

Machine Learning ed applicazioni (Mod.B)

aa. 2018/2019 docente: R. Prevete

Progetto di esame

PARTE A (Comune a tutti).

- Progettazione ed implementazione di funzioni per simulare la propagazione in avanti di una rete neurale multi-strato con almeno: due strati di pesi, con la sigmoide come funzione di output dei nodi interni e l'identità come funzione di output dei nodi di output.
(**FACOLTATIVO**: permettere all'utente di implementare reti con più di uno strato di nodi interni e con qualsiasi funzione di output per ciascun strato)
- Progettazione ed implementazione di funzioni per la realizzazione della back-propagation per reti neurali multi-strato con almeno: due strati di pesi, con la sigmoide come funzione di output dei nodi interni e l'identità come funzione di output dei nodi di output, con la somma dei quadrati come funzione di errore.
(**FACOLTATIVO**: permettere all'utente di realizzare la back-propagation con più di uno strato di nodi interni, con qualsiasi funzione di output per ciascun strato e con qualsiasi funzione di errore derivabile rispetto all'output).

PARTE B.

Dato il dataset “minist” di immagini di cifre scritte a mano (<http://yann.lecun.com/exdb/mnist/>), scegliere uno dei seguenti punti

1. (*Difficoltà bassa*) Si consideri come input le immagine raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale con uno strato di unità interne al variare dei nodi interni e con funzione di errore cross-entropy. Scegliere e mantenere invariati tutti gli altri “parametri” come, ad esempio, le funzioni di output ed i parametri della Rprop. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)
2. (*Difficoltà bassa*) Si consideri come input le immagine raw del dataset mnist. Si ha, allora,

un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la discesa del gradiente con momento come algoritmo di aggiornamento dei pesi, si studi l'apprendimento di una rete neurale con un solo strato interno al variare di eta e del momento per almeno 3 diverse dimensioni dello strato interno. Scegliere e mantenere invariati tutti gli altri "parametri" come, ad esempio, le funzioni di output. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)

3. (*Difficoltà medio-bassa*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale con un solo strato interno di nodi confrontando il caso in cui l'input è costituito dalle immagini raw del dataset mnist, con il caso in cui l'input è costituito da una trasformazione delle immagini raw tramite Principal Component Analysis (PCA). Scegliere opportunamente il numero di nodi interni, ed il numero di componenti principali della PCA. Scegliere e mantenere invariati tutti gli altri "parametri" come, ad esempio, le funzioni di output e la funzione di errore. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)
4. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale con un solo strato interno di nodi confrontando il caso in cui l'input è costituito da una trasformazione delle immagini raw del dataset mnist tramite una rete autoassociativa (paragrafo 8.6.2 del Bishop), con il caso in cui l'input è costituito da una trasformazione delle immagini raw tramite Principal Component Analysis (PCA). Scegliere opportunamente il numero di nodi interni, ed il numero di componenti principali della PCA. Scegliere e mantenere invariati

tutti gli altri “parametri” come, ad esempio, le funzioni di output e la funzione di errore. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)

5. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie (ad esempio, $N=200$). Si fissi la discesa del gradiente come algoritmo di aggiornamento dei pesi, ed una rete neurale con un unico strato di nodi interni. Si scelgano gli iper-parametri del modello, cioè eta della regola di aggiornamento ed il numero di nodi interni, sulla base di un approccio di cross-validation k-fold (ad esempio $k=5$). Scegliere e mantenere invariati tutti gli altri “parametri” come, ad esempio, le funzioni di output e la funzione di errore. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`)
6. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie (ad esempio, $N=200$). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi, ed una rete neurale con un unico strato di nodi interni. Si scelgano gli iper-parametri del modello, cioè i parametri della Rprop ed il numero di nodi interni, sulla base di un approccio di cross-validation k-fold (ad esempio $k=5$). Scegliere e mantenere invariati tutti gli altri “parametri” come, ad esempio, le funzioni di output e la funzione di errore. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
7. (*Difficoltà media*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale con un 1, 2 e 3 strati interni di nodi confrontando il caso in cui si utilizza come funzione di output dei nodi la sigmoide con quello in cui si usa come funzione di output dei nodi la ReLu ($\max(0,a)$). Provare diverse scelte del numero dei nodi per gli strati interni. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in

matlab la funzione `imresize`).

8. (*Difficoltà medio-alta*) Si consideri come input le immagini raw del dataset mnist. Si ha, allora, un problema di classificazione a C classi, con $C=10$. Si estragga opportunamente un dataset globale di N coppie, e lo si divida opportunamente in training, validation e test set (ad esempio, 200 per il training set, 100 per il validation set, 100 per il test set). Si fissi la resilient backpropagation (RProp) come algoritmo di aggiornamento dei pesi (aggiornamento batch). Si studi l'apprendimento di una rete neurale con un solo strato di nodi interni e con la sigmoide come funzione di output al variare del criterio di stop. Facendo riferimento all'articolo "Early Stopping | but when? Lutz Prechelt, 1999", si considerino i due algoritmi di stop GL e PQ con differenti valori del parametro *alfa*. Considerare reti con un diverso numero di nodi interni. Se è necessario, per questioni di tempi computazionali e spazio in memoria, si possono ridurre (ad esempio dimezzarle) le dimensioni delle immagini raw del dataset mnist (ad esempio utilizzando in matlab la funzione `imresize`).
9. (*Difficoltà alta*) Nell'articolo "An Equivalence of Fully Connected Layer and Convolutional Layer, Wei Ma, Jun Lu, arXiv, 2017" è proposta l'implementazione di una rete feed-forward classica che si comporta come una rete convoluzionale. Studiare e descrivere gli aspetti teorici. Implementare tale rete almeno nei suoi aspetti principali e provarne il funzionamento sul dataset mnist.

Matlab come linguaggio di programmazione consigliato.