BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY
Department of Computer Engineering

Programming Assignment-3
Spring 2019-2020
Dr. Nazli Ikizler Cinbis

Due Date: 23:59 pm on Friday, May 15th, 2020

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;

- Understand CNN architectures and build a model from scratch and train on data
- Understand and apply transfer learning from a pretrained CNN.
- Understand and analyze your results.
- Gain experience with a major deep learning frame, PyTorch.

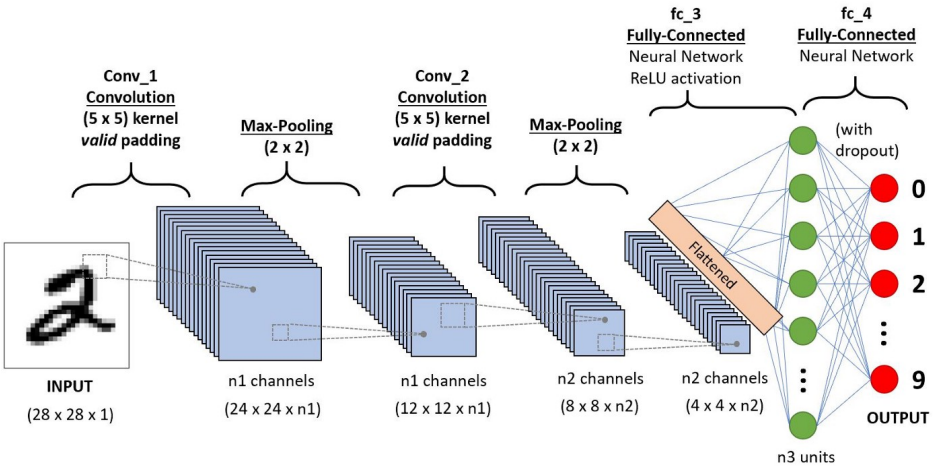

Figure 1: A basic CNN sequence for classification, taken from [1].

Dataset

The dataset [2] includes different hand images photographed from both dorsal and palmar sides with a white background and placed in the same distance from the camera. Also for each image there is a ground-truth record including the subject ID, gender, age, skin color, a set of information of the captured hand (right or left), hand side (dorsal or palmar), and logical values including to whether the hand image contains accessories, nail polish, or irregularities (See Figure 2). *In this assignment, you will only learn to classify 8 classes; aspectOfHand (female-dorsal-right, female-dorsal-left, female-palmar-left, male-palmar-right, male-dorsal-right, male-dorsal-left, male-palmar-left, male-palmar-right) attribute.* . Do not forget to divide your dataset to train, validation, and test sets.

PART 1 - Modeling and Training a CNN classifier from Scratch - [50 pts]

In this part, you are expected to model a CNN classifier and train it. You should first define the components of your model. For instance; a simple architecture might contain 3 convolutional layers and 2 fully connected layers. Define your initial network.

- How many layers are there on your initial classifier? What structure do these layers have? Give parametric details; number of in channels, out channels, stride, etc. Specify your architecture in detail. Write about your choice of activation function, loss function and optimization algorithm [15 pts]

Figure 2: Sample images from the dataset.

After that, start training your network. Now it is time to select hyperparameters for training like; number of epochs, learning rate, batch-size etc.

- Draw a graph to show your model's behaviour with respect to different learning rates on different epochs. Try with four different learning rates and show the change in the loss value for different learning rates. [10 pts]
- After selecting best hyperparameters, give training, validation, and test accuracy. Plot your model's loss and accuracy change. Comment on your findings. [5 pts]

Regularize your network:

- Integrate dropout to your network and give details where you applied. Explore different values to find one that fits. [10 pts]

At the end, you have optimized architecture and ready to test it on the test set. Select the best architecture and parameters according to your results on the validation set. What is your model's test set accuracy? Plot confusion matrix. Explain your findings and results. [10 pts]

PART 3 - Transfer Learning with CNNs - [50 pts]

Now, you will fine-tune the ResNet-18 network. 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 layers. Consequently, the gradients will not be calculated for the layers except the ones that you are going to update (FC layers) 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.

- 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. [10 pts]
- Explore training different number of layers such as; train all FC layers or some of it, or start training the last conv layer and the rest of FC layers and so on. Tune your parameters accordingly and give accuracy on validation set. Compare your results. [20 pts]
- Evaluate your best trained model in the test set and plot the confusion matrix. What do the results tell us? [5 pts]
- Do you think you can improve the performance of your model? What approaches do you think might help to improve the performance? [5 pts]

You are now expected to compare your results in Part-2 and Part-3. Please state your observations clearly and precisely. [10 pts]

The Implementation Details

1. You should pay attention to code readability such as comments, function/variable names and your code quality: 1) no hard-coding 2) no repeated code 3) cleanly separate and organize your code 4) use consistent style, indentation

2. Implement your code with Python 2 or 3 and use the necessary Pytorch, OpenCV, NumPy libraries.
3. You should use the PyTorch as the deep learning framework. You can use Google Colab to run your experiments.

What should you write in the report?

- Give explanations for each step.
- Give experimental results, used parameters and comments on the results in detail.
- Give your model's loss plot and accuracy plot both for training and validation set during training.
- A basic structure of the report might be: 1) Introduction (what is the problem, how do you approach to this problem, what is the content of your report) 2) Implementation Details (the method you followed and details of your solution) 3) Experimental Results (all results for separate parts with different parameters and your comments on the results) 4) Conclusion (what are the results and what are the weaknesses of your implementation, in which parts you have failed and why, possible future solutions)
- You are advised to use LaTeXin report writing (but not required).

What to Hand In

The upload link is; https://classroom.github.com/a/zD4ysFYk
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

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://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
2. https://sites.google.com/view/11khands