HOMEWORK 3

ADRIANA MAUGERI - 1000064219

Progetto di Machine Learning - Ingegneria Informatica, canale MZ

Descrizione del dataset

Il dataset usato è una variante del CIFAR-10, composto come esso da 60.000 immagini a colori suddivise in 10 classi, che rappresentano oggetti e animali in array tridimensionali (32, 32, 3).

Si differenzia per la presenza di etichette rumorose nel training set, presenti sottoforma di quadratini colorati uniformi, di un colore uguale per tutti gli elementi di una classe.

Questi provocano:

- Maggiore rischio di avere overfitting
- Perdita della capacità di generalizzazione

Metodologia usata

FASE 1: CARICAMENTO E PREPARAZIONE DEL DATASET

- **CARICAMENTO**: attraverso la funzione np.load() della libreria numpy carico i dati, che erano già divisi in training, validation e test set.

- **RIMOZIONE DEL RUMORE**: per ogni immagine di training, scorro tutti i possibili sottoblocchi di 5x5 pixel, ne calcolo la varianza e se questa è bassa, sostituisco il quadrato uniforme con il colore medio dei bordi

- **APPIATTISCO LE ETICHETTE**: la funzione ravel() trasforma le etichette da formato (n, 1) ad array monodimensionale di formato (n,)

Metodologia usata

FASE 1: CARICAMENTO E PREPARAZIONE DEL DATASET

- **CONVERSIONE A FLOAT:** con la funzione astype(), le immagini vengono convertite da intero a float. Questo è necessario per applicare normalizzazione e standardizzazione, che non funzionano correttamente con valori interi.

- **RESHAPE E NORMALIZZAZIONE:** reshape() trasforma ogni immagine da matrice (32, 32, 3) a vettore piatto di 3072 elementi per usarlo in MLP, che lavora su input vettoriali. Dividendo per 255 riduco i valori in un range [0,1]

- **STADARDIZZAZIONE**: : Standardizzo i dati usando fit_transform() che calcola la media e la deviazione standard sul training set e poi trasformo i dati del validation e test set nello stesso modo attraverso .transform()

Metodologia usata

FASE 2: MODEL SELECTION

Creo una grid di parametri, che verrano combinati per cercare la combinazione migliore

- hidden_layer_sizes : definisce quanti layer nascosti usare e quanti neuroni per ciascuno ((100,) un hidden layer con 100 neuroni - (100, 50) due hidden layer, il primo con 100 e il secondo con 50 neuroni ecc..)
- learning_rate_init controlla la grandezza dei passi fatti per aggiornare i pesi
- Solver indica l'alrgoritmo di ottimizzazione (adam)
- Batch_size indica quanti esempi usare ad ogni aggiornamento
- Activaction indica la funzione di attivazione (ReLU)

Metodologia usata

FASE 2: MODEL SELECTION E MLP

Per ogni combinazione di iperparametri, ho allenato e valutato sul validation set, utilizzando l'early stopping che permette di fermare l'allenamento prima del tempo, non appena il modello smette di migliorare.

early stopping:

- Alleno un epoca alla volta (max_iter = 1), dopo ogni epoca calcolo l'accuracy sul validation set e se non migliora per 10 epoche consecutive, interrompo l'allenamento.
- **warm_start** permette di procedere l'allenamento dallo stato attuale del modello, senza reinizzializzare i pesi.

Il modello con la migliore accuracy in validation viene applicato al test set.

Metodologia usata

CONFRONTO CON L'UTILIZZO DEGLI ENSEMBLE

La scelta di utilizzare i metodi ensemble, insieme all'utilizzo dell'early stopping e di regolarizzazione, è strettamente legata alla necessità di ridurre l'overfitting.

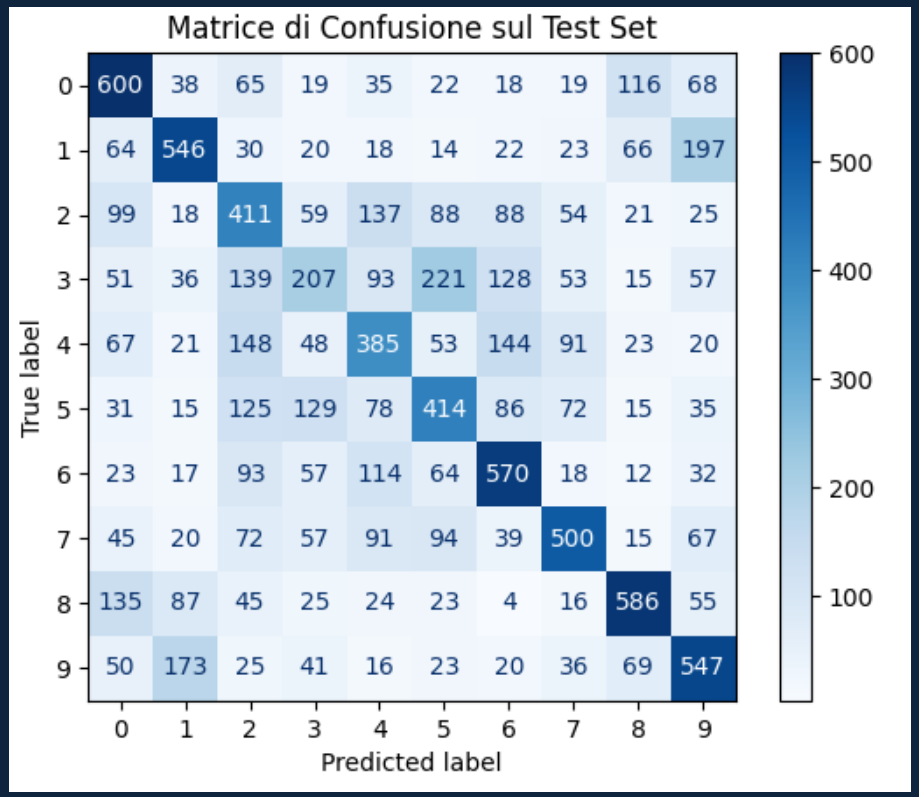
Allenare più modelli MLP su sottoinsiemi casuali del training set e combinare le predizioni (attraverso majoriting voting) fa si che il modello finale possa «mediare» gli errori, migliorando complessivamente la capacità di generalizzazione.

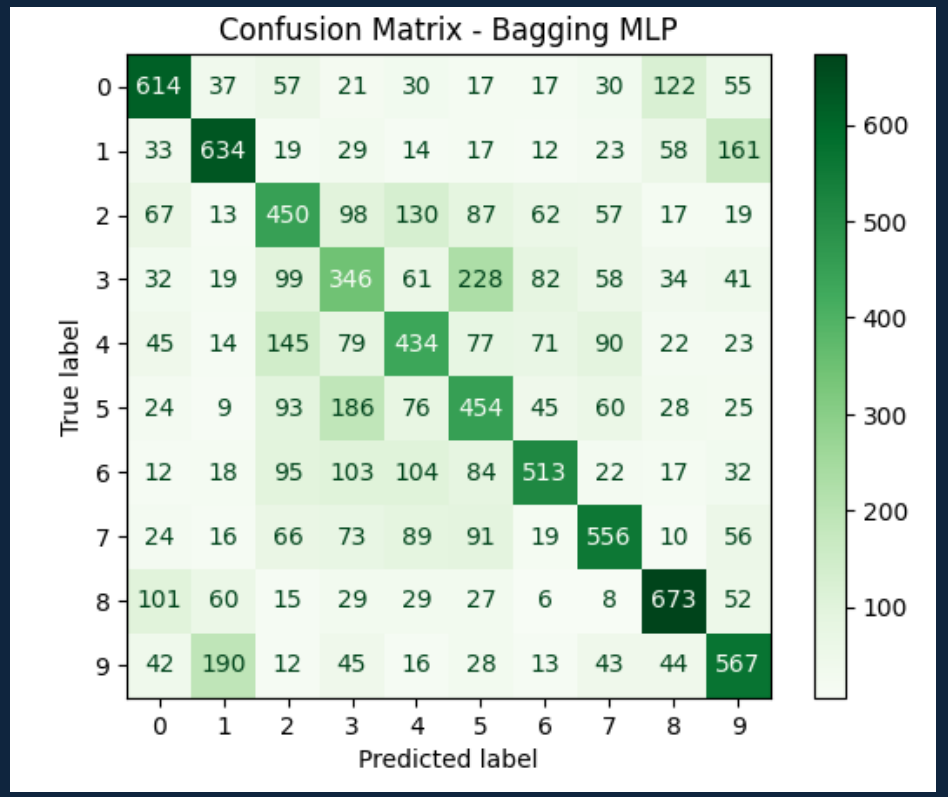
In particolare ho usato 5 MLP, addestrati su sottoinsiemi che usano 80% del training totale, campionato con rimpiazzo.

Questo ha permesso di confrontare le prestazioni rispetto al singolo modello, valutando miglioramenti in termini di accuratezza e bilanciamento degli errori tra le classi.

Metodologia usata

FASE 4: VALUTAZIONE DEL MODELLO FINALE





ACCURACY SU TEST: 0.4766

0.5241