Advanced Machine Learning

Lorenzo Mauri matr 807306

November 20, 2020

Assignment 2

The assignment consists in the prediction of default payments using a neural network.

The dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

1 Descrizione del dataset

Il dataset fornito comprende complessivamente 24000 osservazioni di training, 24 attributi e la variabile target :

- **Features** :
  LIMIT BAL,PAY0,PAY2,PAY3,PAY4,
  PAY5,PAY6,BILLAMT1,BILLAMT2,
  BILLAMT3,BILLAMT4,BILLAMT5,BILLAMT6,
  PAYAMT1,PAYAMT2,PAYAMT3,
  PAYAMT4,PAYAMT5,PAYAMT6,
  default.payment.next.month

2 Analisi esplorativa

2.1 Criticità

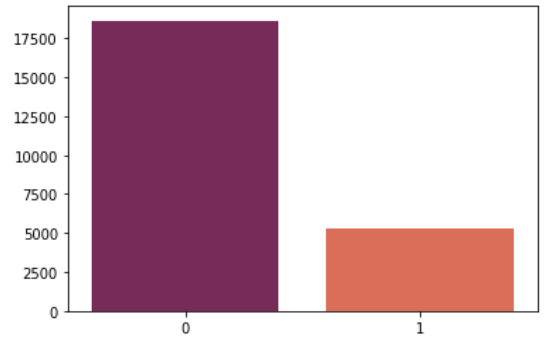
Esplorando i dati sono emerse le seguenti criticità :

- **variabile EDUCATION** : sono presenti valori codificati con i valori 0, 5 e 6 , per i quali non è presente una codifica (nel caso di 0) oppure l'informazione non è nota (nel caso di 5,6). Decido di sostituire a questi valori la moda della variabile (che è pari a 2)
- **variabile MARRIAGE** : vi sono 45 valori non codificati (pari a 0), decido di eliminarli dal dataset
- **Classi sbilanciate** : esiste uno sbilanciamento di classe che potrebbe impattare sulle prestazioni del modello

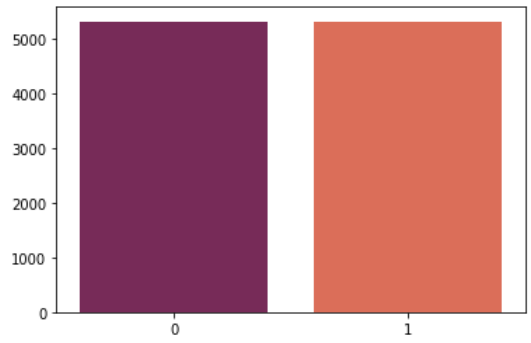
2.2 Classi Sbilanciate

Noto che le due classi della variabile target sono sbilanciate, pertanto decido di bilanciarle applicando la tecnica dell' *undersampling* consistente nel ridurre la classe maggioritaria tramite campionamento casuale (senza reinserimento, in modo tale da non avere doppie osservazioni nel nuovo dataset). Avere le classi della variabile target sbilanciate può introdurre distorsione nel modello e quindi influenzare notevolmente la fase di training, in tale situazione il modello effettuerà più previsioni della classe 1 solo per il fatto che è la classe che ha analizzato maggiormente. Nella prima figura vediamo la situazione di partenza con le relative proporzioni (78% e 22 %) , mentre nella seconda il bilanciamento

```
default.payment.next.month
0    18636
1     5319
dtype: int64
proportion 0 class: 78.0 % of dataset
proportion 1 class : 22.0 % of dataset
```




```
default.payment.next.month
0    5319
1    5319
dtype: int64
proportion 0 class: 50.0 % of dataset
proportion 1 class : 50.0 % of dataset
```



2.3 Normalizzazione

Al fine di ridurre le differenze di scala applico una normalizzazione

$$x_{norm} = \frac{x - min}{max - min}$$

Le differenze di scala potrebbero infatti indurre il modello a dare maggiore importanza ad un attributo piuttosto che ad un altro (è la stessa differenza riscontrabile tra le variabili Età e Reddito, che hanno scale totalmente diverse ma non per questo sono una più rilevante dell'altra).

3 Data Modelling

Dopo il bilanciamento dei dati ottengo un dataset di 10638 osservazioni composto da 24 esplicative e una variabile risposta.

Al fine di valutare correttamente le performance del modello (e in particolare avere maggiori informazioni circa la generalizzazione del modello) ho deciso di dividere il dataset di training in due parti distinte (80% dei dati per il training set e il restante 20 % per il test set, in modo pseudo-casuale). Ottengo in tal modo osservazioni per il training e per il test.

3.1 Struttura della Neural Network

La Neural Network implementata possiede le seguenti caratteristiche :

- **Numero di hidden layers** : 2
- **Numero di unità/neuroni per hidden layer** : 50 , 20
- **Numero di unità/neuroni per input**: 50
- **Numero di unità/neuroni per ouput**: 1
  Il modello prevede un unico valore
- **Funzioni di attivazione** :
  - **Sigmoid** per il layer di output in quanto il valore previsto deve essere compreso tra 0 e 1
  - **Rectifier Linear Unit (ReLU)**
    Si è rivelata la funzione di attivazione più performante per il problema in esame pertanto ho deciso di inserirla in ogni strato della rete
- **Funzione di perdita** : Binary Crossentropy .
  Per il problema che stiamo affrontando questa funzione di perdita è quella maggiormente utilizzata, in quanto

Il numero di layer e le unità di ciascuno sono state scelte su base empirica, testando varie configurazioni e confrontando i risultati.

3.2 Regolarizzazione

Al fine di ridurre l'overfitting, ho deciso di applicare le seguenti strategie di regolarizzazione :

- **Dropout** : $p = 0.6$
- **L1 - Norm** : $\lambda = 0.02$
- **L2 - Norm** : $\lambda = 0.02$

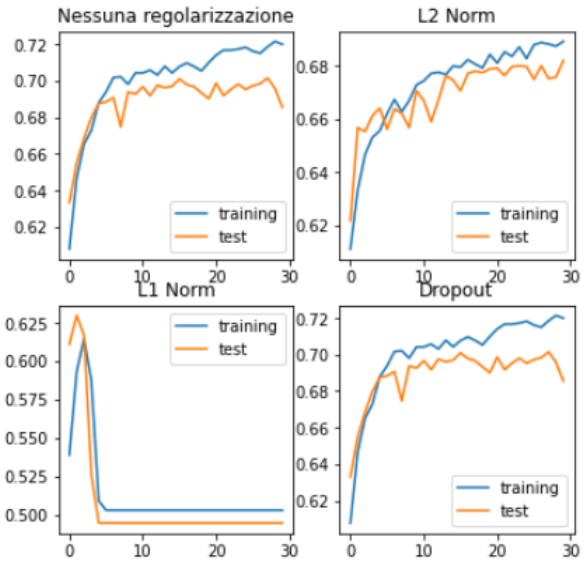
4 Risultati e conclusioni

Il modello inizialmente ottimizzato sembra performare abbastanza bene, sia in fase di training che di test.Tuttavia le regolarizzazioni effettuate conducono in alcuni casi ad un miglioramento delle prestazioni in termini di accuratezza e f1 score.

Di seguito i grafici di performance e gli score ottenuti :

Tipo di regolarizzazione	Accuracy	F1 Score (avg)
Nessuna	0.6855	0.69
L1 Norm	0.4944	0.67
L2 Norm	0.6821	0.33
Dropout	0.7024	0.70

Come è possibile notare il modello con il Dropout è quello con accuratezza e F1 Score maggiori, mentre possiamo affermare che L2 Norm, sotto questo profilo, è il peggiore.



4.1 Weights Control

Confrontiamo ora le due L_p regularization. Data la funzione di perdita $L(x, y)$, la regolarizzazione è la seguente :

$$L(x, y) = L(x, y) + \lambda \sum_{j=0}^{M} |W_j|$$

$$L(x, y) = L(x, y) + \lambda \sum_{j=0}^{M} W_j^2$$

In tabella troviamo i valori ottenuti sommando due pesi di tutti e tre i modelli considerati.

tipo di regolarizzazione	$\sum$
Nessuna	0.49
L1 Norm	0.4518
L2 Norm	0.1566

A parità di λ, osserviamo che la L1 penalizza maggiormente la funzione di perdita rispetto alla L2, in quanto i pesi sono minori di 1. Questo spiegherebbe la così bassa accuracy del modello L1 rispetto a quella di tutti gli altri. Possiamo vedere in ogni caso il funzionamento della regolarizzazione e l'impatto sulle performance del modello.

Conclusione : In ultima analisi scegliamo dunque l'ultimo modello.