BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment 3
Spring 2022-2023
Dr. Nazli Ikizler Cinbis

Due Date: 23:59 pm on Wednesday, May 17th, 2023

Image Classification with Convolutional Neural Networks

In this assignment, you will get familiar with image classification by training Convolutional Neural Networks (CNN).The goals of this assignment are as follows;

1. Understand CNN architectures and build a model from scratch and train on data.
2. Understand and implement transfer learning from a pre-trained CNN.
3. Analyze your results.
4. Experience with a deep learning frame, PyTorch.

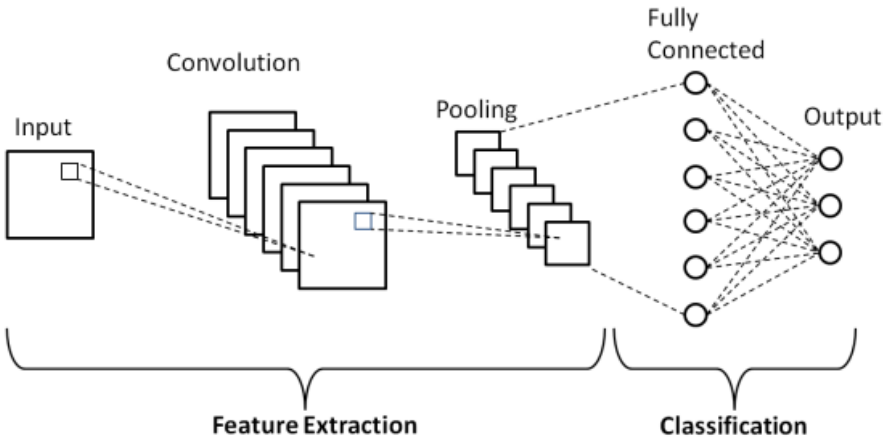

Figure 1: A simple CNN architecture [1].

Dataset

Dataset consists of sample of bacteria images, see Figure 2. There are 8 categories; Amoeba, Euglena, Hydra, Paramecium, Rod bacteria, Spherical bacteria, Spiral bacteria and Yeast. Each of the image belongs to one of the 8 categories. In this assignment, you will train classifiers to classify images to one of the 8 categories. You need to divide your dataset into train, validation, and test sets. If the computation time is a problem, you can use a subset of this dataset. However, make sure that your training set is at least 400 images (50 images per class). The validation and test sets should include at least 10 images per class, a total of 160 images each. Note that the more data you use, the better the learning will be. **Explain how you arrange your dataset into train, validation, and test and write how many images are there in the sets. [5 pts]**

PART 1 - Modeling and Training a CNN classifier from Scratch [55 pts]

In this part, you are expected to model two CNN classifier and train it. You should first define the components of your two model. Both of your models should have 6 convolutional layers and a fully connected layer. After the first layer, add Max Pooling layer. One of your models have residual connection(s) and the other does not. Your model with residual connection(s) must have one or more residual connections.

Figure 2: Dataset class samples.

- Give parametric details with relevant Pytorch code snippets; number of in channels, out channels, stride, etc. Specify your architecture in detail. Write your choice of activation functions, loss functions and optimization algorithms with relevant code snippets. [2.5 pts x 2 = 5 pts]
- Explain how you have implemented residual connection(s) and give relevant code snippets. [5 pts]
- You can apply augmentation on images to obtain better accuracy.

Training and Evaluating your model

For both of your models;

- you will set epoch size to 100 and evaluate your two model with two different learning rates and two different batch sizes. Explain how you calculate accuracy with relevant code snippet. Moreover for each of the points below add relevant code snippet to your document.

1. Draw a graph of loss and accuracy change for two different learning rates and two batch sizes. [5 pts x 2 = 10 pts]
2. Select your best model with respect to validation accuracy and give test accuracy result. [2.5 pts x 2 = 5 pts]
3. Integrate dropout to your best models (best model should be selected with respect to best validation accuracy. You need to integrate to both of your best models). In which part of the network you add dropout and why? Explore four different dropout values and give new validation and test accuracies. [5 pts x 2 = 10 pts]
4. Plot a confusion matrix for your best model's predictions. [5 pts x 2 = 10 pts]
5. Explain and analyze your findings and results. [5 pts x 2 = 10 pts]

PART 2 - Transfer Learning with CNNs [40 pts]

Now, you will fine-tune the pre-trained ResNet-18 network which is available at Pytorch. This network is trained on ImageNet dataset so you are not initializing the weights randomly, instead, you are using the pre-trained weights from this network. Freeze all the layers before training the network, except the FC layer. Consequently, the gradients will not be calculated for the layers except the ones that you are going to update (FC layer) over the pre-trained weights. However, since the number of classes is different for our dataset, you should modify the last layer of the network, which the probabilities will be calculated on. Therefore, the weights will be randomly initialized for this layer.

1. What is fine-tuning? Why should we do this? Why do we freeze the rest and train only FC layers? Give your explanation in detail with relevant code snippet. [5 pts]
2. Explore training with two different cases; train only FC layer and freeze rest, train last two convolutional layers and FC layer and freeze rest. Tune your parameters accordingly and give accuracy on validation set and test set. Compare and analyze your results. Give relevent code snippet. [20 pts]

3. Plot confusion matrix for your best model and analyze results. [5 pts]

4. Compare and analyze your results in Part-1 and Part-2. Please state your observations clearly and precisely. [10 pts]

What to Hand In

Your submission format will be:

- README.txt *(give a text file containing the details about your implementation, how to run your code, the organization of your code, functions etc.)*
- code/ *(directory containing all your code)*
- report.pdf

Archieve this folder as **b<studentNumber>.zip** and submit to `https://submit.cs.hacettepe.edu.tr`.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] `https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697`