Computational Intelligence

Samaneh Hosseini

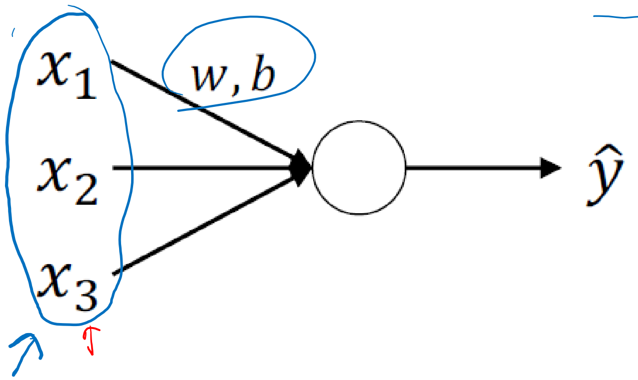
Isfahan University of Technology

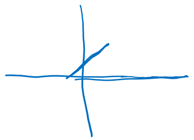
Outline

- Batch Normalization
  - Normalizing Activations in a Network
  - Fitting Batch Norm into a Neural Network
  - Why does Batch Norm work?
  - Batch Norm at Test Time

Batch Normalization:
Normalizing activations in a network

Normalizing inputs to speed up learning





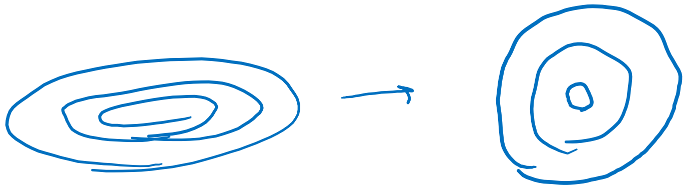
$$\mu = \frac{1}{m}\sum_{i=1}^{m} X^{(i)}$$

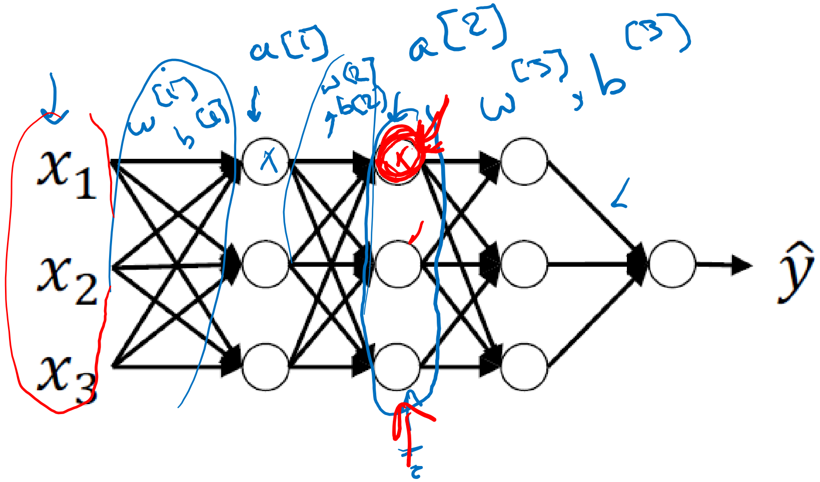
$$X = X - \mu$$

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

element.wise

$$X = X/\sigma^2$$





Can we normalize $a^{[2]}$ so as to train $w^{[3]}, b^{[3]}$ faster

Normalize $z^{[2]}$

64, 128, 512

Implementing Batch Norm

Given some intermediate value in NN $\quad \boxed{z^{[l](1)} \; \dots \; z^{[l](m)}} \rightarrow z^{[l](i)}$

$$\mu = \frac{1}{m} \sum_i z^{(i)}$$

$$\sigma^2 = \frac{1}{m} \sum_i (z_i - \mu)^2$$

$$z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

$$\rightarrow \tilde{z}^{(i)} = \underbrace{\gamma}\, z^{(i)}_{norm} + \underbrace{\beta}$$

learnable parameter of Model

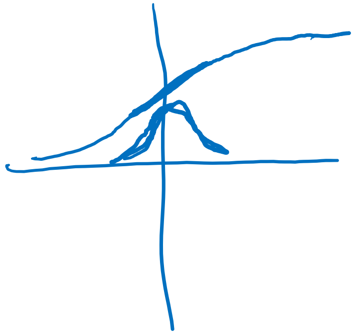
If

$$\gamma = \sqrt{\sigma^2 + \varepsilon}$$

$$\beta = \mu$$

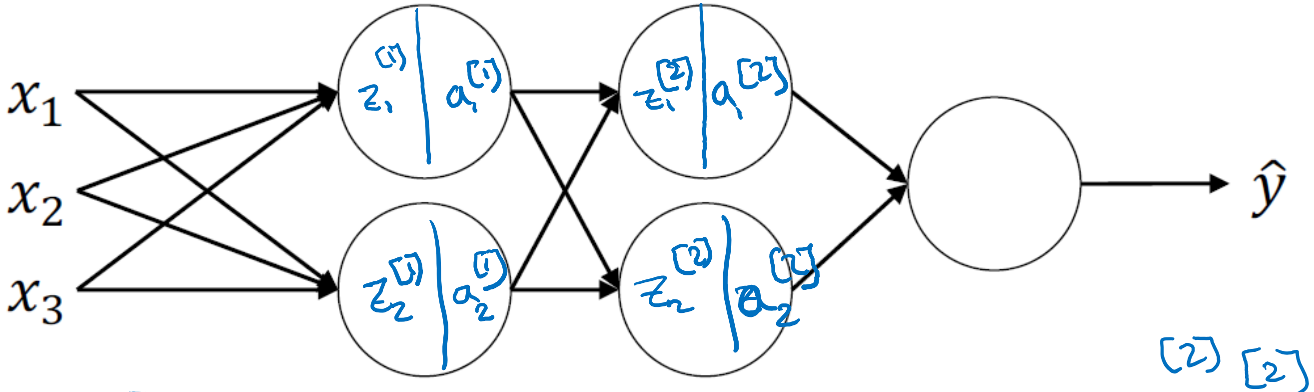
then $\tilde{z}^{(i)} = z^{(i)}$

use $\tilde{z}^{[l](i)}$ instead of $z^{(i)}$



Batch Normalization
Fitting Batch Norm into a Neural Network

Adding Batch Norm to a network



$$X \xrightarrow{w^{[1]}, b^{[1]}} z^{[1]} \xrightarrow[\text{Batch Norm (BN)}]{\beta^{[1]}, \gamma^{[1]}} \tilde{z}^{[1]} \longrightarrow a^{[1]} = g^{[1]}(\tilde{z}^{[1]}) \xrightarrow{w^{[2]}, b^{[2]}} z^{[2]} \xrightarrow[BN]{\beta^{[2]}, \gamma^{[2]}} \tilde{z}^{[2]} \rightarrow a^{[2]} \ldots$$

$$\left.\begin{array}{l} w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}, \ldots, w^{[L]}, b^{[L]} \\ \beta^{[1]}, \gamma^{[1]}, \beta^{[2]}, \gamma^{[2]}, \quad \beta^{[L]}, \gamma^{[L]} \end{array}\right\} \quad d\beta^{[l]} \qquad \beta^{[l]} = \beta^{[l]} - \alpha d\beta^{[l]}$$

tf.nn.batch_normalization(-)

Working with mini-batches

batch $\rightarrow X^{\{1\}} \xrightarrow{W^{[1]}, b^{[1]}} Z^{[1]} \xrightarrow{\beta^{[1]}, \gamma^{[1]}} \tilde{Z}^{[1]} \rightarrow g^{[1]}(\tilde{Z}^{[1]}) = a^{[1]} \xrightarrow{W^{[2]}, b^{[2]}} Z^{[2]} \leftarrow - - -$

batch $\rightarrow X^{\{2\}} \rightarrow Z^{[1]} \xrightarrow[BN]{\beta^{[1]}, \gamma^{[1]}} \tilde{Z}^{[1]} \rightarrow - - -$

$\rightarrow X^{\{\}} \rightarrow$

$W^{[l]}, \cancel{b^{[l]}}, \beta^{[l]}, \gamma^{[l]}$

$Z^{[l]} \quad (n^{[l]}, 1)$

$\beta^{[l]} \downarrow (n^{[l]}, 1)$

$\gamma^{[l]} \downarrow (n^{[l]}, 1)$

$Z - \mu$

$Z^{[l]} = W^{[l]} a^{[l-1]} + \boxed{\cancel{b^{[l]}}}$

$Z^{[l]} = W^{[l]} a^{[l-1]}$

$\rightarrow Z^{[l]}_{norm}$

$\tilde{Z}^{[l]} = \gamma^{[l]} Z^{[l]}_{norm} + \boxed{\beta^{[l]}}$

Implementing gradient descent

for $t = 1$ .... num MiniBatches

compute forward prop on $X^{\{t\}}$

In each hidden layer, use BN to replace $Z^{[l]}$ with $\tilde{Z}^{[l]}$

Use backprop to compute $dw^{[l]}$, $\cancel{db^{[l]}}$, $d\beta^{[l]}$, $d\gamma^{[l]}$

update prameters

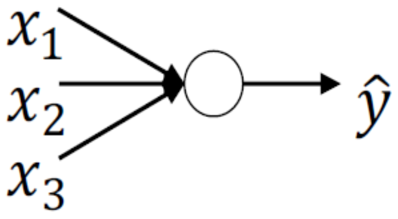
$$\left.\begin{aligned} w^{[l]} &:= w^{[l]} - \alpha dw^{[l]} \\ \beta^{[l]} &:= \beta^{[l]} - \alpha d\beta^{[l]} \\ \gamma^{[l]} &:= \dots \end{aligned}\right\}$$

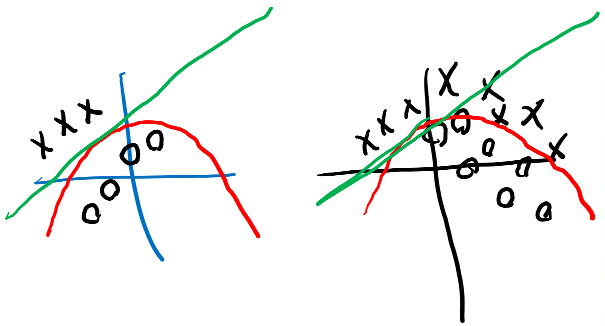
tf . Keras . Layers . BatchNormalization (...)

RMSProb , momentum , Adam

Batch Normalization
Why does Batch Norm work?

Learning on shifting input distribution





Cat

$y = 1$

Non-Cat

$y = 0$

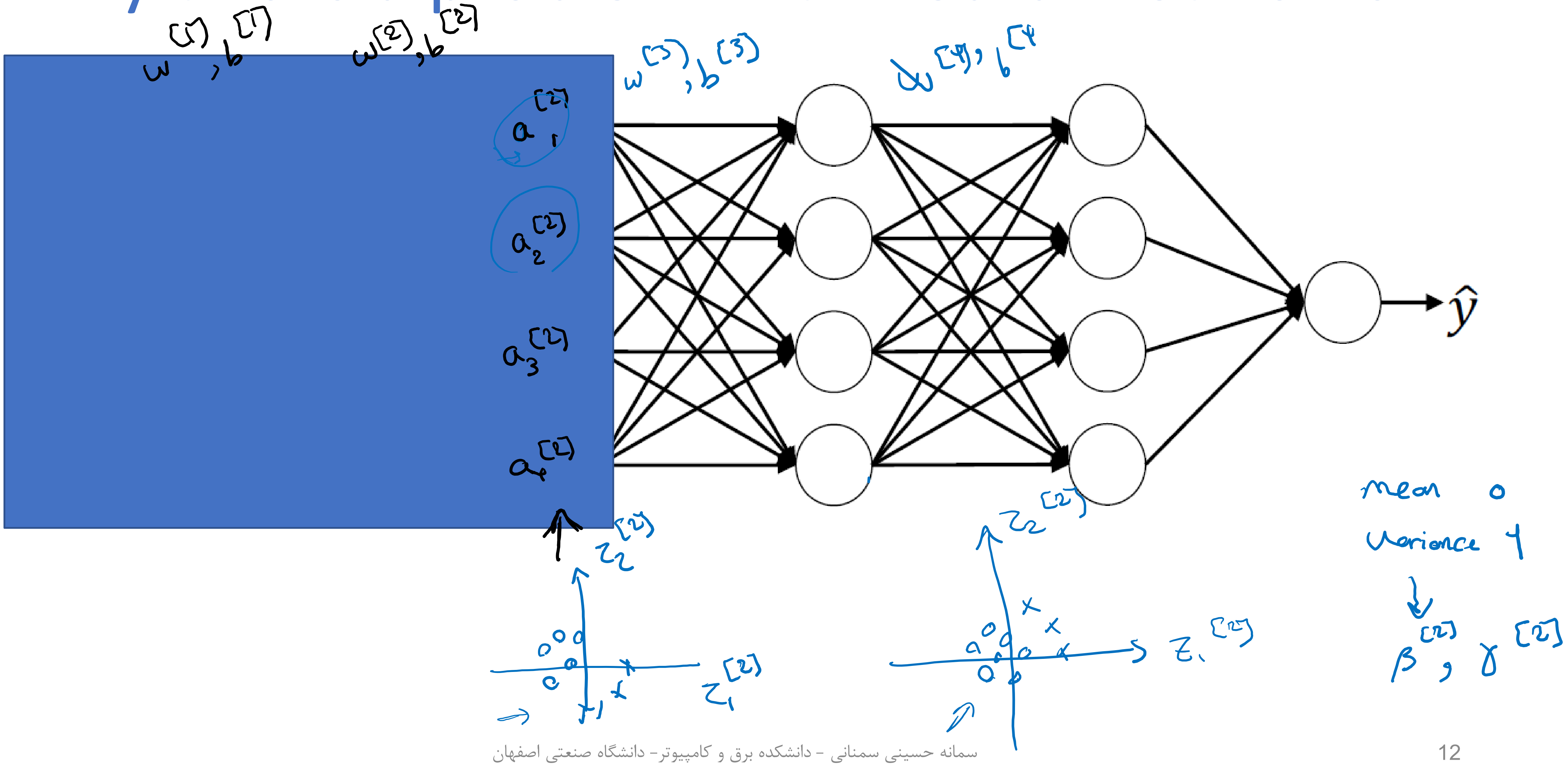
Covariant shift

$X \rightarrow y$

$y = 1$

$y = 0$

Why this is a problem with neural networks?



Batch Norm as regularization

- Each mini-batch is scaled by the mean/variance computed on just that mini-batch.

$\mu^{\{1\}}, \sigma^{\{1\}}$

$\mu^{\{2\}}, \sigma^{\{2\}}$

- This adds some noise to the values +[-] within that minibatch.
- So similar to dropout, it adds some noise to eachhidden layer's activations.
- This has a slight regularization effect.

0 1 σ $+\mu$

Min-batch : 64 $\longrightarrow$ 512

noise ↑ noise ↓

Batch Normalization
Batch Norm at Test Time

exponentially weighted average

Batch Norm at Test Time

$$\mu = \frac{1}{m}\sum_i z^{(i)}$$

$$\sigma^2 = \frac{1}{m}\sum_i (z^{(i)}-\mu)^2$$

$$z_{\text{norm}}^{(i)} = \frac{z^{(i)}-\mu}{\sqrt{\sigma^2+\varepsilon}}$$

$$\tilde{z}^{(i)} = \gamma z_{\text{norm}}^{(i)} + \beta$$

$X^{\{1\}}, \quad X^{\{2\}}, \quad X^{\{3\}}, \quad \ldots$

$\mu^{\{1\}[l]} \quad \mu^{\{2\}[l]} \quad \mu^{\{3\}[l]}$

$\theta_1 \quad \theta_2 \quad \theta_3 \quad \ldots \to \mu, \sigma^2$

$\sigma^{2\{1\}[l]} \quad \sigma^{2\{2\}[l]} \quad \ldots$

$$z_{norm} = \frac{z-\mu}{\sqrt{\sigma^2+\varepsilon}}$$

$$\tilde{z} = \gamma z_{norm} + \beta$$

Core Foundation Review

- Batch Normalization
  - Normalizing Activations in a Network
  - Fitting Batch Norm into a Neural Network
  - Why does Batch Norm work?
  - Batch Norm at Test Time