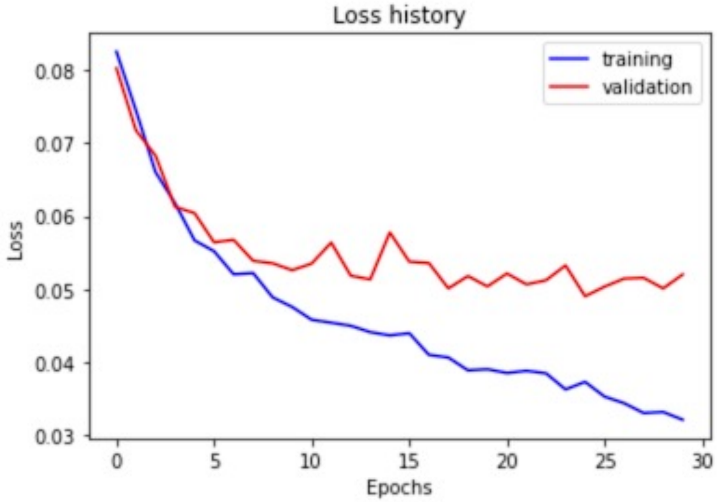
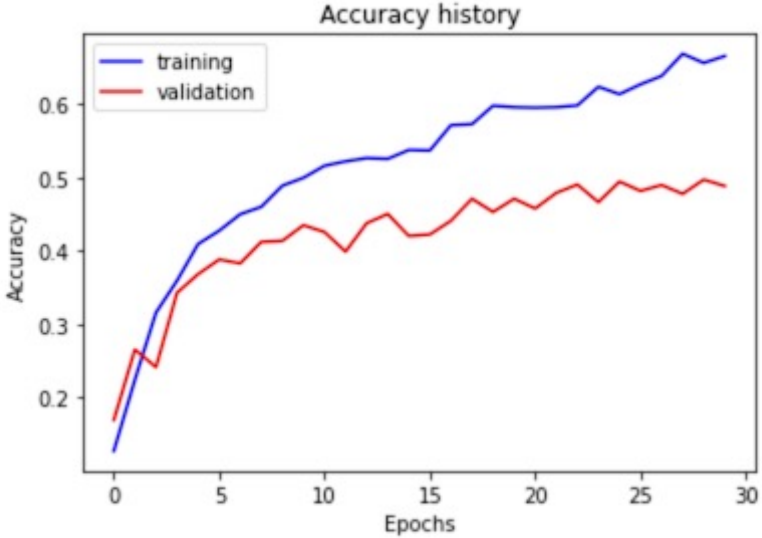
CS 6476 Project 5

Bojun Yang
byang301@gatech.edu
byang301
903254309

Part 1: SimpleNet





Final training accuracy: 0.6650

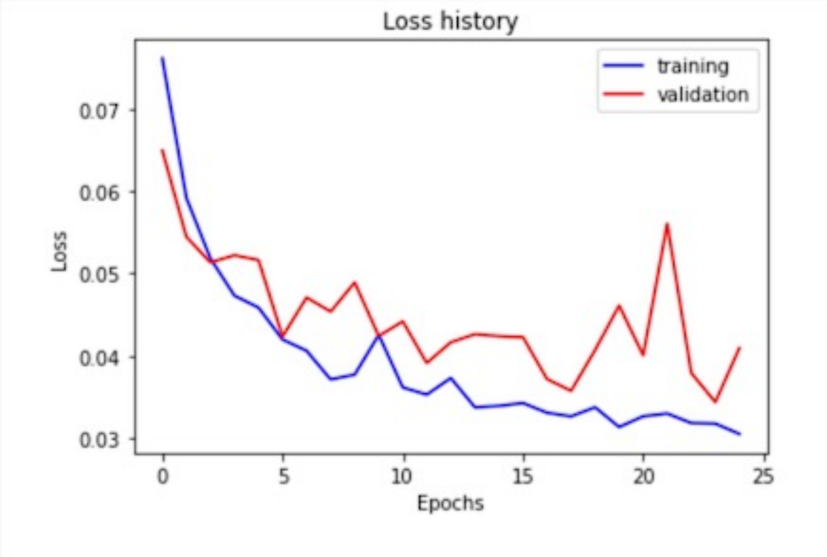
Final validation accuracy: 0.4880

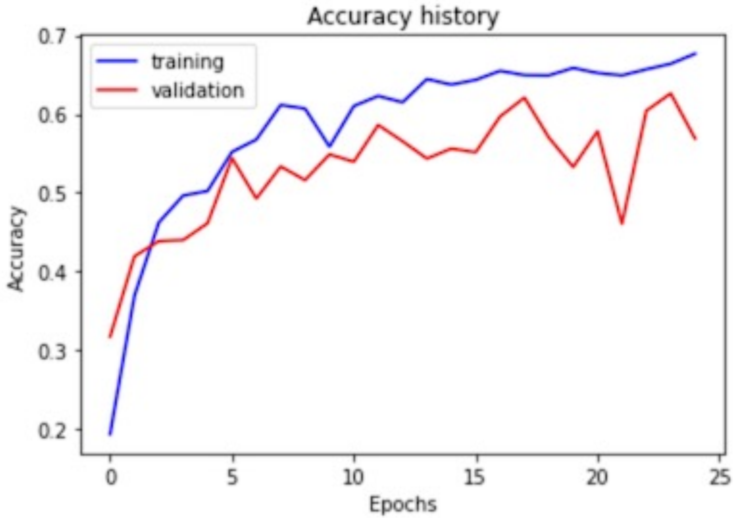
Part 2: SimpleNetFinal

Add each of the following (keeping the changes as you move to the next row):

	Training accuracy	Validation accuracy
SimpleNet	0.6650	0.4880
+ Jittering	0.5585	0.4773
+ Zero-centering & variance-normalization	0.7005	0.5560
+ Dropout regularization	0.6972	0.6082
+ Making network "deep"	0.7819	0.5907
+ Batch normalization	0.8322	0.6313

Part 2: SimpleNetFinal





Final training accuracy: 0.6764

Final validation accuracy: 0.5687

Part 2: SimpleNetFinal

[Name 10 different possible transformations for data augmentation.]
Horizontal flip, vertical flip, scaling, cropping, padding, rotation, translation, increase/decrease brightness, increase/decrease contrast, increase/decrease saturation

[What is the desired variance after each layer? Why would that be helpful?]
In general, we want low variance after each layer. The variance of a model is the difference between validation error and training error, so we want that to be small. A big variance means we are overfitting. In theory, if we have low variance, we can increase the size of the model to increase the accuracy on the training set, and if variance remains small, that means our accuracy on the validation set will also increase.

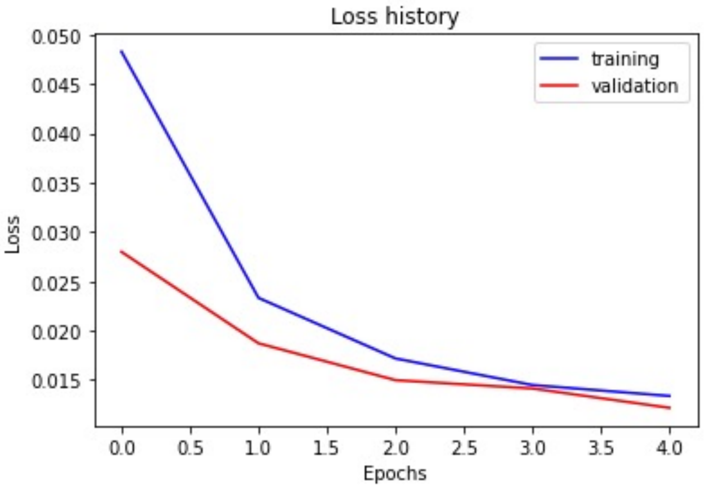
Part 2: SimpleNetFinal

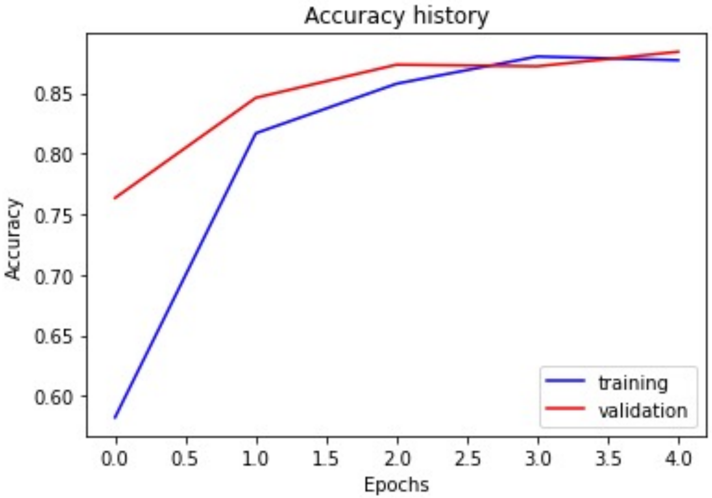
[What distribution is dropout usually sampled from?]
Drop out is usually sampled from the output of a convolutional layer.

[How many parameters does your base SimpleNet model have? How many parameters does your SimpleNetFinal model have?]
SimpleNet has 6 parameters in the conv layers and 2 in the fully connected layers.
SimpleNetFinal has 11 in the conv layers and 2 in the fully connected layers.

[What is the effect of batch norm after a conv layer with a bias?]
If the distribution in the conv layer with a bias changes rapidly, it will also affect the efficiency of the layer after the conv layer. BatchNorm mitigates this by normalizing the hidden unit activation values.

Part 3: ResNet





Final training accuracy: 0.877

Final validation accuracy: 0.884

Part 3: ResNet

[Insert visualization of confusion matrix obtained from your final ResNet model.]

Part 3: ResNet

[Insert visualizations of 3 misclassified images from the most misclassified class according to your confusion matrix. Explain why this may have occurred.]

Part 3: ResNet

[What does fine-tuning a network mean?]
Fine tuning a network means to take a pre-trained model that has been trained for another task and tweak it to make it perform a second similar task. ResNet was trained to recognize 1000 classes so we took it and fine tuned it to recognize our 15 classes.

[Why do we want to "freeze" the conv layers and some of the linear layers from a pre-trained ResNet? Why can we do this?]
We want to freeze the layers for the pre-trained ResNet since updating the weights for these layers will take a lot of computation. We can do this by turning off gradient descent for those layers and only updating weights for the last fully connected layer.

Extra credit (optional)

[Discuss what extra credit you did and provide analyses.]