Deep Learning Check Points*

赵云龙

2018 年 6 月 13 日

目录

1 综述

《Deep Learning》这门课程从 2018 年 03 月始到 2018 年 05 月末结课，历时 13 周半。执教老师是邓世文老师，邓老师在大三上学期就开始教授我们《Deep Learning》的先导课《模式识别》了。对于邓老师我们并不陌生，我个人倒是觉得邓老师挺酷的。

废话不多说，经过 13 周的系统学习，我们也应该具备基本的深度学习理论和一定的实践能力（虽然这并不在要求范围内）。首先让我们先回顾一下我们都学习了哪些东西（姑且称那些枯燥而又有趣的理论为东西）：

第一章——背景知识，了解数据挖掘、机器学习、模式识别和深度学习的定义及其间的联系与区别；Deep Learning 的基本思想及其内的其他术语定义、联系和区别。（详细内容请查看邓老师的《深度学习讲义》或搜索查看）

第二章——前馈神经网络与反向传播算法，学习前馈神经网结构；网络参数的学习（更新）方法——梯度下降；反向传播算法，即残差的回传；矩阵微分，即函数微分与导数，函数梯度，Jacobian 矩阵等。（这一章不仅是考试推导题的重点还是后面章节学习的基石!）

第三章——网络输出层单元，这一章主要对深度学习网络的输出层进行探讨，介绍了四种输出单元。分别是线性输出单元、Sigmoid 输出单元、Softmax 输出单元、Sparse max 输出单元。这一章中的 Sigmoid、Softmax、Sprsemax 会在概念题、填空题、证明题中出现考察形式一般都容易掌握。

第四章——隐藏层单元，我们都知道一个普通的深度学习网络包含输入层（第 1 层）、隐藏层（通常有 n 层）和输出层，残差从输出层反向回传更新隐藏层的学习参数。
学完本章应掌握隐藏单元激活函数与残差的反向传播；Sigmoid 和双曲正切函数作为激活函数使用的特点；整流线性单元 ReLU；Softmax、Sparsemax 作为激活函数使用。（该章进行了一次练习，学生应掌握这次练习，以熟悉深度学习网络的过程。另外，隐藏层饱和态是可能考点出现在概念题中）

第五章——深度网络的正则化，即采用适当的策略显式地来减少泛化误差。

了解参数范数惩罚和 Dropout 两种正则化方法，特别了解 Dropout 的思想、优点及带 Dropout 的信息传播，Dropout 可能会在试题的概念题中出现。

第六章——Autoencode（自编码器 AE），掌握基本 AE 的网络结构和学习算法；了解稀疏自编码器、栈式自编码器、收缩自编码器的基本思想及学习过程。AE 的结构和基本思想会在概念题中出现。

第七章——卷积神经网络（CNN），了解卷积网的历史和种类；熟悉卷积神经网络的整体框架；卷积层学习过程和池化层学习过程；简化架构 CNN 的训练方法，两种残差的回传；最好后，了解组合特征映射的 CNN 的学习过程。第七章的卷积层中的局部感受野与共享参数可能会在概念题中出现，卷积层基本的卷积运算会在填空题中出现，简化架构 CNN 训练方法中的“残差通过卷积层来推导前一层的残差”可能会在证明题中出现。（总而言之，这一章还是比较重要的, 可以通过安装 TensorFlow 来实践卷积神经网络。如果你会 python 那就更好了，大量的数据处理库、机器学习库将为你服务！参考资料：邓世文老师《深度学习讲义》）

2 考点总结

2.1 小阅

课程考察形式：闭卷考试
题型：概念题、填空题、证明题、推导题
考察范围：《Lecture Notes in Deep Learning》

本人将以上述题型的顺序逐一对各考点进行解答与总结！

2.2 概念题

1、Deep Learning 的基本思想
答：自神经生物学表明大脑神经系统处理是以分层的方式来处理信息的，即高层的特征是低层特征的组合，从低层到高层的特征表示越来越抽象，越来越能表现语义或意图。深度学习借鉴这种分层的信息处理方式，其含有多个层，将当前层的输出作为下一层的输入，通过这种方式实现对输入信息进行分级表达和特征提取。
或答：深度学习借鉴了大脑神经系统以分层的方式处理的信息的方式，其含有多个层，将当前层的输出作为下一层的输入，通过这种方式实现对输入信息进行分级表达和特征提取。

2、Dropout 的目的及优点

1) 使用 Dropout 的目的：为了显式地减少泛化误差，以约减参数个数并间接地增加数据量，实现防止过拟合的目的。

2) Dropout 优点：与其他正则方法（稀疏约束等）相比，Dropout 更为有效且具有较小的计算开销。Dropout 还可以与其他正则方法合并使用，得到进一步的性能提升。

3) Dropout 的网络连接：（不在考点内）

3、AE 的构成与基本思想

1) AE 之构成：自编码器内部有一个隐藏层，可以产生编码其可作为输入的一种表示。AE 可以看成由两部分构成：一个由函数表示编码器和一个生成重构的解码器，但输出与输入并不完全相同。

2) AE 基本思想：试图将输入复制到输出。试图将网络的输入转化为网络的输出，常用来实现降维或压缩数据。

3) AE 值网络结构：不在考点范围内.

4、卷积层中蕴含的两个概念：局部感受野与共享参数

1) 局部感受野：在输入图像中，与隐藏层节点相连的区域称为此节点的局部感受野。

2) 共享参数：所谓参数共享，指的是所有隐藏层的节点采用相同的权值和偏置。参数共享意味着隐藏层的所有节点有输入图像的不同位置来检测的是相同的特征。共享参数的好处是极大地减少网络参数。

5、Pooling，池化不变性
答：选择图像中连续范围作为池化区域，并且只是池化相同（重复）的隐藏单元产生的特征，那么，这些池化单元就具有平移不变性。
图像经过一个小的平移、旋转后，仍产生相同的（池化）特征。（*MNIST 是一个手写数字库识别库：http://yann.learn.com/exdb/mnist）

6、隐藏层（Sigmoid）饱和态答：饱和态：当 z 取绝对值很大的整数时，Sigmoid 单元饱和到一个高值；当 z 取绝对值很大的负数时，Sigmoid 单元饱和到一个低值；仅当 z 接近 0 时，Sigmoid 单元才对输入强烈敏感。

7、输出层作用及常用函数

1) 输出层作用：是对这些特征（隐藏层的输出）进行一些额外的变换以得到符合要求的网络输出，并完成整个网络所需要完成的任务。

2) 常用函数: 线性输出单元（高斯输出分布）、Sigmoid 输出单元（Bernouli 输出分布）、Softmax 输出单元（Multinoulli 输出分布）、Sprsemax 输出单元。（详细请阅读深度学习讲义）

8、Machine Learning 中正则化的作用

答: 减少泛化误差，限制模型的学习能力；防止过拟合或欠拟合。常用的方法有参数范数惩罚,l^2 参数正则化,l^1 参数正则化,Dropout 这几种。

9、高维数据的流形假设

答: 假设高维空间（称作外围空间）中的数据 x 近似位于或接近于一个高维空间中的低维流形上，当 x 远离此流形时其概率分布 p(x) 将迅速减少到 0。

2.3 填空题

（1～3 题请牢记讲义中 2.4 矩阵微分节的规则及 2.4.4 的举例）

1、迹的循环等价性

$Tr(\mathcal{ABCD}) = Tr(\mathcal{DABC}) = Tr(\mathcal{CDAB}) = Tr(\mathcal{BCDA})$

2、含有 Hadamard 积的迹函数

$Tr(\mathcal{A}(\mathcal{B} \circ \mathcal{C}) = Tr(\mathcal{A} \circ \mathcal{B}^{\mathrm{T}})\mathcal{C})$

3、求 $f(x) = \|Ax - b\|_2^2$ 的梯度

解:

$$\begin{aligned} \mathrm{d}f &= (A\mathrm{d}x)^{\mathrm{T}}(Ax - b) + (Ax - b)^{\mathrm{T}}A\mathrm{d}x \\ &= (\mathrm{d}x)^{\mathrm{T}}A^{\mathrm{T}}(Ax - b) + (Ax - b)^{\mathrm{T}}A\mathrm{d}x \\ &= 2(Ax - b)^{\mathrm{T}}A\mathrm{d}x \end{aligned}$$

因此，$J = 2(Ax - b)^{\mathrm{T}}A, \bigtriangledown_x = 2A^{\mathrm{T}}(Ax - b)$

4、求 $Sigmoid()$ 的梯度或导数

解：$f(x) = sigmoid(x) = \frac{1}{1+exp(-x)}$

$f(x)' = (\frac{1}{1+exp(-x)})' = \frac{exp(-x)}{(1+exp(-x))^2}$

另解：$f(x)' = f(x) - f(x)^2 = f(x)(1 - f(x))$

5、$Sparsemax(z)$ 函数的优化问题，即要满足 KKT 条件（讲义第 31 页）

解:

$$min_q \frac{1}{2}\|p - z\|_2^2$$

$$s.t. \quad 1^{\mathrm{T}}p = 1$$

$$p \geq 0$$

上述问题含有 1 个等式约束和 K 格不等式约束，其 $Lagrangian$ 为

$$L(z, \mu, \tau) = \frac{1}{2}\|p - z\|_2^2 - (\mu)^{\mathrm{T}}p + \tau(1^{\mathrm{T}}p - 1)$$

KKT 条件：
令 p^*, μ^*, τ^* 为在最优点处的解，则其 KKT 条件为

$$p^* - z - \mu^* + \tau^* 1 = 0$$
$$1^{\mathrm{T}} p^* = 1$$
$$p^* \geq 0$$
$$\mu^* \geq 0$$
$$\mu^* p^* = 0, \quad \forall i \in [k]$$

2.4 证明题

1、已知 $Softmax$ 函数 $a = softmax(z) = exp(z)/1^{\mathrm{T}} exp(z)$, 证明 $softmax$ 函数作为激活函数时的梯度为 $diag(a) - aa^{\mathrm{T}}$.
证：（该题位于讲义第四章隐藏层单元，证明详细且正确。）
求 $softmax$ 函数的微分：

$$\begin{aligned} \mathrm{d}(softmax(z); z) &= diag(a)\mathrm{d}z - a1^{\mathrm{T}} diag(a)\mathrm{d}z \\ &= diag(a)\mathrm{d}z - aa^{\mathrm{T}}\mathrm{d}z \end{aligned}$$

注：$1^{\mathrm{T}} diag(a) = a^{\mathrm{T}}$。
因此，$softmax$ 函数的梯度为

$$\bigtriangledown_z softmax(z) = diag(a) - aa^{\mathrm{T}}$$

注：由于 $softmax()$ 函数的导数形式过于复杂，编译时出现错误，固在此省略！

2、证明残差通过卷积层，已知 $l+1$ 层（卷积层）残差，推导残差通过卷积层回传的公式.
证：（位于讲义第七章第 76～77 页，详细。请自行阅读、推导和证明）

$$\rightarrow_{z_l} Pooling : f(z) \rightarrow_{a_l} down(z) \rightarrow_{z_{l+1}} Convolutional : f(z)$$

信息的正向传播过程为

$$Pooling: a_l = f(z_l)$$

$$Convolutional: z_{l+1} = \omega * a_l + b11^{\mathrm{T}}$$

求微分

$$\begin{aligned} \mathrm{d}(J; z_l) &= Tr(D[J; z_{l+1}]\mathrm{d}(z_{l+1}; z_l)) \\ &= Tr((\delta_{l+1})^{\mathrm{T}}[\omega * \mathrm{d}(a_l; z_l])) \\ &= Tr((\delta_{l+1})^{\mathrm{T}} W(f'(z_l) \circ \mathrm{d}z_l)) \\ &= Tr([(\delta_{l+1})^{\mathrm{T}} W) \circ (f')^{\mathrm{T}}(z_l)]\mathrm{d}z_l) \end{aligned}$$

因此，梯度为

$$\delta_l = \bigtriangledown_{z_l} J = (\widetilde{\omega} * \delta_{l+1}) \circ f'(z_l)$$

相对应的 $MATLAB$ 代码为

$$r1 = conv2(rot90(\omega, 2), r2, 'full'). * f'(z_l)$$

其中，$r: residual$(残差) $r1$: 当前 l 层残差; $r2$: 第 $l+1$ 层残差。$f'() : f()$ 的导数。z_l: 第 l 层的净输入。

2.5 推导题

1、推导: 前馈神经网络的反向传播算法（基于向量）

问:

(1) $J()$ 关于 W 的梯度;

(2) $J()$ 关于 b 的梯度;

(3) 当前层残差与前一层残差的关系;

(4) (W, b) 参数学习（更新）.

答:（本题位于第一版讲义第二章第 7∼9 页，(2.4 节）残差的反向传播贯穿整个深度学习的始末也是基础，希望同学掌握!）
(a) 输入数据 $x \in \mathrm{R}^N$, 输出 $a \in \mathrm{R}^M$。
(b) 网络参数: $W \in \mathrm{R}^{M\times N}$。
(c) 净输入:$z = Wx + b$, 非线性输出:$a = f(z)$。
(d) 深层网络架构:

$$a_1 = x \to z_2|a_2 \to \ldots \to z_l|a_l \to J(z_l)$$

(e) 信息的正向传播
第 1 层输出： $a_1 = x$
第 2 层输出： $a_2 = f(z_2), z_2 = W_1a_1 + b_1$;
$\vdots$
第 l 层： $a_l = f(z_l), z_l = W_{l-1}a_{l-1} + b_l$;
$\vdots$
第 L 层： $J(z_L) = \frac{1}{2}\|z_L - y\|_2^2$.
(f) 反向传播算法
求解第 l 层网络参数相当于求解第 L 层目标函数优化问题
令 y 是 x 的函数，D[y;x] 表示 y 对 x 的 Jacobian 矩阵，即
$\bigtriangledown_x y = \frac{\partial_y}{\partial_x} = D[y;x]^{\mathrm{T}}$。

(1)J 关于 W 的梯度
J 关于参数 W_l 的微分
$\mathrm{d}(J;W_l) = Tr(D[J;z_{l+1}]\mathrm{d}(z_{l+1};W_l))$
$= Tr(D[J;z_{l+1}]\mathrm{d}(W_la_l))$
$= Tr(a_lD[J;z_{l+1}]\mathrm{d}(W_l))$
J 关于参数 W_l 的梯度

$$\bigtriangledown_{w_l} J = D[J;z_{l+1}]^{\mathrm{T}}(a_l)^{\mathrm{T}} = \bigtriangledown_{z_{l+1}} J(a_l)^{\mathrm{T}} = \delta_{l+1}(a_l)^{\mathrm{T}}$$

(2)J 关于 b_l 的梯度
J 关于参数 b_l 的微分
$\mathrm{d}(J;b_l) = Tr(D[J;z_{l+1}]\mathrm{d}(z_l);D[J;z_{l+1}]\mathrm{d}b_l)$

$= Tr(D[J; z_{l+1}]\mathrm{d}b_l)$
J 关于参数 b_l 的梯度

$$\bigtriangledown_{b_l} J = D[J; z_{l+1}]^{\mathrm{T}} = \delta_{l+1}$$

(3) 当前层残差与前一层残差的关系
由信息的正向传播关系，可的到

$$a_l = f(z_l)$$

$$z_{l+1} = W_l a_l + b_l$$

即

$$z_{l+1} = W_l f(z_l) + b_l$$

残差 δ_l 与 δ_{l+1} 的关系

$$\mathrm{d}(J; z_l) = Tr(D[J; z_{l+1}]\mathrm{d}(z_{l+1}; z_l))$$

$$= Tr(D[J; z_{l+1}]W_l \mathrm{d}(f(z_l; z_l)))$$

$$= Tr(D[J; z_{l+1}]W_l(f'(z_l) \circ \mathrm{d}z_l))$$

$$= Tr([(D[J; z_{l+1}]W_l) \circ (f')^{\mathrm{T}}(z_l)]\mathrm{d}z_l)$$

因此，得到

$$\delta_l = ((W_l)^{\mathrm{T}}\delta_{l+1}) \circ f'(z_l)$$

(4) 参数更新方法

$$W_l^{(k+1)} = W_l^{(k)} - \lambda \bigtriangledown_{w_l} J$$
$$b_l^{(k+1)} = b_l^{(k)} - \lambda \bigtriangledown_{b_l} J$$

其中，参数的梯度为

$$\bigtriangledown_{w_l} J = \delta_l a_{l-1}^{\mathrm{T}} \quad (= \delta_{l+1} a_l^{\mathrm{T}})$$

$$\bigtriangledown_{b_l} J = \delta_l \quad (= \delta_{l+1})$$

注：我并不确定哪种形式是对的!

3 Deep Learning 兴趣参考

3.1 视频课程类

1.0 https://www.coursera.org
1.1 网易公开课或者其他视频资料
1.2 阿里云、天池等
...

3.2 工具类

2.0 Tensorflow http://www.tensorfly.cn
2.1 Scikit-learn http://cwiki.apachecn.org/pages/viewpage.action?pageId=10030179
2.2 Python 太多了，不列举了!
2.3 Matlab or Octave
2.4 人脸识别库 $https://github.com/ageitgey/face_recognition$
2.5 像 Dlib、Opencv、Numpy 太多了!
2.6 强大的 Github!
...

3.3 博客文档类

[–]Tip[1]
3.0 https://github.com/PacktPublishing/Python-Machine-Learning-Blueprints/blob/master/Chapter
3.1 http://www.deeplearningbook.org
3.2 http://www.cs.ubc.ca/ schmidtm/
3.3 http://nkonst.com/machine-learning-explained-simple-words/
3.4 http://speech.ee.ntu.edu.tw/ tlkagk/courses.html

[1] You can find this article on my GitHub!

3.5 https://nips.cc/
3.6 https://icml.cc/
3.7 http://www.jmlr.org/
3.8 http://www.tensorfly.cn/
@ https://github.com/tataya[2]/
...

总之，非常多。什么开源中国、CSDN、博客园、各种技术社区、Stackflow、JMLR、NIP、ICCML、阿里云国内国外会议期刊，自己找吧！

强烈推荐 LaTeX!

参考文献

[1] Shi-wen Deng: Lecture Notes in Deep Learning

[2] www.deeplearningbook.org

[2] Follow me on GitHub!