

**UNIVERSITÀ DEGLI STUDI DI  
MILANO BICOCCA**

Dipartimento di Fisica “G.Occhialini”



Tesi di Laurea Triennale in Fisica

**Approcci di deep learning per un task di  
classificazione nell'imaging medicale.**

Relatore:

**Marco Paganoni**

Correlatrice:

**Elisabetta De Bernardi**

Candidato:

**Manenti Federico**

Matr. 790032

Anno Accademico 2017/2018

# Ringraziamenti

Ringrazio Marco Paganoni ed Elisabetta De Bernardi che hanno reso possibile questo lavoro. Elisabetta in particolare mi ha seguito per tutta la durata del lavoro chiarendo tutti i miei dubbi ogni qual'volta ne ho avuto. Ringrazio anche Enrico Panontin che ha cercato di spiegarmi un argomento vastissimo come l'imaging medicale in pochissimo tempo. Un ringraziamento va anche al dottor Valentino Bettinardi e Matteo Bagnalasta per i dati forniti.

I ringraziamenti più sentiti però vanno alla mia famiglia e agli amici, sia in università che non, che mi sono sempre stati vicini e mi hanno dato la forza di continuare.

# Indice

<b>1</b>	<b>Imaging Medicaie</b>	<b>2</b>
1.1	Radiografia . . . . .	2
1.2	Computed Tomography . . . . .	3
1.3	Magnetic Resonance Imaging . . . . .	4
1.4	Positron Emission Tomography . . . . .	5
<b>2</b>	<b>Machine Learning e Deep Learning</b>	<b>7</b>
2.1	Machine Learning . . . . .	7
2.2	Deep learning . . . . .	8
<b>3</b>	<b>Utilizzo del Deep Learning nell’Imaging Medicaie</b>	<b>10</b>
3.1	Classificazione . . . . .	10
3.2	Rivelazione . . . . .	10
3.3	Segmentazione . . . . .	11
<b>4</b>	<b>Descrizione del Task</b>	<b>12</b>
4.1	Preprocessing . . . . .	12
4.2	Descrizione dell’ambiente di lavoro . . . . .	14
<b>5</b>	<b>Lavoro Svolto</b>	<b>16</b>
5.1	Prima prova . . . . .	16
5.2	Seconda prova . . . . .	19
5.3	Terza prova . . . . .	20
5.4	Quarta prova . . . . .	21
5.5	Quinta prova . . . . .	22
5.6	Conclusione ed osservazioni finali . . . . .	23
	<b>Bibliografia</b>	<b>24</b>

# Capitolo 1

## Imaging Medica

L'imaging medica ha un ruolo sempre più importante nelle fasi di diagnosi, definizione del trattamento e monitoraggio della terapia in ambito oncologico, neurologico e cardiologico. Negli ultimi anni stiamo assistendo non solo ad un grosso sviluppo della strumentazione, ma anche ad una forte evoluzione delle tecniche di analisi e quantificazione delle immagini.

L'imaging medica si basa sull'utilizzo di diversi tipi di radiazioni (sia ionizzanti che non) attraverso le quali è possibile osservare un'area di organismo non visibile dall'esterno e quindi ad esempio individuare fratture, tumori, cisti o malformazioni in modo non invasivo.

Esistono diversi tipi di tecniche di imaging, tra cui, radiografia, tomografia computerizzata (CT), risonanza magnetica (MRI) e tomografia ad emissione di positroni (PET).

### 1.1 Radiografia

L'imaging medica ha un'origine più antica di quanto si pensi, infatti la prima tecnica usata è la radiografia che può essere associata alla scoperta dei raggi x da parte di Röntgen nel 1895. Durante i suoi esperimenti si accorse che diversi tipi di materiale e diversi spessori proiettavano un'ombra diversa quando erano attraversati dai raggi da lui scoperti e che poteva fissare questa immagine su una pellicola fotografica. Così chiese alla moglie di tenere ferma la mano sulla lastra e dopo qualche tempo ottenne la prima foto analogica ricavata da una tecnica di imaging (radiografia) della storia.



Figura 1.1: Mano della moglie di Röntgen

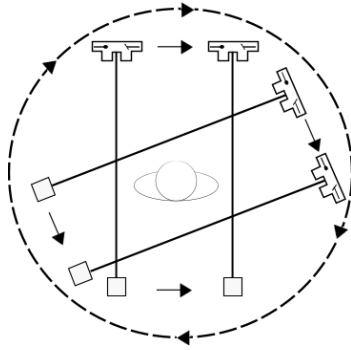
La radiografia si basa quindi sull'invio di raggi X al paziente. I raggi X vengono attenuati dai tessuti in base alla densità e al numero atomico equivalente dei tessuti stessi. Misurando i raggi X a valle del paziente si riescono a ricavare informazioni circa i tessuti attraversati. La radiografia è un'immagine bidimensionale planare che rappresenta tale informazione su un piano di proiezione (antero-posteriore o latero-laterale). I tessuti maggiormente attenuanti (iperdensi), come le ossa, appaiono bianchi sull'immagine radiografica, mentre i tessuti meno attenuanti (ipodensi), come il tessuto polmonare, appaiono scuri. La radiografia è utilizzata ad esempio per lo studio delle fratture, per la ricerca di patologie a livello polmonare e per la ricerca di iperdensità patologiche nella mammella (mammografia).

## 1.2 Computed Tomography

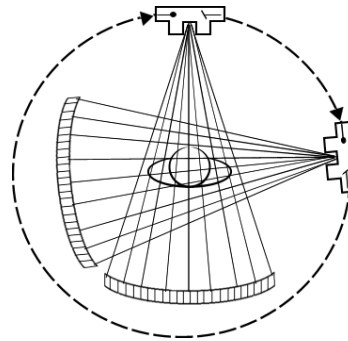
La tomografia computerizzata fu inventata nel 1967 da Godfrey Hounsfield, un ingegnere britannico, e da Allan Cormak, un fisico sudafricano. Tale studio valse il premio Nobel per la medicina nel 1979 ai due scienziati. La prima CT fu installata nel 1971 ed era formata da un'unità di scansione che traslava e ruotava di  $180^\circ$  attorno al paziente immagazzinando 160 immagini ad ogni grado. L'apparecchiatura aveva due grandi limiti:

1. Il tempo, infatti servivano dai 5 ai 10 minuti per raccogliere i dati e circa due ore e mezza per ricostruire le immagini.
2. La scansione poteva essere fatta solo al cervello.

Con l'avanzare degli anni la tecnica venne affinata e il tempo sia di esposizione del paziente che di ricostruzione dati calò drasticamente; nel 1975 Ralph Alfidi riuscì ad utilizzare per la prima volta la CT per scansionare il torace di un uomo. Da questo momento la tecnologia continuò a migliorare.



(a) Schema CT di prima generazione



(b) Schema CT moderne

Le CT moderne sono costituite da una sorgente che emette raggi x con un'ampiezza tra i  $30^\circ$  e  $50^\circ$  e da rilevatori posti dalla parte opposta rispetto al paziente che ruotano attorno ad esso per raccogliere dati in tutte le direzioni. Il paziente posto sopra il lettino scorre in maniera precisa all'interno del macchinario per poter scansionare solo la zona d'interesse. Sorgente di raggi X e rilevatori compiono una rotazione completa in circa un secondo e possono scansionare tutto il corpo in meno di un minuto. Il dato acquisito è quindi costituito da tante proiezioni radiografiche dello stesso oggetto da più angolazioni. A partire da questi dati, mediante algoritmi di ricostruzione tomografica da proiezioni, come ad esempio la *FilteredBackProjection* (FBP) che sfrutta la trasformata di Radon, si ottengono le immagini TC, ossia un set di immagini tomografiche corrispondenti a sezioni assiali virtuali di paziente che rappresentano in 3D la distribuzione di densità dei tessuti e quindi l'anatomia del paziente.

### 1.3 Magnetic Resonance Imaging

La scoperta del principio fisico alla base della MRI (NMR, nuclear magnetic resonance) avvenuta nel 1944 da parte di Felix Bloch ed Edward Purcell precede di circa 40 anni l'introduzione della risonanza magnetica come tecnica di imaging medico. Il fenomeno scoperto all'inizio ha trovato un'applicazione in chimica e biochimica, con una tecnica chiamata spettroscopia a risonanza magnetica. Ogni nucleo ha infatti una frequenza di risonanza diversa dagli altri a seconda della molecola e della posizione che ha al suo interno. Elaborando i dati ottenuti è quindi possibile ricavare uno spettro delle molecole presenti all'interno del campione analizzato.

L'imaging a risonanza magnetica utilizza un campo magnetico di intensità costante (nei macchinari utilizzati in ospedale il campo magnetico varia dai 1,5 ai 3 Tesla, mentre sono in via di sviluppo strumenti che raggiungono fino a 17 Tesla per esperimenti su animali) che serve ad allineare gli spin dei protoni. Vi sono poi delle bobine che forniscono al paziente campi più deboli oscillanti alla frequenza di risonanza dei nuclei (radiofrequenze). Dopo l'impulso di radiofrequenza, i nuclei rilasciano l'energia acquisita in un processo di rilassamento. Grazie a delle bobine di ricezione è possibile misurare l'andamento della magnetizzazione durante il processo di rilassamento. All'interno di ciascun tessuto il rilassamento avviene con due costanti di tempo tessuto-dipendenti chiamate T1 e T2.

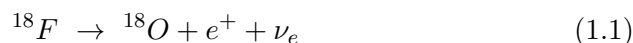
Nell'imaging medico questa caratteristica è sfruttata per creare due serie di immagini diverse pesate in T1 (utilizzata per tessuti molli; le ossa e i liquidi non danno segnale per cui risultano neri) e in T2 (immagini più chiare dove vi è una maggiore concentrazione di liquido). A differenza delle altre tecniche di imaging l'MRI è considerata totalmente sicura perchè non utilizza radiazioni ionizzanti, per questo motivo; spesso è preferita alla CT soprattutto in pazienti come donne incinte o bambini. Anche per gli operatori è più sicuro in quanto non sono stati dimostrati danni per la salute derivanti dalla permanenza vicino a campi magnetici di intensità elevata.

## 1.4 Positron Emission Tomography

La prima tomografia a emissione di positroni fu costruita nel 1952 da William Sweet e Gordon L. Brownell: il principio fisico alla base della PET è il decadimento  $\beta^+$  da parte di isotopi radioattivi iniettati al paziente e la conseguente annichilazione elettrone positrone con emissione di raggi gamma. Le differenze principali rispetto alle altre tecniche di imaging sono:

1. La sorgente di energia non è più esterna, ma è un radionuclide immesso all'interno del paziente (ad esempio Fluorodesossiglucosio  $^{18}F - FDG$ ).
2. L'immagine finale contiene un'informazione fisiologica e non morfologica.

Lo studio PET con  $^{18}F - FDG$  ad esempio consente di individuare le aree con metabolismo particolarmente elevato, tra cui ad esempio alcuni tumori e metastasi. Nel decadimento  $\beta^+$  uno dei protoni dell'atomo iniziale viene convertito in un neutrone emettendo un positrone e un neutrino elettronico. Nella PET l'isotopo radioattivo più utilizzato è il  $^{18}F$  che decade come:



I positroni emessi hanno una vita media molto breve e infatti, dopo aver perso abbastanza energia, interagiscono con un elettrone creando uno sta-

to legato chiamato positronio. Quando le particelle sono quasi a riposo si innesca un decadimento del tipo:

$$e^+ + e^- \rightarrow \gamma\gamma \quad (1.2)$$

chiamato annichilazione elettrone positrone. I due fotoni emessi hanno energia di 511 *KeV* con stessa direzione e verso opposto (*backtoback*). Questa direzione è chiamata *lineofresponse* (LOR). I due fotoni emessi hanno abbastanza energia per uscire dal corpo del paziente e vengono raccolti da una serie di rilevatori posizionati intorno a tutto il paziente (scanner PET). Lo scanner ha il compito di rilevare le coppie di fotoni provenienti dallo stesso decadimento (utilizzando un intervallo di tempo detto finestra di coincidenza) e individuare la corrispondente LOR. I dati, dopo essere stati raccolti, vengono elaborati da un computer al fine di ottenere immagini tomografiche della distribuzione del radioisotopo nel paziente. Tra gli algoritmi di ricostruzione vi sono la *FilteredBackProjection*, e metodi statistici a massima verosimiglianza *MLEM*.



## Capitolo 2

# Machine Learning e Deep Learning

### 2.1 Machine Learning

Il Machine Learning è un campo dell'Intelligenza artificiale che usa tecniche statistiche per dare al computer l'abilità di "imparare" dai dati, ossia di migliorare progressivamente le performance su un determinato compito, senza però essere state esplicitamente e preventivamente programmate.

Solitamente il Machine Learning viene diviso in due categorie: predittivo o con apprendimento supervisionato (*supervised learning*) e descrittivo o con apprendimento non supervisionato.

Il *supervised learning* ha l'obiettivo di predire il particolare valore di una variabile sulla base del valore di altri attributi. Al computer vengono dati come input sia dei set di dati sia le informazioni relative ai risultati desiderati con lo scopo che il sistema individui una regola generale che colleghi l'input con l'output, in modo da poter poi riutilizzare la stessa regola su dati simili. Uno dei task principali di questa categoria è la classificazione dei dati.

Lo scopo dell'*unsupervised learning* è invece quello di trovare una correlazione che associ i dati ricevuti in input. In questo caso, al computer, vengono forniti solo set di dati senza indicazioni sul risultato desiderato. Solitamente queste tecniche hanno bisogno di un lavoro di postprocessing per interpretare e spiegare i risultati ottenuti.

Esiste poi anche una terza categoria chiamata apprendimento per rinforzo (*reinforcement learning*) che sfrutta una routine basata su ricompense e punizioni (errori) e che si applica principalmente alla guida autonoma o nell'intelligenza artificiale dei videogiochi.

## 2.2 Deep learning

Il Deep Learning è una branca del Machine Learning che si basa sull'apprendimento progressivo di rappresentazioni sempre migliori su strati (*layers*) successivi. È definito come una classe di algoritmi di apprendimento automatico che usano diversi livelli di unità non lineari a cascata per estrazione e trasformazione di dati. Ogni *layer* usa come input l'output del precedente. Le tecniche di deep learning sfruttano sia l'apprendimento supervisionato che non supervisionato.

Alla base della maggior parte dei metodi di deep learning ci sono le reti neurali (*Neural Networks*). La struttura delle *NN* ricorda vagamente quella di una rete neurale biologica, da qui deriva il nome. Le *Neural Networks* sono formate da neuroni o unità con funzioni di attivazione e parametri di peso degli ingressi e di bias. I neuroni di ogni strato sono collegati a quello del layer successivo e le informazioni possono scorrere solo in quella direzione senza mai tornare indietro; l'ultimo strato solitamente ha una funzione di attivazione chiamata *softmax* o *sigmoid*. Se alcuni layers sono "nascosti" la rete neurale prende il nome di *Deep Neural Network*.

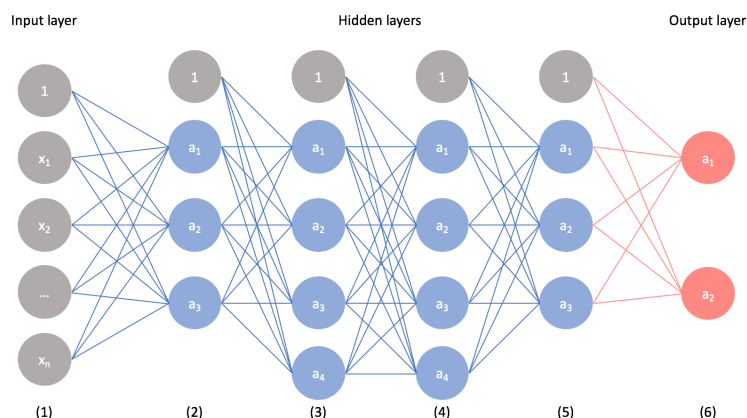


Figura 2.1: Schema di una Deep Neural Network

Ma come funziona una rete neurale? All'inizio dell'esecuzione ad ogni neurone vengono assegnati valori casuali di pesi e bias. Alla rete vengono forniti in ingresso dei dati di training; i layers lavorano sui dati e danno un output predittivo che è paragonato con il target o output desiderato mediante una loss function che calcola un punteggio che rappresenta la distanza tra target predetti e reali. L'abilità del deep learning sta nell'utilizzare questo punteggio come feedback per aggiustare il valore dei pesi dei vari neuroni in modo da abbassare il loss score. Questa operazione è reiterata più volte fino ad ottenere i loss score minimo e quindi l'output desiderato.

Uno dei problemi che affligge il deep learning è l'overfitting, ossia il fatto che la rete, usando troppe reiterazioni su pochi dati può apprendere caratteristiche specifiche del dataset di training e riconoscere pattern anche dove non sono presenti. Per evitare ciò ci sono diverse tecniche; la più banale da utilizzare consiste nell'arrestare l'addestramento all'ultima iterazione in cui il loss score cala in maniera drastica, oppure affiancare al dataset di training un dataset di validazione e arrestare l'addestramento quando il loss score sul dataset di validazione inizia a salire.

## Capitolo 3

# Utilizzo del Deep Learning nell'Imaging Medicale

Le tecniche di machine learning di deep learning sono sempre più utilizzate nell'ambito dell'imaging medico. I tre principali task in cui il deep learning (DL) è stato applicato nell'ambito delle immagini medicali sono la classificazione, la rilevazione di anomalie e la segmentazione.

### 3.1 Classificazione

La classificazione di esami e immagini è una delle prime aree in cui il Deep Learning è stato utilizzato. Solitamente nella classificazione di immagini si ha come input un insieme di immagini e come output una classificazione in due o più classi. Ci si può ricondurre a task di classificazione ad esempio nella diagnosi differenziale di patologia maligna vs benigna oppure nella discriminazione tra più forme patologiche, ad esempio tumorali. Un task di classificazione di immagini medicali può essere affrontato in due modi:

1. Mediante reti pre-trainate (tipicamente su immagini naturali) di cui poi vengono modificati alcuni parametri per applicarla al meglio su immagini mediche.
2. Oppure, più che altro nel caso di piccoli volumi di interesse come ad esempio per la diagnosi di noduli, si possono utilizzare CNN create appositamente, completamente addestrate sul dataset medicale

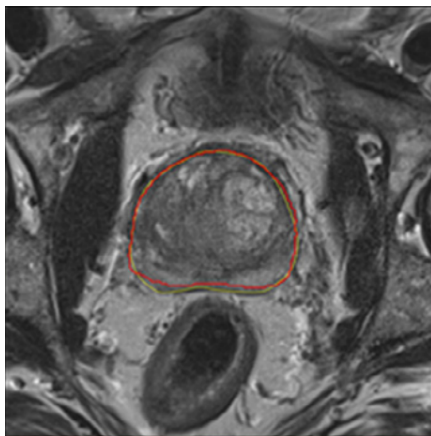
### 3.2 Rivelazione

La rivelazione di organi e punti di riferimento specifici all'interno del corpo umano sono stati un importante precursore della segmentazione di immagini mediche. Per questo task si possono utilizzare sia una CNN che una

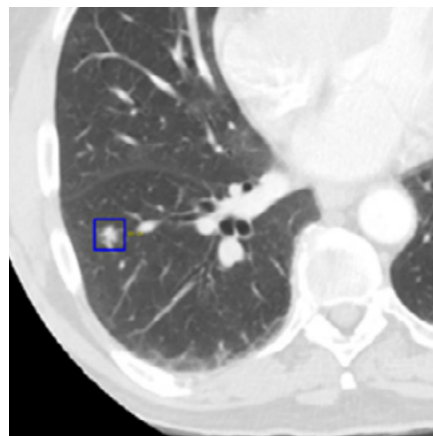
Restricted Boltzman Machine (RBM). Solitamente questo campo si applica alla ricerca di cellule particolari, come quelle tumorali, oppure per localizzare regioni di corpo che sono sospettate di avere un problema. Lo scopo è aumentare il rateo di rilevazione di zone malate e ridurre i falsi positivi.

### 3.3 Segmentazione

La segmentazione di un'immagine nell'elaborazione digitale è il processo di partizione di un'immagine in regioni significative. La segmentazione ha tantissime applicazioni nell'ambito dell'imaging medicale, come ad esempio la rimozione della scatola cranica in studi cerebrali, o il contornamento dei tumori e degli organi ad essi circostanti. Il contornamento di tumori e organi in ambito clinico spesso viene eseguito manualmente da operatori esperti. Trasformarlo in un processo automatizzato consentirebbe di ridurre la variabilità inter-operatore e i tempi di contornamento. Le reti neurali convoluzionali addestrate su una grande quantità di immagini pre-contornate sono molto efficaci nel task di segmentazione.



(a) Segmentazione di una prostata



(b) Rilevazione di un nodulo

## Capitolo 4

# Descrizione del Task

Il task che verrà presentato è un classico del Deep Learning ossia la classificazione di oggetti (in questo caso immagini medicali del distretto cerebrale) in categorie differenti. I dati utilizzati derivano da 3 studi CT, 4 studi PET e 2 studi MRI con dimensione dei voxel diverse:  $(1.95 \times 1.95 \times 3.27) \text{ mm}^3$  per PET,  $(0.5 \times 0.5 \times 0.5) \text{ mm}^3$  per MRI e  $(0.5 \times 0.5 \times 2.5) \text{ mm}^3$  per la CT. Il dataset utilizzato è composto complessivamente da 150 immagini CT, 365 immagini MRI e 124 immagini PET.

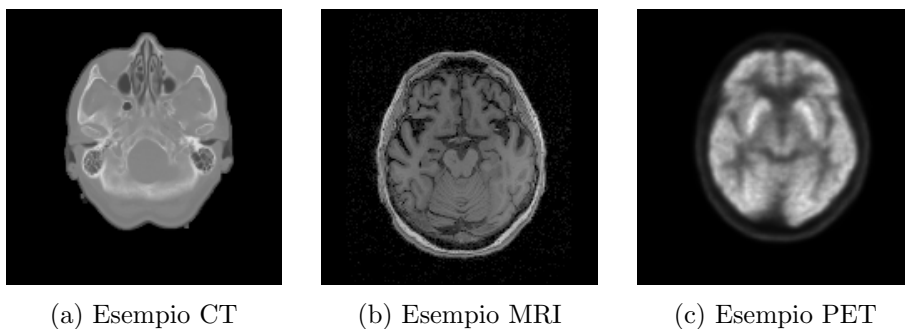


Figura 4.1: Tre immagini prese dal dataset: una per ogni categoria.

### 4.1 Preprocessing

Le immagini a disposizione sono caratterizzate da una dimensione del voxel e da un range di intensità dipendenti dalla tecnica di imaging con cui sono state ottenute. L'obiettivo del preprocessing è quello di elaborare le immagini in modo da uniformare la dimensione del voxel e i range di intensità, allo scopo di fare apprendere alla rete una classificazione basata sul contenuto informativo delle immagini e non sul range dei valori di intensità o sulla dimensione del cranio sulla matrice immagine.

Le foto di ciascuno studio sono state quindi innanzitutto reinterpolate sulla

medesima griglia di voxel ( $1.5 \times 1.5 \times 1.5$ )  $mm^3$  per mantenerle tutte nello stesso sistema di riferimento. A valle della reinterpolazione il dataset risulta costituito da 643 immagini: 259 CT, 121 MRI e 263 PET.

Sulle immagini derivanti dalla CT, a differenza degli altri esami, risulta visibile anche il lettino su cui è posto il paziente. Le CNN sono incredibilmente abili a rilevare pattern per cui il lettino sarebbe stato una differenza e aiuto troppo grande per distinguere le immagini CT. Quindi per complicare il lavoro al programma è stato rimosso via software il lettino.

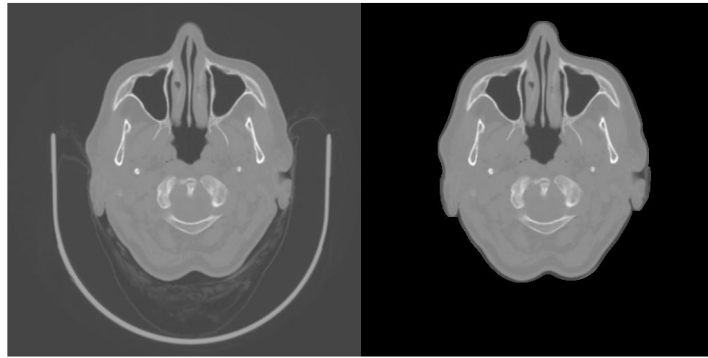


Figura 4.2: Differenza immagine CT con e senza lettino

Successivamente, il range di intensità dei valori dei voxel di ciascuno studio è stato normalizzato tra 0 e 1, per CT e PET in maniera lineare mentre per la MRI è stata applicata una trasformazione non lineare dei livelli di grigio su ciascuno dei due volumi. Tutto ciò per avere la scala di grigi di ogni esame più simile possibile. Fatto ciò da ciascuna immagine è stata ricavata un'immagine png di 166x166 pixel.

Le immagini provenienti da ciascuna tecnica sono state ordinate casualmente, senza più seguire l'ordine di registrazione. Ciò per evitare che il programma impari a classificare semplicemente per il fatto che le immagini siano processate in ordine di categoria.

Il dataset è stato diviso in tre parti: *training*, *validation* e *test set* con il primo gruppo contenente un numero maggiore di elementi rispetto a gli altri a causa dei differenti compiti che possiedono.

1. Il *training set* (336 immagini) è l'insieme dei dati che sono utilizzati per addestrare la rete confrontando l'output generato dalla rete a fronte di un certo input con l'output atteso. I pesi dei singoli neuroni vengono gradualmente modificati andando a minimizzare la differenza tra tali output.

2. Il *validation set* (144 immagini) contiene i dati utilizzati per regolare i parametri del classificatore (ad esempio il numero di neuroni o a che punto arrestare le iterazioni).
3. Il *test set* (163 immagini) sono i dati utilizzati per valutare le performance della rete al termine dell'addestramento.

Come primo passo il modello viene allenato sul training set e contemporaneamente valutato sul validation set, infine quando il modello è pronto viene testato un'ultima volta sul test set. Ma perché avere tre set e non due, ossia allenare sul training set e valutare sul test set? La ragione per cui non si fa è che il programma impara (aggiusta i pesi dei singoli neuroni) grazie ai feedback che riceve sul training set. Gradualmente quindi il modello si adatta al training set. Il nostro obiettivo però non è quello di generare un modello di classificazione che funzioni perfettamente su questo set, ma un modello che apprenda dal training set le regole necessarie per classificare anche dati non ancora visti. Durante le prime fasi dell'addestramento le performance del modello miglioreranno sempre più, sia sul dataset di training che su quello di validazione. Da un certo punto in poi invece le performance continueranno a migliorare sul dataset di training ma inizieranno a peggiorare su quello di validazione. Questo è il punto in cui il modello inizia ad andare in overfitting sulle caratteristiche del dataset di training, e quindi il momento in cui l'addestramento deve essere fermato per ottenere un modello sufficientemente generale. Le performance ottenute sul validation set non possono essere tuttavia considerate come le performance del modello; l'accuratezza sul validation set sarà infatti maggiore rispetto a quella che si otterrebbe su un dataset indipendente perché tale set è utilizzato per scegliere il modello finale. Per questo motivo le performance del modello devono essere valutate su un ulteriore dataset, ossia sul dataset di testing.

## 4.2 Descrizione dell'ambiente di lavoro

Il programma è stato scritto in linguaggio *Python* nella versione 3.6.5 utilizzando *Jupyter Notebook* come editor su una macchina con sistema operativo Manjaro: una distro di Linux. Le principali funzioni utilizzate sono importate da *Keras*, *TensorFlow* e *NumPy*.

Keras è un API open-source di alto livello, sviluppato da Google nel 2015, per reti neurali scritto in Python che lavora al suo meglio con TensorFlow come *backend engine*. Keras è uno degli API per Neural Network più user friendly esistenti: si può pensare ai layers come i blocchi Lego del Deep Learning. Sviluppare un modello di Deep Learning in Keras significa quindi unire layers compatibili tra di loro per creare trasformazioni di dati utili allo scopo. Il concetto di compatibilità tra layer significa che ogni strato accetta



un tensore di una certa dimensione e restituirà un tensore di un'altra specifica dimensione.

TensorFlow è una libreria di funzioni (in particolare manipolazione di tensori) open-source anch'essa sviluppata da Google nel 2015 per calcolo numerico ad alte prestazioni, e studiata prettamente per il Machine Learning. Per lavorare può utilizzare sia CPU che GPU; nel mio caso, non avendo a disposizione una GPU di ultima generazione ed essendo il task non oneroso, ho deciso di utilizzare una CPU.

NumPy è un pacchetto fondamentale per il calcolo scientifico in Python, il framework che verrà utilizzato è *pyplot*: un insieme di funzioni per generare e lavorare su grafici.

## Capitolo 5

# Lavoro Svolto

Come primo passo il dataset è stato suddiviso in due cartelle: una contenente training e validation set mentre l'altra il test set. La rete neurale non può ricevere in input le immagini in formato grafico .png, quindi sono state applicate delle trasformazioni per renderle utilizzabili da TensorFlow. Per iniziare i dati sono stati trasformati in una lista di due entrate. La prima è un tensore a 3 dimensioni (166, 166, 1), dove 166\*166 è la dimensioni dell'immagine e l'1 sulla terza dimensione è il canale del colore (in questo caso significa che le immagini sono in scala di grigio). Quello che otteniamo quindi è una matrice 166\*166 dove ogni numero rappresenta il valore sulla scala di grigi del pixel corrispondente. La seconda entrata invece è un numero (0,1,2) che identifica la classe a cui l'immagine appartiene(CT, MRI o PET). La lista è stata successivamente divisa in due vettori, il primo contenente solo le informazioni sulla matrice immagine, mentre il secondo il tipo di esame.

### 5.1 Prima prova

Preparati i dati, è stata scelta la rete neurale da utilizzare. Essendo il task semplice e dovendo la rete funzionare velocemente su CPU, è stata inizialmente costruita una rete così strutturata:

```
model = models.Sequential()
model.add(layers.Conv2D(16, (3, 3), activation='relu',
                        input_shape=(166,166,1)))
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
model.add(layers.Conv2D(16, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(3, activation='sigmoid'))
```

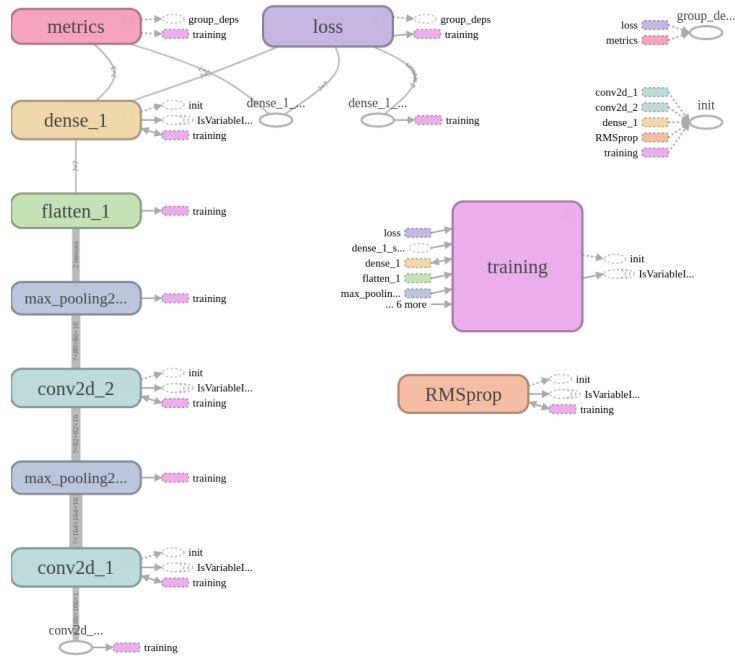


Figura 5.1: Schema rete neurale

La rete è costituita da 2 strati *convoluzionali* di 16 filtri ciascuno, seguiti da strati di *Max Pooling*. Gli strati convoluzionali filtrano l'immagine mediante una matrice kernel i cui pesi vengono appresi durante la fase di addestramento. L'uscita di ciascun filtro è una feature map, che contiene le informazioni estratte dal filtro. Gli strati di *Max Pooling* servono per sottocampionare l'immagine di ingresso estraendo da essa le caratteristiche più importanti, in modo da consentire al successivo strato di convoluzione di estrarre informazioni su un campo recettivo più ampio. Nel nostro caso la dimensione del kernel di *Max Pooling* è 2x2. Questo significa che all'interno di ciascuna area 2x2 sull'immagine viene estratto il valore massimo.

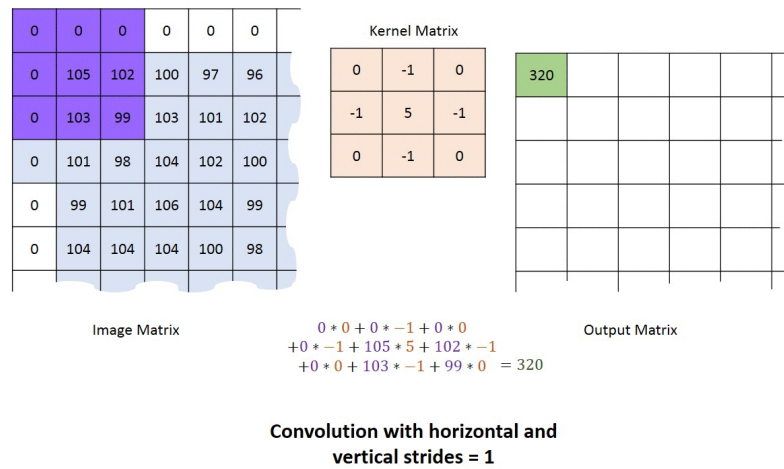


Figura 5.2: Convolutional 2D layer

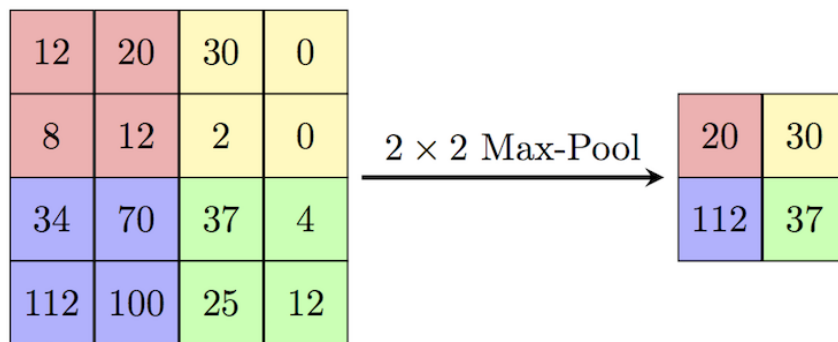
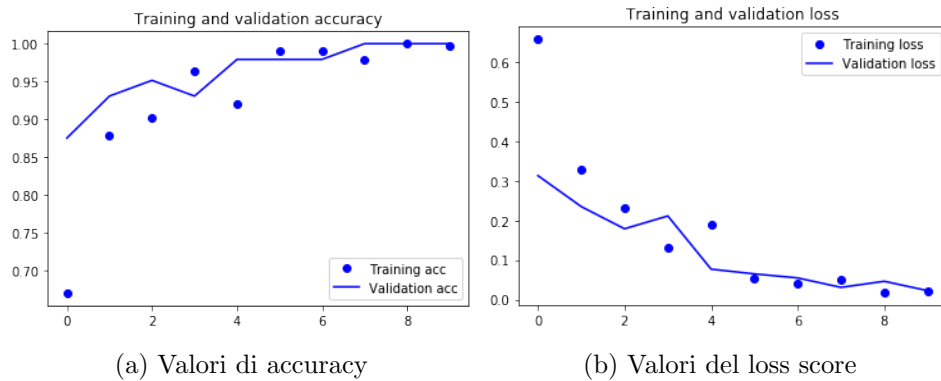


Figura 5.3: Max Pooling layer

Lo strato *flatten* semplicemente trasforma un array di più dimensioni in un vettore. Infine il *Dense layer* è uno strato fully connected che assegna a ciascuna immagine la probabilità di appartenere a ciascuna delle 3 classi. Dopo aver definito l'architettura della rete, il modello è stato compilato utilizzando come loss function la *sparse categorical crossentropy*, che è considerata la migliore per calcolare il loss score quando le classi in uscita sono numeri interi (0,1,2). Come ottimizzatore si è scelto di utilizzare l'*RMSprop*.

Creata la rete è stata quindi addestrata con il dataset di training.

I grafici successivamente rappresentati mostrano il valore dell'accuratezza o loss score in funzione delle epoche utilizzate.



Il modello ottiene un'accuratezza di 1 e un loss score molto basso senza overfit, ciò significa che il compito è svolto molto bene. Il modello è successivamente stato valutato sul dataset di test, ottenendo come valori:

- TEST LOSS: 0.029
- TEST ACCURACY: 0.994

Ciò conferma il buon funzionamento dell'architettura.

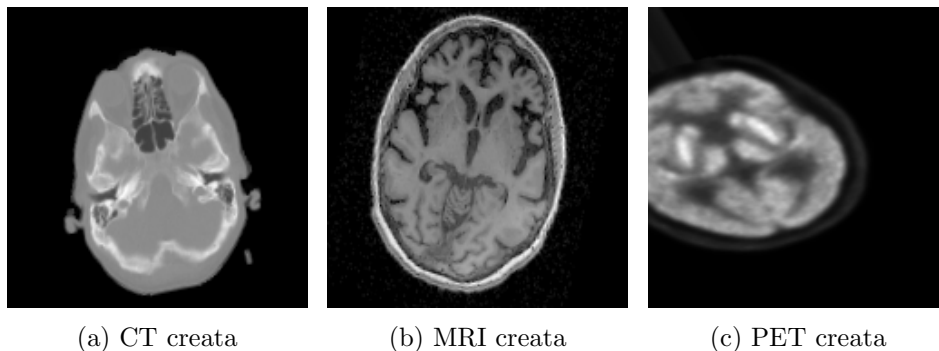
## 5.2 Seconda prova

La rete progettata funziona quindi molto bene per il task assegnato. Ricontrrollando però accuratamente le immagini si può notare che non sono tutte perfettamente centrate: per ogni categoria di esame la matrice cerebrale ha una posizione leggermente differente. Questo è verosimilmente legato alle impostazioni del campo di vista degli scanner. Il dubbio lecito che può venire è che la rete utilizzi la posizione della testa come informazione principale per l'apprendimento piuttosto che le informazioni morfologiche contenute nelle immagini.

Per verificare questa ipotesi, ho deciso di applicare una data augmentation al dataset di test e di valutare il modello addestrato sulle nuove immagini create. Per ciascuna immagine del dataset di test ho creato 10 nuove immagini, utilizzando questi parametri:

```
datagen = ImageDataGenerator(
    rotation_range=45,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```

*Rotation\_range* indica il range di gradi di cui le immagini possono essere ruotate casualmente (da 0 a 45); *width* e *height\_shift\_range* sono il range entro cui le immagini possono essere casualmente shiftate lungo gli assi x e y; *shear\_range* e *zoom\_range* sono il range entro cui l'immagine può essere tagliata e dilatata, mentre *horizontal\_flip* = True significa che l'immagine può essere ribaltata.



Fatto ciò ho ripetuto il procedimento per rendere le immagini processabili dal modello e l'ho valutato ottenendo i valori:

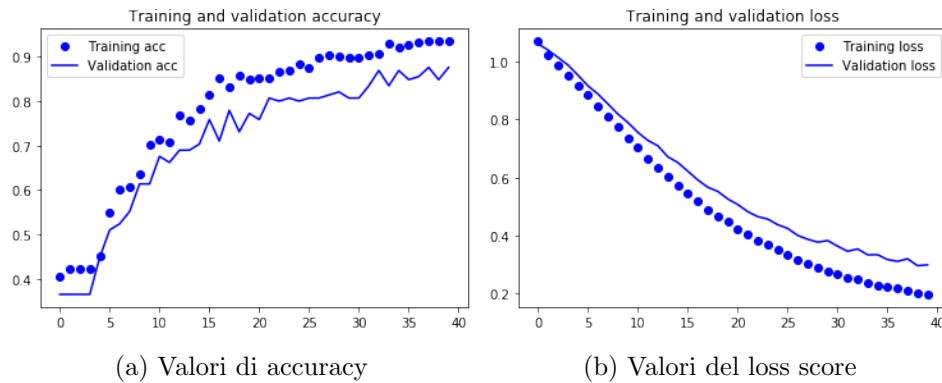
- TEST LOSS: 1.657
- TEST ACCURACY: 0.582

L'intuizione quindi era corretta: il modello allenato sui dati originali impara a classificare le immagini principalmente sulla base della posizione del cervello.

### 5.3 Terza prova

Il passo successivo è stato quindi quello di cercare di rendere il modello più generale possibile, ossia di fargli apprendere la classificazione delle immagini sulla base del contenuto informativo delle stesse, svincolandosi dalle informazioni spaziali. Per fare ciò, ho applicato la data augmentation anche alle immagini dei set di training e validation, ottenendo 10 nuove immagini per ciascuna immagine di partenza. Per poi lavorare su un dataset di dimensioni pari a quelle del dataset iniziale, ho selezionato casualmente 480 immagini dal dataset aumentato.

Ho utilizzato la medesima architettura di rete utilizzata nelle prime due prove, ma per la fase di addestramento ho ridotto il learning rate da 0.001 a 0.0001. Nella fase di addestramento risultati soddisfacenti sono stati raggiunti in 40 epoche.



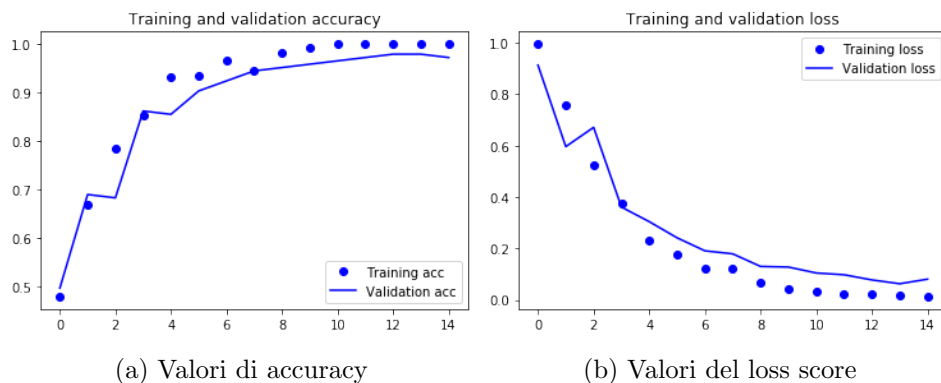
La rete è stata poi valutata sul dataset di testing ottenendo:

- TEST LOSS: 0.323
- TEST ACCURACY 0.855

I valori di accuratezza ottenuti sul validation e sul test set sono circa 0.85, molto lontani dal valore 1 ottenuto nelle prove precedenti. Questo significa che la rete utilizzata è troppo semplice per il task di classificazione, che dopo avere introdotto la data augmentation è diventato più complesso.

## 5.4 Quarta prova

Per migliorare l'accuratezza, ho provato innanzitutto a mantenere inalterata l'architettura della rete e a modificare l'ottimizzatore, passando dallo standard *RMSprop* ad *ADAM*, che risulta in genere molto più performante. I valori di accuratezza e loss score ottenuti in funzione delle epoche da 1 a 15 sono i seguenti.

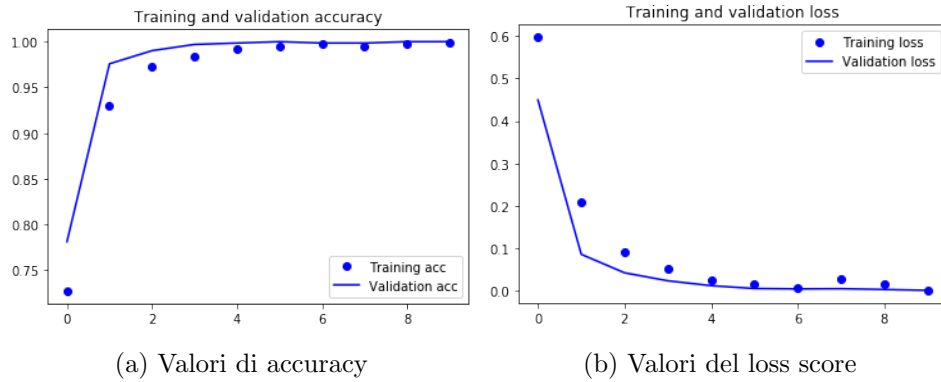


Come si può notare l'accuratezza va quasi ad 1 quindi con questa modifica il modello funziona bene. Anche la valutazione del modello lo conferma:

- TEST LOSS: 0.140
- ACCURACY: 0.945

## 5.5 Quinta prova

Come ultimo studio ho provato ad utilizzare la solita rete con ottimizzatore RMSprop, ma addestrandola sull'intero dataset di immagini creato con la data augmentation, ossia su 4431 immagini (3101 per il train e 1330 per la validazione). Il risultato è stato poi testato sulle 1599 immagini del dataset di test aumentato.



Come si può notare il loss score scende velocemente a circa 0 e l'accuratezza sale velocemente a quasi 1, sia sul set di training che su quello di validazione. Anche in questo caso, come nelle precedenti prove, non è presente overfitting, ossia la rete si comporta allo stesso modo sul dataset di training e su quello di validazione. Anche sul set di test si ottengono:

- TEST LOSS: 0.004
- TEST ACCURACY: 0.999

Questo comportamento è verosimilmente dovuto al fatto che le immagini di ciascuna classe sono molto simili tra loro nei 3 set di dati, in quanto provenienti da pochi studi eseguiti sullo stesso scanner e con gli stessi parametri. I risultati ottenuti sul set di training si ritrovano quindi quasi identici sugli altri due set.



## 5.6 Conclusione ed osservazioni finali

Lo scopo del lavoro è stato quello di approfondire le potenzialità delle tecniche di deep learning nell'ambito dell'imaging medicale e di provare ad affrontare un task di classificazione di immagini mediante queste tecniche. Ho quindi approfondito le reti neurali convoluzionali, approfondito le mie conoscenze di Python e imparato ad utilizzare Keras e i suoi layers per la costruzione delle reti, dei dati in ingresso, per la data augmentation e per l'analisi delle performance. Il task assegnatomi era molto semplice e questo mi ha consentito di operare su una CPU in poco tempo. Ho implementato una rete in grado di classificare correttamente le immagini. Non sono riuscito ad valutare effetti di overfitting durante l'addestramento a causa dell'eccessiva somiglianza tra i dati a mia disposizione.

# Bibliografia

- [1] François Chollet. *Deep learning with Python*. Manning, 2018.
- [2] *Documentazione Keras*. 2015. URL: <https://keras.io>.
- [3] *Documentazione Python*. URL: <https://www.python.org>.
- [4] *Documentazione TensorFlow*. 2015. URL: <https://www.tensorflow.org>.
- [5] Hayit Greenspan; Bram van Ginneken; Ronald M. Summers. «Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique». In: *IEEE Transactions on Medical Imaging* 35 (2016).
- [6] Alex Krizhevsky; Ilya Sutskever; Geoffrey E. Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». In: *Neural Information Processing Systems Conference* (2012).
- [7] Geert Litjens; Thijs Kooi; Babak Ehteshami Bejnordi; Arnaud Arindra Adiyoso Setio; Francesco Ciompi; Mohsen Ghafoorian; Jeroen A.W.M. van der Laak; Bram van Ginneken; Clara I. Sánchez. «A survey on deep learning in medical image analysis». In: *Elsevier* (2017).
- [8] A. Del Guerra; N.Belcari; M.Bisogni. «Positron Emission Tomography: Its 65 years». In: *Rivista del Nuovo Cimento* 39.4 (2016).
- [9] Michael A. Nielsen. *Neural networks and deep learning*. Determination Press, 2015.
- [10] Dinggang Shen; Guorong Wu; Heung-Il Suk. «Deep Learning in Medical Image Analysis». In: *Annual Review of Biomedical Engineering* 19 (2017).