Ex 5 - CNNs & Transfer Learning

Submission Guidelines

Practical part - will be named “ex_9_practical” on submit. You should submit:

- ex_9_code.py
- details.txt
- test.pred

Theoretical part - will be named “ex_10_theoretical” on submit. You should submit:

- report.pdf

Deadline - 14.6.18 (2 weeks) there will be no exceptions.

Please follow the guidelines to avoid any unnecessary points deduction.

Transfer Learning

In this exercise we will implement our first convolutional neural network.
Training a network from scratch might be time consuming. A common practice to mitigate this issue is called “Transfer Learning”.

In transfer learning, we take a pre-trained model and use it in one of the following ways:

1. CNN as feature extractor - we take the pre-trained model and replace its last fully-connected layer with a new fully-connected layer that fits the dimension of our problem (e.g. our output size). We then train (update) only the newly added layer, keeping the rest of the model unchanged.
2. Fine-tuning - same as (1) only we also update the rest of the model.

You can read more about transfer learning here. There are also code example for transfer learning in pytorch here and here.

Instructions

1. You should train a model from scratch on CIFAR-10 dataset:
    a. The basic architecture of your model should be as follows:
        i. Conv Layer
        ii. ReLU activation function
        iii. Pooling layer
        iv. Conv Layer
        v. ReLU activation function
        vi. Pooling layer
        vii. Fully connected
        viii. ReLU activation function
        ix. Fully connected
        x. ReLU activation function

        xi. Fully connected
        xii. Softmax
    b. You should choose the number of filter and filter sizes as you wish.
    c. You can add dropout layers and batchnorm layers as you wish.
    d. You should optimize the cross entropy loss function.

2. To better understand how transfer learning accelerates your training process -- You will load a ResNet-18 model and use it as a feature extractor (see this link).
3. You should replace the last fully-connected layer of the model as shown in the link and train on CIFAR-10 dataset. **Notice**: ResNet-18 expects a 224x224 image while CIFAR-10 images are 32x32. Before feeding CIFAR-10 images to ResNet-18 you should resize them to 224x224. You can use PyTorch’s built-in transformation layer to accomplish that.
4. You should train both models and report the following in a PDF file:
    a. Plot the loss on training set and validation set as a function of the epochs.
    b. Accuracy of the final model on training set and validation set.
    c. Accuracy and loss of the final model on the test set.
    d. Confusion Matrix of the test set using the final model. You can use sklearn to compute the confusion matrix (see the following link).
5. Finally, you should produce a test.pred file with your model’s predictions on examples provided in test.x file.

Good luck!

Deep Learning
What society thinks I do
What my friends think I do
What other computer scientists think I do
What mathematicians think I do
What I think I do
from theano import *
What I actually do