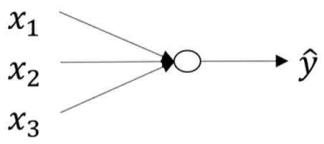
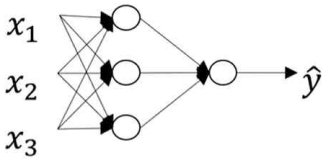
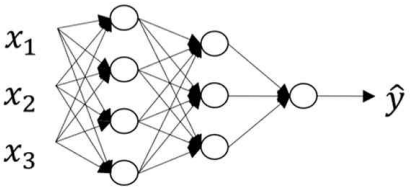
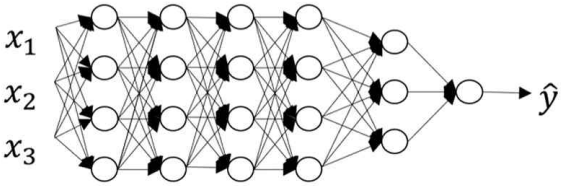
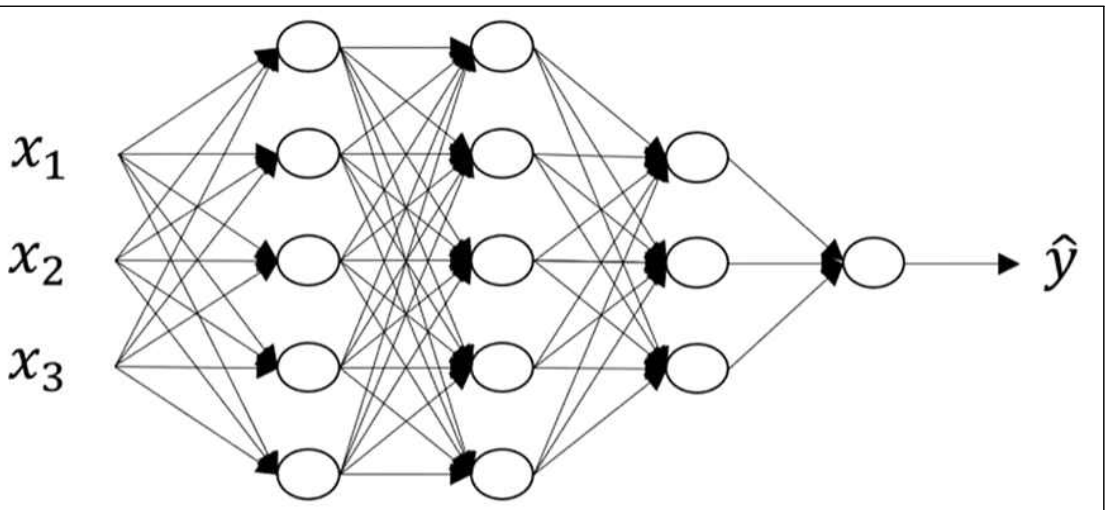
Deep Neural Networks

Deep L-layer Neural Networks

	
logistic regression	1 hidden layer
	
2 hidden layers	5 hidden layers



$L = \#layers$	$n^{[0]} = n_x = 3,$
$n^{[l]} = \#units\ in\ layer\ l$	$n^{[1]} = 5,\ n^{[2]} = 5,\ n^{[3]} = 3,$
$a^{[l]} = activation\ value\ in\ layer\ l$	$n^{[4]} = n^{[L]} = 1$
$W^{[l]}, b^{[l]} = weights\ for\ z^{[l]}\ in\ layer\ l$	
$a^{[l]} = g^{[l]}(z^{[l]})$	

Forward Propagation in a Deep Network

$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$ $A^{[l]} = g^{[l]}(Z^{[l]})$	$Z^{[1]} = W^{[1]} X + b^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$ $A^{[1]} = g^{[1]}(Z^{[1]})$ $Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$ $A^{[2]} = g^{[2]}(Z^{[2]}) = \sigma(Z^{[2]})$ $\vdots$ $\hat{Y} = g^{[4]}(Z^{[4]}) = A^{[4]}$

Getting matrix dimensions right

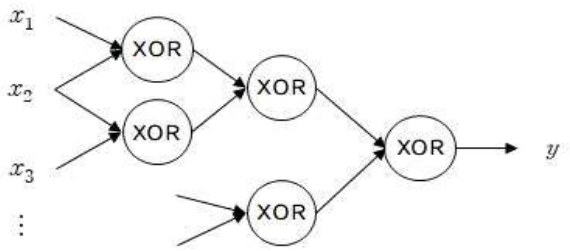
Parameters $W^{[l]}$ and $b^{[l]}$
$W^{[l]} : (n^{[l]}, n^{[l-1]})$ $b^{[l]} : (n^{[l]}, 1)$ $dW^{[l]} : (n^{[l]}, n^{[l-1]})$ $db^{[l]} : (n^{[l]}, 1)$
$l = 0,\ A^{[0]} = X : (n^{[0]}, m)$ $A^{[l-1]} : (n^{[l-1]}, m)$ $Z^{[l]} = A^{[l]} : (n^{[l]}, m)$ $dZ^{[l]} = dA^{[l]} : (n^{[l]}, m)$
$\underset{(n^{[l]}, m)}{Z^{[l]}} = \underset{(n^{[l]}, n^{[l-1]})}{W^{[l]}} \underset{(n^{[l-1]}, m)}{A^{[l-1]}} + \underset{(n^{[l]}, 1) \to (n^{[l]}, m)}{b^{[l]}}$

Why deep representations?

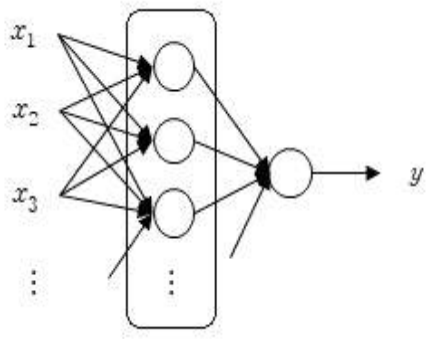
circuit theory and deep learning

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.

$y = x_1$ xor x_2 xor ⋯ xor x_n



$\Rightarrow O(\log n)$



$\Rightarrow O(2^n) = 2^{n-1}$

The deeper layers of a neural network are typically computing more complex features of the input than the earlier layers.

(i) To compute the function using a shallow network circuit, you will need a large network (where we measure size by the number of logic gates in the network), but (ii) To compute it using a deep network circuit, you need only an exponentially smaller network.

"Applied deep learning is a very empirical process."
최근에는 데이터가 새로 들어오면서 가장 좋은 hyperparameter 값이 쉽게 변할 수 있다. 오늘 가장 좋았던 learning rate 값이 한 달 뒤 데이터가 변하거나 새로 추가되면서 가장 좋은 learning rate 값이 변할 수 있다. 그 때문에 경험적 과정인 것이다.

Building blocks of deep neural networks

Forward and Backward functions

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

Forward:

input $A^{[l-1]}$, output $A^{[l]}$

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

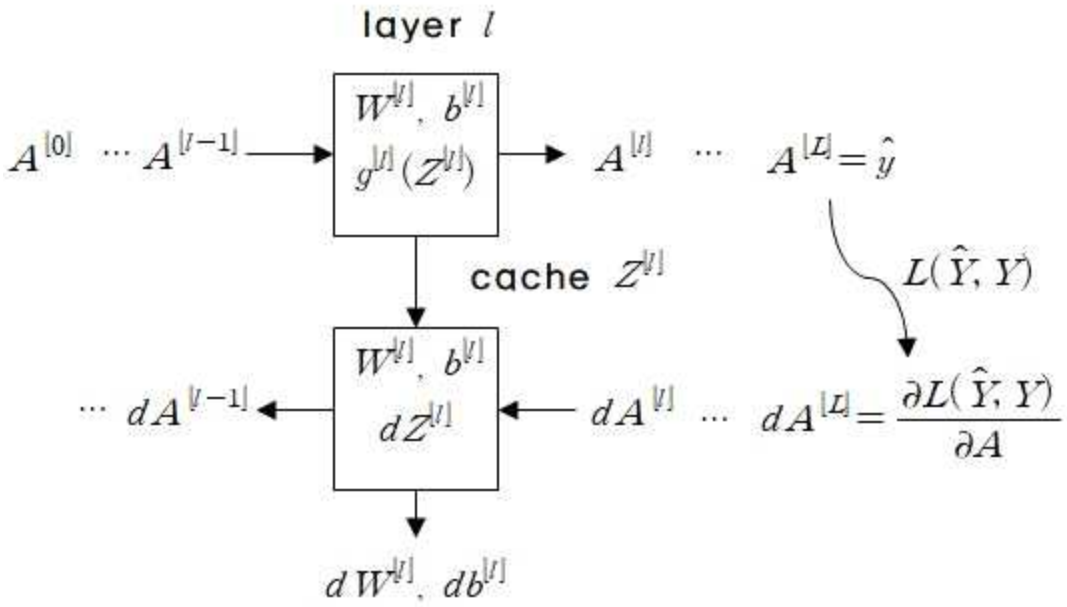
$A^{[l]} = g^{[l]}(Z^{[l]})$

Backward:

input $da^{[l]}$, cache $Z^{[l]}$

ouput $da^{[l-1]}$, $dw^{[l]}$, $db^{[l]}$

$da^{[0]}$은 어느 곳에도 사용되지 않기 때문에 구하지 않아도 괜찮다.



Parameters vs Hyperparameters

Parameters: weights $W^{[l]}$, $b^{[l]}$

Hyperparameters: parameters를 결정하는 요소

- learning rate
- # iterations
- # hidden layers
- # hidden units
- choice of activation function
- momentum, minibatch size, regularization