Task I – Neural Network Design

1. **Fully Connected Neural Network**
   This network was built using the Sequential model and is made up of seven layers: Four(4) dense layers with three(3) dropout layers between the dense layers. The dense layers have activation functions namely relu, relu, tanh and softmax respectively and dropout of 0.2 for input layer and 0.5 for each hidden layer to prevent overfitting. The dense relu layers allow for better convergence, so they come before the tanh activation. One tanh activation was used because of how complex tanh is compared to relu. The softmax activation is used in the last layer to map non-normalized output of the neural network to a probability distribution over the predicted output classes

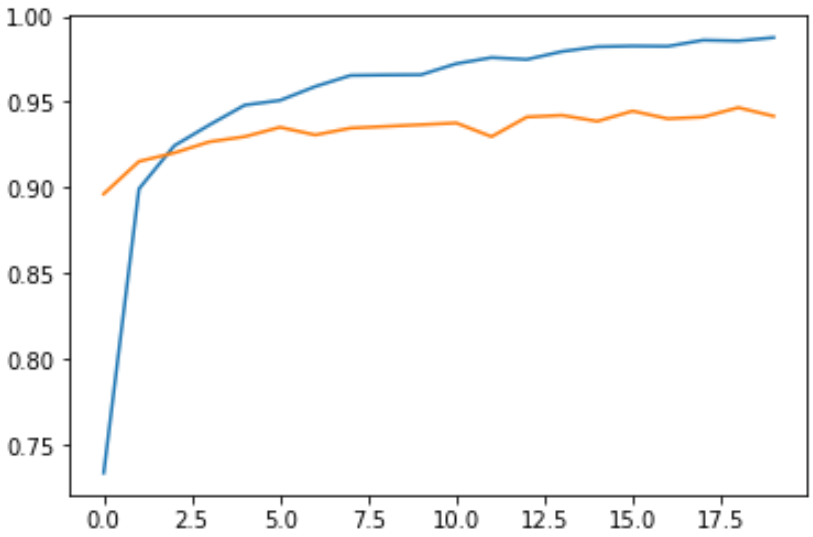
2. **Locally Connected Neural Network**
   The locally connected network was designed using a sequential model. There are nine(9) layers in this network: one locally connected 2d layer, one max pooling layer, three dense layers, three dropout layers and one flatten layer. The dense layers have activation functions namely relu, tanh and softmax respectively and dropout of 0.2 for outer layer and 0.5 for each hidden layer to prevent overfitting. The 2d layers come first, followed by a flatten layer to flatten before the dense layers. The relu layer come before the tanh layer because the relu activation function support better convergence. Sigmoid/tanh functions are more complex than relu so one tanh layer was implemented. The final layer is the softmax layer which maps the non-normalized output of the network to a probability distribution over predicted output classes.

3. **Convolutional Neural Network**
   The convolutional neural network was designed using a sequential model. T There are nine(9) layers in this network: one convolutional 2d layer, one max pooling layer, three dense layers, three dropout layers and one flatten layer. The dense layers have activation functions namely relu, tanh and softmax respectively and dropout of 0.2 for outer layer and 0.5 for each hidden layer to prevent overfitting. The 2d layers come first, followed by a flatten layer to flatten before the dense layers. The relu layer come before the tanh layer because the relu activation function support better convergence. Sigmoid/tanh functions are more complex than relu so one tanh layer was implemented. The final layer is the softmax layer which maps the non-normalized output of the network to a probability distribution over predicted output classes.

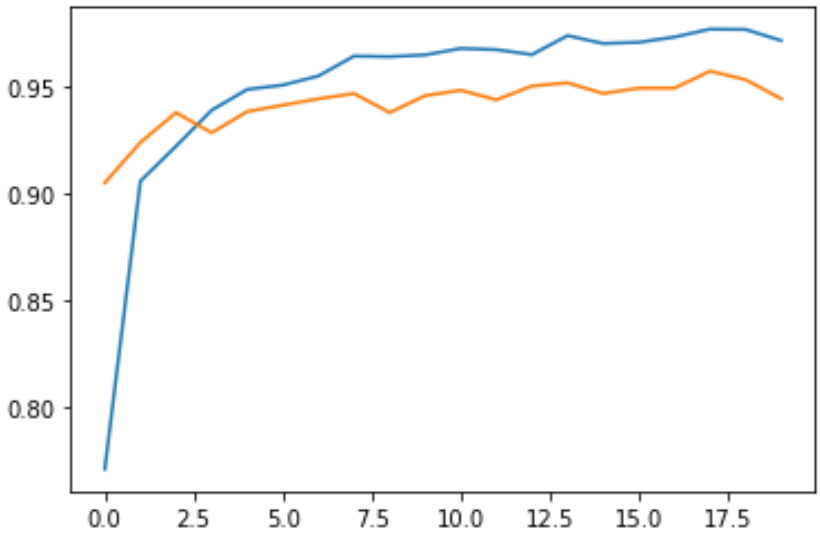
Task II - Techniques for Optimization

1. Parameter Initialization Strategies
   To get the appropriate parameters that will allow the models converge, I set the learning rate and momentums to values with small increments, also I set the epochs and batch size to small values to find the optimal values allow the models converge. I recorded values that

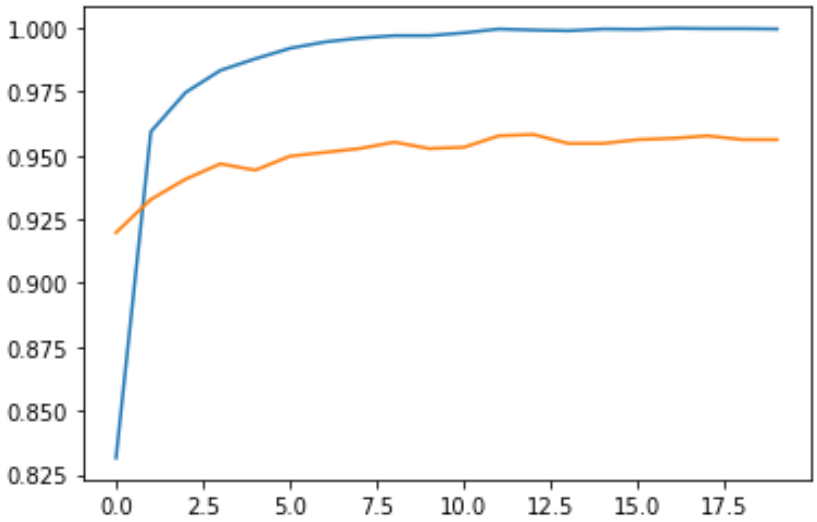
allowed the models converge. For all three(3) neural networks I used **20** epochs and batch size of **16.**



Fully Connected Network(learning rate = 0.01, momentum = 0.5)



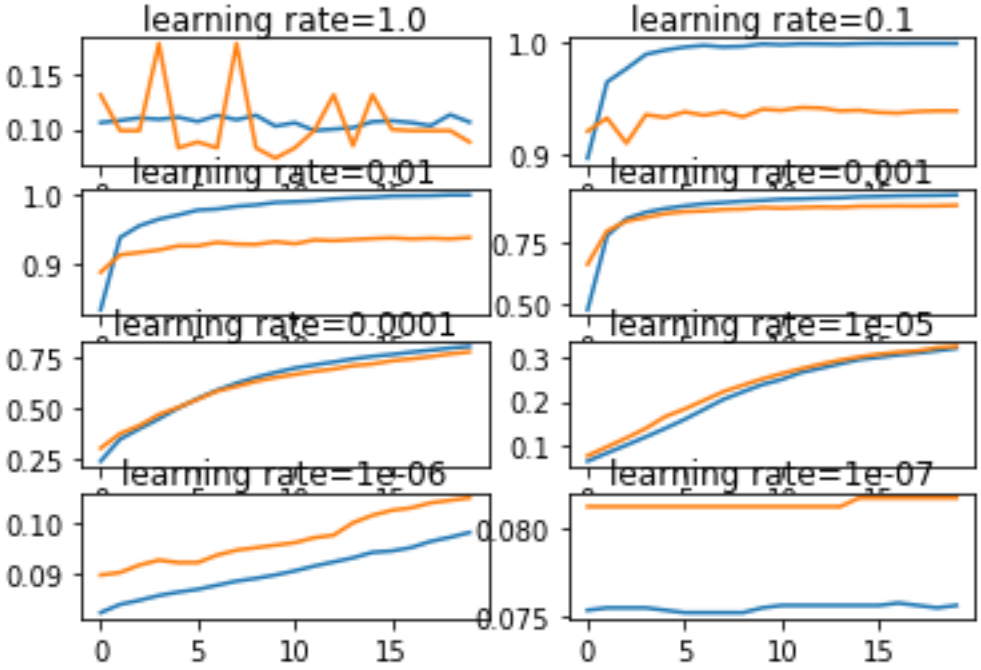
Locally Connected Network(learning rate = 0.1, momentum = 0.5)



Convolutional Network(learning rate = 0.1, momentum = 0.5)

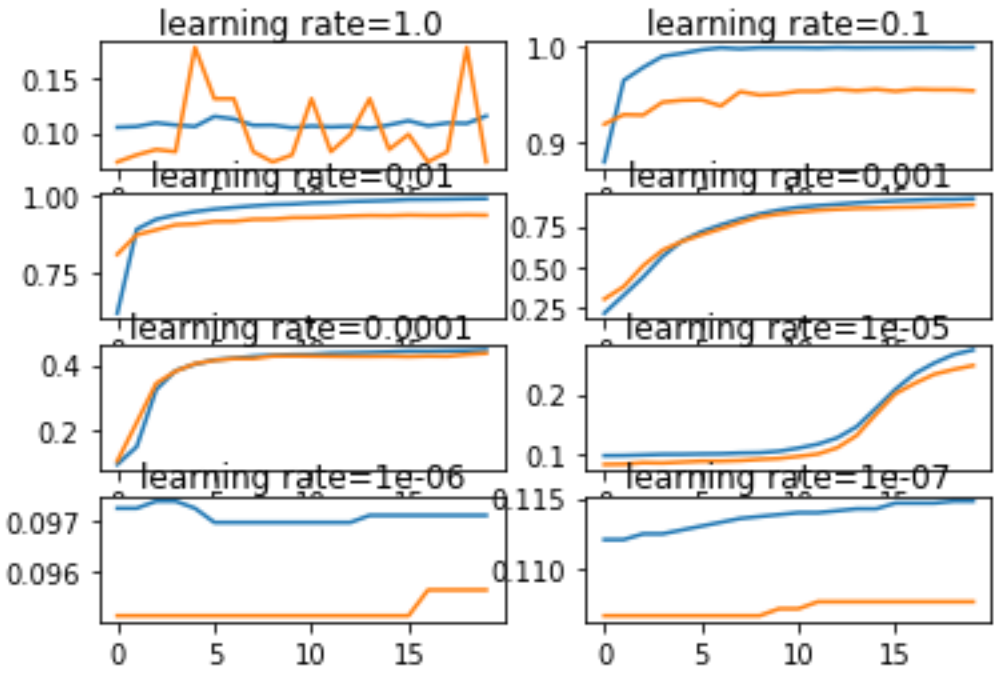
2. Learning Rate

Fully Connected Neural Network



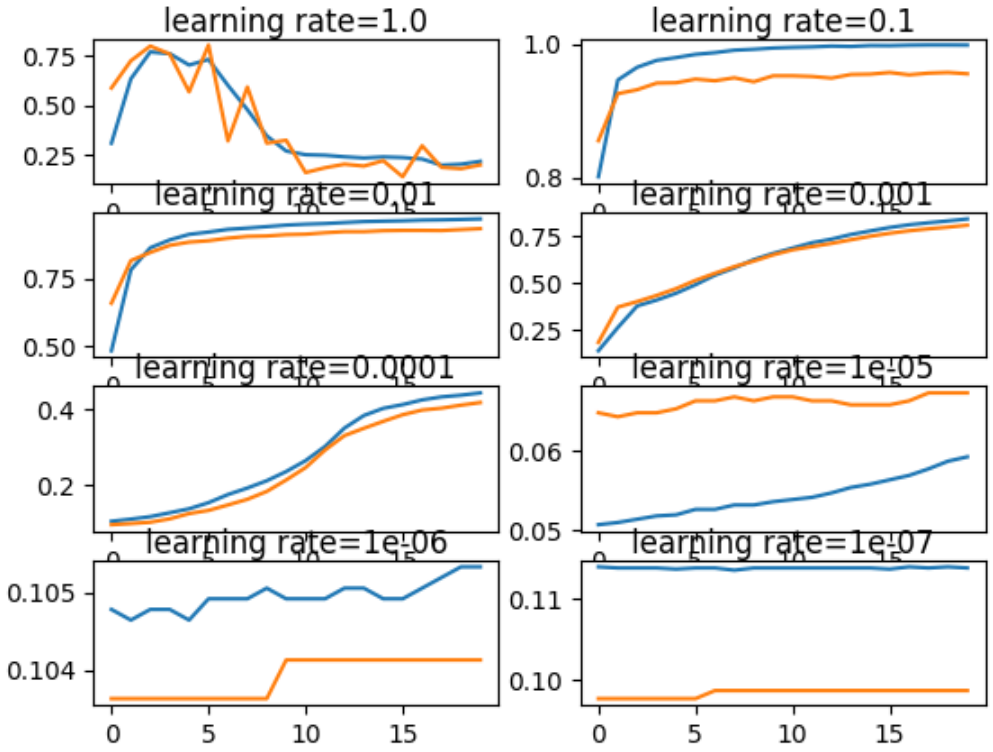
From The above diagram, learning is **slow** when learning rate is **1E-6, 1E-7** and learning is too **fast** when learning rate is **1.0**, learning is **effective** when learning rate is **1E-1, 1E-2, 1E-3.**

Locally Connected Neural Network



From The above diagram, learning is **slow** when learning rate is **1E-6, 1E-7** and learning is **fast** when learning rate is **1.0**, learning is **effective** when learning rate is **1E-1, 1E-2, 1E-3.**

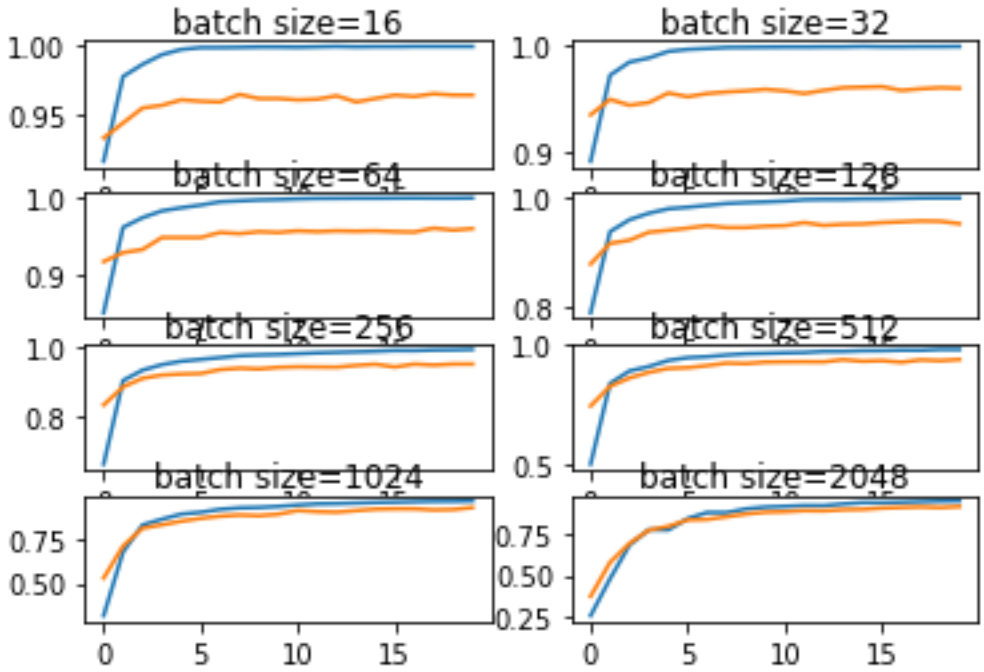
Convolutional Neural Network



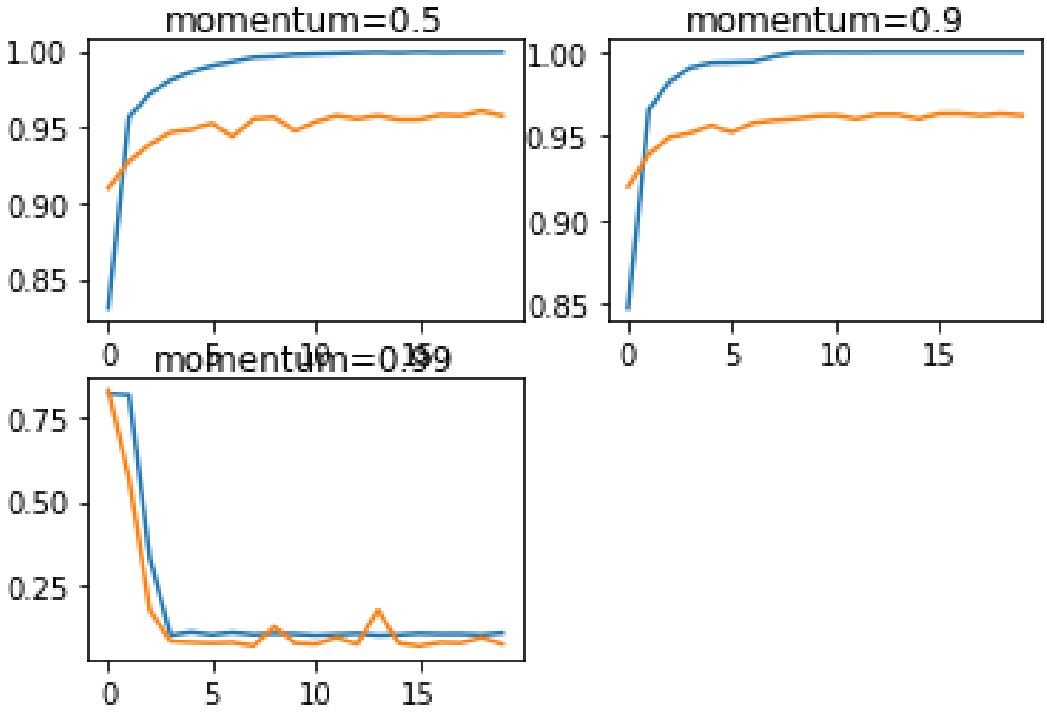
From the above diagram, learning is **slow** when learning rate is **1E-6, 1E-7** and learning is **fast** when learning rate is **1.0**, learning is **effective** when learning rate is **1E-1, 1E-2, 1E-3.**

3. **Batch Size Impact**

The plots below show the performance of the convolutional neural network on batch sizes 16, 32, 64, 128, 256, 512, 1024, 2048 respectively. All these batch sizes generalize well on the data, but the accuracy goes down as the batch size increases.



4. Momentum



The model generalizes well when the momentum is 0.5, 0.9, and the model does not generalize well when the momentum is 0.99.

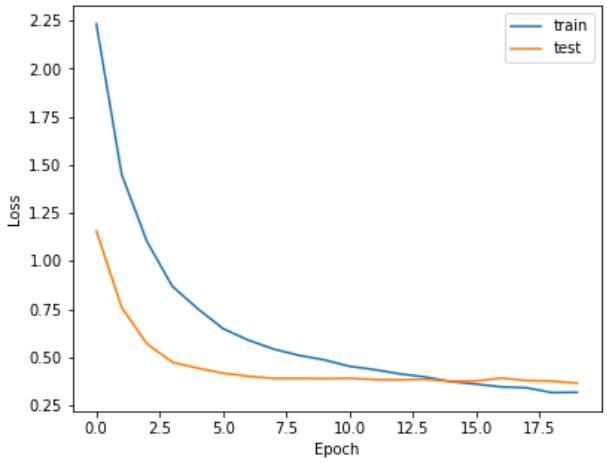
Task III - Techniques for Improving Generalization

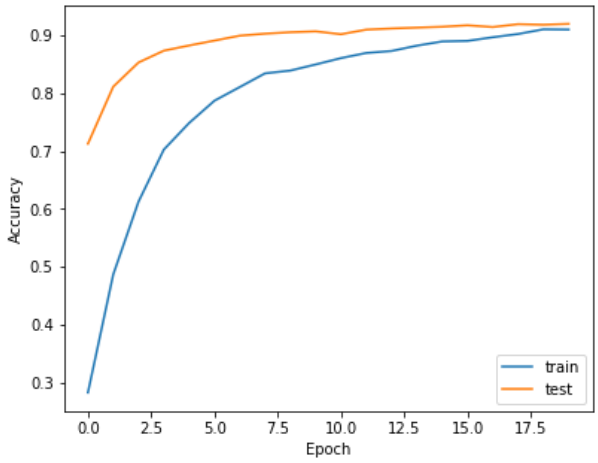
1. Ensemble
   The ensemble model was created using four(4) models(1 fully connected, 1 locally connected, 2 convolutional networks) and finding the average output of the 4 models. I observed **a higher generalization after ensemble.**

   Train Accuracy: 0.9998628446029352
   Test Accuracy: 0.9641255605381166

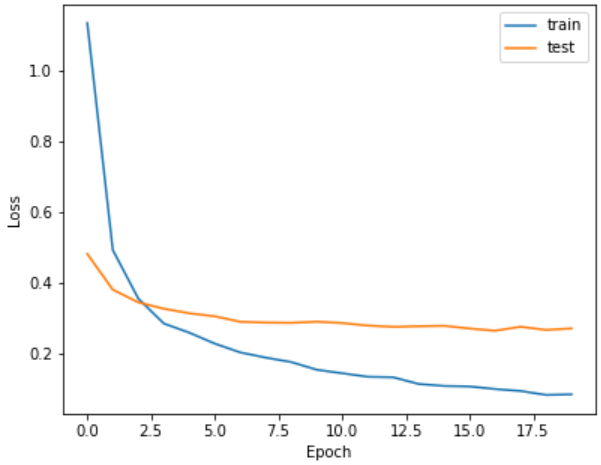
2. Dropout
   Fully Connected
   In the diagram below, an effective dropout case is recorded when the dropout for the visible layer is set to 0.2 and 0.5 for each hidden layer ineffective case is demonstrated when the dropout rate for all dropout layers is set to 0.7 which is high and the network doesn't learn properly.

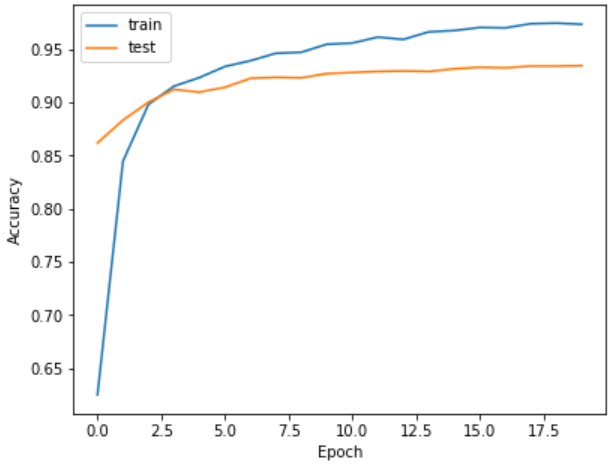
Ineffective Case





Effective Case





Extra Credit Options

1. **Adversarial Training**
   A model for adversarial training was built and some images were created, below is the best image created by the network. The model consists of the generator and discriminator model. The discriminator model was built using the sequential model and has seven(7) layers: 2 conv2D, 2 LeakyRelu, 1 flatten and 1 dense layer. The generator model was built using the sequential model and has seven(7) layers: 2 conv2DTranspose, 3 LeakyRelu, 1 reshape and 1 dense layer. The composite model consist of both the generator and discriminator model. Training was not allowed for the discriminator model.

Image generated by the generator adversarial network.