Deep Learning Lab - Exercise 2

Baohe Zhang

11.11.2018, Uni Freiburg

1 Introduction

In this exercise we implement a convolutional neural network model, which will be used it classify on the most commonly used dataset - MNIST[1]. Besides that, we will get familiar with Tensorflow[2], a Deep learning framework from Google, and with the help of Tensorflow, we build up a "LeNet" which designed by Yann LeCun[3]. The code is based on the framework provided by AIS Lab of University Freiburg.[4]

1.1 CNN Structure

The convolutional neural network(CNN) "LeNet" which we used, consists of three parts. First part: two convolutional layer with 16 3X3 filters and the stride is 1, each follows with a ReLu activation function and a max-pooling layer with size 2. Second part: a fully connected layer with 128 units. Final part: a output fully connected layer with 12 units, Softmax layer, and cross-entropy loss function. We use stochastic gradient descent method to do optimization.

2 Hyperparameter Tuning

Hyperparameter is one of the most important elements of deep learning. A good combination of hyperparameter will significantly increase the training speed and validation performance.

2.1 Learning Rate

Learning rate has a huge impact on training neural network. A good learning rate will achieve better performance. Setting a bad learning rate will cause

[1] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* ().

[2] Martın Abadi et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. vol. 16. 2016, pp. 265–283.

[3] Yann LeCun et al. "LeNet-5, convolutional neural networks". In: *URL: http://yann. lecun. com/exdb/lenet* (2015), p. 20.

[4] Andreas Eitel. *Deep Learning Lab*. `https://github.com/aisrobots/dl-lab-2018`. 2018.

divergence on training procedure. Here we compare different learning rate: This Graph shows that too small learning rate will underfit the model within

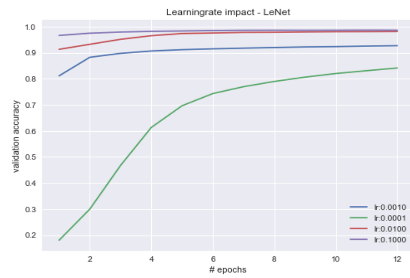

Figure 1: Learning Curve of Different Learning Rate

the same epochs.

2.2 Filter Size

For CNN model, The size of each filter is also a hyperparameter. Here we discuss the influence of filter size by comparing their performance on our validation set. Convolutional layer with filter size bigger than 2 will increase the robustness of

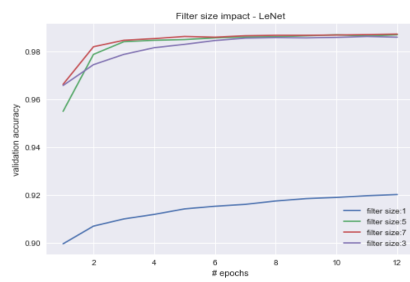

Figure 2: Learning Curve of Different Filter Size

the model for small transition noise and achieve better performance

3 Random Search

Tuning hyperparameters is perhaps the most time-consuming procedure of training a neural network model. With Hpbandster[5], we now can automatically find the best hyperparameter combination. The best combination Hpbandster found for Mnist task by random search is: filter size: 5, learning rate: 0.061334475684028424, batch size: 18, Number of filters: 45.

[5] Stefan Falkner, Aaron Klein, and Frank Hutter. "BOHB: Robust and efficient hyperparameter optimization at scale". In: *arXiv preprint arXiv:1807.01774* (2018).

Here we shows the loss of different budgets over time and the performance of our best model:

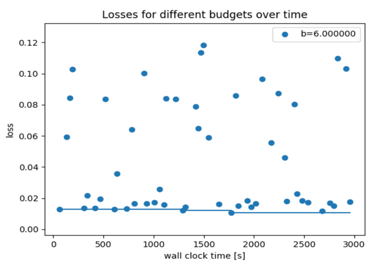

Figure 3: (a) Budgets over TIme (b) Best Model Performance