F.P. 10.3.2019

PROGETTO SABIK – RAPPORTO FINALE

Lo scopo di questo progetto è la realizzazione di un sistema di allarme che permetta di rilevare gli scavalcamenti di una ringhiera metallica, grazie a sensori di accelerazione disposti lungo la ringhiera che inviano i dati a una rete neurale che decreta se un'intrusione sta avendo luogo.

Il progetto è stato realizzato in collaborazione con la Albertolli SA di Taverne.

LO SCENARIO

Gli esperimenti sono stati condotti disponendo 6 accelerometri lungo una ringhiera a 2 m l'uno dall'altro ca. I sensori inviano dati alla frequenza di 26 Hz a un server. Questi dati vengono analizzati da una rete neurale che decreta se un'intrusione è in corso. Una GUI permette di visualizzare uno storico delle intrusioni avvenute.

La ringhiera davanti alla sede della Albertolli SA a Taverne

I sensori inviano dati solo quando sollecitati da eventi. Una volta attivati, inviano dati ininterrottamente per 5 secondi. Se durante questo tempo sono sollecitati nuovamente, questo intervallo di tempo si estende fino a 5 secondi dall'ultimo evento.

1. IL DATASET

I dati per allenare la rete sono stati raccolti eseguendo 28 intrusioni e 44 altri tipi di eventi (percotimenti, serie di pallonate, vento, oggetti che urtano contro la rete).

Le finestre successive di 65 valori dei sensori durante ogni evento costituivano il dataset per addestrare, validare e testare la rete. Questo dataset è stato raddoppiato, considerando che uno scavalcamento in un senso produce le stesse

accelerazioni che uno scavalcamento nel senso inverso, invertendo i valori delle accelerazioni lungo l'asse z.

Il dataset finale include 936 esempi di intrusione e 33384 esempi di non intrusione. Essendo le due classi sbilanciate, si è dato un peso diverso alle due classi durante l'addestramento.

Questo dataset può essere ampliato a piacimento per migliorare la performance del sistema di allarme.

Rotazione dei sensori

I sensori misurano le accelerazioni lungo tre assi x, y e z. Mentre z è sempre perpendicolare alla ringhiera, gli assi x e y, ortogonali tra loro, possono essere ruotati diversamente rispetto alla gravità.

Per allineare l'asse y con l'accelerazione di gravità, calcoliamo l'angolo di rotazione di un sensore quando invia dati a riposo utilizzando la formula:

$$\alpha = \arctan\left(-\frac{x}{y}\right) + b \cdot \pi$$

dove x e y sono le accelerazioni misurate lungo i rispettivi assi a riposo, b = 0 se y >= 0 e b = 1 se y < 0.

Questo angolo permette di trasformare le misure inviate dai sensori secondo la formula:

$$x' = x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$$
$$y' = x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$$

2. LA RETE NEURALE

Si sono confrontate le performance di diverse architetture di rete. I migliori risultati sono stati ottenuti utilizzando una rete neurale convoluzionale con maxpooling (MPCNN) con la seguente architettura:

LAYER	type	filters	neurons	kernel	activation
0	input	1	195		
1	convolution	5	191	5x1	ReLu
2	MaxPooling	5	95	2x1	
3	convolution	5	91		ReLu
4	MaxPooling	5	45	2x1	
5	convolution	20	41		ReLu
6	MaxPooling	20	10	4x1	
7	Flatten	1	200		
8	Dense	1	24	1x1	ReLu
9	Dense	1	2	1x1	softmax

L'input della rete è un vettore di 195 valori, corrispondenti a 65 valori successivi delle accelerazioni lungo gli assi x, y e z.

L'output indica la probabilità, secondo il classificatore, che un'intrusione stia avendo luogo.

Si è usata la *categorical crossentropy* loss e l'*rmsprop* optimizer.

Per evitare l'overfitting, si sono usate le seguenti strategie:

- si è utilizzata una finestra di input relativamente corta (65 valori)
- si è addestrata la rete utilizzando l'Early Stopping sul validation set.

Utilizzando altre architetture di rete (MLP, LSTM, MLP con dropout), i risultati erano meno buoni.

3. VALUTAZIONE

Per valutare l'efficienza della rete neurale si sono misurati quattro valori sul validation set:

i. l'accuratezza

ii. la matrice di confusione

iii. la ROC per le finestre

iv. la ROC per gli eventi.

Il valore più significativo per valutare se gli obiettivi che ci siamo prefissati sono stati raggiunti è il iv.

Nota che per etichettare correttamente un evento di intrusione basta che 1 finestra generata durante l'intrusione venga etichettata correttamente (per decidere se un'intrusione ha avuto luogo si calcola il valore massimo su tutte le finestre). Pertanto, la rete può classificare correttamente tutti gli eventi anche se ci sono dei falsi negativi.

i. Accuratezza

Un classificatore 'baseline' che etichettasse ogni finestra come non intrusione avrebbe un'accuratezza di 0.97.

La CNN raggiunge un'accuratezza di 0.996 sul validation set.

ii. Matrice di confusione

La matrice di confusione offre delle informazioni in più: ci indica, in particolare, quanti sono i falsi negativi (in basso a sinistra) e quanti sono i falsi positivi (in alto a destra).

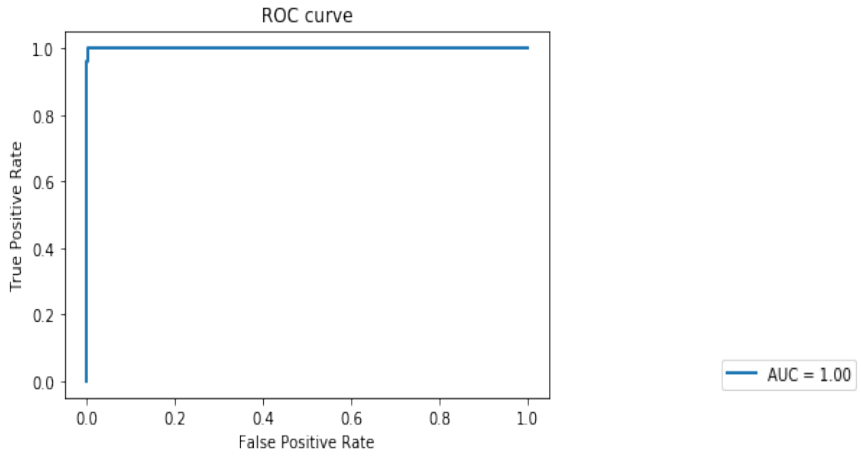
La matrice di confusione sul validation set è:

```
[[21960     0]
[   85   279]]
```

Questi risultati sono ottenuti usando un threshold di 0.5.

iii. ROC per le finestre

Una curva ROC permette di visualizzare essenzialmente le stesse informazioni della matrice di confusione, ma per tutti i possibili threshold. Ogni punto sulla curva rappresenta una coppia di valori: la percentuale di falsi positivi e la percentuale di veri positivi.



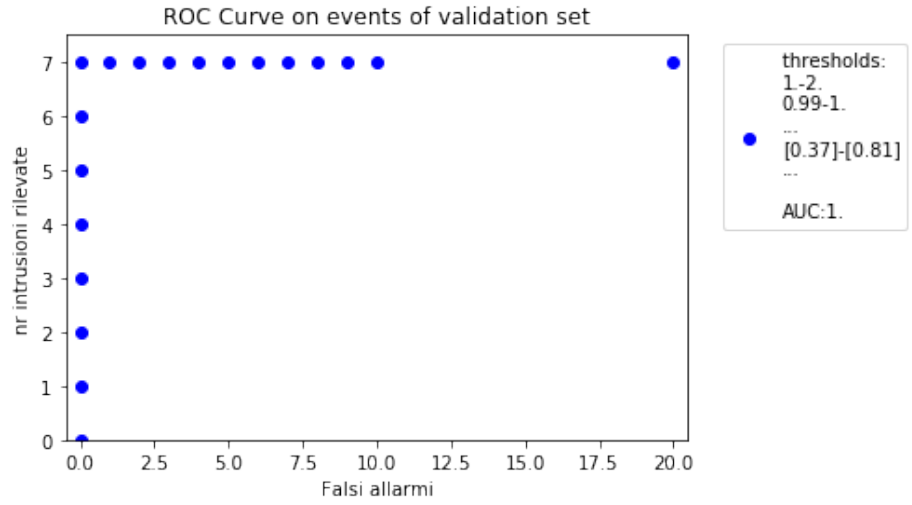
Nel nostro caso, ci importa in particolare che la curva parta in verticale (che significa che possiamo raggiungere un buon tasso di veri positivi pur mantenendo zero falsi positivi).

Nota che il threshold di 0.5 che dà il numero di VP e di FP della matrice di confusione corrisponde al punto in cui la curva ha l'altezza di 0.77 circa.

iv. ROC per gli eventi

Ma la metrica più conclusiva per decidere se il classificatore è in grado di distinguere propriamente intrusioni e non intrusioni è la curva ROC per gli eventi.

Essa indica il numero di intrusioni rilevate correttamente e il numero di falsi allarmi, con diversi threshold. La presenza di un puntino sull'angolo in alto a sinistra indica che tutti gli eventi sono rilevati correttamente con un dato threshold.



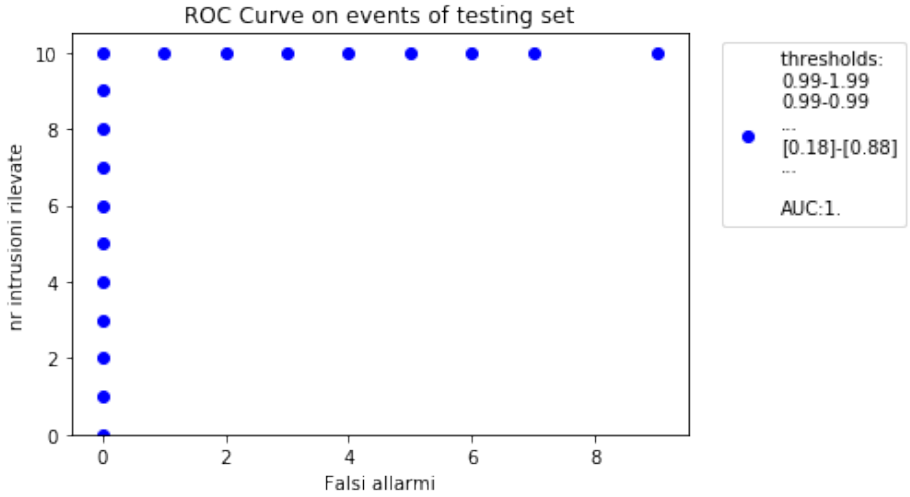
La ROC sugli eventi indica che, utilizzando un threshold compreso tra 0.37 e 0.81, tutti gli eventi sono stati classificati propriamente sul validation set.

Abbiamo deciso di utilizzare un threshold di 0.8 (che dava buoni risultati anche sul training set).

4. TESTING

La valutazione finale della rete neurale è stata fatta su un testing set indipendente (10 intrusioni e 10 non intrusioni).

Anche sul testing set, utilizzando un threshold di 0.8 (o qualsiasi altro valore tra 0.18 e 0.88), la rete neurale etichetta correttamente tutti gli eventi.



Il testing set includeva due scuotimenti e due pallonate.

Il 'Publisher'

Per far girare il programma in 'real time', un programma (il 'Publisher') raccoglie periodicamente gli ultimi dati dei sensori registrati sul server, li elabora e decide, utilizzando la rete neurale, se un'intrusione sta avvenendo. Il 'Publisher' invia infine queste informazioni alla GUI che tiene traccia di uno storico delle intrusioni.

CONCLUSIONI

Utilizzando un threshold ideale (0.8), la rete neurale sembra decretare correttamente tutte le intrusioni e non intrusioni del validation e del testing set Questo threshold generava invece un falso allarme sul training set (una pallonata).

Durante gli esperimenti condotti, è risultato che tra gli eventi di non intrusione, le pallonate erano il test più difficile per il classificatore.

LIMITAZIONI E PROSSIMI PASSI

Negli esperimenti condotti si è usato un dataset relativamente ridotto e non troppo difficile. Il dataset può essere ampliato a piacimento per metterla alla prova con dati più difficili e migliorarne la performance.

Nei presenti esperimenti la maggior parte degli scavalcamenti sono stati eseguiti dalla medesima persona, per la difficoltà di trovare candidati. Sarebbe utile includere più esempi di scavalcamenti di individui con altre corporature.