A Dense Take on Inception for Tiny ImageNet

William Kovacs

Introduction

- Image classification is one of the fundamental problems in computer vision
- Deep learning has been used effectively so solve this problem on the classic ImageNet problem since its introduction in 2012 [1].
- Some notable methods include the use of inception modules to expand the width of the network[2], the use of residuals to provide gradient shortcuts [3], and the use of dense connections to promote feature reuse [4].

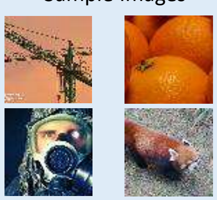
Goal

- To construct a system capable of accurately classifying images using a deep learning framework that is able to utilize the advantages of prior networks
- In particular, inception modules will be connected in a dense fashion in order to construct a wide network that efficiently uses its filters

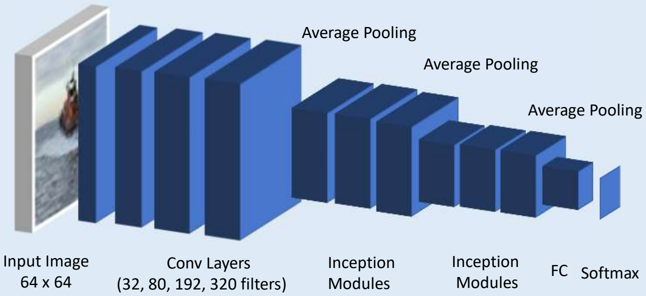
Dataset

- Tiny ImageNet is a miniature version of the commonly used ImageNet, which contains a wide variety of everyday object classes
- 200 classes
- 100,000 training images
- 10,000 validation images
- 10,000 test images
- Preprocessing is used to augment the size by common methods such as random flipping

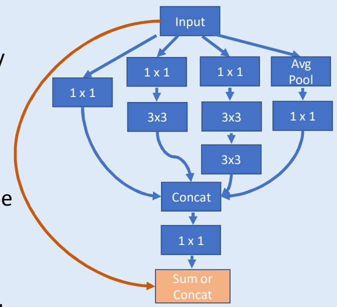
Sample Images



Network Architectures



1. Simple CNN: 5 blocks of 2 two convolutional layers followed by max pooling, and ending with two fully connected layers
2. Inception Resnet: See above figure. Inception modules use a residual connection between the input and output.
3. Inception Densenet: See above figure. Inception modules use concatenation to grow the input between modules of similar dimension, and filter number was reduced by 1/4



A schematic of the inception modules used to increase the width of the network. The main difference between the residual version and the dense version is the summation or concatenation at the end.

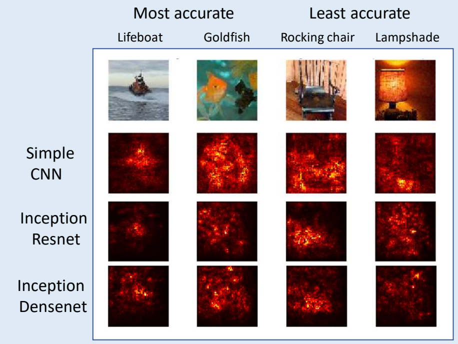
All networks used ReLu activation, batch normalization, L2 regularization, and dropout. RMSProp with a momentum of 0.9, and a learning rate starting at $1x10^{-4}$ that decayed by 0.5 every 5 epochs

Results

Model	Val Err	Val Top-5 Err	Test Err
Basic CNN	0.738	0.511	0.764
Inception Resnet	0.596	0.345	0.612
Inception Densenet	0.586	0.336	0.623

The top-1 and top-5 error metrics are compared between the 3 models. The inception models perform averagely, but the important distinction is that with ¼ of the filters used in the inception dense-net, it was still capable of producing similar results as the Resnet.

Results (Cont.)



A saliency map reflects the importance of a given pixel on the final classification score, and is calculated as the gradient of the correct class's score with respect to the pixel value. For the simple model, the results are noisier showing its lack of focus. The densenet has comparable saliency maps to the resnet, with both demonstrating greater focus, such as on the face of the fish.

Conclusions

- Utilizing the dense paradigm with the network gives similar performance as a non-dense version, while using less filters
- The overall performance is okay, but using this inception architecture should yield lower errors. To improve this, focus should be on looking to better network design, such as utilizing inception blocks after the third pooling, or more robust preprocessing. Current attempts at such an extension have little success, and the time tradeoff could be serious.

References

[1] Krizhevsky, A., Sutskever, I., and Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada.
[2] Szegedy, C., Liu, W., Jia, Y., et al. Going Deeper with Convolutions. (2015) CVPR 2015: Computer Vision and Pattern Recognition. Boston.
[3] He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. (2015) arXiv:1512.03385
[4] Huang, G., Liu, Z., Weinberger, K., and Maaten, L. Densely Connected Convolutional Networks. (2016). arXiv:1608.06993