

B003725 Intelligenza Artificiale (2019/20)

Studente: Giulia Di Costanzo — <2020-07-28 Tue>

Elaborato assegnato per l'esame finale

Istruzioni generali

Il lavoro svolto sarà oggetto di discussione durante l'esame orale e dovrà essere sottomesso per email due giorni prima dell'esame, includendo:

1. Sorgenti e materiale sviluppato in autonomia (non includere eventuali datasets reperibili online, per i quali basta fornire un link);
2. Un file README che spieghi:
 - come usare il codice per riprodurre i risultati sottomessi
 - se vi sono parti del lavoro riprese da altre fonti (che dovranno essere **opportunamente citate**);
3. Una breve relazione (massimo 4 pagine in formato pdf) che descriva il lavoro ed i risultati sperimentali. Non è necessario ripetere in dettaglio i contenuti del libro di testo o di eventuali articoli, è invece necessario che vengano fornite informazioni sufficienti a *riprodurre* i risultati riportati.

La sottomissione va effettuata preferibilmente come link ad un repository **pubblico** su [github](#), [gitlab](#), o [bitbucket](#). In alternativa è accettabile allegare all'email un singolo file zip; in questo caso è **importante evitare di sottomettere files eseguibili** (inclusi files `.jar` o `.class` generati da Java), al fine di evitare il filtraggio automatico da parte del software antispam di ateneo!

Riconoscimento di numeri civici

In questo esercizio si utilizzano implementazioni disponibili di Perceptron (p.es. [scikit-learn](#) in Python o [Weka](#) in Java) per classificare immagini di numeri civici, studiando l'andamento dell'errore di generalizzazione con il numero di esempi. Concretamente si utilizza il dataset [SVHN](#) (nel formato cropped e convertendo le immagini in grayscale) e l'algoritmo perceptron, riportando l'errore di predizione sul training set e sul test set, usando dimensioni del training set di 2^k , per k crescente da 10 fino al massimo valore compatibile con le risorse di calcolo disponibili (eventualmente limitando il numero di iterazioni dell'algoritmo). Si presti attenzione a garantire che, per ogni k , le classi nel training set siano bilanciate.