Sajeed Bakht

Sajeed.bakht@mail.utoronto.ca

289-680-8291

Experience

MediaForce,

Machine Learning Engineer/Researcher   **Jan-2018 to Dec 2019, Ottawa, Ontario**

- I worked in a team of two, that was responsible for researching and developing machine learning products in Marketing. The roles that I was responsible for and not limited to was:
- Researching into Machine Learning Techniques such as collaborative filtering, content similarity, deep learning techniques such as Long-Term-Short-Term Memory Networks, embeddings, convolutional networks, and transfer learning.
- Developing and Serving Machine Learning products using modern machine learning languages and cloud technology tools. These products included and not limited: Recommendation Engines, Conversational Agents, and Computer Vision
- Presented and defended research for the Scientific Research and Experimental Development Tax Incentive Program (SR&ED)

Education

University of Toronto,   Grad 2020   Toronto Ontario

- Honours Bachelor of Science
- Major Statistics with a Specialization in Machine Learning and Data Mining
- 75% GPA over last 2 years

Technical Skills

Experience in Machine Learning, Statistical Analysis, Software Engineering, and Cloud Technology Tools:

- **Data Science/Machine Learning** – TensorFlow,Keras, R, Python with Numpy, Pandas, Tensorflow Transform, TF-Records,OpenCV
- **Software Engineering Tools:** FireBase, ExpressJS, NodeJS,Selenium,Flask,
- **Google Cloud Technologies**: Compute Engine, Storage,SQL, BigQuery,Machine Learning API,DataFlow,Tensorflow Hub

Notable Projects:

License Plate Recognition:

Developed by myself. This program would take a picture of a car and extract the license plate number. First it would crop the license plate out and then take the license plate number using Deep Learning techniques in TensorFlow and Keras. What made this project notable was that labels were a sequence of variable length. For e.g. some license plates have 5 characters, while others were 8 characters. To account for this, I firstly took a pre trained convolutional network from TensorFlow Hub, and connected it to a Recurrent Neural Network. I did this connection so I could output variable length output. I then retrained, the network using photos of license plates that I gathered.

Blog Similarity:

Developed by myself. This program would take as input a body of blogs and compute the most similar blogs. Firstly, the blogs were read in from file format in Python. The words were cleaned and "stop words" were removed. Stop words are commonly used words such as "the", and "and". This was to remove noise from our distance measurement. The blogs were then each represented as their own dictionaries which word as the **key** and frequency of that word as the **value**. The distance measurement was if two blogs shared the same word then we would increment the similarity value. In the end, each blog had a similarity distance to every other blog, stored in a hash map. So given a blog I would be able to return the top 5 most similar blogs. Later, the program would also be able to use web scripting to scan a website for new blogs and recompute the distance measurements.