Sto lavorando anche io su un'applicazione della ANN per la predizione delle ordinate spettrali a corto periodo, a partire da time-histories a bassa frequenza (ottenute tramite simulazione numerica). Sto cercando di valutare diversi approcci per la fase di addestramento, per quantificare l'efficacia della ANN.

In poche parole, ho a disposizione un database di N-TOT records (registrati) da usare come training set. Per esempio, N-TOT = 501. L'idea è di dare in pasto alla ANN le ordinate spettrali a lungo periodo e farle fare una predizione delle ordinate spettrali a corto periodo (ad esempio per periodi T<0.75 s). Di conseguenza, ho iniziato con un addestramento classico: training di 50 o 100 ANNs diverse, ma con le stesse percentuali di training/validation/test (85%, 10% e 5% rispettivamente). Si fanno addestrare 50 o 100 reti per trovare quella migliore. Io avevo fatto addestrare 1000 reti per esser sicura.

Ho selezionato a priori i records per l'addestramento (95% N-TOT) e ho testato la predizione sul restante set di validazione (5% N-TOT) (Non capisco, quindi non hai il set di test? Non hai usato le percentuali 85%, 10% e 5% come hai scritto nella tua frase precedente?). La rete prescelta sarà poi quella che mi dà una performance migliore quando applicata al set di validazione. Diciamo che mi son ispirato pari pari all'esempio classico della doc di matlab (https://fr.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html) per evitare overfitting. La mia rete è una semplice feedforward a due layers. In effetti, anche aumentando il numero di neuroni nel primo/unico hidden layer, per alcuni periodi ottengo un coefficiente R2 di 0.7 (R2 è il coefficiente di correlazione?). Vorrei provare a migliorare la performance...

Leggendo il tuo rapporto (PS: complimenti! è scritto benissimo!! Grazie!!) mi è venuto in mente che effettivamente potrei semplicemente applicare il bootstrap **sperando che l'errore medio della predizione diminuisca** sui cortissimi periodi. È vero che l’errore quadratico medio della predizione che si ottiene utilizzando il bootstrap è minore rispetto a quello che si ottiene senza, soprattutto quando il set dei dati di training è scarso. Ma è quello che stai cercando? Il bootstrap serve per valutare l’incertezza della rete neurale. Quindi (per esempio nel caso che l’output del modello sia solo uno) otterrai un insieme di valori di quell’output.

Quindi, tralasciando i dettagli, vorrei chiederti due chiarimenti su due aspetti pratici:

1). Per il bootstrap, pensavo di considerare tutti i records (N-TOT) e definire a priori gli indici di training/validazione/test. Una volta fatto ciò, ognuna delle **50 o 100 ANNs** viene addestrata su un re-sampling (con ripetizione) del set di training originale, giusto? Dimenticati delle 50 o 100 ANNs precedenti. Crea i nuovi dataset di training per il bootstrap ad esempio io ho considerato B=100 nuovi datasets con resampling. Per ogni dataset di training devi addestrare 50 o 100 ANNs per trovare quella migliore. Siccome il tempo computazionale è molto elevato, io ne avevo fatte addestrare 25, ma puoi anche farne addestrare solo una decina. Ovviamente più ne fai, più la rete che trovi è migliore.

Il set di test viene mantenuto costante e quello di validazione idem. Sì.

Come calcoli poi la performance? Come nell’ANN senza bootstrap.

1. Crei B datasets di training con il bootstrap
2. Addestri ognuna delle B reti e fai la validazione con il dataset della validazione. Ottieni così B ANN
3. Con i dati del dataset del test, testi ognuna delle B ANN e calcoli la performance. Quindi ottieni B performances.

Era questa la domanda?

E soprattutto, potresti spiegarmi rapidamente il test k-fold?

Serve per fare una validazione incrociata. In questo caso metti insieme i dati di training e di validazione creando un solo dataset. Dividi i dati in k parti uguali (più è alto k più oneroso sarà il tempo computazionale). Ad esempio se k=5 avrai 5 datasets (D1, D2, D3, D4, D5) da usare per il training e la validazione. A turno (per k volte) consideri un diverso dataset di validazione e i rimanenti dasets per il training.

Ad esempio, se D1 al passo 1 lo usi per la validazione (Dval = {D1}), D2, D3, D4, D5 costituiranno il data set del training (Dtrain = {D2, D3, D4, D5}). A questo punto addestri la rete e calcoli la performance della rete.

Poi al passo 2, cambi set di validazione. Ora Dval = {D2} e Dtrain = {D1, D3, D4, D5}, addestri la rete e calcoli la performance.

Al passo 3, cambi ancora set di validazione. Ora Dval = {D3} e Dtrain = {D1, D2, D4, D5}, addestri la rete e calcoli la performance.

E così via per k volte, cioè fino a che hai usato per il set di validazione tutti i k datasets identificati.

Alla fine per avere una sola stima dell’errore quadratico medio (RMSE), consideri la media dei k risultati.

Il vantaggio della validazione k-fold è la riduzione del rischio di overfitting.

Ti allego una slide che avevo fatto per spiegare la k-fold cross validation.

Che percentuali di training/validation/test mi consiglieresti?

Questo aspetto è un po’ critico, io avevo utilizzato le percentuali suggerite da matlab, cioè: 70%, 15% e 15% per il training,validazione e test, rispettivamente.

2). Pensi sia il caso di aumentare il numero di layer della rete o di aumentare il numero di neuroni nascosti? Perché io ho una ventina di variabili di input (le circa 20 ordinate spettrali a lungo periodo, ottenute dalla simulazione numerica a basse frequenze) e devo stimare una decina circa di variabili di output (le ordinate spettrali a corto periodo).

Non lo so. Non mi sono mai trovata nella situazione di aumentare il numero di layer nella rete, quindi ti consiglierei di aumentare il numero di neuroni nascosti. È vero però che il tuo caso è più complicato: 20 inputs e 10 outputs sono tanti, il numero di neuroni sarà elevato. Per scegliere il numero di neuroni devi provare ad addestrare tante reti e poi scegliere quella che produce il minore RMSE sulla validazione.

Ricorda che il numero di parametri da stimare deve essere inferiore del numero di dati di training a disposizione e la rete scelta deve avere il minore RMSE sul set dei dati di valutazione.