

**Include the writeup and all your code in a zip file (DATA255\_Assignment2\_Name\_SJSU\_ID) before the due time. (Use one .py file or a Jupyter notebook for each experiment on one model variant. Name the files as minivgg, var1, var2 and var3 and var4.)**

Implement a mini-vgg network and several of its variants. Train and test the models on the cifar-10 dataset. You need to have separate training, validation, and test set. Tune your parameter using validation dataset.

Investigate and compare the performance of the variants to the original model. Write half to one page to 1) summarize your experiment results and discuss 2) the classification performance of the models as well as 3) their size (# of parameters) and 4) the computation time to train them.

Mini-VGG network structure:

Layer	Type(window size) – n filters
1	Conv3 – 64
2	Conv3 – 64
3	Maxpool – 2x2
4	Conv3 – 128
5	Conv3 – 128
6	Maxpool – 2x2
7	Conv3 – 256
8	Conv3 – 256
9	Maxpool – 2x2
10*	fully Connected 512

\* Note your need a reshape layer before this layer to reshape the data

Hint: Use cross entropy loss (torch.nn.CrossEntropyLoss or tf.nn.softmax\_cross\_entropy\_with\_logits()) feed the loss function with the logits without activation function for the last layer but get the prediction accuracy after applying softmax activation)

Report the performance of each network by doing the following:

A) Plot training loss vs validation loss

B) Plot training accuracy vs validation accuracy

C) Calculate test accuracy

1. Implement the mini-vgg model and report its performance. Use ReLU activation function for all the all conv/fc layers except the last one.
2. **Variant 1**: Change the ReLU activation functions to SELU and Swish. Would the performance improve?
3. **Variant 2**: Remove the maxpool layers. Using stride=2 in the conv layer before the maxpool to achieve similar size reduction. Would the performance improve?
4. **Variant 3**: Add a few dropout layers in the model. Would the performance improve? Try 2 different ways to add the dropout layers. Describe the ones you tried and their performance.

5. **Variant 4:** Remove layers 9 and 10. Add two layers of (1, 1) convolution: conv (1, 1) x 128; conv (1, 1) x10. Then add “GlobalAveragePooling2D” to merge feature maps. This is an all-convolutional structure (no fully connected layers).