TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

Università degli Studi di Roma Tor Vergata

Progetto del corso di Metodi di Ottimizzazione per Big Data

Studente :
Andrea Andreoli 0350012

Professore :
Andrea Cristofari

21 marzo 2025

Indice

1 Introduzione

In questo report viene implementato un modello di machine learning per la **classificazione multiclasse**, utilizzando una rete neurale. L'obiettivo è creare un sistema in grado di riconoscere diverse categorie di dati, sfruttando la capacità della rete neurale di apprendere pattern complessi. Verranno descritti i passaggi principali, dalla preparazione dei dati all'addestramento del modello. La rete neurale è stata implementata in Python.
Per la corretta esecuzione del progetto, è necessario aver installato correttamente i seguenti pacchetti:

- Numpy
- Matplotlib
- Scikit-learn
- Imbalanced-learn
- Pandas

2 Struttura della rete neurale

Nella sezione che segue, verranno analizzati in dettaglio sia la struttura della rete neurale utilizzata, sia le principali scelte implementative adottate. Saranno esplorati i vari componenti del modello, le motivazioni alla base delle scelte fatte e come questi elementi contribuiscano al raggiungimento degli obiettivi di classificazione multiclasse.

2.1 Rete neurale

Come descritto nell'introduzione, la rete neurale implementata ha lo scopo di effettuare una classificazione multiclasse. La sua struttura è la seguente:

- **Input layer**: livello che riceve i dati in ingresso composto da N neuroni, dove N coincide con il numero di features del dataset.
- **Hidden layers**: due livelli nascosti responsabili dell'elaborazione dei dati (il numero di neuroni all'interno di questi livelli è variabile per poter studiare il comportamento di diverse configurazioni).
- **Output layer**: livello che produce l'output finale della rete composto da M neuroni, dove M coincide con il numero di classi da predire.

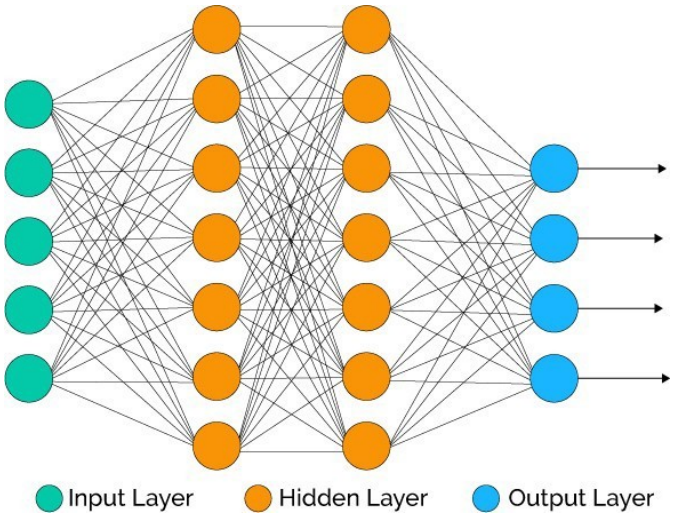

Figura 1: Rete neurale

2.1.1 Livelli nascosti

Per la rete neurale implementata, il numero di neuroni nei due livelli nascosti è stato configurato secondo diverse architetture al fine di valutare l'impatto della complessità del modello sulle prestazioni. Le configurazioni testate includono:

- $(N, 32, 32, M)$
- $(N, 32, 64, M)$
- $(N, 64, 64, M)$
- $(N, 64, 128, M)$
- $(N, 128, 128, M)$
- $(N, 128, 256, M)$
- $(N, 256, 256, M)$
- $(N, 256, 512, M)$
- $(N, 512, 512, M)$

Questo permette di confrontare le prestazioni di reti con differenti capacità rappresentazionali e individuare la struttura ottimale per il problema di classificazione multiclasse.

2.2 Funzione di attivazione

Le funzioni di attivazione sono un componente fondamentale nelle reti neurali, in quanto determinano se un neurone debba essere attivato o meno, influenzando così l'output della rete. Queste funzioni introducono non-linearità nel modello, permettendo alle reti neurali di apprendere e rappresentare relazioni complesse nei dati.
Nella rete neurale in esame sono state implementate due funzioni di attivazione per i livelli nascosti:

- **ReLU**
- **tanh**

Per il livello di output, è stata scelta la funzione di attivazione **softmax**, poiché il problema in esame è di classificazione multiclasse. La funzione softmax trasforma i valori in uscita della rete in probabilità, che rappresentano la probabilità di appartenenza di un dato input a ciascuna delle classi disponibili. Questo la rende particolarmente adatta per compiti in cui è necessario determinare la classe più probabile tra molteplici opzioni.

2.3 Inizializzazione dei parametri

L'inizializzazione dei parametri è un passaggio cruciale per il buon funzionamento della rete neurale. Per la funzione di attivazione **ReLU**, è stata utilizzata l'inizializzazione **He**. Questa scelta aiuta a evitare il problema del vanishing gradient, tipico delle funzioni di attivazione ReLU. Per la funzione **tanh**, invece, è stata utilizzata l'inizializzazione **Xavier** (o Glorot). L'inizializzazione Xavier è particolarmente adatta per le funzioni di attivazione simmetriche come la tanh, migliorando la propagazione del gradiente durante l'addestramento.

2.4 Algoritmi di ottimizzazione

Per l'ottimizzazione della rete neurale, sono stati implementati due metodi principali:

- **Stochastic Gradient Descent**.
- **Stochastic Gradient Descent** con **Momentum**.

Lo Stochastic Gradient Descent aggiorna i parametri della rete in base alla direzione del gradiente calcolato sulla funzione di costo, ma può risultare lento e sensibile alla scelta del learning rate. Il gradiente con Momentum, invece, accelera la convergenza introducendo un termine che accumula i gradienti precedenti, permettendo così di superare meglio i minimi locali e accelerare il processo di apprendimento. Inoltre, il learning rate può essere impostato come costante, oppure può decrescere nel tempo in base al *tasso di decadimento* e al numero di *epoche*.

2.5 Funzione di perdita (loss function)

La scelta della funzione di perdita è fondamentale per il processo di addestramento della rete neurale, in quanto guida l'ottimizzazione dei parametri. Il training set è stato suddiviso in mini-batch di dimensione fissa pari a 64, di conseguenza, la funzione di perdita viene calcolata su ciascun batch di dati invece che su un singolo esempio. Nel caso di classificazione multiclasse, la **cross-entropy loss** per un mini-batch di dimensione N viene definita come la media della funzione di perdita calcolata per ogni esempio del batch. La funzione di perdita nel contesto dei mini-batch è la seguente:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{C} y_i^{(n)} \log(\hat{y}_i{}^{(n)}) \tag{1}$$

dove:

- N è la dimensione del mini-batch.
- C è il numero di classi.
- $y_i^{(n)}$ è la probabilità vera della classe i per l'esempio n.
- $\hat{y_i}^{(n)}$ è la probabilità predetta dalla rete per la classe i nell'esempio n.

2.6 Regolarizzazione

La regolarizzazione è una tecnica fondamentale per prevenire l'overfitting e migliorare la generalizzazione del modello. Nella rete neurale in esame, sono stati utilizzati tre approcci principali di regolarizzazione:

- **L1**
- **L2**
- **Early stopping**

La **regolarizzazione L1** aggiunge alla funzione di perdita una penalizzazione proporzionale al valore assoluto dei pesi. La *funzione di regolarizzazione* per L1 è quindi definita come:

$$\Omega = ||\omega||_1$$

La **regolarizzazione L2**, invece, penalizza i pesi in base al loro quadrato. La *funzione di regolarizzazione* per L2 è:

$$\Omega = ||\omega||^2$$

In entrambe le tecniche, la penalizzazione avviene tramite un **parametro di regolarizzazione** λ. Pertanto, la funzione di perdita regolarizzata diventa:

$$\tilde{\mathcal{L}} = \mathcal{L} + \lambda\Omega$$

Infine, **early stopping** è una tecnica che interrompe l'addestramento quando il modello smette di migliorare le proprie prestazioni, prevenendo l'overfitting. Il parametro chiave di questa metodologia, detto **patience**, determina per quante epoche il modello può continuare l'addestramento senza miglioramenti prima che venga interrotto. Nella rete neurale implementata, il parametro *patience* è impostato a 5.

3 Cross-validation

La strategia di **cross-validation** prevede i seguenti passi:

- La suddivisione del dataset in training set, validation set e test set.
- Al fine di determinare il valore ottimale di λ si definisce un insieme di valori da testare durante l'addestramento, in particolare:
  - Per applicare la regolarizzazione L1: $\lambda \in [0.001, 0.01, 0.1, 0.0015, 0.0125]$.
  - Per applicare la regolarizzazione L2: $\lambda \in [0.01, 0.1, 0.5, 1, 2.5, 5]$.

- Per ogni λ si effettua l'addestramento del modello utilizzando i dati del training set e si stima l'errore sul validation set.
- Si sceglie il parametro λ che minimizza l'errore sul validation set.
- Si utilizza il test set per valutare le prestazioni del modello dopo essere stato addestrato.

4 Dataset

Al fine di valutare le prestazioni della rete neurale implementata, sono stati utilizzati i seguenti dataset:

- **Dry Bean Dataset** [1]: è un insieme di dati utilizzato per la classificazione di sette diversi tipi di fagioli secchi, tenendo in considerazione caratteristiche come la forma, la struttura, il tipo e la configurazione in base alla situazione del mercato.

  Il dataset è stato creato utilizzando un sistema di visione artificiale che ha estratto 16 caratteristiche morfologiche da immagini di 13.611 fagioli. Queste caratteristiche includono area, perimetro, lunghezza dell'asse maggiore, larghezza dell'asse minore, coefficienti di forma e misure di rugosità.

  Caratteristiche del dataset:

  - Numero di campioni: 13.611
  - Numero di features: 16
  - Numero di classi: 7

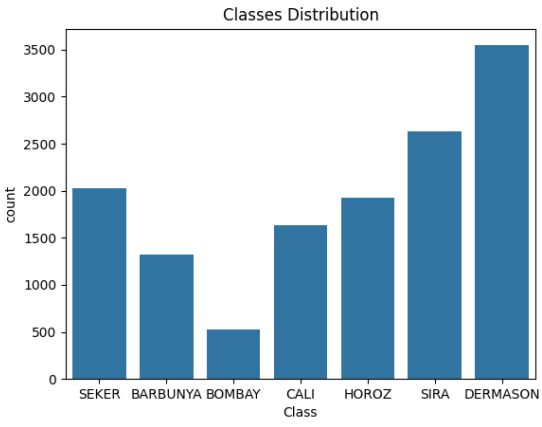

Figura 2: Distribuzione delle classi nel dataset

- **Sloan Digital Sky Survey - DR18 Dataset** [2]: Sloan Digital Sky Survey (SDSS) è un progetto astronomico che mira a mappare una vasta porzione del cielo attraverso osservazioni fotometriche e spettroscopiche.

  Il DR18 include cataloghi di targeting preparati per i programmi scientifici Black Hole Mapper e Milky Way Mapper, oltre a dati spettroscopici raccolti nell'ambito

del programma SPIDERS. Questi dati forniscono informazioni dettagliate su una vasta gamma di oggetti celesti, contribuendo in modo significativo alla comprensione della struttura e dell'evoluzione dell'universo.

Caratteristiche del dataset:

- Numero di campioni: 100.000
- Numero di features: 42
- Numero di classi: 3

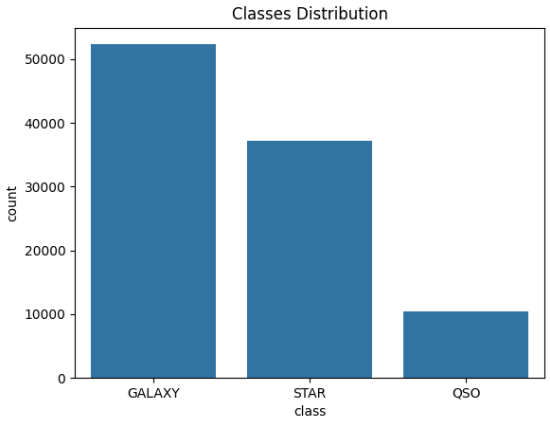

Figura 3: Distribuzione delle classi all'interno nel dataset

- **Air Quality and Pollution Dataset** [3]: questo dataset è una raccolta di dati che mira a valutare la qualità dell'aria in diverse regioni, fornendo metriche ambientali e approfondimenti demografici utili per prevedere i livelli di inquinamento atmosferico.

  Le variabili presenti nel dataset includono l'Indice di Qualità dell'Aria (AQI), le concentrazioni di vari inquinanti, le condizioni meteorologiche e metriche demografiche. Questi dati sono fondamentali per analizzare l'impatto dell'inquinamento atmosferico sulla salute pubblica e per sviluppare modelli predittivi della qualità dell'aria.

  Caratteristiche del dataset:

  - Numero di campioni: 5.000
  - Numero di features: 9
  - Numero di classi: 4

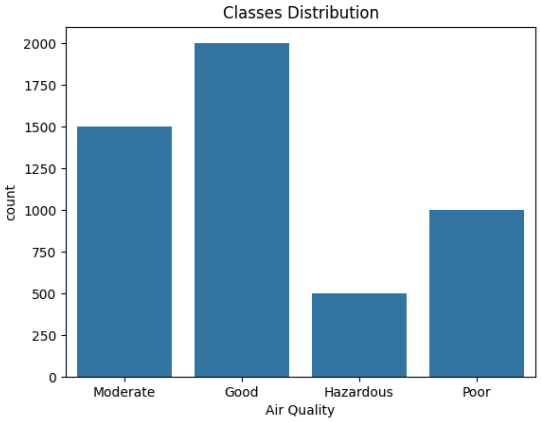

Figura 4: Distribuzione delle classi nel dataset

4.1 Preprocessamento

Per rendere i dati adatti all'addestramento del modello, è stato eseguito un'operazione di preprocessamento volto a trasformarli e strutturarli in una forma utilizzabile.
Poiché tutti i dataset presentano una classe target di tipo categorico, è stato necessario eseguire una fase iniziale di encoding per convertire le categorie in valori numerici. Gli encoding ottenuti, per ciascun dataset, sono i seguenti:

- **Dry Bean Dataset**:
  - BARBUNYA → 0
  - BOMBAY → 1
  - CALI → 2
  - DERMASON → 3
  - HOROZ → 4
  - SEKER → 5
  - SIRA → 6
- **Sloan Digital Sky Survey - DR18 Dataset**:
  - GALAXY → 0
  - QSO → 1
  - STAR → 2

  Notare che la classe QSO sta per Quasi-Stellar Object.
- **Air Quality and Pollution Dataset**:
  - Good → 0
  - Hazardous → 1
  - Moderate → 2

  - Poor → 3

Successivamente, il preprocessamento è stato articolato in diverse fasi per garantire la qualità e l'affidabilità del dataset prima dell'addestramento del modello, in particolare:

- **Rimozione dei duplicati e gestione dei valori mancanti**: rimozione dei record duplicati per evitare ridondanze, separazione della colonna 'target' dalle altre feature e rimozione dei record con variabile target mancante.
- **Standardizzazione delle feature**: è stata applicata la standardizzazione tramite lo StandardScaler, trasformando le feature affinché abbiano una distribuzione con media zero e deviazione standard unitaria.
- **Suddivisione del dataset**: il dataset è stato suddiviso in *training set* (70%), *test set* (20%) e *validation set* (10%).
- **Feature selection**: se richiesto, viene applicata la *feature selection* sul dataset, basata sull'analisi delle correlazioni tra le features e il target, al fine di ridurre la complessità del problema. Verrano tenute in considerazione solo le features che mostrano una correlazione maggiore di una certa costante di soglia.
- **Bilanciamento delle classi**: se richiesto, in presenza di squilibri nella distribuzione della variabile target, è stata applicata una tecnica di bilanciamento delle classi per garantire un'equa rappresentazione di ciascuna classe. In particolare, dopo aver analizzato la distribuzione delle classi nel training set verrà applicata una delle seguenti tecniche in base a soglie predefinite:
  - *Oversampling con SMOTE*: se la classe minoritaria ha una proporzione inferiore alla soglia di oversampling (oversample_threshold, di default 0.3), viene applicata la tecnica SMOTE. SMOTE genera nuovi campioni sintetici per la classe minoritaria creando punti interpolati tra campioni esistenti, aumentando così la sua rappresentatività senza duplicare dati esistenti.
  - *Undersampling con RandomUnderSampler*: se la classe più rappresentata supera la soglia di undersampling (undersample_threshold, di default 0.7), viene applicato l'undersampling. Questa tecnica riduce la quantità di campioni della classe maggioritaria, selezionando casualmente un sottoinsieme di dati per bilanciare meglio la distribuzione.
  - Se la distribuzione delle classi rientra nei limiti definiti dalle soglie, non viene applicato alcun metodo di bilanciamento, poiché il dataset è considerato già sufficientemente equilibrato.

5 Valutazione del modello

La valutazione delle prestazioni del modello avviene considerando le seguenti metriche:

- **Accuratezza**: misura la proporzione di casi classificati correttamente rispetto al numero totale di oggetti nel dataset. Per calcolare questa metrica, si divide il numero di predizioni corrette per il numero totale di predizioni effettuate dal modello.

$$\text{Accuratezza} : \frac{\text{predizioni corrette}}{\text{predizioni totali}}$$

- **Precisione**: nella classificazione multiclasse, è la frazione di istanze correttamente classificate come appartenenti a una specifica classe rispetto a tutte le istanze che il modello ha predetto appartenere a quella classe. In altre parole, la precisione misura l'abilità del modello di identificare correttamente le istanze di una classe c.

  Per cui la precisione calcolata sulla classe c è:

  $$\text{Precisione}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}$$

  dove:

  - TP_c (True Positive) è il numero di istanze correttamente classificate come appartenenti alla classe c.
  - FP_c (False Positive) è il numero di istanze che non appartengono a una specifica classe, ma che il modello ha erroneamente classificato come appartenenti a quella classe.

  Per calcolare il valore complessivo della precisione del modello, si effettua la media dei valori di precisione ottenuti per ciascuna classe (**Macro-average**):

  $$\text{Precisione} = \frac{\text{Precisione}_{c_1} + \cdots + \text{Precisione}_{c_N}}{N}$$

- **Recall**: nella classificazione multiclasse, è la frazione di istanze appartenenti a una determinata classe c che il modello ha correttamente classificato, rispetto al numero totale di istanze che effettivamente appartengono a quella classe.

  Per cui la recall calcolata sulla classe c è:

  $$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$$

  dove:

  - FN_c (False Negative) è il numero di istanze che non appartengono a una specifica classe e che sono state correttamente classificate come non appartenenti a quella classe.

  Per calcolare il valore complessivo della recall del modello, si effettua la media dei valori di recall ottenuti per ciascuna classe (**Macro-average**):

  $$\text{Recall} = \frac{\text{Recall}_{c_1} + \cdots + \text{Recall}_{c_N}}{N}$$

- **F1-Score**: nella classificazione multiclasse è una metrica che bilancia la precisione e la recall di un modello di classificazione.

  L'F1-Score calcolata sulla classe c è:

  $$\text{F1-Score}_c = 2 \cdot \frac{\text{Precisione}_c \cdot \text{Recall}_c}{\text{Precisione}_c + \text{Recall}_c}$$

  Per calcolare il valore complessivo del F1-Score del modello, si effettua la media dei valori di F1-Score ottenuti per ciascuna classe (**Macro-average**):

  $$\text{F1-Score} = \frac{\text{F1-Score}_{c_1} + \cdots + \text{F1-Score}_{c_N}}{N}$$

6 Risultati

Di seguito saranno presentati i risultati ottenuti su ciascun dataset preso in esame. Principalmente, si effettueranno confronti tra i risultati ottenuti con le due funzioni di attivazione: ReLU e tanh. Successivamente, verranno effettuati eventuali approfondimenti utilizzando la funzione di attivazione che fornisce le migliori prestazioni.

6.1 Dry Bean Dataset

6.1.1 ReLU

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 256 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L1 con un parametro $\lambda = 0.0015$. Il tempo totale impiegato per completare la cross-validation è stato di 13 minuti e 44 secondi.

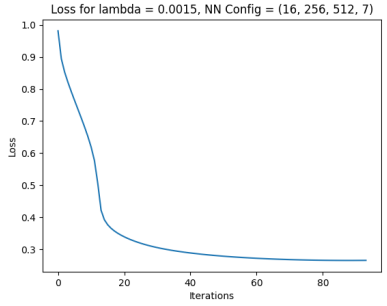

Figura 5: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	90.5131 %
Precisione	91.5242 %
Recall	90.4944 %
F1-Score	0.9091

Tabella 1: Valori ottenuti con il test set

6.1.2 tanh

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 512 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L1 con un parametro $\lambda = 0.0015$. Il tempo totale impiegato per completare la cross-validation è stato di 16 minuti e 58 secondi.

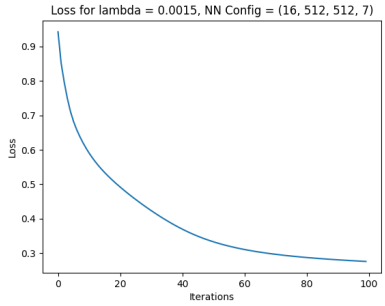

Figura 6: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	90.1809 %
Precisione	91.0885 %
Recall	89.4516 %
F1-Score	0.9007

Tabella 2: Valori ottenuti con il test set

6.1.3 Feature selection

Dopo aver osservato che la funzione di attivazione *ReLU* porta a risultati migliori, si è deciso di applicare la tecnica di feature selection per verificare se fosse possibile ottenere prestazioni accettabili utilizzando solo le features che mostrano una certa influenza (correlazione) sulla variabile target. È stata scelta una soglia di correlazione pari al 30%. La riduzione del numero di features consente di diminuire il tempo necessario per l'esecuzione della cross-validation. Inizialmente erano presenti 16 features, ma dopo l'applicazione della feature selection ne sono state mantenute 10.

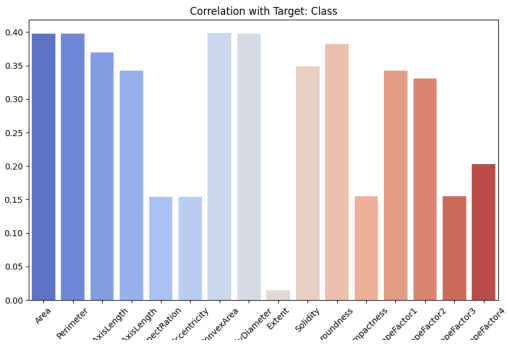

Figura 7: Correlazione tra le features e il target

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 64 neuroni
- Secondo livello: 128 neuroni

Inoltre, il miglior risultato è stato ottenuto senza regolarizzazione. Il tempo totale impiegato per completare la cross-validation è stato di 1 minuti e 58 secondi.

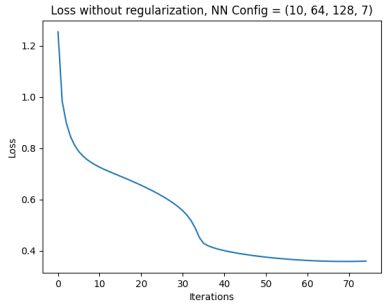

Figura 8: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	88.1506 %
Precisione	89.2508 %
Recall	84.7573 %
F1-Score	0.8607

Tabella 3: Valori ottenuti con il test set

6.2 Sloan Digital Sky Survey - DR18 Dataset

Prima di procedere con l'analisi dei risultati, è opportuna una premessa. Data la grande dimensione del dataset, si è scelto di effettuare la cross-validation considerando esclusivamente due configurazioni di rete neurale, in cui i livelli nascosti sono composti dalle coppie di neuroni: (256, 512) e (512, 512). Questa scelta è motivata dalla necessità di garantire una capacità di apprendimento adeguata, evitando che la rete risulti **sottodimensionata** (modello con una capacità rappresentativa insufficiente rispetto alla complessità dei dati) rispetto alla complessità dei dati.

6.2.1 ReLU

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 512 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto senza regolarizzazione. Il tempo totale impiegato per completare la cross-validation è stato di 30 minuti e 28 secondi.

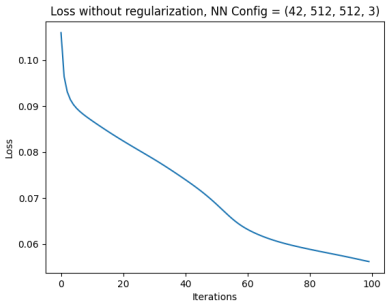

Figura 9: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	98.94 %
Precisione	98.6440 %
Recall	98.0444 %
F1-Score	0.9834

Tabella 4: Valori ottenuti con il test set

6.2.2 tanh

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 512 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L2 con un parametro $\lambda = 0.01$. Il tempo totale impiegato per completare la cross-validation è stato di 1 ora, 5 minuti e 5 secondi.

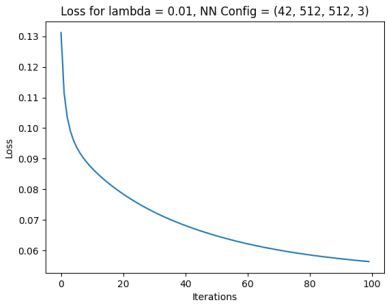

Figura 10: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	98.955 %
Precisione	98.6446 %
Recall	98.1026 %
F1-Score	0.9837

Tabella 5: Valori ottenuti con il test set

6.3 Feature selection

Dopo aver osservato che la funzione di attivazione *tanh* porta a risultati migliori, si è deciso di applicare la tecnica di feature selection per verificare se fosse possibile ottenere prestazioni accettabili utilizzando solo le features che mostrano una certa influenza (correlazione) sulla variabile target. È stata scelta una soglia di correlazione pari al 60%. La riduzione del numero di features consente di diminuire il tempo necessario per l'esecuzione della cross-validation. Inizialmente erano presenti 42 features, ma dopo l'applicazione della feature selection ne sono state mantenute 14.

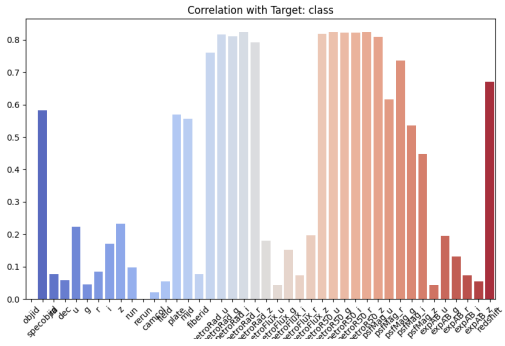

Figura 11: Correlazione tra le features e il target

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 256 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto senza regolarizzazione. Il tempo totale impiegato per completare la cross-validation è stato di 31 minuti e 2 secondi.

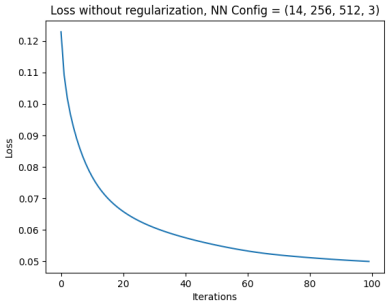

Figura 12: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	98.94 %
Precisione	98.6459 %
Recall	98.0201 %
F1-Score	0.9833

Tabella 6: Valori ottenuti con il test set

6.4 Air Quality and Pollution Dataset

6.4.1 ReLU

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 128 neuroni
- Secondo livello: 128 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L2 con un parametro $\lambda = 1$. Il tempo totale impiegato per completare la cross-validation è stato di 3 minuti e 43 secondi.

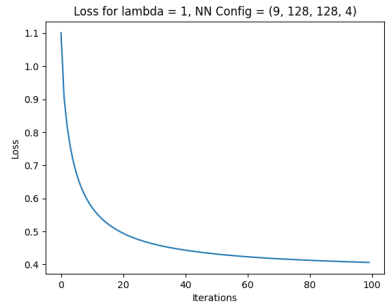

Figura 13: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	91.6 %
Precisione	89.5157 %
Recall	86.0417 %
F1-Score	0.8732

Tabella 7: Valori ottenuti con il test set

6.4.2 tanh

Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 256 neuroni
- Secondo livello: 512 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L1 con un parametro $\lambda = 0.001$. Il tempo totale impiegato per completare la cross-validation è stato di 4 minuti e 6 secondi.

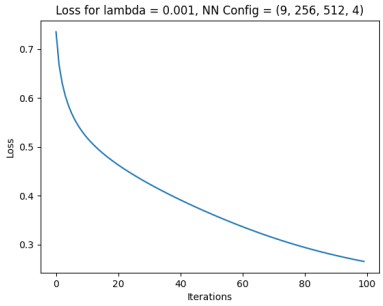

Figura 14: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	91.0 %
Precisione	88.8995 %
Recall	85.8542 %
F1-Score	0.8691

Tabella 8: Valori ottenuti con il test set

6.4.3 Feature selection

Con questo dataset non è stata applicata la tecnica di feature selection, poiché tutte le feature presentano una correlazione significativa con il target. E' stata scelta una soglia di correlazione pari al 30%.

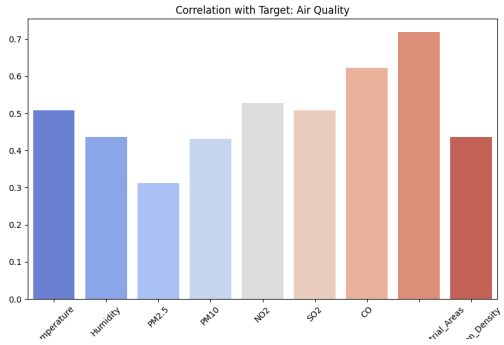

Figura 15: Correlazione tra le features e il target

6.4.4 Bilanciamento delle classi

Dopo aver osservato che la funzione di attivazione *ReLU* porta a risultati migliori, data la ridotta dimensione del dataset e la distribuzione sbilanciata delle classi, è stato deciso di applicare una tecnica di bilanciamento, utilizzando in particolare l'oversampling. Inizialmente, la distribuzione delle classi era la seguente:

Good: 1440, Moderate: 1080, Poor: 720 e Hazardous: 360

Dopo aver applicato l'oversampling, la distribuzione delle classi è stata uniformata come segue:

Good: 1440, Moderate: 1440, Poor: 1440 e Hazardous: 1440

Questo intervento ha portato a un miglioramento delle prestazioni del modello, in particolare per quanto riguarda le metriche di valutazione come recall e F1-score.
Dall'esecuzione della cross-validation è emerso che la configurazione ottimale della rete neurale prevede che i due livelli nascosti siano composti dal seguente numero di neuroni:

- Primo livello: 64 neuroni
- Secondo livello: 64 neuroni

Inoltre, il miglior risultato è stato ottenuto applicando la regolarizzazione L2 con un parametro $\lambda = 1$. Il tempo totale impiegato per completare la cross-validation è stato di 6 minuti e 52 secondi.

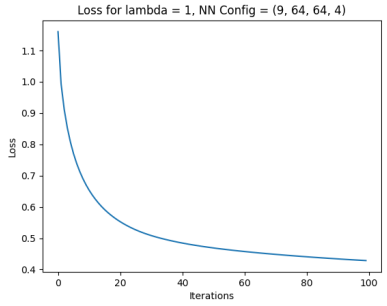

Figura 16: Andamento della loss function al variare delle epoche durante l'addestramento

Metrica	Valore
Accuratezza	91.6 %
Precisione	88.1625 %
Recall	89.2708 %
F1-Score	0.8859

Tabella 9: Valori ottenuti con il test set

7 Conclusioni

È opportuno concludere questo report con alcune osservazioni sui risultati ottenuti. Come prevedibile, il dataset *Sloan Digital Sky Survey - DR18* è quello che ha mostrato le prestazioni migliori, grazie alla grande quantità di dati disponibili per il training del modello. Analizzando i tre dataset, emerge che la funzione di attivazione ReLU è quella che, in generale, ha prodotto i risultati migliori, sebbene nel dataset con le prestazioni ottimali sia stata la funzione tanh a garantire i migliori risultati. Per quanto riguarda l'uso della regolarizzazione, non emerge una tecnica che prevalga nettamente sulle altre; tuttavia, attraverso la cross-validation, è stato possibile identificare il contesto in cui ciascuna tecnica di regolarizzazione fosse più appropriata, e determinare i parametri di regolarizzazione ottimali per ciascun caso.

Riferimenti bibliografici

[1] Nima Pourmoradi. *Dry Bean Dataset Classification*. https://www.kaggle.com/datasets/nimapourmoradi/dry-bean-dataset-classification

[2] Farid R. *Sloan Digital Sky Survey - DR18*. https://www.kaggle.com/datasets/diraf0/sloan-digital-sky-survey-dr18

[3] Mujtaba Mateen. *Air Quality and Pollution Assessment*. https://www.kaggle.com/datasets/mujtabamatin/air-quality-and-pollution-assessment