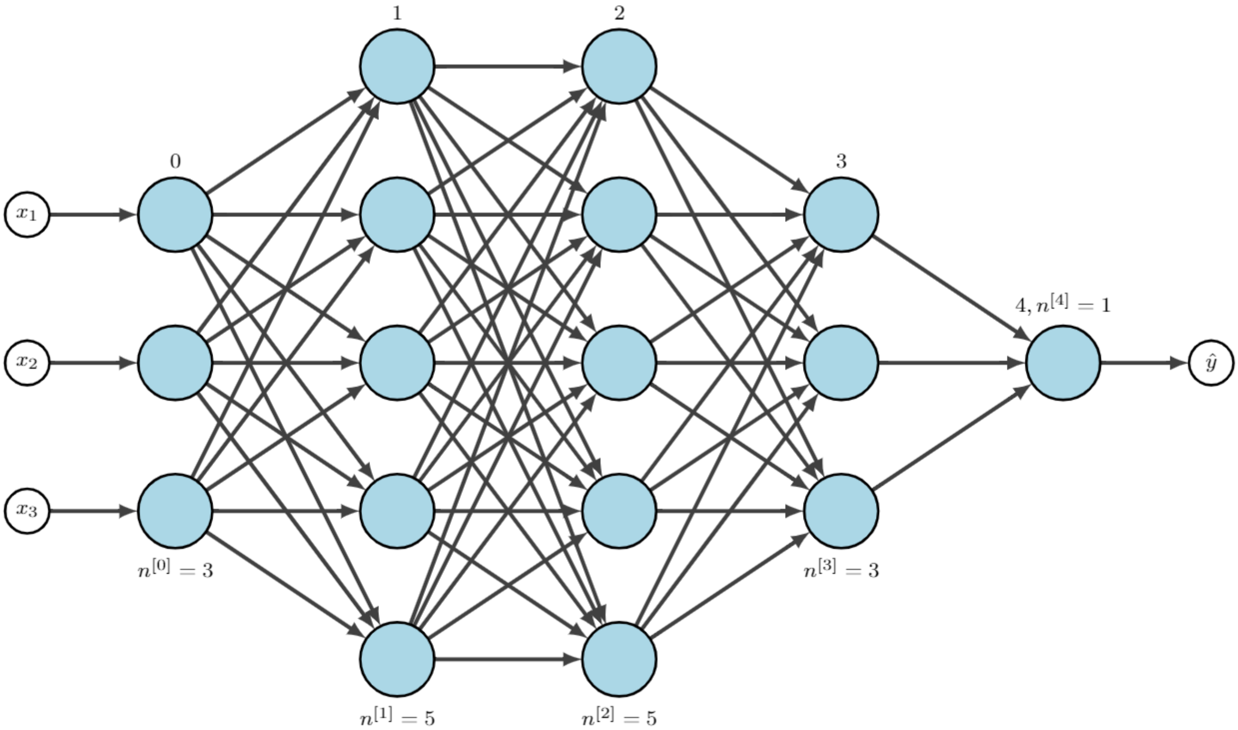
Deep layer neural network

- 

- $n^{[l]} = num\ of units\ in\ l^{th}\ layer$
- $L = num\ of\ total\ layer$
- $a^{[l]} = g^{[l]}(z^{[l]}),\ a^{[0]} = x$
- $W^{[l]},\ b^{[l]} = weights\ for\ z^{[l]}$

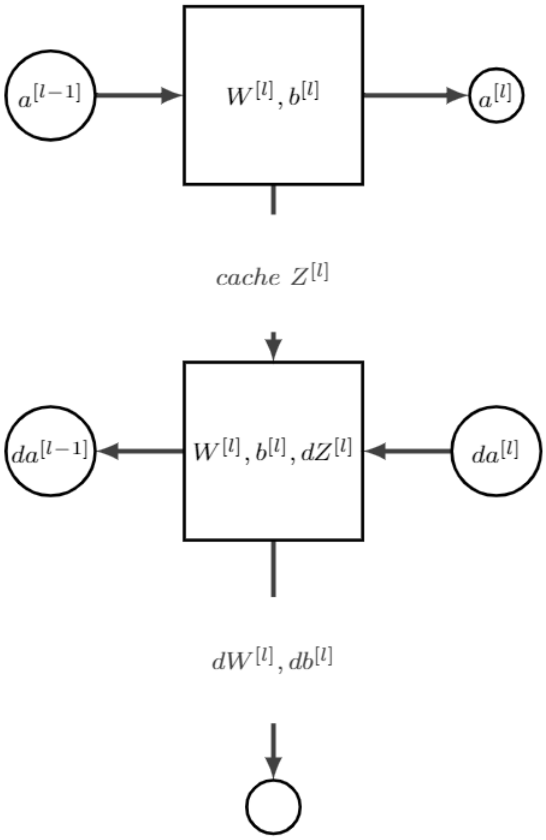
Forward propagation in a deep network

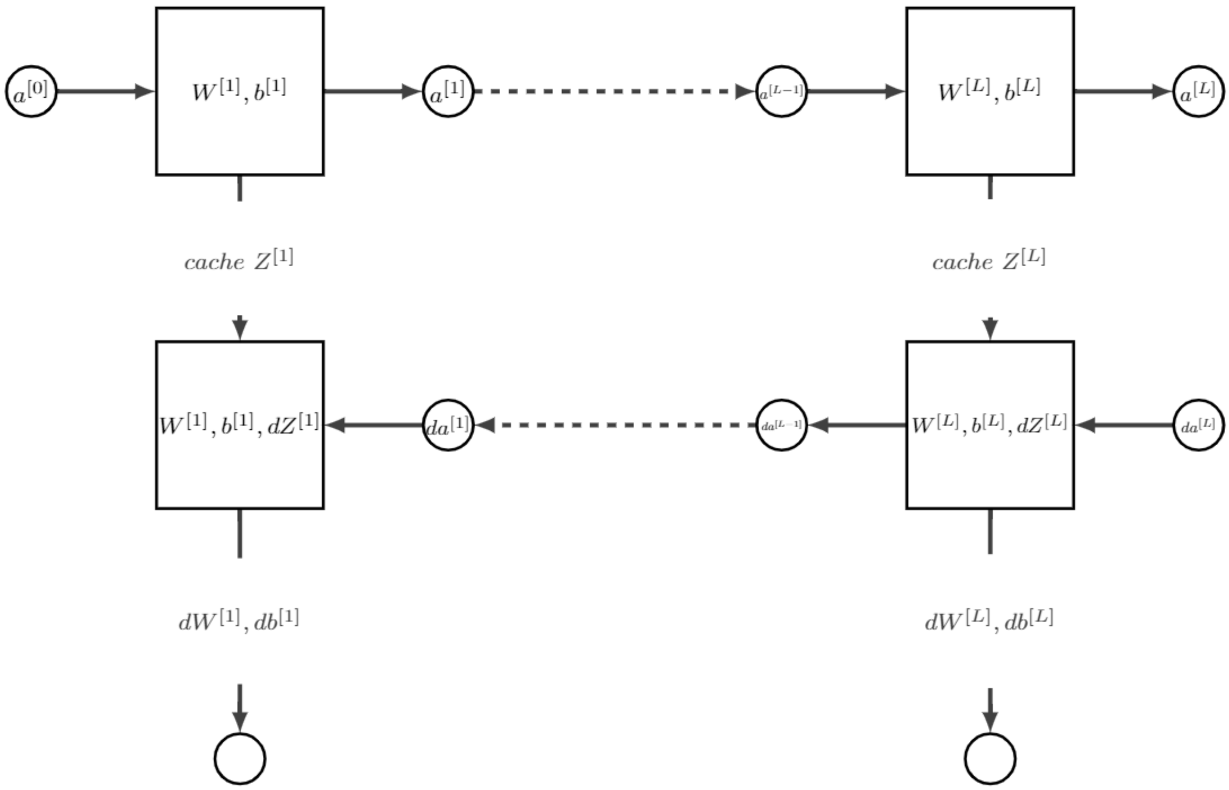
- $A^{[0]} = X$
- $for\ l\ = 1 \dots L$
  - $Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]}$
  - $A^{[l]} = g^{[l]}(Z^{[l]})$
- $\hat{Y} = A^{[L]}$
- Here it is **ok** to use *for loop*

Parameters W and b

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

Building blocks of deep neural network

- 

- 

Forward propagation for layer l

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

Backward propagation for layer l

- individual case
  - $dz^{[l]} = da^{[l]} .* g^{[l]'}(z^{[l]}) = (w^{[l+1]})^T dz^{[l+1]} .* g^{[l]'}(z^{[l]})$
  - $dw^{[l]} = dz^{[l]} a^{[l-1]}$
  - $db^{[l]} = dz[l]$
  - $da^{[l-1]} = (w^{[l]})^T dz^{[l]}$
  - $da^{[L]} = \frac{d}{da^{[L]}} \mathscr{L}(\hat{y}, y) = -\frac{y}{a} + \frac{1-y}{1-a},\ a^{[L]} = \hat{y}$
- vectorized
  - $dZ^{[l]} = dA^{[l]} .* g^{[l]'}(Z^{[l]})$
  - $dW^{[l]} = \frac{1}{m} dZ^{[l]} (A^{[l-1]})^T$
  - $db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis = 1, keepdims = True)$
  - $dA^{[l-1]} = (W^{[l]})^T dZ^{[l]}$
  - $dA^{[L]} = \begin{bmatrix} -\frac{y^{(1)}}{a^{(1)}} + \frac{1-y^{(1)}}{1-a^{(1)}} & \cdots & -\frac{y^{(m)}}{a^{(m)}} + \frac{1-y^{(m)}}{1-a^{(m)}} \end{bmatrix}$

Hyperparameters

- learning rate α
- #iterations
- #hidden layers L
- #hidden units $n^{[l]}$
- choice of activation functions
- ... ...