Artificial Neural Networks

3rd Assignment - Shahid Beheshti University - Master's Program

April 11, 2023

Due date: April 25

Exercise 1

If your GPU runs out of memory while training a CNN, what can you do to solve the problem?

Exercise 2

What is gradient accumulation? When should we use this technique? How to perform this in PyTorch?

Exercise 3

Describe the backpropagation details in the convolutional layers. (For a better understanding, check out this link)

Exercise 4

What are the benefits of the pooling layers? What are the drawbacks of the pooling layers? Are you willing to use these layers? Can you use these layers frequently?

Exercise 5

True/False, Explain the reason:

> Face verification requires comparing a new picture against one person's face, whereas face recognition requires comparing a new picture against K person's faces.
>
> In order to train the parameters of a face recognition system, it would be reasonable to use a training set comprising 100,000 pictures of 100,000 different persons.

You train a CNN on a dataset with 100 different classes. You wonder if you can find a hidden unit that responds strongly to pictures of cats. (I.e., a neuron so that, of all the input/training images that strongly activate that neuron, the majority are cat pictures.) You are more likely to find this unit in layer 4 of the network than in layer 1.

Exercise 6

Traffic Sign Label Prediction. By utilizing CNN models, you are going to predict the labels of traffic sign images, the corresponding data is available here. The dataset consists of 34,799 of 32*32 images with certain labels. You should use `train.pickle`, `valid.pickle`, `test.pickle`, `label_names.csv` files to solve the problem. By using the pickle library, you should import the necessary files, each file is a dictionary that has 2 keys: `features` that consists of images and `lables` that have the corresponding label for each image. Consider the following points when tackling the problem:

- Implement a decent evaluation pipeline and compare all the models that you use to tackle the problem. Some metrics that you can use, are `f1_score`, `precision`, `recall`, and `confusion matrix`.
- Try to make a custom CNN model to perform best on the data. Use various techniques to boost your performance.
- Try to solve the problem using Transfer Learning by utilizing various pre-trained models.
- Analyze which signs are usually mispredicted by the model. What could be the reason?
- Check whether using grayscale images affects the performance of the model or not.
- Check whether the data is balanced or not. If not try to solve the problem and achieve better performance. (Hint: you can try up-sampling and down-sampling approaches to tackle this problem. For an up-sampling approach you can consider augmenting or synthesizing images)(Extra point)
- Visualize the output of multiple random images on different layers of your network and try to explain them. (Extra point)
- Visualize different images in the train set and compare them in terms of illumination, aspect ratio, or other aspects. Try to use appropriate transformation to pre-process the images and boost your model's performance. For, example in terms of illumination, if images differ considerably, you can use `Histogram Equalization` to solve the issue. (Extra point)