Deep Learning for Advanced Robot Perception

Assignment 4: Defeating .99% Deep Learning error in MNIST final evaluation

Classify Handwritten Digits

Due Sunday, October 15th

This homework will provide the foundations for working with Convolutional Neural Networks. As a part of this homework you will:

- Download and load the MNIST dataset in Keras and develop a baseline neural network model for the problem.
- implement and evaluate a Convolutional Neural Network for MNIST.
- develop your Deep Learning model (reduce overfitting using Dropout Regularization)

The MNIST (Modify NIST) problem is a dataset developed by Yann LeCun, Corinna Cortes and Christopher Burges for evaluating machine learning models on the handwritten digit classification problem (http://yann.lecun.com/exdb/mnist/). The dataset was constructed from a number of scanned document datasets available from the National Institute of Standards and Technology (NIST).
Images of digits were taken from a variety of scanned documents, normalized in size and centered. This makes it an excellent dataset for evaluating models, allowing the developer to focus on the machine learning with very little data cleaning or preparation required. Each image is a 28x28 pixel square (784 pixels total). A standard split of the dataset is used to evaluate and compare models, where 60,000 images are used to train a model and a separate set of 10,000 images are used to test it.

The problem is cast as a digit recognition task. As such there are 10 digits (0 to 9) or 10 classes to predict. Results are reported using prediction error, which is nothing more than the inverted classification accuracy ((100-scores[1]*100)).
Excellent results achieve a prediction error of less than 1%. State-of-the-art prediction error of approximately 0.2% can be achieved with large Convolutional Neural Networks. There is a listing of the state-of-the-art results and links to the relevant papers on the MNIST and other datasets on http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

Loading the dataset

Make sure you are connected to the internet.

The Keras deep learning library provides a convenience method for loading the MNIST dataset. The dataset is downloaded automatically the first time this function is called and is stored in your home directory.

```
# load (downloaded if needed) the MNIST dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

Multi-Layer Perceptron (50%)

You will find an implementation of a Multi-Layer Perceptron Model solving this problem in Files/ Week 6/CNN MNIST/**mnist_mlp_baseline.py**. The MLP model should report a prediction error of approximately 1.70%

Your assignment consists in outperform this implementation and cross the barrier of 1% prediction error (less than 1%) by developing your own MLP model.

Your submission should consist in a python code (Keras+Theano) of your algorithm and a report with a detailed explanation about the development and performance of your model. The model that outperform all the models developed in our course will get extra credit (1 point).

Convolutional Neural Network (50%)

You will find an implementation of a Convolutional Neural Network Model solving this problem in Files/ Week 6/CNN MNIST/**mnist_cnn.py**. The CNN model should report a prediction error of approximately 1%.

Your assignment consists in outperform this implementation and cross the barrier of 1% prediction error (less than 1%) by developing your own CNN model.

Your submission should consist in a python code (Keras+Theano or Keras+TensorFlow) of your algorithm and a report with a detailed explanation about the development and performance of your model. The model that outperform all the models developed in our course will get extra credit (1 point).