

Report1 tesi

Data processing, sviluppo ANN, BNN e prima analisi.

Wasim Essbai

Contents

1	Dataset	2
1.1	Sorgente	2
1.2	Elaborazione	2
2	Reti neurali	3
3	Robustezza	4
3.1	Analisi BNN con MAE	4
3.2	Analisi BNN con MAPE	4
3.3	Analisi BNN con NLL	5
3.4	Considerazioni teoriche	5
	Bibliography	7

1 Dataset

1.1 Sorgente

Come anticipato, il dataset usato è una raccolta di segnali PPG, ECG e ABP reperibile a questo link dataset. Questo dataset in realtà è derivato da uno più grande che si chiama MIMIC I (circa 23GB) attraverso un processing e pulizie di vario genere (descritti in un paper che c'è nella pagina del link). Ora c'è addirittura MIMIC III che arriva 2.4T. Sono dataset molto grandi perché contengono rilevamenti su diversi pazienti (circa 10000 nel III) in diverse condizioni fisiche e di salute.

Siccome la tesi non è lo sviluppo di una rete in grado di stimare la pressione sanguigna ho usato il dataset del link che è pulito e soprattutto di dimensioni ridotte.

1.2 Elaborazione

Dal dataset scelto poi c'era bisogno di estrarre le features dai segnali, perché una predizione puntuale è sia inutile che poco fattibile.

Prima però di estrarre le features è stato necessario fare ulteriori pulizie sul dataset, che è pulito ma non abbastanza. Questo perché il dataset era stato pensato per usare sia PPG che ECG insieme per predire la pressione, probabilmente per loro andava bene così. Le pulizie sono state principalmente filtraggio e rilevamento di segnali dove non si riescono a rilevare i picchi. Poi ho preso segnali di durata minima pari a 2 minuti e massima a 7. Un esempio di record è in figura 1

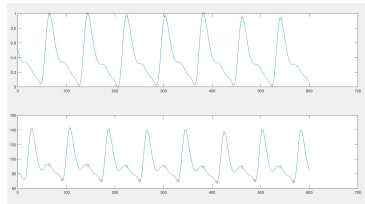


Figure 1: Esempio di coppia di segnali PPG(in alto) e ABP (in basso).

Dopo questa pulizia ho estratto le features seguendo un paper molto noto [1], che ho messo anche su Zotero, risale al 2013 ed è uno dei più citati nel dominio. Quasi tutti i lavori più recenti lo usano come base di partenza. Ci sono diversi modi di estrarre features da un segnale PPG per stimare la pressione sanguigna ma niente di consolidato, è ancora un tema di ricerca.

Una volta estratte le features poi ho fatto un secondo filtraggio, prendendo solo campioni che si riferiscono a valori normali di pressione (sistolica tra 80 e 120, diastolica tra 60 e 100). Questo perché nel dataset ci sono più valori 'normali' di pressione, quelli 'anormali' ci sono in grande numero però si distribuiscono in tanti modi e quindi è come se fossero outliers portando a performance

poco buone (MAE di 9 per la sistolica e 6 per la diastolica). Restrignendo i campioni a valori di pressione normali è stato sia ridotto il dataset, quindi training più veloce, ma anche performance migliori (MAE di 6 per la sistolica e 3 per la diastolica). Il dataset finale contiene 21801 campioni, ogni campione sono le features estratte da un periodo PPG, ovvero ad un ciclo cardiaco (Fig. 2). Nel dataset [1] ne usano poco più di 15000.



Figure 2: Esempio di un periodo PPG.

2 Reti neurali

Una volta che i dati erano pronti ho sviluppato due reti neurali usando tensorflow, una MLP e l'atra bayesiana. Ho cercato di tenere un'architettura semplice,

più che altro la rete bayesiana poi diventa molto pesante. L'architettura usata è la stessa, due hidden layer: primo da 15 e il secondo da 6 neuroni. Per la BNN ho modellato sia incertezza epistemica che aleatoria.

Ho allenato entrambe le reti con diverse funzioni di costo.

- ANN: MAE, MAPE
- BNN: MAE, MAPE, NLL (negative log likelihood)

Sorprendentemente le performance ottenute sono simili tra le due reti con le due reti con tutte le funzioni di costo tranne NLL, dove la bayesiana ha sia MAE che MAPE peggiori. Mi aspettavo risultati peggiori dalla BNN perchè ha veramente molti parametri in più, 33940 contro i 350 della ANN.

3 Robustezza

Ho fatto una prima analisi di robustezza seguendo un po' quello che era stato fatto della tesi precedente. Ho usato solo il rumore gaussiano bianco come alterazione, come range ho usato una deviazione standard da 0 a 2, con step di 0.1. Sembra basso ma in realtà i segnali PPG hanno ampiezze molto piccole e già con 1 il segnale è parecchio perturbato.

Dopo sul dataset alterato ho fatto le predizioni e calcolato le metriche di interesse. In particolare il MAE insieme alle formule discusse l'ultima volta (Eq. 1, Eq. 2), che misurano 'l'inefficienza' della rete. Come misura di incertezza ho usato la larghezza dell'intervallo di confidenza al 95%.

$$ineff1 = \frac{\sigma}{error + 1} + \frac{error}{\sigma + 1} \quad (1)$$

$$ineff2 = \frac{\sigma + 1}{error + 1} + \frac{error + 1}{\sigma + 1} \quad (2)$$

3.1 Analisi BNN con MAE

In figura 3 ci sono i grafici con i risultati, la linea rossa in ciascuno di essi indica le condizioni nominali.

Guardando questi grafici sembrerebbe che la rete sia molto robusta, cioè tolti alcuni picchi (cosa strana) poi il resto si comporta bene, è molto vicina alle condizioni nominali. Per c'è anche da dire che le BNN sono note per la robustezza al rumore gaussiano grazie alla loro natura stocastica.

3.2 Analisi BNN con MAPE

In figura 4 ci sono i grafici come prima. Di nuovo sembrerebbe che si comporti abbastanza bene. Però l'incertenza sembra essere sempre e bassa, tranne in un paio di casi, e questo potrebbe essere un comportamento poco desiderato.

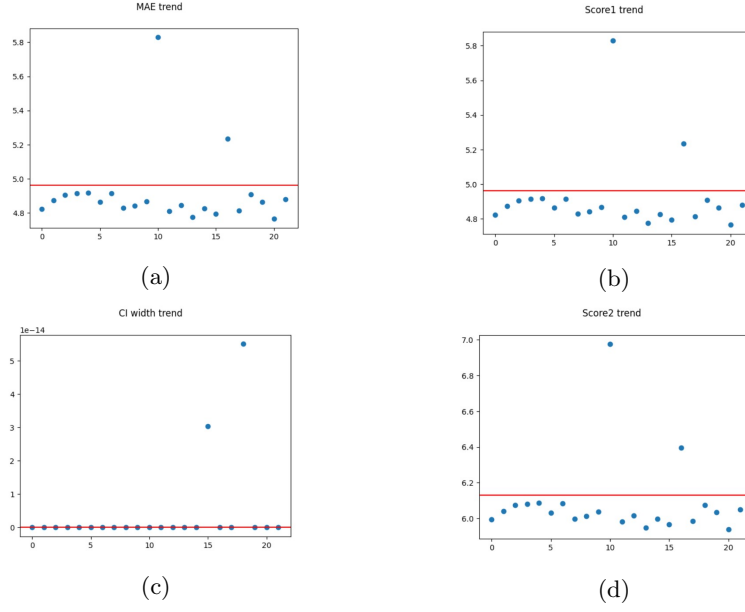


Figure 3: Risultati predizioni sul test set alterato per la BNN con MAPE.

3.3 Analisi BNN con NLL

In questo caso ci sono risultati più interessanti. La rete si comporta molto male a livello di MAE, cosa che era attendibile visto che già in fase di training aveva un MAE basso. In figura 5 ci sono 4 grafici che mostrano questo comportamento. La cosa sorprendente è che fa molto meglio in termini di eff1 ed eff2 rispetto alle condizioni nominali e questo mi sembra più curioso.

Ovviamente queste sono solo prime analisi fatte e sarebbe da investigare meglio, però visto che sono i primi risultati concreti valeva la pena analizzarli.

3.4 Considerazioni teoriche

Le due formule Eq. 1 ed Eq. 2 misurano una sorta di inefficienza della rete. Le due formule sembrerebbe che si comportino in modo molto simile, tranne per il fatto che Eq. 2 può essere invertita senza problema, quindi essere usata come misura di bontà. Però forse non è tanto utile come cosa, nel senso che tutte le metriche su reti neurali tendono a valutare quanto la rete performa male e non il viceversa. Tra due Eq. 1 secondo me ha più senso, anche se non invertibile. Ha senso perché è come se fosse un'estensione dell'errore classico, ovvero mettendosi nel caso deterministico con $\sigma = 0$ di ritorna ad avere solo la misura di errore, cosa che ne giustifica l'utilizzo. Poi anche il suo andamento è più 'deciso' nei primi tratti partendo da (0,0) per poi essere più smooth dopo, cosa che ha senso perché premia di più l'intorno (0,0), che è il punto ideale.

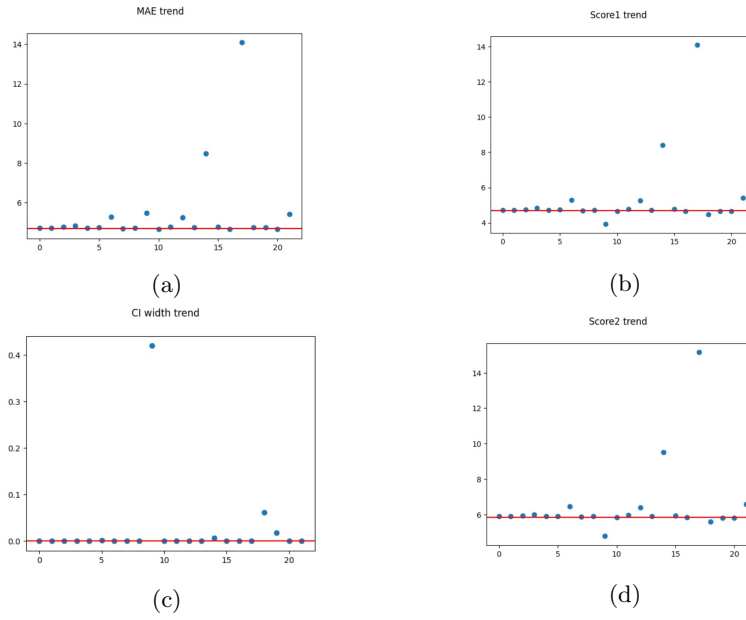


Figure 4: Risultati predizioni sul test set alterato per la BNN con MAPE.

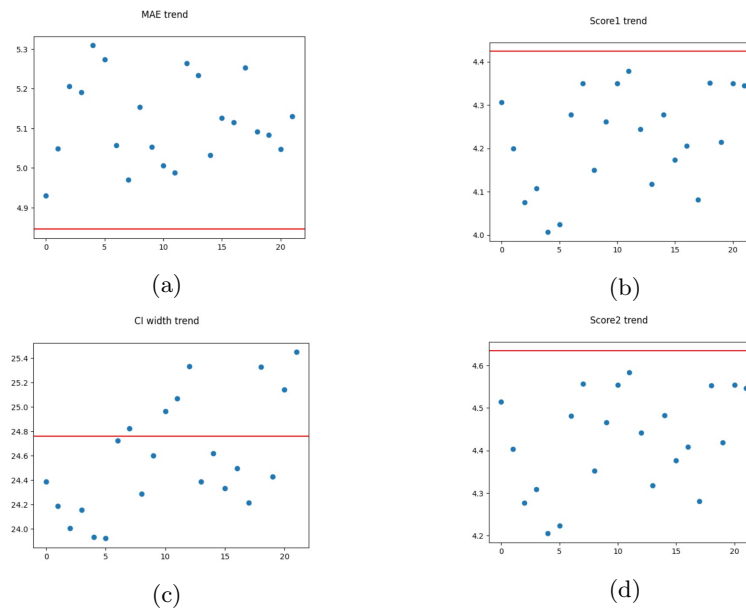


Figure 5: Risultati predizioni sul test set alterato per la BNN con NLL.

References

- [1] Y. Kurylyak, F. Lamonaca, and D. Grimaldi, “A neural network-based method for continuous blood pressure estimation from a ppg signal,” in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 280–283, 2013.