Vectorized version:

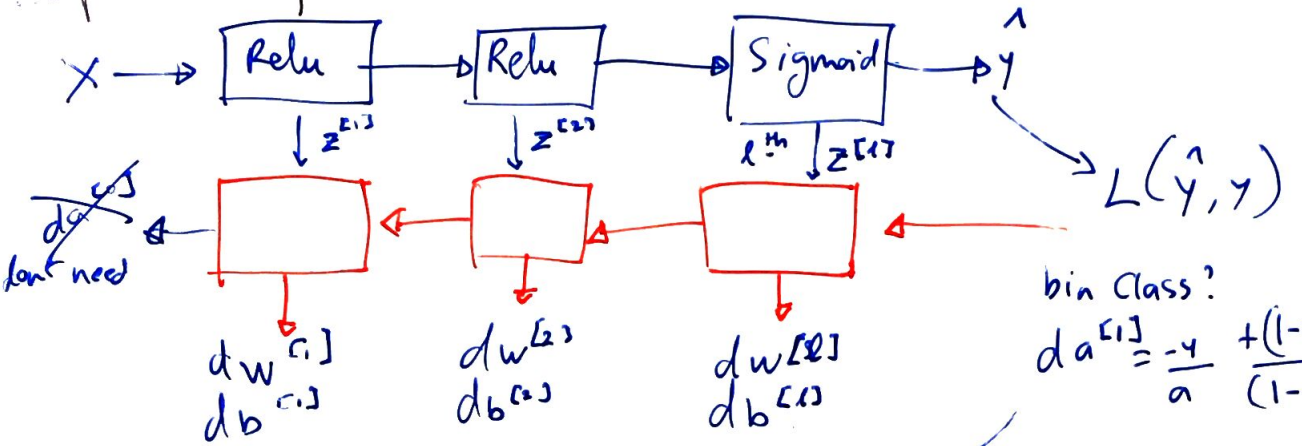
cached → $dZ^{[l]} = dA^{[l]} * g^{[l]}(Z^{[l]})$

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

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

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

Update Cycle:



bin class?

$da^{[l]} = \frac{-y}{a} + \frac{(1-y)}{(1-a)}$

Vectorized v^n =

$dA^{[l]} = \left(\frac{-y^{[1]}}{a^{[1]}} + \frac{1-y^{[1]}}{1-a^{[1]}} \cdots \right.$

$\left. \cdots \frac{y^{[m]}}{a^{[m]}} + \frac{1-y^{[m]}}{1-a^{[m]}} \right)$

Hyperparameters:

Hyperparams values impact learning algo's params

Params: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]} \cdots$

Hyperparams: (controls)

- Learning rate α
- # iterations
- # hidden layer L
- # hidden units $n^{[1]}, n^{[2]} \cdots$
- choice of activation f^n

Later will also see Momentum, minibatch size, regularizations, ---

→ Applied deep learning is a very empirical process

gotta try a lot of values