REPORT DEL PROGETTO D'ESAME PER IL CORSO DI BIOINFORMATICA

PREDIZIONE DI VARIANTI GENETICHE PATOGENICHE DI MALATTIE MENDELIANE IN REGIONI NON CODIFICANTI DEL GENOMA UMANO CON METODI DI DEEP LEARNING.

Studente: Clerici Giulia
Matr. Nr.: 910663
Corso di Bioinformatica
Prof. Valentini Giorgio
CdL in Informatica

ANNO ACCADEMICO 2017-2018

Indice

Capitolo 1

Introduzione

1.1 Obiettivo del progetto

L'obiettivo del progetto consiste nell'implementazione di un modello di Deep Learning al fine di predire le varianti genetiche a livello di singolo nucleotide in regioni non codificanti del genoma umano che causano l'insorgenza di malattie Mendeliane.

1.2 Abstract

L'elaborato consiste nell'implementazione di un modello di Deep Learning, nello specifico di un Percettrone Multistrato (MLP), il cui scopo è quello di effettuare una classificazione binaria, ossia predire la patogenicità della variante a livello di singolo nucleotide. Il dataset fornito presenta un rilevante sbilanciamento tra esempi positivi, la cui presenza è ristretta, ed esempi negativi, fortemente presenti. Dunque è di estrema rilevanza cercare di gestire tale sbilanciamento, che, se non trattato, influenzerebbe negativamente la performance della rete neurale. A tale scopo, è stato effettuato un oversampling dei campioni positivi in modo tale da ottenere un dataset equilibrato che non influenzi negativamente i risultati. Il progetto è stato sviluppato in linguaggio Python tramite l'utilizzo delle librerie Keras, Sklearn, Numpy, e Imblearn.

Capitolo 2

Analisi del Dataset

2.1 Dataset

Il dataset fornito è suddiviso in training set, contenuto nel file Mendelian.train.tsv, e test set, contenuto nel file Mendelian.test.tsv.
Il training set è costituito da 981388 esempi, ognuno rappresentante una variante a livello di singolo nucleotide (SNV - Single Nucleotide Variant). I primi 356 esempi di tale set sono positivi, dunque patogeni, mentre i restanti esempi sono negativi, non patogeni. Gli esempi sono caratterizzati da 26 attributi numerici, che costituiscono il vettore di input della rete neurale. É possibile notare come vi sia un forte sbilanciamento dei dati a favore della classe negativa. Qualora la rete neurale venisse addestrata direttamente su tali dati, la performance ne risulterebbe fortemente influenzata, rendendo estremamente difficile la predizione della patogenicità delle SNV, rischiando così di predire erroneamente gli esempi positivi come negativi. Dunque, per ovviare a tale limitazione, sono state applicate le seguenti tecniche. Innanzitutto, viste le dimensioni elevate del set, gli esempi positivi sono stati separati dagli esempi negativi. In seguito, l'insieme degli esempi negativi, in totale 981032, è stato partizionato in 99 diversi set di uguali dimensioni, pari a 9810 istanze, ed un ultimo set la cui dimensione è pari a 9842 istanze. Ciò è stato fatto con l'obiettivo di sottoporre al modello 100 diversi training set, ognuno bilanciato tra numero di istanze positive e di istanze negative e in modo tale che le istanze negative delle diverse partizioni non si ripetano. Sono state quindi create cento partizioni dagli esempi negativi, le quali non presentano sovrapposizioni, dunque nessun esempio negativo viene ripetuto in più set. Una volta create tali partizioni, ad ognuna sono stati aggiunti i 356 esempi positivi del training set originale. Successivamente, ad ognuno di questi nuovi subset, composti da 356 esempi positivi e 9810 esempi negativi, o 9842 nel caso del centesimo subset, è stato applicato un algoritmo di oversampling in modo tale da bilanciare i dati.

L'algoritmo di oversampling utilizzato è stato l'algoritmo SMOTE, ossia Synthetic Minority Oversampling TEchnique, che consiste nel sintetizzare nuove istanze della classe di minoranza a partire da quelle già esistenti. L'algoritmo genera una nuova istanza tramite interpolazione basandosi sui k punti più vicini, ossia i k nearest neighbors. Utilizzando la classe SMOTE presente nella libreria imblearn.over_sampling con valore di k è pari a cinque, a partire dai 356 esempi positivi del training set, vengono presi in considerazione i cinque punti nearest neighbors al fine di generare, tramite interpolazione, nuovi esempi positivi simili a quelli già esistenti che vadano a bilanciare gli esempi negativi. Dunque, nel nuovo set, che andrà in input alla rete neurale, vi sarà, di volta in volta, un ugual numero di esempi di classe positiva e di esempi di classe negativa. Così facendo, si pone una soluzione allo sbilanciamento dei dati.
Una volta ottenuti questi subset bilanciati, ognuno di essi viene utilizzato come training set per la rete neurale, che viene addestrata di volta in volta su questi 100 insiemi.
Infine, per valutare la capacità di generalizzazione del modello, è stato utilizzato il test set composto da 19018 esempi, di cui i primi 40 positivi ed i restanti negativi, sempre caratterizzati dai medesimi 26 attributi che caratterizzavano le istanze del training set.

Capitolo 3

Architettura della rete neurale implementata

3.1 Percettrone Multistrato - MLP

Il modello di rete neurale implementato è un percettrone multistrato (MLP - Multi-Layer Perceptron). Questa classe di reti neurali consiste in un'evoluzione dell'algoritmo del percettrone. Mentre il percettrone consiste in una rete neurale a singolo strato ed è in grado di classificare unicamente dati linearmente separabili, il percettrone multistrato prevede un'architettura di almeno tre strati e consente di classificare dati non linearmente separabili. Il percettrone consiste nel porre in ingresso ad un neurone diversi dati costituiti da vari attributi accompagnati dai relativi pesi, insieme ad un fattore costante di bias pari a 1, anch'esso pesato. Solitamente i pesi vengono inizializzati con valori casuali molto piccoli. A partire da tali dati e i relativi pesi viene calcolata la somma pesata, il cui risultato è posto in ingresso ad una funzione di attivazione, che consiste solitamente nella funzione segno. Il risultato di tale funzione, posto in output, fornisce una classificazione binaria del dato in ingresso. Tale operazione permette tuttavia di classificare solo dati linearmente separabili.
Al fine di poter classificare dati non linearmente separabili, viene introdotto il percettrone multistrato. L'architettura di questo modello prevede un input layer, un output layer e uno o più hidden layer tra i due. L'input layer e l'output layer si mantengono uguali in entrambi i modelli. Ciò che li distingue è la presenza nel percettrone multistrato di uno o più hidden layer, ossia gli strati nascosti che si interpongono tra input e output layer. Questi hidden layer consistono in strati interni composti da più neuroni le cui uscite costituiscono gli ingressi ai neuroni degli strati successivi. Ogni strato è interamente connesso al successivo. Ogni neurone prevede una funzione di attivazione non lineare, che associa alla somma pesata degli attributi e dei relativi pesi un valore di output del neurone. In questo stesso paragrafo vedremo che

Layer (type)	Output Shape	Number Parameters
Dense_1 (Dense)	(None, 26)	702
Dense_2 (Dense)	(None, 6)	162
Dense_3 (Dense)	(None, 1)	7

Total parameters: 871
Trainable parameters: 871
Non-trainable parameters: 0

Tabella 1: Tabella rappresentate i dettagli del modello MLP avente un hidden layer a 6 neuroni.

nel presente elaborato è stata utilizzata una sigmoide come funzione di attivazione. Come algoritmo di apprendimento viene utilizzato l'algoritmo di backpropagation. Tale algoritmo prevede due fasi: una fase forward, in cui, dati gli ingressi, vengono calcolati gli output di ogni strato, ed una fase backward, in cui si calcola l'errore di output, tra quello calcolato e quello reale, e si aggiornano i pesi dello stato precedente via via fino allo strato di input. Dunque, grazie all'algoritmo di back propagation, è possibile calcolare l'errore tra predizione ed etichetta corretta e, grazie a tale errore, ricalibrare i pesi della rete.
Risulta, quindi, comprensibile come il percettrone multistrato richieda un apprendimento evidentemente più complesso e con un maggior numero di parametri rispetto all'apprendimento del percettrone, ma al contempo permetta di offrire soluzioni più adeguate a problemi più complessi.
Nel caso del progetto sviluppato, sono state implementate due versioni del percettrone multistrato. Una prima versione costituita da un input layer di 26 nodi, tanti quanti sono gli attributi dei dati costituenti il training set, un hidden layer di 6 neuroni ed un output layer di 1 singolo neurone, il cui risultato indica l'appartenenza alla classe positiva o negativa del dato, ossia la sua patogenicità.
Una seconda versione del modello implementata prevede la medesima struttura della prima versione, con l'unica differenza che l'hidden layer presente non è costituito da 6 bensì da 13 neuroni. Tale scelta è stata effettuata nella volontà di poter scorgere dei cambiamenti nella performance della rete neurale a fronte del cambiamento di numero di neuroni costituenti l'hidden layer.
Di seguito sono presentate due tabelle e due grafici che riassumono la struttura delle due varianti del percettrone multistrato implementate.

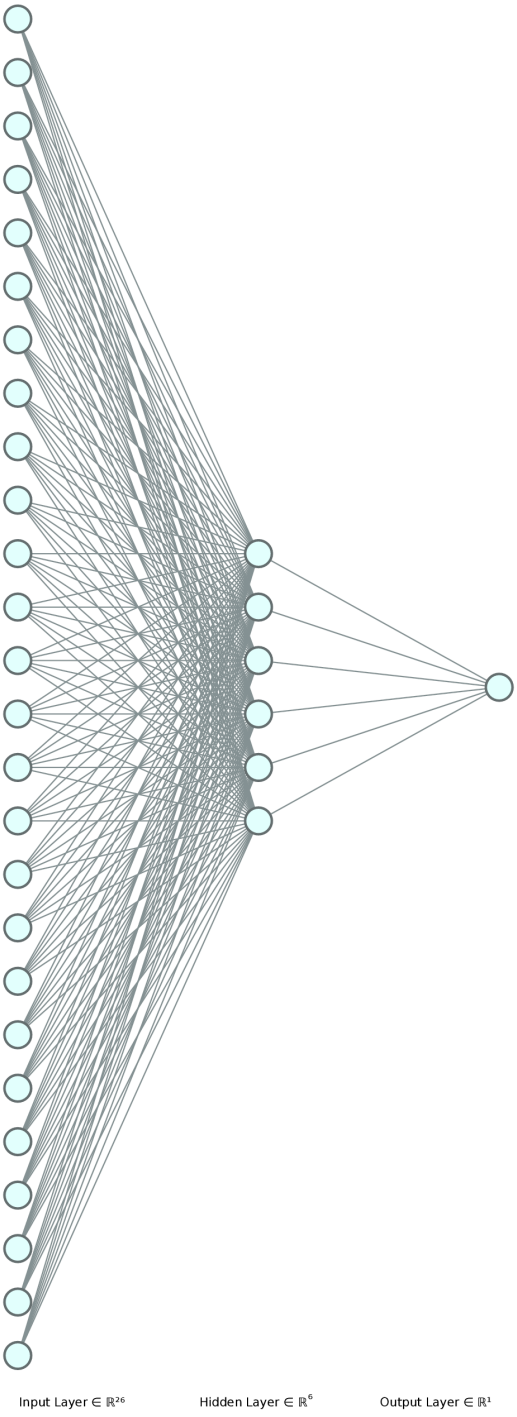

Figura 3.1.1: Architettura del percettrone multistrato, costituito da un input layer, un hidden layer a 6 neuroni e un output layer.

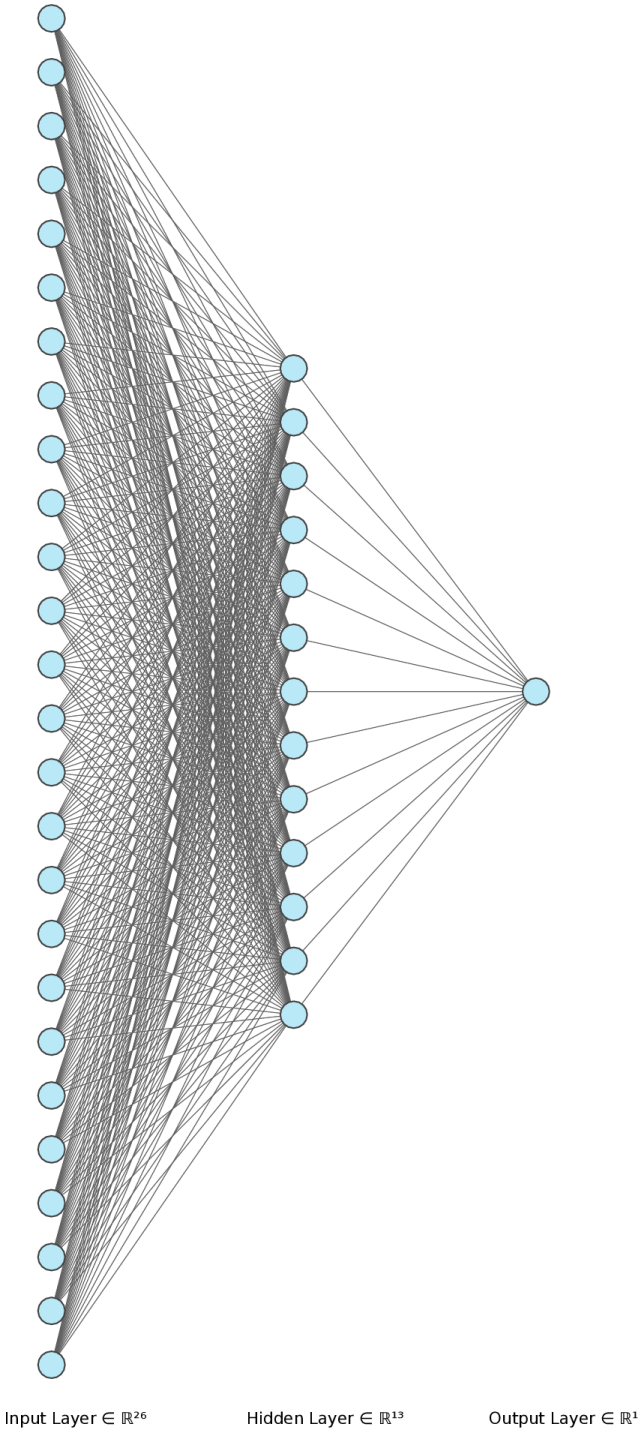

Figura 3.1.2: Architettura del percettrone multistrato, costituito da un input layer, un hidden layer a 13 neuroni e un output layer.

Layer (type)	Output Shape	Number Parameters
Dense_1 (Dense)	(None, 26)	702
Dense_2 (Dense)	(None, 13)	351
Dense_4 (Dense)	(None, 1)	14

Total parameters: 1,067
Trainable parameters: 1,067
Non-trainable parameters: 0

Tabella 2: Tabella rappresentate i dettagli del modello MLP avente un hidden layer a 13 neuroni.

Sono presentate qui di seguito le linee di codice che vanno ad implementare la struttura della rete neurale prodotta.

```
# creazione del modello della NN
model = Sequential()
model.add(Dense(26, input_dim=26, activation='sigmoid'))
# il valore di num neurons viene impostato a 6 o 13,
# a seconda della variante del modello che si vuole implementare,
# tramite l'utilizzo di una flag
flag = 0
if flag == 0:
    num_neurons = 6
else
    num_neurons = 13
model.add(Dense(num_neurons, activation='sigmoid'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='mse', optimizer='sgd', metrics=['accuracy'])
```

Nel progetto implementato la funzione di errore utilizzata è stata la funzione di errore quadratico medio (MSE - Mean Squared Error), che indica la discrepanza quadratica media fra i valori delle etichette e le predizioni effettuate, rappresentata dalla seguente formula.

$$MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2] \tag{3.1.1}$$

Dove $\hat{\theta}$ sta ad indicare l'etichetta predetta e θ l'etichetta corretta, associata al dato all'interno del training set fornito.

Per ogni layer è stata impostata come funzione di attivazione una funzione sigmoidea.

Una sigmoide è definita dalla seguente formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1.2}$$

Tale funzione sigmoidea viene sostituita alla funzione segno utilizzata nel percettrone semplice poiché più smussata e quindi ideale al fine di introdurre la non linearità nella rete neurale. La funzione di attivazione prende in input la somma pesata degli ingressi e restituisce in output un valore che va a determinare l'appartenenza del dato posto in ingresso alla classe positiva o negativa, indicante la patogenicità della variante.
Dunque, la rete neurale implementata è composta da diversi strati, di cui uno strato di input, uno di output e uno strato nascosto. Una prima versione è caratterizzata dalla presenza di 6 neuroni all'interno dell'hidden layer, mentre una seconda versione prevede la presenza nello strato nascosto di 13 neuroni invece di 6.
Inoltre, è bene precisare il numero di epoche e il valore di batch utilizzati nell'apprendimento della rete neurale.
Il valore di batch, detto batch size, è posto pari a 10 per tutte le varianti del modello implementate, mentre il numero di epoche è variato.
Il numero delle epoche definisce quante volte il training set è stato interamente sottoposto al modello in fase di apprendimento. Eseguire l'apprendimento per una sola epoca significa eseguire un training in cui il modello visualizza un'unica volta ogni dato appartenente al training set. Nel caso del presente elaborato, sono stati testati due differenti valori di epoche per entrambe le versioni della rete neurale. Per entrambe le varianti del modello è stato effettuato un apprendimento con un numero di epoche pari a 4 e successivamente pari a 100. Così facendo è stato possibile analizzare quale conseguenza avesse sulla rete un apprendimento caratterizzato da un diverso numero di epoche. I risultati verranno discussi nel seguente capitolo 4.
Così, posti in ingresso alla rete neurale selezionata i dati pesati, il modello implementato restituisce in uscita per ogni variante a singolo nucleotide un valore che permette la classificazione binaria della patogenicità della variante.

Capitolo 4

I risultati ottenuti

4.1 Metriche utilizzate

Dopo aver implementato il percettrone multistrato, eseguito la fase di apprendimento e di test, si procede ora ad analizzare i risultati ottenuti.
Ponendo in ingresso al modello implementato i dati di training come descritto nei capitoli precedenti 2 e 3, è stato possibile ottenere un percettrone multistrato la cui performance è stata misurata tramite la matrice di confusione, AUROC (*Area Under the Receiver Operating Characteristic*) e AUPRC(*Area Under the Precision Recall Curve*).
Nel caso di classificazione binaria, la precisione per una classe consiste nel numero di veri positivi diviso per la somma di veri positivi e falsi positivi, anche detto True Positive Rate (TPR).
Il richiamo per una classe è il numero di veri positivi diviso per la somma di veri positivi e falsi negativi, ossia (1 - FPR), dove FPR è il False Positive Rate, ossia il rapporto tra il numero di falsi positivi e la somma di falsi positivi e veri negativi
L'AUROC è il valore dell'area sotto la curva ROC, che è il grafico in cui i due assi rappresentano la specificità e la sensibilità, ossia rispettivamente il False Positive Rate (FPR) e il True Positive Rate (TPR). Tale grafico studia appunto i rapporti fra allarmi veri e falsi allarmi, rappresenta l'insieme delle coppie di falsi positivi e veri positivi al variare di un parametro del classificatore. Calcolando l'area sottesa alla curva ROC si è in grado di valutare la performance del classificatore nel distinguere una variante a livello di singolo nucleotide patogenica da una non patogenica. Il valore AUROC, che è compreso tra 0 ed 1, misura quanto il parametro sia in grado di distinguere tra classe positiva e negativa, quanto bene il modello sia in grado di identificare gli esempi positivi.
L'AUPRC è il valore dell'area sottesa alla curva di precisione e richiamo, ossia una curva avente sull'asse delle x il richiamo e sull'asse delle y la precisione. Tale grafico

rappresenta il rapporto che vi è tra precisione e richiamo per diversi valori del sistema prodotto.
Infine, un ulteriore metodo per valutare la qualità del classificatore è rappresentato dalla matrice di confusione. Una matrice di confusione è una matrice che riporta l'accuratezza del classificatore, rappresentando quanti veri negativi, falsi positivi, falsi negativi e veri positivi siano individuati a seguito della classificazione.
Di seguito vengono mostrati i grafici, con relative didascalie, delle diverse metriche utilizzate a fronte delle diverse varianti del modello implementato.

4.2 Risultati ottenuti con la rete neurale a 6 neuroni per 4 epoche: Matrice di Confusione, grafici AUROC e AUPRC

In questa sezione sono presentati le tabelle rappresentanti le matrici di confusione, i grafici AUROC e AUPRC valutati sul training set e sul test, il cui scopo è valutare la qualità della performance del modello a 6 neuroni e 4 epoche.
Il sistema è in grado di classificare discretamente bene gli esempi positivi. É da notare come, per quanto riguarda il test set, le varianti a livello di singolo nucleotide patogeniche etichettate dal sistema come non patogeniche costituiscono un decimo delle varianti patogeniche. Tuttavia vi è un'elevata quantità di falsi positivi. Ciò indica come diverse varianti a livello di singolo nucleotide non patogeniche siano etichettate dal modello come patogeniche. Il modello presenta, quindi, degli errori non indifferenti nella classificazione delle istanze. Ciononostante, appare evidente come la decisione di introdurre un algoritmo di sovracampionamento abbia conseguenze positive sui risultati ottenuti, che altrimenti avrebbero mostrato una ancora maggiore presenza di errori.

Matrice di Confusione

su Training Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 341	Falsi Negativi: 15	P′
	n′	Falsi Positivi: 58228	Veri Negativi: 922804	N′
	totale	P	N	

Matrice di Confusione

su Test Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 36	Falsi Negativi: 4	P′
	n′	Falsi Positivi: 873	Veri Negativi: 18105	N′
	totale	P	N	

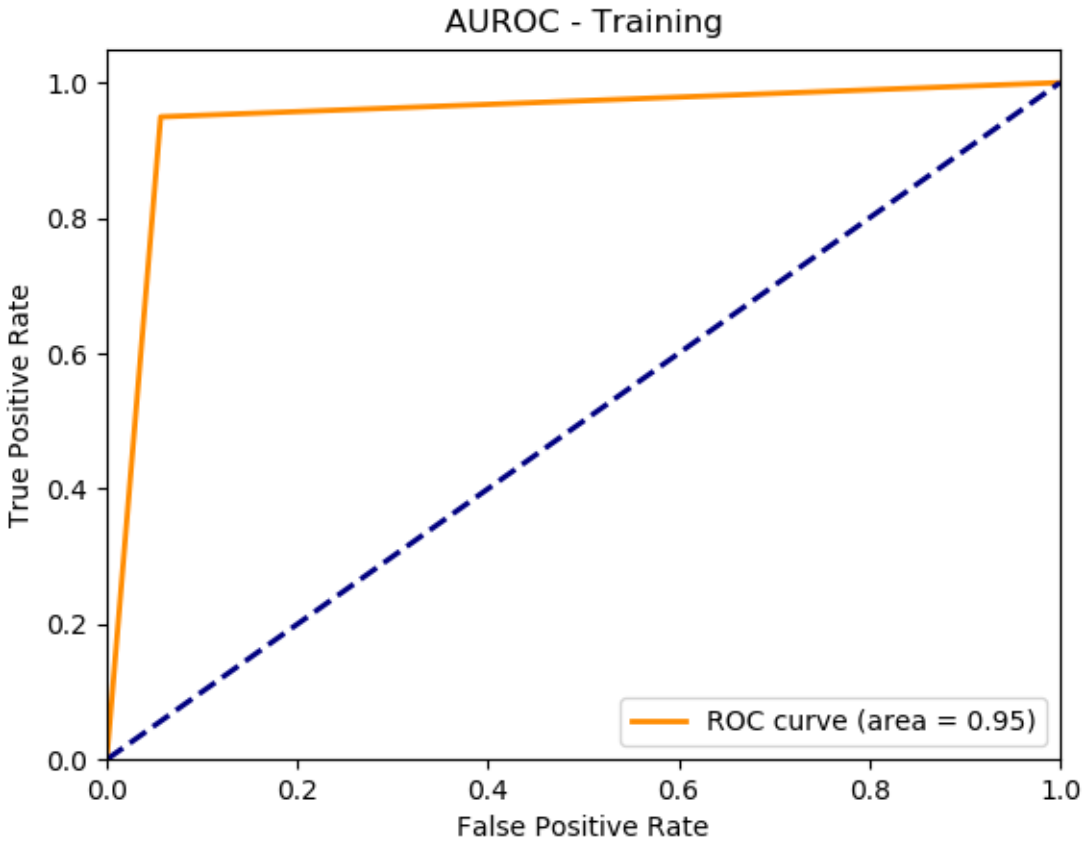

Figura 4.2.1: Grafico AUROC valutato sul training set del percettrone multistrato avente un hidden layer composto da 6 neuroni e calcolato con un valore di epoche pari a 4. Il valore di AUROC è pari a 0.95. Ciò indica come il sistema sia piuttosto efficiente nell'individuazione degli esempi positivi.

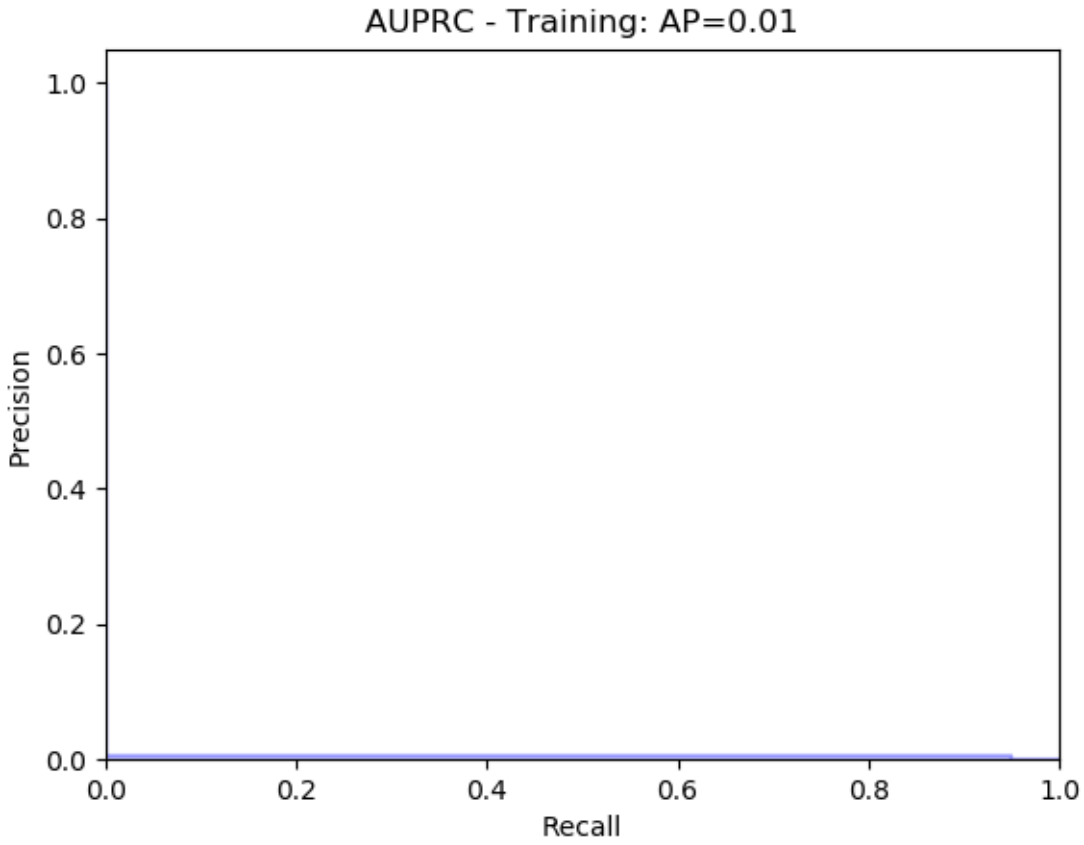

Figura 4.2.2: Grafico AUPRC valutato sul training set della rete neurale implementata, avente un hidden layer di 6 neuroni e calcolato con un valore di epoche pari a 4. É possibile notare come il grafico indichi un valore estremamente basso di AUPRC, con un valore medio di precisione di 0.01. Visualizzando il grafico è possibile notare come il modello sia caratterizzato da un valore estremamente basso di precisione, ma da un valore piuttosto alto di richiamo. Ciò indica come il modello etichetti come positivi un elevato numero di esempi negativi, riducendo drasticamente il valore di precisione. Tuttavia, un valore di richiamo elevato indica come il modello sia in grado di recuperare correttamente una buona quantità degli esempi positivi.

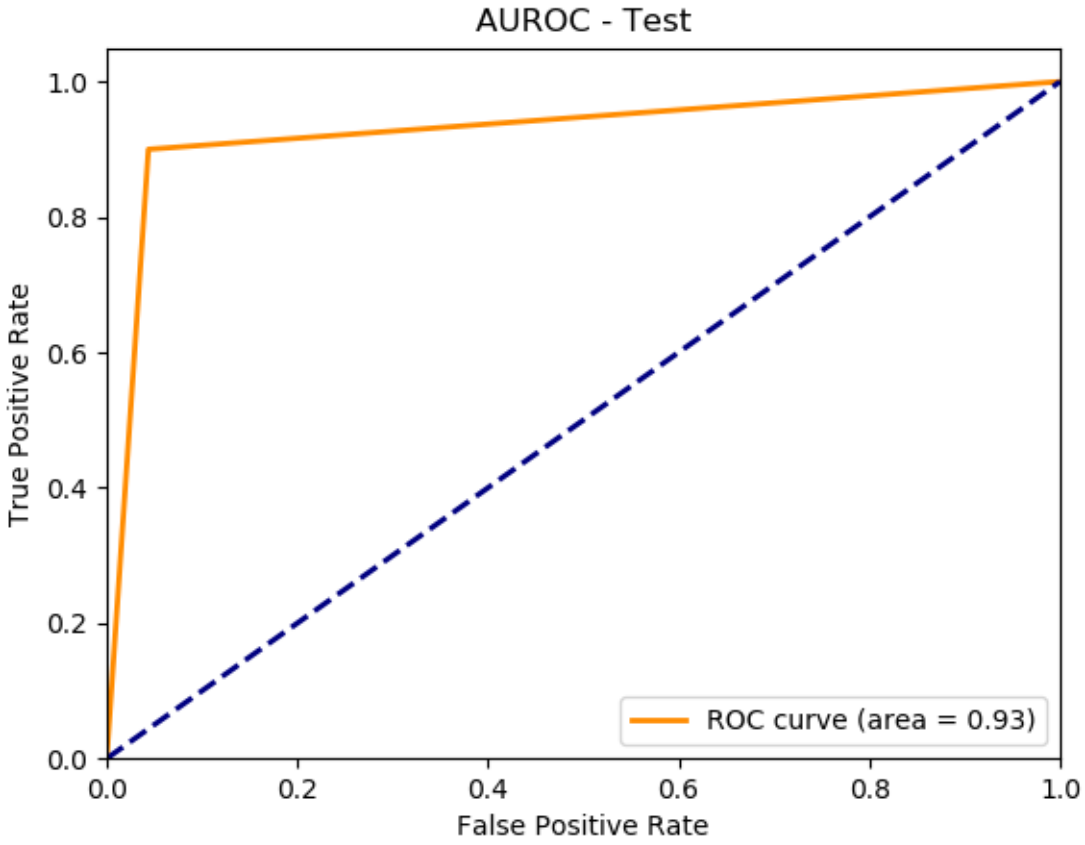

Figura 4.2.3: Grafico AUROC valutato sul test set del percettrone multistrato avente un hidden layer composto da 6 neuroni e calcolato con un valore di epoche pari a 4. Il valore di AUROC è uguale a 0.93.

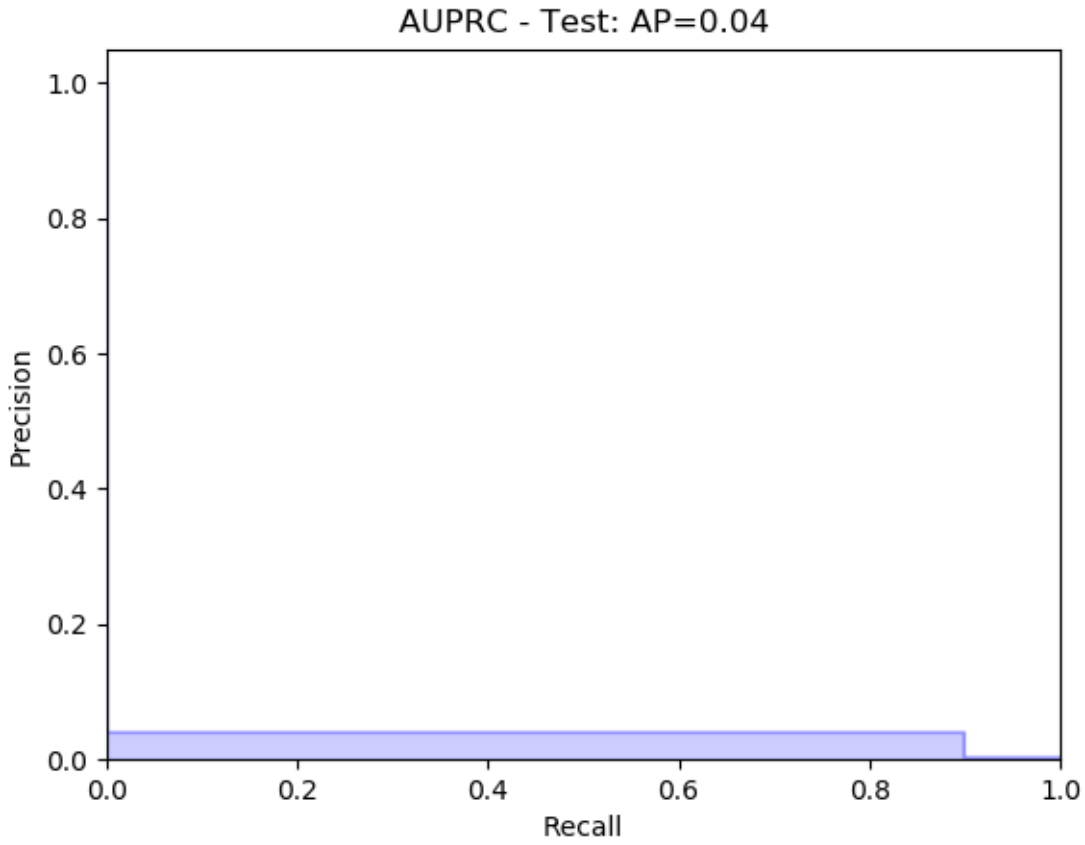

Figura 4.2.4: Grafico AUPRC valutato sul test set della rete neurale implementata, avente un hidden layer di 6 neuroni e calcolato con un valore di epoche pari a 4. É possibile notare come, sebbene subisca un lieve miglioramento, il valore AUPRC rimanga comunque estremamente basso.

4.3 Risultati ottenuti con la rete neurale a 13 neuroni per 4 epoche: Matrice di Confusione, grafici AUROC e AUPRC

In questa sezione sono presentati i grafici e le metriche di valutazione del modello con hidden layer a 13 neuroni e un valore di epoche pari a 4, calcolati sia su training set sia sul test set.
Confrontando questo modello con quello avente un hidden layer costituito da 6 neuroni, è possibile notare come vi sia un miglioramento nell'individuazione degli esempi positivi, ma al contempo vi sia un maggior numero di falsi positivi. Ciò ad indicare come la rete neurale abbia etichettato un maggior numero esempi negativi come esempi positivi.

Matrice di Confusione

su Training Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 348	Falsi Negativi: 8	P′
	n′	Falsi Positivi: 75204	Veri Negativi: 905828	N′
	totale	P	N	

Matrice di Confusione

su Test Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 38	Falsi Negativi: 2	P′
	n′	Falsi Positivi: 1532	Veri Negativi: 17446	N′
	totale	P	N	

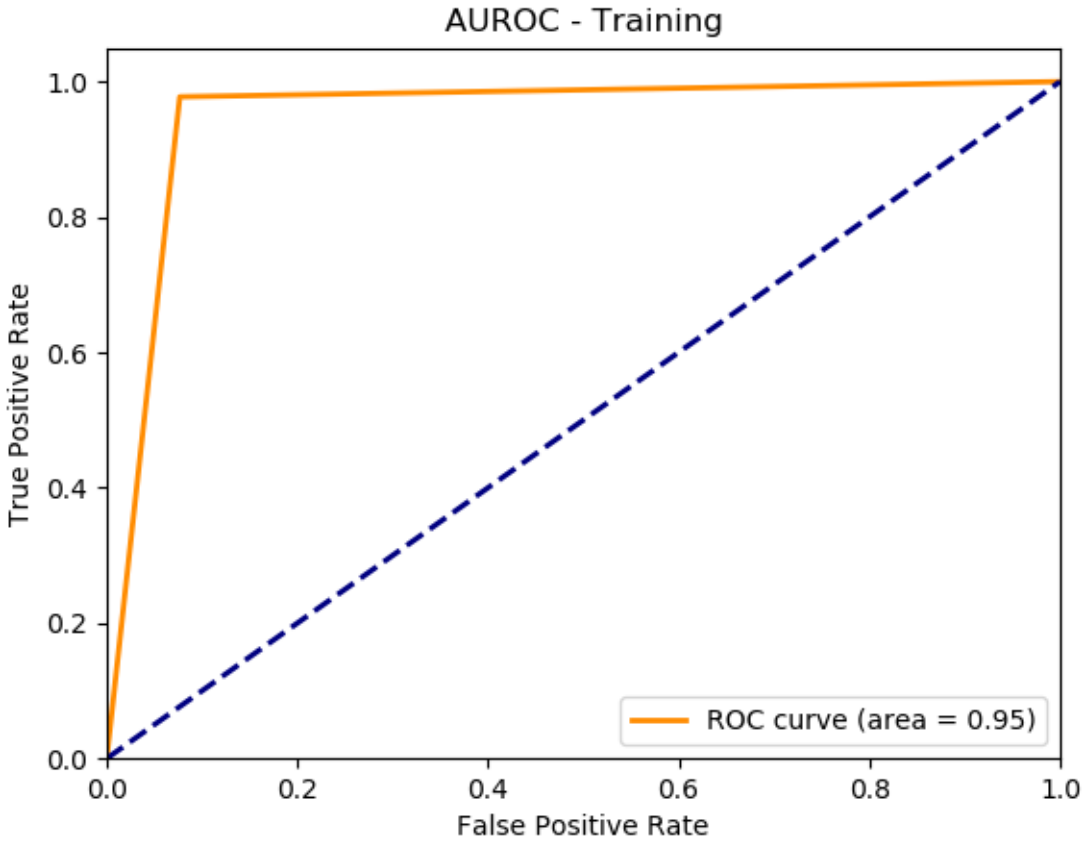

Figura 4.3.1: Grafico AUROC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 4 epoche. Il valore di AUROC è pari a 0.95.

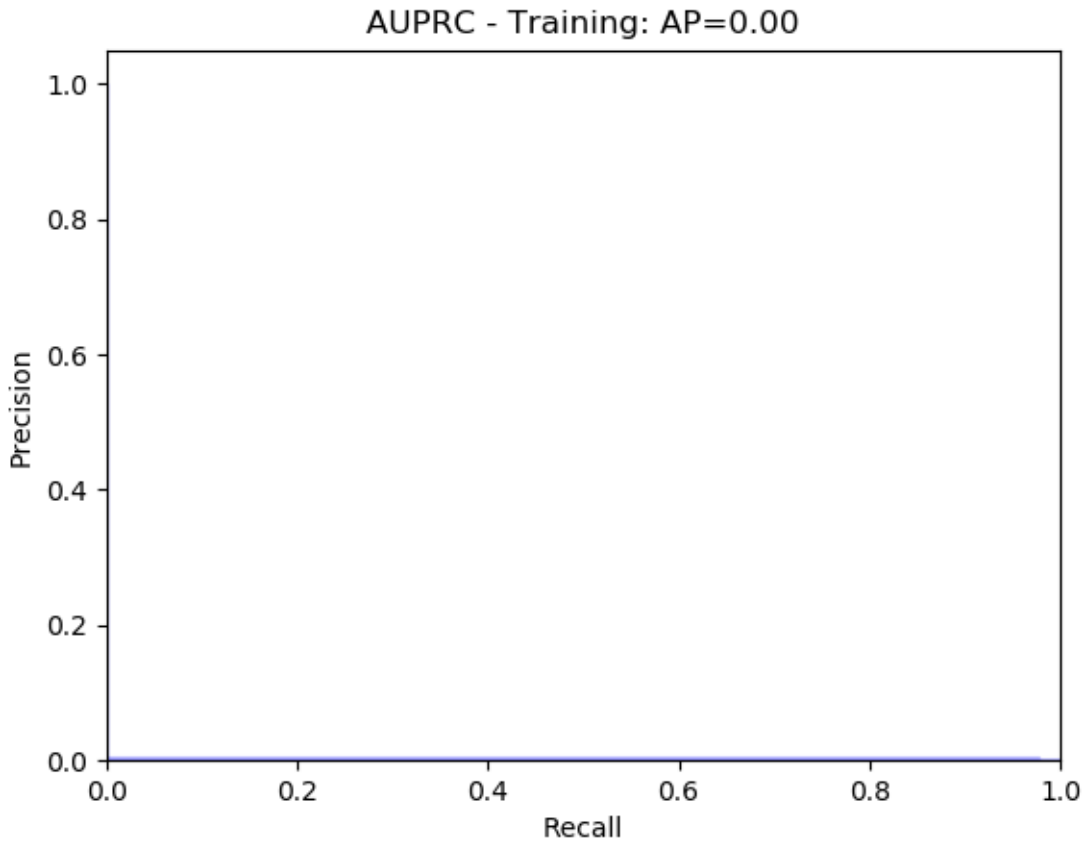

Figura 4.3.2: Grafico AUPRC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 4 epoche. Anche qui, come nel grafico 4.2.2, è possibile notare un valore estremamente basso di AUPRC, evidenziando come vi sia una bassa precisione per il presente modello.

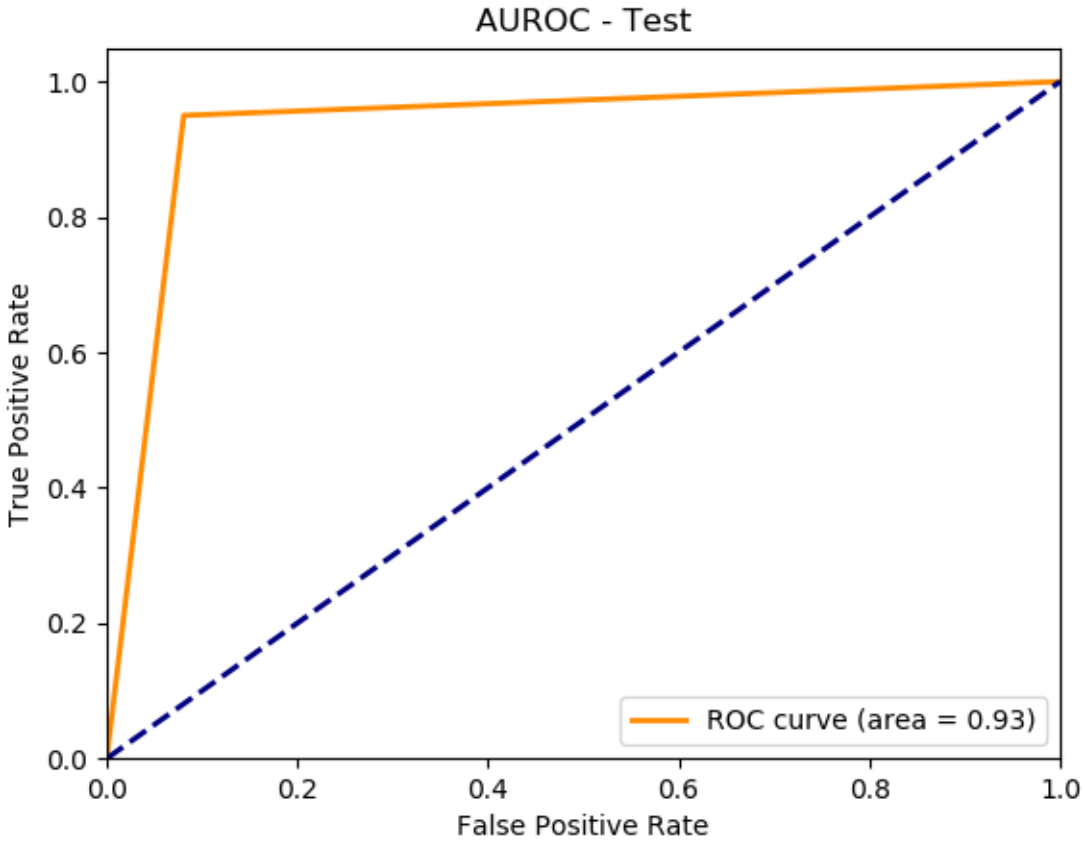

Figura 4.3.3: Grafico AUROC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 4 epoche, con valore di AUROC pari a 0.93.

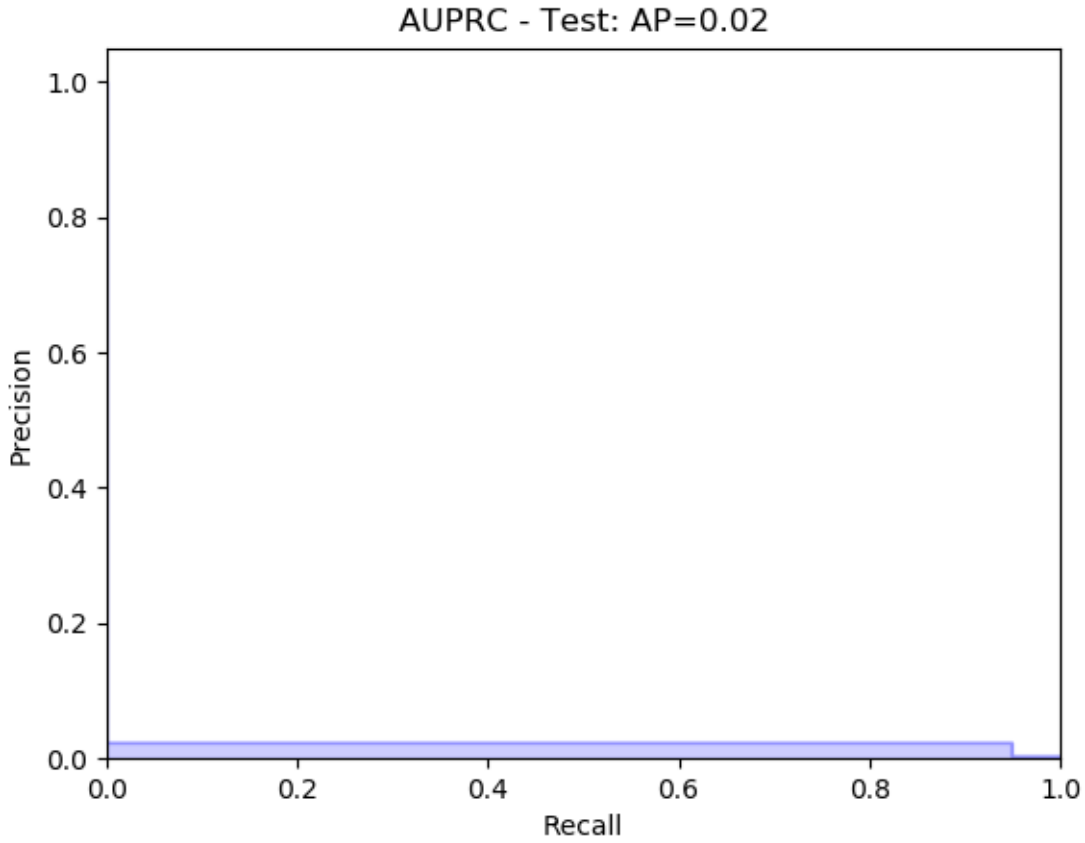

Figura 4.3.4: Grafico AUPRC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 4 epoche, con un valore di precisione media pari a 0.02.

4.4 Risultati ottenuti con la rete neurale a 6 neuroni per 100 epoche: Matrice di Confusione, grafici AUROC e AUPRC

Di seguito sono presentati le metriche di valutazione utilizzate per valutare, sia sul training set che sul test set, la performance del percettrone multistrato con strato interno a 6 neuroni e la cui fase di apprendimento prevede 100 epoche.
Si evidenzia qui come l'aumento del valore di epoche abbia inciso sulla prestazione del percettrone multistrato. Prendendo visione della matrice di confusione relativa alla performance del modello sul test set, si nota come tutti gli esempi positivi siano stati classificati correttamente come tali. Tuttavia, nonostante vi sia un'ottima performance sugli esempi positivi, il modello presenta comunque un numero non indifferente di falsi positivi, ossia campioni negativi classificati però come positivi dal sistema. Ciò è indicato anche dai valori estremamente bassi di AUPRC, sia per quanto concerne il training set sia per quanto concerne il test set.

Matrice di Confusione

su Training Set

		Valori predetti		
		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 355	Falsi Negativi: 1	P′
	n′	Falsi Positivi: 116570	Veri Negativi: 864462	N′
	totale	P	N	

Matrice di Confusione

su Test Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 40	Falsi Negativi: 0	P′
	n′	Falsi Positivi: 2643	Veri Negativi: 16335	N′
	totale	P	N	

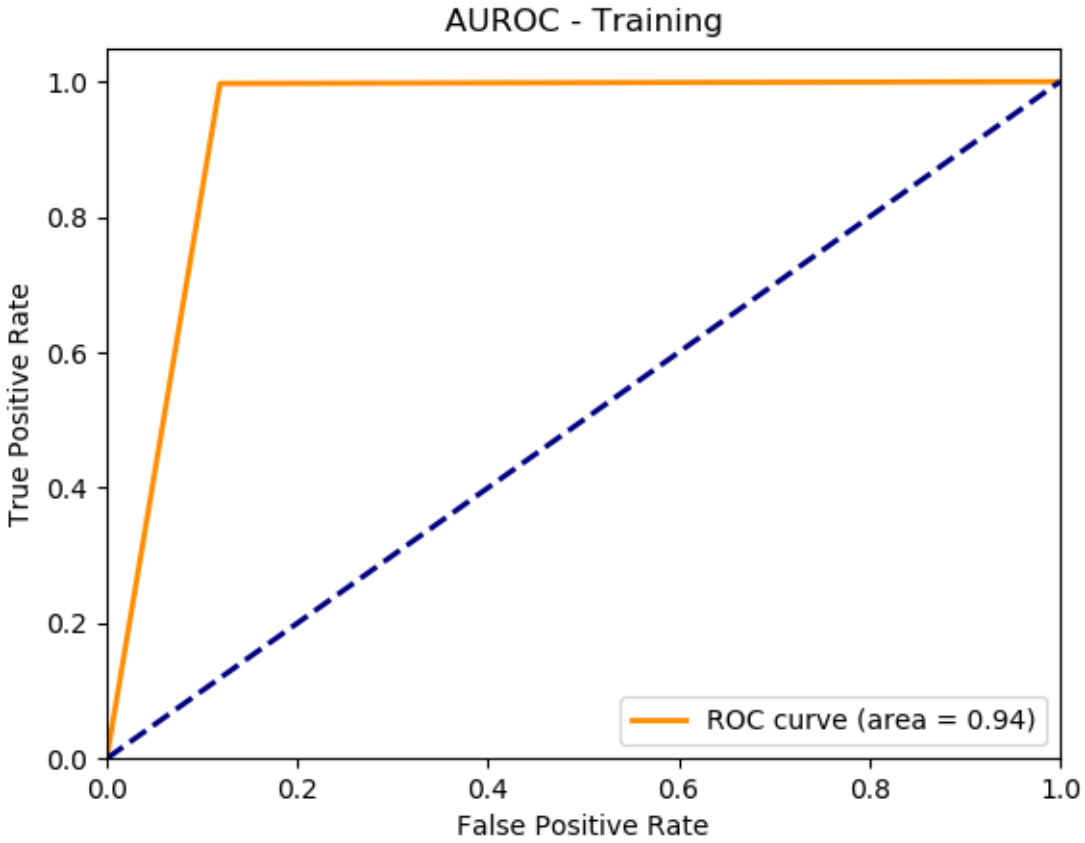

Figura 4.4.1: Grafico AUROC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 6 neuroni e addestrata per 100 epoche, con valore AUROC pari a 0.94.

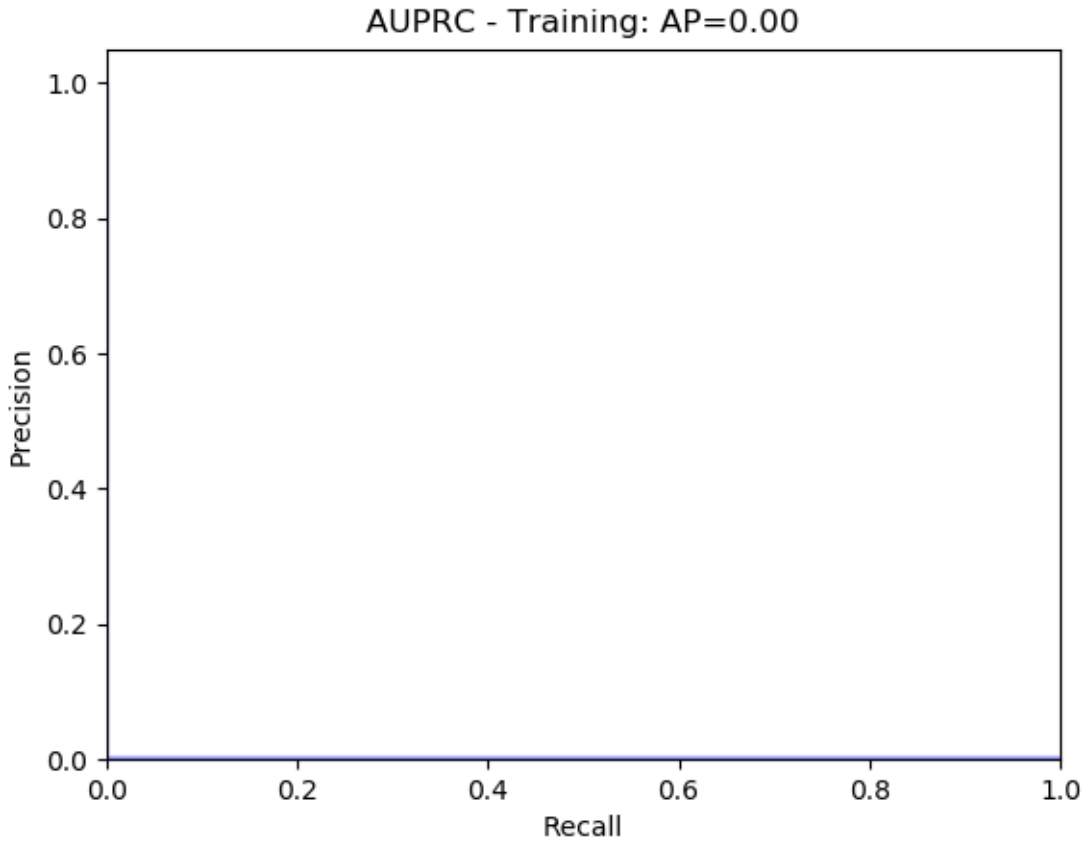

Figura 4.4.2: Grafico AUPRC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 6 neuroni e addestrata per 100 epoche. Come si può notare dal grafico, il valore di medio di precisione qui è estremamente basso, quasi nullo.

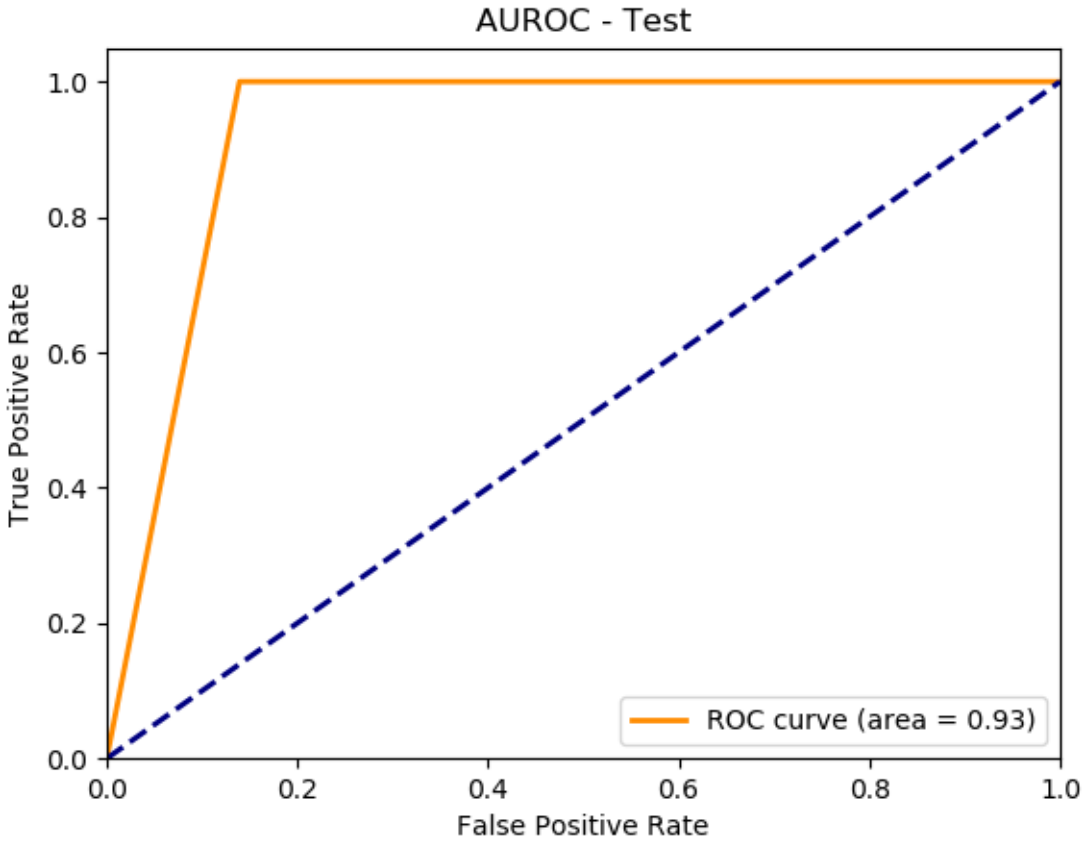

Figura 4.4.3: Grafico AUROC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 6 neuroni e addestrata per 100 epoche. Qui il valore di AUROC è uguale a 0.93

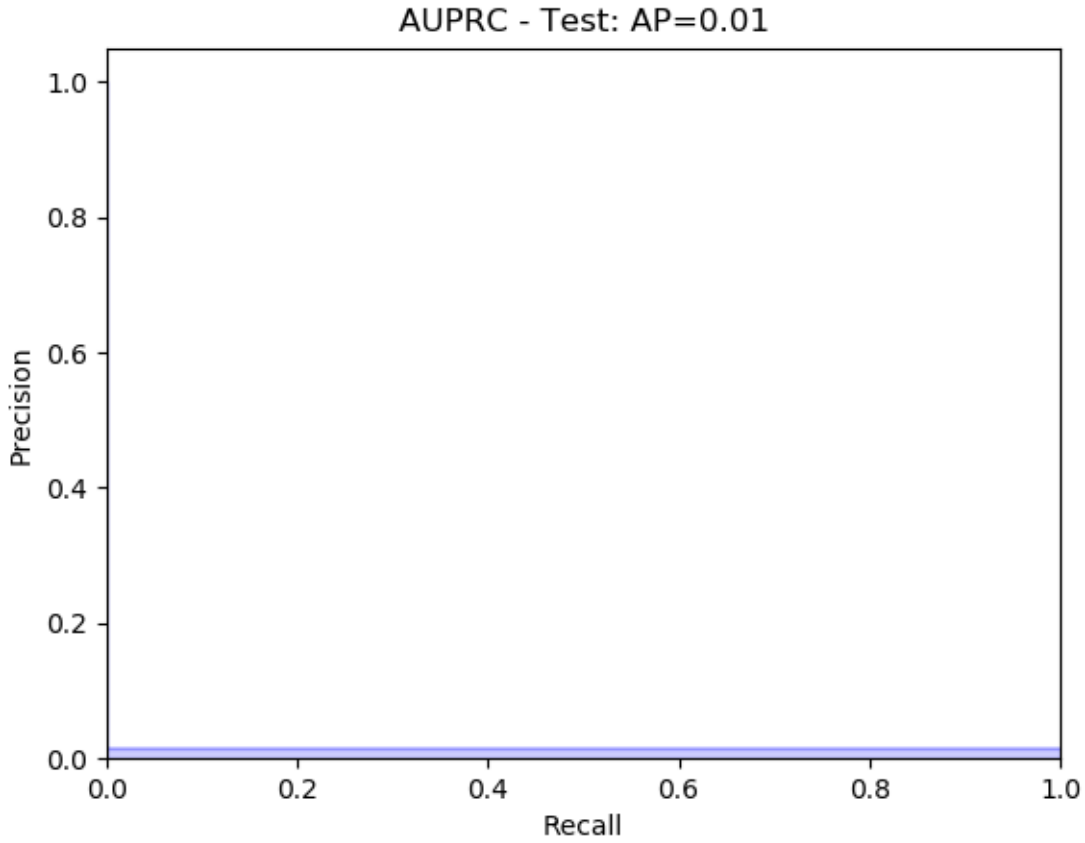

Figura 4.4.4: Grafico AUPRC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 6 neuroni e addestrata per 100 epoche. Il valore medio di precisione è pari a 0.01.

4.5 Risultati ottenuti con la rete neurale a 13 neuroni per 100 epoche: Matrice di Confusione, grafico AUROC e AUPRC

Infine, vengono presentati i grafici e le tabelle relative alle metriche di valutazione della performance del modello di rete neurale che prevede uno strato interno di 13 neuroni e addestrato per 100 epoche.
É possibile notare come per il training set, il numero di veri positivi e il numero di falsi negativi si siano mantenuti uguali sia per il modello a sei neuroni sia per questo modello a tredici neuroni, entrambi addestrati per 100 epoche. Tuttavia, si nota come il presente modello, rispetto al precedente, presenti un dimezzamento del numero di falsi positivi, ad indicare un miglioramento nell'identificazione delle istanze della classe negativa.
Anche per quanto riguarda la performance sul test set, il numero di veri positivi e falsi positivi rimane pressoché uguale, mentre il numero di falsi positivi è all'incirca un terzo rispetto al modello precedente.

Matrice di Confusione

su Training Set

		Valori predetti		
		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 351	Falsi Negativi: 5	P′
	n′	Falsi Positivi: 57670	Veri Negativi: 923362	N′
	totale	P	N	

Matrice di Confusione

su Test Set

Valori predetti

		p	**n**	**totale**
Valori Effettivi	**p′**	Veri Positivi: 39	Falsi Negativi: 1	P′
	n′	Falsi Positivi: 885	Veri Negativi: 18093	N′
	totale	P	N	

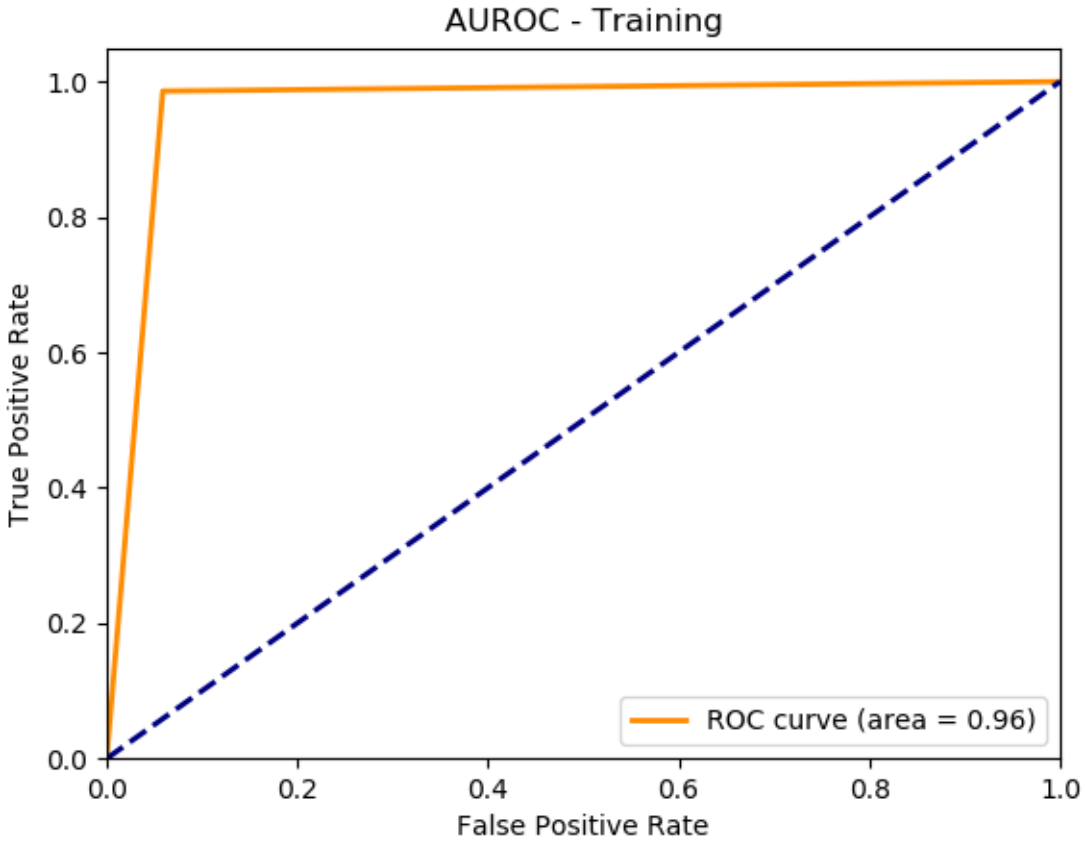

Figura 4.5.1: Grafico AUROC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 100 epoche, con valore AUROC pari a 0.96.

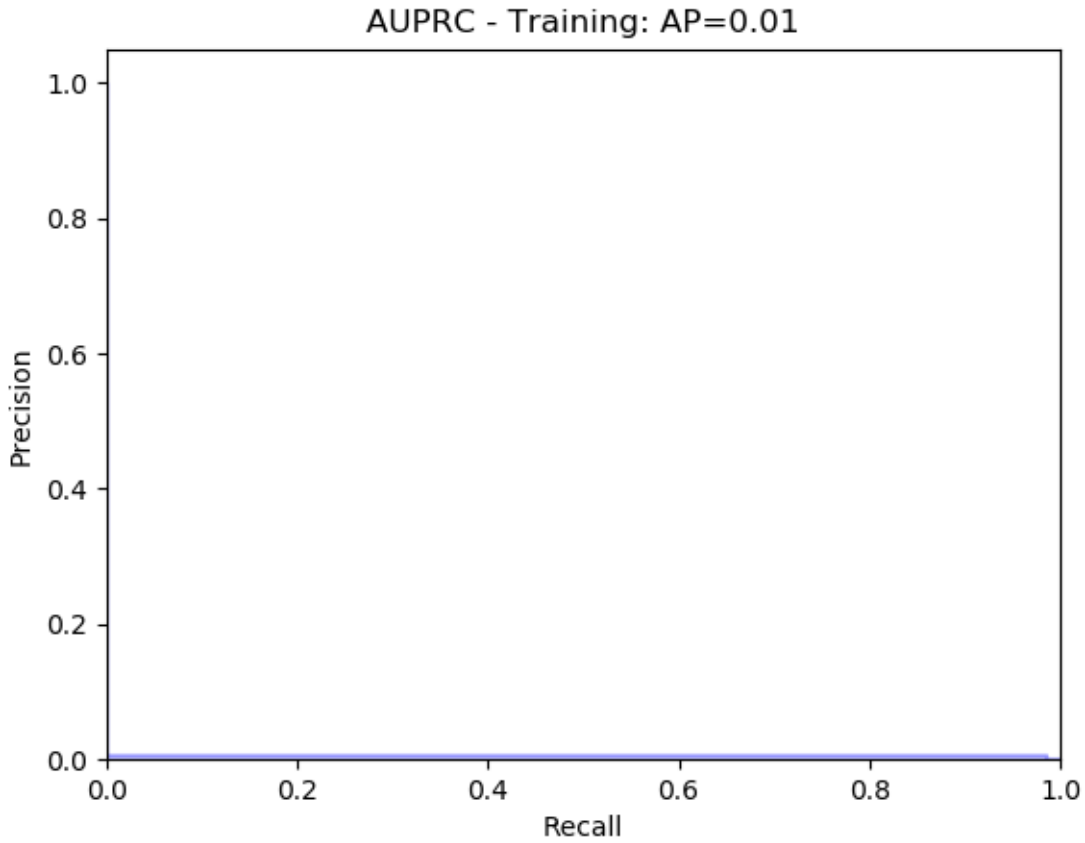

Figura 4.5.2: Grafico AUPRC valutato sul training set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 100 epoche. Il valore di medio di precisione si mantiene basso in quanto vi è ancora un alto tasso di esempi negativi etichettati erroneamente come positivi.

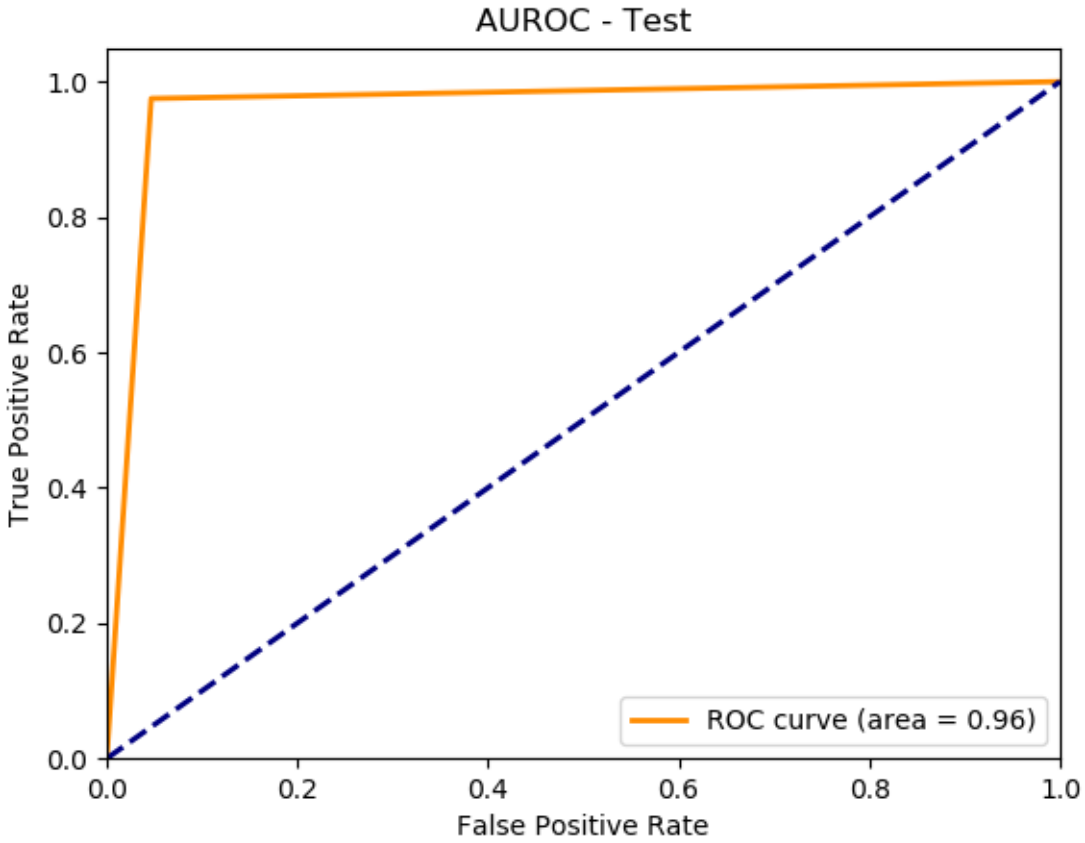

Figura 4.5.3: Grafico AUROC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 100 epoche. Qui il valore di AUROC è uguale a 0.96

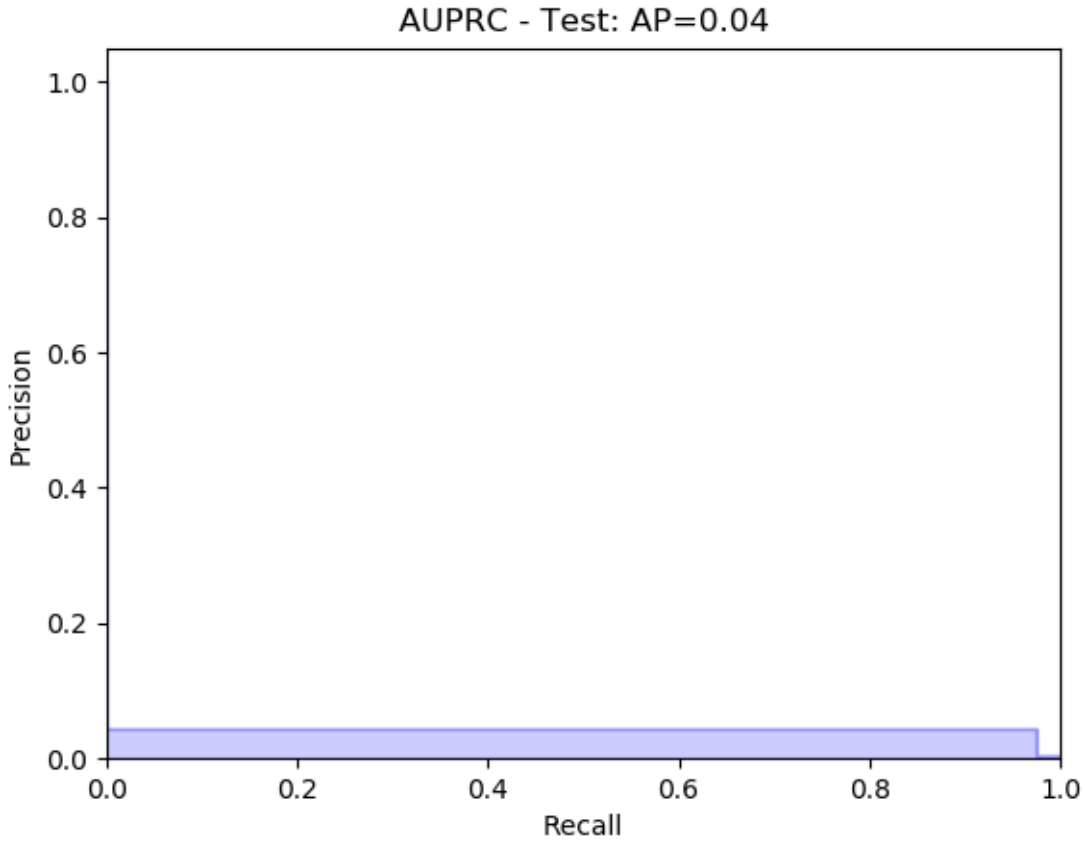

Figura 4.5.4: Grafico AUPRC valutato sul test set e relativo alla rete neurale implementata caratterizzata da un hidden layer con 13 neuroni e addestrata per 100 epoche. Il valore medio di precisione è pari a 0.04.

4.6 Confronto

Visti i risultati prodotti dalle diverse versioni del percettrone multistrato implementate, è possibile trarre alcuni confronti.
In primo luogo, è possibile notare come, fissato il numero di epoche a 4, l'aumento del numero di neuroni da 6 a 13 mostri un miglioramento nell'individuazione degli esempi positivi, ma un peggioramento nella quantità di falsi positivi, indicando come il modello presenti un più alto numero di errori nel classificare gli esempi negativi. Infatti, i valori AUPRC mostrano come vi sia una precisione estremamente esigua, a fronte di un fattore di richiamo elevato.
É possibile notare come il cambiamento del numero di epoche, da 4 a 100, fissato il numero di neuroni, aiuti ad individuare una maggiore quantità di varianti a livello singolo nucleotide patogeniche. Tuttavia, nel caso in cui i neuroni dello strato nascosto siano 6, vi è un peggioramento a discapito degli esempi negativi, di cui una parte non indifferente viene classificata erroneamente come positiva. Ciò è attestato dai valori di precisione e richiamo che mostrano rispettivamente un valore estremamente basso e un valore alto.
Ciononostante, quando il numero di epoche è pari a 100 e il numero di neuroni dello strato nascosto è pari a 13, è possibile notare come la performance subisca un miglioramento generale. Proprio utilizzando questo modello, si riscontrano i migliori valori di AUROC e AUPRC, giustificati da una migliore classificazione non solo per quanto concerne gli esempi positivi, ma anche per quanto riguarda gli esempi negativi. Dunque, la prestazione generale del modello migliora anche nella classificazione degli esempi negativi, di cui sempre meno vengono etichettati in modo errato come positivi.
Tutto ciò indica come il modello si presti bene nell'individuazione delle istanze positive, ma al contempo effettui troppi errori nel classificare erroneamente come positivi diversi esempi negativi. Un miglioramento è evidente nel momento in cui il numero di epoche nella fase di addestramento è aumentato a 100. Questo cambiamento di parametro permette al sistema di migliorare la performance, quindi di discernere in modo migliore tra varianti a livello di singolo nucleotide patogeniche e non.

Capitolo 5

Conclusioni

A fronte di quanto è stato detto nei capitoli precedenti, è possibile affermare come il sistema abbia una scarsa precisione, attestata dalla presenza di diversi errori nell'etichettatura degli esempi, sia per quanto riguarda il training set sia per quanto riguarda il test set. Ciononostante, il percettrone multistrato realizzato mostra di essere in grado di riconoscere buona parte delle varianti a livello di singolo nucleotide patogeniche. Ciò è dovuto al fatto di aver utilizzato un algoritmo di sovracampionamento, nello specifico l'algoritmo SMOTE, nella creazione di esempi positivi sintetici al fine di bilanciare il training set, che si presenta originariamente come estremamente sbilanciato. Dunque, l'utilizzo dell'algoritmo SMOTE si è rivelato essere essenziale nella fase di addestramento del modello. Un ulteriore fattore rivelatosi importante è stato l'incremento del numero di epoche a cui sottoporre i dati in fase di addestramento. Ponendo il numero di epoche pari a 100, è possibile prendere visione di un miglioramento nella prestazione della rete neurale, che, seppur mantenendo una bassa precisione, migliora la performance.
Alcuni sviluppi futuri potrebbero consistere nel testare empiricamente se vi siano miglioramenti aggiungendo un ulteriore hidden layer al percettrone multistrato e aumentare il numero di epoche, al fine di comprendere quanto possa migliorare la performance del modello una volta attuati tali cambiamenti.

Bibliografia

D. Smedley, M, Schubach, J. Jacobsen, S. Kohler, T. Zemojtel, M. Spielmann, M. Jager, H. Hochheiser, N. Washington, J. McMurry, M. Haendel, C. Mungall, S. Lewis, T. Groza, G. Valentini and P.N. Robinson, *A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease*, The American Journal of Human Genetics, September 2016.

M. Schubach, M. Re, P.N. Robinson and G. Valentini, *Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants*, Scientific Reports, Nature Publishing, 2017.

I. Goodfellow, Y. Bengio and A. Courville,*Deep Learning*, MIT Press, 2016.