Machine Learning Engineer Nanodegree

Capstone Proposal

Manouchehr Bagheri
June 12, 2017

Proposal

Convolutional Neural Networks - Optimum Max-Pooling Design

Domain Background

Deep learning have demonstrated impressive results on a number of computer vision and natural language processing problems. At present, state-of-the-art results in image classification (Simonyan & Zisserman (2015) [1]; Szegedy et al. (2015) [2]) and speech recognition (Sercu et al. (2015) [3]), etc., have been achieved with very deep (>16 layer) CNNs. Thin deep nets are of particular interest, since they are accurate and at the same inference-time efficient (Romero et al. (2015) [4]).

The vast majority of modern convolutional neural networks (CNNs) used for object recognition are built using the same principles: They use alternating convolution and max-pooling layers followed by a small number of fully connected layers (e.g. Jarrett et al. (2009) [5]; Krizhevsky et al. (2012) [6]; Since the spatial pooling methods like max-pooling causes limited performance due to the rapid reduction in spatial size, we want to try the fractional version of max-pooling(Graham et al. (2015) [7]) and no pooling method studied by (Springenberg et al. (2015) [8]) on our simple CIFAR-10 image classification project and re-evaluate the results.

Problem Statement

The most common object recognition Convolutional Neural Networks methods are applying the same method: Alternating convolution and max-pooling layers followed by a small number of fully connected layers. The max-pooling act on the hidden layers of the network, reducing their size by an integer multiplicative factor (mostly $\alpha = 2$). This will help simplifying the network and reducing the process time, but its byproduct is discarding 75% of the data that causes a degree of invariance with respect to translations and elastic distortions [7]. It seems that there is a tradeoff problem between the performances of CNNs basic pipeline and process complicity and time. Finding an optimal way of applying spatial pooling or replacing it, is the problem that we try to address in this project.

Datasets and Inputs

This project will use the well-known object recognition dataset, CIFAR-10. It consists of 60000 32x32 RGB color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images [9].

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.

Solution Statement

To address the tradeoff between the performance limitation due to utilization of the max-pooling and the process complicity, this project is looking for an optimal way of max-pooling in Convolutional neural networks. We will apply two alternative studied methods of fractional version of max-pooling [7] and none-pooling [8] on an applied regular CNN method for image classification.

We re-evaluate the three methods and compare their accuracy for a small project of image classification on the CIFAR-10 dataset. We use TensorFlow to build the deep learning networks for each of the above mentioned three methods.

Benchmark Model

Graham reports that his method of Fractional Max-Pooling (FMP) with pseudorandom overlapping pooling network obtained test errors of 4.50% (1 test), 3.67% (12 tests) and 3.47% (100 tests) on CIFAR-10 dataset. Compare to the recent Kaggle competition with a test error of 4.47% using regular CNN methods [7]. Springenberg et al also reported that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks [8].

Evaluation Metrics

In this project we are looking for a comparable improvement in the results of the accuracies obtained from each model. The evaluation metric for the model will be TensorFlow accuracy function on the test images in the CIFAR-10 dataset. It calculates how often predictions matches labels. The accuracy function creates two local variables, total and count that are used to compute the frequency with which predictions match the labels.

Accuracy is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.

$$accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

Since CIFAR-10 dataset contains 60,000 32x32 RGB images, which are divided into 10 classes it is a class balanced classifier. Accuracy is a good metric for class balanced classifiers.

Loss:

The usual method for training a network to perform N-way classification is multinomial logistic regression, aka. Softmax regression. Softmax regression applies a softmax nonlinearity to the output of the network and calculates the cross-entropy between the normalized predictions and a 1-hot encoding of the label. For regularization, we also apply the usual weight decay losses to all learned variables.

Project Design

I will duplicate two studied methods on an applied regular CNN method for image classification on CIFAR-10 dataset through following steps:

1- Applying CNNs by incorporating a spatial max-pooling of the $\alpha \times \alpha$ form with $\alpha = \{2, 3\}$.
2- Applying CNNs by incorporating a fractional version of max-pooling [7] where is allowed to take non-integer values between $1 < \alpha < 3$.
3- Applying CNNs by replacing the max-pooling layers by convolutional layers with increased stride 2 [8] where $\alpha = 1$.

Architectures of these networks are described in a table for comparison. We will use the same number of Convolution and Max Pool layers for each model but with different suggested α values and compare their TensorFelow accuracy levels.

REFERENCES

1 Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale visual recognition. In Proceedings of ICLR, May 2015. URL http://arxiv.org/abs/1409.1556.

2 Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In CVPR 2015, 2015. URL http://arxiv.org/abs/1409.4842.

3 Sercu, T., Puhrsch, C., Kingsbury, B., and LeCun, Y. Very Deep Multilingual Convolutional Neural Networks for LVCSR. ArXiv e-prints, September 2015. URL http://arxiv.org/abs/1509/08967.

4 Romero, Adriana, Ballas, Nicolas, Kahou, Samira Ebrahimi, Chassang, Antoine, Gatta, Carlo, and

Bengio, Yoshua. Fitnets: Hints for thin deep nets. In Proceedings of ICLR, May 2015. URL http://arxiv.org/abs/1412.6550.

5 Jarrett, Kevin, Kavukcuoglu, Koray, Ranzato, Marc'Aurelio, and LeCun, Yann. What is the best multi-stage architecture for object recognition? In ICCV, 2009.

6 Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In NIPS, pp. 1106–1114, 2012.

7 Graham, Benjamin: Fractional Max-Pooling, May 2015. URL https://arxiv.org/abs/1412.6071

8 Jost Tobias Springenberg, Alexey Dosovitskiy———————————————, Thomas Brox, and Martin Riedmiller: STRIVING FOR SIMPLICITY: THE ALL CONVOLUTIONAL NET, April 2015. URL https://arxiv.org/abs/1412.6806

9 "The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton: https://www.cs.toronto.edu/~kriz/cifar.html