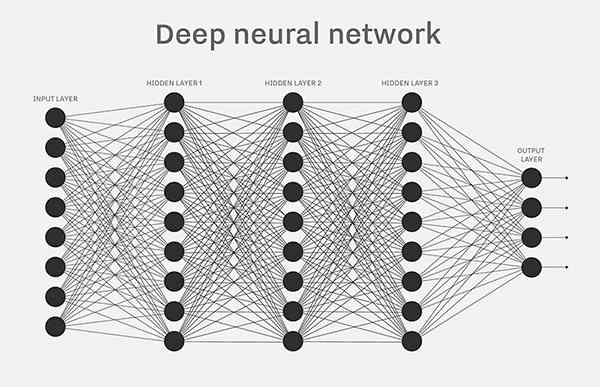
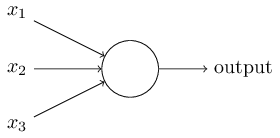
4.2 神经网络结构

先对神经网络的基本概念做一个简介：



神经网络，顾名思义是模拟人类神经元的算法镞，旨在通过大量的训练使得模型能过对新的刺激做出判断，以此模拟人类大脑的思考。其基本模型如下：



该模型称为感知器，x1，x2，x3代表不同的刺激（x1， x2， x3....），output为输出结果，模拟了人类神经末梢感受各种环境变化，最后产生电信号的过程。

在现实生活中，每个刺激的权重会有所不同，因此我们可以给x1，x2，x3赋不同的权值，这样output = Σxi \* pi，再通过给output一个阈值来判断当前刺激能否产生期望结果。

这就是神经网络的基本单元感知器了，当多个感知器相连，前一个感知器的输出当做了其他感知器的输入时，就构成了上图所示的复杂网络。

说完神经网络，再对训练做一个简单的解释：

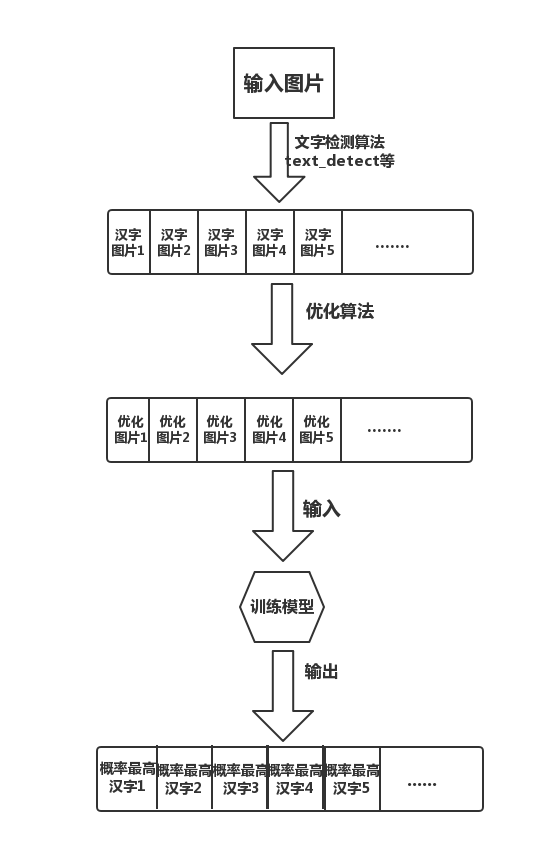
从上边的概念中可以看出，整个神经网络的重点就是确定权值pi和阈值，在现实生活中很难估计它们的值，但通过试错法，可以找到较为接近的答案。

所谓试错法，就是在其他参数都不变的情况下微调权值和阈值，然后观察输出的变化，通过不断重复这个过程，找出最精确的权值和阈值，这个过程就称为模型的训练。

在我们的ocr系统中，检测文字就是通过训练好的模型来识别的，首先我们定义了期望能够识别的文字集，将一些带文字的图片作为输入，输出就是识别出的文字，照片的清晰度等可以设置权重pi。然后将图像的对比算法作为感知器，算法的结果是一个概率值，如结果为75%的概率可以确定汉字是“理”。这样就需要我们确定一个阈值（如80%可信度），当算法得出的概率值低于阈值时，识别无效。高于阈值时，即可认为汉字识别成功。

通过训练集数据输入模型，不断地调整各种参数，找到正确率最大的参数组合，这样就完成了ocr系统的训练。

在ocr系统中,我们先处理出了每个文字的图片,对这些图片进行边缘去噪等优化后,作为输入传给训练模型,再通过训练模型的神经网络进行识别,选择概率最高的作为识别结果。处理的结构图如下：



而在训练中，最重要的算法是比较图片相似的算法，关于该算法，在算法部分会详细介绍。

在训练中,我们基于tensorflow/keras框架,用了八千余张图片作为训练集,经过训练后的模型效果良好,但受限于硬件条件(没有使用GPU运算,使用了传统的CPU)和时间,没有采用更大的训练集进一步进行优化。

参考：

<http://www.ruanyifeng.com/blog/2017/07/neural-network.html>

<https://keras-cn.readthedocs.io/en/latest/>