

第三部分 深度学习技巧与建议 / Deep Learning Tips and Tricks

翻译&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] 数据预处理 / Data Processing

数据增强

深度学习模型通常需要大量数据才能正确训练。使用数据增强技术从现有数据中获取更多数据通常很有用。

主要内容总结在下表中。更具体地说，对于输入图像数据，我们可以应用以下技术：

原始图片 Original	翻转 Flip	旋转 Rotation	随机截取 Random crop
			
原始图片	翻转 (保留图像含义)	小角度旋转	随机聚焦图像的一个部分 可连续进行多次随机裁剪

颜色调整 Color shift	添加噪声 Noise addition	信息损失 Information loss	对比度变化 Contrast change
			
RGB 的细微差别略有变化 捕捉曝光时可能出现噪点	增加噪声 对输入质量变化的容忍度更高	部分图像被(遮挡)忽略 模拟图像部分的潜在损失	亮度变化 控制因一天中的时间而导致的曝光差异

备注：数据通常在训练期间即时增强。

批标准化 Batch normalization

这个步骤通过超参数 γ, β 对批次 $\{x_i\}$ 进行归一化。我们把想要校正的批次的均值和方差记作 μ_B, σ_B^2 ，那归一化过程公式：

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

BN 通常在全连接/卷积层之后和非线性层之前完成，旨在允许更高的学习率并减少对初始化的依赖。

[2] 训练神经网络 / Training a Neural Network

2.1 定义 Definitions

轮次 Epoch

在训练模型的上下文中，epoch 是一个术语，用于指代模型看到整个训练集一次，并进行权重更新迭代。

Mini-batch gradient descent

在训练阶段，由于计算复杂性或由于噪声问题，更新权重通常不是基于整个训练集一次。相反，更新步骤是在小批量上完成的，其中 batch 批量中的数据量是我们调整的超参数。

损失函数 Loss function

损失函数 L 可以量化模型的表现，评估模型输出结果 z 和真实标签 y 的差异程度。

交叉熵损失函数 Cross-entropy loss

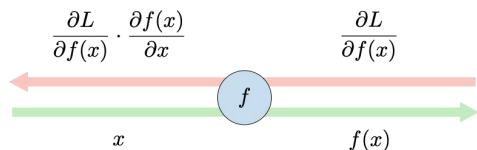
在神经网络二元分类问题中，最常用的损失函数是交叉熵损失 $L(z, y)$ ：

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

2.2 寻找最优权重 Finding Optimal Weights

反向传播 Backpropagation

反向传播是一种通过考虑实际输出和期望输出来更新神经网络中权重的方法。使用链式法则计算每个权重 w 的导数。



反向传播中，每个权重都使用以下规则更新：

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

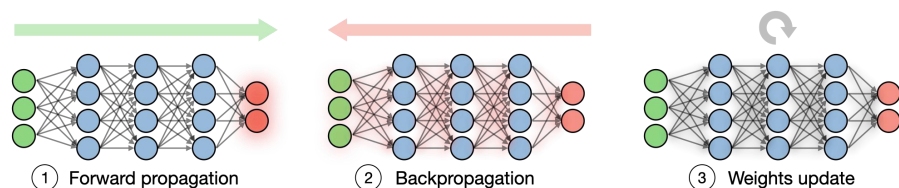
更新权重 Updating weights

在神经网络中，通过以下步骤更新权重：

步骤一：取出一个批次 (batch) 的资料，执行前向传播算法 (forward propagation) 来得到对应的损失值。

步骤二：将损失值通过反向传播算法来得到梯度。

步骤三：使用梯度来更新网络的权重。



[3] 参数调优 / Parameter Tuning

3.1 权重初始化 Weights Initialization

哈维尔初始化 Xavier initialization

Xavier 初始化不是以纯粹随机方式初始化权重，而是使初始权重考虑到网络结构特征。

迁移学习 Transfer learning

训练深度学习模型需要大量数据，更重要的是需要大量时间。在需要数天/数周训练的大型数据集上训练得到的预训练权重通常很有用，可以将其迁移应用于我们的场景。

根据我们手头的数据量多少，有不同的迁移学习方法：

训练数据量	图例	解释
小数据量		冻结所有其他层，在 softmax 上训练权重
中等数据量		冻结大多数层，在最后一层和 softmax 上训练权重
大数据量		通过预训练模型初始化权重，训练所有的层次（包括 softmax ）

3.2 优化与收敛细节 Optimizing Convergence

学习率 Learning rate

学习率，通常记为 α ，有时记作 η ，表示权重更新的速度。它可以是固定的或自适应地改变。目前最流行的方法叫做 **Adam**，是一种自适应学习率的方法。

自适应学习率 Adaptive learning rates

在训练模型时让学习率变化可以减少训练时间并提高数值最优解。**Adam** 优化器是最常用的技术，但其他技术也很有用。

下表是对各种典型方法的一个总结：

Method	Explanation	Update of w	Update of b
Momentum	抑制振荡，改进 SGD 2 个要调整的参数	$w - \alpha v_{dw}$	$b - \alpha v_{db}$

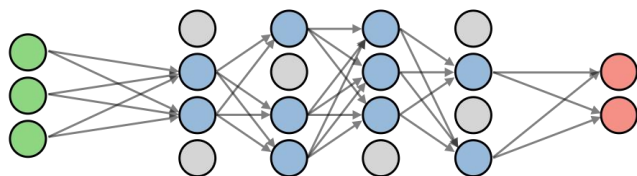
Method	Explanation	Update of w	Update of b
RMSprop	均方根传播 通过控制振荡加速学习算法	$w \leftarrow w - \alpha \frac{dw}{\sqrt{s_{dw}}}$	$b \leftarrow b - \alpha \frac{db}{\sqrt{s_{db}}}$
Adam	自适应矩估计, 最流行的方法 4 个要调整的参数	$w \leftarrow w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \epsilon}$	$b \leftarrow b - \alpha \frac{v_{db}}{\sqrt{s_{db}} + \epsilon}$

备注: 其他方法包括 Adadelta、Adagrad 和 SGD。

[4] 正则化 / Regularization

随机失活 Dropout

Dropout 是神经网络中使用的一种技术, 通过以概率 $p > 0$ 丢弃神经元来防止训练数据过度拟合。它迫使模型避免过分依赖特定的特征集。



备注: 大多数深度学习框架通过“keep”参数来定义 dropout 程度, 设定 keep 为 p 代表 dropout 为 $1 - p$ 。

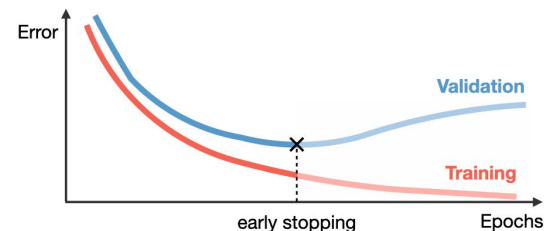
权重正则化 Weight regularization

为确保权重不会太大且模型不会在训练集上过拟合, 通常对模型权重应用正则化操作。

LASSO	Ridge	Elastic Net
将系数缩小到 0, 适合变量选择	获得更小的权重	变量选择和小权重系数之间的一个权衡
$\dots + \lambda \ \theta\ _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda \ \theta\ _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda [(1 - \alpha) \ \theta\ _1 + \alpha \ \theta\ _2^2]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

早停止 Early stopping

早停止: 一旦验证集上的损失不再降低或开始增加, 就会停止训练过程。



[5] 实践技巧 / Good Practices

小批量数据完全拟合测试 Overfitting small batch

在调试模型时, 通常会快速测试以确保模型本身的结构没有重大问题。

具体说来, 为了确保模型可以被正确训练, 在网络内部传递了一个 mini-batch, 看看它是否会过拟合(完全拟合)。

如果不能, 则意味着模型要么太复杂, 要么不够复杂, 以至于在小批量上都不能完全拟合, 更不用说正常大小的训练集了。

梯度检查 Gradient checking

梯度检查是在实现神经网络的反向传递过程中使用的一种方法。具体的做法是, 将分析梯度的值与给定点的数值梯度进行比较, 检查其正确性。

类型	数值梯度	解析梯度
公式	$\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$	$\frac{df}{dx}(x) = f'(x)$
注解	计算代价高: 每个维度必须计算两次损失 用于验证分析实现的正确性 权衡选择 h 既不能太小 (数值不稳定性) 也不能太大 (计算精度差)	结果“精确” 解析公式直接计算 最终实现中会使用

$$\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

Awesome AI Courses Notes Cheat Sheets

Machine Learning CS229	Deep Learning CS230	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。

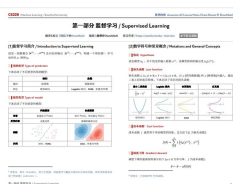
本批两门课程的速查表由斯坦福大学计算机专业学生 **Shervine Amidi** 总结整理。原速查表为英文，可点击【[这里](#)】查看，**ShowMeAI** 对内容进行了翻译、校对与编辑排版，整理为当前的中文版本。

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

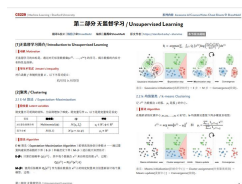
监督学习

Supervised Learning


[中文速查表链接](#)

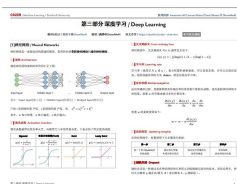
无监督学习

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

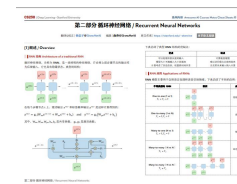
卷积神经网络

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