Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Corso di Laurea Magistrale in Fisica

Segmentazione di Immagini Mammografiche con Convolutional Neural Networks

Relatore:
Prof. Renato Campanini

Correlatore:
Dott. Massimiliano Grandi

Presentata da:
Mattia Neri

Sessione III
Anno Accademico 2012/2013

Abstract

Il tumore al seno si colloca al primo posto per livello di mortalità tra le patologie tumorali che colpiscono la popolazione femminile mondiale [2-I]. Diversi studi clinici hanno dimostrato come la diagnosi da parte del radiologo possa essere aiutata e migliorata dai sistemi di *Computer Aided Detection (CAD)*.

La ricerca automatizzata delle lesioni tumorali è un problema estremamente complicato: le difficoltà principali riguardano la grande variabilità di forma e dimensioni delle masse e la somiglianza di queste con i tessuti che le ospitano.

La maggior parte dei sistemi di CAD è composta da due livelli di classificazione: (a) la *detection*, responsabile dell'individuazione delle regioni sospette presenti sul mammogramma (*region of interest - ROI*) e quindi dell'eliminazione preventiva delle zone non a rischio; (b) la classificazione vera e propria (*classification*) delle ROI in masse e tessuto sano.

Lo scopo principale di questa tesi è lo studio di nuove metodologie di detection che possano migliorare le prestazioni ottenute con le tecniche tradizionali. In particolare si vuole implementare un algoritmo di detection alternativo a quello attualmente usato dal software di CAD Galileo [1-I], per vedere se sia possibile sostituirlo o integrare i due sistemi.

Si considera la detection come un problema di apprendimento supervisionato e lo si affronta mediante le *Convolutional Neural Networks (CNN)*, un algoritmo appartenente al *deep learning*, nuova branca del *machine learning* (una delle aree fondamentali dell'*intelligenza artificiale*). Le CNN si ispirano alle scoperte di Hubel e Wiesel [31] riguardanti due tipi base di cellule identificate nella corteccia visiva dei gatti: le cellule semplici (S) e le cellule complesse (C) . Le cellule S rispondono a stimoli simili ai bordi mentre le cellule C sono localmente invarianti all'esatta posizione dello stimolo. In analogia con la corteccia visiva, le CNN utilizzano un'architettura profonda caratterizzata da un'alternanza di strati di convoluzione e *subsampling*. Le CNN vengono usate per problemi di riconoscimento automatico di *pattern* bidimensionali come la rivelazione di oggetti, facce e loghi nelle immagini o l'analisi di documenti. La scelta di questo algoritmo è dovuta al fatto che le performance ottenute in alcuni problemi sono risultate superiori rispetto a quelle ottenute con altri metodi [25] com'è accaduto, ad esempio, nelle edizioni del 2010 e del 2012 dell'ImageNet Large-Scale Visual Recognition Challenge, una competizione annuale riguardante il riconoscimento di immagini di varia natura (circa 1000 immagini per ognuna delle 1000 categorie).

La struttura della tesi è la seguente. Nel capitolo 1 si introduce la problematica del tumore al seno e vengono descritte la mammografia digitale ed i sistemi di CAD. Il

capitolo 2 descrive il machine learning, il passaggio al deep learning e l’algoritmo delle convolutional neural networks. Nel capitolo 3 viene descritto il lavoro svolto: la creazione del dataset, la selezione del modello, la fase di addestramento del sistema e lo schema di detection delle masse. Nel capitolo 4 si espongono i risultati ottenuti e si confrontano con quelli della letteratura.

Contenuti:

Introduzione

Perchè Deep Learning? Motivi, obiettivi e nuovi contributi

Il *Deep Learning* fa parte della famiglia più ampia dei metodi del *Machine Learning*, una branca dell'*Intelligenza Artificiale (Artificial Intelligence - AI)*. Un'osservazione (e.g., un'immagine), può essere rappresentata in molti modi (e.g., un vettore di pixel) ma alcune rappresentazioni rendono più semplice l'apprendimento di compiti tramite esempi (e.g., questa immagine contiene un volto umano?): le ricerche nel settore del machine learning cercano di capire cosa rende le rappresentazioni migliori e come creare modelli per "impararle".

I metodi del deep learning nascono dalla volontà di imitazione delle dinamiche che caratterizzano l'attività della corteccia visiva umana: essa sembra comportarsi come una gerarchia di filtri dove, a partire dai dati sensoriali, ogni strato cattura alcune delle informazioni provenienti dallo strato precedente passandole, trasformate, ai livelli successivi. Sostanzialmente il deep learning consiste in un insieme di algoritmi che cercano di modellare i dati autonomamente tramite un'architettura profonda, ossia composta da diversi livelli di astrazione, dove i concetti ai livelli più alti sono definiti da quelli ai livelli più bassi tramite una successione di trasformazioni non-lineari.

Recentemente queste tecniche sono state in grado di raggiungere performance superiori a quelle di altri metodi in un'ampia varietà di compiti riguardanti la classificazione di immagini, il riconoscimento e l'elaborazione del linguaggio naturale [25]. Sempre più organizzazioni infatti si stanno interessando al suo utilizzo, ingaggiando i principali "guru" del settore: nel marzo 2013 Geoffrey Hinton, professore di computer science all'università di Toronto, è stato assunto da Google e nel dicembre 2013 Facebook ha annunciato di aver ingaggiato Yann LeCun, professore di computer science all'università di New York, per dirigere il suo nuovo laboratorio di Intelligenza Artificiale.

In un articolo del 1996 [11], Heang-Ping Chan espone risultati incoraggianti ottenuti tramite le *Convolutional Neural Networks*, un algoritmo appartenente al deep learning, nel problema della classificazione di immagini di tessuti sani e tessuti contenenti masse provenienti da mammografie digitali. L'obiettivo principale di questa tesi è quello di proseguire e sviluppare il lavoro descritto nell'articolo con l'utilizzo di un'architettura più profonda (composta da più livelli di rappresentazione) e alla luce delle nuove tecniche già applicate alle CNN nello studio di altri problemi.

L'architettura delle CNN solitamente alterna strati di *convoluzione*, in cui le immagini vengono trasformate con dei filtri (*kernel*), a strati di *subsampling*, e si conclude con una rete neurale classica che esegue la classificazione finale. L'operazione di subsampling ordinaria consiste nella partizione dell'immagine in un

set di rettangoli non sovrapposti seguita dalla sostituzione di ogni sub-regione con la sua media. In un articolo del 2009 Jianchao Yang et al. [24], pur non fornendo una giustificazione teorica, sostengono che le performance di classificazione riguardanti diversi oggetti e scene migliorano se l'operazione di media viene sostituita da quella di massimo.

Per i compiti di classificazione, molti di questi modelli di DL usano, all'ultimo strato della gerarchia, una rete neurale tradizionale con funzione di attivazione di tipo *softmax*, conosciuta anche come *multinomial logistic regression*. Nell'articolo *Deep Learning using Linear Support Vector Machine* di Yichuan Tang [13] viene mostrato come, la sostituzione della rete neurale con una *support vector machine (SVM)* porti a miglioramenti significativi delle performance su diversi datasets popolari come il *MNIST* (database pubblico di cifre scritte a mano), il *CIFAR-10* (10 classi di immagini varie prese da internet) e *ICML 2013* (che riguarda il riconoscimento delle espressioni facciali).

In questa tesi si sono quindi voluti sperimentare entrambi questi approcci innovativi alle CNN, nella speranza di osservare dei miglioramenti nelle performance. A differenza di Chan, che usa una CNN ad un solo livello formato da uno strato di convoluzione e uno di subsampling, noi aumentiamo la profondità introducendo un secondo livello.

Capitolo 1

Mammografia Digitale e CAD

1.1 Tumore al seno

Il tumore al seno colpisce 1 donna su 8 nell'arco della vita. È il tumore più frequente nel sesso femminile e rappresenta il 29% di tutti i tumori che colpiscono le donne. È la prima causa di mortalità per tumore nel sesso femminile, con un tasso di mortalità del 16% di tutti i decessi per causa oncologica [2-I].

La possibilità di guarigione è fortemente dipendente dal grado di sviluppo della patologia nel momento in cui viene diagnosticata: se la malattia viene individuata nella sua fase iniziale, quando le dimensioni della lesione sono ancora ridotte, le possibilità di guarigione sono alte. Si stima che, in questi casi, la sopravvivenza a cinque anni nelle donne trattate è del 98% [2-I]. Ma la completa assenza di sintomi, se non in fase avanzata, rende la diagnosi estremamente difficile. Diventa quindi necessario adottare una politica di prevenzione e diagnosi precoce con controlli periodici.

In caso di tumore di grado avanzato le terapie adottate sono sempre di tipo demolitivo, come l'intervento di asportazione della mammella, seguiti da pesanti trattamenti di radioterapia e chemioterapia, con drammatiche conseguenze psico-fisiche per la paziente. Un tumore allo stadio iniziale, invece, spesso necessita solo di interventi di tipo conservativo, come l'asportazione del nodulo o di una piccola parte della mammella. Ecco quindi che la diagnosi precoce acquisisce un'importanza fondamentale anche da un punto di vista terapeutico.

1.2 Mammografia Digitale

La mammografia digitale è una tecnica a raggi-X che consente di avere una visione interna della mammella con una buona risoluzione spaziale e un alto contrasto.
Un fascio di raggi-X, emesso da un apparecchio radiologico detto mammografo, attraversa il seno e viene assorbito in maniera diversa a seconda del tipo di tessuto che incontra. I raggi rimanenti vengono registrati da un sensore digitale che li converte in potenziale elettrico. Quello che ne risulta è un immagine digitale in scala di grigi, che può essere trasferita numericamente su un computer e visualizzata su uno schermo, rappresentante la struttura interna di una mammella.

La struttura interna del seno può variare notevolmente da soggetto a soggetto e di conseguenza anche il suo aspetto in una mammografia. Se la mammella è formata principalmente da tessuto fibroso la mammografia risulterà luminosa (in questo caso si parla di mammella densa), se invece la mammella è prevalentemente formata da tessuto grasso la mammografia apparirà più scura. Si viene quindi a creare una corrispondenza proporzionale fra l'intensità di colore e la densità del tessuto corrispondente (Figura 1.1).

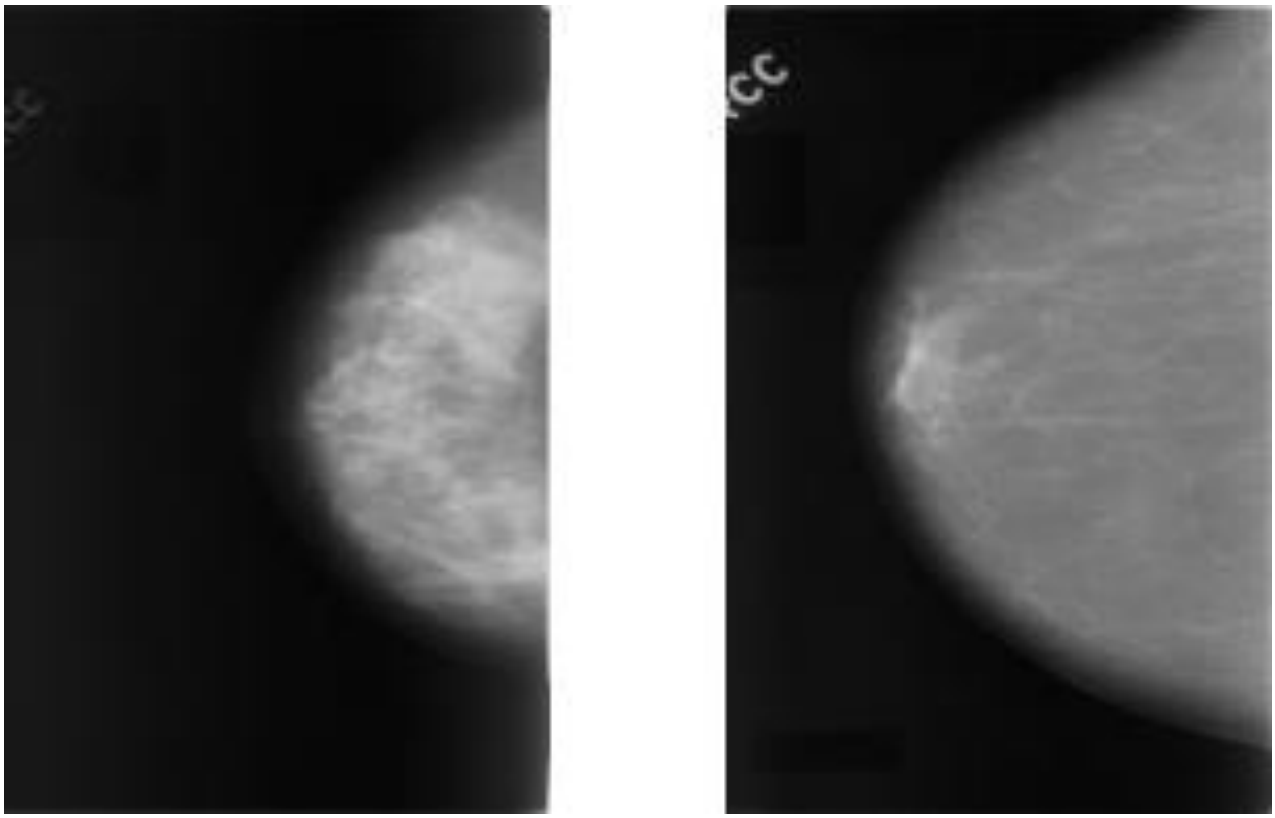


Figura 1.1 mammella densa (sinistra), mammella grassa (destra)

Uno screening di base consiste in due proiezioni per ogni mammella: una lungo l'asse cranio-caudale (CC) ed una obliqua medio-laterale (MLO), inclinata a 45° (Fig. 1.2).

La tecnica tradizionale per la rivelazione del carcinoma mammario è l'analisi, da parte di un radiologo, dell'immagine prodotta sullo schermo. Le lesioni presenti nel seno hanno un grado di assorbimento caratteristico, per questo nella maggior parte dei casi per il radiologo è relativamente semplice individuarle. Tuttavia, nonostante la ricerca abbia migliorato notevolmente le tecniche nel corso negli anni, si stima che sulla totalità delle lesioni neoplastiche mammarie non vengano diagnosticati fra il 10% e il 30% dei casi [32]. I fattori che contribuiscono ad una percentuale di errore così elevata sono diversi: primo fra tutti la natura stessa della massa, caratterizzata da

una grande variabilità di caratteristiche morfologiche; inoltre la sua individuazione è resa difficile dalla somiglianza col tessuto ospitante (Figura 1.3 e 1.4). Grossi problemi si hanno in genere in mammelle di pazienti giovani, nelle quali si presenta un tessuto fibroso molto denso, e quindi un basso contrasto tra le varie zone dei tessuti.

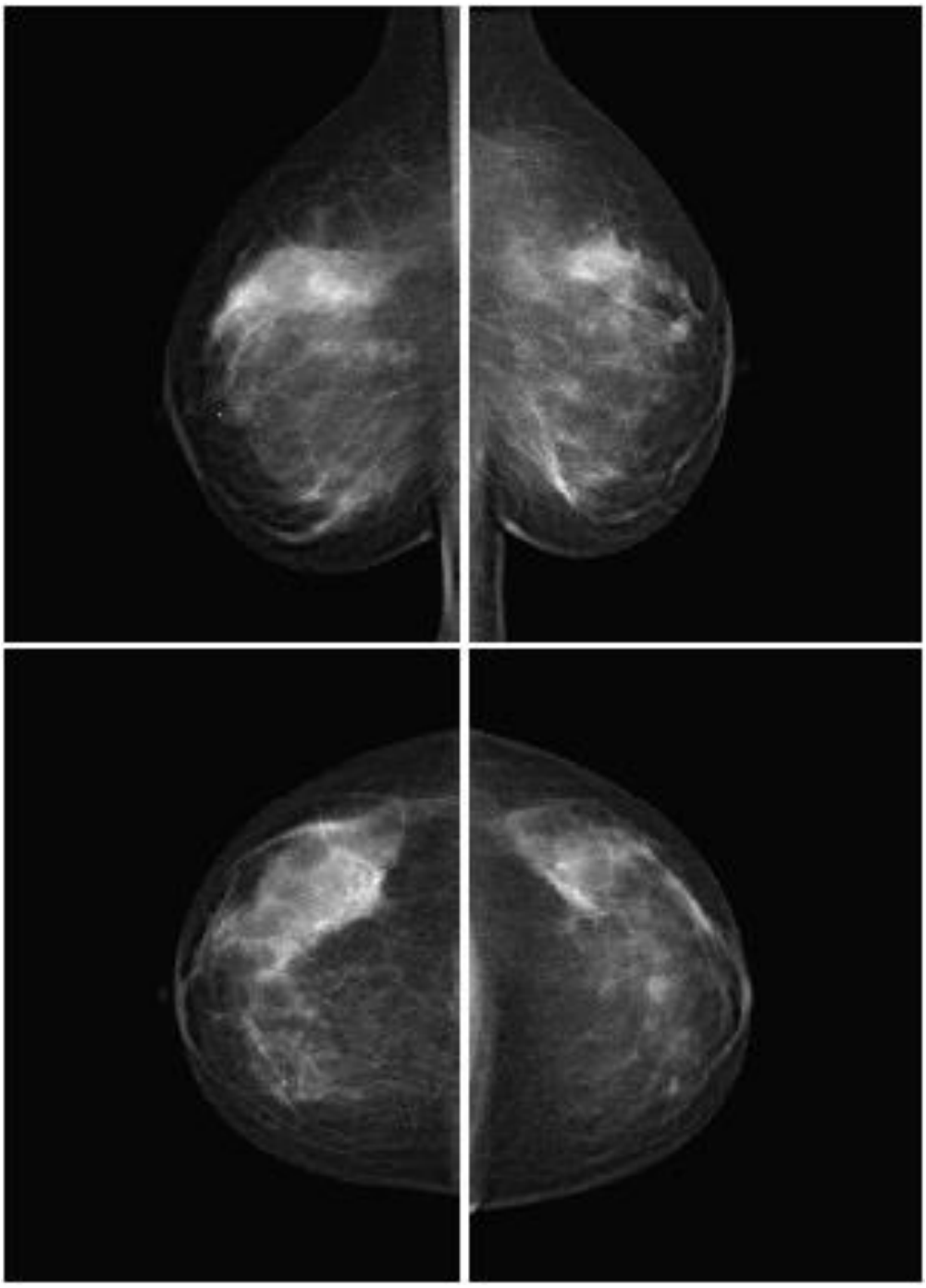


Figura 1.2 diverse proiezioni di un esame di screening

L'ultimo stadio del processo è il verdetto del radiologo stesso. Qui, oltre a quelli sopraindicati, subentrano i fattori di imperfezione umana come, ad esempio, l'affaticamento degli occhi dovuto a lavoro troppo intenso, il fatto che si presti maggiore attenzione alle parti dell'immagine nelle quali si pensa sia più probabile trovare masse o l'intrinseca difficoltà di interpretazione delle immagini.

Si distinguono due tipi di errori:

- falso negativo *(false negative - FN)*, che si verifica quando una mammografia contenente un qualche tipo di lesione viene erroneamente classificata come normale.
- falso positivo *(false positive - FP)* che viene commesso quando in una normale mammografia vengono segnalate lesioni che, invece, non sono presenti.

L'errore chiaramente più grave è quello di tipo falso-negativo, in quanto provoca un ritardo nella diagnosi e nella cura della patologia, che potrebbe compromettere irrimediabilmente la salute della paziente. Un errore di tipo falso-positivo, sebbene non comprometta la probabilità di sopravvivenza della paziente sicuramente comporta danni psicologici non trascurabili.

Studi clinici hanno dimostrato come l'analisi delle mammografie possa essere supportata dai sistemi detti di *Computer Aided Detection (CAD)*.

Figura 1.3 Masse tumorali

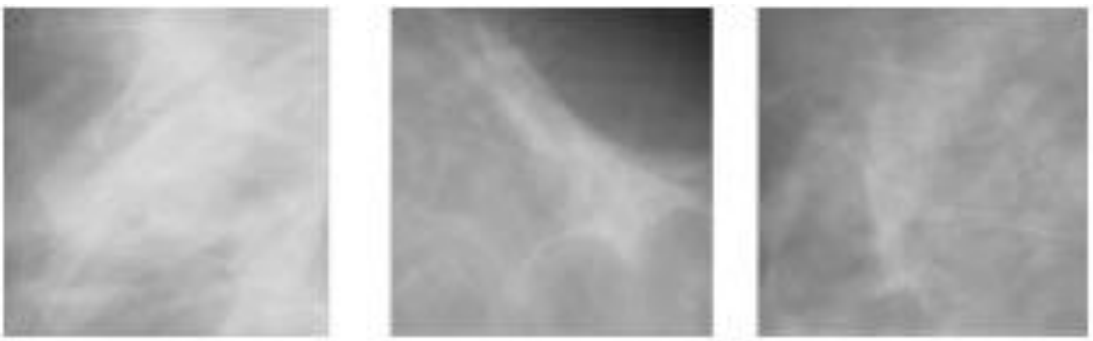


Figura 1.4 Tessuti sani

1.3 Computer Aided Detection - CAD

Vista la fondamentale importanza di una diagnosi precoce e le difficoltà riscontrate dal radiologo, col tempo sono stati introdotti meccanismi che lo possano in qualche modo agevolare, riducendo le possibilità di errore.

All'inizio degli anni '70 è stato introdotto l'uso del computer nella mammografia per il miglioramento dell'immagine (aumento del contrasto, filtri, eccetera…). Tuttavia la mancanza di un sistema di acquisizione di immagini digitali rendeva necessaria la fase di scannerizzazione: questo introduce rumore ed artefatti nelle immagini e comporta una perdita di informazioni. Poi sono state sviluppate tecniche di analisi nelle quali venivano impiegati due radiologi: entrambi facevano la loro diagnosi indipendentemente e se ne discutevano poi i risultati.

Successivamente, l'avvento della mammografia digitale e il successo del metodo di doppia analisi hanno portato ad una sua evoluzione: la sostituzione di uno dei due radiologi con un sistema di rilevamento automatico detto appunto CAD. È importante sottolineare che la sentenza del CAD non è da intendersi esaustiva, ma solo indicativa al radiologo delle regioni sospette. In questo modo lo specialista avrà la possibilità di fare una più attenta valutazione delle suddette regioni.

La maggior parte dei sistemi di CAD è composta da due livelli di classificazione: (a) la *detection*, responsabile dell'individuazione delle regioni sospette presenti sul mammogramma (*Region of Interest - ROI*) e quindi dell'eliminazione preventiva delle zone non a rischio; (b) la classificazione (*classification*) delle ROI in masse e tessuto sano. Nella pratica entrambi i livelli eseguono un'operazione di classificazione. La differenza sta nel fatto che la *detection* classifica le regioni in sospette e non sospette, scartando le seconde mentre la *classification* analizza solo le regioni "sopravvissute" al primo livello e le classifica in masse vere e falsi allarmi.

Una caratteristica essenziale degli algoritmi di *detection* sarà quindi un'alta sensibilità, ossia la capacità di trovare le zone sospette: ogni massa persa in questa fase, infatti, fuggendo all'analisi dei livelli successivi, verrà persa per sempre. Di solito il prezzo da pagare per ottenere una sensibilità elevata consiste in un alto numero di falsi positivi.

1.4 Valutazione performance

<u>Sensibilità e specificità</u>

In questo paragrafo introduciamo alcuni elementi di analisi dell'accuratezza dei sistemi di classificazione dicotomici. Sono problemi di questo tipo tutti quei casi in cui gli elementi da classificare sono divisi in due classi e l'appartenenza di un elemento ad una classe esclude l'appartenenza all'altra. In particolare appartengono a

questa categoria i sistemi di diagnostica medica, che devono distinguere tra positivi e negativi.

L'efficienza di un sistema diagnostico può essere valutata considerando i seguenti parametri:

- TPF: frazione di veri positivi (*true positive fraction*);
- TNF: frazione di veri negativi (*true negative fraction*)

Questi valori, indicati rispettivamente come *sensibilità* e *specificità*, misurano quanti positivi e quanti negativi vengono riconosciuti correttamente rispetto ai relativi totali:

$$sensibilità = TPF = \frac{TP}{TP + FN}$$

$$specificità = TNF = \frac{TN}{TN + FP}$$

Si possono introdurre altre due frazioni:

- FNF: frazione di falsi negativi (*false negative fraction*);
- FPF: frazione di falsi positivi (*false positive fraction*);

Questi indici risultano i complementari dei precedenti in quanto indicano le frazioni dei positivi e dei negativi classificati erroneamente. Si ha quindi:

$$FNF = 1 - TPF$$

$$FPF = 1 - TNF$$

Attraverso le coppie (TPF, TNF) o (TPF, FPF) è quindi possibile conoscere i valori di tutti gli indici definiti per il sistema di diagnosi studiato.

Curve ROC e FROC

Esistono due caratteristiche fondamentali per i sistemi di diagnosi che sensibilità e specificità, o i loro complementari, non permettono di evidenziare.

La prima è la capacità intrinseca di distinguere i positivi dai negativi e dipende dalla sovrapposizione delle distribuzioni di probabilità delle due classi. Queste indicano, per ogni valore della variabile caratteristica usata nella classificazione, la probabilità che un elemento appartenga ad una o all'altra classe (Figura 1.5).

La seconda dipende dal fatto che la distinzione avviene applicando una soglia alla variabile caratteristica e solo applicando questa soglia è possibile ottenere sensibilità e specificità. In Figura 1.5 sono rappresentate le due distribuzioni.

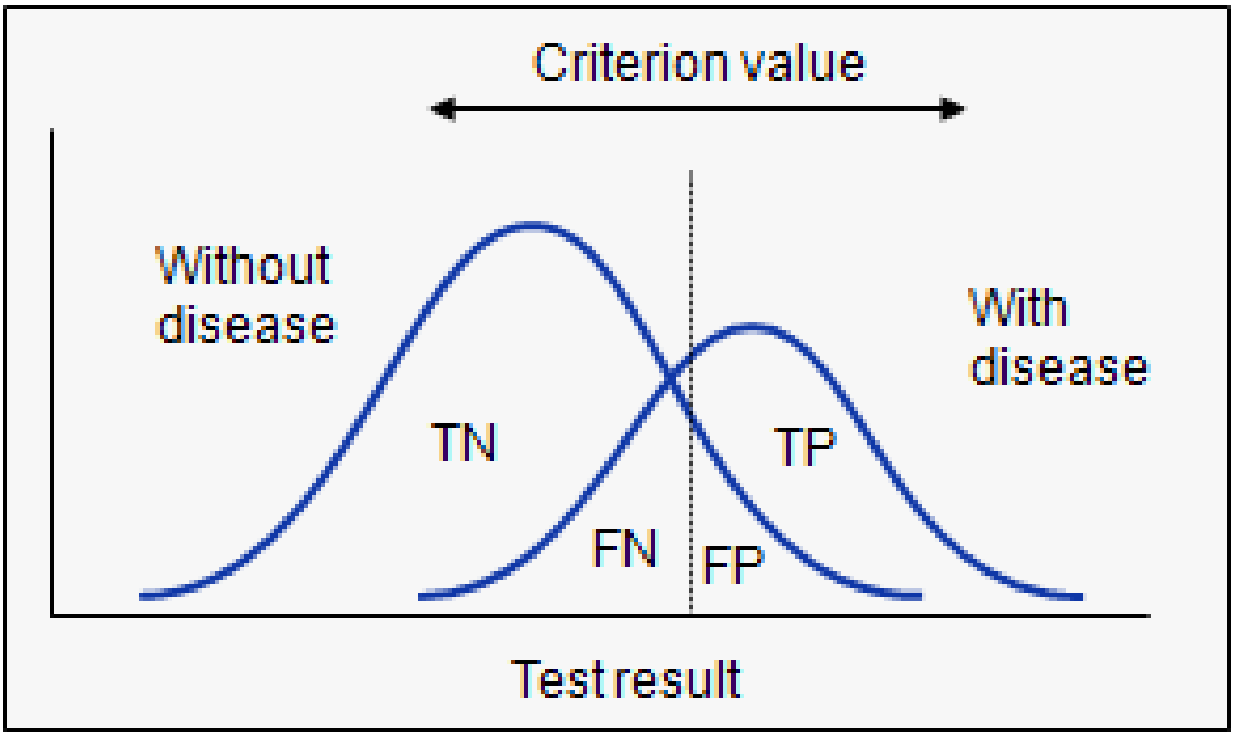


Figura 1.5 Distribuzioni di probabilità dei positivi e dei negativi al variare del valore della variabile caratteristica. Effetto della soglia.

ROC

La soglia definisce un determinato punto di lavoro del sistema di classificazione. Per ottenere un'analisi complessiva del sistema è possibile valutare le coppie (TPF, FPF) per valori differenti della soglia in un intervallo abbastanza ampio da coprire le distribuzioni delle due classi. Il grafico che si ottiene ponendo i valori di FPF sulle ascisse e i rispettivi valori di TPF sulle ordinate è la cosiddetta curva *Receiver Operating Characteristic (ROC)*. Questa permette di visualizzare le prestazioni complessive del sistema, al variare del punto di lavoro.

La diagonale che unisce i punti (0,0) e (1,1) indica un sistema di diagnosi le cui risposte sono indipendenti dagli elementi di ingresso; un tale sistema non distingue in alcun modo le due classi. Un sistema ottimale, cioè che distingua totalmente gli elementi delle due classi, sarà caratterizzato dalla curva formata dai segmenti che uniscono il punto (0,0) con il punto (0,1) e quest'ultimo con il punto (1,1). I sistemi per i quali esiste una sovrapposizione fra le distribuzioni sono caratterizzati da curve intermedie tra i casi limite descritti e le prestazioni migliorano all'avvicinarsi delle curve al punto (0,1). Alcuni esempi sono rappresentati in Figura 1.6.

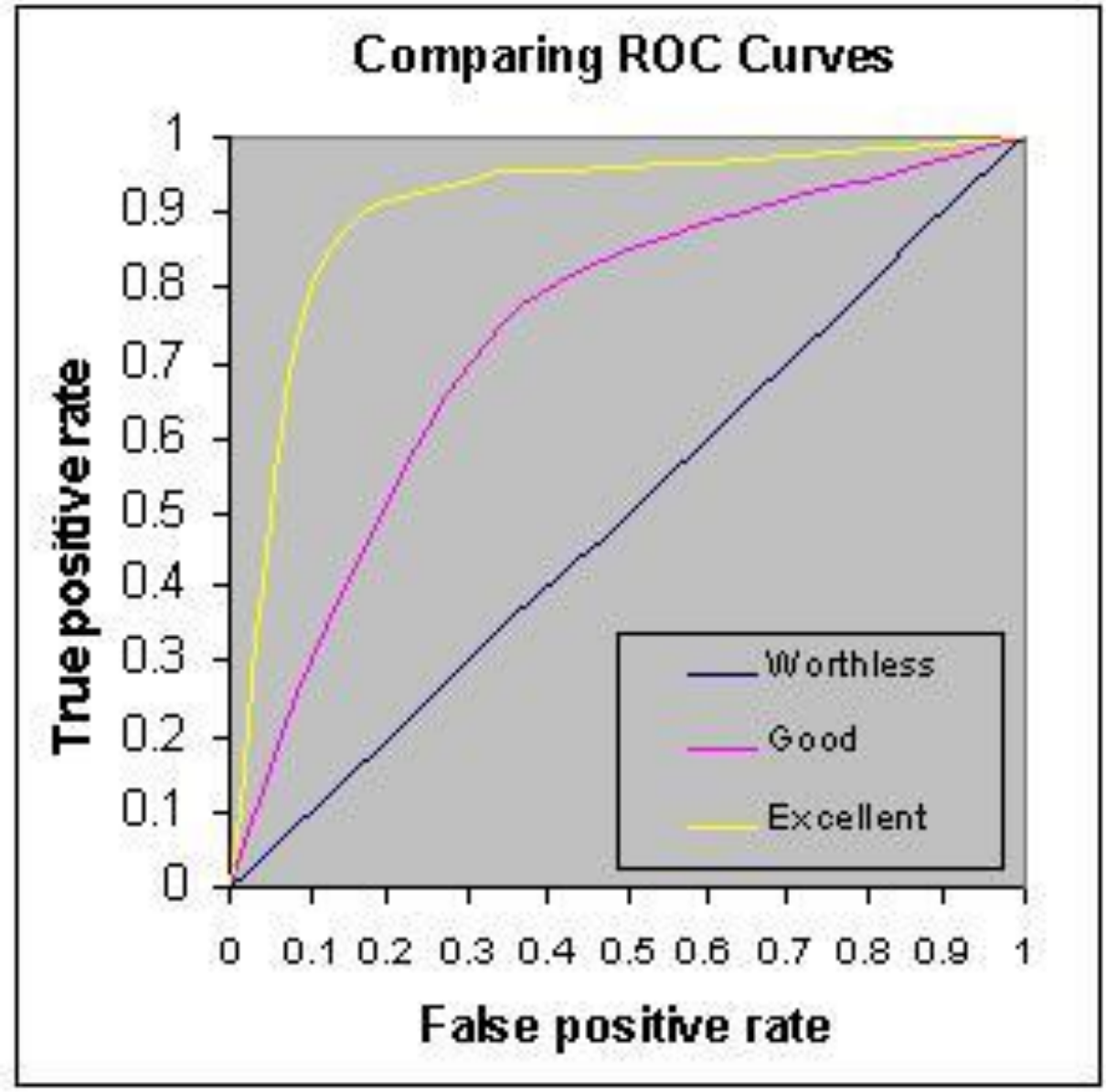


Figura 1.6 Esempi di curve ROC

La curva ROC permette di visualizzare le prestazioni del sistema nei suoi vari punti di lavoro ma fornisce anche un indice quantitativo complessivo per queste prestazioni. Questo si ottiene valutando l'area sotto la curva (A_z), che può essere interpretata come:

- il valore medio della sensibilità per una specificità scelta a caso tra 0 e 1

oppure

- il valore medio della specificità per una sensibilità scelta a caso tra 0 e 1.

FROC

Una variante molto utile della curva ROC è la cosiddetta curva *Free-Response Operating Characteristic (FROC)*, di cui mostriamo un esempio in Figura 1.7.

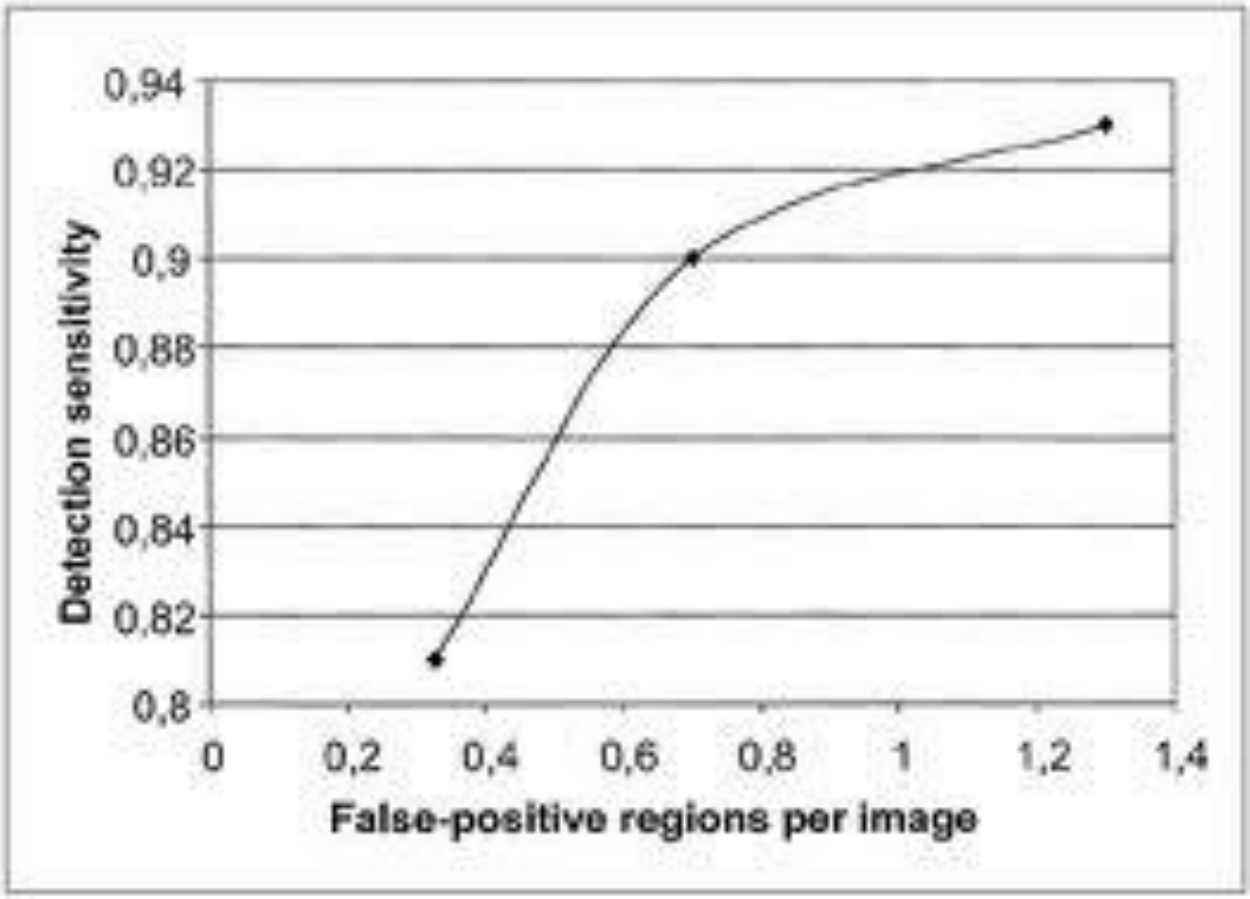


Figura 1.7 Esempio di curva FROC

Essa si ottiene ponendo sulle ascisse il numero di falsi positivi per immagine e sulle ordinate la relativa frazione di veri positivi. In questo modo è possibile conoscere le prestazioni generali del sistema di diagnosi una volta fissata una soglia di falsi positivi per immagine.

Capitolo 2

Deep Learning e CNN

2.1 Introduzione

L'imitazione dell'efficienza e della robustezza con cui il cervello umano rappresenta le informazioni è stata per decenni la sfida centrale delle ricerche di *intelligenza artificiale (artificial intelligence - AI).*

L'ostacolo principale che si incontra in questa sfida, in particolare nell'ambito del *pattern recognition*, ha a che fare con l'elevata dimensionalità dei dati: in questo contesto, infatti, la complessità di apprendimento cresce esponenzialmente con la dimensionalità dei dati. Questo significa che il numero di esempi necessari per stimare, con un dato livello di accuratezza, una funzione arbitraria, cresce esponenzialmente rispetto al numero di variabili (*features*) di ingresso della funzione. Con un numero fissato di esempi di *addestramento*, quindi, la potenza predittiva diminuisce all'aumentare della dimensionalità. Richard Bellman, uno dei primi ad affrontare questo problema, lo ha definito con l'espressione "curse of dimensionality" [15].

L'approccio tradizionalmente più diffuso al superamento di questo ostacolo consiste nel dividere il sistema di *pattern recognition* in due moduli (Figura 2.1): il *feature extractor,* che pre-elabora i pattern in input per diminuirne la dimensionalità, e il *classifier* (da non confondere con il concetto di classificatore introdotto per i sistemi di CAD). Questo approccio ha però un grande limite: l'accuratezza del riconoscimento è fortemente dipendente dall'abilità del realizzatore nello scegliere l'insieme appropriato di features e quindi dal tipo di problema. Nasce quindi la necessità di trovare un meccanismo con validità più generale.

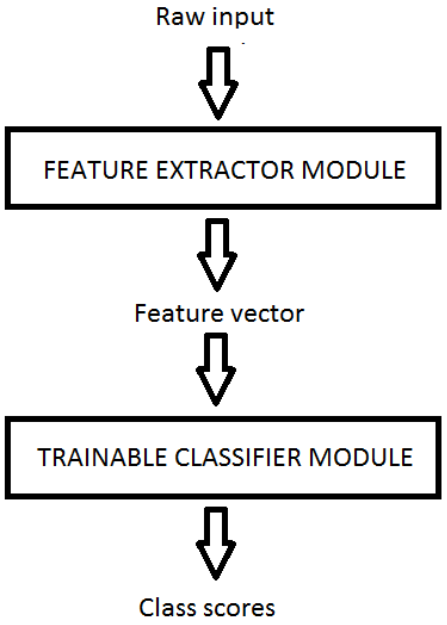


Figura 2.1 Sistema tradizionale di pattern recognition

Alcune scoperte nell'ambito della neuroscienza [15, 16] hanno fornito spiegazioni sui principi che regolano la rappresentazione delle informazioni nel cervello dei mammiferi, portando a nuove idee per la realizzazione di sistemi che imitano queste capacità. Una delle scoperte principali riguarda la neocorteccia, che è responsabile di molte abilità cognitive: essa non pre-elabora esplicitamente i segnali sensoriali ma, piuttosto, li fa scorrere lungo una complessa gerarchia di moduli che, con il tempo, imparano a rappresentare le osservazioni in base alle regolarità che esibiscono.

Questa scoperta ha motivato la nascita del *deep learning (DL)*, resa possibile dalla combinazione di diversi fattori quali la disponibilità di macchine a basso costo con unità aritmetiche veloci, l'aumento delle dimensioni dei database per problemi di ampio mercato ed interesse.

Il DL è una nuova sottocategoria del *machine learning (ML)* che si concentra sullo sviluppo di modelli computazionali per la rappresentazione delle informazioni strutturati con caratteristiche simili a quelle della neocorteccia. Ispirandosi ai risultati delle ricerche nell'ambito della neuroscienza, il DL ha sostanzialmente lo scopo di riportare il machine learning ad uno dei suoi obiettivi originari: l'intelligenza artificiale. Per fare questo esso si serve di un insieme di algoritmi che modellano autonomamente i dati tramite una gerarchia organizzata su diversi livelli (strati) di rappresentazione ed astrazione: i concetti ai livelli più alti sono definiti a partire da quelli ai livelli più bassi tramite una serie di trasformazioni non lineari.

Il cervello dei mammiferi, infatti, è organizzato in un'architettura profonda in cui un certo dato in ingresso viene rappresentato da più livelli di astrazione, ognuno corrispondente ad una diversa area della corteccia. Gli esseri umani descrivono tali concetti in maniera gerarchica, con più livelli di astrazione. Il cervello sembra anche elaborare le informazioni attraverso più fasi di trasformazione e rappresentazione. Questo è particolarmente evidente nel sistema visivo primate, con la sua sequenza di fasi di lavorazione: rilevamento dei bordi, percezione delle forme, da quelle primitive a quelle gradualmente più complesse.

2.2 Machine Learning e apprendimento supervisionato

L'espressione "Intelligenza Artificiale" è stata coniata nel 1956 dal matematico americano John McCarthy, durante uno storico seminario disciplinare svoltosi nel New Hampshire: con questo termine si intende generalmente l'abilità di un computer di svolgere funzioni e ragionamenti tipici della mente umana.

Nel suo aspetto puramente informatico, essa comprende la teoria e le tecniche per lo sviluppo di algoritmi che consentano alle macchine di mostrare un'abilità e/o attività intelligente, almeno in domini specifici.

Il *Machine Learning (ML)* rappresenta una delle aree fondamentali dell'intelligenza artificiale e si occupa della realizzazione di sistemi e algoritmi che si basano su osservazioni come dati per la sintesi di nuova conoscenza. L'apprendimento può avvenire catturando caratteristiche di interesse provenienti da esempi, strutture dati o sensori, per analizzarle e valutarne le relazioni tra le variabili osservate. Per esempio, un sistema di ML può essere addestrato sulle e-mail per distinguere tra messaggi di spam e non-spam.

Apprendimento supervisionato

Il campo dell'apprendimento è molto ampio ed in seguito ci limiteremo a considerare il cosiddetto apprendimento supervisionato: questo termine si riferisce generalmente alle tecniche del machine learning che mirano ad addestrare un sistema informatico a risolvere dei compiti sulla base di una serie di esempi ideali. Il sistema impara cioè ad approssimare una funzione non nota a partire da un insieme di esempi di addestramento formati da coppie ingresso-uscita: per ogni input si comunica al sistema l'output desiderato. L'apprendimento consiste nella capacità del sistema di generalizzazione a nuovi esempi non noti.

Formalmente si definisce apprendimento supervisionato l'approssimazione di funzioni di forma non nota, $f(\boldsymbol{x})$, a partire da un insieme di coppie di valori $(\boldsymbol{x}_i, y_i)$, considerando che questi valori siano ottenuti da f tramite

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i$$

dove con ϵ_i si intendono eventuali contributi di rumore. Va sottolineato che in questo caso la forma della funzione è ignota.

Si distinguono due categorie principali di apprendimento supervisionato:

- classificazione, se i valori assunti dalla funzione sono discreti e rappresentano le classi di appartenenza dei vettori $\boldsymbol{x}_i$;
- regressione, se la funzione assume valori continui.

Nell'ambito della classificazione, il caso particolare in cui si distingua unicamente tra due classi riveste un'importanza notevole. Un esempio di questo tipo è rappresentato proprio dal problema affrontato in questa tesi, in cui la distinzione avviene tra masse e tessuti sani.

Esistono diversi algoritmi di apprendimento supervisionato ma condividono tutti una caratteristica comune: l'addestramento, *training*, avviene minimizzando la cosiddetta funzione costo, *loss function*, che rappresenta l'errore compiuto nella stima dei valori y_i associati ai dati di ingresso $\boldsymbol{x}_i$, tramite la funzione $f(\boldsymbol{x})$.

Sono possibili molteplici scelte per le funzioni costo ma la maggior parte dei sistemi utilizza la funzione di scarto quadratico:

$$\frac{1}{2}\sum_i (y_i - \tilde{y}_i)^2$$

dove $\tilde{y}_i$ indica i valori assunti in uscita dal sistema quando vengano presentati gli $\boldsymbol{x}_i$ in ingresso.

La scelta della funzione da minimizzare sottintende il principio che sta alla base dell'apprendimento. Utilizzare una funzione che, come quella di scarto quadratico, sia un computo diretto dell'errore su tutti gli esempi dell'insieme di training, significa supporre che minimizzando l'errore su questi dati si minimizzi anche l'errore in cui incorre la macchina quando sia chiamata a generalizzare, cioè a determinare il valore della funzione per punti non presenti nell'insieme di addestramento. Nella pratica questa supposizione, specie per un insieme ridotto di dati, non è valida, perché questo tipo di minimizzazione comporta l'adattamento troppo specifico della macchina ai

dati utilizzati per l'addestramento, un effetto chiamato, in letteratura, *overfitting*. Questo comporta un peggioramento delle prestazioni in fase di generalizzazione.

Per ovviare a questi problemi è possibile cercare una stima differente dell'errore in generalizzazione. Questo si può ottenere controllando l'efficienza attraverso un insieme di dati non utilizzato in fase di addestramento, valutando la loss function per questo insieme. Questo approccio, detto *validation*, permette anche di ottimizzare parametri di implementazione non direttamente appartenenti all'algoritmo di apprendimento; per esempio, nel caso di riconoscimento di immagini, le dimensioni delle immagini di ingresso.

Anche ottimizzando il sistema attraverso i dati di validation, si può comunque incorrere in un adattamento troppo specifico. Per questo motivo un terzo insieme, detto di *test*, viene utilizzato come controllo finale. Se l'errore ottenuto risulta simile a quello di validation l'overfitting non si è verificato.

Cross Validation

Lo schema training, validation, test è l'approccio tradizionale al problema dell'ottimizzazione di un sistema di apprendimento supervisionato. Tuttavia, per molti problemi, non è disponibile una base di dati abbastanza ampia da permettere una suddivisione in tre insieme di dimensioni sufficienti. Una tecnica sviluppata per gestire situazioni di questo tipo è la *cross validation.* Essa permette di utilizzare tutti i dati disponibili per ottimizzare il sistema.

L'insieme dei dati viene suddiviso in N partizioni, *fold*, che abbiano approsimativamente la stessa dimensione, per poi addestrare il sitema con $N-1$, utilizzando il restante per la validation. Questo procedimento viene reiterato N volte escludendo ogni volta un insieme diverso. I risultati vengono poi uniti come se la validation fosse stata eseguita su tutto l'insieme dei dati. In questo modo tutti gli esempi disponibili sono utilizzati per l'apprendimento.

Nonostante questa tecnica aumenti notevolmente i tempi di calcolo, in quanto l'addestramento va compiuto N volte, vari studi teorici e simulazioni hanno provato una buona efficacia di questo sistema nell'ottenere buone stime per l'efficienza di generalizzazione.

Gradient-based learning e gradient back-propagation

Il problema generale della minimizzazione di una funzione rispetto ad un insieme di parametri è alla radice di molti problemi informatici.

La loss function può essere minimizzata facendo una stima dell'impatto che hanno su di essa piccole variazioni dei valori dei parametri. Questo può essere misurato tramite il gradiente della loss function rispetto ai parametri. Questa è l'idea base di numerosi algoritmi con parametri a valori continui detti *gradient-based learning algorithms.*

In generale il set di parametri W è un vettore a valori reali, rispetto al quale la loss function $E(W)$ è continua e differenziabile. Date queste condizioni, la procedura di minimizzazione usata più spesso è l'algoritmo del *gradient descent* (Figura 2.2) dove $\boldsymbol{W}$ viene "aggiustato" iterativamente come segue:

$$W_k = W_{k-1} - \eta \frac{\partial E(W)}{\partial W}$$

dove η controlla la velocità di discesa lungo il gradiente e, nei casi più semplici, è una costante. Dato quindi un punto nello spazio dei parametri, per minimizzare $E(W)$, ci si muove in direzione opposta a quella del gradiente.

L'utilità sorprendente delle tecniche basate sul gradient descent per i compiti di machine learning complesso rimase sconosciuta fino alla scoperta dell'algoritmo di *back-propagation* (che vedremo in dettagli nel prossimo paragrafo) da parte di Hinton, Rumelhart e Williams [30], che permette il calcolo del gradiente di sistemi non lineari composti da livelli multipli di elaborazione. L'idea alla base del back-propagation è che il gradiente può essere calcolato in modo efficiente tramite la propagazione degli errori dall'output all'input.

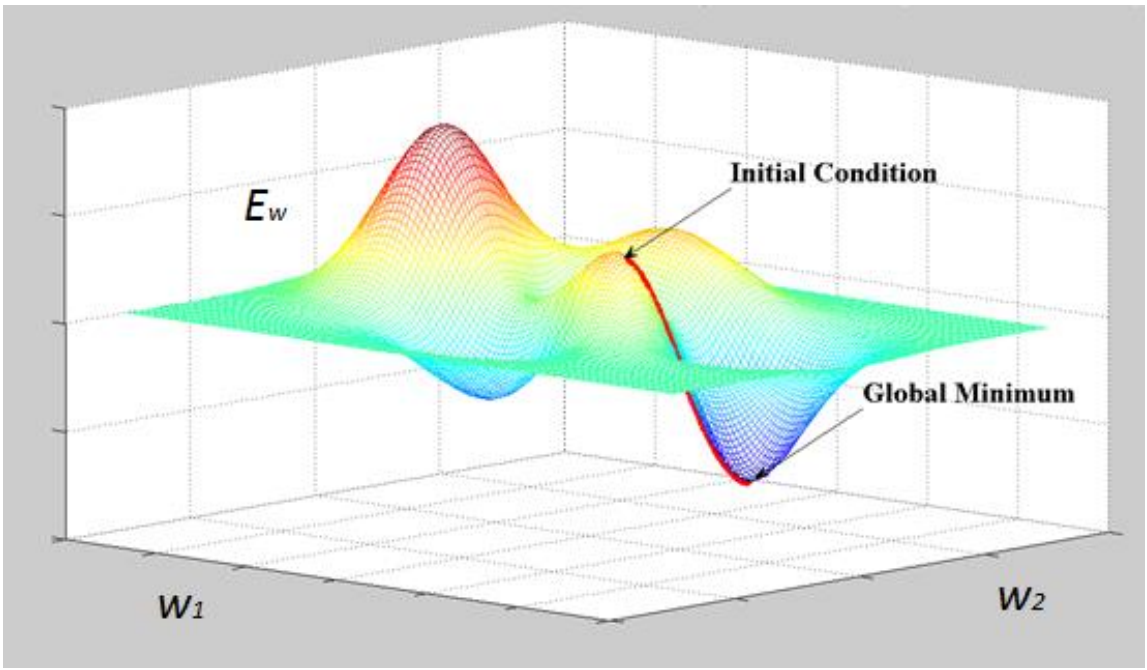


Figura 2.2 Gradient descent

2.3 Neural Networks

Fra gli algoritmi di apprendimento supervisionato, quelli derivati dalle reti neurali hanno avuto un ruolo fondamentale nello sviluppo del campo. L'origine concettuale delle reti neurali si trova nella modellizzazione astratta del sistema di comunicazione tra neuroni.

Il neurone

Un neurone è una cellula composta da una parte centrale, detta *soma*, da cui si dipartono una fibra nervosa principale, l'*assone*, e una serie di ramificazioni, i *dendriti*. I neuroni sono collegati tra loro attraverso le sinapsi, che sono i punti di contatto tra la parte terminale dell'assone e i dendriti o il soma degli altri neuroni.

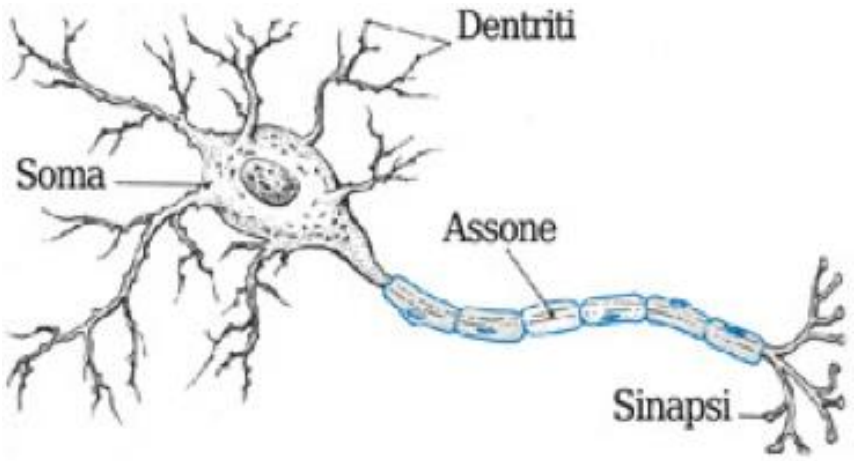


Figura 2.3 Modello schematico del neurone

La comunicazione tra neuroni avviene nel modo seguente. Quando un neurone è attivo invia un impulso elettrico attraverso il suo assone; questo impulso viene trasmesso, attraverso le sinapsi, ai neuroni collegati. La trasmissione avviene attraverso lo scambio di sostanze chimiche, chiamate *neurotrasmettitori*, dall'assone ai recettori presenti sul soma e sulle dendriti. In base al tipo di stimolo e alla configurazione dei recettori, un impulso modifica il potenziale dei neuroni collegati.

Ogni neurone ha un proprio potenziale di attivazione, che può essere raggiunto solo quando si vengano a creare certe configurazioni di attivazione dei neuroni ad esso collegato. Una volta raggiunto tale potenziale, il neurone si attiva e trasmette a sua volta l'impulso ai neuroni collegati alla parte terminale del proprio assone.

La capacità di apprendere sta nella possibilità di modificare le caratteristiche dei recettori in modo da modificare le configurazioni che comportano l'attivazione del neurone.

Reti neurali artificiali

Partendo da questa conoscenza di base del funzionamento del sistema nervoso, sono stati costruiti modelli semplificati sia al fine di studiare il funzionamento dei loro corrispettivi biologici che per verificare se tali modelli siano dotati di capacità di apprendimento.

L'elemento fondamentale di tutti questi modelli è il neurone artificiale, introdotto per la prima volta da McCulloch e Pitts nel 1943 (Figura 2.4). Si tratta di un'unità di calcolo a N ingressi ed una uscita. Ciascuno degli ingressi rappresenta una terminazione sinaptica ed è quindi collegato all'uscita di altri neuroni artificiali. Ad ogni ingresso è associato inoltre un *peso sinaptico* w_i.

I valori di uscita possono essere due valori discreti, solitamente 0 e 1 o -1 e 1 che indicano lo stato attivo o inattivo del neurone; oppure degli intervalli continui, in genere [0,1] o [-1,1], che indicano il grado di attivazione del neurone.

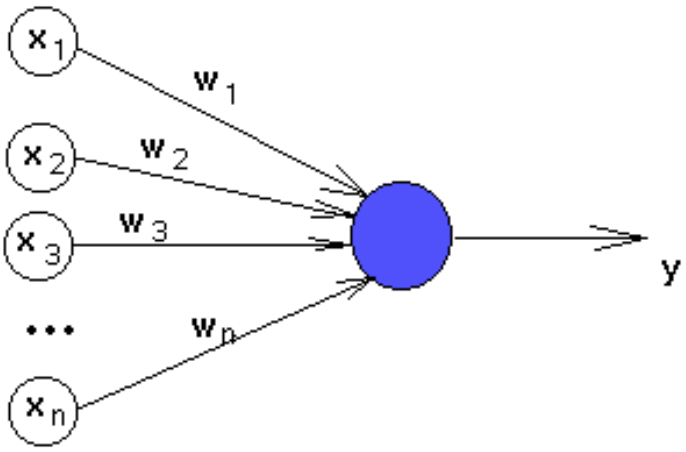


Figura 2.4 Schema del neurone artificiale

Il calcolo del valore di uscita viene effettuato determinando innanzitutto il *potenziale* P, attraverso una somma delle uscite dei neuroni collegati O_i, pesata mediante i pesi sinaptici w_i*:*

$$P = \theta$$

L'uscita si ottiene direttamente dal potenziale attraverso una funzione di trasferimento:

$$O = f(P)$$

Nel caso in cui l'uscita debba essere discreta si possono utilizzare funzioni a soglia come la funzione segno o il gradino di Heaviside ϑ. Negli altri casi si possono scegliere diversi tipi di funzioni. Nella maggior parte dei casi si utilizzano la tangente iperbolica o la funzione *sigmoide*

$$f(x) = \frac{1}{1 + e^{-x}}$$

in quanto esse permettono di regolarizzare rispettivamente la funzione segno e la funzione ϑ, essendo entrambe derivabili e avendo lo stesso comportamento di queste funzioni agli estremi.

Connettendo queste unità secondo varie topologie è possibile ottenere diversi tipi di reti. L'apprendimento avviene modificando i pesi sinaptici in modo tale da minimizzare l'errore in fase di *training*, secondo le metodologie descritte nel paragrafo 2.4. Per fare questo è necessario definire una legge di apprendimento che dia i nuovi valori dei pesi partendo dal valore della loss function.

Queste metodologie verranno esposte, per un caso particolare, nel prossimo paragrafo, dove descriveremo una tipologia di rete molto diffusa adatta all'apprendimento supervisionato.

Multi-Layer Perceptron – MLP

Il *multi-layer perceptron* è una tipologia di rete neurale artificiale studiata specificamente per l'apprendimento supervisionato.

Il modello consiste in almeno tre strati di neuroni artificiali collegati in un'unica direzione, *feed forward*.

Ogni neurone, oltre agli ingressi relativi alle uscite dei neuroni precedenti, possiede un ulteriore ingresso: il *bias*. Ad esso viene dato un valore fisso pari ad 1. Questo comporta un termine costante nella somma.

Spesso, nel calcolo del numero degli strati, il primo non viene considerato perché i suoi elementi non possiedono pesi sinaptici: infatti questo strato si limita a presentare i valori di ogni input ai neuroni dello strato successivo, eseguendo semplicemente una trasformazione tramite la funzione di trasferimento. La rete rappresentata in Figura 2.5, ad esempio, viene considerata a due o a tre strati, contando rispettivamente il numero di pesi o di neuroni. È preferibile contare gli strati di pesi, in quanto è proprio la modifica di questi ultimi che rende la rete in grado di apprendere.

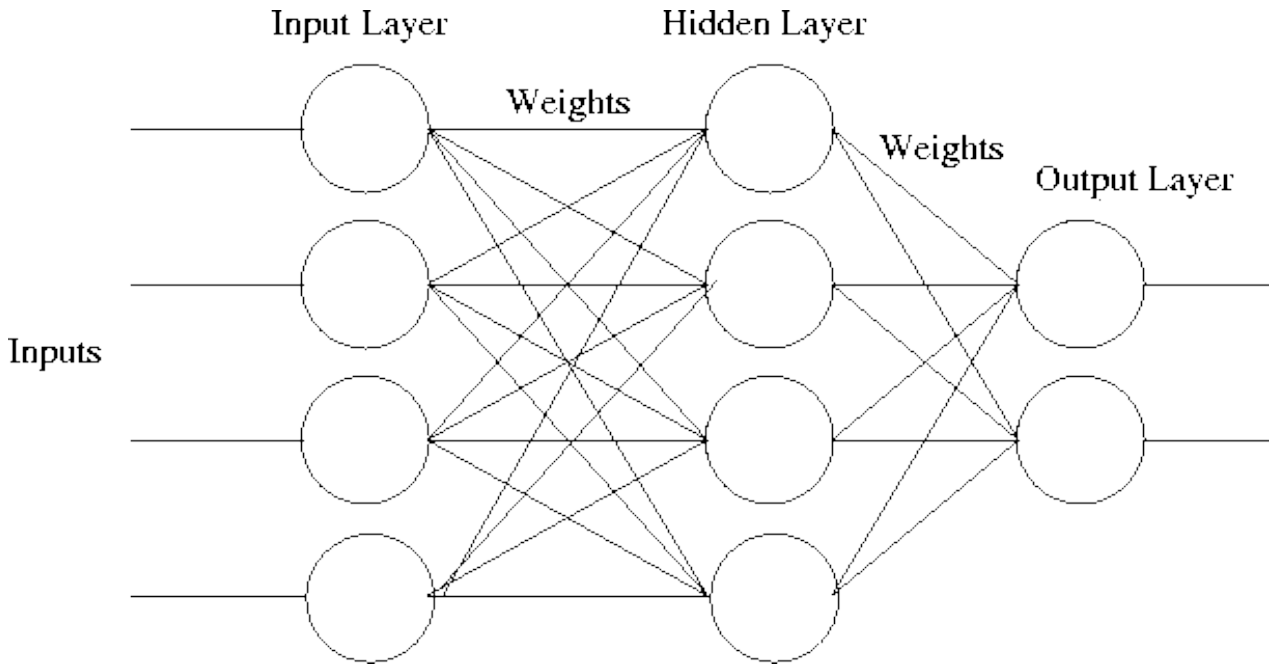


Figura 2.5 Schema di una rete MLP

Il primo e l'ultimo strato devono avere un numero di unità pari rispettivamente alla dimensione dello spazio di ingresso e di quello di uscita: queste sono le terminazioni della "scatola nera" che rappresenta la funzione da approssimare. Per la classificazione a due classi è sufficiente una sola uscita, il cui valore, passato attraverso la soglia, indicherà l'appartenenza di un vettore in ingresso ad una delle due classi. Nel caso in cui si abbiano più di due classi si utilizza un numero di uscite pari al numero delle classi stesse, anch'esse sottoposte a soglia.

Il valore di uscita della rete si calcola nel modo seguente. Ad ogni neurone di uno strato si applicano la somma e la funzione di trasferimento. I valori calcolati sono gli ingressi dello strato successivo, le cui uscite si calcolano allo stesso modo. Si procede così fino a giungere all'ultimo strato le cui uscite rappresentano l'uscita di tutta la rete.

Vedremo che la legge di apprendimento richiede che le funzioni di trasferimento dei neuroni siano derivabili. In seguito, per semplicità, ci limiteremo al caso particolare di una rete a due strati di pesi (come quella in Figura 2.5) che utilizzi come funzione di trasferimento quella sigmoidale.

Introduciamo la seguente notazione: consideriamo *I* ingressi, *J* elementi interni, *K* uscite ed utilizziamo *i*, *j* e *k* rispettivamente per indicare gli indici riferiti agli elementi degli strati. w_{ij} rappresenta il peso sinaptico del j-esimo neurone dello strato interno riferito all'i-esimo ingresso e analogamente w_{jk} rappresenta il peso del k-esimo neurone di uscita riferito al j-esimo neurone interno. I valori di *bias* sono

analogamente indicati con w_{0j} e w_{0k}. Si considera la funzione di trasferimento f. Indichiamo con O_i, O_j, e O_k i valori di uscita di ogni strato e con P_j e P_j i potenziali degli strati successivi al primo. Inoltre consideriamo posti uguale a 1 i valori di uscita di indice 0.

Il valore di uscita del k-esimo neurone dell'ultimo strato risulta quindi:

$$O_k = f\left(\sum_{j=1}^{J} w_{jk} f\left(\sum_{i=0}^{I} w_{ij} O_i\right) + w_{0k}\right)$$

L'apprendimento si ottiene mediante l'algoritmo *error back-propagation* che permette di modificare i pesi sinaptici delle celle anche in presenza di strati interni. Esso modifica i pesi sinaptici in base alla discrepanza tra la risposta fornita dalla rete e la risposta corretta impiegando il metodo del gradient descent della funzione di errore. Per fare questo si attribuisce un valore casuale, di solito compreso tra -1 e +1, ad ogni peso della rete. Si ottiene quindi un punto iniziale nello spazio dei parametri dal quale ci si muove in direzione opposta a quella del gradiente. Nelle reti MPL la funzione di costo più utilizzata è lo scarto quadratico.

Nel casi in cui l'insieme di training sia composto di *L* esempi ed utilizzando *l* come indice per questi esempi, l'errore diventa:

$$E = \frac{1}{2}\sum_{l}\sum_{k}\left(O_k^l - y_k^l\right)^2$$

con O_k^l dato dalla formula vista prima calcolata per l'l-esimo esempio. Dal calcolo del gradiente di questa funzione, rispetto ai pesi dei vari strati, si ottiene la seguente legge di apprendimento, dove il parametro η controlla la velocità di discesa lungo il gradiente. Per lo strato di uscita:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \delta_k O_j$$

dove $\delta_k = \frac{\partial E}{\partial P_k} = (O_k - y_k) y_k (1 - y_k)$

Per lo strato nascosto:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{ik}} = -\eta \delta_j O_i$$

dove $\delta_j = \frac{\partial E}{\partial P_j} = \sum_k \delta_k w_{jk} O_j (1 - y_j)$

Le suddette leggi di apprendimento hanno la stessa forma. Ciò che le differenzia sono le definizioni di δ_k e δ_j. Il secondo risulta dipendente dal primo ed, in questo senso, si parla di propagazione all'indietro dell'errore, da cui il nome dell'algoritmo.

Risulta necessario almeno uno strato interno affinché si possa introdurre la non linearità, che risulta l'elemento fondamentale della flessibilità delle reti. Un esempio classico dell'inefficacia di un singolo strato è l'incapacità di ottenere una rete di questo tipo che possa calcolare lo XOR logico.

L'architettura interna della rete ha un notevole impatto sulle prestazioni. La molteplicità di configurazioni possibili comporta un numero di soluzioni subottimali. Inoltre, lo stesso algoritmo di apprendimento può fermarsi raggiungendo minimi locali. Un problema che affligge in generale tutti i sistemi di apprendimento supervisionato è la *curse of dimensionality,* un progressivo decadimento delle prestazioni all'aumentare della dimensione dello spazio di ingresso. Questo avviene perché il numero di esempi necessari ad ottenere un campionamento sufficiente dello spazio di ingresso aumenta esponenzialmente con il numero delle dimensioni. Il probema risulta particolarmente significativo nelle reti neurali.

2.4 Support Vector Machine – SVM

Le *Support Vector Machines (SVM)* sono un insieme di metodi di apprendimento supervisionato per la regressione e la classificazione di *pattern*, sviluppati negli anni '90 da Vladimir Vapnik ed il suo team presso i laboratori Bell AT&T.

Appartengono alla famiglia dei classificatori lineari generalizzati e sono anche noti come classificatori a massimo margine, poiché allo stesso tempo minimizzano l'errore empirico di classificazione e massimizzano il margine geometrico.

Le Support Vector Machine sono state inventate da Vladimir Vapnik e i suoi colleghi del AT&T Bell Laboratories ed introdotte per la prima volta nel 1992 alla Computational Learning Theory conference. Le radici di questo approccio, il metodo dei Support Vectors per costruire l'iperpiano di separazione ottimale per la classificazione di pattern, risalgono al lavoro del 1964 di Vapnik e Chervonenkis [18]. Oggi le SVM sono diventate uno degli strumenti standard nella comunità del machine learning per problemi di classificazione, regressione e stima delle densità.

Il metodo più intuitivo per comprendere l'applicazione delle SVM consiste nello studio del problema della separazione di due classi da un punto di vista geometrico. Consideriamo per semplicità uno spazio di ingresso bidimensionale. Gli elementi delle due classi saranno sparsi nel piano e, in base alle loro caratteristiche, occuperanno regioni diverse. Nel caso in cui risulti possibile separare tali classi tramite una retta o, in caso di un numero superiore di dimensioni, un iperpiano, allora

le SVM permettono di scegliere univocamente quello ottimale (Figura 2.6). Si ottiene che tale iperpiano ottimale è quello per cui risulta massimo il margine fra le due classi, cioè la distanza minima dei punti di una classe dall'iperpiano.

L'iperpiano di separazione divide lo spazio in due parti. Il sistema classifica un nuovo elemento in ingresso come appartenente all'una o all'altra classe in base a quale delle due regioni contiene il punto dello spazio individuato da questo elemento.

Può accadere che la separazione completa delle due classi utilizzando un iperpiano non sia possibile. Qualunque iperpiano si scelga ci saranno alcuni punti di una classe che si trovano dalla parte dello spazio associata all'altra classe (Figura 2.7). Questo problema si riesce ad affrontare modificando l'algoritmo: si introduce un costo associato agli errori di classificazione nell'ottimizzazione.

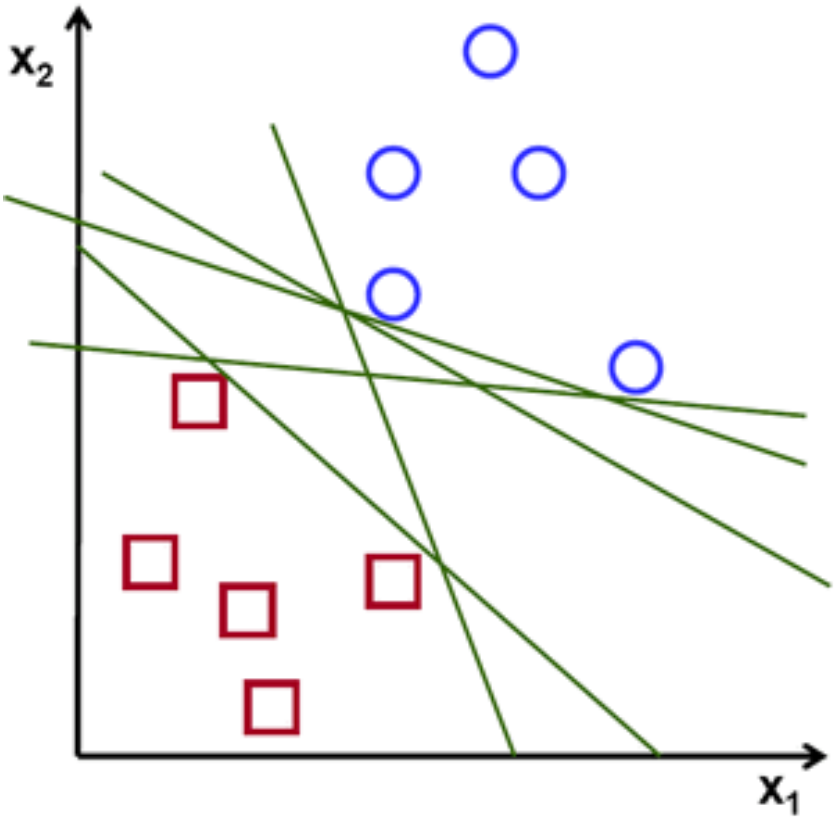


Figura 2.6 Separazione completa mediante un iperpiano (che in due dimensioni si riduce ad una retta)

Nei casi di separazione completa si cerca l'iperpiano di separazione direttamente nello spazio di ingresso. In generale, tuttavia, la separazione attraverso un iperpiano è poco efficiente. In Figura 2.8 si nota come la distribuzione sia troppo complessa per essere separata in questo modo, anche considerando gli errori introdotti in precedenza. Risultano molto più efficaci curve come quelle tracciate. La complessità della distribuzione può essere ridotta "mappando" gli elementi in uno spazio di

dimensioni maggiori, detto delle *features*, attraverso una funzione ϕ non lineare, come in Figura 2.8.

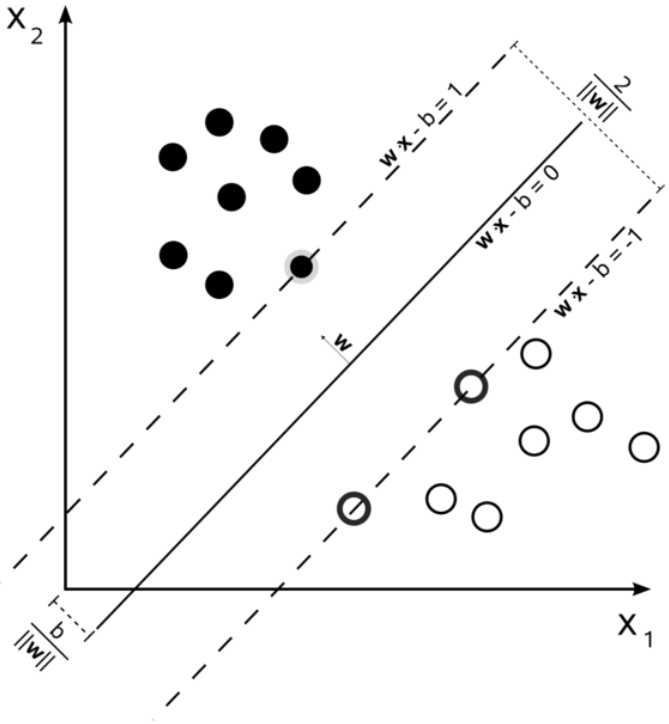


Figura 2.7 Individuazione dell'iperpiano di separazione nel caso non separabile

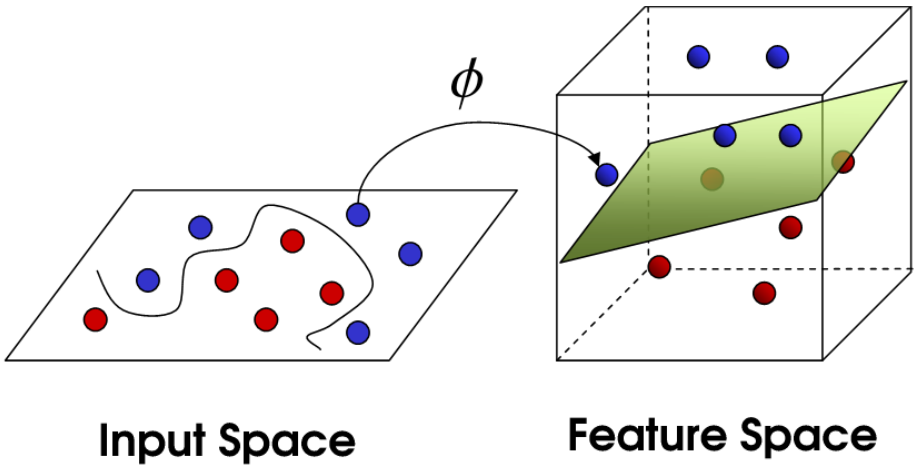


Figura 2.8 Separazione in uno spazio di dimensioni maggiori, attraverso una funzione non lineare

La non linearità della funzione permette di ridistribuire gli elementi in modo da creare una separazione efficiente utilizzando un iperpiano. È quindi possibile applicare il metodo indicato per le SVM in questo spazio. In generale la conoscenza di ϕ è un problema computazionale troppo complesso; tuttavia si può dimostrare che per applicare l'algoritmo è unicamente necessario conoscere il prodotto scalare di due

vettori nello spazio delle features in funzione dei loro corrispettivi vettori nello spazio di ingresso.

L'algoritmo SVM possiede alcune proprietà che lo rendono preferibile ad altri algoritmi per l'apprendimento supervisionato. Prendiamo in considerazione in particolare le reti neurali a molti strati. Abbiamo visto che una caratteristica di queste reti è la possibilità che l'ottimizzazione si fermi in minimi locali. Questo problema non sussiste nel caso delle SVM in quanto la funzione di ottimizzazione è convessa, quadratica e a vincoli lineari. Questo comporta la globalità del minimo.

Dal teorema di Karush-Kuhn-Tucker risulta che la soluzione del problema dipende solamente da un sottoinsieme di vettori di *training*, i *support vectors.* L'algoritmo, cioè, permette di scegliere automaticamente quali informazioni utilizzare e quali scartare, definendo quindi l'architettura ottimale. Nel caso delle reti neurali a molti strati, invece, l'architettura è da determinare attraverso un processo di *validation*.

La possibilità di usare funzioni di *kernel* permette di rendere l'algoritmo adatto a separare distribuzioni di vettori molto complesse. Se infatti questo non fosse possibile, la non linearità delle funzioni di trasferimento dei neuroni nelle reti neurali renderebbe queste molto più flessibili rispetto alle SVM.

Infine, vari studi hanno dimostrato che, rispetto alle reti neurali, il problema della *curse of dimensionality* risulta notevolmente ridotto.

2.5 Deep Learning

Il concetto di *profondità* (*depth, da cui deep learning*), riferito ad uno strumento di modellazione di dati può essere illustrato mediante un esempio. Le operazioni eseguite per arrivare dall'input all'output possono essere rappresentate tramite un grafico di flusso composto da vari nodi che rappresentano un'operazione elementare e danno un risultato dipendente da quelli dei nodi vicini.

Ad esempio, il grafico di flusso per l'espressione $\sin(a^2 + b/a)$, in Figura 2.9, consiste in due nodi di input a e b, un nodo per la divisione b/a che ha come input a e b (i.e. come figli), un nodo per il quadrato (che prende a come input), un nodo per l'addizione (input a^2 e b/a) e, infine, il nodo di output che calcola il seno, con un solo input proveniente dal nodo dell'addizione.

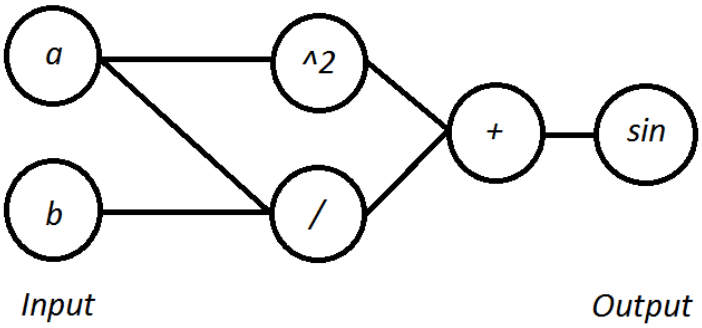


Figura 2.9 Grafico di flusso di $\sin(a^2 + b/a)$

La profondità, definita formalmente come lunghezza del percorso più lungo che collega input e output, è una proprietà importante di tali grafici. Una profondità pari a 2 è di solito sufficiente per rappresentare qualsiasi funzione con precisione arbitraria. Questo però comporta un prezzo da pagare: il numero di nodi richiesti dal grafico può diventare molto grande. Quindi per l'apprendimento di funzioni complicate, che possono rappresentare alti livelli di astrazione, può essere necessario ricorrere ad architetture profonde (composte da livelli multipli di operazioni non lineari), come nelle reti neurali a molti strati nascosti.

Questo però non è l'unico motivo per lo studio di algoritmi di apprendimento per architetture profonde. Infatti:

- Il cervello ha un'architettura profonda
- Gli uomini organizzano le loro idee gerarchicamente
- Gli uomini prima imparano concetti semplici e poi li compongono per rappresentare concetti più astratti
- Architetture non profonde possono essere esponenzialmente inefficienti
- Gli ingegneri scompongono le soluzioni in livelli multipli di astrazione ed elaborazione

Spinti da questi motivi i metodi del deep learning mirano, tramite l'utilizzo di architetture profonde, all'apprendimento di gerarchie di features, con le features ai livelli più alti formate tramite la composizione di quelle ai livelli più bassi.

Ispirandosi all'architettura profonda del cervello, i ricercatori nel campo delle reti neurali hanno provato per decenni ad addestrare reti a molti strati, ma fino al 2006 non ci sono stati tentativi di successo: i risultati erano positivi fino a due o tre livelli (cioè uno o due strati nascosti), ma aumentando il numero si osservava un peggioramento dei risultati. Nel 2006 Hinton et al. presso l'Università di Toronto

hanno dato alla scena la spinta decisiva introducendo le Deep Belief Networks (DBN), ossia reti che, sfruttando un algoritmo di apprendimento non supervisionato (Restricted Boltzman Machine - RBM), addestrano singolarmente ogni strato.

Dal 2006, le architetture profonde sono state applicate con successo non solo nei compiti di classificazione ma anche in regressione, riduzione della dimensionalità, modelli di texture, modellazione del movimento, segmentazione di oggetti, recupero delle informazioni, robotica, elaborazione del linguaggio naturale.

2.6 Convolutional Neural Networks

L'abilità delle reti neurali multi-strato nell'apprendere, a partire da grandi insiemi di esempi, funzioni complesse, non-lineari e ad alta dimensionalità, le rende perfetti candidati per i compiti di riconoscimento di immagini.

Immagini digitali e filtri

Un'immagine digitale può essere considerata come una matrice *A* di dimensione $M \times N$ valori reali o discreti. Ogni valore della matrice prende il nome di pixel e i suoi indici sono anche chiamati coordinate: ogni pixel *A(m,n)* rappresenta l'intensità nella posizione indicata dagli indici.

Si definisce filtro una trasformazione applicata ad un'immagine. Possiamo definire vari tipi di filtri (*kernel*) in base a quale regione dell'immagine di partenza sia necessaria per ottenere il valore di un pixel:

- Globali: per determinare il valore dell'immagine di uscita è necessario conoscere ogni pixel dell'immagine di ingresso. Il filtro è dunque rappresentabile tramite una funzione $R^{M \times N} \rightarrow R^{M \times N}$.
- Locali: il valore di un pixel dell'immagine di uscita dipende dal valore dei pixel di una sottoregione dell'immagine di ingresso. Se tale sottoregione ha lato pari a *L*, il valore di ogni pixel dell'immagine di uscita sarà quindi ottenuto da una funzione $R^{L \times L} \rightarrow R$
- Puntuali: il valore di un pixel dipende solo da un altro pixel, in genere quello che ha le stesse coordinate del pixel da calcolare. La funzione in questo caso diventa $R \rightarrow R$.

Riguardo ai filtri globali e locali, un particolare tipo di operazione che si può eseguire per passare da ingresso ad uscita è rappresentata dalla *convoluzione*.

La convoluzione, nel caso si operi su immagini digitali (convoluzione discreta), si può definire come:

$$c[m,n] = a[m,n] \otimes b[m,n] = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} a[j,k]b[m-j,n-k]$$

considerando $a[m,n]$ l'immagine di ingresso, $c[m,n]$ l'immagine filtrata e definendo la matrice di convoluzione $b[m,n]$. Otteniamo un filtro locale scegliendo per b una matrice di dimensioni pari alla sottoregione che influenza il valore di un pixel. Ogni pixel è così il risultato di una somma pesata tramite la matrice $b[m,n]$ dei valori della sottoregion che ha per centro le coordinate del pixel.

È importante notare che, perché i filtri siano ben definiti, è necessario considerare gli estremi dell'immagine di partenza: infatti la sottoregione centrata su un punto del bordo dell'immagine tocca punti non definiti. Esistono due possibilità, da scegliere in base all'utilizzo che si deve fare dell'immagine filtrata:

- estendere l'immagine, ottenendo un'immagine filtrata avente le stesse dimensioni dell'immagine di ingresso
- non considerare la cornice non definita, ottenere un'immagine di uscita più piccola: se l'immagine d'ingresso ha dimensioni $M \times N$ e la matrice di convoluzione $m \times n$, l'immagine filtrata avrà dimensioni:

$$(M - m + 1) \times (N - n + 1)$$

Vedremo in seguito che le convolutional neural networks sfruttano questa seconda opzione.

Convolutional Neural Networks

In linea teorica, una rete neurale tradizionale completamente connessa con dimensioni sufficientemente grandi può imparare a riconoscere immagini raw (grezze) senza l'estrazione di features. Tuttavia, con un approccio di questo tipo, sorgerebbero dei problemi legati al numero di parametri da addestrare (e quindi al numero di esempi di training necessari), che diventerebbe ingestibile da parte di molti sistemi hardware. Inoltre, un altro difetto delle reti neurali ordinarie è legato al fatto che l'input ha la forma di un vettore, quindi una sola dimensione. Per questo motivo, se si vuole classificare un'immagine, bidimensionale, questa andrà prima "srotolata" concatenandone le righe o le colonne in un unico vettore. In questo modo la topologia dell'input verrà quasi completamente ignorata. Le immagini però hanno una forte struttura locale: pixel vicini hanno di solito una forte correlazione spaziale.

Le *Convolutional Neural Networks (CNN)*, un particolare tipo di reti neurali appartenenti al settore del deep learning, evitano questi problemi grazie all'uso di particolari idee architetturali. Le CNN sono state ideate da Yann LeCun, che si è ispirato al *Neocognitron* di Fukushima del 1980.

La differenza sostanziale tra le convolutional neural networks e le reti neurali ordinarie consiste nel fatto che le prime operano direttamente sulle immagini mentre le seconde su *features* estratte da esse. L'input di una CNN quindi, a differenza di quello di una rete neurale ordinaria, sarà bidimensionale e le *features* saranno i pixel stessi.

In Figura 2.7 viene rappresentata la struttura base di una CNN. Vediamo come ad un'immagine in input corrispondano, all'interno degli strati nascosti, diversi gruppi di immagini chiamate *feature maps*: le *feature maps* di uno strato sono il risultato di trasformazioni eseguite sulle immagini dello strato precedente mediante filtri di convoluzione o *subsampling*, anche detti *kernel*. Il percorso dall'input all'output è caratterizzato da un'alternanza di strati di convoluzione e *subsample* e si conclude con una rete neurale tradizionale.

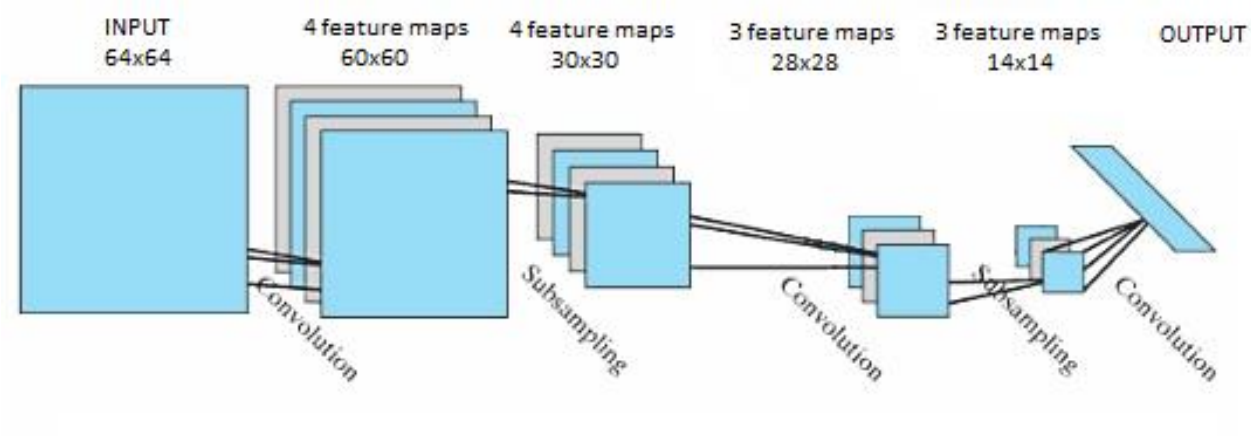


Figura 2.7 Convolutional neural network: alternanza tra strati di convoluzione e *subsampling*

Negli strati di convoluzione n immagini in input vengono convolute con k diversi filtri. Ciascun kernel produce dunque n immagini, una per ogni immagine proveniente dallo strato precedente. Queste n immagini vengono sommate e il risultato è una delle k feature maps prodotte dai k diversi filtri: queste, una volta passate attraverso una funzione sigmoide, rappresenteranno l'input per lo strato successivo.

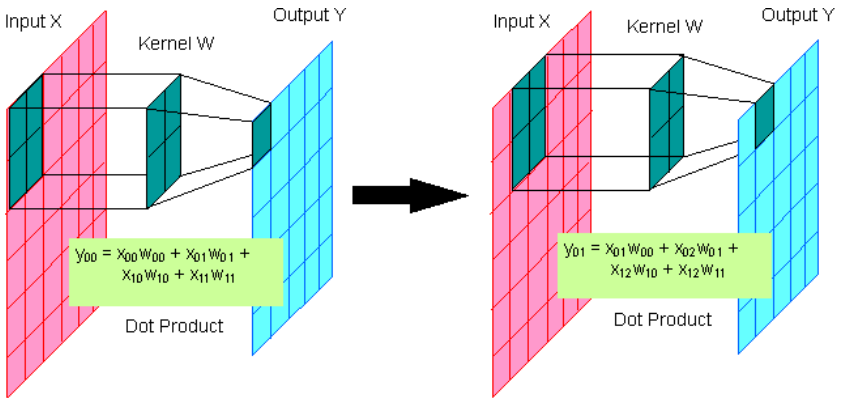


Figura 2.8 Convoluzione con un filtro: primi due step

In base a quanto visto nel paragrafo sulla convoluzione, la dimensione delle feature maps sarà data da:

$$(M - m + 1) \times (M - m + 1)$$

dove M è la dimensione dell'immagine in input e m è la dimensione del kernel.

Ad esempio, nella rete in Figura 2.9, le 4 feature maps del primo strato di convoluzione sono prodotte dalla convoluzione dell'immagine originale con dimensione 32×32 con 2 diversi filtri (5×5). La loro dimensione sarà data quindi da:

$$(32 - 5 + 1) \times (32 - 5 + 1) = 28 \times 28$$

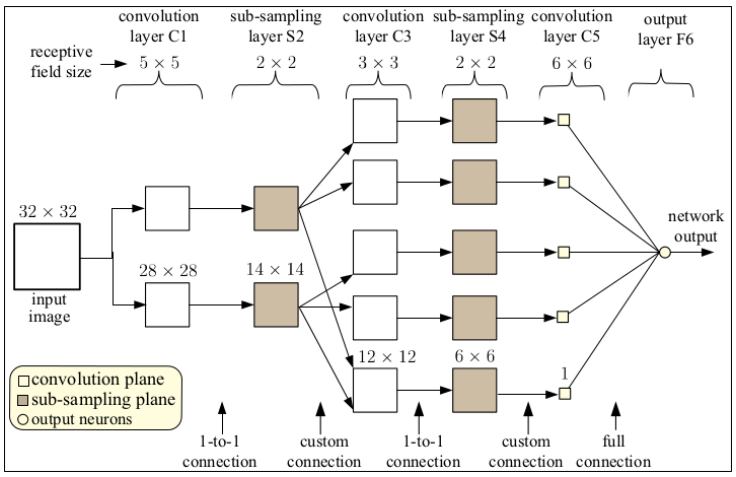


Figura 2.9 Convolutional Neural Network: modifica delle dimensioni delle immagini lungo il percorso da input ad output

Le unità in uno strato di convoluzione saranno quindi organizzate in piani, le *feature maps* stesse, all'interno dei quali tutte le unità condividono gli stessi pesi, dati dai valori degli elementi del kernel corrispondente. Tali pesi, inizializzati in modo random, vengono aggiustati durante l'addestramento della rete tramite l'algoritmo di *back-propagation*. In questo modo CNN impara autonomamente ad estrarre le *features* più significative. Le unità in una stessa *feature map*, condividendo gli stessi pesi, eseguono la stessa operazione su tutta l'immagine in input.

I kernel possono essere considerati come una sorta di campi recettivi locali, che fungono da rilevatori elementari di *features*. Quindi forzando le unità di una *feature map* ad avere gli stessi valori dei pesi, esse cercheranno le stesse "strutture" in tutta l'immagine con cui i kernel sono convoluti. Unità appartenenti a *feature maps* differenti hanno invece pesi differenti e cercano quindi strutture diverse.

Per mezzo di questa struttura abbiamo, in ogni strato, la possibilità di estrarre diversi tipi di features da ogni posizione delle immagini in input.

In sintesi, ogni unità di uno strato di convoluzione riceve in input informazioni provenienti da un insieme di unità localizzate in un piccolo "intorno" dello strato precedente, le cui dimensioni sono date dalle dimensioni dei kernel. Tramite i campi recettivi locali i neuroni estraggono dalla scena *features* elementari (come bordi orientati, angoli o *end-points*). Queste *features* vengono poi combinate dagli strati successivi per formare *features* più complesse.

L'idea di connettere le unità a campi recettivi locali nell'input risale al *Perceptron* dei primi anni '60, coincidente con la scoperta da parte di Hubel e Wiesel dei neuroni con caratteristiche *locally-sensitive* e *orientation-selective* nel sistema visivo dei gatti. Sono stati identificati due tipi base di cellule: le cellule semplici (S) e le cellule complesse (C) . Le cellule S rispondono a stimoli simili ai bordi, interni ai loro campi recettivi. Le cellule C sono localmente invarianti all'esatta posizione dello stimolo.

Una proprietà interessante degli strati di convoluzione consiste nel fatto che se l'immagine in input viene traslata, l'output della feature map sarà traslato della stessa quantità ma rimarrà invariato altrove. Questa proprietà è alla base della robustezza delle convolutional networks rispetto alle traslazioni e alle distorsioni dell'input.

Una volta che le features vengono rilevate, la "conoscenza" della loro posizione non solo diventa meno importante per identificare il pattern (solo la posizione rispetto alle altre features è rilevante), ma è potenzialmente "dannosa" poiché l'obiettivo è quello di cercare le stesse strutture in tutta l'immagine. Un modo semplice per ridurre la precisione con cui la posizione delle diverse features è registrata su una feature map è

quello di ridurre la risoluzione spaziale della stessa. Questo può essere fatto tramite i cosiddetti strati di subsampling, che solitamente eseguono una media locale, riducendo la risoluzione della feature map e quindi la sensibilità alle traslazioni e distorsioni. L'operazione di subsampling consiste nella partizione dell'immagine in un set di rettangoli non sovrapposti seguita dalla sostituzione di ogni sub-regione con la media corrispondente. Ad esempio, se le unità in uno strato di subsampling hanno dimensione 2×2, ognuna di esse restituirà la media di 4 punti. Le feature maps in uno strato di subsampling avranno, in questo modo, metà delle righe e delle colonne di quelle nello strato precedente.

A differenza di ciò che accade negli strati di convoluzione, il numero di immagini prodotte da uno strato di subsampling è sempre uguale a quello di immagini entranti.

Terminati gli strati di convoluzione e subsampling, le feature maps dello strato finale vengono "srotolate" in vettori e affidate ad una rete neurale tradizionale che esegue la classificazione finale.

La combinazione convoluzione/subsampling, ispirata dalle nozioni di Hubel e Wiesel di cellule semplici e complesse, era già stata implementata nel Neocognitron di Fukushima ma, all'epoca, non esistevano procedure di addestramento supervisionato come il back-propagation. Con questa progressiva riduzione della risoluzione spaziale, compensata da un progressivo aumento della ricchezza di rappresentazione (il numero di feature maps) è possibile raggiungere un ampio grado di invarianza rispetto alle trasformazioni geometriche dell'input.

Poiché tutti i pesi vengono aggiustati tramite back-propagation, le convolutional networks producono autonomamente il loro estrattore di features. La tecnica di condivisione dei pesi ha l'interessante conseguenza di ridurre il numero di parametri liberi.

Negli ultimi anni le CNN sono protagoniste di molte applicazioni di successo, come il riconoscimento di caratteri scritti a mano, quello di caratteri stampati e quello facciale[26], [27], [28].

Mean vs max-pooling

Un concetto importante riguardante le CNN è quello del cosiddetto *max-pooling*, operazione che rappresenta una sorta di subsampling non-lineare. Come descritto qualche riga sopra, l'operazione tradizionalmente eseguita negli strati di subsampling consiste nella partizione dell'immagine in un set di rettangoli non sovrapposti seguita dalla sostituzione di ogni sub-regione con la sua media.

In un articolo del 2009 Yang et al. [24], pur non fornendo una giustificazione teorica, sostengono che le performance di classificazione riguardo a diversi "oggetti" e "scene" migliorano se l'operazione di media viene sostituita da quella di massimo (max-pooling).

Softmax vs SVM

Per i compiti di classificazione, molti modelli di DL usano, all'ultimo strato della gerarchia, una rete neurale con funzione di attivazione di tipo *softmax* (conosciuta anche come *multinomial logistic regression*).

La logistic regression è un classificatore probabilistico lineare. È parametrizzato da una matrice di pesi W e da un vettore di bias b. La classificazione viene eseguita proiettando i dati (punti nello spazio delle features) su un set di iperpiani, la distanza dai quali riflette la probabilità di appartenenza ad una data classe. Matematicamente, si può scrivere:

$$P(Y = i|x, W, b) = softmax_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

L'output del modello (o predizione) è quindi ottenuto prendendo l'argmax del vettore il cui i-esimo elemento è $P(Y = i|x)$

$$y_{pred} = argmax_i P(Y = i|x, W, b)$$

Nell'articolo *Deep Learning using Linear Support Vector Machine* di Yichuan Tang viene mostrato come, la sostituzione della rete neurale con una support vector machine porti a miglioramenti significativi delle performance su diversi datasets popolari come il *MNIST* (database pubblico di cifre scritte a mano), il *CIFAR-10* (10 classi di immagini varie prese da internet) e *ICML 2013* (che riguarda il riconoscimento delle espressioni facciali).

Capitolo 3

Metodi

3.1 Introduzione

I sistemi di CAD considerano il rilevamento di masse come un problema di classificazione binaria. Gli oggetti "bersaglio", le masse tumorali, devono essere separate dal tessuto normale. La maggior parte di tali sistemi è composta da due livelli: (a) la *detection*, responsabile dell'individuazione delle regioni sospette presenti sul mammogramma (*region of interest - ROI*) e quindi dell'eliminazione preventiva delle zone non a rischio; (b) la classificazione (*classification*) delle ROI in masse e tessuto sano.

In realtà entrambi i livelli eseguono un'operazione di classificazione. La differenza sta nel fatto che la *detection* classifica le regioni in "sospette" e "non sospette", scartando le seconde mentre la *classification* analizza solo le regioni "sopravvissute" al primo livello e le classifica in masse vere e falsi allarmi.

Una caratteristica essenziale degli algoritmi di detection sarà quindi un'alta sensibilità (TPF), in quanto ogni massa persa in questa fase, sarà esclusa dall'analisi in quelle successive. Di solito il prezzo da pagare per avere un'alta TPF consiste in un aumento anche della FPF.

L'obiettivo principale delle questa tesi è lo studio di nuove metodologie che si occupino della prima di queste due fasi e che possano migliorare le prestazioni ottenute con le tecniche tradizionali. In particolare si vuole implementare un algoritmo di *detection* alternativo a quello attualmente usato dal software di CAD Galileo [1-I], per vedere se sia possibile sostituirlo o integrare i due sistemi. Sistemi di questo tipo sono solitamente seguiti da fasi successive responsabili prima di una riduzione dei falsi positivi e poi della vera e propria classificazione delle ROI in tessuto sano e masse.

L'algoritmo usato è una *convolutional neural network (CNN)* che, come visto nel capitolo 2, consiste sostanzialmente in una rete neurale *back-propagation* con filtri bidimensionali addestrabili che operano sull'immagine.

La fase di detection comprende due tipi di segmentazione: una esterna, responsabile dell'eliminazione dal mammogramma delle parti che non appartengono alla

mammella (fondo) ed una interna, che esclude le regioni della mammella che con alta probabilità non contengono masse.

Per l'addestramento della rete sono stati usati ritagli quadrati di mammografie, alcuni contenenti masse ed altri porzioni di tessuto sano, ricavati da immagini del *DDSM (Digital Database for Screening Mammography)* [3-I].

Per valutare le prestazioni del sistema si è usato il metodo della curva *ROC (Receiver Operating Characteristic)*. Con la configurazione migliore dei parametri della CNN si è raggiunto un valore dell'area sotto la curva ROC pari a:

$$A_z = 0.94 \pm 0.01$$

Dove 0.01 rappresenta l'indeterminazione sulla media.

3.2 Digital Database for Screening Mammography

Il Digital Database for Screening Mammography (DDSM) è una risorsa di immagini mammografiche digitalizzate raccolte dalla University of Florida e disponibili gratuitamente su internet [3-I]. Lo scopo del progetto è quello di fornire un grande database di mammografie in formato digitale per facilitare lo sviluppo, la valutazione e il confronto di algoritmi informatici di Computer Aided Detection.

Il database, completato nel 1999, è composto da circa 2600 casi, ognuno dei quali contiene quattro immagini, due per ogni seno *(right MLO, right CC, left MLO, left CC)*, insieme ad alcune informazioni sul paziente (età, densità del seno) ed informazioni sull'immagine (tipo di scanner usato per la digitalizzazione, risoluzione spaziale).

Alle immagini che contengono lesioni tumorali sono associati files con informazioni sulla posizione, sul tipo e sul grado di evidenza della regione sospetta, che rendono possibile una verifica automatizzata delle prestazioni degli algoritmi di ricerca delle lesioni.

Bisogna dire che le immagini mammografiche del DDSM, pur rappresentando una fonte preziosissima per gli studi di sistemi di CAD, sono in un formato che complica leggermente la situazione rispetto a quella alla quale ci si troverebbe di fronte in un contesto medico reale. Esse infatti non sono propriamente mammografie digitali, ma sono state digitalizzate tramite scanner. Contengono infatti porzioni danneggiate a causa di diversi fattori: deposito di polvere sulle pellicole, graffi, artefatti introdotti dal processo di scannerizzazione; in più provengono da scanner diversi, con

risoluzione spaziale differente, e questo comporta una grande variabilità di luminosità che, sommata alla già ampia variabilità naturale nell'aspetto delle mammografie, complica un problema già in partenza non semplice.

3.3 Dataset

Come accennato nell'introduzione di questo capitolo, l'addestramento del sistema viene eseguito presentando alla CNN due insiemi di ritagli quadrati di immagine chiamati *crop*: il primo è formato dagli esempi positivi, ossia crop contenenti masse tumorali; il secondo, quello degli esempi negativi, è formato da crop contenenti tessuto senza lesioni.

Le masse considerate hanno dimensioni comprese tra 1 cm e 2.5 cm. Va puntualizzato però che tali dimensioni si riferiscono, in realtà, alla zona indicata dalle gound-truth: questa contiene interamente la massa ed ha quindi un'area maggiore rispetto ad essa (Figura 3.1). Le dimensioni effettive delle masse che consideriamo, quindi, saranno comprese tra valori inferiori a 1 cm e a 2.5 cm.

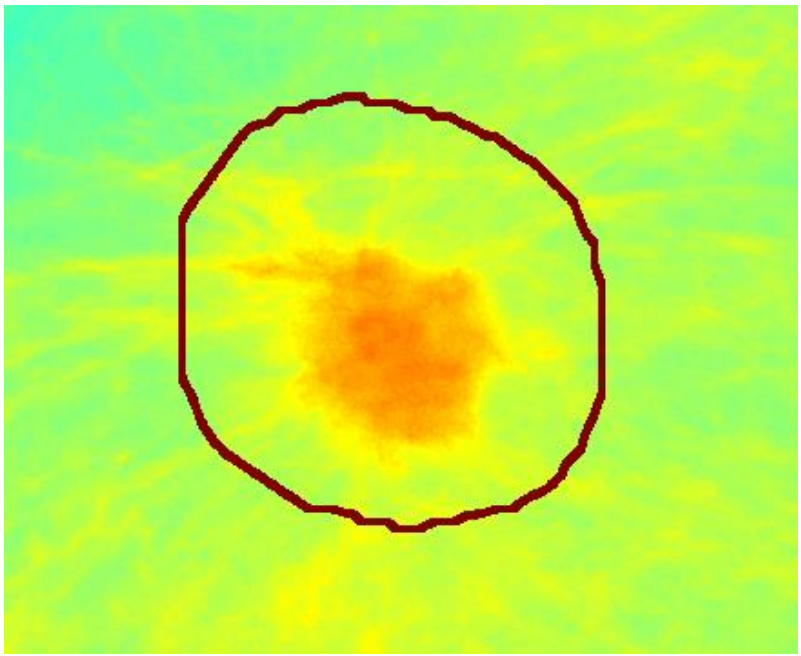


Figura 3.1 Contorno di una lesione generato tramite le informazioni *ground-truth* contenute nei file *overlay:* si noti come l'area interna al contorno sia superiore all'area effettiva della lesione

Gli esempi sono ottenuti ritagliando le finestre e ridimensionandole, per questioni computazionali, ad una grandezza prefissata di 64×64 pixel, la stessa dell'input della CNN.

Poiché le mammografie del DDSM non sono state tutte digitalizzate con il medesimo scanner, ma con scanner aventi risoluzioni differenti, nell'estrazione dei crop è necessario usare i parametri dimensionali corretti. Poiché le lesioni che ci interessano hanno dimensione massima paria a 2.5 cm, vogliamo dei crop il cui lato nella realtà corrisponda a questo valore. I tre scanner usati hanno risoluzioni paria a 42 µm, 43.5 µm, 50 µm: dovremmo quindi ritagliare finestre con un lato *L* pari rispettivamente a:

- $R = 50\ \mu m \rightarrow L = \frac{2.5\ cm}{50 \cdot 10^{-4}\ cm} = 500\ pixel$
- $R = 43.5\ \mu m \rightarrow L = \frac{2.5\ cm}{43.5 \cdot 10^{-4}\ cm} = 575\ pixel$
- $R = 42\ \mu m \rightarrow L = \frac{2.5\ cm}{42 \cdot 10^{-4}\ cm} = 595\ pixel$

Dimensioni simili, però, non sono generalmente gestibili dalle CNN per questioni computazionali riguardanti il numero di parametri coinvolti e quindi il numero di esempi necessari per un corretto addestramento. Dobbiamo quindi eseguire sull'immagine un'operazione di *subsampling* (Figura 3.2) per ridurre la dimensione dei crop ad una più ragionevole. Questa operazione consiste, nella sua versione classica, nella partizione dell'immagine in un set di quadrati non sovrapposti e nella sostituzione di ogni sub-regione con la sua media. Il lato di tali sub-regioni è detto fattore di subsampling *N* e, indicando una misura di lunghezza in pixel, deve appartenere ai numeri interi. Quindi, data una certa risoluzione dell'immagine, *R*, una volta eseguito il subsampling la risoluzione sarà *R·N*.

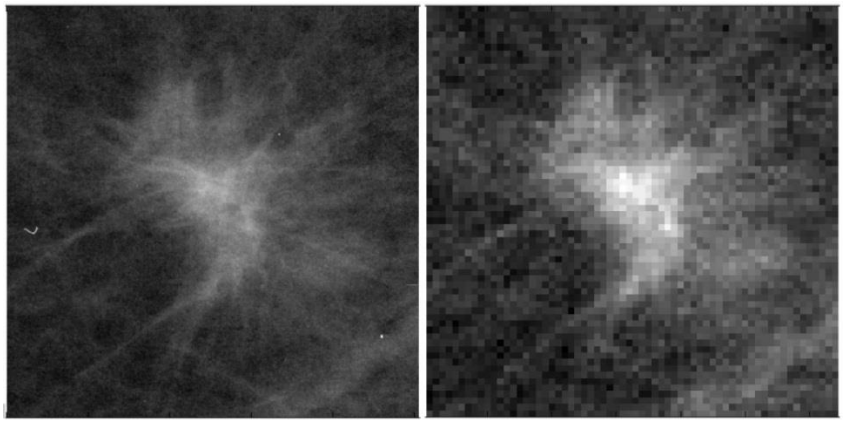


Figura 3.2 Immagine di una massa prima e dopo il *subsampling*

In base alla risoluzione dello scanner, quindi, dovremo usare un fattore di subsampling differente, calcolato dividendo la dimensione della finestra per 64 ed approssimando all'intero più vicino:

- $L = 500 \quad \rightarrow \quad L/64 = 7.8 \rightarrow \quad N = 8;$
- $L = 595 \quad \rightarrow \quad L/64 = 9.3 \rightarrow \quad N = 9;$
- $L = 575 \quad \rightarrow \quad L/64 = 9.0 \rightarrow \quad N = 9;$

Così a seconda dello scanner e quindi della risoluzione, avremo:

- $R = 50\ \mu m \quad \rightarrow \quad N = 8$
- $R = 43.5\ \mu m \quad \rightarrow \quad N = 9$
- $R = 42\ \mu m \quad \rightarrow \quad N = 9$

Quindi, poiché la dimensione lineare delle finestre estratte dev'essere divisibile per il fattore di subsampling, introduciamo un'altra approssimazione: dalle mammografie con risoluzione 50 µm estraiamo crop con lato pari a 512 pixel, ossia l'intero divisibile per 8 più vicino a 500; da quelle con risoluzione 43.5 µm e 42 µm estraiamo crop con lato 576, l'intero divisibile per 9 più vicino a 595 e a 575. Infatti $512/8 = 64$ e $576/9 = 64$. Abbiamo quindi:

- $R = 50\ \mu m \quad \rightarrow \quad N = 8 \quad \rightarrow \quad L = 512$
- $R = 43.5\ \mu m \quad \rightarrow \quad N = 9 \quad \rightarrow \quad L = 576$
- $R = 42\ \mu m \quad \rightarrow \quad N = 9 \quad \rightarrow \quad L = 576$

In questo modo le dimensioni reali effettive (D) delle finestre e quindi delle masse saranno pari a:

- $R = 50\ \mu m \quad \rightarrow \quad D = 512 \times 50\ \mu m = 2.56\ cm$
- $R = 43.5\ \mu m \quad \rightarrow \quad D = 576 \times 43.5\ \mu m = 2.51\ cm$
- $R = 42\ \mu m \quad \rightarrow \quad D = 576 \times 42\ \mu m = 2.42\ cm$

Ripetendo un ragionamento analogo con la dimensione minima (1 cm) avremo:

- $R = 50\ \mu m \quad \rightarrow \quad L = 200$
- $R = 43.5\ \mu m \quad \rightarrow \quad L = 230$
- $R = 42\ \mu m \quad \rightarrow \quad L = 238$

corrispondenti a dimensioni reali di:

- $R = 50\ \mu m \quad \rightarrow \quad D = 200 \times 50\ \mu m = 1.00\ cm$
- $R = 43.5\ \mu m \quad \rightarrow \quad D = 230 \times 43.5\ \mu m = 1.00\ cm$

- $R = 42\ \mu m \quad \rightarrow \quad D = 238 \times 42\ \mu m = 1.00\ cm$

In sintesi le dimensioni reali delle masse considerate saranno comprese tra

$$1.00\ cm < D < 2.56\ cm$$

Esempi negativi

Per l'acquisizione degli esempi negativi partendo da mammografie appartenenti a casi di tipo *normal*, che sicuramente non contengono lesioni, si sono estratti i crop in posizioni casuali della mammella, con almeno l'80% dell'area del crop appartenente ad essa.

Prima dell'estrazione, le immagini sono state modificate con qualche operazione di *pre-processing*: filtro mediano per l'attenuazione del rumore, *dilation*, *erosion*, *opening* e *closing* per l'eliminazione degli artefatti.

È necessaria inoltre un'operazione di binarizzazione: questa serve per determinare la posizione della mammella nell'immagine e quindi evitare l'estrazione di crop da zone appartenenti al fondo. Data la grande variabilità di luminosità delle mammografie considerate, dovuta a diversi fattori, non è stato però possibile usare una *soglia assoluta*. Oltre alla grande eterogeneità intrinseca dei tessuti da paziente a paziente, un altro fattore consiste nel fatto che le immagini sono state acquisite analogicamente e poi digitalizzate, oltretutto con scanner differenti.

L'utilizzo di una soglia assoluta infatti introdurrebbe due tipi di errore: nelle mammografie meno luminose avremmo un'eliminazione di alcune parti della mammella (quelle più scure), impedendo al sistema di analizzarle e quindi di rilevare eventuali lesioni presenti (Figura 3.3); in quelle più luminose invece verrebbero selezionate anche regioni appartenenti al fondo e non alla mammella (quelle più chiare) che, per ovvie ragioni, non possono contenere lesioni: questo implicherebbe un aumento inutile del tempo di scansione (Figura 3.4).

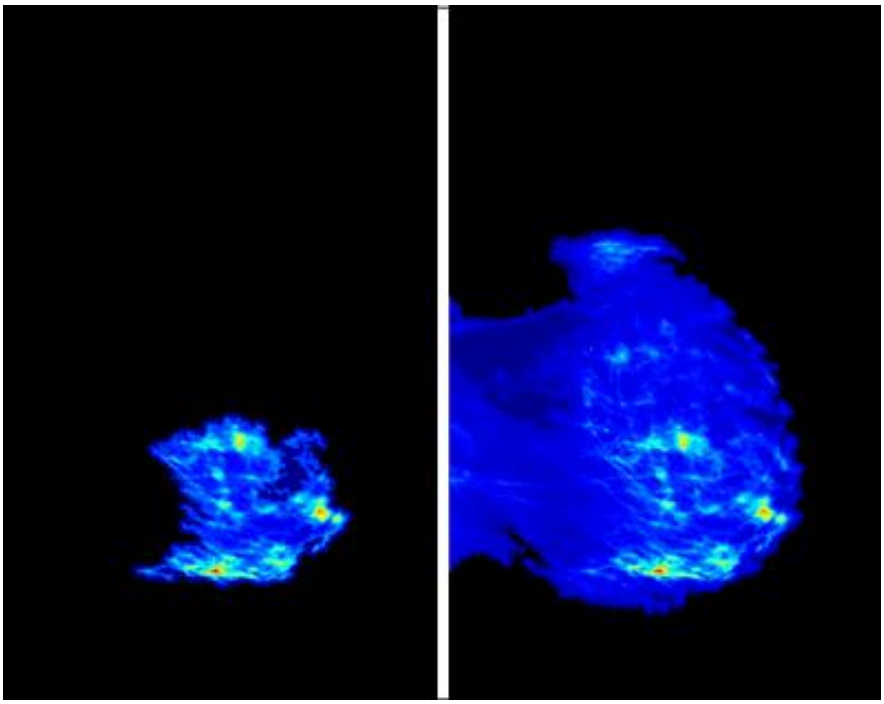


Figura 3.3 Soglia assoluta (sinistra) e soglia adattiva (destra) nel caso di mammografia poco luminosa (le zone nere sono quelle escluse dall'analisi)

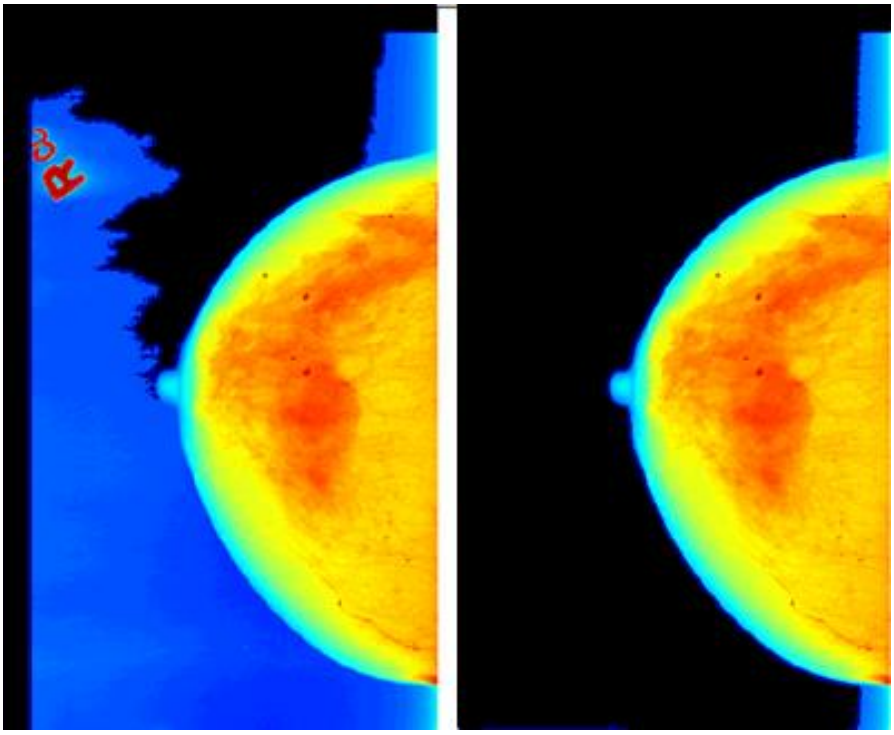


Figura 3.4 Soglia assoluta (sinistra) e soglia adattiva (destra) nel caso di mammografia molto luminosa (le zone nere sono quelle escluse dall'analisi)

È stato quindi necessario trovare un metodo per la ricerca automatica della soglia adatta alla mammografia considerata (*soglia adattiva*). Si è proceduto in questo modo: dopo aver eseguito un'operazione di stretching dei livelli di grigio, si è interpolato il primo picco dell'istogramma con una gaussiana (Figura 3.5) e, una volta ricavate la media (μ) e la deviazione standard (σ) si è impostata la soglia (S) a:

$$S = \mu + 0.4 \times \sigma$$

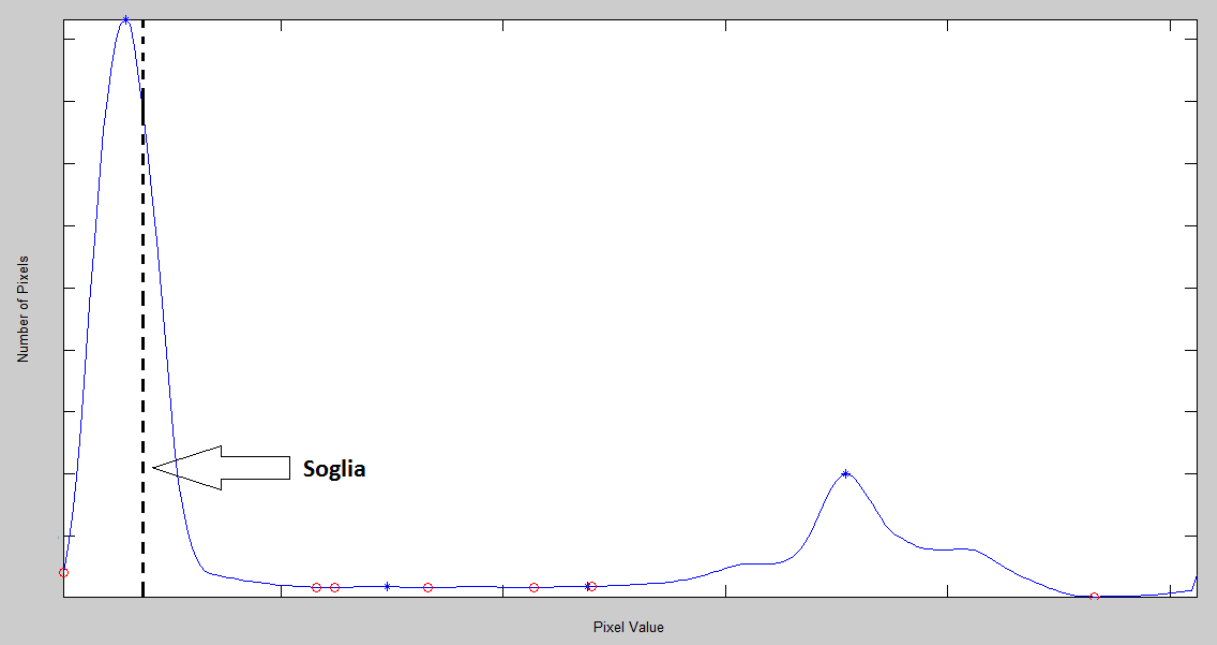


Figura 3.5 Soglia con interpolazione del primo picco (fondo) con una gaussiana

Quindi si è proceduto con l'estrazione vera e propria dei crop, ottenendo l'insieme degli esempi negativi di addestramento.

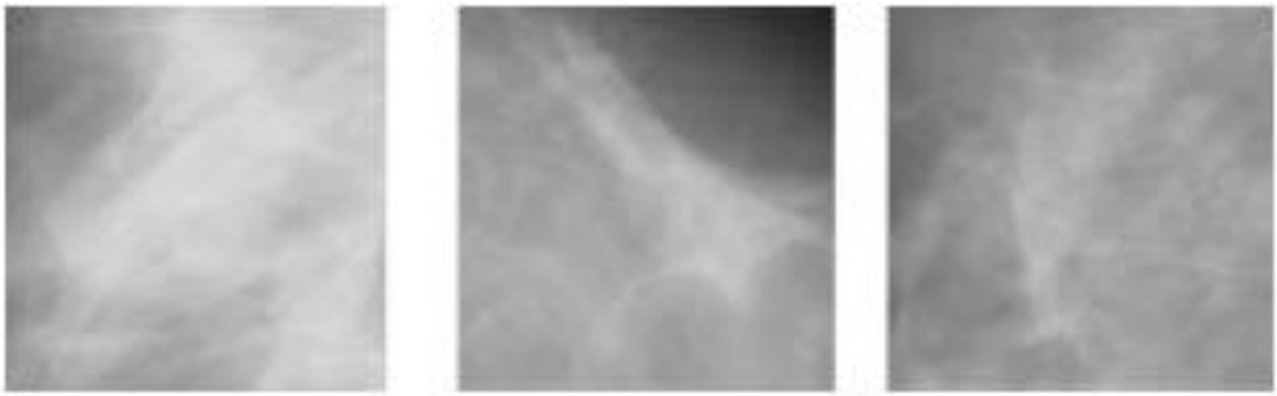


Figura 3.6 Esempi negativi

Esempi positivi

Per quanto riguarda gli esempi positivi, per automatizzare il processo di estrazione, ci siamo serviti dei file *overlay* a disposizione sul DDSM. Tali file contengono le cosiddette *ground-truth information*, che contengono, per le mammografie appartenenti ai casi *cancer*, la posizione, le dimensioni e la morfologia delle lesioni: da questi file si possono ricavare le istruzioni necessarie per generare il contorno della lesione, pixel per pixel, sulla mammografia. In questo modo possiamo selezionare automaticamente solo le lesioni che ci interessano, cioè quelle di tipo massa e con dimensioni comprese tra 1 cm e 2.5 cm. Dopo aver eseguito sull'immagine le stesse operazioni di pre-processing usate per i negativi, ritagliamo il crop centrato sulla lesione procedendo nel modo seguente:

1. Si crea il bounding box della massa e si sceglie la maggiore tra le due dimensioni, l.
2. Si centra il bounding box in una finestra quadrata con lato l.
3. La finestra con lato l viene centrata in un'altra con lato $L > l$ (L è calcolato tramite il procedimento esposto in precedenza).
4. Si crea il crop estraendo i pixel interni alla finestra quadrata più grande
5. Si esegue il subsampling del crop per ridurne la dimensione a 64×64, che è anche la dimensione dell'input della CNN. L'operazione di subsampling consiste nella partizione dell'immagine in un set di rettangoli non sovrapposti seguita dalla sostituzione di ogni sub-regione con la sua media.

Figura 3.7 Esempi positivi

Il dataset così ottenuto, formato da 418 esempi negativi e 174 esempi positivi, verrà usato per l'addestramento del sistema.

3.4 Selezione del modello

Il passo successivo alla creazione del dataset è la selezione del modello, ossia la ricerca dell'architettura ottimale e del giusto valore dei parametri ed iperparametri del sistema. Come detto, uno degli obiettivi della tesi è il tentativo di miglioramento dei risultati ottenuti da Chan. Per farlo si è scelto, innanzitutto, di utilizzare un'architettura più profonda, con due livelli (ognuno composto da uno strato di convoluzione ed uno di subsamling). Inoltre, vista la maggior potenza computazionale fornita dai computer attuali rispetto a quelli disponibili nel 1996, si è fissata la dimensione dell'input a 64×64, contro il 16×16 e 32×32 usate da Chan.

La scelta dell'architettura riguarda:

- Dimensione dell'input
- Numero di livelli, ossia il numero di coppie di strati di convoluzione e subsampling
- Numero e dimensione dei filtri degli strati di convoluzione
- Tipo di operazione eseguita negli strati di subsampling: media o max
- Tipologia dello strato finale: softmax o SVM

I parametri del modello, i cui valori vengono inizializzati in modo casuale e il cui aggiustamento avviene durante la fase di addestramento sono:

- I valori delle unità dei filtri (kernel) di convoluzione
- I pesi dello strato finale completamente connesso

I valori iniziali delle unità dei kernel sono inizializzati random come segue:

$$w = (rand - 0.5) \cdot 2\sqrt{\frac{6}{fan_{in} + fan_{out}}}$$

dove $rand$ rappresenta un numero casuale estratto tra 0 e 1 fan_{in} e fan_{out} sono dati rispettivamente da:

- $fan_{in} = f_{l-1} \cdot {k_{l-1}}^2$
- $fan_{in} = f_l \cdot {k_l}^2$

dove: l e $l-1$ sono rispettivamente l'indice dello strato dei kernel considerati e l'indice di quello precedente; k_{l-1} e k_l sono le dimansioni dei kernel dei rispettivi strati.

I valori dei pesi dello strato finale invece sono inizializzati a:

$$w = (rand - 0.5) \cdot 2\sqrt{\frac{6}{n}}$$

dove n è il numero di pesi dello strato.

Gli iperparametri riguardanti la fase di training del sistema fase sono:

- Il numero di epoche di addestramento
- Il tasso di apprendimento (learning rate)

La scelta dell'architettura della rete e dei parametri ottimali è stata fatta tramite un procedimento di *shaking* casuale del dataset, simile alla *cross-validation*. Al variare dell'architettura e della configurazione dei parametri

1. si divide il dataset in due partizioni, il training set (80% degli esempi) ed il test set (20% degli esempi)
2. si addestra il sistema con gli esempi del training set e si calcola l'errore sul test set
3. si ripetono i primi due passi per 10 volte, con diverse composizioni del train e del test ottenute da permutazioni random degli esempi, e si calcola la media degli errori sul test set
4. si scelgono l'architettura e la configurazione con errore medio minore

L'architettura ottimale, scelta secondo il criterio descritto è risultata quella con le seguenti caratteristiche:

1° strato: convoluzione con 5 filtri di dimensione 5×5

2° strato: subsampling con max-pooling

3° strato: convoluzione con 3 filtri di dimensione3×3
4° strato: subsampling con max-pooling
5° strato (finale): SVM

Gli iperparametri di training ottimali, ottenuti con il medesimo procedimento sono:

- Numero di epoche = 3000
- Learning rate = 0.02

3.5 Training

Una volta scelta l'architettura migliore e la configurazione ottimale dei parametri, si usano gli esempi ottenuti col procedimento descritto nel paragrafo 3.1 per l'addestramento del sistema.

Normalizzazione

Per rendere l'addestramento efficace è necessario normalizzare gli esempi di training rispetto ai livelli di grigio. La normalizzazione più efficace dal punto di vista delle performance è stata ottenuta col seguente procedimento:

1. Si sottrae ad ogni immagine la propria media, ottenendo così dei valori dei pixel centrati attorno allo zero
2. Per ogni posizione dei pixel $((1,1),(1,2),\dots,(64,64))$ si calcolano media e deviazione standard su tutte le immagini
3. Si sottrae ad ogni pixel la media di quel pixel fatta su tutti gli esempi di training
4. Si divide ogni pixel per la deviazione standard di quel pixel (rispetto alla media fatta sugli esempi di training)

Successive Enhancement Learning (SEL)

La maggiore difficoltà nell'addestramento di un tale sistema è legata alla caratterizzazione delle "non masse". Infatti, mentre gli esempi positivi hanno una struttura abbastanza definita, quelli negativi hanno una gande variabilità e non ne esistono esempi tipici. Avendo come obiettivo quello di trovare un piccolo numero di casi positivi tra una quantità enorme di casi negativi, anche se l'errore commesso dal sistema in classificazione è basso, tendenzialmente esso troverà un numero molto grande di falsi positivi. Si potrebbe quindi pensare di fare un addestramento con più esempi negativi. Tuttavia l'utilizzo delle CNN mostra una particolarità: come per altri algoritmi di machine learning, il training set deve essere bilanciato tra numero di esempi negativi e positivi, altrimenti il sistema "impara negativo" (nel senso che tende a classificare tutto come negativo) e si osserva un degrado delle prestazioni (Tabella 3.1, Figure 3.8 e 3.9). Risulta quindi necessario trovare un criterio per la selezione degli esempi negativi, in modo da mantenere basso sia il numero di falsi positivi che quello di falsi negativi.

Composizione train set (n° es. neg. : n° es. pos.)	1 : 1	1.5 : 1	2 : 1	2.5 : 1	3 : 1	3.5 : 1
***Media F.P. per mammografia* ± indeterminazione sulla media**	47 ± 3	46 ± 3	42 ± 3	34 ± 3	21 ± 2	Il sistema predice tutto negativo
FN (lesioni mancate)	2%	4%	4%	10%	28%	100%

Tabella 3.1 Performance al variare del bilanciamento del train set

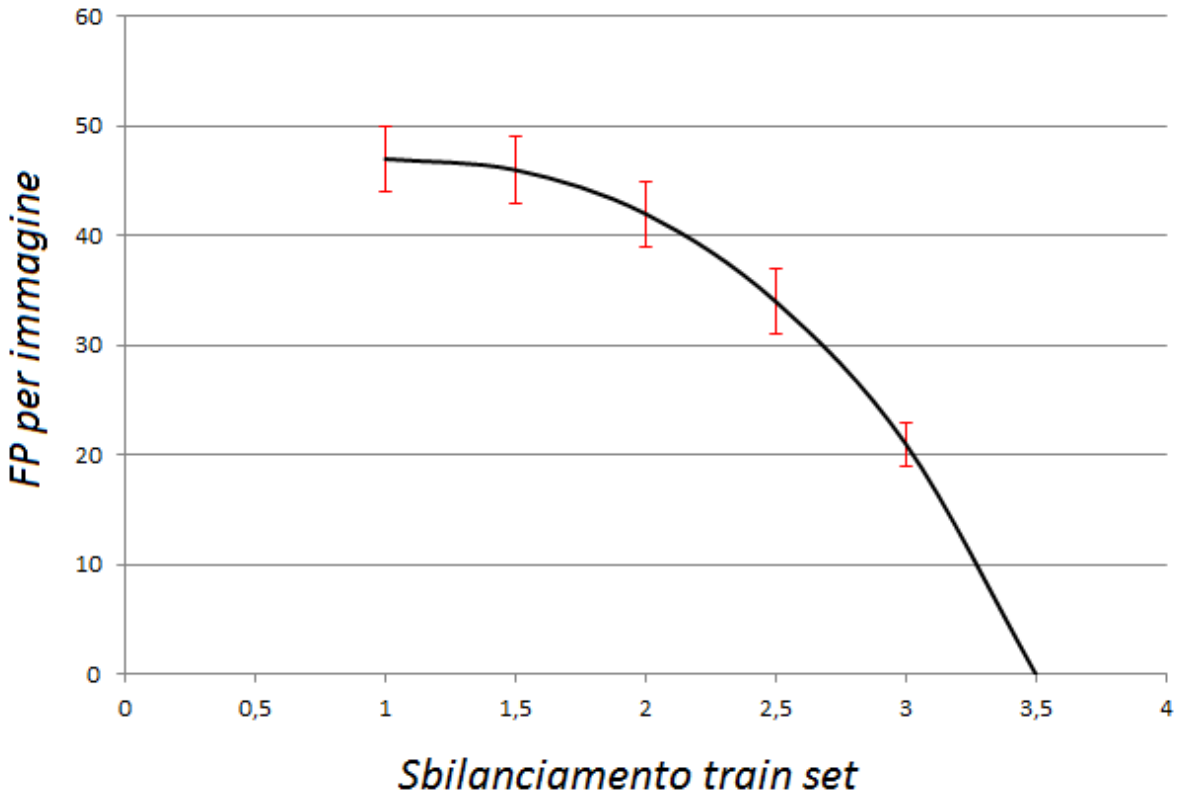


Figura 3.8 Performance al variare del bilanciamento del train set: falsi positivi per immagine

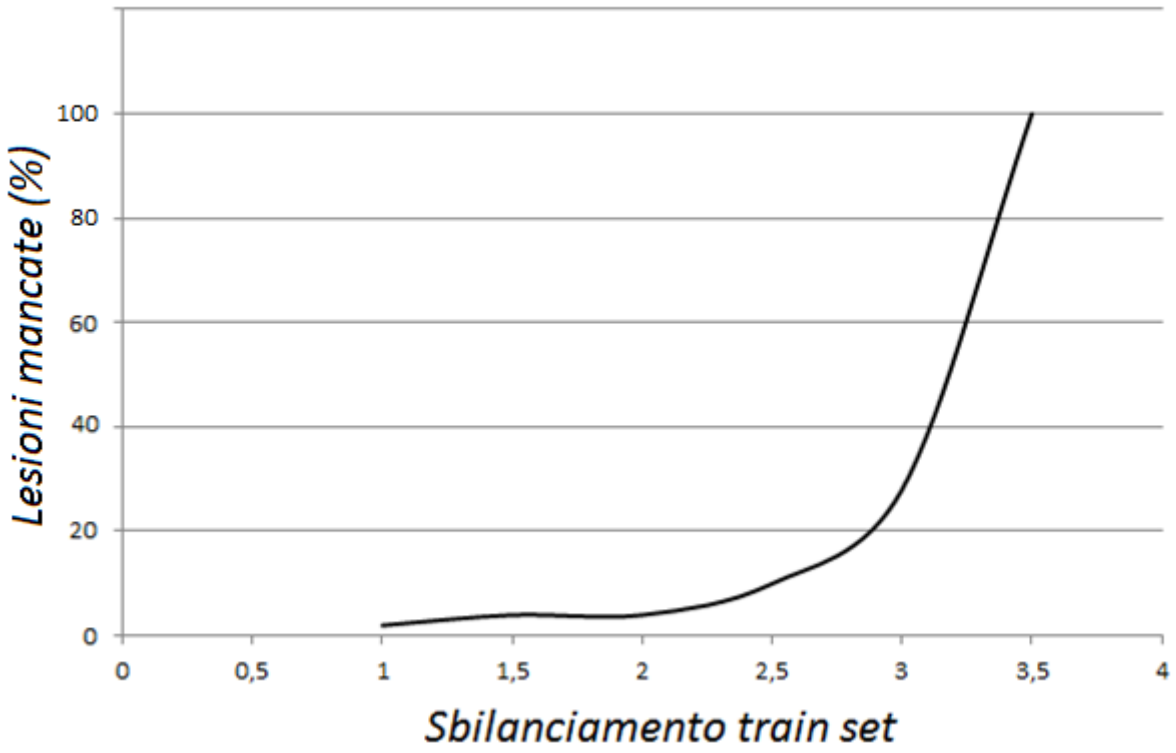


Figura 3.9 Performance al variare del bilanciamento del train set: lesioni mancate

Si è notato innanzitutto che addestrando il sistema con esempi negativi completamente casuali, in fase di test si ha una concentrazione dei falsi positivi soprattutto in due zone: quella del muscolo pettorale e quella a cavallo tra la pelle e il tessuto interno alla mammella. Si è quindi deciso di estrarre dei crop da queste regioni e di sostituirli ad alcuni dei negativi random: il 10% circa del numero totale di negativi è sostituito da crop provenienti dal pettorale e un altro 10% da crop provenienti dalla zona della pelle.

Questa tecnica porta già ad una diminuzione non trascurabile della frazione di falsi positivi senza perdite significative sulla sensibilità. In Figura 3.10 è mostrata la differenza di risultati riguardanti la scansione di una mammografia normale (senza lesioni) prima e dopo l'aggiunta al train set dei crop provenienti dalla zona del bordo pelle-tessuto. La differenza nel numero di falsi positivi è evidente.

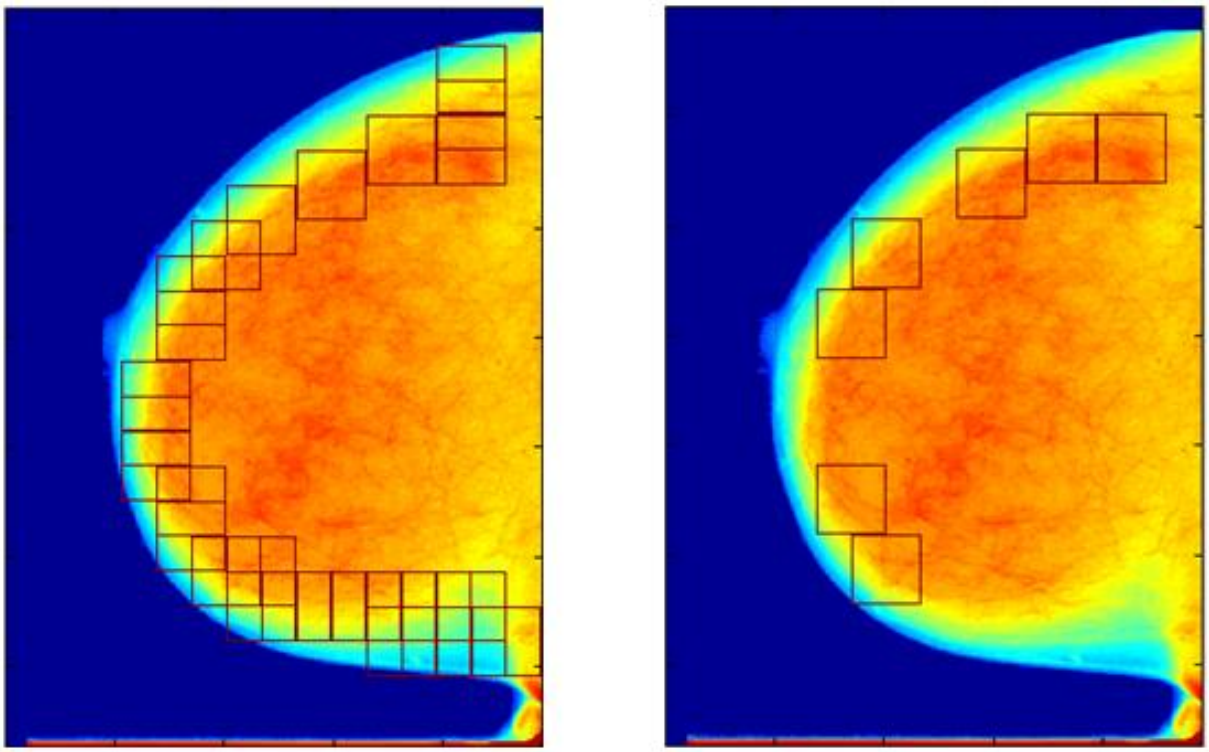


Figura 3.10 scansione prima e dopo l'aggiunta al train set dei crop provenienti dal bordo pelle-tessuto: le finestre con bordo rosso rappresentano i crop classificati positivi (tutti FP perché la mammografia non contiene lesioni)

Volendo migliorare ulteriormente la specificità del sistema si è cercato un metodo, più efficace di quello casuale, per la selezione del rimanente 80% dei negativi: tale criterio è definito in letteratura come *successive enhancement learning (SEL)* [8], la cui versione tradizionale funziona in questo modo:

1. Si addestra il sistema con un set N_1 esempi negativi scelti in modo casuale tra quelli a disposizione N_{tot}
2. Si testa su tutti i rimanenti (insieme Z)
3. Si sostituiscono gli esempi più "facili" di N_1 (cioè quelli che in output della rete vengono classificati negativi con le probabilità maggiori) con quelli "difficili" di Z (cioè quelli classificati impropriamente dalla rete)
4. Si ripetono i passi 2. e 3. finché le prestazioni raggiunte sono considerate accettabili (o fino all'esaurimento degli esempi a disposizione)

Ad ogni iterazione la rete viene riaddestrata per 1000 epoche inizializzando i pesi con quelli ottenuti all'iterazione precedente. Il compromesso tra FP per mammografia e numero di lesioni rilevate, è stato raggiunto alla 2 iterazione del SEL, quindi dopo un numero di epoche complessive pari a 3000.

Si è notato tuttavia che questo procedimento, pur portando ad una considerevole diminuzione dei falsi positivi, ha il difetto di aumentare la frazione di falsi negativi. Questo si traduce in una perdita di lesioni, una situazione ovviamente da evitare. Per controbilanciare questa tendenza si è provato ad eliminare dal training set più negativi di quanti poi se ne inseriscano (in un rapporto circa uguale a 1.5:1), "sbilanciando" quindi verso i positivi. In questo modo si è riusciti a ridurre considerevolmente la frazione di falsi positivi senza perdite significative sulla sensibilità. Nella tabella 3.2 sono mostrate le performance relative alle diverse iterazioni del SEL ottenute fissando lo stride al 23% della dimensione lineare della finestra di scansione: sono riportati i dati relativi alla percentuale media del numero di crop falsi positivi per mammografia. Si è scelto questo valore dello stride poiché un valore maggiore comporta la perdita di un numero di lesioni troppo grande, mentre uno minore ha come conseguenza un aumento inaccettabile delle false positive fraction.

Iterazione SEL	0	1	2	3
Crop F.P. per mammografia	144 ± 11	42 ± 3	35 ± 3	31 ± 3
FN (lesioni mancate)	0%	4%	2%	4%

Tabella 3.2 Performance iterazioni SEL

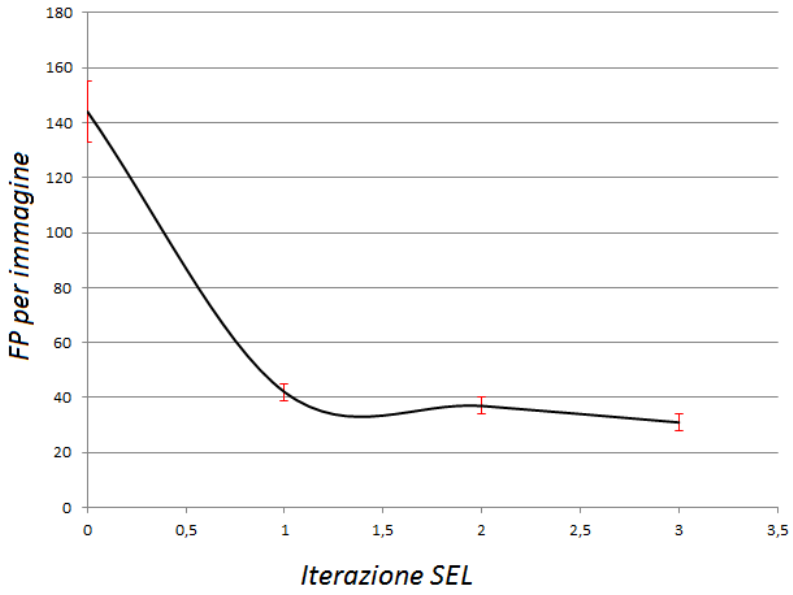


Figura 3.11 Performance iterazioni SEL: falsi positivi per immagine

Per quanto riguarda gli esempi di training positivi, invece, si è notato che dopo un addestramento eseguito solo con lesioni centrate nei crop il sistema non riusciva ad individuare le masse situate nella zona vicina al bordo della mammografia. Questo è dovuto al fatto che tali lesioni, durante la scansione, non sono mai centrate nella finestra e spesso sono anche parzialmente visibili (cioè sono tagliate dal bordo della mammografia): questa situazione, in effetti, è "sconosciuta" al sistema (addestrato con lesioni sempre centrate nel crop). Quindi si è deciso di aggiungere agli esempi di training dei casi di questo tipo (circa un 3% del numero totale di esempi positivi), con un evidente aumento della sensibilità.

3.6 Schema di detection

Il sistema di analisi di ricerca delle masse prende in input l'intera immagine mammografica ed è sostanzialmente composto da due fasi:

- segmentazione: riduzione dell'area dell'immagine alle sole zone che rappresentano il tessuto mammario
- individuazione delle ROI: ricerca, nelle immagini segmentate, di segnali con caratteristiche di una lesione

Solitamente, i sistemi di CAD fanno seguire a queste due fasi diversi livelli, responsabili della cosiddetta false positive reduction, ossia la riduzione del numero di falsi positivi provenienti dal primo livello.

Dopo qualche operazione di pre-processing per la riduzione del rumore e per l'eliminazione degli artefatti (le stesse effettuate sulle mammografie usate per l'addestramento), si esegue il subsampling con un fattore dipendente dalla risoluzione dello scanner (vedi paragrafo 3.1) e sulla mammografia viene fatta scorrere una finestra quadrata.

Per il "passo" di scorrimento (*stride*) di solito viene scelto un valore uguale ad una piccola percentuale della dimensione lineare della finestra (20-30%): in questo modo aumentano le probabilità che un'eventuale lesione sia ben centrata in almeno una finestra, evitandone la perdita. In Tabella 3.3, infatti, si può notare come all'aumentare dello stride aumenti anche la percentuale di lesioni mancate. Diminuendo troppo lo stride, invece, si ha un aumento dei crop totali estratti e un conseguente aumento di falsi positivi. Con uno stride molto basso, quindi, la sensibilità del sistema risulta elevata ma il numero di falsi positivi diventa inaccettabile.

Per ovvi motivi l'estrazione dei crop avviene solo nelle zone appartenenti alla mammella e non al fondo, anche se, in realtà, una piccola percentuale di

sovrapposizione con il fondo è necessaria per evitare di perdere eventuali lesioni situate lungo il bordo della mammella (Figura 3.12).

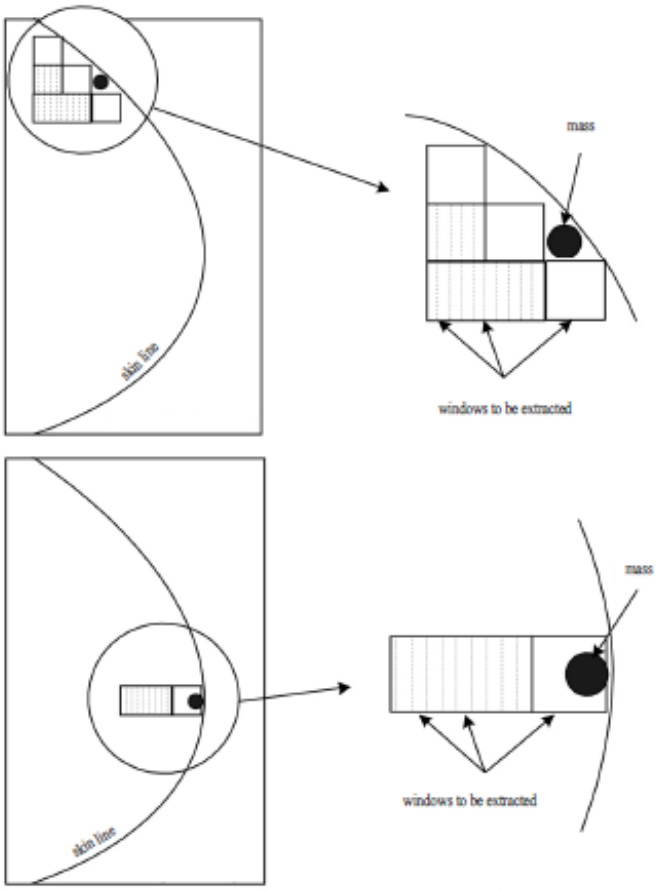


Figura 3.12 Problemi in fase di detection

Una volta estratti, i crop vengono inviati alla CNN addestrata: questa restituisce in output le probabilità di appartenenza alle due classi, positiva e negativa: ogni crop classificato come positivo identifica un'area giudicata come sospetta.

Le performance riguardanti il numero medio di falsi positivi per mammografia e le lesioni mancate al variare dello *stride* sono mostrati nella tabella 3.3 Essa contiene, dall'alto al basso:

- il valore dello stride, in percentuale rispetto alla dimensione lineare del crop
- il numero medio di crop totali estratti per mammografia
- il numero medio di crop classificati erroneamente come positivi (FP)
- la percentuale media di crop FP rispetto al numero totale
- la percentuale di lesioni mancate (secondo il criterio precedentemente esposto)

È chiaro che, come anticipato, al diminuire dello stride aumenterà anche il numero di crop estratti e, di conseguenza, il numero di FP. Ciò che rimarrà circa costante sarà invece il rapporto tra il numero di FP e il numero di crop totali estratti. Questo risulta, per tutti i valori dello stride, circa pari a:

$$\frac{FP}{CROP_{tot}} \approx 8.3\%$$

Stride (%)	**17**	**20**	**23**	**27**	**30**
Crop totali estratti	814 ± 50	587 ± 36	421 ± 25	348 ± 21	280 ± 17
Crop FP per mammografia	68 ± 6	49 ± 4	35 ± 3	29 ± 2	23 ± 2
Percentuale media di FP (%)	8.3 ± 0.7	8,3 ± 0.7	8,3 ± 0.7	8,2 ± 0.7	8,4 ± 0.7
Lesioni mancate	4%	4%	2%	6%	10%

Tabella 3.3 Risultati riguardanti la scansione delle 50 mammografie di test al variare dello *stride*

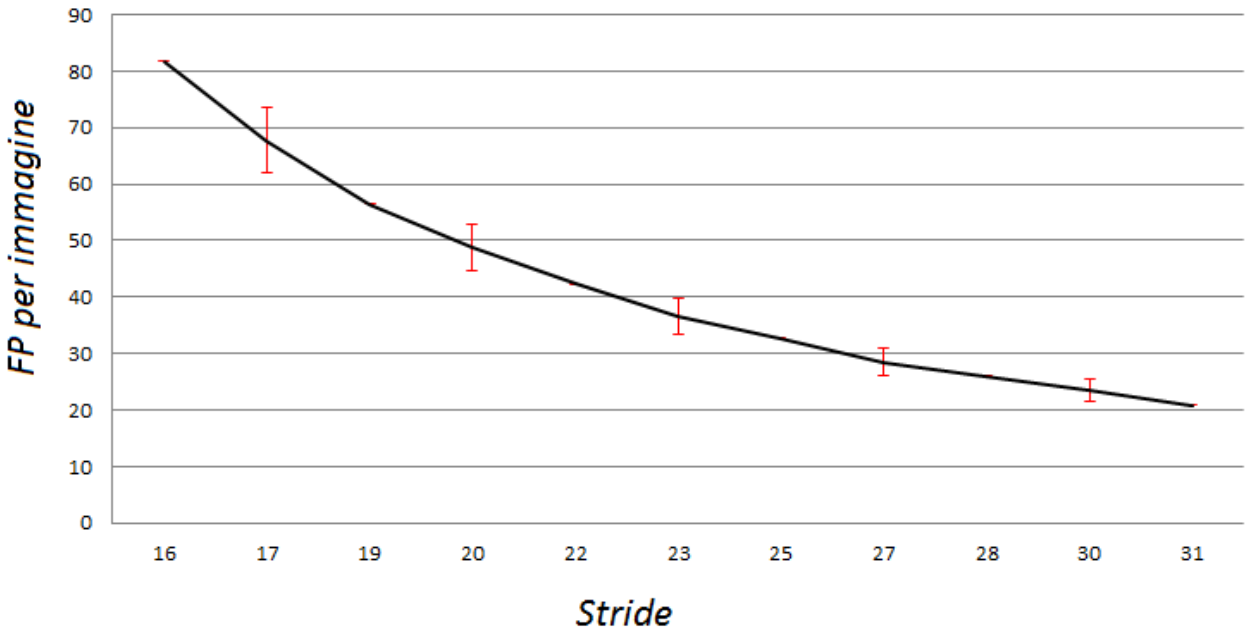
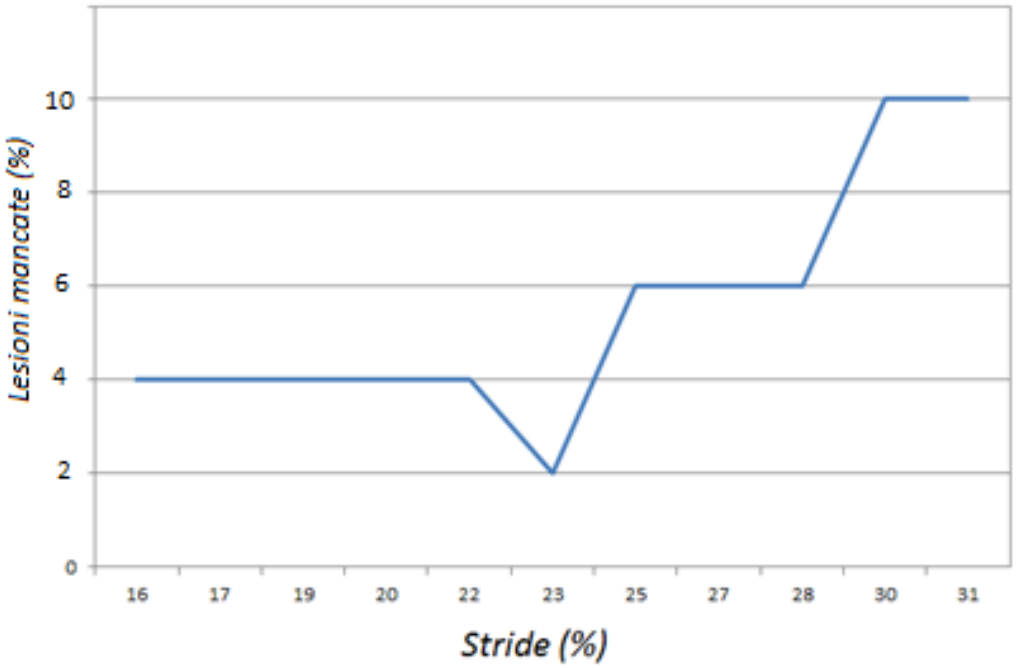


Figura 3.13 Numero medio dei crop FP per mammografia al variare dello *stride* (espresso in percentuale della dimensione della finestra di scansione)



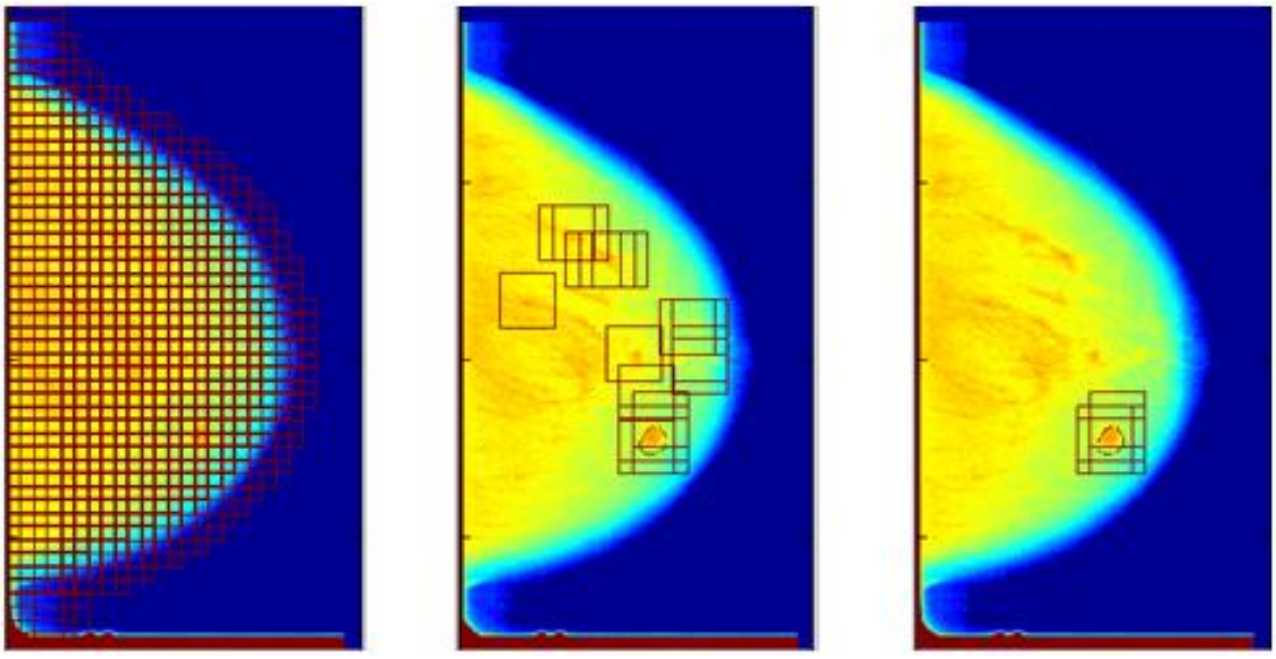


Figura 3.14 Schema di detection mediante scansione: tutti i *crop* estratti (sinistra), *crop* classificati come positivi (centro), *crop* veri positivi (destra); il contorno della lesione è ottenuto tramite i file overlay che contengono le *ground-truth*

3.7 False positive reduction

Abbiamo già detto che l'obiettivo primario della fase di detection di un CAD consiste nell'ottenere un'alta sensibilità, poiché una lesione persa in questa fase viene ignorata dai livelli successivi. Di solito il prezzo da pagare per ottenere una sensibilità elevata consiste in un alto numero di falsi positivi.

Come è stato accennato in precedenza l'addestramento del sistema con negativi completamente casuali comporta una FPF inaccettabile. È quindi necessario trovare un modo per ridurre il numero di falsi positivi. Per prima cosa si sono aggiunti al dataset di training crop provenienti dalla zona dei pettorali e dal confine pelle-tessuto perché una percentuale rilevante di FP riguarda proprio queste zone. Poi si è proceduto con la sopracitata tecnica del successive enhancement learning, utilizzata per selezionare i negativi più rappresentativi della loro classe (vedi paragrafo 3.3).

Capitolo 4
Risultati e conclusioni

4.1 Risultati

Per valutare le performance del sistema, lo abbiamo testato su 50 mammografie eterogenee dal punto di vista della densità dei tessuti (scelta random) e contenenti ognuna una sola massa. Il test set conterrà quindi mammografie che possono essere più o meno dense o più o meno grasse. Per quanto riguarda le dimensioni delle masse, invece, esse sono state scelte all'interno dello stesso *range* usato per l'addestramento.

Lo schema di detection è quello descritto nel paragrafo 3.6.

Per la verifica del rilevamento delle lesioni abbiamo usato, come per l'estrazione dei casi positivi di training, le *ground-truth information* contenute nei file overlay.

Una lesione viene considerata "trovata" se almeno uno dei crop estratti dalla mammografia considerata classificati come positivi dalla CNN rispetta una delle seguenti due condizioni:

- il suo centro giace all'interno della zona del contorno
- contiene almeno l'80% dell'area racchiusa dal contorno

Fissato lo stride al 23% della dimensione lineare della finestra di scansione, ovvero 15 pixel (rispetto ai 64 del lato della finestra), il sistema ha rilevato 49 masse su 50.

Questo equivale ad un valore della TPF pari a:

$$TPF = 0.98$$

A questo valore della sensibilità corrisponde un numero medio di falsi positivi per mammografia pari a:

$$FPI = 37 \pm 3$$

che rappresenta, in termini percentuali, l'(8.3 ± 0.7)% rispetto al numero medio totale di crop estratti per mammografia:

$$CROP_{tot} = 444 \pm 27$$

Si può notare come incertezza sul numero dei crop FP e su quelli totali sia relativamente elevata. Questo è dovuto fondamentalmente alla variabilità delle dimensioni delle mammelle (Figura 4.1). All'aumentare della dimensione della

mammella aumenterà anche il numero di crop estratti e quindi anche il numero di falsi positivi.

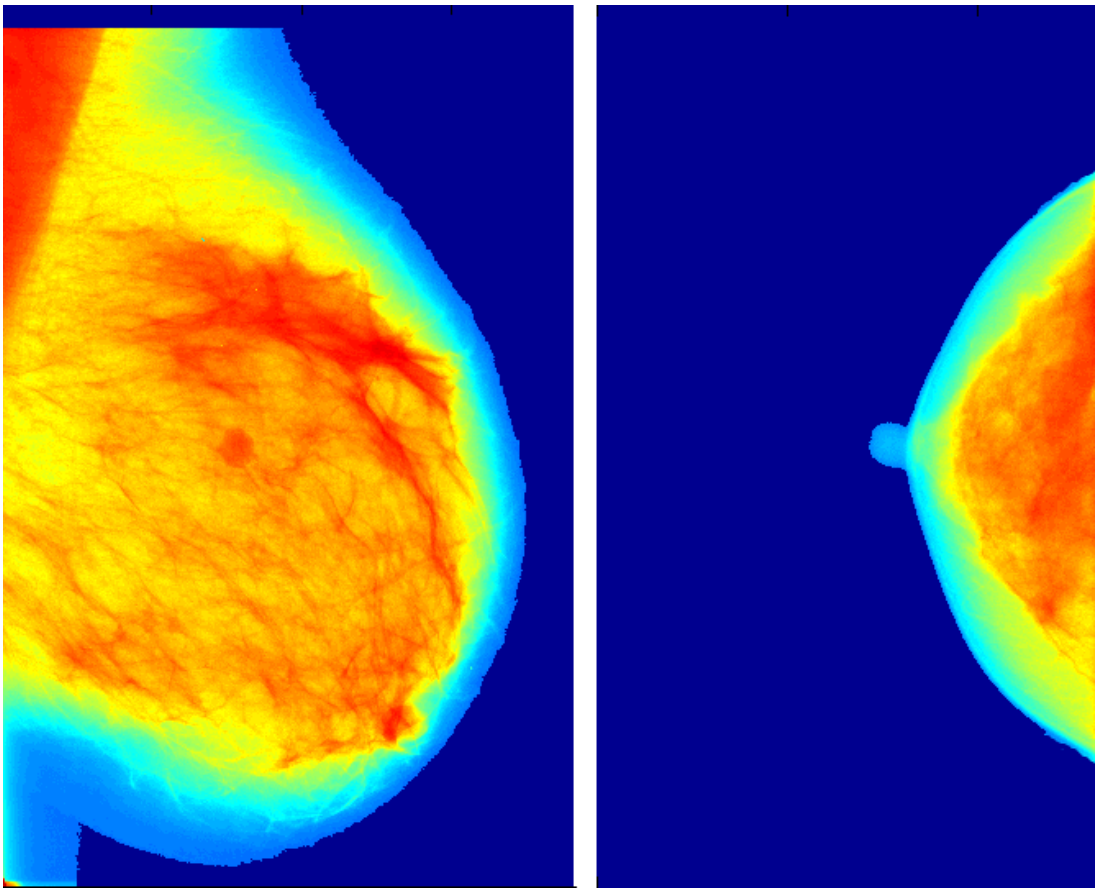


Figura 4.1 Variabilità delle dimensioni delle mammelle

Come detto, per un sistema di detection, soprattutto ai primi livelli, anche la perdita di una sola lesione rappresenta un grave problema. Tuttavia la massa persa (Figura 4.2) ha una caratteristica che, per via del tipo di addestramento a cui è stato sottoposto il sistema, la rende estremamente difficile da rilevare: è situata all'interno della zona del pettorale, una regione che, come detto nel paragrafo 3.5, il nostro sistema è stato forzato ad imparare a classificare negativa (proprio tramite l'aggiunta al train set di esempi negativi provenienti da questa zona). Quasi tutti i sistemi di CAD descritti in letteratura, addirittura, segmentano la zona del pettorale escludendola dall'analisi (per diminuire il tempo computazionale).

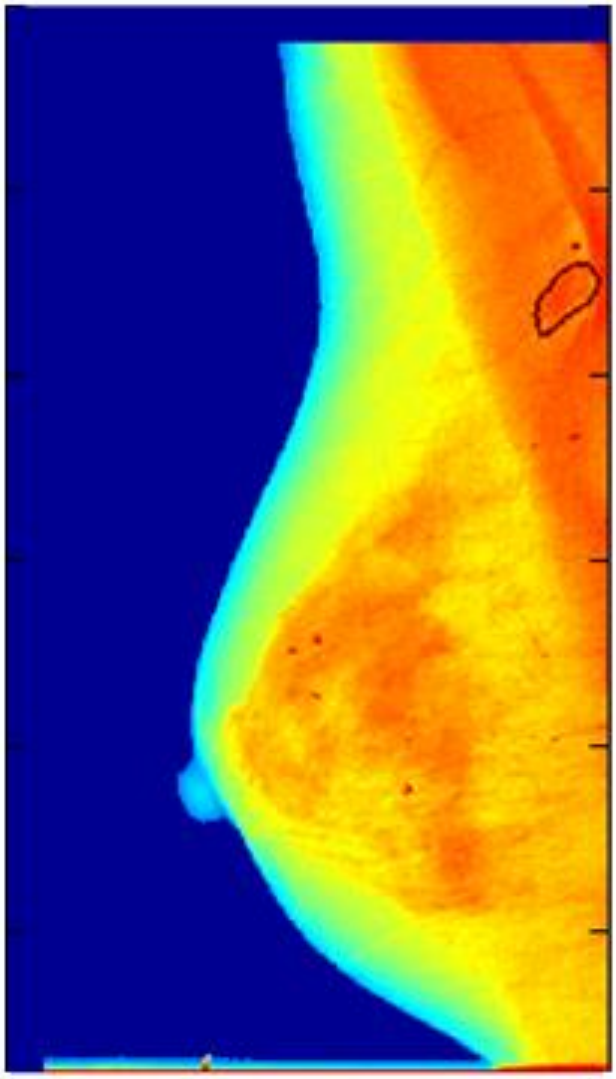


Figura 4.2 Lesione persa dal sistema: è situata nella regione del muscolo pettorale (una zona che il sistema ha imparato a classificare come negativa)

Di seguito vengono presentate le curve ROC e FROC che, come detto in precedenza, permettono di valutare le prestazioni del sistema al variare del punto di lavoro.

In Figura 4.3 viene mostrata la curva ROC. Questa è stata ottenuta tramite una partizione del *dataset* in due parti, *train set* e *test set*, composte rispettivamente dall'80% e dal 20% degli esempi totali. L'area sotto la curva relativa al test set è pari a $A_z = 0.94 \pm 0.01$, dove 0.01 rappresenta l'indeterminazione sulla media. Si può notare come, ad esempio, ad una TPF pari circa all'80% corrisponda una FPF pari a circa al 20%.

L' errore sul valore dell'area è stato calcolato eseguendo sul *train set* una *cross validation* stratificata *10 fold*.

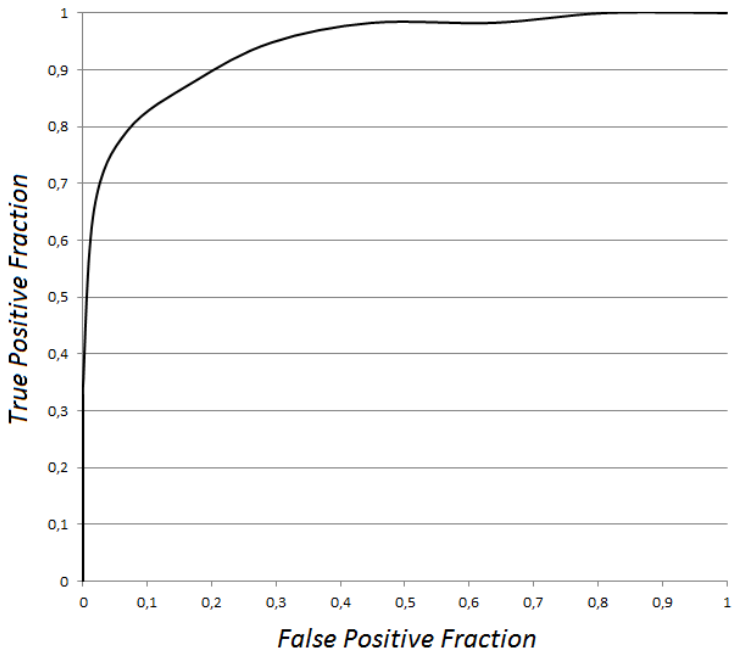


Figura 4.3 Curva ROC ottenuta con la miglior configurazione dei parametri

In Figura 4.4 è graficata invece la curva FROC che, come detto nel Capitolo 1, mostra la frazione di positivi correttamente classificata al variare del numero di falsi positivi per paziente. Osservando la curva si può inferire che, ad esempio, circa il 90% delle lesioni viene classificato ottenendo mediamente circa 15 FP per immagine.

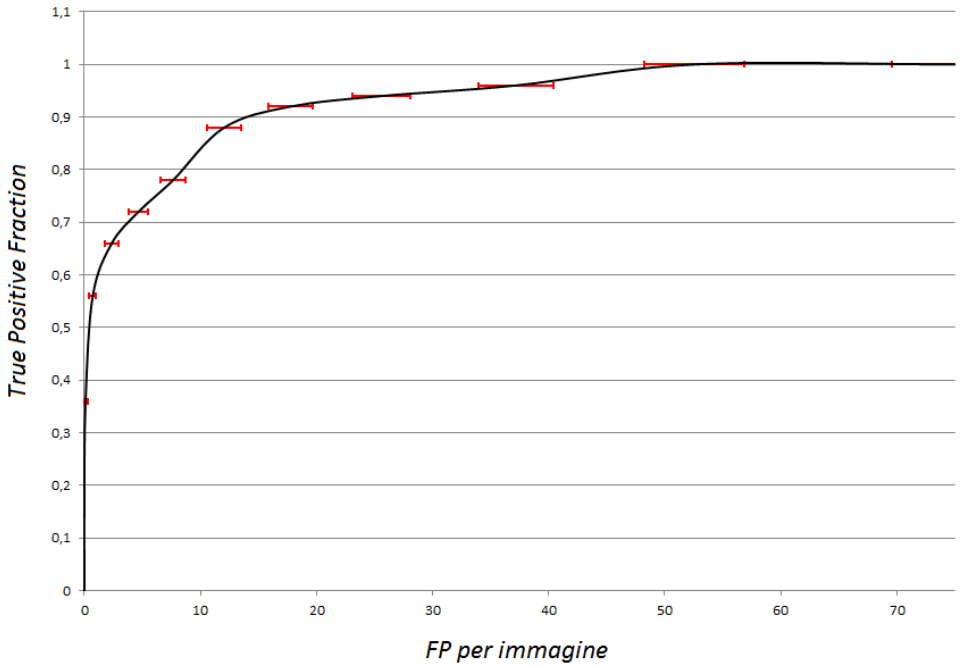


Figura 4.4 Curva FROC (sensibilità vs FP per immagine)

4.2 Confronto con letteratura e conclusioni

Come detto, il problema della detection di masse con le CNN era già stato affrontato da Chan nel 1996 [10]. Nell'articolo in cui espone i risultati, Chan afferma di aver provato due differenti approcci al problema: il primo effettua sui crop solo operazioni di media e subsampling agendo in sostanza solo sui livelli di grigio delle immagini, senza necessità di pre-processing; il secondo comporta l'estrazione di particolari tipi di features (*texture feature*) dalle ROI. Noi, come descritto nei capitoli precedenti, abbiamo intrapreso il percorso featureless, in accordo con l'idea alla base del deep learning di minimizzare il lavoro di pre-elaborazione dei dati. Il confronto tra i nostri risultati e quelli ottenuti da Chan riguardanti l'area sottesa dalla curva ROC (A_z) è mostrato nella tabella 4.2.

	$\boldsymbol{A_z(featureless)}$	$\boldsymbol{A_z(features)}$
Tesi	0.94 ± 0.01	/
Chan	0.83	0.87

Tabella 4.2 Confronto performance con Chan (1996) – Area sotto la curva ROC

In Figura 4.5 è mostrato il confronto tra la nostra curva ROC e quella ottenuta da Chan.

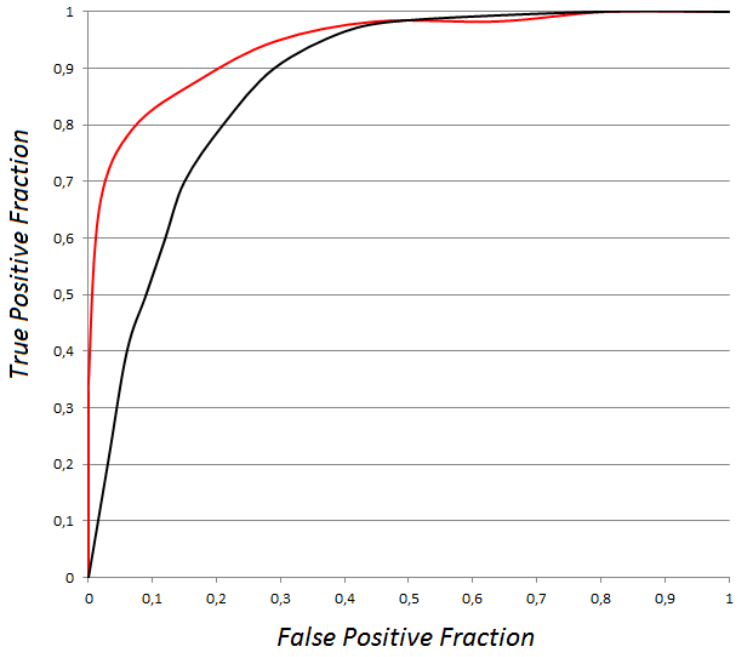


Figura 4.5 Confronto tra curva ROC ottenuta (in rosso) e curva ROC ottenuta da Chan con texture features (in nero)

I migliori risultati ottenuti da Chan sono quelli ottenuti con la tecnica di estrazione delle features, con un'area sotto la curva ROC pari a $A_z = 0.87$. Mediante la tecnica featureless, invece, egli ha ottenuto una $A_z = 0.83$

Come detto, noi abbiamo approfondito la tecnica "featureless" che, pur avendo registrato performance inferiori nel lavoro di Chan, è interessante perché non comporta il pre-processing dei dati. L'area sotto la curva ROC, nel nostro caso, è pari a $A_z = 0.94 \pm 0.01$.

Come detto nei capitoli precedenti, la tecnica usata si differenzia da quella di Chan per l'utilizzo di un'architettura più profonda, formata da due livelli anziché uno, e per l'uso combinato del max-pooling negli strati di subsampling e dell'SVM nello strato finale. Questo approccio ha portato, complessivamente, ad un miglioramento delle performance sia rispetto all'approccio fatureless che a quello con estrazione di features di Chan.

Nell'articolo [18], Mehul P. Sampat, Mia K. Markey e Alan C. Bovik presentano un resoconto delle performance ottenute con diversi algoritmi di detection di masse presenti in letteratura. In Tabella 4.3 riassumiamo i risultati riguardanti la sensibilità e il numero di falsi positivi per immagine (FPI) di diversi sistemi di CAD al primo livello di detection. Alcuni di questi usano dei metodi detti *pixel-based* che, per ogni pixel dell'immagine, ricavano una feature da un suo intorno locale; altri usano metodi detti *region-based*, in cui prima si estraggono le ROI con tecniche di segmentazione o filtraggio e poi da ognuna di queste regioni si ricavano delle features.

Autore	Metodo	**N° immagini**	**TP**	**FPI**
Li et al., 2001	Pixel	200	97.3%	14.8
Petrick et al., 1996	Region	168	95.5%	20.0
Polakowsi et al., 1997	Region	254	92%	8.4

Tabella 4.3 Performance varie di algoritmi di detection al primo livello

In Tabella 4.4 mostriamo le performance del nostro sistema ottenute al variare della true positive fraction (TPF): i dati sono stati ottenuti fissando un valore della TPF e

ricavando, tramite la curva FROC il valore corrispondente di falsi positivi per immagine (FPI). I risultati si riferiscono alle 50 immagini usate come test.

TPF	FP per immagine
88%	12.0 ± 1.5
92%	17.8 ± 1.9
94%	25.6 ± 2.5
98%	37 ± 3

Tabella 4.4 Performance del sistema al variare della sensibilità

4.3 Possibili sviluppi del lavoro

Vista l'ancora alta percentuale di FP per immagine si è pensato, come proseguimento futuro del progetto, di far seguire a questo primo livello del sistema di CAD, un secondo livello con il compito di una *false positive reduction*.

Il secondo livello potrà essere rappresentato da un'altra CNN, da un SVM o dall'algoritmo degli auto-encoders (AE), un'altra tecnica appartenente al campo del deep learning. Questo, per quanto riguarda gli esempi negativi, dovrà essere addestrato mediante i falsi positivi uscenti dal primo livello. L'insieme degli esempi positivi, invece, potrà essere aumentato "artificialmente" tramite un metodo usato spesso in molti tipi di problemi di riconoscimento: si utilizzano trasformazioni spaziali quali rotazioni, riflessioni, traslazioni e operazioni di *scaling* che, a partire da un singolo esempio, permettono di ricavarne altri. Questo aumento del dataset dei positivi potrà essere sfruttato sia per incrementare la robustezza del primo livello che per procurarsi nuovi esempi da usare per l'addestramento del secondo.

Riuscendo ad ottenere un maggiore livello di invarianza rispetto alle traslazioni, in particolare, il sistema imparerà a riconoscerle anche se non perfettamente centrate nella finestra di scansione. Questo potrebbe permettere di ottenere la stessa sensibilità con uno stride maggiore. In questo modo diminuirebbe il numero di crop totali estratti mediamente dalle mammografie e, di conseguenza, il numero di falsi positivi.

L'utilizzo di più livelli in un sistema di CAD è una tecnica comunemente impiegata che permette l'eliminazione progressiva di zone sane sempre più simili a quelle delle masse tumorali. Le zone scartate dai primi livelli sono quelle che presentano caratteristiche palesemente differenti da quelle delle lesioni, mentre quelle che

“sopravvivono” ai vari livelli sono sempre più simili a quelle che il sistema ha imparato a riconoscere come masse. Questo permette di ottenere un valore della *false positive fraction* che non sarebbe possibile con l’utilizzo di un solo livello addestrato su esempi negativi generici.

Bibliografia

[1] J. Roebuck. *Clinical Radiology of the Breast*. Heinemann Medical Books, Oxford, 1990

[2] Rasmus Berg Palm. *Prediction as a candidate for learning deep hierarchical models of data*. Technical University of Denmark, DTU Informatics, 2012

[3] Renato Campanini, Danilo Dongiovanni, Emiro Iampieri, Nico Lanconelli, Matteo Masotti, Giuseppe Palermo, Alessandro Riccardi, and Matteo Roffilli. *A novel featureless approach to mass detection in digital mammograms based on Support Vector Machines*, Physics in Medicine and Biology, vol. *49, no. 6, 2004*

[4] Matteo Roffilli. *Advanced Machine Learning Techniques for Digital Mammography*, PhD thesis, Tech. Rep. UBLCS-2006-12, University of Bologna, Department of Computer Science, 2006

[5] Massimiliano Zanoni. *Algoritmi avanzati per la rivelazione di masse tumorali in mammografia digitale*, 2003

[6] Dario Floreano. *Manuale sulle reti neurali*. Il Mulino, pp. 11-60, 2002

[7] Jawad Nagi, Sameem Abdul Kareem, Farrukh Nagi, Syed Khaleel Ahmed – *Automated Breast Segmentation for ROI Detection Using Digital Mammograms*, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia, 2010

[8] Issam El-Naqa, Yongyi Yang, Miles N. Wernick, Nikolas P. Galatsanos, Robert M. Nishikawa – *A Support Vector Machine Approach for Detection of Microcalcifications*, IEEE Trans. Med. Imaging 21(11): 1552-1563 (2002), 2002

[9] Yann LeCunn, Leon Bottou, Yoshua Bengio, Patrick Haffner – *Gradient-Based Learning Applied to Document Recognition*, Intelligent Signal Processing, 306-351, IEEE Press, 2001

[10] Matteo Masotti – *Optimal Image Representation for Mass Detection in Digital Mammography*, PhD Defense, Bologna, Italia, 2005

[11] Berkman Sahiner, Heang-Ping Chan – *Classification of Mass and Normal Breast Tissue: A Convolution Neural Network Classifier with Spatial Domain and Texture Images*, Dept. of Radiol., Michigan Univ., Ann Arbor, MI. IEEE Transactions on Medical Imaging. 1996

[12] Renato Campanini, Enrico Angelini, Emiro Iampieri, Nico Lanconelli, Matteo Masotti, Matteo Roffilli, Omar Schiaratura, Massimiliano Zanoni – *A fast Algorithm for intra-breast segmentation of digital mammograms for CAD systems,* 2004

[13] Yichuan Tang – *Deep learning using support vector machine,* International Conference on Machine Learning 2013: Challenges in Representation Learning Workshop. Atlanta, Georgia, USA. 2013

[14] D. George. *How the brain might work: A hierarchical and temporal model for learning and recognition,* Dileep, Ph.D., Syanford University, Stanford 2008

[15] R. Bellman. *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957

[16] T. Lee, D. Mumford. *Hierarchical Bayesian inference in the visual cortex*. J. Opt. Soc. Amer., vol. 20, 2003

[17] T. Lee, D. Mumford, R. Romero, V. Lamme. *The role of the primary visual cortex in higher level vision*. Vision Res., vol. 38, 1998

[18] Mehul P. Sampat, Mia K. Markey, Alan C. Bovik. *Computer-Aided Detection and Diagnosis in Mammography*, in Handbook of Image and Video Processing (ed. Bovik), 2nd edition 2005, pgs. 1195-1217 University of Texsas at Austin, 2005

[19] C. Cortes and V. Vapnik. *Support vector networks*. Machine Learning September 1995, Volume 20, Issue 3, pp 273-297

[20] Y. LeCun. *A theoretical framework for back-propagation* in Proceedings of the 1988 Connectionist Models Summer School, (D. Touretzky, G. Hinton, and T. Sejnowski, eds.), (CMU, Pittsburgh, Pa), pp. 21-28, 1988

[21] Gianluca Ferri. *Classificazione di noduli al polmone mediante filtro Support Vector Regression*. 2006

[22] Fukushima, K and Miyake, S. *Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position*. Pattern Recognition, vol. 15(6), pp. 455-469, 1982

[23] Hubel, D. H and Wiesel, T. N. *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*. J. Physiol. Jan 1962

[24] Yang, J, Yu, K, Gong, Y, and Huang, T. *Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification*, in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009

[25] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems. University of Toronto, Canada. 2012.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel. *Handwritten digit recognition with a back-propagation network, in Advances in Neural Information Processing Systems 2*, David Touretzsky, Ed., Denver, CO, 1990, Morgan Kaufmann.

[27] G. L. Martin. *Centered-object integrated segmentation and recognition of overlapping hand-printed characters, Neural Computation*, vol. 5, no. 3, pp. 419-429, 1993.

[28] S. Lawrence, C. Lee Giles, A. C. Tsoi, A. D. Back. *Face recognition: A convolutional neural network approach,* Transactions on Neural Networks, vol. 8, 1997.

[29] R. O. Duda, P. E. Hart. Pattern Classification And Scene Analysis. John Wiley & Sons, New York, NY, 1973

[30] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by error propagation, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, pp. 318-362, MIT Press Cambridge, MA, USA, 1986

[31] D. H. Hubel, T. N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex, Journal of Physiology, London, vol. 160, pp. 106-154, 1962.

[32] J. Roebuck, *Clinical Radiology of the Breast*, Heinemann Medical Books, Oxford, 1990

Fonti internet

[1-I] http://www.arcadialab.com/

[2-I] http://www.airc.it/tumori/tumore-al-seno.asp

[3-I] The Digital Database for Screening Mammography, Michael Heath, Kevin Bowyer, Daniel Kopans, Richard Moore and W. Philip Kegelmeyer, in *Proceedings of the Fifth International Workshop on Digital Mammography*, M.J. Yaffe, ed., 212-218, Medical Physics Publishing, 2001. ISBN 1-930524-00-5

[4-I] Current status of the Digital Database for Screening Mammography, Michael Heath, Kevin Bowyer, Daniel Kopans, W. Philip Kegelmeyer, Richard Moore, Kyong Chang, and S. MunishKumaran, in *Digital Mammography*, 457-460, Kluwer Academic Publishers, 1998; Proceedings of the Fourth International Workshop on Digital Mammography

[5-I] http://deeplearning.net/