目录

第五章　神经网络

0.1　神经元模型

神经网络是由具有适应性的简单单元组成的广泛**并行互连**的网络，它的组织能够模拟生物神经系统对真实世界物体所做出的反应。神经元（neuron）模型是神经网络最基本的组成成分。M-P 神经元模型：神经元收到来自 n 个其他神经元传递过来的输入信号，这些输入信号通过带权重的连接进行传递，将神经元接收到的总输入值和神经元的阈值进行比较，然后通过激活函数（activation function）处理以产生神经元的输出。

$$y = f(\sum_{i=1}^{n} w_i x_i - \theta) \tag{1}$$

激活函数一般要求的性质：

- 非线性：内在反映的是模型的非线性
- 可微性：以支持梯度下降法等基于微分的优化方法
- 单调性：以保证单层网络是凸约束，从而存在最优解
- 输出值范围受限：输出值有限时，寻优算法（如梯度下降）变得更稳定。

常用的激活函数：
对数几率函数：$\sigma(x) = \text{sigmod}(x) = \frac{1}{1+e^{-x}}$
双曲正切函数：$\tanh(x) = 2\sigma(2x) - 1$
修正线性函数：$\text{ReLU}(x) = \max(0, x)$
径向基函数（Radial Basis Function）：高斯径向基函数：$P(x, x') = e^{-\frac{(x-x')^2}{2\sigma^2}}$，其中 σ 为自由参数。
softmax 函数：softmax 函数常用于将多个标量映射为一个概率分布。$\text{softmax}(x_k) = \frac{e^{x_k}}{\sum_{i=1}^{n} e^{x_i}}$

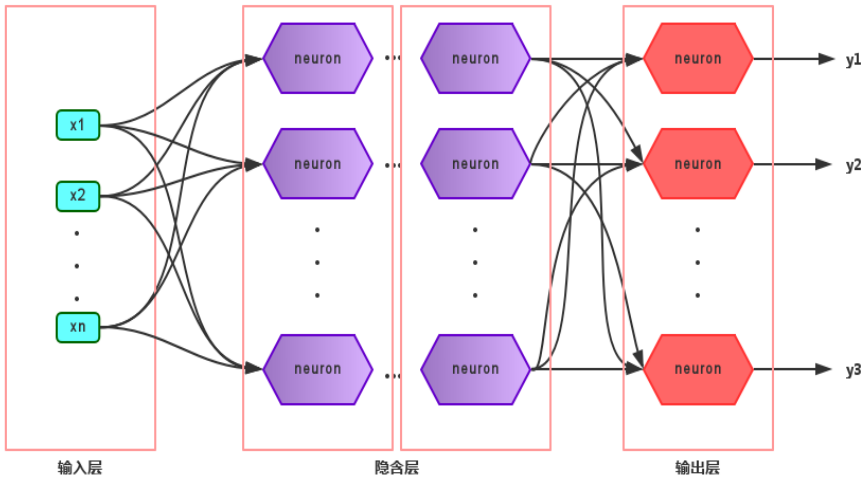
0.2　感知机与多层网络

感知机（Perceptron）是由输入层和输出层是 M-P 神经元的两层神经元组成。因为感知机只有输出层神经元进行激活函数处理，即只拥有一层功能神经元，**只能解决线性可分问题**。

多层功能神经元可以解决非线性可分问题。输入层和输出层之间的隐含层与输出层神经元都是拥有激活函数的功能神经元。最为常见的多层神经网络——**多层前馈神经网络**（multi-layer feedforward neural networks），它有以下特点：

- 每层神经元与下一层神经元全互连
- 神经元之间不存在同层连接
- 神经元之间不存在跨层连接

神经网络的学习其实就是调整各神经元之间的连接权（connection weight）以及各神经元的阈值。



多层前馈神经网络

0.3 误差逆传播算法

误差逆传播算法（error BackPropagation，简称 BP）也称为反向传播算法，是最为成功的一种神经网络学习方法之一。一般而言，BP 神经网络是指用 BP 算法训练的多层前馈神经网络，但 BP 算法也能训练其他类型的神经网络，如递归神经网络。

0.3.1 标准 BP 算法

假设要训练的是一个单隐层的前馈神经网络，BP 算法使用**均方误差作为性能度量**，基于梯度下降（gradient descent）策略，以目标函数的负梯度方向对参数进行调整。

输入：训练集 $D = (\mathbf{x}_k, \mathbf{y}_k)_{k=1}^m$，学习率 η。
过程：
1：在 $(0, 1)$ 范围内随机初始化网络中所有连接权和阈值
2：**repeat**
3：　**for all** $(\mathbf{x}_k, \mathbf{y}_k \in D)$ **do**
4：　　根据当前参数计算出样本的输出 $\hat{\mathbf{y}}_k$
5：　　计算输出层神经元的梯度项 g_j
6：　　计算隐层神经元的梯度项 e_h
7：　　更新连接权与阈值
8：　**endfor**
9：**until** 达到停止条件
输出：连接权与阈值确定的多层前馈神经网络

所谓逆传播其实就是从输出层开始逐步往后更新，因为输出层的误差确定后就可以对输出层的连接权和阈值进行更新，并且可以推算出隐含层输出的“真实值”，从而计算出隐含层的“误差”，然后更新隐含层的连接权和阈值。BP 就是这样一种利用一层层倒推来最终更新整个神经网络的方法，每一层的更新公式其实和感知机用的是类似的。

在学习过程中，学习率 η 控制着每一轮迭代的更新步长，太大则容易振荡，太小则收敛速度太慢。所以常常设置 η 和迭代次数相关。

0.3.2 累积 BP 算法

BP 算法的目标是最小化训练集 D 上的累积误差：

$$E = \frac{1}{m}\sum_{k=1}^{m} E_k$$

而标准 BP 算法每输入一个样例就进行一次更新，所以它的参数更新非常频繁，而且不同样例可能会对更新起到抵消效果，从而使得模型需要更多次迭代才能到达累积误差的极小点。

如果把更新方式变为每输入一遍训练集进行一次更新，就得到累积 BP 算法，更新公式需要重新推导一下。这样更新一次就称为一轮（one round，亦称 one epoch）学习。

0.3.3 过拟合问题

鉴于 BP 神经网络强大的表达能力，很容易会遇到过拟合问题。主要有以下两种应对策略：

- **早停**（early stopping）：把数据集分为训练集和验证集，若训练集误差降低但测试集误差升高，就停止训练，并返回具有最小验证集误差的连接权和阈值。
- **正则化**（regularization）：在目标函数中添加一个用于描述网络复杂度的部分，比如连接权和阈值的平方和。这样训练时就会偏好较小的连接权和阈值，从而令输出更"光滑"。带正则化项的目标函数如下：

$$E = \lambda\frac{1}{m}\sum_{k=1}^{m} E_k + (1-\lambda)\sum_{i} w_i^2$$

其中 λ 是用于对经验误差和网络复杂度折中的参数，常通过交叉验证法来估计。

为什么可以用正则化来避免过拟合呢？ 过拟合的时候，**拟合函数的系数往往非常大**【过拟合，就是拟合函数需要顾忌每一个点，最终形成的拟合函数波动很大。在某些很小的区间里，函数值的变化很剧烈。这就意味着函数在某些小区间里的导数值（绝对值）非常大】，而正则化是通过约束参数的范数使其不要太大，所以可以在一定程度上减少过拟合情况。

0.4 局部最小

因为用梯度下降搜索最优解可能会陷入非全局最小解的局部极小解，在现实任务中，人们会使用以下这些策略试图跳出局部极小：

- **以多组不同参数值初始化多个神经网络**：经过训练后，取误差最小的解作为最终参数。这种方法相当于从多个不同的初始点开始搜索，从而增加找到全局最小解的可能性。
- **模拟退火**（simulated annealing）技术：每次迭代都以一定的概率接收比当前解差的结果，从而有机会跳出局部极小（当然也有可能跳出全局最小），每次接受"次优解"的概率会随着时间推移逐渐减小，从而保证了算法的稳定。
- **随机梯度下降**（stochastic gradient descent，简称 SGD）：在计算梯度时加入了随机因素，因此即便陷入局部极小点，梯度依然可能不为 0，从而有机会跳出局部极小，继续搜索。

除了这些方法之外，**遗传算法**也常用于训练神经网络以逼近全局最小。当这些技术大多是启发式，没有理论的保障。

0.5 其他常见神经网络

0.5.1 RBF 网络

RBF（Radial Basis Function）网络是一种单隐层前馈神经网络，它使用径向基函数作为隐层神经元的激活函数，输出层则直接使用隐层神经元的线性组合。

0.5.2 ART 网络

竞争型学习（competitive learning）是神经网络中常用的一种无监督学习策略。使用该策略时，网络中的输出神经元相互竞争，每次只有一个竞争获胜的神经元被激活，其它输出神经元被抑制，这种机制又称为胜者通吃（winner-take-all）。

ART（Adaptive Resonance Theory，自适应谐振理论）网络是竞争型学习的重要代表。该网络由四部份组成：**比较层、识别层、识别阈值、重置模块**。比较层就是输入层，只负责把样本传递给识别层。识别层也即输出层，但识别层的每个神经元对应一个模式类，而且神经元的数目可以在训练过程中动态增加以增加新的模式类。

识别层的每个神经元有一个对应的模式类的代表向量，每次输入一个样本，各识别层神经元相互竞争，代表向量与样本距离最小的胜出并被激活。获胜神经元会向其他神经元发送信号，抑制其激活。如果样本与获胜神经元的距离小于识别阈值，则被分到对应的模式类。否则，重置模块会在识别层新增一个神经元，代表向量为该样本。

ART 能有效缓解竞争型学习中的**可塑性-稳定性窘境**（stability-plasticity dilemma）。可塑性指神经网络要有学习新知识的能力，稳定性则指神经网络在学习新知识时要保持对旧知识的记忆。

ART 具备可塑性和稳定性，因此能进行**增量学习**（incremental learning）和**在线学习**（online learning）。

增量学习可以理解为建立模型后再收到新的样例时可以对模型进行更新，但不用重新训练整个模型，先前学得的有效信息会被保存。它可以逐个新样例进行更新，也能以批模型（batch-mode），每次用一批新样例来更新。

在线学习则可以理解为每拿到一个新样本就进行一次模型更新，不需要一开始就准备好完整的训练集，每次收到新样例都能继续训练。可以看作增量学习的一个特例。

0.5.3 SOM 网络

SOM（Self-Organizing Map，自组织映射）网络，又称为自组织特征映射网络或 Kohonen 网络。同样是一种竞争学习型无监督神经网络，只有输入层和输出层两层，输出层以矩阵形式排列。与样本距离最近的输出层神经元获胜，称为最佳匹配单元（best matching unit）。最佳匹配单元和邻近神经元的权向量会被调整，使得下次遇到相似的样本时距离更小。如此迭代，直至收敛。

0.5.4 级联相关网络

级联相关（Cascade-Correlation）网络是一种典型的结构自适应网络，这类网络不仅通过训练来学习合适的连接权和阈值等参数，还会在训练过程中找到最符合数据特点的网络结构。

级联相关神经网络有两个主要成分：

- 级联：指建立层次连接的层级结构。开始训练时，只有输入层和输出层，随着训练进行逐渐加入隐层神经元，从而建立层级结构。注意，**隐层神经元的输入端连接权是冻结固定的**。
- 相关：指通过**最大化新神经元的输出与网络误差之间的相关性**（correlation）来训练相关的参数。

0.5.5 Elman 网络

递归神经网络（recurrent neural networks，简称 RNN）允许网络中出现环形结构，即一些神经元的**输出可以反馈回来当输入信号**，从而能够处理与时间有关的动态变化。

Elman 网络是最常用的递归神经网络之一，只有一个隐层，并且隐层神经元的输出会被反馈，在下一时刻与输入层神经元的输入信号一起作为隐层神经元的新输入。隐层神经元一般采用 Sigmoid 函数作为激活函数，并用 BP 算法训练整个网络。

0.5.6 Boltzmann 机

神经网络中有一类基于能量的模型（energy-based model），**把网络状态定义为一个能量，能量最小时网络达到理想状态**，模型的学习过程就是最小化能量函数。Boltzmann 机就是这样的模型，同时也

是一种 RNN。

Boltzmann 机的神经元分为显层与隐层，显层用于表达数据的输入与输出，隐层则是数据的内在。每个神经元只有 0、1 两种状态，也即抑制和激活。

标准的 Boltzmann 机是全连接图，即任意两个神经元之间都相连。但复杂度太高，难以用于解决现实任务，实际应用中用的是**受限 Boltzmann 机**（Restricted Boltzmann Machine，简称 RBM），把标准 Boltzmann 机退化为二部图，只保留显层和隐层之间的连接，同一层直接不相连。

0.6 深度学习

理论上，参数越多，模型复杂度就越高，容量（capability）就越大，从而能完成更复杂的学习任务。怎么增大模型复杂度呢？两个办法，一是**增加隐层的数目**，二是**增加隐层神经元的数目**。前者更有效一些，因为它不仅增加了功能神经元的数量，还增加了激活函数嵌套的层数。但是对于多隐层神经网络，经典算法如标准 BP 算法往往会在误差逆传播时发散（diverge），无法收敛到稳定状态。

那要怎么有效地训练多隐层神经网络呢？一般来说有以下两种方法：

- **无监督逐层训练**（unsupervised layer-wise training）：每次训练一层隐结点，把上一层隐结点的输出当作输入来训练，本层隐结点训练好后，输出再作为下一层的输入来训练，这称为预训练（pre-training）。全部预训练完成后，再对整个网络进行微调（fine-tuning）训练。一个典型例子就是深度信念网络（deep belief network，简称 DBN）。这种做法其实可以视为把大量的参数进行分组，先找出每组较好的设置，再基于这些局部最优的结果来训练全局最优。
- **权共享**（weight sharing）：令同一层神经元使用完全相同的连接权，典型的例子是卷积神经网络（Convolutional Neural Network，简称 CNN）。这样做可以大大减少需要训练的参数数目。

事实上，深度学习可以理解为一种特征学习（feature learning）或者表示学习（representation learning），无论是 DBN 还是 CNN，都是通过多个隐层来把初始与输出目标联系不大的输入表示转化为与输出目标更密切的表示，使原来只通过单层映射难以完成的任务变为可能。