Bias / Variance

High Bias ? → Bigger DataSet

(training data performance) $\xrightarrow{Y}$ train longer

↓ N

High Variance ?

(validation set performance) → More data

→ Regularization

↓ N Y

Done

Regularization

logistic Regression

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\|w\|_2^2$$

L_2 Regularization : $\|w\|_2^2 = w^T w$

L_1 Regularization : $\frac{\lambda}{2m}\sum_{j=1}^{n_x}|w_j| = \frac{\lambda}{2m}\|w\|_1$ } w will be sparse

⇓

help a little to compress model

Neural Network

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

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

w $n^{[l-1]} \times n^{[l]}$

# of hidden

"Forbenius norm"

unit in l-1

$$dw^{[l]} = \frac{\partial J}{\partial W^{[l]}} + \frac{\lambda}{m} w^{[l]}$$

$\rightarrow w^{[l]} = w^{[l]} - \alpha \, dw^{[l]}$ $\Rightarrow$ from backprop

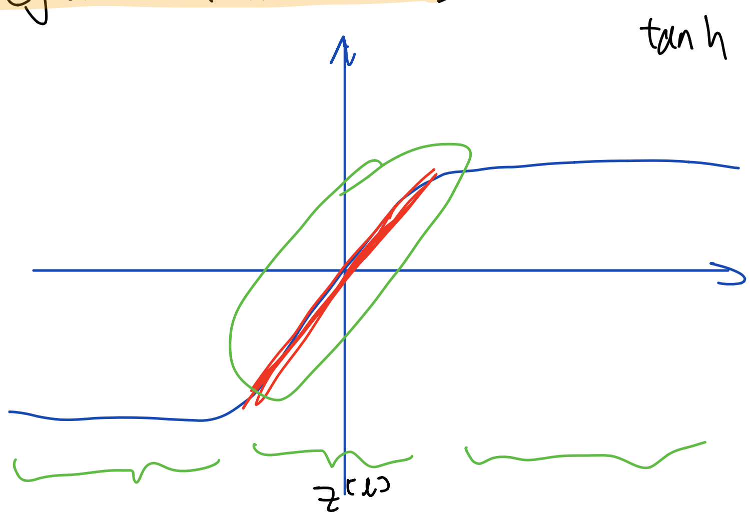
$\Rightarrow$ Weight Decay:

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

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

$$= \underbrace{\left(1 - \frac{\alpha \lambda}{m}\right) w}_{\text{always gets smaller}} - \alpha (\text{from backprop})$$

Why Regularization Works?



$\lambda \uparrow$ $\quad w^{[l]} \downarrow$ $\quad z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$

force the weight to be closer to linear.

therefore reduce the complexity of Network.

Dropout Regularization (No Dropout at test time)

randomly turning off hidden unit during training.

Implementation : "inverted dropout"

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

← zero out 20%

then /= 0.8

Why Dropout Works?

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

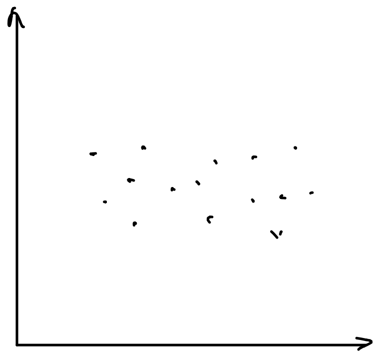
similar to L2.

[When debugging gradient descent: turn off dropout and see if gradient monotonically decreasing]

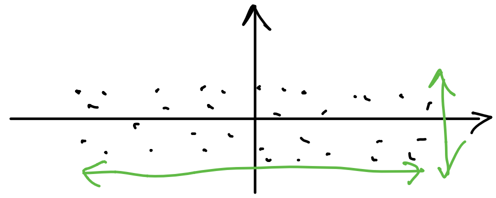
Other method :

- Data augmentation
- Early Stopping → orthognolization

Normalizing Input



⇒

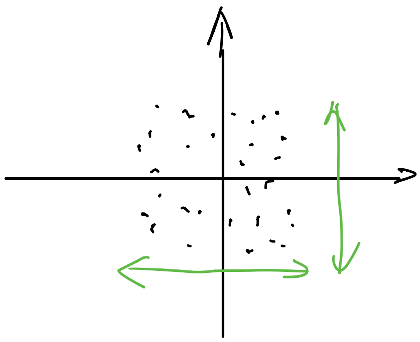


subtract mean

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$x \mathrel{-}= \mu$$

⇙



Normalize Variance

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} {*}{*} 2$$

$$x \mathrel{/}= \sigma^2$$