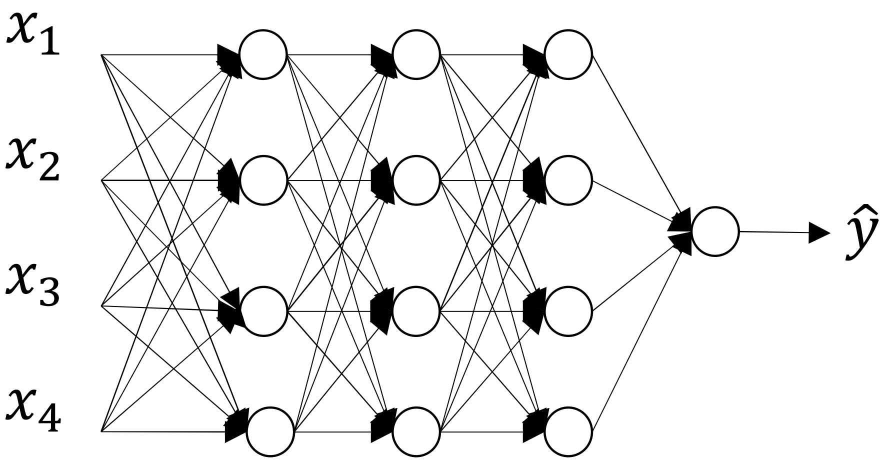
deeplearning.ai

Regularizing your neural network

Dropout regularization

Dropout regularization



↑ ↑ ↑

0.5 0.5 0.5

Implementing dropout ("Inverted dropout")

Illustrate with layer $l=3$. keep-prob = 0.8

0.2

→ $\boxed{d3}$ = np.random.rand(a3.shape[0], a3.shape[1]) < keep-prob

a3 = np.multiply(a3, d3)

a3 *= d3.

→ a3 /= ~~0.8~~ keep-prob ←

50 units. ⇝ 10 units shut off

$$z^{[4]} = w^{[4]} \cdot a^{[3]} + b^{[4]}$$

reduced by 20%.

Test

/= 0.8

Andrew Ng

Making predictions at test time

$a^{[0]} = X$

No drop out.

$$z^{[1]} = W^{[1]} \underline{a^{[0]}} + b^{[1]}$$

$$a^{[1]} = g^{[1]}(z^{[1]})$$

$$z^{[2]} = W^{[2]} \underline{a^{[1]}} + b^{[2]}$$

$$a^{[2]} = \ldots$$

$$\downarrow$$

$$\hat{y}$$

$/=$ keep-prob

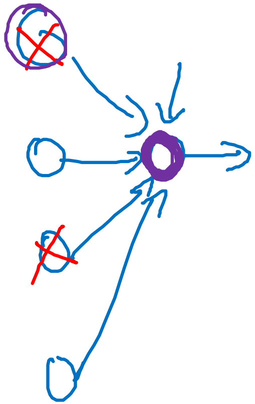
deeplearning.ai

Regularizing your neural network

Understanding dropout

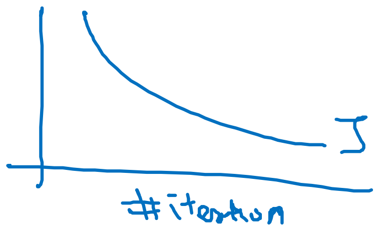
Why does drop-out work?

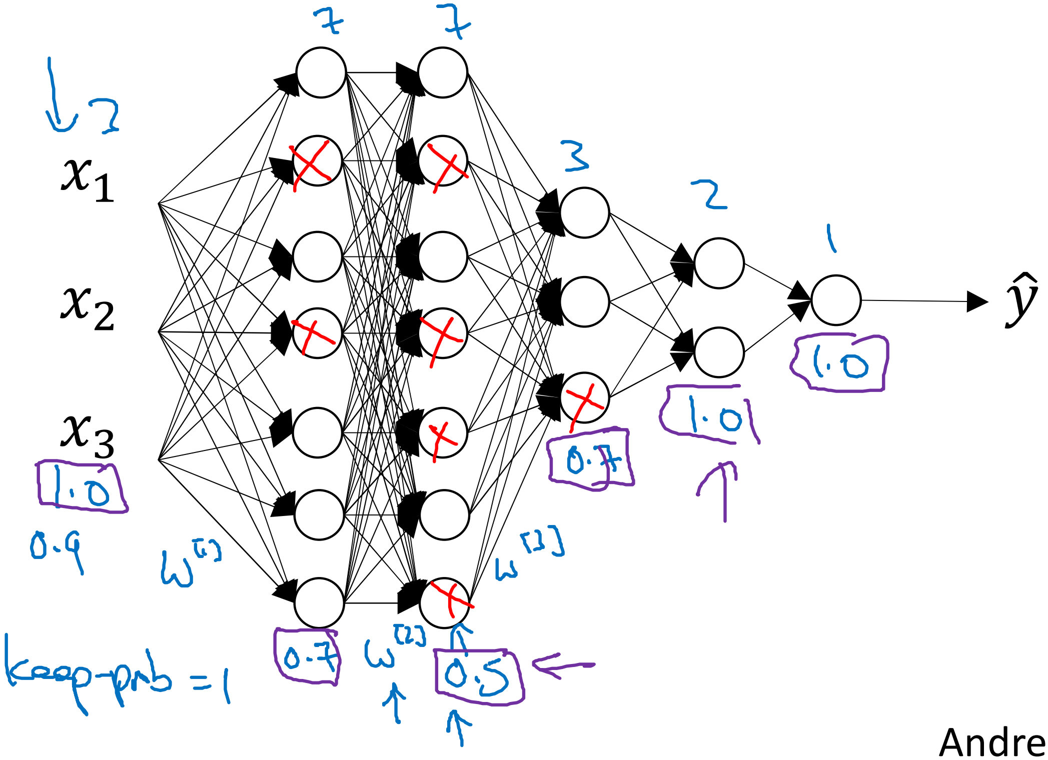
Intuition: Can't rely on any one feature, so have to spread out weights. ⇝ Shrink weights. L2



→ Computer Vision

Downside: J





Andrew Ng