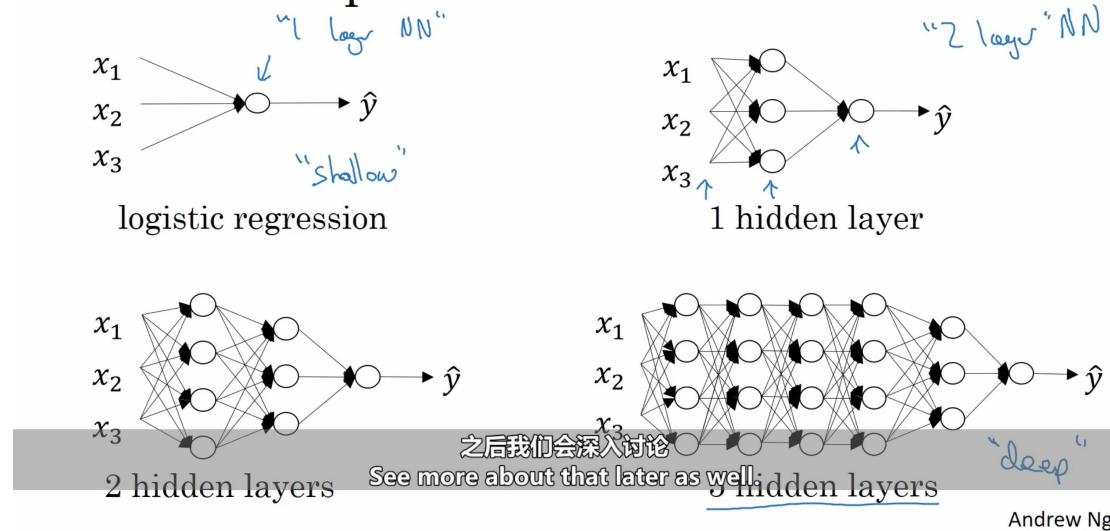


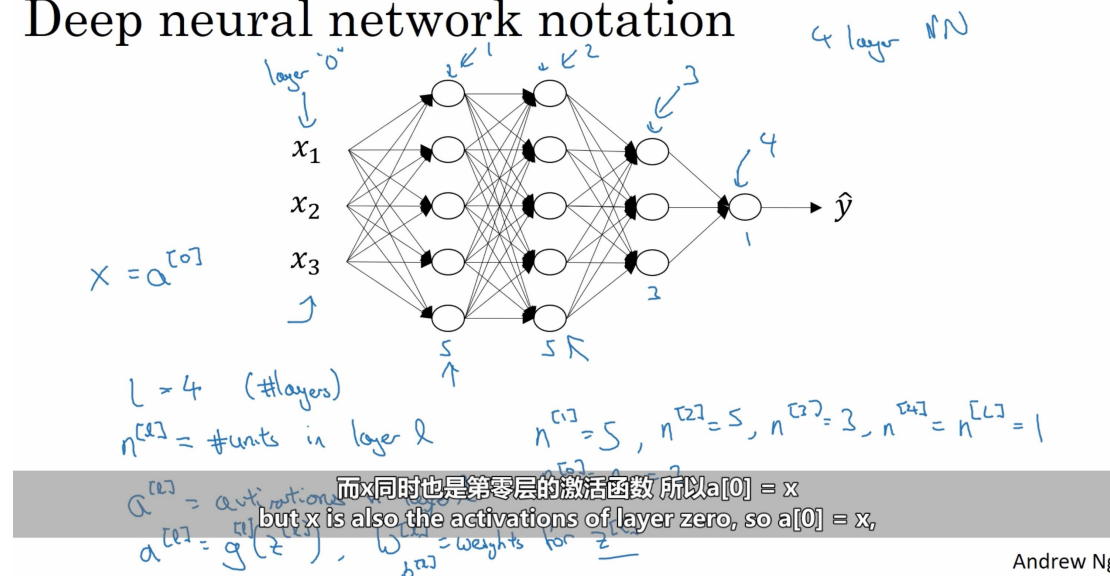
## 4 Deep Neural Networks

### 4.1 Deep L-layer Neural Network

What is a deep neural network?



Deep neural network notation



## 4.2 Forward Propagation in a Deep Network

### Forward propagation in a deep network

Handwritten equations for forward propagation:

$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g(z^{[l]})$$

$$A^{[l]} = g(z^{[l]})$$

For  $l=1 \dots 4$

So, in this place, it's perfectly okay to have an explicit For loop.

Andrew Ng

## 4.3 Getting your matrix dimensions right

### Parameters $W^{[l]}$ and $b^{[l]}$

Handwritten equations for matrix dimensions:

$$W^{[l]}: (n^{[l]}, n^{[l-1]})$$

$$b^{[l]}: (n^{[l]}, 1)$$

$$dW^{[l]}: (n^{[l]}, n^{[l-1]})$$

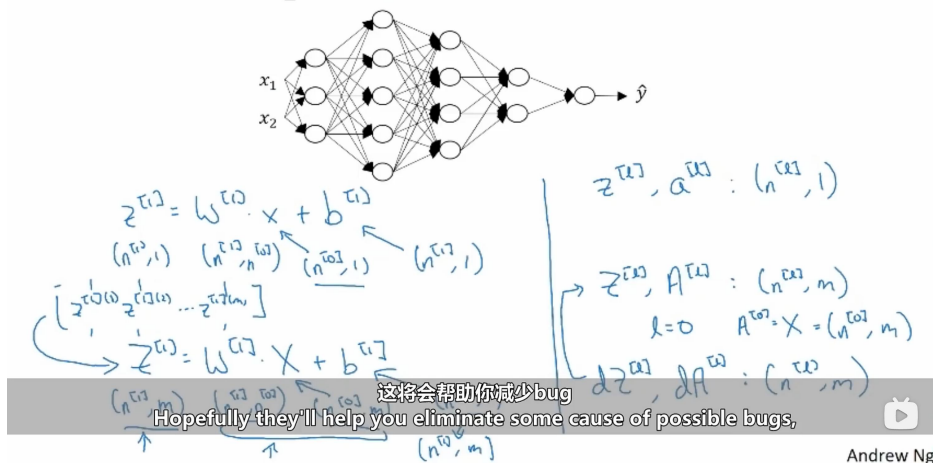
$$db^{[l]}: (n^{[l]}, 1)$$

For  $l=1 \dots 4$

But the dimensions of  $z$ ,  $a$ , as well as  $x$  will

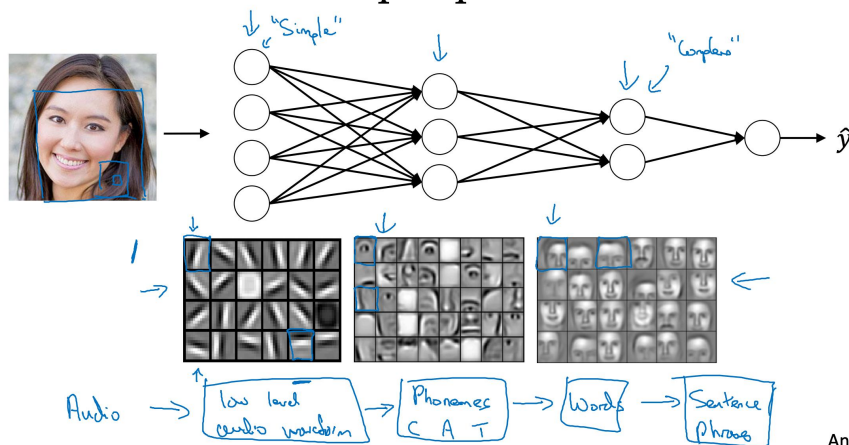
Andrew Ng

## Vectorized implementation



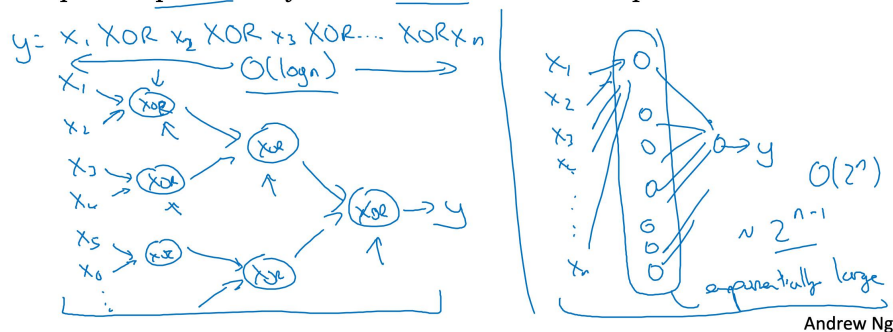
## 4.4 Why deep representations?

### Intuition about deep representation



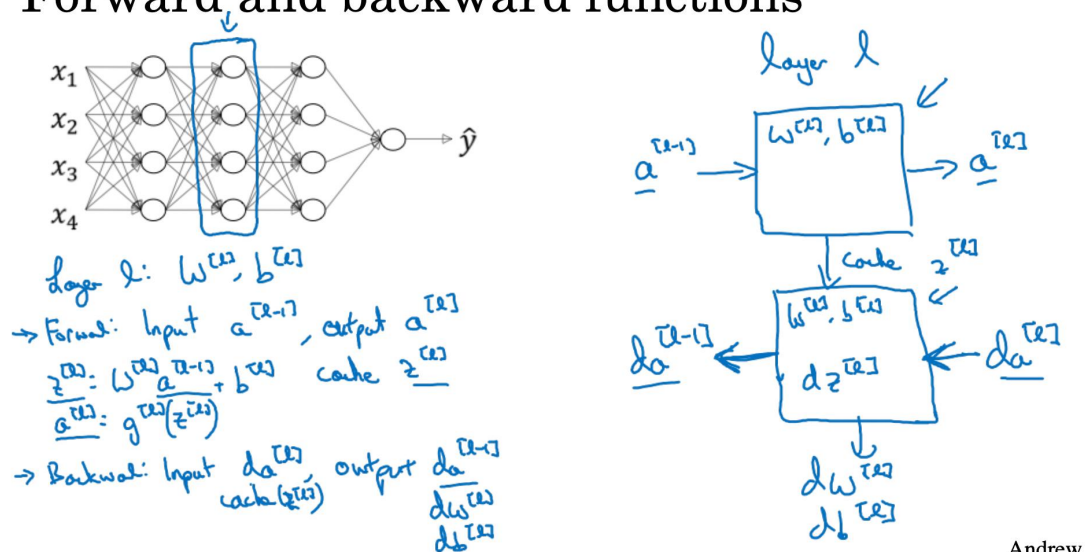
## Circuit theory and deep learning

Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.

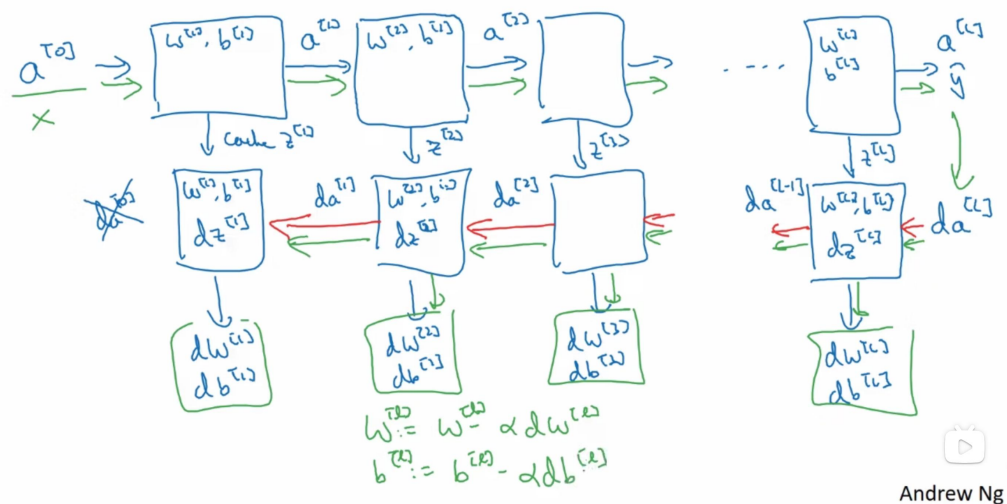


## 4.5 Building blocks of deep neural networks

### Forward and backward functions



### Forward and backward functions



## 4.6 Forward and backward propagation

### Forward propagation for layer $l$

→ Input  $a^{[l-1]} \leftarrow$

→ Output  $a^{[l]}$ , cache  $(z^{[l]})$

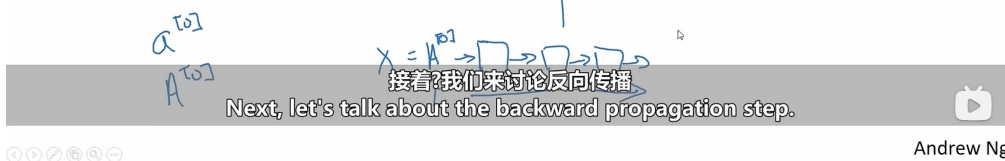
$$z^{[l]} = W^{[l]} \cdot a^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vectorized:

$$z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(z^{[l]})$$



Andrew Ng

### Backward propagation for layer $l$

→ Input  $da^{[l]}$

→ Output  $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$$dz^{[l]} = da^{[l]} * g'^{[l]}(z^{[l]})$$

$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$

$$db^{[l]} = dz^{[l]}$$

$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

$$dz^{[l-1]} = W^{[l+1]T} \cdot dz^{[l]} * g'^{[l+1]}(z^{[l+1]})$$

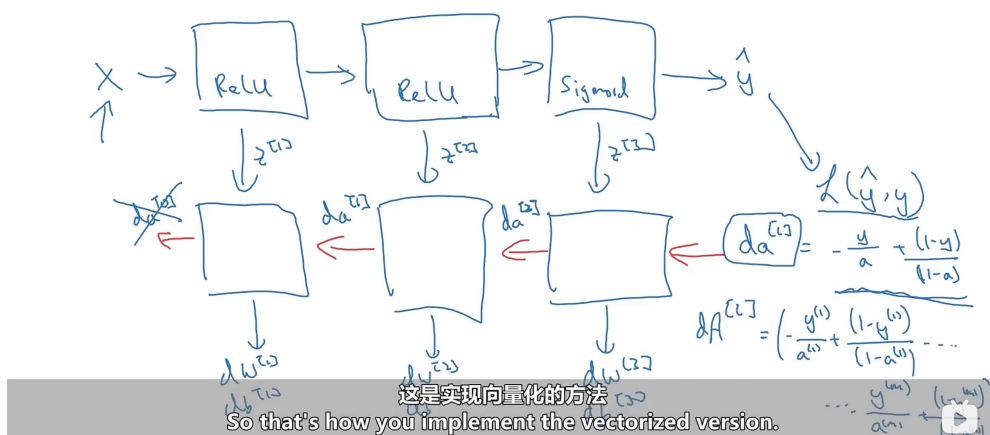
$$dz^{[l]} = dA^{[l]} * g'^{[l]}(z^{[l]})$$

$$dW^{[l]} = \frac{1}{n} dz^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{n} \text{np.sum}(dz^{[l]}, \text{axis}=1, \text{keepdims}=True)$$

$$dA^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

### Summary



Andrew Ng



## 4.7 Parameters vs Hyperparameters

What are hyperparameters?

Parameters:  $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \dots$

Hyperparameters:  $\left. \begin{array}{l} \text{learning rate } \alpha \\ \text{\# iterations} \\ \text{\# hidden layers } L \\ \text{\# hidden units } n^{[1]}, n^{[2]}, \dots \\ \text{choice of activation function} \end{array} \right\}$

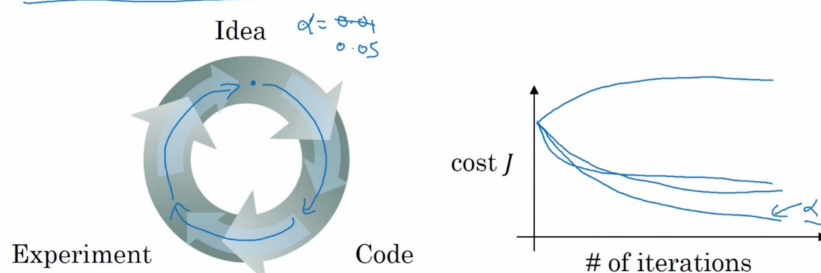
Also: Momentum, mini-batch size, regularizations, ...

会有很多不同的选择  
be a lot of possible settings for the



Andrew Ng

Applied deep learning is a very empirical process



Vision, Speech, NLP, Ad Sent Recommendation  
或其他集合上进行评估

cross-validation set or something and



Andrew Ng

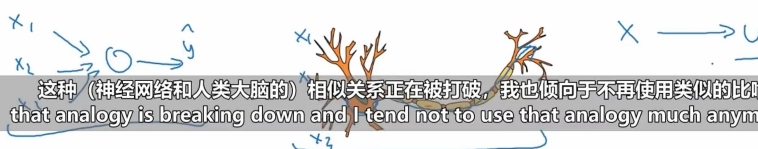
## 4.8 What does this have to do with the brain?

Forward and backward propagation

$$\begin{aligned} Z^{[1]} &= W^{[1]}X + b^{[1]} \\ A^{[1]} &= g^{[1]}(Z^{[1]}) \\ Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\ A^{[2]} &= g^{[2]}(Z^{[2]}) \\ &\vdots \\ A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y} \end{aligned}$$

"It's like the brain"

$$\begin{aligned} dZ^{[L]} &= A^{[L]} - Y \\ dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]T} \\ db^{[L]} &= \frac{1}{m} \text{np.sum}(dZ^{[L]}, \text{axis} = 1, \text{keepdims} = \text{True}) \\ dZ^{[L-1]} &= dW^{[L]T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\ &\vdots \\ dZ^{[1]} &= dW^{[L]T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\ dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]T} \\ db^{[1]} &= \frac{1}{m} \text{np.sum}(dZ^{[1]}, \text{axis} = 1, \text{keepdims} = \text{True}) \end{aligned}$$



这种 (神经网络和人类大脑的) 相似关系正在被打破, 我也倾向于不再使用类似的比喻  
that analogy is breaking down and I tend not to use that analogy much anymore.



Andrew Ng