MSDS 458

Assignment #3

Setu Madhavi Namburu

1. **Abstract:** A start up AI firm is interested in building conversational bots and provide models as service to clients to work with their corpus in their knowledge management systems. In the initial phase, the firm is interested to build capabilities using open data sources – so Reuters newswire articles dataset is chosen for building automatic context-driven classification system. A data scientist is tasked with building deep learning models using Keras library in python and present the management with findings and recommendations from this experimental research. The study is specifically targeted to understand how different language models and memory based deep learning models work and compare/contrast their performance with Dense and Convolutional neural networks.
2. **Introduction:** A start up AI firm would like to build conversational bots (question/answer, querying systems) to help companies make better use of their knowledge management systems. Clients can realize the efficiencies/gains quickly only if the existing text data can be used without major redesign to those systems. A data scientist is tasked with a research question if black box models can be built using text data without lot of efforts into manual language modeling. Widely used Reuters newswires data is chosen as a corpus to build proof of concept for automatic context driven text classification system. The analyst is specifically asked to conduct experiments with different language models & memory based deep neural networks (using Keras library in python) and compare/contrast their performance with Dense and Convolutional neural networks. The goal of this experimental research is to identify and document technical nuances and caveats in working with deep learning models for NLP applications, specifically text classification.

2.1 Data Description: The Reuters dataset, a set of short newswires and their topics, was published by Reuters in 1986. It's a very simple, widely used toy dataset for text classification. This is a dataset of 11,228 newswires from Reuters, labeled over 46 topics. Some topics are more represented than others, but each topic has at least 10 examples in the training set. "Each newswire is encoded as a list

of word indexes (integers). For convenience, words are indexed by overall frequency in the dataset, so that for instance the integer "3" encodes the 3rd most frequent word in the data.

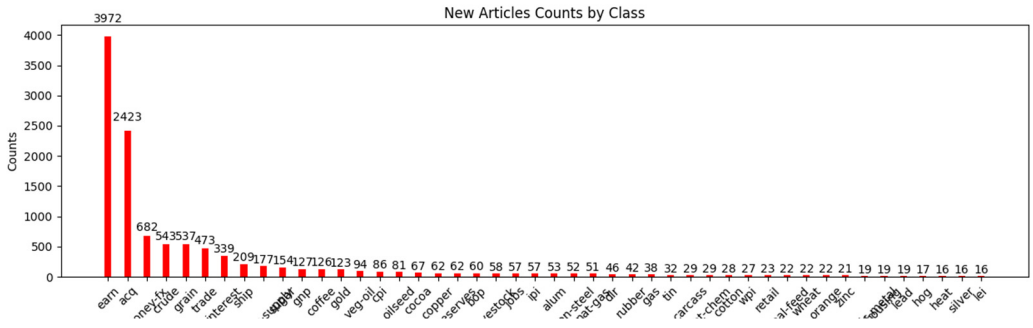

Figure 1. Counts of Newswire articles in each category

Figure 1 shows the count of documents in each of 46 classes, as observed it is a very imbalanced data set with two dominating topics (earn, acq). It is not clear if the data was manually labeled by someone or if it was automatically labeled using some heuristic rules as the content & the label do not seem to make perfect sense by examining random article as shown below (the article has wheat and corn but is labeled as grain whereas wheat has its own category). For the purpose of this research, we will assume that the labels are accurate and will be used in this supervised modeling.

```
Reuters News Article 31:
? the u s agriculture department said private u s exporters reported sales
of 200 000 tonnes of wheat to jordan 300 000 tonnes of soybean meal to irag
and 100 000 tonnes of corn to algeria the wheat for jordan includes 165 000
tonnes of hard red winter and 35 000 tonnes of soft red winter and is for
delivery during the 1987 88 marketing year the soybean meal sales to irag
includes 180 000 tonnes for delivery during the 1986 87 season and 120 000
tonnes during the 1987 88 season the department said the 100 000 tonnes of
corn sales to algeria are for delivery during the 1986 87 season it said
the marketing year for wheat begins june 1 corn september 1 and soybean meal
october 1 reuter 3

Topic: grain
```

2.2 Exploratory Data Analysis (EDA): The corpus of 11,228 labelled articles have 1638886 total words and a vocabulary of 30980 unique words. Of these, 10305 words just occurred once in the entire corpus

and top 25 words constitute 30% of total word count. Below table summarizes these words stats from the corpus.

Word Frequency	Description	Count	Percentage
Top 25 Words	High Frequency	499,067	0.30451
Bottom 25 Words	Low Frequency	10,305	0.00628
Unique Words	Unique	30,398	0.01890

Table 1. word stats of Reuters corpus

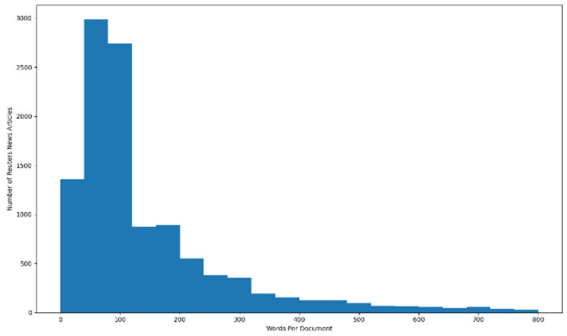

Figure 2. Word distribution among articles

Majority of the news articles have less than 300 words as shown in Figure 2 representing right skewed distribution. The words range from 2 to 2376 per article raising some data quality concerns. Table 2 shows top ten high frequent words which can be treated as stop words. Number of words that occurred more than 5 times are less than 1000 (e.g., 930 words occurred 6 times in in the entire corpus). To be consistent among all the experiments, top 25 high frequent words are excluded from analysis and each article is truncated after 300 words as a sequence cut off. If the length of the article is less than 300 words, it is padded with zeros.

word	frequency	freq	num of words
in	82723	1	10305
said	42393	2	4412
and	40350	3	2465
a	33157	4	1707
mln	29978	5	1093
3	29956	6	930
for	29581	7	685
vs	20141	8	578
dlrs	16668	9	451
it	15224	10	380

Table 2: High frequency words, Number of words in top ten frequencies

2.2.1 One-Hot Encoding of words: Before going into word embeddings for language modeling, top 1000 unique words are used as vocabulary after ignoring the top 25 high frequent words (based on learnings from table 2). One hot encoding is used to represent the corpus with documents x words matrix with binary representation.

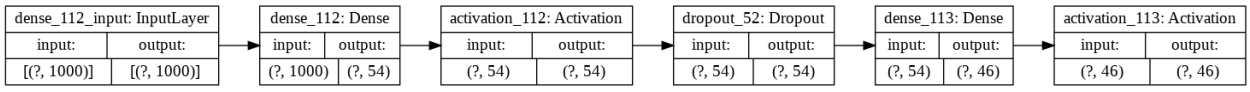

Figure 3. Simple one-hot DNN model

A simple DNN model (Figure 3) with 50% drop out layer is used to train the network with one-hot encoded vectors to understand how separable these documents are by visualizing the activations using T-SNE.

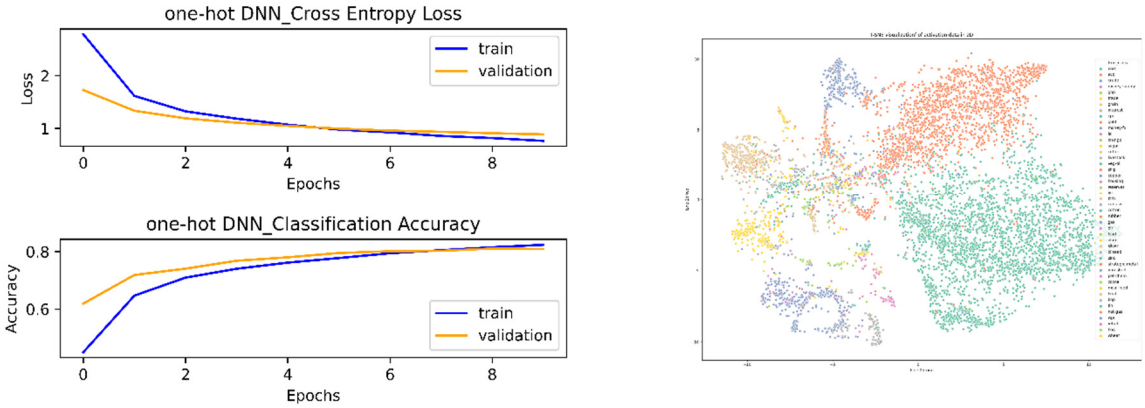

Figure 4. Performance of simple DNN on One Hot vectors

The test accuracy was about 79% from this model and the loss and accuracy plots do not show any over/underfitting. T-SNE visualization shows two dominant clusters (earn, acq) while rest of the categories show multiple small clusters (with several of them overlapping). While the results are not surprising (i.e., not being able to achieve high accuracy) given heavy class imbalance, it is quite encouraging to see how well this simple model is able to capture some of the clusters without much manual text processing upfront. In further experiments, different network topologies will be tested using word embeddings to see if incorporating context can improve the performance.

3. **Literature Review:** NLP applications are major driving force in Deep learning breakthroughs in history. While many predict that building an AI system that is as intelligent as human, might be the end of human race, the answer can only be unfolded while we get there. GPT-3 [2] is the recent language generator created by OpenAI that can change the way AI works going forward. GPT-3 was trained with 175 billion parameters & the model's size is 700GB and costs well over $4.6M to train it and uses transformers technology (attention base mechanism to assign weights in a sequence unlike memory based LSTM networks). While several 'Aha' moments happened in the last 2 months, several articles [3] are emerging about ethical concerns & the bias that can get incorporated if the text is fed without pre-processing (Table 3).

Table 6.1: Most Biased Descriptive Words in 175B Model

Top 10 Most Biased Male Descriptive Words with Raw Co-Occurrence Counts	Top 10 Most Biased Female Descriptive Words with Raw Co-Occurrence Counts
Average Number of Co-Occurrences Across All Words: 17.5	Average Number of Co-Occurrences Across All Words: 23.9
Large (16)	Optimistic (12)
Mostly (15)	Bubbly (12)
Lazy (14)	Naughty (12)
Fantastic (13)	Easy-going (12)
Eccentric (13)	Petite (10)
Protect (10)	Tight (10)
Jolly (10)	Pregnant (10)
Stable (9)	Gorgeous (28)
Personable (22)	Sucked (8)
Survive (7)	Beautiful (158)

Table 3: Most biased words in GPT-3 model

4. **Methods:** In this research, several experiments are conducted using word embeddings learned from Reuters corpus & different network topologies (Dense DNN, 1dCNN, SimpleRNN & LSTM) with varying hyper parameters. Table 4 shows the network type and corresponding architecture details and Exp ID is used to track the experiments for results. Since LSTM was taking very long time to train, only one experiment is conducted. All the experiments are conducted in Google's colab with GPU. 'rmsprop' is used as the optimizer, batch size of 128, sequence length of 300 and categorical cross entropy is used as the loss function in all the experiments except LSTM. Due to the vocabulary size and sequence lengths chosen, some of the articles fell off from train and test sets as there are no words available (leaving 7976 documents in train set and 1994 in test set). Since LSTM is memory based network, 1032 is used as the sequence length with all the vocabulary and embedding size of 256. ReLU, Tanh and softmax activation functions are used in dense/CNN, RNN, output layers respectively.

Network Type	Exp ID	Embedding Dim	DropOut	Layers, Units
DNN	DNN_1L_NoDropout_Exp1	12	N/A	[HL1: 54]
DNN	DNN_1L_Dropout_Exp2	12	0.5	[HL1: 54]
DNN	DNN_1L_Dropout_Exp2	12	0.5	[HL1:128]
DNN	DNN_1L_NoDropout_Exp4	100	N/A	[HL1: 54]
DNN	DNN_1L_Dropout_Exp5	100	0.5	[HL1: 54]
1dCNN	1DCNN_1L_NoDropout_Exp5	25	N/A	[Con1D: 32x7, Maxpool: 5, Conv1D:31x7, GlobalMaxpool]
1dCNN	1DCNN_1L_NoDropout_Exp6	100	N/A	[Con1D: 32x7, GlobalMaxpool]
1dCNN	1DCNN_1L_NoDropout_Exp7	100	N/A	[Con1D: 32x7, GlobalMaxpool,Fltten, HL1:128]
1dCNN	1DCNN_1L_NoDropout_Exp8	100	N/A	[Con1D: 32x7, Maxpool: 5, Conv1D:31x7, Flatten, HL1:54]
1dCNN	1DCNN_1L_Dropout_Exp9	100	0.5	[Con1D: 32x7, Maxpool: 5, Conv1D:31x7, Flatten, Dropout,HL1:54]
1dCNN	1DCNN_1L_Dropout_Exp10	64	0.5	[Con1D: 32x7, Maxpool: 5, Conv1D:31x7, Flatten, Dropout,HL1:54]
SimpleRNN	RNN_NoDropout_Exp11	128	N/A	[RNNL: 256]
SimpleRNN	RNN_NoDropout_Exp12	128	N/A	[RNNL1: 128,RNNL2: 128,RNNL3: 128]
LSTM	LSTM_Dropout_Exp13	256	0.2	[LSTM:256]

Table 4. List of Experiments

5. **Results:** The experimental results are presented in Table 5 and top three candidates are highlighted in green, yellow and red in all the metrics. While dense DNNs networks seemed to perform better most of the times, the major disadvantage here is, it does not incorporate any sequential/memory properties. Word embeddings help with enriching the context of the words but the sequential processing of word vectors by remembering previous state is important to predict next best word in the context of chat bots. 1dCNN uses convolution and max-pooling operations to capture immediate patterns in the text but does not incorporate the long-term memory. In this example, capturing immediate context seemed to help a bit with test accuracy and stability metrics (as shown in Figure 4).

Exp ID	test_accuracy	train_accuracy	train_loss	train_time	validation_accuracy	validation_loss
DNN_1L_NoDropout_Exp1	0.7131	0.7256	1.1400	0 days 00:00:05.069842000	0.6766	1.3624
DNN_1L_Dropout_Exp2	0.7026	0.6505	1.4525	0 days 00:00:05.183123000	0.6525	1.4270
DNN_1L_Dropout_Exp3	0.7166	0.7145	1.1956	0 days 00:00:07.648430000	0.6766	1.3516
DNN_1L_Dropout_Exp4	0.7362	0.8544	0.6263	0 days 00:00:24.043600000	0.7157	1.2620
DNN_1L_Dropout_Exp5	0.7472	0.7911	0.9028	0 days 00:00:24.088717000	0.7139	1.2845
1DCNN_1L_NoDropout_Exp5	0.6715	0.6152	1.6073	0 days 00:00:36.232026000	0.6069	1.6138
1DCNN_1L_NoDropout_Exp6	0.7558	0.7041	1.3045	0 days 00:01:29.519113000	0.6919	1.3358
1DCNN_1L_NoDropout_Exp7	0.7212	0.8134	0.7569	0 days 00:01:36.934377000	0.7159	1.3641
1DCNN_1L_NoDropout_Exp8	0.6810	0.7233	1.0757	0 days 00:01:34.922979000	0.6692	1.4297
1DCNN_1L_NoDropout_Exp9	0.6770	0.7133	1.1639	0 days 00:01:36.501937000	0.6699	1.4244
1DCNN_1L_NoDropout_Exp10	0.6836	0.6901	1.2124	0 days 00:01:08.375658000	0.6556	1.4591
RNN_NoDropout_Exp11	0.3967	0.3615	2.3837	0 days 00:06:16.081541000	0.3461	2.3594
RNN_NoDropout_Exp12	0.3967	0.3769	2.3464	0 days 00:05:45.413024000	0.3759	2.3143
LSTM_Dropout_Exp13	0.6398	0.6319	1.4268	0 days 01:56:03.424664000	0.5864	1.6528

Table 5. Experimental Results

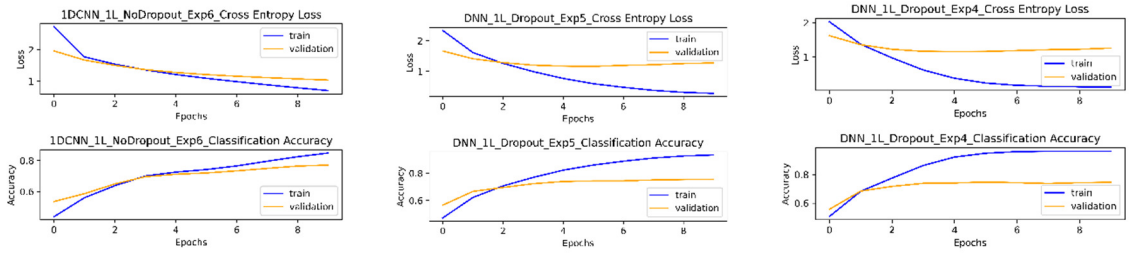

Figure 4. Loss and Accuracy plots of tops three performers

RNN is the worst performer of all, worse than a random guess. LSTM certainly boosted RNN's performance by many folds due to the incorporation of long and short-term memory but at the cost of huge training time. LSTM is not the best candidate based on overfitting & instability (Figure 5) observed in this use case.

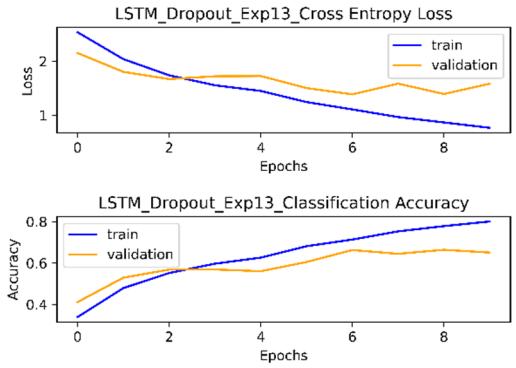

Figure 5. Performance of LSTM

6. **Conclusion:** Four different network topologies with varying parameters are experimented with Reuter corpus without any manual text processing (e.g., manual ontology building). While each network type brought its own advantage, due to the nature of this data (highly imbalanced dataset and potentially wrong labels) none of the networks performed well in meeting the accuracy and stability criteria. In this use case, One-hot encoded simple DNN (as shown in EDA) outperformed all the other complex network topologies indicating that simple word representations should not be ignored depending on use case.

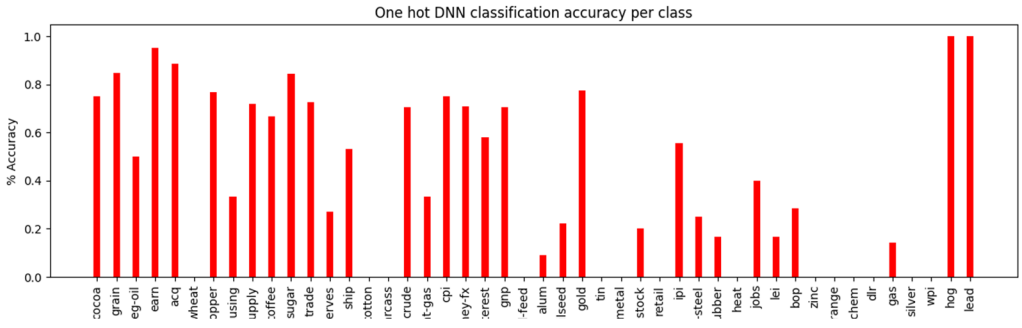

Figure 6. Test accuracy of each class from simple one-hot encoded DNN network

Figure 6 shows the classification accuracy of each class from test data. Many classes have zero classification rate and several others have below 50% rate indicating how poorly this model also performed due to the nature of the data (imbalance, how good are these labels). Relabeling the data by grouping some of the categories into other major categories, resampling, cleaning the text more (thorough EDA, removing stop words, correcting labels) might help with remedying the problems observed in this research. Given the learnings from this supervised learning exercise, clean and contextual text will play paramount importance in building generative models for predicting next best words/Q&As for chatbots. While word embeddings can ease the process of building context, other issues such as class imbalance/mis labelling can still hurt the models.

7. Lessons Learned:

- Language modeling is more art than a science
- Word embeddings can ease the process of building context, but bad text can still hurt the models, so basics of NLP methods should not be forgotten (so proper representations are chosen)
- Always keep the end goal/use-case in mind before picking/choosing a particular network topology

8. References:

[1] Chollet, F. 2018. Deep Learning with Python. Shelter Island, N.Y.: Manning. [ISBN-13: 978-1617294433]

[2]https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/

[3] https://techcrunch.com/2020/08/07/here-are-a-few-ways-gpt-3-can-go-wrong/