End-Sem (Major) Exam Report

Submitted By : Anurag Saraswat (M20CS066)

Paper Link	LINK
Author's Code	LINK
Part 2 Collab Link (With dataset use in paper)	LINK
Part 3 Collab Link (With dataset not use in paper)	LINK
Part 2 and 3 Comparison Collab Link	LINK

Learning compositional functions via multiplicative weight updates

1. Summary

Author proposes and benchmark Madam—a multiplicative version of the Adam optimiser.Empirically, Madam seems to not require learning rate tuning. Further, it may be used to train neural networks with low bit width synapses(a **synapse** is a small gap at the end of a neuron that allows a signal to pass from one neuron to the next) stored in a logarithmic number system.

MADAM optimiser is defined as below:

```
Madam(net.parameters(), lr=0.01, p_scale=3.0, g_bound=10.0)
```

Where, **net.parameters()** are parameters of the network to be learnt. **lr** is the learning rate. **p_scale** controls the size of the optimization domain. **g_bound** is a gradient bound used for clipping gradients.

A typical MADAM update is as follow:

$$w \dashrightarrow w \exp(\pm lr)$$

Largest possible update to parameter is :

$$w \dashrightarrow w \exp(\pm g\_bound \times lr)$$

And finally parameters are clipped to lie within range ± init_scale x p_scale

Algorithm for Multiplicative Adaptive Moments based optimiser(MADAM) Optimizer:

σ : initial weight scale
σ* : max weight
η : typical perturbation
η* : max perturbation
β : averaging constant

Weight initialisation: initialise weights randomly on scale σ,
for example: W ~ NORMAL(0, σ).

1. $\check{g} \leftarrow 0$, initialise second moment estimate repeat.
2. $g \leftarrow$ StochasticGradient() ,collecting gradient.
3. $\check{g}^2 \leftarrow (1 - \beta)g^2 + \beta\check{g}^2$, update second moment estimate.
4. $W \leftarrow W \odot \exp[-\eta \text{ sign } W \odot \text{clamp}_{\eta*/\eta}(g/\check{g})]$,update weights multiplicatively.
5. $W \leftarrow \text{clamp}_{\sigma*}(W)$, clamp weights between ±σ*.
6. Repeat step 2 , until convergence.

Author's Finding:

- Across various tasks, including image classification, language modeling and image generation—Madam without learning rate tuning is competitive with a tuned SGD or Adam.
- MADAM performed worse on the Imagenet experiment, achieving a test error 4.8% worse than SGD.
- For Madam, the best setting (of η) was independent of task.
- The multiplicative nature of Madam suggests storing synapse strengths in a logarithmic number system, where numbers are represented just by a sign and exponent.

2. Reproduced results on one of the databases used in the paper

Dataset

Reproducing result on CIFAR-10 dataset use by author. It consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.
Class Present in the dataset is Air Plane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck.

Visualization of 10 random sample is as follow:

Model Information:

In author code various CNN architectures are used for evaluation. Here, I am defining my own architecture for evaluation.
CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers.

Arrangement of these layers are as follow:

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 30, 30]             168
         MaxPool2d-2            [-1, 6, 15, 15]               0
            Conv2d-3            [-1, 8, 13, 13]             440
            Conv2d-4           [-1, 10, 13, 13]              90
            Conv2d-5           [-1, 12, 13, 13]             132
            Conv2d-6           [-1, 14, 13, 13]             182
       BatchNorm2d-7           [-1, 14, 13, 13]              28
            Conv2d-8           [-1, 16, 13, 13]             240
            Linear-9                 [-1, 1024]       2,769,920
           Linear-10                  [-1, 512]         524,800
      BatchNorm1d-11                  [-1, 512]           1,024
           Linear-12                  [-1, 256]         131,328
           Linear-13                   [-1, 10]           2,570
================================================================
```

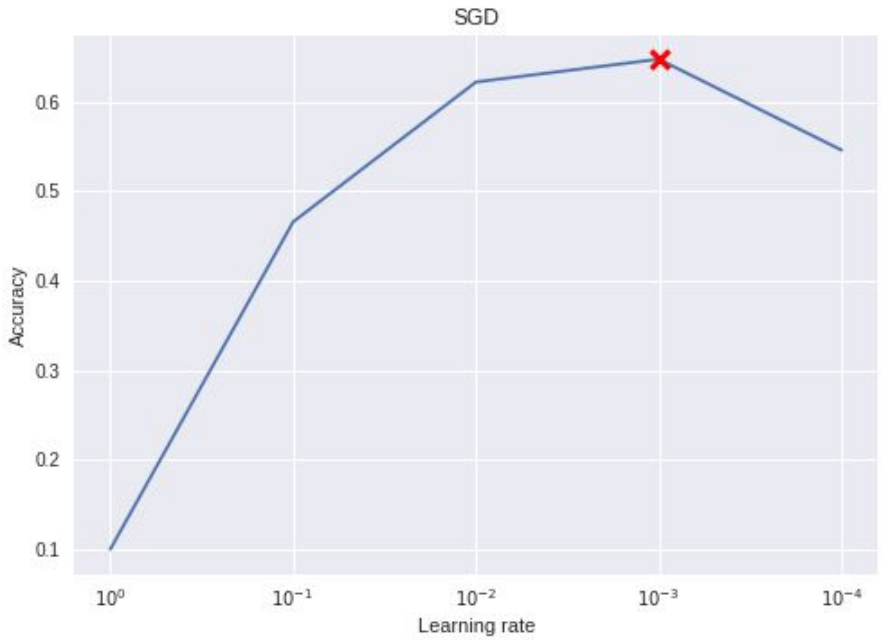
Details of each layer are as follows:

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(8, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(10, 12, kernel_size=(1, 1), stride=(1, 1))
  (conv5): Conv2d(12, 14, kernel_size=(1, 1), stride=(1, 1))
  (BatchNorm): BatchNorm2d(14, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv6): Conv2d(14, 16, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=2704, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=512, bias=True)
  (BatchNorm1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc3): Linear(in_features=512, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=10, bias=True)
)
```

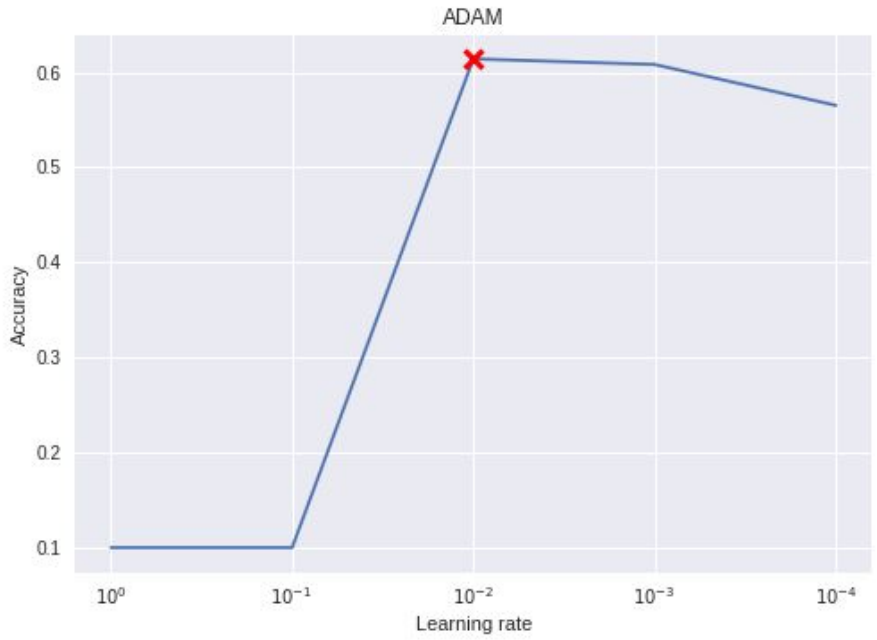
Model is trained for 10 epochs using SGD, ADAM and MADAM optimizer. Performance is observed by varying learning rate as [1, 0.1, 0.01, 0.001, 0.0001].

Following curve is obtained between learning rate and accuracy achieved:

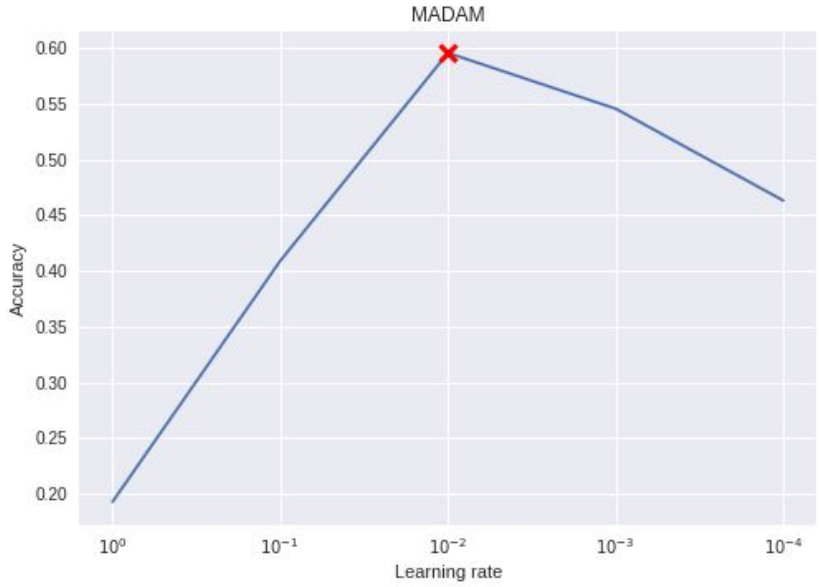
Using SGD optimizer:



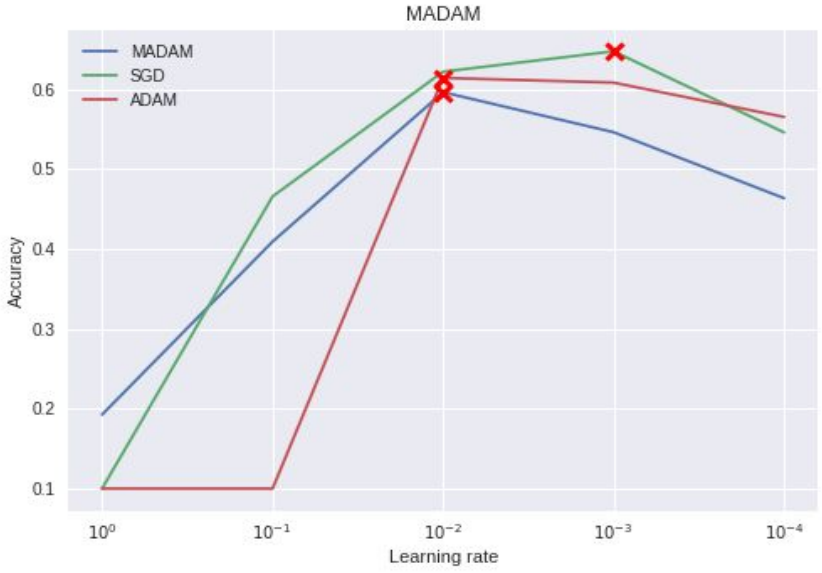
Using ADAM optimizer:



Using Madam Optimizer:



Combined plot of all the curve presented above:



3. Results on one database which was not used in the paper as per the database protocol.

Dataset

Fashion MNIST dataset is used. It consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.Each training and test example is assigned to one of the following labels:

```
('T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat','Sandal',
'Shirt', 'Sneaker', 'Bag', 'Ankle boot')
```

Visualization of 10 random sample is as follow:

Model Information:

CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers.
Arrangement of these layers are as follow:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 6, 30, 30]	168
MaxPool2d-2	[-1, 6, 15, 15]	0
Conv2d-3	[-1, 8, 13, 13]	440
Conv2d-4	[-1, 10, 13, 13]	90
Conv2d-5	[-1, 12, 13, 13]	132
Conv2d-6	[-1, 14, 13, 13]	182
BatchNorm2d-7	[-1, 14, 13, 13]	28
Conv2d-8	[-1, 16, 13, 13]	240
Linear-9	[-1, 1024]	2,769,920
Linear-10	[-1, 512]	524,800
BatchNorm1d-11	[-1, 512]	1,024
Linear-12	[-1, 256]	131,328
Linear-13	[-1, 10]	2,570

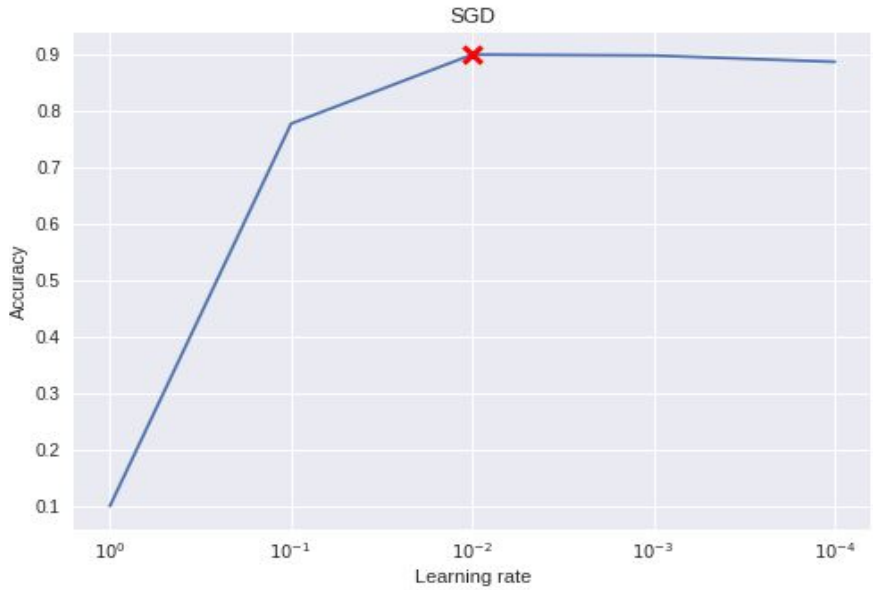
Details of each layer are as follows:

```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(8, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(10, 12, kernel_size=(1, 1), stride=(1, 1))
  (conv5): Conv2d(12, 14, kernel_size=(1, 1), stride=(1, 1))
  (BatchNorm): BatchNorm2d(14, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv6): Conv2d(14, 16, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=2704, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=512, bias=True)
  (BatchNorm1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc3): Linear(in_features=512, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=10, bias=True)
)
```

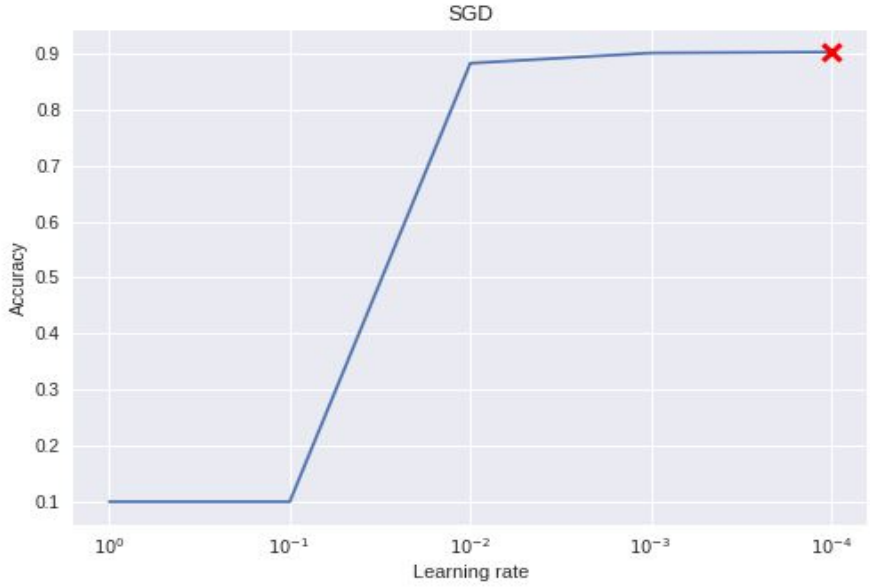
Model is trained for 10 epochs using SGD, ADAM and MADAM optimizer.
Performance is observed by varying learning rate as [1, 0.1, 0.01, 0.001, 0.0001].

Following curve is obtained between learning rate and accuracy achieved:

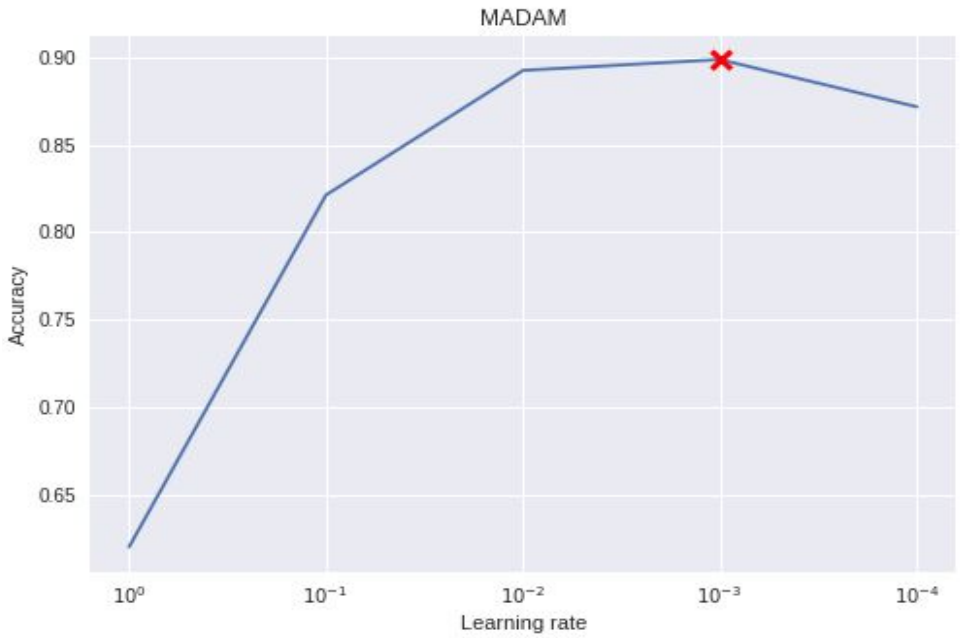
Using SGD optimizer:



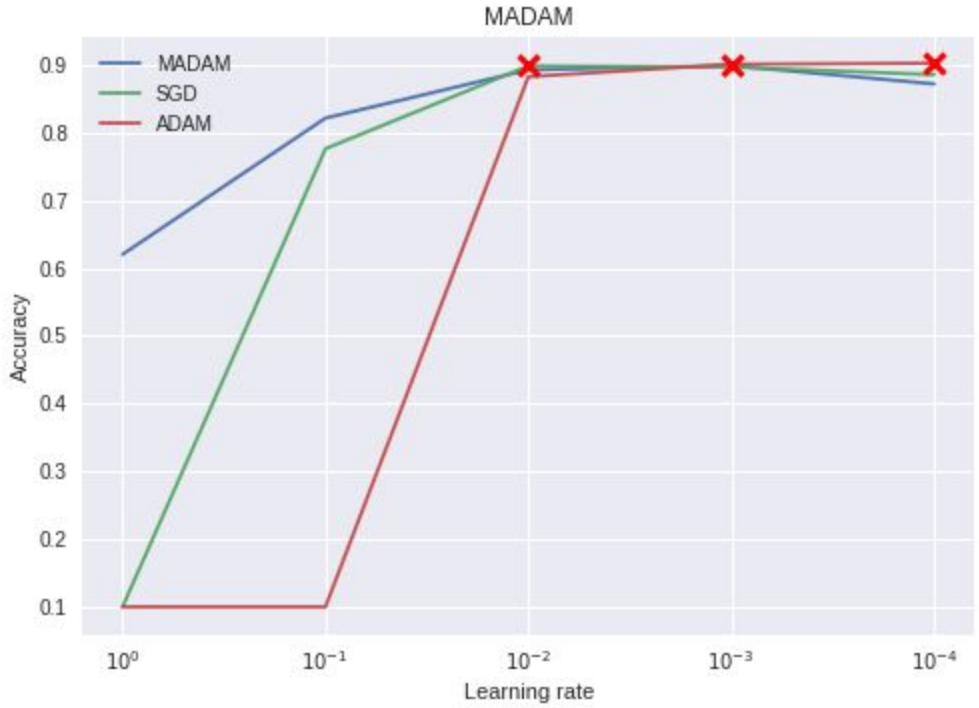
Using ADAM Optimizer:



Using MADAM optimizer:

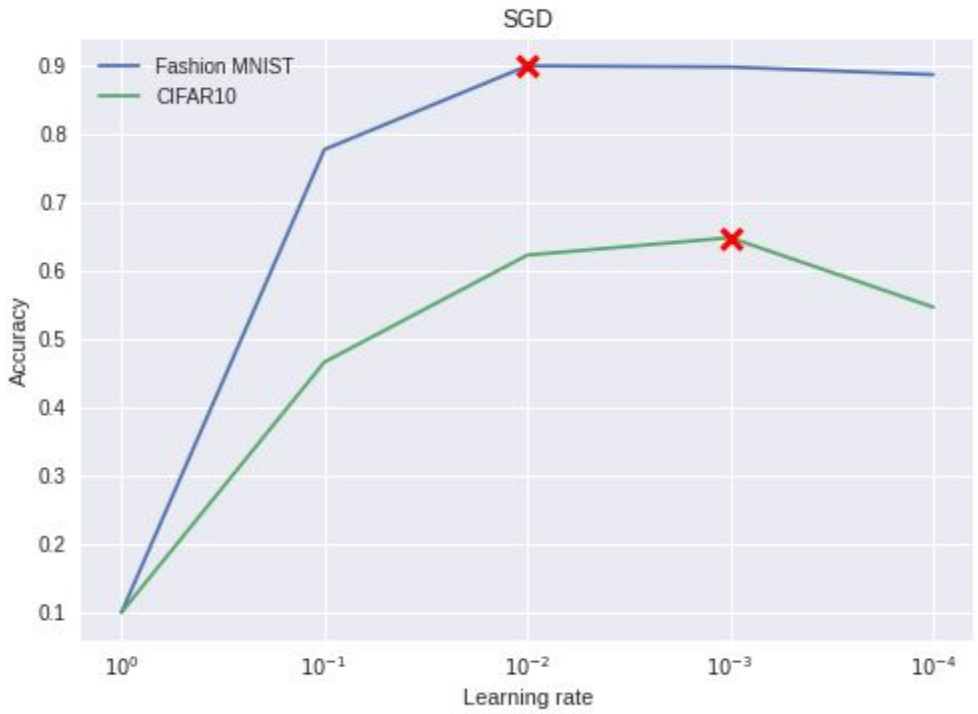


Combined plot of all the curve presented above:

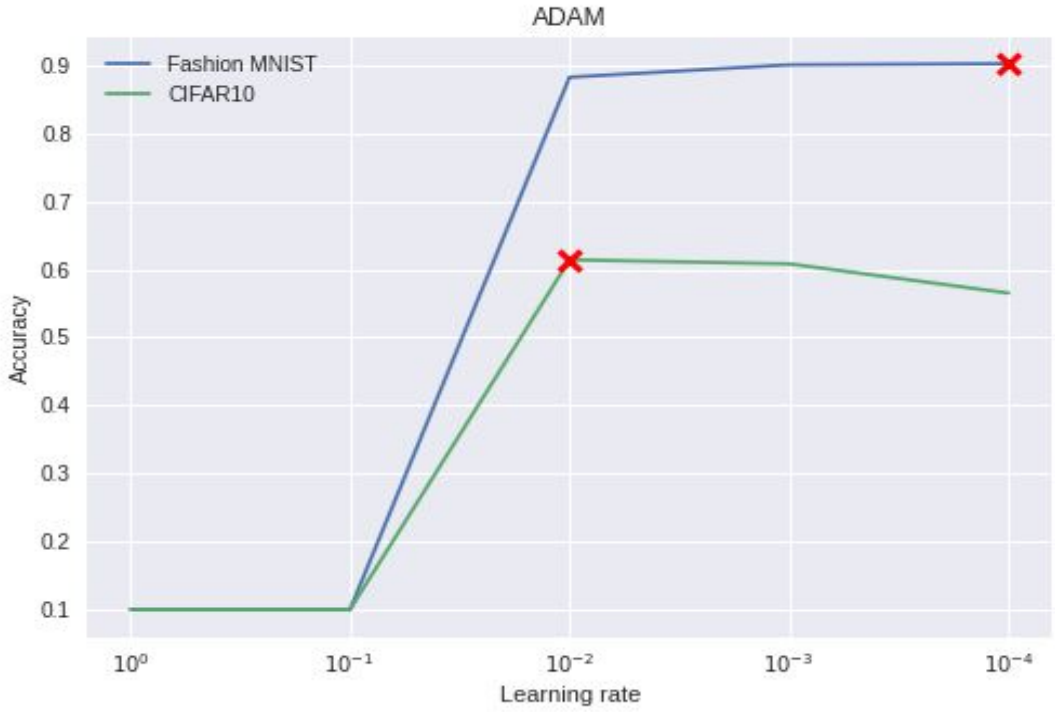


Comparison of result of Accuracy of above two dataset i.e CIFAR-10 and Fashion MNIST over different optimizer

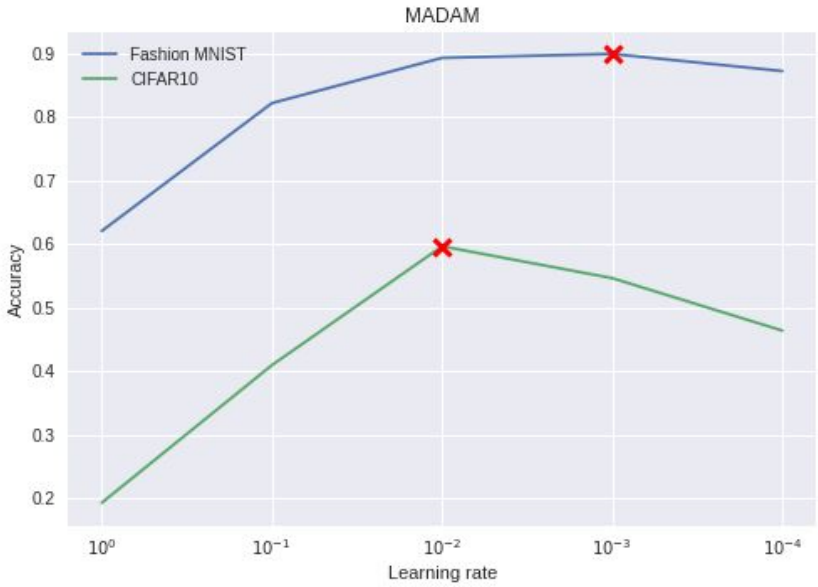
Over SGD Optimizer:



Over ADAM Optimizer:



Over MADAM Optimizer:



Tabulation of Accuracy Obtained:

DataSet	SGD		ADAM		MADAM	
CIFAR-10	lr =1	10	lr =1	10	lr =1	19.25
	lr =0.1	21.9	lr =0.1	10	lr =0.1	40.91
	lr =0.01	50.83	lr =0.01	48.6	lr =0.01	59.61
	lr =0.001	48.65	lr =0.001	50.33	lr =0.001	54.6
	lr=0.0001	33.3	lr=0.0001	48.33	lr=0.0001	46.34

Fashion MNIST	lr =1 | 10 lr =0.1 | 63.98 lr =0.01 | 86.06 lr =0.001 | 86.5 lr=0.0001 | 80.58	lr =1 | 10 lr =0.1 | 10 lr =0.01 | 84.13 lr =0.001 | 86.01 lr=0.0001 | 86.25	lr =1 | 36.67 lr =0.1 | 69.11 lr =0.01 | 86.74 lr =0.001 | 85.74 lr=0.0001 | 74.2