原文链接：<https://blog.csdn.net/dujuancao11/article/details/105138648>

人工神经网络大类型：**前馈型神经网络、反馈型神经网络**

**前馈型神经网络：**单向多层结构，整个网络中无反馈。**常用于图像识别、检测、分割**

**反馈型神经网络：**具有反馈连接的神经网络，**当前的结果受到先前所有的结果的影响**。 **常用于语音、文本处理、问答系统等（语境）**

**一、输入输出、样本和标签等数据集**

数据集准备 （已有的数据集MNIST是一个著名的手写体数字识别数据集，训练集包含60,000 个样本和标签，测试集包含10,000 个样本和标签。）

**训练集：**类似于日常学习中使用到的各种学习资料（练习册），帮助我们提升学习能力；

**测试集：**类似于考试试卷，检测我们的学习成绩；

**标签：** 可以认为是习题和考试的标准答案。

**样本：**每个样本图像的输入都是一组784个数值，称为一个784 维向量 ，28\*28=784

**标签：**每个标签数据中，把数字n将表示成一个只有在第n维度数字为1的10维向量。例如，比如 ， 数字2 的标签将表示成： [0,0,1,0,0,0,0,0,0,0,0]

**输出：**输入的784维向量经过两层中间计算后，输出为一个10维的向量，每个维的值代表是该数字的概率。

**三、构建网络（输入层、隐藏层、输出层）**

隐藏层（隐藏层主要包括**卷积层、全连接层、池化层、归一化指数层、激活层**等）

通过多个顺序链接的隐藏层的组合，神经网络就可以将原始图像变换为高层次抽象的图像特征。

**卷积层：**卷积层的作用是**提取图像的二维特征**，**通过不同的算子可以检测图像不同边缘。**

**池化层：** 池化层**作用是减少训练参数**，是对原始特征信号进行采样。当输入数据过大时，卷积层的计算量就会很大，这时需要减少参数，因此池化层常出现在卷积层之后。

**全连接层：** 全连接层的**作用是将所有特征融合到一起**。通常用来**把计算得到的蕴含特征的大维度向量转换成与输出层向量的维度相同** 。

**激活层（阈值）**

电脑：非线性激活层就是**决定哪些神经元的活跃程度高，哪些神经元的活跃程度低**。

**归一化指数层**

〉 归一化指数层的作用就是**完成最后输出分类时每个类别概率的计算**。

〉 通过“归一化”函数，使得**最后输出的十维向量的每个值加起来总和为1**。

〉 每个单独的数值就可以理解为AI认为**图像是该类型的概率**。

**四、人工神经网络的训练与评估--------计算损失函数**

**模型训练**

搭建人工神经网络之后，进行训练。

〉 对于一个2层的小网络，就可能有一万三千多个参数需要我们进行计算，计算量大。

〉 网络的训练是通过一系列**带有标记的样本数据来进行的**，这种学习方式称为**监督学习。**

〉 神经网络训练的过程就是**最小化损失函数的过程**，通过不断优化网络参数，使模型的预测值更加接近真实值。

**模型评估**

〉 既然样本数据带有标签，我们就要把神经网络预测得到的结果与标签进行对比，从而判断预测的准确性。

〉 如果发现预测结果不够好，那么我们将调整网络的参数，使得网络能够做出更好的预测。

〉 如何评价预测结果好坏？通过损失函数来定量计算

〉 损失函数（loss function）又叫做代价函数（cost function），是用来评估模型的预测值与真实值的误差大小。

〉 神经网络训练的过程就是最小化损失函数的过程，通过不断优化网络参数，使模型的预测值更加接近真实值。

**损失函数类型**

**（1）0-1损失函数：**一种常用于分类任务的损失函数。

预测值与真实值不同 ， 就是预测错误 ， 则损失是1 ；

预测值与真实值相等 ， 就是预测正确 ，是 损失是 0 ，就是没有损失

**注意：**手写数字识别中不同的数字有不同的概率，不是非0即1，不适用于0-1损失函数

**（2）平方损失函数：** 预测值与真实值的差的平方和。**通过不断减小平方损失函数来提高分类的准确率**。

平方损失函数比0-1损失函数能够反映多的信息，在手写数字识别实例中更有利于参数调整。

**总结：**

（1）通过损失函数来定量计算神经网络预测结果的好坏。

（2）神经网络训练的过程就是最小化损失函数的过程

（3） 损失函数越小说明预测结果越接近真实结果（标签）

**五、根据评价结果优化参数----模型优化**

通过损失函数能够定量神经网络预测结果的好坏，那我们如何根据这个评价结果来更改优化我们的参数？

〉 神经网络中的参数是海量的，如果修改的方法不当，就会耗费大量的计算时间。

〉 在损失函数确定情况下，优化网络的任务就交给了优化器。 优化器代表了调整网络参数，使损失函数达到最小的过程。

〉 不同的优化器有不同的算法，改变损失函数的过程和效率也各不相同。

函数优化例子（类比下山）

〉 把损失函数想像成一个起起伏伏的山谷。

〉 假设小明现在处于图中最高点a，要去到最低点b，那该如何选择最优路径？

梯度--------表示函数在某一点变化最快的方向

梯度下降算法---------梯度下降法的基本思想可以类比为一个下山的过程。

〉 假设小明被困在山上，浓雾很大可见度低，下山的路径就无法确定，他必须利用自己周围的信息去找到下山的路径。他就可以利用梯度下降算法来帮助自己下山。以他当前的所处的位置为基，寻找这个位置最陡峭的地方，然后朝着山的高度下降的地方走。每次沿着当前位置下山一小步，这样就能保证越来越接近山谷，即得到更小的损失函数，直到到达b，得到损失函数最小值。

〉 如果不小心进入此低点c、d点，那么梯度下降的算法就无法再回到b，这时就需要其他优化算法

结：

〉 优化器代表了调整网络参数，使损失函数达到最小的过程。

〉 不同的优化算法的计算过程和效率也各不相同。

〉 梯度下降算法是最常用也是最有效的优化

**六、反向传播**

〉 通过多层神经网络得出的结果与实际标签的结果作比较，得到了损失函数。

〉 根据梯度下降的方法，将当前的损失函数反馈给之前各层的神经网络，并调整各层网络参数的权值，这个过程称为反向传播。

〉对于标签为2的数据，为了预测得到正确的值2，我们希望代表2的神经元的值提高，而把其他的神经元值降低。

〉 提高和降低的程度和它现有的值成比例，提高2的值就比降低8的值来的重要，所以要选用梯度。

神经元的参数

〉 一个神经元的输入输出可以类比为一个多元一次的线性方程。

〉 二元一次：𝑦 = 𝑤 1\* 𝑥 1 +𝑤 2 \*𝑥 2 + b

〉 可以通过调整w、x、b的值来调整y的值

〉 根据梯度大小不同，调整不同的参数对损失

函数影响不同，梯度值：3.6 调整它的影响大 梯度值：0.1 调整它的影响小

调整系数w

〉 调整输出y的一个方法是调整系数w。

〉 前一层的神经元激活值越大，表示越有可能“看到”某些与2相关的特征。

〉 调整激活值大的神经元的参数，影响比调整激活值小的参数要大。

调整输入x

〉 调整输出y的另一个方法是调整输入x。

〉 增加原始参数w是正值的神经元的激活值，减少原始参数w是负值的神经元的激活值，增加和减少的程度与参数w大小相关。

〉 但我们无法直接改变每个神经元的激活值。将每个输出需要改变的大小叠加之后，能够得到总的改变量。

这样不断层层传播回去，最后就能知道该如何修改整个网络的参数。