**Qiren Sun**

Tel: 1-6172597843 Email: sunqiren@bu.edu

**EDUCATION**

**School of Applied Physics, Beijing Institute of Technology** Sept 2013- Jun 2017

* Bachelor of Science in Physics, Department of Applied Physics
* Overall GPA:81/100

**College of Metropolitan, Boston University** Sept 2018- present

* Master of Computer Science, Department of Computer Science
* Overall GPA 4.0/4.0

**RESAERCH & PROJECT EXPERIENCE**

**CycleGAN in Pytorch**  Jun 2018 – Sept 2018

2018

* The idea of CycleGAN comes from unpaired image-to-image translation using cycle-consistent adversarial networks which are written by Jun-Yan Zhu, etc.. Using a set of images of Yosemite national park taken either during the summer of winter and aimed to generate the pictures of summer set but have the feature of the images in winter dataset.
* A Cycle has two discriminators and two generators. The inputs of the discriminator are 128\*128\*3 tensor image. The two generators are made of an encoder and a decoder.
* In the latent space, compressing the 128\*128\*3 tensor image into a feature representation and pass it through the residual block (something like ResNet50 for image classification). Training part transforms the image style from domain X to domain Y or from Y to X. Implementing the CycleGAN through Pytorch.

**Drum Track Generator in Tensorflow** Jan 2018 – Apr 2019

2019

* Drum track generator produces drum track which based on the input drum track. The generator model based on RNN (LSTM) and the idea comes from Coconet which developed by the Google AI team. One of the crucial keys is to build the time step which limits the note to the specific position and extend the drum track to the time step.
* To obtain the data type which can be input into the RNN model, transforming the input midi file to the standard note sequence and convert the standard note sequence to embedding space (event sequence) through the encoder. This model uses the beam search algorithm, and the beam contains the beam size (=1) sequence with the highest score. The returns which include the sequence with the highest likelihood will feed to the model.
* Implemented CudnnLSTM (train) and CudnnCompatibleLSTMCell (generate) because the training part and generator part have different tensor shape. After decoding the event sequence, the returns (inputs batch) are fed into the generator and predict the next event for every sequence. Getting the extend event sequence after taking the list of event sequence and passing the input batch through the softmax function (changes the similarity between input and generate part).

**Quantum dot solar cell in the ISCAS** Jul 2016 – Sept 2016

2016

*Institute of Semiconductors of Chinese Academy of Sciences| Research Assistant*

* Design and manufacture of quantum dot solar cells
* Proposed to change the structure of the solar cell and carry out the plasmons on the surface of the quantum dot solar cell
* Aimed to shorten the distance between quantum dots and improve the efficiency of photoelectric conversion
* Carried out experiments and obtained the photoelectric conversion efficiency higher than the original solar cell structure

**Generation of jewelry models through machine learning** May 2017 – Aug 2017

2017

*The application of deep learning in the traditional jewelry industry at Shanghai Jiaotong University*

* Set mechanical parameters to meet the physical architecture through Auto CAD
* Set parameters to achieve the physical architecture due to the automatic generation of jewelry models through machine learning and GAN
* Signed an agreement with Chinese Gold and Zhou Dafu for software delivery, which is of great potential value

**INTERNSHIP**

**Dream workshop** Jan 2017-Apr 2017

*Testing engineer*

* This website aims to offer users with the latest news of the company and collect the reviews of the users after they purchase the product to do an analysis. Website building based on Django, MySQL, and AWS
* My job is testing the models, forms, and views in Django, deploying the machine learning model on AWS sagemaker and, creating endpoint collect the data from users. The ML model uses the XGBoost (sklearn) algorithm.
* Inspected the Shenzhen factory and designed software that can be industrialized
* Planned a variety of application development to realize the user's experience after the ML model analyze the reviews of the users to the product they bought

**Lenovo** Feb 2018- Jun 2018

2018

*Product assistant, department of the product innovation*

* The job of the Lenovo product innovation team is to create a new type of electronic products, and the project I participated in is smart speaker development.
* My work is to test the 6 (+1) ring microphone array in different situations. Adjusted the distance and path between microphone and speaker through mathematics to make the acoustic echo cancellation (AEC).
* Participated in the design of the speaker through market research. Implemented turing robot (a platform to do the semantic recognition and DeepQA) through API and using turing robot to answer the user’s question if cannot be recognized.
* Based on the NLTK Framenet structure to process the text data and build a personal corpus.

**COURSES PROJECT**

**Generating the human face through Deep Convolutional GAN**

* The DCGAN trained on the Street View House Number Dataset (SVHN) and planned to generate the RGB number pictures.
* The discriminator is a convolutional classifier and without any pooling layers. We generated some RGB pictures which can be recognized. Implemented the DCGAN through Pytorch.
* The DCGAN trained on the CelebFaces Attributes Dataset (CelebA) and aimed to generate the human face.
* The structure of the network is almost the same as the above DCGAN, except that I initialized the weights of the convolutional and linear layers.
* The DCGAN can generate human face with the 32\*32\*3 tensor images, but it sometimes will create the human face with the noise. Implemented the DCGAN through Pytorch.

**Information Extraction based on Web Crawler in Python**

* This project collected random websites (wiki text) through crawler and used the 2-grams model to obtain high-frequency vocabulary.
* An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of an (n − 1)-order Markov model. A Markov chain is a stochastic model describing a series of possible events in which the probability of each occurrence depends only on the state attained in the previous event.
* Both the n-gram model and Markov chain model are useful in NLP(natural language processing). This project chose the highest frequency vocabulary that occurred in the 2-grams model to do the information extraction.

**Amazon Database in Microsoft SQL Server**

* The amazon database based on the relational database and designed through the ER diagram.
* Creating the tables, and constraints which are consistent with my ERD design. Creating data insert scripts (products, customers, etc.). Implemented the stored procedure which allows the seller to add a new product and change details of an existing product. Achieved the aggregation and the subquery to operate the Amazon database.

**A dog identification app through CNN**

* Creating a CNN to classify dog breeds used transfer learning.
* Implemented the densenet121 to pre-train my CNN and got the result accuracy 86%. Implementing the CNN through Pytorch.

**Sentiment Prediction RNN in Pytorch**

* Implemented a recurrent neural network that performs sentiment analysis, and they used a dataset of movie reviews.
* The RNN used the embedding layer with the Skip-gram Word2Vec model, LSTM, and sigmoid output layer.
* The embedding layer can get a more efficient representation than one-hot encoded vectors. This sentiment prediction RNN accompanied by sentiment labels (positive or negative), and the final accuracy of the model is 80%.

**TV Scripts Generation in Pytorch**

* **Implemented the RNN** generates the Seinfeld TV scripts and I use part of the Seinfeld dataset of scripts from 9 seasons. I construct a LSTM model with PyTorch**.**

**Predicting Bike-Sharing Patterns, Sentiment Analysis Network, and Student Admissions**

* These three projects used the MLP thought and I used the mathematics knowledge to build the neural network.

**LEADERSHIP & ACITVITIES**

**Physics School basketball team captain**  Sept 2014-Jul 2016

* Organized trainings and joined basketball matches

**Co-society Minister of Practice** Sept 2014-Jul 2016

* Organized activities
* Made plans for association activities

**Focus & Follow**

* Conferences (NIPS, AAAI, CVPR, EMNLP, etc.)
* GitHub (https://github.com/QirenSun)
* PyTorch Community, Google Developer Group, Deeplearning.ai, arXiv.org, Experiments with Google, etc..

**SKILLS & INTERESTS**

**Programming Language**: C/C++, Pyhon, Html5, and SQL

**Software:** AWS sagemaker, Matlab, Colab, Jupyter Notebook,Weka, MySQL, and Auto CAD

**Interests:** NLP related topic (encoder, decoder, information extraction, long term memory in RNN, etc.), play the drums, combing music and AI