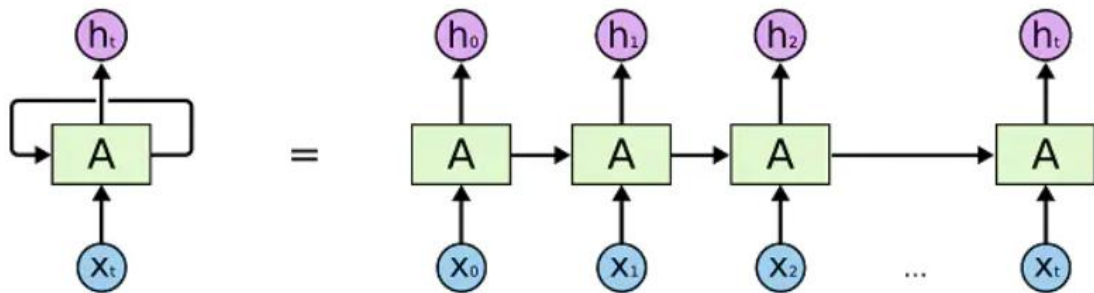


LSTM 长短时记忆神经网络（20min）

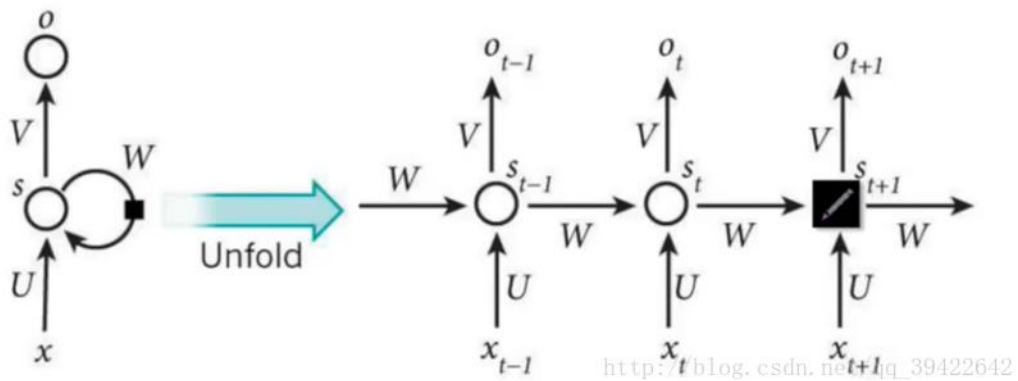
一 模型理解

循环神经网络结构图



网络特点：MLP—多层神经网络，可以实现降维操作，但是循环神经网络存在梯度为零或者梯度下降等问题，对于长时依赖问题不能够很好地解决（从结构也能看出，只能短时记忆），所以 LSTM 应运而生。

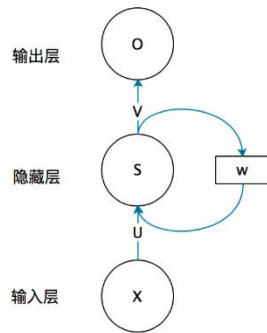
对循环神经网络的理解



循环神经网络的结构图

循环神经网络的定义是 在每一个时刻的输入都是多个 输入包括之前时刻的输入以及当前时刻的输入。

通过一个 **sigmoid** 函数的取值实现对之前消息部分保留量的判断，但是对于较长的字符串来说，也就是很久远的时间点来看，循环神经网络不太合适。



左面是一个简单的循环神经网络的结构图

W 的含义：循环神经网络 隐藏层的值不仅仅取决于输入层，还取决于上一次隐藏层的数值，**w** 是一个权重矩阵，**u** 也是一个权重矩阵这两个权重矩阵共同影响隐藏层的数值。

在训练的过程中，实际上就是矫正误差值的过程，通过对误差值的矫正能够不断提高模型的精度。

首先 **反向传播算法**的关键就在于误差项。通过不断矫正误差项来实现修改权重矩阵，最终实现对模型的改进，实现提高模型精度的目的。

反向传播算法的定义：利用链式求导计算损失值 **f** 对每个权重的偏导数，根据梯度下降算法更新权重。更新权重的过程就是模型训练的过程。

卷积神经网络分三个部分 卷积层 池化层 全连接层

全连接层是最简单的，就是一个由全连接的神经元组成的层结构

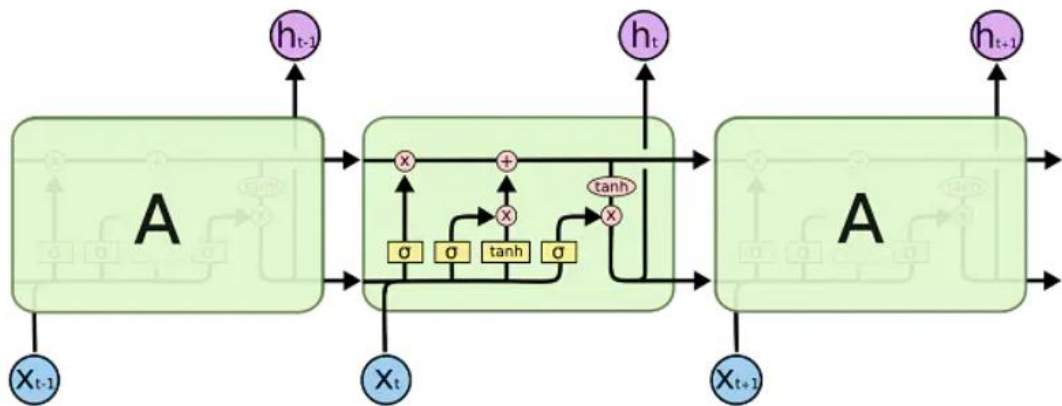
卷积层的实际操作是矩阵运算，通过矩阵的卷积操作实现降维操作 或者说降低图像的复杂度，然后是卷积层的深度要和输入图像的深度相对应，池化层的操作何卷积层大同小异，只不过池化层是 **mean** 或者 **max** 这样的操作在卷积层处理的基础之上实现再一次的数据降维此操作。

长短时记忆神经网络是循环神经网络的进化版

长短时记忆神经网络的记忆结构中存在三个部分，由于循环神经网络只能关联距离自己比较近的之前的数值或者说数据，所以 **LSTM** 神经网络改善了这一问题

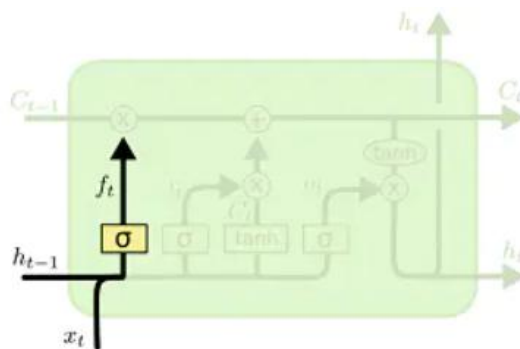
使得之前的输出或者之前的数据无论多么遥远都能在这一次的训练过程中保存下来。

LSTM 神经网络基础结构



LSTM 网络的结构：遗忘层+输入层+输出层

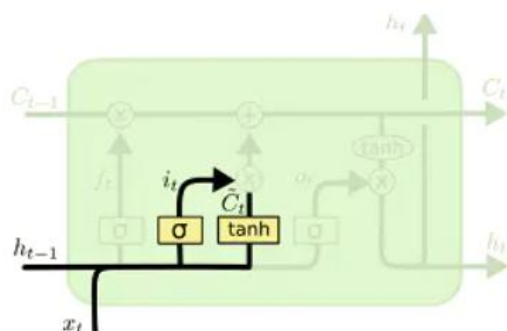
遗忘层结构



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

从上面结构能很好地看到，遗忘层的输出，是由输入层 x_t 和上一个时间点的输出层 h_{t-1} 共同组成的（矩阵向量运算）， α 实际上是 sigmoid 函数参与的运算，确定矩阵中需要保留的信息（1）和需要去除的信息（0），由于 sigmoid 函数的取值范围在 0-1 之间，所以利用该函数能够很好地决定需要保留的信息多少。

输入层结构



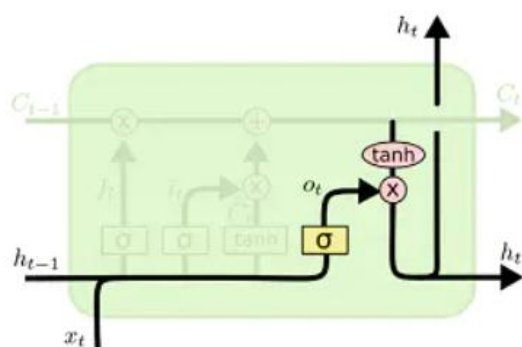
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

输入层采用了两个中间函数，sigmoid 函数和 tanh 函数，其大致原理同遗忘层类似，通过这两个函数来决定有多少信息可以输入到网络中，bi 与 bc 是偏置项，偏置项的学习也是通过反向传播算法来获得，由于原理类似，在此处不再赘述。

将两个输出进行积操作获得输入部分的信息。

输出层结构



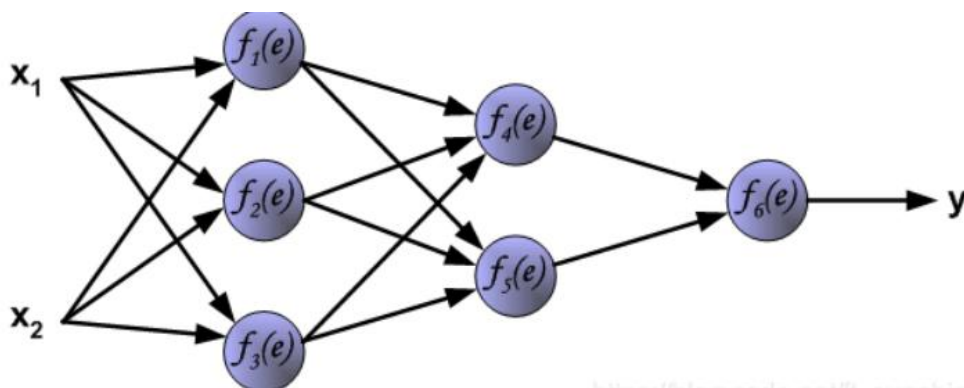
$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

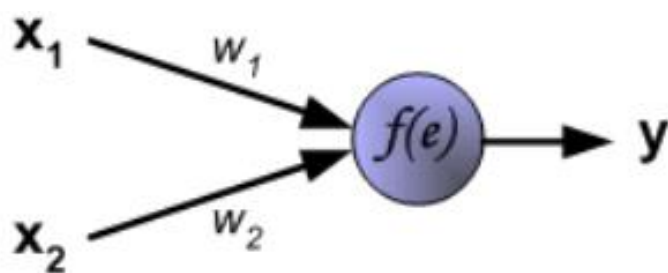
$$h_t = o_t * \tanh(C_t)$$

输出层的最终结果受到多方面的影响，包括单元状态、输入层等因素，首先输入层所得到的信息经过 sigmoid 函数进行过滤（添加偏置项），然后与经历 tanh 函数的单元状态进行矩阵相乘，获得需要的输出层。

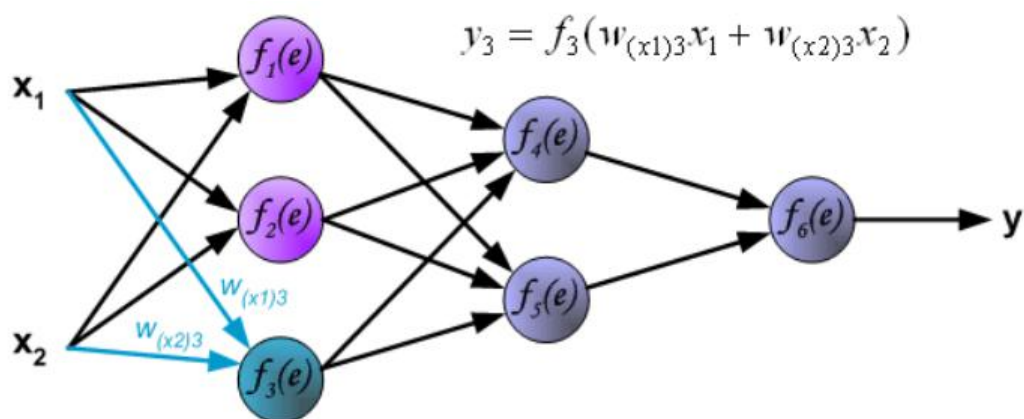
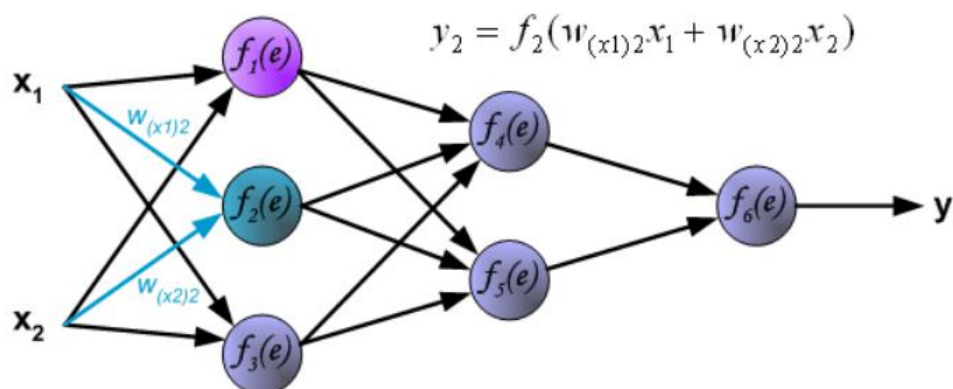
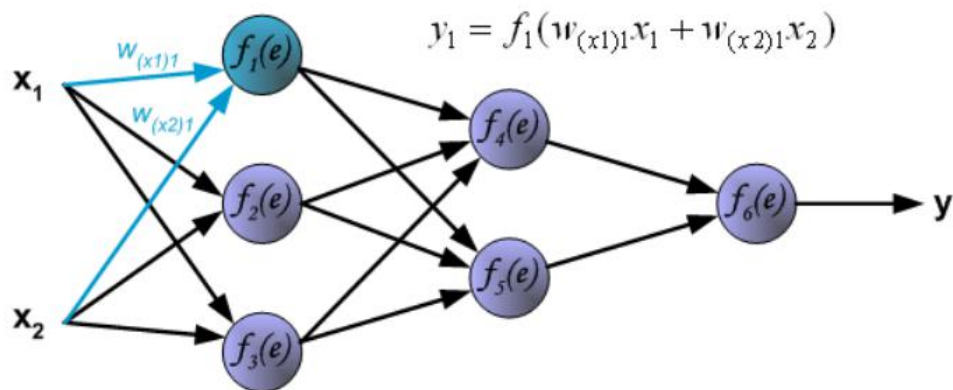
训练过程中最重要的算法（BP 算法——链式法则）

直观模型推导

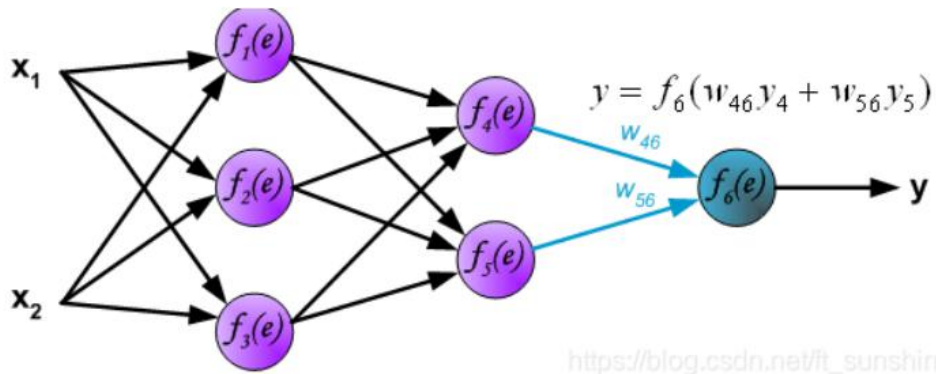




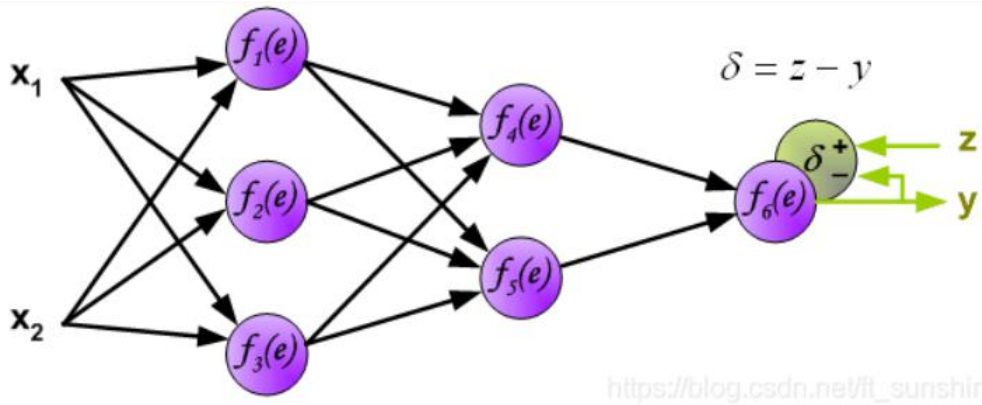
(纸上推导 1)



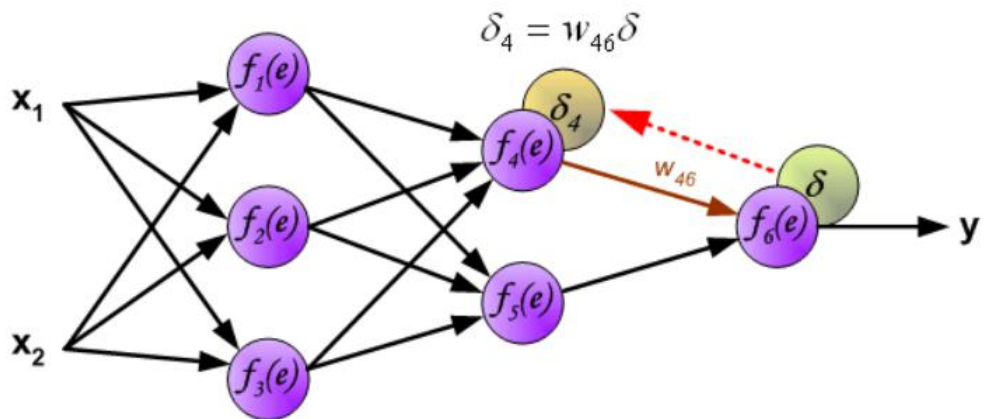
.....
.....
.....

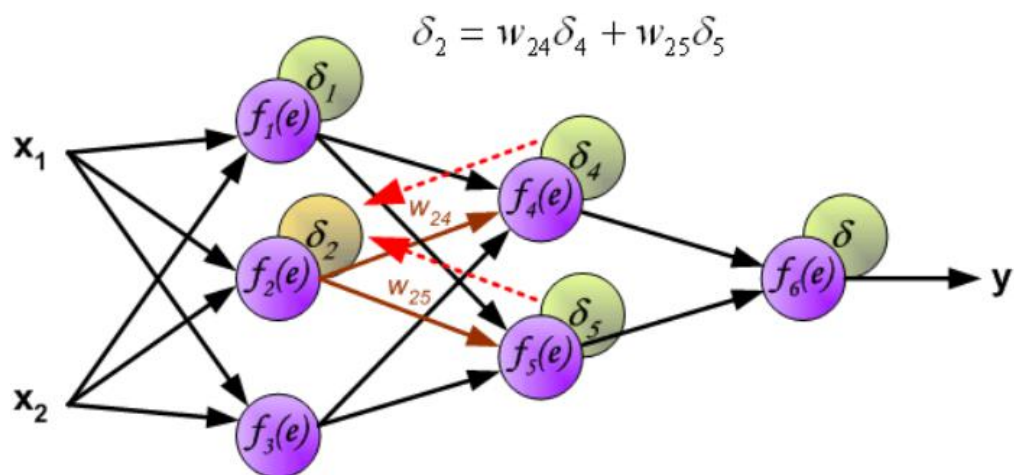
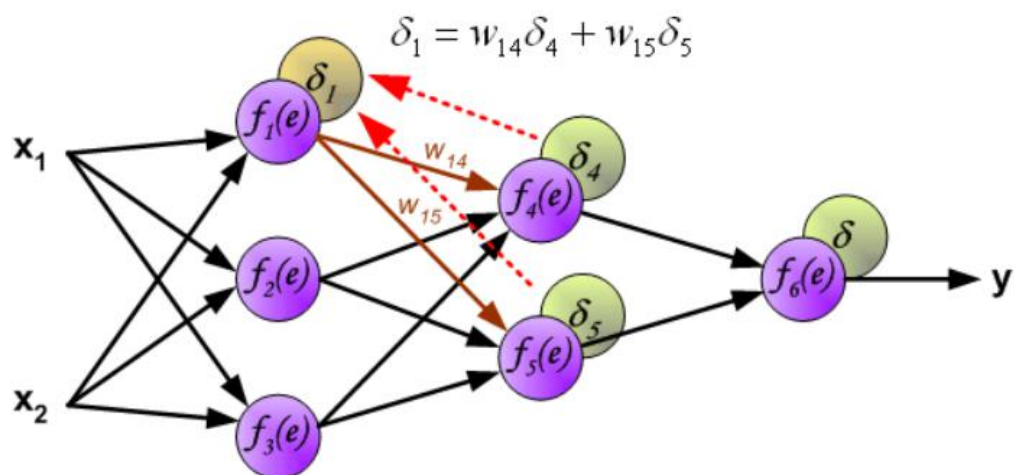
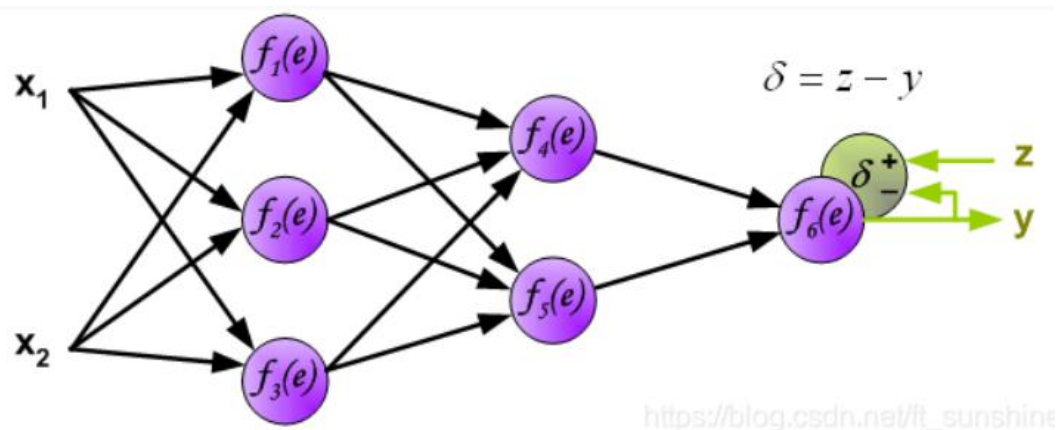


通过上面的前向传播获得一个输出值，输出值与理想值进行比较，判别，通过反向传播算法更新权重矩阵，这也就是神经网络的训练学习过程。

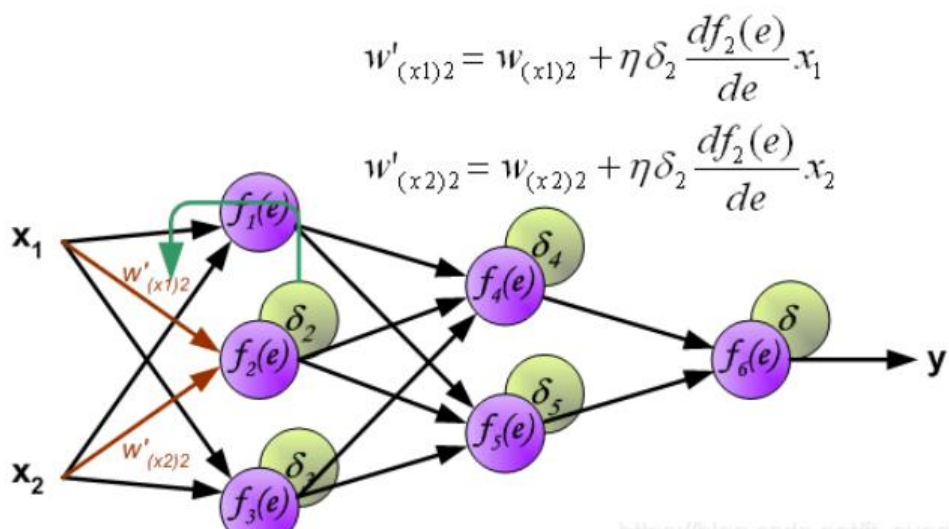
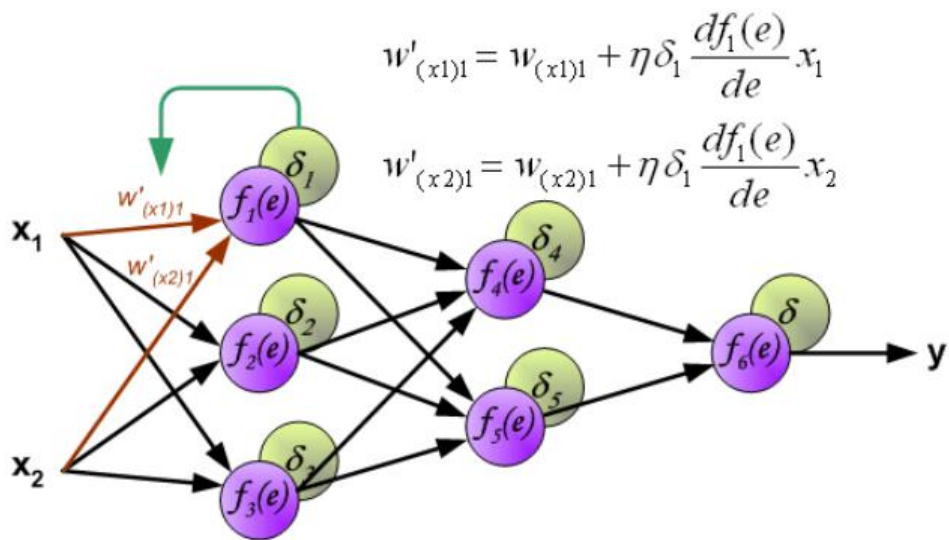
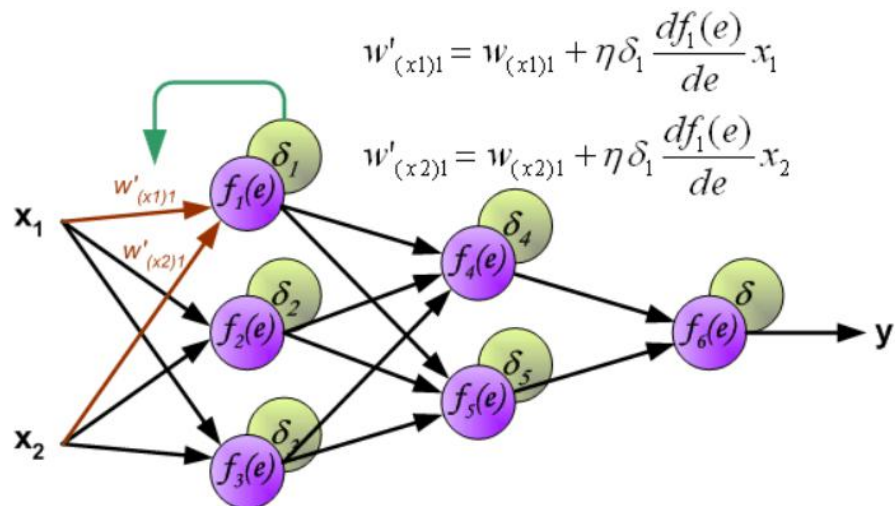


下面开始计算每个神经元的误差 (δ) :

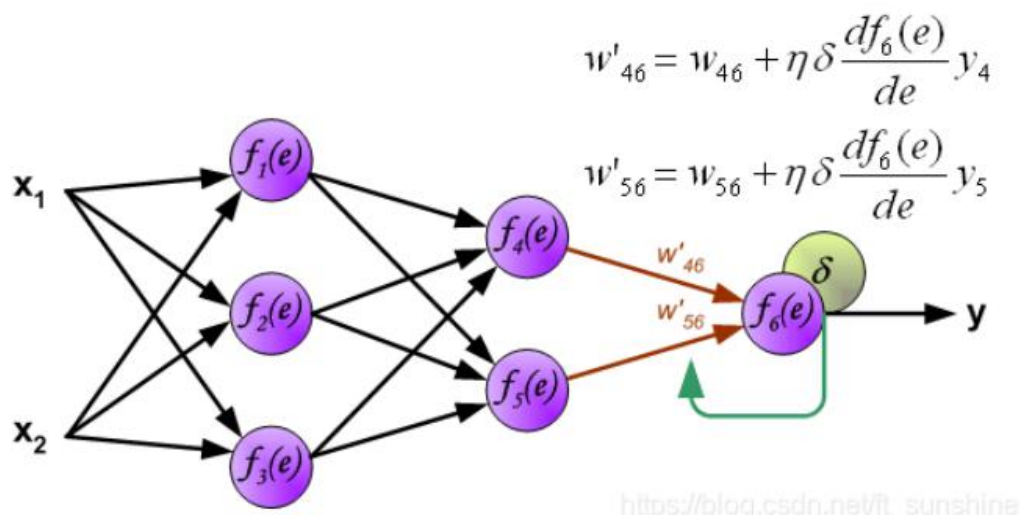




下面是修改权重的过程



.....



到此为止，整个网络的前向，反向传播和权重更新已经完成

“模拟退火算法”

“仿真示例”——能够运行，正在理解