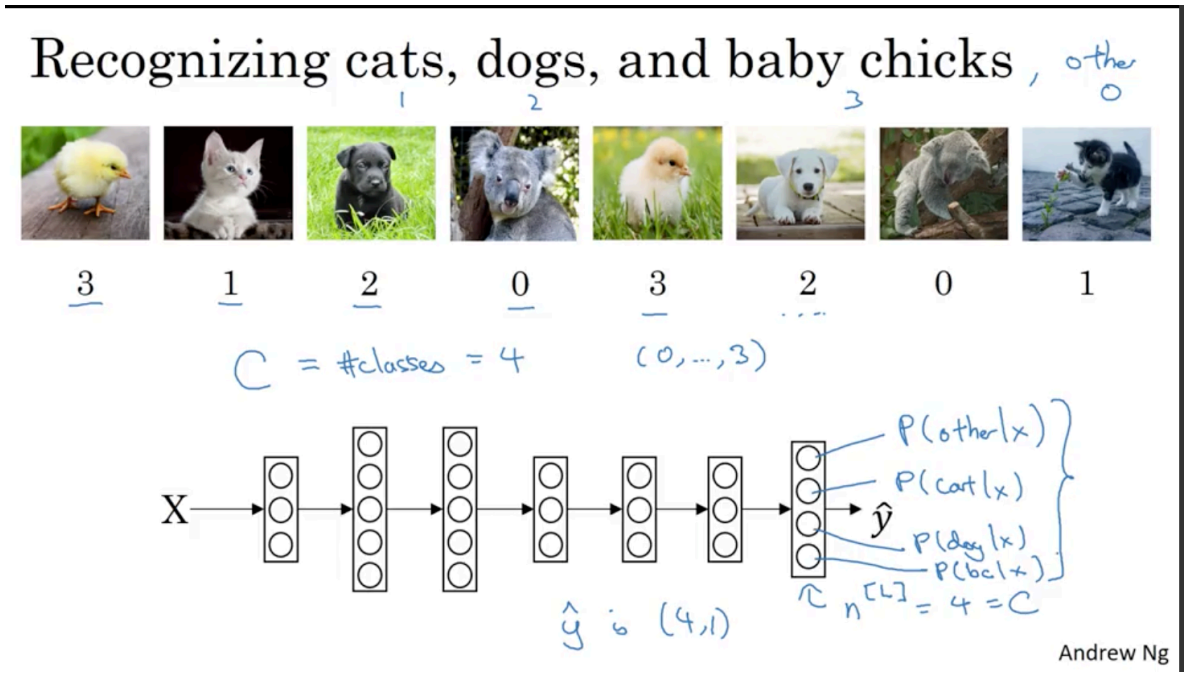
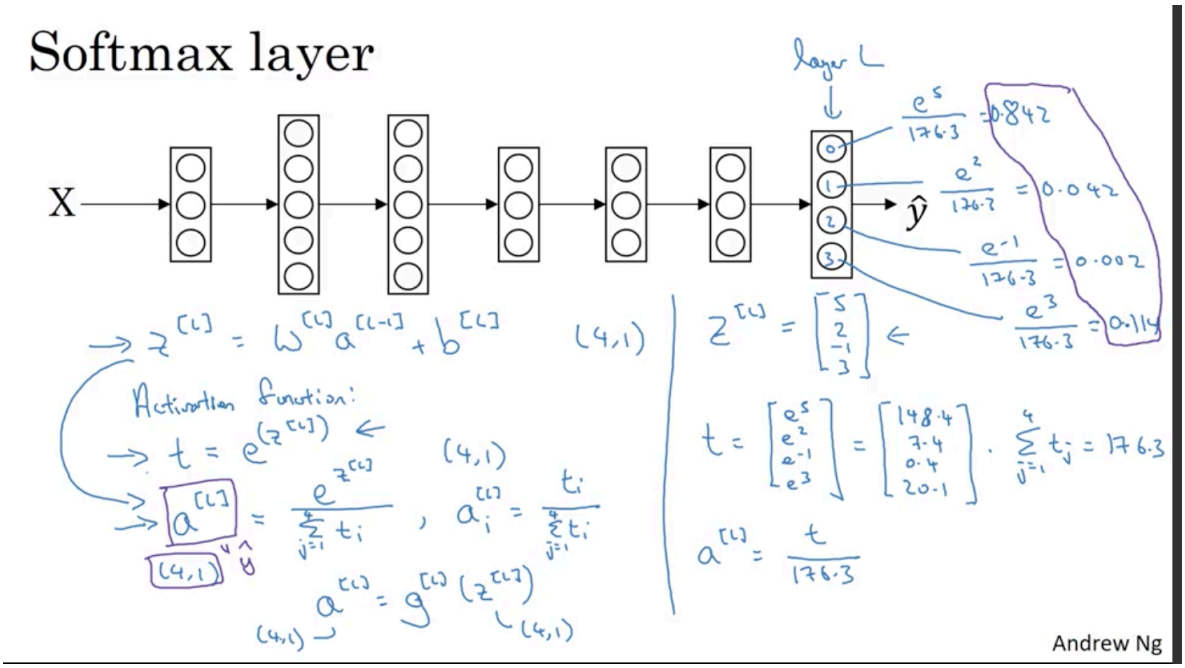
softmax



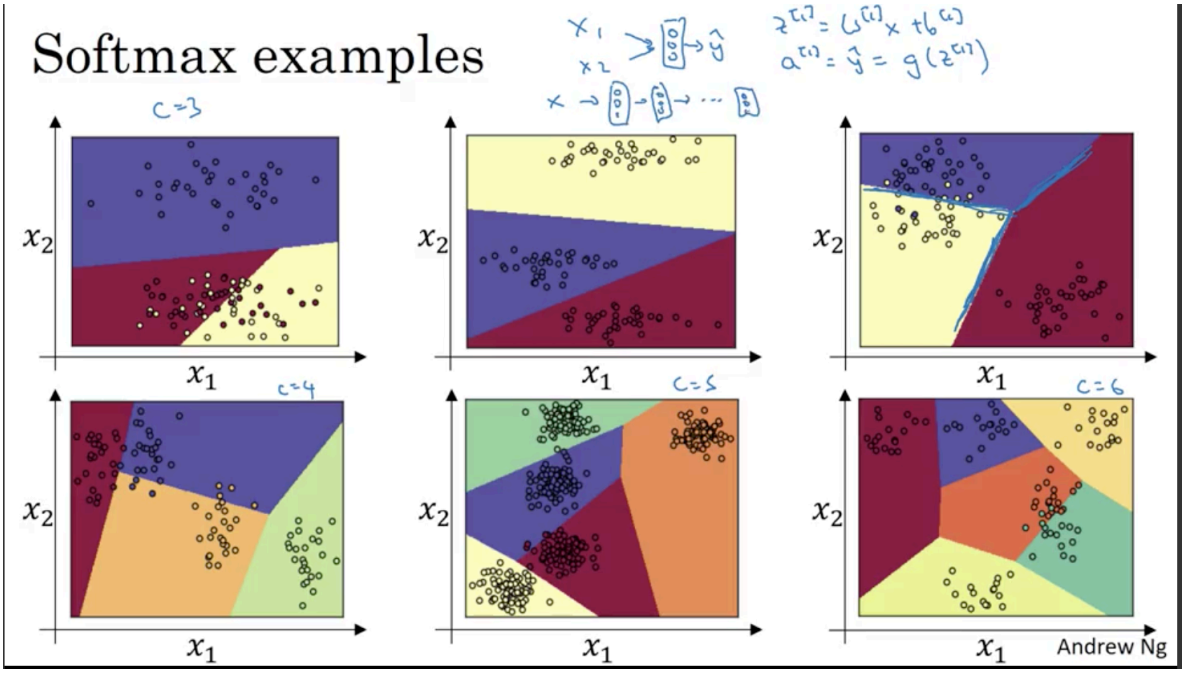
我们之前说的都是判断一个图是不是猫，只能分类一个
现在是想分类好几种：
我们约定C： 是classes的数量，我们这里是4（猫，狗，小鸡，其他）
然后我们的y hat 也不再是一个实数：0或1
而是一个vector:
dim = (C, 1)

然后y hat的前一层：输出层的unit数也要等于 C
即n[L] = C
然后每个unit的含义是：
P(other|x) 当我们输入x的时候，它是other的概率
==========
==========



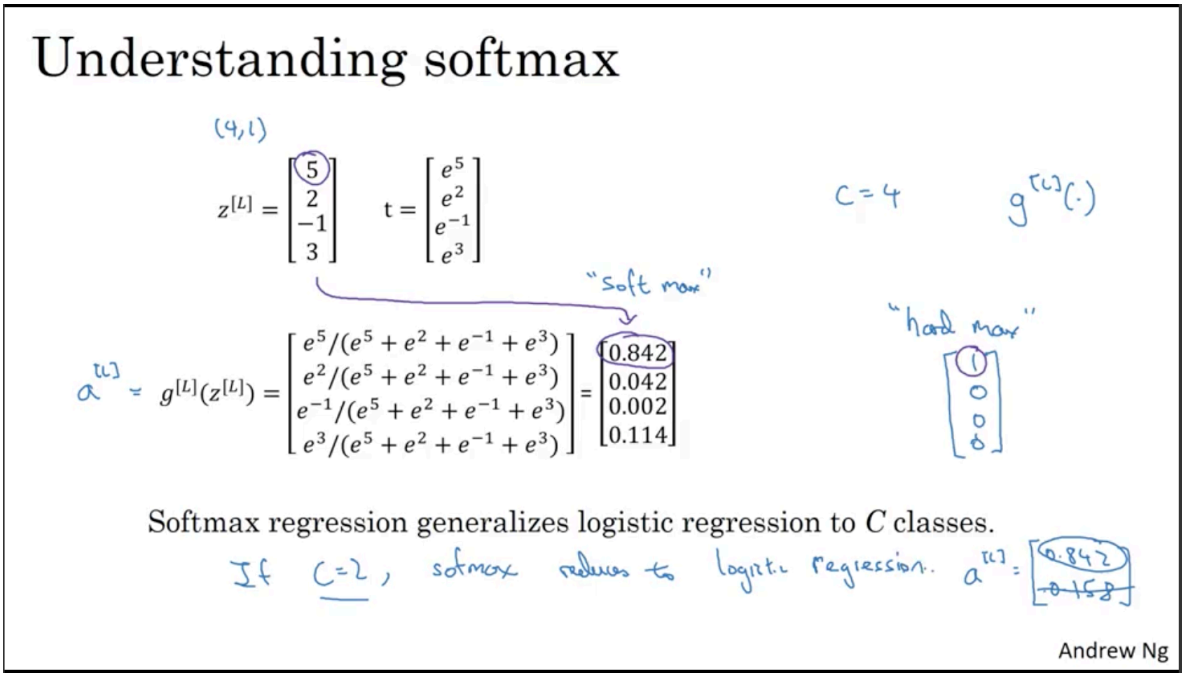
这一页具体说了softmax的做法
首先和之前不同的是
之前是z = w*x + b然后 a = af(z)
a只是一个数字，然后这个数字再pass到下一层

现在不一样了
af()不再是sigmoid， relu, tenh 这样只能返回一个值的activation function
而是：
图中所示的：加权平均（先计算出e^(z[l]),再算每个unit占总和的比例）
a[l] 的dim也将是(C,1)
老师举的例子中：
a[l]
= y hat
= [0.842
0.042
0.002
0.114]
==========
==========



这里说的是：我们之前的y hat只是一个实数
现在却是一个vector了：dim = (C, 1)

==========
==========

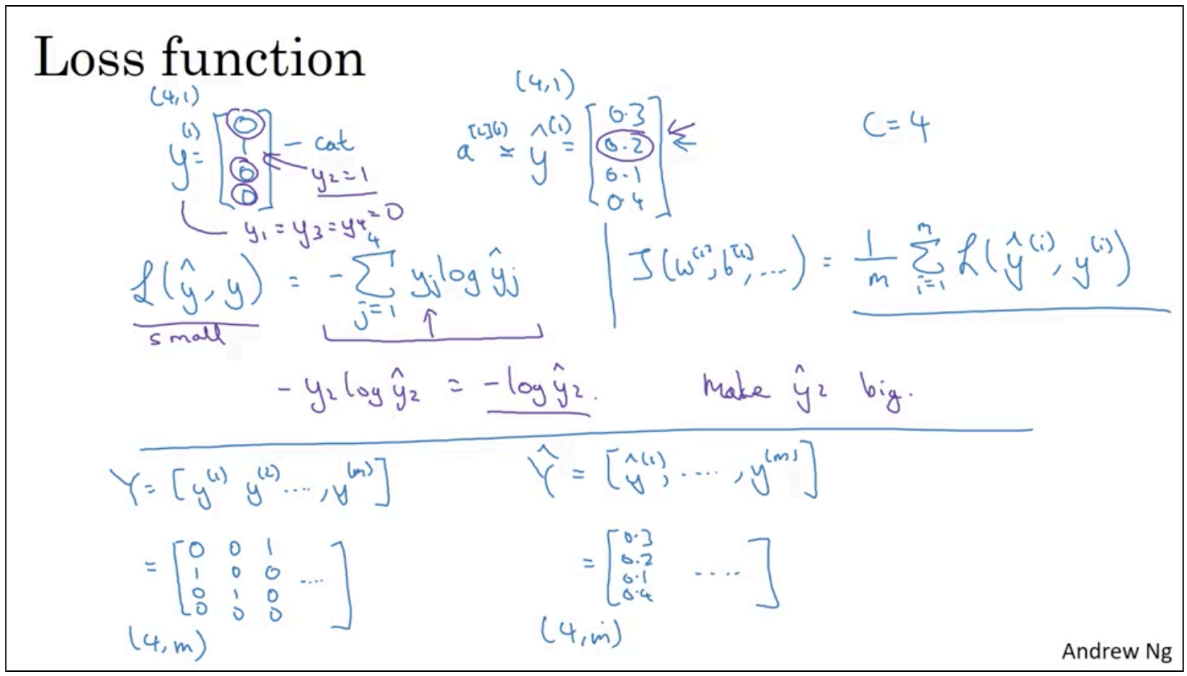


这里说的是：
softmax是logistic regression的泛化版
因为logistic regression相当于是hard max
让softmax中的最大的那个变成1，其余都是0（优点马太效应的感觉）
然后logistic regression相当于C = 2时候的softmax

老师右边的a[L] = [0.842

0.158]
的意思是说：
C= 2 的softmax
我们获得了这样的结果
a[L] = [0.842,0.158]
由于只有两类，所以没有必要计算0.158，因为当0.842大于0.5的时候，我们就已经知道它属于哪一类了
所以就写成了logistic regression的形式，只有0,1

==========
==========



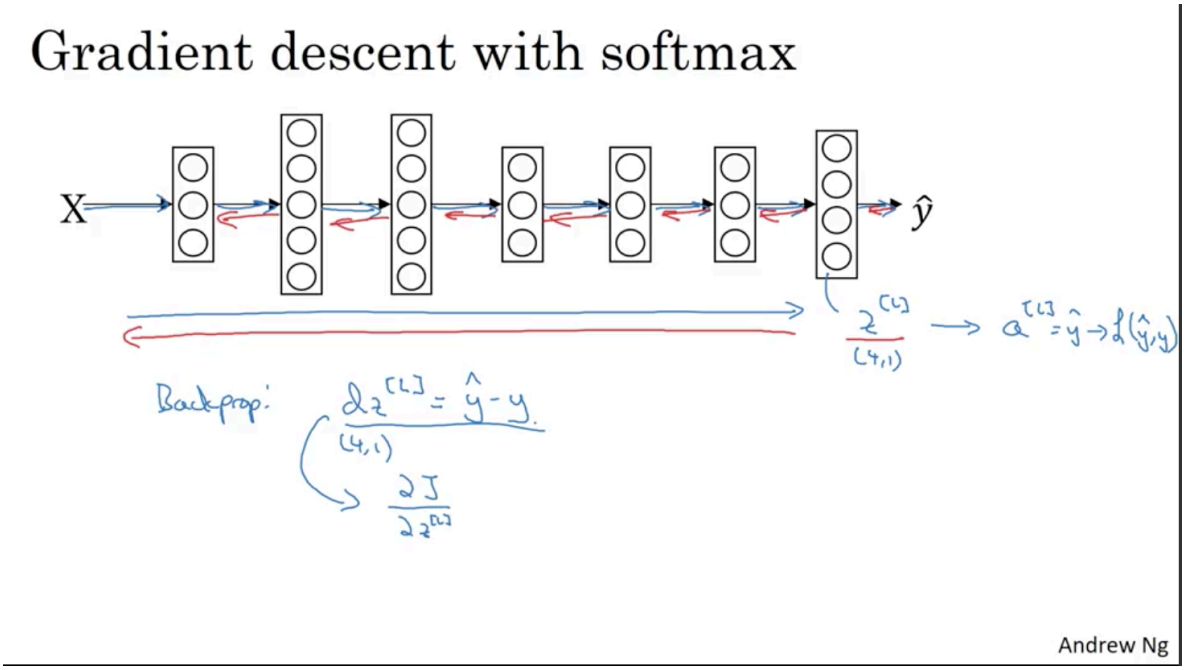
这一页说的是
如果使用softmax的时候的cost function
对于一个样本来说：
假设它的真实label是这样的：
y(1) ... 代表着第一个样本的 label:
[0
1
0
0]
说明这是一只cat
然后我们通过模型计算出来的
y hat(1): 第一个样本的label
[0.3
0.2
0.1
0.4] ...可见预测的真差。预测成了其他动物了

然后我们的loss function是：
L(y hat, y) = 如图所示，j是我们的classes数量

因为y2 = 1，
y1,y3,y4= 0
所以L(y hat, y) = -log(y hat2) = -log(0.2)
如果希望L(y hat, y)小的话，我们希望y hat2 能够大一些，也就印证了y hat2应该要趋近于1

接下来是所有m个样本的cost function：
J（...）= ...
很简单，就是m个L(y hat, y) 的平均数

==========
==========



这里说的是我们要计算的时候
也是需要计算back prop的
但是我们不要求掌握。。