Realization Of Identity Function Using Two Node Neural Network

A simple function, the Identity Function y = x, is used to explore Stochastic, Mini-Batch and Batch processing with Gradient Descent(GD), Gradient Descent With Momentum(GDM), RMSProp and Adam optimizer.

# **Network**

The predicted output, , is calculated given the input, , along with weight, , and bias, . The predicted outputis then compared against the actual output using Mean Squared Error Loss, , which gives a measure of how far apart the two outputs are. The gradient of the loss with respect to weight and bias, and respectively, helps in moving in the right direction so as to minimize the loss.

Actual Output :

Predicted Output :

The general flow is defined as below:

Forward Propagation **:**

Loss **:**

Gradients **:**

**:**

Updates :

:

The data processing method, Stochastic, Mini-Batch or Batch processing, determines how often is the weight and bias updated. The optimizer method, GD, GDM, RMSProp or Adam, determines how to use the gradient information.

# **Input Sample Processing**

## **Stochastic Processing**

The weight and bias is updated for every input sample.

## **Mini-Batch Processing**

The gradient is averaged over M(Mini-Batch Size) samples. The weight and bias is updated after every M samples.

**,**

**,**

## **Batch Processing**

**,**

**,**

The gradient is averaged over all the N samples. The weight and bias is updated after N samples.

# **Optimizer**

and will be used in the below equations and they can be either Stochastic, Mini-Batch or Batch processing output.

## **Gradient Descent**

The gradient is used as it is to update the weight and bias.

## **Gradient Descent With Momentum**

The exponential moving average of the gradient is used to update the weight and bias.

## **Root Mean Square Prop**

The exponential moving average of the squared gradient is used to update the weight and bias.

## **Adam**

The exponential moving average of both the gradient and squared gradient is used to update the weight and bias.

# **Performance**

As this is an Identity Function, we know that that the weight should be 1 and bias 0. The training is stopped when both the weight and bias reach within 10E-4 of the expected value.

## **Stochastic Processing with Gradient Descent**

With learning rate, set to 0.001, the weight and bias explodes(obtain high magnitude) immediately thereby making the network useless.

Within 1815 samples, the weight and bias approach infinity. Below graph shows the high oscillation in weight within just ~50 samples.

Lowering to 0.0001 prevents the weights from exploding and we get a usable network.

## **Stochastic Processing with Gradient Descent with Momentum(** = 0.0001)

## **Stochastic Processing with RMS(** = 0.001)

## **Stochastic Processing with Adam(** = 0.1)

## **Mini-Batch Processing with Gradient Descent(** = 0.0001)

## **Mini-Batch Processing with Gradient Descent with Momentum(** = 0.001)

## **Mini-Batch Processing with RMS(** = 0.001)

## **Mini-Batch Processing with Adam(** = 0.1)

## **Batch Processing with Gradient Descent(** = 0.0001)

## **Batch Processing with Gradient Descent with Momentum(** = 0.0001)

## **Batch Processing with RMS(** = 0.00001)

## **Batch Processing with Adam(** = 0.00001)

## **Comparative** **Graph**

Below table shows the weight profile under different methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weight | GD | GDM | RMSProp | Adam |
| Stochastic |  |  |  |  |
| Mini-Batch |  |  |  |  |
| Batch |  |  |  |  |

Below table shows the bias profile under different methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bias | GD | GDM | RMSProp | Adam |
| Stochastic |  |  |  |  |
| Mini-Batch |  |  |  |  |
| Batch |  |  |  |  |

Below table shows the maximum learning rate used(above which the weights explode or there is no learning) to achieve the required accuracy under different methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weight | GD | GDM | RMSProp | Adam |
| Stochastic | 0.0001 | 0.0001 | 0.001 | 0.1 |
| Mini-Batch | 0.0001 | 0.001 | 0.001 | 0.1 |
| Batch | 0.0001 | 0.0001 | 0.00001 | 0.00001 |

One can observe that the oscillations reduce when moving from Stochastic to Batch. This is because the weights are updated frequently, after every sample, in Stochastic processing whereas the weights are updated only once for the entire batch of data in Batch processing. Also the amount by which the weights are updated in Batch Processing is very small due to the need of very small learning rate.

# **Timing**

Training time of the model, for weight and bias to reach an accuracy of 0.0001.

## **Absolute** **Value**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Training Time(s) | GD | GDM | RMSProp | Adam |
| Stochastic | **1.2531726** | **0.1571339** | **0.0252543** | **0.0081627** |
| Mini-Batch | **1.4777563** | **0.0142058** | **0.0204918** | **0.0049806** |
| Batch | **180.1724363** | **157.7865816** | **50.1907914** | **48.8349409** |

## **Relative** **Value**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Training Time | GD | GDM | RMSProp | Adam |
| Stochastic | **252** | **32** | **6** | **2** |
| Mini-Batch | **297** | **3** | **5** | **1** |
| Batch | **36175** | **31681** | **10078** | **9806** |

# **Conclusion**

Mini-Batch Processing with Adam Optimizer is the best model and Batch Processing with traditional Gradient Descent optimizer is the worst model taking 36175 times more time to reach the required weight and bias accuracy.

Model parameters used in Mini-Batch Processing with Adam Optimizer:

|  |  |
| --- | --- |
| Epochs taken | 1 |
| Mini-Batch Size, M | **64** |
| Learning Rate, | **0.1** |
| Beta1, | **0.9** |
| Beta2, | **0.999** |