UE21CS343BB2

Topics in Deep Learning

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD),
Centre for Data Sciences & Applied Machine Learning (CSSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Anirudh Chandrasekar, Teaching Assistant

PES UNIVERSITY

Topics in Deep Learning

Batch Normalization

Problems with NN Training process

- Internal Covariate Shift
- Unstable and Slow training
- Sensitivity to weight initialization
- Limited Generalization
- Increased dependency on Hyper-parameters

Covariate Shift

- Variables that affect the response variable, but are not of interest in the study. (according to statistics)
- In ML : They are not variables but essentially features.

Ex:

- Age: Study between physical activity and health.
- Gender: Study between job satisfaction and work-life balance.
- Neural Networks: Activations in Hidden layers and features.

Covariate Shift

- **Shift in the distribution of features.**
- Usually shift of data/feature distribution between training and testing.

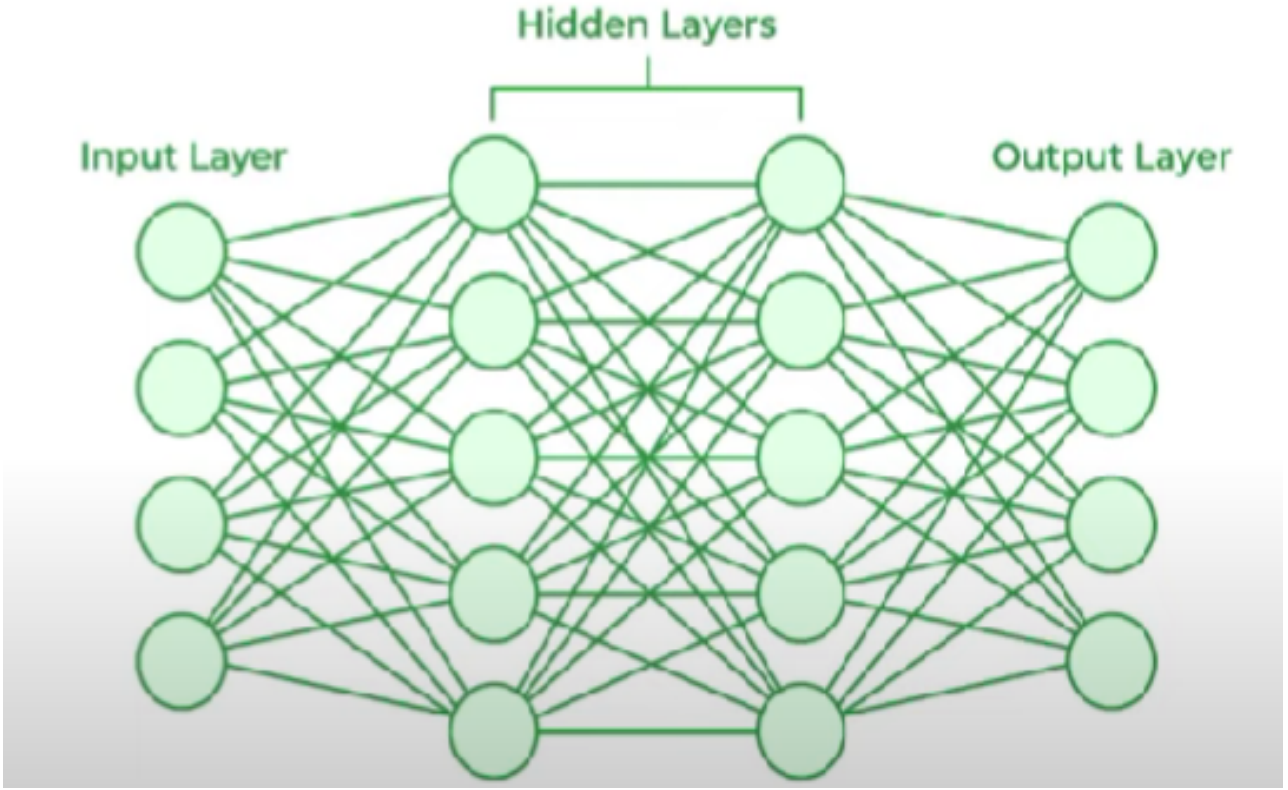
Ex:

- Image Classification: Day (Train) vs Night (Test)
- Speech Recognition: Trained on different accents but tested on another.
- Medical Diagnosis: Trained on patients below 20 years and testing on patients above 50 years.

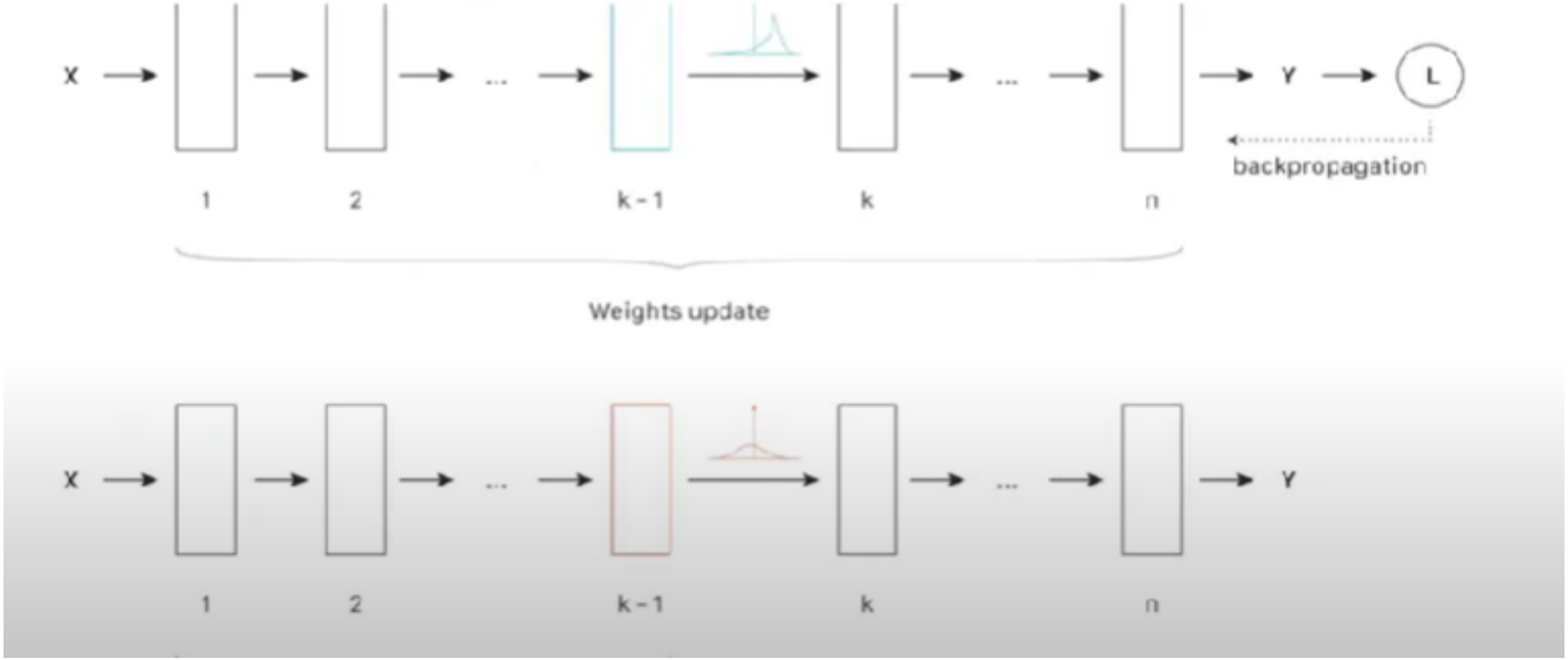
Internal Covariate Shift

- **Shift in the distribution of hidden layer features during training.**

- We know that the Input to the 2nd hidden layer is the output of the 1st hidden layer.
- Similarly all the inputs to the intermediate layers are outputs of the previous layer, which are dependent on the Weights.
- These weights get updated while training.
- After weight update the input distributions could change.



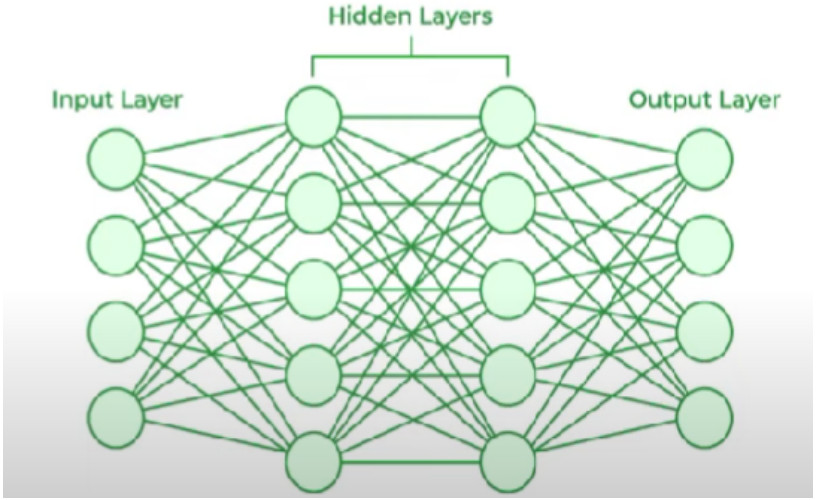
Internal Covariate Shift

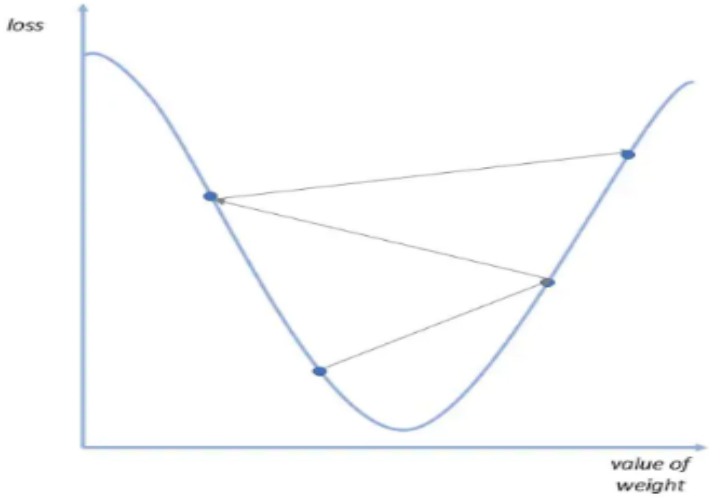


- The distribution of input has changed in the k^{th} layer after weight updation.

Unstable and Slow training

- Since the distribution of inputs is changing continuously the model has to re-learn the weights which results in slow training.
- As the distribution changes the model is prone to large errors, leading to larger weight updates.
- Larger weight updates causes fluctuation and makes the model unstable as shown in the graph.
- Due to this reaching convergence takes a lot of time.

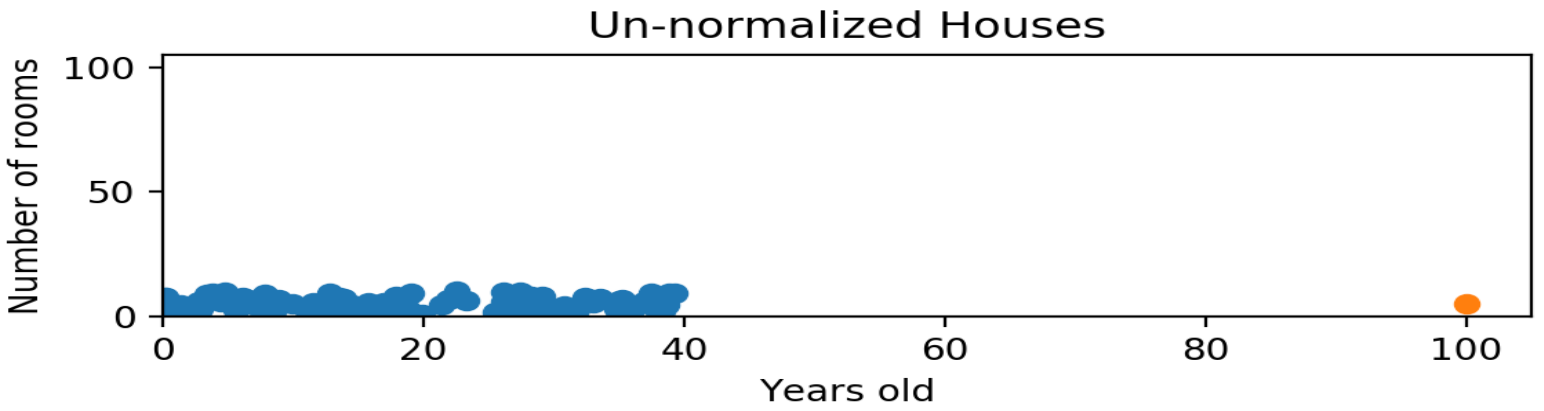


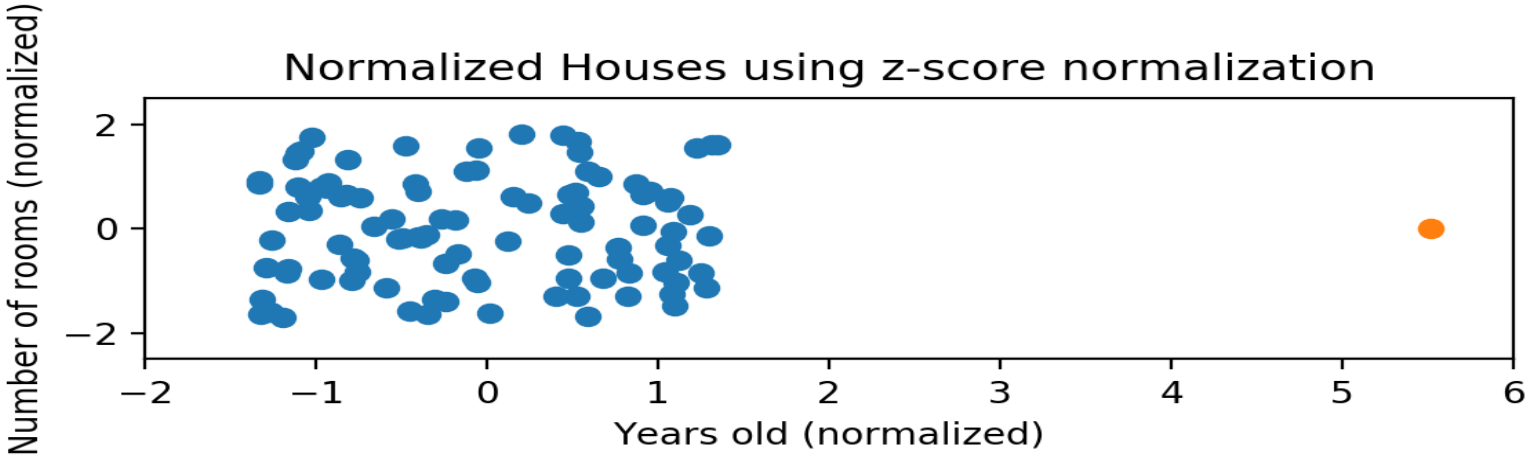


Problems with NN Training process

- **Weight initialization** is done randomly in neural networks. Initialization must be done with caution as poor initialization can lead to never attaining convergence.
- **Limited Generalization**: Since the distribution keeps changing, we cannot generalize the model as it fits to the data that is already seen.
- **Increased dependency on Hyper-parameters**: Learning rate is the hyper-parameter which plays a crucial role in time taken in attaining convergence, but without normalization there are chances that certain values(higher values) of learning rates do not lead to convergence at all.

Input Standardization





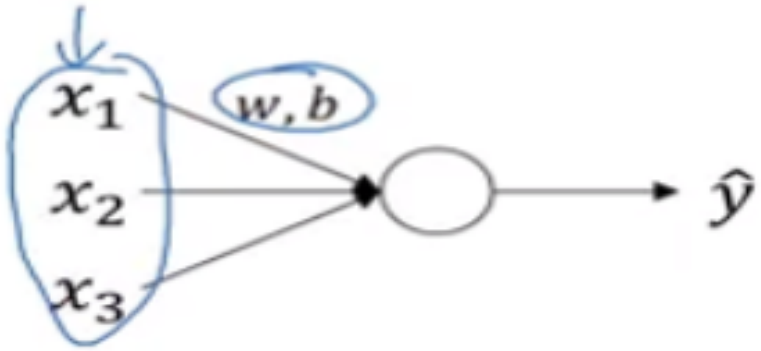
Convert input data into a distribution with 0 mean and unit variance.

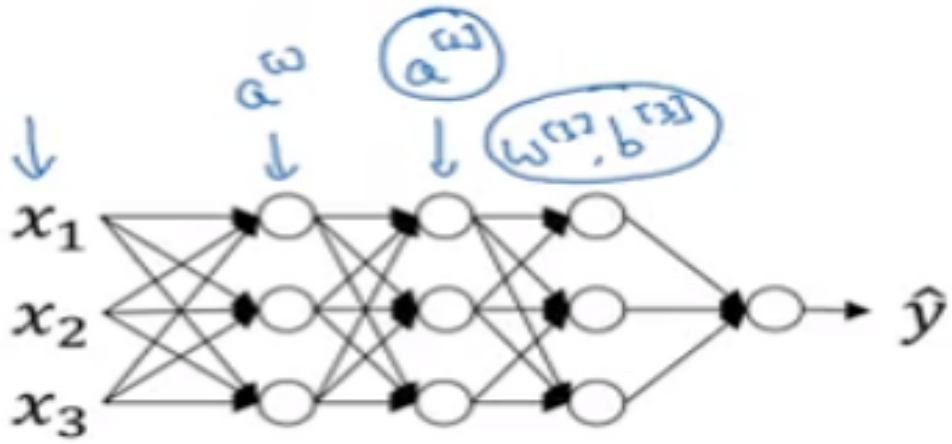
This takes care of the input layers.

What about hidden layers?

Normalizing Activations in a Network

Normalizing inputs is done to speed up learning



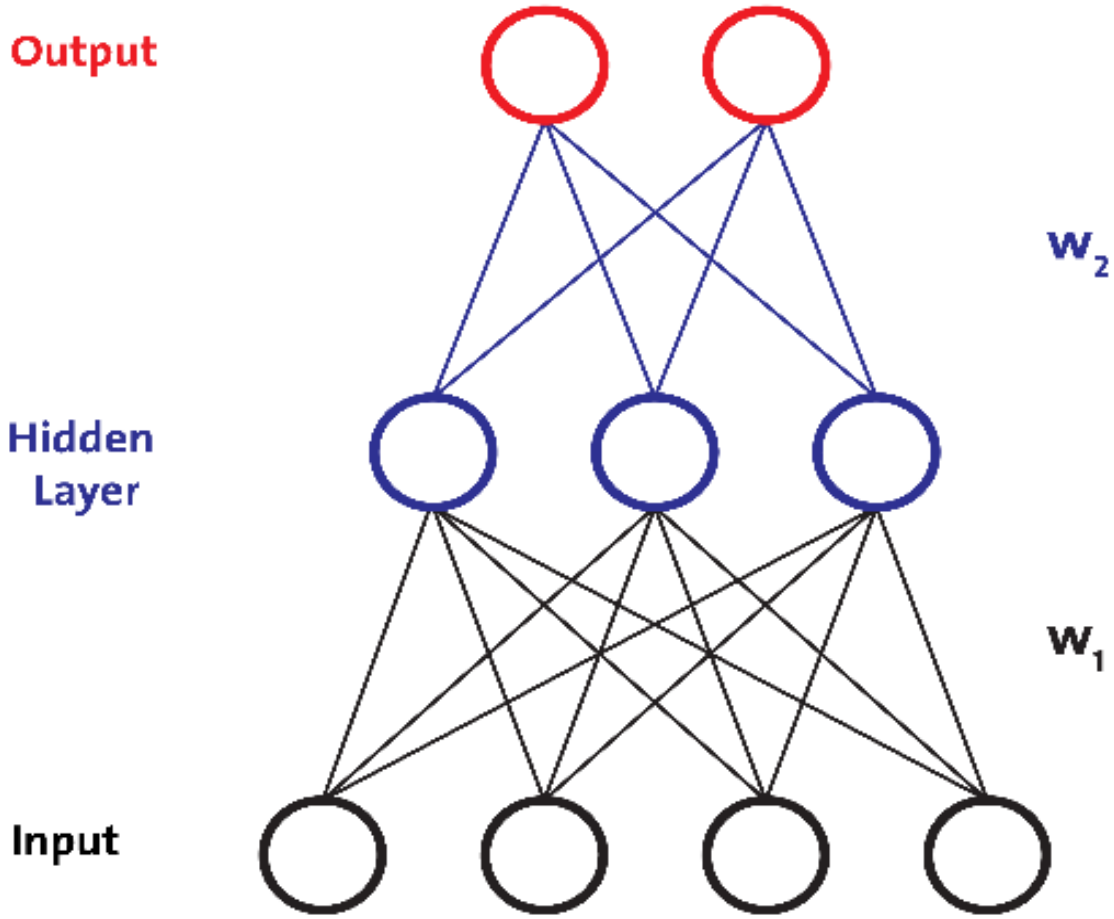


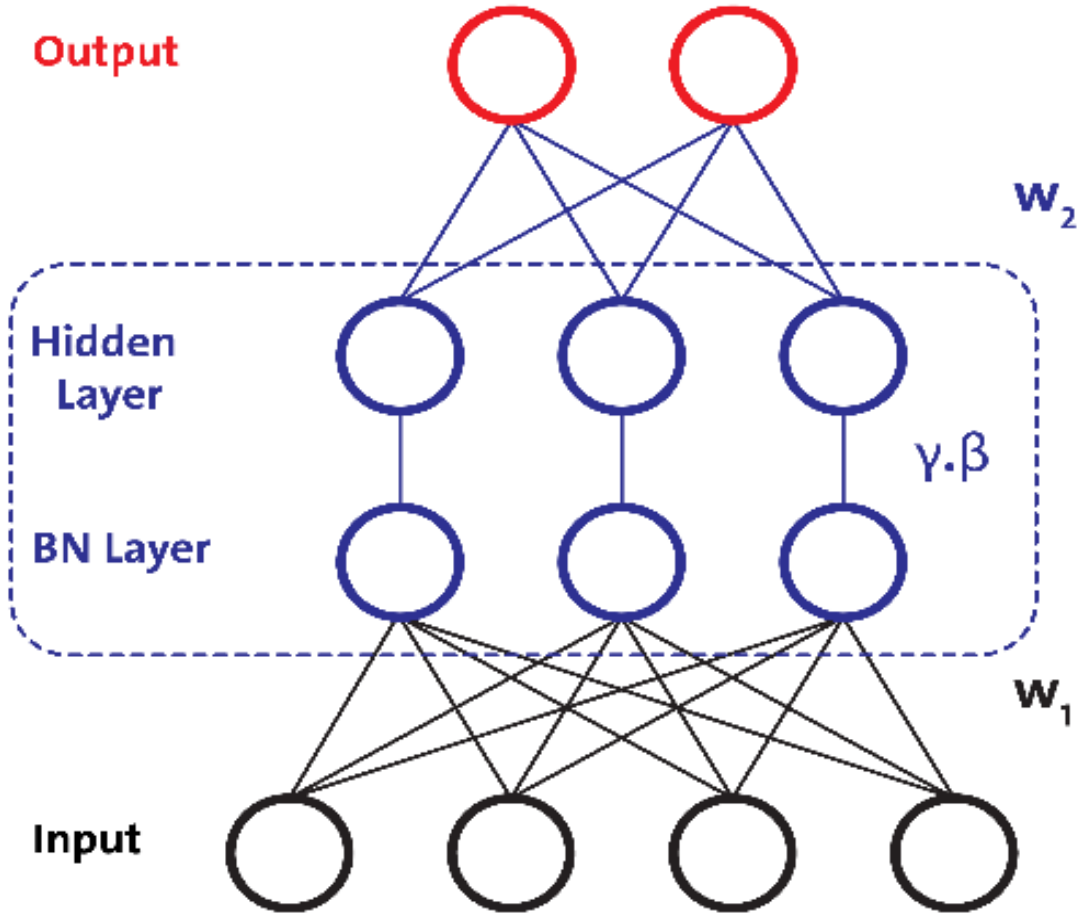
This is done by computing the mean and variance of the input and then subtracting the mean and normalizing the data by the variance.

Considering a deep network, Can we normalize $a^{[2]}$ (or any hidden layer) so as to train $w^{[3]}$, $b^{[3]}$ faster?

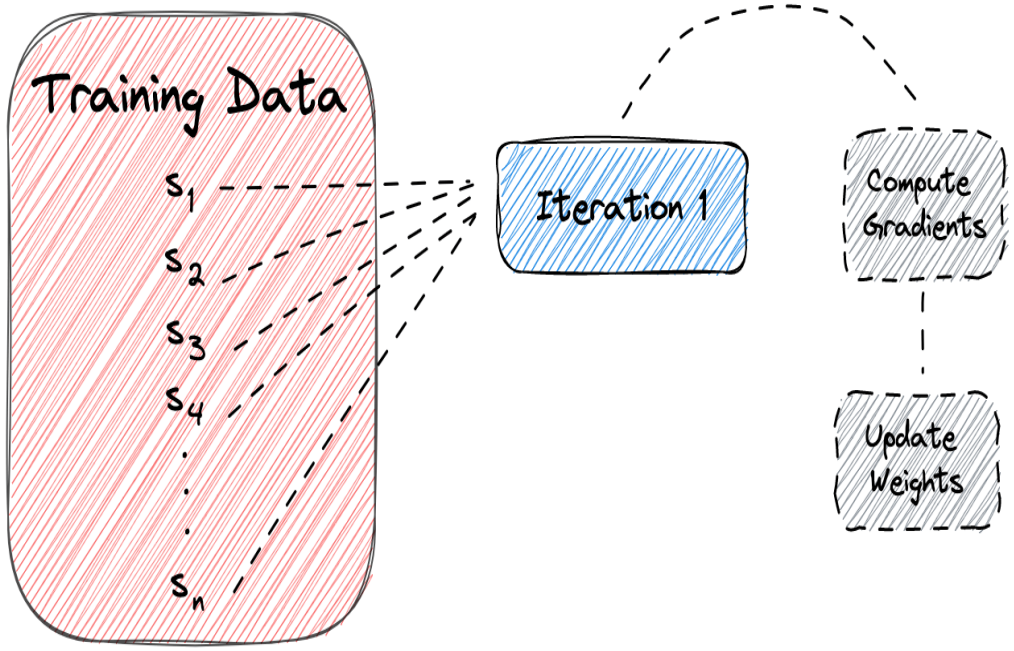
- This is what batch normalization does.
- But we normalize $z^{[2]}$ instead of $a^{[2]}$ i.e. before applying the activation function.

Implementing Batch Normalization



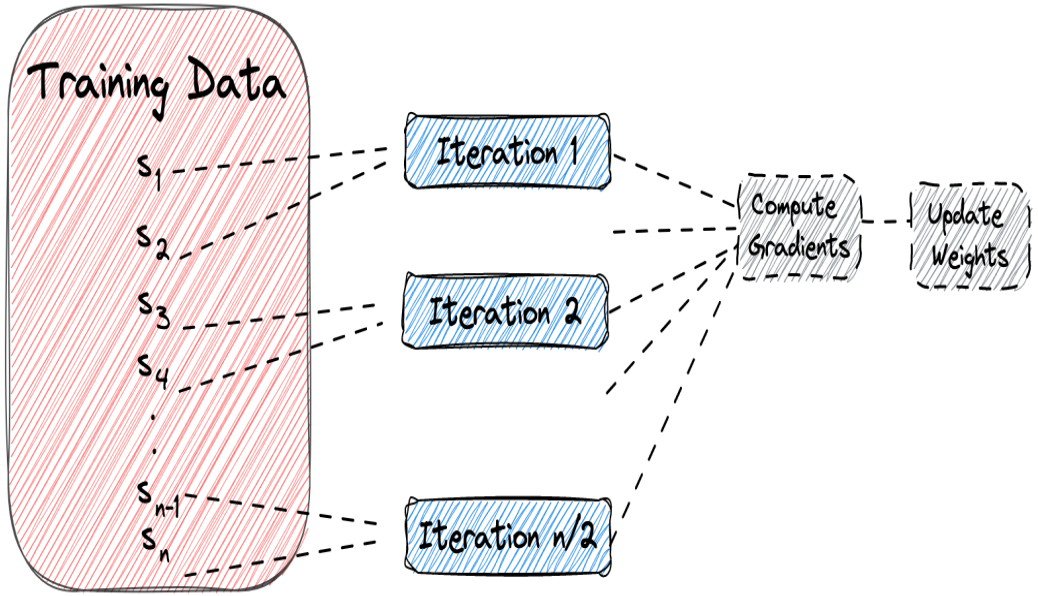


Batch v/s Mini-Batch



Batch - all the data is passed at once.

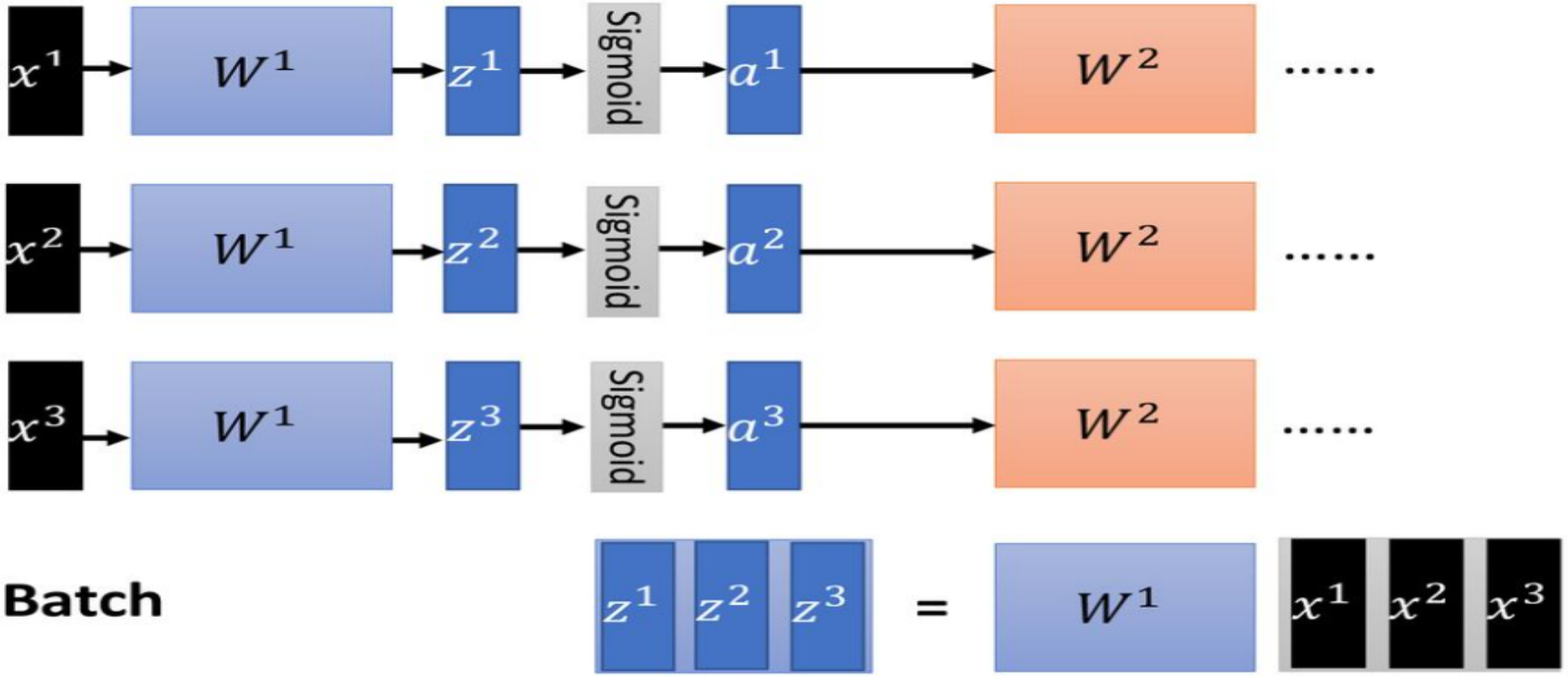
In this case we do not require batch normalization as weight updates happen only once.



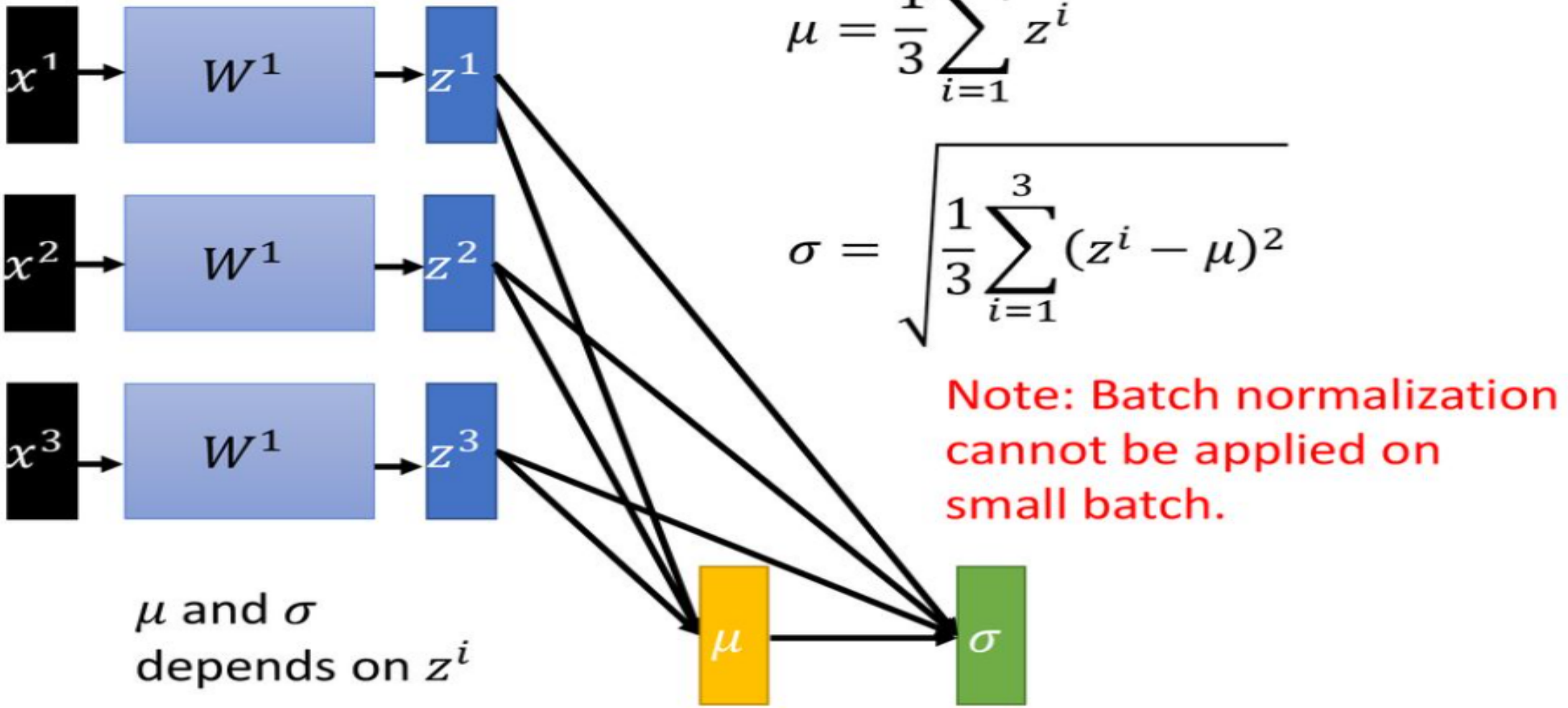
Mini-batch - Data is split into smaller sets and each set is passed in a new iteration.

In this case Batch Normalization is needed since weights are updated before each iteration.

Batch


x^1
W^1
z^1
Sigmoid
a^1
W^2
.....
x^2
W^1
z^2
Sigmoid
a^2
W^2
.....
x^3
W^1
z^3
Sigmoid
a^3
W^2
.....
Batch
z^1 z^2 z^3
=
W^1
x^1 x^2 x^3

Batch Normalization



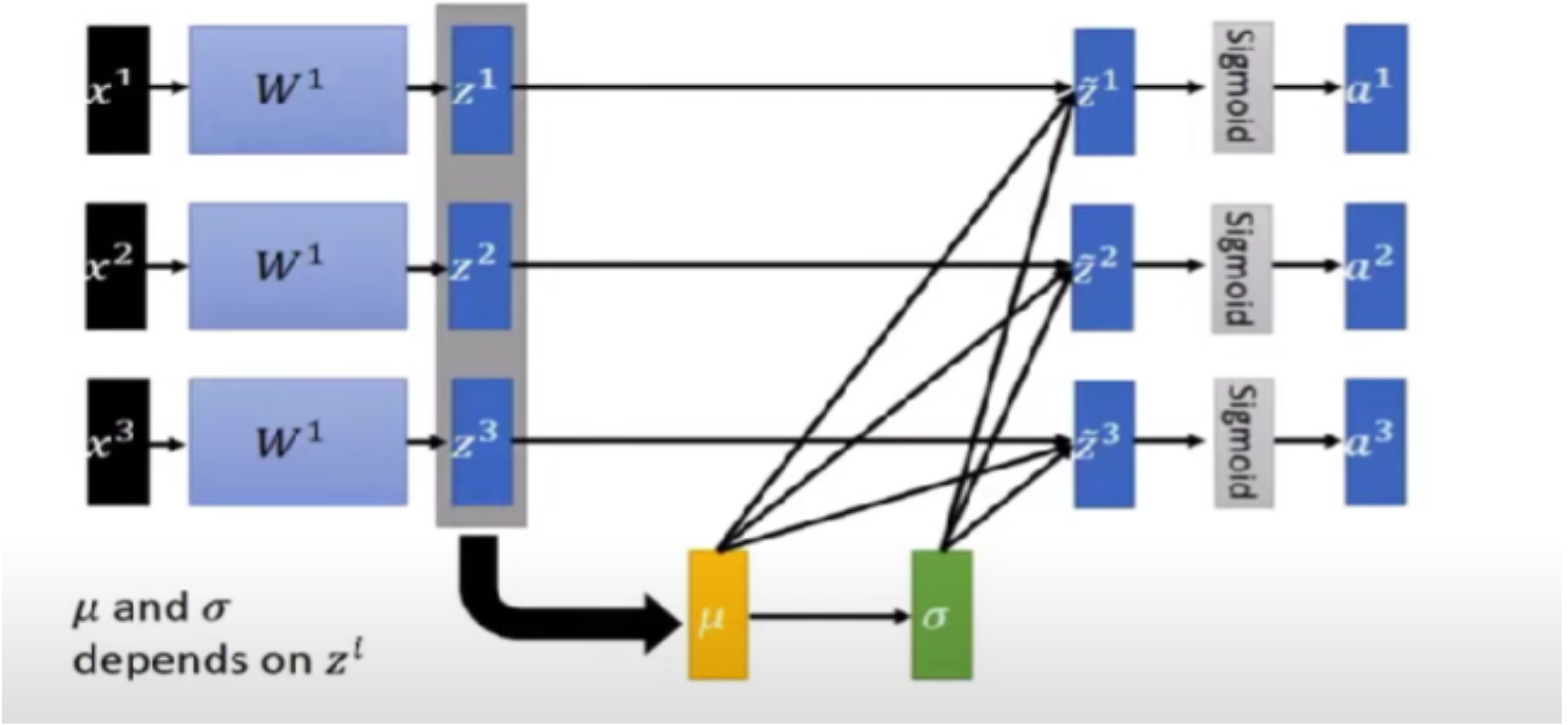
$$\mu = \frac{1}{3}\sum_{i=1}^{3} z^i$$

$$\sigma = \sqrt{\frac{1}{3}\sum_{i=1}^{3}(z^i - \mu)^2}$$

Note: Batch normalization cannot be applied on small batch.

Batch Normalization

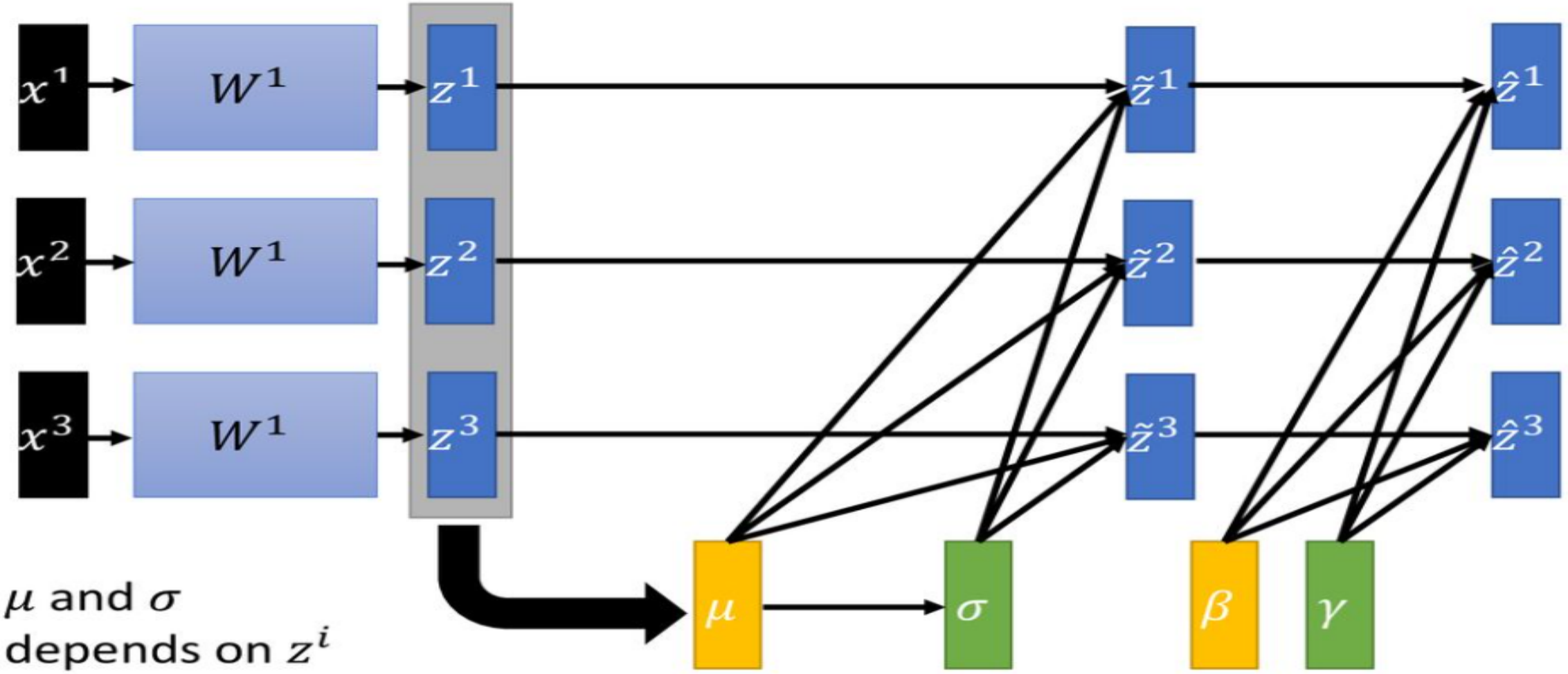
$$Z^i_{norm} = \frac{z^i - \mu}{\sigma}$$



Batch Normalization

$$Z^{(i)}_{norm} = \frac{z^i - \mu}{\sigma}$$

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



Implementing Batch Normalization

Given some intermediate values of the NN $z^{(1)}, z^{(2)}, z^{(3)}, \ldots z^{(i)}$ for some hidden layer l.

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

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

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

where

- $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\sigma}^2$ is the variance.
- $\boldsymbol{\varepsilon}$ is added to ensure mathematical stability. (incase variance turns out to be 0).

Implementing Batch Normalization

Now we have normalized the values of z such that they have mean 0 and standard unit variance.

But we do not want this to be the case always and might want them to have a different distribution.

So we compute,

$$\tilde{Z}^{(i)} = \boldsymbol{\gamma}\, Z^{(i)}_{norm} + \boldsymbol{\beta}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable parameters of the model.

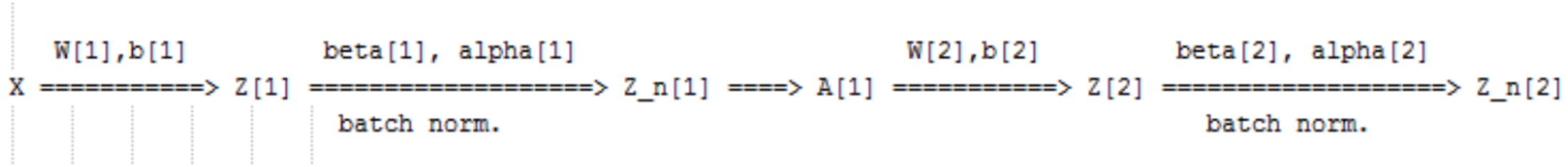
Note: $\boldsymbol{\gamma}$ scales and $\boldsymbol{\beta}$ shifts.

Question: For what values of $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ will $\tilde{Z}^{(i)} = Z^{(i)}$?

Ans: $\boldsymbol{\gamma} = \sqrt{\boldsymbol{\sigma}^2 + \boldsymbol{\varepsilon}}$ and $\boldsymbol{\beta} = \boldsymbol{\mu}$

Adding Batch Normalization to a Network

Using Batch Norm in 3 hidden layers NN:



Our NN parameters will be:

`W[1]`, `b[1]`, ..., `W[L]`, `b[L]`, `beta[1]`, `gamma[1]`, ..., `beta[L]`, `gamma[L]`

`beta[1]`, `gamma[1]`, ..., `beta[L]`, `gamma[L]` are updated using any optimization algorithms (like GD, RMSprop, Adam)

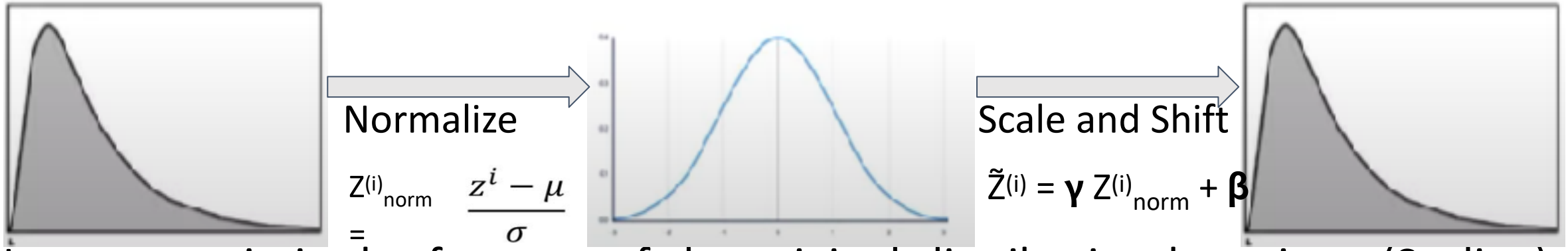
If you are using a deep learning framework, you won't have to implement batch norm yourself:

- Ex. in Tensorflow you can add this line: tf.nn.batch-normalization()

Why are β and γ needed?

As shown in the figure once we normalize the inputs, all of them them follow a normal distribution.

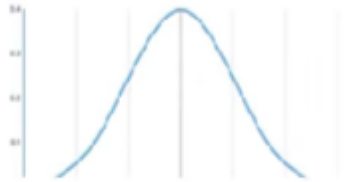
This leads to the model learning the same distribution over and over again, hindering the learning capability of the model.



Hence we mimic the features of the original distribution by using **γ**(Scaling) and **β**(Shifting).

Note: When $\gamma = \sqrt{\sigma^2 + \varepsilon}$ and $\beta = \mu$ we get the above result.

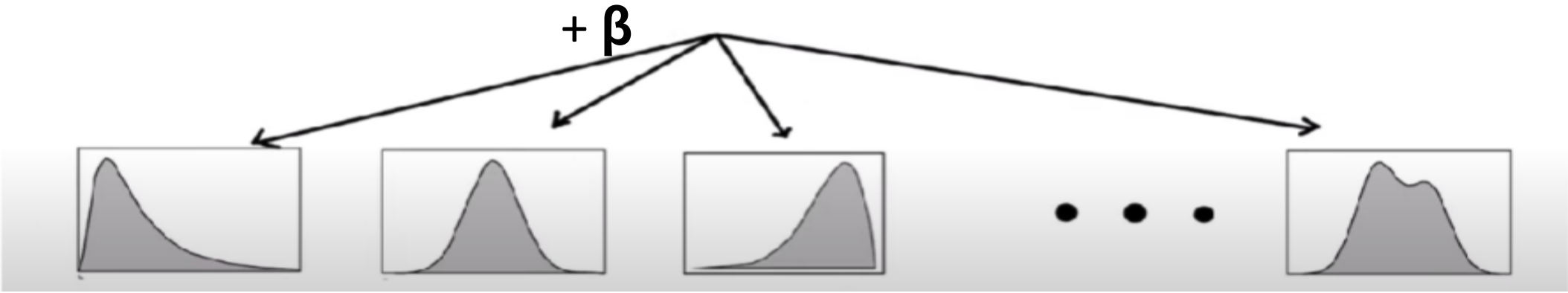
Why are β and γ learnable?



γ, β are network parameters

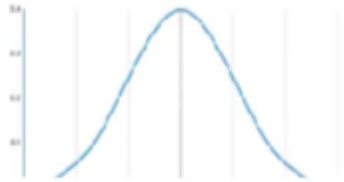
$$\tilde{Z}^{(i)} = \boldsymbol{\gamma}\, Z^{(i)}_{norm}$$

$$+ \boldsymbol{\beta}$$



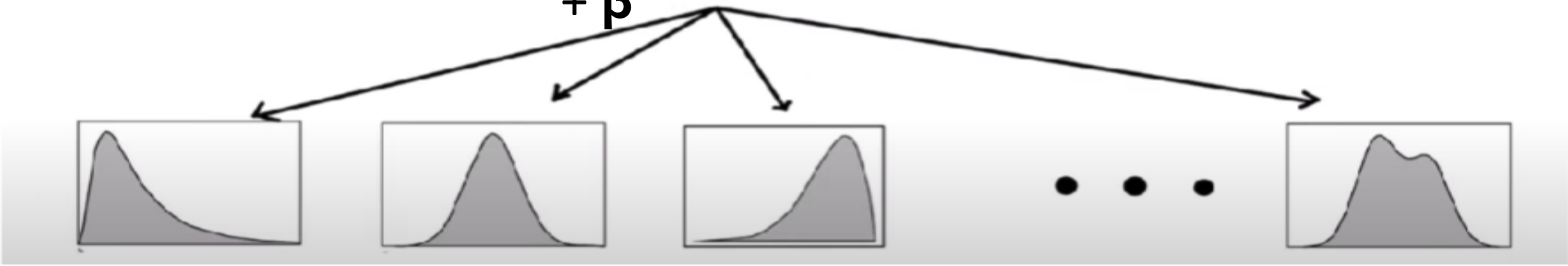
As shown in the figure, different values of **γ** and **β** produces different distributions and what is chosen depends on how the distributions are varying across batches(Learning factor).

Why are β and γ learnable?



γ, β are network parameters

$$\tilde{Z}^{(i)} = \boldsymbol{\gamma}\, Z^{(i)}_{norm} + \boldsymbol{\beta}$$



Using **γ** and **β**, the network is trying to learn the overall distribution of the training dataset even though we are passing the input as batches.

PES UNIVERSITY

Advantages of Batch Normalization

- Networks train faster.
- Allows for higher learning rates.
- Makes weights easier to initialise.
- Provides some regularization.

Acknowledgements & References

- https://deeplearning.ai
- https://medium.com/@rishavsapahia/5-min-recap-for-andrew-ng-deep-learning-specialization-course-2-8a59fd58ca0d
- https://youtu.be/PRQPyQeq6C4?si=7j_lT2i03mS4O94N
- https://youtu.be/2xChdY2qkmc?si=ten5Owc--g8ZaKOk

Thank You

Dr. Shylaja S S
Director of Cloud Computing & Big Data (CCBD), Centre for Data Sciences & Applied Machine Learning (CSSAML)
Department of Computer Science and Engineering
shylaja.sharath@pes.edu

Ack: Anirudh Chandrasekar, Teaching Assistant