YUSHENG JIN

7 DEKALB AVE, APT 20F, BROOKLYN, NY, 11201
(646) 468-8010 • king0133@live.com

OBJECTIVE

Seeking for a full-time Software Development Machine Learning or Data Science position starting immediately.

EDUCATION BACKGROUND

College of Staten Island — **New York, United States**
Master of Computer Science (GPA 3.875/4.0) — 01/2019-present
Hunter College — **New York, United States**
Master of Computer Science — 01/2018-01/2019
Harbin University of Science and Technology — **Harbin, China**
Bachelor of Engineering — 08.2010-06.2014

SKILLS

- Languages: C/C++, Python, MATLAB, HTML, Swift
- Frameworks and tools: Weka, TenserFlow, numpy, scikit-learn, matplotlib, pandas and so on

PROJECT EXPERIENCES

• DLMLP for Classification and Prediction Using Keras

•MNIST (Modified National Institute of Standards and Technology database) is a commonly used handwritten digit dataset consisting of 60,000 images in the training set and 10,000 images in the test set. The task is to train a machine learning algorithm to recognize a new sample from the test set correctly using feedforward Deep Learning Multilayered Perceptron (DLMLP).

- Use the Sequential model for building the network with two hidden layers.
- Plot the loss and accuracy curves.
- Add regularization to the model to prevent overfitting.
- Check performance after regularization.
- Getting the predicted class and probabilities.

• Image Classification with Convolutional Neural Networks (CNN)

•Dogs vs. Cats dataset will be used as image dataset. It contains 25,000 images of dogs and cats (12,500 from each class). Using only 4,000 pictures of cats and dogs (2,000 cats, 2,000 dogs). Split the set in the following way: 2,000 pictures for training; 1,000 for validation, and 1,000 for testing.

- Create a new one containing three subsets: a training set with 1,000 samples of each class, a validation set with 500 samples of each class, and a test set with 500 samples of each class.
- Use Sequential model and build the topology CNN implementing augmentation by including four convolutional layers Conv2D (with Relu activation), four MaxPooling2D layers and a Dense layer of size 1 and a sigmoid activation as the number of classes is two. Use RMSprop optimizer.
- Display some randomly augmented images (four images of two different cats and four images of two different dogs).
- To avoid overfitting add Dropout layer and do the training of CNN using data- augmentation generators. Plot training and validation accuracy as well as training and validation loss.
- Visualize every channel in every intermediate activation.
- Visualize convolutional filters: get the gradient of the loss with regard to the input, apply stochastic gradient descent, include a code for filter visualizations and generate a grid of all filter response patterns in a layer.

• Simulation: CPU scheduling algorithms comparison

• In this project, I would implement and evaluate the following four different CPU scheduling algorithms by writing a CPU simulator. First Come First Serve (FCFS); Shortest Job First (SJF)(Non-preemptive); Shortest Remaining Time Next (SRTN)(Preemptive); Roundrobin (RR).

• Given a set of processes to execute with CPU and I/O requirements, the CPU simulator should simulate the execution of these processes based on a given CPU scheduling policy. This simulation should collect the following statistics: the CPU utilization (NOT CPU efficiency), the total time required to execute the set of processes, and the service time (or CPU time), I/O time, and turnaround time for each individual process.