Machine Learning Engineer Nanodegree

Capstone Proposal

Hussein Al-barazanchi
April 9th, 2017

Domain Background

Most of the research on image classification using CNN (Convolutional Neural Networks) concentrate on enhancing the accuracy in the classification task. However, the common CNN architectures need a lot of memory and computation to run and also require moderate storage size. These reasons make CNN very difficult to use on devices with limited memory such as smartphones. Just few research were dealing with the issues of CNN related to the memory and storage requirements. One of the early attempts to compress CNN was by Denton [1]. After that, Han [2] developed a technique known as deep compression. Also, Han [3] presented another method less popular before deep compression. The mentioned methods try to compress CNN models by reducing the floating point operations.

The other direction of research is trying to compress CNN by reducing the number of parameters used in the network. The famous network in this direction is SqueezeNet developed by Andola [4]. Where SqueezeNet minimizes the number of parameters by working the number of convolutional filters and the reception fields and how they are connected. SqueezeNet was able to achieve an accuracy very similar to AlexNet network [5]. The number of parameters used in SqueezeNet is less than AlexNet by 50 times.

I am very excited to work on this fundamental research. The number of applications that could be built on compressed CNN are countless.

Problem Statement

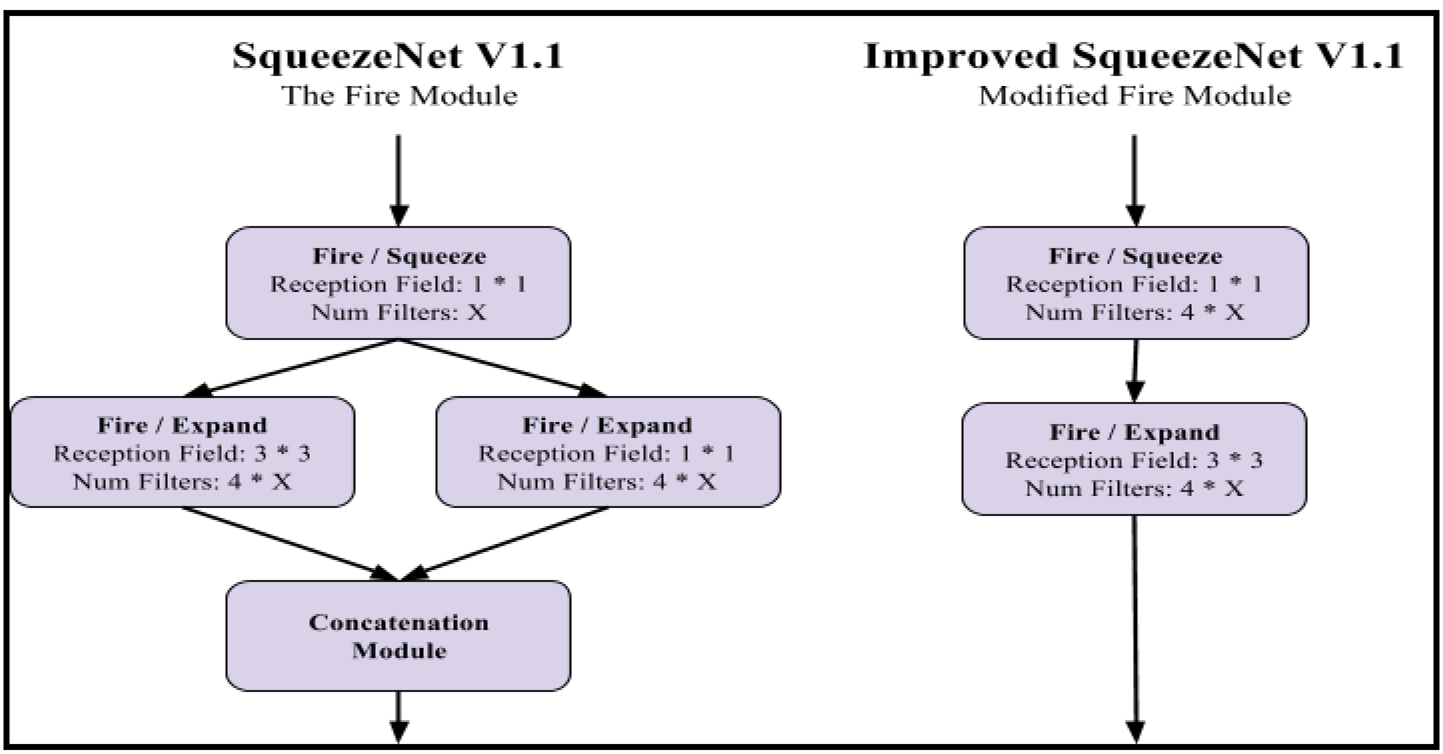
One of the problems with SqueezeNet is time required to train. SqueezeNet is slow to train comparing with the small number of parameters it has. So, I will work on improving the architecture of SqueezeNet to enhance the training time and also to improve the accuracy resulted.

Datasets and Inputs

The datasets that will be used in the experiments are two large scale scene images. The first one is MIT places scene 205 [6]. The Scene 205 contains 205 different scene category with 2.5 million image. Each image is associated with a specific label. The second dataset that will be used is MIT Places2. This dataset contains 365 different category with about 1.8 million images. The images in the two datasets come in different resolutions. However, this will be solved in the preprocessing step as explained in the Theoretical Flow section below . The number of images across labels is not equal so they are not balanced datasets. The division of the data into training and testing will take into consideration the unbalanced nature of the data.

Solution Statement

The fundamental idea behind SqueezeNet is to reduce the number of parameters through architecture design. The intuitive solution is to reduce the number of convolutional layers of reception field 3*3. To implement this technique, they suggested a basic module named Fire. Fire module consists from 3 parts. The first part is Fire/Squeeze which consists from convolutional layers of reception field 1. This layer is followed by two parallel convolutional layers named Fire/Expand. The first layer reception field is 3*3 while the other layer is 1*1. The outputs from these two layers are combined through a module named concatenation. The CNN network will be constructed from the basic Fire module. The figure below show a basic illustration of the Fire module.



Based on my experience of building different types of CNN, the slow training phase is because the use of two parallel convolutional layers in the expand level followed by the concatenation module. So, my suggested solution to this issue is to remove the Fire/Expand layer of reception field 1*1 so there will be no need for the concatenation module. The network will start training faster. However another issue will arise which is the performance will degrade because the number of layer is reduced. To solve this, I will increase the number of filers in the Fire/Squeeze layer.

Benchmark Model

The model will be benchmarked on two large scale image datasets. First, I will train the original SqueezeNet network on the two datasets to get a baseline results. After that, I will train the new suggested network on the same datasets and compare the results.

Evaluation Metrics

As the task is image classification, the evaluation metrics used is accuracy score. Where accuracy is calculated by computing the number of correctly predicted labels to the total number of images. This score will be denoted as Top-1 accuracy. Another measure is Top-5 accuracy. Where accuracy will be computed as the number of labels predicted correctly within the top 5 most likely labels to the total number of images.

Theoretical Workflow

Software and Preprocessing.

1. The used framework is Nvidia Digits
2. The used underlying deep learning library is Caffe
3. All images will be resized to 256 * 256
4. The mean pixel for each color channel will be subtracted from all images
5. 80% of the images will be used for training and 20% for testing
6. The division of training and testing will be as a percentage in regard to the number of images per label
7. All images will be cropped randomly to be 227 * 227

Training and Testing

1. I will use the SqueezeNet V1.1 network provided by its author from this repository [8].
2. I will train it on the MIT Places205 and MIT Places2
3. The top-1 and top-5 accuracies will be reported.
4. I will use my suggested modifications mentioned in the solution statement above
5. It is very expensive (limited budget) to try many hyper-parameters configurations. So, I will conduct two experiments on each dataset using the new architecture as follow:

    A. Use the same hyper parameters provided by SqueezeNet V1.1.
    B. Increase the number of filters in Fire/Squeeze to be Equal to the number of filters in Fire/Expand.
6. The top-1 and top-5 accuracies will be reported and compared with the original networks.
7. A summary of advantages and disadvantages of the new architecture will be presented.

References

1. Denton, Emily L., et al. "Exploiting linear structure within convolutional networks for efficient evaluation." Advances in Neural Information Processing Systems. 2014.
2. Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).
3. Han, Song, et al. "Learning both weights and connections for efficient neural network." Advances in Neural Information Processing Systems. 2015.
4. Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." arXiv preprint arXiv:1602.07360 (2016).
5. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
6. Zhou, Bolei, et al. "Learning deep features for scene recognition using places database." Advances in neural information processing systems. 2014.
7. Zhou, Bolei, et al. "Places: An image database for deep scene understanding." arXiv preprint arXiv:1610.02055 (2016).
8. Retrieved from Github by April 9, 2017, https://github.com/DeepScale/SqueezeNet.