# 《Python 深度学习》读书笔记

## 深度学习基础概念

### 学习与深度的含义

学习本质上是自动寻找更好数据表示方法的过程。而 “深度” 指的是利用一系列连续的表示层。数据模型中包含的层数体现了模型的深度。

### 机器学习与深度学习的关联

机器学习通过观察大量输入和目标的示例，将输入（如图像）映射到目标（如标签 “猫”）。深度学习则是机器学习中的一类特殊函数形式，通常以多层神经网络的形态出现。

### 有监督学习、无监督学习与半监督学习

1. **有监督学习**：训练数据包含已知结果，模型依据这些结果进行训练。基本目标是给定训练样本\{x\_i,y\_i\}构造预测器，将实际映射到标签上。若y为真实值，属于回归任务；若y是有限集合中的值，则为模式分类。
2. **无监督学习**：训练数据不包含已知结果，算法自行探寻数据中的联系。主要利用数据的可变性和可分性等特征，确定属性 / 特征的相关性，目标是将未标注数据分成不同聚类。
3. **半监督学习**：结合大量未标注数据和少量标注数据，适用于仅有有限带标注观测值的情况。

## 神经网络的数学基础

### 相关术语定义

1. **类（class）**：分类问题中的某个类别。
2. **样本（sample）**：数据点。
3. **标签（label）**：某个样本对应的类。
4. **训练集（training set）**：用于模型学习的数据，如 MNIST 数据集中的train\_images和train\_labels。
5. **测试集（test set）**：用于测试模型性能的数据，如 MNIST 数据集中的test\_images和test\_labels。

### 神经网络的数据表示 - 张量

1. **标量（0D 张量）**：仅包含一个数字的张量，如s = 12。
2. **向量（1D 张量）**：数字组成的数组，只有一个轴。例如v = [1, 2, 3, 4, 5]是一个 5D 向量。
3. **矩阵（2D 张量）**：向量组成的数组，有两个轴。如m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]，形状为(3, 3)。
4. **3D 张量与更高维张量**：多个矩阵组合成的新数组为 3D 张量，可直观理解为数字组成的立方体。更高维张量依此类推。例如，一个形状为(2, 3, 4)的 3D 张量，可看作是由 2 个形状为(3, 4)的矩阵组成。
5. **关键属性**
   * **轴的个数（阶）**：张量的维度数量。
   * **形状**：一个整数元素的元组，表示张量沿每个轴的维度大小。例如，向量[1, 2, 3]的形状为(3,)，标量5的形状为()。
   * **数据类型**：张量中所包含数据的类型，如float32、uint8等。
6. **在 Numpy 中操作张量**：通过张量切片选择特定元素，如train\_images[i]选择沿着第一个轴的第i个数字。
7. **数据批量的概念**：深度学习中数据张量的第一个轴通常是样本轴，如 MNIST 例子中的数字图像样本。

### 简单神经网络构建步骤（以 MNIST 数据集为例）

1. **加载 Keras 中的 MNIST 数据集**

|  |
| --- |
| from keras.datasets import mnist  (train\_images, train\_labels), (test\_images, test\_labels) = mnist.load\_data() |

图像编码为 Numpy 数组，标签为数字数组，取值0 - 9，图像与标签一一对应。

2. **网络架构搭建**

|  |
| --- |
| from keras import models  from keras import layers  network = models.Sequential()  network.add(layers.Dense(512, activation='relu', input\_shape=(28 \* 28,)))  network.add(layers.Dense(10, activation='softmax')) |

神经网络核心组件层（layer）可看作数据过滤器。这里构建了一个包含两个全连接层的网络，输入层将28\times28的图像数据展平为一维向量，第一个隐藏层有512个神经元，使用 ReLU 激活函数；输出层有10个神经元，使用 softmax 激活函数，输出 10 个数字类别的概率。

3. **编译网络**

|  |
| --- |
| network.compile(optimizer='rmsprop',  loss='categorical\_crossentropy',  metrics=['accuracy']) |

编译需选择三个参数：

* **损失函数（loss function）**：衡量网络在训练数据上的性能，引导网络朝着正确方向优化，如分类问题常用的交叉熵损失函数。
* **优化器（optimizer）**：基于训练数据和损失函数更新网络的机制，如 RMSProp 优化器。
* **监控指标（metric）**：训练和测试过程中关注的指标，如精度（accuracy）。

1. **准备图像数据**

|  |
| --- |
| train\_images = train\_images.reshape((60000, 28 \* 28))  train\_images = train\_images.astype('float32') / 255  test\_images = test\_images.reshape((10000, 28 \* 28))  test\_images = test\_images.astype('float32') / 255 |

训练前对数据预处理，将图像数据形状变换为网络所需，缩放到0 - 1区间。原图像数据为uint8类型，形状(60000, 28, 28)，取值0 - 255，转换为float32类型，形状(60000, 28 \* 28)。

5. **准备标签**

|  |
| --- |
| from keras.utils import to\_categorical  train\_labels = to\_categorical(train\_labels)  test\_labels = to\_categorical(test\_labels) |

对标签进行分类编码，将数字标签转换为 one - hot 编码形式，便于计算损失函数。

6. **开始训练网络**

|  |
| --- |
| network.fit(train\_images, train\_labels, epochs=5, batch\_size=128) |

通过调用网络的fit方法在训练数据上拟合模型。epochs指定训练轮数，batch\_size指定每次训练使用的样本数量。

### 神经网络的学习算法

1. **反向传播法**
   * 确定权重初始值。
   * 前向反馈：输入数据经各层神经元处理，得到输出结果。
   * 误差评估：计算输出结果与实际标签的误差。
   * 使用输出层的误差重新调整权重，并计算相对于权重值变化的误差变化的梯度。
   * 以降低变化梯度对权重做出调整。根据激活函数的导数、网络输出和实际目标结果之间的偏差，以及神经元输出调整每个神经元的权重和偏差。神经网络初始化时，权重和偏差设置为( - 2n ~ 2n)内的随机值，n为输入属性数量。
2. **梯度下降算法及问题解决**
   * **梯度下降算法**：沿着损失函数梯度的反方向更新权重，使损失函数值减小。但容易陷入局部极小值。
   * **解决方法**
     + 指定一个冲量参数，使权重更新具有一定惯性，避免陷入局部极小值。
     + 使用随机梯度下降（SGD）：随机选择样本更新参数，沿着样本相关梯度移动，路径曲折但能找到全局最小值（若损失函数为凸函数）。
3. **学习速率**

学习速率决定梯度下降算法达到极小值的步长大小。大学习速率使网络学习快速，但可能错过全局最小值；学习速率过慢则训练时间长。最佳学习速率通常接近最大学习速率的一半，可从大学习速率开始尝试，若偏离训练标准，将学习速率设为原来的1/3，反复调整直到无偏离。

### 激活函数

1. **sigmoid 函数（logistic）**：接收实数值，将其压缩到0 - 1范围，输出可解释为人工神经元发射的概率，运算成本低，公式为f(u)=\frac{1}{1 + exp( - cu)}，c为参数（如1.5）。
2. **ReLU 函数**：常用于隐藏层激活函数，在图像压缩等任务中表现良好，能使网络更好地泛化，减少实际压缩 - 解压时间，公式为f(x) = max(0, x)。

### 深度神经网络简介（DNN）

可视为多个回归模型的组合，在某些情况下，每个隐藏层可解释为简单的对数线性模型。一个隐藏层足以模拟任何分段连续函数，如研究人员 Hornik 等人的定理所示：设F是n维空间有限子集上的连续函数，存在包含有限个隐藏单元的双层神经网络\hat{F}，对于F域中的所有x，\vert F(x)-\hat{F}(x)\vert<\sigma。

## 深度学习应用案例

### 图像识别

在 MNIST 手写数字识别任务中，通过构建合适的神经网络模型，对图像数据进行处理和特征提取，实现对数字的准确分类。在更复杂的图像识别任务如 ImageNet 大赛中，使用更深层次的卷积神经网络（如 VGG、GoogleLeNet、ResNet 等），不断提升图像识别的准确率。这些网络通过增加网络深度、引入新的结构（如 Inception 结构、快捷结构）等方式，提高网络对图像特征的学习能力。

### 物体检测与图像分割

深度学习在物体检测任务中，能够识别图像中不同物体的类别和位置。图像分割则是将图像中的每个像素划分到对应的物体类别中。通过构建特定的神经网络模型，如基于卷积神经网络的物体检测模型和全卷积神经网络的图像分割模型，利用大量标注数据进行训练，实现对复杂场景中物体的准确检测和分割。