Convolutional Neural Networks for CIFAR-10 Image Classification

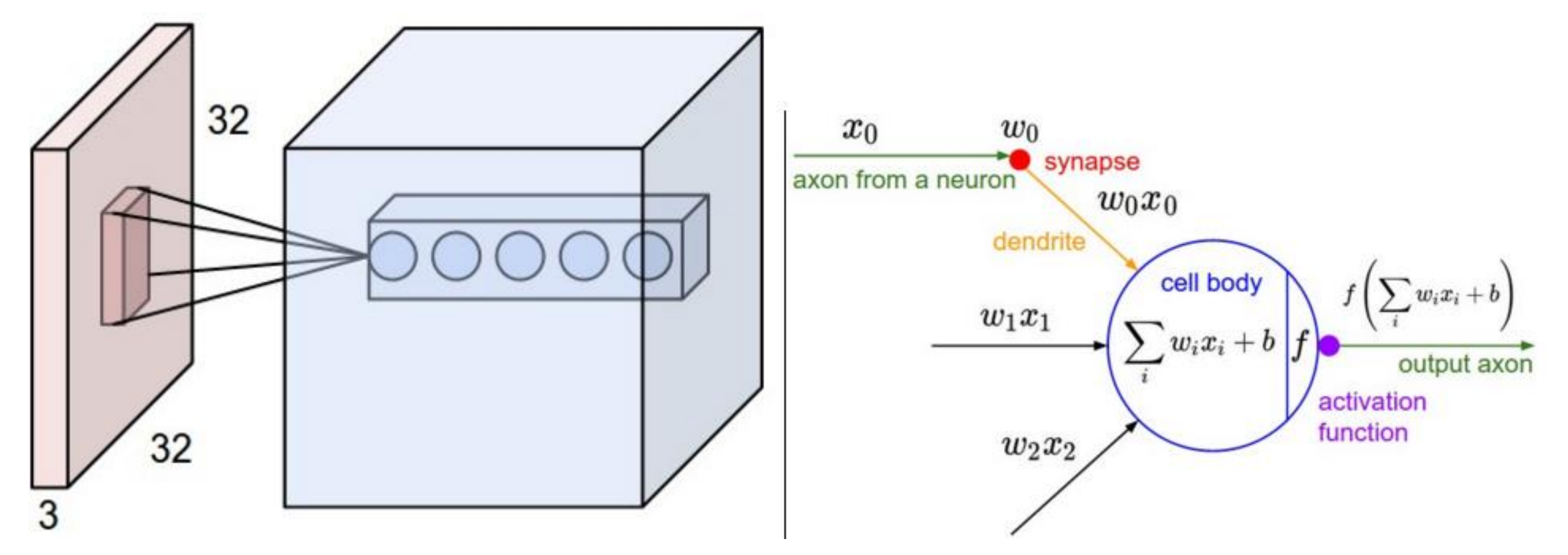
Chia-Sheng Hsu, Jingting Yao, Mythili Vadali, Yi-Chi Shao

Motivation: Convolutional neural networks (CNNs) are specialized regular neural networks models which exploit the spatial correlation and variabilities of 2D shapes to extract features, and therefore are one of the best among all the online and offline image classification networks. This project is intended to build up understanding of the architecture of CNNs and implement CNN based image classification.

Methodology

Convolutional Neural Networks (CNNs)

When used for image recognition, CNNs consist of multiple layers of small neuron collections which process portions of the input image, called receptive fields. The outputs of these collections are then tiled so that their input regions overlap, to obtain a better representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image

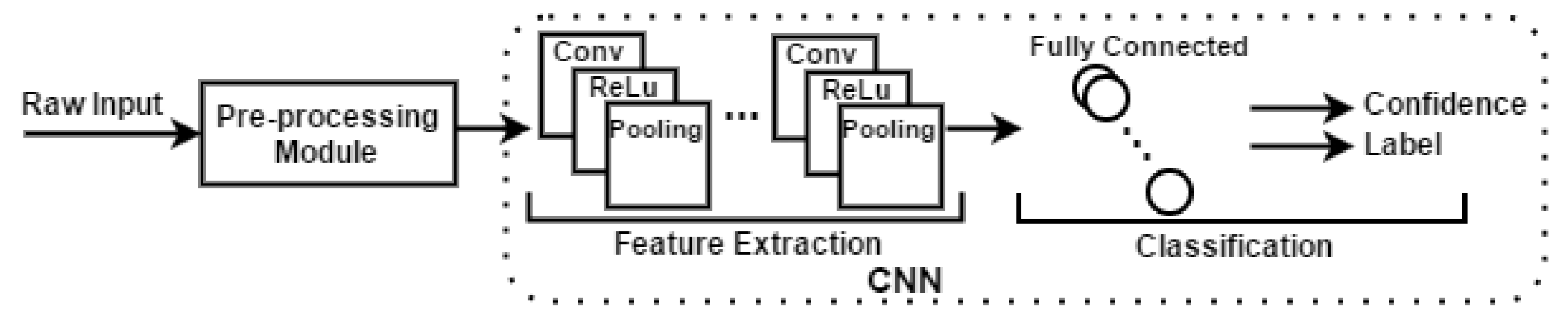


Main types of layers in common Convolutional Neural Networks architectures:
1. **Convolutional Layer**: computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and the region they are connected.
2. **Activation Layer:** applies an elementwise activation function, such as a rectified linear unit (ReLU).
3. **Pooling Layer:** performs a down-sampling operation in spatial dimensions.
4. **Fully-Connected Layer**: exactly the same as layers in regular Neural Networks

Implementation

The specific contributions of this project are:

1. Trained the CIFAR-10 images using a baseline CNN architecture



2. Trained and tested **4** variations of the proven algorithm which is a tiny VGGNet architecture.

Simulation platform: Keras, a highly modular neural networks library, written in **Python** and capable of running on top of **TensorFlow** API.

Dataset: 60,000 32x32 tiny images in 10 classes from **CIFAR – 10** dataset, with 6000 images per class. The training datasets consist of 50,000 images and testing 10,000 images.

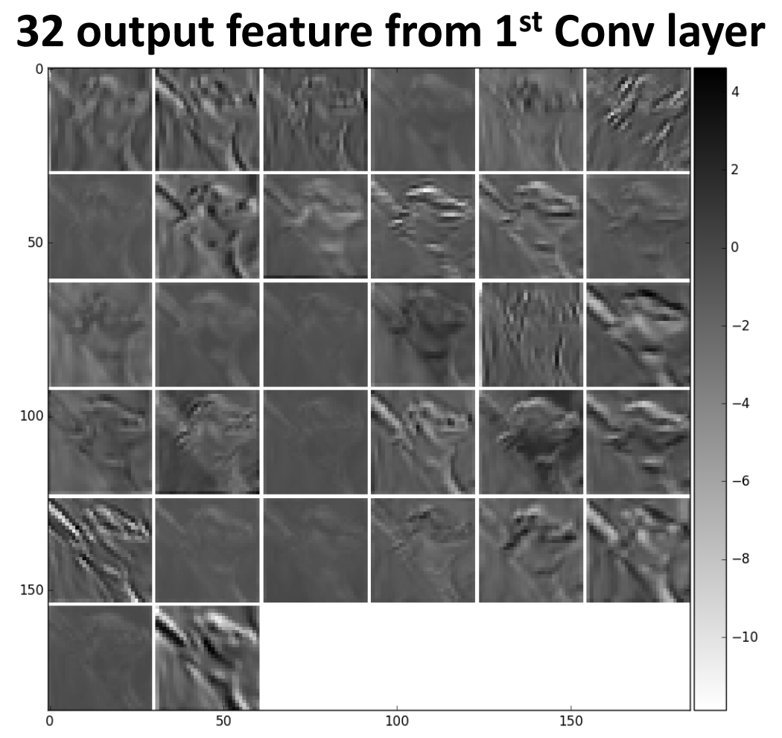
Experiment	Description
No Dropout Layer	Remove three dropout layers
Sigmoid activation	Replace the last activation function to sigmoid
Additional Conv Layers	Add two more convolution layers: Conv3-128
SGD MINI-BATCH SIZE 16	Change the batch size from 32 to 16

Baseline Architecture SGD batch size 32
Data Augmentation
Input Layer (32*32) RGB Images
Conv3-32 activation-relu **Conv3-32** activation-relu maxpool Dropout
Conv3-64 activation-relu **Conv3-64** activation-relu maxpool Dropout
Flatten **Dense-512** activation-relu Dropout **Dense - 10** Activation-Softmax

Visualization of Layers

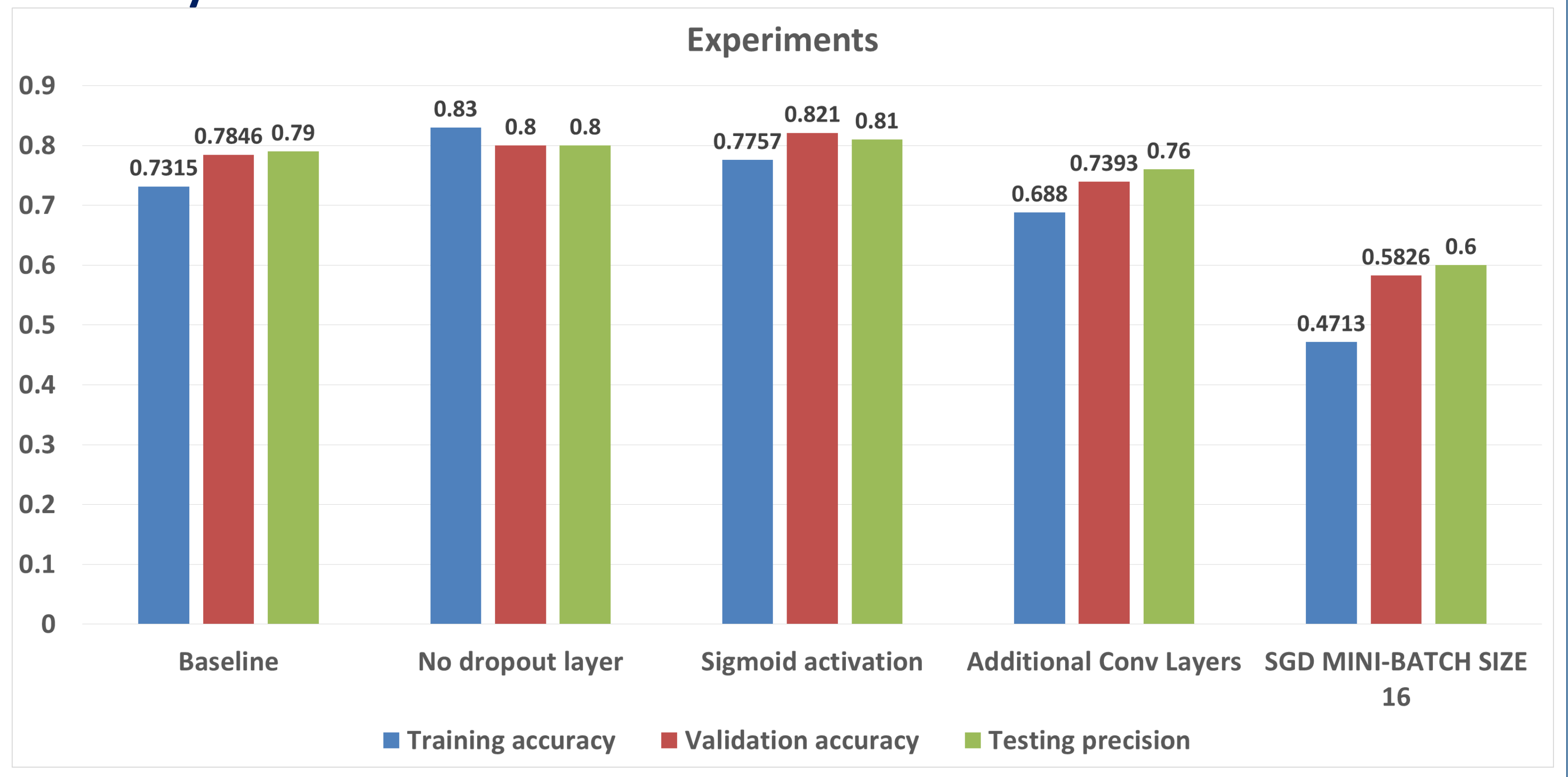
32 different 3x3 shared weights

32 output feature from 1st Conv layer



Results and Discussion

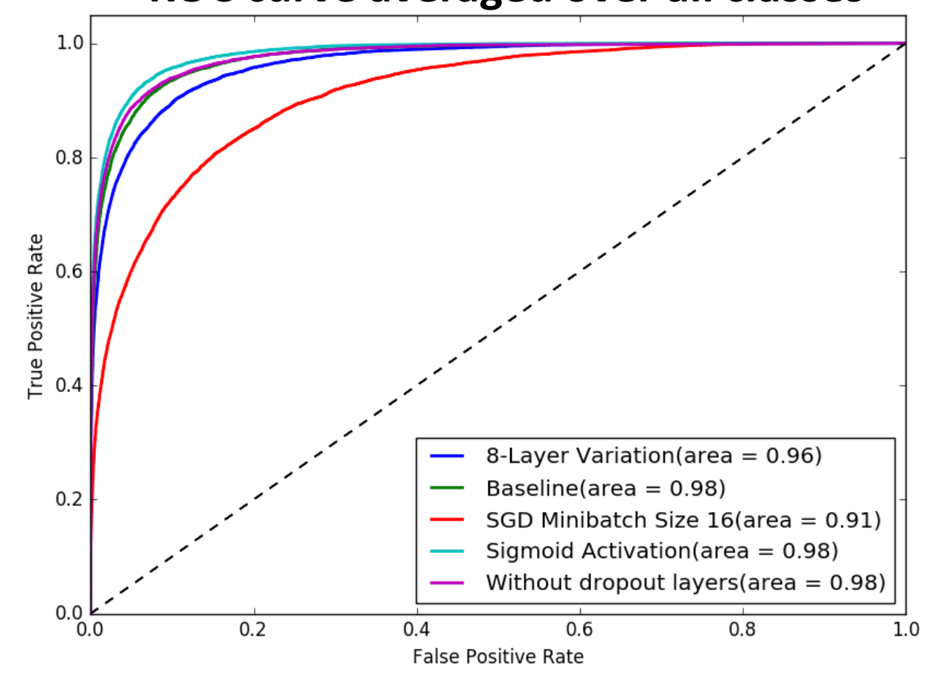
Accuracy



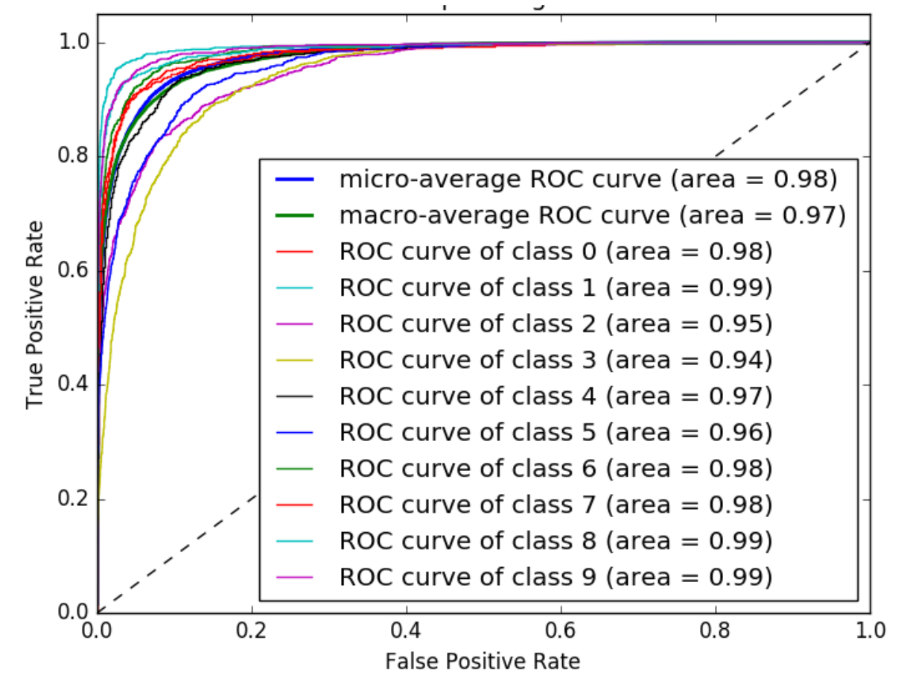
- No dropout layer has higher training accuracy compared with testing and validation. This is because the model without dropout is overfitting. The baseline model has dropout layers included, and hence generalizes well.
- Sigmoid function has output range [0,1] whereas the ReLU has [0, ∞]. Hence sigmoid function can be used to model probability. Applying sigmoid function preserves more output features of the corresponding layer, which could be the reason that makes it achieve better performance. The sharp cutoff of ReLU at zero makes the output more sparse and reduces likelihood of vanishing gradient. The more ReLU layers exist in the model, the more sparse the output representation.
- For more layers and smaller batch size models, the results are not comparable since both of them are not converging at 50 epochs. More number of epochs may lead to a better performance than the baseline model.

ROC

ROC curve averaged over all classes



Multiclass ROC of baseline model

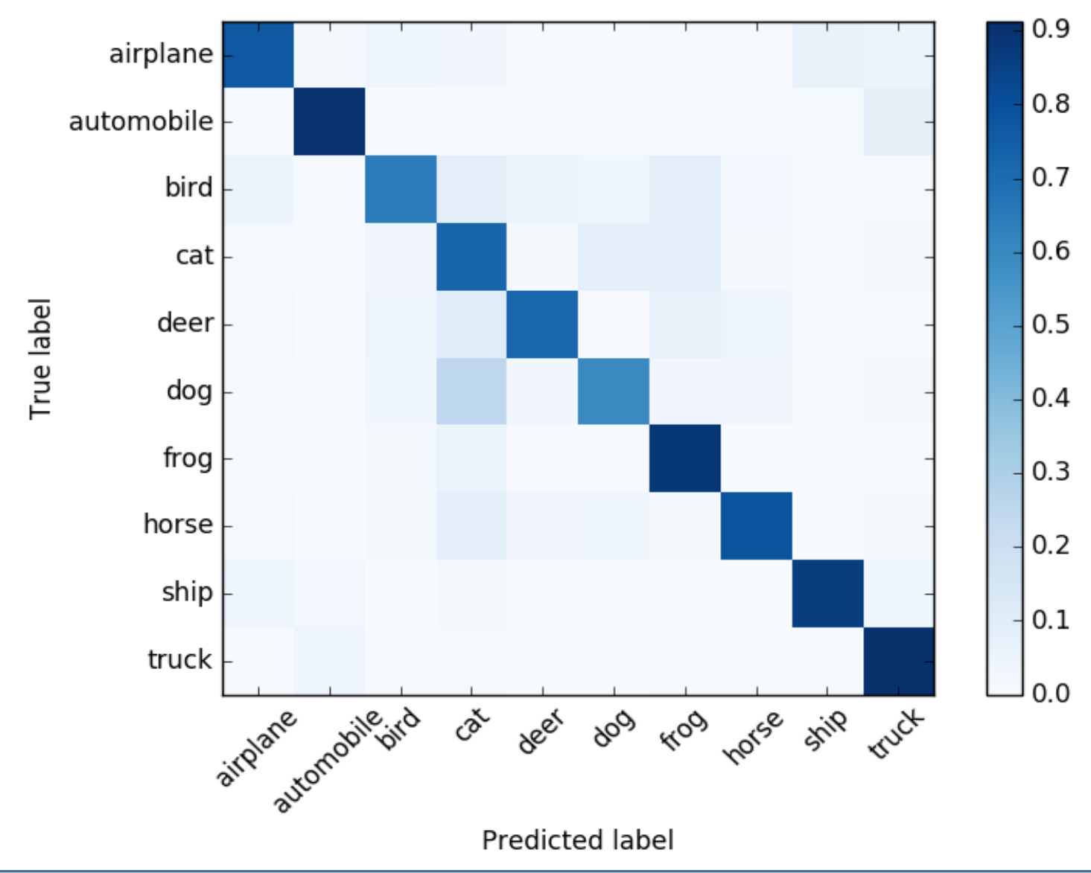


Performance of different models. Sigmoid variation achieves the most area under the curve. The model without dropout is superior to the base line model. SGD minibatch with size 16 seems to have the worst performance, so does the 8-layer model.

ROC curve for each class corresponding to the prediction of the baseline model. It can be seen that the performance is worst for class 3 corresponding to 'cat' classification.

Confusion Matrix

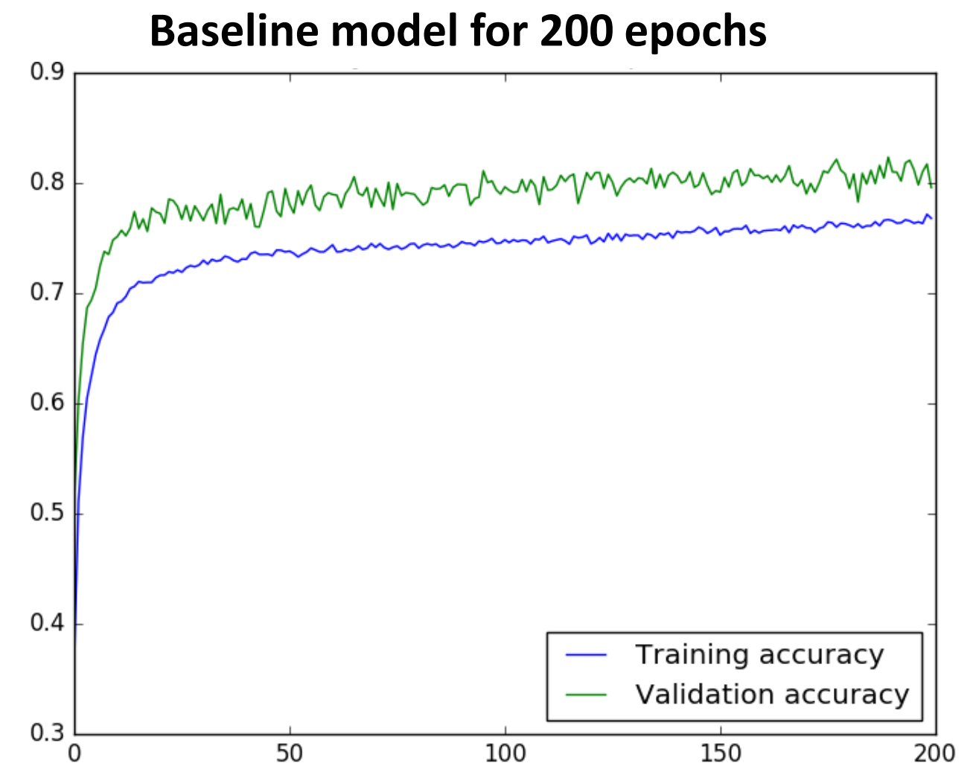
Trucks and automobiles are easier to classify because their feature sets could be smaller, for instance, their curves and edges are more well-defined than animals such as cats.



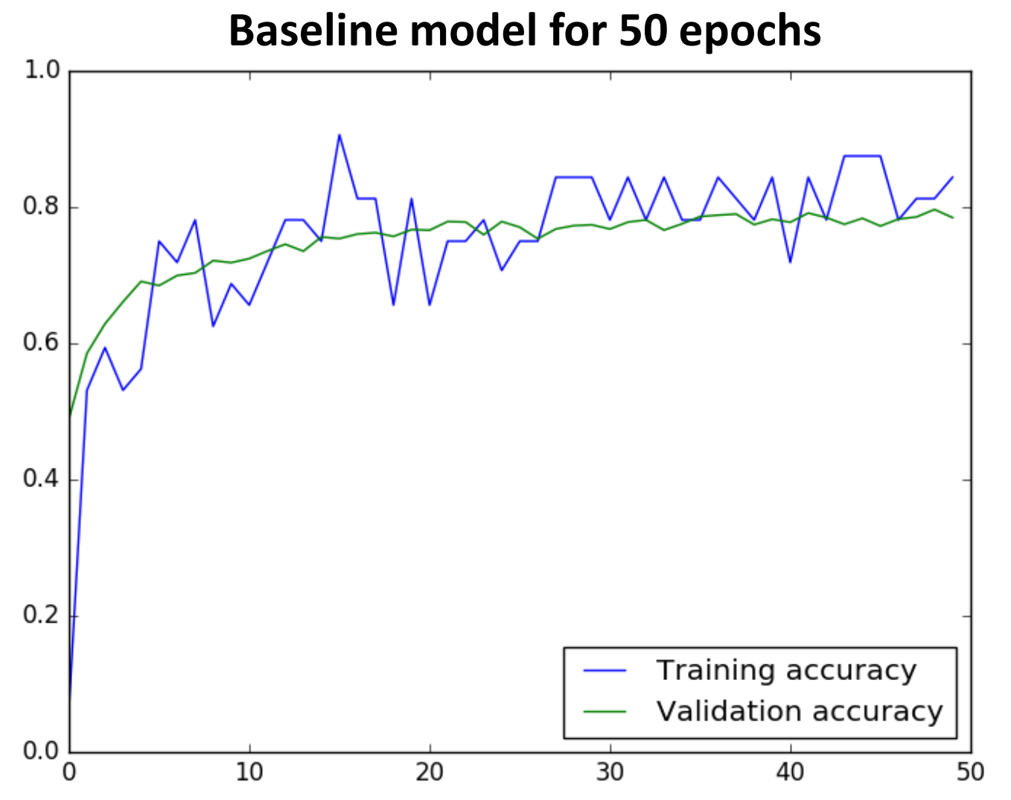
Convergence of SGD

The baseline model converges approximately at 50 epochs. Due to the computational limitation, each experiment was run for 50 epochs. It was found that two of the experiments did not converge for 50 epochs, and their performance metrics are not true reflection of achievable performance of these models.

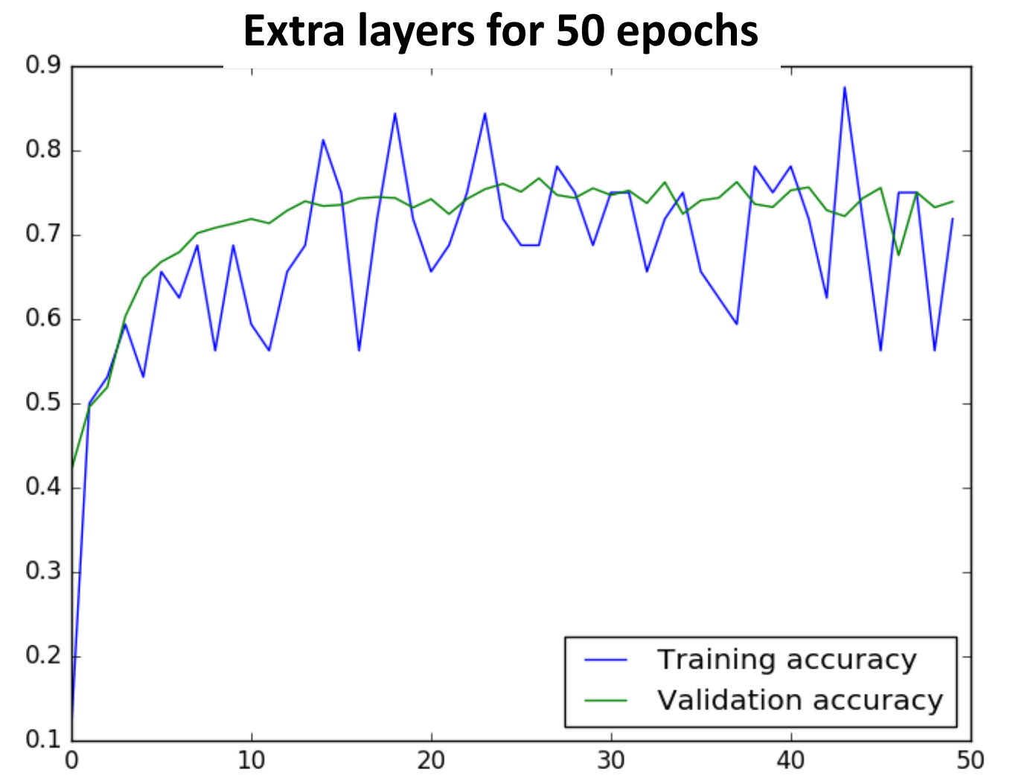
Baseline model for 200 epochs



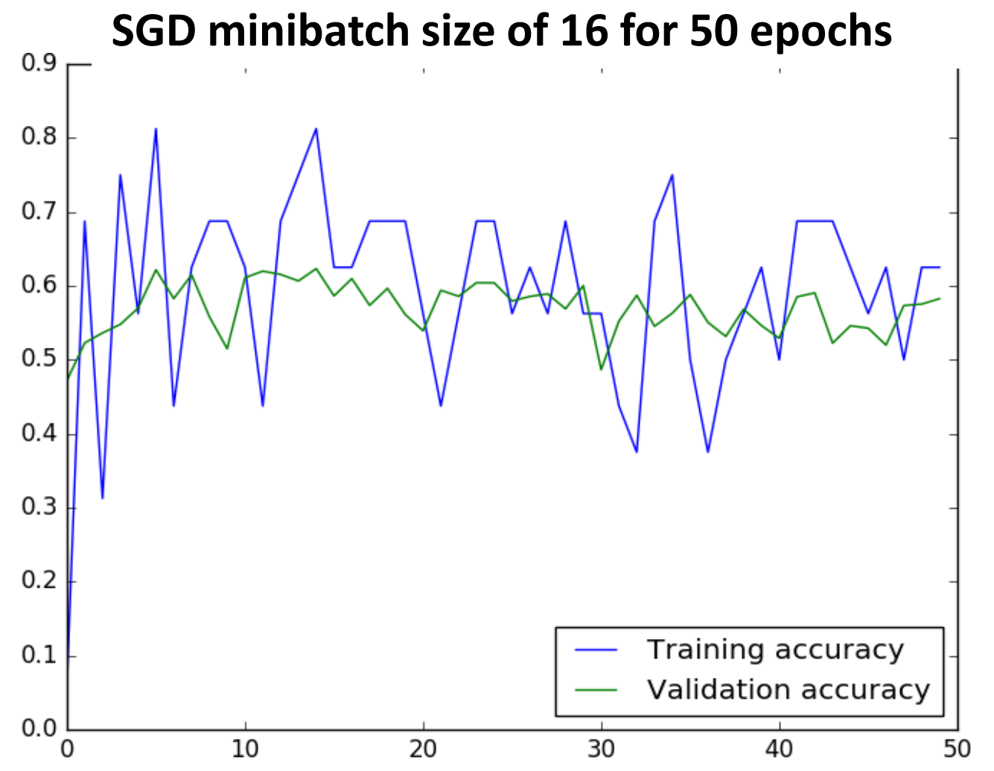
Baseline model for 50 epochs



Extra layers for 50 epochs



SGD minibatch size of 16 for 50 epochs



Conclusions

1. Limitations of computation resources lead to a compromise between variety of architectures that could be experimented.
2. It was demonstrated that the layer depth of baseline model is beneficial for the classification accuracy.
3. Performance of the proven VGGNet architecture on the CIFAR – 10 dataset can be further increased by replacing the last activation layer of ReLU with Sigmoid.
4. Feature extraction for images depends significantly on the complexity of objects in the images.
5. Adding dropout layers leads to regularization of the network.
6. VGGNet CNN architecture can be further extended to other popular datasets.

References

[1] LeCun, Yann, et al. "Gradient-based learning applied to document recognition."Proceedings of the IEEE 86.11 (1998): 2278-2324.

[2] Karpathy, Andre, (2016). CS231n Convolutional Neural Networks for Visual Recognition.

[3] Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton. "The CIFAR-10 dataset." Learning Multiple Layers of Features from Tiny Images. N.p., 8 Apr. 2009.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[5] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).