CS231n Convolutional Neural Networks for Visual Recognition

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 the core parameter update loop of **mini-batch gradient descent**
- 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

Setup

You can work on the assignment in one of two ways: locally on your own machine, or on a virtual machine through Terminal.

Working locally

Get the code: Download the starter code here.

[Optional] virtual environment: Once you have unzipped the starter code, you might 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 on your machine. To set up a virtual environment, run the following:

```
cd assignment2
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
```

You can reuse the virtual environment that you created for the first assignment, but you will need to run `pip install -r requirements.txt` after activating it to install additional dependencies required by this assignment.

Download data: Once you have the starter code, you will need to download the CIFAR-10 dataset. Run the following from the `assignment2` directory:

```
cd cs231n/datasets
./get_datasets.sh
```

Compile the Cython extension: Convolutional Neural Networks require a very efficient implementation. We have implemented of the functionality using Cython; you will need to compile the Cython extension before you can run the code. From the `cs231n` 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 `assignment2` directory. If you are unfamiliar with IPython, you should read our IPython tutorial.

Working on Terminal

We have created a Terminal snapshot that is preconfigured for this assignment; you can find it here. Terminal allows you to work on the assignment from your browser. You can find a tutorial on how to use it here.

Submitting your work:

Whether you work on the assignment locally or using Terminal, once you are done working run the `collectSubmission.sh` script; this will produce a file called `assignment2.zip`. Upload this file to your dropbox on the coursework page for the course.

Q1: Two-layer Neural Network (30 points)

The IPython Notebook `two_layer_net.ipynb` will walk you through implementing a two-layer neural network on CIFAR-10. You will write a hard-coded 2-layer Neural Network, implement its backprop pass, and tune its hyperparameters.

Q2: Modular Neural Network (30 points)

The IPython Notebook `layers.ipynb` will walk you through a modular Neural Network implementation. You will implement the forward and backward passes of many different layer types, including convolution and pooling layers.

Q3: ConvNet on CIFAR-10 (40 points)

The IPython Notebook `convnet.ipynb` will walk you through the process of training a (shallow) convolutional neural network on CIFAR-10. It wil then be up to you to train the best network that you can.

Q4: Do something extra! (up to +20 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.

---

cs231n
cs231n
karpathy@cs.stanford.edu