DNN

Let **X** be the data matrix, where i[th] example is in the i[th] row of **X**. We have our forward computation in Neural Network as follows:

$$\boldsymbol{a^1 = XW^1 + b^1}$$
$$\boldsymbol{z^1} = \mathrm{ReLU}(\boldsymbol{a^1})$$
$$\boldsymbol{a^2 = z^1 W^2 + b^2}$$
$$\boldsymbol{z^2} = \mathrm{ReLU}(\boldsymbol{a^2})$$
$$\boldsymbol{a^3 = z^2 W^3 + b^3}$$
$$\boldsymbol{z^3} = \mathrm{Sigmoid}(\boldsymbol{a^3})$$

Loss (E) = Cross Entropy (z^3, y)

$$z_i^3 = \frac{e^{a_i^3}}{\sum_{j=1}^{C} e^{a_j^3}}$$

if i = j: $$\frac{\delta z_i^3}{\delta a_i^3} = \frac{e^{a_i^3}\left(\sum_{j=1}^{C} e^{a_j^3}\right) - e^{a_i^3} e^{a_i^3}}{\left(\sum_{j=1}^{C} e^{a_j^3}\right)^2} = \frac{e^{a_i^3}}{\sum_{j=1}^{C} e^{a_j^3}}\left(1 - \frac{e^{a_i^3}}{\sum_{j=1}^{C} e^{a_j^3}}\right) = z_i^3(1 - z_i^3)$$

if i ≠ j: $$\frac{\delta z_i^3}{\delta a_j^3} = \frac{0 - e^{a_i^3} e^{a_j^3}}{\left(\sum_{j=1}^{C} e^{a_j^3}\right)^2} = -\frac{e^{a_i^3}}{\sum_{j=1}^{C} e^{a_j^3}} \frac{e^{a_j^3}}{\sum_{j=1}^{C} e^{a_j^3}} = -z_i^3 z_j^3$$

$$\Delta_i^3 = \frac{\delta E}{\delta a_i^3} = \frac{-\sum_{j=1}^{C} \delta t_j \log(z_j^3)}{\delta a_i^3} = -\sum_{j=1}^{C} t_j \frac{\delta \log(z_j^3)}{\delta a_i^3} = -\sum_{j=1}^{C} t_j \frac{1}{z_j^3} \frac{\delta z_j^3}{\delta a_i^3}$$

$$= -\frac{t_i}{z_i^3} \frac{\delta z_i^3}{\delta a_i^3} - \sum_{j \neq i} \frac{t_j}{z_j^3} \frac{\delta z_j^3}{\delta a_i^3} = -\frac{t_i}{z_i^3} z_i^3(1 - z_i^3) - \sum_{j \neq i} \frac{t_j}{z_j^3}(-z_i^3 z_j^3)$$

$$= -t_i + t_i z_i^3 + \sum_{j \neq i} t_j z_i^3 = -t_i + \sum_{j=1}^{C} t_j z_i^3$$

$$= z_i^3 - t_i$$

Since a^3 is single dimension $\Delta^3 = z^3 - t$

$$\frac{\delta E}{\delta b^3} = \frac{\delta E}{\delta a^3} \frac{\delta a^3}{\delta b^3} = \Delta^3$$

In vector notation $\boldsymbol{\nabla_{b^3} E = \Delta^3}$

$$\frac{\delta E}{\delta W_{ij}^3} = \frac{\delta E}{\delta a_j^3} \frac{\delta a_j^3}{\delta W_{ij}^3} = \Delta_j^3 z_i^2$$

W^3 is a vector, so we have $\boldsymbol{\nabla_{W^3} E = \Delta^3 z^2}$

$$\Delta_i^2 = \frac{\delta E}{\delta a_i^2} = \frac{\delta E}{\delta a^3} \frac{\delta a^3}{\delta a_i^2} = \frac{\delta E}{\delta a^3} \frac{\delta a^3}{\delta z_i^2} \frac{\delta z_i^2}{\delta a_i^2}$$

$$= \Delta^3 W_i^3 1_{\{a_i^2 > 0\}}$$

1_z is indicator function for z.

In vector notation $\boldsymbol{\Delta^2 = \Delta^3 (W^3 \odot 1_{\{a^2>0\}})}$

$$\frac{\delta E}{\delta b_i^2} = \frac{\delta E}{\delta a_i^2}\frac{\delta a_i^2}{\delta b_i^2} = \Delta_i^2$$

In vector notation $\boldsymbol{\nabla_{b^2} E = \Delta^2}$

$$\frac{\delta E}{\delta W_{ij}^2} = \frac{\delta E}{\delta a_j^2}\frac{\delta a_j^2}{\delta W_{ij}^2} = \Delta_j^2 z_i^1$$

In Matrix notation $\boldsymbol{\nabla_{W^2} E = z^1 (\Delta^2)^T}$

$$\Delta_i^2 = \frac{\delta E}{\delta a_i^1} = \sum_j \frac{\delta E}{\delta a_j^2}\frac{\delta a_j^2}{\delta a_i^1} = \sum_j \frac{\delta E}{\delta a_j^2}\frac{\delta a_j^2}{\delta z_i^1}\frac{\delta z_i^1}{\delta a_i^1}$$

$$= \sum_j \Delta_j^2 W_{ij}^2 1_{\{a_i^1>0\}}$$

In vector notation $\boldsymbol{\Delta^1 = 1_{\{a^1>0\}} \odot (W^2 \Delta^2)}$

$$\frac{\delta E}{\delta b_i^1} = \frac{\delta E}{\delta a_i^1}\frac{\delta a_i^1}{\delta b_i^1} = \Delta_i^1$$

In vector notation $\boldsymbol{\nabla_{b^1} E = \Delta^1}$

$$\frac{\delta E}{\delta W_{ij}^1} = \frac{\delta E}{\delta a_j^1}\frac{\delta a_j^1}{\delta W_{ij}^1} = \Delta_j^1 x_i$$

In Matrix notation $\boldsymbol{\nabla_{W^1} E = x (\Delta^2)^T}$

Once forward pass is done we can calculate $\nabla_{W^3}(t), \nabla_{b^3}(t), \nabla_{W^2}(t), \nabla_{b^2}(t), \nabla_{W^1}(t), \nabla_{b^1}(t)$ and update the weight as:

$$\boldsymbol{v_{W^3}(t+1) = \gamma v_{W^3}(t) + \eta \nabla_{W^3}(t)}$$
$$\boldsymbol{W^3(t+1) = W^3(t) - v_{W^3}(t+1)}$$

$$\boldsymbol{v_{b^3}(t+1) = \gamma v_{b^3}(t) + \eta \nabla_{b^3}(t)}$$
$$\boldsymbol{b^3(t+1) = b^3(t) - v_{b^3}(t+1)}$$

$$\boldsymbol{v_{W^2}(t+1) = \gamma v_{W^2}(t) + \eta \nabla_{W^2}(t)}$$
$$\boldsymbol{W^2(t+1) = W^2(t) - v_{W^2}(t+1)}$$

$$\boldsymbol{v_{b^2}(t+1) = \gamma v_{b^2}(t) + \eta \nabla_{b^2}(t)}$$
$$\boldsymbol{b^2(t+1) = b^2(t) - v_{b^2}(t+1)}$$

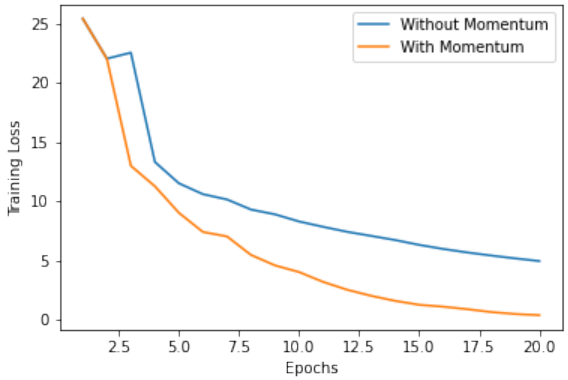
$$\boldsymbol{v_{W^1}(t+1) = \gamma v_{W^1}(t) + \eta \nabla_{W^1}(t)}$$
$$\boldsymbol{W^1(t+1) = W^1(t) - v_{W^1}(t+1)}$$

$$\boldsymbol{v_{b^1}(t+1) = \gamma v_{b^1}(t) + \eta \nabla_{b^1}(t)}$$
$$\boldsymbol{b^1(t+1) = b^1(t) - v_{b^1}(t+1)}$$

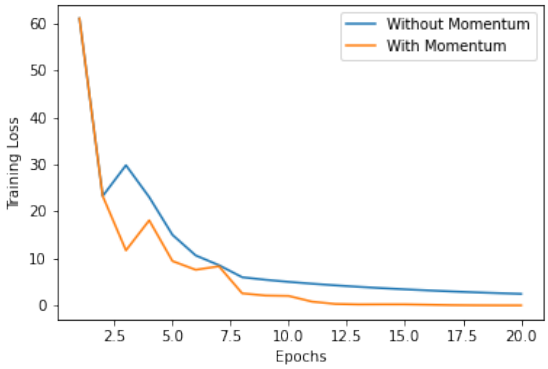
Here γ is momentum parameter, its set to zero when we don't use momentum. All the momentum vectors are initially initialized to zero.

We took two neural network with first neural network having 12 neurons in hidden layer and second having 15 neurons in hidden layer. We trained both the neural network with momentum and without

momentum, in both the cases, we initialized them same wights. We used Xavier initialization and He initialization for initializing weights of out neural network. Following figure shows comparison of momentum for both Neural Network (we choose initial weight of the neural network to be the same)



Neural Network with 12 hidden nodes



Neural Network with 15 hidden nodes

We notice that, with momentum the network is converging fast. So momentum is improving our learning process.

When we increased the number of layers from 12 to 15, most of time the network was over-fitting the data. So test accuracy of 15-node hidden layer network is less than 12-node hidden layer network.
The accuracy of neural network with 12 nodes in hidden layer was about 80%, while the accuracy of neural network with 15 nodes in hidden layer was only 60%.