

UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE



Corso di Laurea magistrale in
Informatica

IL TITOLO DELLA TESI

Relatore: Relatore 1
Correlatore: Correlatore 1

Tesi di Laurea di:
Lorenzo D'Alessandro
Matr. Nr. 939416

ANNO ACCADEMICO 2020-2021

Dedica

Ringraziamenti

Questa sezione, facoltativa, contiene i ringraziamenti.

Indice

| | |
|--|------------|
| Ringraziamenti | ii |
| Indice | iii |
| 1 Introduzione | 1 |
| 1.1 I contenuti | 1 |
| 1.2 Organizzazione della tesi | 1 |
| 2 Stato dell'arte | 2 |
| 3 Classificatore | 3 |
| 3.1 Limitazioni recsys collaborative filtering | 3 |
| 3.2 Struttura classificatore | 3 |
| 4 Datasets | 5 |
| 5 Risultati | 6 |
| 6 Conclusioni | 7 |
| 6.1 Conclusioni | 7 |
| 6.2 Sviluppi futuri | 7 |
| Bibliografia | 8 |

Capitolo 1

Introduzione

Introduzione...

1.1 I contenuti

Spiegazione problema...

1.2 Organizzazione della tesi

Organizzazione tesi...

Capitolo 2

Stato dell'arte

Capitolo 3

Classificatore

3.1 Limitazioni recsys collaborative filtering

La principale limitazione dei recommender system dei collaborative filtering è definire il numero di users e items prima di iniziare l'aggiornamento/training del modello. Nel caso degli algoritmi di matrix factorization la matrice di input è composta da un numero di righe pari al numero di users, e un numero di colonne pari al numero di items. Nel caso dei modelli basati su reti neurali descritti nel Capitolo 2 è necessario dare in input ai modelli le lunghezze dei vettori di users e items che corrispondono rispettivamente al numero di users e items.

In entrambi se si vuole aggiungere un nuovo user/item è necessario ricompilare il modello ed eseguire nuovamente da zero la fase di training. Questo non è un problema in un ambiente desktop in cui è possibile aggiornare giornalmente lato server le preferenze di tutti gli utenti con una cadenza regolare (es. una volta al giorno).

Eseguire di nuovo il training da zero ogni volta che nuovi users/items vengono scoperti è un problema se si vuole fare il training direttamente su dispositivo mobile. In questo caso c'è un problema sia di dispendio energetico, sia di tempo effettivo per il training.

3.2 Struttura classificatore

Il modello proposto in questa tesi è una rete feed-forward fully-connected. Una rete feed-forward non contiene cicli nel suo grafo [1], mentre fully connected indica che ogni neurone in un layer è connesso a tutti i neuroni del layer successivo. Il numero di layer nascosti e il numero di neuroni in ogni layer è stato deciso nella fase di tuning descritta nel Capitolo 5. Il numero di neuroni nel layer di output è sempre 1 come è prassi nei problemi di classificazione binaria. La funzione di

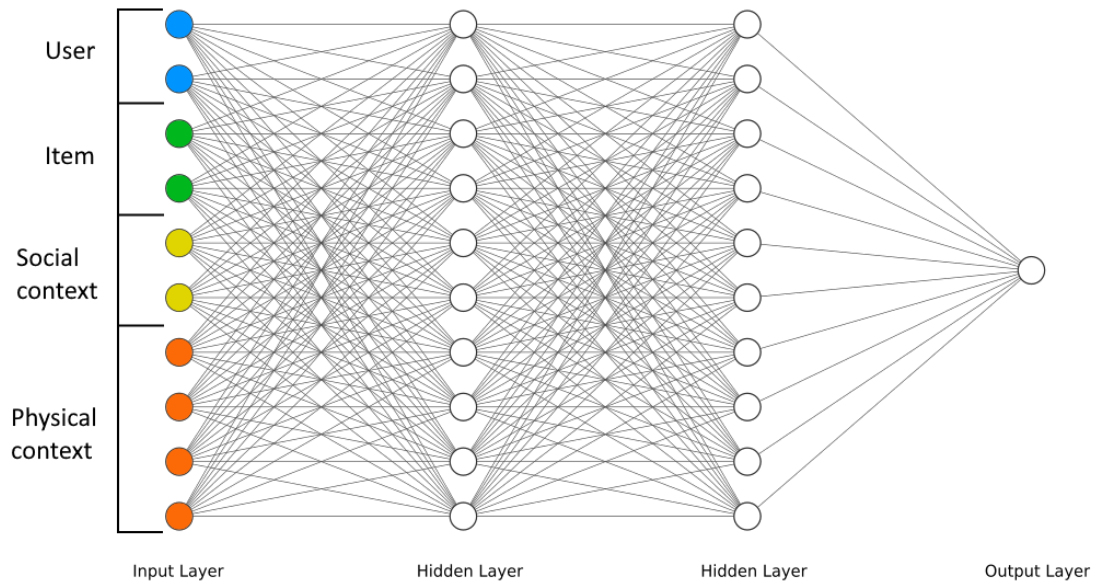


Figura 1: Schema rete feed-forward

attivazione scelta per i layer nascosti è ReLU (rectified linear unit), che è più plausibile biologicamente rispetto ad altre funzioni come sigmoide hyperbolic tangent, non viene saturata e aiuta a prevenire l'overfitting del modello [2]. La funzione di attivazione scelta per il layer di output è la sigmoide che limita l'output a valori compresi tra 0 e 1. La funzione di loss della rete è binary cross entropy.

Capitolo 4

Datasets

Capitolo 5

Risultati

Capitolo 6

Conclusioni

6.1 Conclusioni

Conclusioni...

6.2 Sviluppi futuri

Sviluppi futuri...

Bibliografia

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [2] Xavier Glorot, Antoine Bordes, and Y. Bengio. Deep sparse rectifier neural networks. volume 15, 01 2010.