

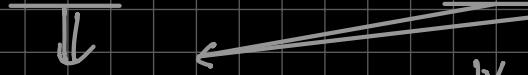
Deep Learning :

来源：

DL的前身为 Neural Network，为建立模拟人脑进行学习的神经网络。e.g.: 通用的深度学习结构：含多隐层的多层次感知器

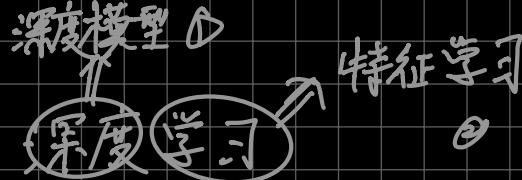
应用领域：自然语言处理、机器人、自动驾驶等新兴、高新技术领域

应用方面：特征提取。二）深度学习用大数据来学习特征，学习到的特征更本质地刻画数据丰富的内在信息。



特征：

与传统机器学习不同点：



① 模型 结构深度较大 (Deep) \Rightarrow ≥5 层隐层结点

② 更强调特征的学习 \Rightarrow 通过逐层特征变换，将样本于原空间特征变换至一个新特征空间。

③ 学习特征的方式为通过大数据自主学习·提取数据特征。而传统机器学习为人工特征选择。

与浅层结构算法不同点：

分类、回归等机器学习算法。

对复杂函数的表现能力强。传统的浅层结构算法由于有限样本和计算单元，结构简单，对复杂分类问题其泛化能力受限，而深度学习通过深层非线性网络结构，实现对复杂函数的逼近，从而使任何函数（几乎）均可被表现出来。

与传统神经网络不同点：

① 通过无监督学习实现逐层初始化，从而降低训练中难度。而传统神经网络为直接随机地进行初始化，设定初值。

② 训练网络方式不同。传统神经网络通过反向传播法训练网络，而深度学习为一个层层传导的训练机制。

原因：在深层网络，若采用反向传播法，会出现梯度扩散问题。

训练过程：

A：自底向上的无监督学习 \Rightarrow 无监督学习无标注数据
分层训练各层参数（将 k 层输出作为 $k+1$ 层输入）

B：自顶向下的监督学习 \Rightarrow 监督学习，有标签数据，通过误差向下传递修改参数。

Deep Learning Neural Network

名词解释:

Neural Networks	LSTM: Long short Term Model	长短期神经网络
	RNN: Recurrent Neural Network	递归神经网络
	CNN: Convolution Neural Network	卷积神经网络
	DNN: Deep Neural Network	深度神经网络

应用领域:

RNN \Rightarrow NLP (自然语言处理)

Reason: 相比普通神经网络, RNN能更好处理 序列数据, 因此能较好地应用于NLP领域.
如上下文语境中单词、语音

CNN \Rightarrow CV (计算机视觉)

Reason: 相比普通神经网络, CNN通过卷积、池化操作自动学习图像在各个层次上的特征, 符合人类大脑理解图像时层级处理的过程。

↓
1. 颜色、亮度 \rightarrow 2. 局部细节特征 (边缘、角点、直线) \rightarrow 3. 纹理、几何形状等
 \Rightarrow 形成整个物体的概念.

WideDeep Model:

Description: WideDeep model is a typical multi-input single output model.

Advantages: 兼备浅层模型记忆能力与深层模型泛化能力.

Activation Function (激活函数)

Linear \Rightarrow 此时并无增加非线性因素，因此此时 Neural Network 退化为 perceptron，只能处理线性问题。

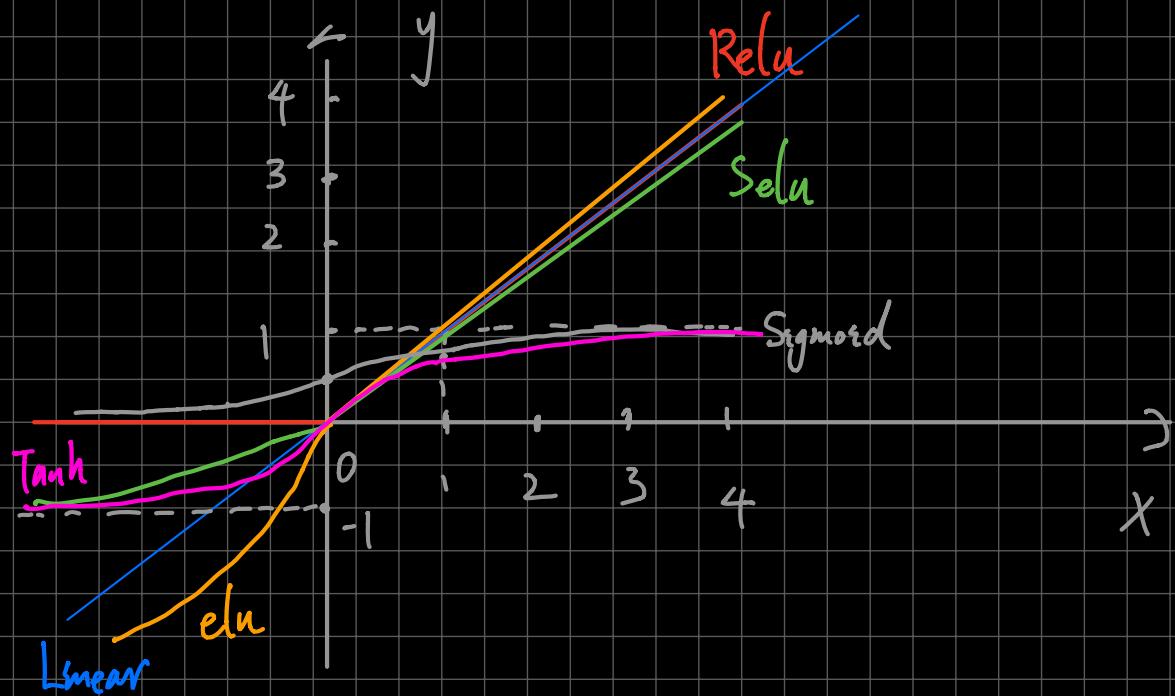
Sigmoid \Rightarrow 一般用于二分类问题

Softmax \Rightarrow 一般用于多分类问题

Relu \Rightarrow 最为流行，效果非常好，但只用于 Hidden layer

selu·elu \Rightarrow 解决 Relu 出现的神经元死亡问题。

Tanh \Rightarrow 绝大多数情况下避免使用，易出现梯度消失，神经元死亡



Loss Function (损失函数):

适用场景

Binary Cross entropy \Rightarrow 二分类情景，计算预测与真实值间
(二元交叉熵) 交叉熵损失

Sparse Categorical Crossentropy \Rightarrow 多分类情景。

(多分类交叉熵 \Rightarrow 此时 y -true 是 single value)

Categorical Crossentropy \Rightarrow 此时 y -true 是 one-hot 编码

MSE (Mean Square Error) \Rightarrow 回归情景

MSE \Rightarrow Mean Square Error



RMSE \Rightarrow Root Mean Square Error

MAE \Rightarrow Mean Absolute Error

} 用于回归问题的评价。

AUC \Rightarrow 用于二分类问题

Recall Rate (覆盖率 || 召回率)

Precision Rate (精确率)

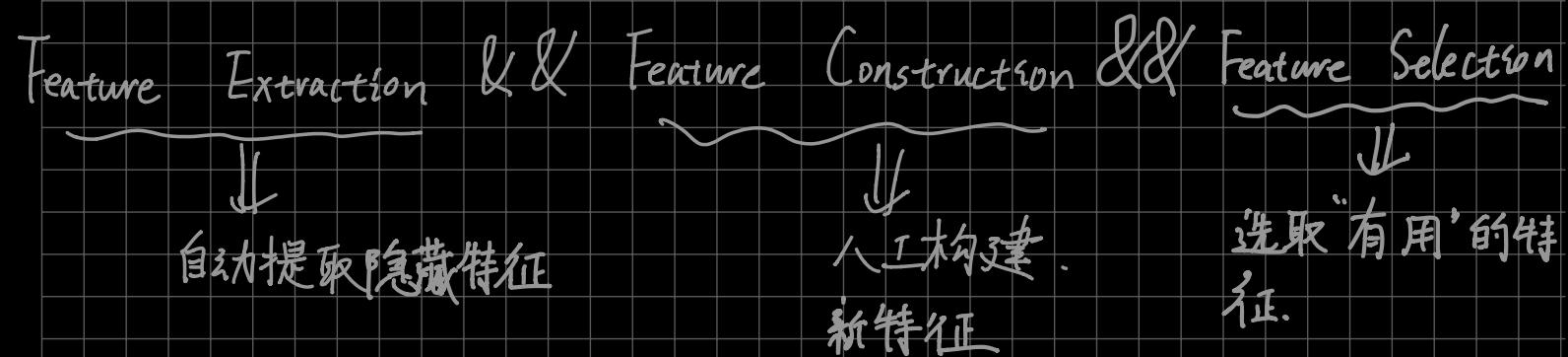
} 用于分类问题的评价。

几种典型优化器optimizer

- SGD随机梯度下降: keras.optimizers.SGD(lr=0.01, momentum=0.0, decay=0.0, nesterov=False)
 - 可以支持动量参数momentum, 支持Nesterov动量 (给momentum增加了校正因子)
- Adagrad: keras.optimizers.Adagrad(lr=0.01, epsilon=None, decay=0.0)
 - 具有特定参数学习率的优化器, 根据参数在训练期间的更新频率进行自适应调整。
 - 应用于凸问题时快速收敛, 但在非凸函数时可能是局部最优
- RMSProp: keras.optimizers.RMSprop(lr=0.001, rho=0.9, epsilon=None, decay=0.0)
 - 修正Adagrad在非凸情况下的问题
 - 已证明是有效实用的优化算法, 被经常的大量使用
- Adam: keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999)
 - 派生自短语adaptive moments
 - 早起被看做对RMSProp加上一些重要动量的变种, 但其实和RMSProp难分伯仲

选择:

- 简单问题用SGD
- 复杂问题用RMSProp和Adam均可
- 使用RMSProp一般都没错



Feature Extraction \Rightarrow 特征提取

