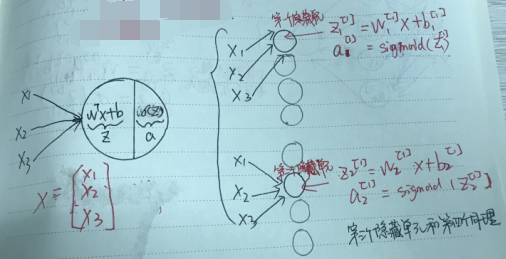
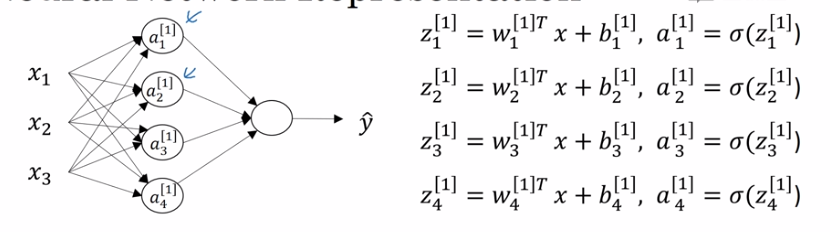
**2019.1.21总结**

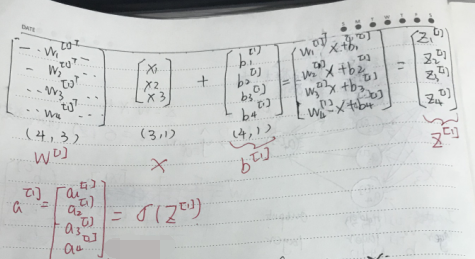
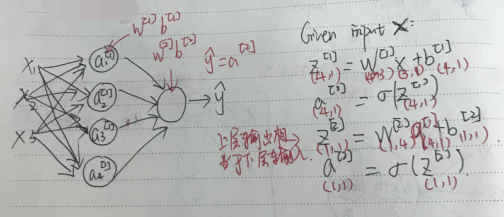
**第三周**

一个节点的计算





对应矩阵形式的运算

**m个训练样本**

for i=1 to m:#（双层NN训练m个样本）







X



X



X

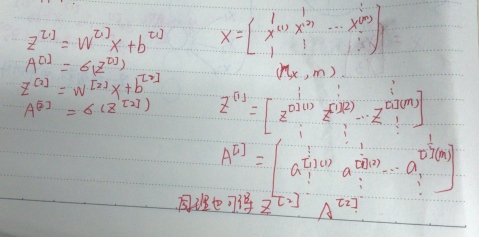


X

**...**

[2]表示第二层

(i)表示第i个样本



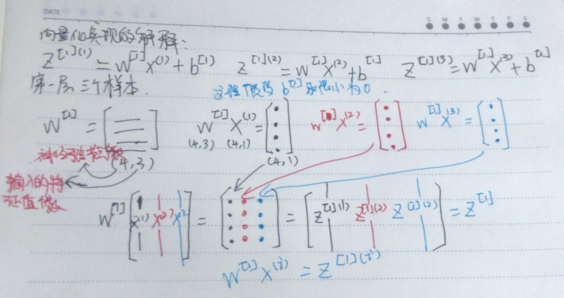
上述矩阵中从左往右横向看表示：表示不同样本

从上往下竖向看表示：神经网络中不同节点

横向矩阵A扫过不同的训练样本

竖向是矩阵A中的不同指标

**向量化实现解释：**



**Activation functions(激活函数)**

Sigmoid function 几乎已经不用了，tanh( )函数比sigmoid 函数好用，函数平均值更接近0。tanh()函数适合于任何场合，但是在输出层中不适合，因为如果y为1或者0，在0到1·之间而不是-1到1之间。（在二元分类时，输出层可用sigmoid函数）

Sigmoid 和tanh 函数都有一个缺点，如果z非常大或非常小，那么导数的梯度（函数的斜率）可能很小，很接近0，这样可能会拖慢梯度算法速度。

