Report del progetto di Machine Learning

Ferrario Tommaso Matr. 869005 (@TommasoFerrario18)

Terzi Telemaco Matr. 865981(@Tezze2001)

Vendramini Simone Matr. 866229(@Svendra4MySelf)

3 febbraio 2024

Indice

Capitolo 1

Introduzione

Questo è un progetto per l'esame di Machine Learning del primo anno del corso di laurea magistrale in informatica dell'Università degli Studi di Milano-Bicocca.

L'intero progetto si basa sul riconoscimento della presenza di un tumore al cervello data l'immagine di una risonanza magnetica. Il dataset scelto per questo progetto è scaricabile dal seguente link.

Per il riconoscimento del tumore sono stati allenati i sueguenti modelli di machine learning:

- **Percettrone**: è stato scelto questo modello vista la sua semplicità e il suo ridotto numero di paramentri rispetto gli altri.
- **SVM**: è stato scelto questo modello vista la buona capacità teorica nel geralizzare.
- **Naive Bayes Gaussiano**: è stato scelto questo modello dal momento che è l'unico ad essere probabilistico.

La relazione è stata suddivisa nei seguenti capitoli:

- **Introduzione**: descrizione del dominio e presentazione dei modelli che verranno presi in considerazione per questo progetto.
- **Dataset**: descrizione di come è stato costruito il dataset a partire dalle immagini, ovvero come sono state ricavate le features, e analisi esplorativa.
- **Percettrone**: descrizione e analisi delle performance del percettrone.
- **SVM**: descrizione e analisi delle performance delle SVM.
- **Naive Bayes Gaussiano**: descrizione e analisi delle performance per Naive Bayes Gaussian.
- **Analisi dei risultati**: analisi comparata dei risultati tra i tre modelli considerati.
- **Conclusioni**: conclusioni sull'elaborato.

Capitolo 2

Dataset

Il dataset è stato a partire da un set di 3762 immagini ottenute dalla risonanza magnetica del cervello di 3762 persone, ettichettato manualmente da professionisti del settore nelle rispettive classi:

- **presenza del tumore**: $T = 1$
- **assenza del tumore**: $T = 0$

Il valore della label cade sotto al colonna *Class*.

Le features del dataset sono state ottenute calcolando i **momenti Hu** sulle immagini della risonanza magnetica. I momenti Hu catturano le informazioni di base sull'immagine come l'area dell'oggetto, il centroide, l'orientazione e altre proprietà.

Le feature sulle immagini si dividono in base a 2 gruppi[1]:

- **First Order Features**: forniscono informazioni legate alla distribuzione dei livelli di grigio dell'immagine. Queste features corrispondono alle statistiche descrittive calcolate sui valori di ciascun pixel dell'imamgine:
  - **media**
  - **varianza**
  - **deviazione standard**
  - **indice di asimmetria**
  - **indice di kurtosis**
- **Second Order Features**: forniscono informazioni a livello di composizione della texture dell'immagine.
  - **contrast**
  - **energy**
  - **asm**
  - **entropy**
  - **homogeneous**
  - **dissimilarity**
  - **correlation**
  - **coarseness**

Capitolo 3

Rete neurale

La precedente fase di analisi ha permesso di acquisire informazioni utili sulla struttura del dataset e di conseguenza permettere la selezione di un modello adatto a svolgere il compito di classificazione.

In questo capitolo verrà presentato uno dei tre modelli che sono stati realizzati per svolgere il compito di classificazione, ovvero la rete neurale. Nello specifico si andranno a presentare i passaggi che sono stati effettuati per la realizzazione di questo modello, partendo dalla preparazione dei dati, passando per la definizione della struttura della rete neurale, fino ad arrivare ai risultati ottenuti.

3.1 Preparazione dei dati

Prima di passare alla presentazione della rete neurale nel dettaglio, risulta necessario specificare come sono stati preparati i dati per l'addestramento della rete neurale.

La prima operazione svolta sui dati è stata un operazione di standardizzazione, ovvero per ogni feature è stata calcolata la media e la deviazione standard e questi valori sono stati utilizzati per standardizzare i dati.

Eseguendo questa operazione i dati sono stati trasformati in modo tale che la loro media sia 0 e la loro deviazione standard sia 1. Questa operazione è stata eseguita per garantire che la rete neurale non sia influenzata da valori di input con scale diverse.

La seconda operazione svolta sul dataset è stata la suddivisione in training set e test set. Questa operazione è stata fatta per poter utilizzare una parte dei dati per addestrare la rete neurale e una parte dei dati per valutare le prestazioni della rete neurale. La suddivisione del dataset è stata effettuata in modo tale che il training set contenesse il 80% dei dati, mentre il test set contenesse il 20% dei dati.

Questa operazione è stata effettuata in modo da mantenere la stessa percentuale di dati positivi e negativi in entrambi i set, con lo scopo di evitare che la rete neurale sia addestrata su un dataset sbilanciato e di conseguenza non sia in grado di generalizzare correttamente.

3.2 Struttura della rete neurale

Per definire la struttura della rete neurale sono stati utilizzati i risultati ottenuti dalla fase di analisi a cui si è aggiunta una fase di grid search. Questa fase è stata effettuata per valutare la combinazione migliore di iperparametri per la rete neurale.

Dai risultati ottenuti dalla fase di analisi e dal dominio del problema, si è scelto di utilizzare una rete con una struttura di dimensioni ridotte, in modo tale da evitare l'overfitting.

Per svolgere il compito di classificazione si è scelto di utilizzare una rete neurale feedforward, la cui struttura, a meno del layer di input e di output, è stata definita attraverso un processo di grid search.

Dalla fase di analisi è stato selezionato un sottoinsieme di feature le quali sono state utilizzate come input della rete neurale. Questo sottoinsieme è composto da 5 elementi, il che ha permesso di definire la struttura del layer di input della rete neurale. Questo primo strato è composto da 5 neuroni, in cui la funzione di attivazione è stata definita come il resto delle funzioni di attivazione.

Ottimizzazione degli iperparametri

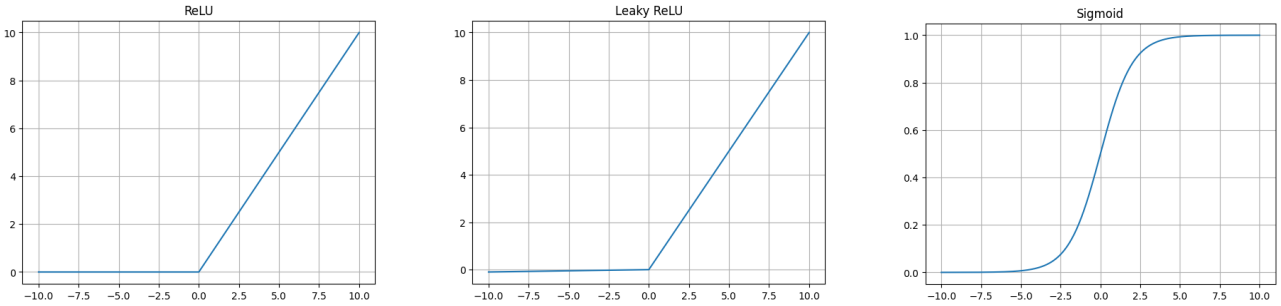
Come già accennato in precedenza, la ricerca degli iperparametri della rete neurale è stata effettuata attraverso un processo di grid search. Questo processo ha permesso di valutare le prestazioni della rete neurale al variare della funzione di attivazione, del numero di layer nascosti e del numero di neuroni per ogni layer nascosto. Questo

processo è stato effettuata attraverso una cross validation a 5 fold, prendendo in considerazione solamente i dati del training set.

Visti i risultati ottenuti nella fase di analisi e la volontà di mantenere i tempi di addestramento bassi, si è scelto di mantenere una struttura di dimensioni ridotte per la rete neurale. Per questo motivo, l'operazione di grid search è stata effettuata prendendo in considerazione un numero di neuroni per layer tra 5, 10, 50, mentre il numero di layer nascosti è stato valutato tra 1 e 2.

Per quanto riguarda la funzione di attivazione, sono state valutate le seguenti funzioni di attivazione:

- *ReLU*
- *Leaky ReLU*
- *sigmoid*



(a) ReLU (b) Leaky ReLU (c) Sigmoide

Figura 3.1: Funzioni di attivazione utilizzate nella fase di grid search

Durante il processo di grid search, per ogni modello che è stato addestrato, sono state raccolte delle informazioni relative all'accuratezza, al tempo di addestramento richiesto. In aggiunta a queste informazioni, sono stati calcolati gli intervalli di confidenza al 95% per entrambe le misure.

Ottenuti i risultati, si è proceduto con l'analisi di questi, in modo tale da definire la struttura della rete neurale. Per effettuare questa valutazione sono state utilizzate le misure precedentemente citate.

Il modello selezionato è stato scelto in base al seguente criterio: si è scelto il modello che ha ottenuto il valore più alto combinando, attraverso dei pesi, i valori relativi all'accuratezza media, al tempo di addestramento medio e agli intervalli di confidenza ottenuti. Per questa operazione si è scelto di assegnare un peso maggiore all'accuratezza media ottenuta dalla cross validation e al tempo di addestramento medio.

Nello specifico, sono stati utilizzati i seguenti pesi: 2 per l'accuratezza media, 2 per il tempo di addestramento medio e 1 per gli intervalli di confidenza. Questi pesi sono stati scelti in modo tale da dare più importanza all'accuratezza media e al tempo di addestramento medio, in quanto sono le due misure che permettono di valutare le prestazioni della rete neurale.

Per verificare la validità del modello scelto si è proceduto con il confronto tra il modello che ha ottenuto la migliore accuratezza e il modello che ha ottenuto il tempo di addestramento minore, ottenendo i risultati riportati in tabella 3.1.

Modello	**Accuratezza**	**Tempo di addestramento**
Tempo di addestramento minore	98.1%	0.96s
Accuratezza maggiore	99.5%	24.54s
Modello scelto	98.7%	2.7s

Tabella 3.1: Risultati ottenuti dalla fase di grid search

Dai valori riportati nella tabella 3.1 si può notare che il notare che il modello che è stato selezionato fornisce un buon compromesso tra accuratezza e tempo di addestramento. Nello specifico, perdendo lo 0.8% di accuratezza si è ottenuto un tempo di addestramento minore di 21.84s secondi.

Definizione della struttura della rete neurale

I risultati ottenuti dalla fase di grid search hanno permesso di definire la struttura della rete neurale. In particolare, la rete neurale è composta da 1 layer di input, 1 layer nascosto e 1 layer di output.

Il layer nascosto è composto da 50 neuroni, in cui la funzione di attivazione è la funzione Leaky ReLU.

Per concludere la descrizione della struttura della rete neurale, è necessario specificare come è composto l'ultimo layer, ovvero quello di output. Vista la natura del problema di classificazione, il layer di output è composto da un solo neurone, in cui la funzione di attivazione è la funzione sigmoide 3.1c. Questa scelta è dovuta al fatto che tale funzione restituisce un valore compreso tra 0 e 1, il che permette di interpretare l'output della rete neurale come la probabilità che l'input appartenga alla classe positiva.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

La struttura della rete neurale è riassunta nella figura 3.2.

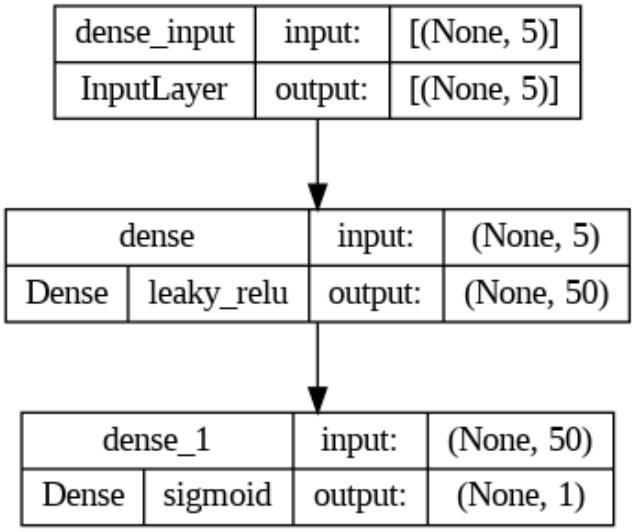


Figura 3.2: Struttura della rete neurale

Altri iperparametri della rete neurale

Oltre alla ricerca della struttura della rete neurale, la fase di grid search è stata utilizzata per valutare l'algoritmo di ottimizzazione, il numero di epoche e la dimensione del batch.

Per quanto riguarda l'algoritmo di ottimizzazione, il confronto è stato eseguito tra *Adam* e *SGD*, mentre per il numero di epoche e la dimensione del batch sono stati valutati i valori 100, 300 per il numero di epoche e 50, 100, 300 per la dimensione del batch.

I risultati ottenuti dalla fase di grid search hanno permesso di definire i valori degli iperparametri che hanno permesso di ottenere i migliori risultati. In particolare, l'algoritmo di ottimizzazione scelto è *Adam*, mentre il numero di epoche e la dimensione del batch sono stati impostati a 100 e 100 rispettivamente.

In questa fase è stato necessario definire la funzione di perdita. Si è scelta la *binary crossentropy* in quanto adatta a problemi di classificazione binaria. La scelta di questa loss è dovuta alla natura del problema di classificazione che si vuole risolvere.

3.3 Addestramento della rete neurale

La fase di addestramento della rete neurale è stata effettuata utilizzando il training set precedentemente definito. L'addestramento della rete neurale è stato effettuato utilizzando la libreria *Keras* in quanto permette di definire e addestrare reti neurali in modo intuitivo.

3.4 Risultati

Il modello addestrato in precedenza è stato valutato sui dati che compongono il test set. In particolare sono state valutate le seguenti metriche: accuratezza, precisione, richiamo e F1 score. Oltre al calcolo di queste metriche, si è deciso di realizzare la curva ROC per il modello e di rappresentare la matrice di confusione.

Prima di presentare i risultati ottenuti, è necessario specificare che essendo il dataset riferito a un ambito medico, si è deciso di aggiustare il valore di threshold per la predizione del modello. In particolare, il valore di threshold è stato impostato a 0.3, in modo tale da ridurre il numero di falsi negativi.

Fatta questa precisazione, si può procedere con la presentazione dei risultati ottenuti. In particolare, nella tabella 3.2 sono presentati i risultati ottenuti dal modello addestrato.

Metrica	**Valore**
Accuratezza	98.80 %
Precisione	99.10 %
Richiamo	98.21 %
F1 score	98.65 %

Tabella 3.2: Risultati ottenuti dal modello addestrato

I risultati ottenuti sono giustificati dal fatto che le due classi sono linearmente separabili.

In aggiunta al calcolo di queste metriche, è stata calcolata la matrice di confusione per il modello addestrato. La matrice di confusione ottenuta è riportata in figura 3.3.

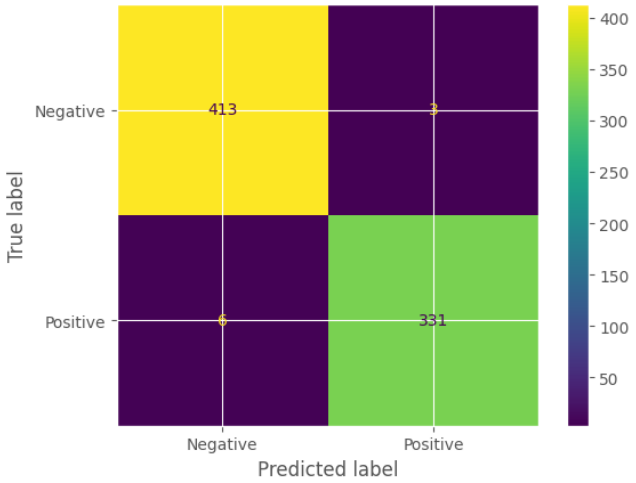


Figura 3.3: Matrice di confusione ottenuta dal modello addestrato

Infine, è stata realizzata la curva ROC per il modello addestrato. La curva ROC ottenuta è riportata in figura 3.4. Oltre alla curva ROC è stata calcolata l'area sotto la curva, la quale ha ottenuto un valore di 1.00.

Figura 3.4: Curva ROC ottenuta dal modello addestrato

Per avere una visione più chiara dei risultati ottenuti, si è deciso di effettuare una valutazione del modello attraverso 10 fold di cross validation. In questo processo ogni modello che è stato addestrato è stato valutato attraverso le metriche di accuratezza, precisione, richiamo e F1 score.

Sui risultati ottenuti da questo processo sono stati calcolati gli intervalli di confidenza al 90%. I risultati ottenuti sono riportati in tabella 3.3.

Metrica	**Valore**	**Intervallo di confidenza**
Accuratezza	98.32 %	[97.92%, 98.72%]
Precisione	98.74 %	[98.40%, 99.08%]
Richiamo	97.50 %	[96.57%, 98.43%]
F1 score	98.11 %	[97.65%, 98.57%]

Tabella 3.3: Risultati ottenuti dalla cross validation

Gli intervalli ottenuti sono stati successivamente rappresentati in un grafico riportato in figura 3.5. Questo grafico permette di avere una visione più chiara dei risultati ottenuti dalla cross validation.

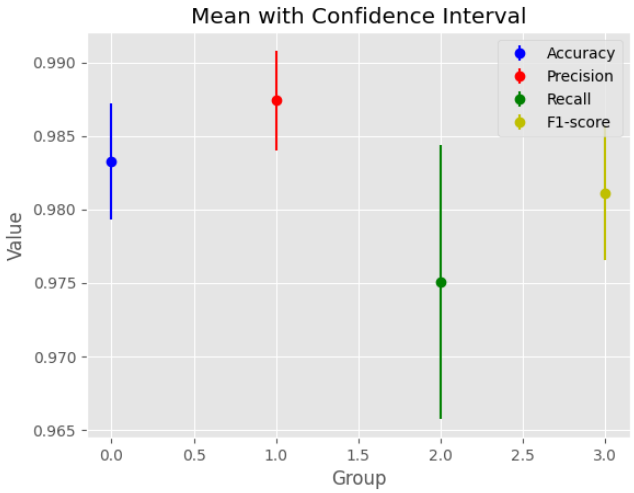


Figura 3.5: Risultati ottenuti dalla cross validation

Bibliografia

[1] Namita Aggarwal e RK Agrawal. “First and second order statistics features for classification of magnetic resonance brain images”. In: (2012).