

Homework and Programming Set 4

In this assignment you will practice writing backpropagation code, and training Neural Networks and Convolutional Neural Networks. The goals of this assignment are as follows:

- understand the basic Image Classification pipeline and the data-driven approach (train/predict stages)
- implement and apply a Multiclass Support Vector Machine (SVM) classifier
- understand **Neural Networks** and how they are arranged in layered architectures
- understand and be able to implement (vectorized) **backpropagation**
- implement various **update rules** used to optimize Neural Networks
- effectively **cross-validate** and find the best hyperparameters for Neural Network architecture
- understand the architecture of **Convolutional Neural Networks** and train gain experience with training these models on data
- understand the architecture of recurrent neural networks (RNNs) and how they operate on sequences by sharing weights over time
- understand and implement both Vanilla RNNs and Long-Short Term Memory (LSTM) RNNs

Setup

Here's how you install the necessary dependencies:

(OPTIONAL) Installing GPU drivers: You are at no disadvantage for most questions of the assignment. For Question 3, which is in TensorFlow or PyTorch, however, having a GPU will be a significant advantage. We recommend using a Google Cloud Instance with a GPU, at least for this part. If you have your own NVIDIA GPU, however, and wish to use that, that's fine -- you'll need to install the drivers for your GPU, install CUDA, install cuDNN, and then install either [TensorFlow](#) or [PyTorch](#). You could theoretically do the entire assignment with no GPUs, though this will make training much slower in the last part.

Installing Python 3.5+: To use python3, make sure to install version 3.5 or 3.6 on your local machine. If you are on Mac OS X, you can do this using [Homebrew](#) with `brew install python3`. You can find instructions for Ubuntu [here](#).

Virtual environment: We recommend using [virtual environment](#) for the project. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed globally on your machine. This assignment contains three parts and you may set up a virtual environment for each part. For example, to set up a virtual environment for Part 1, run the following:

```
cd part1
sudo pip install virtualenv
virtualenv -p python3 .env
source .env/bin/activate
pip install -r requirements.txt
```

```
#
# This may already be installed
# Create a virtual environment (python3)
# Activate the virtual environment
# Install dependencies
```

```
# Note that this does NOT install TensorFlow or PyTorch,  
# which you need to do yourself.  
  
# Work on the assignment for a while ...  
# ... and when you're done:  
deactivate                                # Exit the virtual environment
```

Note that every time you want to work on each part of the assignment, you should run `source .env/bin/activate` (e.g., from within your `part1` folder) to re-activate the virtual environment, and `deactivate` again whenever you are done.

NOTE 2: If you are working in a virtual environment on OSX, you may *potentially* encounter errors with matplotlib due to the [issues described here](#). In our testing, it seems that this issue is no longer present with the most recent version of matplotlib, but if you do end up running into this issue you may have to use the `start_ipython_osx.sh` script from the `part1` directory (instead of `jupyter notebook` above) to launch your IPython notebook server. Note that you may have to modify some variables within the script to match your version of python/installation directory. The script assumes that your virtual environment is named `.env`.

Submitting your work:

Once you are done working, go to the directory of each part and run the `collectSubmission.sh` script, which will produce a zip file, e.g., `assignment1.zip`. Please submit all three zip files. Please don't submit downloaded data.

Part 1 (15 points)

Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Run the following from the `part1` directory:

```
cd iiisai/datasets  
./get_datasets.sh
```

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the `part1` directory, with the `jupyter notebook` command.

Q1: Training a Support Vector Machine (15 points)

The IPython Notebook `svm.ipynb` will walk you through implementing the SVM classifier.

Part 2 (45 points)

Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Run the following from the `part2` directory:

```
cd iiisai/datasets
./get_datasets.sh
```

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the `part2` directory, with the `jupyter notebook` command.

Q2: Fully-connected Neural Network (15 points)

The IPython notebook `FullyConnectedNets.ipynb` will introduce you to our modular layer design, and then use those layers to implement fully-connected networks of arbitrary depth. To optimize these models you will implement several popular update rules.

Q3: Convolutional Networks (20 points)

In the IPython Notebook `ConvolutionalNetworks.ipynb` you will implement several new layers that are commonly used in convolutional networks.

Q4: PyTorch / Tensorflow on CIFAR-10 (10 points)

For this last part, you will be working in either TensorFlow or PyTorch, two popular and powerful deep learning frameworks. **You only need to complete ONE of these two notebooks.** You do NOT need to do both.

Open up either `PyTorch.ipynb` or `TensorFlow.ipynb`. There, you will learn how the framework works, culminating in training a convolutional network of your own design on CIFAR-10 to get the best performance you can.

Part 3 (40 points)

Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Run the following from the `part3` directory:

```
cd iiisai/datasets
./get_assignment3_data.sh
```

Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the `part3` directory, with the `jupyter notebook` command.

Q5: Image Captioning with Vanilla RNNs (20 points)

The Jupyter notebook `RNN_Captioning.ipynb` will walk you through the implementation of an image captioning system on MS-COCO using vanilla recurrent networks.

Q6: Image Captioning with LSTMs (20 points)

The Jupyter notebook `LSTM_Captioning.ipynb` will walk you through the implementation of Long-Short Term Memory (LSTM) RNNs, and apply them to image captioning on MS-COCO.