Output Layer

Estimate $\hat{y}_{ui}$ ← Training → y_{ui} Target

Training is performed by minimising the point-wise loss between $\hat{y}_{ui}$ and y_{ui}

Layer X

Neural Collaborative Filtering Layers

. . . .

Input vector, $\mathbf{x}_{ui}$, is fed through multi-layer perceptron architecture, applying non-linear activation functions. Model learns latent representations of users and items

Layer 1

Layer 1

$\mathbf{x}_{ui}$

User and Item embeddings, $\mathbf{p_u}$ and $\mathbf{q_i}$ are concatenated to form input vector, $\mathbf{x}_{ui}$

Embedding Layer

User Latent Vector

Item Latent Vector

$\mathbf{P}_{M \times K} = \mathbf{p}_{uk}$

$\mathbf{Q}_{N \times K} = \mathbf{q}_{ik}$

User and Item embeddings, $\mathbf{p_u}$ and $\mathbf{q_i}$, are unknown and are updated iteratively in training

Input Feature Layer

$\mathbf{v}_u^U$ | 3 | 0 | 5 | 5 | ...

$\mathbf{v}_I^I$ | 3 | 0 | 5 | 5 | ...

User (u)

Item (i)