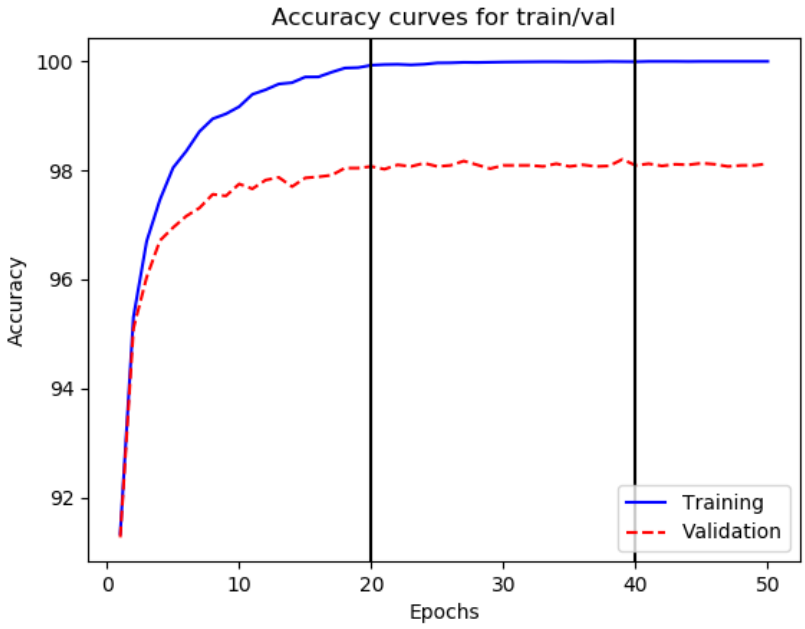
BEST MODEL --- with highest validation accuracy

- **MODEL 1**
- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) , where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 512, 10] -- 1 hidden layer
- Batch = 50
- Learning rate = 0.1
- Activation function = reLu
- Epochs = 50

On Train and Val----

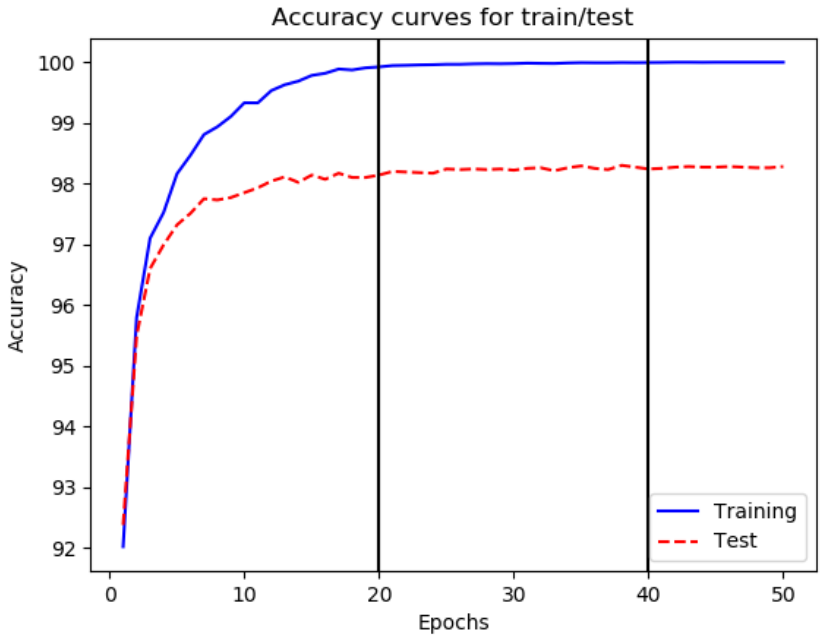


Final Training accuracy : 99.996

Final Validation accuracy : 98.11999999999999

Auto – Grade score: 98.1 Accuracy

On Train and Test ----



Final train accuracy : 99.44053333333335

Final Test accuracy : 98.28

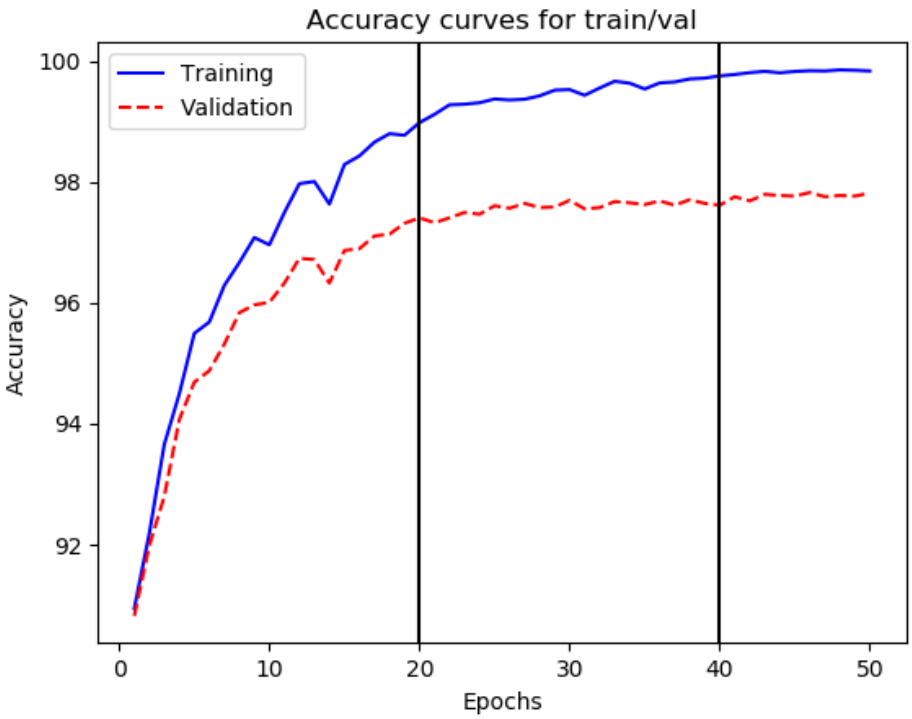
II. MODEL – 2

- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) , where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 512, 10] -- 1 hidden layer
- Batch = 50
- Learning rate = 0.1
- Activation function = Tanh
- Epochs = 50

On Train and Val----



Final Training accuracy : 99.842

Final Validation accuracy : 97.82

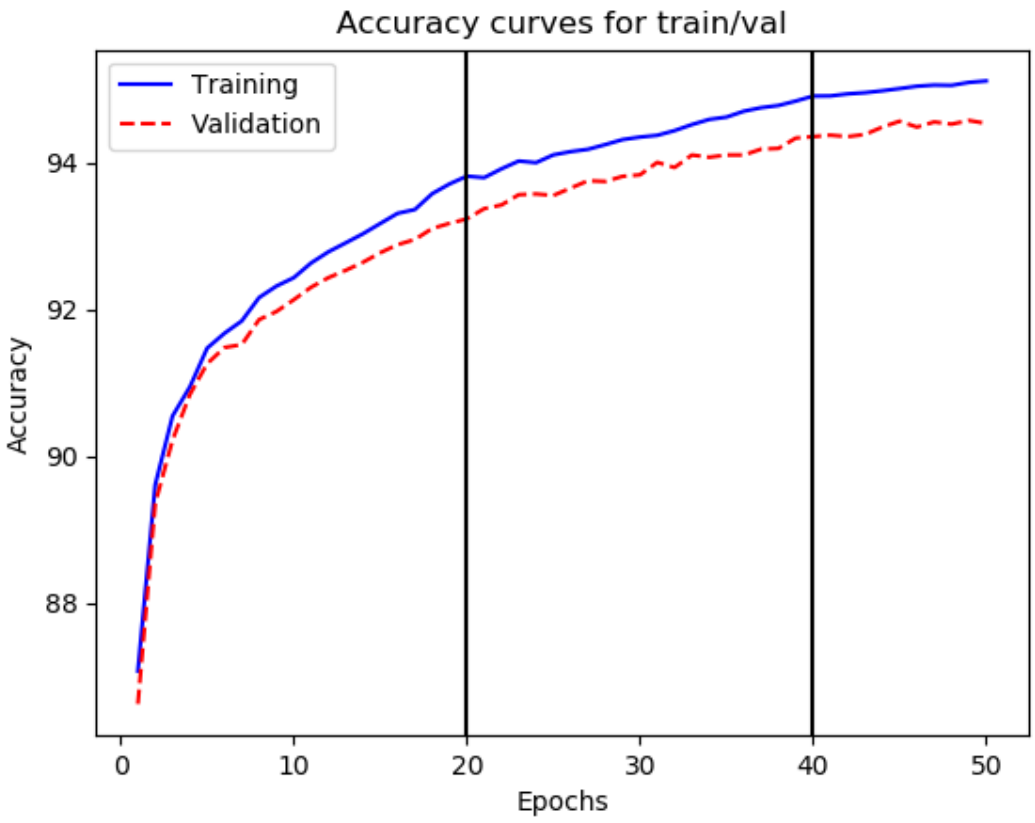
III MODEL – 3

- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) , where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 512, 10] -- 1 hidden layer
- Batch = 50
- Learning rate = 0.01
- Activation function = Tanh
- Epochs = 50

On Train and Val----



Final Training accuracy : 95.108

Final Validation accuracy : 94.53

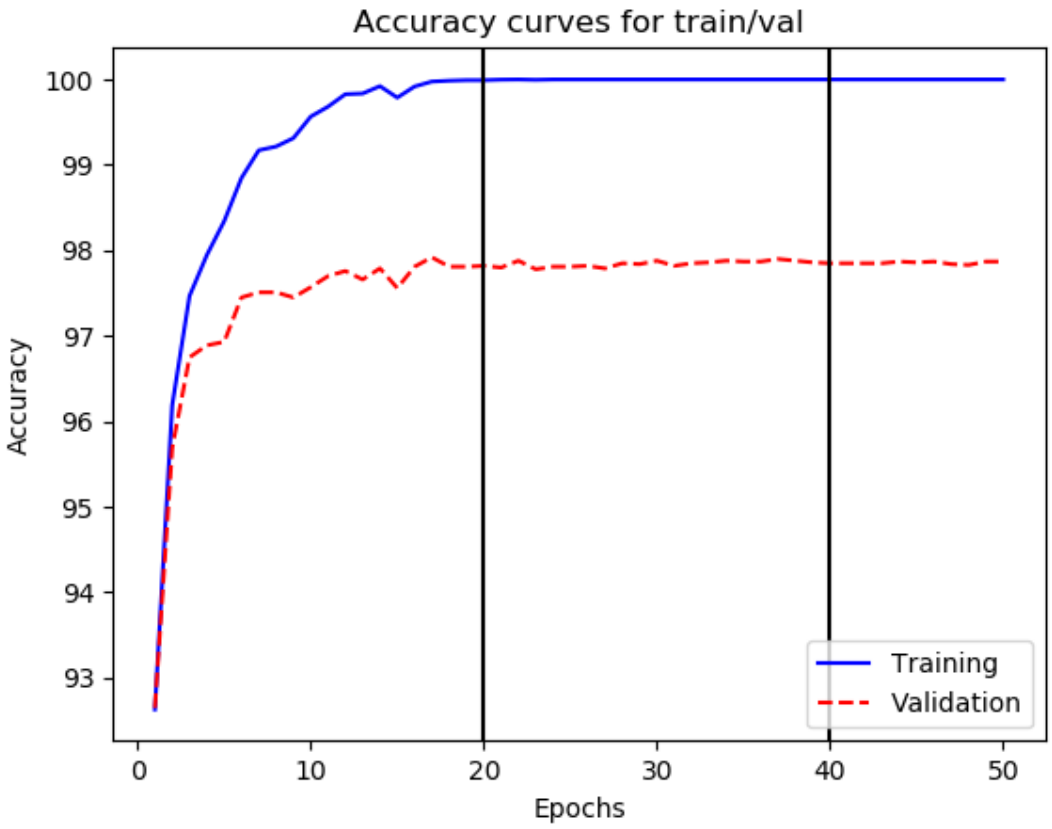
IV. Model – 4

- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) ,
  where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 200,100, 10] -- 2 hidden layers
- Batch = 100
- Learning rate = 0.15
- Activation function = ReLu , ReLu (both layers)
- Epochs = 50

On Train and Val----



Final Training accuracy : 100.0

Final Validation accuracy : 97.87

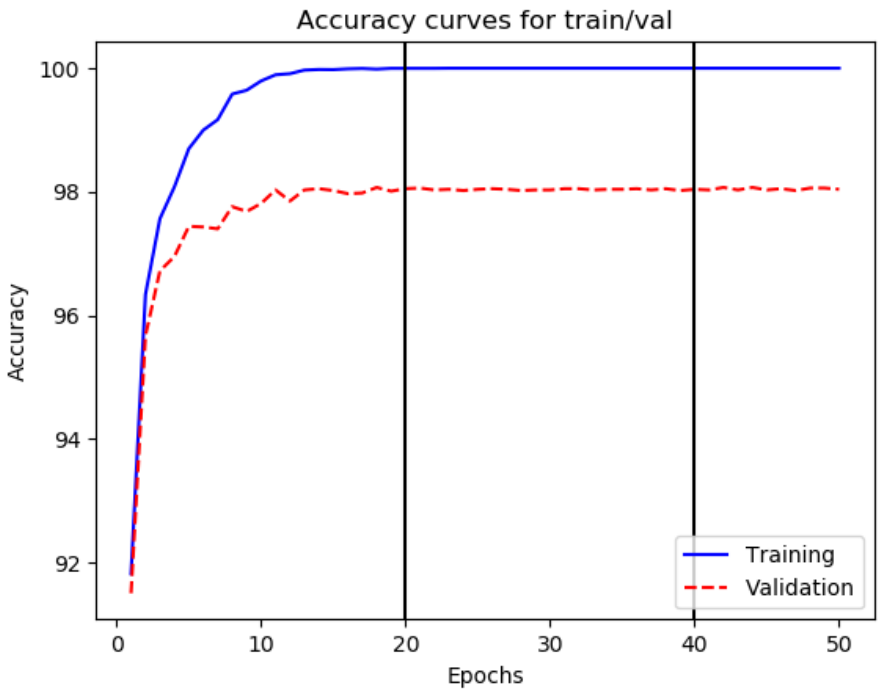
V. MODEL -- 5

- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) ,
  where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 512, 32 , 10] -- 2 hidden layers
- Batch = 50
- Learning rate = 0.1
- Activation function = ReLu , ReLu (both layers)
- Epochs = 50

On Train and Val----



Final Training accuracy : 100.0

Final Validation accuracy : 98.04

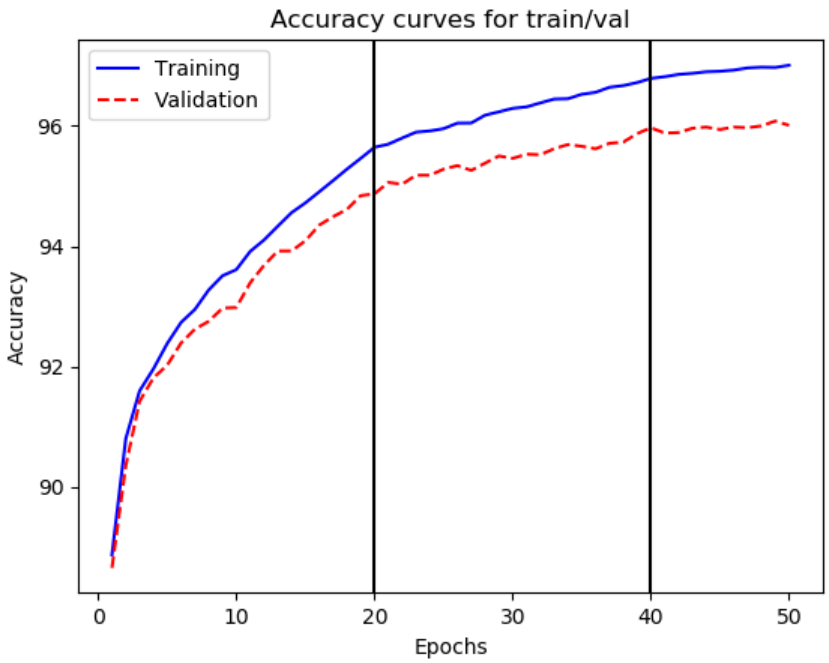
VI . MODEL – 6

- Initialization: He Initialization given by the below numpy command,

  np.random.randn(layers_config[i],layers_config[i-1]) * np.sqrt(2.0/layers_config[i-1]) , where layers_config[i] gives current layer and layers_config[i-1] gives the second output.

- Layer configuration in terms of neurons in each layer = [784, 512, 10] -- 1 hidden layers
- Batch = 100
- Learning rate = 0.025
- Activation function = Tanh
- Epochs = 50

On Train and Val----



Final Training accuracy : 97.008

Final Validation accuracy : 96.009