



deeplearning.ai

Regularizing your neural network

Regularization

First thing you try when you get high variance.

Logistic regression

W is usually a pretty high dimensional parameter vector, especially with a high variance problem. Maybe w just has a lot of parameters, so you aren't fitting all the parameters well, whereas b is just a single number. So almost all the parameters are in w rather than b.

And usually, you set this using your development set, or using hold-out cross validation set.

$$\min_{w,b} J(w, b)$$

$$\underline{w \in \mathbb{R}^{n_x}}, \underline{b \in \mathbb{R}}$$

λ = regularization parameter
lambda lambda

$$J(w, b) = \underbrace{\frac{1}{m} \sum_{i=1}^m \ell(y^{(i)}, \hat{y}^{(i)})}_{\text{cost function}} + \frac{\lambda}{2m} \underbrace{\|w\|_2^2}_{\text{L2 regularization}}$$

~~$$+ \frac{\lambda}{2m} b^2$$~~
 omit

if you have the last term, in practice, it won't make much of a difference.

L2 regularization

$$\underline{\|w\|_2^2} = \sum_{j=1}^{n_x} w_j^2 = w^T w \leftarrow$$

You're using the Euclidean norms, just called the L2 norm with the parameter vector w.

L1 regularization

$$\frac{\lambda}{2m} \sum_{j=1}^{n_x} |w_j| = \frac{\lambda}{2m} \|w\|_1$$

w will be sparse

Although, I find that, in practice, L1 regularization to make your model sparse help only a little bit in reducing the storage.

Neural network

$$\rightarrow J(w^{[1]}, b^{[1]}, \dots, w^{[L]}, b^{[L]}) = \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(y^{(i)}, \hat{y}^{(i)})}_{\text{loss}} + \underbrace{\frac{\lambda}{2n} \sum_{l=1}^L \|w^{[l]}\|_F^2}_{\text{weight decay}}$$

$$\|w^{[l]}\|_F^2 = \sum_{i=1}^{n^{[l]}} \sum_{j=1}^{n^{[l-1]}} (w_{ij}^{[l]})^2$$

$w^{[l]}: \begin{matrix} n^{[l]} & n^{[l-1]} \\ \uparrow & \uparrow \end{matrix}$

"Frobenius norm"

$\|\cdot\|_2^2$

$\|\cdot\|_F^2$

$$dw^{[l]} = \left[(\text{from backprop}) + \frac{\lambda}{n} w^{[l]} \right]$$

$$\rightarrow w^{[l]} := w^{[l]} - \alpha dw^{[l]}$$

$$\frac{\partial J}{\partial w^{[l]}} = dw^{[l]}$$

"Weight decay"

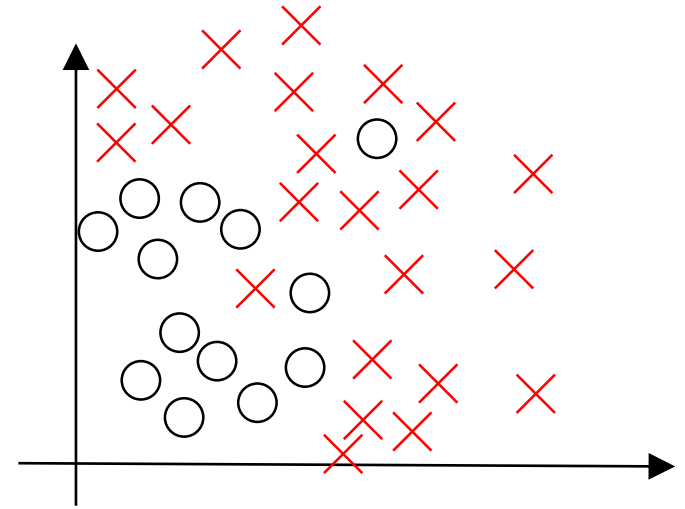
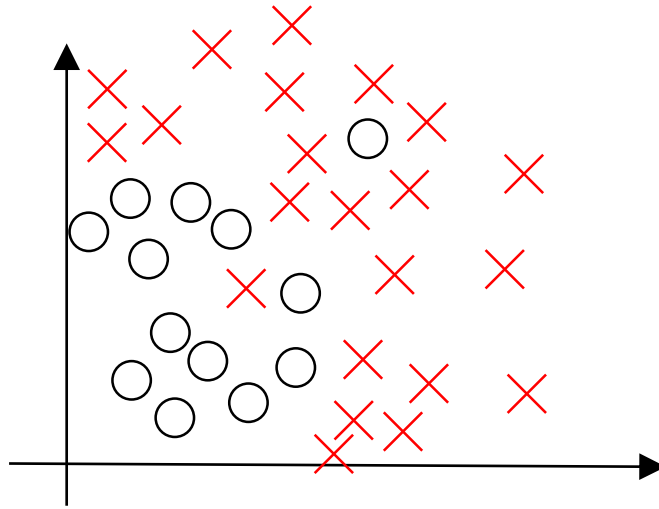
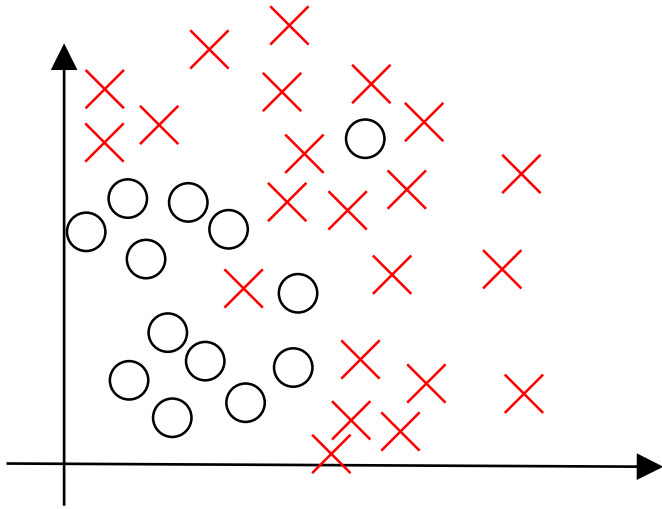
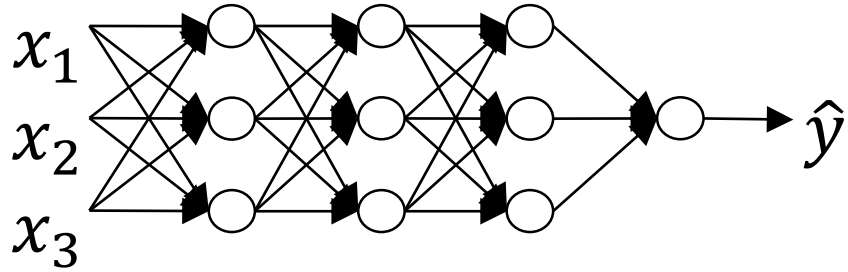
$$w^{[l]} := w^{[l]} - \alpha \left[(\text{from backprop}) + \frac{\lambda}{n} w^{[l]} \right]$$

$$= w^{[l]} - \frac{\alpha \lambda}{n} w^{[l]} - \alpha (\text{from backprop})$$

$$= \underbrace{\left(1 - \frac{\alpha \lambda}{n}\right)}_{\leq 1} \underbrace{w^{[l]}}_{\text{weight}} - \alpha (\text{from backprop})$$

So you're just multiplying the weight metrics by a number slightly less than 1

How does regularization prevent overfitting?



How does regularization prevent overfitting?