Best Practices for Neural Network Applied to Image Semantic Segmentation

Teng Ma

The Australian National University
Canberra ACT 2600 Australia

U6123792@anu.edu.au

Abstract. There are many aspects can determine the performance of a specific neural network. In this paper, I will mainly discuss the influence caused by the dataset scale in image semantic segmentation field. I used five common networks and two kinds of datasets to conduct the experiment. The result shows that the different neural network performed many differences in different dataset. And I gave my opinion about how to choose the best practice for neural network applied to specific dataset scale.

Keywords. Neural Networks, Performance Improvement, Dataset Scale, Semantic Segmentation

1. Introduction

Nowadays, semantic segmentation has been applied to many fields, such as image classifier, image recognition, even autonomous driving systems. (Garcia-Garcia, et al, 2017). The idea of semantic segmentation is recognizing and understanding what's in the image in pixel level. And there are two main aspects for semantic segmentation, one is accuracy, the other is efficiency, because this technology is always used in real-time operations. Recently, convolutional neural networks (CNN) have achieved a great success in this field and is benefits from the large public image repositories, such as ImageNet, and high-performance hardware, such as GPUs (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Sermanet et al., 2014; Simonyan & Zisserman, 2014). And there are many mature networks come from CNN, such as AlexNet and VGG. These two networks achieved excellent results in the ILSVRC-2012 and ILSVRC-2013, respectively (Garcia-Garcia, et al, 2017). Moreover, recurrent neural networks (RNN) also made some achievements in generating image description. RNN works well on sequential processing in absence of sequences. Sometimes it is rare that having sequences as the inputs and outputs, but some images, such as facial detection, that we tackled are fixed vectors and we can process them by using this powerful formalism in a sequential method. Therefore, we can get many methods in image processing, and we should determine the best way to conduct it, that is exploring the network which could perform well in both two aspects, accuracy and time cost.

There are two kinds of datasets used in this experiment. MNIST database is the dataset for handwritten digits, distributed by Yann Lecun. It is a famous dataset for image classification and recognition. The dataset consists of pairs, handwritten digit image and label. The digit image is a series of gray scale image with size 28x28 pixel, and the label is the actual number corresponding the image pattern. I used it as simple and small amount of dataset in this experiment. The other dataset I used is from ImageNet. There are over 1000 kinds of categories in the ImageNet database. And I chose the category of cats and tigers, because they are similar and can be used to do transfer learning. I used the ImageNet dataset as complicated and large amount of dataset in this experiment. Thus, we can evaluate the results in both horizontal comparison and vertical comparison.

In this experiment, our aims are twofold. We aim to improve the performance of neural network in MNIST. In the experiments conducted by LeCun et al in 1998, the accuracy result is 91.6% with the linear classifier (1-layer NN). And we will apply the dataset to 4 modern neural network and hope to

get better results and compare the time cost. We also aim to determine what situation is the best practice for the specific neural network.

The remainder of this paper is organized as follows. It describes and detailly explains the methods I used in this experiment in second section. Next, Section 3 shows the experiments results and proposes discussions. At last, Section 4 summaries the paper, draws conclusions and raises the future work on this topic.

2. Methods

As we previously mentioned, we applied four kinds of neural network, CNN, RNN, AlexNet and VGGnet on the two datasets respectively and will introduce the experiment in detail. And as the hardware limited, I only use CPU to conduct the experiment and without GPUs.

2.1 Data pre-processing

The dataset MNIST has been separated into training set and test set, so we did not do any pre-processing on it. We mixed the categories of cats and tigers and did 10 cross-validation on the dataset. We want to classify the image of cats from the entire dataset, so we added the tiger's image as negative samples. And we used the API from scikit-learn to do the cross-validation.

2.2 CNN

The first neural network we choose is CNN. In this experiment, we used the convolutional neural network with 2 convolution layers, 2 pooling layers, 1 fully-connected layer and 1 output layer. We used 2 convolution layers to extract the feature map of the images for each 5*5 patch. And after convolution, the image size has been 7*7, and we added a fully-connected layer with 1024 neurons to process all the images. Then, we reshape the output of pooling layer to a 1-dimensional vector and process with weights and biases, and executes the ReLU function. And we use dropout to avoid overfitting, and the probability of keeping is set to 0.5 as training. The dropout will not be used while testing. Finally, we got the distribution by using the softmax layer. Moreover, we recorded the accuracy and time cost to do the comparing work.

2.3 RNN

RNN is always used in natural language processing or machine translation because it has more advantages in process the sequential dataset. But it also can be used in image recognizing field, such as image captioning, which can add the description to a specific image. When we used RNN in image processing, it may be slower than CNN, but it can save many memory spaces.

In this experiment, we also used RNN to do the image semantic segmentation. We let RNN read every line of the image from the first to the last, and make the classification. First, we set some hyper-parameters, such as learning rate, iteration times and batch sizes. The architecture of this recurrent neural network consists of three components, input layer, lstm cell and output layer. We used the basic LSTM cell to build the network, and we also added dropout to avoid the overfitting while training.

2.4 AlexNet

AlexNet was proposed by the SuperVision group and won the championship of ILSVRC-2012 by achieving the test accuracy of 84.6%, more than 10.8% ahead of the traditional architecture in the same challenge (Garcia-Garcia, et al, 2017). The AlexNet's architecture was quite simple (Krizhevsky, et al, 2012). The network consists of 8 weight layers, including 5 convolutional layers and 3 fully-connected layers. The first and second convolutional layer was followed by LRN layers, every LRN layer and the last convolutional layer was followed by the max pooling layers, and every layer connected to the RELU activation functions. After fully-connected layer, Dropout will solve over-fitting issues. The figure 1 below shows the architecture of AlexNet.

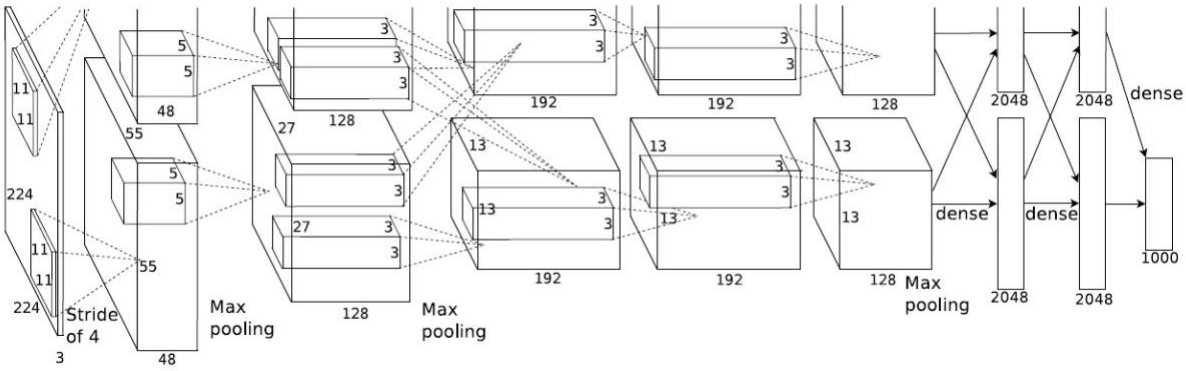

Figure 1: AlexNet architecture. (Krizhevsky, et al, 2012)

In this experiment we used the AlexNet as a representation for multiple-layer neural network. In AlexNet, we define a norm function to normalize the pooling results. And there are 3 convolution layers with max pooling layer. After convolution, we deform the outputs and then connect three fully-connected layers. Then, we get the output.

2.5 VGGnet

VGGnet was proposed by Visual Geometry Group(VGG) from Oxford University, and it is a deep convolutional neural network. VGGnet explored the relationship between network's depth and performance, and it successfully constructed deep convolutional network in 16~19 layers by constantly heaping up 3*3 convolution kernel and 2*2 max pooling layer (Simonyan & Zisserman, 2014). The Figure 2 shows the architecture of VGGnet.

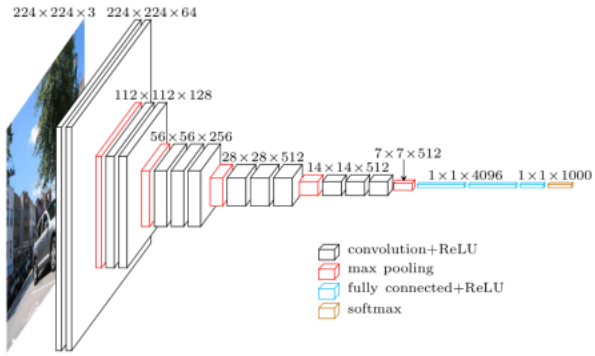

Figure 2: VGGnet architecture layer (Simonyan & Zisserman, 2014).

In this experiment we used VGGnet as another representation for multiple-layer neural network. We built the VGGnet as the figure presented above. It has 5 groups of convolutions, 2 fully-connected layers for image features and 1fully-connected layer for classifier feature and uses softmax layer to output the results.

3. Results and Discussion

This part we will compares the results of five neural networks, and how they perform in two kinds of dataset.

For MNIST, we used the batch size as 100 for all networks. We controlled that the learning rate for RNN, AlexNet and VGG are same, 0.001 and the dropouts are also same, 0.5. In this situation, we can compare the performance of the neural network instead of influencing by the hyper-parameters.

The results are shown below.

MNIST:

	1-layer NN	CNN	RNN	AlexNet	VGGnet
Accuracy	91.94%	97.63%	96.09%	99.21%	96.87%
Time Cost	49.187s	264.966s	382.686s	893.263s	1292.122s

For the ImageNet, we only got the results from AlexNet and VGGnet, because the performances of other three are unacceptable. And we also used transfer learning to improve the performance of VGGnet. We used the pre-trained model provided by Machrisaa (2017). This model has trained in the 1000 categories in ImageNet, and the parameters have been trained already. The transfer learning made it possible for us to conduct the experiments on the machines with poor hardware and can also save much time.

The results are shown below.

ImageNet:

	1-layer NN	CNN	RNN	AlexNet	VGGnet	Transfer Learning
Accuracy	-	-	-	73.7%	83.9%	84.7%
Time Cost	-	-	-	6 hours	8 hours	43min

From the results above, we can have some discussions.

First, the accuracy is really improved by the more complicated neural network. But the time cost is also increased, and we can see that the more layers the network has, it will need more time to train the neural network itself. Sometimes, the time cost is acceptable, but in real life we are limited by the efficiency. And in the networks above, the RNN is the special one, because when we train the networks, it cost much less memory than others, and the process of training was accelerating. Although RNN may not be good at image processing, it has shown the advantages.

Second, we can see that the VGGnet did not performs better than the simple CNN networks, and it cost more than 4 times time. The reason may be that not all the multiple-layer networks suit the small dataset scale. Although the dropout layer may reduce influence of overfitting problem, the large scale of network could not be trained perfectly by the small dataset. That is one of the reasons, why the accuracy is less than CNN. But on the other hand, the multiple-layer network could do more than small scale networks, such as processing the large image dataset.

Third, transfer learning is really a good assistance for us to do the image processing work. We could use the trained model to classify the image instead of training many models by ourselves. It not only improves the performance, but also reduces the time cost.

4. Conclusion and Future Work

In this report, we conducted the experiments by applying 5 kinds of neural networks on 2 kinds of scale's dataset. We can make some conclusion. When we consider the time cost as one aspect of the network's performance, the complex and multiple-layer networks do not perform as well as we expect. When these networks used in small scale dataset, they may face many possible problems, for example, there is not enough data for the large network to train, and overfitting. But the large-scale network can be applied to more situations and they performs comprehensively. Therefore, when we get a dataset, we should check the dataset first, and do some data pre-processing on it; then, we can choose the suitable network to work with it and get the best practice for it. Sometimes, we can use transfer learning to reduce our training time and get relevantly acceptable results.

And there are still some works we can do in the semantic segmentation, because the dataset for the real life is extremely large and we cannot train a model for every situation. But we will try to use the transfer learning in this field and get more accuracy and efficient models.

Reference:

Ciregan, D., Meier, U., & Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 3642-3649). IEEE.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Machrisaa. (2017). Retrieved from https://github.com/machrisaa/tensorflow-vgg

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.