

Task 5: AI Books—stage 2

任务描述

选择Task5(AIbooks)，第二阶段选择的书为Artificial Intelligence Machine Learning and Deep Learning
第二阶段翻译的部分为本书4、5章

- 深度学习概论
- 深度学习：RNN和LSTM模型

一、深度学习概论（报告人：李逸博）

本章第一部分介绍了深度学习的概念、深度学习可以解决的问题。第二部分讲了神经网络的基本概念和构建，同时还介绍了几个基本的神经网络，包括ANN、MLP。第三部分详细介绍了卷积神经网络(CNN)，之后还给出了一个使用MNIST数据及训练基于keras的CNN示例。

1.原理介绍

1.1 与无监督学习的关系

- 用于研究神经网络、训练神经网络。
- 需要神经网络中至少有两个隐藏层
- 深度学习是学习样本数据的内在规律和表示层次，这些学习过程中获得的信息对诸如文字，图像和声音等数据的解释有很大的帮助。

1.2 深度学习可以解决的问题

因为反向传播涉及到更新边之间的权重，在权重更新的过程中，会产生梯度消失和梯度爆炸的问题。深度学习可以解决这些问题（可以通过LSTM实现）。深度学习模型用ReLU函数替代sigmoid激活函数。

1.3 深度学习中的挑战

- 算法中的偏差（包括数据中心的偏差和运算过程中的偏差）
- 缺乏可解释性
- 概括能力有限
- 结果为相关关系，无法表达因果关系

2.深度学习模型

2.1 感知器

感知器使用特征向量来表示的前馈神经网络，是一种二元分类器，把矩阵上的输入映射到输出值 $f(x)$ （一个二元的值）。

$$f(x) = \begin{cases} 1 & w * x + b > 0 \\ 0 & else \end{cases}$$

x 是输入向量, w 是实数的表式权重的向量, $w * x$ 是点积。 b 是偏置，一个不依赖于任何输入值的常数。偏置可以认为是激励函数的偏移量，或者给神经元一个基础活跃等级。 $f(x)$ (0 或 1)用于对进行分类，看它是肯定的还是否定的，这属于二元分类问题。

2.2 深度学习中的超参数

超参数包括隐藏层数量、隐藏层神经元数量、边的权重、激活函数、损失函数、优化器、学习率、丢弃率。

- 初始化模型的超参数

隐藏层数量、隐藏层神经元数量和边的权重是初始化模型中必要的部分。隐藏层是中间的计算层，神经元的数量可以自己决定，神经元之间的连接也可以决定。连接神经元之间的边具有权重，权重的初始值通常介于0，1之间，这个权值会在训练过程中进行调整。

- 激活函数

激活函数应用于每对连接层之间的权重，通过计算改变神经元的输出。具有许多层的神经网络通常包含不同的激活函数。

- 损失函数

损失函数、优化器、学习率都是在反向误差传播的过程中使用的。常见的损失函数有MSE函数、交叉熵函数等。

- 优化器

优化器是结合损失函数选择的算法，目的是在训练阶段收敛到损失函数的最小值。常见的优化器包括：SGD、Adagrad、Adam等。

- 学习率

学习率是一个很小的数字，通常在0.001和0.05之间，在使损失函数取最小值得过程中，可以使得越靠近最低点，降低的速度越慢。

- 丢弃率

是一个介于0和1之间的数，确定的是前向传播的过程中丢弃的神经元的比例。例如：如果丢弃率为0.2，则在前向传播的每个步骤中随机选择20%的神经元忽略不计。

2.3 反向误差传播

反向传播的神经网络是由一个输入层、一个输出层和一个或多个隐层构成的，它的激活函数采用 *sigmoid* 函数。在这其中，输入信号经输入层输入，通过隐层计算由输出层输出，输出值与标记值比较，若有误差，将误差反向由输出层向输入层传播，在这个过程中，利用梯度下降算法对神经元权值进行调整。

3. 多层感知器 (MLP)

- MLP实际上就是人工神经网络。除了输入输出层，它中间可以有多个隐层，最简单的MLP只含一个隐层，即三层的结构。多层感知机层与层之间是全连接的（全连接的意思就是：上一层的任何一个神经元与下一层的所有神经元都有连接）。
- 输入层：比如输入是一个 n 维向量，就有 n 个神经元。
- 隐藏层的神经元，与输入层是全连接的，假设输入层用向量 X 表示，则隐藏层的输出就是

$$f(W_1 X + b_1)$$

， W_1 是权重（也叫连接系数）， b_1 是偏置，函数 f 可以是常用的 *sigmoid* 函数或者 *tanh* 函数。

- 输出层，隐藏层到输出层可以看成是一个多类别的逻辑回归，也即 *softmax* 回归，所以输出层的输出就是 $softmax(W_2 X_1 + b_2)$ ， X_1 表示隐藏层的输出 $f(W_1 X + b_1)$ 。

4. 卷积神经网络

卷积神经网络是一种多层神经网络，擅长处理图像特别是大图像的相关机器学习问题。

卷积网络通过一系列方法，成功将数据量庞大的图像识别问题不断降维，最终使其能够被训练。

卷积神经网络的层级结构包括：

- 卷积层
- 池化层
- 全连接层

卷积层与池化层配合，组成多个卷积组，逐层提取特征，最终通过若干个全连接层完成分类。

CNN通过卷积来模拟特征区分，并且通过卷积的权值共享及池化，来降低网络参数的数量级，最后通过传统神经网络完成分类等任务。

4.1 卷积层(Conv2D)

- 有两个关键操作：
 - 局部关联。每个神经元看做一个滤波器(filter)
 - 窗口(receptive field)滑动， filter对局部数据计算
- 卷积核大小 (kernel size)

决定一个点的输出值与多大的范围的输入有关->感受野 (respective filed) 通常是奇数，因为奇数大小的核，才能保证中心点四周的长度是相等的。通常卷积核越大效果越好，但增大卷积核伴随着参数个数平方增长。

- 填充 (padding)
目的是不减小feature map的大小
常见策略：周边补0；向外对称。
- 步长 (stride)
用于降低feature maps的尺寸

4.2 ReLU激活函数

在创建每个特征图之后，特征图上的某些值可能是负值。ReLU激活函数的功能是将这些负值替换为0。ReLU的定义如下：

$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

4.3 池化层(pooling)

即使做完了卷积，图像仍然很大（因为卷积核比较小），所以为了降低数据维度，就进行采样。

即使减少了许多数据，特征的统计属性仍能够描述图像，而且由于降低了数据维度，有效地避免了过拟合。

在实际应用中，分为最大值采样（Max-Pooling）与平均值采样（Mean-Pooling）。

eg:原始图片是20x20的，对其进行采样，采样窗口为10x10，最终将其采样成为一个2x2大小的特征图。

4.4 Dropout

- 训练时，每次随机去掉全连接层中的一些神经元
- 防止或减轻过拟合

4.5 Batch normalization

- 对输入数据进行减均值，除方差的操作
- 加快训练速度

5.总结

第四章深度学习概论主要讲了相关概念、神经网络的基本结构并介绍了具体的神经网络。

在翻译的过程中，我对深度学习有了更清晰的认识，对神经网络的基本结构和构建方法有了了解。通过书中利用keras在MNIST数据集上训练神经网络模型的具体示例，我也学到了如何在代码层面实现一个具体的神经网络。

6.规划

我们已经了解了机器学习、深度学习等基本知识，第三阶段我们选择翻译的书是*Unsupervised Learning in Space and Time*,阅读并翻译前三章的部分。