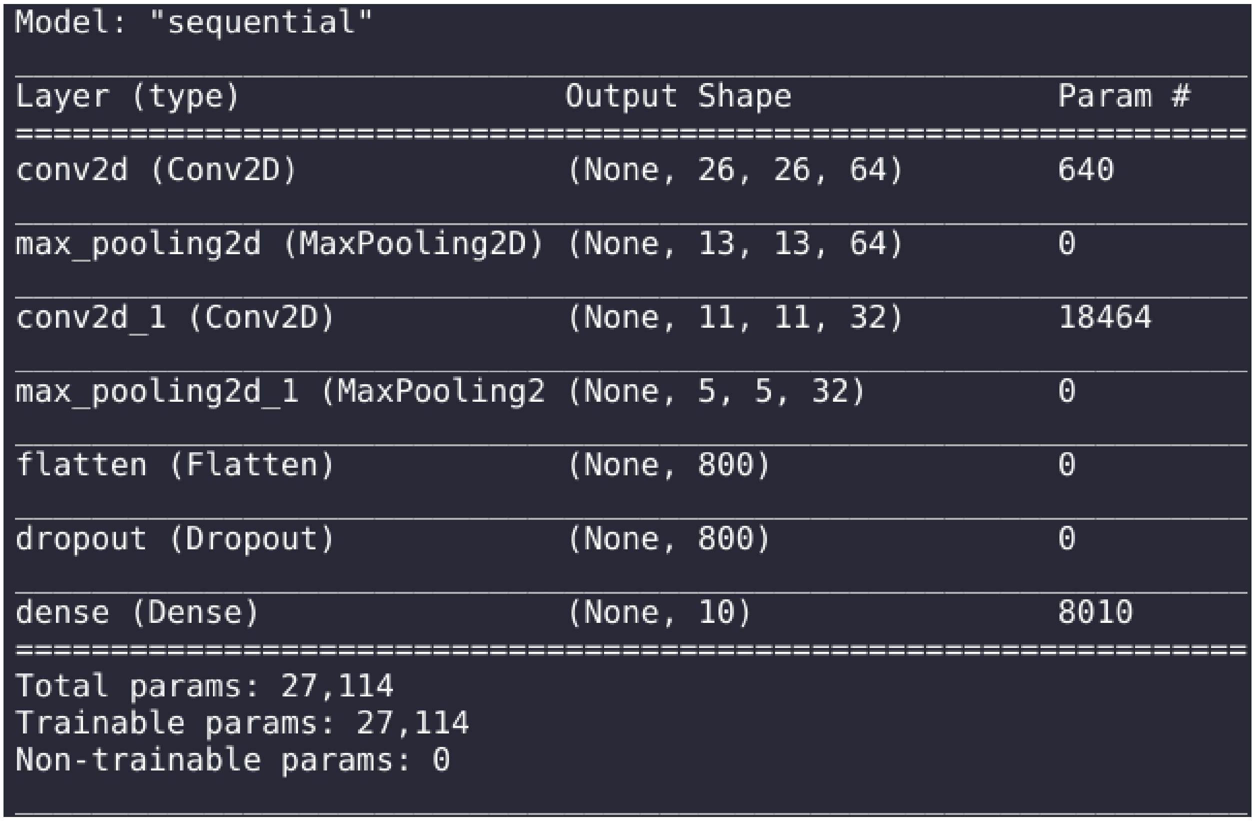
CSE517 Optimization for Machine Learning Project
Hardik Garg, 2019040 | Tathagat Pal, 2019211

Problem Statement - To analyze the convergence of different optimizers for the supervised learning task performed on the Fashion-MNIST dataset.

Model Summary - The CNN created using Keras is as shown below -



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 26, 26, 64)        640       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 64)        0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 32)        18464     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 32)          0         
_________________________________________________________________
flatten (Flatten)            (None, 800)               0         
_________________________________________________________________
dropout (Dropout)            (None, 800)               0         
_________________________________________________________________
dense (Dense)                (None, 10)                8010      
=================================================================
Total params: 27,114
Trainable params: 27,114
Non-trainable params: 0
_________________________________________________________________
```

For all the optimizers, the batch size was 128 and the number of epochs was 100

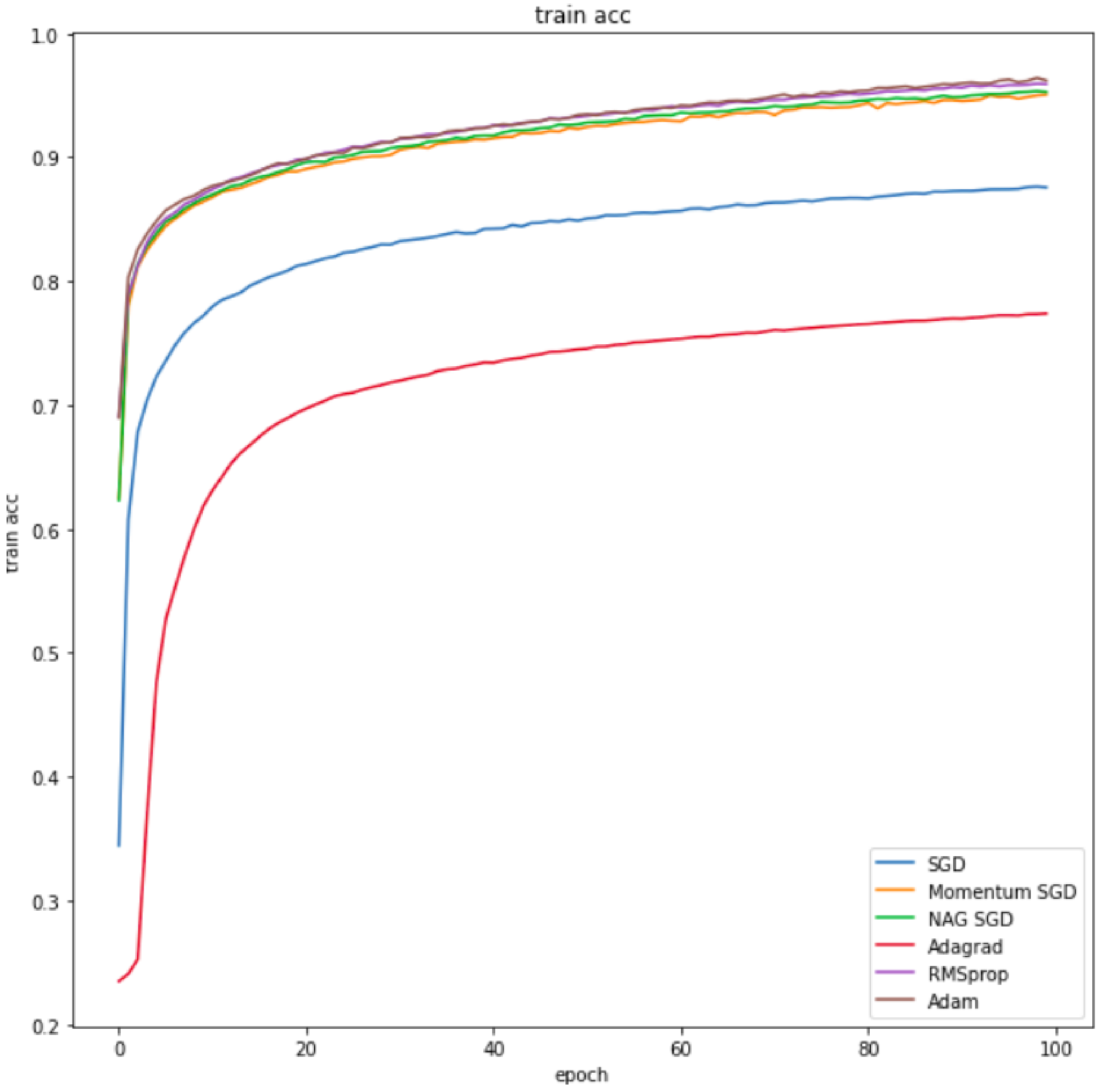
Hyperparameter Tuning - In order to do the hyperparameter tuning, keras_tuner was used and the results obtained are as follows -

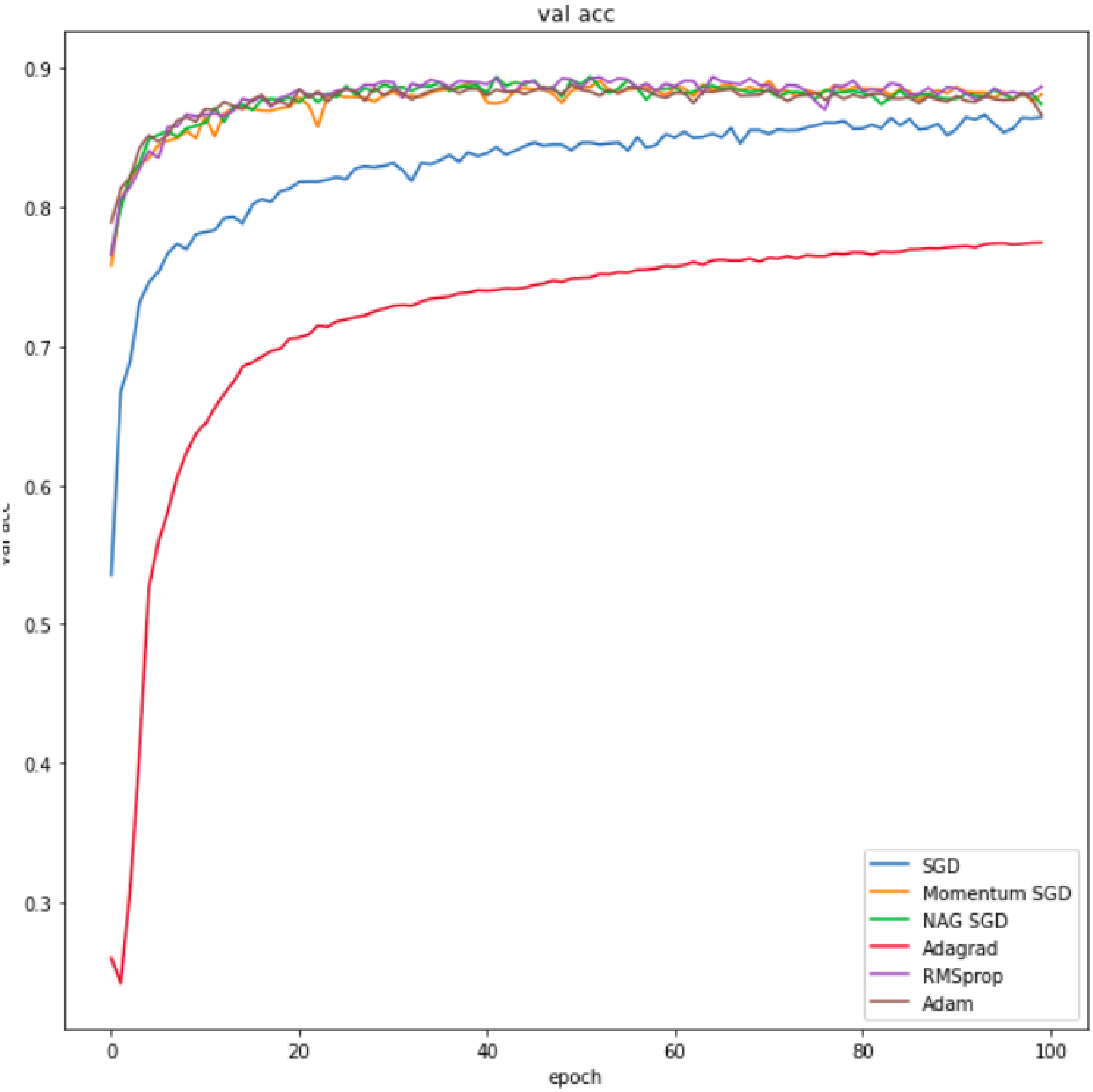
1. **SGD -** learning_rate = 0.01
2. **Momentum SGD -** learning_rate=0.01, momentum=0.9
3. **NAG SGD -** learning_rate=0.01, momentum=0.9, nesterov=True
4. **Adagrad -** learning_rate=0.001, epsilon=1e-7, initial_accumulator_value=0.1
5. **RMSprop -** learning_rate=0.001, epsilon=1e-7, momentum=0.75, rho=0.5, centered=True
6. **Adam -** learning_rate=0.001, beta_1=0.3, beta_2=0.999

The model was run with the default values of these optimizers as well as the aforementioned values. The metric used for evaluation is Accuracy. The loss plots and the accuracy plots obtained are shown below -

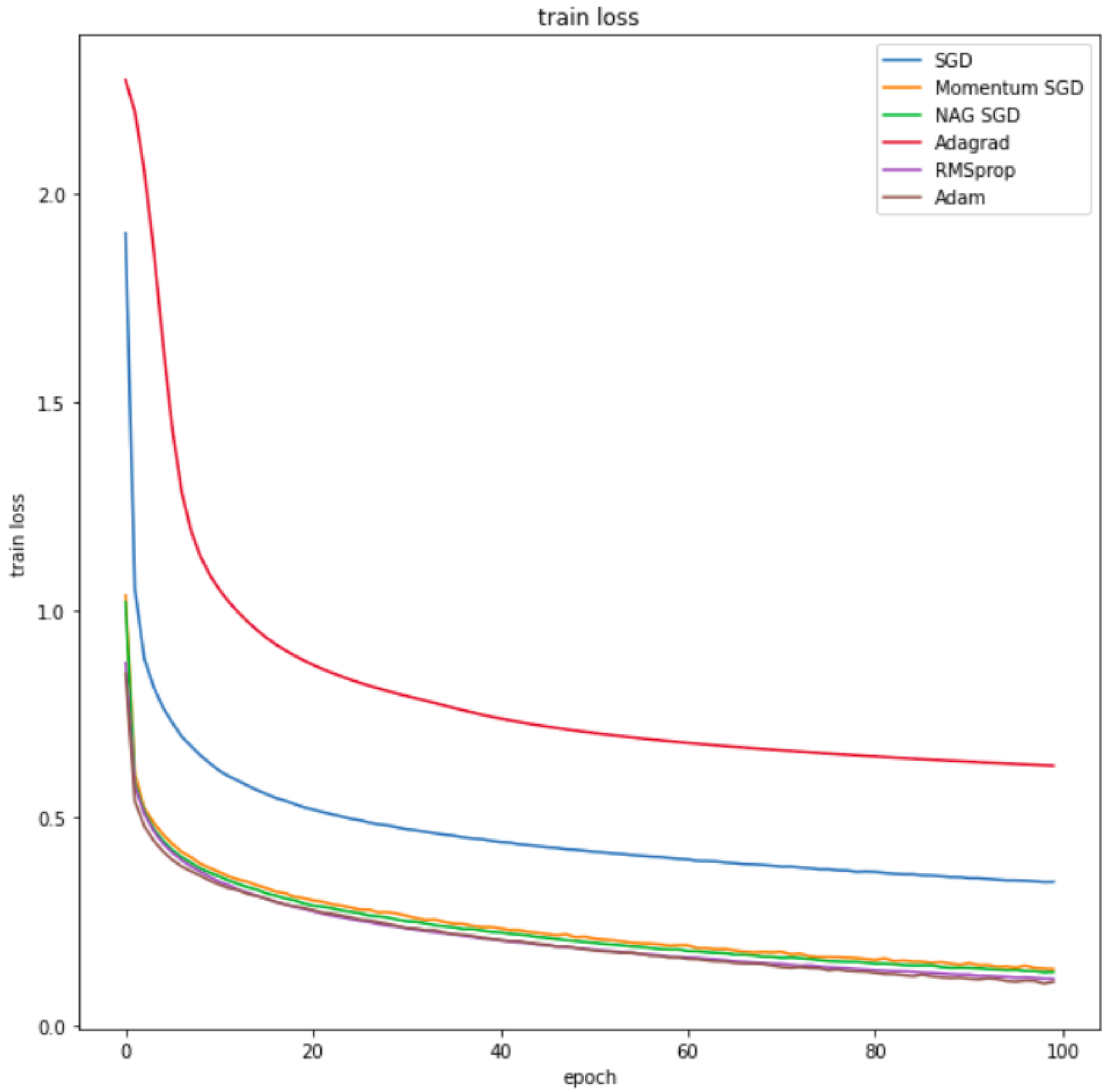
1. **For default optimizer values**

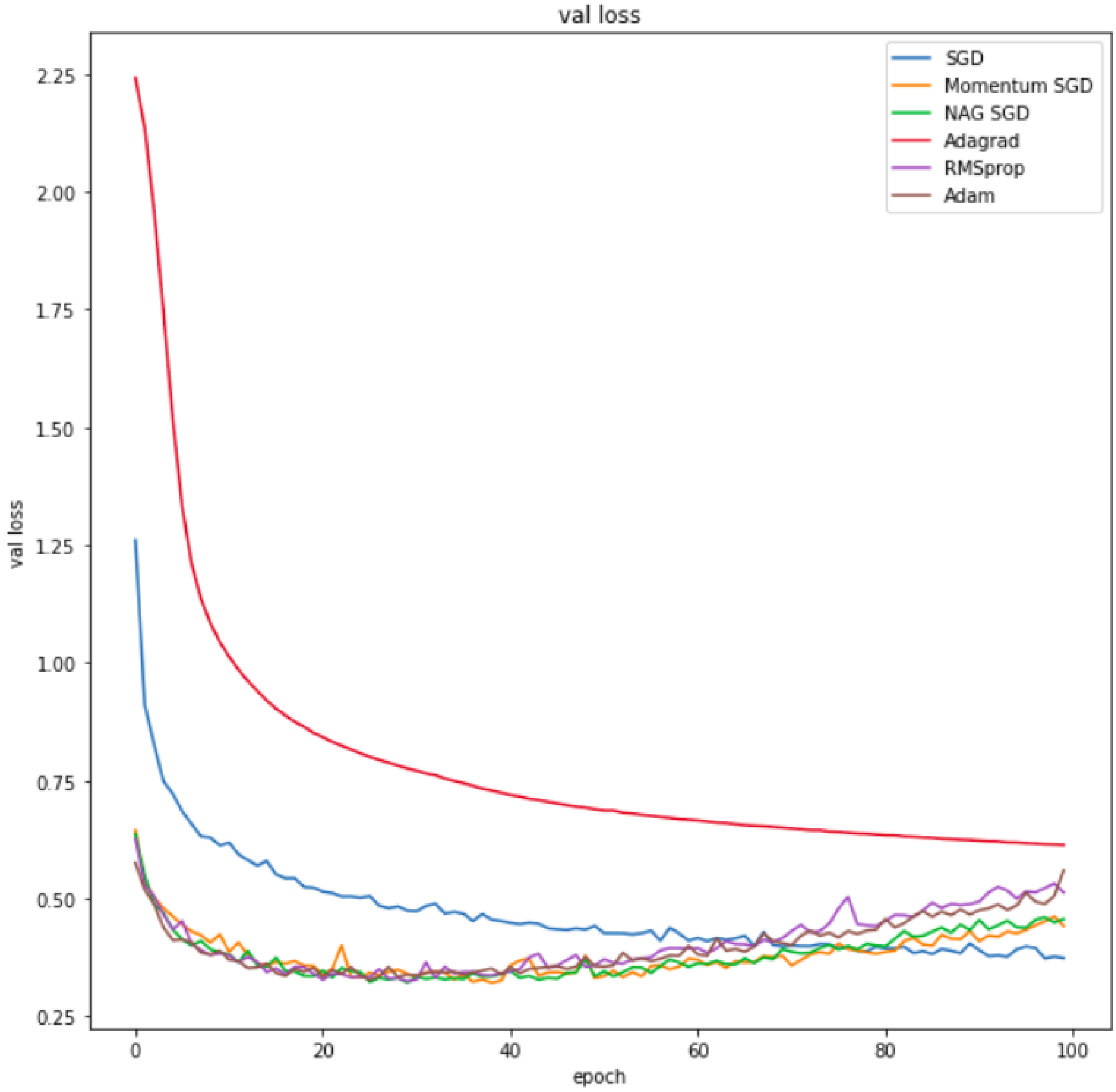
   (a) Accuracy Plots




val acc
val acc
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
epoch
SGD
Momentum SGD
NAG SGD
Adagrad
RMSprop
Adam

(b) Loss Plots

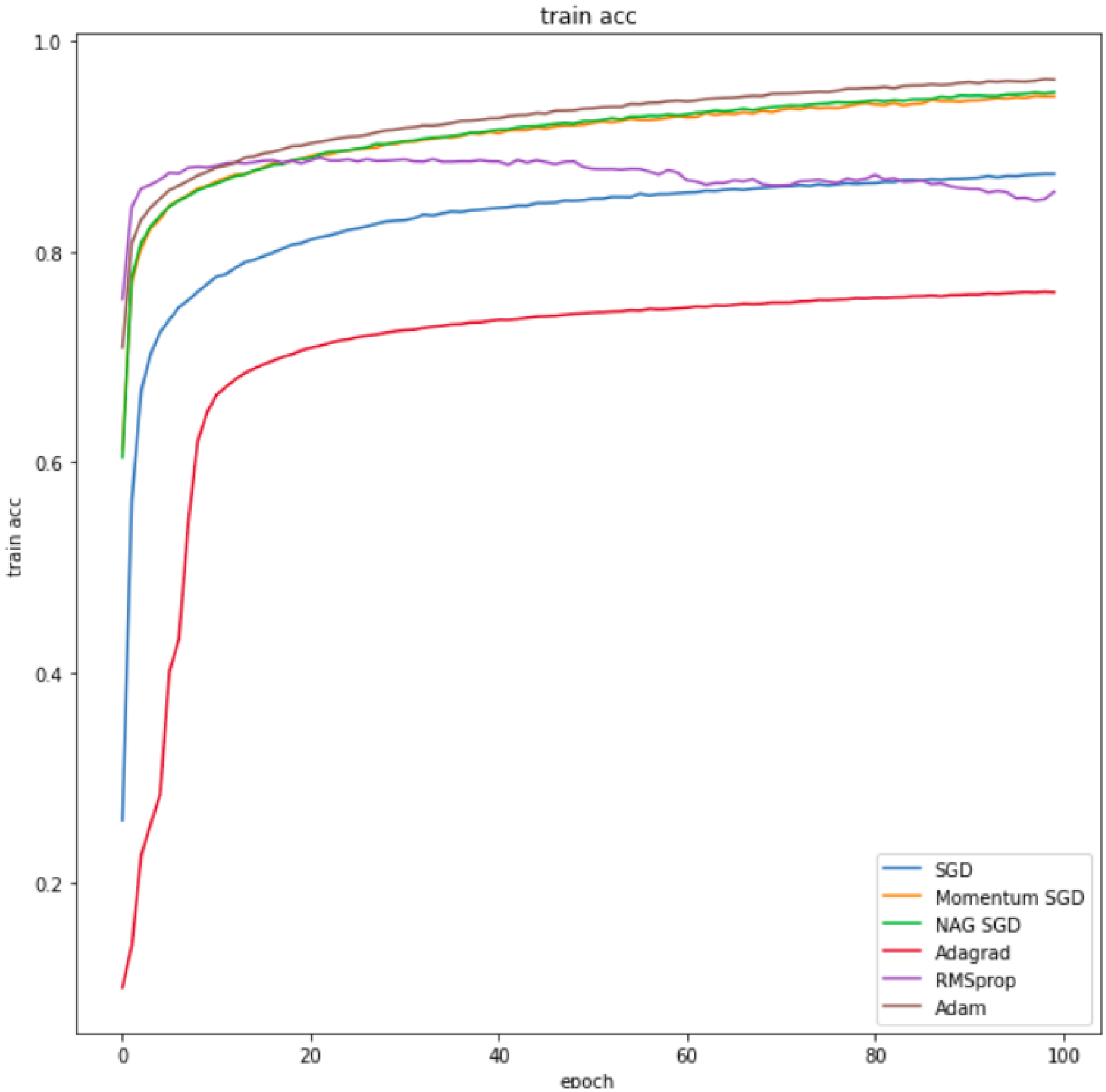

train loss
SGD
Momentum SGD
NAG SGD
Adagrad
RMSprop
Adam
train loss
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
epoch

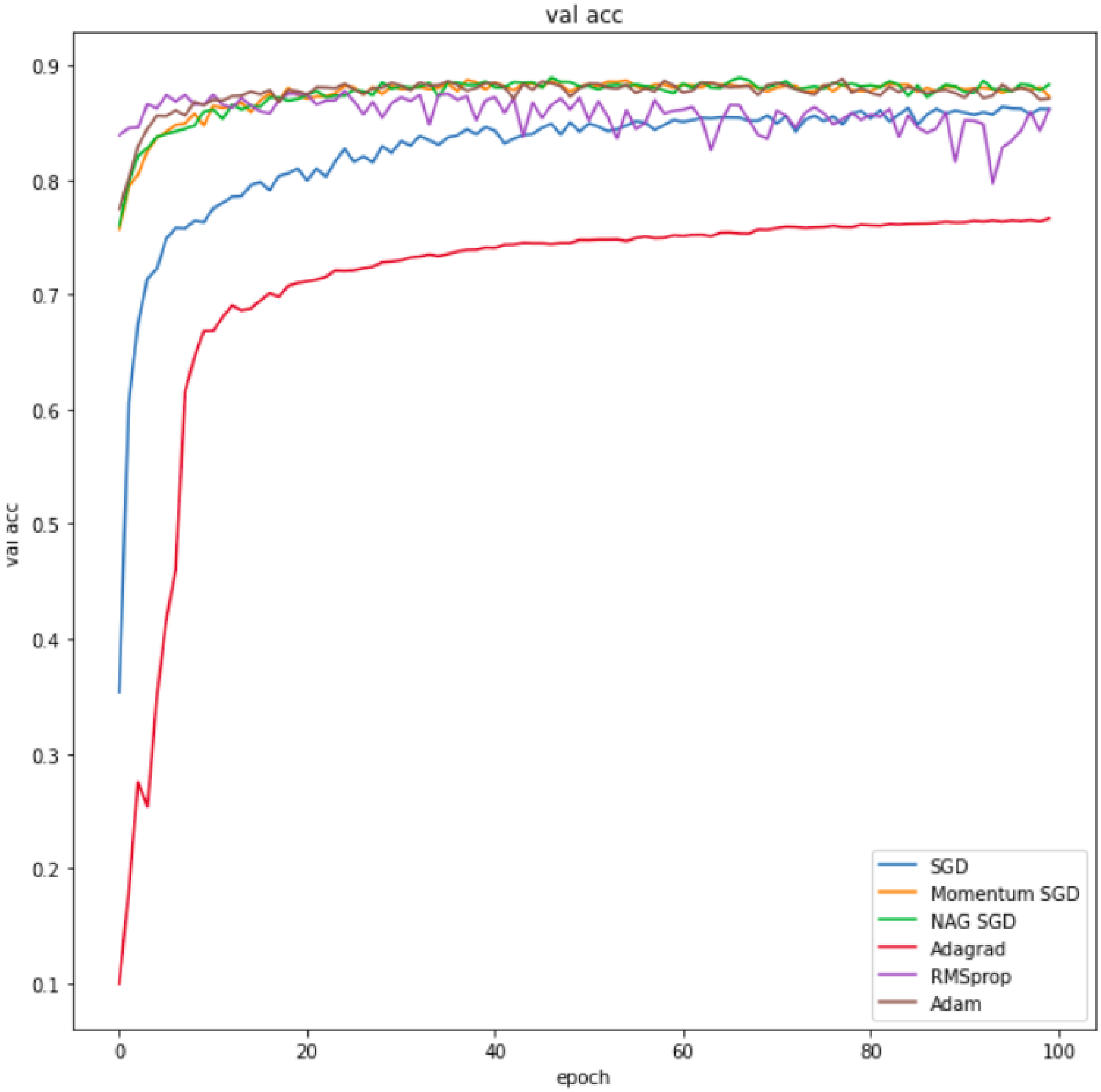


For default parameters, for accuracy plots, Momentum SGD, NAG SGD, RMSprop, Adam all show the best convergence whereas for validation plots, along with the former optimizers, SGD shows improvement. As for the loss plots, for the train set, Momentum SGD, NAG SGD, RMSprop, Adam all show the best convergence whereas for the validation set, SGD also joins the group. Note that though SGD shows the best convergence in loss plots, its accuracy is not the highest.

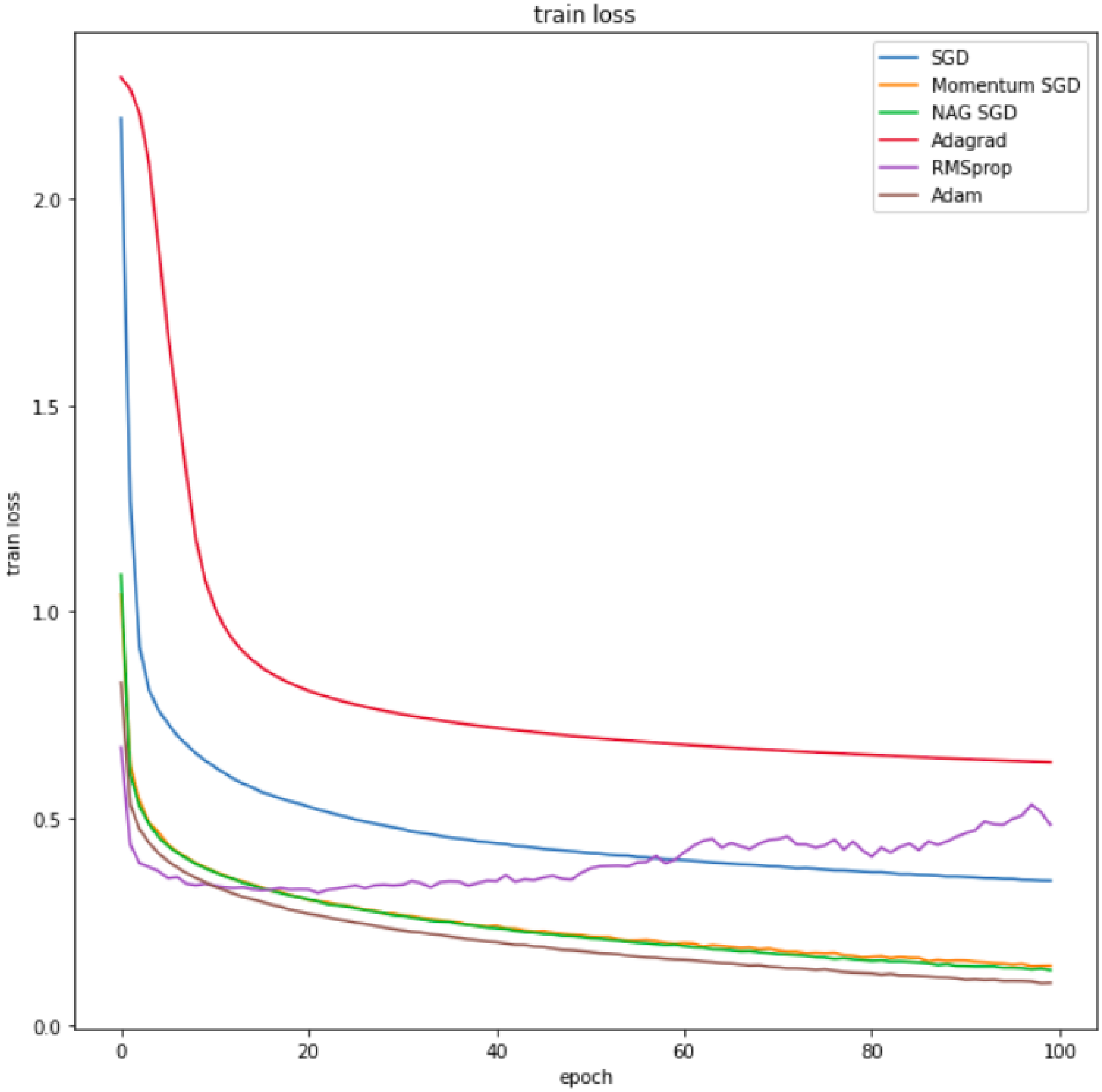
2. For Calculated Optimizer values

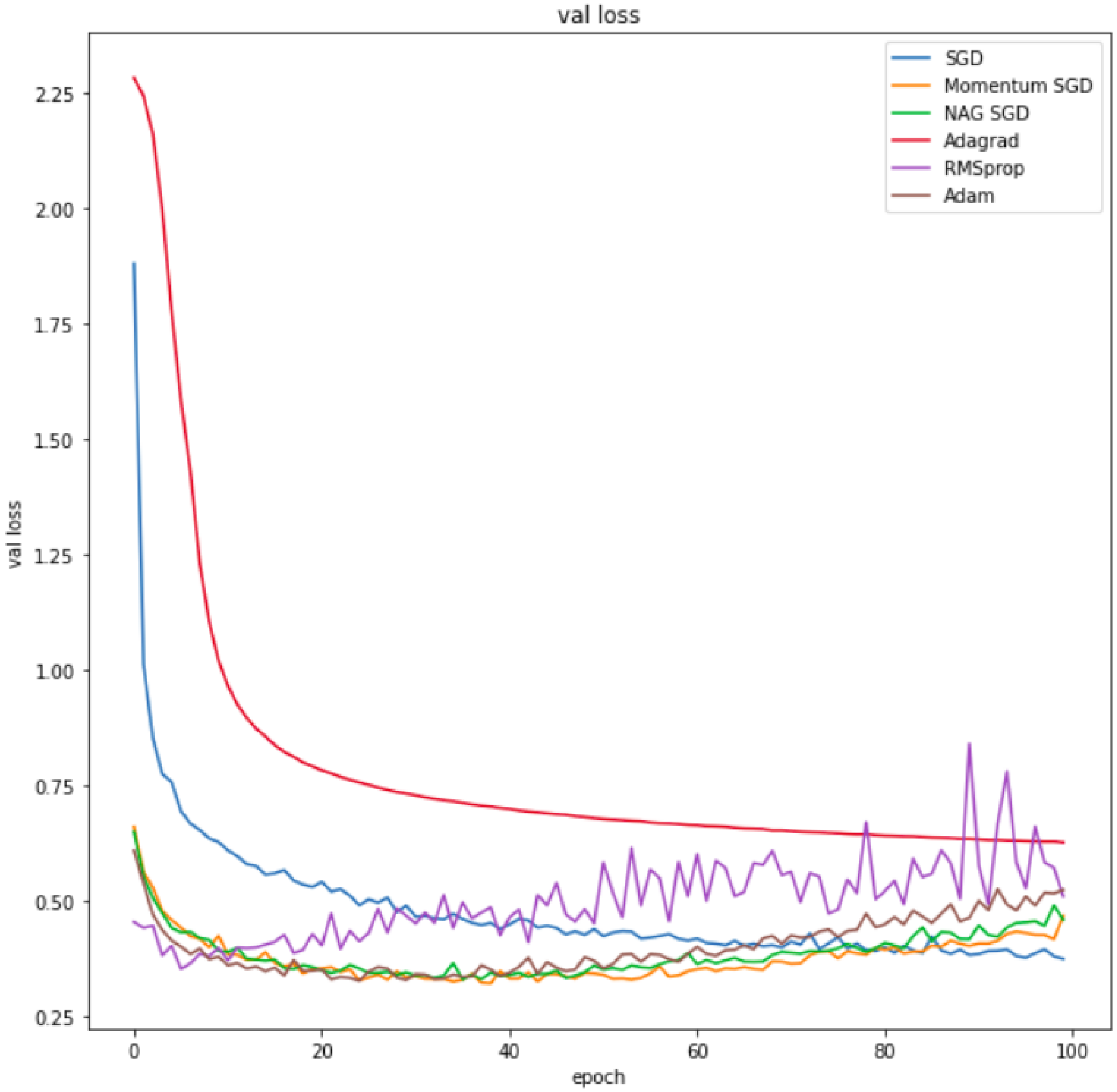
(a) Accuracy Plots




val acc
val acc
epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
SGD
Momentum SGD
NAG SGD
Adagrad
RMSprop
Adam

(b) Loss Plots


train loss
SGD
Momentum SGD
NAG SGD
Adagrad
RMSprop
Adam
train loss
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
epoch



After hyperparameter tuning, for accuracy plots, on the training set, Adam, NAG SGD and Momentum SGD have the best convergence whereas on the validation set, Adam, NAG SGD and Momentum SGD once again have the best convergence. As for the loss plots, on the training set, Adam, Momentum SGD and NAG SGD have the best convergence. On the validation set, SGD shows the best convergence, followed by NAG SGD and Momentum SGD. Point to note here is that despite the best convergence in loss plots, SGD does not perform analogously better in terms of accuracy.

Observations

1. After hyperparameter tuning, SGD outperformed RMSprop.
2. After hyperparameter tuning, all optimizers give slightly better results, however the overall comparison remains the same, with the exception of SGD.
3. RMSprop showed a slight decrease in the accuracy / increase in error towards the last few iterations.
4. Even a simple CNN gives a high accuracy (~0.9 for train set and ~0.8 for validation set)

Conclusion

1. With default hyperparameter values, order of convergence of optimizers is - Adagrad < SGD < Momentum SGD < NAG SGD < RMSprop < Adam
2. After hyperparameter tuning, the order of convergence of optimizers is - Adagrad < RMSprop < SGD < Momentum SGD < NAG SGD < Adam