ECE 539: Final Project
Fine-tune Alexnet and VGG-16

Feng Rong (fr214)
feng.rong@rutgers.edu

Abstract

In this project, I fine-tuned two pretrained networks, Alexnet and VGG16, using the LFW(Labeled Faces in the Wild) dataset. I plot the ROC curve of each step and compared the performances of the network before and after fine-tune and see the ability of classification of the networks.
For this report, I introduced the architecture of CNN, Alexnet and VGG-16 in section one. And in section two, I introduced the dataset I used to fine-tune the two network, as well as the detailed set-up of the fine-tune process. As for section three, I show the ROC curve and discussed the result.

1 Introduction

1.1 Convolutional Neural Network

In machine learning, a convolutional neural network is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery, which were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers.

- Convolutional Layer
  Assume the input is a 32*32*3 array of pixel values, we have a filter which is an array of numbers (the numbers are called weights). As the filter is convolving around the input image, it is multiplying the values in the filter with the original pixel value of the image. These multiplications are all summed up and become a single number. Repeat this process for every location on the input volume. After sliding the filter over all the locations, we get a smaller array of numbers which we called a feature map. By using more filters, we are able to preserve the spatial dimensions better. The filters in the first conv layer are designed to detect low level features. When go through another conv layer, the output of the first conv layer becomes the input of the 2nd conv layer which is a activation map, the output will be activations that represent higher level features. As we go through more conv layers, the activation maps represent more complex features.

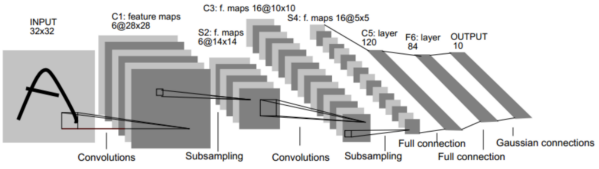

Figure 1: CNN

- ReLU Layer
  After each conv layer, there usually is a nonlinear layer immediately afterward. This layer introduce nonlinearity to a system that basically has just been computing linear operation during conv layer. ReLU layers work far better than sigmoid and tanh because the network is able to train a lot faster without making a significant difference to the accuracy. It also helps to alleviate the vanishing

gradient problem, which is the issue where the lower layer of the network train veru slowly because the gradient decreases exponentially through the layer. The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input volume. In basic terms, this layer just changes all the negative activations to 0. This layer increases the nonlinear properties of the model and the overall network without affecting the receptive fields of the conv layer.

- Pooling Layer
  After some ReLU layers, we may choose to apply pooling layer. This serves two main purposes. The first is that the amount of parameters or weights is reduced by 75 per, thus lessening the computation cost. The second is that it will control overfitting. Maxpooling is the most popular one. It takes a filter and a stride of the same length and then applies to the input volume and outputs the maximum number in every subregion that filter convolves around.

- Fully Connected Layer
  The fully connected layer looks at the output of the previous layer and determines which features most correlate to a particular class. In other words, FC layer looks at what high level features most strongly correlate to a particular class and has particular weights so that when we compute the products between the weights and the previous layer, we get the correct probabilities for the different class.

1.2 Alexnet

Alexnet, designed by the SuperVision group, consisting of Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever, is a convolutional neural network, which won the 2012 ImageNet LSVRC-2012 competition by a large margin (15.3% VS 26.2% (second place) error rates).

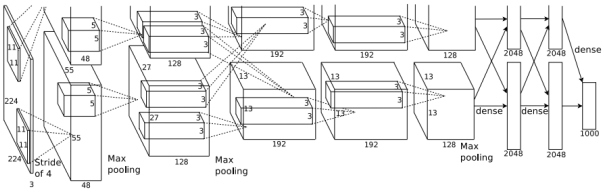

Figure 2: Alexnet

Alexnet contains eight learned layers: five convolutional layers and three fully-connected layers. Relu is applied after every convolutional and fully connected layer. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. This network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the log-probability of the correct label under the prediction distribution. The network has 62.3 million parameters, and needs 1.1 billion computation units in a forward pass. We can also see convolution layers, which accounts for 6% of all the parameters, consumes 95% of the computation.
The first concolutional layer filters the 224*224*3 input image with 96 kernels of size 11*11*3 with a stride of 4 pixels. The second convolutional layer takes as input the output of the first convolutional layer and filters it with 256 kernels of size 5*5*48. The third, fourth, and fifth convolutional layers are connected to one another without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size 3*3*256 connected to the (normalized, pooled) outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size 3*3*192 , and the fifth convolutional layer has 256 kernels of size 3*3*192. The fully-connected layers have 4096 neurons each.
Alexnet uses ReLu for non-linear part, instead of a Tanh or Sigmoid function which was the earlier standard for traditional neural network. ReLu is given by

$$f(x) = max(0, x)$$

The advantage of the ReLu over sigmoid is that it trains much faster than the latter, since the derivative of sigmoid becomes very small in the saturating region and therefore the updates to the weights almost vanish which is called vanishing gradient problem.
Another problem that this architecture solved was reducing the over-fitting by using a Dropout layer after every FC layer. Dropout layer has a probability,(p), associated with it and is applied at every neuron of the response map separately. It randomly switches off the activation with the probability p.

1.3 VGG-16

VGG-16 is the a ConveNet model of the 16-layer network used by the VGG team in the ILSVRC-2014 competition. It takes input RGB image of size 224*244*3. It makes the improvement over AlexNet by replacing large kernel-sized filters(11 and 5 in the first and second convolutional layer, respectively) with multiple 3X3 kernel-sized filters one after another. With a given receptive field(the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost. The image is passed through a stack of convolutional layers, where it use filters with a very small receptive field: 3*3. In one of the configurations it also utilise 1*1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3 3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2 2 pixel window, with stride 2.
The VGG convolutional layers are followed by 3 fully connected layers. The width of the network starts at a small value of 64 and increases by a factor of 2 after every sub-sampling/pooling layer. It achieves the top-5 accuracy of 92.3 % on ImageNet.

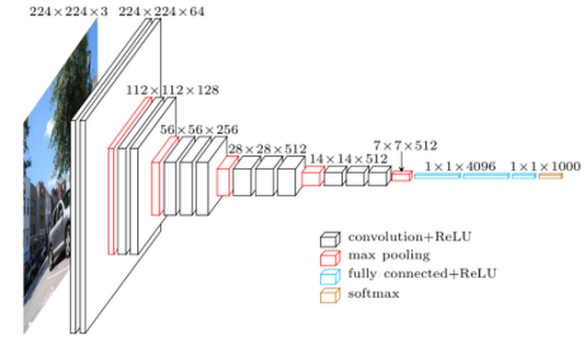

Figure 3: VGG-16

1.4 The Dataset

Both networks are pretrained using ImageNet. ImageNet is an image dataset organized according to the WordNet hierarchy, which is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labeled by human labelers using Amazon's Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.

2 Fine-tune

To fine-tune Alexnet and VGG-16, I used tensorflow as working environment.
The source code is included in the project file that I uploaded.

2.1 The Dataset

The dataset I used to fine-tune the network is Labeled Faces in the Wild(LFW), which is a database of face photographs designed for studying the problem of unconstrained face recognition. The dataset contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. There are total 5479 people and 1680 of the people pictured have two or more distinct photos in the data set.
To fine-tune the network, I divided the LFW dataset into training set and validation set. I randomly choose 75% of the dataset as training set, and rest of the picture as validation set.

2.2 Alexnet

To fine-tune the alexnet, I loaded the pretrained weight and trained the final three layers, which are fc6, fc7 and fc8 with the given dataset. To extract features, I pooled out the results of fc7. I compared the features of the image pairs using cosin similarity to decide weather the given image pairs are belongs to the same class, and get the ROC curve as well. The learning rate I set for alexnet is 0.002 and I trained it 80 times.

2.3 VGG-16

To fine-tune the VGG-16, I also loaded the pre-trained weight and trained the final three layers, which are fc6, fc7 and fc8 with the given dataset. To extract features, I also pooled out the results of fc7. I compared the features of the image pairs using cosin similarity to decide weather the given image pairs are belongs to the same class, and get the ROC curve as well. The learning rate I set for VGG-16 is 0.002 and I trained it 45 times.

3 Results Analysis

For this part, I trained the two networks using the setting I mentioned in the previous section, and compared the result.
The results are shown by ROC curve, which is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. ROC gives two bits of relative information for each threshold.
For the ROC curve, the testing set contains 1000 pairs of data. Half of them belongs to the same class while others not.

3.1 Alexnet

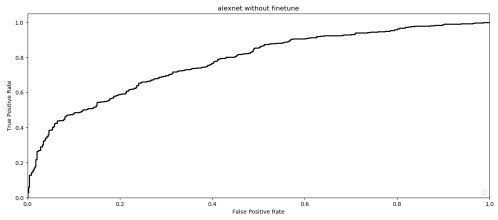

Figure 4: Alexnet without Fine-tune

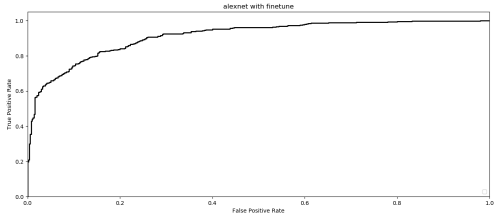

Figure 5: Alexnet without Fine-tune

To fine-tune the Alexnet, I set the learning rate to 0.002 and I trained it 80 times.
From the ROC curve we can see that the original Alexnet has the ability of recognizing different class, however, the accuracy is not high enough. After fine-tune, it can tell that the accuracy has improved.

3.2 VGG-16

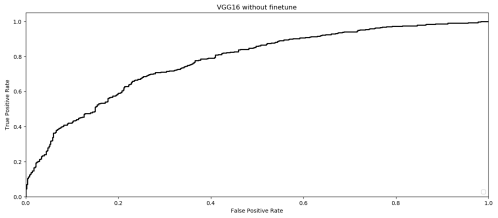

Figure 6: VGG-16 without Fine-tune

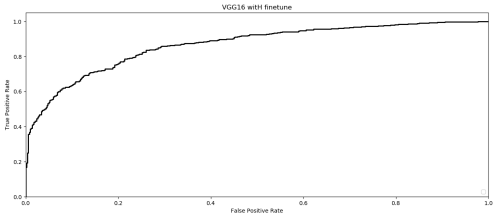

Figure 7: VGG-16 with Fine-tune

To fine-tune VGG-16, I set the learning rate to 0.002 and I trained it 45 times.
The original VGG-16 also has the ability of recognizing different class, although, it's accuracy seems better than Alexnet, it still can be better. After fine-tune, it can tell from the pictures that the accuracy has improved. However, the accuracy is not as good as I expect. From what I have learned, VGG-16 is supposed to have a better performance than the Alexnet, but the results I got did not show that. This may be because that I only trained the final three layers of the network and the pretrained process uses a different dataset. I do not have enough time to train the whole network.

4 Conclusion

From the results in the previous section, we can see that after fine-tune using the LFW dataset, the accuracy of both network have been improved. However, due to the time limit, the results are not as good as I expected, as I am not able to train the whole network. A huge improvement can still be seen from the results I achieve.
For future work, I want to train more layers of the network to see if it can achieve better performance.

5 Acknowledgement

This work would not have been possible without the help and support of Professor Vishal M. Patel and Professor Peter Meer.

6 Reference

1. CNNWikipedia:https://en.wikipedia.org/wiki/Convolutional_neural_network
2. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks
3. Karen Simonyan, Andrew Zisserman: VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION
4. VGG-16:http://cv-tricks.com/cnn/understand-resnet\-alexnet-vgg-inception/
5. Labeled Faces in the Wild:http://vis-www.cs.umass.edu/lfw/
6. CNN Architecture:http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf
7. VGG-16: https://www.quora.com/What-is-the-VGG-neural-network