POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Biomedica

Tesi di Laurea Magistrale

Classificazione di cisti ovariche in immagini ecografiche tramite reti neurali profonde

Relatore

Prof. Filippo Molinari

Candidato

Nicole Riberti

Correlatore

Ing. Massimo Salvi

A.A 2019/2020

Sommario

Il progetto di tesi si concentra sull'implementazione di una rete neurale convoluzionale adatta alla classificazione di cisti ovariche in immagini ecografiche. L'Intelligenza Artificiale consente, per mezzo di algoritmi automatici, di memorizzare un certo pattern di input e di riconoscerne di nuovi. L'oggetto in questione è la cisti ovarica, cioè una neoformazione che si sviluppa a carico delle ovaie in forme diverse in donne di ogni età. Data la frequenza in ambito diagnostico, diventa fondamentale per il medico riuscire ad identificare la tipologia di cisti durante l'analisi ultrasonografica e decidere la terapia da seguire, alcune cisti devono essere asportate chirurgicamente mentre altre presentano una regressione spontanea. Lo studio dimostra come una rete neurale convoluzionale può apprendere le caratteristiche delle cisti ed essere di supporto al medico per svolgere una corretta diagnosi.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Il Machine Learning o Apprendimento Automatico è un campo di ricerca in continua evoluzione, che ha come scopo quello di insegnare a computer e robot a svolgere attività allo stesso modo degli esseri umani. L'apprendimento avviene in modo spontaneo, direttamente dall'esperienza, proprio come per gli uomini. Le macchine, infatti, non vengono programmate direttamente con equazioni o modelli matematici, ma per mezzo di metodi computazionali al fine di riconoscere i dati in ingresso e distinguerne di nuovi una volta che l'apprendimento è stato completato. Nel seguente progetto di tesi sono stati sviluppati algoritmi di questo tipo che si occupano in particolare della classificazione di cisti ovariche in immagini ecografiche, denominati "Reti neurali convoluzionali". L'idea è partita dall'azienda SynDiag in collaborazione con l'ospedale Mauriziano di Torino, che ha mostrato l'esigenza di un software in grado di sostenere il medico nella diagnosi ecografica. Il linguaggio di programmazione utilizzato è Python, in particolare le librerie open-source di Tensorflow, un insieme di procedure atte allo svolgimento di un determinato compito rese disponibili online.
Nel primo capitolo vengono approfondite conoscenze relative all'ambito di descrizione e discriminazione di tumori e cisti ovariche, necessarie per la loro classificazione. Nel secondo sono elencati i principi fisici dietro alla formazione dell'immagine ultrasonografica e gli artefatti che ne conseguono. Nel terzo si descrive il Machine Learning e poi nello specifico il concetto di Deep Learning, alla base dei modelli computazionali a più livelli di astrazione, per seguire con le tecniche utilizzate in fase sperimentale e lo stato dell'arte delle reti neurali convoluzionali. Esse hanno portato a progressi nell'elaborazione di immagini, video, voce e audio, mentre le reti classiche hanno fatto luce su dati sequenziali come il testo e il parlato. Nel progetto saranno utilizzate le prime, ma si partirà dalla descrizione dei modelli di base per poi passare a quella delle reti neurali profonde. Per finire nell'ultimo capito sono esposti i risultati in seguito alla scelta di organizzazione del dataset di input, del preprocessing delle immagini e delle tecniche di regolarizzazione adottate.

Capitolo 2

Cenni di Medicina

Le ovaie sono organi sede della maturazione di oociti, rilasciati al momento dell'ovulazione, con il compito di secernere gli ormoni sessuali femminili come il progesterone e gli estrogeni. Le unità funzionali dell'ovaio sono i follicoli oofori, i quali maturando nelle fasi di follicolo primario e secondario arrivano alla formazione di involucri tecali (vescicoloso di Graaf), fino a formare il follicolo maturo (19-24 mm di diametro) che rilascerà l'oocito per completare il processo di ovulazione.[1] Talvolta questi organi vengono colpiti da lesioni come tumori solidi e cisti funzionali o benigne. Le neoplasie si suddividono in base alla loro origine cellulare: cellule epiteliali, cellule germinali e cellule dei cordoni sessuali-stromali, mentre le cisti possono essere suddivise in base alla fase del ciclo mestruale in cui si sviluppano, oppure sulla base del loro contenuto.[2]

2.1 Tumori Ovarici

Il tumore ovarico è un tumore che ha origine sulla superficie delle ovaie e rappresenta la causa principale di morte per tumore ginecologico e la quinta per tumore che colpisce le donne nei paesi più sviluppati. Ogni anno vengono diagnosticati in Europa 65.000 casi di cui 5.000 in Italia con un'alta percentuale di mortalità sulle donne tra i 50 e 65 anni. Il tumore ovarico viene definito un "killer silenzioso" dal momento che i sintomi vengono spesso scambiati per disturbi minori. Per questo è fondamentale incentivare test genetici di identificazione del gene BRCA, così da accedere a delle terapie mirate che possono migliorare la prognosi della malattia.
La suddivisione principale dipende dalla gravità del tumore e per questo si distinguono in benigni e maligni. I primi non provocano metastasi, mentre i secondi possono metastatizzare nei diversi distretti corporei e si dividono in carcinomi (90% dei tumori ovarici maligni) e tumori stromali.

Figura 2.1: Utero
[3]

2.1.1 Fattori di rischio

I principali fattori di rischio sono:

- nulliparità;
- prima gravidanza dopo i 35 anni;
- terapia sostitutiva ormonale;
- menopausa tardiva;
- malattia infiammatoria pelvica;
- tumore alla mammella diagnosticata in giovane età;
- endometriosi;
- mutazione BRCA1 e BRCA2, malattie genetiche presenti rispettivamente sul cromosoma 17 e 13, che controllano la proliferazione cellulare;
- familiarità.

Al contrario sono stati stimati alcuni fattori protettivi come l'età inferiore o uguale a 25 anni per la prima gravidanza, un alto numero di gravidanze, l'impiego di contraccettivi orali e l'allattamento al seno. Si tratta infatti di comportamenti di protezione, mentre per quanto riguarda la prevenziove vera e propria, questa è data dallo screening diagnostico, in particolare delle ecografie transvaginali.

2.1.2 Classificazione dei tumori ovarici

I tumori ovarici vengono classificati in base alle cellule da cui derivano:

- **Tumori epiteliali**: corrispondono circa al 90% dei tumori maligni e possono essere sierosi, mucinosi, endometriosi e i tumori a cellule chiare tipo 2.
- **Tumori germinali**: sono circa il 5% dei tumori ovarici e hanno origine dalle cellule da cui si genera l'ovocita. Nell'80% dei casi si manifesta prima dei 30 anni e comprende i teratomi e i disgerminomi (più rari).
- **Tumori stromali o dei cordoni sessuali**: sono rari e derivano da strutture connettivali che producono progesterone ed estrogeno. Le metastasi sono tardive dunque è importante rilevarli in tempo. Tra questi si possono distinguere i tumori della granulosa, granulosa-tesa e i tumori di Sertoli-Leydig.
- **Tumori borderline**: hanno basso grado di malignità e la possibilità di essere asportati per via chirurgica totalmente senza ridurre eccessivamente il tessuto ovarico. La diagnosi in giovane età prevale in questo caso e la tendenza a recidivare è possibile. Nell'Istituto Europeo di Oncologia si è condotto uno studio per cui le recidive crescono 1 mm al mese, permettendo il follow-up delle pazienti per lunghi periodi senza intervenire tempestivamente con l'intervento chirurgico.
- **Tumori peritoneali primari**: derivano dalle cellule sierose che ricoprono l'addome e la pelvi e possono manifestarsi anche in donne sottoposte ad annessiectomia.

2.1.3 Sintomi del tumore ovarico

I sintomi dei tumori ovarici sono personalizzati in base alla paziente che ne soffre e possono essere di diversi tipi, tra questi:

- malessere addominale;
- diarrea o stitichezza;
- crampi;
- aumento della frequenza di minzione;
- improvvise perdite o guadagni di peso;
- sanguinamento vaginale anomalo.

In generale questi sintomi sono molto comuni, e non sempre corrispondono a un tumore ovarico. Quindi è raccomandato valutarne la frequenza, se appaiono per meno di sei mesi e la durata, da più di tre mesi. In questo caso è opportuno contattare il medico per una visita ginecologica con TVS (ecografia transvaginale) e un esame del sangue con controllo del marker tumorale CA-125.[4]
L'esame con CA-125 è molto utilizzato dai medici, sia per i casi in cui lo screening non è sufficiente ai fini diagnostici che nel monitoraggio di neoplasie ovariche in seguito a trattamento chirurgico. L'acronimo sta per *Cancer Antigen 125*, si tratta di una glicoproteina presente nelle tube di Falloppio, nella cervice uterina e nell'utero che aumenta la sua rigenerazione quando uno di questi tessuti è danneggiato, come nel caso di un tumore ovarico. Si rileva grazie alle analisi del sangue e i valori normali vanno da 0 a 35 U/ml. Valori superiori ai 35 U/ml potrebbero indicare la presenza di una patologia maligna in atto. Nonostante ciò tale indice può essere elevato anche in presenza di altri tumori (es. tube di Falloppio, intestino, utero) oppure a causa di cisti benigne. In particolare dunque l'esame del CA-125 è utile in fase post operatoria per verificare l'eventuale ritorno di recidiva, in tal caso i valori non si stabilizzerebbero nel range di normalità e questo indicherebbe che la paziente non reagisce bene alle cure. Un altro marker tumorale è l' HE4 (*Human epididymis protein* che aumenta in presenza di cancro dell'ovaio in modo più specifico, così consente di discriminarlo dalla presenza di una cisti o di altre masse ovariche benigne.[5]

2.2 Cisti non neoplastiche e funzionali

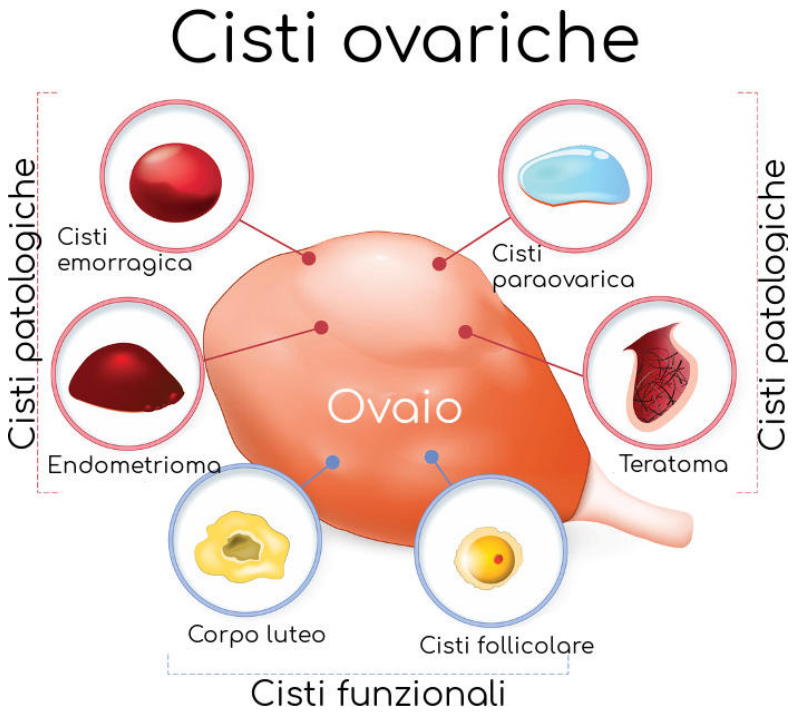


Figura 2.2: Esempi di cisti ovariche
[6]

Le cisti sono neoformazioni che si sviluppano a carico delle ovaie connesse all'utero per mezzo delle tube di Falloppio, la formazione di cisti è un fenomeno molto frequente che non sempre è accompagnato da una forma di malignità. Le ovaie sono ghiandole dedite allo sviluppo di ovociti necessari per l'attività riproduttiva femminile, influenzata da ormoni sessuali che intervengono nella produzione e maturazione dei follicoli. Le cisti possono essere classificate in base alla funzionalità dell'ovaio oppure a seconda della fase del ciclo mestruale in cui si sviluppano, in questo caso vengono definite "cisti funzionali". Altre invece non dipendono da questo e sono chiamate "cisti non neoplastiche" o "tumori benigni dell'ovaio" [2]:

- **Cisti funzionali**: si distinguono in cisti *follicolari* o *luteiniche*, le prime si formano dai follicoli cistici originati dai follicoli di Graaf che non sono andati incontro a deiscenza oppure da follicoli che si sono rotti e richiusi immediatamente nelle fasi follicolari successive. Mentre le seconde si sviluppano dal corpo luteo, una ghiandola endocrina temporanea che si occupa della produzione di ormoni. Essa si sviluppa nella fase luteinica del ciclo mestruale

quando il follicolo ovarico rilascia un ovulo maturo. I corpi lutei dunque sono proprio i resti dei follicoli stessi una volta concluso il processo di ovulazione e si caratterizzano dalla presenza di una corona di tessuto luteale di colore giallo chiaro, fortemente vascolarizzato. In linea di massima vengono naturalmente smaltiti, ma in alcuni casi possono andare incontro a rottura provocando una reazione del tessuto peritoneale. Talvolta la presenza di emorragie e di fibrosi può renderne complicata la distinzione dalle cisti endometriosiche.
Gran parte delle cisti funzionali hanno un diametro inferiore a 2 cm, si risolvono spontaneamente nel giro di pochi giorni o settimane e sono difficilmente riscontrabili in una donna post gravidanza.

- **Cisti non neoplastiche**: possono essere dei *cistoadenomi sierosi o mucinosi*, che si distinguono in base al loro contenuto, appunto siero o muco; *fibromi*, composti di cellule muscolari lisce e tessuto fibroso; *teratomi*, comunemente chiamati anche dermoidi, sono pacchetti cellulari non ben differenziati che possono contenere unghie, capelli e altro tessuto cheratinico; per finire gli *endometrioidi* sono cisti di natura emorragica prevalentemente presenti in premenopausa.

- **Policisti**: ingrossamenti follicolari facilmente identificabili, non portano a ripercussioni gravi ma solo irregolarità del ciclo e sono molto frequenti nelle donne di giovane età. Identificabili come puntini neri che si distinguono nell'omogeneità dell'ovaio.

2.2.1 Classificazione IOTA

IOTA è l'acronimo di *International Ovarian Tumor Analysis* e sta ad indicare il gruppo di medici in Belgio che ha raggruppato, sotto un lessico comune, le caratteristiche delle cisti ovariche al fine di favorire la diagnosi differenziale. La classificazione IOTA permette di distinguere le cisti sulla base della valutazione dei setti (distinzione tra uniloculare e multiloculare), della presenza di parti solide o papille (uniloculare solida o multiloculare solida), dell'ecogenecità che è indice diretto del contenuto della cisti e della vascolarizzazione e che consente di determinare la malignità e lo *score* tramite analisi del segnale Doppler.
Le regole IOTA vengono comunemente definite *Simple Rules* e attribuite a cisti con un diametro pari o superiore a 3 cm.
Una cisti può essere classificata come **benigna** se:

1. è uniloculare;
2. presenta una componente solida di diametro sotto i 7 mm;
3. presenta un'ombra acustica;
4. è multiloculare con regolarità di parete e un diametro massimo di 100 mm;
5. non è vascolarizzata (*color score* = 1).

Una cisti, invece, può essere classificata come **maligna** se:

1. presenta componente solida con parete irregolare;
2. presenta ascite (accumulo patologico di liquido nella cavità peritoneale);
3. almeno 4 strutture papillari;
4. è una multiloculare solida con irregolarità di parete di diametro superiore ai 100 mm;
5. ha una forte vascolarizzazione (*color score* = 4).

Nonostante le regole siano specifiche e facilitino notevolmente la diagnosi del medico, comunque non sono applicabili a tutte le cisti, come a quelle sotto i 3 cm o a quelle superiori a 10 cm. I *small tumors* sono carcinomi invasivi che presentano una difficoltà diagnostica, inoltre cisti come l'endometrioma vengono spesso confuse come cisti borderline (cisti con un indice di malignità e presenza di papille). Spesso i medici per ovviare a questo problema di diagnosi si affidano alla storia clinica della paziente, alla sua età e al marker tumorale CA-125. In caso contrario se l'interpretazione resta incerta è necessario osservare l'evoluzione della cisti.

Nella tabella sottostante vengono caratterizzate le cisti descritte in precedenza andando ad analizzare l'indice di malignità, se devono essere asportate chirurgicamente, il loro contenuto e l'effetto che hanno sull'immagine ecografica.

Cisti	Malignità	Chirurgia	Contenuto	Effetto immagine
Corpo Luteo	No	No, smaltito naturalmente	Non vascolarizzato internamente ma si ingrossa comprimendo i tessuti	Vascolarizzazione esterna
Cistoadenoma	No	Si, provoca dolore	Sieroso o mucinoso	Uniloculare con pareti esterne regolari e spesse
Dermoide	No	Si, tende ad ingrossarsi	Pacchetti cellulari di cheratina (unghie, capelli etc.)	*White ball*, regione iperecogena "a cielo stellato", assenza di vascolarizzazione
Endometrioma	No	Si	Parete intrauterina con zone mucose ecogene	*Ground glass*, contenuto ipoecogeno omogeneo, privi di setti, assenza di vascolarizzazione centrale
Corpo Solido	Si	Si	Irregolarità e presenza di papille, aree cistiche anecogene	Vascolarizzazione interna, struttura disomogenea

Tabella 2.1: Caratteristiche delle cisti

Capitolo 3

Ultrasonografia

3.1 Principi fisici degli ultrasuoni

Il suono è un fenomeno fisico che ha bisogno di una sorgente, il corpo vibrante e del mezzo elastico di propagazione per viaggiare. Il termine "ultrasuono" sta ad indicare la radiazione acustica che viaggia a delle frequenze oltre l'udibile umano, dunque oltre i 20 kHz, generalmente le frequenze utilizzate da un ecografo variano tra 1-2 MHz e 20 MHz. L'onda meccanica provoca un moto armonico di compressione e rarefazione continuo del tessuto elastico in cui si propaga, così che esso risponda con una riflessione, l'"eco di ritorno", verso la sorgente, e l'energia raccolta consenta di costruire l'immagine. Il fatto di trasferire al tessuto solamente energia meccanica rende l'ecografo un dispositivo sicuro dal punto di vista della sicurezza del paziente, gli ultrasuoni infatti sono radiazioni non ionizzanti usate spesso per il monitoraggio di pazienti a rischio, come le donne in gravidanza. Inoltre l'elevata risoluzione temporale (circa 300 immagini al secondo) consente di seguire fenomeni in rapida evoluzione come quelli cardiaci, praticamente in real time. Un altro vantaggio di questa tecnica è la focalizzazione della sorgente, quindi la possibilità di concentrare l'onda nella zona di maggior interesse.
Prima equazione fondamentale: **Legame velocità-frequenza**

$$v = \lambda f \tag{3.1}$$

Dove:

- v = velocità di propagazione dell'ultrasuono nel mezzo
- λ = lunghezza d'onda che determina la risoluzione spaziale minima ottenibile per la generazione dell'immagine
- f = frequenza dell'onda

Dall'inversione della relazione si può dedurre che maggiore sarà la frequenza, minore sarà la risoluzione, quindi migliore la possibilità di distinguere oggetti più piccoli e avere una miglior qualità dell'immagine. Il limite è dato dal fatto che la frequenza è direttamente proporzionale all'attenuazione dei tessuti, quindi tessuti più in profondità non sono visibili con una frequenza elevata. Per questo l'operatore dovrà regolare la frequenza in base al distretto corporeo in indagine e in base alla sua locazione.
Seconda equazione fondamentale: **Impedenza acustica**

$$Z = \rho v \tag{3.2}$$

Dove:

- Z = impedenza acustica dei tessuti espressa in kg/m^2/s o R(rayl)
- ρ = densità del tessuto
- v = velocità di propagazione dell'ultrasuono nel mezzo

Da questa relazione si deduce che l'ultrasuono non è in grado di distinguere la vera densità del tessuto ma la sua impedenza acustica, una caratteristica variabile dal momento che varia la velocità con cui l'ultrasuono di propaga nel mezzo. Per questo è necessario assumere un valore di velocità media dei tessuti (ad eccezione dell'aria, dell'osso e dei polmoni) pari a 1540 m/s, noto come "tessuto molle medio" con ρ 1060 kg/m^3 e Z 1.63 kg/m^2/s. E sarà proprio dalla variazione di Z che si avranno riflessioni diverse in grado di discriminare le diverse porzioni corporee in esame.

3.2 Flussimetria Doppler

I flussimetri sono dei dispositivi che utilizzano comunque gli ultrasuoni ma misurano la velocità del flusso ematico. Questo diventa fondamentale nell'ambito di una diagnosi differenziale, cioè quando si vuole scoprire se la cisti è irrorata da una circolazione sanguigna che ne determinerebbe una forma di malignità. Il principio di base, come si deduce dal nome, è l'"effetto Doppler", secondo cui se una sorgente è in movimento rispetto al ricevitore, quest'ultimo riceve una frequenza diversa in base al moto della sorgente. La frequenza sarà più elevata se il moto è in avvicinamento e più attenuata se in allontanamento. Nella flussimetria la sonda funge sia da sorgente che da ricevitore e il flusso ematico è il bersaglio in movimento, in particolare è la parte corpuscolare del sangue che genera una retro-diffusione andando a modificare la frequenza di emissione.

Questa variazione della frequenza è definita "scarto Doppler" e si ottiene in questo modo:

$$f_{\mathrm{D}} = \frac{2 f_{\mathrm{o}} v \cos(\theta)}{c} \tag{3.3}$$

Dove:

- f_o = frequenza sorgente
- v = velocità incognita del flusso
- θ = angolo formato dall'asse del vaso e dalla direzione di insonazione della sonda*
- c = velocità di propagazione dell'ultrasuono nei tessuti in riferimento al "tessuto molle medio" pari a 1540 m/s

*E' fondamentale che la sonda non venga mai posta a 90° rispetto al vaso, perché in questo modo il coseno sarebbe nullo e si annullerebbe lo scarto Doppler.

3.3 Immagini ecografiche

Le immagini ecografiche utilizzate sono frame in B-mode di video ecografici procurati dall'ospedale Mauriziano di Torino. L'indagine ecografica risulta enormemente complessa per due principali motivi: il primo è che le cisti sono di diversi tipi e spesso a livello visivo hanno caratteristiche in comune che rendono la discriminazione complessa e per secondo, l'ecografia è strettamente operatore dipendente, per cui la sonda micro-convex transvaginale utilizzata può o meno andare a visualizzare ogni lato della cisti sotto esame. Un altro fatto da considerare è l'opinione stessa dei medici che non spesso è concorde sulla diagnosi.
L'ecografia transvaginale è uno degli approcci più utilizzati per la diagnosi di masse benigne o maligne delle ovaie. L'elevata accuratezza e la buona risoluzione permette di descrivere le caratteristiche morfologiche e l'ecogenecità degli organi pelvici in una modalità nettamente superiore rispetto alla sonda transaddominale. Questo in particolare è dovuto alle anse intestinali e ad altri tessuti interposti che ostacolano la visualizzazione immediata dell'apparato uterinico. Inoltre l'ecografia transvaginale permette di riconoscere anche la consistenza della cisti, la sua elasticità e il movimento reciproco tra la massa e i piani anteriori e posteriori.[7] Questo consente all'ecografista di comprendere la natura persistente o transitoria/funzionale della cisti che spesso va incontro a cambiamenti morfologici, per questo si tende a lasciar passare qualche tempo per ripetere l'esame e poi dare il referto definitivo. Gli elementi transitori sono le strutture funzionali come follicoli e corpi lutei, i quali regrediscono spontaneamente dopo circa 60 giorni e dunque non sono soggetti a trattamento chirurgico. Da qui si evince l'importanza diagnostica per evitare o meno l'operazione.[8]

3.3.1 Artefatti in ecografia

Un artefatto ultrasonografico può essere definito come un'informazione falsa, distorta o multipla generata dalla macchina o dall'interazione degli ultrasuoni con i tessuti, che determina quindi una corrispondenza non perfetta tra realtà e modellizzazione fisica sfruttata per produrre le immagini. Esistono quattro tipologie di artefatti:

Caratteristiche del fascio

- **Artefatto in risoluzione assiale e laterale**: avviene quando due o più strutture vengono rappresentate come una sola, nel caso della risoluzione assiale l'artefatto è strettamente legato alla PRF (Pulse Repetition Frequency), più è bassa peggiore sarà la risoluzione, quindi la capacità di distinguere oggetti vicini.

- **Beam width**: è provocato dalla presenza di oggetti fuori dal fascio principale che generano un eco sull'immagine, la zona focale non è ottimizzata pertanto è necessaria una regolazione del fascio tramite lente apposita per eliminare gli echi.

- **Lobi laterali**: il fascio primario non è il solo a interferire con i tessuti, ma esistono fasci secondari generati dai cristalli che hanno direzioni diverse ad intensità inferiore rispetto al primario, questi non vanno a creare effetti visibili in un'immagine ecogena, ma nell'immagine parzialmente anecogena i deboli echi di tali fasci sono visibili come spot chiari attorno al fascio principale. La vescica, la cistifellea e grosse raccolte di liquidi anecogeni sono le principali sedi in cui si riscontra l'artefatto. Per ridurli in genere è sufficiente diminuire il gain generale allo scopo di sopprimere gli echi a bassa energia, così da eliminarli quasi completamente senza ridurre il dettaglio dell'immagine.

Generazione degli echi

- **Riverbero**: la causa delle riverberazioni è la riflessione multipla tra oggetto e trasduttore, generata da strutture che producono una riflessione reciproca del fascio che le colpisce perpendicolarmente, producendo una falsa informazione sulla profondità del tessuto. Il risultato è dato da bande ecogene distanziate tra loro da uno spazio costante pari alla distanza tra oggetto e sonda e con intensità decrescente. La *Coda di cometa* è un esempio di questa tipologia di artefatti; si verifica quando la porzione di tessuto coinvolta è di piccole dimensioni e ad elevata impedenza acustica, come micro-bolle gassose, micro-calcificazioni e cristalli di colesterolo che producono sull'immagine echi paralleli molto vicini tra loro realizzando una forma molto vicina ad una coda di cometa.

In questo caso non ci sono metodiche pratiche per la rimozione dell'artefatto, ma è l'esperienza dell'operatore a guidare l'interpretazione di quanto osservato ai fini della diagnostica. Non vale lo stesso per un altro tipo di riverberazione, l'artefatto *Ring down*, che produce posteriormente a raccolte miste liquido-gassose una striscia iperecogena o una serie di bande parallele trasversali alla direzione di propagazione del fascio. A differenza della *Coda di cometa* questo scompare regolando la sonda e focalizzando il fascio.

- **Effetto Pioggia**: si produce quando il fascio investe aree che generano echi diffusi di medio-bassa intensità poste a monte di una struttura contenente liquido (come la vescica). Il risultato è un effetto puntinato in prossimità della parete che rimanda appunto alla pioggia. Può essere eliminato nel momento in cui ci si allontana da tale interfaccia.

- **Specchio**: le strutture poste presso interfacce ricurve (tessuto/diaframma) possono generare una riflessione riprodotta sia nella loro posizione reale che al di là dell'interfaccia stessa, la quale funge da specchio. Il PC interpreta questi secondi echi come oggetti posti più profondamente e quindi riproduce anche una seconda immagine oltre l'interfaccia a minor luminosità. Anche in questo caso la capacità del medico di distinguere l'oggetto reale dal suo riflesso è fondamentale in ambito di esame.

- **Ombre laterali**: sono legate al fenomeno della rifrazione degli ultrasuoni nel momento in cui essi impattano con i profili laterali di strutture rotondeggianti od ovalari, solide o liquide generando bande laterali ipoecogene in direzione distale. In questo caso l'artefatto risulta utile ai fini di diagnosi per discriminare il tessuto con cui il fascio è entrato in contatto, pertanto non è necessaria la sua eliminazione.

Velocità del suono

- **Distorsione geometrica**: provocata dalla variazione di velocità degli ultrasuoni nei diversi tessuti, reputata costante dagli ecografi che ricostruiscono l'immagine sotto questa ipotesi. In tessuti come acqua e grasso, la distanza calcolata dall'ecografo sarà più piccola poiché la velocità di propagazione è inferiore rispetto a quella di settaggio, e l'immagine che ne deriva risulta rappresentata più in profondità, mentre nei tessuti in cui la velocità di propagazione è superiore la distanza stimata sarà maggiore di quella reale e le formazioni appariranno più in prossimità. È necessario che sia il medico a tener conto di questa differenza.

Attenuazione dei tessuti

- **Shadowing** (Cono d'ombra posteriore): attenuazione o completa riflessione del fascio da parte di tessuti come calcificazioni, collagene, masse solide o maligne che determinano l'eliminazione degli echi di ritorno nella zona sottostante che è quindi "muta". Questo spot di ridotta intensità deve essere correttamente interpretato dall'ecografista perché potrebbe essere un messaggio chiave di presenza di cisti o masse tumorali.

- **Rinforzo di parete posteriore**: il fascio attraversando una raccolta liquida omogenea non produce echi e per questo si attenua poco generando ultrasuoni più intensi a valle di essa rispetto a quelli che non l'hanno attraversata. Dunque, i tessuti sottostanti appariranno più ecogeni di quanto sono realmente. Talvolta questo artefatto può essere utile nella distinzione di una lesione cistica da un nodulo solido ipoecogeno.

Capitolo 4

Machine Learning

4.1 Reti Neurali

4.1.1 Panoramica Machine Learning

Con il termine "Machine Learning" si definisce quella branca dell'informatica dedita allo studio dell'Intelligenza Artificiale (AI), in cui si utilizzano algoritmi automatici di apprendimento computazionale di un certo pattern di input. L'idea di fondo è quella di addestrare l'algoritmo partendo da dati disponibili, i cosiddetti "training data", e in seguito far si che esso acquisisca la capacità di predire informazioni nuove, "test data", sulla base di quelle apprese. Una volta impostato il modello di base, il training avviene in automatico sulla base di osservazioni. Inoltre, i tempi computazionali sono abbastanza brevi (qualche ora) e possono essere applicati in tantissimi campi: motori di ricerca, riconoscimento ottico di caratteri, computer vision, riconoscimento di immagini come volti o come nel nostro caso di immagini mediche.[9] Ogni tipo di algoritmo deve avere un obiettivo. Il criterio utilizzato si basa sulla minimizzazione di una funzione matematica, definita in generale *loss function*. Le funzioni di perdita sono scelte sulla base della tipologia di addestramento, in particolare le due classi principali sono le funzioni utilizzate nella classificazione e quelle utilizzate nella regressione. Nei problemi di regressione i dati in input vengono interpolati per fornire in output un valore numerico continuo, il dominio stesso dell'output, infatti, risulta continuo e non determinato da un insieme discreto di possibilità. Mentre nei problemi di classificazione il compito è quello di assegnare ad ogni input un'etichetta che corrisponde alla classe di appartenenza. Può essere sia binaria (0 o 1) oppure nel caso in cui vi siano più di due etichette, si fa riferimento alla classificazione multi-classe. Il metodo più comunemente utilizzato per trovare il minimo della funzione è il *Gradient Descendent*, descritto in seguito. La scelta della funzione di perdita dipende da una serie di fattori, tra cui la presenza di valori anomali (*outliers*), la scelta dell'algoritmo di apprendimento della

macchina, l'efficienza temporale della discesa del gradiente, la facilità di trovare le derivate e la fiducia nelle previsioni. Per questo vengono mostrate in seguito alcune di esse:

- *Square Error Loss*:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i*)^2 \tag{4.1}$$

  Funzione prevalentemente utilizzata nel Machine Learning, il vantaggio è legato alla convergenza sicura dell'algoritmo anche con un tasso di apprendimento fisso. Il termine n rappresenta il numero di osservazioni, mentre y e y^* sono rispettivamente il valore predetto e il valore desiderabile.

- *Absolute Error Loss*:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i*| \tag{4.2}$$

  L'utilizzo del modulo rende l'algoritmo più robusto agli *outliers*, questo perché se si pensa di minimizzare l'MSE la previsione sarà la media di tutti i target considerati, mentre quando si va a minimizzare l'MAE quello che si ottiene è la mediana delle osservazioni che, per la statistica, è una stima più robusta rispetto alla media. Lo svantaggio principale è che il suo gradiente sarà grande anche per piccoli valori di perdita. Ciò può essere risolto con un tasso di apprendimento dinamico che diminuisce ad ogni iterazione, così che l'algoritmo possa arrivare a convergere.

Generalmente per i classificatori si utilizzano altre tipologie di funzioni di perdita, dal momento che l'obiettivo è quello di separare gli input in un numero prestabilito di classi di output, ottenendo come risultato un valore discreto:

- *Binary cross entropy*:

$$H(p,q) = -(p(x)log(q(x)) + (1-p(x))log(1-q(x))) \tag{4.3}$$

  Funzione derivante dalla teoria dell'informazione che esprime la distanza tra le probabilità dell'input di appartenere alla classe 0 o alla classe 1. Il termine p rappresenta il "target" mentre q la "prediction" per ciascuna classe.

- *Categorical cross entropy*:

$$H(p,q) = -\sum_{i=1}^{n} p_i log(q_i) \tag{4.4}$$

  Funzione analoga alla precedente su cui calcola la probabilità dell'input di appartenere a una classe tra le n stabilite. In questo caso il termine p rappresenta il "target" e q la "prediction" degli n valori possibili dell'input.[10]

Esistono due tipologie di apprendimento per addestrare una macchina:

- **Supervisionato**: il set di dati di training è noto e contiene sia gli input che le *labels*, cioè le etichette associate per l'identificazione della classe di appartenenza. Gli esempi classici di applicazione sono la classificazione e la regressione.
- **Non supervisionato**: in questo caso il dataset contiene esclusivamente gli input senza le etichette, di conseguenza l'algoritmo non può fare associazioni ma tenderà invece a raggruppare i dati in base a criteri di somiglianza, come la distanza euclidea o quella probabilistica. Il tipico esempio è il clustering.

La classificazione delle cisti verrà fatta per mezzo di un modello computazionale di classificazione supervisionato basato su Reti neurali artificiali (abbreviate in ANN o NN). Le reti neurali sono algoritmi di apprendimento ispirati al funzionamento delle reti neurali biologiche, sia dal punto di vista strutturale che dal punto di vista funzionale. Si tratta di un grafo a più strati, definiti appunto *layers*, formato da archi e nodi. I primi rappresentano delle sinapsi artificiali, i pesi della rete aggiornati ad ogni iterazione, la sua vera fonte di apprendimento. Invece i nodi rappresentano i neuroni ad ogni strato, caratterizzati dalla loro funzione di attivazione. Tale modello costituisce un gruppo di interconnessioni tra le quali passa l'informazione dall'input all'output senza tornare indietro. Un'altra caratteristica è legata alla profondità, più la rete è "profonda", più la complessità della struttura aumenta. Aumentare i tempi computazionali andando a costruire più *layers* diventa necessario per avere una maggior accuratezza dei risultati finali. Lo svantaggio principale è che l'apprendimento è una "scatola nera", cioè non conosciamo il modo in cui la rete riesce ad apprendere e quindi interpretare i singoli aggiornamenti dei pesi è praticamente impossibile. Ciascun neurone rappresentato ha una propria capacità di elaborazione e trasforma l'input in output sulla base della funzione di attivazione scelta.

$$output = f(\sum pesi * input) + bias \tag{4.5}$$

Le funzioni di attivazione possono essere di diverso tipo, le più comuni sono:

- *Threshold function*: funzione soglia che restituisce 1 se la sommatoria degli input è maggiore o uguale di zero mentre restituisce 0 negli altri casi.
- *Sigmund function*: funzione con codominio continuo tra 0 e 1, ma l'output non è più 1 ma la probabilità che il valore di uscita sia uguale a 1. Lo stesso vale per lo zero.
- *Rectifier function*: funzione che restituisce 0 quando la sommatoria degli input è minore o uguale di 0, mentre la sommatoria dei pesi per i rispettivi input in

tutti gli altri casi. Con un codominio da 0 a infinito. Definita comunemente ReLU (Rectified Linear Unit) è applicata nelle reti convoluzionali al fine di accelerare la discesa del gradiente, sfruttando una funzione poco costosa dal punto di vista computazionale.

- *Hyperbolic Tangent function*: simile alla Sigmund ma con un codominio da -1 a 1, una caratteristica utile in determinate applicazioni.
- *Softmax*: applicata nelle classificazioni multi-classe, ha come scopo quello di mettere in luce il valore massimo e nascondere quelli più piccoli, a cui sarà attribuito un peso minore.

4.1.2 Perceptron

Inizialmente sono state fatte delle prove con il Perceptron, per aver una maggior praticità con il linguaggio di programmazione Python e per comprendere il funzionamento alla base di questa tipologia di algoritmi. Infatti il percettrone è il modello più semplice, caratterizzato da una serie di input, ciascuno moltiplicato per il valore del suo peso corrispondente, che una volta sommati insieme andranno in pasto alla funzione segno. Il dataset utilizzato nelle prove è il MNIST, una base di dati di cifre scritte a mano (60000 per il training e 10000 per il test), impiegato per l'addestramento dei sistemi di elaborazione di immagini che in output dovranno restituire la cifra corrispondente. Come prima cosa si fissa il *seed* dal momento che la rete contiene componenti random tra cui l'inizializzazione dei pesi. Il seme con cui viene inizializzato il generatore di numeri random produce le istanze di questi processi. In questo modo ogni volta che il codice verrà lanciato si avrà la stessa distribuzione di valori di input e sarà possibile paragonare le diverse prove. La tecnica di ottimizzazione della funzione costo è il *Gradient Descendent*, un algoritmo basato sulla ricerca locale. Il fine ultimo, infatti, è quello di convergere in un punto che sia il minimo della funzione obiettivo a cui è applicato il gradiente. Un limite è la dipendenza dalla condizione iniziale, cioè dal primo punto della funzione obiettivo in cui si calcola il gradiente, perché è proprio da esso che ne dipenderà la direzione di spostamento per raggiungere il minimo. Considerando la funzione costo come una curva, essa sarà fatta da una serie di minimi locali ed un minimo globale, che è quello ricercato. Il limite di questo algoritmo è che spesso non trova quello globale ma quello locale perché, accettando sempre la soluzione migliore, potrebbe cadere in uno di essi e credere che sia quello assoluto. Un altro limite è la dipendenza dalle dimensioni del "vicinato", cioè l'intervallo di osservazione attorno al punto considerato ad ogni iterazione. Questo influenza la dimensione dello spostamento, lo step con cui ci si avvicina al minimo, definito *learning rate*. Si tratta della costante di proporzionalità utilizzata anche se, in molte applicazioni pratiche, viene fatta variare man a mano che l'algoritmo di avvicina alla convergenza. Quando si trova lontano dal minimo ha un valore maggiore, per ridurre i tempi e dunque avvicinarsi più velocemente, mentre una volta in prossimità di esso il suo valore si riduce per evitare delle divergenze indesiderate.[10]

Esistono tre varianti del *Gradient Descendent*:

- *Batch gradient descent*: il gradiente viene calcolato sull'intero dataset eseguendo una sommatoria di tutte le osservazioni, per produrre un unico aggiornamento dei parametri nella direzione opposta a quella del gradiente. Lo svantaggio è legato alla lentezza dell'algoritmo e ai costi computazionali per iterare sul dataset totale ad ogni epoca. Il vantaggio è che garantisce convergenza al minimo globale per superfici convesse e al minimo locale per quelle non convesse.
  Nell'equazione seguente θ corrisponde ai parametri da aggiornare ad ogni iterazione, η al tasso di apprendimento e ΔJ al gradiente della funzione costo.

$$\theta = \theta - \eta \Delta J(\theta) \tag{4.6}$$

- *Stochastic gradient descent*: l'aggiornamento dei parametri della loss avviene ad ogni esempio di training x, questo favorisce una maggior velocità rispetto al *Batch* ma anche fluttuazioni maggiori della funzione obiettivo che danno la possibilità all'algoritmo di saltare per finire in un minimo locale migliore. Nonostante ciò, questo può complicare la convergenza definitiva e rendere le rappresentazioni più rumorose. Anch'esso garantisce la convergenza al minimo globale per superfici convesse e al minimo locale per quelle non convesse.

$$\theta = \theta - \eta \Delta J(\theta; x(i); y(i)) \tag{4.7}$$

- *Mini-batch gradient descent*: è una via di mezzo dei due precedenti ed è quello che andremo ad applicare nelle simulazioni. Divide il dataset in un numero n di batch (generalmente potenze di 2: 32, 64, 128, 256, 512, etc.) e per ognuno aggiorna i pesi, così riduce la varianza del parametro aggiornato e garantisce una convergenza più stabile. E' necessario inoltre rimescolare il dataset dopo averlo suddiviso in batch, prima di iniziare l'allenamento.[11]

$$\theta = \theta - \eta \Delta J(\theta; x(i : i + n); y(i : i + n)) \tag{4.8}$$

Una volta definito l'errore, questo si "propaga indietro" al fine di aggiornare i pesi e quindi favorire l'apprendimento. Il processo è iterativo fino a minimizzare la loss e ad arrivare ad un alto livello di accuratezza. Questo algoritmo appena descritto è il "Back propagation". Vi sono due fasi in tale tecnica: la prima è la *Forward phase* in cui pesi e connessioni sinaptiche restano inalterate e il segnale in input viene "propagato in avanti" tra i layers ad ogni livello di profondità, fino ad arrivare all'output. All'uscita si valuta la differenza tra l'output ottenuto e quello desiderato, questa differenza, rilevata dalla funzione costo, stabilisce di quanto i pesi devono essere modificati all'iterazione successiva. Il parametro che quantifica

ciò è l'errore, che nella seconda fase *Backward phase*, viene "propagato indietro". L'algoritmo si basa sulla derivazione a catena e quindi sul calcolo del gradiente della loss function per ogni parametro della rete.

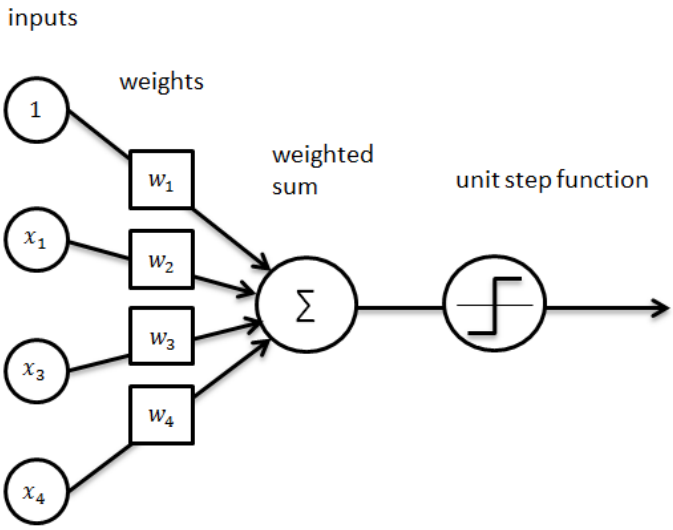


Figura 4.1: Perceptron
[12]

4.1.3 Multi-Layer Perceptron

Come nel percettrone anche nel Multi-Layer Perceptron gli obiettivi rimangono gli stessi: minimizzare la funzione costo e incrementare l'accuratezza di previsione della classificazione. La differenza sta nel fatto che al modello viene inserito uno strato di neuroni intermedi "hidden layers", i quali favoriscono una maggior profondità della rete e una classificazione più accurata. Nel Perceptron l'output della classificazione avviene per mezzo della Binary Cross Entropy, dal momento che il risultato deve essere 0 o 1. In questo caso però, in cui abbiamo una classificazione multi-classe con una serie di neuroni di output, si fa riferimento alla Categorical Cross Entropy, in cui le etichette sono vettori con 1 nella classe di appartenenza corrispondente e dalle altre parti 0.

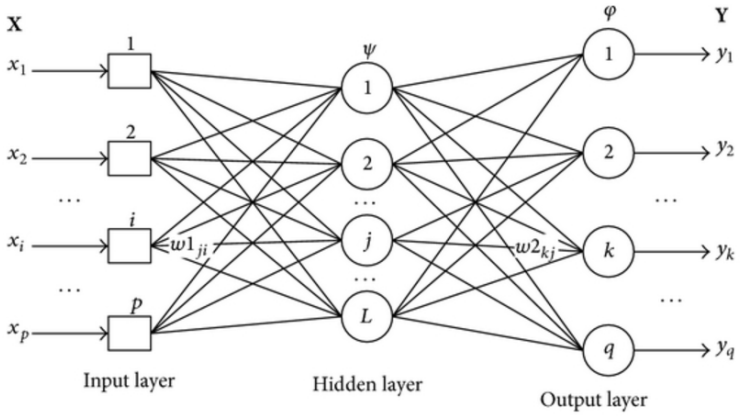


Figura 4.2: Multi-layer Perceptron
[13]

Un problema che è spesso affrontato quando si tratta di reti neurali multi-layer è l'*overfitting*, letteralmente un "adattamento eccessivo" che si verifica quando la rete riesce a riconoscere bene unicamente gli esempi del training e non quelli del test. In alcuni casi è necessario interrompere l'apprendimento e quindi riconoscere il numero di epoche ottimale per cui si ha una riduzione sufficiente della *loss function* tale che la rete mantenga la capacità di generalizzare. Nonostante ciò, definire un numero di epoche ottimale non è facile, per questo esistono delle "tecniche di regolarizzazione" che hanno il compito di introdurre del "rumore" in ingresso alla rete nella fase di training e far in modo che gli esempi non vengano imparati nel modo ottimale, cosa che può avvenire man a mano che si incrementa la profondità della rete. Alcune tecniche sono le seguenti:

- **Dropout**: consiste nell'eliminazione di alcuni neuroni nella fase di addestramento secondo un certo rate (probabilità di "drop"). Dunque la rete sarà costretta ad ogni step a fare previsioni in assenza di alcuni neuroni, questo consentirà alla rete di essere meno sensibile a specifici pesi, migliorando la capacità di generalizzare. Ogni neurone sarà costretto ad apprendere unicamente le features più robuste per poi classificarle nel modo corretto in fase di test.
- **Batch-Normalization** (Blocco di normalizzazione aggiunto): permette di migliorare la velocità, la performance e la stabilità della rete per mezzo di due tecniche : "Re-center" e "Re-scaling" applicate a gruppi di input. Questo consente di risolvere il problema dell'*Internal covariate shift*, cioè dello spostamento della distribuzione degli input durante l'apprendimento. Tale problema causa il rallentamento del training e dunque richiederebbe dei *learining rate* più bassi affinché la rete riesca comunque ad apprendere.[14] La Batch-Normalization va a normalizzare i *mini-batch* sottraendo la loro media e dividendo per la deviazione standard di ciascun gruppo di dati. Considerando l'insieme *B=Xi...m* come *mini-batch*:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.9}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{4.10}$$

$$x_i* = \frac{x_i - \mu_B}{\sqrt{\sigma_B + \epsilon}} \tag{4.11}$$

- **L1**(sparsity) e **L2**(weight decay): le tecniche di regolarizzazione L1 e L2 vanno ad applicare una norma al vettore dei pesi la prima va a sommare i moduli degli elementi mentre la seconda fa la radice quadrata della somma degli elementi al quadrato. Questo viene fatto per rendere i coefficienti del modello piccoli andando a ridurne la complessità. Il parametro in ingresso (attorno a 0.001/0.005) corrisponde al tasso di regolarizzazione cioè a quel compromesso tra trovare pesi sufficientemente piccoli e minimizzare la funzione costo. E' necessario tener presente che la funzione di regolarizzazione aggiunta deve essere della stessa scala della funzione costo, in caso contrario si avrebbero problemi di apprendimento.[15]

$$x_{L1} = \sum_{i=1}^{m} |x_i| \tag{4.12}$$

$$x_{L2} = \sqrt{\sum_{i=1}^{m} x_i^2} \tag{4.13}$$

4.1.4 Reti neurali profonde

Il Deep Learning permette ai modelli computazionali, composti da più livelli, di apprendere in modo autonomo le rappresentazioni di dati con molteplici livelli di astrazione. Questi metodi hanno migliorato notevolmente lo stato dell'arte del riconoscimento vocale, del riconoscimento visivo degli oggetti, del rilevamento di oggetti e molti altri ambiti come la scoperta di droghe e la genomica. Le reti neurali profonde si occupano di sottoporre i dati ad un'elaborazione maggiore, su più livelli, creando così una rete profonda che possa estrarre delle features dagli input. Le features sono le caratteristiche del dataset estratte dall'algoritmo in modo automatico. Questo è reso possibile grazie all'aggiunta di numerosi hidden layers di neuroni.

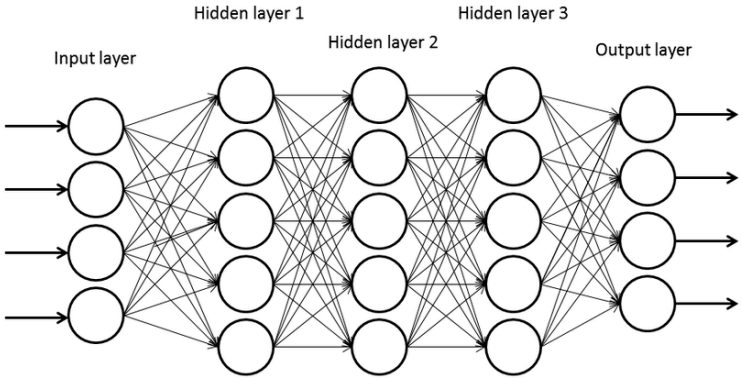


Figura 4.3: Deep Neural Network
[16]

Il principio di funzionamento cardine è analogo a quello delle reti neurali classiche in cui si utilizza il metodo del gradiente e il back propagation per l'aggiornamento dei pesi, con la differenza del fissaggio degli iperparametri, come la dimensione dei dati in input, la dimensione in termini di numero di strati degli hidden layers, il learning rate, l'inizializzazione dei pesi iniziali e l'operazione di ottimizzazione. Questo ci fa comprendere come, per ottenere un'accuratezza migliore nei risultati, sia necessario far variare questi parametri e controllare il problema del *Vanishing gradient* derivante dalla struttura profonda della rete. Si definisce anche "degradazione del gradiente", dal momento che le funzioni di attivazione producono dei valori molto bassi in output (-1;1) che, moltiplicati tra loro a causa della computazione in catena, vengono propagati indietro senza produrre un aggiornamento consistente dei pesi. Se i pesi non vengono aggiornati, dunque, la rete non riesce ad apprendere. Questo problema può essere risolto con l'utilizzo delle reti residuali, stato dell'arte delle reti neurali profonde.

4.1.5 Reti neurali convoluzionali

La rete neurale convoluzionale è una tipologia di rete profonda adatta in particolare alle immagini. Le CNN (Convolutional Neural Net) si specializzano nella classificazione e nella segmentazione di oggetti in determinate immagini, andando ad estrarvi le features particolari che poi andranno ad essere classificate. Essendo comunque una sottoclasse delle reti neurali profonde, anche queste saranno organizzate gerarchicamente con almeno due hidden layers. Il vantaggio di tale impostazione è la condivisione e il riutilizzo delle informazione su più strati, andando a selezionare features particolari e scartando i dettagli inutili. Tale idea si ispira ai neuroni del nostro sistema visivo, i quali compongono una rete cellulare complessa. Le cellule possono essere di due tipologie: le *simple cells*, sensibili a piccole sottoregioni del campo visivo (campo ricettivo), agiscono come feature detector comportandosi come filtri locali sullo spazio di ingresso (mappa dell'immagine di input). Essi vanno a sfruttare la correlazione spaziale presente nelle immagini naturali; mentre le *complex cells* fondono gli output delle *simple cells* in un intorno, compiendo l'operazione di *pooling*.
Anche il numero stesso di livelli utilizzati (una decina) è ispirato a quelli tra la retina e i muscoli attuatori, i quali se fossero in un numero maggiore sarebbero troppo lenti per rispondere agli stimoli. Essendo inoltre la corteccia visiva animale il sistema di elaborazione visiva più potente esistente, sembra naturale emulare il suo comportamento.[17]

Caratteristiche principali

Tornando alla descrizione delle CNN il principio di base, come è espresso dal nome, è la convoluzione dell'immagine con un kernel fisso che, passato sopra di essa, fa prodotti elemento per elemento e ad ogni spostamento, li somma ottenendo in uscita una *feature map*. Altri parametri da controllare saranno dunque le dimensioni e il numero dei kernel, definiti nell'architettura della rete. Per rilevare aspetti particolari dell'immagine, come bordi ad esempio, sarà necessario utilizzare kernel di dimensioni molto più piccole rispetto all'immagine, così è possibile analizzare milioni di pixels rilevando soltanto alcune caratteristiche significative e questo riduce l'utilizzo di memoria. L'operazione di convoluzione per input continui è la seguente:

$$s(t) = \int_{-\infty}^{+\infty} x(a)w(t-a)da \tag{4.14}$$

Dal momento che le immagini utilizzate sono, a livello pratico, delle mappe di interi si può definire la convoluzione discreta:

$$s(t) = \sum_{-\infty}^{+\infty} x(a)w(t-a) \tag{4.15}$$

Nel caso dei layer di convoluzione il filtro scorre su delle piccole porzioni della mappa dell'immagine, mentre i layers di max pooling sono utilizzati per ridurre le dimensioni delle feature maps. In quest'ultimo caso il filtro, con dimensioni note, mantiene solo il massimo di ogni finestra su cui è applicato, quindi tanto più grande sarà la dimensione di esso, tanto più la feature map in uscita sarà piccola.
Il motivo principale dell'utilizzo di tali operazioni nella rete è quello di aumentare il suo campo ricettivo, permettendo ad ogni neurone in uscita di vedere una porzione più grande dell'ingresso. Inoltre, il *downsampling* consente di utilizzare un numero maggiore di feature maps senza aumentare di troppo il peso computazionale dell'algoritmo, un limite importante per immagini 3D.[18]
Un'altra idea di base delle reti convoluzionali è il *Weights sharing*, cioè la condivisione dei pesi, che crea una dipendenza tra i livelli. Infatti un peso non è legato al singolo layer ma può essere riutilizzato altrove così da evitare il calcolo di kernel nuovi che aumenterebbero i tempi computazionali del sistema. Oltre a garantire l'efficienza, questa tecnica consente alla convoluzione di acquisire la proprietà di equivarianza alla traslazione, cioè creare una dipendenza tra input e output tale per cui la convoluzione di un input traslato è uguale alla traslazione della convoluzione dello stesso input:

$$f(g(x)) = g(f(x)) \tag{4.16}$$

La riduzione del numero dei pesi e delle connessioni, inoltre, riduce notevolmente i costi computazionali dell'algoritmo, poiché neuroni diversi dello stesso hidden layer eseguono lo stesso tipo di elaborazione (stessi pesi) su porzioni diverse dell'input.

Architettura della rete

I layers comunemente utilizzati per una rete convoluzionale sono:

- **Convolutional layer**: il layer di convoluzione è utilizzato per l'omonima operazione e seguito dalla funzione di attivazione ReLU al fine di introdurre la non linearità ed evitare la regressione lineare.

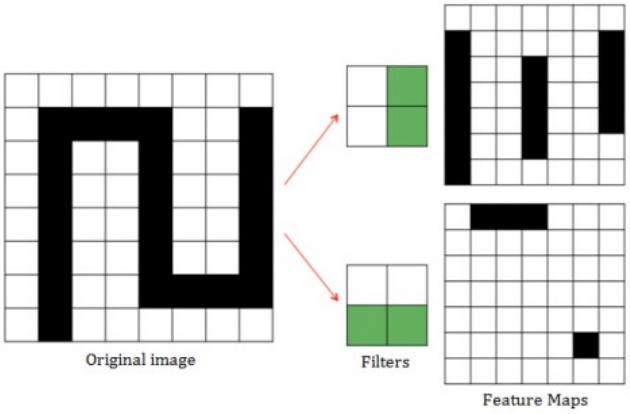


Figura 4.4: Feature maps generate da filtri per linee orizzontali e verticali [19]

- **Max Pooling** e **Average Pooling**: questi layers hanno lo scopo di eliminare il superfluo riducendo letteralmente le dimensioni delle feature maps. Esso viene applicato dopo gli strati convoluzionali per ridurne la complessità, infatti questo "raggruppamento" crea una *condensed feature map*, cioè una feature map più piccola tramite applicazione di un filtro alle sotto-regioni. Il Max Pooling prende il valore massimo tra quelli della sotto-regione considerata, mentre l'Average Pooling prende il valore medio.

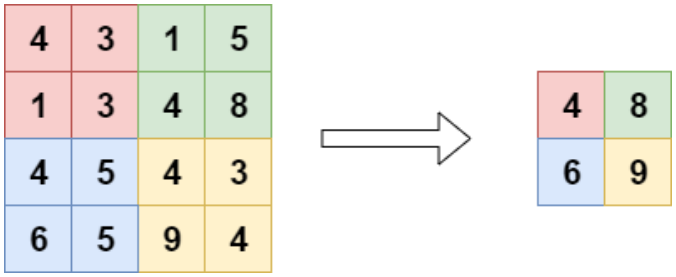


Figura 4.5: Max Pooling [19]

- **Dense layer**: i layers Dense sono quelli utilizzati per la classificazione vera e propria, qui ogni neurone è connesso ad ogni neurone nel livello successivo, per questo la struttura viene definita "fully connected". Questa porzione dell'architettura ha caratteristiche analoghe alle reti neurali classiche, utilizza infatti bias e pesi per classificare gli input, obiettivo finale della rete.

Data Augmentation

L'operazione di "Aumento del dataset" viene inserita nel momento in cui il supporto di dati in ingresso risulta insufficiente per allenare una rete, oppure non è sufficientemente vario. Quello che si vuole ottenere è un dataset più robusto e ampio evitando le ridondanze che potrebbero portare all'overfitting. L'idea di base è quella di andare a "deformare" i dati tramite tecniche di tipo geometrico come: *flipping*, *rotation*, *scaling*, *zooming* e *cropping* oppure tecniche che vanno ad occuparsi dell'Image Enhancement come *gamma correction* e *histogram equalization*. Così da avere in uscita sia le immagini originali che le stesse modificate. La rete neurale non si accorgerà di questa differenza e vedrà i due gruppi come unici andando a fare previsioni su entrambi.
Prima di aggiungere la funzione che applica queste operazioni è necessario capire bene a quale punto dell'algoritmo posizionarla e soprattutto per quali immagini, dal momento che alcune operazioni potrebbero portare a una mancata identificazione delle cisti. Pertanto è necessario comprendere quale tecnica adottare. Le opzioni sono due: applicare la Data augmentation prima di inserire i dati nel modello, cioè eseguire le modifiche in anticipo aumentando la dimensione del set di dati iniziale; oppure applicarla su un mini-batch appena prima del training vero e proprio. Il primo metodo è definito "aumento offline" ed è preferito per set di dati più piccoli, in quanto si finirebbe per aumentare la dimensione del set di un fattore pari al numero di trasformazioni effettuate. Il secondo metodo è noto come "aumento online" o "aumento al volo" ed è preferito per dataset più grandi, in quanto non ci si può permettere l'aumento esplosivo delle dimensioni e si esegue solo su mini-batch. Per ogni tecnica implementata si specifica un fattore di aumento della dimensione del set di dati, al fine di avere un aumento controllato. Le tecniche proposte sono le seguenti:

- **Flipping**: si basa sulla riproduzione dell'immagine a specchio, può essere orizzontale o verticale, il secondo equivale a ruotare un'immagine di 180° e poi applicare un flip orizzontale.

- **Rotation**: rotazione dell'immagine di 90° o 180°, è necessario tener conto che le dimensioni dell'immagine originale possono variare nel caso in cui non sia una matrice quadrata, pertanto si applica prima l'operazione di cropping e in seguito la rotazione.

- **Scaling**: si tratta di fare uno zoom dell'immagine, andando ad evidenziarne alcuni particolari. Anche in questo caso le dimensioni varieranno, mentre alcune porzioni verranno messe in luce, altre verranno tagliate fuori dal bordo dell'immagine.

- **Cropping**: operazione che va a ritagliare porzioni dell'immagine non adatte al training, quali le scritte sulle immagini ecografiche ad esempio, pertanto viene eseguita prima delle altre.

- **Translation**: va a spostare l'immagine lungo l'asse x o y, oppure lungo entrambi. Questa operazione fa si che la rete possa concentrarsi su punti diversi dell'immagine, andando a visualizzare l'oggetto nelle sue diverse porzioni.

- **Gaussian Noise**: il rumore gaussiano viene aggiunto, in generale, per evitare l'overfitting aumentando le capacità di apprendimento della rete. Questo avviene perché le componenti ad ogni frequenza vengono distorte incentivando la rete neurale a guardare oltre.

- **Gamma Correction**: modifica il contrasto dell'immagine andando a pesare i pixels con γ, variabile uniforme nell'intervallo [0.4,2.0]. Per i valori inferiori a 1 l'immagine sarà più chiara, mentre per quelli maggiori di 1 l'immagine apparirà più scura.

- **Shearing**: sposta un lato dell'immagine trasformando un quadrato o un rettangolo in un trapezio. Il parametro in input è l'angolo di deformazione dell'immagine.

Quando vengono eseguite queste trasformazioni si deve tener conto del fatto che non si hanno informazioni dell'immagine al di là dei bordi. Quindi se si dovesse scalare l'immagine verso l'interno quello che potrebbe apparire è l'immagine al centro circondata da uno sfondo nero. Dunque si impongono una serie di zeri sullo sfondo dell'immagine, ottenendo una regione nera in cui l'immagine non è definita. Ma non è l'unica soluzione, ad esempio per immagini di paesaggi si tende a riflettere la stessa immagine oltre i bordi, oppure è possibile estendere i bordi fino alle estremità della dimensione originale. Secondo lo studio [20] aumentare le dimensioni del dataset ha portato a ottimi riscontri nella classificazione, nonostante ciò è necessario tener presente che non tutte le trasformazioni possono avere senso. Se si andasse a considerare l'immagine di un'automobile, ad esempio, la sua rotazione di 180° potrebbe non essere coerente in uno scenario di automobili su strada. Pertanto si andrebbero a creare dati irrilevanti che porterebbero a un errore nella classificazione. I metodi proposti sono stati validati su tre casi di studi medici: diagnosi di melanomi della pelle, immagini istopatologiche e analisi di risonanza magnetica del seno (RMN) per la classificazione delle immagini al fine di fornire una corretta diagnosi. La carenza di dati è tipica nel settore medicale dal momento che il dato trattato è "sensibile" e dunque non disponibile a tutti.[20]

Esempi di modelli

- **AlexNet**: architettura nata nel 2012 chiamata così dall'autore Alex Krizhevsky, vincitore dell'ILSVRC 2012 (ImageNet Large-Scale Visual Recognition Challenge), permette di risolvere il problema della classificazione di immagini dando in output un vettore di lunghezza pari al numero di input. L'iesimo elemento del vettore di uscita rappresenta la probabilità che l'immagine appartenga alla classe iesima. Di conseguenza la somma di tutti gli elementi di tale vettore è pari a 1. La particolarità di tale rete è che offre la possibilità di avere in input immagini di dimensioni abbastanza elevate: 256x256, a discapito dei tempi computazioni di circa 6 giorni. La struttura è composta da 5 layer convoluzionali, intervallati da layer di overlapping max pooling e da 3 layer fully connected. Gli overlapping max pooling sono simili ai max pooling ad eccezione del fatto che i filtri applicati sono sovrapposti l'un l'altro. Gli autori di [21] hanno utilizzato filtri di dimensione 3x3 con un passo di 2 tra le finestre adiacenti. Grazie a questa idea si ha avuto una riduzione del tasso di errore. Analogamente a reti come LeNet anche AlexNet utilizza la funzione di attivazione non lineare ReLU al fine di avere una convergenza più rapida.[21]

- **GoogLeNet**: rete proposta da una ricerca di Google con diverse università nel 2014 nel documento [22]. Questa architettura è stata vincitrice dell'ILSVRC 2014, offrendo una riduzione del tasso di errore di classificazione rispetto ad AlexNet e a VGG (seconda classificata nel 2014). Il modello si ispira a LeNet ma con l'aggiunta di un modulo iniziale basato sulla convoluzione 1x1, convoluzioni molto piccole per diminuire drasticamente il numero di parametri (da 60 milioni in AlexNet a 4 milioni), velocizzando l'addestramento.[22]

- **VGG**: seconda classificata all'ILSVRC 2014 è soprannominata VGGNet o VGG16 dalla comunità che l'ha sviluppata [23]. Il numero 16 indica gli strati convoluzionali con filtri 3x3, solo alcuni intervallati da 5 layers di max pooling, per finire con 3 layers fully connected. La struttura è molto uniforme ma anche profonda, formata da 138 milioni di parametri. Questo fa si che i tempi computazionali sono di circa 1/2 settimane su 4GPU.[23]

- **ResNet**: la "rete neurale residuale" vince l'ILSVRC nel 2015 diventando lo stato dell'arte delle reti convoluzionali. Nasce dall'osservazione delle performance di una rete convoluzionale di base a cui sono stati aggiunti più strati. Contro ogni prognostico si ottenevano dei peggioramenti nell'accuratezza di classificazione, questo almeno durante l'allenamento non dovrebbe mai accadere perché non c'è il rischio di overfitting. Di conseguenza si è compreso che mappature dirette sono difficili da allenare, invece risultava più semplice allenare il loro "residuo", cioè la sottrazione tra una mappa e la successiva, da questo la definizione di “rete residuale”. L'architettura è formata da una

serie di blocchi tra i quali sono interposte delle inserzioni, “skip connection”, in cui si somma al blocco una “identity mapping” e si fa una normalizzazione. [24] Questo viene fatto in modo tale che ciascun strato abbia una sorta di riferimento, una mappa di identità appunto, da cui poter apprendere nonostante la profondità della rete. Così si è stati in grado di addestrare una NN con 152 strati pur avendo una complessità inferiore rispetto a VGGNet. Si evitano inoltre problemi come il *Vanishing gradient*, grazie all'operazione di normalizzazione dei blocchi applicata.[25]

4.1.6 LeNet

LeNet è una rete convoluzionale creata da Yann LeCun nel 1998 e ampiamente utilizzata nel riconoscimento della scrittura. Inizialmente è stata utilizzata dalle banche per riconoscere i numeri scritti a mano sugli assegni, digitalizzati in immagini 32x32 pixel in scala di grigi. In seguito è stata applicata ad immagini con più alta risoluzione portando ad un aumento dei livelli della rete e incrementando i costi computazionali dell'algoritmo. Inizialmente si è studiato il modello base, mettendo in input i campioni dei MNIST, immagini in scala di grigi su cui sono scritte cifre fino a 10. Il compito della rete è quello di interpretare al meglio tali immagini dando come output il numero corrispondente. Il pacchetto di Tensorflow, Keras offre i layers necessari per formare il modello LeNet.

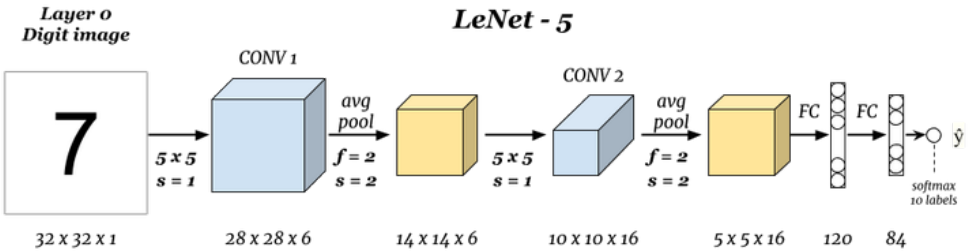


Figura 4.6: Rete neurale convoluzionale LeNet-5
[26]

Prima deve essere definito il layer di input in cui si specifica la dimensione delle immagini, prendiamo l'esempio base dell'immagine del MNIST 32x32. Questo layer diventerà poi l'input di un layer convoluzionale, che si occuperà dell'operazione di convoluzione 2D. Gli argomenti sono il numero dei filtri (6), la loro dimensione (5, sia per l'altezza che per la larghezza), e la funzione di attivazione ReLU. Rispetto al Multi-Layer, in cui la funzione prevalentemente utilizzata è la Sigmoide, nelle reti profonde si predilige la ReLU, per evitare il problema del *Vanishing gradient*. Infatti al momento della retro-propagazione del gradiente, la derivata della Sigmoide è tipicamente minore di 1 e, man a mano che a catena si applica l'operazione di derivazione, si arriva a moltiplicare tra loro termini molto piccoli e sempre minori di 1, portando ad una riduzione dei valori del gradiente nei livelli più profondi. Invece, la derivata della ReLU vale 0 per i valori negativi o nulli, mentre vale 1 per i valori positivi. Così attivazioni sparse dei neuroni possono conferire maggiore robustezza. Le feature maps in uscita hanno dimensioni che dipendono dai parametri di input, la profondità, ad esempio, corrisponde al numero di filtri del layer precedente, quindi 6 feature maps in questo caso. Mentre le dimensioni altezza e larghezza saranno pari a 28x28, dal momento che filtri 5x5 vengono passati su ogni porzione dell'immagine perdendo i 4 pixel alle estremità di ogni angolo. Per preservare le dimensioni

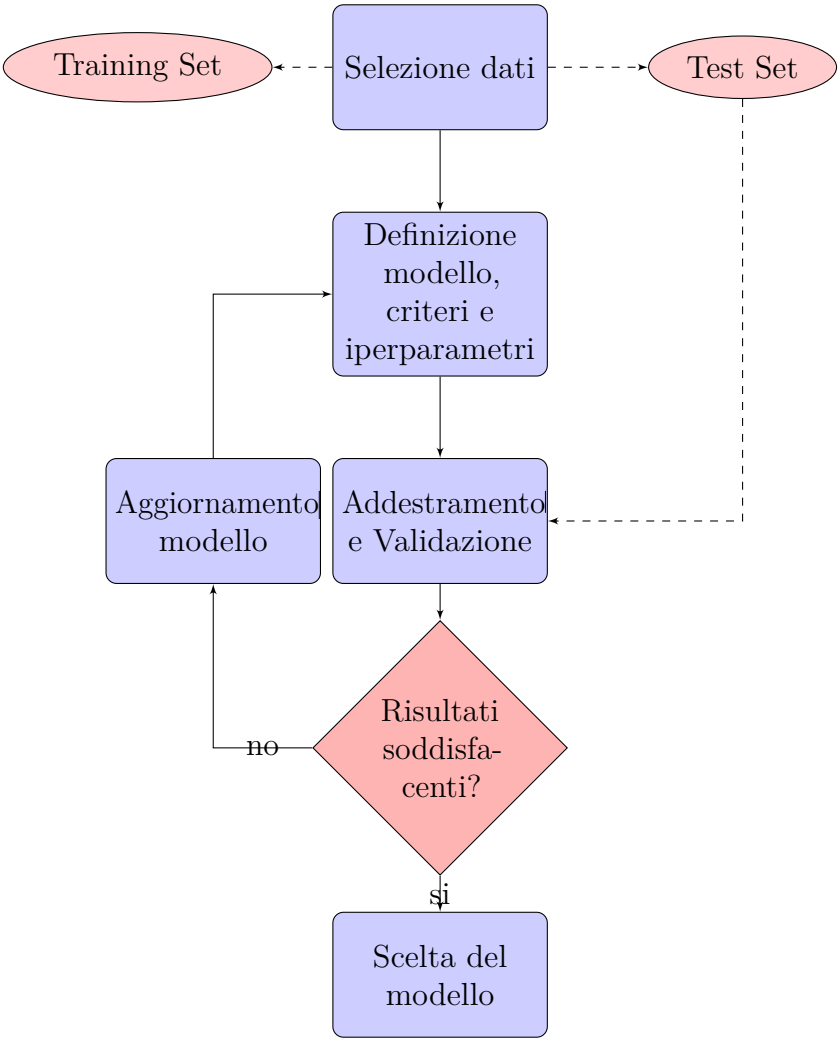
originali è necessario applicare lo "zero padding", cioè aggiungere un bordo di zeri alla matrice originale così da non perdere informazioni. In cascata il layer Max Pooling compierà l'operazione di riduzione delle dimensioni della feature map, preservando unicamente le aree di maggior attivazione, così da ridurre notevolmente i costi computazionali dell'algoritmo. Con il vantaggio di rendere la rete indifferente alle distorsioni e di ridurre il rischio di overfitting. Tale operazione è a livello di ogni feature map, così che il numero nel volume di input è lo stesso dell'output. Si settano le dimensioni di pooling (2) e il parametro "strides", che indica il numero di pixel su cui passa il filtro di pooling ad ogni istante sull'immagine. Mettendo, come nel nostro caso, strides pari a 2, i filtri salteranno due pixel alla volta dando in uscita feature maps di dimensioni 14x14. Quindi più alto sarà tale valore più "jump" si avranno, ottenendo così delle feature maps più piccole. In seguito si riapplicherà nuovamente un layer convoluzionale e uno di pooling, ma con un numero di filtri convoluzionali pari a 16, che permette di ottenere una profondità di feature maps 6x16 con dimensioni 10x10. Gli altri argomenti resteranno invariati. Una formula per calcolare le dimensioni delle feature maps in output ad ogni layer è la seguente (relazione valida sia per la dimensione verticale che orizzontale):

$$Wout = \frac{(Win - F + 2 * P)}{S} + 1 \tag{4.17}$$

In cui F corrisponde alla dimensione del filtro, P all'operazione di zero-padding (0 se non è applicata) e S sta per "stride", il passo dell'operazione. Nel nostro caso: Win=32, F=5, P=0, S=1 -> Wout=28. Man a mano che si riapplica la concatenazione di layer convoluzionali e di Max Pooling si va a ridurre la dimensione della matrice e si aumenta il livello di astrazione. Passando da filtri elementari, come linee verticali e orizzontali, a filtri più particolari in grado di riconoscere forme come setti di una cisti. Per poi arrivare all'ultimo livello in cui si distinguerebbero proprio le cisti uniloculari da quelle multiloculari, nelle loro tipologie. Dopo ci saranno i layers fully connected o dense come nel modello del Multi-Layer Perceptron. In questo secondo livello del modello entreranno esclusivamente le feature maps, perché solo queste possono dare l'informazione sulla locazione delle caratteristiche dell'immagine. L'obiettivo del primo blocco di convoluzioni è proprio quello di prendere il dato grezzo ed estrarre le caratteristiche utili dell'immagine, così da creare mappe facilmente classificabili. Se avessimo una rete unicamente fully connected il numero dei nodi e degli archi utilizzati porterebbe ad un'esplosione computazionale. Per questo si preferisce "condividere" i pesi e ridurre tramite "pooling" le dimensioni degli input ai livelli successivi. I dense layers, i cui nodi ad ogni layer sono collegati a tutti i nodi dei layers precedenti e successivi, avranno come obiettivo quello di classificare le features. Sono stati utilizzati due hidden layers, il primo con 120 neuroni e il secondo con 84. Il layer di output avrà un numero di neuroni in base alla classificazione scelta.

Capitolo 5

Risultati



5.1 Tensorflow

Tensorflow è un software opensource sviluppato da Google nel progetto Google Brain, utilizzato nel calcolo numerico per la *data flow graph*, cioè la modellazione dei grafi. Come accennato in precedenza i tensori sono gli elementi alla base di tutto e consistono in un insieme di valori primitivi modellati sotto forma di un array multi-dimensionale da cui deriva la prima parte del nome "Tensor", mentre "Flow" sta ad indicare il flusso delle operazioni matematiche applicate al fine di creare un vero e proprio "grafo". Questa libreria viene utilizzata in particolare per il Machine Learning nelle reti neurali e ingloba numerose API (application programming interfaces) sia di alto che di basso livello. Nel secondo caso tali procedure tendono ad andare maggiormente nel dettaglio al fine di avere un maggior controllo dell'algoritmo e di controllare i parametri inseriti, e sono quelle che andremo a prediligere. Inoltre Tensorflow presenta numerosi vantaggi nel lato pratico come il calcolo automatico delle derivate, la ricca documentazione e il tool di visualizzazione grafica contro alcuni svantaggi come l'utilizzo alto di memoria, tempi computazionali piuttosto lunghi e scarsità di modelli di reti pre-addestrate. Ma nel nostro caso questo avrà una rilevanza lieve dal momento che cercheremo un modello più "custom", che si adatti al meglio ad una tipologia di immagini "nuove" in questo ambito. Attualmente imprese, sviluppatori e ricercatori come il CERN, la NASA e l'NIH utilizzano API come Keras, una libreria scritta in Python che consente la creazione di modelli per il riconoscimento di immagini, suoni e caratteri, con una flessibilità di basso livello per l'implementazione di ricerche arbitrarie, e allo stesso tempo offre funzionalità di alto livello al fine di velocizzare i cicli di sperimentazione.[27] Le nuove applicazioni di Tensorflow in campo medico sono state utilizzate per la rilevazione di malattie respiratorie e per la prescrizione di antibiotici. Nel primo caso l'algoritmo automatico va ad analizzare i suoni fisiologici, in particolare le frequenze che il medico con il solo fonendoscopio non sarebbe in grado di rilevare, il sistema mima l'orecchio umano andando ad evidenziare eventuali anomalie respiratorie tramite la creazione di uno spettrogramma. Mentre nella seconda applicazione l'algoritmo va a supportare il medico nella scelta della terapia corretta al fine di combattere il batterio responsabile della specifica malattia. L'idea di queste tecnologie, infatti, è quella di sostenere il medico nella diagnosi e dunque nelle successive scelte che riguarderanno il percorso terapeutico del paziente. Nell'ambito di individuazione di cisti ovariche questo punto diventa fondamentale, dal momento che alla fine sarà il medico a decretare la classificazione definitiva. L'algoritmo però andrà ad individuare le zone più a rischio e a mettere in luce delle caratteristiche particolari che lo porteranno a scegliere una tipologia di cisti piuttosto che un'altra. Questo metterà in dubbio il medico al momento della scelta e lo porterà ad analizzare meglio il caso.

5.2 Applicazione di LeNet-5

Il modello della rete è quello proposto in 4.6, con qualche modifica relativa alla dimensione delle immagini di input e alla tecnica di pooling utilizzata (Max Pooling). La scelta è stata fatta per avere in input un'immagine con buona risoluzione ma che comunque non abbia grandezze eccessive che incrementino di troppo i tempi computazionali, così da avere un algoritmo accurato e veloce allo stesso tempo. Per ridurre la rumorosità dell'accuratezza in uscita si è generato il dataset per 10 volte (10 seeds), per avere ad ogni iterazione sia una suddivisione di pazienti in training e test differente che uno *shuffle* distinto dei dati di training in ingresso alla rete. In seguito la tabella che racchiude le scelte degli iper-parametri del modello.

Metodi e Parametri	Modello
Metodo di ottimizzazione	Adam
Funzione di perdita	Categorical Cross Entropy
Funzione di accuracy	Categorical Accuracy
Numero di epoche	100
Layer di output	2 (Background e Uniloculare)
Size delle immagini di input	50x50x3
Batch Size	64
Numero di seeds	10
Processore della macchina	Intel Core i5
Memoria della macchina	8 GB

Tabella 5.1: Caratteristiche del modello

5.3 Analisi Statistica del DataSet

L'obiettivo principale del progetto di tesi sarà quello di identificare le cisti e distinguerle in sei classi: Background, cioè i frame in cui la sonda ecografica non è rivolta verso la cisti, Uniloculare (da intendersi come Uniloculare sierosa), Multiloculare (sierosa), Uniloculare solida, Multiloculare solida e Solida. Per farlo è necessaria un'operazione di conteggio dei frames iniziale e quindi un'analisi statistica del dataset, così da dare in pasto alla rete un set bilanciato e fare in modo che la classificazione sia il più possibile uniforme. Un altro problema è sorto dal momento che i frames sono estratti da video e quindi un frame risulterà strettamente correlato al successivo, pertanto sarà valutata anche la cross-correlazione tra le immagini così da vedere quali devono essere scartate per evitare ridondanze all'interno del dataset. Innanzitutto è necessario considerare la tipologia di dati in ingresso:

omogenei, bilanciati e di un numero sufficiente affinché la rete possa imparare a riconoscere frames diversi allo stesso modo. Questo controllo è necessario dal momento che i video ecografici hanno durate differenti per ogni paziente e quindi il numero di frames sarà diverso per ognuno di essi e per ogni classe considerata. Dunque come prima cosa si tiene in considerazione il fatto che il dataset non è omogeneo e che è sbilanciato. Questo potrebbe portare la rete a prediligere la classificazione di una classe piuttosto che un'altra, facendo si che alcune risultino più rappresentative, pertanto è necessaria una pipeline per la gestione del dataset. In seguito il dataset dovrà essere diviso in training set e test set, rispettivamente nell'80% e nel 20% facendo si che i pazienti del training non siano gli stessi di quelli del test. Quindi come prima cosa dai file excel in cui erano stati registrati i file e le loro assegnazioni sono stati conteggiati tutti i frame per le rispettive etichette. Come si nota dal grafico 5.1, c'è uno squilibrio importante tra le diverse classi, pertanto sarà necessaria un'operazione di sotto-campionamento così da riportare i frames selezionati allo stesso numero per ognuna di esse.

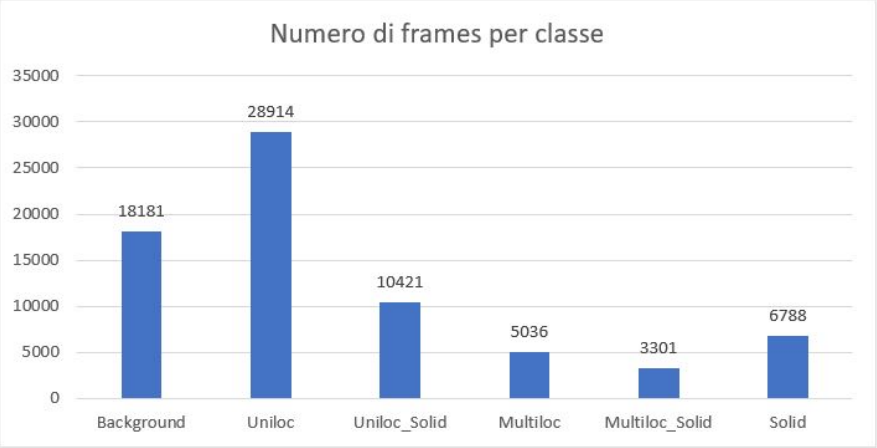


Figura 5.1: Numero di frames per classe

In seguito i passaggi relativi alla preparazione del dataset:

1. suddivisione dei pazienti in training e test;
2. suddivisione dei frames ed etichette per ogni paziente;
3. rimozione delle transizioni;
4. rimozione delle ridondanze;
5. rimozione dei frames e delle corrispettive etichette non selezionate;
6. operazione di sotto-campionamento;

7. preprocessing delle immagini;

8. scrittura su file tfrecord.

Come prima cosa i pazienti vengono suddivisi in training e test secondo la proporzione 80-20, per far si che ci sia una quantità di dati sufficiente ad allenare la rete e una minor quantità per testarla così da verificarne il comportamento con immagini nuove. Ogni video ecografico selezionato viene spacchettato in una serie di frames che andranno a rappresentare o la cisti o il background sulla base dell'istante preso in considerazione. Dunque anche se la paziente ha una cisti multiloculare solida, i frames all'interno del video possono avere etichette differenti in base alla posizione della sonda nel corso del video. In alcuni istanti, seguendo l'esempio, un solo loculo potrebbe essere rappresentato e quindi quel frame sarà etichettato come uniloculare, così come le porzioni dell'ovaio in cui non appare la cisti saranno etichettate come background. Come seconda cosa, è necessario considerare i frames di transizione, cioè le fasi del video in cui si intravede la cisti ma non è ancora ben rappresentata. Questi potrebbero confondere la rete ed indurla ad errori. Così si è scelto un range di eliminazione in corrispondenza delle transizioni valutando la dimensione del dataset per ogni valore settato, come si nota in 5.3 la variabile *expansion* indica il numero di frames rimossi attorno al punto di transizione. Inoltre sono state calcolate le cross-correlazioni tra frames adiacenti e si è notato che il loro valore è molto vicino al picco massimo della funzione. Questo potrebbe portare la rete neurale ad un overfitting, pertanto è necessario scartare frames vicini tra loro. Anche qui viene settata la variabile *stepsize*, che indica il numero di frames tra un'immagine e l'altra che devono essere rimossi.

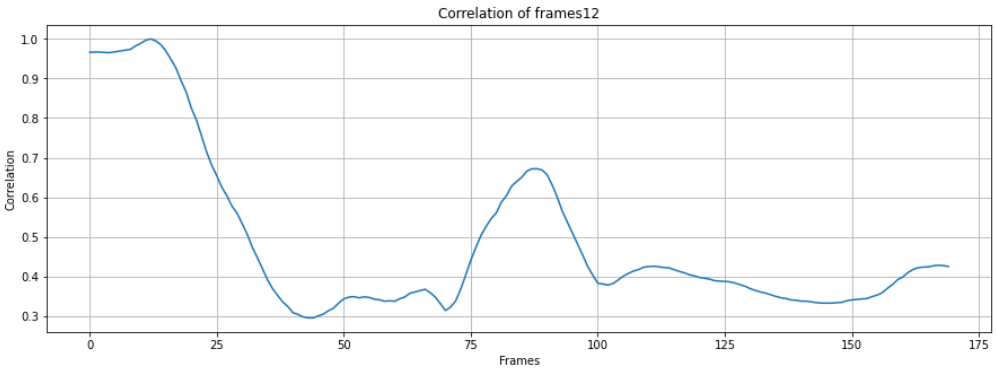


Figura 5.2: Cross-correlazione tra il dodicesimo frame e quelli adiacenti

Il picco a 1 nella figura di esempio evidenzia una correlazione tra il frame considerato con se stesso, mentre i valori sottostanti fanno notare come i frames siano strettamente correlati tra loro e che quindi è necessaria un'operazione di

esclusione delle transizioni per avere un dataset il più omogeneo possibile e senza ricorrenze. A seguire, nel caso in cui vengano selezionate soltanto alcune classi dell'intero dataset per fare classificazione, è necessario escludere i corrispettivi frames ed etichette associate.
Per finire con l'ultimo blocco segue l'operazione di sotto-campionamento, che consente di avere delle classi bilanciate per evitare che la rete possa imparare meglio una classe piuttosto che un'altra.
La tabella sottostante mostra come varia la dimensione del dataset al variare dei parametri *stepsize*, il range di esclusione tra ogni frame ed il successivo; e *expansion*, il range di esclusione delle transizioni (per singolo lato a partire dal punto di transizione).

SEED	stepsize	expansion	TRAIN	TEST
0	3	3	4229	1332
1	3	3	8904	2200
2	3	3	9048	2072
3	3	3	9088	2032
4	3	3	8340	2802
5	3	3	8706	2406
6	3	3	9578	1554
7	3	3	9152	1966
8	3	3	9344	1790
9	3	3	7956	3192

SEED	stepsize	expansion	TRAIN	TEST
0	3	5	8104	2556
1	3	5	8578	2102
2	3	5	8686	1994
3	3	5	8748	1896
4	3	5	8010	2676
5	3	5	8362	2328
6	3	5	9234	1476
7	3	5	8790	1872
8	3	5	8970	1698
9	3	5	7648	3064

SEED	stepsize	expansion	TRAIN	TEST
0	5	3	5066	1582
1	5	3	5350	1318
2	5	3	5432	1242
3	5	3	5446	1210
4	5	3	4988	1668
5	5	3	5228	1448
6	5	3	5724	932
7	5	3	5486	1182
8	5	3	5600	1064
9	5	3	4768	1910

SEED	stepsize	expansion	TRAIN	TEST
0	5	5	4862	1534
1	5	5	5136	1258
2	5	5	5210	1208
3	5	5	5272	1144
4	5	5	4818	1614
5	5	5	4998	1392
6	5	5	5520	876
7	5	5	5286	1124
8	5	5	5382	1022
9	5	5	4564	1846

SEED	stepsize	expansion	TRAIN	TEST
0	10	3	2550	800
1	10	3	2680	658
2	10	3	2716	622
3	10	3	2708	610
4	10	3	2504	832
5	10	3	2606	728
6	10	3	2846	464
7	10	3	2724	594
8	10	3	2804	534
9	10	3	2406	954

SEED	stepsize	expansion	TRAIN	TEST
0	10	5	2436	774
1	10	5	2584	620
2	10	5	2630	602
3	10	5	2630	572
4	10	5	2390	812
5	10	5	2508	698
6	10	5	2766	432
7	10	5	2660	562
8	10	5	2690	510
9	10	5	2288	932

Figura 5.3: Dimensioni del dataset al variare della stepsize e dell'expansion

Si è scelto di settare sia *stepsize* che *expansion* a 5. Così da prendere un'immagine ogni 5 ed escludere 10 frames ad ogni transizione ed avere un dataset comunque ampio (circa 5000 dati di train e 1200 di test), privo di ridondanze.
Nella fase di preprocessing si è tenuto conto dell'eliminazione dei bordi dell'immagine

ecografica, contenenti le scritte che individuano frequenza, codice identificativo, misurazioni, potenza acustica etc., le dimensioni della finestra di "cropping" sono: [106,-61,95,-125]. Mentre viene applicato un "resize" di [50x50x3]. La dimensione originale delle immagini ecografiche è pari a [566x800x3], dunque dimensioni nettamente superiori rispetto alle immagini del MNIST [32x32x1], con cui si erano fatte le simulazioni dei codici. Da questo ne consegue un'alta richiesta di memoria e un'elevata latenza delle operazioni. La compressione eccessiva invece comporterebbe una scarsa qualità delle immagini in termini di risoluzione e una conseguente riduzione dell'accuratezza di classificazione. Pertanto è necessario mediare tra i due problemi e trovare il giusto compromesso.[28]
Per finire si è scelto di organizzare il dataset creando una libreria TFRecord, che possa organizzare al meglio le immagini, cioè i frames di ogni video associati alla propria label, etichetta che corrisponde alla tipologia di cisti o al background. Quando si lavora con grandi basi di dati, utilizzare un formato di file binario per la memorizzazione può avere un impatto significativo sulle prestazioni della pipeline e di conseguenza sul tempo di formazione del modello. I dati binari sono più facili da gestire e occupano meno spazio sul disco di memoria, inoltre richiedono meno tempo per la copia e la lettura risulta più efficiente. Un altro vantaggio è l'ottimizzazione per l'utilizzo combinato con Tensorflow, così da rendere più facile la combinazione di più set di dati integrando le funzionalità di importo ed elaborazione di essi fornite dalla libreria. Poiché solo i dati necessari al momento vengono caricati da disco e poi elaborati. Anche la memorizzazione in sequenza può essere semplificata con l'utilizzo di questo formato, in particolare per serie temporali o codifiche di parole. Tuttavia ci sono anche degli svantaggi da considerare, come la limitata documentazione sulla conversione dei dati nel formato "tfrecords" e poi la riconversione al momento della lettura. I dati vengono dunque serializzati tramite un buffer di protocollo e inseriti in modo efficiente in una struttura. L'immagine contiene una serie di caratteristiche, come la prima, la seconda e la terza dimensione, la label della classe corrispondente e il path del video da cui è stata estratta, pertanto tali features devono essere memorizzate sul file. Una volta inizializzate le liste, i dati serializzati vengono scritti su disco e poi letti al momento dell'allenamento della rete.[29]
I risultati presentati in seguito saranno relativi ad una classificazione tra cisti uniloculari e il background, dunque una classificazione binaria. Questo è possibile andando a selezionare in input all'algoritmo unicamente queste due tipologie, così da discriminare la presenza o l'assenza della cisti. Si è deciso di procedere per step e quindi allenare la rete su due classi e verificare le performance, per poi poter estendere la classificazione ad un livello più alto e complesso come quello multiclasse. Questa scelta è stata intrapresa anche per la scarsità di dati ecografici inerenti e cisti multiloculari, multiloculari solide e solide, pertanto una volta che arriveranno ulteriori immagini sarà possibile allenare nuovamente la rete con un dataset di input più ampio.

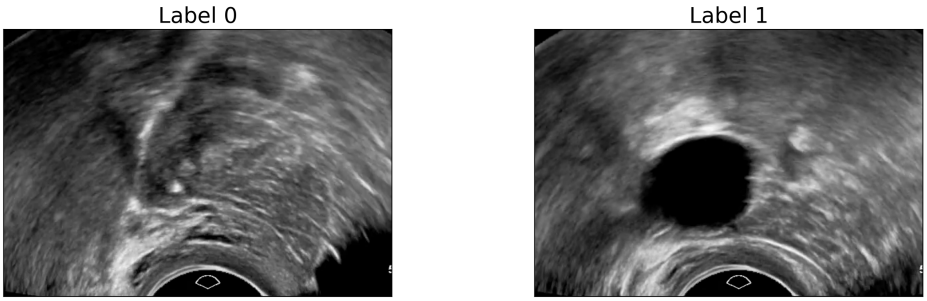


Figura 5.4: Labeling delle immagini

La tabella sottostante indica le prestazioni della rete a seguito della scelta dei parametri, in particolare i valori di training e test, sia totali che su ogni classe, mediati su 10 seeds. Si può notare che le performance, anche in assenza

Training		Test		Time
0.9981		0.8054		240min
Background 0.9978	Uniloculare 0.9984	Backgroud 0.8083	Uniloculare 0.8025	

Tabella 5.2: Valori iniziali di training e test

di miglioramenti aggiuntivi, risultano piuttosto buone. Più di dell'80% sul test. Questo probabilmente dovuto al fatto che si sta andando a classificare l'assenza e la presenza della cisti, sui frames più numerosi del dataset.

5.4 Regolarizzazioni

Le tecniche di regolarizzazione sono fondamentali in una rete neurale al fine di evitare l'overfitting. Aggiungere le regolarizzazioni significa aggiungere "rumore" al dataset, far si che l'algoritmo venga ostacolato nell'apprendimento eccessivo dei dati di input del training set. Inizialmente sono state fatte prove senza regolarizzazioni e successivamente sono state provate separatamente le regolarizzazioni: L2, Dropout e Batch Normalization.

5.4.1 Tecniche di regolarizzazione: L2

Come prima cosa nell'applicazione delle regolarizzazioni si è valutato in quali layers applicarle, se solo in quelli convoluzionali, solo nei fully connected, oppure in entrambi. In seguito alle diverse prove e all'analisi in letteratura, L2 con tasso di regolarizzazione di valore pari a 0.001 applicata sui fully connected ha prodotto risultati migliori.
I seguenti grafici riporteranno media e deviazione standard delle *accuracy* sui 10 seeds, dunque su ogni generazione del dataset e rispettivo training.

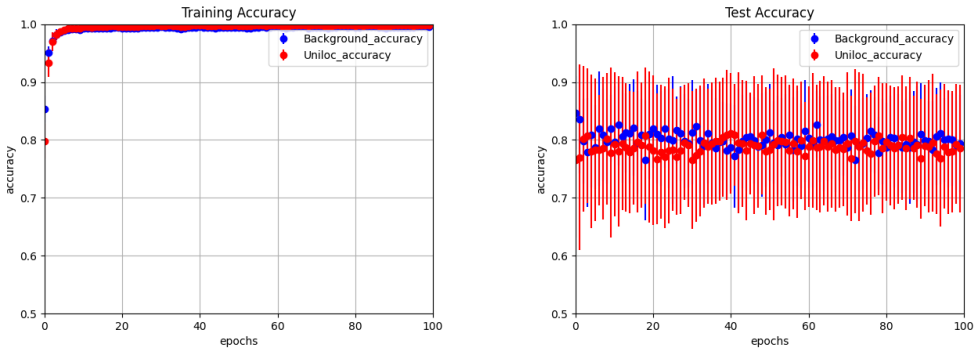


Figura 5.5: Training e Test con regolarizzazione L2

5.4.2 Tecniche di regolarizzazione: Dropout

Così come per la regolarizzazione L2 anche con il Dropout vengono fatte diverse prove: il range di probabilità di "dropping" viene variato da 0.2 a 0.5 a passi di 0.1 e il layer viene applicato prima sui layers dense, poi tra layers convoluzionali e quelli di max pooling e per finire su entrambi. Da letteratura sono stati proposti diversi modelli: nel documento originale [30] sono state fatte prove prima inserendo il Dropout sui layers fully connected con rate a 0.5, in seguito aggiungendolo anche ai layer convoluzionali con rate 0.2. In questo secondo caso le performance dimostravano un effettivo miglioramento.[31] L'implementazione standard di Tensorflow

utilizza il Dropout unicamente sui layer fully connected, come anche in [32] perchè sono quelli che possiedono un grande quantità di parametri da allenare e che quindi sono tendenzialmente portati all'overfitting.
Un'altra questione affrontata è legata al posizionamento nei layer convoluzionali se prima o dopo quelli di max pooling. Considerando che questa tecnica funziona per neurone, fare il "dropping" del neurone significa eliminare la sua mappa caratteristica. Il pool agisce separatamente su ognuna di esse, quindi teoricamente non c'è differenza nell' applicazione del Dropout prima o dopo questo strato.

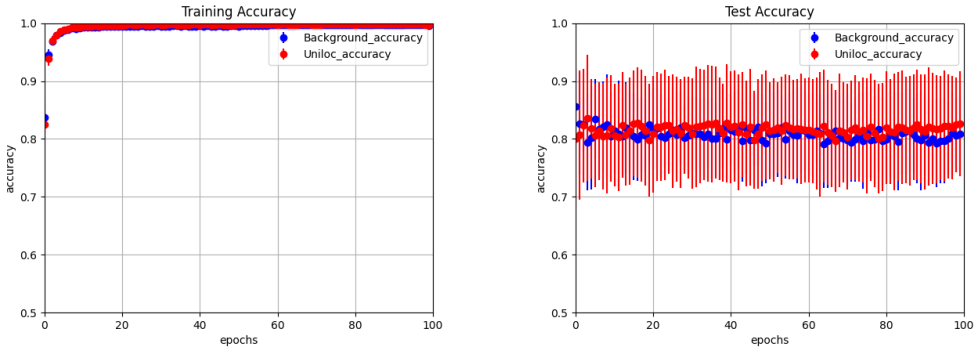


Figura 5.6: Training e Test con regolarizzazione Dropout

5.4.3 Tecniche di regolarizzazione: Batch Normalization

La Batch Normalization è ad oggi la tecnica che ha prodotto risultati migliori in letteratura ed è quella prevalentemente utilizzata, come illustrano i grafici la percentuale di accuratezza tende ad avere un leggero miglioramento rispetto alle performance di base e rispetto alle altre tecniche di regolarizzazione.

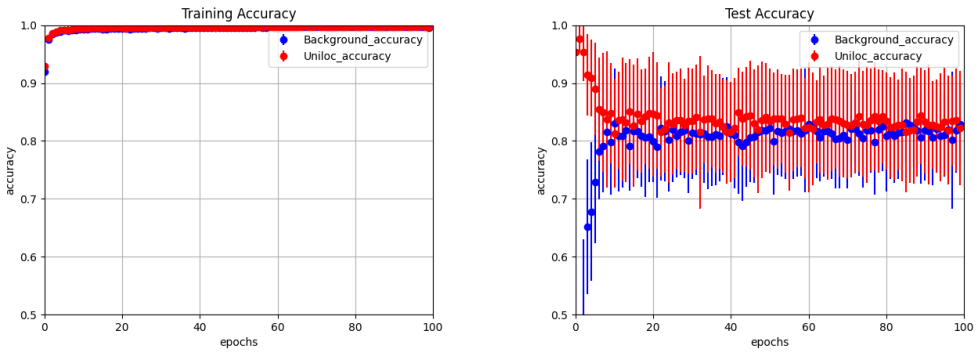


Figura 5.7: Training e Test con regolarizzazione BatchNorm

A seguito il grafico che mette a confronto la Test Accuracy di ogni tecnica di regolarizzazione applicata rispetto alle performance di base, seguito dalla tabella dei tempi di training. Come si nota la Batch Normalization risulta superiore rispetto alle altre pertanto verrà inglobata nel modello nel corso delle prove successive.

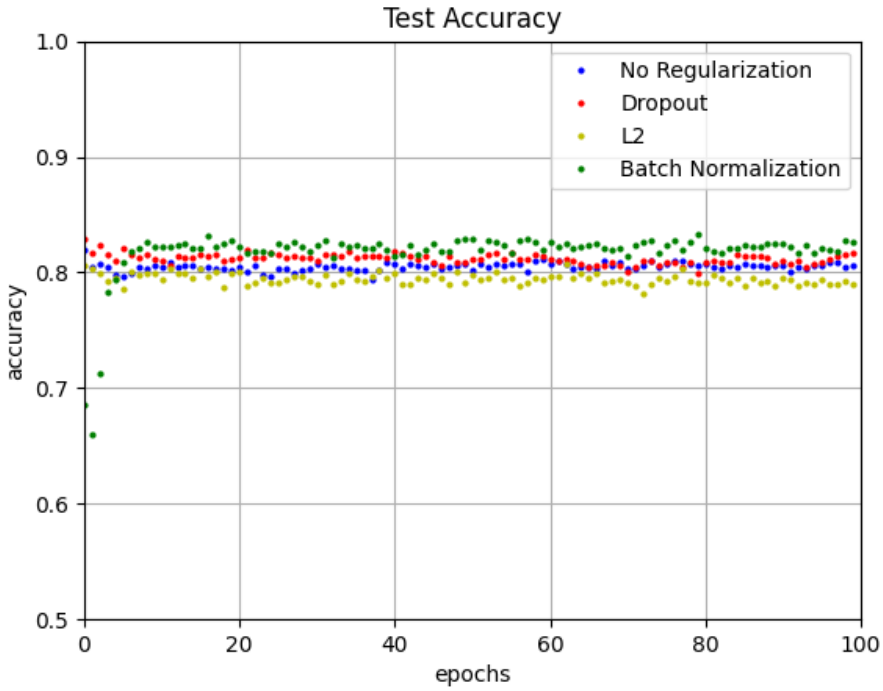


Figura 5.8: Test Accuracy delle tecniche di regolarizzazione

Metodo	Time
No Regularization	240min
L2	213min
Dropout	252min
Batch Normalization	252min

Tabella 5.3: Tempi di Training per 100 epoche

5.5 Data Augmentation

Allo stato dell'arte le reti neurali convoluzionali hanno un dataset dell'ordine dei milioni, mentre le immagini ecografiche utilizzate sono dell'ordine delle migliaia. Nel campo medico, infatti, aumentare i dati diventa fondamentale data la mancata disponibilità pubblica, poiché l'accesso alle cartelle cliniche individuali è fortemente protetto dalla legislazione e deve essere dato un consenso adeguato. In gran parte dei casi, questo processo è ostacolato dalla burocrazia e dai costi, mentre la raccolta dei risultati è sbilanciata tra soggetti patologici e normali.
E' necessaria pertanto un'operazione di "aumentazione" del dataset, tramite due principali tipologie di tecniche: quelle geometriche e quelle dedite all'*Image Enhancement*. Mentre per le prime è necessario controllare che le cisti siano sempre contenute all'interno dell'immagine e che lo *zero-padding* venga applicato correttamente, per le seconde è importante capire come impostare i parametri che attuino le modifiche di luminosità e contrasto. La tabella sottostante riporta le operazioni scelte e i parametri settati per ognuna.

Tecniche di Augmentation	Parametri	Valori
Zooming	Percentuale di zoom	10%
Rotation	Gradi di rotazione	+/- 10°
Adding noise	Media e Deviazione Standard	10,20
Shearing	Gradi di deformazione	15
Shiftx Shifty	Valori di traslazione per ogni asse	-0.1,-0.05,0.05,0.1
Gamma Correction	Valori di γ	0.4,0.5,0.6
Scaling	Percentuale di scaling su altezza e larghezza per ogni asse	1.1,1.2

Tabella 5.4: Data Augmentation: Primo Metodo

Inizialmente vengono applicate le aumentazioni al 20% del training set scelto randomicamente, dunque ciascuna classe sarà incrementata del 10% e ogni immagine selezionata aumentata con una sola tecnica, scelta anch'essa in modo random dalla lista delle operazioni. In seguito, allo stesso modo, si andrà ad aumentare la percentuale al 30% e al 40%.

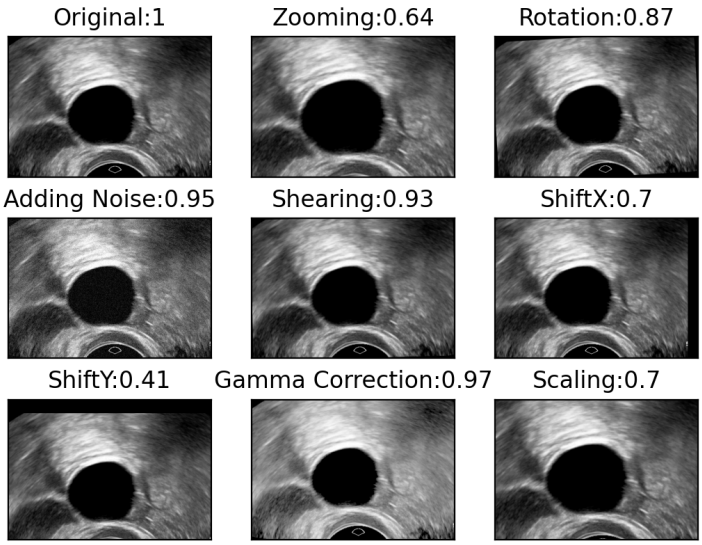


Figura 5.9: Data Augmentation con Coefficiente di correlazione corrispondente: Primo Metodo

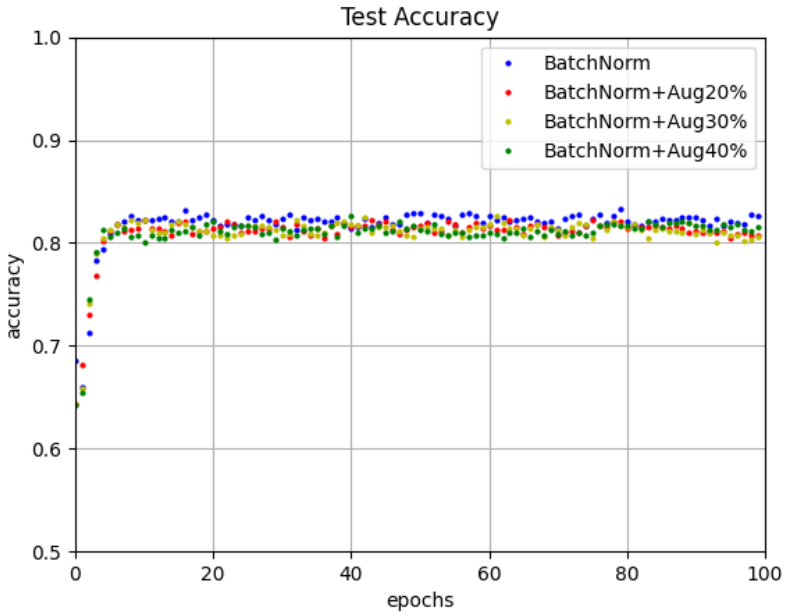


Figura 5.10: Test Accuracy al variare della percentuale di aumentazione del dataset

Nonostante la percentuale di aumentazione viene incrementata non si ottengono risultati soddisfacenti, è necessario quindi cambiare la pipeline per il processamento delle immagini. Dunque le aumentazioni saranno applicate in cascata ad ogni immagine selezionata settando dall'esterno la quantità di immagini che si vuole aumentare. In modo random saranno scelte nel dataset i frames (nella giusta proporzione per le due classi) e ad ognuno di essi saranno applicate tutte le aumentazioni. Per dare una maggior randomicità al dataset, anche l'ordine di selezione delle tecniche viene scelto in modo random. In 5.5 sono riportate le modifiche dei parametri applicate.

Tecniche di Augmentation	Parametri	Valori
Zooming	Percentuale di zoom	10%
Rotation	Gradi di rotazione	+/- 10°
Adding noise	Media e Deviazione Standard	10,range(10,20)
Shearing	Gradi di deformazione	+/- 20°
Shiftx Shifty	Valori di traslazione per ogni asse	-0.1,-0.05,0.05,0.1
Gamma Correction	Valori di γ	0.5,0.6,0.7
Scaling	Percentuale di scaling su altezza e larghezza per ogni asse	1.1,1.2

Tabella 5.5: Data Augmentation: Secondo Metodo

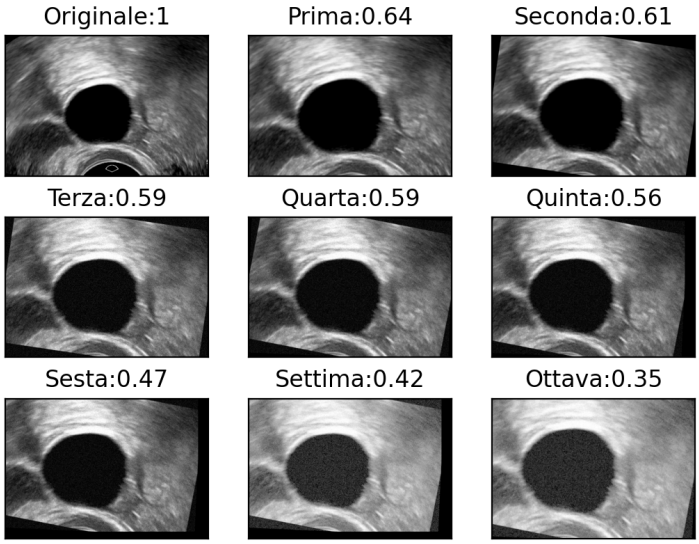


Figura 5.11: Data Augmentation con Coefficiente di correlazione corrispondente: Secondo Metodo

Il confronto seguente viene svolto tra il primo metodo con percentuale di augmentation pari al 40% e il secondo metodo in cui si va a raddoppiare il dataset iniziale, così da paragonare due tecniche partendo circa dalla stessa numerosità di dati posti in ingresso alla rete.

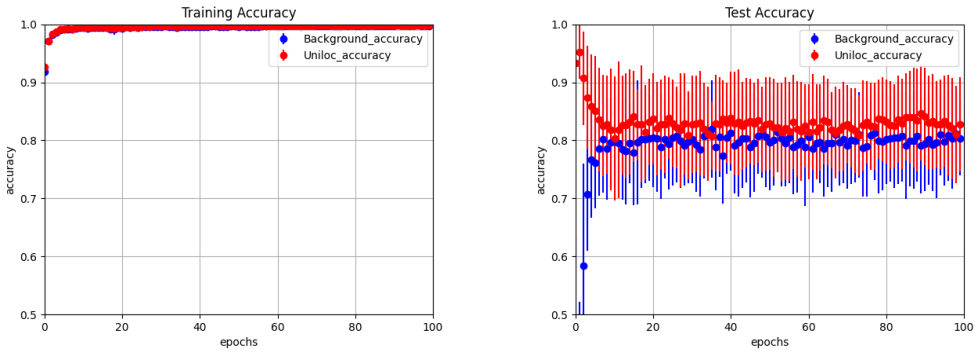


Figura 5.12: Medie e Deviazioni Standard di Training e Test Accuracy: Primo Metodo di Augmentation

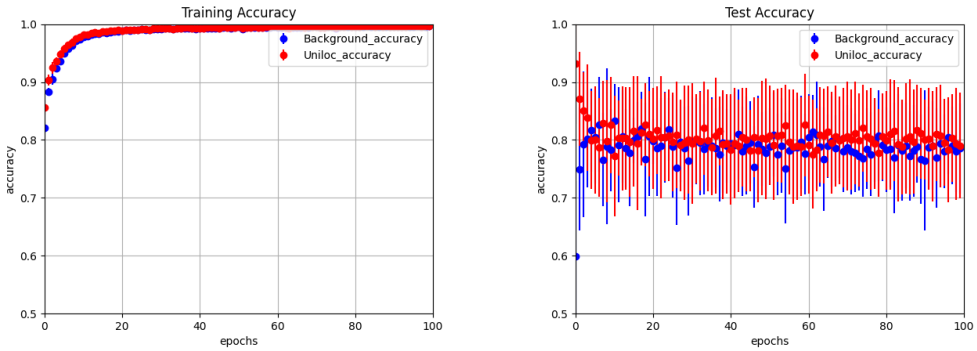


Figura 5.13: Medie e Deviazioni Standard di Training e Test Accuracy: Secondo Metodo di Augmentation

	Training	Test	Time
Primo	0.99703	0.81593	390min
Secondo	0.99657	0.78771	444min

Tabella 5.6: Confronto tra metodi di Augmentation

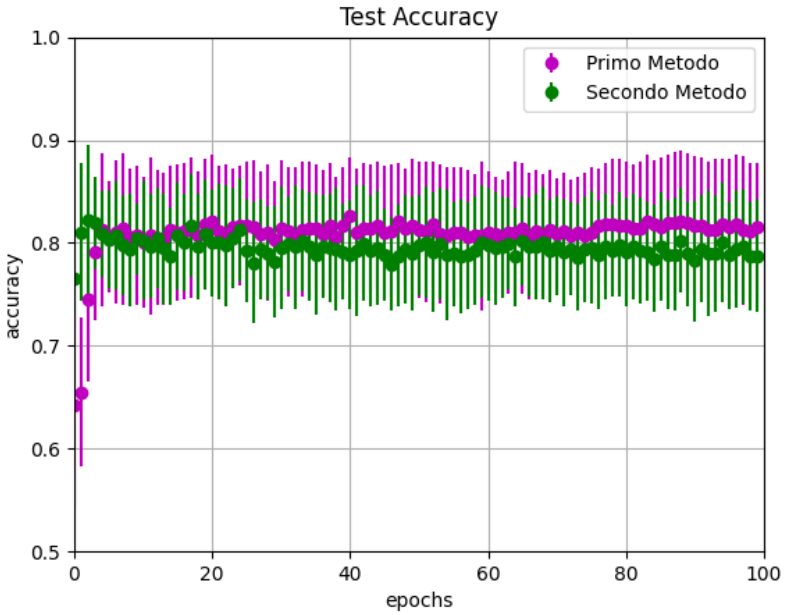


Figura 5.14: Confronto tra metodi di Augmentation

Analizzando complessivamente i due metodi non si notano grandi differenze, se non per il lieve sbilanciamento tra le due classi nel primo che invece scompare nel secondo. Far si che la rete possa imparare allo stesso modo entrambe le classi è un grande vantaggio, pertanto nonostante mediamente la prima da risultati migliori (5.6), la seconda è preferibile.
Come prove successive si decide di introdurre nuove reti, andato a modificare completamente sia il modello che il tipo di approccio di "basso livello", adottato precedentemente. Questo viene sostituito con uno di "alto livello" al fine di utilizzare la tecnica del *Transfer Learning*.

5.6 Transfer Learning

Dal momento che l'aumento del dataset non portava a risultati significativi, si è optato per l'applicazione di una tecnica che utilizza i pesi di un'altra rete convoluzionale, addestrata su ImageNet, alla quale si aggancia un layer di fully connected al fine di far previsioni sulle nostre classi. Tale procedura viene definita *Transfer Learning*. Il trasferimento dell'apprendimento ha come target il miglioramento del medesimo in un nuovo compito attraverso il trasferimento di conoscenze da un compito correlato che è già stato appreso.[33]
ImageNet è il dataset utilizzato in questo caso, esso ingloba oltre 15 milioni di immagini raccolte dal web a risoluzione variabile e classificate su 22.000 categorie. Tale dataset viene utilizzato anche nel concorso annuale *ImageNet Large-Scala Visual Recognition Challenge* (ILSVRC) che utilizza un sottoinsieme di ImageNet con circa 1000 immagini in ciascuna delle 1000 classi. Lo scopo principale del "trasferimento di apprendimento" è quello di utilizzare agenti di training addestrati su delle *source task* e in seguito trasferirli sulle *target task* a cui sono correlati. Dunque si va ad aggiungere ai dati standard di formazione un'informazione aggiuntiva determinata dagli strati fully connected che determineranno la classificazione delle classi Background e Uniloculare. Il vantaggio principale si basa sui tempi di training nettamente ridotti rispetto a quelli che impiegherebbero delle reti più complesse (VGG16, Inception etc...) per imparare il target da zero. In questo modo è stato possibile fare un confronto tra LeNet-5, utilizzata inizialmente, e una rete convoluzionale maggiormente profonda. Stando ai dati utilizzati, se escludessimo l'utilizzo del transfer learning, il training impiegherebbe diverse ore su un unico seed per ottenere i risultati, mentre in questo caso i tempi verranno ridotti a circa un'ora. Al fine di evitare l'overfitting ed avere una maggior selettività del dato in ingresso, in coda al modello vengono posti gli ultimi tre strati fully connected originali utilizzati in precedenza.
E' necessario, inoltre, tener conto della diversità dei dati di input con cui è stata pre-addestrata una rete come VGG16, e i nostri dati originali. Le ConvNet possiedono caratteristiche generiche, come i rilevatori di blob di colore e i rilevatori di bordi, mentre gli strati finali possiedono caratteristiche man a mano sempre più vicine al dataset di input. Per questo diventa fondamentale l'aggiunta degli ultimi due strati, così che negli strati convoluzionali si utilizzano i pesi di ImageNet, mentre i pesi degli strati profondi vengono riaddestrati per divenire specifici per il caso corrente.
In questo modo la rete utilizzata apprende le caratteristiche di un dominio più ampio grazie ad un pre-addestramento, dopo di che la funzione di classificazione viene aggiunta per ottimizzare la rete e farle apprendere le caratteristiche di un dominio più specifico. Le reti prevalentemente utilizzate per la classificazione di più di mille diverse classi di oggetti nel dataset ImageNet sono VGG16, ResNet50

e InceptionV3.[34] Un'altra considerazione è rivolta alla dimensione delle immagini di input con cui sono state pre-addestrate le reti.[35] Non tutte le dimensioni sono valide, ma sono quelle nel range [32x32:512x512] Di default VGG16 e ResNet50 hanno immagini in ingresso di dimensione 224x224x3. Per motivi di memoria e di tempi computazionali si applica un *resize* alle immagini di input pari a 128x128x3. Seguirà un confronto delle performance delle reti con un dataset di queste dimensioni: il modello iniziale LeNet-5 con l'applicazione della Batch Normalization e la rete VGG16 tramite *Transfer Learning*.

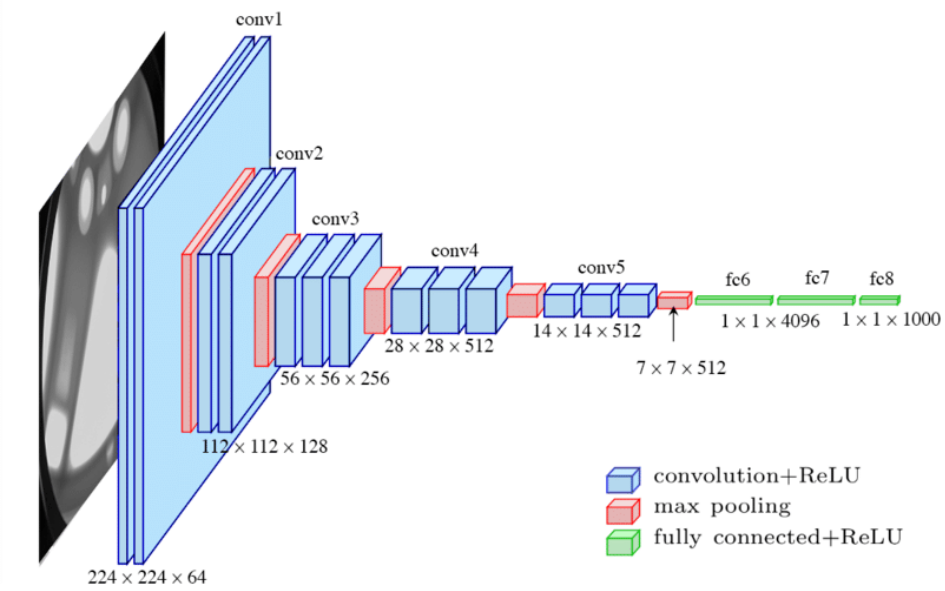


Figura 5.15: Rete neurale convoluzionale VGG16 [36]

5.7 Confronto tra Reti Convoluzionali

Seguono le specifiche relative alle reti neurali convoluzionali poste a confronto. Il numero di epoche è sceso a 50 al fine di evitare l'overfitting, visualizzando la convergenza delle reti a questo valore.

Metodi e Parametri	Modello
Metodo di ottimizzazione	Adam
Funzione di perdita	Categorical Cross Entropy
Funzione di accuracy	Categorical Accuracy
Numero di epoche	50
Layer di output	2 (Background e Uniloculare)
Size delle immagini di input	128x128x3
Batch Size	32
Numero di seeds	5
Processore della macchina	Intel Core i5
Memoria della macchina	8 GB

Tabella 5.7: Caratteristiche del modello

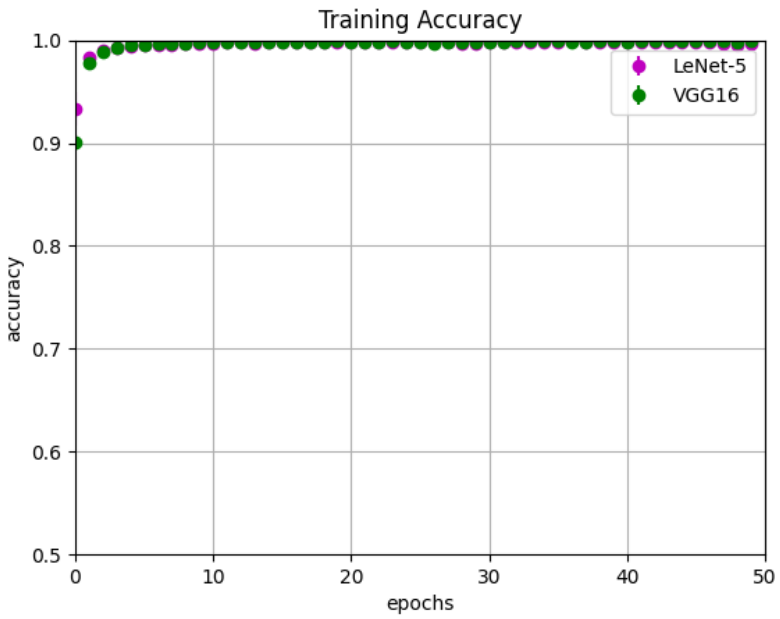


Figura 5.16: Confronto Training Accuracy

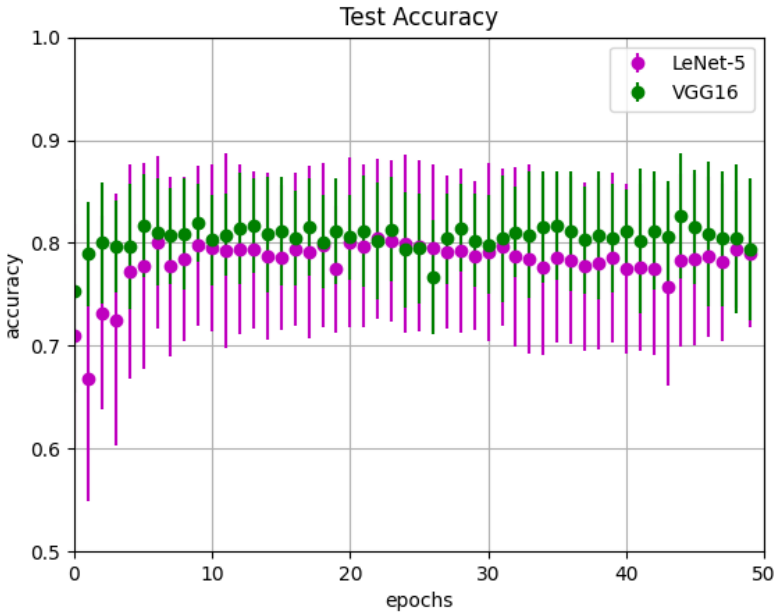


Figura 5.17: Confronto Test Accuracy

Reti	Training	Test
LeNet-5	0.996	0.794
VGG16	0.999	0.804

Tabella 5.8: Performance per 50 epoche

Le accuracy delle due reti sono molto simili sia nel Training che nel Test, leggermente più alte quelle della VGG16. Mentre invece ci si aspettava che la rete VGG16 desse dei risultati nettamente superiori. Questo non accade probabilmente per la dimensione delle immagini di input, infatti andando a ridurre la size delle immagini a 128x128x3 rispetto a 224x224x3 (dimensione standard di VGG16) si vanno a sottrarre delle *features* all'immagine che porterebbero la rete ad apprendere maggiormente.

Capitolo 6

Conclusioni

Il tema innovativo del progetto di tesi è stato spinto dall'idea di creare un supporto al medico in fase di diagnosi ecografica di cisti ovariche. Attualmente quello che è stato presentato è solo l'incipit del progetto globale e si è sviluppato cercando di ottimizzare al meglio il dataset di input andando a rimuovere frames ridondanti, per poi introdurre tecniche di regolarizzazione e Data Augmentation al fine di migliorare le performance iniziali. Complessivamente le performance risultano soddisfacenti per il raggiungimento di un 80% di accuratezza di classificazione sia per la rete LeNet-5 che per VGG16. Considerando inoltre che la rete LeNet-5 è una CNN poco profonda e che la qualità delle immagini di input era piuttosto bassa, ottenere risultati buoni non era affatto scontato.
Il progetto procederà con l'incremento del dataset al fine di raggiungere una classificazione multiclasse e la costruzione di modelli sempre più complessi che possano andare ad analizzare l'immagine in una maggiore qualità con il supporto di un cloud su cui allenare le reti.
Entrare in possesso di un dataset ampio di immagini mediche è una necessità per approcci di machine learning come questo, pertanto creare una piattaforma condivisibile tra aziende sanitarie dei dati "sensibili" porterebbe ad un grande vantaggio per la generazione di software di riconoscimento, per il confronto tra radiologi ed ecografisti e di conseguenza per una più facile diagnosi delle patologie rare.

Bibliografia

[1] Rosario F. Donato Paolo Castano. *Anatomia dell'Uomo*. Milano: Edi.Ermes, 2008 (cit. a p. 2).

[2] J.C.Aster V.Kumar A.K.Abbas. *Robbins e Cotran. Le basi patologiche delle malattie. Patologia generale*. Milano: Elsevier srl, 2010 (cit. alle pp. 2, 6).

[3] Atlante Anatomico. *APPARATO GENITALE FEMMINILE Ovaie, Tube e Utero*. URL: `http://www.benessere.com/salute/atlante/genitali_f_02.htm` (cit. a p. 3).

[4] Erling Ekerhovd, Heinrich Wienerroith, Alf Staudach e Seth Granberg. «Preoperative assessment of unilocular adnexal cysts by transvaginal ultrasonography: a comparison between ultrasonographic morphologic imaging and histopathologic diagnosis». In: *American journal of obstetrics and gynecology* 184.2 (2001), pp. 48–54 (cit. a p. 5).

[5] Giulia Bertelli. *CA 125: Antigene Tumorale 125*. URL: `https://www.mypersonaltrainer.it/salute/CA-125.html` (cit. a p. 5).

[6] Dr. Roberto Gindro. *Cisti ovariche: sintomi, cause, pericoli, cura*. URL: `https://www.farmacoecura.it/donna/cisti-ovariche-sintomi-cause-pericoli-cura/` (cit. a p. 6).

[7] Dirk Timmerman et al. «Simple ultrasound-based rules for the diagnosis of ovarian cancer». In: *Ultrasound in Obstetrics and Gynecology: The Official Journal of the International Society of Ultrasound in Obstetrics and Gynecology* 31.6 (2008), pp. 681–690 (cit. a p. 12).

[8] Frank De Smet et al. «New models to predict depth of infiltration in endometrial carcinoma based on transvaginal sonography». In: *Ultrasound in Obstetrics and Gynecology: The Official Journal of the International Society of Ultrasound in Obstetrics and Gynecology* 27.6 (2006), pp. 664–671 (cit. a p. 12).

[9] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. «Deep learning». In: *nature* 521.7553 (2015), pp. 436–444 (cit. a p. 16).

[10] Simon S Haykin et al. *Neural networks and learning machines/Simon Haykin.* 2009 (cit. alle pp. 17, 20).

[11] Sebastian Ruder. «An overview of gradient descent optimization algorithms». In: *arXiv preprint arXiv:1609.04747* (2016) (cit. a p. 21).

[12] Claudio Casellato. *Qual è la differenza tra funzione di attivazione e funzione di trasferimento in una rete neurale?* URL: `https://it.quora.com/Qual-%C3%A8-la-differenza-tra-funzione-di-attivazione-e-funzione-di-trasferimento-in-una-rete-neurale` (cit. a p. 22).

[13] William W Guo e Heru Xue. «Crop yield forecasting using artificial neural networks: A comparison between spatial and temporal models». In: *Mathematical Problems in Engineering* 2014 (2014) (cit. a p. 23).

[14] Sergey Ioffe e Christian Szegedy. «Batch normalization: Accelerating deep network training by reducing internal covariate shift». In: *arXiv preprint arXiv:1502.03167* (2015) (cit. a p. 24).

[15] Mohamed Abdel-Nasser e Osama Ahmed Omer. «Ultrasound image enhancement using a deep learning architecture». In: *International Conference on Advanced Intelligent Systems and Informatics*. Springer. 2016, pp. 639–649 (cit. a p. 24).

[16] Luis Miralles-Pechuán, Dafne Rosso, Fernando Jiménez e Jose M García. «A methodology based on Deep Learning for advert value calculation in CPM, CPC and CPA networks». In: *Soft Computing* 21.3 (2017), pp. 651–665 (cit. a p. 25).

[17] Baris Kayalibay, Grady Jensen e Patrick van der Smagt. «CNN-based segmentation of medical imaging data». In: *arXiv preprint arXiv:1701.03056* (2017) (cit. a p. 26).

[18] Michael A Nielsen. *Neural networks and deep learning*. Vol. 2018. Determination press San Francisco, CA, 2015 (cit. a p. 27).

[19] Andrea Missinato. *Le Reti Neurali Convoluzionali, ovvero come insegnare alle macchine a riconoscere per astrazione*. URL: `https://www.spindox.it/it/blog/reti-neurali-convoluzionali-il-deep-learning-ispirato-alla-corteccia-visiva/` (cit. a p. 28).

[20] Agnieszka Mikołajczyk e Michał Grochowski. «Data augmentation for improving deep learning in image classification problem». In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, pp. 117–122 (cit. a p. 30).

[21] Alex Krizhevsky, Ilya Sutskever e Geoffrey E Hinton. «Imagenet classification with deep convolutional neural networks». In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. a p. 31).

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke e Andrew Rabinovich. «Going deeper with convolutions». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9 (cit. a p. 31).

[23] Karen Simonyan e Andrew Zisserman. «Very deep convolutional networks for large-scale image recognition». In: *arXiv preprint arXiv:1409.1556* (2014) (cit. a p. 31).

[24] Jie Peng et al. «Residual convolutional neural network for predicting response of transarterial chemoembolization in hepatocellular carcinoma from CT imaging». In: *European radiology* 30.1 (2020), pp. 413–424 (cit. a p. 32).

[25] Alfredo Canziani, Adam Paszke e Eugenio Culurciello. «An analysis of deep neural network models for practical applications». In: *arXiv preprint arXiv:1605.07678* (2016) (cit. a p. 32).

[26] Andrea Provino. *LeNet-5 CNN Networks*. URL: `https://andreaprovino.it/lenet-5/` (cit. a p. 33).

[27] TensorFlow. *An end-to-end open source machine learning platform*. URL: `https://www.tensorflow.org` (cit. a p. 36).

[28] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber e Danny Z Chen. «Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation». In: *Advances in neural information processing systems*. 2016, pp. 3036–3044 (cit. a p. 41).

[29] Thomes Gamauf. *Tensorflow Records? What they are and how to use them*. URL: `https://medium.com/mostly-ai/tensorflow-records-what-they-are-and-how-to-use-them-c46bc4bbb564` (cit. a p. 41).

[30] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever e Ruslan R Salakhutdinov. «Improving neural networks by preventing co-adaptation of feature detectors». In: *arXiv preprint arXiv:1207.0580* (2012) (cit. a p. 43).

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever e Ruslan Salakhutdinov. «Dropout: a simple way to prevent neural networks from overfitting». In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958 (cit. a p. 43).

[32] Y Gao, Mohammad Ali Maraci e J Alison Noble. «Describing ultrasound video content using deep convolutional neural networks». In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2016, pp. 787–790 (cit. a p. 44).

[33] Lisa Torrey, Jude Shavlik, Trevor Walker e Richard Maclin. «Transfer learning via advice taking». In: *Advances in Machine Learning I*. Springer, 2010, pp. 147–170 (cit. a p. 52).

[34] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled e Fahmy Aly. «Deep learning approaches for data augmentation and classification of breast masses using ultrasound images». In: *Int. J. Adv. Comput. Sci. Appl* 10.5 (2019), pp. 1–11 (cit. a p. 53).

[35] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox e Martin Riedmiller. «Striving for simplicity: The all convolutional net». In: *arXiv preprint arXiv:1412.6806* (2014) (cit. a p. 53).

[36] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee e Kincho H Law. «Automatic localization of casting defects with convolutional neural networks». In: *2017 IEEE international conference on big data (big data)*. IEEE. 2017, pp. 1726–1735 (cit. a p. 53).

Ringraziamenti

Ringrazio l'azienda SynDiag per avermi dato l'opportunità di collaborare nel loro progetto e avermi fatto capire cosa significa lavorare in un team. Ognuno è un tassello di un puzzle e come tale deve saper imparare dagli altri pezzi per conoscere la sua posizione e qui esprimersi al meglio, portando il suo contributo.
Un grazie alla mia famiglia, che nonostante il periodo difficile che sta affrontando è sempre rimasta unita e mi ha dato la forza per concludere questo percorso. A mia madre, che mi ha supportato durante le mie crisi e mi ha saputo perdonare, che ha sacrificato tanto di sé per mettere me e mio fratello al primo posto, spero che un giorno ti possa ripagare per tutto quello che mi hai dato. Al mio Incredibile bro, che mi ha ricordato il bello di stare "nel chilling" ogni tanto nella vita, spero di esserti di supporto nelle tue scelte future e di saperti consigliare al meglio la strada da seguire. A mia nonna, che con i suoi gratinati mi faceva tornare a Torino con il sorriso e con la pancia sempre piena, ti prometto che ci abbracceremo forte di nuovo una volta finita questa emergenza. A mio padre, che avrei voluto qui con me in questo momento così importante della mia vita. In quest'ultimo anno senza di te ho pensato più volte di non farcela, ma non potevo deluderti. Così sono cresciuta e ho continuato a studiare anche quando gli occhi erano troppo appannati per leggere le parole sui libri. Grazie per avermi insegnato a lottare.
Un grazie a Imp, il mio compagno di viaggio che non mi ha mai fatto sentire sola ma parte sempre di un qualcosa. Per avermi fatto vivere questa esperienza con il sorriso e con 200 mA di vitalità.
Grazie alle Allodole e alle Vecchie Amicizie per avermi donato quell'amicizia unica e rara, per cui ogni volta non vedevo l'ora di riprendere il treno per tornare a casa. Ovunque mi porterà la mia vita so che ci ritroveremo sempre per ballare YMCA come solo noi sappiamo fare.