Zihao Wang

204 10th St, Apt 314, Jersey City, New Jersey, 07302

Mail:zw1074@nyu.edu, Website:zw1074.github.io
Phone: 551-(225)-9955

Summary

I am currently a candidate for Master of Science in Data Science, prior to which I had a Bachelor's degree in computational mathematics. To solve some huge PDE, I learned parallel programming by myself. Now, I am interested in CUDA C++, torch7 and deep learning. As for me, I am a quick learner and hard worker.

Education

- **New York University (CIMS)** — New York, USA
  Master of Data Science — *Sep. 2015 - Now*
  - **Expected graduation day**: Jun. 2017
  - **GPA**: 3.8/4.0
  - **Relevant Courses**
    Machine Learning, Deep Learning, Statistical Learning, Nature Language Processing, Inference and Graph Model, Time Theory, Logic
- **Nanjing University** — Nanjing, China
  Bachelor of Science, Computational Mathematics — *Sep. 2011–Jun. 2015*
  - **GPA**: 3.5/4.0
  - **Relevant Courses**
    Advanced Linear Algebra, Discrete mathematics, Numerical methods & experiments, Operations research, Mathematical modeling, Foundation of information theory, Numerical methods for PDEs, Foundations of Probability Theory, Stochastic processing, Foundation of Mathematical Statistics, Parallel Programming

Experience

- **Research Assistant Intern** — AIG Inc, NYC USA
  Science Group — *Aug. 2016 - Dec. 2016*
  - Contribute to the whole automatic damage appraisal project, especially for license plate detection and heat map generation of damage part.

Projects

- **Class Activation Map Improvement** — AIG Inc, USA
  Teamwork: response for designing and implement — *Oct. 2016*
  - Class activation map is the response heat map of class. It reflects which part of image the model focus on for getting the class. The current problem is that we need to add average pooling layer on the top of the model instead of fully connected layer to get the weight of each feature map. And also we want a more precise heat map.
  - I solve this restriction by using gradient method. For each class, calculate the correspond gradient of it and thus calculate the weight from the gradient. This method is called guided class activation map.
  - I implement this method on both tensorflow and theano platform and also modify some details for fitting our situation. I also design some experiments for testing its performance which shows that it works better than previous method on the same model.
- **License Plate Detection** — AIG Inc, USA
  Teamwork: response for designing and implement — *Aug. 2016*
  - Detect the license plate from a car image. There are two challenges. One is the image is quiet hard such as low resolution or high color contrast. The other one is that it requires fast processing.
  - To solve the question of hard image, I train a character-based classification. It only contains two layers of convolution neural net for reducing processing time. The training set is the combination of the global character training set and the background of the car image. With translation and rescaling, the classification can handle many hard situations.
  - To generate the candidate of the license plate from the saliency map generated from classification process, I use dilation to connect the nearby element and delete some members by its size and shape. Finally the top-5 error is 15% and top-3 error is 20%.
- **Efficient auto-encoder for physics particle collision event** — New York University, USA
  Teamwork: response for designing and implement — *Oct. 2016*
  - Use collision event data from CERN to produce an auto-encoder to compress data. The data is 3-D tensor while the index represent the location of the energy detector and the value represent the energy. The challenge is that this noisy data has 14400 dimension and we need to preserve the most relevant part and thus reduce the noise. The compress ratio is 32:14400.
  - Compare three compressors: Multilayer perceptron auto-encoder, convolutional auto-encoder and PCA. We evaluate them by two ways. One is calculating the reconstruction error and the other is apply reconstructed data in real application to see its performance.
  - Add threshold RELU on the last layer to make the output sparse. We find that multilayer perceptron is the best auto-encoder because it has over 0.95 AUC score between the reconstruction data and the original data. Also unlike PCA, it does not focus on the biggest value but indeed extract the hidden factor via our training process.
  - We use this technique to do anomaly detection. We compare the mean square error between the normal one and the abnormal one. And we find that multilayer perceptron performs best in this case.

- **Duplication Detection** New York University, USA
  Teamwork: response for designing and implement *May. 2016*
  - Use the data from health care system to predict possible duplication of information. The challenge is that the whole pair set is around 10^{11}, which is time consuming if we build the model on it. And it is also not a balanced dataset because in ground truth we only have around 120,000 pairs.
  - Construct an efficient parallel method to get a smaller set of interesting pairs, which we think that they are duplication. The processing time is 10 minutes using 8 workers on CPU. After this process, we reduce the pair set from 10^{11} to $3700k$. Then we generate a balanced training set by randomly select same number negative pairs (pairs that are not duplication) as the positive pairs (pairs that are duplication). Then we use a feature extractor to generate feature vector for each pair. Finally use random forest to make the prediction.
  - The smaller set of interesting pairs includes over 95% ground truth. And we finally get 99.4% accuracy with our fine tunning classifier.
- **Effective classification of STL-10** New York University, USA
  Teamwork: response for designing and implement *Mar. 2016*
  - STL-10 is a famous image processing database for testing semi-supervised learning containing 4000 training data, 1000 validation data, 8000 testing data and 100000 unlabeled data for 10 different class. The challenge of this dataset is that the training dataset is not enough for training compared to testing data.
  - We find a good initial kernel for first four CNN layers by applying k-means clustering to unlabeled data, which makes our accuracy improved to 76%.
  - We generate extra training data by applying some augmentation technology such as scaling, translation for balancing the size of training data and testing data. We find that when we augment twice for each training data, it performs best and finally get 78% accuracy.
- **Yelp Restaurant Rating Prediction** New York University, USA
  Teamwork: response for designing and implement *Dec. 2015*
  - Use the data from Yelp Dateset Challenge to fit different models. The challenge of this dataset is that the business attribute of the restaurant is not enough for well prediction so that combining the review as the additional feature is necessary.
  - Create a new model by tagging words of each review as adjective then apply Google pre-trained word2vec model which can improve the accuracy by 50%. Also evaluate the model by using AUC of the micro-ROC curve, which is equal to the probability that the confident score of true sample is higher than the score of false sample. For our model, the AUC/probability is 0.86.
- **Efficient Algorithm for Solving Tridiagonal Matrix** Personal, CN
  Personal: response for code and idea *Mar. 2014*
  - Convert the traditional tridiagonal problem to a recursive problem. Then using prefix algorithms finish the calculation.
  - The theoretical running time should be $O(\log n)$. In practice, it spends about 2s on GTX 970 when size is 2^{24}, and for compared to traditional LUdecomposition algorithm, it spends over hours on MATLAB.

Skills and Interest

- **Programming:** C++/C, SQL, CUDA, Python, Tensorflow, Theano, Torch7, MATLAB, Mathematica, LaTeX, Spark, Hadoop.
- **Interest:** Parallel Programming, Theorem and Application in NLP, Adversarial Network, Transfer Learning.