CSA 250: Deep Learning
(Spring 2020)

Sargur N. Srihari
Indian Institute of Science,
Bengaluru, Karnataka
email : srihari@buffalo.edu

February 25, 2020

1 Problem Statement

In this project, we would develop machine learning models on text input. To feed in text as input to a model, we need to convert each atomic discrete entity in the input (words/characters) into real vectors (R^d) so that their semantics are captured meaningfully. This mapping from text to real numbers is called feature extraction which can be either fixed by loose assumptions like bag of words, IID etc. or learnt by the model itself through a carefully crafted loss function. Unlike images that have spatial dependencies, text has temporal dependencies i.e. a word at position t may depend on words at position $t \pm k, k \geq 0$. In project 2, we used CNN which is good in capturing spatial dependencies and in this project we would explore RNNs that can handle long term temporal dependencies.

In particular, we would work with the task of Natural Language Inference. In this task, we would be given two sentences called premise and hypothesis. We should determine whether the "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given that the "premise" is true.

2 Task

The aim of the project is to implement a deep learning network for the task of Natural Language Inference. Given two sentences, we will have to predict if the sentence pair constitutes entailment/contradiction/neutral. Following are the tasks to be performed.

1. Build a simple Logistic regression classifier using TF-IDF features.
2. Build a Deep model specific for text like RNN/GRU/LSTM or any of the new approaches like Transformers, BERT etc for NLI.

3 Dataset

For this project, we would use one of the standard datasets for this task called **S**tanford **N**atural **L**anguage **I**nference (SNLI) [1]. The SNLI dataset can be downloaded from the link [2]. Once we unzip thew dataset, we would find a ReadME.txt file which gives a detailed description about the dataset. For this project we would primarily use the files snli_1.0_train.jsonl for training the model and snli_1.0_test.jsonl for evaluating the performance of the model. In each of these files, the relevant fields to be considered are gold_label, sentence1 and sentence2.

Field	**Description**
gold_label	represents the target label. Entailment means that the two sentences agree with each other, contradiction means that the two sentences contradict each other and neutral means that the two sentences neither entail nor contradict each other.
sentence1	This is the first sentence of the pair. This is also called premise.
sentence2	This is the second sentence in the pair. This is also called hypothesis.

Here are a few examples from the dataset.

SNLI dataset		
premise	**hypothesis**	**gold_label**
A man inspects the uniform of a figure in some East Asian country.	The man is sleeping	contradiction
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral
A soccer game with multiple males playing.	Some men are playing a sport.	entailment

Before training the model, it is a standard practice in NLP to do text pre-processing which involves, but not limited to :

- conversion to lower case
- stop words removal
- stemming
- tokenisation
- special character removal, if required
- HTML tags processing
- Including UNK token to handle out-of-vocabulary words

[1] https://nlp.stanford.edu/projects/snli/

[2] https://nlp.stanford.edu/projects/snli/snli_1.0.zip

One may use a standard python library for text processing called NLTK [3] for text pre processing tasks.

4 Plan of Work

1. **Train Logistic regression using TF-IDF features** using scikit in python
2. **Train using Deep network** with high level Neural Network library
3. **Tune hyper-parameters:** For steps 1 and 2: Use a part of the training set as validation set for tuning the model. Tune your hyper parameters for higher validation accuracy. One may try grid search , random search or any other well known techniques for hyper parameter tuning.
4. **Try different architectures:** For step 2 : We would recommend you to try different architectures of Neural networds mentioned earlier. Major hyper parameters that one should be tuning for Neural nets are learning rates, choice of optimizers, number of layers, number of units in each layer, regularization methods. Document your observation on these hyper parameters.
5. **Test your machine learning scheme on the testing set:** For steps 1 and 2: Once you finalize the hyper parameters, predict the labels for each of the sentence pairs in the test set (snli_1.0_test.jsonl) and submit your results.

5 Evaluation

1. Plot graph of training loss and validation loss vs number of epochs while training on each classifier.
2. For each classifier, we would evaluate your solution on the test set using classification accuracy:

$$Accuracy = \frac{N_{correct}}{N} \tag{1}$$

Where $N_{correct}$ is the number of corrected classified data samples, and N is the total number of samples of the test set.

6 Deliverables

The following documents/codes must be submitted:

1. Prepare a report and name it **Deep Learning Report 3.pdf**. In your report, you need to briefly describe what you have done, present the results (in a form you think is good) and provide a brief discussion of the results you obtained. Please feel free to play with the model architecture and hyper-parameter settings. Please make it concise and to-the-point.

[3] https://www.nltk.org/

2. You need to save the trained model in a respective format and put it in model directory.

3. Python code for training.

4. Submit the outputs generated by your classifiers in the files "tfidf.txt" and "deep_model.txt" containing the predictions of TF-IDF and deep models respectively. The output file should contain one line for each sentence pair in the test data. Print entailment/contradiction/neutral in each line depending on the model's prediction.

The submission instructions will be provided in a separate document at a later stage. Please check piazza periodically for updates.