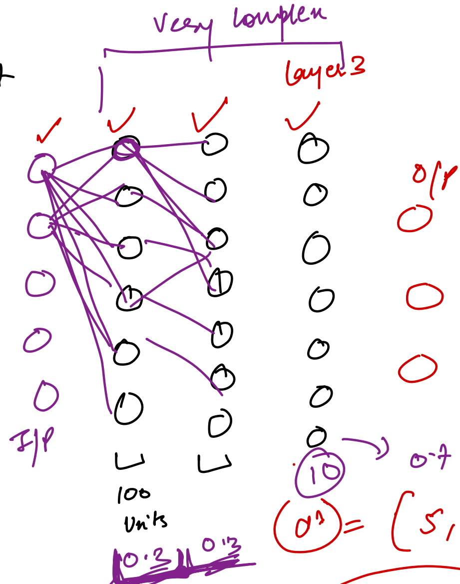
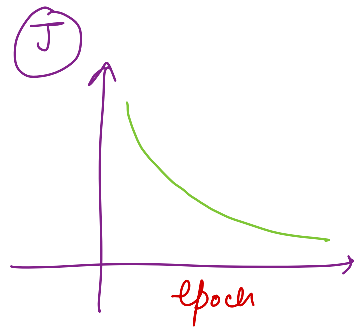
Regularization

Dropout



$0^3 = (5, 3)$



Iteration — $\dfrac{\text{Training set}}{\text{Batch size}} = \dfrac{10000}{1000}$

10 → 14 ⇒

Inverted dropout

layer = 3

keep_prob = 0.7

d_3 = np.random.rand(a_3.shape[0], a_3.shape[1]) < keep_prob

True

$$\begin{bmatrix} 0.9 & 0.7 & 0.6 \\ 0.8 & 0.9 & 0.5 \\ 0.3 & 0.4 & 0.6 \end{bmatrix} \geq 0.7$$

Matrix

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

a_3 = np.multiply(a_3, d_3)

a_3 = a_3 / keep_prob

$$a_3 / 0.7 = \frac{a_3 \times 10}{7}$$

L1/L2 Regularization → stopping the weights from becoming large.

Dropouts → minimizing the no. of parameters.

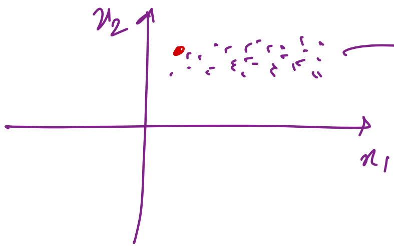
→ complexity ↓

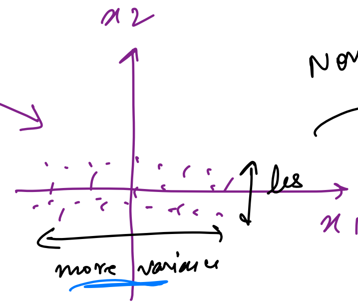
Stopping N/W from overfitting

→ Every weight has to be learned in a way, that even in the absence of a few connection, you are making better predictions.

Batch normalization

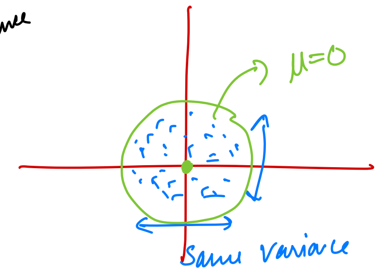
Standardization





Normalize the variance

$X - \mu$



Subtract the mean

X, μ

$X = X - \mu$

$Z = X - \mu$

take element wise

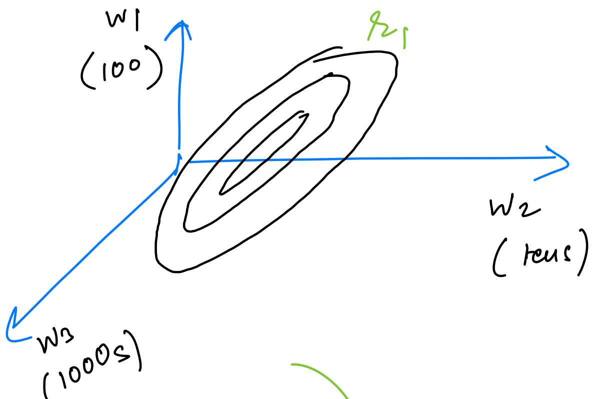
$$\frac{X^2}{\sigma^2 + \varepsilon}$$

$$Z = \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} \Rightarrow Z_{norm}$$

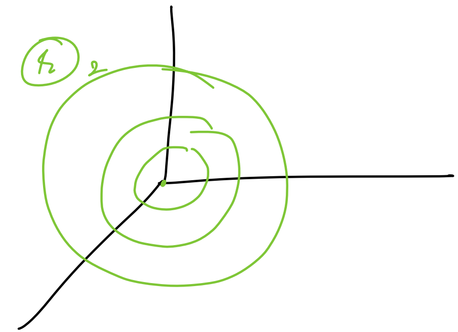
Std => 0

$\varepsilon \Rightarrow 10^{-8}$

MNIST → $X/255$ → Accuracy ↑



$\mu \neq 0$
SD $\neq 1$



$lr_2 < lr_1$

Znorm ⇒ $\mu = 0$
$\sigma^2 = 1$

layer

$$\hat{z} = \gamma \times z_{norm}^{(i)} + \beta$$

2 1

Batch Normalization

adam

gd with momentum

RMS prop