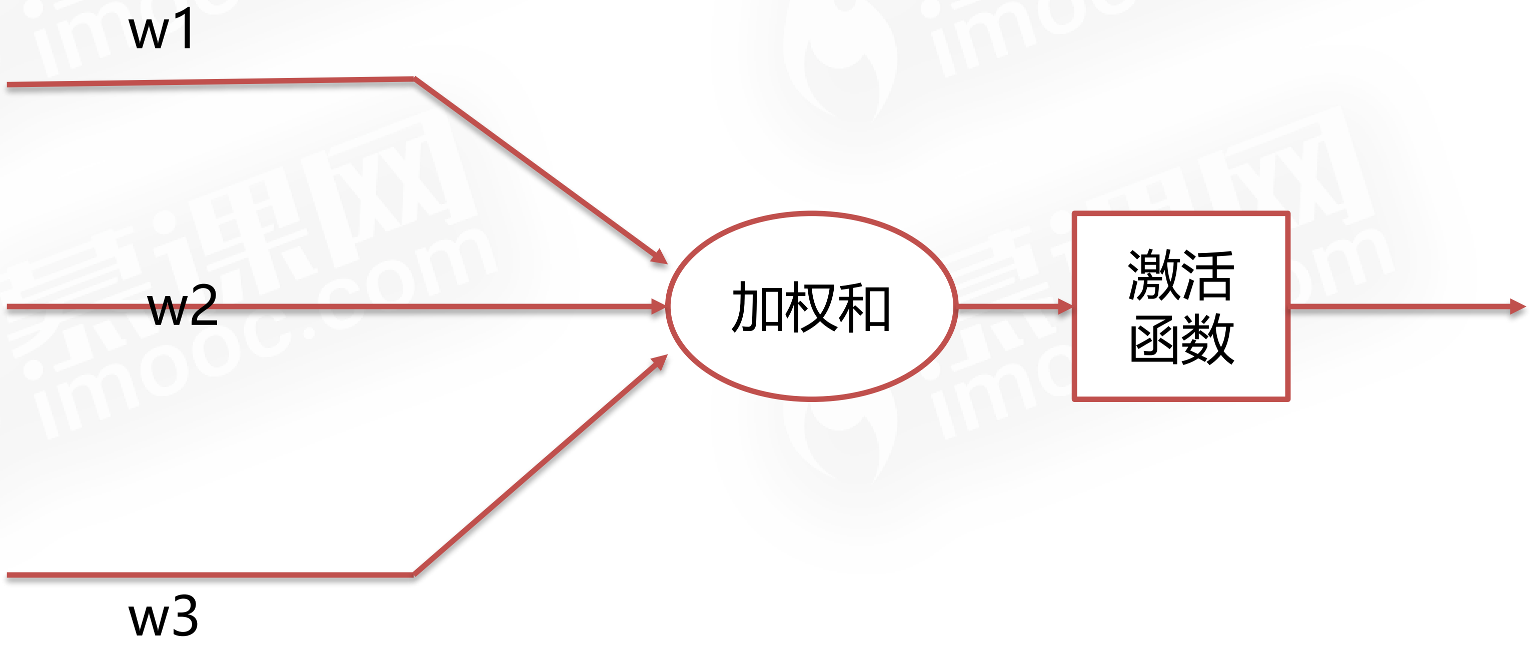
Personal Recommendation Algorithm

Main Flow

- 背景知识介绍之深度学习
- DNN网络结构与数学原理
- WD(wide and deep)网络结构与数学原理

神经元

- 什么神经元？

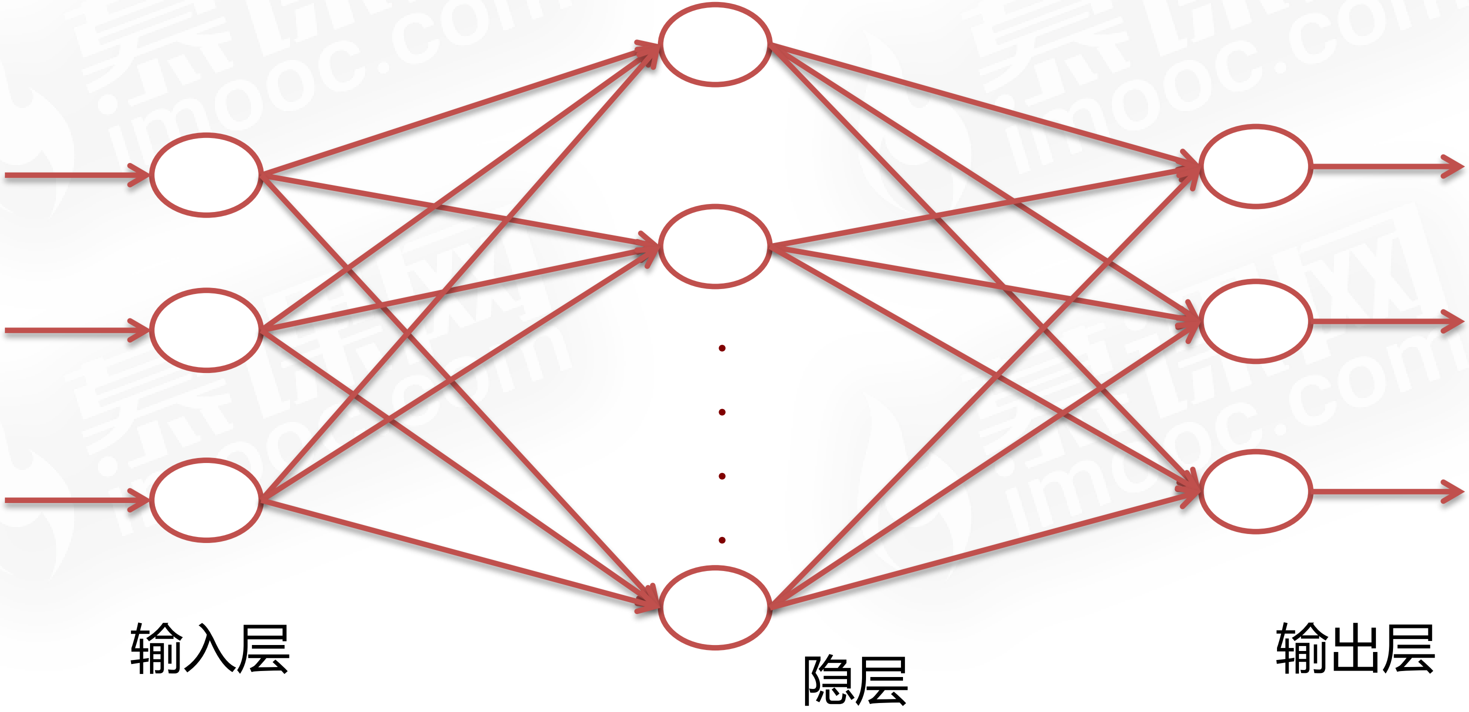


激活函数

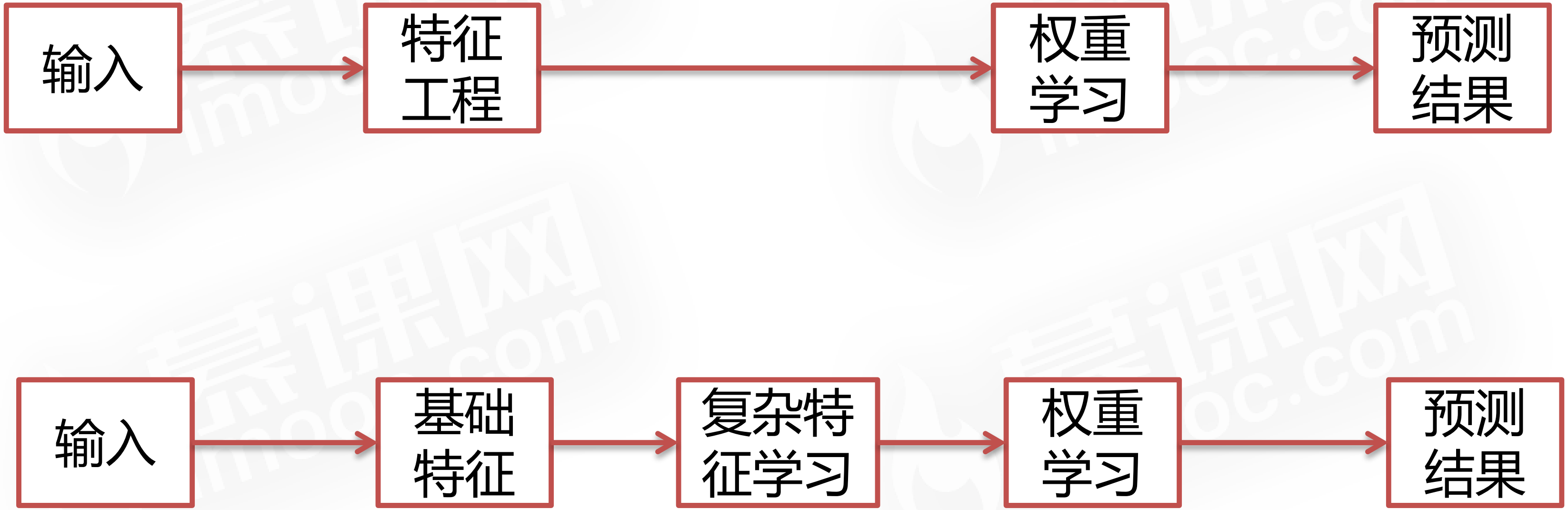
- sigmoid
- tanh
- relu

神经网络

- 什么是神经网络?

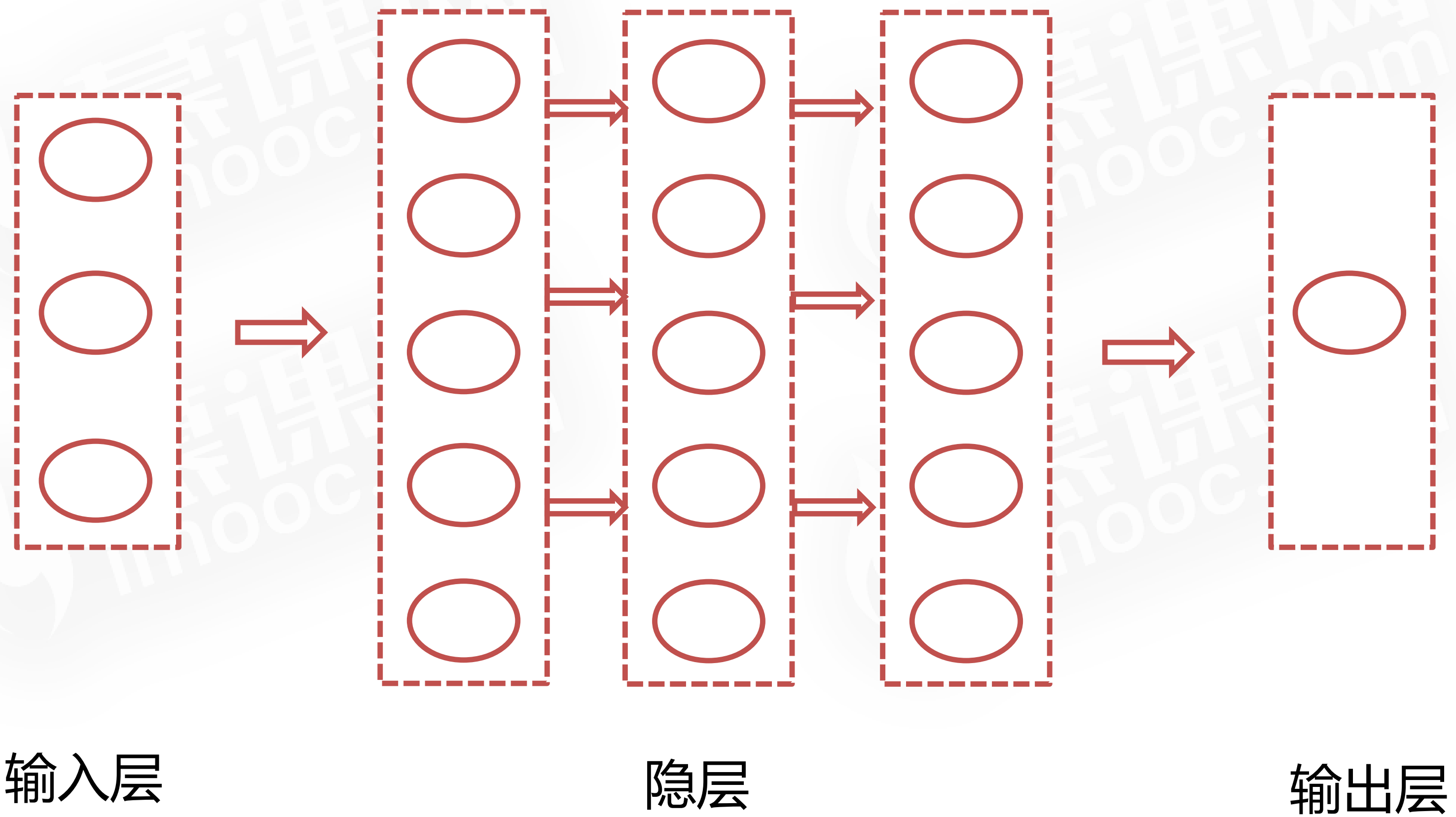


DL DIFFS ML



Personal Recommendation Algorithm

DNN网络结构



DNN模型参数

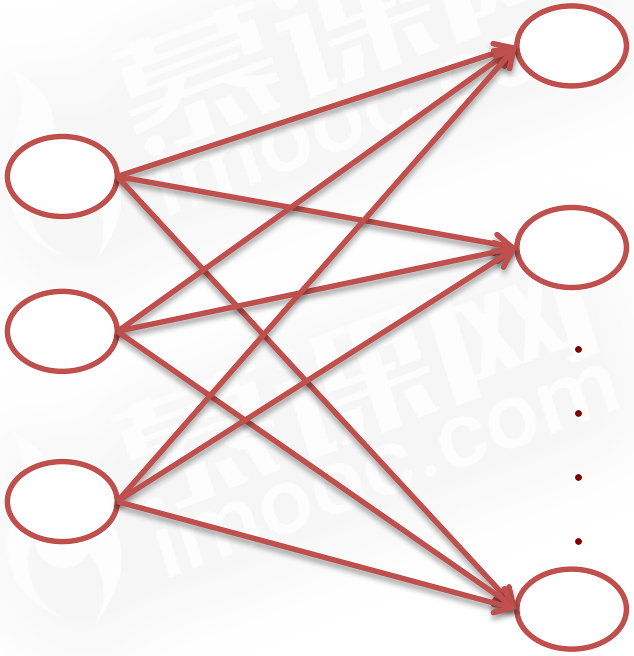
- 隐层层数，每个隐层神经元个数，激活函数
- 输入输出层向量维度
- 不同层之间神经元的连接权重W与偏移值B

前向传播

- 节点的输出值

$$a_j^t = f\left(\sum_k w_{jk}^t a_k^{t-1} + b_j^t\right)$$

$$z_j^t = \sum_k w_{jk}^t a_k^{t-1} + b_j^t$$



反向传播

- Our Target

$$\frac{\partial L}{\partial w^t_{jk}} \qquad \frac{\partial L}{\partial b^t_j}$$

- What We Have

$$\frac{\partial L}{\partial a^T_j} \qquad \frac{\partial L}{\partial z^T_j}$$

反向传播

- 推导

$$\frac{\partial L}{\partial b_j^t} = \frac{\partial L}{\partial z_j^t} * \frac{\partial z_j^t}{\partial b_j^t} = \frac{\partial L}{\partial z_j^t}$$

$$z_j^t = \sum_k w_{jk}^t a_k^{t-1} + b_j^t$$

$$\frac{\partial L}{\partial w_{jk}^t} = \frac{\partial L}{\partial z_j^t} * \frac{\partial z_j^t}{\partial w_{jk}^t} = \frac{\partial L}{\partial z_j^t} * a_k^{t-1}$$

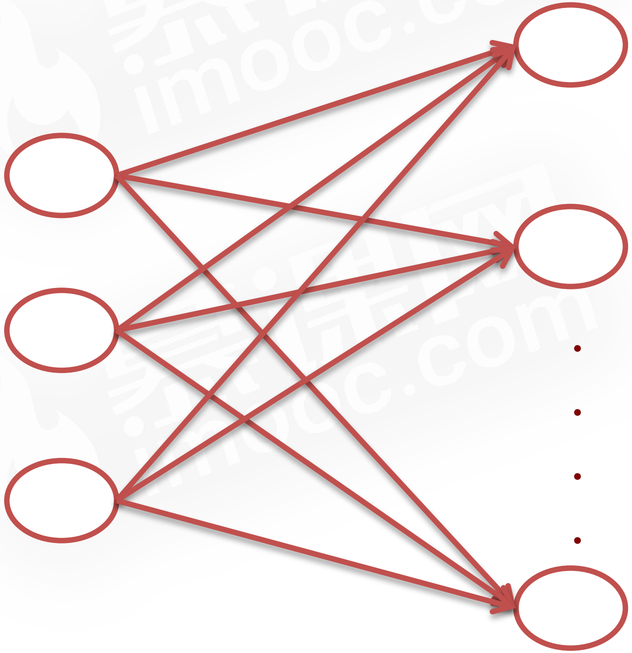
反向传播

- 核心

$$\frac{\partial L}{\partial z_j^{t-1}} = \sum_k \frac{\partial L}{\partial z_k^t} \frac{\partial z_k^t}{\partial z_j^{t-1}}$$

$$z_k^t = \sum_j w_{kj}^t a_j^{t-1} + b_k^t$$

$$\frac{\partial z_k^t}{\partial z_j^{t-1}} = \frac{\partial z_k^t}{\partial a_j^{t-1}} \frac{\partial a_j^{t-1}}{\partial z_j^{t-1}} = w_{kj}^t \frac{\partial a_j^{t-1}}{\partial z_j^{t-1}}$$



反向传播

- 对于输入x，设置合理的输入向量
- 前向传播逐层逐个神经元求解加权和与激活值

反向传播

- 对于输出层求解输出层损失函数对于z值的偏导
- 反向传播逐层求解损失函数对z值的偏导
- 得到w与b的梯度

Personal Recommendation Algorithm

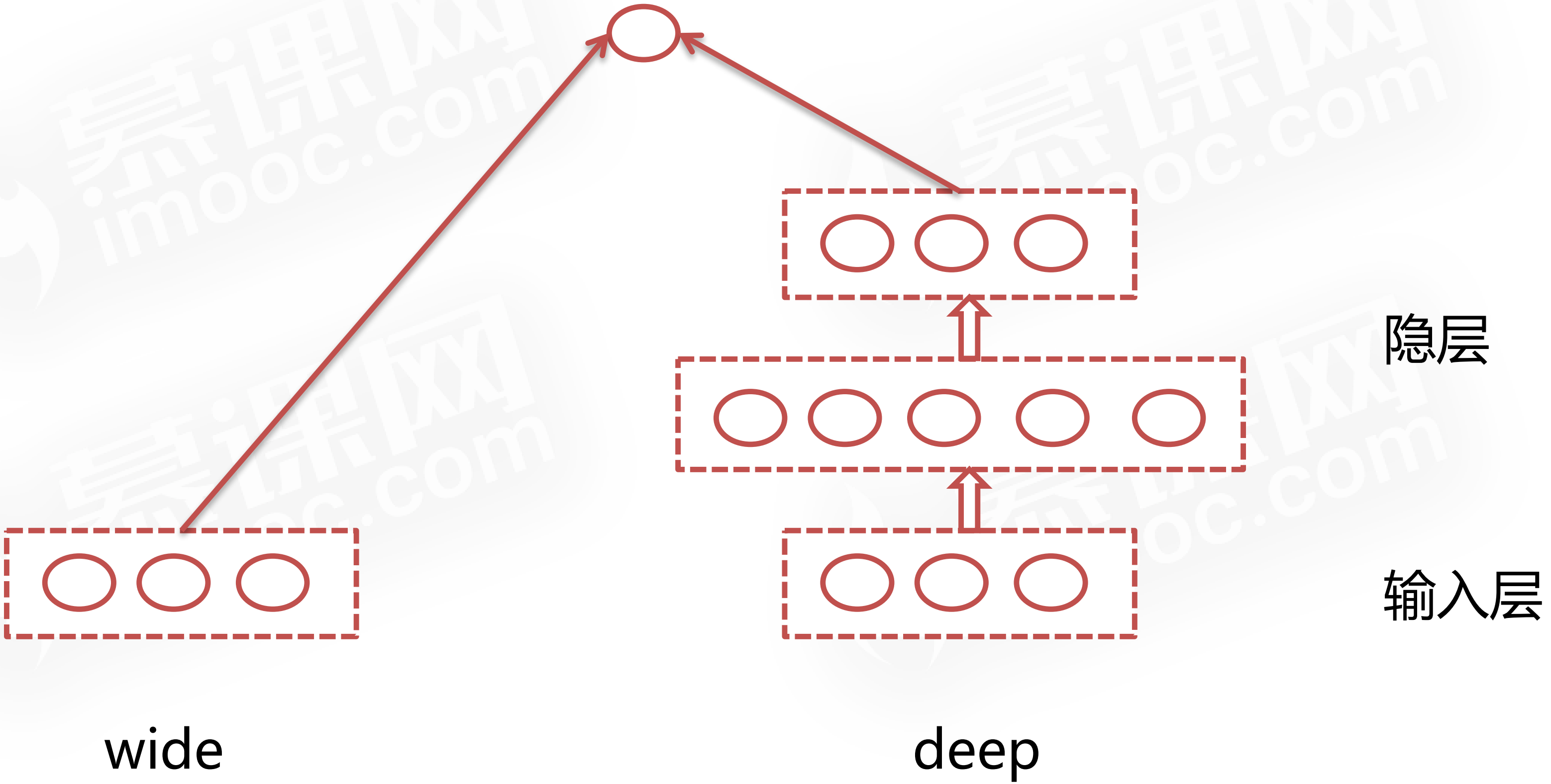
Wide And Deep Model

- w & d的物理意义
- w & d的网络结构
- w & d的数学原理

物理意义

- 论文：wide & deep learning for Recommender Systems
- Generalization and Memorization

网络结构



模型输出

- 模型输出

$$a_{out}^{T} = h\left(w_{wide}, w_{deep}\right) = \sigma\left(w_{wide}\left[x, x_{cross}\right] + w_{deep} a_{out}^{T-1} + b^{T}\right)$$

WD model的反向传播

- wide侧参数学习

$$\frac{\partial L}{\partial w_{widej}} = \frac{\partial L}{\partial a^T} \frac{\partial a^T}{\partial z^T} \frac{\partial z^T}{\partial w_{widej}} = \frac{\partial L}{\partial a^T} \sigma'\left(z^T\right) x_{widej}$$

WD model的反向传播

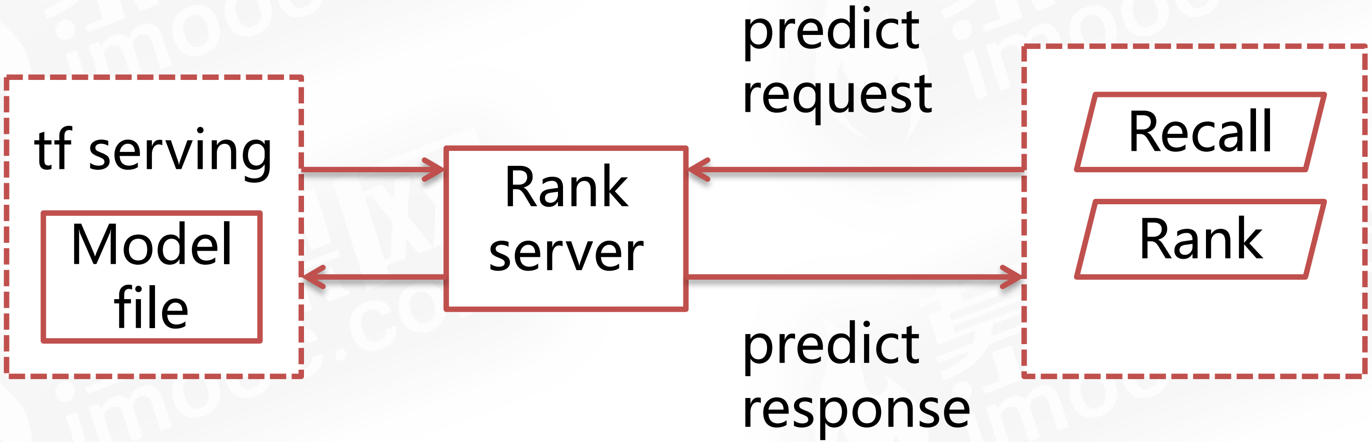
- Deep侧参数学习

$$\frac{\partial L}{\partial z_j^{t-1}} = \sum_k \frac{\partial L}{\partial z_k^t} \frac{\partial z_k^t}{\partial a_j^{t-1}} \frac{\partial a_j^{t-1}}{\partial z_j^{t-1}} = \sum_k \frac{\partial L}{\partial z_k^t} w_{deepkj}^t \frac{\partial a_j^{t-1}}{\partial z_j^{t-1}}$$

$$z_k^t = \sum_j w_{deepkj}^t a_j^{t-1} + b_k^t \rightarrow t \neq T \qquad z_k^t = \left(\sum_j w_{deepkj}^t a_j^{t-1} + b_k^t \right) + w_{wide} * X \rightarrow t = T$$

$$\frac{\partial L}{\partial b_j^t} = \frac{\partial L}{\partial z_j^t} * \frac{\partial z_j^t}{\partial b_j^t} = \frac{\partial L}{\partial z_j^t} \qquad \frac{\partial L}{\partial w_{jk}^t} = \frac{\partial L}{\partial z_j^t} * \frac{\partial z_j^t}{\partial w_{jk}^t} = \frac{\partial L}{\partial z_j^t} * a_k^{t-1}$$

server架构



tf serving
Model file
Rank server
predict request
predict response
Recall
Rank