

Assignment #3: PyTorch Convolutional Network

Start Assignment

- 기한 3월2일 오후 11:59까지
- 배점 50
- 제출 방식 파일 업로드
- 파일 유형 ipynb


In this assignment, you will use PyTorch to create a convolutional neural network, and also utilize TensorBoard to visualize the training process. TensorBoard is a nice visualization tool for deep neural networks. This assignment is due on **Sunday Mar. 2nd 11:59PM**.

1) Create a ResNet-14 that classifies the CIFAR-10 dataset. The ResNet-14 architecture can be found in Section 4.2 of the paper.

<https://arxiv.org/pdf/1512.03385.pdf> (<https://arxiv.org/pdf/1512.03385.pdf>)

with $n = 2$. Note that the paper itself never used $n = 2$. But it should still lead to a pretty good network.

This tutorial should be helpful, or you can also use chatGPT/coPilot or any language-model based helper as well:

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html 
(https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)

For the global average pooling specified in the paper, you can use `torch.nn.AvgPool2d` with a kernel size the same as the image.

For the training loss function, you can directly use `torch.nn.CrossEntropyLoss` which by default handles multi-class. `torch.nn.CrossEntropyLoss` will automatically put a softmax, together with the log-likelihood loss function, so you **DON'T** have to make another softmax layer before it.

(20 points)

2) Setup TensorBoard on Google Colab:

This tutorial might be helpful:

https://pytorch.org/tutorials/intermediate/tensorboard_tutorial.html
(https://pytorch.org/tutorials/intermediate/tensorboard_tutorial.html)

Note: To use TensorBoard on Colab, you may need to run this first in Colab:

```
%load_ext tensorboard
```

Then:

```
%tensorboard --logdir runs
```

launches TensorBoard, instead of the corresponding command in the tutorial.

NOTE #2:

There might be a strange bug when you call `add_embedding`, it can be solved by adding the following 3 lines to Colab before you invoke a tensorboard writer.

```
import tensorflow as tf
import tensorboard as tb
tf.io.gfile = tb.compat.tensorflow_stub.io.gfile
```

3) Train the model for at least 30 epochs. Make tensorboard to display the training curve (training loss vs. epoch) as well as some images from the training set. (10 points)

4) Try to add visualizations of testing error and testing loss with TensorBoard. (10 points)

5) Try to tune the network in one other way of your choice. Train it again to check how does the performance change? Visualize the loss with TensorBoard. (10 points)

Please put all these in an iPython notebook, with **clear labels of your name and different sections of the assignment**. Then submit the iPython notebook on Canvas.

Rubric note:

If your code runs but the learning diverges (e.g. going to infinity or not a number, or 10% accuracy), the points for that particular part (part 3, part 5) will be deducted. For each setting, this will be 5 points deduction out of 50 total. If the code does not run, then you will get at most 10 points for part 1.