Normalization, Regularization

11-785 Introduction to Deep Learning

- lecture 8 -

TAVE Research DL001
Heeji Won

Contents

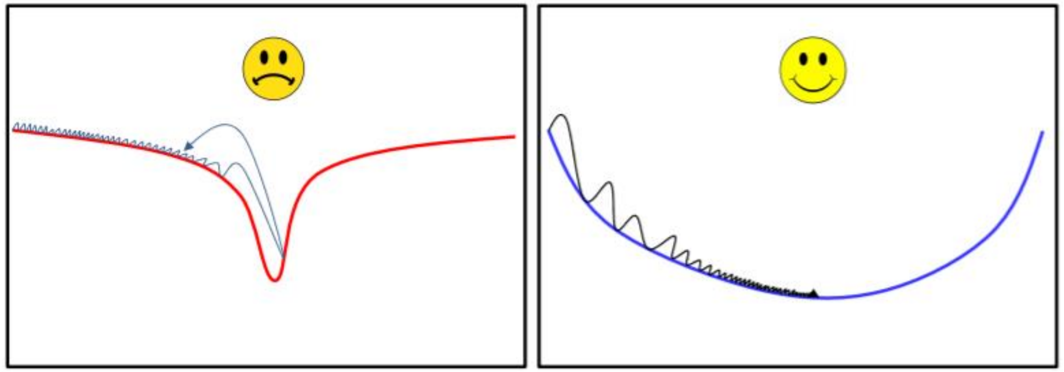
Contents

01. Divergence

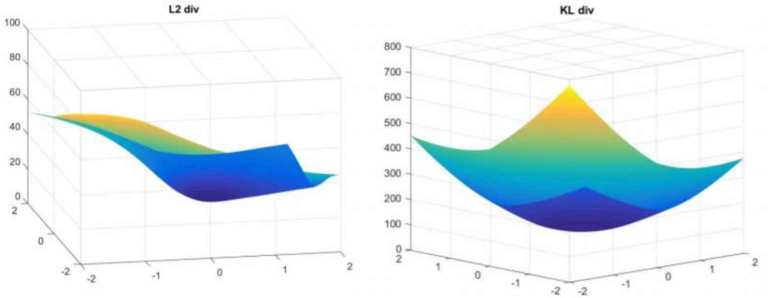
"The convergence of the gradient descent depends on the divergence"

$$Loss = \frac{1}{T}\sum_{t} Div(\mathbf{Y}_t, \mathbf{d}_t; \mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K)$$

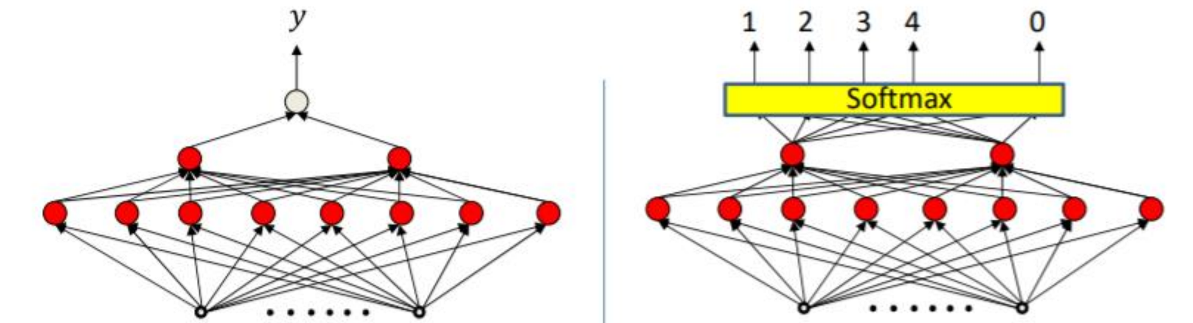
The best type of divergence is steep far from the optimum, but shallow at the optimum



- L2 vs KL



  - L2 is popular for networks that perform numeric prediction/regression
  - KL is popular for networks that perform classification
  - L2 is not convex while KL is convex

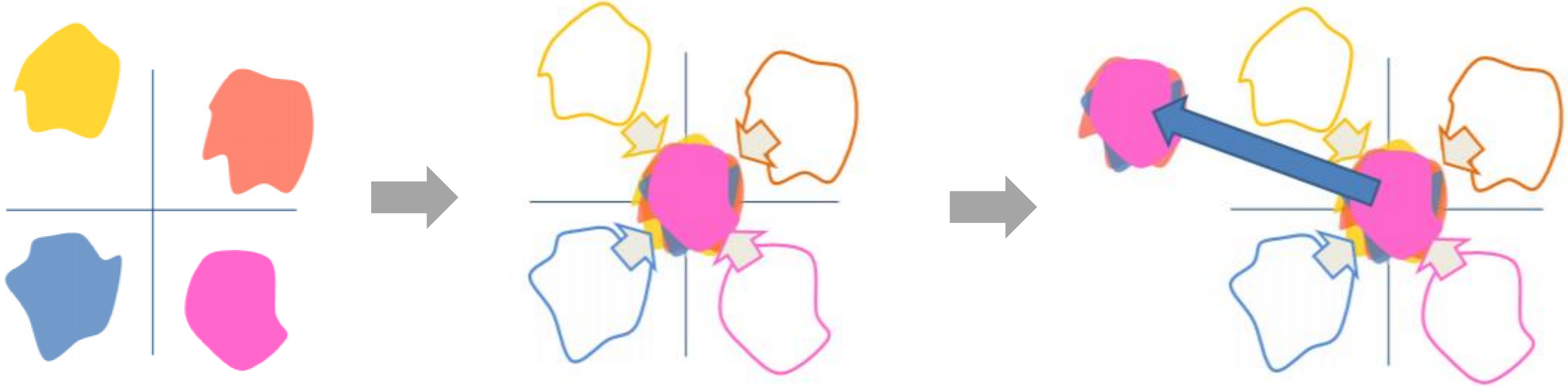


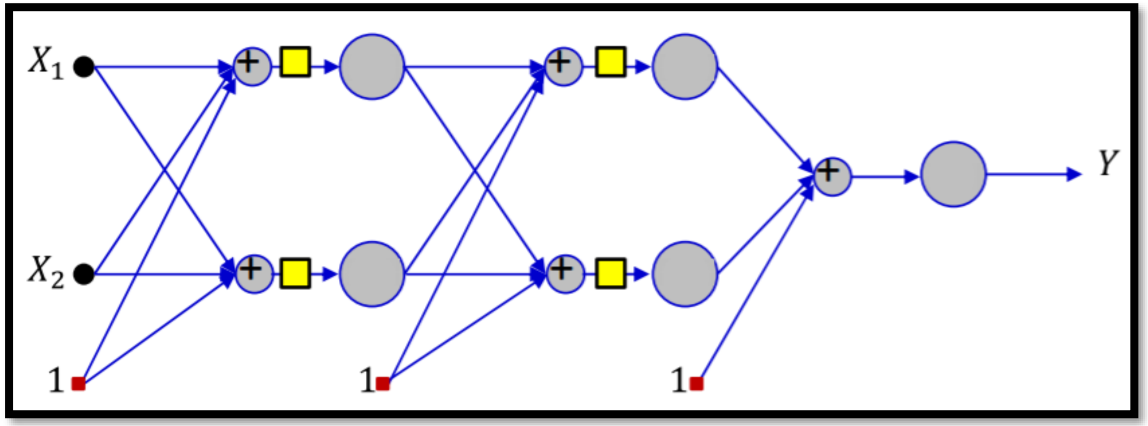
Desired output:	d	Desired output: $[0,0,\ldots,1,\ldots,0]$
L2	$Div = \frac{1}{2}(y-d)^2$	$Div = \frac{1}{2}\sum_i (y_i - d_i)^2$
KL	$Div = -d\log(y) - (1-d)\log(1-y)$	$Div = \sum_i d_i \log(d_i) - \sum_i d_i \log(y_i)$

Contents

02. Batch Normalization

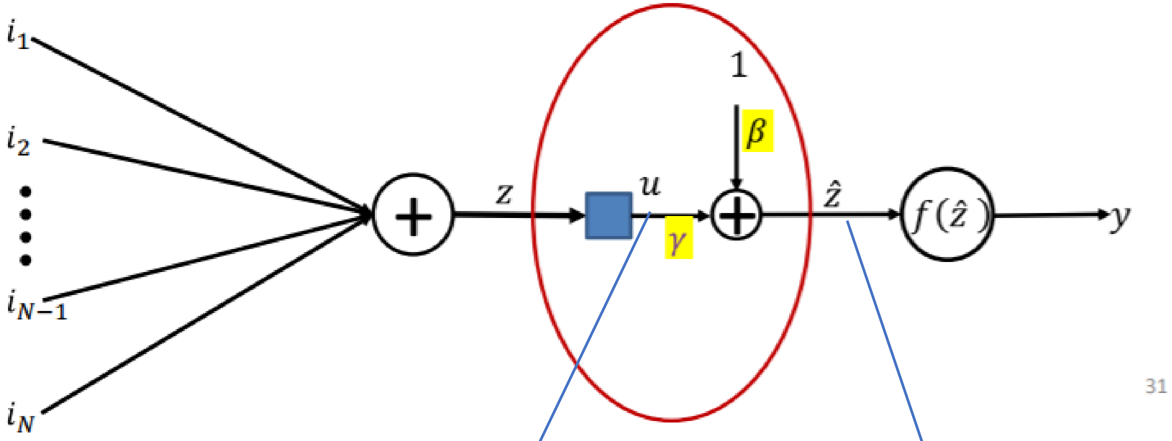
- The solution for covariate shifts
  - The problem is each minibatch may have a different distribution
  - So, normalize batches





- Batch normalization is a covariate adjustment unit that happens after the weighted addition of inputs
- Is done independently for each unit
- The adjustment occurs over individual minibatches

02. Batch Normalization



$$\boxed{u_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}}$$

Normalize minibatch to
zero−mean unit variance

$$\boxed{\hat{z}_i = \gamma u_i + \beta}$$

Neuron-specific terms

Shift to right
position

$$Loss(minibatch)$$
$$= \frac{1}{B}\sum_t Div\left(Y_t\left(X_t, \mu_B(X_t, X_{t'\neq t}), \sigma_B^2\left(X_t, X_{t'\neq t}, \mu_B(X_t, X_{t'\neq t})\right)\right), d_t(X_t)\right)$$

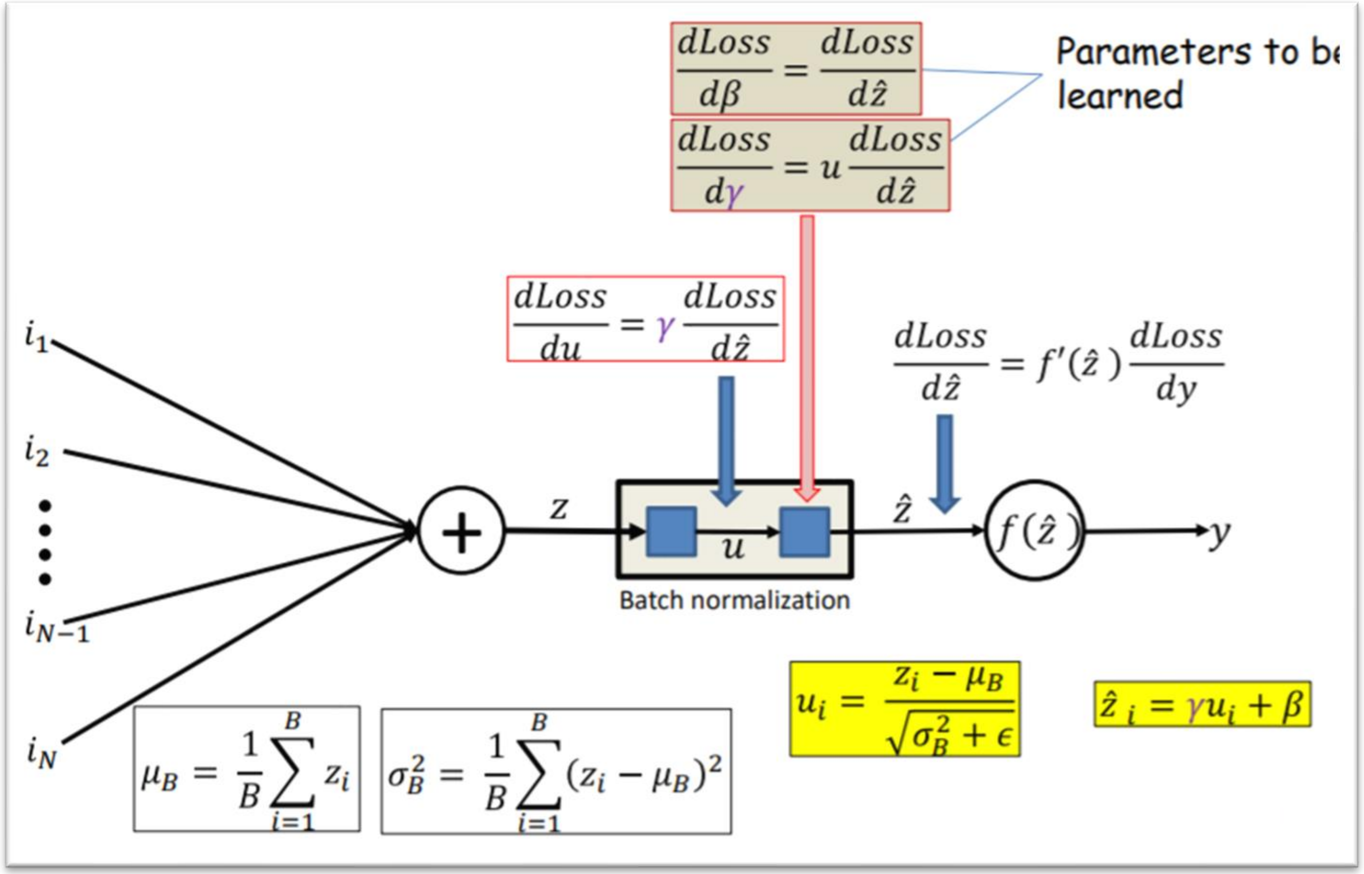
✓ In the case of Inference,
use the average over all training minibatches

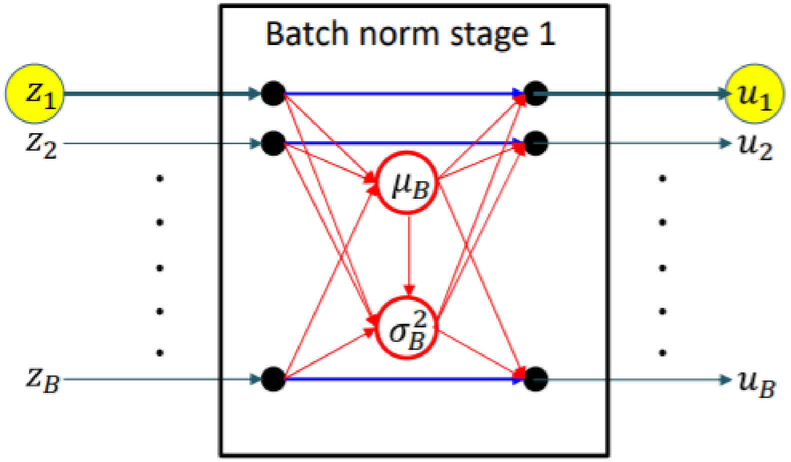
$$\mu_{BN} = \frac{1}{Nbatches}\sum_{batch} \mu_B(batch)$$

$$\sigma_{BN}^2 = \frac{B}{(B-1)Nbatches}\sum_{bat} \sigma_B^2(batch)$$

02. Batch Normalization

- Backpropagation





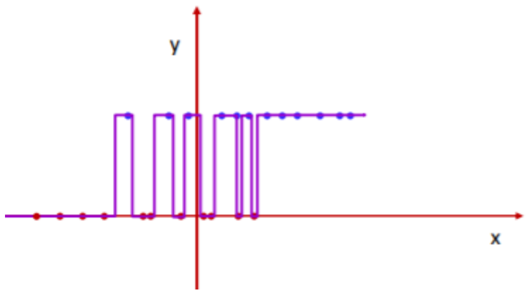
$$\frac{du_i}{dz_i} = \frac{\partial u_i}{\partial z_i} + \frac{\partial u_i}{\partial \mu_B}\frac{d\mu_B}{dz_i} + \frac{\partial u_i}{\partial \sigma_B^2}\frac{d\sigma_B^2}{dz_i}$$

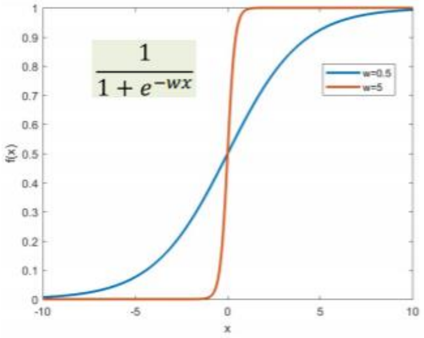
$$\frac{du_j}{dz_i} = \begin{cases} \frac{1}{\sqrt{\sigma_B^2+\epsilon}} + \frac{-1}{B\sqrt{\sigma_B^2+\epsilon}} + \frac{-(z_i-\mu_B)^2}{B(\sigma_B^2+\epsilon)^{3/2}} & \text{if } j = i \\ \frac{-1}{B\sqrt{\sigma_B^2+\epsilon}} + \frac{-(z_i-\mu_B)^2}{B(\sigma_B^2+\epsilon)^{3/2}} & \text{if } j \neq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

Contents

03. Solutions for overfitting

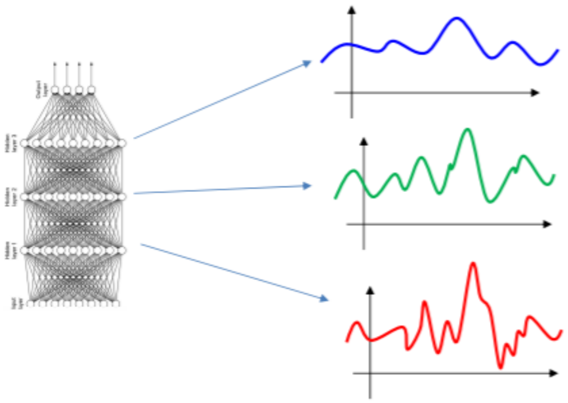
- The unconstrained model





As |w| increases, the response becomes steeper

- Deeper networks



Deeper networks impose more smoothness than shallow ones

- Regularized training

$$L(W_1, W_2, \dots, W_K) = \frac{1}{T}\sum_t Div(Y_t, d_t; W_1, W_2, \dots, W_K) + \frac{1}{2}\lambda \sum_k \|W_k\|_F^2$$

– Increasing λ assigns greater importance to shrinking the weights

$$L(W_1, W_2, \dots, W_K) = \frac{1}{T}\sum_t Div(Y_t, d_t) + \frac{1}{2}\lambda \sum_k \|W_k\|_F^2$$

- Batch mode:

$$\Delta W_k = \frac{1}{T}\sum_t \nabla_{W_k} Div(Y_t, d_t)^T + \lambda W_k$$

- SGD:

$$\Delta W_k = \nabla_{W_k} Div(Y_t, d_t)^T + \lambda W_k$$

- Minibatch:

$$\Delta W_k = \frac{1}{b}\sum_{\tau=t}^{t+b-1} \nabla_{W_k} Div(Y_\tau, d_\tau)^T + \lambda W_k$$

- Update rule:

$$W_k \leftarrow W_k - \eta \Delta W_k$$

Thank you