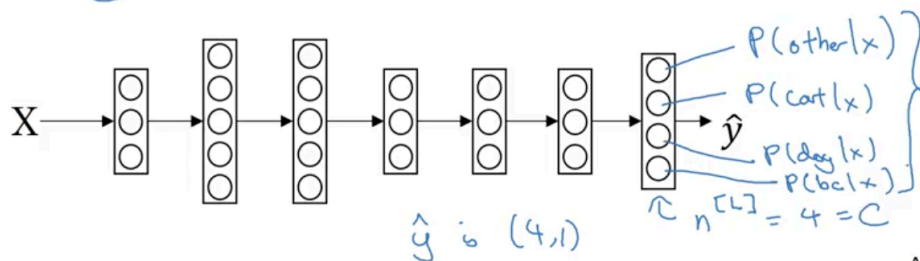


# Recognizing cats, dogs, and baby chicks, <sup>other</sup>



3      1      2      0      3      2      0      1

$C = \#classes = 4$       (0, ..., 3)



Andrew Ng

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 $n[L] = C$

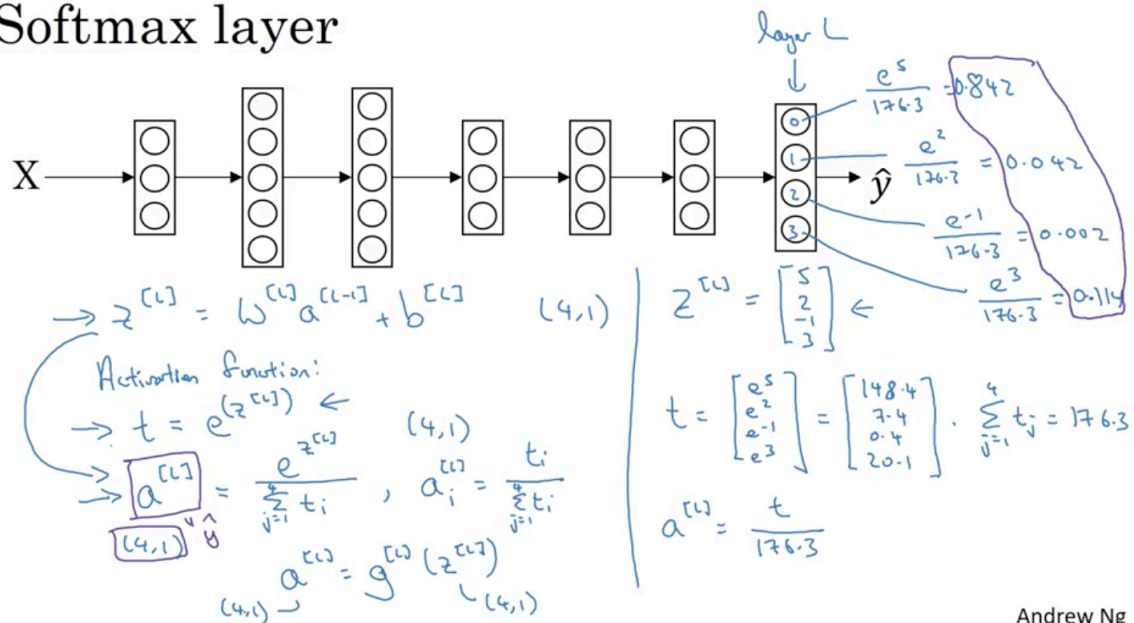
然后每个unit的含义是：

$P(\text{other}|x)$  当我们输入x的时候，它是other的概率

=====

=====

# Softmax layer



Andrew Ng

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  $a = \text{af}(z)$

$a$  只是一个数字，然后这个数字再pass到下一层

现在不一样了

$\text{af}()$  不再是 sigmoid, relu, tanh 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：

$a[l]$

=  $\hat{y}$

= [0.842

0.042

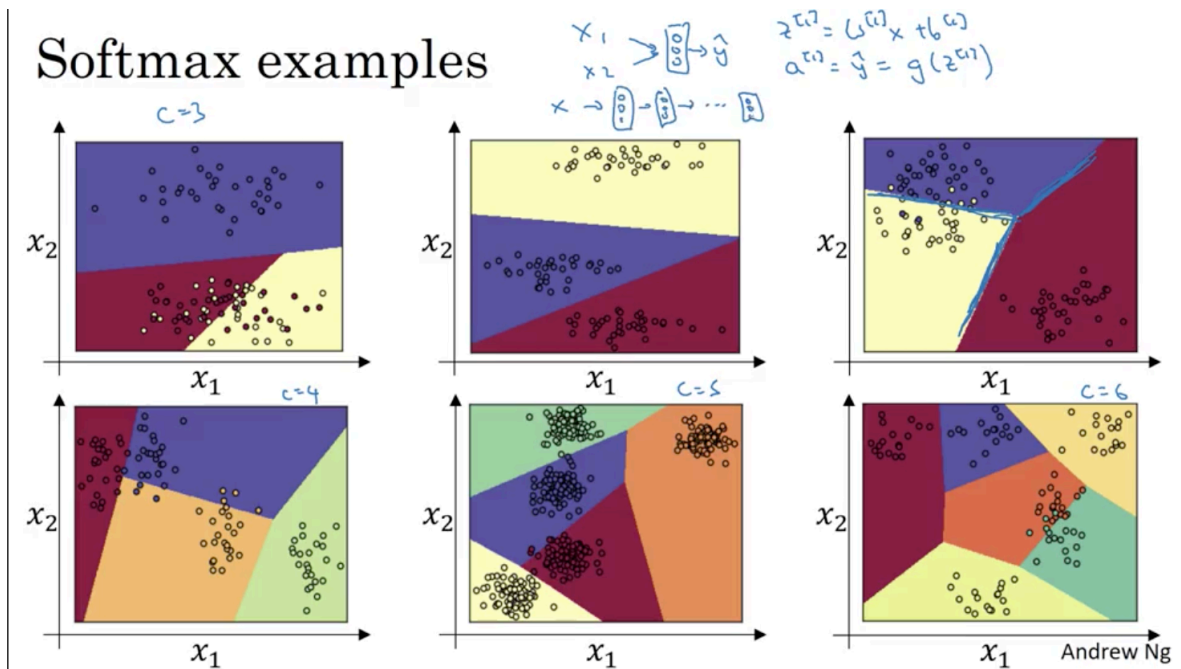
0.002

0.114]

=====

=====

## Softmax examples



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

=====

=====

## Understanding softmax

Handwritten notes:

$$z^{(L)} = \begin{bmatrix} 5 \\ 2 \\ -1 \\ 3 \end{bmatrix} \quad t = \begin{bmatrix} e^5 \\ e^2 \\ e^{-1} \\ e^3 \end{bmatrix}$$

"Soft max"

$$a^{(L)} = g^{(L)}(z^{(L)}) = \begin{bmatrix} e^5 / (e^5 + e^2 + e^{-1} + e^3) \\ e^2 / (e^5 + e^2 + e^{-1} + e^3) \\ e^{-1} / (e^5 + e^2 + e^{-1} + e^3) \\ e^3 / (e^5 + e^2 + e^{-1} + e^3) \end{bmatrix} = \begin{bmatrix} 0.842 \\ 0.042 \\ 0.002 \\ 0.114 \end{bmatrix}$$

Handwritten notes:

$$C=4 \quad g^{(L)}(\cdot)$$

"hard max"

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Softmax regression generalizes logistic regression to  $C$  classes.

If  $C=2$ , softmax reduces to logistic regression.  $a^{(L)} = \begin{bmatrix} 0.842 \\ 0.158 \end{bmatrix}$

Andrew Ng

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

=====  
=====

**Loss function**

$(4,1)$   
 $y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$  - cat  
 $y_2 = 1$   
 $y_1 = y_3 = y_4 = 0$

$(4,1)$   
 $a^{(1)} \approx \hat{y} = \begin{bmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{bmatrix}$

$C = 4$

$\mathcal{L}(\hat{y}, y) = - \sum_{j=1}^C y_j \log \hat{y}_j$   
s mall

$\mathcal{J}(w^{(1)}, b^{(1)}, \dots) = \frac{1}{m} \sum_{i=1}^n \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

$- y_2 \log \hat{y}_2 = - \log \hat{y}_2$       make  $\hat{y}_2$  big.

$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}]$   
 $= \begin{bmatrix} 0 & 0 & 1 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix}$   
 $(4, m)$

$\hat{Y} = [\hat{y}^{(1)} \ \dots \ \hat{y}^{(m)}]$   
 $= \begin{bmatrix} 0.3 & \dots \\ 0.2 & \dots \\ 0.1 & \dots \\ 0.4 & \dots \end{bmatrix}$   
 $(4, m)$

Andrew Ng

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

然后通过模型计算出来的

$\hat{y}(1)$ : 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：

$L(\hat{y}, y)$  = 如图所示，j是我们的classes数量

因为  $y_2 = 1$ ,

$y_1, y_3, y_4 = 0$

所以  $L(\hat{y}, y) = -\log(\hat{y}_2) = -\log(0.2)$

如果希望  $L(\hat{y}, y)$  小的话, 我们希望  $\hat{y}_2$  能够大一些, 也就印证了  $\hat{y}_2$  应该要趋近于 1

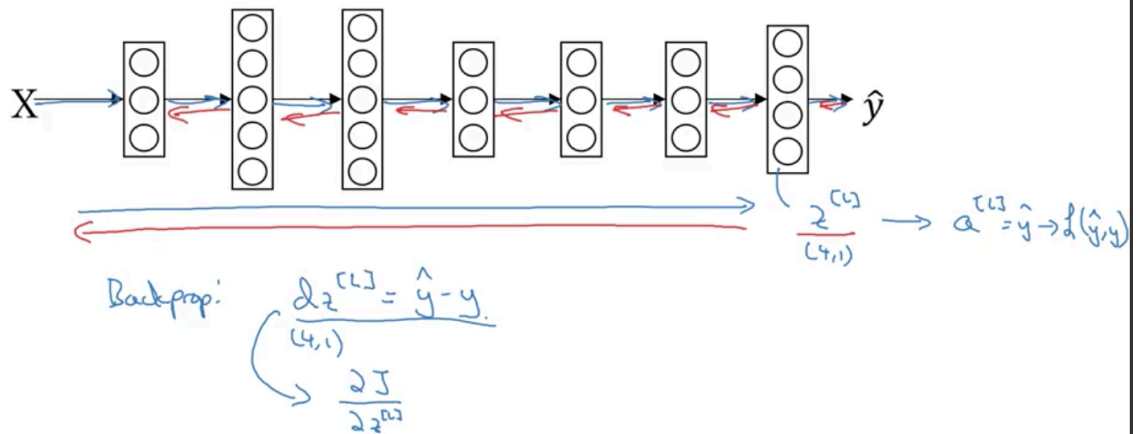
接下来是所有  $m$  个样本的 cost function :

$J(\dots) = \dots$

很简单, 就是  $m$  个  $L(\hat{y}, y)$  的平均数

=====

## Gradient descent with softmax



Andrew Ng

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。。