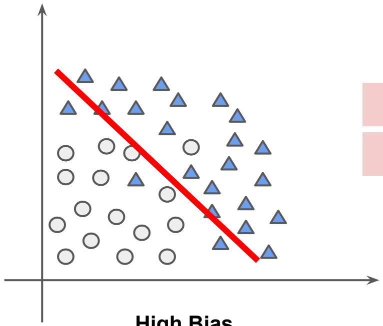
Deep Learning

Week 3

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Bias and Variance



High Bias

Underfitting

Train Error: 15%	Low Accuracy
Test Error: 16%	Low Accuracy

Bigger Network
Train Longer

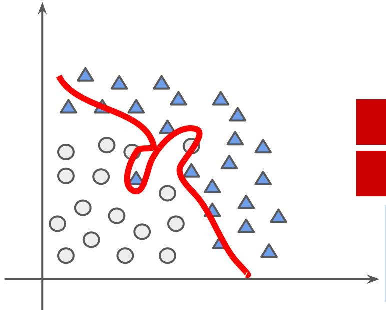
High Variance

Overfitting

Train Error: 1%	High Accuracy
Test Error: 11%	Low Accuracy

More Data
Regularization

Make sure you dev test and training test should come from same distribution



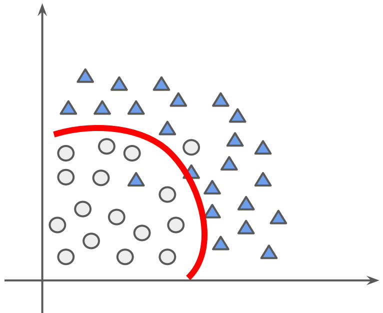
High Bias - High Variance

Train Error: 15%	Low Accuracy
Test Error: 30%	Very Low Accuracy

Bigger Network
Train Longer

More Data
Regularization

Make sure you dev test and training test should come from same distribution



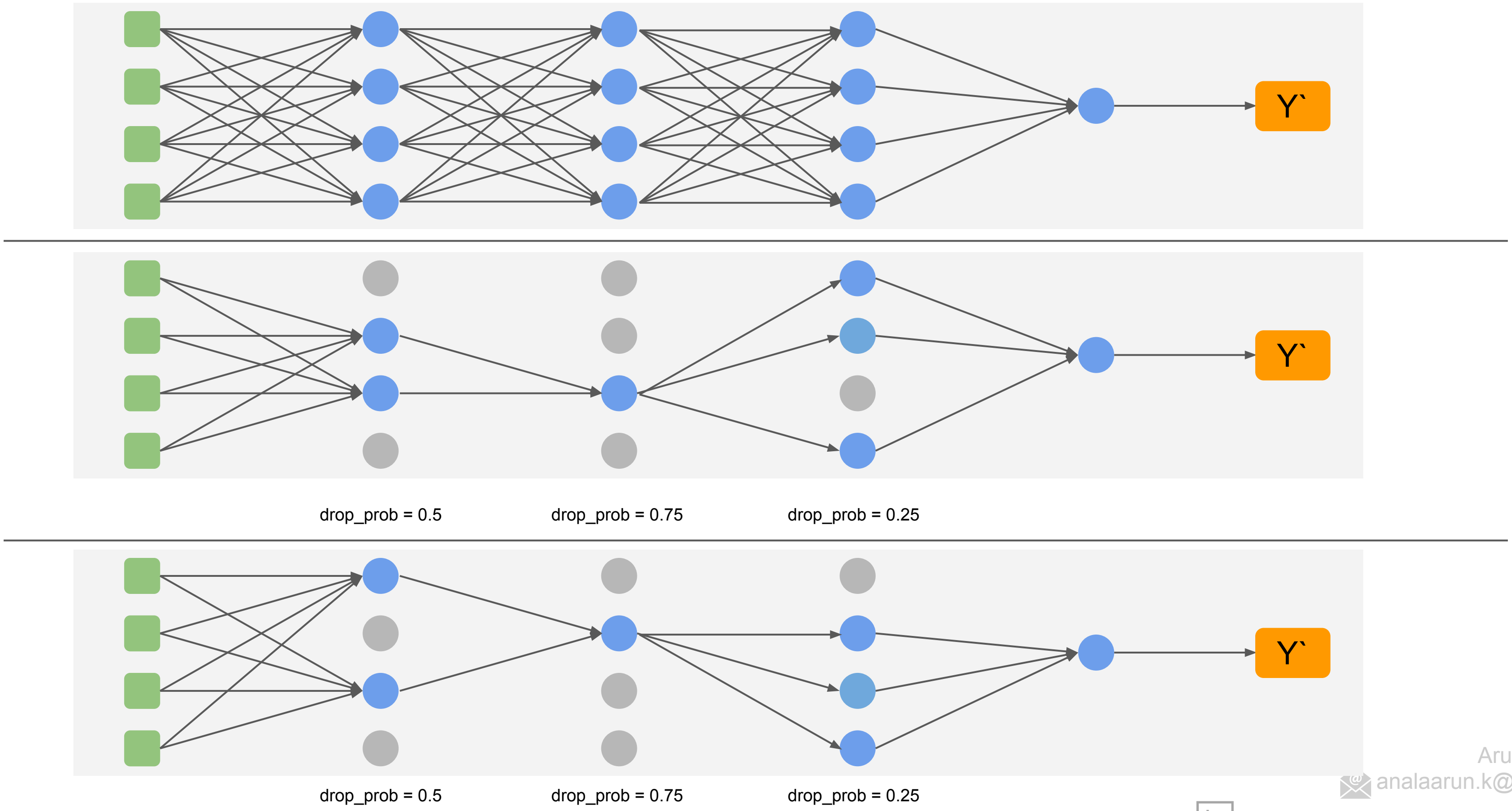
Low Bias - Low Variance

Just Right

Train Error: 0.5%	High Accuracy
Test Error: 1%	High Accuracy

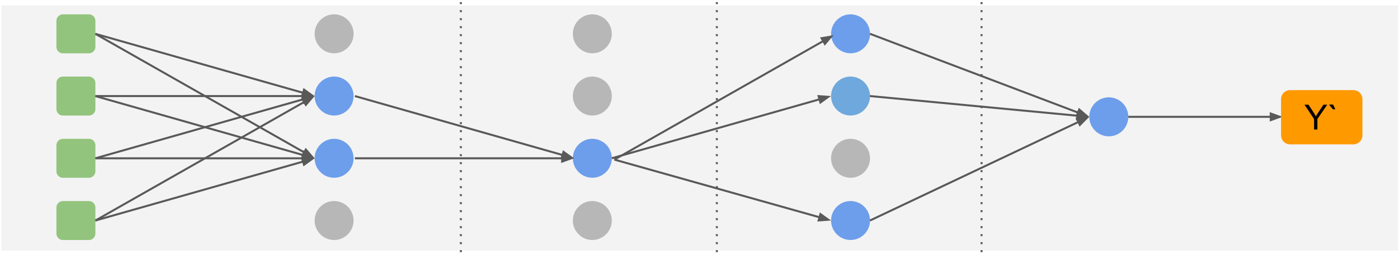
Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Regularization Drop-Out



Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Regularization Drop-Out (Inverted dropout)



$drop\_prob^1 = 0.5$ | $drop\_prob^2 = 0.75$ | $drop\_prob^3 = 0.25$

$keep\_prob^1 = 0.5$ | $keep\_prob^2 = 0.25$ | $keep\_prob^3 = 0.75$

$z^1 = w^1 a^0 + b^1$

$a^{\sim 1} = f^1 (z^1)$

$z^2 = w^2 a^{\sim 1} + b^2$

$z^2 = w^2 a^{\sim 1}/0.5 + b^2$

$a^{\sim 2} = f^2 (z^2)$

$z^3 = w^3 a^{\sim 2} + b^3$

$z^3 = w^3 a^{\sim 2}/0.25 + b^3$

$a^{\sim 3} = f^3 (z^3)$

$z^4 = w^4 a^{\sim 3} + b^4$

$z^4 = w^4 a^{\sim 3}/0.75 + b^4$

$Y` = a^4$

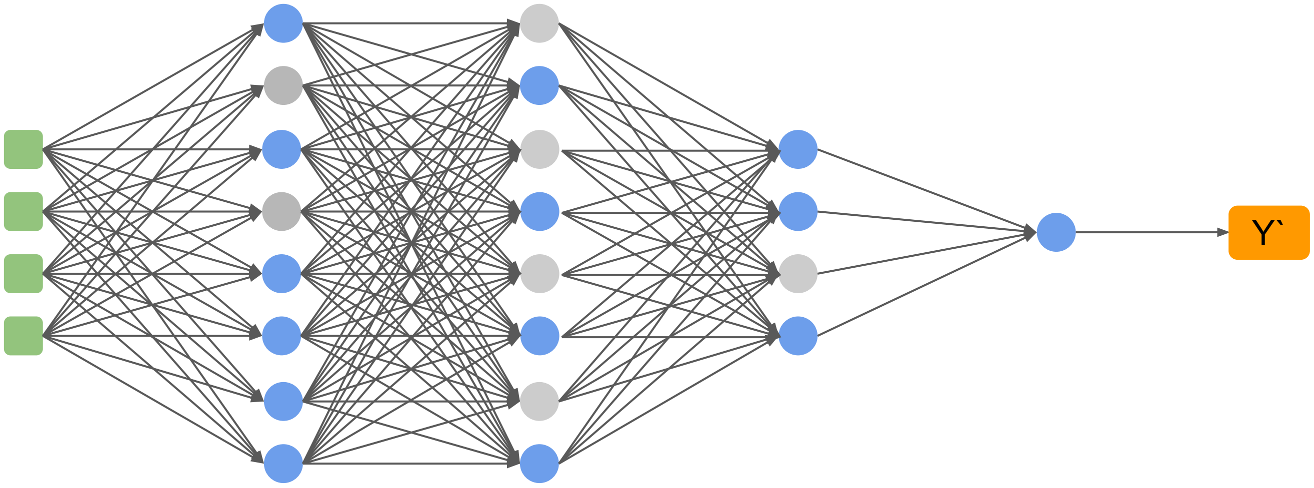
$$\mathbf{z^{[L]} = w^{[L]}\, a^{\sim[L-1]} / keep\_prob^{[L-1]} + b^{[L]}}$$

$$\mathbf{a^{\sim[L]} = f^{[L]} (z^{[L]})}$$

100 * 0.25 = 25
25 / 0.25 = 100

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Regularization Drop-Out (Why it works?)

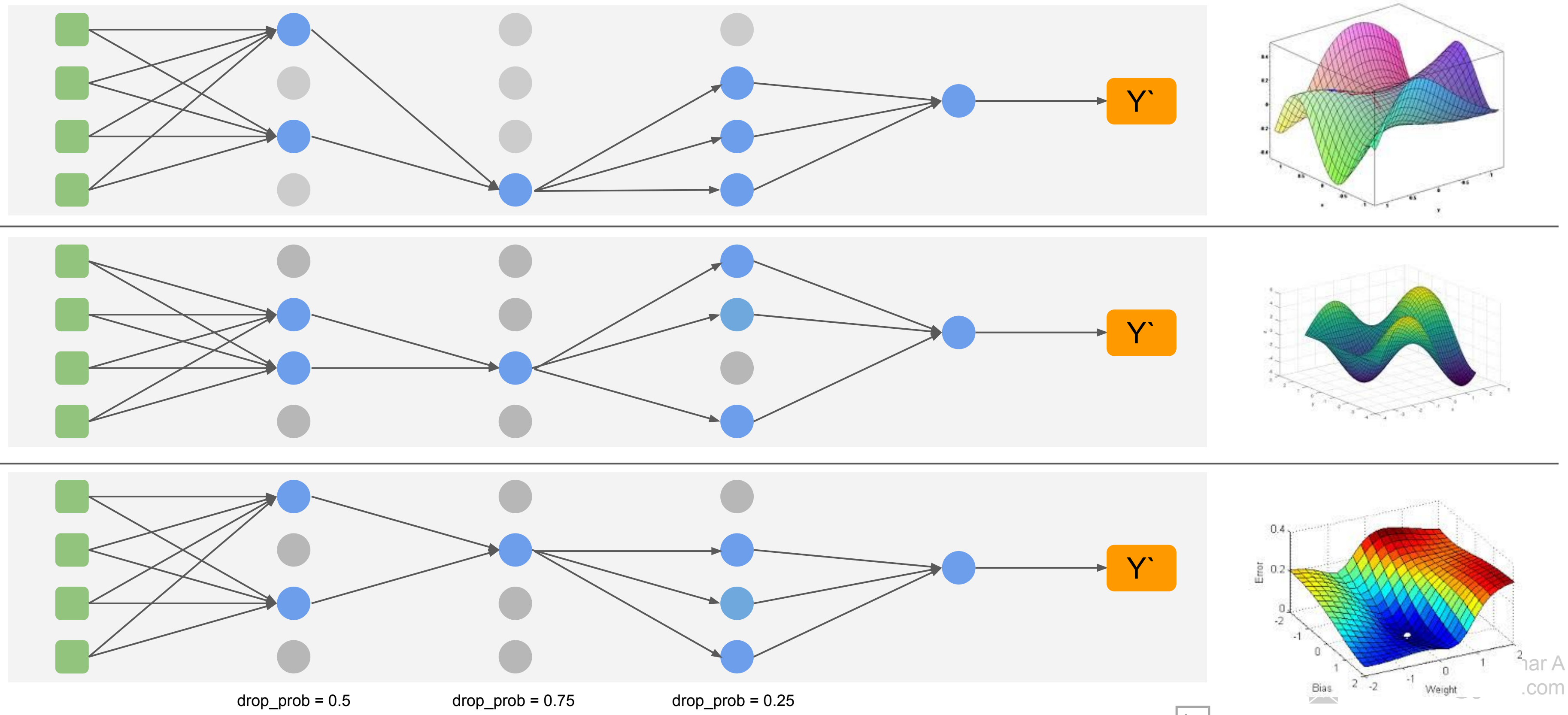


Keep_prob = 0.75 Keep_prob = 0.50 Keep_prob = 0.75 Keep_prob = 1.0

Can’t rely on single feature, so spread out weights.
Reduces overfitting and high variance, since more weights between layers can cause more learning and so overfitting.

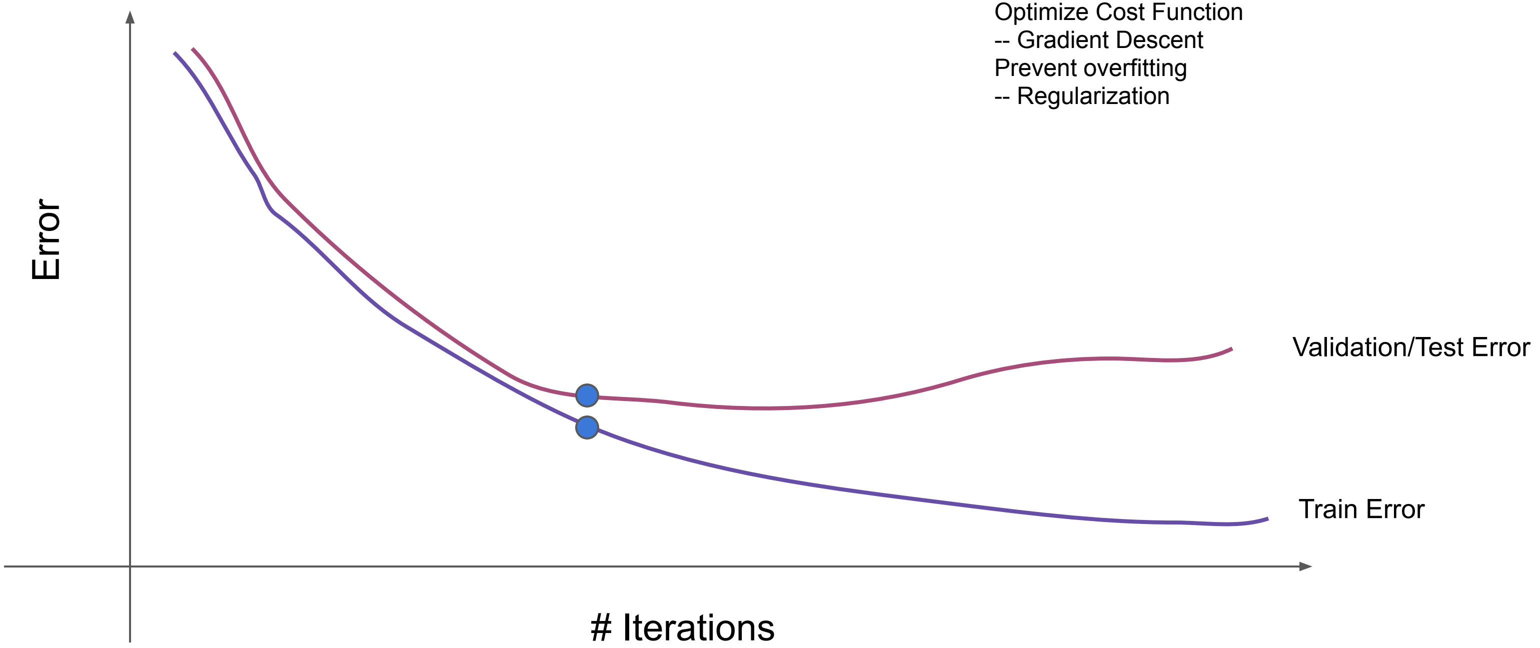
Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Regularization Drop-Out (Downside - Gradient Descent)



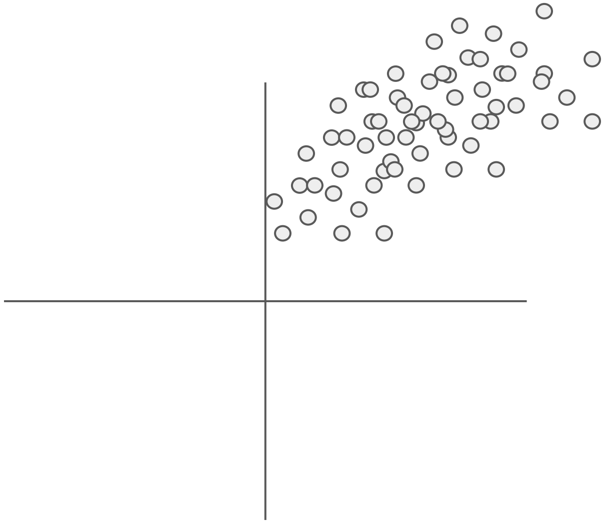
Early Stopping

Optimize Cost Function
-- Gradient Descent
Prevent overfitting
-- Regularization



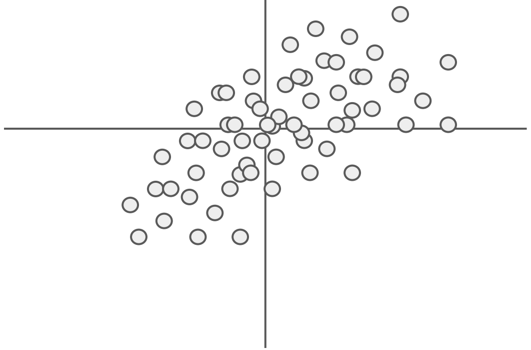
Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Normalizing a distribution



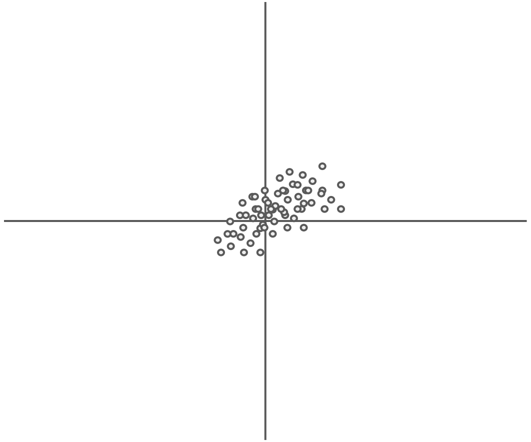
Original Distribution

z^1



Zero Centered Distribution

$(z^1 - \mu)$

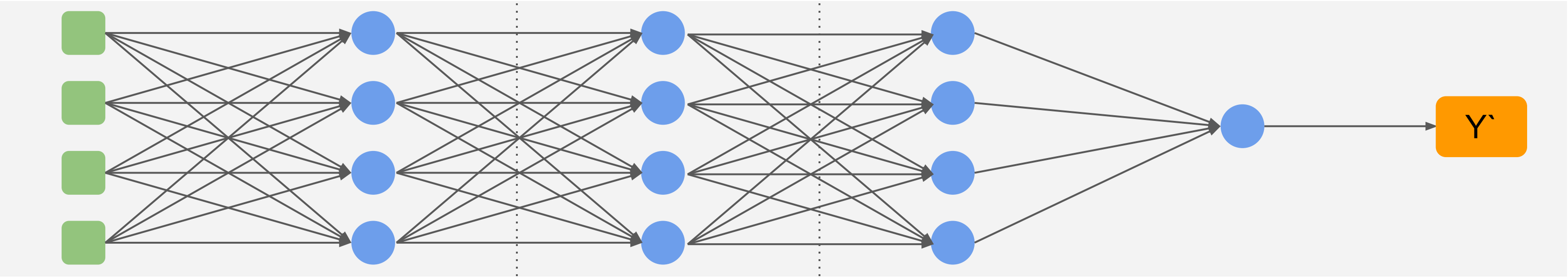


Normalized Distribution

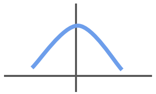
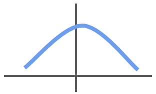
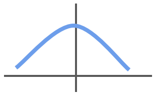
$(z^1 - \mu^1) / \sqrt{(\sigma^2 + \varepsilon)}$

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization



z value	$z^1 = w^1 a^0 + b^1$	$z^2 = w^2 a^1 + b^2$	$z^3 = w^3 a^2 + b^3$	$z^4 = w^4 a^3 + b^4$	Y` = a^4
Batch Statistics / mean	$\mu^1 = 1/m\ \Sigma z^1$	$\mu^2 = 1/m\ \Sigma z^2$	$\mu^3 = 1/m\ \Sigma z^3$		
Non-Trainable Parameters / variance	$\sigma^{2[1]} = 1/m\ \Sigma(z^1 - \mu^1)$	$\sigma^{2[2]} = 1/m\ \Sigma(z^2 - \mu^2)$	$\sigma^{2[3]} = 1/m\ \Sigma(z^3 - \mu^3)$		
z norm value	$z^1_{norm} = (z^1 - \mu^1) / \sqrt{(\sigma^{2[1]} + \varepsilon)}$	$z^2_{norm} = (z^2 - \mu^2) / \sqrt{(\sigma^{2[2]} + \varepsilon)}$	$z^3_{norm} = (z^3 - \mu^3) / \sqrt{(\sigma^{2[3]} + \varepsilon)}$		
z with gamma & beta / Trainable Parameters	$z^{\sim 1} = \gamma^1 z^1_{norm} + \beta^1$	$z^{\sim 2} = \gamma^2 z^2_{norm} + \beta^2$	$z^{\sim 3} = \gamma^3 z^3_{norm} + \beta^3$		
Activation value	$a^1 = f^1 (z^{\sim 1})$	$a^2 = f^2 (z^{\sim 2})$	$a^3 = f^3 (z^{\sim 3})$		

$z^1_{norm} = (z^1 - \mu^1) / \sqrt{(\sigma^{2[1]} + \varepsilon)}$

$z^1_{norm} \sqrt{(\sigma^{2[1]} + \varepsilon)} = (z^1 - \mu^1)$

$z^1_{norm} \sqrt{(\sigma^{2[1]} + \varepsilon)} + \mu^1 = z^1$

$z^1 = \sqrt{(\sigma^{2[1]} + \varepsilon)} * z^1_{norm} + \mu^1$

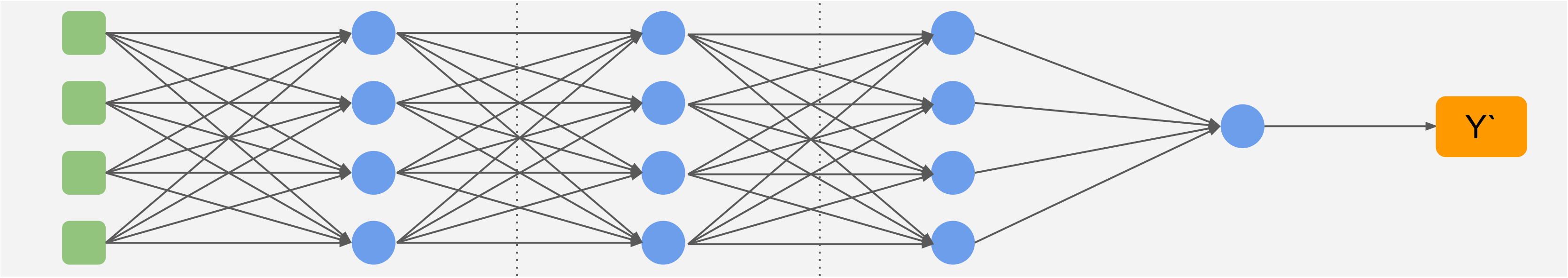
$z^{\sim 1} = \quad \gamma^1 \quad z^1_{norm} + \beta^1$

$\gamma^1 = \sqrt{(\sigma^{2[1]} + \varepsilon)} \qquad \beta^1 = \mu^1$

$z^{\sim 1} = z^1$

Without Batch Normalization - Covariate Shift

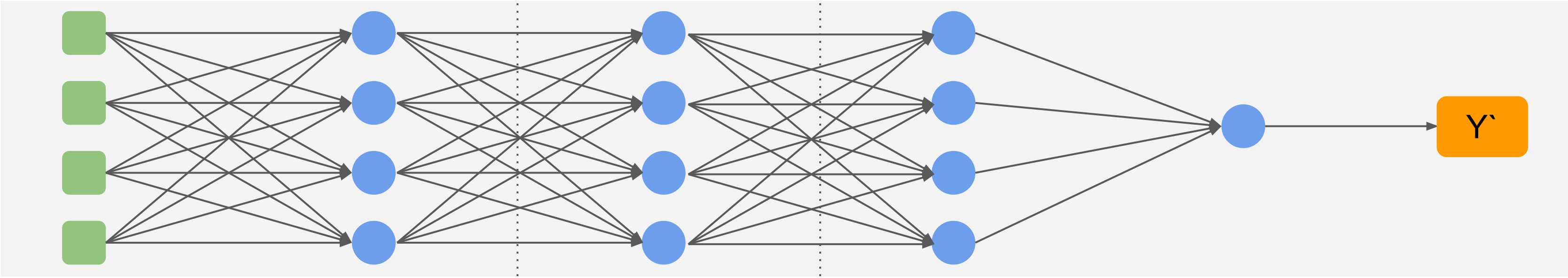
Learning of Shifting Input Distribution

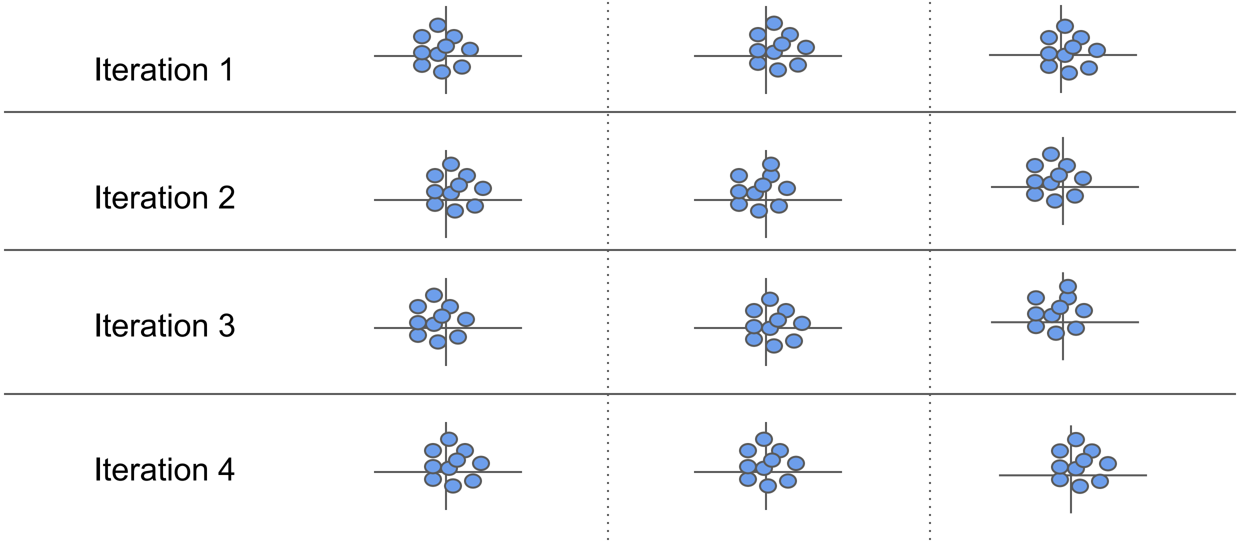


Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Covariate Shift

Learning of Shifting Input Distribution



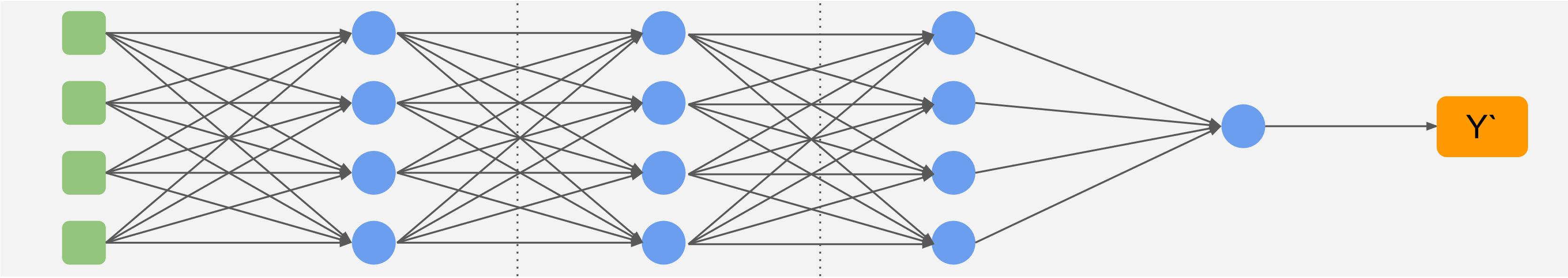


Less prone to noise of the input.
This makes the learning faster and efficient
Improves gradient flow through the network.

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



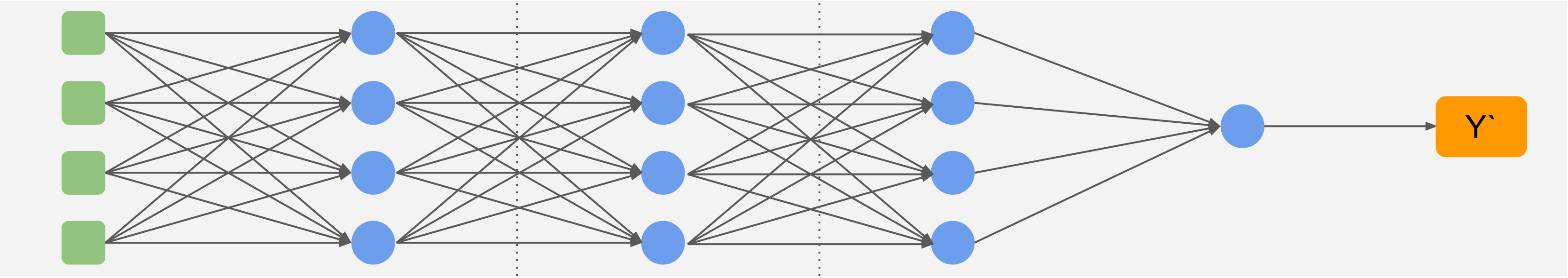
1 Epoch or Iteration

Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



1 Epoch or Iteration

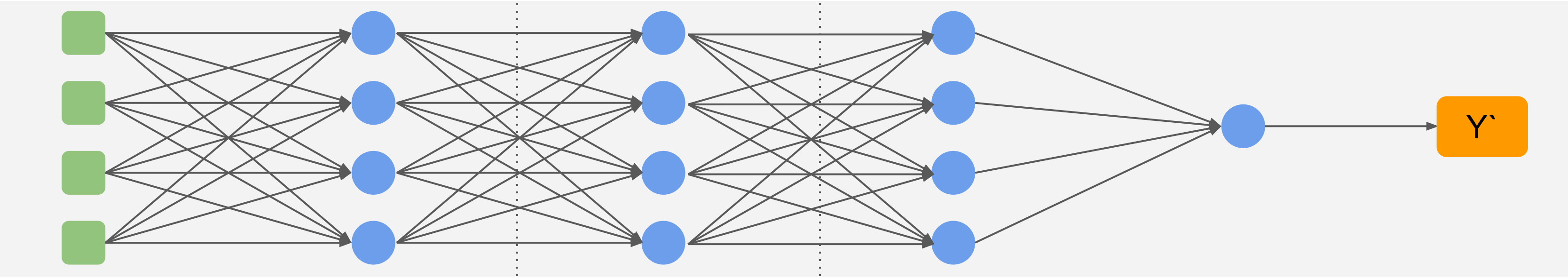
Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

Compute Moving Average

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



1 Epoch or Iteration

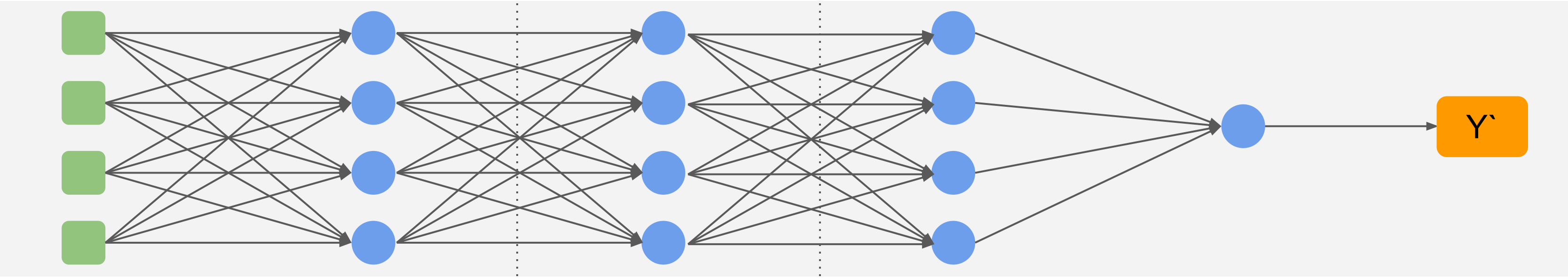
Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

Compute Moving Average

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



1 Epoch or Iteration

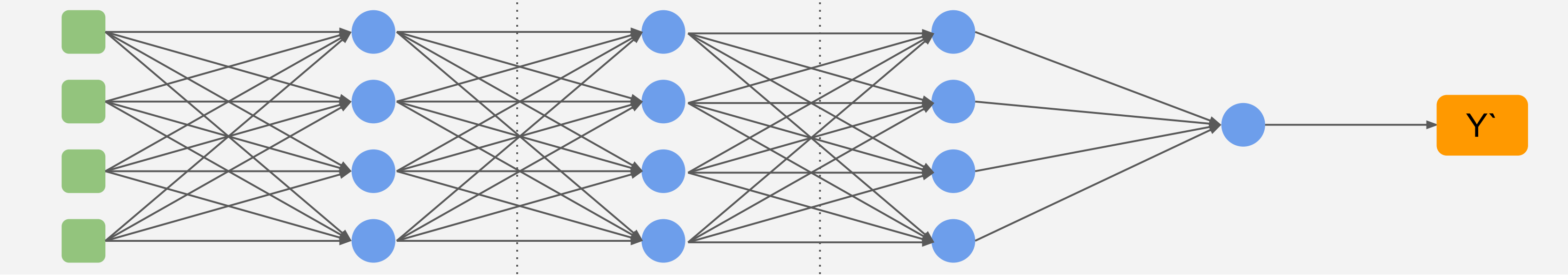
	Layer 1	Layer 2	Layer 3
Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

Compute Moving Average

Arun Kumar A
analaarun.k@gmail.com
https://www.linkedin.com/in/arun-kumar-anala-35760523/

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



1 Epoch or Iteration

	Layer 1	Layer 2	Layer 3
Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

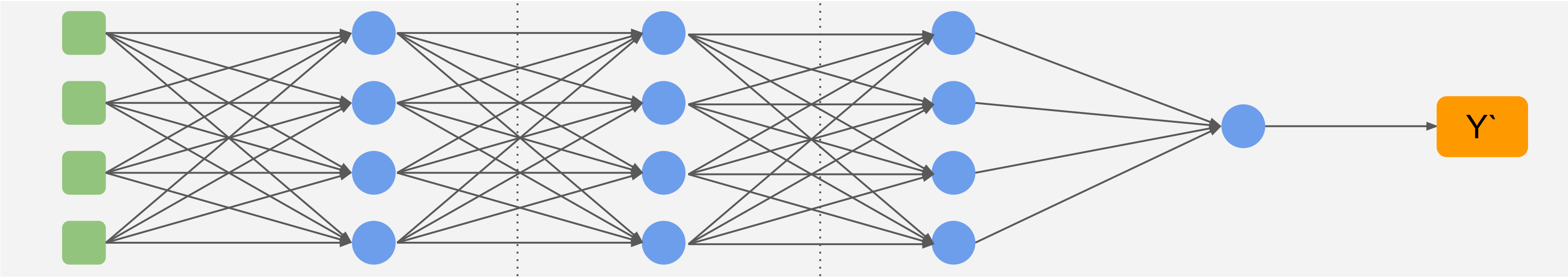
In Exponential Weighted Average, high weightage is given to new values

Compute Moving Average

These values would be finally used for testing to compute the z_{norm} value

Batch Normalization - Mini batch

Train set : 1000
Mini Batch : 200



1 Epoch or Iteration

Mini Batch 1	$\mu^{1[1]}$, $\sigma^{2[1][1]}$	$\mu^{2[1]}$, $\sigma^{2[2][1]}$	$\mu^{3[1]}$, $\sigma^{2[3][1]}$
Mini Batch 2	$\mu^{1[2]}$, $\sigma^{2[1][2]}$	$\mu^{2[2]}$, $\sigma^{2[2][2]}$	$\mu^{3[2]}$, $\sigma^{2[3][2]}$
Mini Batch 3	$\mu^{1[3]}$, $\sigma^{2[1][3]}$	$\mu^{2[3]}$, $\sigma^{2[2][3]}$	$\mu^{3[3]}$, $\sigma^{2[3][3]}$
Mini Batch 4	$\mu^{1[4]}$, $\sigma^{2[1][4]}$	$\mu^{2[4]}$, $\sigma^{2[2][4]}$	$\mu^{3[4]}$, $\sigma^{2[3][4]}$
Mini Batch 5	$\mu^{1[5]}$, $\sigma^{2[1][5]}$	$\mu^{2[5]}$, $\sigma^{2[2][5]}$	$\mu^{3[5]}$, $\sigma^{2[3][5]}$
Estimate using Exponential Weighted Average across mini-batch	μ^{1} , $\sigma^{2[1]}$	μ^{2} , $\sigma^{2[2]}$	μ^{3} , $\sigma^{2[3]}$

The Batch Normalization in Mini batch generates noise which propagates to next hidden layers, so similar to drop out it has slight regularization effect.

So less records in mini-batch, more noise it adds to the next layer.

Arun Kumar A
analaarun.k@gmail.com
in https://www.linkedin.com/in/arun-kumar-anala-35760523/