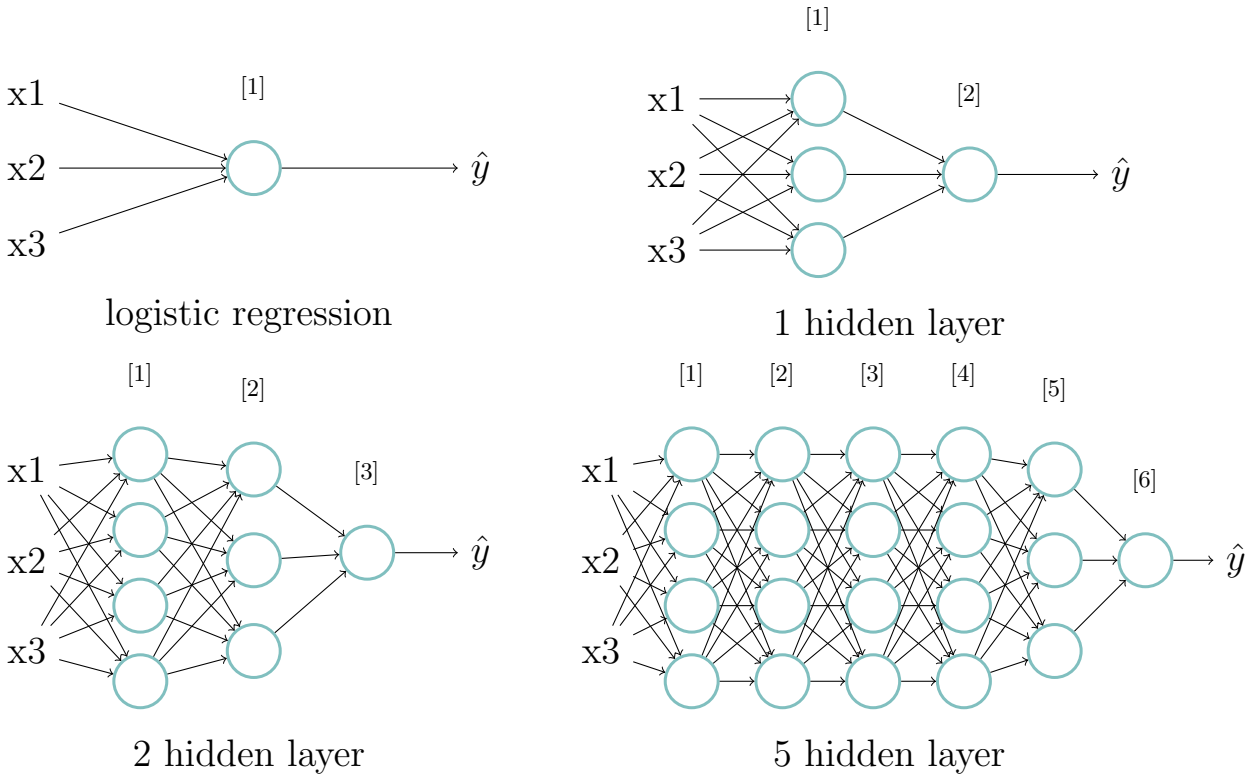
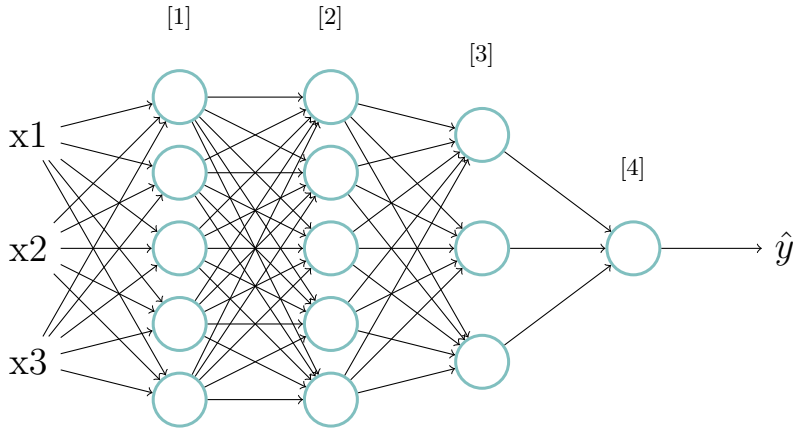
1 Week4

Deep Neural Network.

1.1 Depp L-layer Neural Network



Neural network with different hidden layers



$L = 4 =$ Total number of layers.

$X = a^{[0]}$

$\hat{y} = a^{[L]}$

$n^{[l]} =$ number of units in layer l

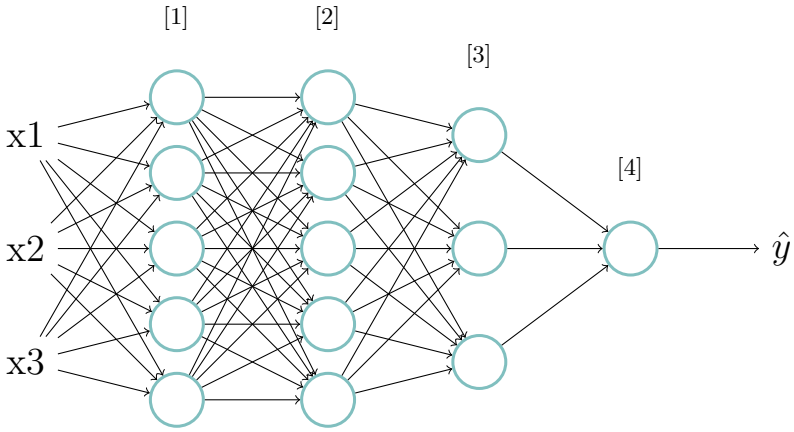
$a^{[l]} = g^{[l]}(z^{[l]}) =$ activation in layer l

$W^{[l]} =$ weights for $Z^{[l]}$

$b^{[l]} =$ bias for $Z^{[l]}$

$n^{[0]}=n_x=3,\ n^{[1]}=5,\ n^{[2]}=5,\ n^{[3]}=3,\ n^{[4]}=n^{[L]}=1$

1.2 Forward Propagation in a deep network



$$Z^{[1]} = W^{[1]}A^{[0]} + b^{[1]} \quad _{where\ A^{[0]}=X}$$

$$A^{[1]} = g^{[1]}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(Z^{[2]})$$

$$\dots$$

$$Z^{[4]} = W^{[4]}A^{[3]} + b^{[4]}$$

$$A^{[4]} = g^{[4]}(Z^{[4]}) = \hat{Y}$$

Using for loops on $[l]$ layer for $l = 1, 2, \dots, L$:

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$$

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

1.3 Getting Matrix Dimensions Right

For a single training example:

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$$

$(n^{[l]},1)=(n^{[l]},n^{[l-1]})\times(n^{[l-1]},1)+(n^{[l]},1))$

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

For m training examples:

$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$$

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

$$dZ^{[l]},\ dA^{[l]}$$

$(n^{[l]},m)=(n^{[l]},n^{[l-1]})\times(n^{[l-1]},m)+(n^{[l]},1)$, where $b^{[l]}$ is broadcasted to $(n^{[l]},m)$

1.3.1 Why Deep Representations?

Intuition about deep representation: Each layer could be edge dectector, facial detector, etc going from lower to deeper layers. Simple to complex hierarchical representation(compositional) could be applied to other types of data.(image, audio, ...).

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.

1.4 Building Blocks of a Deep Neural Network

Forward:

$a^{[l-1]} \rightarrow w^{[l]}, b^{[l]} \rightarrow a^{[l]}, \text{ cache } z^{[l]}$

Backward:

$da^{[l-1]}, dw^{[l]}, db^{[l]}, dz^{[l]} \leftarrow w^{[l]}, b^{[l]}, dz^{[l]} \leftarrow da^{[l]}$

Update parameters:

$w^{[l]} = w^{[l]} - \alpha dw^{[l]}$

$b^{[l]} = b^{[l]} - \alpha db^{[l]}$

1.5 Forward and Backward Propagation

Forward propagation for layer l: $a^{[l-1]} \rightarrow a^{[l]}, z^{[l]}, w^{[l]}, b^{[l]}$

$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$

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

(for $i = 1, \ldots, L$ with initial value $A^{[0]} = X$)

Backward propagation for layer l: $da^{[l]} \rightarrow da^{[l-1]}, dW^{[l]}, db^{[l]}$

$dZ^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})$

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

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

$dA^{[l-1]} = W^{[l]^T} dZ^{[l]}$

(with initial value $dA^{[L]} = A^{[L]} - Y$)

1.6 Parameter vs. Hyperparameter

Paramters: $W^{[l]}, b^{[l]}$

Hyperparameters: (it affects paramters)

learning rate α

number of iterations

number of hidden layers L

number of hidden units $n^{[l]}$

Choice of activation function

later: Momentum, minibatch size, regularization paramters, ...

Applied deep learning is a very empirical process

Idea, Code, Experiment

number of iterations $\rightarrow$ cost J decreases.

1.7 What does this have to do with the brain?