Ji, Chao

ji.chao.stern@gmail.com githut.com/chao-ji https://chao-ji.github.io

10 years of experience & expertise in Machine Learning and extensive hands-on experience in building and experimenting Deep Learning models on public datasets. Looking for full-time opportunities in Machine Learning/Deep Learning based in China.

Education

Indiana University, Bloomington **Bloomington, IN, USA**
Doctor in Philosophy *May, 2016*
Informatics

Beihang University **Beijing, China**
Bachelor of Engineering *July 2007*
Software Engineering

Experience

Indiana University **Bloomington, IN**
Visiting Research Scholar *August 2016 - July 2017*

- Worked on Data Science projects involving a large real world dataset containing users' posts from `reddit.com`
- Applied a multitude of NLP and Machine Learning techniques to characterize the polarization between two conflicting subreddits by analyzing the semantic content of user's posts.

Indiana University **Bloomington, IN**
Research Associate *August 2008 - May 2016*

- Proposed and developed multiple statistical machine learning algorithms for protein quantification and peptide identification problems in mass spectrometry based computational proteomics.

PROJECTS

Building and experimenting with Deep Learning models

NumPy-based Framework for Automatic Differentiation

- A framework that provides Python program interfaces for defining and running the forward/backward passes of a computational graph to train neural networks. `https://github.com/chao-ji/np-auto-diff`

Word2Vec: Learning Word Representation

- Implemented Word2Vec in TensorFlow. Provides options to define the training task as either Continuous Bag-Of-Words or Skip-Gram, and performs training using either Negative-Sampling or Hierachical Softmax. `https://github.com/chao-ji/tf-word2vec`

Doc2Vec: Learning Paragraph/Document Representation

- Implemented Doc2Vec in TensorFlow. Provides options to define the training task as either Distributed Memory or Distributed Bag-Of-Words, and performs training using either Negative-Sampling or Hierarchical Softmax. Evaluated the model by training on the IMDB dataset to obtain vectors of unlabeled movie reviews, and comparing the classification accuracy based on document vectors vs. the accuracy based on the word-count feature vectors. `https://github.com/chao-ji/tf-doc2vec`

RNN Language Model

- Implemented a multi-layer RNN Language Model in TensorFlow. Provides options to train and evaluate the model on the character level or on the word level. Evaluated on the PennTreeBank dataset in terms of word-level Perplexity. `https://github.com/chao-ji/tf-RNN-Language-Model`

Neural Machine Translation with Attention Mechanism

- Implemented a Neural Machine Translation system in TensorFlow using the Encoder-Decoder architecture with Attention Mechanism. Evaluated the model on English-Viatnamese translation task (BLEU score 25.9, IWSLT13 dataset) and some generic sequence transduction tasks (i.e. sorting). `https://github.com/chao-ji/tf-seq2seq`

ResNet for classifying CIFAR10 images

- A lightweight TensorFlow implementation of the Residual Networks and the counterpart Plain Networks with no residual connections for classifying images in CIFAR10 dataset. Reproduced the results presented in the original paper. `https://github.com/chao-ji/tf-resnet-cifar10`

DeepLab for Semantic Segmentation

- TensorFlow implementation of DeepLabv3 plus (the latest incarnation of DeepLab models) to generate pixelwise semantic masks for images. Provides options to choose from a selection of feature extractors (e.g. ResNets, MobileNets) for the desired speed-accuracy trade-off. Evaluated on PASCAL VOC 2012 validation set (mIOU=0.765). `https://github.com/chao-ji/tf-deeplabv3`

Object Detection Models

- TensorFlow Implementation of *multiple* Object Detection Models (currently Single Shot Detector and Faster R-CNN) in a *single* framework by abstracting out common sub-tasks such as anchor box generator, anchor-wise target assignment, non-maximum suppression, etc. to make them reusable across different models. Evaluated on PASCAL VOC 2007 test set (mAP@.5=0.788 for Faster RCNN w/ ResNet) and COCO 2017 val set(mAP@[.5, .95]=0.275 for Faster RCNN w/ Inceptionv2). `https://github.com/chao-ji/detection`

Past Data Science projects

Collaborative filtering algorithm based on latent factor models

- Implemented latent factor model based collaborative filtering algorithm using TensorFlow. Tested and evaluated performance (RMSE = 0.85) on MovieLens dataset. Performed user profiling and categorization based on learned user embeddings.

Semantic modeling of subreddit polarization

- Extracted user posts from two subreddits (r/MensRights and r/Feminism) with presumably polarized sentiment. Identified words/phrases that appear statistically more often in one subreddit versus the other. Run doc2vec model to obtain vectors of user posts and vectors of the subreddit-biased words/phrases. Found evidence of polarization by showing that the user posts that are semantically more similar to the words biased towards the "home" subreddit received higher scores (i.e. upvotes minus downvotes).

Predict the amount of insurance claims

- Ran truncated SVD to reduce the dimensionality of a high-dimensional dataset. Used cross-validation to determine the optimal number of features. Led to improved accuracy of prediction.

Predict of likelihood of fraudulent credit card transaction

- Addressed the class imbalance problem (the majority of transactions are negatives, i.e. not fraudulent) by up-weighting the penalties for misclassified positives relative to the negatives.

Past work on applying machine learning in science domain

A k-neighbor-based approach for predicting peptide fragmentation spectra

- Proposed an algorithm that predicts the target value (the "y") of a function by (weighted) averaging the target values of other data points (other "x"'s) that are similar to the given input (the "x") of the function. The similarity (hence the weighting) of data points is determined by a binary classifier that predicts a "soft" class label (between 0 and 1), which was trained on *pairs* of data points labeled as either similar or not similar.

Latent variable based algorithm for estimating protein quantity

- Designed an iterative algorithm that estimates the target value (protein quantity, Q) based on an unknown latent variable (peptide response rate, R). Q is a function of R and can be directly computed if R were known. On the other hand R is a different function of Q and can be computed if Q were known. Alternatively R can be predicted from a feature vector (X) extracted from an entity (peptide sequence) using a regression model. The algorithm repeats the following steps. In Step-1 an initial random value of Q is used to generate target values of R to train a regressor. In Step-2 the regressor predicts the estimated value of R from X. In Step-3 an updated value of Q is computed from the estimated value of R.

Probability-based model for scoring peptide-spectrum matches of cross-linked peptides

- Designed a supervised learning model that predicts a probability score (S) given a structure-valued input (a set). Uses logistic regression outputs to approximate conditional probabilities given elements of the input. Formulates the probability score as a function of the conditional probabilities under a probability framework.

TECH SKILLS

- **Programming languages**: Python, C, Java
- **Frameworks and Tools**: TensorFlow, NumPy, scikit-learn, Protocol Buffer, Git

SELECTED PUBLICATIONS

- **Ji C**, Arnold RJ, Sokoloski KJ, Hardy RW, Tang H, Radivojac P, Extending the coverage of spectral libraries: a neighbor-based approach to predicting intensities of peptide fragmentation spectra. *Proteomics* 2013, 13, 756-765.
- **Ji C**, Li YF, Bellinger EP, Li S, Arnold RJ, Radivojac P, Tang H, A maximum-likelihood approach to absolute protein quantification in mass spectrometry. Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics 2015, 296-305.
- **Ji C**, Li S, Reilly JP, Radivojac P, Tang H, XLSearch: a probabilistic database search algorithm for identifying cross-linked peptides. *J. of Proteome Res.* 2016, 15(6), 1830-1841.