

## 第三部分 深度学习 / Deep Learning

翻译&amp;校正 | 韩信子@ShowMeAI

编辑 | 南乔@ShowMeAI

原文作者 | <https://stanford.edu/~shervine>

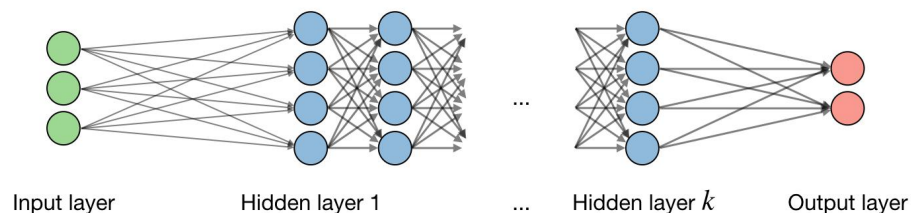
本节原文超链

### [1]神经网络 / Neural Networks

神经网络是一类按层结构搭建的模型，常用的包括卷积神经网络和递归神经网络。

#### 架构 Architecture

描述神经网络架构的常用词汇见下图：



备注：Input layer[输入层] → Hidden layer[隐藏层] → Output layer[输出层]

已知  $i$  是网络的第  $i$  层， $j$  是网络的第  $j$  层，有： $z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$

其中， $w$  表示权重， $b$  表示偏差， $z$  表示输出。

#### 激活函数 Activation function

激活函数被用在隐含单元后，向模型引入非线性复杂度。下表总结了常见激活函数：

Sigmoid	双曲正切函数(Tanh)	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z) \epsilon \ll 1$

#### 交叉熵损失 Cross-entropy loss

神经网络中，交叉熵损失  $L(z, y)$  通常定义如下：

$$L(z, y) = -[y \log(z) + (1 - y) \log(1 - z)]$$

#### 学习率 Learning rate

学习率（通常记为  $\alpha$  或  $\eta$ ），表示权重的更新速度，可以是固定值，也可以自适应改变。现阶段最常用的方法 Adam，就是自适应学习率。

#### 反向传播 Backpropagation

反向传播的过程，是根据网络实际输出和标准答案计算损失函数，进而更新神经网络中的权重。权重  $w$  的导数由链式法则计算而来：

$$\frac{\partial L(z, y)}{\partial w} = \frac{\partial L(z, y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

权重  $w$  的更新规则如下：

$$w \leftarrow w - \alpha \frac{\partial L(z, y)}{\partial w}$$

#### 更新权重 Updating weights

在神经网络中，权重按照下方步骤进行更新：

第一步	第二步	第三步	第四步
取一个批次(batch)的训练数据	通过前向传播来得到相关损失	通过反向传播损失来得到梯度	利用梯度更新网络的权重

#### 随机失活 Dropout

随机失活是一种通过丢弃神经网络单元来防止训练数据过拟合的技术。实际应用中，神经元以概率  $p$  被丢弃/置为 0（或以概率  $1 - p$  被保留）。

## [2]卷积神经网络 / Convolutional Neural Networks

### 卷积神经网络要求 Convolutional layer requirement

记  $W$  为输入图像尺寸,  $F$  为卷积层神经元尺寸,  $P$  为零填充的大小, 那么给定的输入图像能够容纳的神经元数目  $N$  为:

$$N = \frac{W - F + 2P}{S} + 1$$

### 批量规范化 Batch normalization

它是超参数  $\gamma, \beta$  标准化样本批  $\{x_i\}$  的一个步骤。记  $\mu_B$  为当前批样本的均值,  $\sigma_B^2$  为当前批样本的方差, 则有:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

备注: 通常应用于全连接层/卷积层之后、在非线性层之前, 目的是允许更高的学习率、减少对初始化的强依赖。

## [3]递归神经网络 / Recurrent Neural Networks, RNN

### 门控的种类 Types of gates

在一个典型的递归神经网络中, 有多种门控结构:

输入门	忘记门	门控	输出门
信息要不要写入单元?	要不要清空单元信息?	在单元写入多少信息?	从单元输出多少信息?

### 长短期记忆网络 LSTM

长短期记忆网络 (long short-term memory, LSTM) 是 RNN 的一种, 它可以通过增加“忘记门”来避免梯度消失问题。

## [4]强化学习和控制 / Reinforcement Learning and Control

强化学习的目标, 是让代理去学习如何在环境中进化。

### 马尔可夫决策过程 Markov decision processes

一个马尔可夫决策过程 (Markov decision processes, MDP) 是一个 5 维元组  $(\mathcal{S}, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ 。

- 1)  $\mathcal{S}$  是状态的集合
- 2)  $\mathcal{A}$  是动作的集合
- 3)  $\{P_{sa}\}$  是对于  $s \in \mathcal{S}$  和  $a \in \mathcal{A}$  的状态转换概率
- 4)  $\gamma \in [0, 1]$  是折扣系数
- 5)  $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  or  $R: \mathcal{S} \rightarrow \mathbb{R}$  是算法希望最大化的回报函数

### 策略 Policy

策略  $\pi$  函数, 将状态  $\mathcal{S}$  映射到动作  $\mathcal{A}$ :

$$\pi: \mathcal{S} \rightarrow \mathcal{A}$$

备注: 对于某指定状态  $s$ , 完成了动作  $a = \pi(s)$ , 被认为执行了一个指定的策略  $\pi$ 。

### 价值函数 Value function

对于一个指定的策略  $\pi$  和指定的状态  $s$ , 价值函数  $v^\pi$  定义如下:

$$V^\pi(s) = E[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi]$$

### 贝尔曼方程 Bellman equation

最优贝尔曼方程描述了最优策略  $\pi^*$  的价值函数  $V^{\pi^*}$ :

$$V^{\pi^*}(s) = R(s) + \max_{a \in \mathcal{A}} \gamma \sum_{s' \in \mathcal{S}} P_{sa}(s') V^{\pi^*}(s')$$

备注: 对于某特定状态  $s$ , 其最优策略  $\pi^*$  是:

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P_{sa}(s') V^*(s')$$

## 值迭代算法 Value iteration algorithm

值迭代算法分为两步：

1) 首先初始化值：

$$V_0(s) = 0$$

2) 通过之前的值进行迭代：

$$V_{i+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left[ \sum_{s' \in \mathcal{S}} \gamma P_{sa}(s') V_i(s') \right]$$

## 极大似然估计 Maximum likelihood estimate

状态转移概率的极大似然估计，定义如下：

$$P_{sa}(s') = \frac{\text{#状态 } s \text{ 下进行动作 } a \text{ 并且进入状态 } s' \text{ 的次数}}{\text{#状态 } s \text{ 下进行动作 } a \text{ 的次数}}$$

## Q-学习 Q-learning

Q 学习是一种 Q 的无模型 (model-free) 估计，定义如下：

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[ R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

## Awesome AI Courses Notes Cheat Sheets

<b>Machine Learning CS229</b>	<b>Deep Learning CS230</b>	Natural Language Processing CS224n	Computer Vision CS231n	Deep Reinforcement Learning CS285	Neural Networks for NLP CS11-747	DL for Self-Driving Cars 6.S094	...
Stanford	Stanford	Stanford	Stanford	UC Berkeley	CMU	MIT	...

是 **ShowMeAI** 资料库的分支系列，覆盖最具知名度的 TOP20+ 门 AI 课程，旨在为读者和学习者提供一整套高品质中文速查表，可以点击【[这里](#)】查看。

斯坦福大学（Stanford University）的 **Machine Learning（CS229）** 和 **Deep Learning（CS230）** 课程，是本系列的第一批产出。

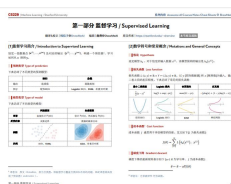
本批两门课程的速查表由斯坦福大学计算机专业学生 **Shervine Amidi** 总结整理。原速查表为英文，可点击【[这里](#)】查看，**ShowMeAI** 对内容进行了翻译、校对与编辑排版，整理为当前的中文版本。

有任何建议和反馈，也欢迎通过下方渠道和我们联络 (\*^\_^\*)

## CS229 | Machine Learning @ Stanford University

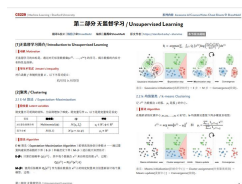
### 监督学习

Supervised Learning


[中文速查表链接](#)

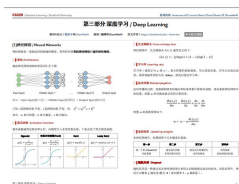
### 无监督学习

Unsupervised Learning


[中文速查表链接](#)

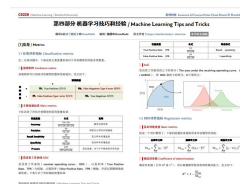
### 深度学习

Deep Learning


[中文速查表链接](#)

### 机器学习技巧和经验

Tips and Tricks


[中文速查表链接](#)

## CS230 | Deep Learning @ Stanford University

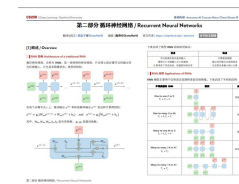
### 卷积神经网络

CNN


[中文速查表链接](#)

### 循环神经网络

RNN


[中文速查表链接](#)

### 深度学习技巧与建议

Tips and Tricks


[中文速查表链接](#)

### 概率统计

Probabilities / Statistics


[中文速查表链接](#)

### 线性代数与微积分

Linear Algebra and Calculus


[中文速查表链接](#)

GitHub  
ShowMeAI

<https://github.com/ShowMeAI-Hub/>



ShowMeAI 研究中心

扫码回复“**速查表**”  
下载**最新**全套资料