

HW2

- You will learn how to train a deep network using PyTorch tool. Please read the following tutorial. You may skip the data parallelism section.

https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

- CIFAR10 is a dataset of 60,000 color images of size 32×32 from 10 categories. Please download the PyTorch tutorial code for CIFAR10 to start:

https://pytorch.org/tutorials/_downloads/cifar10_tutorial.py

- When you run the tutorial code, it will download CIFAR10 dataset for you. Please follow the instructions in the following link to install PyTorch: <https://pytorch.org/>

- To learn more, you can also find a tutorial for MNIST here:

https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html#sphx-glr-beginner-blitz-neural-networks-tutorial-py

and the sample model for MNIST here:

<https://github.com/pytorch/examples/blob/master/mnist/main.py>

and the sample code for Imagenet here:

<https://github.com/pytorch/examples/blob/master/imagenet/main.py>

- For all the following sections, train the model for 50 epochs and plot the curve for loss, training accuracy, and test accuracy evaluated every epoch.

1. Run the tutorial code out of the box and make sure you get reasonable results. You will report these results in Section 2, so no report needed here.
2. Change the code to have only a single fully connected layer. The model will have a single layer that connects the input to the output. What is the number of parameters? In PyTorch, "nn.Linear" can be used for fully connected layer.
3. Change the code to have multiple fully connected layers. Try having a layer from input to 110 neurons and then a layer to 74 neurons, and finally a layer to 10 neurons, one for each category. What happens if you use or not use ReLU after each layer? Describe why.
4. Change the code by adding two convolutional layers along with maxpooling layers before the fully connected layers. This will be similar to the example in the tutorial. Use this model for the following sections.
5. Try multiple batch sizes to see the effect and describe the findings. Please use batch size of 1, 4, and 1000. If 1000 does not fit into the memory of your machine, please feel free to reduce it to the largest possible number.
6. Try multiple learning rates to see the effect and describe the findings. Please use learning rates of 10, 0.1, 0.01, and 0.001.
7. Please add some data augmentation to avoid overfitting. Note that you need to do this only for the training and not the testing. You may use line 244-260 from Imagenet sample code:

<https://github.com/pytorch/examples/blob/master/imagenet/main.py>

"RandomResizedCrop" samples a random patch from the image to train the model on. "RandomHorizontalFlip" flips randomly chosen images horizontally.

8. Change the loss function from Cross Entropy to Mean Squared Error and report the effect.