Manish Kumar **Keshri**

(612)298-0689 | keshr005@umn.edu | ManishKumarKeshri | manish-kumar-keshri-b094158b

Education

University of Minnesota, Twin Cities *MN, U.S.A*

M.S. in Data Science (GPA:3.74/4.00) *Sep 2017 - May 2019*

Courses: Artificial Intelligence, Statistical Analysis, Data Mining, Machine Learning, Applied Regression, Advanced Algorithms, Introduction to Machine Learning, Principles of Database Systems, Dispersed Data-driven Computing, Applied Parallel Programming, Applied Multivariate Methods, Distributed Systems, **NLP for protein classification(Major Project)**

National Institute of Technology, Tiruchirappalli *TN, India*

B.Tech in Electrical and Electronics Engineering (GPA:8.5/10.0) *Jun 2012 - May 2016*

Courses: Data structures and algorithms, Operating systems, Computer Architecture, Fuzzy logic and genetic algorithms, Numerical methods, Microprocessors and Microcontrollers

Deeplearning.ai, Coursera *Online*

Deep Learning Specialization(Grade: 95/100) *Dec 2017 - Mar 2018*

Courses: Neural Networks and Deep Learning, Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization, Structuring Machine Learning Projects, Convolutional Neural Networks, Sequence Models

Technical Skills

C, C++, Python, R, SQL, Verilog, TensorFlow, Keras, Scikit-Learn, Numpy, Scipy, Git, CUDA, MATLAB, Machine/Reinforcement/Deep Learning, Data Mining, Statistics, Algorithms, Parallel Programming

Experience

University of Minnesota *Twin Cities, MN, USA*

Teaching Assistant *Sep 2018 - Present*

- TA for Course CSCI 4041 Algorithms and Data Structures (Fall 2018)
- TA for Course CSCI 4041 Algorithms and Data Structures (Spring 2019)

Synopsys Inc. *Mountain View, CA, USA*

R&D Intern(Machine Learning) *May 2018 - Aug 2018*

- Test sequence generation: Implemented Policy Gradient with **LSTM Network, SeqGAN and Genetic Algorithm** to generate sequences such that the test coverage is maximized. Grouping of tests by their coverage areas using DBSCAN, and Hierarchical clustering.
- Test selection using **reinforcement learning**: Formulated test subset selection as a Multiagent coordination problem and implemented Multiagent Reinforcement Learning algorithm with reward shaping. Languages Used: Python, C++, System Verilog

SanDisk, A WD Brand *Bengaluru, India*

Systems Software Engineer *Jul 2016 - Aug 2017*

- Data analysis in characterization tests of NAND Flash memory dies (Used: Python, R)
- Firmware verification of command sequences for the successful operation of NAND Flash memory (Used: C)
- Presented a **paper on use of Machine Learning** in the flash memory product flow to improve yield and reliability of memory dies.

Texas Instruments *Bengaluru, India*

Intern *May 2015 - Jun 2015*

- Implemented FPGA-based architecture simulator for Serial Peripheral Interface (SPI) (Used: C)

Research

- **NLP for Protein Classification**: Learning protein sequence embeddings using its structural information and using it in the prediction of global structural similarity between protein sequences as defined by shared membership in the SCOP(Structural Classification of Proteins database) hierarchy.

Projects

- **LSTM-based Startup name and poem generation**: Character and word level language modeling with deep RNN-LSTM network to learn from a corpus of names/text and generate similar ones. At each time step during training, the input character/word to the LSTM is mapped to the next character/word in the sequence.

- **NLP system to emojify a sentence using RNN-LSTM and GloVe Word Embeddings**: Built an NLP system using RNN with LSTM units to assign an emoji to a sentence. Used 50-dimensional GloVe Word Embedding as the features for sentence words and emojis were assigned using 5 directional SoftMax output.
- **CUDA-enabled Parallel Implementation of Collaborative Filtering**: Achieved speed-up of 402x in similarity matrix calculation on NVIDIA GTX 1080; used coalescing, thread coarsening and shared memory tiling.
- **Finding Fraud in Enron Email Data**: Explored dataset with 146 data points (i.e. ”employees of Enron”), each of which has 21 features(i.e. email info, salary, compensation etc.) to find the fraud. Implemented features selection, outliers removal, classification, and validation. Compared confusion matrices and F1 scores of Decision tree, SVM and AdaBoost classifiers (Used: Python)
- **Pattern Mining in transactional database**: Implemented FP Tree algorithm to find frequently purchased itemsets and corresponding association rules in more than 63000 transactions using C++
  - Compared results for different values of minimum support and confidence.
- **Artistic image generation using deep learning and Neural Style Transfer(NST) algorithm** : The idea is to compose images in the style of another image. Implemented NST algorithm which uses transfer learning from a pre-trained VGG-19 layer CNN model to get hidden layer activations. The overall cost was reduced by updating the pixel values of the generated image.
- **Car detection application for autonomous driving using YOLO algorithm**: Implemented YOLO architecture which takes input image of 608x608 passes and encodes it into 19x19 grid with information about 5 anchor boxes. Predicts bounding boxes and class probabilities directly from full images in one evaluation.
- **Geo-distributed Recommender System**: Developed a geo-distributed recommender system using Decentralized-SVD algorithm. Acheived over 90% reduction in data transfer without decrease in accuracy.
- **Instance Segmentation - Nuclei finding and Segmentation**: Trained Mask R-CNN for object instance segmentation by adapting the existing model configurations to detect small nuclei in images with varying size and modality. Used Resnet50 as a backbone encoder and Adam as an optimizer for Mask R-CNN.
- **Image compression with EM and K-Means clustering Algorithms**: Implemented EM and K-Means Algorithms to cluster the pixels in an image. RGB encoding to reduce the number of colors from 16,777,216 in a normal 24-bit color image to K = 16 and 256 colors. Algorithms select best K colors to represent the compressed image.
- **MNIST digit classification using SVM**: Implemented SVM to classify MNIST digit dataset.