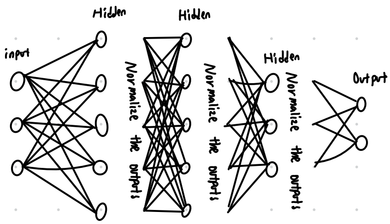
Normalization

: We can design our models such that they are easier to optimize

Types of Normalization

- Batch normalization : Used in ResNets and many other models
- Instance normalization
- Layer normalization : Used in Transformer
- Group normalization
- ...



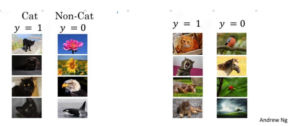
the quality or condition of being in good condition

Normalization layers are often used to increase training robustness of neural network

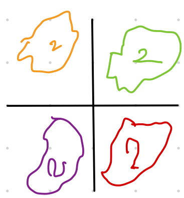
Why we need normalization? : Main issue is "Covariate shift".

- Training assumes the training data are similarly distributed. However, in practice, each minibatch may have different distribution : "Covariate Shift". It may occur in each layer of the network.

Ex)



Ex2)



: Think MNIST dataset.

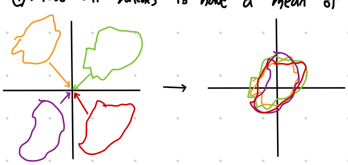
Training assumes that distributions of digit '2' are all similarly distributed, and it's very naive approach.

We need to apply normalization to make each digits' distributions as similar as possible.

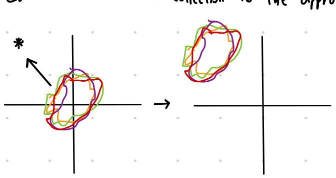
Solution: Move all minibatches to a "standard" location

General Concept of Normalization

① Move all batches to have a mean of 0 and unit standard deviation : This process eliminates covariate shift between batches



② Move the entire collection to the appropriate location

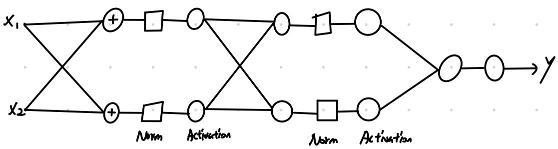


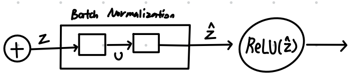
Batch normalization is a "shift-adjustment unit" that happens after the weighted addition of inputs but before the application of activation.

- Is done independently for each unit, to simplify computation.

Ex)
```
self.basis = nn.Sequential(
        nn.Linear(x_dim, 100),
        nn.BatchNorm(100),
        nn.ReLU(),
          ⋮
)
```

In pictorial depiction

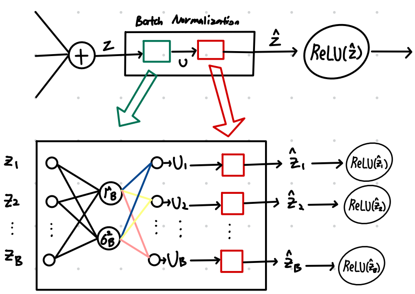




- $z = \sum_j w_j x_j + b$, $w_1x_1 + w_2x_2 + b = w^T x + b$.
- $u = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$: Centering : Scaling , where $\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$ (B: Batch size) and $\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i - \mu_B)^2$
- $\hat{z} = \gamma u_i + \beta$, where γ, β are learnable features, and they're responsible for shifting distributions into appropriate locations.

☆Tip: Batch Normalization prefers larger batches.

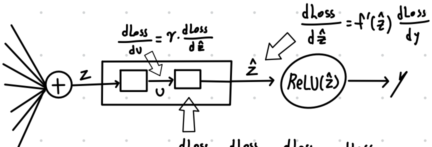
A better picture for batch norm more explicitly



where $u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$, $\hat{z}_i = \gamma u_i + \beta$

We can compute $\frac{\partial Loss}{\partial u_i}$ individually for each u_i because the processing after the computation of u_i is independent for each u_i

Batch normalization : Backpropagation



$\frac{dLoss}{d\hat{z}} = f'(\hat{z})\frac{dLoss}{dy}$

$\frac{dLoss}{du} = \gamma \cdot \frac{dLoss}{d\hat{z}}$

$\frac{dLoss}{d\beta} = \frac{dLoss}{d\hat{z}}$, $\frac{dLoss}{d\gamma} = u \cdot \frac{dLoss}{d\hat{z}}$

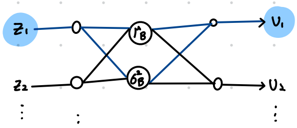
Complete derivative of the batch-loss w.r.t. z_i

$$\frac{d\,Loss}{dz_i} = \sum_j \frac{d\,Loss}{dU_j}\,\frac{dU_j}{dz_i}$$

We know $\frac{d\,Loss}{dU_j} = \gamma \cdot \frac{d\,Loss}{d\hat{z}_j} = \gamma \cdot f'(\hat{z}_j)\,\frac{d\,Loss}{dy}$

$\frac{dU_j}{dz}$?

① The derivative for the "through" line $(i=j)$



$$\frac{dU_i}{dz_i} = \frac{\partial U_i}{\partial z_i} + \frac{\partial U_i}{\partial \mu_B}\frac{\partial \mu_B}{\partial z_i} + \frac{\partial U_i}{\partial \sigma_B^2}\frac{d\sigma^2}{dz_i}$$

1.1: $U_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2+\epsilon}}$, then $\frac{\partial U_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left(\frac{1}{\sqrt{\sigma_B^2+\epsilon}}\cdot(z_i-\mu_B)\right) = \frac{1}{\sqrt{\sigma_B^2+\epsilon}}$

1.2: $U_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2+\epsilon}}$, then $\frac{\partial U_i}{\partial \mu_B} = \frac{-1}{\sqrt{\sigma_B^2+\epsilon}}$

1.3: $\mu_B = \frac{1}{B}\sum_{i=1}^{B} z_i$, then $\frac{\partial \mu_B}{\partial z_i} = \frac{1}{B}$

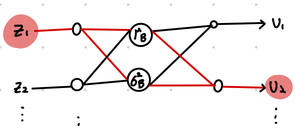
1.4: $U_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2+\epsilon}}$, then $\frac{\partial U_i}{\partial \sigma_B^2} = \frac{\partial}{\partial \sigma_B^2}\left((z_i-\mu_B)\cdot(\sigma_B^2+\epsilon)^{-\frac{1}{2}}\right) = -\frac{(z_i-\mu_B)}{2}\cdot(\sigma_B^2+\epsilon)^{-\frac{3}{2}} = \frac{(z_i-\mu_B)}{2(\sigma_B^2+\epsilon)^{\frac{3}{2}}}$

1.5: $\sigma_B^2 = \frac{1}{B}\sum_{i=1}^{B}(z_i-\mu_B)^2$, then $\frac{d\sigma_B^2}{dz_i} = \frac{\partial \sigma_B^2}{\partial z_i} + \frac{\partial \sigma_B^2}{\partial \mu_B}\frac{d\mu_B}{dz_i} = \frac{2(z_i-\mu_B)}{B} + 0$

Hence, the derivative for the through line $(i=j)$

$$\frac{dU_i}{dz_i} = \frac{1}{\sqrt{\sigma_B^2+\epsilon}} - \frac{1}{B\sqrt{\sigma_B^2+\epsilon}} + \frac{-(z_i-\mu_B)^2}{B(\sigma_B^2+\epsilon)^{\frac{3}{2}}}$$

② The derivative for the "cross" lines $(i \neq j)$



$$\frac{dU_j}{dz_i} = \frac{\partial U_j}{\partial \mu_B}\frac{d\mu_B}{dz_i} + \frac{\partial U_j}{\partial \sigma_B^2}\frac{d\sigma_B^2}{dz_i}$$

$$= \frac{-1}{B\sqrt{\sigma_B^2+\epsilon}} - \frac{(z_i-\mu_B)^2}{B(\sigma_B^2+\epsilon)^{3/2}}$$

① + ②

$$\frac{dU_j}{dz_i} = \begin{cases} \dfrac{1}{\sqrt{\sigma_B^2+\varepsilon}} - \dfrac{1}{B\sqrt{\sigma_B^2+\varepsilon}} + \dfrac{-(z_i-\mu_B)^2}{B(\sigma_B^2+\varepsilon)^{\frac{3}{2}}}, & \text{if } i=j \\ \dfrac{-1}{B\sqrt{\sigma_B^2+\varepsilon}} - \dfrac{(z_i-\mu_B)^2}{B(\sigma_B^2+\varepsilon)^{3/2}}, & \text{if } i\neq j \end{cases}$$

And, recall the complete derivative of the mini-batch loss w.r.t. z_i

$$\frac{d\,Loss}{dz_i} = \sum_j \frac{d\,Loss}{dU_j}\,\frac{dU_j}{dz_i}$$