Batch Normalization

Pavlos Protopapas

Feature Normalization

Good practice to normalize features before applying learning algorithm:

Feature vector

Vector of mean feature values

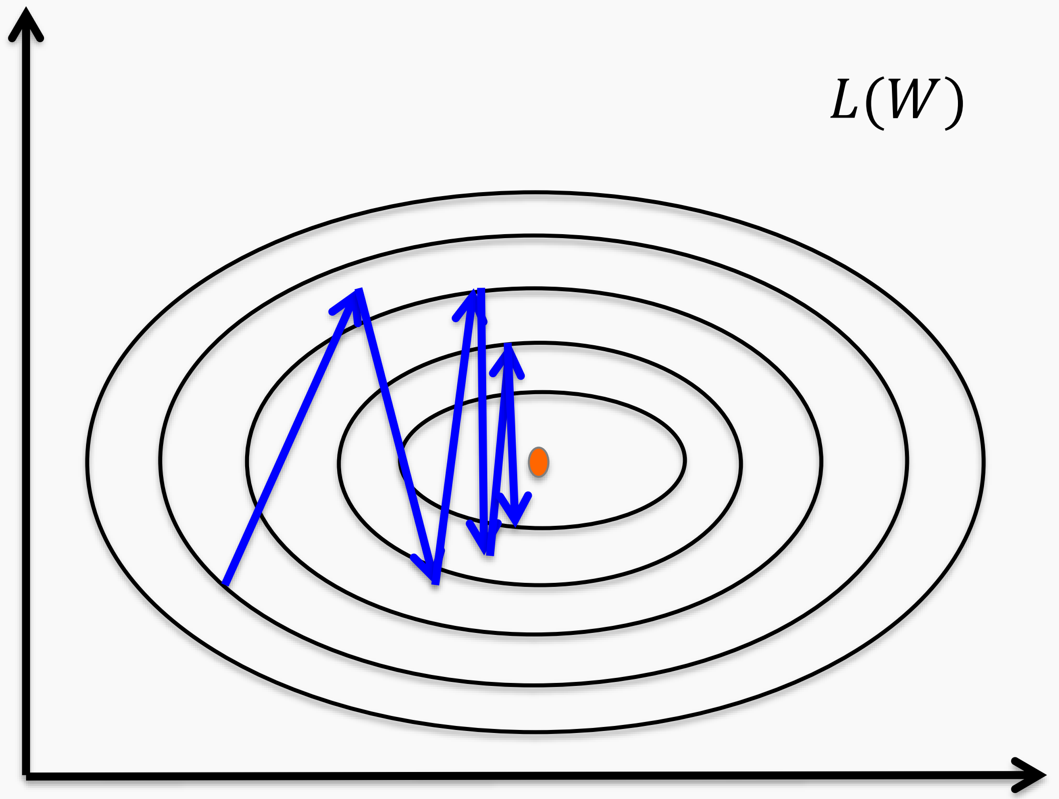
$$\boxed{x' = \frac{x - \mu}{\sigma}}$$

Vector of SD of feature values

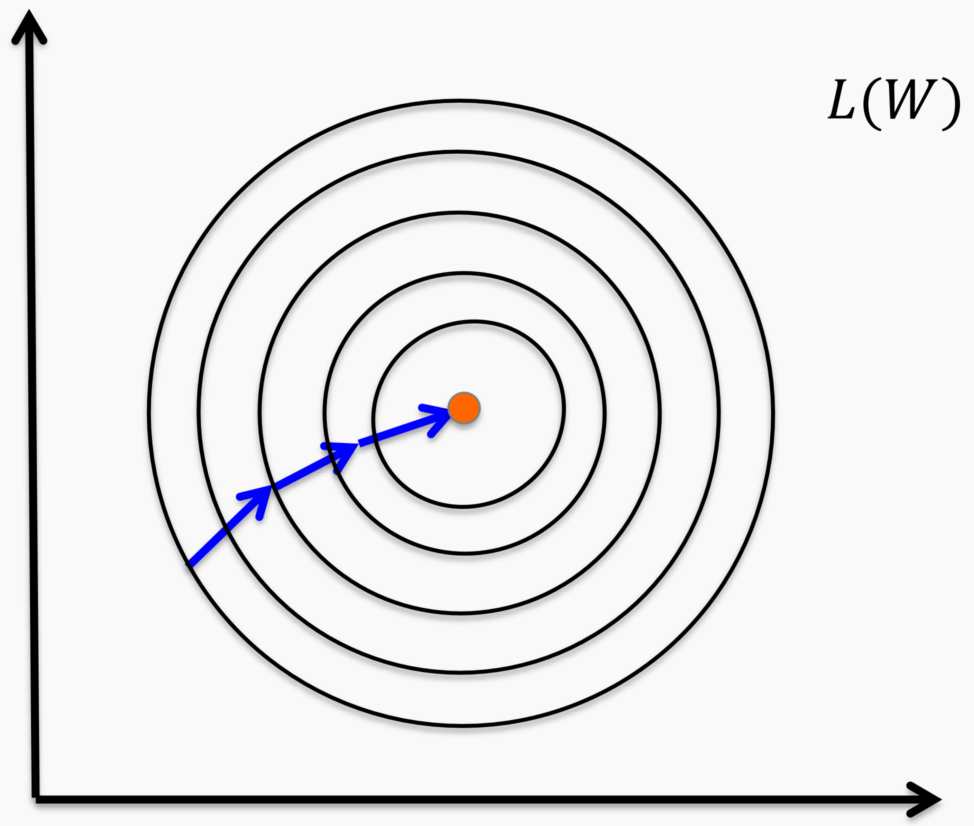
Features in same scale: mean 0 and variance 1

Feature Normalization

Speeds up learning



Before normalization

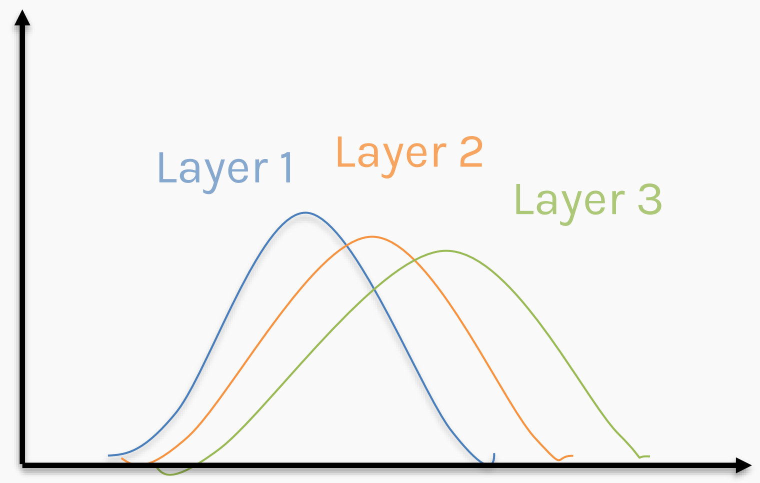


After normalization

Note: This is an ideal case scenario. In real, loss landscapes are much more complex.

Internal Covariance Shift

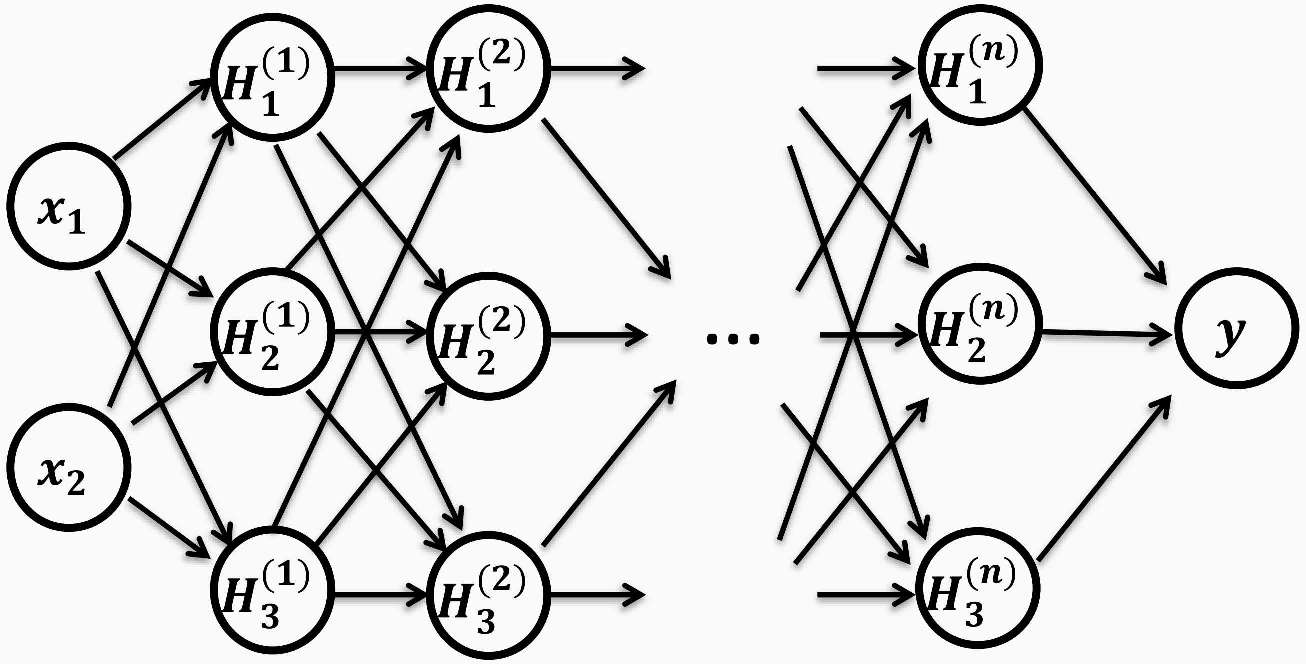
Even if you normalize your data to mean 0 and variance 1, the shape of the distribution may still change as you propagate through the layers of your neural network.



Distribution of the outputs of different layers

Internal Covariance Shift

Each hidden layer changes distribution of inputs to next layer: *slows down learning*



Internal Covariance Shift

We know that:

$$\frac{dL}{dW^{(1)}} = \frac{dL}{dH^{(n)}} \times \frac{dH^{(n-1)}}{dH^{(n-2)}} \times \cdots \times \frac{dH^{(2)}}{dH^{(1)}} \times \frac{dH^{(1)}}{dW^{(1)}}$$

Internal Covariance Shift

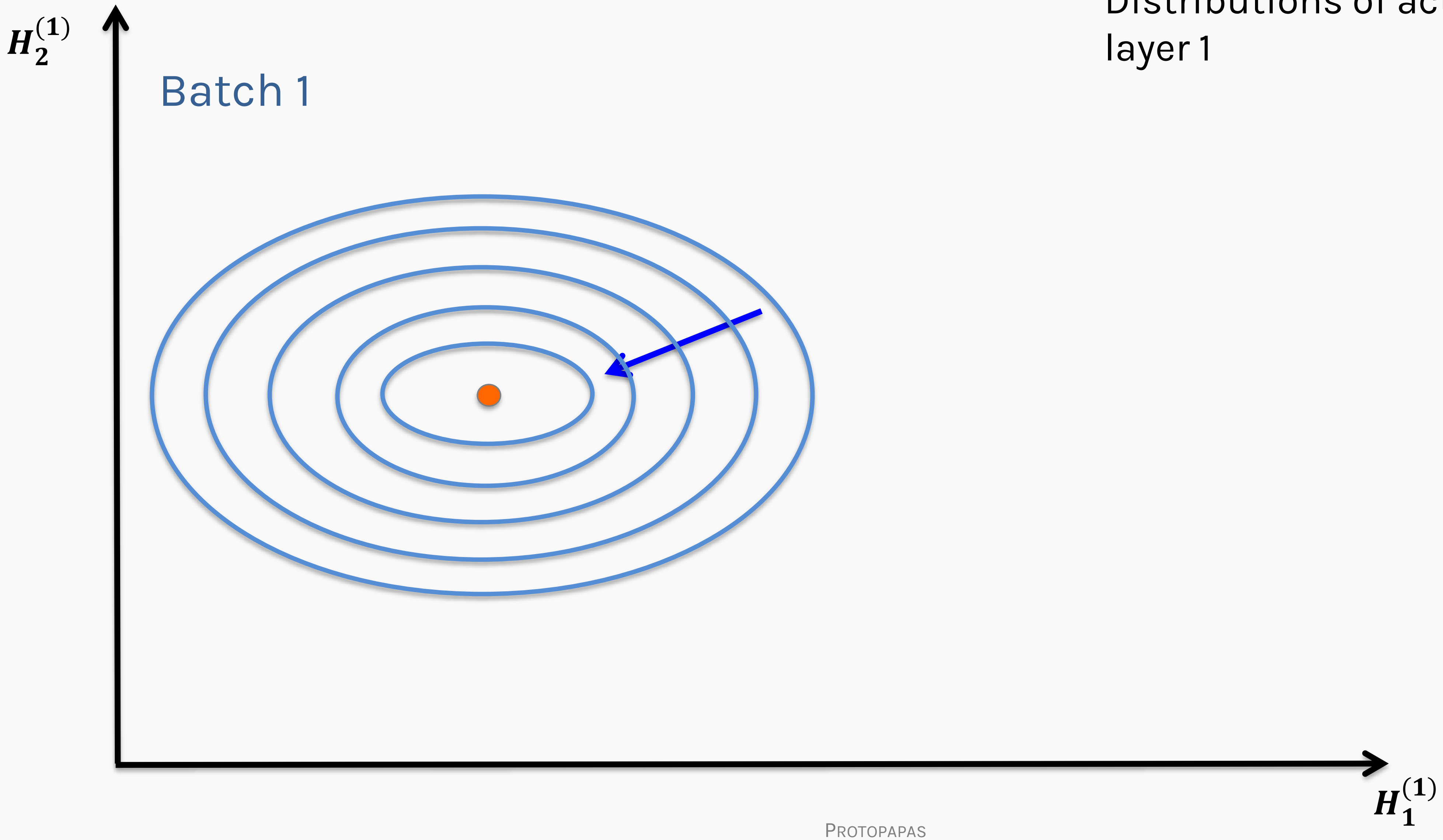
We know that:

$$\frac{dL}{dW^{(1)}} = \frac{dL}{dH^{(n)}} \times \frac{dH^{(n-1)}}{dH^{(n-2)}} \times \cdots \times \frac{dH^{(2)}}{dH^{(1)}} \times \frac{dH^{(1)}}{dW^{(1)}}$$

Let's look at this gradient.

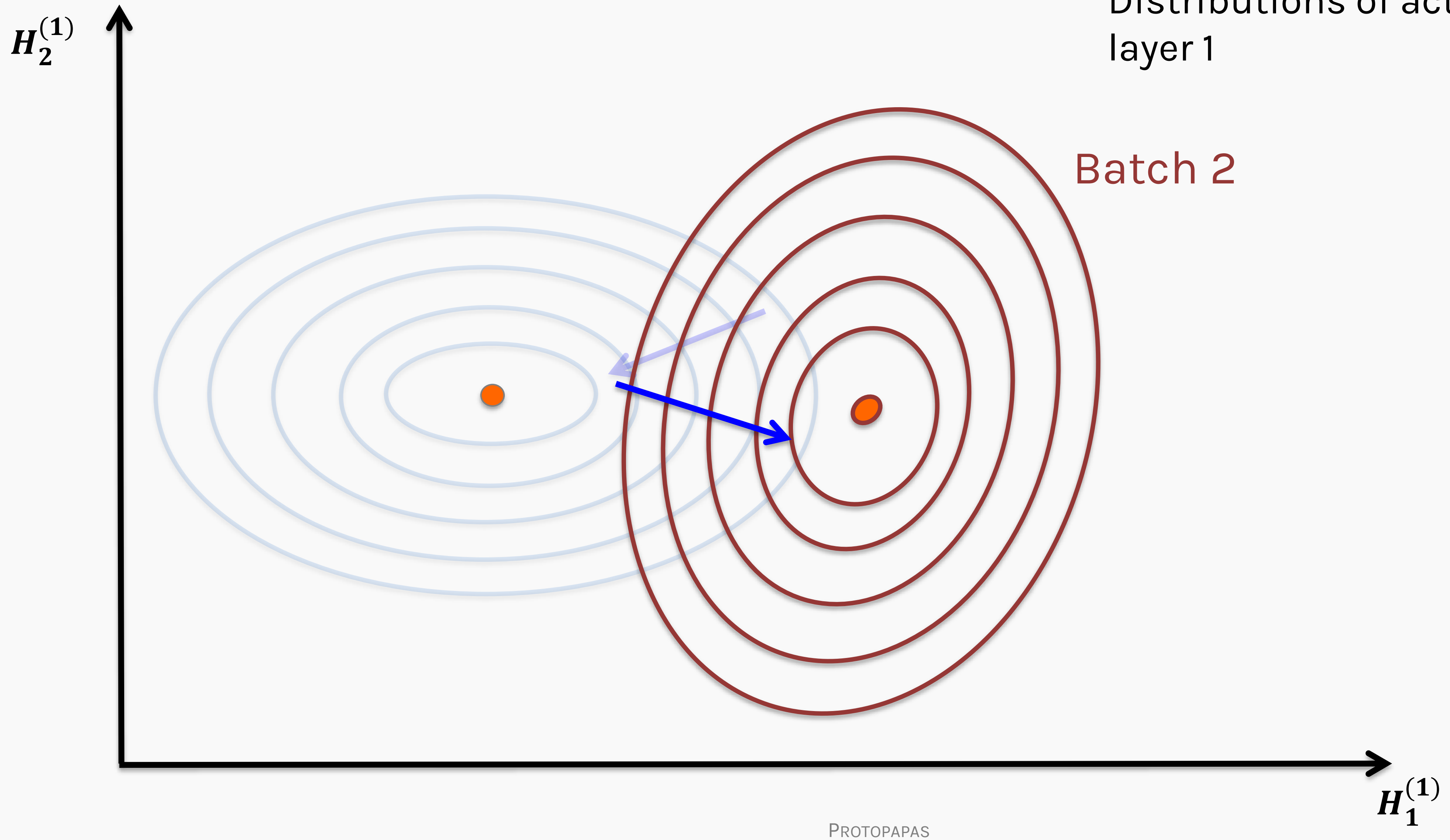
Internal Covariance Shift

Distributions of activations of hidden layer 1

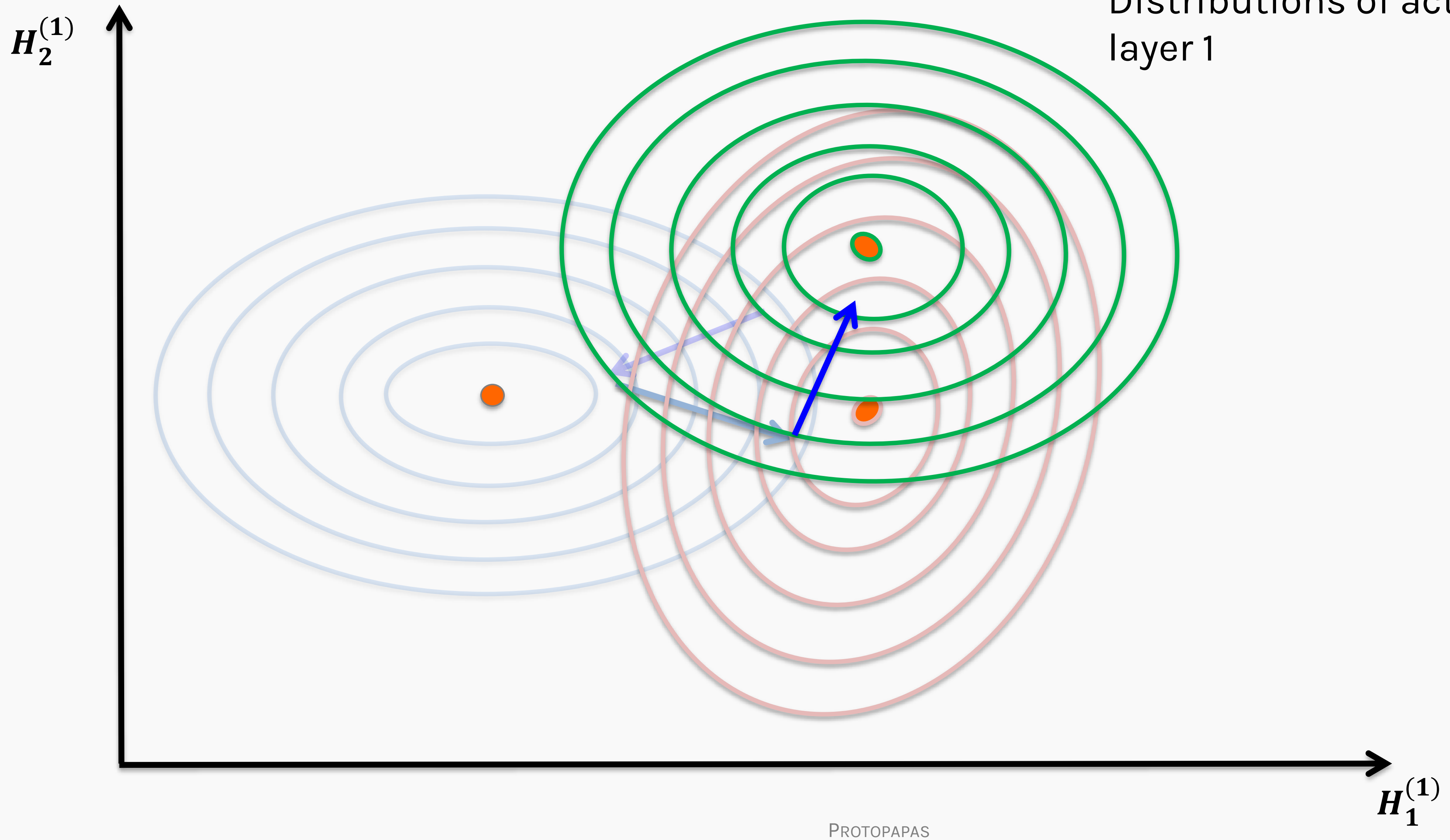


Internal Covariance Shift

Distributions of activations of hidden layer 1



Internal Covariance Shift



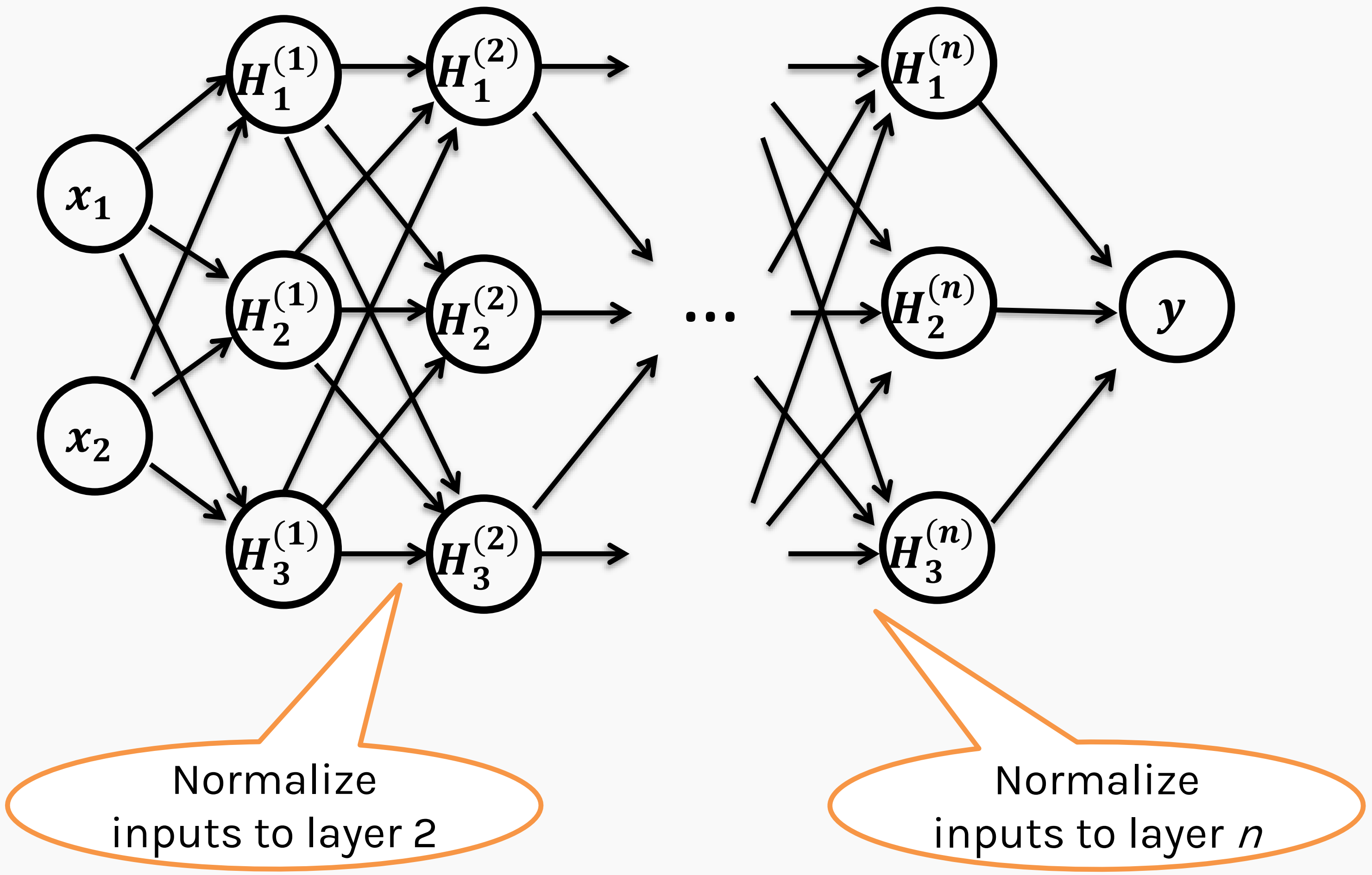
Distributions of activations of hidden layer 1

Internal Covariance Shift

So, How can this problem be solved?

Internal Covariance Shift Solution

We normalize inputs to every hidden layer.



Batch Normalization

Training time:

Batch of activations for a given layer to normalize

For a given hidden layer

$$\boxed{H = \begin{bmatrix} H_{11} & \cdots & H_{1K} \\ \vdots & \ddots & \vdots \\ H_{N1} & \cdots & H_{NK} \end{bmatrix}}$$

N data points in batch

K hidden units activations

Batch Normalization

Training time:

Batch of activations for a given layer to normalize

$$\boxed{H = \begin{bmatrix} \boxed{\begin{matrix} H_{11} \\ \vdots \\ H_{N1} \end{matrix}} & \begin{matrix} \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} H_{1K} \\ \vdots \\ H_{NK} \end{matrix} \end{bmatrix}}$$

$$H'_{ik} = \frac{H_{ik} - \mu_k}{\sigma_k}$$

Batch Normalization

Training time:

Batch of activations for a given layer to normalize

$$\boxed{H = \begin{bmatrix} H_{11} & \cdots & H_{1K} \\ \vdots & \ddots & \vdots \\ H_{N1} & \cdots & H_{NK} \end{bmatrix}}$$

$$H'_{ik} = \frac{H_{ik} - \mu_k}{\sigma_k}$$

$$\mu_k = \frac{1}{N} \sum_i H_{ik}$$

Mean activations across batch for node k.

Batch Normalization

When calculating the variance, we add a small constant to the variance to prevent potential divisions by zero.

Training time:

Batch of activations for a given layer to normalize

$$H = \begin{bmatrix} H_{11} & \cdots & H_{1K} \\ \vdots & \ddots & \vdots \\ H_{N1} & \cdots & H_{NK} \end{bmatrix}$$

$$H'_{ik} = \frac{H_{ik} - \mu_k}{\sigma_k}$$

$$\mu_k = \frac{1}{N} \sum_i H_{ik}$$

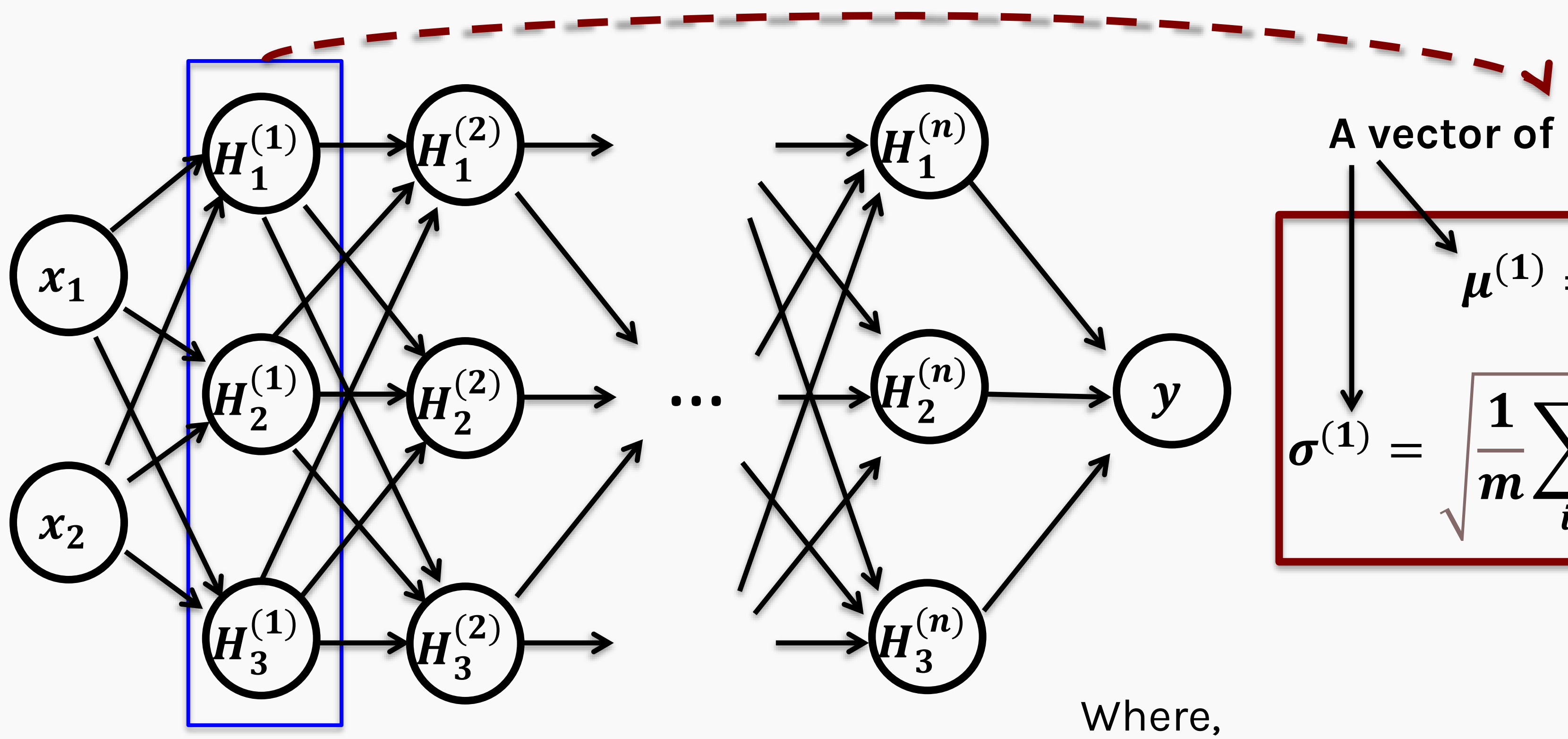
Mean activatio across batch for node

$$\sigma_k = \sqrt{\frac{1}{N} \sum_i (H_{ik} - \mu_k)^2 + \delta}$$

SD of each unit across batch

Batch Normalization

Batch 1



A vector of 3 values

$$\boxed{\begin{aligned}\mu^{(1)} &= \frac{1}{m}\sum_i H_i^{(1)} \\ \sigma^{(1)} &= \sqrt{\frac{1}{m}\sum_i \left(H_i^{(1)} - \mu^{(1)}\right)^2 + \delta}\end{aligned}}$$

Where,

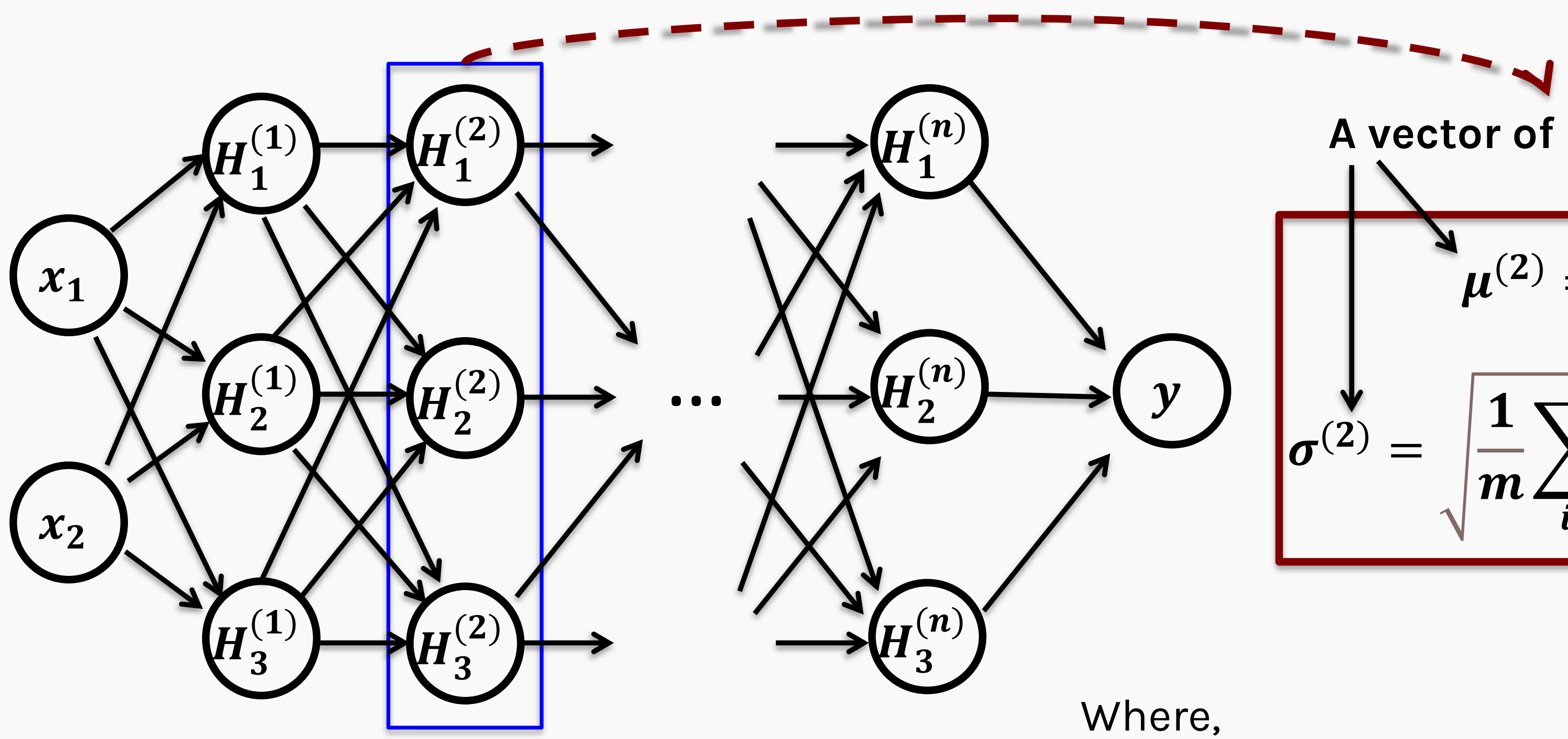
m: Number of training examples in the batch

$H_i^{(1)}$: Hidden layer activation of the first hidden layer for the i^{th} training example

δ: A small constant

Batch Normalization

Batch 1



A vector of 3 values

$$\mu^{(2)} = \frac{1}{m}\sum_i H_i^{(2)}$$

$$\sigma^{(2)} = \sqrt{\frac{1}{m}\sum_i \left(H_i^{(2)} - \mu^{(2)}\right)^2 + \delta}$$

Where,

m: Number of training examples in the batch

$H_i^{(2)}$: Hidden layer activation of the second hidden layer for the i^{th} training example

δ: A small constant

Batch Normalization

Training time:

- Normalization can reduce expressive power
- Instead use:

$$\boxed{H'_{ik} = \gamma H'_{ik} + \beta}$$

Batch Normalization

Training time:

- Normalization can reduce expressive power
- Instead use:

$$\boxed{H'_{ik} = \underbrace{\gamma}_{\text{Learnable parameters}} H'_{ik} + \beta}$$

Learnable parameters

Batch Normalization

When do we normalize: before or after activation?

Before activation

Batch Normalization

We have the equation

$$h^{(2)} = Wa^{(1)} + b$$

where

$a^{(1)}$: Activation of the first hidden layer

$h^{(2)}$: the output of the second hidden layer w/o activation

If we do batch normalization **after** activation:

The shape of the distribution of $a^{(1)}$ is likely to change during training and limiting its mean and standard deviation will not eliminate covariate shift.

Batch Normalization

We have the equation

$$h^{(2)} = Wa^{(1)} + b$$

where

$a^{(1)}$: Activation of the first hidden layer

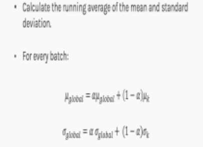
$h^{(2)}$: the output of the second hidden layer w/o activation

If we do batch normalization **before** activation:

$Wa^{(1)} + b$ is very likely to have a symmetric, non-sparse distribution; normalizing it is likely to produce activations with a stable distribution.

Evaluation

We saw how batch normalization works during training, but what about **prediction**, when we might not have a complete batch!


• Calculate the running average of the mean and standard deviation.

• For every batch:

$\mu_{global} = \alpha\mu_{global} + (1-\alpha)\mu_k$

$\sigma_{global} = \alpha\sigma_{global} + (1-\alpha)\sigma_k$

Evaluation

Evaluation time:

- Calculate the running average of the mean and standard deviation.

- For every batch:

Decay parameter

Use this for evaluation

$$\mu_{global} = \alpha\mu_{global} + (1-\alpha)\mu_k$$

$$\sigma_{global} = \alpha\sigma_{global} + (1-\alpha)\sigma_k$$

Batch Normalization

Evaluation time:

Hidden activations will be a vector as there are no batches.

$$\boxed{H = [H_1 \quad \ldots \quad H_K]}$$

Batch Normalization

Evaluation time:

Use the global statistics to normalize the node activations.

$$\boxed{H = [H_1 \quad \dots \quad H_K]}$$

For each hidden node k:

$$H_k' = \frac{H_k - \mu_{global}}{\sigma_{global}}$$

Estimated global mean of each unit activation.

Estimated global SD of each unit activation.