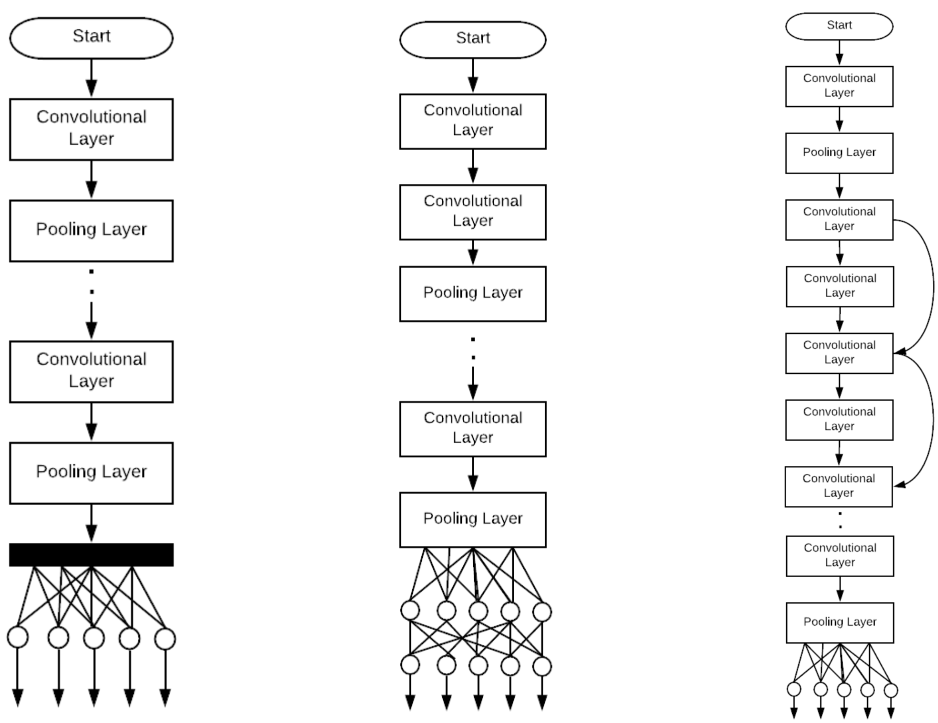
James Hurt
CS 8780
9 May 2018

Comparison of Convolutional Neural Networks for Remote Sensing

1 Technical Description

Remote Sensing is a lucrative application for computational intelligence, with much of the recent research being in Convolutional Neural Networks. Within this discussion of neural networks for remote sensing, Deep Convolutional Neural Networks (DCNN) are the most popular. The popularity of convolutional neural networks can be attributed to the ability of a convolutional network to learn the features of the image, thus eliminating one of the greatest challenges in all of machine learning: feature selection and extraction. While CNN have proven very useful, one of the issues with DCNN is the high training time, even with GPU acceleration. In this project, I attempt to compare a shallow convolutional network with one of the first deep convolutional networks. I then take my comparison one step farther, by looking at the relative cost and benefit of residual neural networks.



(a) Custom Shallow CNN (b) VGG19 (c) ResNet-50

Figure 1: Architectures of the Compared CNN

1.1 Shallow Convolutional Neural Network

The shallow convolutional neural network used in this project combines many techniques of fully connected networks previously used for pattern recognition. The architecture of the network can be see in Figure 1a.

The architecture of the network has a total of 12 layers. The first 10 layers are alternating 2D Convolutional layers layers with a 3x3 kernel size and 2D Max Pooling layers with a 2x2 Pooling Window. After these 10 layers, feature extraction is finished, and a flattening layer is introduced to put the data into a single vector for the final fully connected softmax layer.

1.2 Deep Convolutional Neural Network

The Deep Convolutional Neural Network used for comparison in this project is VGG 19 [1]. The VGG 19 network, as the name suggests, has 19 total layers. This count, however, does not include the pooling layers. The basic architecture can been seen in Figure 1b.

VGG19 has 16 convolutional layers and 4 pooling layers. The interesting quality about VGG19 is that the final pooling layer does not connect directly to the softmax output layer. Instead, the final pooling layer is a 3-layer multilayer perceptron (MLP) [2] The final layer of this MLP is a softmax layer for classification.

1.3 Deep Residual Convolutional Neural Network

The final neural network used for comparison in this project is a Deep Residual Convolutional Neural Network (DRNN). I chose to use ResNet-50 [3], a standard DRNN for comparison developed by Microsoft Research. The architecture of this network can be seen in Figure 1c.

ResNet-50 begins with a single convolutional layer, followed by a single pooling layer, followed by 49 convolutional layers that are residual, meaning that there is a connection from layer N to both layer $N{+}1$ as well as layer $N{+}2$. This connection exists so that the error from the two connections to the $N{+}2$ layer can be summed when performing backpropagation, thus slowing the disappearance of the gradient. The disappearing gradient is one of the greatest challenges of using the gradient descent method.

2 Algorithm Design

3 Experiment Results

4 Analysis

5 Code

References

[1] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Sept. 2014. arXiv: 1409.1556.

[2] J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems, and Evolutionary Computation.* Piscataway, NJ: Wiley-IEEE Press, 1 edition ed., July 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015. arXiv: 1512.03385.