Lab 1 : ML Environment Setup

Introduction

The main goal of this lab is to set up the environment on the Guppy server for Machine learning experiments and toolchains for Machine learning development. To connect to the Guppy server, we need the SSH program to create a connection. On the server, we use Docker program to create a virtual environment to perform machine learning training tasks. We use PyCharm IDE as a primary IDE for developing python application/program. We also do some testing by executing the AlexNet sample code and the modified version which using the CIFAR-10 dataset as training data.

Methods

SSH set up

We generated the RSA key and upload the public key to the Guppy server which makes the Guppy server recognize our computer when we do SSH. We also write the SSH config file to easily do SSH to Guppy server.

Docker image building

To build a docker image, we need to install some additional packages which are required an internet connection. Therefore, we declare the environment variable name http_proxy and https_proxy in .cshrc file.
Next, we create a file named Dockerfile and copy this code below to the file.

```
FROM nvidia/cuda:9.2-base
ENV http_proxy http://192.41.170.23:3128
ENV https_proxy http://192.41.170.23:3128
RUN apt-get update && apt-get upgrade -y && apt-get install -y openssh-server
RUN mkdir /var/run/sshd
RUN mkdir /root/.ssh/
COPY id_rsa.pub /root/.ssh/authorized_keys
EXPOSE 22

# Set the locale to en_US.UTF-8
RUN apt-get install -y locales
ENV DEBIAN_FRONTEND noninteractive
RUN echo "en_US.UTF-8 UTF-8" > /etc/locale.gen     && locale-gen en_US.UTF-8
&& dpkg-reconfigure locales     && /usr/sbin/update-locale LANG=en_US.UTF-8
ENV LC_ALL en_US.UTF-8

RUN apt-get install -y python3-pip vim
RUN pip3 install --upgrade pip
RUN pip3 install numpy matplotlib
RUN pip3 install torch==1.4.0+cu92 torchvision==0.5.0+cu92 -f
https://download.pytorch.org/whl/torch_stable.html
```

```
# Set root password
RUN echo "root:Docker!" | chpasswd
CMD ["/usr/sbin/sshd", "-D"]
```

This code above has a little modification (reset the root password) due to SSH to the Docker image problem. The problem is SSH requires a root password which we don't know.

Next, we execute the command to build the docker image.

```
docker build . -t nopp-lab1-old-driver
```

Set up PyCharm and AlexNet sample code Experiment

In this part, we start the docker image using this command below.

```
ssh -L 12501:localhost:12501 guppy "docker run -p 12501:22 --gpus all
nopp-lab1-old-driver"
```

After executing the command, we connect the PyCharm IDE with the interpreter on Guppy server through the SSH tunnel on port 12501 which specify in the command above

Individual Part

In this part, we modified the AlexNet sample code to learn to classify images from the CIFAR-10. In this experiment, I choose the parameter learning rate = 0.002 and momentum = 0.9 and training for 25 epochs. All of the data loader (Traning set loader, Validation set loader and Test set loader) use batch size = 16 and the number of workers = 2.

Results

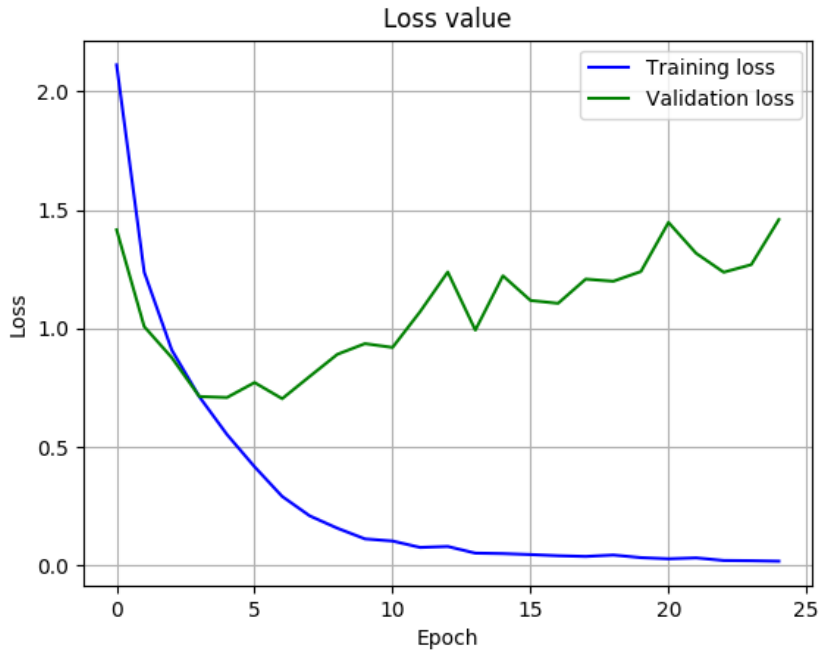

Figure 1. The loss value between training set and validation set.

From figure 1, after 4 epochs of training, the loss value of the validation set start to increase while training loss keeps going down.

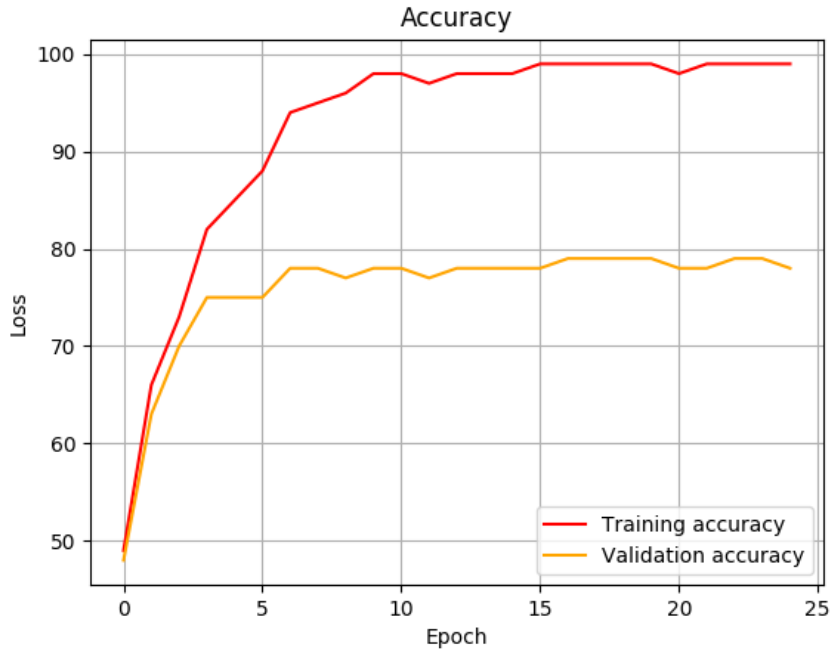

Figure 2. The accuracy of the model between training set and validation set.

From figure 2, the first 5 epochs, the accuracy is getting higher as normal. Near the end of 25 epochs, it has some evidence of overfitting since the validation accuracy decrease while training accuracy has no change.

In the final, the accuracy of test set data which contains 10000 images is 78%.

Conclusion

In this lab, I have learned about how to set up the environment for Machine learning development with the GPU server, how the SSH and Docker work and the PyTorch framework works. I also found that training CNN requires a lot of memory space. We can change the parameter in the data loader to reduce the memory space cost.

For the next step, we can tweak some hyperparameters of the model to gain more accuracy. My suggestion is to make some modifications to train the model and every epoch we save the model until the accuracy of the validation set starts to drop more than the threshold then we stop. We will choose the mode which gives the best test set accuracy from the saved model. This method may cost a lot of computing resource but it is guaranteed that we will get the best model on each hyperparameter configuration.