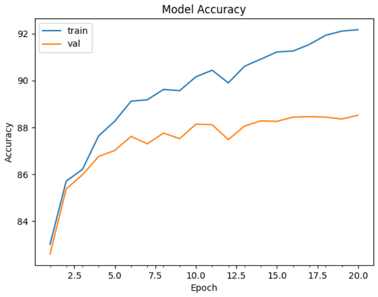
CS7015-Deep Learning
Programming Assignment#1:Report

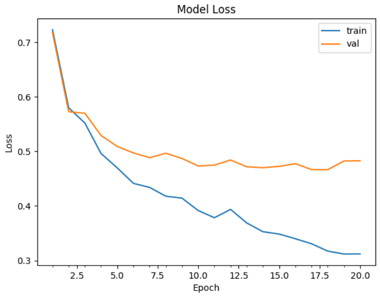
Team name: MagicalAI
Ajmeera Balaji Naik(CS14B034)
Satish G(CS14B042)

February 17, 2018

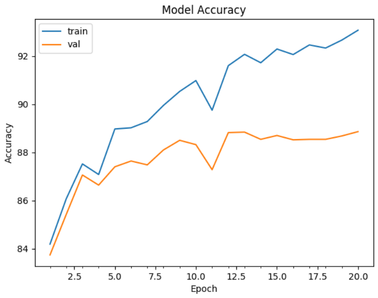
1. One Hidden Layer

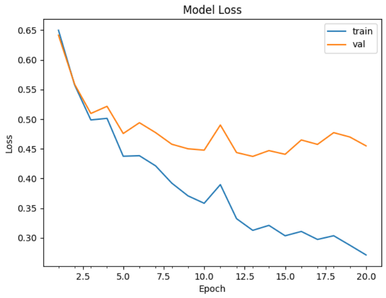
- Configuration: 50 Neurons



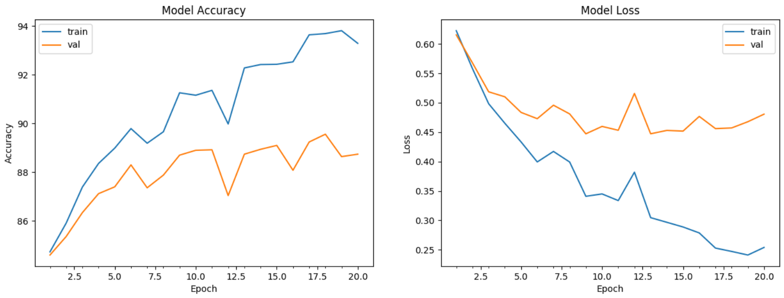


- Configuration: 100 Neurons

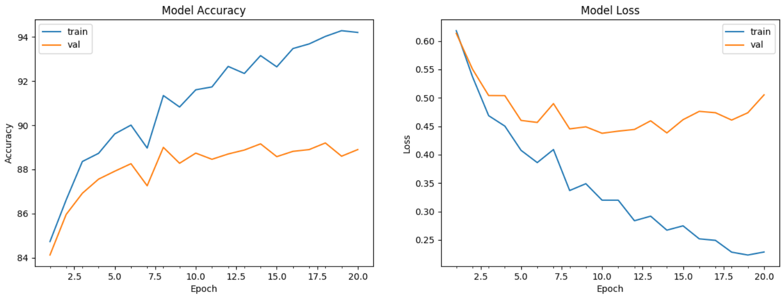


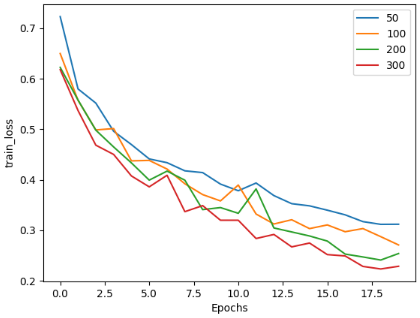


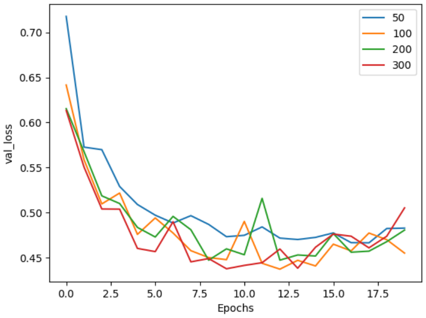
- Configuration: 200 Neurons



- Configuration: 300 Neurons







no.of.neurons	best train_acc @epoch	best val_acc @epoch
50	92.17@20	88.52 @20
100	93.07@20	88.86 @20
200	**93.81**@19	**89.56** @18
300	94.29@19	89.2 @18

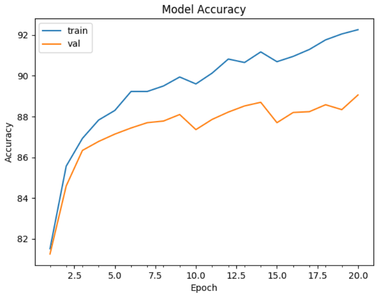
The best model for one layer is for **200** neurons

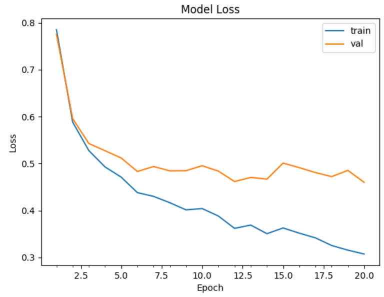
Observations and Comments:

- As number of neurons increases, the training accuracy increases as expected since the complexity of model increases which implies that over fitting tendency increases.
- One can see that validation accuracy increases till 200 neurons and thereafter it deceases(The model complexity at 200 neurons might be corresponds to sweet spot in **Bias-Variance** graph)

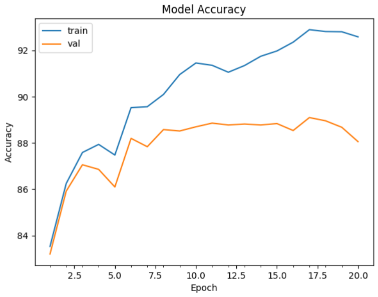
2. Two Hidden Layers

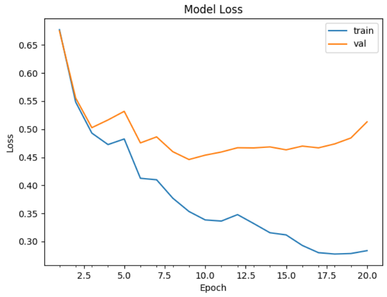
- Configuration: 50 Neurons



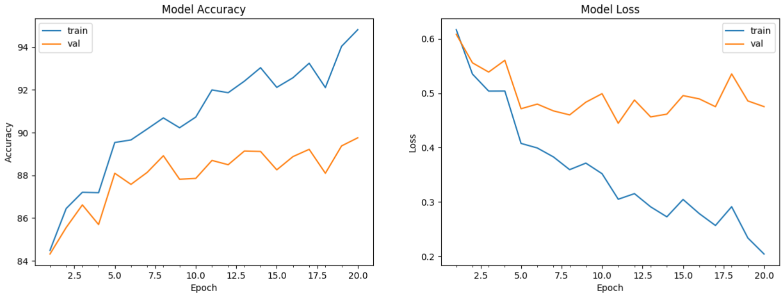


- Configuration: 100 Neurons

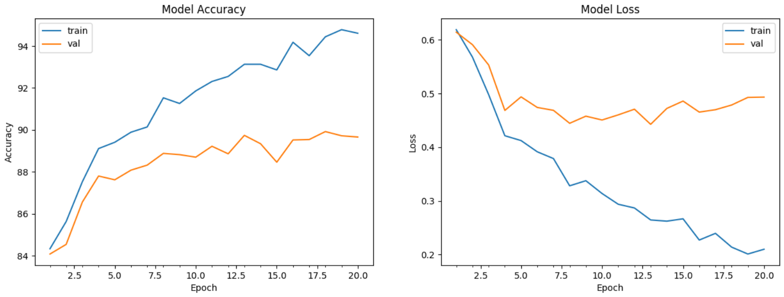


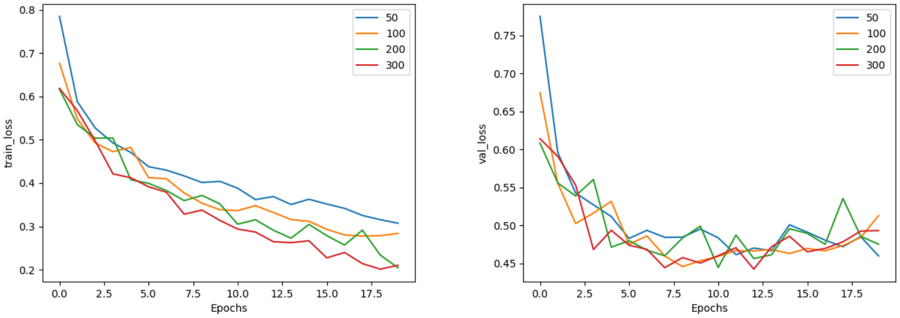


- Configuration: 200 Neurons



- Configuration: 300 Neurons



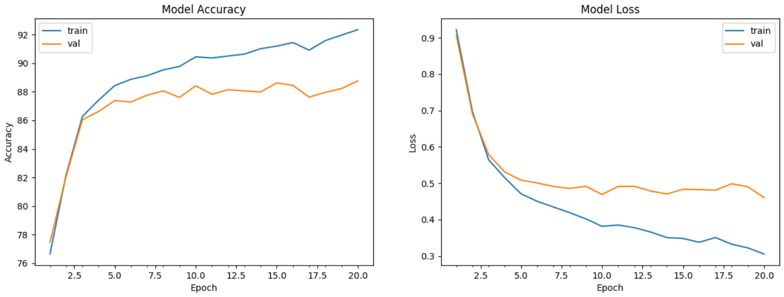


no.of.neurons	best train_acc @epoch	best val_acc @epoch
50	92.26@20	89.06 @20
100	92.9@17	89.1 @17
200	94.82@20	89.76 @20
300	**94.78**@19	**89.92** @18

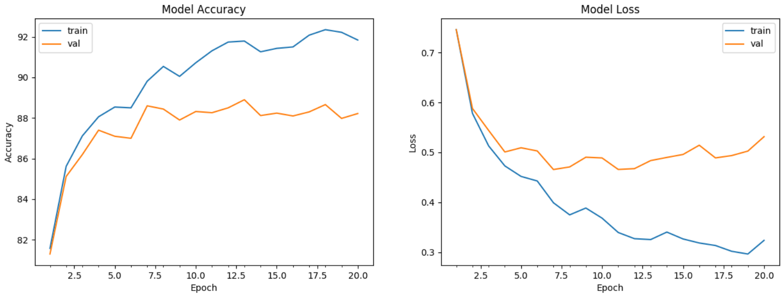
The best model for two layer is for **300** neurons

3. Three Hidden Layers

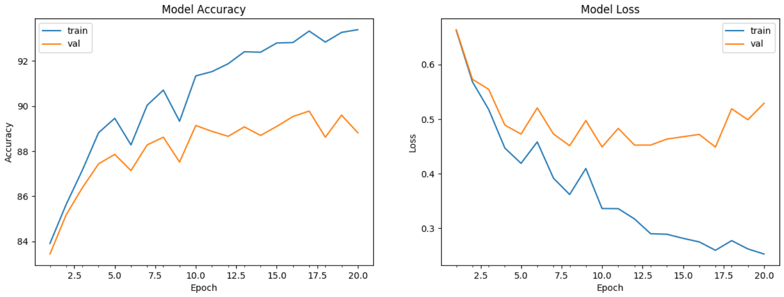
- Configuration: 50 Neurons



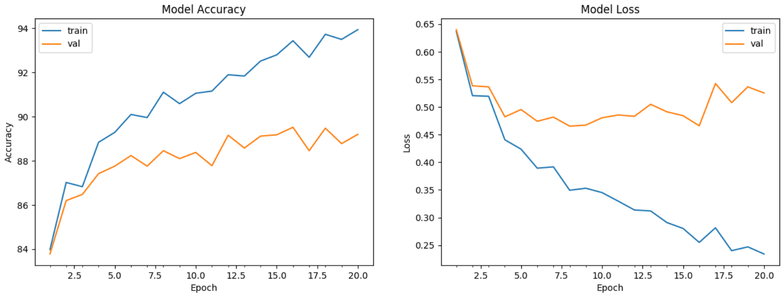
- Configuration: 100 Neurons

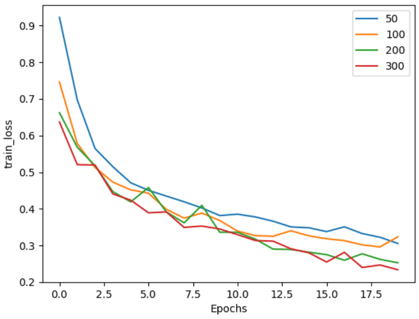


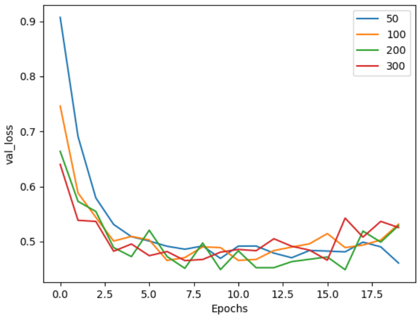
- Configuration: 200 Neurons



- Configuration: 300 Neurons





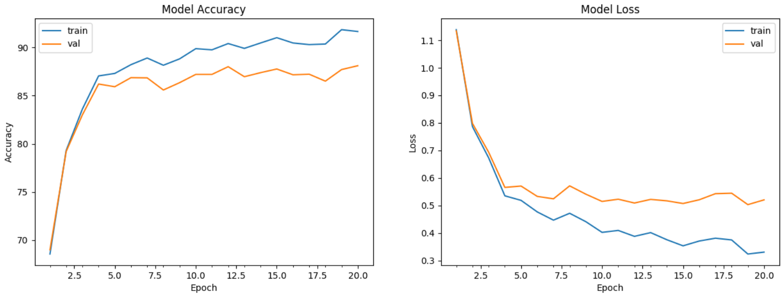


no.of.neurons	best train_acc @epoch	best val_acc @epoch
50	92.34@20	88.76 @20
100	92.35@18	88.9 @13
200	**93.39**@19	**89.78** @17
300	93.94.78@20	89.52 @16

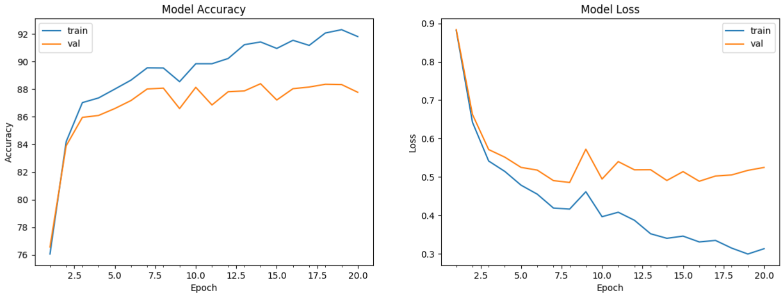
The best model for three layer is for **200** neurons

4. Four Hidden Layers

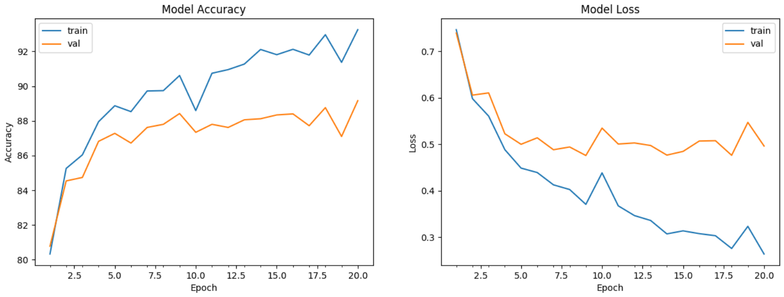
- Configuration: 50 Neurons



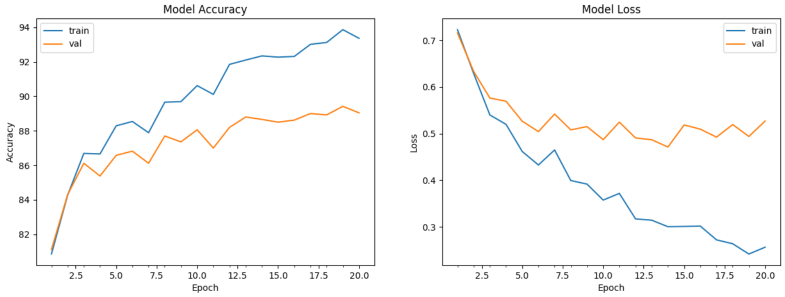
- Configuration: 100 Neurons

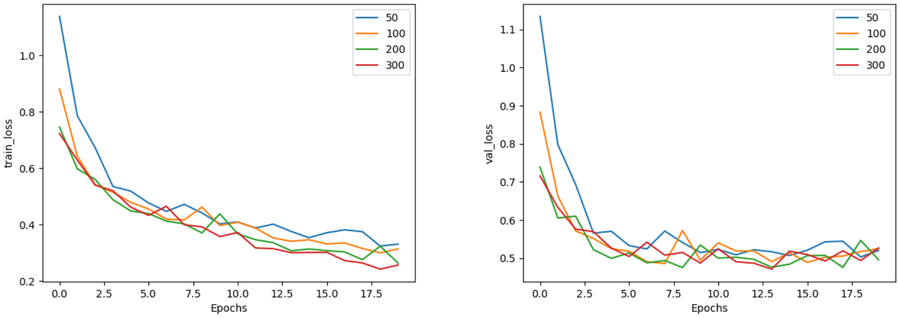


- Configuration: 200 Neurons



- Configuration: 300 Neurons



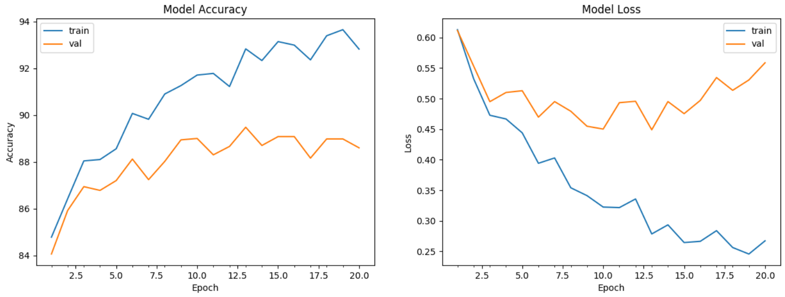


no.of.neurons	best train_acc @epoch	best val_acc @epoch
50	91.84@19	88.1 @20
100	92.32@19	88.4 @14
200	93.25@20	89.16 @20
300	**93.86**@19	**89.42** @19

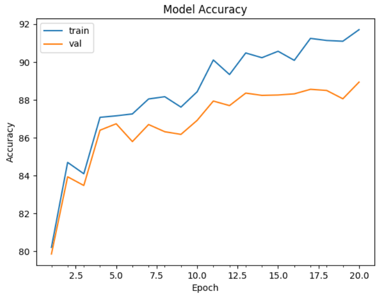
The best model for four layer is for **300** neurons

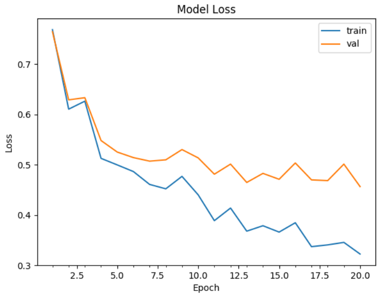
5. Adam, NAG, Momentum,GD

- Algorithm: Adam

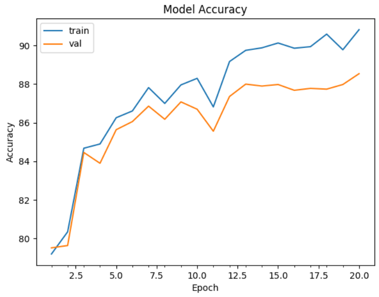


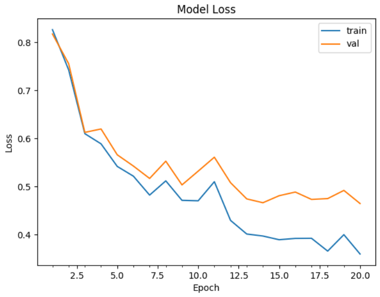
- Algorithm: NAG



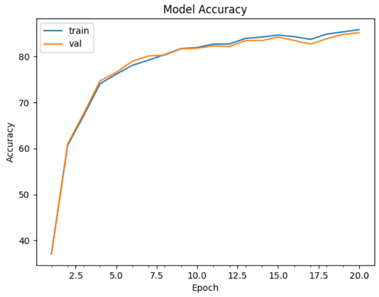


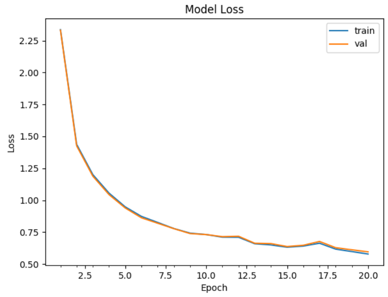
- Algorithm: Momentum

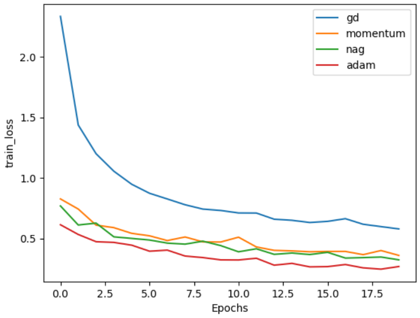


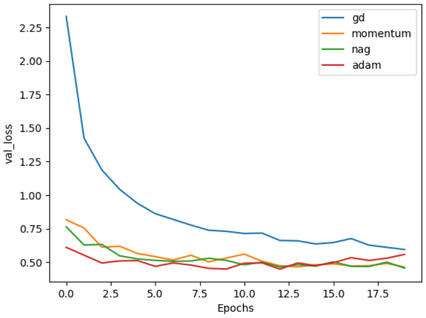


- Algorithm: GD









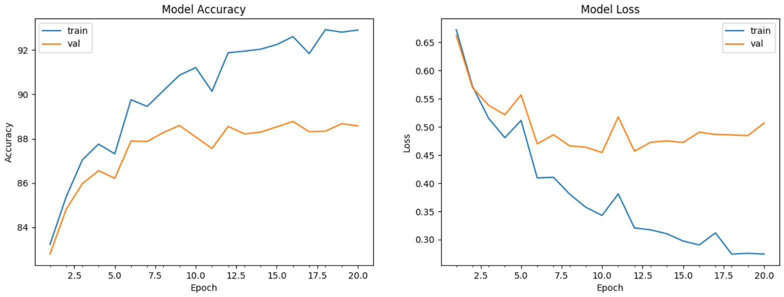
Optimizier	best train_acc @epoch	best val_acc @epoch
gd	85.83 @20	85.16 @20
momentum	90.82 @20	88.54 @20
nag	91.71 @20	89.94 @20
adam	**93.65**@19	**89.48** @12

Observations:

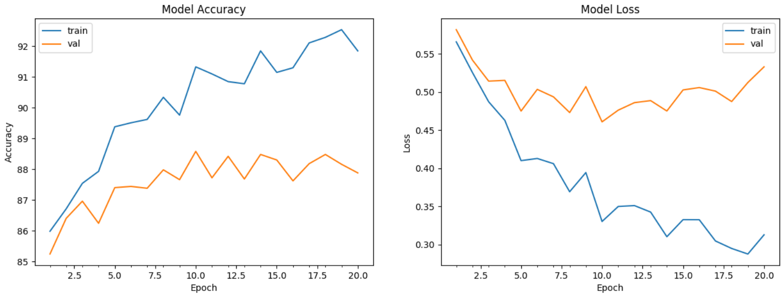
Based on above set of experiments, it is observed that "Adam" did good job when compared to others. In cased of Adam, Momentum and NAG, one can see the oscillations.Whereas in GD, nothing as such which is expected.

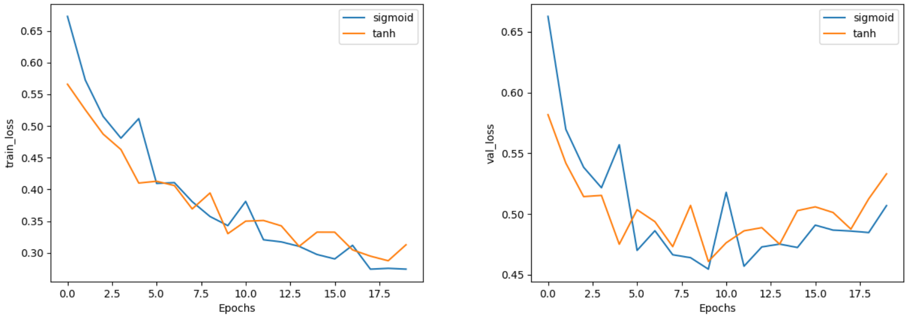
6. Sigmoid vs Tanh Activation Function

- Sigmoid



- Tanh

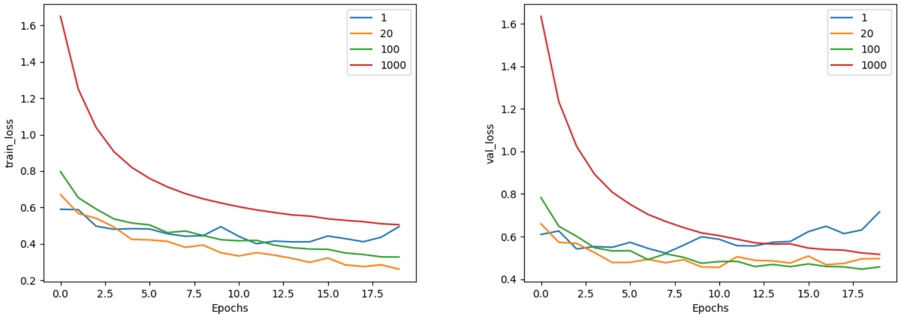




Activation function	best train_acc @epoch	best val_acc @epoch
sigmoid	92.92@18	88.78 @16
tanh	92.54@18	88.58 @9

Observation:
Here, from the accuracy estimates, it is observed that architecture with *Sigmoid* activation function is performing well when compared to *Tanh* function

7. Batch size



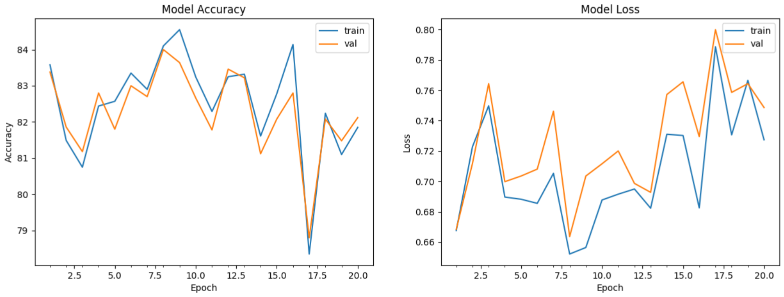
Batch size	best train_acc @epoch	best val_acc @epoch
1	89.98@12	87.66 @8
20	93.11@20	89.08 @17
100	91.71@19	88.7 @19
1000	87.63@20	86.8 @20

Observations:
As the batch size increased, the algorithm is almost working as standard GD which is implied from oscillations in above graph.

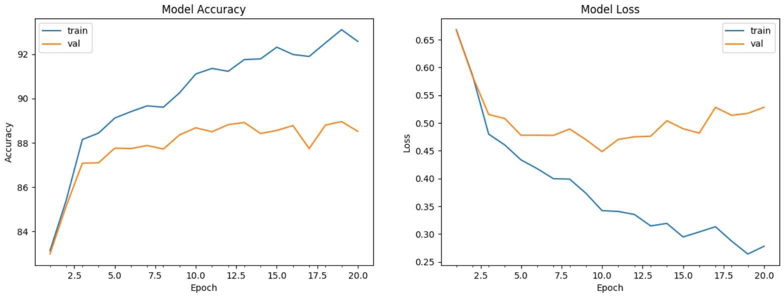
8. Tuning Learning Rate

exponential search

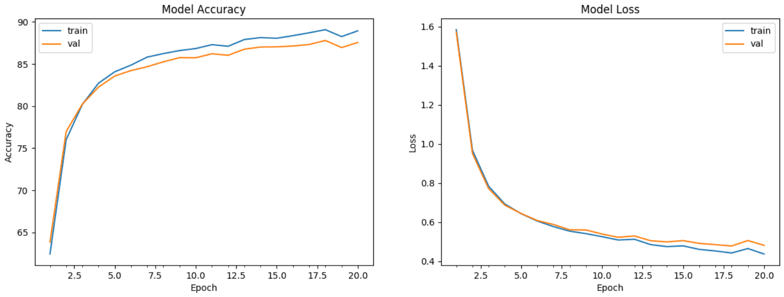
- $\eta = 0.01$



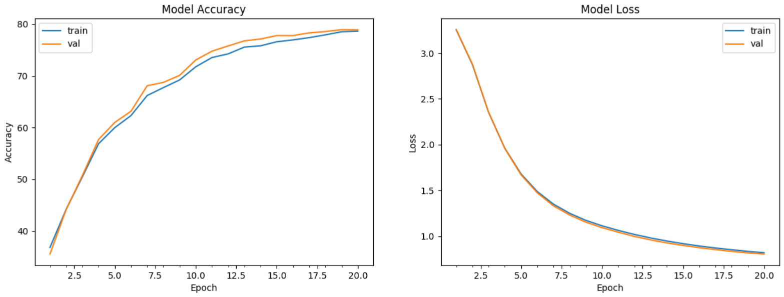
- $\eta = 0.001$



- $\eta = 0.0001$

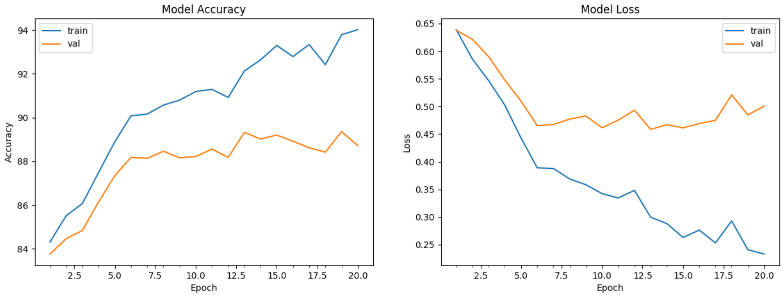


- $\eta = 0.00001$

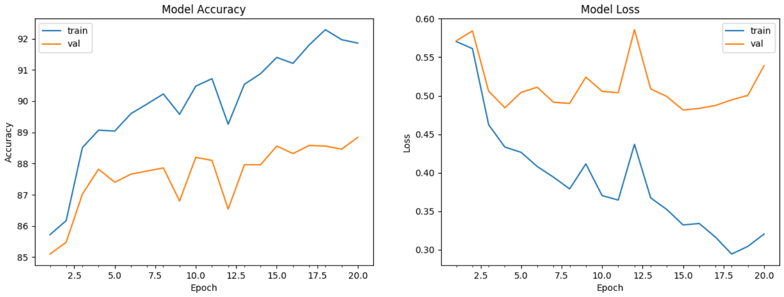


Fine(linear) Search:

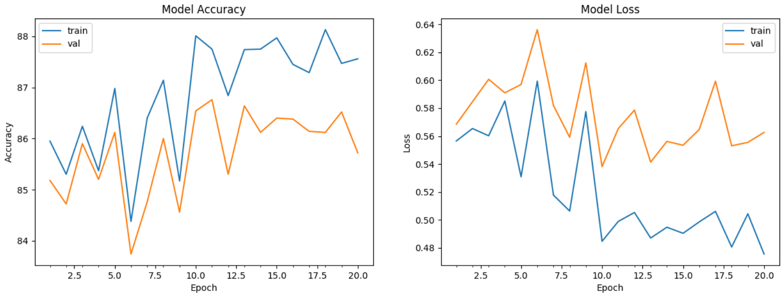
- $\eta = 0.001$



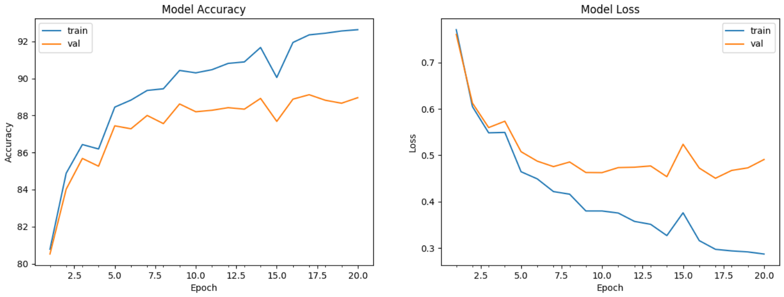
- $\eta = 0.002$



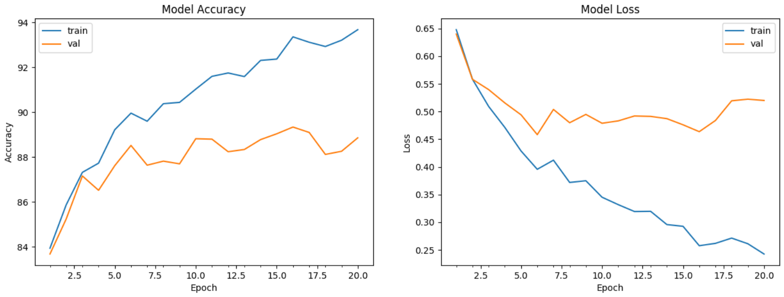
- $\eta = 0.005$



- $\eta = 0.0005$



- $\eta = 0.000095$

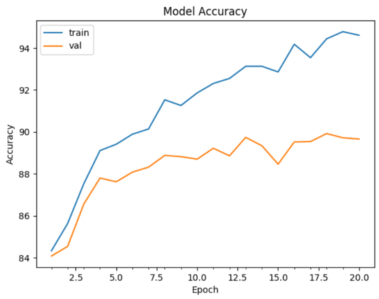


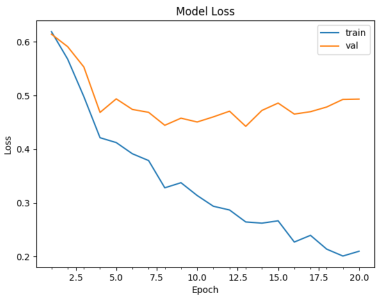
Observations:

- It was observed that between $\eta = 0.001$ and $\eta = 0.0001$, better learning rate occurs.Again we have searched in between these two, it is observed that better learning rate is $\eta = 0.001$
- One can observe that the oscillations decreases as learning rate decreases.

9. Best Models

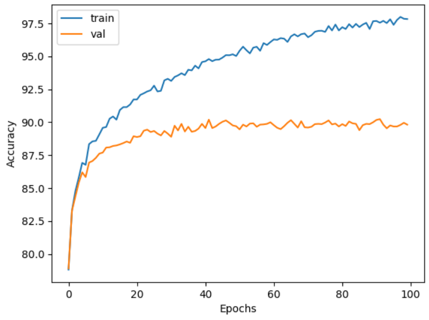
- **Model 1**
  num hidden: 2, sizes: 300,300
  activation: sigmoid, loss: ce, batch size: 50
  opt: adam, lr: 0.001, momentum1: 0.9, momentum2: 0.99, epsilon: $1e-8$
  anneal: False, dropout: False, Epochs: 20

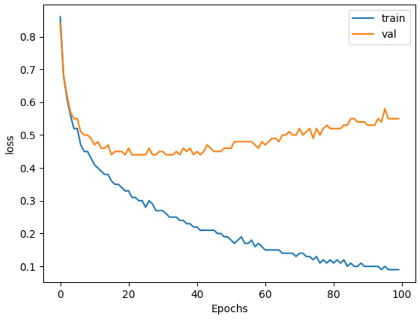




Train accuuracy	94.78@18
Valid accuracy	89.92@18
Test accuracy	89.23@18

- **Model 2**
  num hidden: 3, sizes: 200,200,200
  activation: relu, loss: ce, batch size: 50
  opt: adam, lr: 0.0001, momentum1: 0.9, momentum2: 0.99, epsilon: $1e-8$
  anneal: False, dropout: 0.8, Epochs: 100





Train accuuracy	94.8@41
Valid accuracy	90.02@41
Test accuracy	90.02@41

10. Other Experiments:

- For Model 1, we have tried DROPOUT regularization technique without changing above mentioned configuration.It was observed that model(valid) accuracy increased up to 88.8 and after that it drastically fell.
- With annealing factor 0.8, it was observed that there wasn't much improvement in Model 1.
- For Model 2, we have tried L2 regularization.But it didn't seem to perform better than DROPOUT.