BP神经网络

胡家田

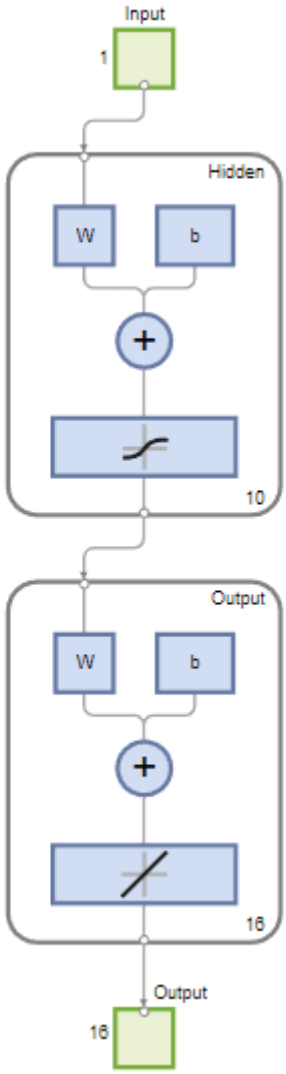
BP神经网络简单理解

函数拟合是在一组输入上训练神经网络以产生一组相关的目标输出的过程。一旦神经网络对数据进行了拟合，它就形成了输入-输出关系的泛化，并可用于生成未经训练的输入的输出。

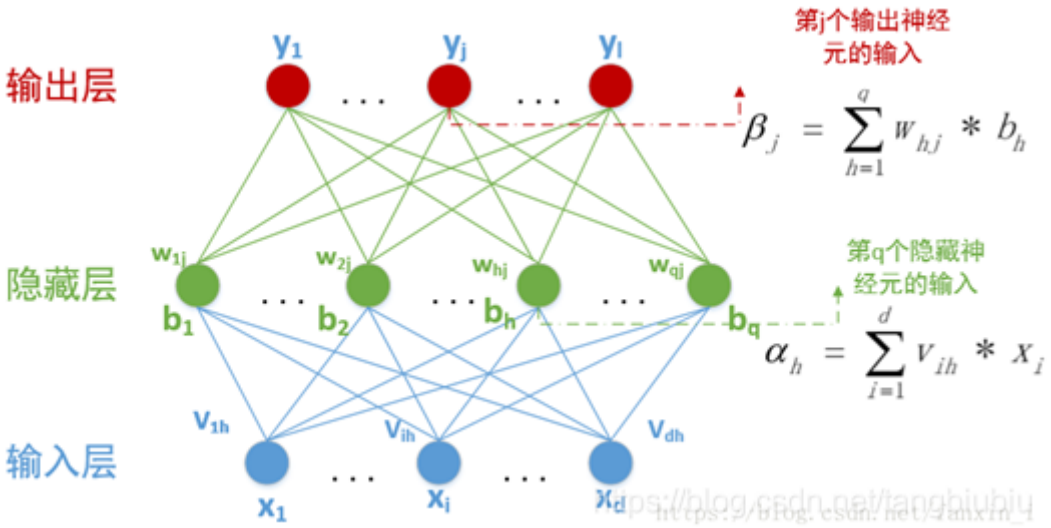
$$f\ (x_1,\ x_2 \dots x_n)\ =\ (y_1,\ y_2 \dots y_m)$$

BP神经网络做的事情就是在给出的$(x_1 \sim x_n, y_1 \sim y_m)$特征的观测值数集上进行训练，给出其关系的泛化f，并用其进行对应数据集的预测工作。

其关键在于泛化关系f的生成方式：

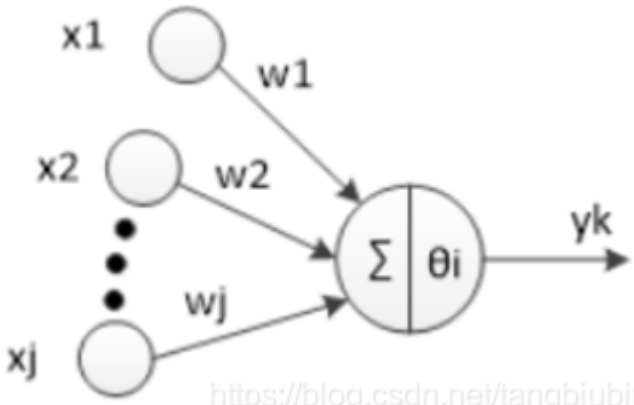


- BP神经网络的拓扑结构：



分为三层每层都有若干神经元，上一层的输出作为下一层的输入。

- 单个神经元：



x i代表输入w i代表权重，θ i 代表阈值。Y k代表输出。

- 激活函数：

满足：yk=f（neti+b）

-neti为加权和（大于阈值则激活，继续运算，小于则没有输出）

-b为偏置值（给予偏置补偿，防止欠拟合）

激活函数给神经元引入了非线性因素，使得神经网络可以任意逼近任何非线性函数，

形成了原始的感知机。

- 损失函数：

定义预测值与真实值的差距，用损失函数（loss function）来衡量。

解决分类问题和回归问题各细分类分别使用不同的损失函数。

（对于分类问题，我认为可以使输出的范围划分为几个类来实现）

- 反向传播（即BP）：

对损失函数利用梯度下降法，调整w（我认为也有对偏置值的调整）使损失函数沿着越

来越小的方向下降。每调整一次即完成一次循环，然后进行迭代，直至达到收敛条件

跳出循环。（许多参数在工具箱里提前设定好了一定的数据集）

- 泛化关系生成：

对所有参数进行整合，形成泛化关系。

Matlab集成BP神经网络

BP神经网络作为一种经典的神经网络，现在已经形成了集成模块。

输入和输出：

- 必须为double类型

  将table转化为double：table2array( )

- 输入输出为特征*观测值的double数组，输入和输出的观测值必须一样（例：对506个社区的13个参数进行研究，而后输出该社区房价的均值。）

参数数据：

- 训练数据：70
- 验证数据：15
- 测试数据：15
- 隐含神经元数量：10

输出结果：

- 回归图：显示了网络输出对培训、验证和测试集的目标。为了完美的配合,数据应该沿着45度的线下降,网络输出等于目标。

- 性能图：均值平方差来确定误差，并给出最佳训练的轮数。

- 训练状态图：

-gradient：梯度

-mu：控制权重的更新速度

-gamk：：参数数目

-ssX：参数平方和

-valfail：验证检验

- 误差直方图：代表网格训练结果，与样本量、训练次数等都有关。（R为相关系数、MSE代表方差）

- 泛化关系及其拟合效果图（预测变量仅有一个特征）

结果调用：

- 定义对应矩阵
- Y=sim(net,X)

- 例：

```
% 这里要注意，我们要将指标变为列向量，然后再用sim函数预测
sim(net, new_X(1,:)')

% 写一个循环，预测接下来的十个样本的辛烷值
predict_y = zeros(10,1); % 初始化predict_y
for i = 1: 10
   result = sim(net, new_X(i,:)');
  predict_y(i) = result;
end
disp('预测值为：')
disp(predict_y)
```