EECS 498-007 / 598-005
Deep Learning for Computer Vision
Fall 2020

Assignment 4

From this assignment forward, you will use autograd in PyTorch to perform backpropgation for you. This will enable you to easily build complex models without worrying about writing code for the backward pass by hand.

The goals of this assignment are:

- Understand how autograd can help automate gradient computation
- See how to use PyTorch Modules to build up complex neural network architectures
- Understand and implement recurrent neural networks
- See how recurrent neural networks can be used for image captioning
- Understand how to augment recurrent neural networks with attention
- Use image gradients to synthesize saliency maps, adversarial examples, and perform class visualizations
- Combine content and style losses to perform artistic style transfer

This assignment is due on **Friday, October 30 at 11:59pm EDT**.

Q1: PyTorch Autograd (30 points)

The notebook **pytorch_autograd_and_nn.ipynb** will introduce you to the different levels of abstraction that PyTorch provides for building neural network models. You will use this knowledge to implement and train Residual Networks for image classification.

Q2: Image Captioning with Recurrent Neural Networks (40 points)

The notebook **rnn_lstm_attention_captioning.ipynb** will walk you through the implementation of vanilla recurrent neural networks (RNN) and Long Short Term Memory (LSTM) RNNs. You will use these networks to train an image captioning model. You will then augment your implementation to perform spatial attention over image regions while generating captions.

Q3: Network Visualization (15 points)

The notebook **network_visualization.ipynb** will walk you through the use of image gradients for generating saliency maps, adversarial examples, and class visualizations.

Q4: Style Transfer (15 points)

In the notebook **style_transfer.ipynb**, you will learn how to create images with the artistic style of one image and the content of another image.

Steps

1. Download the zipped assignment file

- Click here to download the starter code

2. Unzip all and open the Colab file from the Drive

Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the *.ipynb file. We recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. For more information on using Colab, please see our Colab tutorial.

3. Work on the assignment

Work through the notebook, executing cells and writing code in `*.py,` as indicated. You can save your work, both `*.ipynb` and `*.py,` in Google Drive (click "File" -> "Save") and resume later if you don't want to complete it all at once.

While working on the assignment, keep the following in mind:

- The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks**.
- **Do not add or delete cells from the notebook**. You may add new cells to perform scratch computations, but you should delete them before submitting your work.
- **Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.

4. Evaluate your implementation on Autograder

Once you want to evaluate your implementation, please submit the `*.py` , `*.ipynb` and other required files to Autograder for grading your implementations in the middle or after

implementing everything. You can partially grade some of the files in the middle, but please make sure that this also reduces the daily submission quota. Please check our Autograder tutorial for details.

5. Download .zip file

Once you have completed a notebook, download the completed `uniqueid_umid_A4.zip` file, which is generated from your last cell of the `style_transfer.ipynb` file. Before executing the last cell in `style_transfer.ipynb`, please manually **run all the cells of notebook and save your results** so that the zip file includes all updates.

Make sure your downloaded zip file includes your most up-to-date edits; the zip file should include:

- *pytorch_autograd_and_nn.ipynb*
- *rnn_lstm_attention_captioning.ipynb*
- *network_visualization.ipynb*
- *style_transfer.ipynb*
- *pytorch_autograd_and_nn.py*
- *rnn_lstm_attention_captioning.py*
- *network_visualization.py*
- *style_transfer.py*
- *pytorch_autograd_and_nn.pkl*
- *rnn_lstm_attention_submission.pkl*
- *saliency_maps_results.jpg*
- *adversarial_attacks_results.jpg*
- *class_viz_result.jpg*
- *style_transfer_result.jpg*
- *feature_inversion_result.jpg*

6. Submit your python and ipython notebook files to Autograder

When you are done, please upload your work to Autograder (UMich enrolled students only). Your `*.ipynb` files *SHOULD include* all the outputs. Please check your outputs up to date before submitting yours to Autograder.
Note: Autograder for A4 will start working on October 13.

EECS 498-007 / 598-005: Deep Learning for Computer Vision

Justin Johnson
justincj@umich.edu

Website for UMich EECS course