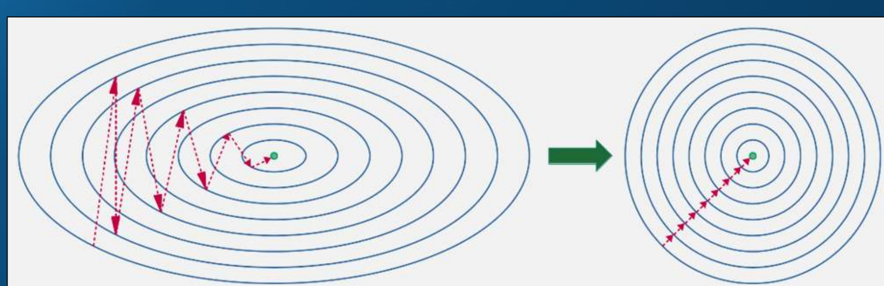
Batch Normalization

Deep Neural Network
Session 13
Pramod Sharma
pramod.sharma@prasami.com

Batch Normalization

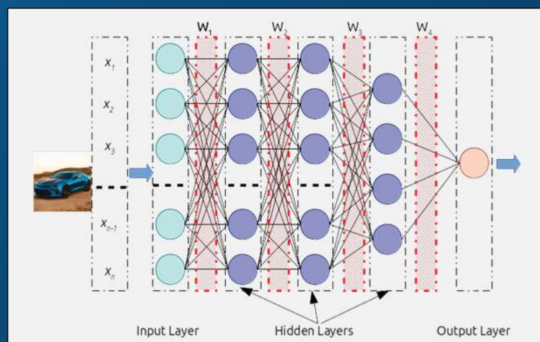
- It definitely helps to normalize input data
- Gradient converges faster



3

Batch Normalization

- What about hidden layer?
- After all activations from previous layer are inputs for current layer…



- Will it help if we normalize the hidden layers too?

4

Batch Normalization

- Batch normalization (also known as batch norm) [by Sergey Loffe and Christian Szegedy in 2015]
  - Make artificial neural networks faster
  - More stable through normalization of the input layer by re-centering and re-scaling
  - Wider choices of hyper- parameter…
- In theory, its normalizing activation values of the respective layers
- In practice, it works better if we normalize ‘z’
  - Look at the documentation for details

5

Batch Normalization

- In General, any Z^i can be normalized

  mean $\mu = \frac{\Sigma Z^i}{m}$

  std $\sigma^2 = \frac{1}{m}\Sigma(Z^i - \mu)^2$

- $Z^i Norm = \frac{z^i - \mu}{\sqrt{\sigma^2}}$
- $\hat{z} = \gamma . Z^i$ Norm + β
- where γ *and* β *are paramters we can tune*
- if $\gamma = \frac{1}{\sqrt{(\sigma^2)}}$ and $\beta = \frac{\mu}{\sqrt{\sigma^2}}$; Z^i Norm $= \hat{z}$

11/25/2024

pra-sāmi

6

Batch Normalization

- In General, any Z^i can be normalized

  mean $\mu = \frac{\Sigma Z^i}{m}$

  std $\sigma^2 = \frac{1}{m}\Sigma(Z^i - \mu)^2$

  $z^i_{Norm} = \frac{z^i - \mu}{\sqrt{\sigma^2}}$

  $\hat{z} = \gamma . z^i_{Norm} + \beta$

- where γ and β are parameters, we can **Train**
- if $\gamma = \frac{1}{\sqrt{\sigma^2}}$ and $\beta = \frac{\mu}{\sqrt{\sigma^2}}$; $Z^i_{Norm} = \hat{z}$

Instead of using $z^i_{Norm,}$ researchers realized that its better to derive $\hat{z}$ with two trainable parameters.

Intuition is that by normalizing z, we are introducing bias in the system. Hence it makes sense to train these parameters

11/25/2024

pra-sāmi

7

Batch Normalization

- In General, any Z^i can be normalized
- mean $\mu = \frac{\sum Z^i}{m}$
- std $\sigma^2 = \frac{1}{m}\Sigma(Z^i - \mu)^2$
  - Z^iNorm $= \frac{Z^i - \mu}{\sqrt{\sigma^2 + \varepsilon}}$
  - $\hat{z} = \gamma . Z^i$ Norm $+ \beta$
- where γ and β are parameters, we can train
- if $\gamma = \frac{1}{\sqrt{\sigma^2 + \varepsilon}}$ and $\beta = \frac{\mu}{\sqrt{\sigma^2 + \varepsilon}}$; Z^i Norm $= \hat{z}$

Lets add a small ε to prevent zero divide error…

11/25/2024

pra-sâmi

8

Batch Normalization

- In General, any Z^i can be normalized
- mean $\mu = \frac{\sum Z^i}{m}$
- std $\sigma^2 = \frac{1}{m}\Sigma(Z^i - \mu)^2$
  - Z^i Norm $= \frac{\text{z^i} - \mu}{\sqrt{\sigma 2 + \varepsilon}}$
  - $\hat{z} = \gamma$. zi Norm $+ \beta$
- where γ and β are parameters, we can train
- if $\gamma = \frac{1}{\sqrt{\sigma 2 + \varepsilon}}$ and $\beta = \frac{\mu}{\sqrt{\sigma 2 + \varepsilon}}$; zi Norm $= \hat{z}$i

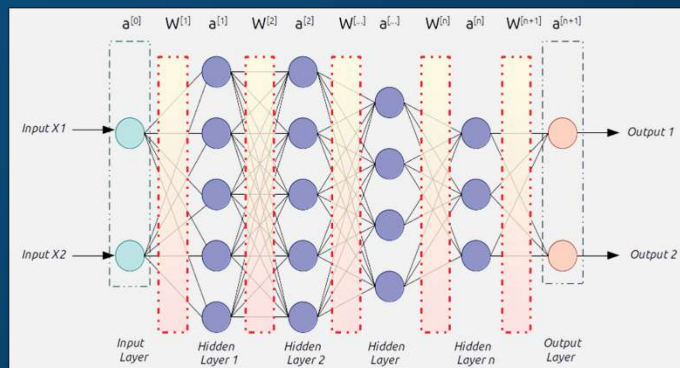
11/25/2024

pra-sâmi

9

Batch Normalization

- Notes:
  - Batch norm is used along with mini batches
  - Batch norm is applied to the batch under consideration only irrespective of other mini batches
- Where does it fit in overall scheme?

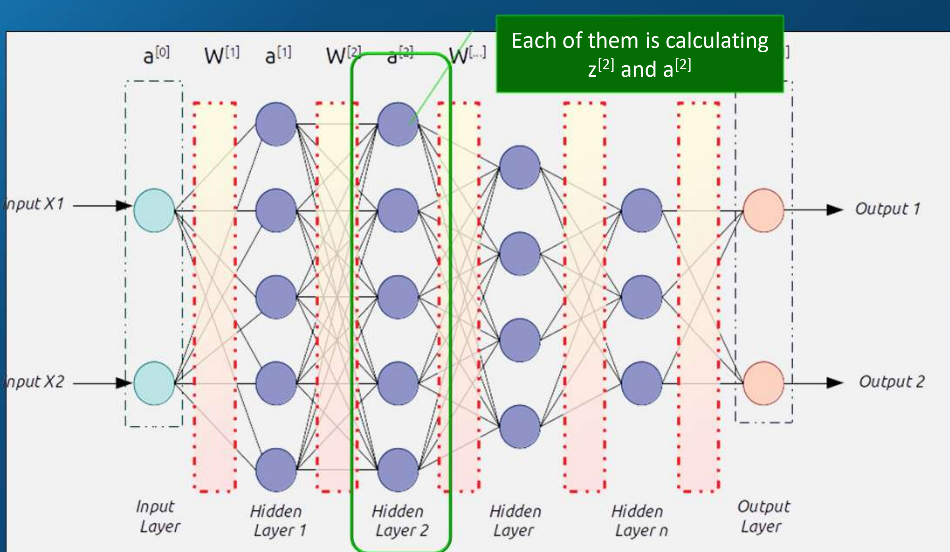


11/25/2024

pra-sâmi

10

Batch Normalization



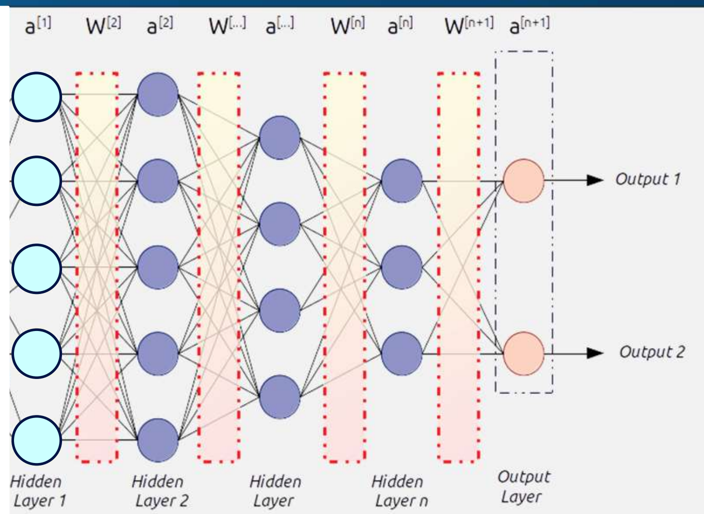
11/25/2024

pra-sâmi

11

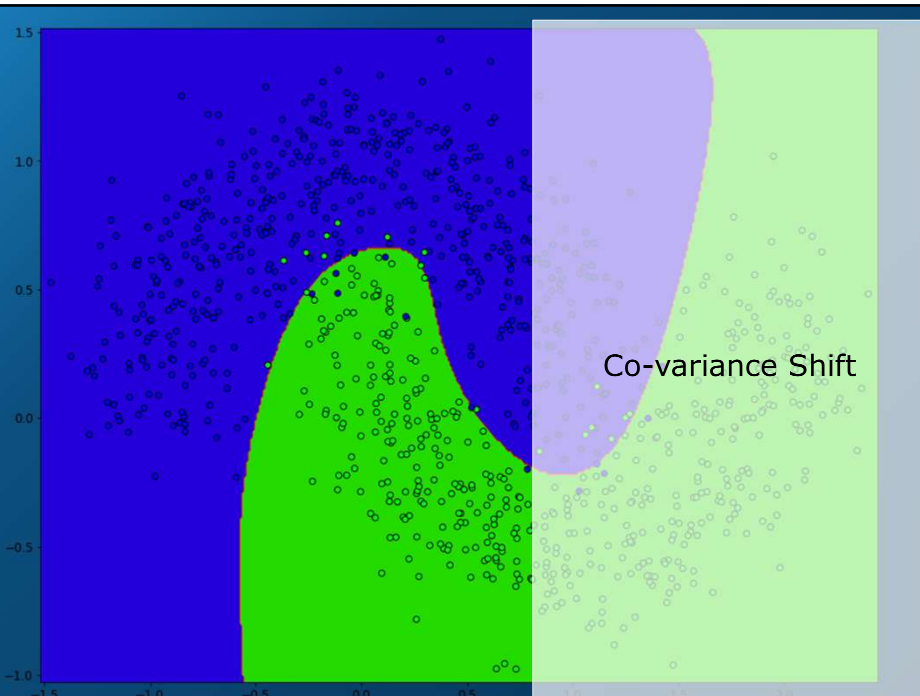
Batch Normalization

For $a^{[2]}$ all $a^{[1]}$ are acting as input features



11/25/2024

12

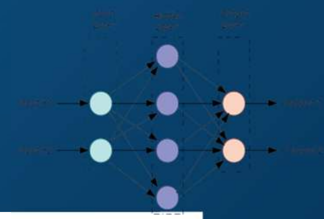


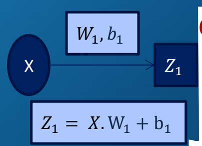
11/25/2024

13

Batch Normalization

- Forward and back propagation with batch norm:





Our standard equation to calculate z_1.

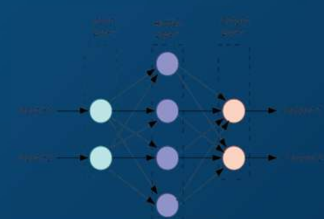
11/25/2024

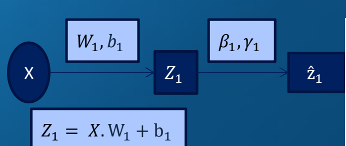
pra-sâmi

14

Batch Normalization

- Forward and back propagation with batch norm:





Calculate $\hat{z}_1$, based on β_1, γ_1

11/25/2024

pra-sâmi

15

Batch Normalization

- Forward and back propagation with batch norm:

W_1, b_1 β_1, γ_1

X → Z_1 → $\hat{z}_1$ → $a_1 = g_1(\hat{z}_1)$

$Z_1 = X.\mathrm{W}_1 + \mathrm{b}_1$

Apply activation function $g_1(\hat{z}_1)$

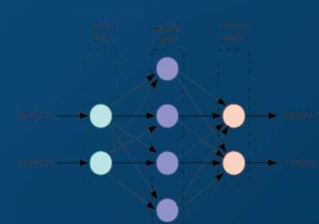
11/25/2024

pra-sâmi

16

Batch Normalization

- Forward and back propagation with batch norm:



W_1, b_1 β_1, γ_1 $\mathrm{W}_2, \mathrm{b}_2$

X → Z_1 → $\hat{z}_1$ → $a_1 = g_1(\hat{z}_1)$ → z_2

$Z_1 = X.\mathrm{W}_1 + \mathrm{b}_1$ $z_2 = a_1.\mathrm{W}_2 + \mathrm{b}_2$

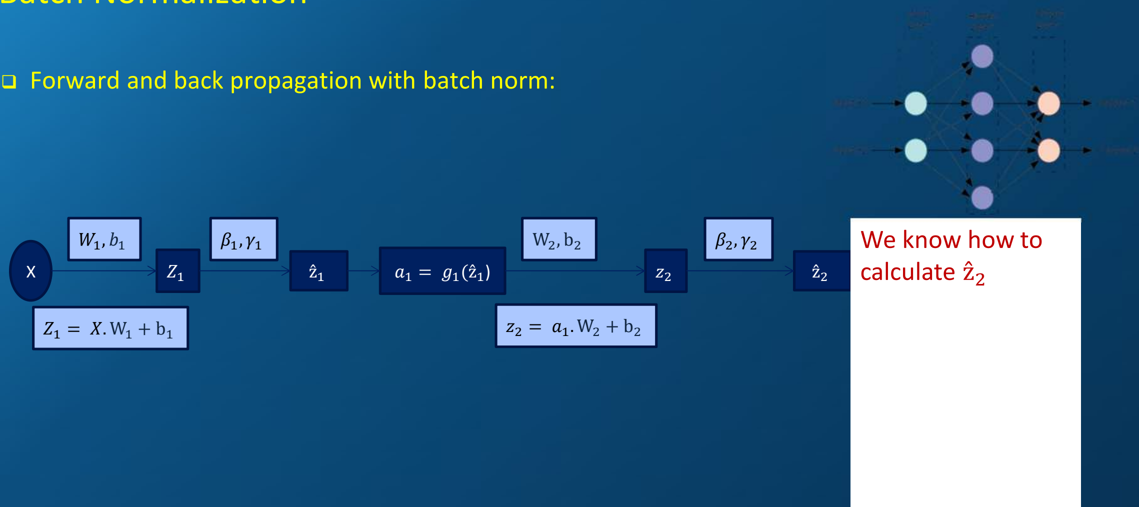
Calculate z_2 as usual

11/25/2024

pra-sâmi

17
Batch Normalization

- Forward and back propagation with batch norm:

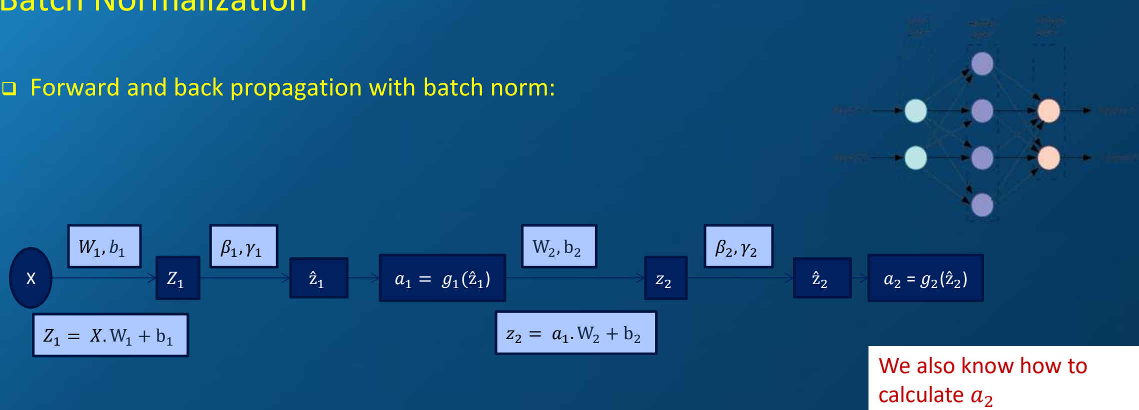


11/25/2024

pra-sâmi

18
Batch Normalization

- Forward and back propagation with batch norm:



11/25/2024

pra-sâmi

19

Batch Normalization

- Forward and back propagation with batch norm:

$X \rightarrow Z_1 \rightarrow \hat{z}_1 \rightarrow a_1 = g_1(\hat{z}_1) \rightarrow z_2 \rightarrow \hat{z}_2 \rightarrow a_2 = g_2(\hat{z}_2)$

W_1, b_1 ; β_1, γ_1 ; W_2, b_2 ; β_2, γ_2

$Z_1 = X.W_1 + b_1$

$z_2 = a_1.W_2 + b_2$

$\beta_1 = \beta_1 - \alpha . \partial\beta_1$
$\gamma_1 = \gamma_1 - \alpha . \partial\gamma_1$

Using the gradient descent, update β's, γ's along with W's and b's

$\beta_2 = \beta_2 - \alpha . \partial\beta_2$
$\gamma_2 = \gamma_2 - \alpha . \partial\gamma_2$

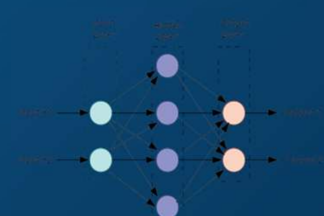
11/25/2024

pra-sâmi

20

Batch Normalization

- Forward and back propagation with batch norm:



$X \rightarrow Z_1 \rightarrow \hat{z}_1 \rightarrow a_1 = g_1(\hat{z}_1) \rightarrow z_2 \rightarrow \hat{z}_2 \rightarrow a_2 = g_2(\hat{z}_2)$

$W_1, \cancel{b_1}$; β_1, γ_1 ; $W_2, \cancel{b_2}$; β_2, γ_2

$Z_1 = X.W_1 \cancel{+ b_1}$

$z_2 = a_1.W_2 \cancel{+ b_2}$

$\beta_1 = \beta_1 - \alpha . \partial\beta_1$
$\gamma_1 = \gamma_1 - \alpha . \partial\gamma_1$

$\beta_2 = \beta_2 - \alpha . \partial\beta_2$
$\gamma_2 = \gamma_2 - \alpha . \partial\gamma_2$

One more thing, since we are normalizing our Z's, keeping b's in the equation does not make any sense now. Being the constant it will get eliminated!!

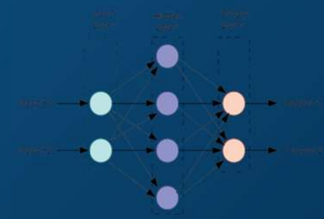
11/25/2024

pra-sâmi

21

Batch Normalization

- Forward and back propagation with batch norm:



$X \rightarrow Z_1 \rightarrow \hat{z}_1 \rightarrow a_1 = g_1(\hat{z}_1) \rightarrow z_2 \rightarrow \hat{z}_2 \rightarrow a_2 = g_2(\hat{z}_2)$

$W_1, \cancel{b_1}$ $\quad \beta_1, \gamma_1$ $\quad W_2, \cancel{b_2}$ $\quad \beta_2, \gamma_2$

$Z_1 = X.W_1 \cancel{+ b_1}$

$z_2 = a_1.W_2 \cancel{+ b_2}$

$\beta_1 = \beta_1 - \alpha . \partial \beta_1$
$\gamma_1 = \gamma_1 - \alpha . \partial \gamma_1$

$\beta_2 = \beta_2 - \alpha . \partial \beta_2$
$\gamma_2 = \gamma_2 - \alpha . \partial \gamma_2$

And at test/validation time using a exponentially weighted average!
So while training do not forget to save exponentially weighted values or simply running average!!

11/25/2024

pra-sâmi

22

Batch Normalization

Forward and back propagation with batch norm:

Too many calculation steps…

Don't worry…

Most frameworks have one line code to do it.

11/25/2024

pra-sâmi

23

Batch Normalization – Code Sample

```
model = tf.keras.models.Sequential(
     [
       tf.keras.layers.RNN( keras.layers.LSTMCell(units), input_shape=(None, input_dim)  ),
       tf.keras.layers.BatchNormalization(),
       tf.keras.layers.Dense(output_size),
     ]
   )


class Net(nn.Module):
          def __init__(self):
          super(Net, self).__init__()
          self.dense1 = nn.Linear(in_features=320, out_features=50)
          self.dense1_bn = nn.BatchNorm1d(50)
          self.dense2 = nn.Linear(50, 10)
```

- And it is applied to mini batches only….
- Batch Norm can be updated using any of the optimization functions…

11/25/2024

24

Batch Normalization

Remember β, γ *are parameters you train*!

11/25/2024

25

Reflect…

- What is the primary purpose of Batch Normalization in deep learning?
  - A) To prevent overfitting
  - B) To reduce the number of parameters in the model
  - C) To accelerate training and reduce internal covariate shift
  - D) To increase the depth of the neural network
- Answer: C) To accelerate training and reduce internal covariate shift
- At which stage is Batch Normalization applied in a neural network?
  - A) After the input layer
  - B) After the activation function
  - C) Before the loss calculation
  - D) Before or after the activation function, depending on the implementation
- Answer: D) Before or after the activation function, depending on the implementation
- Which of the following is a key step in Batch Normalization?
  - A) Normalizing the gradient updates
  - B) Normalizing the activations by subtracting the batch mean and dividing by the batch standard deviation
  - C) Initializing weights to zero
  - D) Adding noise to the input data
- Answer: B) Normalizing the activations by subtracting the batch mean and dividing by the batch standard deviation
- What are the two learnable parameters introduced in Batch Normalization?
  - A) Gamma and Beta
  - B) Alpha and Beta
  - C) Theta and Gamma
  - D) Sigma and Mu
- Answer: A) Gamma and Beta

11/25/2024

pra-sâmi

26

Next Session… Recurrent Neural Networks

11/25/2024

pra-sâmi