### 第二章 深度学习基本理论

### 2.1 从人工智能到深度学习

人工智能的浪潮正在席卷全球，相关概念也越来越被人所熟知：人工智能（Artificial Intelligence，AI）、机器学习（Machine Learning，ML）、深度学习（Deep Learning，DL）等。

简而言之，机器学习是一种实现人工智能的方法，深度学习是一种实现机器学习的技术。可以用同心圆来展现出它们三者的关系，如图2.1所示。



图1 三者关系示意图

1956年，在美国汉诺斯小镇宁静的达特茅斯学院中，John McCarthy、Marvin Minsky、Claude Shannon、Allen Newell、Herbert Simon等科学家正聚在一起，讨论着一个完全不食人间烟火的主题：用机器来模仿人类学习以及其他方面的智能，提出了“人工智能”的概念。2012年以后，得益于数据量的上涨、运算力的提升和机器学习新算法（深度学习）的出现，人工智能开始大爆发。人工智能的研究领域也在不断扩大，包括：专家系统、机器学习、进化计算、模糊逻辑、计算机视觉、自然语言处理、推荐系统等。但目前的科研工作都集中在弱人工智能这部分，“智能”的实现主要归功于一种实现人工智能的方法——机器学习。

机器学习最基本的做法是使用算法来解析数据并从中学习，然后对真实世界中的事件做出决策和预测。与传统的为解决特定任务、硬编码的软件程序不同，机器学习是用大量的数据来“训练”，通过各种算法从数据中学习如何完成任务。机器学习方法直接来源于早期对人工智能领域的研究，传统的机器学习算法包括：决策树、聚类、贝叶斯分类、支持向量机、EM、Adaboost等。从学习方法上来分，可以分为监督学习、无监督学习、半监督学习、集成学习、深度学习和强化学习。传统的机器学习算法在指纹识别、基于Haar的人脸检测、基于HOG特征的物体检测等领域的应用基本达到了商业化的要求，但每前进一步都异常艰难，直到深度学习算法的出现。

深度学习本来并不是一种独立的学习方法，其本身也会用到有监督和无监督的学习方法来训练深度神经网络。但由于近几年该领域发展迅猛，一些特有的学习手段相继被提出（如残差网络），因此越来越多的人将其单独看作一种学习的方法。最初的深度学习是利用深度神经网络来解决特征表达的一种学习过程。深度神经网络本身并不是一个全新的概念，可大致理解为包含多个隐含层的神经网络结构。为了提高深层神经网络的训练效果，人们对神经元的连接方法和激活函数等方面做出相应的调整。深度学习摧枯拉朽般地实现了各种任务，使得似乎所有的机器辅助功能都变为可能。无人驾驶汽车，预防性医疗保健，甚至是更好的电影推荐，都近在眼前或者即将实现。

### 2.2 神经网络学习机制

神经网络的学习机制主要体现在神经网络模型的训练过程，在网络模型训练的过程中实现对模型参数的不断更新，从而通过这些参数拟合出最佳的函数。在训练神经网络之前要定义好损失函数（或代价函数），实现对前向传播结果的评估得到网络计算结果的误差。参数是否更新就取决于是否存在误差，然而在深层的神经网络中存在激活函数，各层的输入与输出是非线性的映射关系，无法通过方程式求解。因此，针对深度神经网络的训练过程提出了不断迭代的、基于梯度方向的参数更新过程。在深神经网络训练时经常使用的思想包括：反向传播算法(Backpropagation)、随机梯度下降(Stochastic gradient descent，SGD)算法。

### 2.2.1误差反向传播算法

误差反向传播法是Rumelhart等在1986年提出的，即BP(error BackPropagation)算法，影响最为广泛。直到今天，BP算法仍然是自动控制上最重要、应用最多的有效算法。是用于多层[神经网络](https://baike.baidu.com/item/%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C/174248)训练的著名算法，有理论依据坚实、推导过程严谨、物理概念清楚、通用性强等优点。

误差反向传播算法的主要思路是通过前向传播计算出网络输出结果，利用构建的损失函数计算出网络输出结果与真实值之间的误差，然后逐层将误差从输出层向前层反向传播直至各层网络，每一层网络利用传递过来的误差沿着梯度下降的方向更新模型参数。完全更新一次网络参数为一次迭代周期，迭代直至模型收敛。

下面详细介绍误差反向传播算法的推导过程，首先，定义输入x的代价函数为，

(2.1)

其中w，b代表网络模型参数，y(x)为输入x对应的真实输出值，a为网络模型的最终输出值。对于神经网络模型最终层L层而言，a为L层网络的非线性映射输出，z表示神经元在非线性变换前的输出值，激活函数为，L层网络又是通过L-1层网络进行权值连接的方式得到的，由此可以得到以下表达式

(2.2)  
 (2.3)

因此，对于输出层L，其神经元结构如图2.1所示，

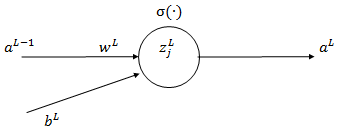


图2.1 神经元结构连接

每一个神经元的误差可以通过代价函数对每一个神经元求偏导数得到，用表示L层第j个神经元上的误差，即

(2.4)

其中，表示代价函数关于关于的梯度，代表两个矩阵逐元素乘积（Hadamard 乘积）。得到最终L层的误差值之后，可以根据反向传播的思路，依次得到前层误差值，即任一层的误差值都可以由后一层的误差值计算得到，即

(2.5)

其中根据神经网络结构可以得到，

(2.6)

因此，有

(2.7)

将式（2.7）带入到式（2.5）可以得到任一层的误差计算公式，

(2.8)

通过上式可以看到，在计算网络误差时存在对激活函数求导的过程，且随着网络隐含层数的不断增加，激活函数的导数对误差值的影响将变得越来越大，也就是说，

当时，误差经过反向传递的过程将会变得越来越大。这时候将会产生梯度爆炸，导致网络模型训练失败，

而当时，误差将会变得越来越小导致对前层网络参数的影响甚微，不能够帮助前层参数进行迭代更新，

尤其是当时，就出现了梯度消失的现象，也因此丧失了网络迭代更新进行模型训练的能力。

因此，在进行深度网络模型激活函数选择的时候，需要考虑梯度消失、梯度爆炸等因素，合理选择隐含单元的非线性映射激活函数。在本文中，一般采用修正线性单元(Rectified Linear Unit，ReLU)作为隐含层的激活函数，

(2.9)

修正线性单元易于优化，当其处于激活状态时，其导数保持一致并且保持为1，在对误差进行传递进行梯度学习时具有较好的效果，这在配合随机梯度下降算法时得到了证明，在设置恰当的学习率后，其可以帮助随机梯度下降算法实现快速收敛。

2.2.2梯度下降算法

梯度下降算法是机器学习中应用较为广泛的优化算法，是众多机器学习算法中最节用的优化方法。当前先进的机器学习库或者深度学习库都会包含梯度下降算法的不同变种实现。接下来就通过数学推导来介绍基于梯度下降算法的模型参数更新方式。

对于含有多个样本的数据集，机器学习算法中的代价函数通常可以分解成每个样本代价函数的和，即

(2.10)

当权值w进行参数更新时，需要计算代价函数对权值的导数得到其梯度值。首先对最终层L层的权值进行求解，对于最终L层有，则可以计算最终层L层对任一权值的偏导数得到梯度值，

(2.11)

同时，由上一节已经得到了误差的表达方式，即，即权值参数的梯度值可以表示为，

（2.12）

同理可以得到偏差的梯度值

(2.13)

得到最终层的梯度之后。可以根据反向传播的思路对前层的网络参数进行求其梯度值，得到相应的计算公式。在这里，对于任一层对权值参数的梯度求解，有

(2.14)

其中，又有

(2.15)

由式（2.6）可以得到

(2.16)

整理式（2.15）可得

（2.17）

将式（2.17）代入到式（2.14）中，并结合上一章对误差的表达方式，可以得到任一层对权值的梯度值，即

(2.18)

同理可以得到对偏导数的任一层梯度值，

(2.19)

以上就可以得到对于模型第层参数的梯度值，即

(2.20)

有了以上梯度值的计算方法，基于梯度下降的方式可以得到第层相应的送代参数更新过程，即

(2.21)

其中，为参数更新过程的学习率，它控制着梯度更新的速度，过大的学习率会加快模型优化速率，但同时也有可能造成模型错过最优值而产生振荡的现象，过小的学习率虽然可以避免最优化的振荡过程，但其计算过程冗长往往增加了模型训练时间，导致最优化过程非常缓慢。

### 2.3深层神经网络的学习率优化策略

由上一节对深度神经网络训练算法的推导可以看出，对于算法学习过程，学习率的设置是模型成功的关键之一，学习率不同的选择策略会对模型最终效果产生显著的影响。在深度模型应用的过程中，许多学者也提出了一些不同的策略来设置学习率以求得到最优效果，其中包含采用动量的方法来实现恰当的学习率选择。

基于动量的方法是为了加速网络模型的训练学习，在参数更新阶段引入变量v，代表了参数在参数空间进行迭代更新的方向和速率。动量的概念借助于物理学中的定义，在动量学习算法中，假设需要更新的参数为物理学中的粒子，质量为单位质量，那么速度向量v则表示了需要更新的参数在寻求最优解路径时的速率和方向，即称为动量。设置学习率为，动量参数为，初始参数为，初始速度为v，那么在进行参数更新时，规则为

(2.22)

在这里，通过速度来积累梯度，动量参数表示速度衰减力度，当越大时表示之前梯度的积累对目前参数更新运动的影响越大。简而言之，若当前时刻的梯度与历史梯度方向相似，参数更新趋势则会被加强，如若当前时刻的梯度相较历史梯度方向有异，则参数更新将会被减弱。假设，当所有时刻的梯度方向均为相似时，可以得到更新步长为

(2.23)

直观地讲，将动量参数设置为0.5时，该模型的参数更新速率变为之前的2倍，当动量参数设置为099时，则参数更新的速度则提升到原来的100倍。和学习率相同，动量参数也是需要随着时间不断进行调整的，一般设置较小的初始值，并随着训练过程逐渐变大。

基于动量的方法在一定程度上加速了训练过程，但由于其引入了速度v这一超参数，增加了网络模型参数与训练复杂度，因此，一些自适应学习率的算法可以更好地取得模型优化。基于自适应学习率算法主要有Adagrad( Adaptive Gradient Algorithm)算法、RMSProp( Root Mean Square Prop)算法和Adam( Adaptive Moment Estimation)算法。

AdaGrad算法主要思想是通过缩放每个参数从而得到不同的更新速度，也就是说，对于具有较大偏导的参数可以得到快速下降的学习率，而对于偏导较小的参数则会得到相应较小的下降。主要过程为，首先设置全局学习率n以及初始梯度积累量r=0，在计算梯度△的同时，累计历史梯度平方值，

(2.24)

针对于每一个参数，得到相应的参数更新策略，即

(2.25)

在这里，需要每个元素地进行求和及除平方根操作，是一个为了进行数值稳定的小常数值，大约设置在左右。对于参数空间中较为平缓的参数更新路径，历史梯度平方和较小，对应的学习率衰减幅度小，参数更新过程会具有更好的效果。但该方法从训练的最初阶段就开始累计梯度平方值，会导致学习率衰减地过早，所以在实际应用过程中，会出现不够理想的表现。

在Ada Grad算法的基础上，改变梯度积累的方式，变为指数加权的移动平均就得到了RMSProp算法。通过指数衰减平均可以丢弃掉过早的梯度历史，使其能够更快的进行收敛，避免学习率衰减得过早或过量问题的发生。该方法的梯度积累方式为

(2.26)

而对于参数更新策略则与式(2.25)相同，该方法是在实际应用中也取得了不错的效果，是一种有效且实用的深度神经网络优化方法。

基于以上对于学习率的讨论研究，可以得到另一种学习率自适应的优化算法，其主要思想是带有动量的 RMSProp优化算法，被称为Adam最优化，它需要计算梯度的阶矩估计和二阶矩估计来动态地调整每个参数的学习率，其主要优点在于该算法包含了偏置修正，使得学习率始终处于较平稳的参数范围内，因此对于超参数的选择而言，基于Adam的最优化策略具有更好的稳定性。基于Adam的最优化算法中，参数更新过程如下，首先需要计算梯度的一阶矩估计s和二阶矩估计r，

(2.27)

其中，和矩估计的指数衰减速率，在一般情况下设置为0.9和0.99接下来对一阶矩和二阶矩偏差进行修正，即

(2.28)

其中，t为时间步数，初始化为0，由此可以得到每个元素的参数更新策略，

(2.29)

Adam最优化的方法结合了动量和RMSProp方法，在处理非平稳目标优化时具有较强的能力，对于不同的参数计算自适应的学习率，有助于深度网络模型较快收敛，适用于大数据集和高维空间的模型训练。在本文的模型建立和优化过程中，均采用Adam最优化方法，得到自适应的学习率用于模型训练学习，得到收敛较快的优化过程。

2.4小结

随着机器学习的发展和大数据的应用，深度学习技术在研究领域与应用领域受到了专家与学者们的青睐，深度学习技术是机器学习领域之中发展较为迅速的一个分支，其通过对经验和数据的学习来帮助网络模型优化，是实现人工智能的技术支持之深度模型可以分层地从输入信息中学习有利于特定任务的特征表达，这在计算机视觉、自然语言处理以及自然科学领域都得到了成功的应用。对于深度网络模型而言，最为重要的就是建立和优化网络模型，如何搭建含有多个隐含层的网络模型结构，如何利用训练集对模型进行充分地训练，以及如何最优化模型参数空间使其达到最优的效果，这些都是影响深度学习应用的关键因素。

在本章，介绍了深度网络模型的训练学习基本方法，包括误差反向传播方法、基于梯度的参数更新方法、以及迭代式更新方法，这些是深度网络模型最通用且最有效的训练方式，本章给出了数学推导过程，介绍了基于深度学习算法的参数更新方式，在本文的后续四章研究工作中，基于深度学习的故障诊断模型设计与模型训练优化过程，均是基于本章所述的参数更新方式进行的。此外，本章还介绍了关于选代式参数更新方式中自适应学习率的选择策略，旨在获得最快速且最有效的模型收敛，发挥深度模型的特征学习能力，在实际应用场景中能够得到最优效果。