An Image Classification Service Based On Convolutional Neural Network *

Peiran Yao, Yilun Wang and Zhengyan Zhang†

Abstract

We trained an artificial neural network model to classify images of four distinct tourist attractions of Tsinghua University. A sixteen-layer convolutional neural network was applied and with some image augmentations the model, trained on a rather limited training dataset, has successfully reached an accuracy of approximately 80 percent.

1 Introduction

Throughout the history of mankind, inventors have been dedicated to create machines that can think. And the invention of electronic computer has sharply shortened the distance towards that great aspiration and several approaches towards artificial intelligence (AI) have been proposed. The idea of composing several simpler functions to form a mathematical function mapping some set of input values to output values to create an AI was first proposed back in the 1940s [McCulloch and Pitts, 1943]. Such approach, now known as *deep learning*, was presumed to be preciously valuable since neuroscience suggested that a single deep learning algorithm may be able to solve many different tasks [Von Melchner et al., 2000]. It is even presumed now to be the only viable approach towards building AI systems that can operate in complicated, real-world environments [Goodfellow et al., 2016]. With the advent of general purpose GPUs, deep learning works perfectly with large models and large datasets. And the combination of deep learning and big data has dramatically improved the state-of-the-art of speech recognition, computer vision, motion planning, natural language processing

*Final assignment of the course *Big Data and Machine Intelligence*, conducted by Fundamental Industry Training Center of Tsinghua University.

†Dept. of Computer Science and Technology, Tsinghua University. E-mail: {ypr15, yl-wang15, zhangzhengyan14}@mails.tsinghua.edu.cn

and other fields [LeCun et al., 2015]. Deep learning has now outperformed competing AI systems [Goodfellow et al., 2016].

Convolutional neural networks(CNN) [LeCun et al., 1989] employ convolution operation in at least one layer of an neural network, which reduces the scale of matrix multiplications and hence brings about the capability of processing larger input. It has been proved potent in many fields of artificial intelligence especially in computer vision, where it was first used to read checks [LeCun et al., 1998] and is now the forerunner of many contests [Krizhevsky et al., 2012].

We exploited CNN to build an image classification service that can classify images of four tourist attractions in Tsinghua University: the Auditorium, the Old Gate, the Main Building and Tsinghua School. Even with a limited set of training data we have achieved an accuracy no less than 80 percent. With more labeled training data the model can be extended to classify more landmarks and have better accuracy, which can be later embedded in intelligent applications such as tour guide systems and augmented reality applications.

2 Image Classification

Machine learning algorithms were defined by [Mitchell, 1997] as a computer program that learns from experience E with respect to some class of tasks T, whose performance P improves with experience E. Among the various tasks, classification, especially image classification (along with object recognition) is one of the most common ones. Modern image classification [Ioffe and Szegedy, 2015, Krizhevsky et al., 2012] and object recognition [Taigman et al., 2014] is best accomplished with deep CNN. We built a deep convolutional neural network with 16 layers and 1212708 parameters that classifies input images into 4 categories.

2.1 Training Data

Most of our training data was retrieved by a web crawler[1] from Microsoft Cognitive Service. Some of the photos were taken manually. The dataset was stratified sampled into a training set and a validation set in respect of a ratio of approximately 3:1. The detailed constitution of the dataset is listed in Table 1.

[1] See Section 6

Category	Training	Validation
The Auditorium	77	23
The Old Gate	107	40
Tsinghua School	74	25
The Main Building	27	10
Total	285	98

Table 1: Constitution of the dataset.

2.2 Augmentation

Our neural network takes an input of 150 pixels × 150 pixels with three channels, therefore all images were first resized to fit the model. All pixels were then applied a mapping to be rescaled from $\{n \in \mathbb{N} \mid 0 \leq n \leq 255\}^3$ to $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}^3$. To make up for the limited set of inputs, all images were randomly sheared, zoomed and performed horizontal flips so that the network will not be fed with duplicated inputs.

2.3 Network

The neural network we applied is a simplified version of VGG-16 [Simonyan and Zisserman, 2014]. The overall structure of the network is illustrated in Figure 1, while layer detail of each component is shown in Figure 2. In general, the neural network represents a mapping from $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}^{3\times150\times150}$ to $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}^4$.

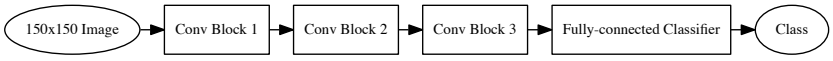
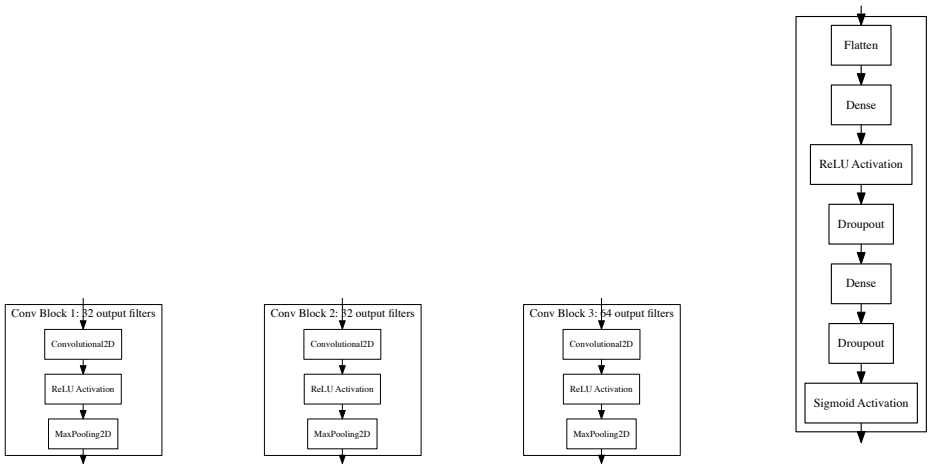


Figure 1: Overview of the neural network.

As is shown in Figure 2a, 2b and 2c, the network begins with a cascading of a similar structure: In the first stage, the network performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a nonlinear activation function, in this case the rectified linear activation function, which is called *the detector stage*. In the third stage, we use a pooling function to modify the output of the layer further [Goodfellow et al., 2016]. Rectified linear units, or ReLU,



(a) Conv Block 1 (b) Conv Block 2 (c) Conv Block 3 (d) Classifier

Figure 2: Detailed model of the neural network.

uses the activation function

$$g(z) = \max\{0, z\}$$

and the overall activation of a *conv block* would be

$$\boldsymbol{h} = g(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}).$$

ReLUs are similar to linear units so they are easy to optimize. *Max pooling* [Zhou and Chellappa, 1988] replaces the output of the network at a certain place with the maximum output within its rectangular neighborhood, which helps to make the representation become approximately invariant to small translations of the input [Goodfellow et al., 2016].

After all convolution operations, the tensor was then flattened to a column vector, which represents features the net extracted. Then we applied fully-connected *dense* layers, which simply takes the form of

$$\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{h} + \boldsymbol{b}.$$

Typically, as the neural network becomes more complicated, the training error decays but the generalization error increases, causing overfitting [Goodfellow et al., 2016], which is illustrated in Figure 3. In this case the training

dataset is too small compared with the deep neural network, so it is prone for the model to overfit. We introduced two layers of *dropout* [Srivastava et al., 2014] that randomly remove units form the net to avoid overfitting.

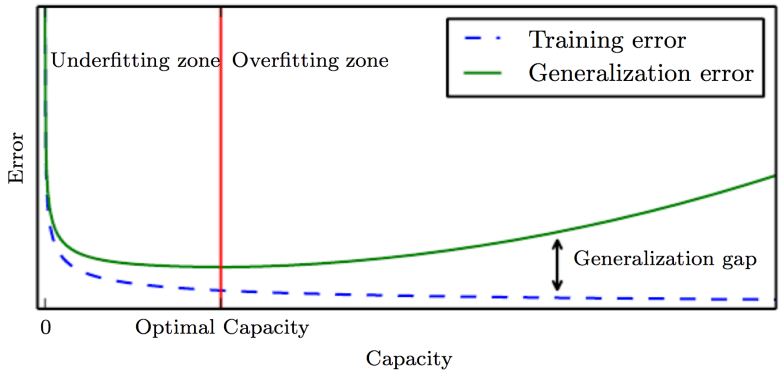


Figure 3: Relationship between capacity and error [Goodfellow et al., 2016].

As the output, we would like to get a probability distribution over all categories, based on which reason a *softmax* output layer was chosen. The final result would be

$$\hat{y} = \mathrm{P}(y = 1 \mid \boldsymbol{x}).$$

The softmax layer first predicts unnormalized log probabilities $z_i = \log \tilde{P}(y = i \mid \boldsymbol{x})$ by

$$\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{h} + \boldsymbol{b}$$

and then normalizes it with the softmax function

$$\mathrm{softmax}(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_j \exp z_j}.$$

As we shall see in next subsection, softmax function works well when maximizing the log-likelihood.

2.4 Cost Function

Our objective, same as the objective of any other supervised learning, is to maximize the maximum likelihood, which attempts to make the model distribution match the empirical distribution drawn from the training set. Here we use the cross-entropy function

$$H_{y'}(y) = -\sum_i y'_i \log y_i$$

and attempts to minimize it. As can be proved mathematically, minimizing the cross-entropy function defined above is equivalent to minimizing the Kullback-Leibler divergence, which in turn is equivalent to maximizing the maximum likelihood [Goodfellow et al., 2016].

2.5 Training

The neural network can be depicted as a parametric function $f(\boldsymbol{X};\boldsymbol{\theta})$, and our objective is to optimize the parameter $\boldsymbol{\theta}$ to minimize the cost function. Generally, the cost function and the layers of a typical neural network are all differentiable, which makes neural networks capable of exploiting an algorithms gradient descent. By continuously move from a point $\boldsymbol{x}$ to another point $\boldsymbol{x}'$ where

$$\boldsymbol{x}' = \boldsymbol{x} - \epsilon \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$$

our algorithm will eventually get to an approximative minimal or minimum of the cost function. The chain rule of calculus

$$\nabla_{\boldsymbol{x}} z = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^{\top} \nabla_{\boldsymbol{y}} z$$

enables the back-propagation algorithm to update every parameter in the neural network to obtain a minimal. It is obvious that the choice of ϵ is essential because if ϵ is too large the algorithm may oscillate near a minimal and if too small it will take a long time to converge. In particular we adopted Adam [Chilimbi et al., 2014] which is a derivative of the classical stochastic gradient descent algorithms that adopts adaptive momentum that simply put changes ϵ dynamically to build an efficient and scalable deep learning training system.

After 5,000 iterations, both the training error and the validation error which reflects the generalization error came to a relative low rate. The process of how the training error and the validation error evolves through iterations is plotted in Figure 4.

2.6 Evaluation

The trained model correctly predicts 80 percent of the samples in the randomly sampled validation set and performed satisfyingly in real-world tests in our live demonstration.

After training, the error on the training set and that on the validation set did not seem to converge at that time but in order to avoid overfitting the training was terminated. But there exists possibility that the training error and the generalization error could be further lowered.

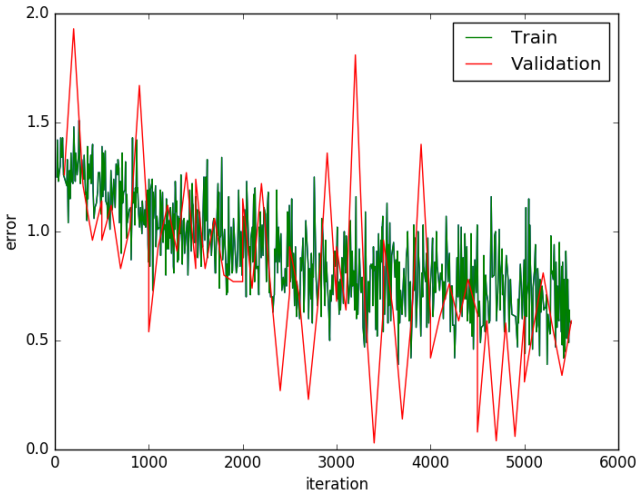


Figure 4: The evolvement of training error and validation error.

Compared with other typical applications of deep neural network, our dataset is definitely too limited that can not utilize the full potential of deep neural networks. Other approaches, like support vector machine [Boser et al., 1992] or traditional computer vision methods like manual feature extraction might perform better or equally given the same task and training set.

3 Future Work

Since neither the training error or the generalization error seems to have reached a minimum value, the first step to improve the model would be retraining it with more iterations. Since our deep feedforward neural network has more than 1,200,000 parameters, with this high capacity the network also requires a dataset with *big data* scale to have better generalization ability. Correspondingly, if the dataset is enlarged, the network can be made deeper with more convolutional layers and larger fully-connected linear and rectified linear layers.

Another significant regularization method of training a neural network is using a strategy called greedy layer-wise pretraining [Hinton, 2007] which takes the advantage of pretrained models and makes them fit into your training set, assuming that a pretrained model has a better *feature extraction* function. Layer-wise pretraining and other regularization methods like image augmentations can be applied as well.

One state-of-the-art technology of image classification is object detection

[Krizhevsky et al., 2012], in which face recognition [Taigman et al., 2014] is a hot topic in industry. Object detection requires not only categories but also strong geometric information. Generally object detection has the same basic structure of image classification but the last layer is replaced with a regression layer based on the insight that "networks which to some extent encode translation invariance, can capture object locations as well" [Szegedy et al., 2013].

4 Conclusion

By building a functioning image classification network we have verified that, convolutional neural network, or deep neural network in general, is a powerful approach towards artificial intelligence, and would be the foundation of intelligent services and systems. Compared with traditional attempts, neural networks relieves engineers from writing "hard code" for each specific task by *learning* through experience itself, which reflects its *intelligence*. We have also recognized that large training data is essential to neural network and big data is and will for a long time be an important field of computer science and engineering.

5 Acknowledgements

We would like to give our special thanks to the instructor of this course, Dr. Zhen Chen and the T.A. Wenxun Zheng, without whose comprehensive guidance and introduction to the fascinating and charming field of deep learning we would not have reaped so much. The workstations and VPS offered generously by iCenter helped with the training and the demonstration.

6 Supplementary Materials

An online demonstration of our network can be found at `http://img.mobsafe.cc/` where you can upload a photo and see the result. All source codes, datasets and trained models are hosted on iCenter GitLab at `http://gitlab.icenter.tsinghua.edu.cn/BDMI_Group1/AI_Neural_Network/`.

References

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

[Chilimbi et al., 2014] Chilimbi, T., Suzue, Y., Apacible, J., and Kalyanaraman, K. (2014). Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 571–582.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Hinton, 2007] Hinton, G. E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–547.

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[LeCun et al., 1989] LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, pages 143–155.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[Mitchell, 1997] Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45:37.

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

[Srivastava et al., 2014] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

[Szegedy et al., 2013] Szegedy, C., Toshev, A., and Erhan, D. (2013). Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561.

[Taigman et al., 2014] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

[Von Melchner et al., 2000] Von Melchner, L., Pallas, S. L., and Sur, M. (2000). Visual behaviour mediated by retinal projections directed to the auditory pathway. *Nature*, 404(6780):871–876.

[Zhou and Chellappa, 1988] Zhou, Y. and Chellappa, R. (1988). Computation of optical flow using a neural network. In *Neural Networks, 1988., IEEE International Conference on*, pages 71–78. IEEE.