神经网络

1.神经元模型

目前对于神经网络的定义多种多样，其中使用最广泛的一种定义是：

> "神经网络是具有适应性的简单单元组成的广泛并行的互联网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应。"

神经网络中最基本的组成单元是神经元 $(neuron)$ 模型，即定义中的简单单元。

在1943年，诞生了沿用至今的 “M-P神经元模型”。在这个模型中，神经元接收到来自 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有权重的连接进行传递，神经元接收到的总输入值将与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。

理想中的激活函数是阶跃函数，他将输入值映射为输出值 “0” 和 “1”，“1”对应神经元兴奋，“0”对应神经元抑制。但是，阶跃函数存在不连续，不光滑等不好的属性。因此，常使用 sigmod 函数作为激活函数。sigmod函数可以把较大的值，对应到0~1的范围内。sigmod函数有时也称为“挤压函数$(squashing\ function)$”

神经网络就是把多个简单的神经元模型组合到一起构成的。同时也可以把神经网络看作成包含许多参数的数学模型，模型由多个函数构成。

2.感知机与多层网络

感知机有两层结构组成，一层是输入层，接受外部输入后传递给输出层。另一层是输出层，输出层是M-P神经元。也称“阈值逻辑单元”。

感知机可以实现逻辑与，或，非运算。

一般的，给定一个训练集，权重 $w_i(i=1,2,3,\dots,n)$ 以及阈值 θ 可通过学习得到。阈值 θ 可以看作一个固定输入为-1的哑节点$(dummy\ node)$ 所对应的连接权重 w_{n+1} ,这样权重和阈值的学习就可以统一为权重的学习。感知机学习的规则简单，对训练样例 (x,y) ,若当前感知机的输出为 $\hat{y}$,则感知机的权重将进行调整，

$$w_i \leftarrow w_i + \Delta w_i \tag{1}$$

$$\Delta w_i = \eta(y-\hat{y})x_i \tag{2}$$

其中 $\eta \in (0,1)$ 称为学习率。若感知机对训练样例结果预测正确，即 $y=\hat{y}$，感知机则不发生变化。反正，将根据错误的程度进行权重调整。

需要注意到，感知机的学习能力非常有限。若两类模式都是线性的，存在一个线性超平面将其分开，则感知机的学习过程中一定会收敛，从而求得适当的权向量，反之，则感知机会出现振荡。权向量难以稳定下来。

要解决这个问题，可以考虑使用多层神经元。输入层与输出层之间的神经元被称为隐含层或者隐层，隐含层和输出层都是拥有激活函数的功能神经元。

3.误差逆传播算法

多层网络的学习能力强于单层感知机。要想训练多层网络需要使用更强大的学习算法。误差逆传播

($Error\ Back\ Propagation$,简称 BP)算法就是其中杰出的代表。现实任务中使用神经网络时，大多是在使用BP算法在训练。其实BP算法不仅可用于多层前馈网络，也可以用在其他类型的神经网络，比如递归神经网络。但是通常说"BP网络",一般是指使用BP算法的多层前馈网络。

对每个训练样例，BP算法执行如下操作：先将输入示例提供给输入层神经元，然后逐层将信号前传，直到产生输出层的结果；然后计算输出层的误差，再将误差逆向传播至隐层神经网络，最后根据隐层神经元的误差来对连接权和阈值进行调整，该迭代过程循环进行，直到达到某些停止条件为止。

BP算法的目标是最小化训练集 D 的累计误差

$$E = \frac{1}{m}\sum_{k=1}^{m} E_k \tag{3}$$

上面的训练只是针对一个训练样例的训练结果，如果推导出基于累计最小化误差的规则，那么就得到了累计误差逆传播算法。

由于强大的拟合能力，BP神经网络经常会遇到过拟合的问题。训练误差持续降低，测试误差却不断上升。通常由两种策略来缓解过拟合问题。

第一种策略是“早停”。将数据分为训练集和验证集。演练集用来计算梯度，更新连接权和阈值。验证集用来更新误差，若训练集误差降低，验证集误差上升，则停止训练，同时返回具有最小验证集误差的连接权和阈值。

第二种策略是“正则化”。主要思想是，误差目标函数中增加一个用于描述网络复杂度的部分。例如连接权和阈值的平方和。仍然使用E_k 表示第k个训练样例上的误差，w_i 表示连接权和阈值，则误差目标函数更新为

$$E = \lambda\frac{1}{m}\sum_{k=1}^{m} E_k + (1-\lambda)\sum_{i} w_i^2 \tag{4}$$

> 增加连接权和阈值平方和这一项后，训练过程将会偏好比较小的连接权和阈值，使网络输出更“光滑”，从而对过拟合有所缓解。

其中$\lambda \in (0,1)$，用于对经验误差与复杂网络这两项进行折中，常通过交叉验证法来进行验证。

4.全局最小与局部最小

用 E 表示神经网络的误差，则它显然是关于连接权w 阈值 θ 的函数。此时，神经网络的训练过程可以看作一个参数寻优过程，即在参数空间中寻找一组最优参数使得E 最小。

我们常常会提到两种最优：局部极小($local\ minimum$)和全局最小($global\ minimum$)。直观的看，局部极小解是参数空间中的某个点，其邻域点的误差函数值均不小于该点的函数值；全局最小解则是指参数空间中所有点的误差函数值均不小于该点的误差函数值。两者对应的E分别称为误差函数的局部极小值和全局最小值。

在参数空间中可能会存在多个局部极小值，但只会有一个全局最小值。这就是说全局最小值一定是在多个局部极小值中的一个。我们在参数寻优的过程中是希望找到全局最小。

基于梯度的搜索时是使用最为广泛的参数寻优方法。在此类方法中，我们从初始解出发，迭代寻找最优参数值。每次迭代中，我们先计算误差函数在当前点的梯度，然后根据梯度确定搜索方向。比如，由于负梯度方向是函数值下降最快的方向，因此梯度下降法就是沿着负梯度方向搜索最优解。若误差函数在当前点的梯度为零，这就说明这个点是局部极小值，他的更新量将为零，其实也就是说参数的迭代更新将在这一点停下来。如果一个函数仅有一个局部极小值，那么这个点就是全局极小。当误差函数有多个局部极小，就不能保证找到的解就是全局最小。后一种情况我们称参数寻优陷入了局部极小。这显然不是我们所希望的。

在现实情况中，人们想到了跳出局部极小的策略。有以下几种：

- 设置不同的参数值初始化神经网络，按标准方法训练后，取其中误差最小的解作为最终参数。这其实相当于，从不同的初始点开始搜索，最终就会陷入不同的局部极小，在多个局部极小中选择最有可能获得接近全局最小的结果。
- 使用“模拟退火” ($Simulated\ Annealing$) 技术。模拟退火技术在每一步都以一定的概率接受比当前解更差的结果，从而有助于“跳出”局部极小。每步迭代的过程中，接受“次优解”的概率要随着时间的推移而逐渐降低，从而保证算法稳定。
- 使用随机梯度下降。与标准梯度下降精确计算梯度不同，随机梯度下降在计算梯度时加入了随机因素。即便陷入局部极小点，它计算的梯度仍可能不为零，这样就有机会跳出局部极小继续搜索。

5.神经网络常见算法

5.1 RBF网络

RBF($Radial\ Basic\ Function$,径向基函数)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激活函数，而输出层是对隐层神经元输出的线性组合。假定输入为 d 维向量 x ,输出为实值，则RBF网络可表示为

$$\varphi(\boldsymbol{x}) = \sum_{i=1}^{q} w_i \rho(\boldsymbol{x}, \boldsymbol{c_i}), \tag{5}$$

其中q为隐层神经元个数，$\boldsymbol{c_i}$ 和w_i 分别是第i 个隐层神经元所对应的中心和权重，$\rho(\boldsymbol{x}, \boldsymbol{c_i})$ 是径向基函数，这是某种沿径向对称的标量函数，通常定义为样本 $\boldsymbol{x}$ 到数据中心 $\boldsymbol{c_i}$ 之间欧氏距离的单调函数。常用的高斯径向基函数形如

$$\rho(\boldsymbol{x}, \boldsymbol{c_i}) = e^{-\beta_i ||\boldsymbol{x}-\boldsymbol{c_i}||^2} \tag{6}$$

[Park and Sandberg,1991]证明，具有足够多的隐层神经元的RBF网络能以任意精度逼近任意连续函数。

通常采用两部过程来训练RBF网络：第一步，确定神经元中心$\boldsymbol{c_i}$,常用的方式包括随机采样，聚类等；第二部，利用BP算法来确定参数w_i 和 β_i

5.2 ART网络

ART($Adaptive\ Resonance\ Theory$)网络是采用竞争型学习策略。是竞争型学习的重要代表。该网络有比较层，识别层，识别阈值和重置模块构成。其中，比较层负责接收输入样本，并将其传递给识别曾神经元。识别层每一个神经元对应一个模式类，神经元数目可在训练过程中动态增加以增加新的模式类。

接收到比较层的输入信号之后，识别层神经元之间相互竞争以产生获胜神经元。竞争最简单的方式是，计算输入向量与每个识别神经元所对应的模式类的代表向量之间的距离，距离最小者获胜。获胜神经元将向其他识别曾神经元发送信号，抑制其激活。若输入向量与获胜神经元所对应的代表向量之间的相似度大于识别阈值，则当前输入样本将被归为该代表向量所属类别，同时，网络连接权将会更新，使得以后在接受到相似输入样本时该模式会计算出更大的相似度，从而使得获胜神经元有更大可能获胜；若相似度不大于识别阈值，则重置模块将在识别层增设一个新的神经元，其代表向量就设置为当前输入量。

由上面的过程可以看出识别阈值对 ART 网络有重要影响。当识别阈值较高时，输入类别将会被分成比较多，比较精细的模式类；当识别阈值较低时，则会产生比较少，比较粗略的模式类。

ART 比较好的解决了竞争性学习中的“可塑性-稳定性窘境($Stability-Plasticity\ Dilemma$)”,可塑性是指神经网络要有学习新知识的能力，而稳定性则是指神经网络在学习新知识时要保持对旧知识的记忆。这就使得ART 网络有一个很重要的优点：可进行增量学习（$Incremental\ Learning$ ）或在线学习($Online\ Learning$)

早期的ART 网络只能处理布尔型输入数据，此后ART 发展成了一个算法族，包括能处理实值输入的$ART2$网络，结合模糊处理的$FuzzyART$网络，以及可进行监督学习的$ARTMAP$网络等

5.3 SOM网络

$SOM(Self-Organizing\ Map,$自动组织映射$)$ 网络是一种竞争学习型的无监督网络，它能将高位输入数据映射到低维空间(通常映射为二维)，同时保持输入数据在高维空间的拓扑结构，即将高维空间中相似样本点映射到网络输入层的邻近神经元。

SOM网络中的输入层神经元以矩阵方式排列在二维空间中，每个神经元都拥有一个权向量，网络在接受输入向量后，将会确定输出层获胜神经元，它决定了该输入向量在低维空间中的位置。SOM的训练目标就是为每个输入层神经元找到合适的权向量，以达到保持拓扑结构的目的。

SOM的训练过程很简单：在接收到一个训练样本后，每个输入层神经元会计算该样本与自身携带的权向量之间的距离，距离最近的神经元会成为竞争获胜者，成为最佳匹配单元($Best\ Matching\ Unit$).然后，最佳匹配单元及其邻近神经元的权向量被调整，以使得这些权向量与当前输入样本的距离缩小，这个过程不断迭代，直至收敛。

5.4 Elman网络

与前馈神经网络不同，“递归神经网络”($Recurrent\ Neural\ Networks$) 允许网络中出现环形结构，从而可让一些神经元的输出反馈回来作为输入信号。这样的结构与信息反馈过程，使得网络在 t 时刻的输出状态不仅与t 时刻的输入有关，还与$t-1$ 时刻的网络状态有关，从而能处理与时间有关的动态变化。

$Elman$ 网络时最常用的递归神经网络之一，其结构与多层前馈网络很相似，但隐层神经元的输出被反馈回来，与下一刻输入神经元提供的信号一起，作为隐层神经元在下一时刻的输入。隐层神经元通常采用$Sigmod$ 激活函数，而网络的训练则常通过推广的BP 算法进行。