Neural Collaborative Filtering on MovieLens 1M

Kevin Bretz
Leiden University

Michail Protonotarios
Leiden University

Maftukhatul Q. Virati
Leiden University

Abstract

We implement Neural Collaborative Filtering (NCF) [2] on the MovieLens 1M dataset using implicit binary feedback. A sequential ablation over four hyperparameters (embedding dimension, negative sampling ratio, dropout and MLP architecture) using 12 experiments reveals that negative sampling ratio ρ is the primary performance driver. The best configuration (E=64, ρ=8, p=0.1, Wide-MLP) achieves a Test Recall@10 of 0.1149 and NDCG@10 of 0.1027, with Recall@10 increasing monotonically with ρ and diminishing returns suggesting an optimal value beyond the tested range.

CCS Concepts

• **Information systems** → **Collaborative filtering**; • **Computing methodologies** → *Neural networks*.

Keywords

Neural Collaborative Filtering, Recommender Systems, Implicit Feedback, Negative Sampling

1 Introduction

Collaborative filtering is the backbone of modern recommender systems. Classical Matrix Factorisation (MF) [5] models user–item affinities as inner products in a shared latent space, which is effective but inherently linear and unable to capture complex interaction patterns. Neural Collaborative Filtering (NCF) [2] addresses this limitation by augmenting the MF inner product with a Multi-Layer Perceptron (MLP), allowing the model to learn arbitrary non-linear interaction functions.

In this work we implement the NeuMF variant of NCF in PyTorch, apply it to the MovieLens 1M benchmark [1] and using the implicit feedback protocol and we evaluate it with Recall@10 and NDCG@10 using full-ranking. We conduct a sequential ablation over four hyperparameters to identify which design choices most influence performance.

2 Dataset and Preprocessing

2.1 MovieLens 1M

The MovieLens 1M dataset [1] contains 1,000,209 explicit ratings from 6,040 users on 3,706 movies, giving a sparsity of about 95.5%. NCF operates on implicit feedback rather than explicit ratings, so we binarise the data: ratings ≥ 4 (on a 1–5 scale) are treated as positive interactions (label = 1), yielding 575,281 positive pairs. Ratings below this threshold are discarded rather than used as negatives, following the convention in [2, 3], since a low rating does not necessarily imply disinterest.

2.2 Negative Sampling

Training directly on all un-rated items is impractical due to the extreme sparsity of the interaction matrix. For each positive pair (u, i) we sample ρ items that user u has *not* interacted with, drawn uniformly at random from the complement of u's positive set. These receive label = 0. At the baseline setting of ρ=4, this yields approximately 2.88 million training samples in total (≈575K positive + ≈2.3M negative).

2.3 Data Split

All samples (positives and sampled negatives) are split randomly into **70% train**, **15% validation** and **15% test**. The split is performed at the sample level, ensuring each partition contains a proportional mix of positive and negative labels.

3 Model Architecture

3.1 Overview

NeuMF [2] decomposes the user–item interaction function into two parallel branches that capture complementary aspects of the signal. Each branch maintains its own pair of embedding tables so that GMF and MLP can learn independent representations suited to their respective interaction functions. Both tables have the same embedding dimension E.

3.2 GMF Branch

Let $\mathbf{p}_u^{\text{GMF}}, \mathbf{q}_i^{\text{GMF}} \in \mathbb{R}^E$ denote the GMF user and item embeddings. The branch output is their element-wise (Hadamard) product:

$$\mathbf{o}^{\text{GMF}} = \mathbf{p}_u^{\text{GMF}} \odot \mathbf{q}_i^{\text{GMF}} \in \mathbb{R}^E. \tag{1}$$

This generalises standard MF: instead of collapsing the element-wise product to a scalar immediately, the vector is passed to the fusion layer, giving a learnable linear readout.

3.3 MLP Branch

The MLP branch concatenates separate user and item embeddings and passes them through a stack of fully-connected layers with ReLU activations and dropout:

$$\mathbf{z}_0 = \left[\mathbf{p}_u^{\text{MLP}} \parallel \mathbf{q}_i^{\text{MLP}}\right] \in \mathbb{R}^{2E}, \tag{2}$$

$$\mathbf{z}_l = \text{Dropout}(\text{ReLU}(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l)) \,. \tag{3}$$

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

The MLP can represent arbitrary non-linear interaction patterns that the linear GMF branch cannot express. We experiment with three tower configurations: **Shallow** [64, 32], **Default** [64, 32, 16, 8] and **Wide** [128, 64, 32].

3.4 Fusion and Prediction

The outputs of both branches are concatenated and mapped to a scalar prediction via a single linear layer followed by sigmoid:

$$\hat{y}_{ui} = \sigma\Big(\mathbf{h}^\top \big[\mathbf{o}^{\text{GMF}} \parallel \mathbf{z}_L\big]\Big), \tag{4}$$

where $\hat{y}_{ui} \in (0, 1)$ is the predicted interaction probability.

4 Training

4.1 Loss and Optimisation

We minimise binary cross-entropy (BCE) loss over all labelled pairs:

$$\mathcal{L} = -\sum_{(u,i,y)\in\mathcal{D}} \big[y \log \hat{y}_{ui} + (1-y)\log(1-\hat{y}_{ui})\big]. \tag{5}$$

All experiments were run on a LIACS PC-Lab machine equipped with an NVIDIA RTX 4060 GPU (8 GB VRAM). All models are trained with the Adam optimiser [4] with learning rate 10^{-3}, weight decay 10^{-5} and mini-batch size 1,024. Embedding and linear layers are initialised with Xavier uniform. Training runs for at most 30 epochs with early stopping: if the validation loss does not improve by more than 10^{-6} for 5 consecutive epochs, training halts and the best checkpoint is restored.

4.2 Convergence

All configurations converge smoothly without overfitting: the validation loss closely tracks the training loss throughout and the train validation gap remains small across all hyperparameter settings, indicating that dropout and early stopping together provide effective regularisation.

5 Results and Comparison

5.1 Evaluation Protocol and Metrics

We use a full-ranking protocol: for each user in the evaluation split, all items are scored and ranked in descending order.

Recall@10 measures the fraction of ground-truth positives recovered in the top 10 list.

NDCG@10 additionally rewards hits ranked higher, using a logarithmic discount $1/\log_2(k+1)$ at rank k, normalised by the ideal ranking.

5.2 Hyperparameter Ablation

To identify which design choices most influence performance, we conducted a sequential ablation over the four NCF specific hyperparameters Dropout Rate p, Embedding Dimension E, MLP Architecture and Negative Ratio ρ, varying one at a time while fixing the others at the best value found so far (starting baseline: E=32, ρ=4, p=0.2, Default-MLP), yielding 12 experiments in total.

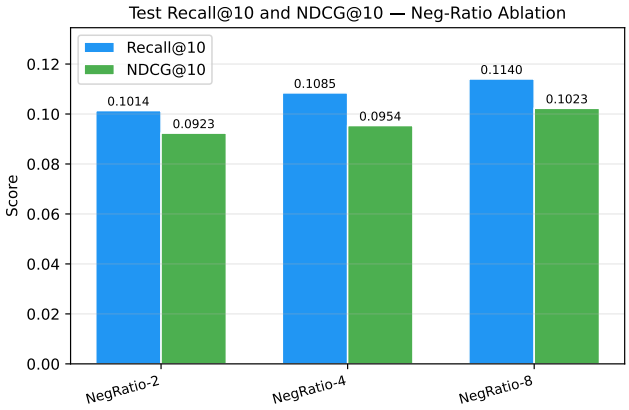

Figure 1: Test Recall@10 and NDCG@10 for each negative sampling ratio (other hyperparameters fixed at the best values from prior ablation phases). Both metrics increase monotonically with ρ but with diminishing returns between consecutive values.

Table 1: Test results for each negative sampling ratio (E=64, p=0.1, Wide-MLP). Bold: best per column.

ρ	Val R@10	Val N@10	Test R@10	Test N@10
2	0.1070	0.0951	0.1061	0.0950
4	0.1081	0.0981	0.1107	0.0989
8	**0.1156**	**0.1014**	**0.1149**	**0.1027**

The dominant finding is that **negative sampling ratio** is the primary driver of recommendation quality. Figure 1 and Table 1 show Test Recall@10 and NDCG@10 for each value of ρ (with the other hyperparameters fixed at the best values selected in prior phases). Performance increases monotonically — from Recall@10 = 0.106 at ρ=2 to 0.111 at ρ=4 and 0.115 at ρ=8 — with **diminishing returns**: the gain from ρ=2→4 (+0.005) is slightly larger than from ρ=4→8 (+0.004).

The loss curves in Figure 2 illuminate the mechanism. As ρ increases, the model converges to a substantially lower final validation loss (from ≈0.34 at ρ=2 to ≈0.19 at ρ=8) and requires more epochs to do so (best checkpoints at epochs 7, 10 and 14 respectively). The train–validation gap remains small throughout, indicating that the gain comes from richer supervision rather than overfitting. This suggests a performance ceiling exists at some $\rho > 8$ — beyond which additional negatives become redundant — but it lies outside our tested range.

6 Conclusions

We implemented NeuMF on MovieLens 1M and conducted a systematic sequential ablation over four hyperparameters (embedding dimension, negative sampling ratio, dropout and MLP architecture) using 12 experiments. The ablation establishes that negative sampling ratio is the most important hyperparameter: increasing ρ

yields consistent, monotonically improving performance with slight diminishing returns, pointing to an optimal value beyond the range we tested. The best configuration found (E=64, ρ=8, p=0.1, Wide-MLP) achieves Test Recall@10 of 0.1149 and NDCG@10 of 0.1027. The remaining hyperparameters have comparatively minor effects on performance.

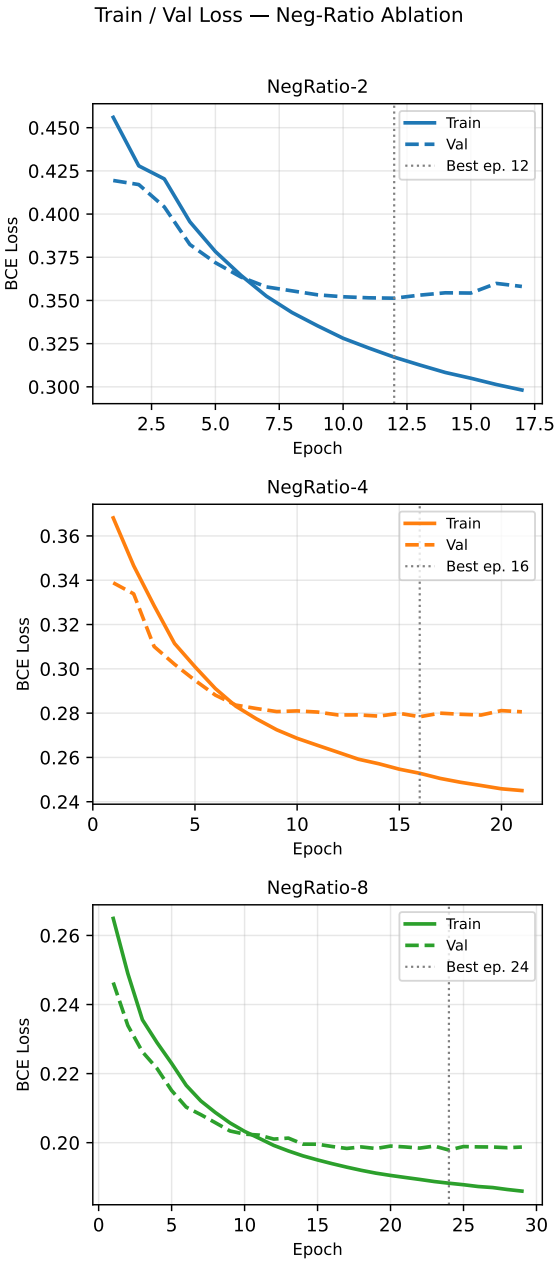

Figure 2: Train (solid) and validation (dashed) BCE loss for $\rho \in \{2, 4, 8\}$ (E=64, p=0.1, Wide-MLP). Higher ρ achieves a lower final validation loss and requires more epochs to converge.

Potential improvements. (1) *Pre-training*: training GMF and MLP branches independently before fine-tuning the fused model, as in the original paper [2], may improve convergence and final performance. (2) *Ranking-aware loss*: Bayesian Personalised Ranking [6] optimises a pairwise ranking objective directly and may yield better Recall and NDCG than pointwise BCE. (3) *Graph-based propagation*: methods such as LightGCN propagate collaborative signals over the user–item bipartite graph and have shown strong improvements on this benchmark.

Acknowledgments

This work was completed as part of Assignment 1 in the 2026 Recommender Systems master course at Leiden University.

References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 19:1–19:19. doi:10.1145/2827872

[2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, 173–182. doi:10.1145/3038912.3052569

[3] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*. 263–272. doi:10.1109/ICDM.2008.22

[4] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR '15)*.

[5] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8, 30–37. doi:10.1109/MC.2009.263

[6] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. 452–461.

A Complete Ablation Results

Table 2: Complete ablation results (all 12 runs). Metrics are Recall@10 and NDCG@10.

Config	E	ρ	p	MLP Layers	Val R@10	Val N@10	Test R@10	Test N@10
E32_NR4_D0.1_Default	32	4	0.1	[64, 32, 16, 8]	0.0743	0.0642	0.0756	0.0655
E32_NR4_D0.2_Default	32	4	0.2	[64, 32, 16, 8]	0.0581	0.0511	0.0546	0.0490
E32_NR4_D0.5_Default	32	4	0.5	[64, 32, 16, 8]	0.0365	0.0347	0.0353	0.0340
E16_NR4_D0.1_Default	16	4	0.1	[64, 32, 16, 8]	0.1104	0.0969	0.1107	0.0979
E32_NR4_D0.1_Default	32	4	0.1	[64, 32, 16, 8]	0.1084	0.0964	0.1091	0.0962
E64_NR4_D0.1_Default	64	4	0.1	[64, 32, 16, 8]	0.1116	0.0992	0.1147	0.1010
E64_NR4_D0.1_Shallow	64	4	0.1	[64, 32]	0.1101	0.0977	0.1099	0.0991
E64_NR4_D0.1_Default	64	4	0.1	[64, 32, 16, 8]	0.1099	0.0987	0.1135	0.0998
E64_NR4_D0.1_Wide	64	4	0.1	[128, 64, 32]	0.1138	0.1022	0.1157	0.1019
E64_NR2_D0.1_Wide	64	2	0.1	[128, 64, 32]	0.1070	0.0951	0.1061	0.0950
E64_NR4_D0.1_Wide	64	4	0.1	[128, 64, 32]	0.1081	0.0981	0.1107	0.0989
E64_NR8_D0.1_Wide	64	8	0.1	[128, 64, 32]	0.1156	0.1014	0.1149	0.1027