

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Metodiche di Deep Learning e applicazioni all'imaging medico: la Radiomica

Relatore:

Prof. Gastone Castellani

Presentata da:

Rita Maestri

Anno Accademico 2016/2017

Sommario

Questa tesi ha lo scopo di presentare il deep learning e una delle sue applicazioni che ha avuto molto successo nell'analisi delle immagini: la rete neurale convoluzionale. In particolare, si espongono i vantaggi, gli svantaggi e i risultati ottenuti nell'applicazione delle reti convoluzionali alla radiomica, una nuova disciplina che prevede l'estrazione di un elevato numero di feature dalle immagini mediche per elaborare modelli di supporto a diagnosi e prognosi.

Nel primo capitolo si introducono concetti di machine learning utili per comprendere gli algoritmi di apprendimento usati anche nel deep learning. Poi sono presentate le reti neurali, ovvero le strutture su cui si basano gli algoritmi di deep learning. Infine, viene spiegato il funzionamento e gli utilizzi delle reti neurali convoluzionali.

Nel secondo capitolo si espongono le tecniche e gli utilizzi della radiomica e, infine, i vantaggi di usare le reti neurali convoluzionali in quest'ambito, presentando alcuni recenti studi portati a termine in merito.

Indice

1	Introduzione al Deep Learning	3
1.1	Machine Learning	3
1.1.1	Il compito	4
1.1.2	L'esperienza	4
1.1.3	Misura della performance	5
1.1.4	Metodo di ottimizzazione: la discesa del gradiente	5
1.1.5	Esempio di addestramento: la regressione lineare	6
1.1.6	Underfitting e overfitting	7
1.1.7	Evitare l'overfitting: le tecniche più comuni	8
1.2	Il Deep Learning	9
1.2.1	Le reti neurali	10
1.2.2	Reti neurali profonde: il multilayer perceptron	15
1.2.3	Addestramento delle reti neurali profonde	16
1.3	Reti neurali convoluzionali	17
1.3.1	Struttura delle CNN	18
1.3.2	Addestramento delle CNN	23
1.3.3	Applicazioni delle CNN	24
2	La Radiomica	27
2.1	Cos'è la radiomica	27
2.1.1	Il processo radiomico	28
2.1.2	Successi della radiomica	32
2.1.3	I problemi attuali della radiomica	33
2.2	Le reti convoluzionali applicate alla radiomica	35

2.3 Conclusioni	41
Bibliografia	41

Capitolo 1

Introduzione al Deep Learning

1.1 Machine Learning

I computer sono in grado di risolvere in pochissimo tempo problemi di natura matematica che un essere umano affronterebbe con difficoltà. Ma allo stesso tempo, compiti che noi portiamo a termine ogni giorno in modo naturale e con poca difficoltà, sono per un computer una vera sfida. Alcuni esempi sono il riconoscimento visivo degli oggetti o quello delle parole.

La programmazione a codifica fissa si rivela poco efficace per affrontare questo tipo di problemi. Perciò si ricorre ad algoritmi che simulano il modo di conoscere il mondo che usa l'essere umano: l'apprendimento. Metodi che permettono al computer di apprendere senza essere esplicitamente programmato sono detti di *machine learning*, o di apprendimento automatico.

La definizione più appropriata di apprendimento nell'ambito informatico è stata data da Tom Mitchell: **"Si può dire che un programma impara da un'esperienza E rispetto a una certa classe di compiti T e a una misura della performance P, se la sua performance nel compito T, misurata tramite P, migliora con l'esperienza E"**.

In seguito presentiamo diversi tipi di compiti, misure della performance ed esperienze che possono essere usate per costruire un algoritmo di machine learning.

1.1.1 Il compito

Come già detto, l'importanza degli algoritmi di machine learning sta nella loro capacità di risolvere compiti facili per gli esseri umani ma impossibili per programmi classici. Alcuni di questi possono essere:

- **Classificazione:** un algoritmo di machine learning può imparare ad inserire un dato in una categoria, scegliendo fra un set di categorie. Ad esempio può riconoscere un oggetto in un'immagine, che viene data in input. Può anche fornire la probabilità che un certo input appartenga a una categoria oppure a un'altra.
- **Raggruppamento (*clustering*):** si raggruppano nello stesso insieme dati che presentano caratteristiche simili.
- **Regressione:** il programma deve prevedere variabili numeriche continue dato uno o più numeri in input.

1.1.2 L'esperienza

La maggior parte degli algoritmi di machine learning ricevono, come informazione da cui imparare un determinato compito, un *dataset*, ovvero una collezione di esempi.

Un dataset per un algoritmo classificatore può essere un insieme di immagini, all'interno delle quali deve riconoscere certi oggetti. Per un algoritmo di regressione, invece, il dataset è tipicamente costituito da una collezione di numeri o vettori di numeri, che fungono da argomenti della funzione che si vuole trovare.

Ogni esempio ha caratteristiche quantificabili, dette *feature*. Un dataset viene generalmente descritto nel programma da una matrice di dati, in cui ogni riga corrisponde a un esempio, ogni colonna corrisponde al valore di una specifica feature nei diversi esempi.

L'algoritmo può imparare a svolgere un compito a partire dal dataset in due modi: con l'apprendimento *supervisionato* o con quello *non supervisionato*. Negli algoritmi di apprendimento supervisionato (usati, ad esempio, per la classificazione o la regressione) ai dati in input sono associate delle etichette che tipicamente indicano l'output atteso per quel dato. L'apprendimento è infatti detto supervisionato perchè l'etichetta funge da "insegnante". Nell'apprendimento non supervisionato (usato ad esempio per il clustering) è richiesto all'algoritmo di riconoscere proprietà utili della struttura del dataset, come somiglianze tra le feature di alcuni degli esempi.

1.1.3 Misura della performance

Ciò che distingue un algoritmo di machine learning da uno di ottimizzazione è la capacità che deve avere il primo di generalizzare i risultati ottenuti a un set di dati mai visto (si parla di *generalizzazione*). Per questo motivo, tipicamente la performance dell'algoritmo è misurata su una collezione di dati sconosciuti, detta *test set*, distinta da quella grazie a cui l'algoritmo viene addestrato, detta *training set*.

La funzione che misura la performance varia da compito a compito. Per esempio, nella classificazione si misura spesso l'accuratezza del modello, ovvero la frazione degli esempi contenuti nel test set classificati correttamente dall'algoritmo. Nella regressione, invece, si misura la varianza tra il risultato stimato dall'algoritmo su un dato del test set e quello atteso.

1.1.4 Metodo di ottimizzazione: la discesa del gradiente

Gli algoritmi di machine learning usano metodi di ottimizzazione per convergere alla soluzione del problema che gli viene sottoposto. Ottimizzare significa trovare il minimo (o il massimo) di una certa funzione. Nel caso del machine learning la funzione da minimizzare è detta *funzione di costo*. Essa quantifica quanto bene l'algoritmo raggiunge il risultato atteso sul training set in fase di addestramento.

Uno dei metodi di ottimizzazione più usati è quello della *discesa del gradiente* o *gradient descent*, che descriviamo di seguito.

Sappiamo dall'analisi matematica che il valore della derivata di una funzione $f(w)$ calcolata in un punto w_0 quantifica la pendenza della funzione in quel punto. Il gradiente è la generalizzazione del concetto di derivata a funzioni $f(\mathbf{w})$ definite su uno spazio multidimensionale a valori reali. Il gradiente $\nabla f(\mathbf{w})$ di una funzione, calcolato in un punto \mathbf{w}_0 , è un vettore che punta alla direzione di massima pendenza della funzione in quel punto. Quindi la direzione di massima discesa è data dall'opposto del gradiente: $\mathbf{d} = -\nabla f(\mathbf{w})$. Se si varia il valore della variabile \mathbf{w} nella direzione di \mathbf{d} iterativamente, secondo la formula $\mathbf{w}' = \mathbf{w} + \epsilon \mathbf{d}$, dove ϵ è la lunghezza dello spostamento, si raggiunge un punto di minimo della funzione (o un suo intorno piccolo a piacere).

1.1.5 Esempio di addestramento: la regressione lineare

Vediamo un'applicazione dei concetti introdotti fin ora con l'esempio della regressione lineare.

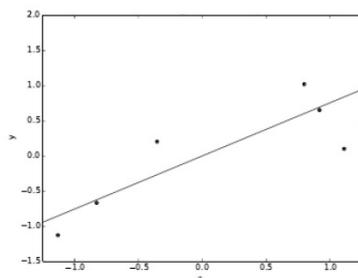


Figura 1.1: Esempio di regressione lineare, con curva stimata tramite un algoritmo di machine learning. Da [1].

Il training set è costituito da due vettori di dati $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$ e $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_n)$. Ogni coppia (\hat{x}_i, \hat{y}_i) individua un punto in un piano. L'obiettivo è di trovare i parametri a , di pendenza, e b di intercetta tali che la funzione $y = a\hat{x}_i + b$ più si avvicini ai valori del training set.

Una possibile funzione di costo per l'apprendimento può essere data dallo scarto quadratico medio (*mean squared error*) tra i valori di y calcolati dal programma e quelli del data set \hat{y} :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (ax_i + b - \hat{y}_i)^2.$$

Minimizzando lo scarto quadratico medio si ottiene una buona stima della retta (metodo dei minimi quadrati).

L'algoritmo è strutturato come segue. Inizialmente i valori di a e b sono inizializzati in modo casuale e forniscono una predizione sbagliata. Si può usare quindi il metodo della discesa del gradiente per minimizzare la funzione di costo rispetto ad a e b , ottenendo la stima ottimale. Il metodo prevede il calcolo del gradiente della funzione di costo rispetto alle variabili a e b :

$$\nabla MSE = \left(\frac{\partial}{\partial a} MSE, \frac{\partial}{\partial b} MSE \right),$$

dove:

- $\frac{\partial}{\partial a} MSE = \frac{2}{n} \sum_{i=1}^n -x_i (y_i - (ax_i + b))$

- $\frac{\partial}{\partial b}MSE = \frac{2}{n} \sum_{i=1}^n -(y_i - (ax_i + b))$

Si aggiorna quindi il valore della coppia (a, b) secondo la formula:

$$(a', b') = (a, b) - \epsilon \nabla MSE.$$

Si ripete il processo finchè non si raggiunge il punto di minimo della funzione di costo, che ha gradiente nullo. In questo modo otteniamo una retta che approssima bene la distribuzione di punti del training set.

Notiamo che l'algoritmo ha appreso il corretto valore dei parametri $a^{trained}$ e $b^{trained}$ tramite un confronto con i valori \hat{y}_i contenuti nel data set, quindi il tipo di apprendimento è quello supervisionato.

Possiamo misurare la bontà del modello calcolando lo scarto quadratico medio tra i valori di una collezione di punti (*test set*) che seguono la stessa distribuzione di quelli del *training set* e i valori calcolati coi parametri $a^{trained}$ e $b^{trained}$.

$$MSE^{test} = \frac{1}{n} \sum_{i=1}^n (y_i^{trained} - \hat{y}_i^{test})^2.$$

1.1.6 Underfitting e overfitting

Abbiamo visto che, in un algoritmo di machine learning, viene prima preso in considerazione training set e viene usato per ridurre al minimo la funzione di costo (cioè l'errore sul training set), poi viene usato il test set, rispetto al quale è calcolato l'errore, al fine di misurare la bontà della predizione. Quindi i fattori che bisogna rendere il più possibile piccoli durante l'addestramento dell'algoritmo sono due: l'errore sul training set e la differenza tra errore del training e del test set.

Per raggiungere questi obiettivi è necessario fronteggiare i fondamentali problemi degli algoritmi di machine learning: l'*underfitting* e l'*overfitting*.

Un modello è soggetto ad *overfitting* se è in grado di predire bene i risultati del training set ma non quelli del test set (differenza tra errore del training e del test set alta).

Si parla invece di *underfitting* quando non è in grado di modellizzare neanche il training set (errore del training set alto) (vedi Figura 1.2).

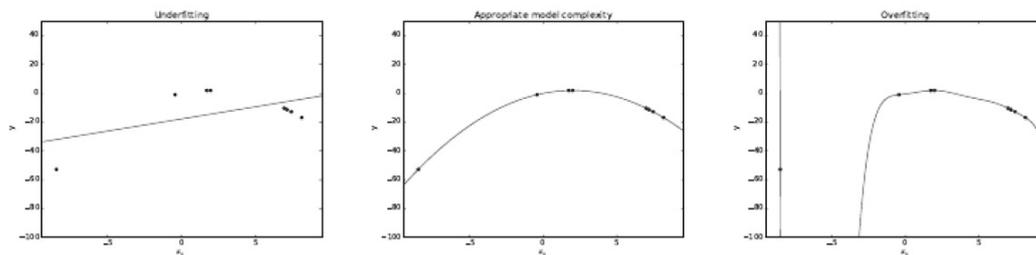


Figura 1.2: In figura si mostrano tre diversi modelli di fitting degli stessi punti. I punti sono stati generati con una distribuzione quadratica in x . A sinistra vediamo un modello in underfitting. Al centro vediamo il modello che meglio approssima i dati. A destra invece il modello è in overfitting. Da [1].

Quest'ultimo può essere causato dal fatto che il modello è troppo semplice: o le features usate sono troppe poche o non rappresentative del training set oppure sono usati pochi o sbagliati *iperparametri*. Gli iperparametri sono parametri utilizzati nel modello che non vengono imparati durante l'addestramento ma impostati dal programmatore a priori.

Per esempio, nel caso della regressione, se prendiamo in considerazione un modello che usa una funzione a due (iper)parametri per approssimare punti giacenti su una parabola, cioè un polinomio di secondo grado, funzione a tre parametri, allora il modello incorre in underfitting.

L'overfitting avviene invece quando il modello usa troppi iperparametri rispetto a quelli necessari per stimare la soluzione, adattandosi troppo a caratteristiche specifiche solo del training set (come il rumore).

Quindi funzioni più semplici sono in grado di generalizzare meglio, ma bisogna comunque scegliere ipotesi abbastanza complesse da minimizzare l'errore sul training set.

1.1.7 Evitare l'overfitting: le tecniche più comuni

Per evitare l'overfitting si può ricorrere a diverse soluzioni.

- Aumentare il numero di elementi contenuti nel training set: avere un training set molto numeroso aiuta ad elaborare un modello il più possibile generico. Questa necessità è uno dei maggiori problemi nel caso dei metodi di machine learning applicati all'imaging medico: spesso sono disponibili troppe poche immagini per

il grado di complessità della previsione che si vuole raggiungere. Per risolvere il problema si può creare un database condiviso di immagini provenienti da diversi ospedali ma accessibile a tutti. In alternativa si deve ricorrere alle tecniche qui esposte.

- La tecnica della *k-fold cross validation*: può essere utile suddividere il training set in due insiemi: uno contenente i dati che vengono usati per aggiustare i parametri del modello (training set vero e proprio), il secondo, detto *validation set* che serve per stimare, tra diversi possibili modelli ottenuti a partire dal training set, quello che generalizza meglio la soluzione (stima quindi gli iperparametri). Questa suddivisione viene usata nella tecnica della *k-fold cross validation*, in cui si partiziona l'insieme dei dati contenuto nel training set in k insiemi più piccoli che non si sovrappongono e di uguale numerosità, ciascuno dei quali viene escluso a turno per calcolare la soluzione. Viene poi selezionata la soluzione che meglio prevede il comportamento dell'insieme escluso (*validation set*).
- Rimuovere delle features: alcune delle features usate per creare il modello possono essere ridondanti, irrilevanti o legate strettamente agli esempi del training set.
- Le tecniche di *regolarizzazione*: sono tecniche pensate per ridurre l'errore di generalizzazione ma non quello di addestramento. La tecnica è strettamente legata al tipo di algoritmo usato.

1.2 Il Deep Learning

Abbiamo accennato nel paragrafo 1.1.2 ogni elemento del data set ha caratteristiche quantificabili dette feature. L'insieme delle feature che vengono fornite all'algoritmo è detto *rappresentazione* dell'elemento del data set.

La performance degli algoritmi di machine learning dipende molto dalla rappresentazione dei dati che viene loro fornita: per esempio in un algoritmo di riconoscimento vocale, non saranno importanti le caratteristiche della voce del parlante che derivano dal sesso o dall'accento, ma piuttosto l'altezza del suono. Quindi un'algoritmo molto semplice a cui viene data quest'ultima caratteristica in input svolgerà il suo compito molto meglio rispetto ad algoritmi complicati che ricevono caratteristiche irrilevanti.

Dato che fornire all'algoritmo dati complessi come immagini o suoni in forma grezza è estremamente inefficiente, è necessario elaborare tecniche per ricavare dai dati le informazioni utili. Questo compito, chiamato *estrazione delle feature*, può essere svolto tramite la scrittura di programmi appositi, che cercano all'interno dei dati grezzi delle caratteristiche che il programmatore ritiene essere importanti per l'algoritmo di machine learning.

Questo approccio può presentare dei problemi: quando le feature sono estratte "a mano" (*hand-crafted features*) è richiesto l'intervento di un esperto che conosca bene il compito sottoposto all'algoritmo di machine learning, e che possa quindi scegliere le features rilevanti. Può avvenire che sia molto difficile, anche per uno specialista, riconoscere quali siano le feature rilevanti.

Una possibile soluzione è fare in modo che l'algoritmo apprenda non solo a classificare l'input ma anche a riconoscere le feature essenziali. Questo tipo di apprendimento è detto *representation learning*.

Il *deep learning* è un insieme di metodi di representation learning con molti livelli di rappresentazione: si parte da una rappresentazione di livello inferiore, simile al dato grezzo, e si affina arrivando ad una rappresentazione di livello superiore e più astratta, costruita come una composizione di quelle più semplici.

Il deep learning si è rivelato un metodo efficace per svolgere compiti quali riconoscimento vocale, riconoscimento visuale di oggetti, filtro di immagini sui social network. Un'applicazione importante è anche quella all'imaging medico: in questo ambito il deep learning è usato per la segmentazione delle immagini, per l'etichettatura automatica delle immagini con l'organo o la malattia corrispondente, per la diagnosi di malattie (computer-aided detection), nell'ambito della radiomica (correlazione di feature delle immagini mediche con esiti clinici), ecc.

1.2.1 Le reti neurali

Il deep learning si basa sull'utilizzo di sistemi computazionali chiamati *reti neurali artificiali* (*artificial neural network*). L'architettura delle reti neurali artificiali è ispirata a modelli biologici del cervello umano, e nasce col fine di riprodurre attività come il riconoscimento di immagini, comprensione del linguaggio, ecc.

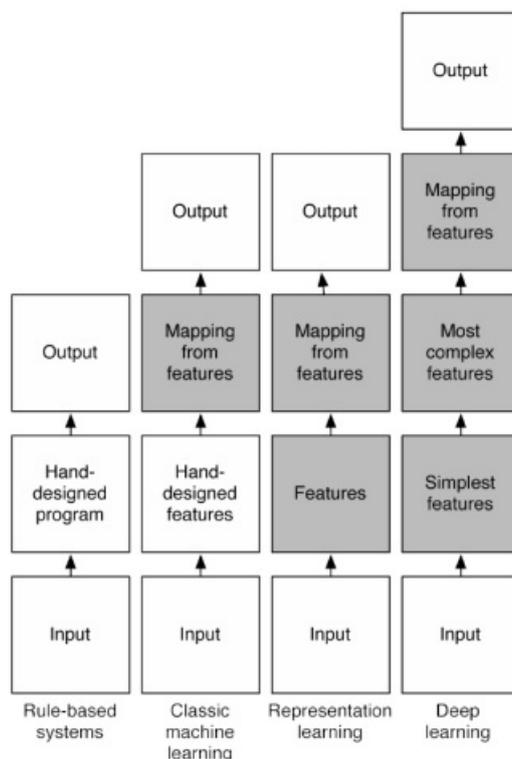


Figura 1.3: Diagramma che mostra in che modo i diversi step di apprendimento si susseguono nei vari algoritmi di intelligenza artificiale. I quadrati scuri indicano componenti che imparano dai dati. Da [1].

Una rete neurale è costituita da nodi detti *neuroni* che, in base all'input che ricevono, cambiano il loro stato (attivo o non attivo) e mandano un certo output. Gli output di alcuni neuroni sono collegati agli input di altri, a formare una rete, tramite collegamenti pesati e direzionati.

Il perceptrone

Per capire il funzionamento di una rete neurale, partiamo analizzando la sua unità fondamentale: il neurone.

La Figura 1.4 mostra una rete neurale costituita da un solo neurone. Questo modello è chiamato *perceptrone*. Il neurone ha n canali di input x_1, \dots, x_n , ciascuno dei quali è collegato alla prima funzione $y(x_1, \dots, x_n)$ tramite un collegamento pesato, di peso w_1, \dots, w_n . Il valore assoluto del peso indica la forza del collegamento. La funzione y non

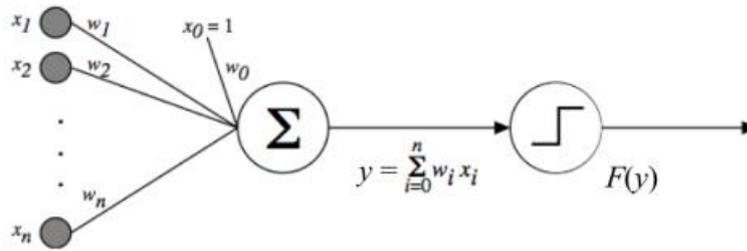


Figura 1.4: Modello di un neurone.

è altro che la combinazione lineare degli input con i pesi come coefficienti (la riportiamo scritta anche in notazione vettoriale):

$$y = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^n w_i x_i.$$

Il valore di y viene passato, assieme ad un'altra variabile Θ , detta *valore di soglia*, a una seconda funzione F detta *funzione di attivazione*. Quest'ultima determina lo stato di attivazione del neurone (acceso o spento). Le più comuni funzioni di attivazione sono:

- la funzione di Heaviside (a gradino)

$$F_H(y, \Theta) = \begin{cases} 1 & \text{se } y + \Theta > 0 \\ 0 & \text{se } y + \Theta < 0 \end{cases} \quad (1.1)$$

- la funzione logistica:

$$F_L(y, \Theta) = \frac{1}{1 + e^{-y+\Theta}} \quad (1.2)$$

Il valore di soglia è il valore dell'input oltre il quale si attiva il neurone. Ha infatti l'effetto di far slittare a destra o a sinistra la funzione F .

È facile notare che esso può essere rappresentato da un canale di input aggiuntivo x_0 che rimane sempre attivo (quindi ha valore 1), e pesato con l'inverso del valore di soglia: $w_0 = -\Theta$. Questo canale è chiamato *bias*. In questo modo possiamo riscrivere la funzione y come:

$$y = \sum_{i=0}^n w_i x_i,$$

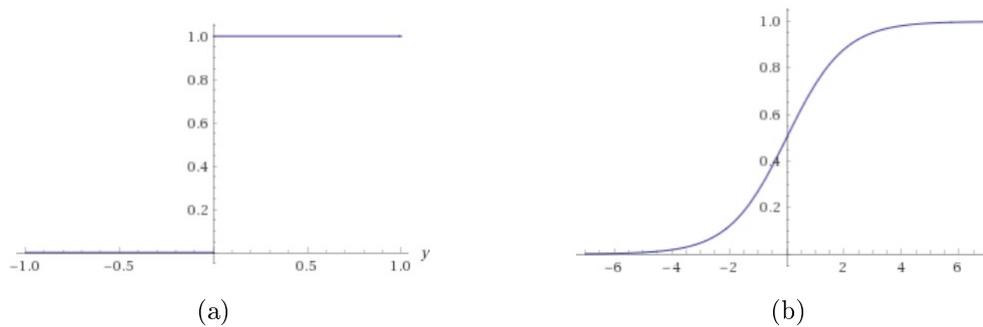


Figura 1.5: (a) La funzione di Heaviside e (b) la funzione logistica per $\Theta = 0$.

e togliere la dipendenza esplicita da Θ nelle funzioni di attivazione.

Esempio di addestramento del perceptrone: la classificazione

Un perceptrone può svolgere compiti di classificazione lineare.

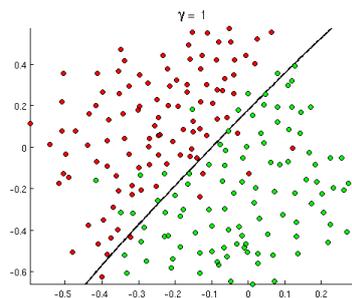


Figura 1.6: Oggetti classificabili linearmente. I punti in verde identificano oggetti appartenenti alla prima classe, quelli in rosso alla seconda. La retta rappresentata è quella che approssima meglio la separazione tra le due classi. Da openclassroom.stanford.edu.

Supponiamo ad esempio di avere oggetti rappresentabili come punti in un piano, quindi individuabili da due coordinate $\mathbf{x}_i = (x_{1,i}, x_{2,i})$, e che questi oggetti appartengano a due classi linearmente separabili. Vogliamo usare il perceptrone per individuare la retta che, all'interno del piano, separa le due classi.

Il perceptrone avrà due ingressi x_1 e x_2 , a ciascuno dei quali è associato il peso w_1 e w_2 , più l'ingresso di bias di peso Θ . La funzione y avrà quindi la forma:

$$y = w_1x_1 + w_2x_2 + \Theta.$$

La funzione y viene data in input alla funzione di attivazione che supponiamo essere a gradino.

L'algoritmo classifica il punto come appartenente alla classe C_1 quando la funzione di attivazione vale 1, alla classe C_0 altrimenti. Quindi, dalla definizione 1.1, risulta

$$x \in C_0 \text{ se } x_2 \leq -\frac{w_1}{w_2}x_1 + \frac{\Theta}{w_2} \quad (1.3)$$

dove la disequazione individua le due porzioni di piano separate dalla retta.

L'apprendimento è supervisionato: si etichettano i dati in input \mathbf{x}_i con l'uscita attesa, si inizializzano i pesi in modo casuale e si aggiustano nel corso dell'addestramento, che viene effettuato con i metodi illustrati nella sezione 1.1, fino ad ottenere i pesi che meglio approssimano la retta di separazione tra le due classi.

Il perceptrone è efficace solo quando le classi si possono separare con una linea retta. Ma in molti casi la classificazione non è lineare (ne è riportato un esempio in figura 1.7)

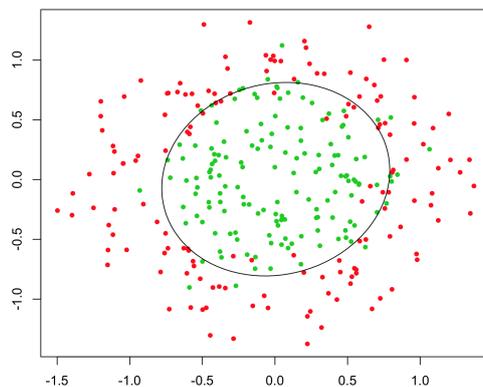


Figura 1.7: Esempio di oggetti non classificabili linearmente. Da openclassroom.stanford.edu.

Per questi compiti si possono usare reti neurali profonde (*deep neural network*), che sono le architetture usate negli algoritmi di deep learning.

1.2.2 Reti neurali profonde: il multilayer perceptron

Il tipo più semplice di rete neurale profonda è il *multilayer perceptron*, un tipo di rete feedforward.

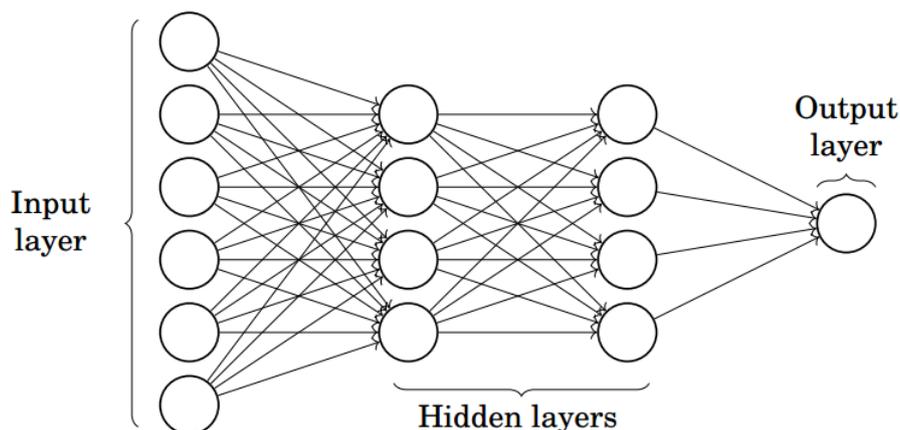


Figura 1.8: Multilayer perceptron. Da <https://www.vaetas.cz/blog/introduction-artificial-neural-networks/>.

In questo tipo di rete (illustrata in Figura 1.8) ci sono diversi strati (*layer*) di neuroni, che si dividono in *input layer*, *hidden layer* e *output layer*. I collegamenti esistono solo tra i neuroni di uno strato e quelli dello strato successivo, e sono direzionati verso l'output (perciò si chiamano feedforward: l'informazione passa dall'input all'output senza tornare indietro).

L'input layer riceve i dati del data set. Ci sono tanti neuroni quanti sono gli input da processare: per esempio in un'immagine ogni neurone corrisponderà a un pixel.

Gli hidden layer possono essere molti: più hidden layer ci sono più il dato sarà rappresentato in modo astratto. Infatti ciascun neurone di uno strato trasforma il suo input in modo non lineare (la funzione di attivazione può essere logistica, arcotangente...), e componendo molti neuroni in diversi strati si possono ottenere trasformazioni dello spazio dell'input fortemente non lineari. In questo modo si possono rendere separabili regioni dello spazio degli input che non lo erano col singolo perceptrone.

È utile notare che l'operazione svolta da ciascun hidden layer può essere scritta in notazione matriciale. Consideriamo uno strato i che riceve in input dei dati $\mathbf{x}^{(i)} =$

$(x_1^{(i)}, \dots, x_n^{(i)})$, che sono a loro volta gli output dello strato $i - 1$. La funzione che manderà in output può essere scritta come:

$$f(\mathbf{x}^{(i)}) = F(W^{(i)}\mathbf{x}^{(i)}), \quad (1.4)$$

dove $W^{(i)}$ è la matrice dei pesi e F è la funzione di attivazione.

L'output layer contiene uno o più neuroni, a seconda del compito che la rete deve svolgere. Per esempio, nei compiti di classificazione, ci sono tanti neuroni quante sono le classi individuabili dall'algoritmo.

1.2.3 Addestramento delle reti neurali profonde

L'addestramento (supervisionato) delle reti profonde è analogo a quello di tutti gli altri algoritmi di machine learning: dato un training set in cui ciascun elemento è etichettato con la classe di appartenenza attesa, ognuno di tali elementi è dato in input alla rete, i cui pesi sono inizializzati in modo casuale. La rete produrrà un certo output, che viene confrontato con quello atteso tramite la funzione di costo.

La funzione di costo per queste reti può essere il classico scarto quadratico medio, ma per compiti di classificazione è risultato più conveniente usare funzioni specifiche per output attesi $y \in \{0, 1\}$ e output calcolati $f(\mathbf{x}) \in [0, 1]$, con \mathbf{x} vettore degli input (cosa che avviene quando la funzione di attivazione dell'output layer è logistica o a gradino). La funzione di costo più usata che abbia queste caratteristiche è la *cross entropy*, definita da:

$$L(f(\mathbf{x}), y) = -y \log f(\mathbf{x}) - (1 - y) \log(1 - f(\mathbf{x})) \quad (1.5)$$

Per quanto riguarda il metodo di ottimizzazione, con un numero N altissimo di esempi nel data set, che è una condizione necessaria per addestrare algoritmi con così tanti parametri, usare il metodo gradient descent standard rallenta tantissimo la computazione. Perciò viene generalmente usato lo *stochastic gradient descent*, che è identico al primo se non per il fatto che viene calcolata la direzione del gradiente per un set ristretto di n dati, a turno:

$$\mathbf{w}' = \mathbf{w} - \epsilon \nabla(f(\mathbf{w}), x_1, \dots, x_n, y_1, \dots, y_n). \quad (1.6)$$

Ciò che comporta l'utilizzo di questo metodo è che la discesa del gradiente non avviene sempre lungo la direzione di massima pendenza, ma ogni passo avrà una direzione che devia di poco da quest'ultima, dato che viene scelto basandosi su un set ristretto di dati.

La convergenza a un punto di minimo può risultare abbastanza lenta utilizzando lo stochastic gradient descent, soprattutto dopo i primi passi. Un modo per velocizzare l'algoritmo è il *momento*.

L'algoritmo di stochastic gradient descent con momento prevede che l'aggiornamento dei pesi ad ogni iterazione dipenda dagli aggiornamenti precedenti, secondo la formula:

$$\mathbf{w}' = \mathbf{w} - \epsilon \nabla f(\mathbf{w}) + \alpha \Delta \mathbf{w}, \quad (1.7)$$

dove $\Delta \mathbf{w}$ è la differenza tra i pesi \mathbf{w} dell'iterazione attuale e di quella precedente, mentre $\alpha \in (0, 1]$ è il momento, ed indica che frazione dell'aggiornamento precedente è tenuta in considerazione per il calcolo di \mathbf{w}' . In questo modo si riduce l'oscillazione della direzione di discesa ad ogni iterazione.

Il calcolo del gradiente per una funzione stratificata e a molti parametri come quella dell'output di un multilayer perceptron è un'operazione non banale. L'algoritmo che lo svolge è chiamato *backpropagation*.

1.3 Reti neurali convoluzionali

Le reti neurali convoluzionali (CNN) sono un tipo di rete neurale feedforward che viene usata per l'analisi delle immagini. Si chiamano *convoluzionali* perchè almeno un layer della rete usa l'operazione della convoluzione sugli input, al posto della matrice dei pesi. In generale, però, in questo tipo di rete i layer convoluzionali sono accompagnati da layer di *pooling*, layer *fully connected* e layer di normalizzazione. Descriviamo in seguito il funzionamento di ciascuno di questi.

Le CNN si sono dimostrate efficacissime nei compiti di classificazione visuale (come quella di caratteri scritti a mano), o nell'individuazione di oggetti (come i volti) nelle immagini.

Si è riaccessi, negli ultimi anni, l'interesse verso le potenzialità delle CNN a causa di una maggiore disponibilità di set di dati già classificati (come ImageNet, un database di immagini etichettate creato nel 2009 per favorire la ricerca nei software di ricono-

scimento di oggetti nelle immagini), oltre che per il miglioramento delle strategie di regolarizzazione degli algoritmi e per l'implementazione di GPU più potenti ([3]).

Ciò che rende le CNN così efficaci nel riconoscimento delle immagini è la capacità che hanno di comporre caratteristiche generali (come la presenza di bordi, di punti chiari o scuri, ecc..) in forme più complesse (nel caso di un volto: occhi, labbra, ecc..), fino a riconoscere l'oggetto completo (il volto stesso). Ciò che rende possibile questa progressiva composizione delle caratteristiche delle immagini è la struttura a molti strati della rete.

Un'altro vantaggio fondamentale è che l'utilizzo di layer convoluzionali aumenta esponenzialmente l'efficienza dell'algoritmo perchè, al contrario di quanto avviene in un multilayer perceptron, ogni neurone di un layer convoluzionale è connesso solo ai neuroni in input contenuti in una piccola regione. Questo fatto, oltre che incrementare l'efficienza, fa sì che sia possibile individuare caratteristiche come i bordi e i punti di luminosità all'interno di regioni ristrette, per poi riconoscere la totalità dell'immagine nei layer successivi.

Infine, il riconoscimento degli oggetti all'interno di un'immagine è reso possibile dall'invarianza traslazionale di cui gode l'individuazione delle caratteristiche in un layer convoluzionale. Questo vuol dire che un oggetto può essere riconosciuto indipendentemente dalla sua posizione all'interno dell'immagine.

1.3.1 Struttura delle CNN

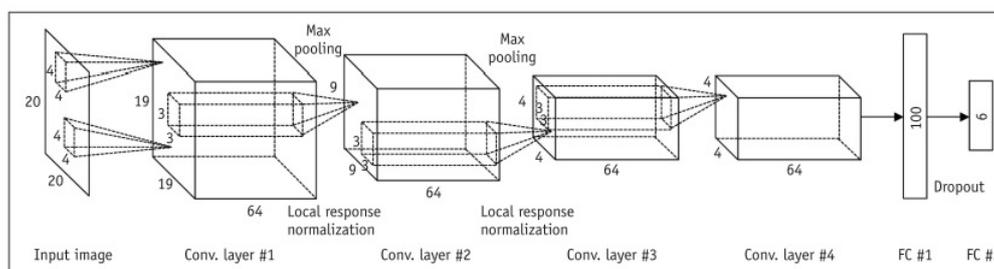


Figura 1.9: Struttura di una rete neurale convoluzionale. L'immagine in input passa attraverso layer convoluzionali (di profondità uguale al numero dei kernel convoluti sull'immagine), poi il segnale viene normalizzato, generalmente con la funzione ReLU, definita nell'equazione 1.11, e passato al layer di pooling, che ne diminuisce la dimensione in altezza e larghezza. Questo processo è ripetuto diverse volte, finché il segnale non arriva all'ultimo layer completamente connesso. Da [4].

Layer convoluzionali

La convoluzione è un'operazione matematica tra due funzioni f e w , che produce una terza funzione g definita come:

$$g(x) = \int f(s)w(x-s)ds. \quad (1.8)$$

In informatica non abbiamo mai a che fare con funzioni continue, quindi è utile introdurre la convoluzione discreta:

$$g(x) = \sum_n f(n)w(x-n). \quad (1.9)$$

Nell'ambito del machine learning, la funzione w è generalmente chiamata *kernel*, mentre la funzione in uscita g è chiamata *mappa del kernel* o *mappa della feature*. La funzione $f(x)$ è in genere una funzione della posizione o del tempo che viene data in input al layer convoluzionale, mentre il kernel $w(x)$ è la funzione che viene imparata durante l'algoritmo di deep learning.

La convoluzione per funzioni unidimensionali trova un utilizzo nell'ambito del riconoscimento vocale: la funzione di input è la caratteristica del suono in funzione del tempo.

Per processare le immagini invece, dato che la funzione di input è bidimensionale, si usa la convoluzione discreta per funzioni a due dimensioni:

$$g(x_1, x_2) = \sum_n \sum_m f(n, m)w(x_1 - n, x_2 - m), \quad (1.10)$$

dove $f(n, m)$ individua uno specifico pixel della figura di input, $w(n, m)$ individua una specifica posizione nel kernel. Il kernel è un rettangolo di dimensione molto minore dell'immagine: generalmente qualche decina di pixel contro le migliaia di cui sono costituite in generale le immagini.

Il kernel rappresenta una feature (può essere una piccola linea o un punto luminoso), che viene cercata all'interno dell'immagine in input come descritto in seguito.

Si può pensare al kernel come il neurone della rete, e al valore di ogni suo pixel come il peso che deve essere aggiustato durante l'algoritmo di apprendimento: alla fine verranno imparati kernel dotati di pesi tali che la feature individuata è utile per riconoscere l'oggetto da classificare: per esempio, se si vuole individuare un'automobile, è probabile

che la rete impari kernel che permettono di individuare le ruote.

In un layer convoluzionale, il kernel viene sovrapposto a una certa porzione dell'immagine, centrata in (x_1, x_2) . Il valore di pixel di posizione $(x_1 - n, x_2 - m)$ del kernel viene moltiplicato per il valore del pixel dell'immagine nella posizione corrispondente. I risultati ottenuti per ogni pixel sono mediati tra loro (quest'operazione è rappresentata dalla sommatoria nell'equazione 1.10): il risultato finale di quest'operazione costituisce il valore del pixel in posizione (x_1, x_2) della mappa del kernel, $g(\mathbf{x})$. Quest'operazione è chiamata *filtering*, e per questo motivo il kernel è anche chiamato *filtro*.

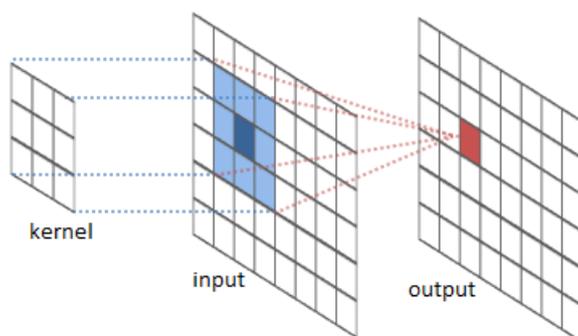


Figura 1.10: Filtering dell'immagine.

Da <http://intellabs.github.io/RiverTrail/tutorial/>.

Per ottenere il valore dei rimanenti pixel, si *convolve* il kernel sull'immagine, ovvero si trasla il kernel e lo si sovrappone con un'altra porzione dell'immagine, ripetendo il filtering finchè non si satura l'immagine, e si ottiene la mappa del kernel completa.

Si può scegliere la dimensione della mappa del kernel a partire da tre parametri: la profondità, il passo e lo *zero padding*.

- *Profondità*. I kernel usati per la convoluzione sono molteplici, in modo tale da poter individuare più feature nell'immagine. Il fatto di avere diversi kernel nello stesso layer si può rappresentare attribuendo al layer una profondità D , dove D è il numero di kernel convoluti sull'immagine (vedi figura 1.9).
- *Passo*. La convoluzione del kernel avviene spostando quest'ultimo in una posizione che dista uno, due o, più raramente, tre o più pixel da quella precedente. Nel caso lo spostamento sia di un pixel, la dimensione dell'output rimane dell'input, nel caso sia due o più si restringe.

- *Zero padding*. Se la dimensione dell'immagine non è un multiplo del passo, convolvere il kernel significa anche diminuire la dimensione dell'immagine, poichè, dato che il kernel non si può spostare oltre il bordo, i pixel ai bordi non vengono filtrati e non hanno un corrispettivo nell'output. Questa diminuzione della dimensione dell'output può essere evitata aggiungendo pixel di valore zero ai bordi dell'immagine. Quest'operazione è chiamata *zero padding*.

Connettività locale e condivisione dei parametri

Il fatto che il kernel sia più piccolo dell'immagine in input dà luogo ad un'importante proprietà delle reti convoluzionali: la connettività locale (*sparse connectivity*). Infatti possiamo pensare al kernel come a un neurone collegato solo alle unità di input (pixel dell'immagine) adiacenti ad esso.

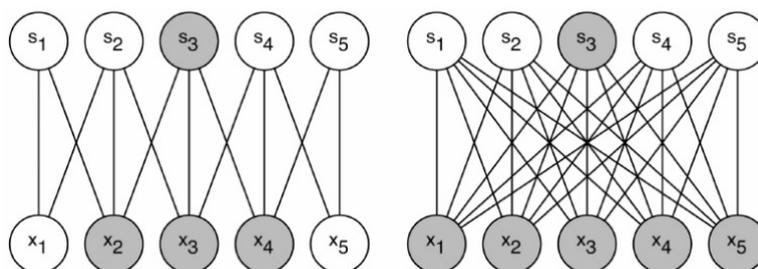


Figura 1.11: Connettività tra unità di input x_i e neuroni s_i del layer convoluzionale a sinistra, di un layer completamente connesso a destra. Nell'immagine di sinistra sono evidenziati in grigio le unità di input che sono connessi a un kernel di dimensione 3. A destra, invece, si nota che i neuroni classici (come quelli del multilayer perceptron) sono collegati a tutte le unità di input. Ciò provoca un aumento del numero dei pesi da aggiustare durante l'apprendimento. Da [1]

La dimensione del kernel, nonché il numero di unità di input collegate con ciascun neurone, è un iperparametro chiamato *campo recettivo* (*receptive field*).

Il fatto di avere connettività locale riduce di molto il numero dei parametri (pesi) e di conseguenza la velocità di calcolo e l'efficacia del modello.

Un altro fattore che contribuisce ai vantaggi citati sopra è la *condivisione dei parametri*: infatti tutti i neuroni che si trovano a un fissato livello di profondità del layer convoluzionale hanno identici pesi, poichè cercano di individuare la stessa feature in punti diversi dell'immagine.

Fase di normalizzazione

Nella fase di normalizzazione la mappa ottenuta viene data in input a una funzione di attivazione non lineare: aumentando la non linearità del modello se ne aumenta la complessità.

Ci sono diverse possibili funzioni di attivazione, come *tanh* e la funzione logistica, ma si è verificato che la più efficace è la *rectified linear unit* o *ReLU*, definita come:

$$f_{ReLU}(x) = \max(0, x) \quad (1.11)$$

Notiamo che la ReLU non fa altro che sostituire con zero i valori negativi dei pixel della mappa.

La ReLU è preferibile alle altre funzioni di attivazione perchè rende la computazione molto più veloce e risolve problemi di gradiente zero.

Layer di pooling

La mappa delle feature creata dal layer convoluzionale è data in input al layer di pooling.

L'immagine in input del layer viene suddivisa in un set di rettangoli che non si intersecano tra loro, e ogni rettangolo viene mappato con una funzione statistica del valore dei pixel dello stesso. La funzione più usata è quella di *max pooling*: viene mandato in output il pixel con il valore più alto tra quelli del rettangolo.

La funzione di pooling fa sì che non sia individuata la precisa posizione della feature ma la sua zona, rendendo l'algoritmo invariante per piccole traslazioni delle feature. Quest'invarianza è una proprietà fondamentale: vuol dire che interessa più se la feature è presente o meno, piuttosto che dove si trovi.

Allo stesso tempo riduce la dimensionalità dell'immagine, assieme al numero di parametri e quindi alla possibilità di overfitting. La riduzione dei parametri inoltre rende l'algoritmo molto più veloce rispetto a quello di una rete completamente connessa.

Il pooling layer, in alcuni compiti, è anche essenziale per gestire immagini di input con diverse dimensioni. Infatti il classificatore alla fine della rete generalmente deve ricevere in input immagini della stessa dimensione. Il problema viene risolto modificando, in modo opportuno per ogni immagine, la dimensione dei rettangoli di pooling del layer subito precedente al classificatore.

Layer fully connected

Diversi layer convoluzionali si succedono, intervallati da layer di pooling e di normalizzazione all'interno di una CNN: il primo layer di questo tipo individua feature più semplici, i successivi compongono quelle più semplici in una mappa complessa.

Dopo che il segnale è passato attraverso diversi strati convoluzionali, di normalizzazione e di pooling, viene dato in input al layer completamente connesso (fully connected). Al contrario dei neuroni nel layer convoluzionale, qui ciascun neurone è connesso a tutti i neuroni dello strato precedente. Servono a integrare tutte le feature trovate nell'immagine fornendo il risultato finale (può, ad esempio, classificare l'immagine).

1.3.2 Addestramento delle CNN

Per l'addestramento delle reti neurali convoluzionali si usano, come nel multilayer perceptron, gli algoritmi di stochastic gradient descent con momento per l'ottimizzazione e di backpropagation per il calcolo del gradiente.

Per contrastare l'overfitting, vengono usate tecniche di regolarizzazione come il *dropout* o l'incremento dei dati. Il primo è una tecnica che consiste nell'impostare una probabilità P che un neurone sia momentaneamente eliminato, assieme ai suoi collegamenti di input e di output inizializzati casualmente, durante la fase di addestramento, e che sia invece reinserito nella rete durante quando l'algoritmo è provato sul test set. Questo aumenta la capacità di generalizzazione.

La tecnica di incremento dei dati invece si utilizza quando il data set è ristretto e consiste nel creare nuovi esempi perturbando le immagini già contenute nel data set.

Transfer Learning

In diverse applicazioni, mediche e non, è impossibile addestrare una CNN da zero, ovvero partendo da pesi inizializzati in modo casuale, a causa dell'esiguità degli esempi nel data set. In questi casi si può addestrare la rete tramite il *transfer learning*, un metodo che consiste nell'utilizzare come pesi di inizializzazione quelli di una rete già addestrata su un data set più ampio. Si parla in questo caso di CNN pre-addestrata (*pre-trained*).

Data una rete pre-addestrata, la si può utilizzare come estrattore di feature fissato, e addestrare solo l'ultimo layer completamente connesso specificatamente per il compito

proposto, al posto dell'intera rete. In alternativa si possono ricalibrare tutti i pesi o quelli degli ultimi layer.

1.3.3 Applicazioni delle CNN

Gli algoritmi di deep learning, grazie alla loro capacità di imparare feature a diversi livelli di complessità, hanno trovato un utilizzo in moltissimi ambiti.

In particolare, le CNN si sono rivelate tra i migliori algoritmi per la soluzione di una serie di compiti quali: riconoscimento facciale (questa categoria comprende compiti quali il riconoscimento dei volti in un'immagine, ma anche identificare le feature tipiche di uno specifico volto e associarlo al nome della persona); etichettatura di scene (è un compito che consiste nell'etichettare ciascun pixel con la classe dell'oggetto a cui appartiene); classificazione di immagini; riconoscimento dell'azione che avviene nell'immagine; riconoscimento della posa della persona presente nell'immagine; analisi dei documenti (compito che consiste nel riconoscere caratteri scritti a mano all'interno di un testo). Nell'ambito dell'elaborazione del linguaggio naturale (*natural language processing*), le CNN sono utilizzate con successo nei compiti di riconoscimento vocale o la classificazione di un testo (che può essere utile, ad esempio, per separare le mail di spam dalle rimanenti) ([6]).

Per quanto riguarda, invece, l'ambito medico, le CNN sono usate generalmente per l'analisi delle immagini radiologiche, acquisite ad esempio con le tecniche di *computed tomography* (CT), *positron emission tomography* (PET), risonanza magnetica (MR) e raggi X, o istopatologiche, acquisite al microscopio.

I maggiori utilizzi delle CNN per l'analisi delle immagini mediche sono i seguenti.

- *Segmentazione.* La segmentazione dell'immagine consiste nel partizionare l'immagine in regioni aventi pixel con caratteristiche omogenee, al fine di rappresentarla in un modo più semplice da analizzare.

Le CNN sono state usate, ad esempio, per la segmentazione del cervello, distinguendo tra il cranio e il cervello stesso nelle immagini MR tridimensionali, ottenendo sensibilità pari e sensibilità maggiore rispetto ad altri metodi ([7]).

- *Individuazione di organi.* Data un'immagine (ad esempio MR) che rappresenta una sezione di una certa parte del corpo, può essere utile distinguere con precisione

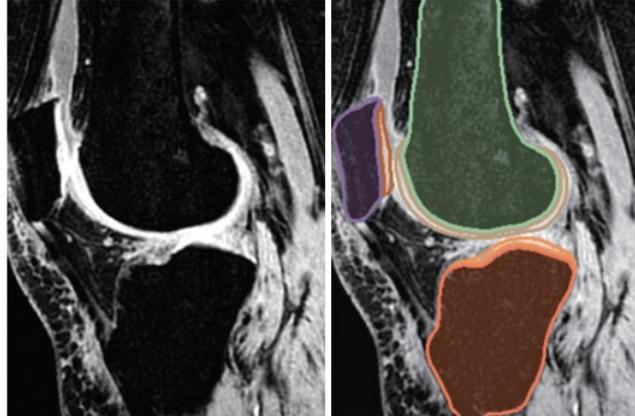


Figura 1.12: Segmentazione dell'immagine MR di un ginocchio.
Da [4]

all'interno della sezione i diversi organi. È stata testata una CNN pre-addestrata a questo scopo ([7]), ottenendo una buona accuratezza (92,2%).

- *Individuazione di cellule.* Può essere necessario, per la valutazione di alcune malattie, analizzare ogni singola cellula di un tessuto a partire da immagini al microscopio. Ad esempio, è stata addestrata una CNN al riconoscimento di mitosi nelle immagini istologiche del seno ([7]). La CNN era capace di individuare con grande accuratezza il pixel corrispondente al centro della cellula.
- *Sistemi di diagnosi assistita dal computer (CAD).* I CAD sono sistemi informatici che aiutano il medico nella diagnosi di malattie interpretando le immagini MR o a raggi X. I sistemi CAD individuano nell'immagine i punti dove più probabilmente è presente una malattia o una lesione. Essi si sono rivelati particolarmente efficaci nell'individuazione di cancro al seno tramite mammografie, tanto da diventare parte della routine clinica negli Stati Uniti ([8]).

Un sistema CAD si può suddividere in quattro compiti principali, che hanno lo scopo di modificare l'immagine ed analizzarla ([9]). Il primo è la pre-elaborazione dell'immagine medica grezza, al fine di rimuovere eventuale rumore che provoca la distorsione delle feature dell'immagine. Il secondo consiste nella segmentazione, per separare la regione d'interesse dal resto dell'immagine. Poi avviene l'estrazione e la

successiva selezione delle feature utili per il quarto stadio, ovvero la classificazione delle immagini sospette come corrispondenti a tumori benigni o maligni.

In generale la performance dei sistemi CAD nella diagnosi non è eccellente ([4]), per questo vengono usate solo come strumento di supporto all'analisi dell'immagine da parte di un radiologo esperto.

Uno dei motivi del problema è l'alta sensibilità dei sistemi CAD al rumore. Per risolvere il problema si possono usare sistemi basati sul deep learning (in particolare sulle CNN) nella fase di estrazione delle feature e/o di segmentazione. Sono già stati implementati sistemi di questo tipo per la diagnosi di cancro al seno, ai polmoni e Alzheimer ([4]), ed hanno mostrato ottime performance.

- *Medicina di precisione.* La medicina di precisione è un modello che propone un'assistenza sanitaria adattata al singolo paziente, che comporta diagnosi più accurate e trattamenti mirati. Per raggiungere questo obiettivo, la medicina di precisione si basa sull'analisi delle immagini mediche o del genoma di un paziente, in cui si cerca la presenza di biomarker che permettono di determinare il rischio di malattia, la prognosi o la risposta al trattamento.

Il deep learning può essere usato per individuare biomarker a partire da una gran quantità di immagini mediche o di dati genetici.

Un campo in evoluzione della medicina di precisione è la *radiomica*, il cui scopo è di estrarre, dalle immagini mediche, feature utili ad esempio per la predizione dell'evoluzione di certi tipi di cancro. Nel prossimo capitolo verrà approfondita questa branca della medicina di precisione.

Capitolo 2

La Radiomica

2.1 Cos'è la radiomica

La radiomica è un campo dell'analisi delle immagini mediche, in cui da queste ultime vengono estratte feature quantitative (dette feature radiomiche), sulla base delle quali è costruito un modello utilizzato come sistema di supporto alle decisioni mediche per migliorare la diagnosi, la prognosi e l'accuratezza predittiva. Le feature radiomiche forniscono quindi informazioni utili per costruire terapie personalizzate: per questo motivo la radiomica è considerato uno strumento efficace nella medicina di precisione.([10])

Anche se la radiomica assomiglia ai sistemi CAD (descritti nella sezione 1.3.3), differisce da questi negli scopi([11]): mentre i sistemi CAD servono come aiuto all'individuazione oppure alla diagnosi della malattia(quindi un singolo compito), la radiomica è una tecnica di *data mining* : fa uso di una grande quantità di dati (feature estratte da centinaia o migliaia di immagini mediche contenute in database condivisi fra ospedali), che vengono usate per la generazione di un modello *predittivo* e per una sua verifica.

La radiomica trova la sua principale applicazione nell'ambito dell'oncologia di precisione, in cui le feature estratte vengono utilizzate per predire possibili esiti clinici, come l'insorgere di metastasi, di recidiva locoregionale o di decesso causato dal cancro.

Infatti la radiomica può rivelarsi uno strumento utilissimo per superare il problema dall'eterogeneità dei tumori: in una stessa zona possono manifestarsi tipi di tumore diversi in morfologia cellulare, espressione genica, metabolismo, potenzialità di metastasi, ecc (si parla di eterogeneità inter-tumorale). All'interno di uno stesso tumore, inoltre,

possono essere presenti tipi di cellule diverse (si parla di eterogeneità intra-tumorale).

Ogni tipo di tumore ha un decorso tipico che può variare di molto rispetto agli altri. È quindi di fondamentale importanza saper riconoscere il tipo di tumore per fare una stadiazione e una prognosi accurate e personalizzate. È difficile stimare l'eterogeneità del tumore con un campionamento casuale dello stesso o con una biopsia perchè il campione potrebbe non rendere conto della variabilità genica all'interno del tumore stesso ([12]). Inoltre la biopsia è un metodo invasivo e che non può essere ripetuto entro un breve arco temporale.

Per questo è preferibile ricorrere a strumenti come le immagini CT, MR e PET. La valutazione dell'eterogeneità del tumore viene spesso fatta dai radiologi a partire dalla semplice visualizzazione delle immagini mediche. Le descrizioni del fenotipo del tumore risultano quindi qualitative e soggettive. Inoltre ci sono caratteristiche che possono aiutare a classificare il tumore non visibili ad occhio nudo.

La radiomica può identificare i diversi fenotipi tumorali per effettuare una prognosi accurata. Ciò può far sì che questo metodo venga in futuro utilizzato nella routine clinica per personalizzare i trattamenti.

Molti studi sottolineano l'esistenza di un legame tra le feature delle immagini mediche di diversi tipi di cancro e il genoma del cancro stesso([13]). Dato che si pensa che la variabilità fenotipica di certi tipi di cancro (come la risposta alla radioterapia) dipenda dalla sua espressione genica, diventa importante trovare un metodo per indagarla con metodi non invasivi. La radiogenomica risponde a questa esigenza: si tratta dell'identificazione di feature radiomiche corrispondenti ad alleli che causano una differenza nel fenotipo. Conoscere l'espressione genica può migliorare grandemente la diagnosi, la prognosi e la previsione sulla risposta al trattamento.

2.1.1 Il processo radiomico

Per giungere alla formulazione di un modello basato sulla radiomica si passa attraverso diverse fasi di elaborazione dei dati. Le descriviamo in seguito.

Acquisizione delle immagini

Le immagini mediche analizzate sono acquisite tramite scan CT, PET e MR . Si lamenta in molti studi ([11], [10]) la mancanza di un protocollo di acquisizione delle immagini

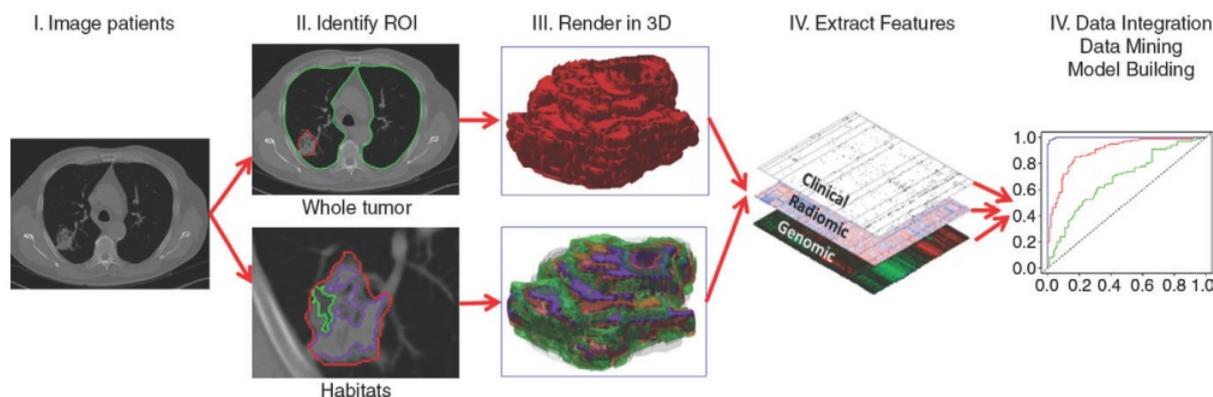


Figura 2.1: Il processo radiomico.
Da [11]

standardizzato, che è l'unico modo per rendere riproducibili gli esperimenti di radiomica.

Dato che l'estrazione delle feature deve essere fatta sulla regione in cui è presente il cancro, e non sull'intera immagine, deve essere individuato il cosiddetto *volume d'interesse*. All'interno del volume d'interesse possono essere presenti sotto-regioni con caratteristiche dell'imaging differenti (che possono essere indicatori di eterogeneità del tumore). Queste sotto-regioni sono chiamate *habitat* del tumore.

L'immagine acquisita, in alcuni studi, è soggetta a una pre-elaborazione per eliminare eventuale rumore, applicando ad esempio filtri di smoothing.

Segmentazione dell'immagine

La segmentazione dell'immagine è una fase importante perchè molte delle feature radiomiche dipendono dal metodo di delineazione del contorno ([14]). La segmentazione può avvenire manualmente, ma questo metodo può essere dispendioso in termini di tempo e variabile a seconda del soggetto che lo esegue.

Per garantire la riproducibilità, quindi, nell'ambito della radiomica è preferibile la segmentazione automatica.

Estrazione delle feature

La radiomica è incentrata sull'estrazione delle feature per la descrizione quantitativa delle caratteristiche del volume d'interesse.

Le feature si possono suddividere in *semantiche* e *agnostiche* ([11]). Le prime sono feature che vengono utilizzate nell'ambito della radiologia per caratterizzare una certa regione d'interesse(ad esempio dimensione, forma, necrosi, vascolarità,...), mentre le feature agnostiche sono caratteristiche dell'immagine estratte matematicamente che non hanno un corrispettivo nel lessico radiologico(ad esempio curtosi e simmetria degli istogrammi, texture di Haralik, trasformata wavelet e di Laplace).

Queste ultime possono essere classificate come feature statistiche di primo ordine, di secondo o di ordini più alti. Le statistiche di primo ordine sono le feature che descrivono certe caratteristiche della distribuzione delle intensità dei voxel dell'immagine (come la media, il massimo, il minimo, etc.). Le statistiche di secondo ordine sono feature che descrivono la texture dell'immagine, quindi danno informazioni sulla disposizione spaziale dell'intensità dei voxel. Si basano sull'uso di matrici che evidenziano quest'informazione. Ne sono un esempio la matrice di co-occorrenza del livello di grigio, che conta le coppie di voxel con certi valori di grigio poste a una certa distanza e direzione tra loro, o la matrice di differenza del tono di grigio tra voxel vicini, dal nome autoesplicativo, che dà una misura della percezione che ha un umano della complessità della texture. Quelle dei ordini più alti danno informazioni sulla presenza di pattern.

Analisi esplorativa dei dati e selezione delle feature

In questa fase ci si propone di fare un'analisi statistica delle feature estratte, investigando le relazioni tra queste ultime. Infatti possono essere individuati cluster di feature altamente correlate tra loro: in questo caso, è conveniente scegliere una feature rappresentante del gruppo invece di utilizzarle tutte per la creazione del modello. Questa selezione riduce la dimensionalità del problema e la possibilità di overfitting.

A questo scopo, è inoltre necessario eliminare dal modello le feature che mancano di robustezza nei confronti della variabilità delle immagini.

Ci sono diversi metodi per la selezione delle feature([16]), che prevedono di classificare queste ultime in base all'assegnazione di un punteggio. In particolare i metodi di selezione possono essere di tipo univariato o multivariato. Nel primo, l'assegnazione del punteggio dipende dalla rilevanza della singola feature, ovvero dalla misura di quanto la feature da sola è associata a un possibile esito clinico. Nei metodi multivariati il punteggio si assegna tenendo conto non solo della rilevanza, ma anche della ridondanza, che quantifica

quanto la feature è superflua nella previsione dell'esito clinico rispetto al set di feature già selezionate.

Modellizzazione

Le feature estratte costituiscono un grande database da cui è possibile ricavare modelli predittivi degli esiti clinici. Questi modelli sono costruiti con tecniche di machine learning, che rendono il processo automatico e accurato.

Nel caso della radiomica applicata all'oncologia, si possono usare le feature per creare modelli di valutazione del rischio di metastasi o ricorrenze locoregionali o in generale per stimare la probabilità di sopravvivenza al cancro ([15]).

I modelli radiomici possono basarsi non solo sulle feature estratte ma anche su variabili cliniche come l'età del paziente e lo stadio del cancro. L'aggiunta di queste variabili può rendere il modello più accurato([15]).

Le tecniche di machine learning utilizzate negli studi di radiomica sono diverse([10], [16]). Quando lo studio vuole raggruppare le varie feature in cluster per, ad esempio, rilevare quelli di questi sono associati a performance prognostiche migliori o alla probabilità di sopravvivenza, allora vengono usati metodi di machine learning con apprendimento non supervisionato. Più spesso però vengono usati algoritmi di classificazione, che hanno quindi un tipo di apprendimento supervisionato. Il classificatore elabora una funzione delle feature di input che manda in output la probabilità di un possibile esito clinico.

Esempi di algoritmi utilizzati per creare un modello radiomico sono le reti neurali, gli alberi di decisione, le reti Bayesiane, gli algoritmi di random forest ([16]).

Verifica

Per verificare la validità del modello, il totale delle immagini mediche a disposizione viene diviso in training set, usato nella fase di addestramento dell'algoritmo, e validation set, usato appunto per testare la capacità predittiva del modello su dati mai visti.

La performance predittiva del modello è misurata usando il metodo dell'area sotto la curva ROC (AUC). La curva ROC è ottenuta plottando la frazione di veri positivi (specificità) in funzione dei falsi positivi generati dal modello considerato. L'area sotto la curva è un numero compreso tra 0 e 1. Più questa misura è vicino a 1 più il modello è specifico: infatti la AUC rappresenta la probabilità che un paziente selezionato

casualmente che presenta un determinato esito clinico sia inserito nella giusta classe dal modello con maggior probabilità rispetto a un paziente che non presenta quell'esito.

Per stimare gli intervalli di confidenza della stima della performance del modello si usa il metodo di ricampionamento bootstrap sui pazienti del training set. Questo metodo prevede di ricampionare in modo casuale il training set per diverse volte, calcolare l'intervallo di confidenza per ciascuno dei campioni casuali e, infine, stimare l'intervallo di confidenza totale come una media di quelli dei singoli campioni.

2.1.2 Successi della radiomica

Le pubblicazioni [11], [10] e [13] mettono in luce gli ambiti in cui la radiomica è stata applicata con successo.

- *Diagnosi e valutazione dell'aggressività del tumore.* Molti studi hanno mostrato che l'analisi radiomica permette di distinguere tessuti benigni e maligni aiutando nella valutazione dell'aggressività del timore. Ad esempio, è stato mostrato che le feature radiomiche hanno permesso di distinguere tra tessuti della prostata cancerogeni e non. La feature radiomica della matrice di co-occorrenza del livello di grigio ha permesso di distinguere tra le lesioni maligne e benigne nel seno con grande accuratezza. L'analisi della texture, invece, ha permesso di distinguere tra glioma di basso e alto grado di malignità.
- *Prognosi del tumore.* Diversi studi mostrano la validità delle feature radiomiche come indicatori prognostici per diversi tipi di cancro. Ad esempio feature come la dimensione, la texture, la forma e le wavelet erano predittive degli esiti clinici del cancro ai polmoni, alla testa e al collo. La distribuzione degli habitat in pazienti con glioblastoma multiforme ha permesso di distinguere tra tumori che si sviluppano rapidamente o meno.
- *Selezione del trattamento.* Ci sono fenotipi individuabili tramite l'imaging che corrispondono alla risposta ad una certa terapia.

Ad esempio si è dimostrato che certe caratteristiche delle immagini mediche sono correlate alla risposta alla dexorubicina nel carcinoma epatocellulare. L'analisi della texture, invece, ha consentito la previsione della risposta alla chemioterapia nel cancro al seno in stato avanzato.

Ad esempio ci sono studi che dimostrano che la radiosensibilità, la riossigenazione delle cellule circostanti al tumore, la frazione ipossica e la capacità di ripopolazione sono caratteristiche fondamentali per prevedere l'esito di una radioterapia. L'analisi radiomica può rivelare la presenza di queste caratteristiche.

Un'altra terapia il cui successo può essere valutato con metodi di radiomica è l'immunoterapia. Quest'ultima si rivela efficace solo se è presente nelle cellule del cancro un antigene che fa sì che vengano attaccate dalle cellule T del sistema immunitario. La presenza dell'antigene si manifesta in certe caratteristiche fenotipiche.

- *Stadiazione del tumore.* È stato mostrato che molte feature sono in grado di differenziare le malattie allo stato avanzato o precoce. Ad esempio, tre feature agnostiche si sono rivelate strettamente correlate agli stadi T (che quantifica la dimensione del tumore e può assumere valori da 1 a 4) e N (che quantifica il numero di linfonodi coinvolti nella malattia) nel tumore esofageo. Feature radiomiche di texture sono state usate con successo per effettuare la stadiazione, sia in fase precoce che avanzata, nel cancro alla cervice.
- *Individuazione della zona dove effettuare la biopsia.* Dato che i tumori sono, al loro interno, eterogenei, quando si effettua la biopsia si rischia di campionare un pezzo di tessuto che porta poche informazioni. Le feature radiomiche possono aiutare ad individuare le zone del tumore che più probabilmente contengono informazioni prognostiche e diagnostiche.

2.1.3 I problemi attuali della radiomica

Ogni fase del processo radiomico presenta dei problemi che devono essere affrontati per rendere la radiomica parte della routine clinica. I principali problemi sono esposti da [17], [13] e [11]. .

- *Riproducibilità degli studi.* Gli studi che usano la radiomica sono di difficile riproducibilità a causa di diversi fattori come la complessità tecnica, l'overfitting dei dati su uno specifico campione, la comunicazione incompleta dei dati. Dovrebbero perciò essere elaborate delle linee guida standardizzate che diano indicazioni su come acquisire ed analizzare i dati.

- *Standardizzazione dell'acquisizione delle immagini.* È stata dimostrata una grande variabilità delle feature radiomiche a fronte di diversi modi di acquisizione delle immagini, come la posizione del paziente, la dimensione dei pixel, la sottigliezza della sezione, gli algoritmi di ricostruzione. Questo rende difficile confrontare i risultati ottenuti con macchine diverse. È perciò necessaria la standardizzazione dell'acquisizione delle immagini.
- *Segmentazione.* Per quanto riguarda la segmentazione, questa è forse la fase che presenta più problemi. Molti tumori hanno margini indistinti, quindi è difficile la riproducibilità della loro delineazione. Spesso questa viene eseguita manualmente, ma questo metodo è soggetto ad una variabilità tra i soggetti ed è dispersivo in termini di tempo. Anche se sono stati sviluppati diversi algoritmi di segmentazione automatica e semi-automatica questi non sono universalmente applicabili a tutte le immagini mediche. È quindi necessario trovare una tecnica di segmentazione efficace e riproducibile.
- *Condivisione dei dati per la formulazione delle ipotesi e la verifica.* La presenza di database in cui reperire immagini mediche dalla provenienza eterogenea è necessaria per essere usata sia come training set, in modo tale da raggiungere un potere statistico sufficiente, sia come validazione. Infatti la verifica del modello deve auspicabilmente essere fatta su pazienti provenienti da coorti indipendenti, preferibilmente da diversi ospedali.

A questo fine, le immagini contenute nel database devono essere fornite di annotazioni, dati sull'esito clinico, metodo di acquisizione, scanner e altre informazioni.

2.2 Le reti convoluzionali applicate alla radiomica

I metodi computerizzati di analisi delle immagini mediche tradizionali, con feature definite dai radiologi, hanno mostrato la capacità di trovare biomarker (cioè feature) utili per l'oncologia e non solo ([18]), per effettuare diagnosi, facilitare l'identificazione del migliore trattamento, etc.

Negli ultimi anni è stato esplorato un nuovo modo di condurre l'analisi delle immagini mediche, che ha provato di costituire lo stato dell'arte in diverse applicazioni, portando a modelli accurati: il deep learning.

Il deep learning presenta diversi vantaggi rispetto all'estrazione manuale delle feature. ([18]) Per prima cosa, dato che il deep learning impara le feature direttamente a partire dai dati di training, questo compito (che può essere molto dispendioso in termini di tempo, soprattutto per il design delle feature di ordine più alto) non grava sui ricercatori. Questo fatto permette anche l'estrazione delle feature indipendentemente dalla presenza di un esperto (radiologo). Inoltre le feature imparate direttamente dalle immagini possono essere anche più efficienti per l'elaborazione del modello, scoprendo pattern complessi che non sono evidenti all'occhio umano ed estraendo feature più astratte (di ordine più alto). Quest'ultimo vantaggio è permesso dalla struttura a strati delle reti neurali profonde, in cui ogni strato corrisponde a un più alto livello di astrazione delle feature, composte a partire da quelle dello strato precedente. Un altro vantaggio fondamentale è che tre fasi diverse del processo radiomico, l'estrazione delle feature, la selezione e la classificazione, possono essere effettuate nello stesso algoritmo, tramite una rete neurale profonda.

Il deep learning presenta anche alcuni svantaggi ([19]): l'addestramento di una rete neurale profonda richiede un training set molto ampio e dotato di annotazioni sull'esito clinico e altre informazioni, che è difficile da reperire nell'ambito medico. Inoltre, data la scarsa reperibilità di immagini mediche, si rischia di andare incontro all'overfitting del modello. Per questo motivo spesso si sfrutta la trasferibilità delle feature delle immagini usando reti pre-addestrate su ImageNet o altri database.

Sono stati svolti diversi studi che sfruttano le CNN per l'estrazione delle feature dalle immagini mediche, al fine di elaborare un modello. In particolare, nell'ambito della computer-aided detection (CAD), si è dimostrato che per l'identificazione delle lesioni le CNN in molti casi costituiscono lo stato dell'arte ([19]). Ad esempio, i sistemi

CAD che usano le CNN per l'identificazione di polipi nel colon, metastasi nella spina dorsale e linfonodi allargati hanno una sensitività del 13-34% superiore rispetto ai metodi tradizionali. L'identificazione di noduli polmonari a partire da immagini CT tramite le CNN pre-addestrate ha raggiunto lo stato dell'arte, con una sensitività dell'85%.([20]).

È utile notare, in quanto si tratta di una fase del processo radiomico, che le CNN sono state usate anche per la segmentazione delle immagini mediche., ottenendo ottimi risultati([19]).

Il successo del deep learning nell'estrazione delle feature per i sistemi CAD ha incentivato l'esplorazione di questo metodo anche in radiomica ([21]). Infatti, sebbene sia i sistemi CAD che i modelli radiomici si basino su un procedimento simile (acquisizione delle immagini, segmentazione, estrazione e selezione delle feature, modellizzazione), la radiomica si propone di dare informazioni sulla prognosi, mentre i sistemi CAD servono solo per la diagnosi e l'identificazione di lesioni.

Nell'ambito dell'oncologia, tre studi possono essere portati come esempio.

Li *et al.* hanno condotto uno studio ([22]) in cui hanno elaborato un metodo radiomico basato sul deep learning, in particolare sulle CNN, per predire la mutazione dello stato dell'enzima IDH1(isocitrato deidrogenasi), che è altamente correlata alla comparsa del glioma, il più diffuso tra i tumori maligni al cervello.

Lo stato dell'enzima IDH1 infatti può essere usato per personalizzare il trattamento in quanto tendenzialmente i pazienti con IDH1 mutato hanno una prognosi positiva.

Lo studio è stato effettuato analizzando le immagini MR di 119 pazienti a cui è stato diagnosticato il glioma. È stata usata una rete convoluzionale sia per la segmentazione dell'immagine che per distinguere i tumori che presentano una mutazione dell'enzima IDH1 da quelli senza mutazione. La CNN usata è dotata di 6 layer convoluzionali, i cui neuroni hanno una ReLU come funzione di attivazione.

Per il compito di segmentazione è stato usato l'output dell'ultimo layer, quello fully connented, che classificava i pixel come appartenenti o non appartenenti alla regione del tumore. È stato notato dai ricercatori che aumentando il numero di neuroni nel layer fully connected la segmentazione risultava essere più accurata. Perciò è stata scelta una CNN con 4096 neuroni nell'ultimo layer. La rete è stata addestrata tramite un algoritmo di Stochastic Gradient Descent che usava la backpropagation per il calcolo del gradiente. La rete si è mostrata in grado di segmentare l'immagine con accuratezza.

La CNN è stata usata anche per il compito di estrazione delle feature. Le feature sono state generate dalle mappe delle feature dell'ultimo layer convoluzionale. In particolare sono state usate come feature del modello le statistiche di primo e secondo ordine di kernel Gaussiani applicati alle mappe dei filtri profondi.

Sono state selezionate le feature più significative e non ridondanti con diversi metodi, per poi classificare i vari casi con una SVM (support-vector machine).

L'accuratezza del classificatore è stata stimata con diversi metodi tra cui il metodo AUC. Facendo un confronto tra la performance del modello basato sul deep learning e di quello basato su feature estratte a mano, ne risulta che il primo ha un'area sotto la curva (AUC) del 92%, mentre il secondo dell'86%.

Gli autori mettono in luce diversi vantaggi nell'utilizzare un approccio alla radiomica basato sul deep learning. In primis il processo radiomico può essere reso completamente automatizzato segmentando l'immagine e successivamente estraendone le feature con una rete convoluzionale. I secondo luogo, la grande capacità della rete nel riconoscimento del tumore fa sì che le feature estratte dall'ultimo layer siano altamente indicative dello stato dell>IDH1. Infine, la CNN può essere adattata al problema specifico, quindi le caratteristiche estratte da quest'ultima sono anch'esse specifiche. Questo si contrappone alle feature radiomiche classiche: queste ultime sono raggruppate in un set fisso che viene usato per gli studi su qualsiasi tipo di tumore.

Lao *et al.* ([23]) hanno condotto uno studio con lo scopo di determinare se le feature profonde estratte da una CNN addestrata con il transfer-learning possano generare un modello radiomico per la previsione della sopravvivenza al glioblastoma multiforme, un tumore maligno alle cellule gliali, che costituiscono una parte del sistema nervoso.

Lo studio è stato condotto sulla base delle immagini MR di 112 pazienti. Le immagini sono state pre-processate e successivamente è stata effettuata una segmentazione manuale della regione del tumore (da parte di un neurochirurgo). La zona segmentata è stata usata per l'estrazione delle feature.

Sono state estratte sia feature radiomiche tradizionali sia feature profonde, elaborate dalla rete convoluzionale. Le prime possono essere suddivise in feature di forma, di intensità e di texture e sono un totale di 1403. Le feature profonde sono state estratte da una CNN pre-addestrata sul dataset ImageNet ILSVRC 2012, contenente 5 layer convoluzionali e 3 layer fully connected. È stato scelto di utilizzare una rete pre-addestrata poiché

il numero di immagini mediche a disposizione, utilizzabili come data set, era limitato. Sono state estratte un totale di 98304 feature profonde dai layer fully connected numero 6 e 7.

Dato l'alto numero delle feature estratte, è stata operata una selezione in più fasi, che prediligeva le feature robuste rispetto a variazioni nella segmentazione, informative, dall'alto valore prognostico e non ridondanti. Le feature selezionate sono 6, e tutte estratte con la CNN.

Le feature selezionate sono state usate per costruire il modello, che consiste nella classificazione dei pazienti come appartenenti al gruppo di rischio alto o basso, dove queste categorie identificano pazienti con tempo di sopravvivenza significativamente diverso. In particolare, è stato calcolato in punteggio delle feature radiomiche come la combinazione lineare dei valori delle feature con opportuni coefficienti. Qualora questo punteggio superi un certo valore di soglia, il paziente viene assegnato al gruppo di rischio alto, in caso contrario al basso.

La performance del modello è valutata tramite il metodo AUC, che vale 0,731 per il training set, 0,710 per il validation set, mostrando che il modello, oltre ad essere accurato, ha un'ottima capacità di generalizzazione. La performance è stata confrontata con quella degli indicatori che vengono usati di solito per fare una previsione della probabilità di sopravvivenza, che sono l'età del paziente e la scala di Karnofsky (che tiene conto della qualità della vita del paziente). Nessuno dei due ha stratificato efficacemente i pazienti nei due gruppi di rischio.

È stato infine creato un modello che combina le feature radiomiche ai fattori di rischio sopra citati, che ha ottenuto un valore di AUC pari a 0,739. Il modello è quindi dotato di un buon potere predittivo.

Gli autori sottolineano che il fatto che la selezione delle feature abbia escluso quelle estratte a mano indica che le feature profonde sono più informative, in quanto riflettono pattern delle immagini di ordini più alti rispetto alle prime, registrando una maggiore eterogeneità nell'immagine.

Cha K. *et al.* hanno condotto uno studio il cui scopo è di stimare la risposta alla chemioterapia neoadiuvante nel cancro alla vescica usando un modello radiomico basato sul deep learning.

Lo studio è stato condotto sulle immagini CT, acquisite prima e dopo il trattamento,

di 123 pazienti, di cui 82 usati come training set e 41 come test set.

Sono stati elaborati tre modelli, che sono stati confrontati tra loro e con la performance di due radiologi esperti. Il primo (DLR-CNN) è quello che fa uso della CNN. L'immagine data in input alla rete neurale è costituita dalle immagini appaiate delle regioni di interesse, ottenute grazie a una segmentazione automatica della lesione. La CNN è usata per l'estrazione e per la classificazione. Le possibili classi sono due: pazienti che presentano residui del cancro dopo il trattamento e pazienti che non li presentano. La CNN utilizzata è costituita da 5 layer, di cui due convoluzionali, due parzialmente connessi e uno totalmente connesso. Il tipo di apprendimento è supervisionato, in quanto il compito è di classificazione. Il secondo modello (RF-SL) consiste nell'estrazione di 91 feature radiomiche classiche a partire da immagini segmentate della lesione, corrispondenti a caratteristiche morfologiche, di livelli di grigio e di texture delle immagini. Sono poi state selezionate 5 features ed è stata fatta una classificazione dei diversi casi con un algoritmo random forest. Il terzo modello (RF-ROI) consiste nell'estrazione di feature classiche dalle immagini delle regioni di interesse prima e dopo il trattamento appaiate. 5 features sono poi selezionate e la classificazione è effettuata tramite un algoritmo random forest.

La stima della performance tramite il metodo della curva AUC ha dato come risultati 0.73 per il modello DLR-CNN, 0.77 per RF-SL, 0.69 per RF-ROI. La performance dei due radiologi è stata valutata come 0.76 per il primo e 0.77 per il secondo.

I metodi DLR-CNN e RF-SL hanno avuto performance comparabili e simili a quelle dei radiologi esperti, ma non ottimali. Secondo gli autori, questo può essere dovuto al basso numero di esempi nel data set, che ha impedito alla rete e al classificatore random forest di essere addestrati al meglio.

La radiomica può essere utilizzata anche in ambito non oncologico. In particolare, è stato condotto uno studio, che presentiamo di seguito, con l'intento di trovare trattamenti preventivi personalizzati. Lo studio ha usato con successo il metodo radiomico basato su algoritmi di deep learning.

Oakden-Rayner *et al.* hanno condotto uno studio con lo scopo di individuare biomarker che indichino lo stato generale di salute dell'individuo e la presenza di malattie croniche legate all'età, in modo tale da prevedere la mortalità entro 5 anni, ed eventualmente attuare trattamenti di prevenzione adatti.

Lo studio è stato condotto sulle immagini CT del petto di 48 pazienti sopra ai 60 anni, di cui 24 sono sopravvissuti oltre 5 anni dopo l'acquisizione dell'immagine, i rimanenti no. In primis l'immagine del torace è stata segmentata in modo semiautomatico separando muscolo, grasso corporeo, aorta, colonna vertebrale, cuore e polmoni.

Sono stati elaborati due modelli per lo studio, che sono stati confrontati tra di loro e con la performance delle feature cliniche che sono di solito usate per fare previsioni sulla longevità. Il primo modello fa uso di feature radiomiche estratte a mano da radiologi esperti, che si riferiscono a caratteristiche di intensità e di texture dell'immagine e caratteristiche che riflettono la conoscenza preesistente dei biomarker nell'imaging. Le feature sono state selezionate (passando da un numero di 15957 a 5). In seguito è avvenuta la classificazione dei casi tramite un algoritmo di random forest: le classi possibili erano pazienti ad alto e a basso rischio di mortalità. I pazienti ad alto rischio hanno una probabilità di morire nei 5 anni successivi all'acquisizione di circa l'87%, quelli a basso rischio dell'8%.

Il secondo metodo prevede l'utilizzo di una CNN costituita da 3 layer convoluzionali e un layer fully connected. L'operazione di pooling era effettuata dopo i primi due layer convoluzionali. In questo caso, la CNN estraeva le feature e, allo stesso tempo, classificava le immagini.

La performance dei due modelli è stata stimata usando il metodo della cross validation, usando 40 pazienti come training set e 8 come test set. La stima tramite il metodo AUC ha dato come risultati 0,677 per il metodo della CNN, 0,646 per le feature estratte a mano, mentre la classificazione fatta con le caratteristiche cliniche tradizionali ha un'accuratezza di 0,79.

La capacità di classificazione di entrambi i metodi è buona anche se lo studio è soggetto a delle limitazioni importanti. In primo luogo, il numero di casi nel training set è abbastanza esiguo. Questo è dovuto alla scarsa reperibilità di immagini di questo tipo e alla quantità proibitiva di tempo impiegato nella segmentazione.

Gli autori sottolineano che il metodo che usa il deep learning è almeno tanto efficace quanto quello che usa le feature estratte a mano. Inoltre presenta il vantaggio che le feature vengono estratte automaticamente e sono adattate al compito specifico. Questo fa sì che tramite questa tecnica si possano scoprire biomarker mai presi in considerazione.

2.3 Conclusioni

In questo lavoro, dopo un'introduzione sul funzionamento degli algoritmi di deep learning e, in particolare, delle reti neurali convoluzionali, abbiamo mostrato come queste possono essere applicate alla radiomica. Abbiamo portato come esempi studi nell'ambito oncologico che hanno usato le CNN in diversi modi: in un caso si per la segmentazione sia per l'estrazione delle feature ([22]); una CNN pre-addestrata per l'estrazione e la classificazione([23]); una CNN non pre-addestrata per l'estrazione e la classificazione([24]). Abbiamo inoltre presentato un'applicazione delle CNN a uno studio di radiomica non in ambito oncologico ma a scopo di prevenzione, in cui la rete veniva usata per l'estrazione e la classificazione([25]).

In questi studi si mostra come le CNN applicate alla radiomica abbiano un grande potenziale, ma anche che devono essere superate certe limitazioni come il basso numero di esempi presenti nel training set.

Bibliografia

- [1] Goodfellow I., Bengio Y., Courville A., *Deep Learning*, MIT Press, 2016.
- [2] LeCun Y., Bengio Y., Hinton G., *Deep Learning*, Nature, Vol. 521, No. 7553., 2015.
- [3] Zeiler M. D., Fergus R., *Visualizing and Understanding Convolutional Networks*, EECV, 2014.
- [4] Lee J. *et al.*, *Deep Learning in medical imaging: General Overview*, Korean J Radiol., 2017.
- [5] Namatēvs I., *Deep Convolutional Neural Networks: Structure, Feature Extraction and Training*, Informational technology and Management Science, 2017.
- [6] Bhandare A. *et al.*, *Applications of Convolutional Neural Networks* , International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2016.
- [7] Shen D., Wu G., Suk H., *Deep Learning in Medical Image Analysis* Annu. Rev. Biomed. Eng., 2017.
- [8] Doi K., *Computer-Aided Diagnosis in Medical Imaging: Historical Review, Current Status and Future Potential*, Comput Med Imaging Graph., 2007.
- [9] Jalalian A. *et al.*, *Foundation and methodologies in computer-aided diagnosis systems for breast cancer detection*, EXCLI Journal, 2017.
- [10] Lambin P., *Radiomics: the bridge between medical imaging and personalized medicine*, Nature Reviews Clinical Oncology, Vol. 14, 2017.
- [11] Gillies R. J., Kinahan P. E., Hricak H., *Images Are More than Pictures, They Are Data*, Radiology, Vol. 278 (2), 2016.

- [12] Davnall F. *et al.*, *Assessment of tumor heterogeneity: an emerging imaging tool for clinical practice?*, Insights Imaging, 2012.
- [13] Yip S., Aerts H. J.W.L., *Applications and limitations of radiomics*, Phys Med Biol., 2016.
- [14] Larue R.T.H.M., Defraene G., De Ruyscher D., Lambin P., Van Elmpt W., *Quantitative radiomics studies for tissue characterization: a review of technology and methodological procedures*, Br J Radiol, 2017.
- [15] Vallières M. *et al.*, *Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer*, Sci Rep., 2017.
- [16] Parmar C., Grossmann P., Bussink J., Lambin P., Aerts H. J. W. L., *Machine Learning methods for Quantitative Radiomic Biomarkers*, Sci. Rep., 2015.
- [17] Kumar V. *et al.*, *QIN “Radiomics: The Process and the Challenges”*, Magn Reson Imaging, 2012.
- [18] Oakden-Rayner L. *et al.*, *Precision Radiology: Predicting longevity using feature engineering and deep learning methods in a radiomics framework*, Sci Rep, 2017.
- [19] Greenspan H., Van Ginneken B., Summers R.M., *Guest editorial: Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique*, IEEE transaction on medical imaging, Vol. 35 (5), 2016.
- [20] Shin H. *et al.* *Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning*, IEEE, Vol 35, 2016.
- [21] Carneiro G. *et al.*, *Automated 5-year Mortality Prediction using Deep Learning and Radiomics Features from Chest Computed Tomography*, IEEE, 2017.
- [22] Li Z. , Wang Y., Yu J., Guo Y., Cao W., *Deep Learning based Radiomics (DLR) and its usage in noninvasive IDH1 prediction for low grade glioma*, Sci Rep, 2017.
- [23] Lao J. *et al.*, *A Deep Learning-Based Radiomics Model for Prediction of Survival in Glioblastoma Multiforme*, Sci Rep, 2017.

- [24] Cha k. *et al.*, *Bladder Cancer Treatment Response Assessment in CT using Radiomics with Deep-Learning*, Sci. Rep., 2017.
- [25] Oakden-Rayner L. *et al.*, *Precision Radiology: Predicting longevity using feature engineering and deep learning methods in a radiomics framework*, Sci. Rep., 2017.