 1119-1125, 1994

[50] W. Siedlecki, J. Sklansky. ***"A Note on Genetic Algorithms for Large-Scale Feature Selection"***. Pattern Recognition Letters, vol. 10, pp. 335-347, 1989.

[51] F. J. Ferri, P. Pudil, M. Hatef, J. Kittler. ***"Comparative study of techniques for large-scale feature selection"***. Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hybrid Systems, eds. E. S. Gelsema and L. S. Kanal. Amsterdam: Elsevier, pp. 403-413, 1994.

[52] Douglas Zongker, Anil Jain. ***"Algorithms for feature selection: an evaluation"***. IEEE Proceedings of ICPR, vol. 96, pp. 18-22, 1996.

[52] C.E. Shannon. "***A Mathematical Theory of Communication***" The Bell System Technical Journal, vol.27, pp.379-423, 623-656, July, October, 1948

[54] Roberto Battiti. ***"Using Mutual Information for Selecting Features in Supervised Neural Net Learning"***. IEEE Transactions on Neural Networks. Vol. 5 n. 4, pp. 537-549, July 1994

[55] W. Bialek, F. Rieke, Rob de Ruyter van Steveninck, D. Warland. ***"Reading a neural code"***. Science, vol.252, pp.1854–1857, 1991

[56] M. Hutter, M. Zaffalon. ***"Distribution of Mutual Information for Robust Feature Selection. Tech"***. rept. IDSIA-11-02. IDSIA, Manno (Lugano), CH. Submitted, 2002.

[57] Wentian Li. ***"Mutual Information Functions Versus Correlation Functions"***. Journal of Statistical Physics, vol.60, pp. 823-837, 1990.

[58] Liam Paninski. ***"Estimation of Entropy and Mutual Information"***. Neural Computation vol.15, pp. 1191–1253, 2003

[59] S. P. Strong, Roland Koberle, Rob R. de Ruyter van Steveninck, William Bialek. ***"Entropy and information in neural spike trains"***. Physical Review Letters, vol. 80, n. 1, pp. 197–202, 1998.

[60] Georgia D. Tourassi, Erik D. Frederick, Mia K. Markey, Carey E. Floyd. ***"Application of the mutual information criterion for feature selection in computer-aided diagnosis"***. Med. Phys, vol. 28, n. 12, pp. 2394-2401, 2001.

[61] H.H. Yang, S. Van Vuuren, S. Sharma, H. Hermansky. ***"Relevance of time frequency features for phonetic and speaker-channel classification"***. Speech Commun, vol. 31, pp. 35-50, 2000.

[62] Mohamed Kamal Omar, Ken Chen, Mark Hasegawa-Johnson, Yigal Brandman. ***"An Evaluation of Using Mutual Information for Selection of Acoustic Features Representation of Phonemes for Speech Recognition"***. Int. Conference on Spoken Language Processing, Denver, CO, Sept. 2002.

[63] K. S. Fu. ***"A step towards unification of syntactic and statistical pattern recognition"***. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 5, n. 2, pp. 200-205, March 1983

[64] A. Jain, Y. Zhong, and S. Lakshmanan. ***"Object matching using deformable templates".*** IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18, n. 3, pp. 207-277, Mar. 1996.

[65] A. Jain, Y. Zhong, and M. Dubuisson-Jolly. "***Deformable template models: A review".*** Signal Processing, vol. 71, pp. 109-129, 1998.

[66] Hiromichi Fujisawa and Cheng-Lin Liu." ***Directional Pattern Matching for Character Recognition Revisited***". Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)

[67] Richard P. Lippmann, ***"A critical overview of neural network pattern classification"***. Neural Networks for Signal Processing, IEEE Workshop, pp. 266-275, 1991.

[68] N.R. Pal, V.K. Eluri. ***"Two efficient connectionist schemes for structure preserving dimensionality reduction"***. Neural Networks, IEEE Transactions on , vol. 9, Issue: 6, pp. 1142 –1154, Nov 1998.

[69] Bishop. ***"Nevral Networks for Pattern Recognition"***. Oxford; Clarendon Press, 1995.

[70] Lasse Holmstrom, Petri Koistinen, Jorma Laaksonen, Erkki Oja. ***"Neural Network and Statistical Perspectives of Classification"***. IEEE Proceeding of ICPR, pp. 286-290, 1996.

[71] Dan W. Patterson. ***"Artificial Neuronal Networks. Theory and Application"***. Prentice hall. 1996.

[72] L. Cordella, C. Sansone, F. Tortorella, M. Vento, C. De Stefano. "***Neural network classification reliability: problems and applications***" in Image Processing and Pattern Recognition, Acaddemic Press 1998, pp. 161-199

[73] D.E. Rumelhart, G.E. Hinton, R.J. Williams. "***Learning internal representations by error propagation***" Parallel Distribuited Processing pp. 318-362 MIT Press, Cambridge, MA, 1986.

[74] D.S. Broomhead, D. Lowe. "***Multivariable functional interpolation and adaptive networks***" Complex System Vol. 2, pp. 321-355,1988

[75] T. Kohonen. "***The self-organizing maps***". Proceeding of the Institute of Electrical and Electronics Engineers, vol 78, pp. 1464-1480, 1990.

[76] T. Kohohnen. "***Self-Organization and Associative Memory"***. Springer-Verlag, New York, 1984.

[77] S. Grossberg. ***"Competitive Learning: From Interactive Activation to Adaptive Resonance"***. Cognitive Science, vol. 11, pp. 23-63, 1987.

[78] D.F. Specht. "***Probabilistic Neural Networks***" Neural Networks Vol. 3, pp. 109-118,1990

[79] J.J. Hopfield. " ***Neural networks and physical systems with emergent collective abilities***" Proc. Nat. Acad. Science USA, vol. 79, pp. 2554-2558,1982

[80] B. Widrow, M.E. Hoff. ***"Adaptive switching circuits"***. IRE WESCON Convention Record, pp. 96-104, 1960.

[81] F. Rosenblatt. ***"The perceptron: A probabilistic model for information storage and organization in the brain"***. Psychological Review, vol. 65, pp. 386-408, 1958.

[82] J. Park, I. W. Sandberg. ***"Universal approximation using radial-basis function networks"***. Neural Computation, vol. 3, pp. 246-257, 1991.

[83] A.R. Barron. ***"Universal approximation bounds for superposition of sigmoid functions"***. IEEE Trans. Information Theory, vol. 39, pp. 930-945, 1993.

[84] Lippmann, Richard P., "***Neural Nets for Computing***" IEEE pp. 1-9, April 1988.

[85] E. Wan. ***"Neural network classification: A Bayesian interpretation"***. IEEE Trans. Neural Networks, vol. 1, n. 4, pp. 303–305, 1990.

[86] P. A. Shoemaker. ***"A note on least-squares learning procedures and classification by neural network models"***. IEEE Trans. Neural Networks, vol. 2, pp. 158–160, 1991.

[87] M. D. Richard, R. Lippmann. ***"Neural network classifiers estimate Bayesian a posteriori probabilities"***. Neural Comput., vol. 3, pp. 461–483, 1991.

[88] H. Gish. ***"A probabilistic approach to the understanding and training of neural network classifiers"***. in Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing, pp. 1361–1364, 1990.

[89] Guoqiang Peter Zhang . ***"Neural Networks for Classification: A Survey"***. IEEE Transaction Systems, Man, and Cybernetics – Part C: Applications and Reviews, vol. 30, n. 4, November 2000.

[90] R. P. Lippmann, P. E. Beckman. ***"Adaptive Neural-Net Processing for Signal Detection in Non-Gaussian Noise"***. Advances Neural Info.

Processing Syst., 1989.

[91] P. Gallinari, S. Thiria, R. Badran, F. Fogelman-Soulie. ***"On the relationships between discriminant analysis and multilayer perceptrons"***. Neural Networks, vol. 4, pp. 349–360, 1991.

[92] H. Asoh, N. Otsu. ***"An approximation of nonlinear discriminant analysis by multilayer neural networks"***. in Proc. Int. Joint Conf. Neural Networks, San Diego, CA, pp. III-211–III-216, 1990.

[93] A. R.Webb, D. Lowe. ***"The optimized internal representation of multilayer classifier networks performs nonlinear discriminant analysis"***. Neural Networks, vol. 3, n. 4, pp. 367–375, 1990.

[94] T. Kohoonen, G. Barna, and R. Chrisley. ***" Statistical Pattern Recognition With Neural Networks Bench Marking Studies"*** . IEEE Annual Int. I. Conf. on Neural Networks, San Diego, July 1988.

[95] S. P. Curram and J. Mingers. ***"Neural networks, decision tree induction and discriminant analysis: An empirical comparison"***. J. Oper. Res. Soc., vol. 45, n. 4, pp. 440–450, 1994.

[96] G. S. Lim, M. Alder, P. Hadingham. ***"Adaptive quadratic neural nets"***. Pattern Recognit. Lett., vol. 13, pp. 325–329, 1992.

[97] D. Michie, D. J. Spiegelhalter, C. C. Taylor (eds.). ***"Machine Learning, Neural, and Statistical Classification"***. London, U.K.: Ellis Horwood, 1994.

[98] V. Kurkova. "***Kolmogorov's theorem***". In Michael A. Arbib, editor, The Handbook of Brain Theory and Neural Networks, pages 501--502. MIT Press, Cambridge, Massachusetts, 1995

[99] E. B. Baum, D. Hausler, ***"What Size Net Gives Valid Generalization?"***. Neural Computation, vol. 1, pp. 151-160, 1989.

[100] J.M.J. Murre, R.H. Phaf, G. Wolters. ***"CALM: Categorizing and Learning Module"***. Neural Networks, vol. 5, pp. 55-82, 1992.

[101] McCloskey, M. & Cohen, N.J. ***"Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem"***. The Psychology of Learning and Motivation vol. 24, pp. 109-165, 1989

[102] S. Grossberg. ***"Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control"***. Boston, MA: Reidel Press, 1982.

[103] N. Morgan, H. Bourlard. ***"Generalization and parameter estimation in feedforward nets: Some experiments"***. Adv. Neural Inform. Process. Syst., vol. 2, pp. 630–637, 1990.

[104] R. Reed. ***"Pruning algorithms—A survey"***. IEEE Trans. Neural Networks, vol. 4, pp. 740–747, Sept. 1993.

[105] A.Weigend, D. Rumelhart, and B. Huberman. ***"Predicting the future: A connectionist approach"***. Int. J. Neural Syst., vol. 3, pp. 193–209, 1990.

[106] T.G. Dietterich. ***"Overfitting and undercomputing in machine learning"***. Comput. Surv., vol. 27, n. 3, pp. 326–327, 1995.

[107] N. Qian, T. J. Sejnowski. ***"Predicting the Secondary Structure of Globular Proteines using Neural Network Models"***. J. Molecular Biology, vol. 202, pp. 865-884, 1988.

[108] S. Tamura, A. Waibel. ***"Noise Reduction using Connectionist Models"***. Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 553-556, 1988.

[109] Ching-Piao Tsai, Tsong-Lin Lee ***"Back-Propagation Neural Network in Tidal-Level Forecasting"***Journal of Waterway, Port, Coastal and Ocean Engineering, Vol. 125, No. 4, pp. 195-20 July/August 1999

[110] Saeed Moshiri, Norman E Cameron, David Scuse." ***Static, Dynamic, and Hybrid Neural Networks in Forecasting Inflation***". Computational Economics, Kluwer Academic Pubblishers, vol. 14 (3) pp. 219-35 Dicember 1999

[111] Richard P. Lippmann. ***"Review of Neural Networks for Speech Recognition"***. Neural Comp., vol. 1, n. 1, pp. 1-38, 1989.

[112] Marcelo N. Kapp, Cinthia O. De A. Freitas, Júlio C. Nievola." ***Evaluating the Conventional and Class-Modular Architectures Feedforward Neural Network for Handwritten Word Recognition***". XVI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'03) October 12 - 15, 2003 São Carlos, Brazil

[113] M. Nakamura, K. Maruyama, T. Kawabata, K. Shikano. "***Neural network approach to word category prediction for English texts***". In Proceedings of the International Conference on Computational Linguistics. 213—218, 1990.

[114] J. P. Kerr, E. B. Bartlett. ***"A statistically tailored neural network approach to tomographic image reconstruction"***. Med. Phys., vol. 22, pp. 601-610, 1995.

[115] H. Yan, J. Mao. ***"Data truncation artefact reduction in MR imaging using a multilayer neural neural network"***. IEEE Trans. Med. Imaging, vol. 12, pp. 73-77, 1993

[116] U. Raff, A. L. Scherzinger, R. F. Vargas, J. H. Simon. ***"Quantitation of grey matter, white matter, and cerebrospinal fluid from spin-echo magnetic resonance images using an artificial neural network technique"***. Med. Phys., vol. 21, pp. 1933-1942, 1994.

[117] E. R. Kischell, N. Kehtarnavaz, G. R. Hillman, H. Levin, M. Lilly, T. A. Kent. ***"Classification of brain compartments and head injury lesions by neural network applied to MRI"***. Neuroradiology, vol. 37, pp. 535-541, 1995.

[118] B. Zheng, W. Qian, L. P. Clarke. ***"Digital mammography: Mixed feature neural network with spectral entropy decision form detection***

of microcalcifications". IEEE Trans. Med. Imaging, vol. 15, pp. 589-597, 1996.

[119] Y. Sun. ***"On quantization error of self-organizing map network"***. Neurocomputing, vol. 34, pp. 169-193, 2000.

[120] Juha Vesanto, Mika Sulkava, Jaakko Hollmen. ***"On the Decomposition of the Self-Organizing Map Distortion Mearure"***. Helsinki University of Technology, Laboratory of Computer and Information Science, pp. 11-16, 2003.

[121] T. Kohonen. "***Self-Organizing Maps***". Springer Series in information Sciences, second edition, vol. 30 Berlin Springer, 1997.

[122] A. Ultsch, C. Vetter. ***"Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark"***. University of Marburg, FG Neuroinformatik & Kuenstliche Intelligenz, Research Report 0994, 1994.

[123] A. Ultsch, H. P. Siemon. ***"Kohonen's self organizing feature maps for exploratory data analysis"***. In Proc. Int. Neural Network Conf. Dordrecht, The Netherlands, pp. 305-308, 1990.

[124] M. A. Kraaijveld, J. Mao, A. K. Jain. ***"A nonlinear projection method based on Kohonen's topology preserving maps"***. IEEE Trans. Neural Network, vol. 6, pp. 548-559, May 1995.

[125] Simone Marinai, Emanuele Marino, Giovanni Soda. ***"Seventh International Conference on Document Analysis and Recognition Volume I"***. Edimburgh, Scotland, Agoust 2003.

[126] M. C. Benitez, A. Rubio, P. Garcia, A. de la Torre. ***"Different confidence measures for word verification in speech recognition"***. Speech Communication, vol. 32, n. 1, pp. 79–94, 2000.

[127] Ugur Halici, Guclu Ongun. ***"Fingerprint Classification Through Self Organizing Feature Maps Modified to Treat Uncertainties"***. Proceedings of the IEEE, vol. 84, n. 10, pp. 1497-1512, October 1996.

[128] N. M. Allinson, H. Yin. ***"Self-organising maps for pattern recognition"***. In Oja, E. and Kaski, S., editors, Kohonen Maps, pp. 111–120, Elsevier, Amsterdam, 1999.

[129] J. S. Baras, S. Dey. ***"Adaptive classification based on compressed data using learning vector quantization"***. In Proceedings of the 38th IEEE Conference on Decision and Control, vol. 4, pp. 3677–3683, Piscataway, NJ. IEEE Service Center, 1999.

[130] J. S. Baras, S. Dey. ***"Combined compression and classification with learning vector quantization"***. IEEE Transactions on Information Theory, vol. 45, pp. 1911–1920, 1999.

[131] S. Di Bona e O. Salvetti. ***"Approch for 3d volumes matching. In Proc. ISETI SPIE: 13th International Symposium on Electronic Imaging 200"***. Artificial Neural Networks in Image Processing, vol. 4305, San Jose, CA, USA, January, pp.20-26, 2001.

[132] S. Di Bona e O. Salvetti. ***"A neural method for three-dimensional image matching"***. Journal of Electronical Imaging vol. 11, n. 4, pp. 1-10, Oct. 2002.

[133] T. Kohonen, K. Torkkola, M. Shozakai, J. Kangas, O. Venta. ***"Phonetic Typewriter for Finnish and Japanese"***. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88), pp. 607-- 610, New York City, USA, April 1988.

[134] D. L. James, R. Miikkulainen. ***"SARDNET: a selforganizing feature map for sequences"***. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, Proceedings of the Advances in Neural Information Processing Systems, vol. 7. pp. 1-9, Morgan Kaufmann, 1995.

[135] Raúl Saavedra. "***Self-Organizing Map as a clustering tool in monitoring deterioration of three-phase induction motors***". *A Thesis submitted the 24 of March 1999, to the Department of Electrical Engineering and Computer Science of the school of Engineering of Tulane University, in partial fulfillment of the requirements for degree of Master of Science in Computer Science.* Copyright 1999 by Raúl Saavedra

[136] K. Lagus, T. Honkela, D. Kaski, T. Kohonen. ***"WEBSOM for textual data mining"***. Artificial Intelligence Review, vol. 13, issue 5/6, pp. 345-364, 1999

[137] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela. ***"Self organization of a massive text document collection"***. IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, vol. 11, pp. 574-585, 2000.

[138] Udo Seiffert, Bernd Michaelis. "***Three-dimensional Self-Organizing Maps for Classification of Image properties***". *Institute for Measurement Technology and Electronics, Magdeburg, Germany. www: http://ipe.et.uni-magdeburg.de.*

[139] S. R. Waterhouse, A. J. Robinson. ***"Classification using Hierarchical Mixtures of Experts"***. IEEE Workshop on Neural Networks for Signal Processing IV, pp. 177-186, 1994.

[140] Z. H. Zhou, J .Wu, W. Tang. ***"Ensembling neural networks: many could be better than all"***. Artificial Intelligence, vol. 137, n. 1-2, pp. 239-263, 2002.

[141] M. P. Perrone, ***"Putting it all together: Methods for combining neural networks"***. in Advances in Neural Information Processing Systems, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. San Mateo, CA: Morgan Kaufmann, vol. 6, pp. 1188–1189, 1994.

[142] S. Hashem, B. Schmeiser. ***"Improving model accuracy using optimal linear combinations of trained neural networks"***. IEEE Trans. Neural Networks, vol. 6, n. 3, pp. 792–794, 1995.

[143] L. K. Hansen and P. Salamon. ***"Neural network ensembles"***. IEEE Trans. Pattern Anal. Machine Intell., vol. 12, n. 10, pp. 993–1001, 1990.

[144] D. N. Osherson, S. Weinsten, M. Stoli. ***"Modular Learning"***. Computational Neuroscience, E.L. Schwartz, ed., pp. 369-377, 1990.

[145] Dean W. Abbott. ***"Combining Models to Improve Classifier Accuracy and Robustness"***. Presented at the 1999 International Conference on Information Fusion— Fusion99, Sunnyvale, CA, July 6-8, 1999. http://www.abbott-consulting.com/pubs/fusion99_abbott_dist.pdf

[146] R.E. Schapire. ***"The strength of weak learnability"***. Machine Learning, vol. 5, n. 2, pp. 197-227, 1990.

[147] R.E. Schapire. ***"A brief introduction to boosting"***. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.

[148] R. Meir and G. Rätsch. ***"An introduction to boosting and leveraging"***. In S. Mendelson and A. Smola, editors, Advanced Lectures on Machine Learning, LNCS, pp. 119-184. Springer, 2003.

[149] R. A. Jacobs, M. I. Jordon, S. J. Nowlan, G. E. Hinton. ***"Adaptive mixtures of local experts"***. Neural Computation, vol. 3, pp. 79--87, 1991

[150] M. I. Jordan, R. A. Jacobs. ***"Hierarchical mixtures of experts and the EM algorithm"***. Neural Computation, vol. 6, pp. 181-214, 1994.

[151] R. K. Speer, W. E. Moore. ***"Hierarchical Artificial Neural Network Architecture"***. 0-7803-4859-1/98©1998 IEEE

[152] L. Srivastava, S. N. Singh, J. Sharma. ***"Parallel self-organizing hierarchical neural network-based fast voltage estimation"***. IEEE Proc. Gener. Transm. Distrib., vol. 145, n. 1, pp. 98-104, January 1998.

[153] T. Paulpandian, V. Ganapathy. ***"Translation and Scale Inveriant Recognition of Handwritten Tamil Characters Using a Hierarchical Neural Network"***. 0-7803-1254-6/93 © 1993 IEEE.

[154] Hui Su, Wei Wang, Xinyou Li, Shaowei Xia. ***"Hierarchical Neural Network for Recognizing Hand-written Characters in Engineering Drawings"***. 0-8186-7128-9/95 © 1995 IEEE.

[155] Murat A. Ozbayoglu, Cihan H. Dagli, Bill Fulkerson. ***"A Hierarchical Neural Network Implementation for Forecarting"***. 0-7803-1901-X/94 © 1994 IEEE.

[156] Sung Joo Park, Jin Seol Yang. ***"A Hierarchical Neural Network Approach to Intelligent Traffic Control"***. 0-7803-1901-X/94 © 1994 IEEE

[157] Javier Herrero, Alfonso Valencia, Joaquìn Dopazo. "***A hierarchical unsupervised growing neural network for clustering gene expression patterns***". Bioinformatics, vol. 17, n. 2, pp. 126-136, 2001.

[158] Bong-Su Kang, Sung-Il Chien, Kil-Taek Lim, Jin-Ho Kim. ***"Large Scale Pattern Recognition System using Hierarchical Neural Network and False-Alarming Nodes"***. 1082-3409/97 © 1997 IEEE.

[159] L. Vladutu, S. Papadimitriou, S. Mavroudi, A. Bezerianos. ***"Ischemia Detection using Supervised Learning for Hierarchical Neural Networks based on Kohonen-maps"***. 2001 Proceeding of the 23rd Annual EMBS International Conference, pp. 1688-1691, 2001.

[160] Paul Sajda, Clay Spence, John Pearson. ***"A Hierarchical Neural Network Architecture That Learns Target Context: Applications to Digital Mammography"***. International Conference on Image Processing, vol. 3, pp.149-151, 1995.

[161] S. Keem, H. Meadows, H. Kemp. ***"Hierarchical Neural Networks in Quantitative Coronary Arteriography"***. In: Proceedings of the ath International Conference on Artificial Networks—ANN'95. London, UK, pp. 459-464, 1995.

[162] Sergio Di Bona, Heinrich Niemann, Gabriele Pierri, Ovidio Salvetti. ***"Brain volumes characterisation using hierarchical neural networks"***. Artificial Intelligence in Medicine vol. 28, pp. 307-322, 2003.