Account
Dashboard
Courses
Groups
Calendar
82
Inbox
Commons
Help

*Spring 2019*
Home
Assignments
Discussions
Grades
People
Pages
Files
Syllabus
Conferences
Collaborations
Chat
Course Captures

View All Pages

# Assignment 1 Description

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 **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
- implement **batch normalization** for training deep networks
- implement **dropout** to regularize networks
- effectively **cross-validate** and find the best hyperparameters for Neural Network architecture
- understand the architecture of **Convolutional Neural Networks** and gain experience with training these models on data

Submit by 11pm on Tuesday 2/19

## Setup

Get the code as a zip file here:

assignment1.zip

You will need a unix shell for the command scripts. Its possible, although probably quite difficult to do the assignment on Windows. We officially support Linux and MacOS. If you're on windows, you need to install cygwin or a Linux VM such as VirtualBox. Make sure you include a C++ compiler in your cygwin install. If you;re not already a cygwin user, you will probably find it easier to set up Virtual Box. That will also give you a standard Python distribution.

**Note:** this is a Python 2.7 project. It will not work under Python 3.

As for the dependencies:

**[Option 1] Use Anaconda:** The preferred approach for installing all the assignment dependencies is to use Anaconda, which is a Python distribution that includes many of the most popular Python packages for science, math, engineering and data analysis. Once you install it you can skip all mentions of requirements and you're ready to go directly to working on the assignment.

**[Option 2] Manual install, virtual environment:** If you'd like to (instead of Anaconda) go with a more manual and risky installation route you will likely want to create a 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. To set up a virtual environment, run the following:

```
cd assignment1
sudo pip install virtualenv      # This may already be installed
virtualenv .env                  # Create a virtual environment
source .env/bin/activate         # Activate the virtual environment
pip install -r requirements.txt  # Install dependencies
# Work on the assignment for a while ...
deactivate                       # Exit the virtual environment
```

**Download data:** Once you have the starter code, you will need to download the CIFAR-10 dataset. Run the following from the `assignment1` directory:

```
cd deeplearning/datasets
./get_datasets.sh
```

**Compile the Cython extension:** Convolutional Neural Networks require a very efficient implementation. We have implemented the functionality using Cython; you will need to compile the Cython extension before you can run the code. From the `deeplearning` directory, run the following command:

```
python setup.py build_ext --inplace
```

**Start IPython:** After you have the CIFAR-10 data, you should start the IPython notebook server from the `assignment1` directory. If you are unfamiliar with IPython, you should read this IPython tutorial. Note that the default ipython will probably use python3. You can start a python2 notebook by typing `ipython2 notebook`.

**NOTE:** If you are working in a virtual environment on OSX, you may encounter errors with matplotlib due to the issues described here. You can work around this issue by starting the IPython server using the `start_ipython_osx.sh` script from the `assignment1` directory; the script assumes that your virtual environment is named `.env`.

## Submitting your work:

Once you are done working run the `collectSubmission.sh` script from the assignment1 directory; this will produce a file called `assignment1.zip`. Submit this file at the end of this assignment.

## Q1: Fully-connected Neural Network (30 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.

## Q2: Batch Normalization (30 points)

In the IPython notebook `BatchNormalization.ipynb` you will implement batch normalization, and use it to train deep fully-connected networks.

## Q3: Dropout (10 points)

The IPython notebook `Dropout.ipynb` will help you implement Dropout and explore its effects on model generalization.

## Q4: ConvNet on CIFAR-10 (30 points)

In the IPython Notebook `ConvolutionalNetworks.ipynb` you will implement several new layers that are commonly used in convolutional networks. You will train a (shallow) convolutional network on CIFAR-10, and it will then be up to you to train the best network that you can.

## Q5: Do something extra! (up to +10 points)

In the process of training your network, you should feel free to implement anything that you want to get better performance. You can modify the solver, implement additional layers, use different types of regularization, use an ensemble of models, or anything else that comes to mind. If you implement these or other ideas not covered in the assignment then you will be awarded some bonus points.

## Check your Submission

You can check your submission on a standard Python implementation in Virtual Box. First download and install virtual box from here. Then grab this zip file, save and unzip it.

- Open VirtualBox and click the 'New' button.
- Select the following options in the VM creation wizard that appears:
  - *Name and operating system*
    - Type: Linux
    - Version: Ubuntu (64-bit)
  - Memory size: at least 1024 MB, preferably half the physical memory on your machine.
  - *Hard drive*
    - Use an existing virtual drive file; select the disk image (`.vdi` file) you unzipped
  - *CPU cores*
    - Under Settings → System → Processor
      allocate half the machine's cores to the virtual machine

Then you can start your virtual machine, and test the assignment inside it. You can mount directories from your host machine or use the network to copy the assignment into the VM. The account name is "deep" and has password "deep". The account has sudo access. Sorry, only Python 2.7 support for now.

## Submit

Go to the assignment 1 submission page, and click the "submit" button on the top right. Submit by 11pm on Tuesday 2/19
Based on assignment 2, CS231n