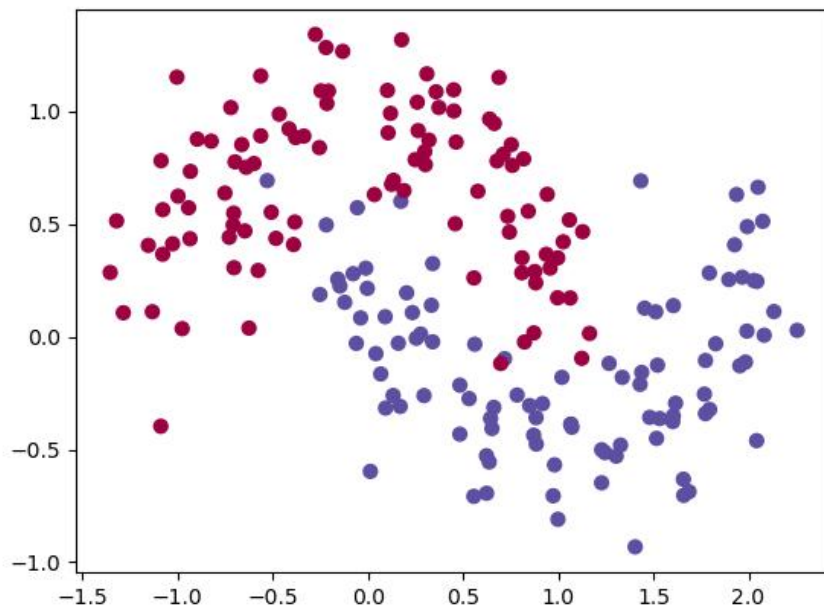


Assignment1

Author: Ruiqi Kuang

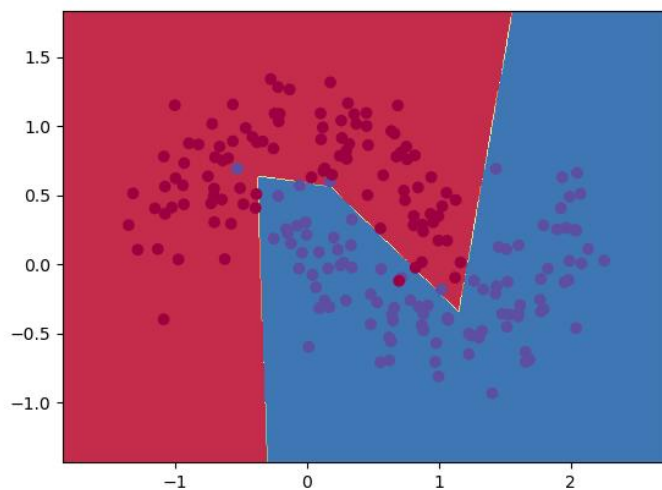
1 Backpropagation in a Simple Neural Network

TASK1: visualize Make-Moons dataset

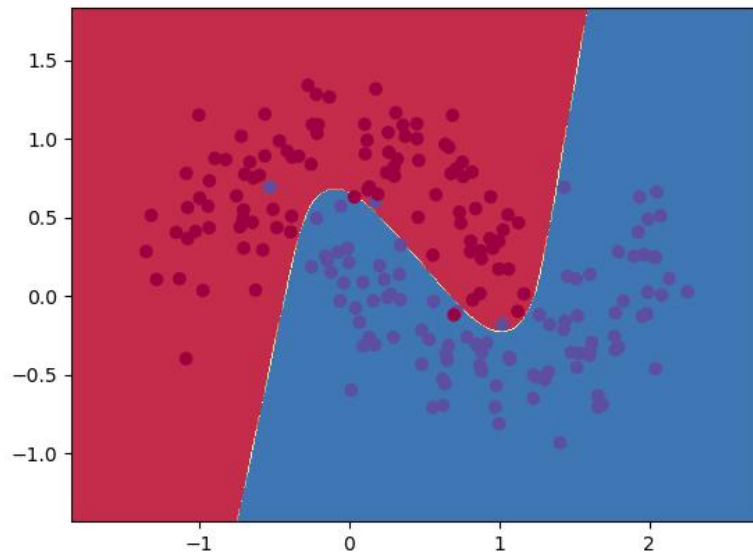


Task2: train with different activation function

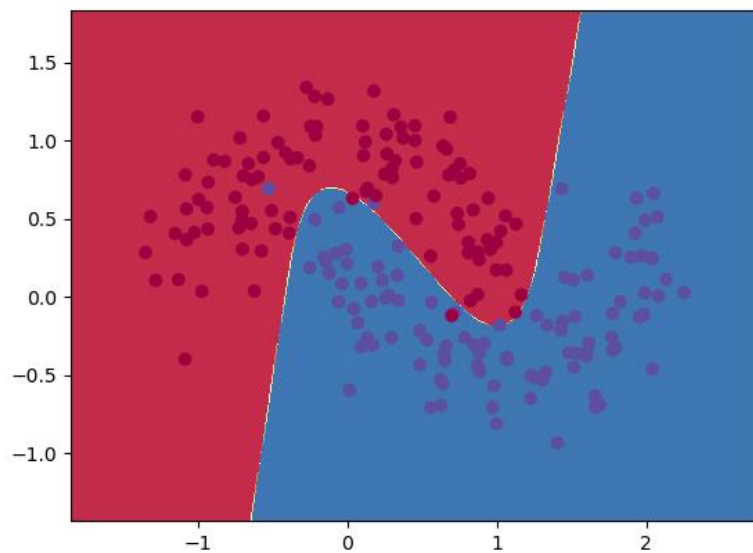
Relu:



Tanh:



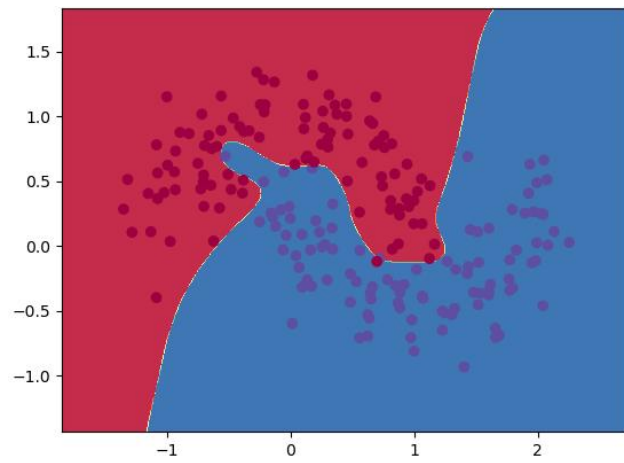
Sigmoid:



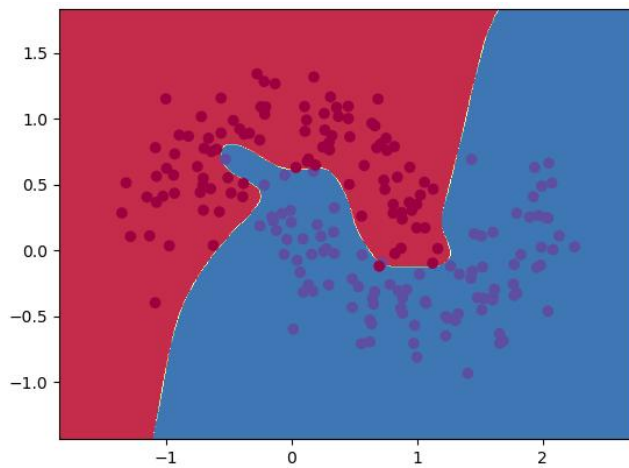
The boundary relu created is several straight lines, while tanh and sigmoid create smooth curves. That may be cause by the fact that relu is a piece-wise function where each part is a proportional function.

Task3: train with different the number of hidden units

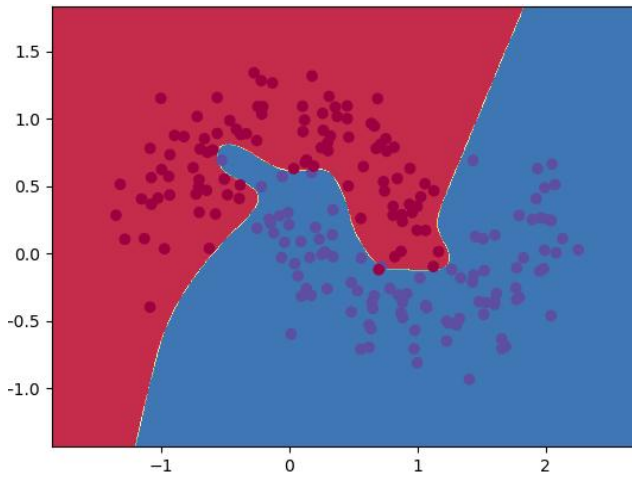
nn_hidden_dim=10



nn_hidden_dim=20



nn_hidden_dim=30



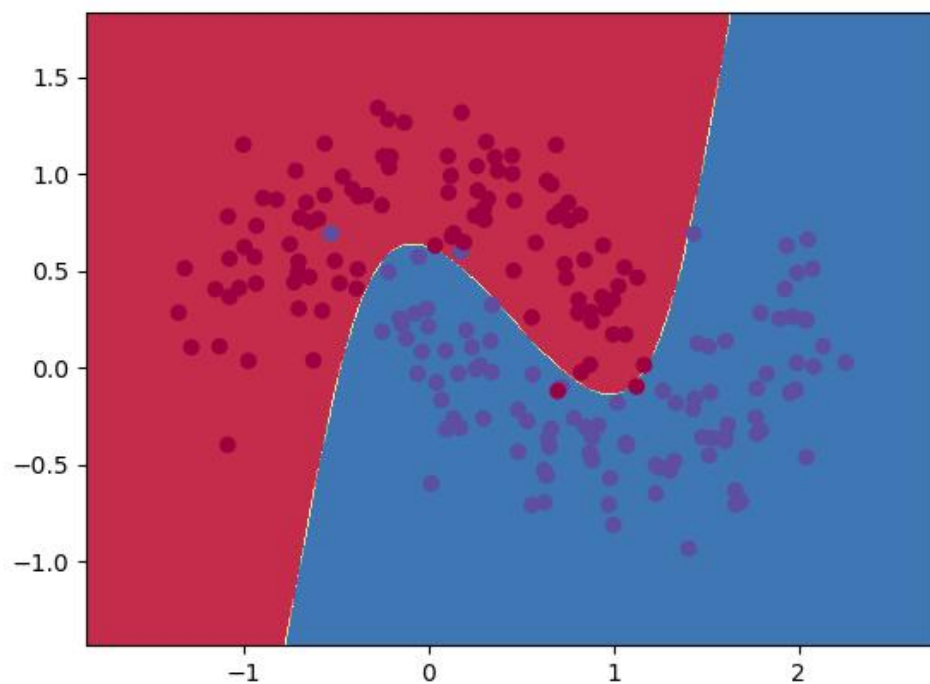
The loss decrease while the dimension of the hidden layer increase.

The decision boundary doesn't change dramatically while dimension increase.

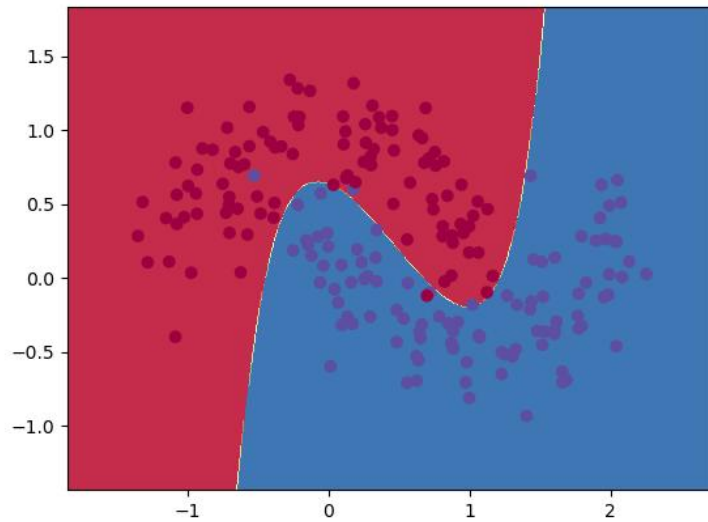
Task4: n layer neural network train make_moon

1. nn_input_dim: dimension of input layer
2. nn_hidden_dim: dimension of hidden layers
3. nn_num_layers: numbers of layers in the NN(including input layer)
4. nn_output_dim: dimension of output layer

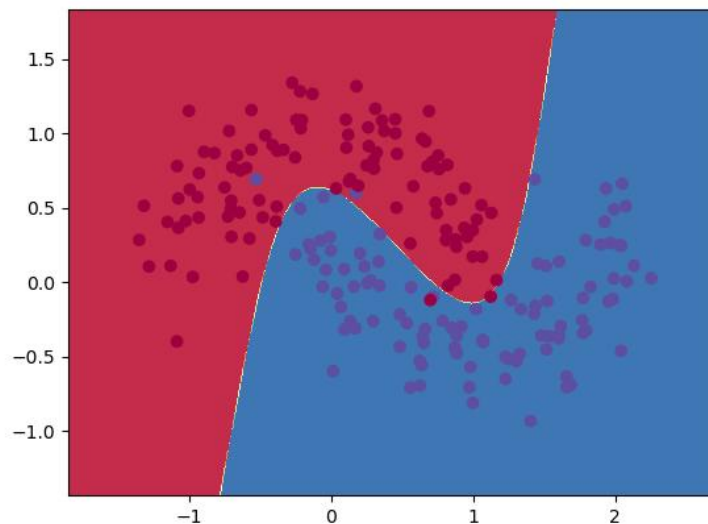
```
model = NeuralNetwork(nn_input_dim=2, nn_hidden_dim=3,  
nn_num_layers=5, nn_output_dim=2, actFun_type='tanh')
```



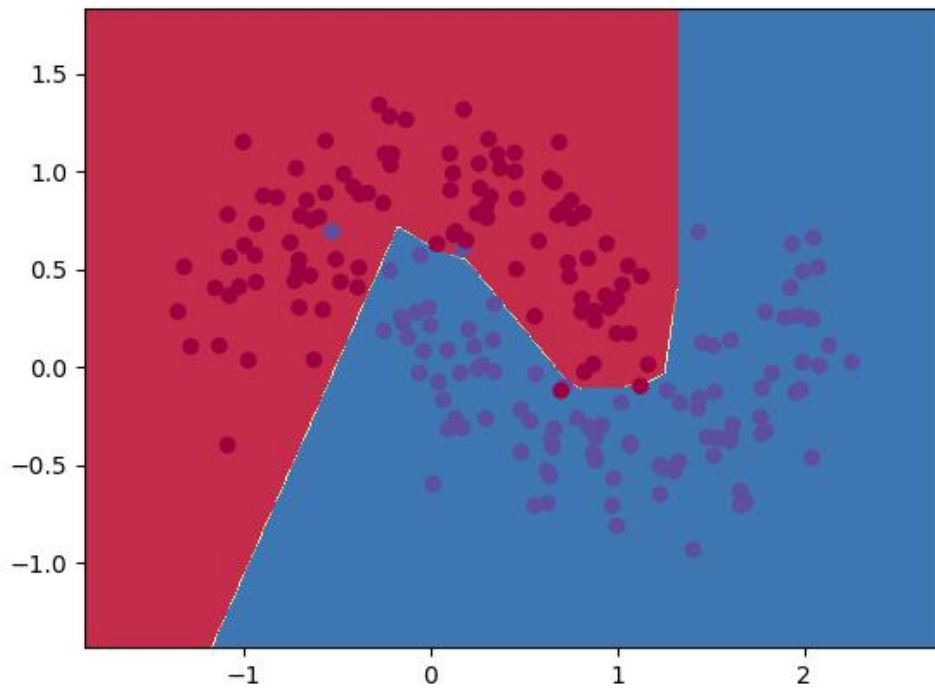
```
model = NeuralNetwork(nn_input_dim=2, nn_hidden_dim=3,  
nn_num_layers=10, nn_output_dim=2, actFun_type='tanh')
```



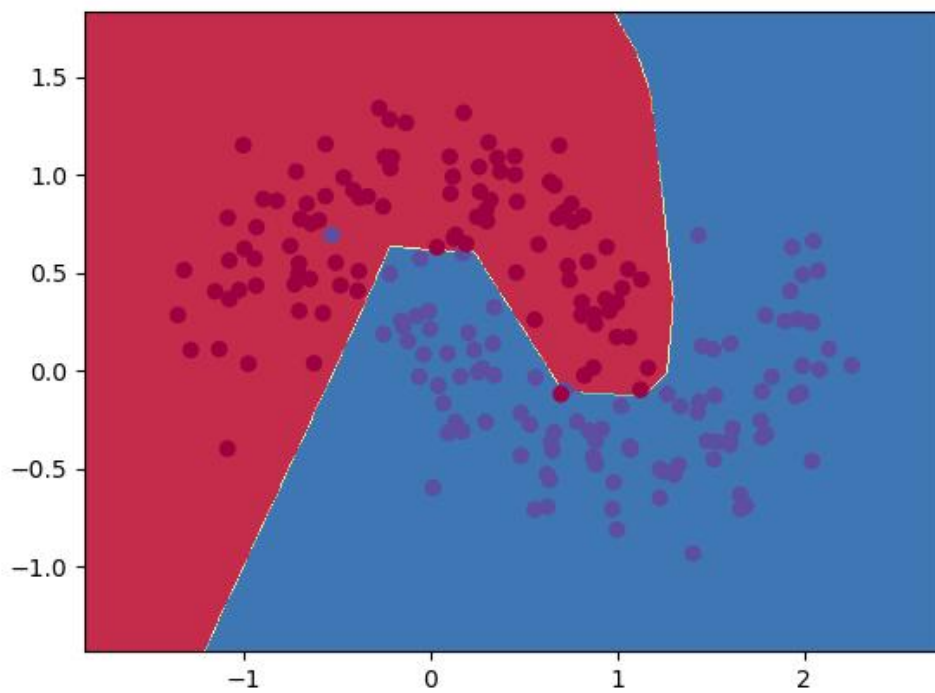
```
model = NeuralNetwork(nn_input_dim=2, nn_hidden_dim=6,  
nn_num_layers=5, nn_output_dim=2, actFun_type='tanh')
```



```
model = NeuralNetwork(nn_input_dim=2, nn_hidden_dim=6,  
nn_num_layers=5, nn_output_dim=2, actFun_type='relu')
```



```
model = NeuralNetwork(nn_input_dim=2, nn_hidden_dim=6,  
nn_num_layers=10, nn_output_dim=2, actFun_type='relu')
```



Task5: n layer neural network train make_blobs

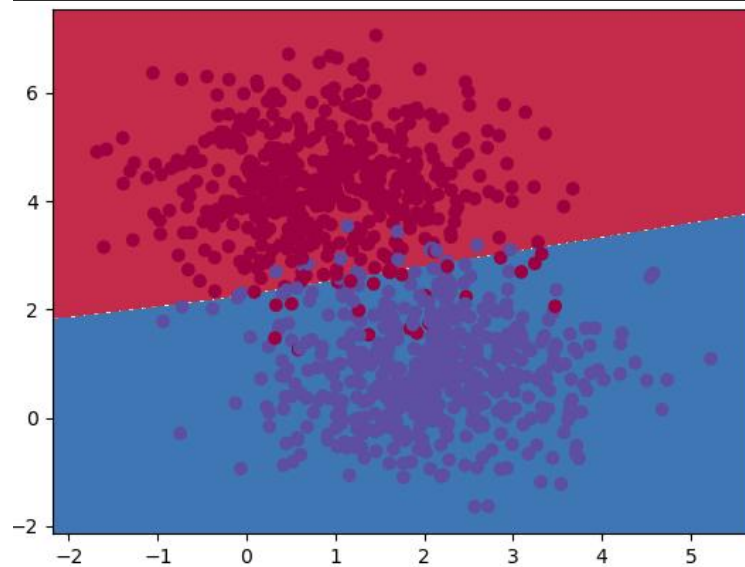
Make_blobs: Generate clustering data with normal distribution

n_sample: number of samples

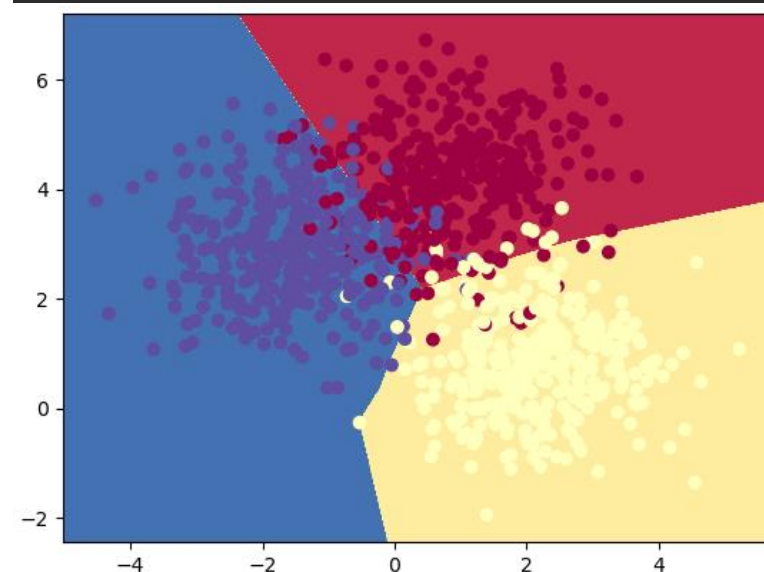
n_features: dimension of sample point(equal to the dimension of the input layer)

centers: number of cluster(equal to the dimension of the output layer)

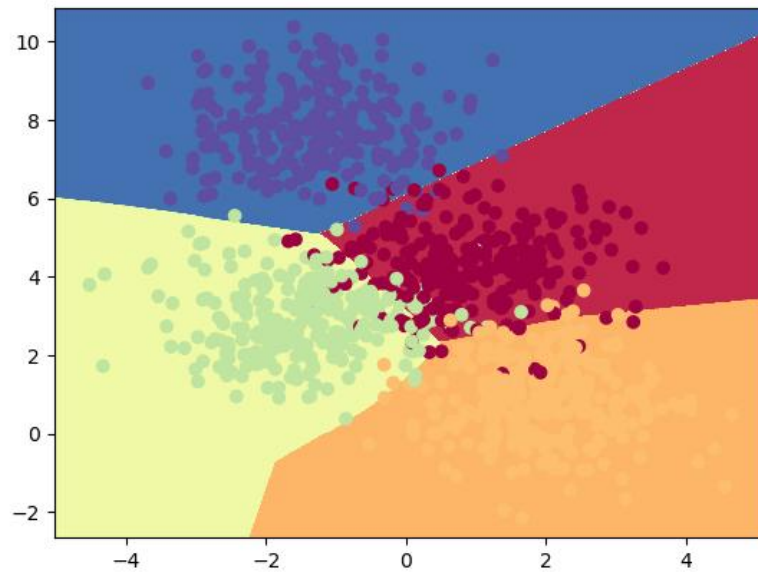
```
X, y = datasets.make_blobs(n_samples=1000, n_features=2, centers=2,  
cluster_std=1.0, center_box=(-10.0, 10.0), shuffle=True, random_state=None)
```



```
X, y = datasets.make_blobs(n_samples=1000, n_features=2, centers=3,  
cluster_std=1.0, center_box=(-10.0, 10.0), shuffle=True, random_state=None)
```

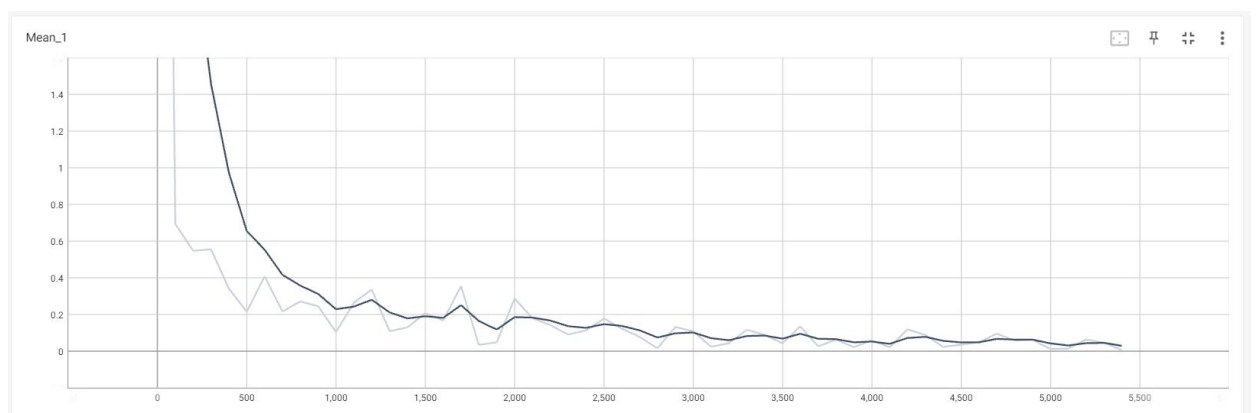


```
X, y = datasets.make_blobs(n_samples=1000, n_features=2, centers=4,
cluster_std=1.0, center_box=(-10.0, 10.0), shuffle=True, random_state=None)
```



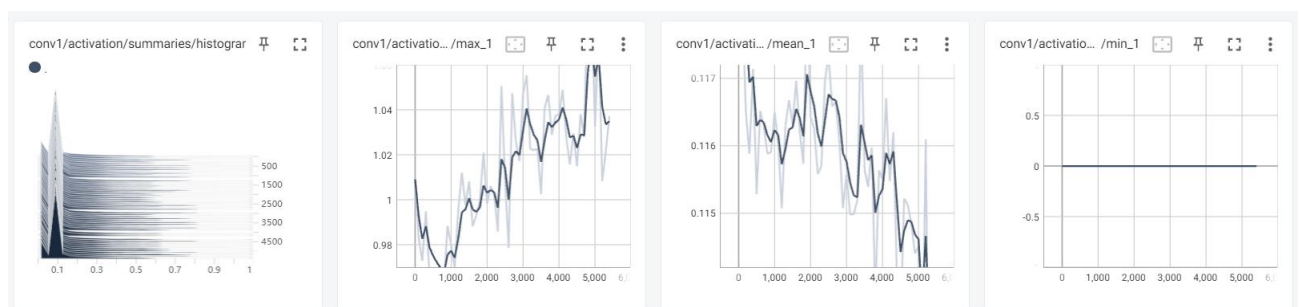
2 Training a Simple Deep Convolutional Network on MNIST

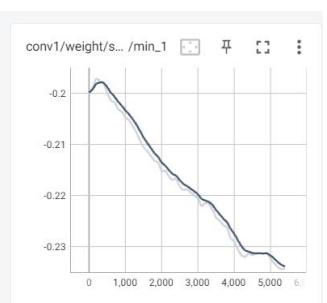
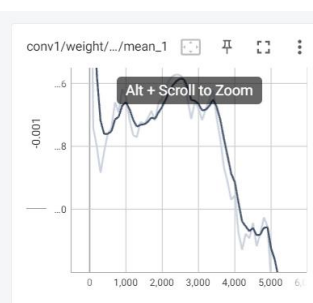
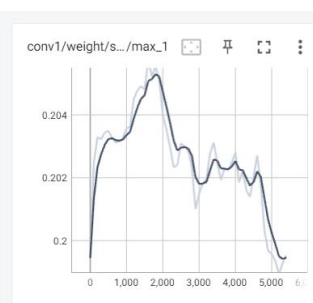
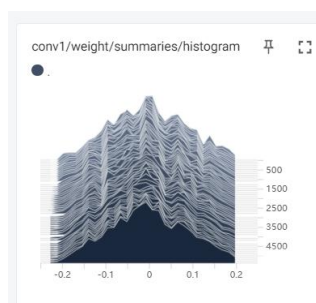
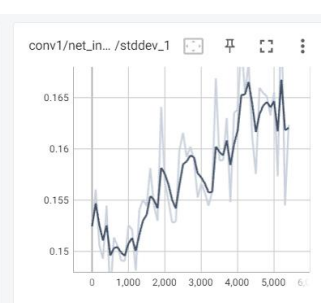
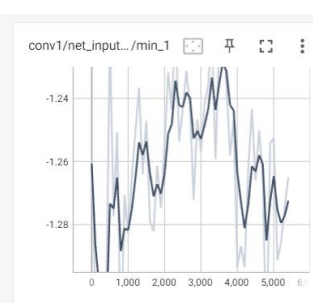
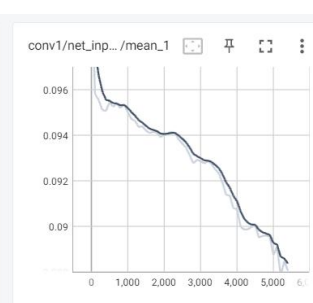
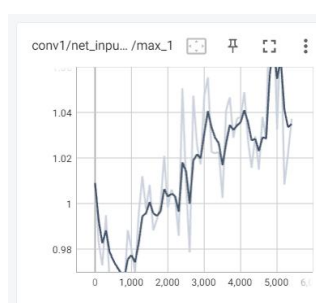
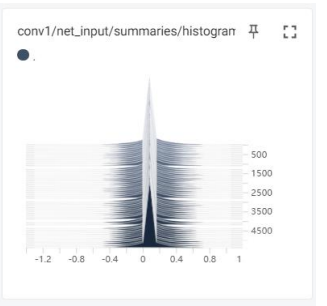
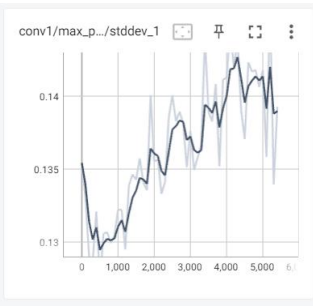
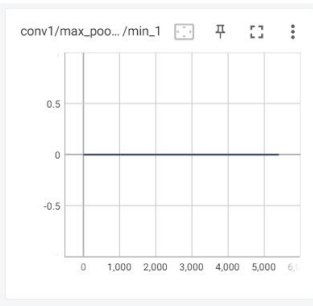
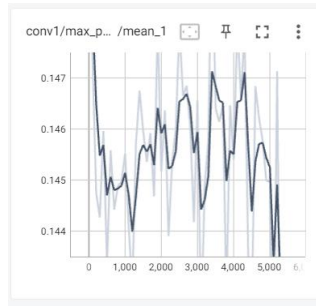
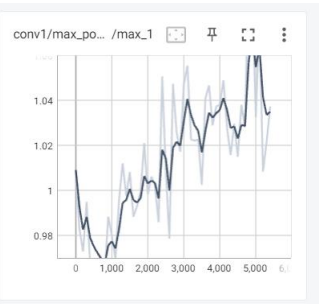
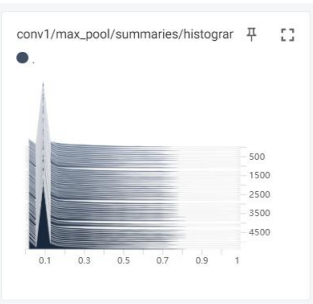
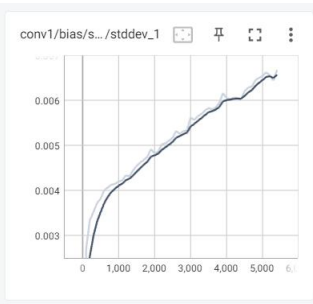
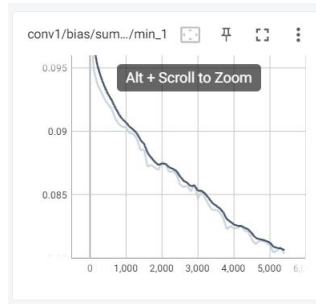
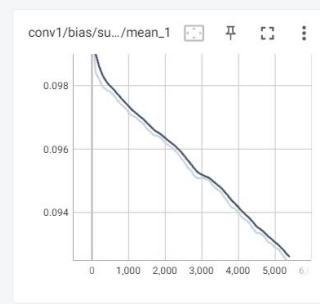
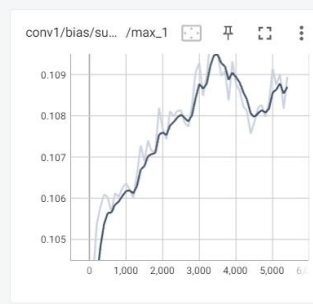
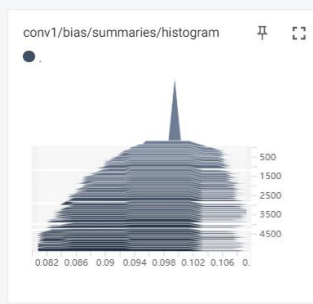
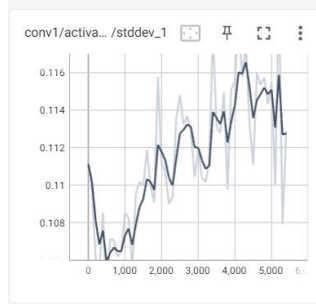
a) Build and Train a 4-layer DCN

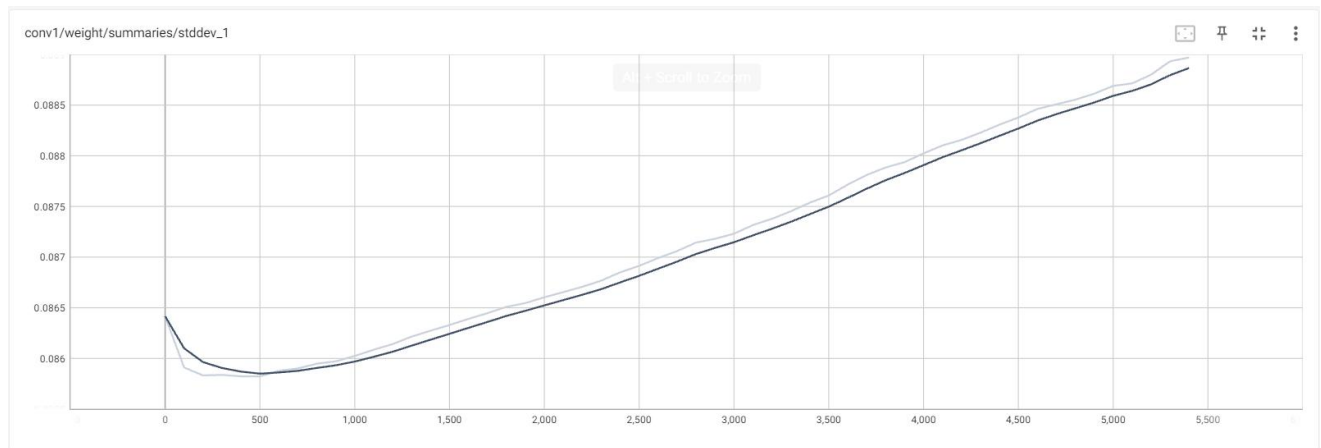


b) More on Visualizing Your Training

Conv1:







Conv2:

