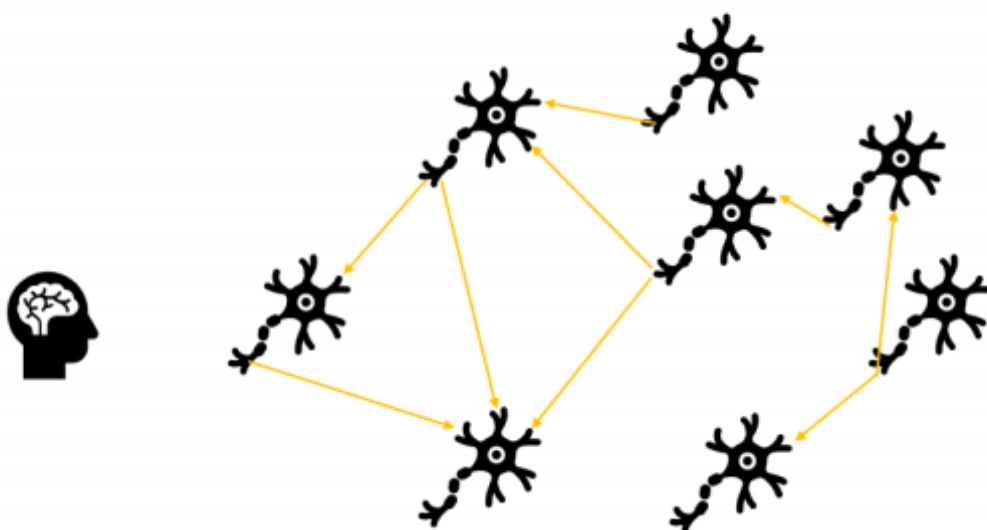


自从 2016 年 Alpha Go 打败了李世石，神经网络和深度学习就已经进入了广大人民群众视野，这个方向已经变得越来越火热，各路专家层出不穷地改进算法以及预训练模型，让人眼花缭乱。但是实际上人工神经网络算法早在几十年前就已经有了，只不过随着算力的进步以及科学家们不懈的改良，现在有了更加优秀的效果和广阔的应用，那么今天我就来介绍一下人工神经网络的基础算法。先来看一个例子。

一个例子

人的大脑是一个很好的记忆、逻辑、运算、推理的设备，其实人工智能就是在不断地模拟大脑的判断逻辑，那么能不能构建一个算法来完全模拟生物学上大脑的运行方式以实现大脑的功能呢？伴随着这个思路，我们先来看看人的大脑是怎么构成和工作的。



人在刚出生的时候有 2000 亿的脑细胞，虽然脑细胞不会再生，而且在生命的过程中一直在死亡，但是活到 100 岁仍然会有 27 亿的脑细胞。在我们的脑细胞中，起到记忆和运算的功能单位是神经元，神经元包含了轴突和树突，树突负责接收信号、轴突负责发送信号。在不同的神经元之间依靠突触进行连接，一个神经元可能会与很多神经元连接，这样形成了错综复杂的关联关系，最终构成了我们的神经系统。

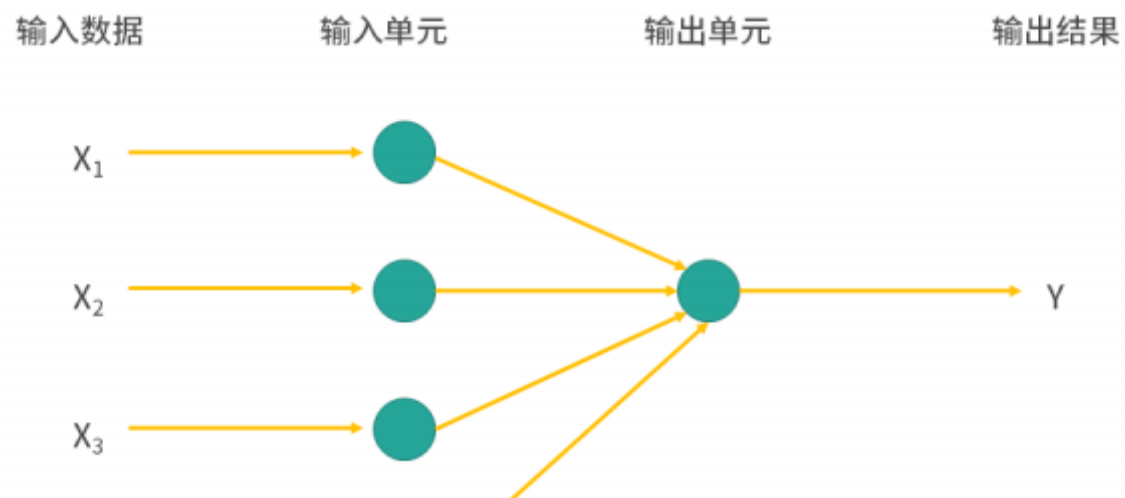
神经元可以发送电脉冲信号，如果信号足够强，会跨过突触间隙，到达另一个神经细胞，就可以激活神经细胞释放“化学物质”；如果信号强度不够强，则不会释放化学物质；同时多个输入信号可以进行叠加从而达到激活强度。不仅如此，神经元还可以形成新的连接，甚至改变连接。

算法原理

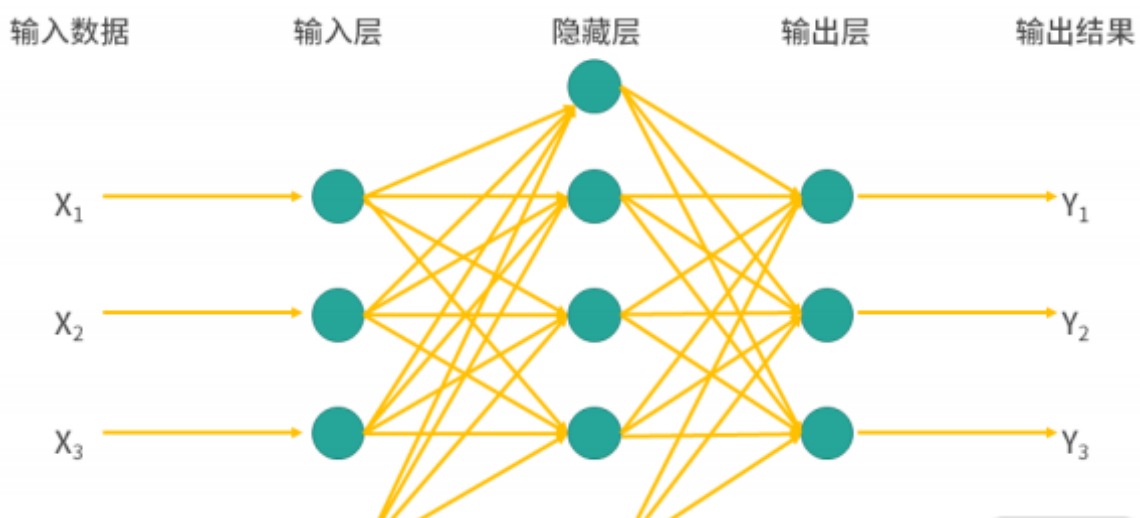
我们先考虑一个最简单的神经系统，这个神经系统里面总共有两层神经元：一层输入单元和一层输出单元。我们现在要根据已知数据去预测一个结果，这个结果符合方程：

$$Y = X_1 + 2X_2 + 3X_3 + 4$$

假设一个输入神经元处理一个特征，那么我们需要 4 个输入神经元、1 个输出神经元输出结果，由此可知，我们构建的这个模型就会具备下图所示的结构，这个模型的目的就是去寻找从输入单元到输出单元这条线上的权重，来根据数据拟合结果，这就是一个最简单的人工神经网络模型 ANN (Artificial Neural Network) 。



你会想，天呐，这不是在“坑”我吗？这不就是一个线性回归模型吗？是的，这个确实跟线性回归模型没有太大的区别。但是不要着急，我们再来看下面这张图：



这样是不是有点“网络”的意思了？我们在输入层和输出层之间加入了一个隐藏层，那么这个模型就可以处理非线性关系了。所以可想而知，根据不同的问题，可以加入多个隐藏层，由图中这种全连接改为部分连接，甚至是环形连接等。

所以，神经网络算法处理过程如下。

第一步，我们要**预先设定一种网络结构和激活函数**，这一步其实很困难，因为网络结构可以无限拓展，要知道什么样的结构才符合我们的问题需要做大量的试验。

第二步，**初始化模型中的权重**。模型中的每一个连接都会有一个权重，在初始化的时候可以都随机给予一个值。

第三步，**就是根据输入数据和权重来预测结果**。由于最开始的参数都是随机设置的，所以获得的结果肯定与真实的结果差距比较大，所以在这里要计算一个误差，误差反映了预测结果和真实结果的差距有多大。

最后一步，**模型要调节权重**。这里我们可以参与的就是需要设置一个“**学习率**”，这个学习率是针对误差的，每次获得误差后，连接上的权重都会按照误差的这个比率来进行调整，从而期望在下次计算时获得一个较小的误差。经过若干次循环这个过程，我们可以选择达到一个比较低的损失值的时候停止并输出模型，也可以选择一个确定的循环轮次来结束。

人工神经网络算法的大致流程就如上面介绍的这样，只要按照步骤去进行搭建就可以完成一个网络模型的构建。

关于激活函数

上面我提到了一个激活函数，这个词可以参考前面神经元的故事加以理解。**神经元的激活需要电信号累加到一定的水平**，这个神经元才会激活，所以我们在模拟神经网络的时候也需要有这样一个方法，在上层节点的输出和下层节点的输入中间加入一个激活函数，来实现这个功能。

如果没有激活函数，那不管有多少层网络，神经元之间也仍然是线性关系，就像没有隐藏层的时候一样，加入了非线性函数作为激活函数，这样深层次的网络就可以去拟合任意类型的函数了。常见的激活函数有 ReLU、tanh、Sigmoid 等，在不同的场景下可能需要不同的激活函数。

了解完了人工神经网络模型的基本原理，下面我们来了解了一下它的优缺点。

算法优缺点

优点

经过上面的介绍，我不知道你是否了解到了关于神经网络的一个巨大的优点，那就是**可以像搭积木一样不断地扩展模型的边界**，而对于内部具体的运行不需要加以太多的干涉。通过不同的搭建手段，神经网络几乎可以去模拟任何算法的结果，只要数据量够多，构建的模型够完善，最终都会有一个很好的结果。

缺点

然而神经网络的优点反过来想，也变成了它的缺点。首先神经网络缺乏可解释性，它的内部纷繁复杂，就像一个神奇的黑匣子，你告诉它数据，然后它告诉你结果，至于为什么会这样，它不做任何解释。所以在很多对解释性要求比较高的场景，比如信用评级、金融风控等情况下没办法使用。

其次，神经网络非常消耗资源，不管是数据、网络节点，还是硬件设备，要构建一套完美的神经网络模型开销是非常大的，不光训练时间长，还需要耗费很大的人力物力。

到这里，你应该已经对人工神经网络有了一个比较全面的认知，接下来进入我们的动手环节，用代码去实际感受神经网络的魅力。

尝试动手

在代码中，前半部分的数据处理环节仍然是使用的鸮尾花数据，与之前的处理没有任何区别。

```
from sklearn import datasets

import numpy as np

from sklearn.neural_network import MLPClassifier

np.random.seed(0)
```

```
iris=datasets.load_iris()

iris_x=iris.data

iris_y=iris.target

indices = np.random.permutation(len(iris_x))

iris_x_train = iris_x[indices[:-10]]

iris_y_train = iris_y[indices[:-10]]

iris_x_test  = iris_x[indices[-10:]]

iris_y_test  = iris_y[indices[-10:]]
```

后半部分，需要注意的是我们设置的参数，其中有一个比较重要的是 `hidden_layer_sizes`，它用来设置隐藏层大小，长度就是隐藏层的数量。在第一次，我们设置的是 `[5,2]` 这个数组，这样我们的网络有两层隐藏层，第一层有 5 个神经元，第二层有 2 个神经元。

```
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,

                    hidden_layer_sizes=(5, 2), random_state=1)

clf.fit(iris_x_train,iris_y_train)

iris_y_predict = clf.predict(iris_x_test)

score=clf.score(iris_x_test,iris_y_test,sample_weight=None)

print('iris_y_predict = ')

print(iris_y_predict)

print('iris_y_test = ')

print(iris_y_test)

print('Accuracy:',score)
```

```
print('layers nums :',clf.n_layers_)
```

再看结果，这里需要注意一下，在第一次输出的时候，可以看到准确率只有 20%，是不是让人大跌眼镜？不要担心，神经网络的重点就在这里，我们接着往下看。

```
iris_y_predict =  
  
[2 2 2 2 2 2 2 2 2]  
  
iris_y_test =  
  
[1 1 1 0 0 0 2 1 2 0]  
  
Accuracy: 0.2  
  
layers nums : 4
```

在第二次输出的时候，对于代码其他部分没有做任何调整，只是对 hidden_layer_sizes 进行了修改，改成了三个隐藏层，每个隐藏层有 10 个神经元，这个时候神奇的事情发生了，我们的准确率已经提升到了 90%。

```
iris_y_predict =  
  
[1 2 1 0 0 0 2 1 2 0]  
  
iris_y_test =  
  
[1 1 1 0 0 0 2 1 2 0]  
  
Accuracy: 0.9  
  
layers nums : 5
```

到这里相信你对人工神经网络已经有了认识，并且学会了使用 sklearn 搭建一个简单的神经网络，在实际工作中可以应用这种算法去处理自己的工作。当然，上面的代码我们可以看到一个现象，在我们的网络规模增长了之后，效果就变得更好了，所以，人工神经网络的发展也展现出了“大力出奇迹”的现象。

扩展内容：大力出奇迹

近些年，人工神经网络重新大放异彩，并且有了一个新的名字——**深度学习**。研究深度神经网络的人越来越多，所以也构建出了很多优秀的模型，尤其是在图像处理和自然语言处理方面产生了巨大的成果。

在图像处理方面，原来有一个著名的比赛就是 ImageNet 大规模视觉识别挑战赛 ILSVRC，除了模型本身的结构调整外，随着模型的隐藏层数量不断扩充，所用的参数量不断增加，预测的效果也越来越好，以至于在 2017 年图像识别的准确率已经达到了 97%，超过了人类对图像的识别准确率，这个比赛也就此停止了。

在自然语言处理方面，最新的由 Open AI 推出的 GPT-3 模型，最大模式使用了 45TB 的数据量、1750 亿的参数，这简直是一个爆炸级的数量。当然，要训练一个这样的模型所付出的算力也必然不菲。

可以看出，神经网络在大数据量，大规模网络的道路上一骑绝尘，虽然效果上有了长足的发展，但是也让很多个人和小公司望而却步。

总结

这节课又到了跟大家说再见的时候，在本小节的内容中，我们先认识了人的神经元，并了解了神经元的工作过程，从而引入到人工神经网络的构建上来。在介绍了人工神经网络的原理和优缺点之后，我在代码中展示了不同规模的神经网络对预测效果产生的影响。当然，现在火热的深度学习可不止这么一点内容，很多开源的框架不断涌现出来，比如 TensorFlow、PyTorch、PaddlePaddle、MXNet 等都是非常好用的深度学习框架，为我们构建和使用神经网络创造了极大的便利，在实际的工作中，通常也会选择这些框架来进行模型的构建和训练。如果你对深度学习有更多的兴趣，那可以在这些框架的官网找到更多的资料进行学习。

