CS4670/5670: Computer Vision, Fall 2015  
**Project 5: Convolutional Neural Networks**

**PA5 was adapted from an assignment for the Stanford course taught by**[**Andrej Karpathy and Fei-Fei Li**](http://cs231n.stanford.edu/)**.**

Brief

* **Assigned:** Wednesday, April 22, 2015
* **Code Due:** Monday, May 4, 2015 (by 9:00am)
* **Artifact:** No artifact
* This assignment should be done in teams of 2 students.

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

Getting started

**Get the code:**

git clone http://www.cs.cornell.edu/courses/cs4670/2015sp/projects/pa5/skeleton.git

This will create the directory skeleton. To get updates to the code you can then simply run git pull. Note: before diving into the code, we suggest reading the notebooks below to familiarize yourself with what you will be doing. The actual assignment is inside the notebooks; the notebooks will call code inside the cs4670 folder (which you will write).

**Getting set up on the virtual machine:** For the VirtualBox [virtual machine](https://cornell.box.com/s/rhth59tk4uxjvqnsxno15w9wwhd9zumh), there are only a few missing dependencies to install:

# Install missing dependency

sudo apt-get install libncurses-dev

# There's no need for a virtualenv on the virtual machine since there's nothing

but the assignments there. (expect this to take ~20min)

sudo pip install -r requirements.txt

**Getting set up on Mac OSX:** You're also welcome to use the virtual machine, but here are steps to getting this to work natively on OSX:

# Set up homebrew (if you don't already have it)

ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

# Install python and dependencies

brew install wget gcc pkg-config python libjpg libpng freetype

# fix a link problem

ln -s /usr/local/opt/freetype/include/freetype2 /usr/local/include/freetype

# \*\* IF NOT USING VIRTUALENV (see below) \*\*

# (expect this to take ~20min)

pip install -r requirements.txt

**Getting set up on Windows:** If someone gets it working on Windows and posts a walkthrough, we can include it here. Otherwise, we recommend using the VirtualBox [virtual machine](https://cornell.box.com/s/rhth59tk4uxjvqnsxno15w9wwhd9zumh).

**[Optional] Python virtual environment:** Once you have unzipped the starter code, you might want to create a [virtual environment](http://docs.python-guide.org/en/latest/dev/virtualenvs/) 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 pa5

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 (will take 20min)*

*# Work on the assignment for a while ...*

deactivate *# Exit the virtual environment*

If not using a virtual environment, run sudo pip install -r requirements.txt.

**Download data:** Once you have the starter code, you will need to download the CIFAR-10 dataset. Run the following from the pa5 directory:

cd cs4670/datasets

bash get\_datasets.sh

**Compile the Cython extension:** Convolutional Neural Networks require a very efficient implementation. We have implemented of the functionality using [Cython](http://cython.org/); you will need to compile the Cython extension before you can run the code. From the cs4670 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 pa5 directory. If you are unfamiliar with IPython, you should read our [IPython tutorial](http://cs231n.github.io/ipython-tutorial). Note that the command ipython notebook is the actual text you type; notebook is not the name of the notebook.

Submitting your work:

Once you are done working run the collectSubmission.sh script; this will produce a file called pa5.zip.

Q1: Two-layer Neural Network

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

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.

**Extra credit:** Implementing the conv\_backward\_naive function is extra credit.

Q3: ConvNet on CIFAR-10

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!

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.