

IA - Progetto

Introduzione

Iper-parametri:

Hidden Channels

| Indica il numero di neuroni nei layer nascosti della rete neurale.

Valori alti :

- Migliorano la capacità della rete di catturare pattern complessi.
- Possono portare a una migliore generalizzazione se combinati con una buona regolarizzazione.

Valori bassi :

- Possono limitare la capacità di apprendimento, soprattutto con dati complessi.
- Tuttavia, possono prevenire l'overfitting su dataset più piccoli.

Learning Rate

| Controlla quanto velocemente il modello aggiorna i pesi durante l'addestramento.

Valori bassi :

- Consentono aggiornamenti più stabili e una convergenza più precisa.

Valori alti :

- Permettono aggiornamenti rapidi, ma possono causare oscillazioni e mancata convergenza.

Batch Size

| Indica quanti campioni vengono elaborati simultaneamente prima di aggiornare i pesi.

Batch size alti :

- Stabilizzano l'ottimizzazione e riducono la varianza tra gli aggiornamenti.
- Spesso migliorano la generalizzazione.

Batch size bassi :

- Consentono aggiornamenti più frequenti e adattamenti più rapidi, ma con maggiore rumore.

Num Neighbors

| Indica quanti vicini vengono considerati nel grafo per ogni nodo. Un valore più alto include più contesto.

Valori alti :

- Aiutano con grafi complessi, migliorando la qualità delle rappresentazioni apprese.

Valori bassi :

- Possono ridurre il rumore, ma rischiano di perdere informazioni importanti.

Negative Sampling Ratio

| Indica il numero di esempi negativi generati per ogni esempio positivo durante l'addestramento.

Valori alti :

- Aiutano quando il dataset è sbilanciato, fornendo più esempi negativi per un migliore apprendimento.

Valori bassi :

- Sono sufficienti se il dataset è già ben bilanciato.

Metriche di valutazione:

AUC

| Area sotto la curva ROC.

F1-score:

| Media armonica di precision e recall.

Precision:

| Numero di esempi positivi classificati correttamente diviso per il numero totale di esempi classificati come positivi

Recall:

| Numero di esempi positivi classificati correttamente diviso per il numero totale di esempi positivi effettivi nel set di test

Loss:

| Funzione di perdita.

Metodo per la scelta delle configurazioni

Poiché AUC e F1-score devono essere massimizzati mentre la Loss deve essere minimizzata, ho normalizzato i valori per renderli comparabili su una scala comune tra 0 e 1:

- $AUC_{\text{norm}} = \frac{AUC - AUC_{\min}}{AUC_{\max} - AUC_{\min}}$
- $F1_{\text{norm}} = \frac{F1 - F1_{\min}}{F1_{\max} - F1_{\min}}$
- $Loss_{\text{norm}} = \frac{Loss - Loss_{\min}}{Loss_{\max} - Loss_{\min}}$

La Loss è invertita perché vogliamo che valori più bassi siano considerati migliori.

1. Creazione di uno Score combinato

Per ogni configurazione, ho calcolato uno **Score finale** come combinazione ponderata delle metriche normalizzate:

$$Score = 0.4 \cdot AUC_{\text{norm}} + 0.4 \cdot F1_{\text{norm}} + 0.2 \cdot Loss_{\text{norm}}$$

Le pesature sono state scelte così:

- **AUC (40%)**: Indica la capacità del modello di distinguere tra classi positive e negative.
- **F1-score (40%)**: Misura il bilanciamento tra Precision e Recall, cruciale per modelli con dati sbilanciati.
- **Loss (20%)**: Considerata meno rilevante rispetto a AUC e F1-score, ma comunque indicativa della qualità dell'addestramento.

1. Selezione delle 5 migliori configurazioni

Dopo aver calcolato lo Score per tutte le configurazioni presenti nei file, ho selezionato le **5 configurazioni con il punteggio più alto**.

Motivazione della scelta dei criteri

1. Perché AUC e F1-score?

- L'AUC è una misura globale della qualità del modello. Un AUC più alto significa che il modello separa meglio le classi.
- L'F1-score bilancia Precision e Recall, quindi assicura che il modello non sia troppo sbilanciato verso falsi positivi o falsi negativi.

2. Perché includere la Loss?

- La Loss misura direttamente l'errore del modello. Anche se non sempre un minor valore di Loss significa migliori metriche di valutazione, aiuta a prevenire selezioni di modelli con ottime metriche ma addestramento instabile.

3. Perché pesi diversi?

- AUC e F1-score sono più importanti per la qualità del modello finale, quindi ricevono un peso maggiore (40% ciascuno).
- La Loss ha un peso minore (20%) perché, sebbene importante, non sempre è correlata direttamente alla performance finale in classificazione.

Analisi dei dati

No estensioni

Optimizer: Adam

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Precision
64	0.005	128	[20, 10]	1	0.931089	0.796777	0.80924
128	0.001	64	[30, 15]	1	0.931229	0.796395	0.79970
128	0.001	128	[30, 15]	2	0.935655	0.788131	0.82940
32	0.005	128	[30, 15]	1	0.931749	0.803575	0.79159
64	0.005	128	[30, 15]	2	0.930632	0.777145	0.84740

Analisi Adam:

Le configurazioni migliori tendono ad avere un numero maggiore di Hidden Channels (64-128) e un Learning Rate relativamente basso (0.001-0.005). La combinazione di Batch Size 128 e un maggiore numero di vicini ([30, 15]) sembra favorire risultati migliori. L'AUC rimane costantemente alto, sopra 0.93, mentre l'F1-score si mantiene attorno a 0.78-0.80.

Optimizer: AdamW

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Precision
256	0.0005	128	[20, 10]	1	0.929335	0.798373	0.780017
128	0.0005	64	[20, 10]	2	0.940899	0.791337	0.851302
64	0.0050	64	[30, 15]	2	0.930821	0.787841	0.83865
128	0.0010	64	[30, 15]	1	0.929097	0.795486	0.79057
256	0.0010	128	[10, 5]	1	0.928361	0.775956	0.819526

Analisi AdamW:

Le migliori configurazioni utilizzano un numero elevato di Hidden Channels (128-256) e un Learning Rate molto basso (0.0005-0.001). Il numero di vicini varia leggermente, ma generalmente configurazioni con [20, 10] e [30, 15] sembrano avere performance elevate. L'AUC è generalmente superiore a 0.92, mentre l'F1-score si mantiene tra 0.77 e 0.80.

Optimizer: SGD

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Precision
256	0.01	64	[20, 10]	1	0.911210	0.739666	0.787370
128	0.01	64	[20, 10]	1	0.899091	0.747698	0.738512
256	0.01	128	[30, 15]	1	0.892388	0.730655	0.728991
256	0.01	64	[10, 5]	3	0.905186	0.657782	0.839071
128	0.005	64	[10, 5]	1	0.879099	0.715174	0.711941

Analisi SGD:

A differenza di Adam e AdamW, le configurazioni ottimali con SGD tendono ad avere un Learning Rate più alto (0.01). L'uso di 256 Hidden Channels sembra produrre buoni risultati, mentre il numero di vicini varia. L'AUC è generalmente inferiore rispetto ad Adam e AdamW, con un massimo di 0.91. Anche l'F1-score è leggermente più basso, attorno a 0.73-0.75. Inoltre, il Loss è più elevato rispetto agli altri ottimizzatori.

Considerazioni Finali

L'analisi degli ottimizzatori mostra che **Adam e AdamW sono le scelte migliori** per la link prediction su MovieLens, con performance superiori a **SGD** in termini di **AUC e F1-score**. Adam offre un miglior bilanciamento tra le metriche, mentre AdamW tende ad avere un AUC leggermente più alto. **SGD, invece, presenta una Loss più elevata e risultati peggiori**, suggerendo che potrebbe non essere l'ideale per questo compito.

Tra i parametri, **il numero di vicini (Num Neighbors) influenza le prestazioni**, con configurazioni come **[20,10] e [30,15]** che emergono come le più efficaci. Inoltre, **Batch Size 128** è ricorrente tra le migliori configurazioni, soprattutto con Adam e AdamW. **Adam e AdamW funzionano meglio con Hidden Channels elevati (128-256), un Learning Rate basso (0.0005-0.001) e Batch Size 128**, mentre **SGD preferisce un Learning Rate più alto (0.01) e può funzionare con meno Hidden Channels**.

Infine, **Num Neighbors e Negative Sampling Ratio influenzano le prestazioni**, con **[30,15] e un ratio di 1-2** che si rivelano parametri comuni tra le configurazioni migliori.

Estensione 5

Risultati per Adam

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
32	0.0010	64	[20, 10]	1	0.9282	0.7947	0.1104
32	0.0050	64	[30, 15]	1	0.9286	0.7925	0.1217
128	0.0010	128	[20, 10]	1	0.9319	0.8040	0.1633
128	0.0010	128	[30, 15]	2	0.9328	0.7675	0.1099
256	0.0005	64	[30, 15]	1	0.9295	0.7993	0.1591

Considerazioni su Adam

Adam predilige **Hidden Channels più bassi (32-128) e un Learning Rate di 0.001 o inferiore**, indicando che un aggiornamento più controllato migliora la performance. Il **Batch Size di 64** rimane la scelta più frequente, e il **Neg Sampling Ratio tende a 1**, suggerendo che, a differenza di SGD, Adam non necessita di molti esempi negativi per generalizzare bene.

Risultati per AdamW

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
64	0.0010	64	[30, 15]	2	0.9339	0.7660	0.0700
128	0.0010	64	[30, 15]	2	0.9334	0.7823	0.1153
64	0.0050	128	[30, 15]	1	0.9270	0.7946	0.1259
32	0.0010	64	[20, 10]	1	0.9228	0.7877	0.1022
128	0.0005	64	[30, 15]	2	0.9302	0.7823	0.1218

Analisi AdamW

AdamW funziona meglio con **Learning Rate più bassi (0.0005 - 0.0010)** e **Batch Size di 64**, mostrando maggiore stabilità rispetto a SGD. Le configurazioni migliori ottengono **AUC e F1-score più alti**, suggerendo che AdamW possa essere più efficace nel bilanciare precisione e richiamo. Il **Neg Sampling Ratio intorno a 2** sembra fornire buoni risultati, evitando sia il sovracampionamento che la scarsità di esempi negativi.

Risultati per SGD

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
128	0.01	64	[10, 5]	1	0.8939	0.7393	0.1805
256	0.01	64	[30, 15]	1	0.9005	0.7438	0.2298
256	0.01	64	[10, 5]	1	0.9050	0.7452	0.2630
256	0.01	64	[20, 10]	3	0.9073	0.7267	0.2559
256	0.01	64	[20, 10]	2	0.9101	0.7689	0.3579

Analisi SGD

Le migliori configurazioni per **SGD** mostrano che **un alto numero di Hidden Channels (128-256)** e un **Learning Rate di 0.01** sono scelte ottimali. Il **Batch Size di 64** appare come un buon compromesso. Il **Neg Sampling Ratio tra 1 e 3** suggerisce che un moderato numero di esempi negativi aiuta il modello senza peggiorarne la stabilità. Non sempre la Loss più bassa corrisponde alle migliori prestazioni, confermando che AUC e F1-score sono metriche più rilevanti.

Considerazioni finali

AdamW e Adam tendono a essere più stabili di SGD, con AdamW che ottiene il punteggio più alto complessivo. **SGD funziona meglio con Learning Rate più alti e Hidden Channels elevati**, mentre **Adam e AdamW preferiscono Learning Rate più bassi e Hidden Channels più contenuti**. Il **Batch Size ottimale è quasi sempre 64**. Il **Neg Sampling Ratio ideale varia**: SGD beneficia di valori più alti, mentre Adam e AdamW lavorano meglio con valori più bassi.

Estensione 20

Adam

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
128	0.0005	64	[20, 10]	1	0.9303	0.7879	0.0533
128	0.0010	128	[20, 10]	1	0.9310	0.7976	0.0981
32	0.0050	128	[20, 10]	1	0.9329	0.8040	0.1183
128	0.0010	64	[20, 10]	1	0.9302	0.7977	0.1173
32	0.0050	64	[20, 10]	2	0.9367	0.8013	0.1606

Analisi Adam

Adam si conferma l'ottimizzatore più **bilanciato e stabile**, con valori di **AUC e F1-score costantemente alti** e una Loss generalmente più bassa rispetto agli altri. Funziona bene con un **learning rate tra 0.0005 e 0.005**, hidden channels sia bassi (32) che alti (128), e batch size di **64 o 128**. Il **Negative Sampling Ratio = 1** è sufficiente nella maggior parte dei casi, anche se in alcuni scenari **valori più alti migliorano l'AUC**.

AdamW

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
32	0.001	64	[20, 10]	1	0.932	0.796	0.065
32	0.005	128	[20, 10]	1	0.931	0.802	0.102
128	0.001	64	[30, 15]	2	0.936	0.790	0.117
128	0.001	128	[30, 15]	3	0.932	0.765	0.098
128	0.001	64	[20, 10]	1	0.933	0.802	0.178

Analisi AdamW

AdamW mostra un comportamento simile ad Adam ma con un **learning rate più basso (0.001 - 0.005)** e una maggiore variabilità nei parametri ottimali. Risulta più efficace con **hidden channels tra 32 e 128**, con **batch size 64 o 128** e **Num Neighbors più flessibile ([20,10] o [30,15])**. Alcune configurazioni ottimali sfruttano **Negative Sampling Ratio = 2 o 3**, suggerendo che una maggiore quantità di esempi negativi possa migliorare le prestazioni in alcuni contesti.

SGD

Hidden Channels	Learning Rate	Batch Size	Num Neighbors	Neg Sampling Ratio	AUC	F1-score	Loss
256	0.005	64	[30, 15]	1	0.891	0.739	0.178
256	0.010	128	[20, 10]	1	0.892	0.742	0.208
256	0.010	64	[20, 10]	2	0.910	0.714	0.236
256	0.010	64	[30, 15]	3	0.904	0.701	0.245
128	0.010	64	[30, 15]	1	0.896	0.734	0.292

Analisi SGD

SGD ha un **AUC inferiore rispetto ad Adam e AdamW**, ma le sue migliori configurazioni puntano su **hidden channels elevati (128-256)** e **learning rate più alti (0.005 - 0.01)**. Batch size **64 o 128** funzionano meglio, mentre il Negative Sampling Ratio varia tra **1 e 3**. Questo suggerisce che, per competere con Adam, SGD ha bisogno di un modello più complesso e di un apprendimento più aggressivo.

Considerazioni Finali

Adam e AdamW sono gli ottimizzatori migliori per la link prediction su MovieLens, garantendo **stabilità e alte performance**. Adam è il più bilanciato, con buoni risultati già con **Negative Sampling Ratio = 1**, mentre AdamW è più sensibile agli esempi negativi e beneficia di **valori tra 2 e 3**. SGD, invece, richiede **hidden channels elevati e learning rate più alti** per ottenere risultati competitivi, ma resta meno efficace rispetto ad Adam e AdamW.