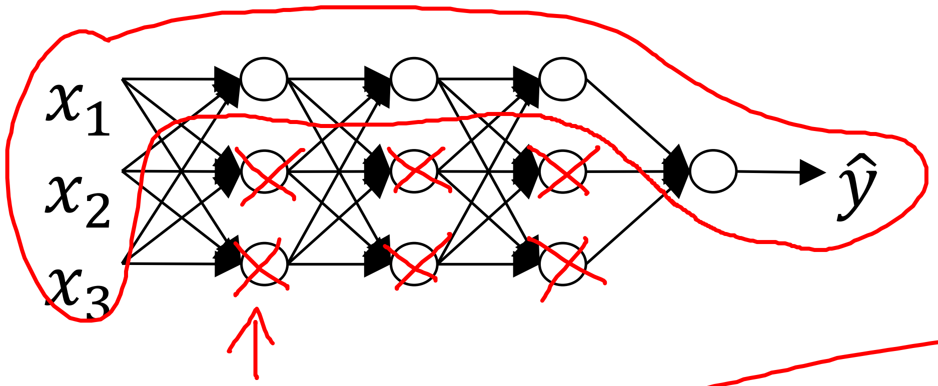
deeplearning.ai

# Regularizing your neural network

# Why regularization reduces overfitting

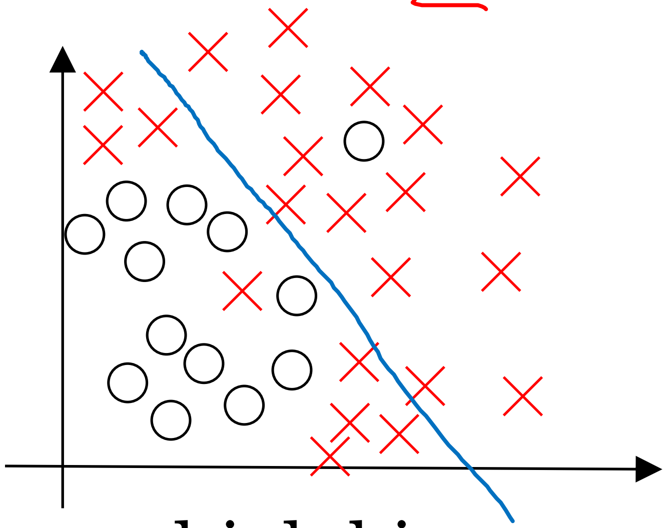
# How does regularization prevent overfitting?



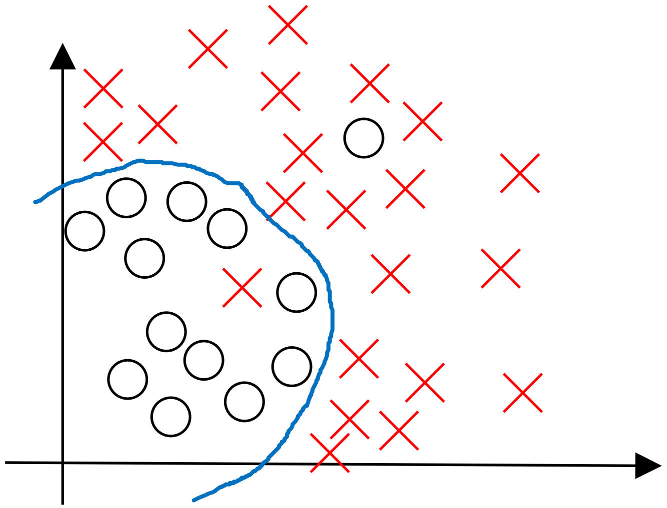

$$J(w^{[l]}, b^{[l]}) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L} \|w^{[l]}\|_F^2$$

add this term that penalizes the weight matrices from being too large. So that was Frobenius norm. So why is it that shrinking the L2 norm or the Frobenius norm...

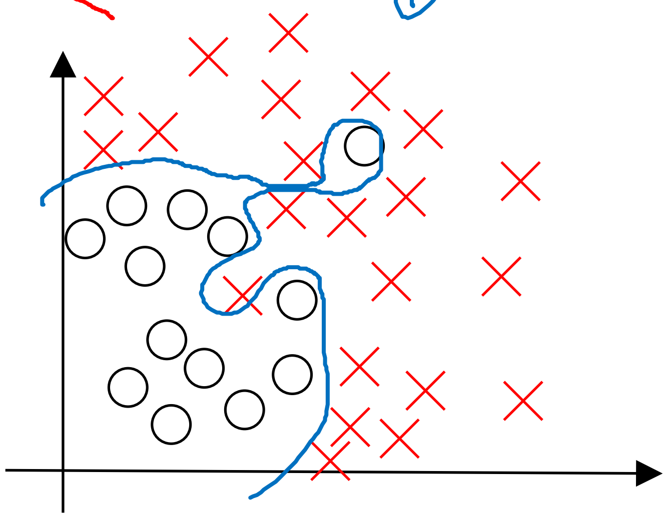
$$w^{[l]} \approx 0$$



high bias

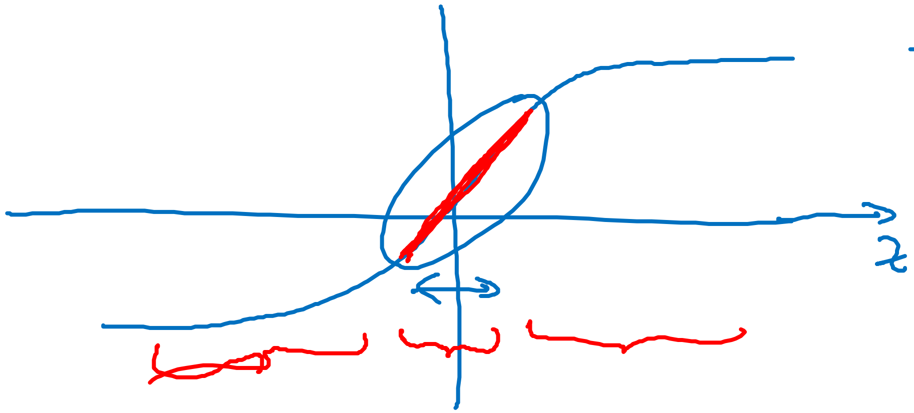


"just right"



high variance

# How does regularization prevent overfitting?



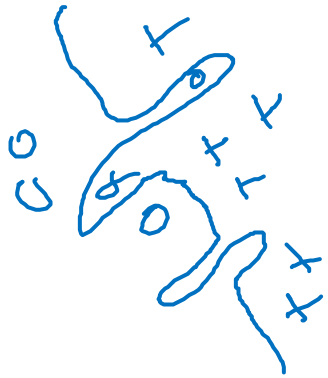

tanh

$g(z) = \tanh(z)$

So it's not able to fit those very very complicated decision, very non-linear decision boundaries that allow it to really overfit right to data sets.

$\lambda \uparrow \qquad W^{[l]} \downarrow$

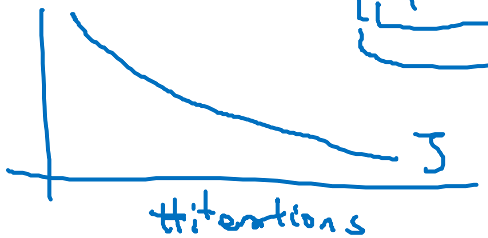
$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$



And so your whole neural network will be computing something not too far from a big linear function which is therefore pretty simple function rather a very complex highly non-linear function. So is also less able to overfit.

Every layer $\approx$ linear.

$$J(\dots) = \boxed{\sum_i \mathcal{L}(\hat{y}^{(i)}, y^{(i)})} + \frac{\lambda}{2m} \sum_l \|W^{[l]}\|_F^2$$




Including the second term as well. Otherwise you might not see J decrease monotonically on every single iteration.

Andrew Ng