

Tozzi Senesi Leonardo
7013640

Il lavoro è stato effettuato utilizzando JupyterLab.

Il foglio di lavoro è diviso in 9 blocchi (il numero del blocco è scritto alla prima riga di ciascuno).
Premessa: ho utilizzato i datasets (train_32x32.mat ; test_32x32.mat ; train_32x32.mat) reperibili online su [SVHN](http://svhn.song-da.net/).

Nel blocco 1 ci si occupa di caricare nell'ambiente di lavoro i datasets dati in formato .mat. In particolare questi erano già divisi tra dati di train, identificati da MatTrain e divisi in X e y, e dati di test, identificati da MatTest e divisi in X_t e y_t. Inoltre veniva messo a disposizione un ulteriore dataset extra, diviso in X_e e y_e, necessario per bilanciare le classi nel training set.

In questo blocco ci si occupa anche di trasformare y,y_t e y_e (vettori colonna) in y_train, y_test e y_extra (vettori riga). Come ultima cosa si stampano le dimensioni di X, X_t, y_train e y_test.

Nel blocco 2 si contano le occorrenze degli elementi nel dataset di train e in quello extra e infine, si calcola quanti elementi sono necessari per far sì che ciascuna classe abbia lo stesso numero di elementi.

Nel blocco 3 ci si occupa di suddividere i valori presenti in X_e in modo da ottenere vettori omogenei e aventi dimensione pari al numero di elementi necessari a rendere bilanciate le classi.

Nel blocco 4 si caricano nel notebook X_train, X_test, e gli extra in formato grayscale e se ne stampano le dimensioni.

Nel blocco 5 inizialmente suddividiamo il dataset di training in vettori omogenei (come fatto precedentemente con il dataset extra) e dopodiché ad ogni vettore andiamo a concatenare i vettori extra contenenti i valori corrispondenti, ottenendo così vettori omogenei e con lo stesso numero di elementi.

Nel blocco 6 andiamo a creare il dataset di training, che poi utilizzeremo, andando a “pescare” un elemento per volta da ciascun vettore omogeneo. Il dataset di training ottenuto avrà quindi il vettore contenente le etichette (YTRAIN) così formato: [1 . 2 . 3 10. 1. 2. 3.].

Nei blocchi 7 e 8 si aggiustano le dimensioni dei vari dataset, trasformandoli da tensori 3d a tensori 2d per renderli compatibili tra di loro.

Nel blocco 9 si effettua un ciclo *for* che aumenta i valori di k, considerando come dimensione 2^k quando $10 \leq k < 19$, mentre quando $k=18$ si pone la dimensione pari alla dimensione totale del training set. La dimensione del test set è sempre proporzionale a quella del training set ($subdim = \text{int}(dim/2.8141)$). Si definisce poi un classificatore *reg* addestrato con i dati di train (X_train, y_train) e testato con i dati di test (X_test, y_test). Successivamente si misura l'accuratezza e conseguentemente l'errore di predizione sul training set e sul test set.

Il blocco termina con il codice necessario a mostrare l'andamento di quest'ultimo in un grafico e riportando per ogni iterazione le dimensioni dei dati di train e di test.