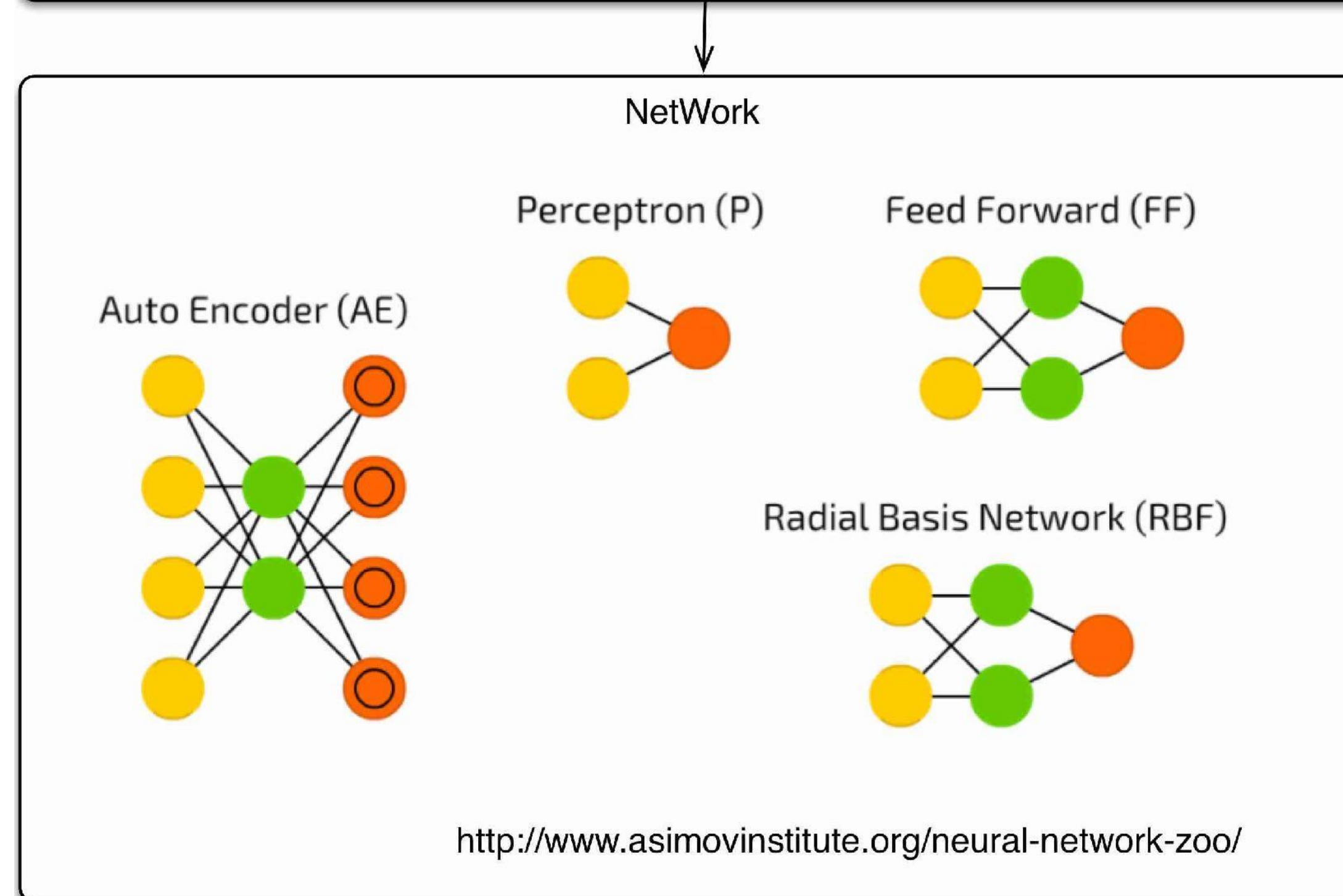
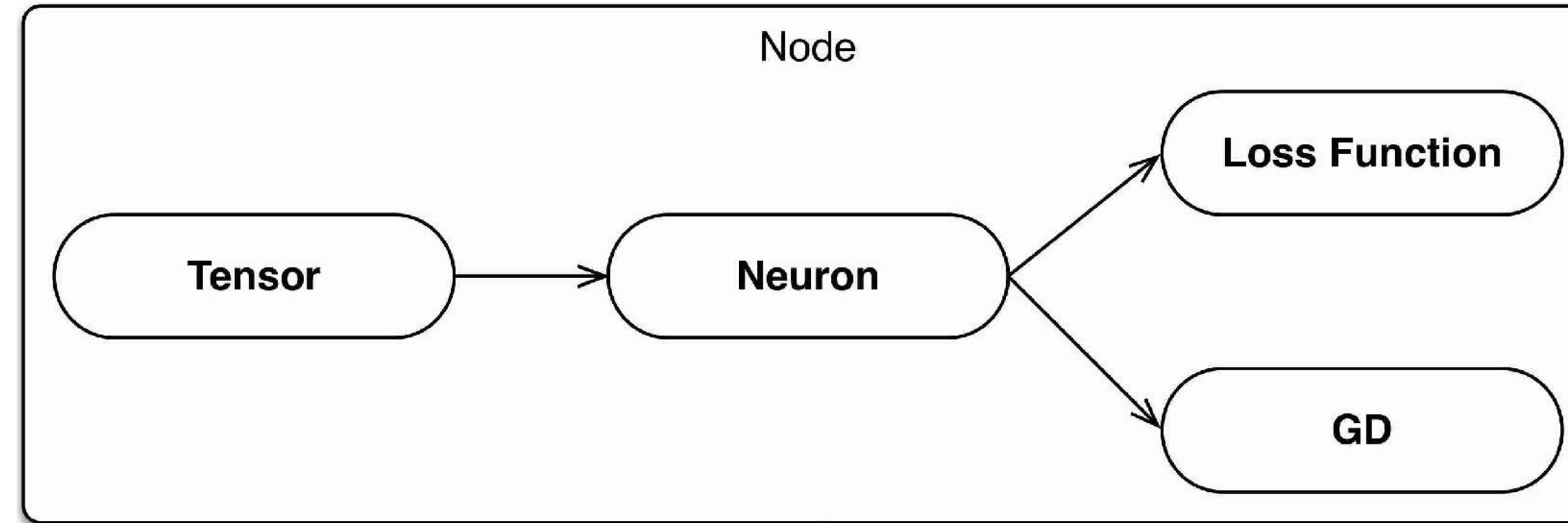


Tensorflow Basic

学习路线

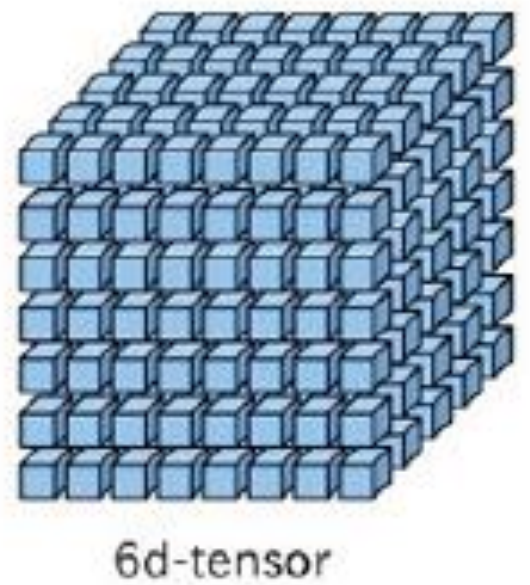
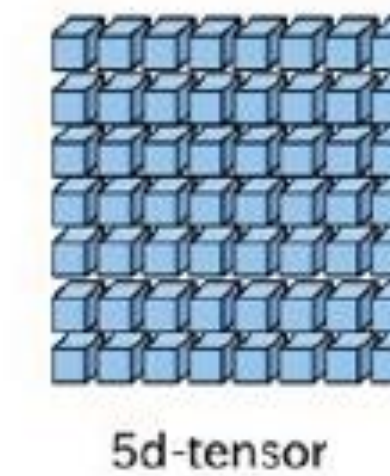
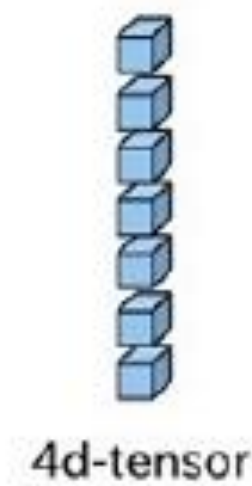
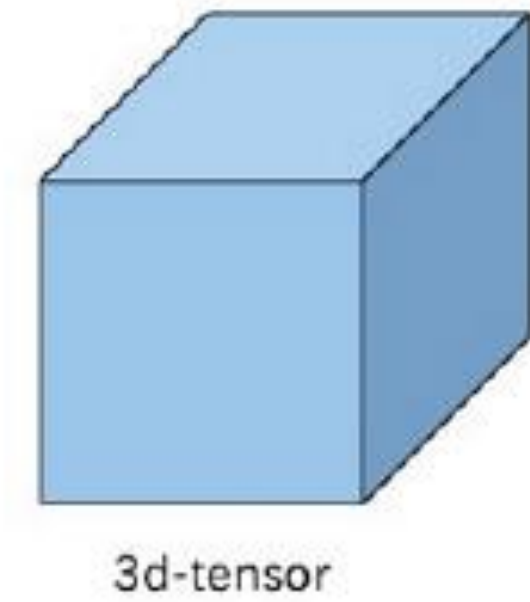
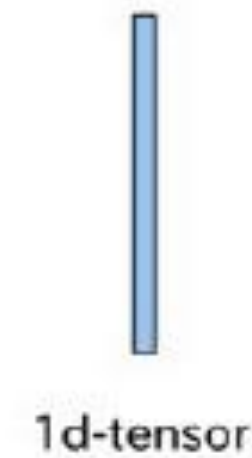


TF预备知识

- 即时执行(Eager Execution), 计算图(Graph), 会话(Session)
- Tensor
 - Constant, Placeholder, Variable 三种表现
 - reshape, rank

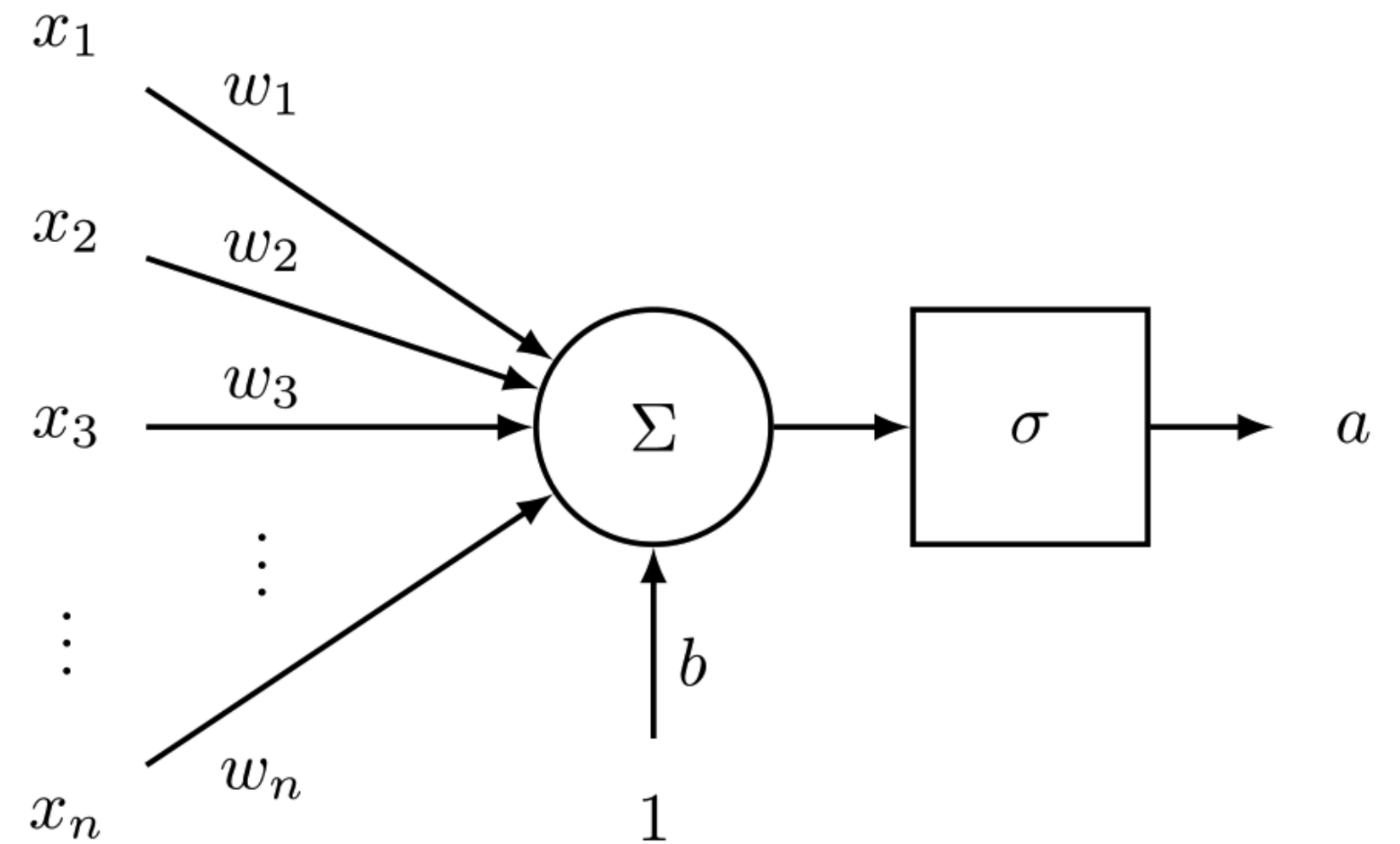
Tensor

- `basic_operations.py`
 - 用Constant实现简单计算
- Matrix操作



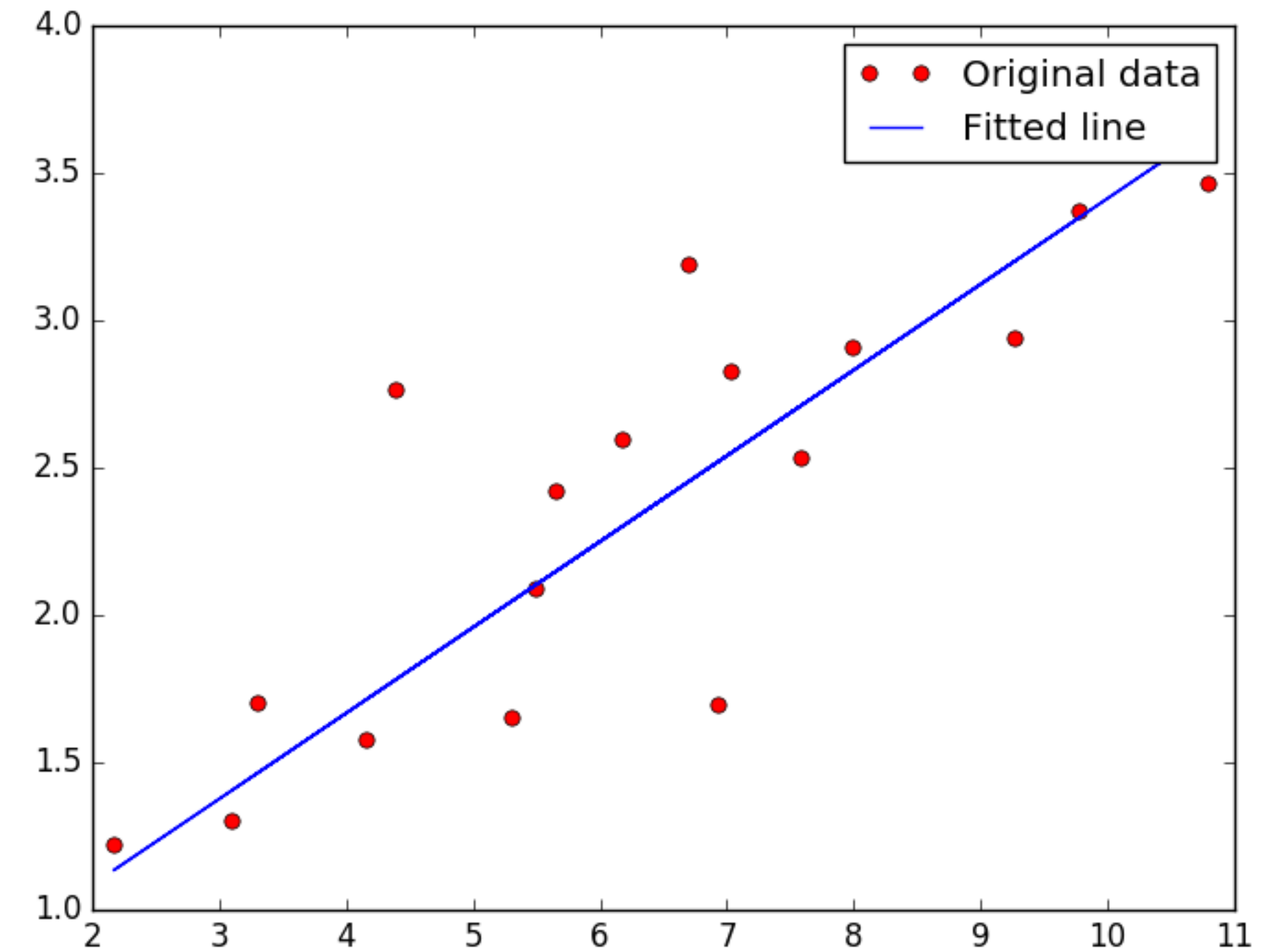
人工神经元Neuron

- 人工神经元被建模为一个函数 $\sigma(Wx+b)$ ，其中 w 是权重， x 是输入， b 是偏移量
- 输入线性加权叠加
- 一个非线性函数 σ 作用，进行输出， σ 称为激活函数
- neuron.py
- 用Variables, Placeholder实现一个简单的线性人工神经元



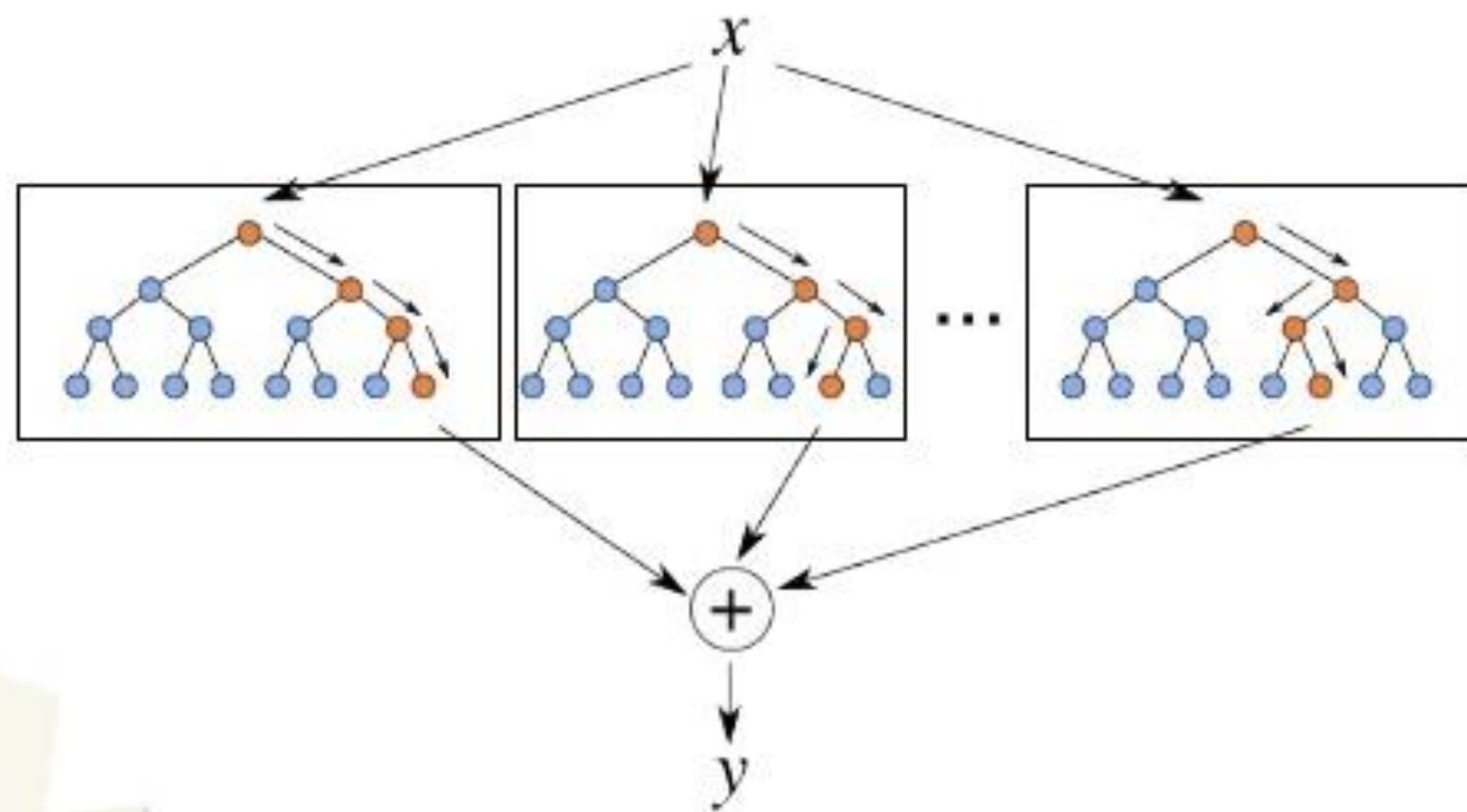
Loss & GD

- `linear_regression.py`
 - 定义距离Loss
 - GD迭代优化
- `logistic_regression.py`
 - 多一个sigmoid激活函数



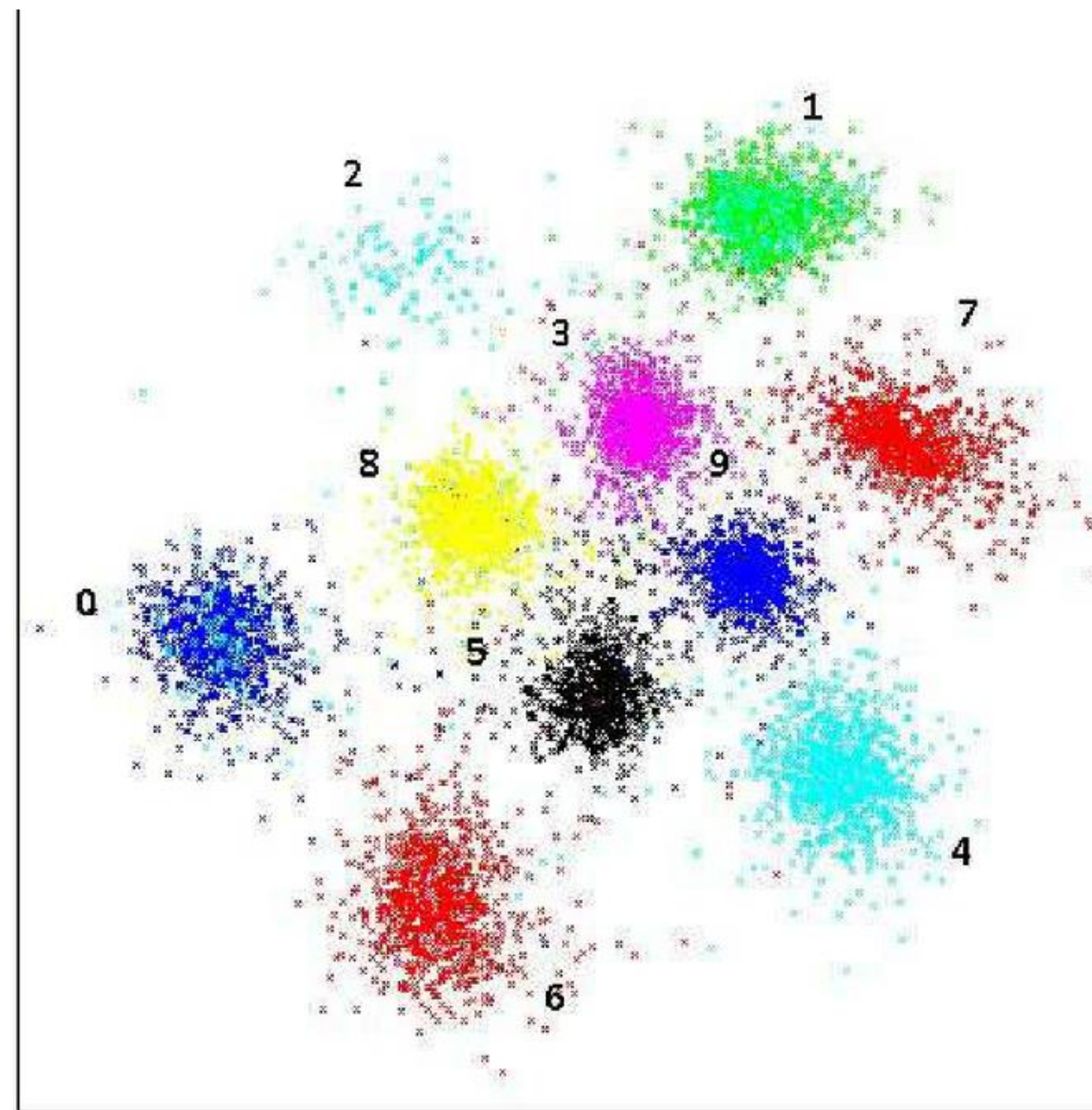
Mnist传统解法

Random Forest



随机森林

random_forest.py



最近邻方法

DNN思考

- 打开<https://playground.tensorflow.org/>
- 思考如下几个问题：
 - 调节学习率，看收敛速度有什么差别
 - 1-2个神经元分别能处理什么样的分类问题
 - 最少需要多少个神经元可解XOR问题
 - 做出螺旋线问题, 比比训练速度和神经元数 ($\text{loss} < 0.05$)

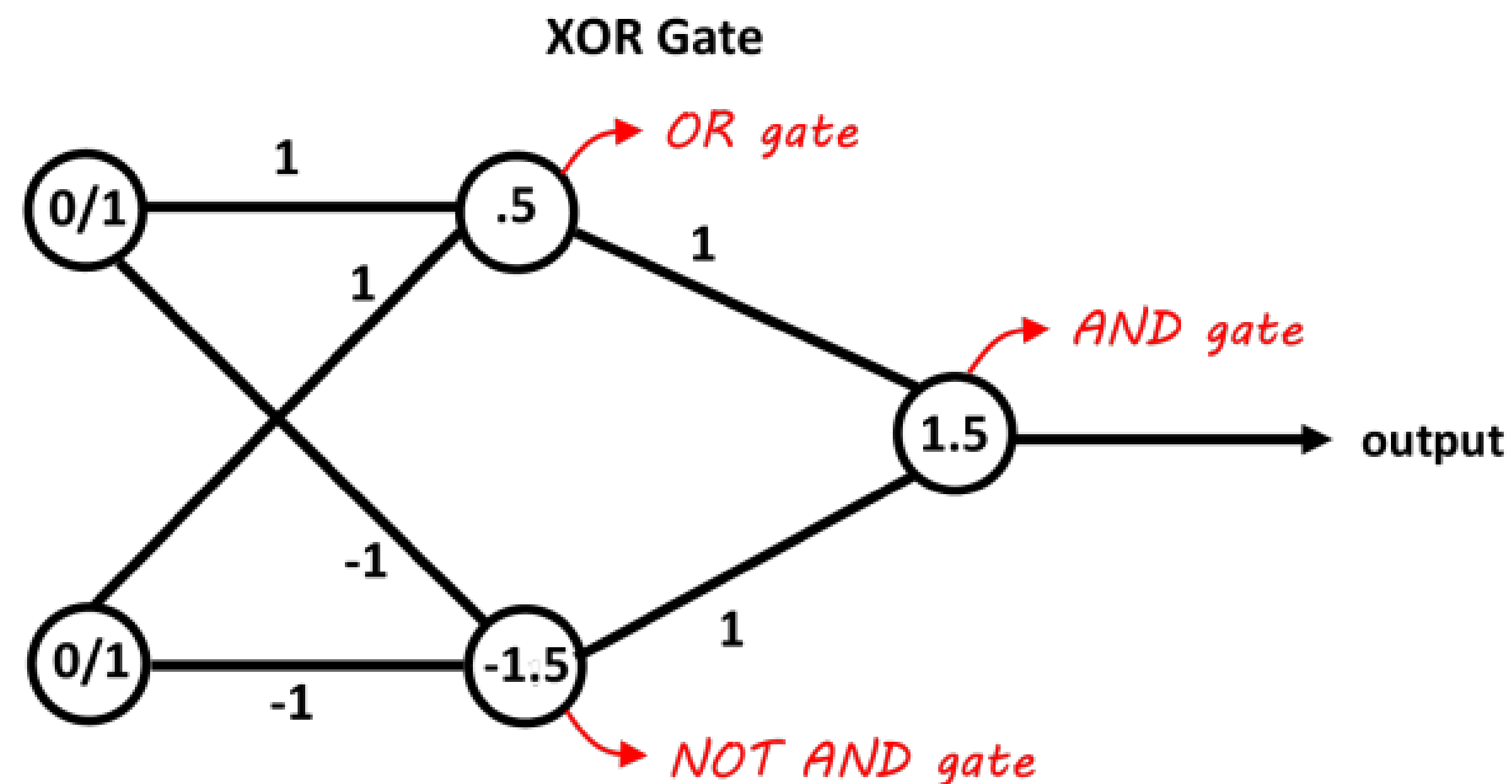
收敛讨论

- 学习率过大会引起震荡，太小学习速率太慢
- NN对初始值的选择比较敏感
- 测试集和训练集的Loss差异

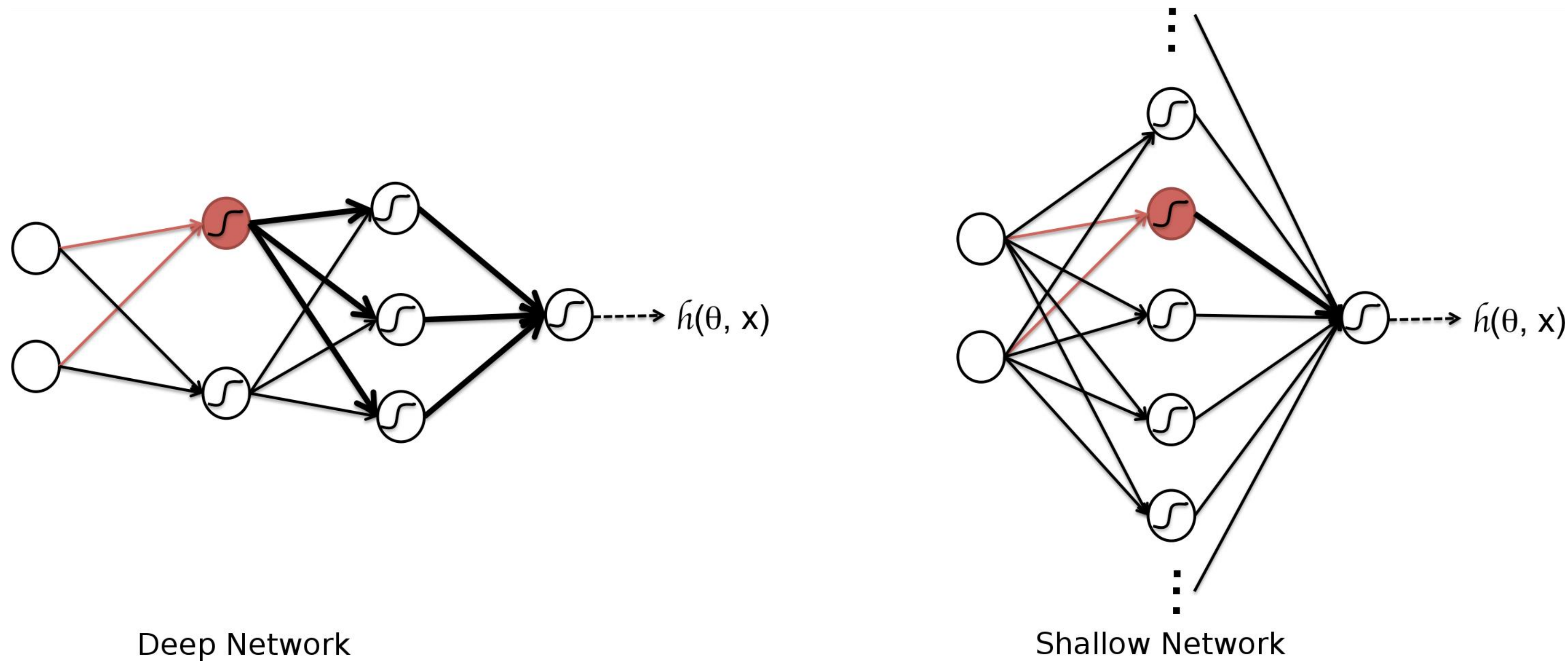
XOR问题

- 不是个线性问题
- 2个ReLU神经元可解

	0	1
0	0	-0.5
1	-0.5	0



宽度 vs 深度



when a function can be **compactly** represented by a deep architecture, it might need a very large architecture to be represented by an insufficiently deep one.

Reference

- Q. V. Le. A Tutorial on Deep Learning Lecture Notes, 2015.
- Y. Bengio. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1):1 – 127, 2009.
- L. Ba and R. Caurana. Do deep nets really need to be deep? arXiv preprint arXiv:1312.6184, 2013.
- Y. LeCun, L. Bottou, G. Orr, and K. R. Muller. Efficient backprop. In Neural networks: Tricks of the trade. Springer, 1998.

谢谢大家！