Table of Contents

DEEPLEARNING INTRO

Data Ingestion Case Study: Text

Table of Contents

Basic Concepts of Text Representation for Neural Networks

- PreProcessing and tokenization
- Bag of Words
- N-Grams
- Word2Vec
- Paragraph Vectors
- GloVE
- Words as Sequence of Characters

PreProcessing and tokenization

- Tokenizer
  - Splits stream of words into individual words
    - DefaultTokenizer
    - NGramTokenizer
      - PosUimaTokenizer
    - UimaTokenizer
- PreProcessors
  - LowCasePreProcessor
  - StemmingPreprocessor

Bag of Words

Corpus is represented as the bag(multiset) of its words.

- No Grammar
- No order
- Frequency only

"Bob and Carol and Ted and Alice"

Becomes the List ["Bob","and","Carol","Ted","Alice"]

Term frequency [1,3,1,1,1]

Bag of Words uses

- TfIDF
  - Frequency of word/document compared to word/corpus of documents

Bag of Words Example

- Lab Folder has example
- Tokenizer to read files from directory and label with filename

```
TokenizerFactory tokenizerFactory = new DefaultTokenizerFactory();

      LabelAwareIterator iterator = new FilenamesLabelAwareIterator.Builder()
              .addSourceFolder(new ClassPathResource("bow").getFile())
              .useAbsolutePathAsLabel(false)
              .build();
```

Bag of Words Example continued

- Code to show contents of iterator ``` while(iterator.hasNext()){

```
        LabelledDocument doc = iterator.nextDocument();
        System.out.println(doc.getContent());
        System.out.println(doc.getLabels().get(0));
    }

    iterator.reset();
```

```
------------------
<div style="page-break-after: always;"></div>

# Bag of Words Example Continued
```

BagOfWordsVectorizer vectorizer = new BagOfWordsVectorizer.Builder() .setMinWordFrequency(1) .setStopWords(new ArrayList()) .setTokenizerFactory(tokenizerFactory) .setIterator(iterator) .build(); vectorizer.fit();

```
------------------
<div style="page-break-after: always;"></div>

# Bag of Words Example Continued

* Code to explore the contents of the Bag of Words
```

```
log.info(vectorizer.getVocabCache().tokens().toString());
System.out.println(vectorizer.getVocabCache().totalNumberOfDocs());
System.out.println(vectorizer.getVocabCache().docAppearedIn("two."));
System.out.println(vectorizer.getVocabCache().docAppearedIn("one."));
System.out.println(vectorizer.getVocabCache().docAppearedIn("world"));
```

```
------------------
<div style="page-break-after: always;"></div>

# NGrams
* Contiguous sequence of n items from a sequence of text

Example "It is the year 2016"

Bi-grams "It is" "is the" "the year" "year 2016"
Tri-grams "It is the" "is the year" "the year 2016"

------------------
<div style="page-break-after: always;"></div>

# NGram uses

* Provide more context than Bag of Words
* Used in some Neural Net for Speech Recognition to narrow the scope of prediction
  * RNN predicts next word out of top x percent of trigram for previous 2 word pre
dictions


------------------
<div style="page-break-after: always;"></div>

# NGram code Example
```

public static void main(String[] args) throws Exception{ String toTokenize = "To boldly go where no one has gone before."; TokenizerFactory factory = new NGramTokenizerFactory(new DefaultTokenizerFactory(), 1, 2); Tokenizer tokenizer = factory.create(toTokenize); factory = new NGramTokenizerFactory(new DefaultTokenizerFactory(), 2, 3); List tokens = factory.create(toTokenize).getTokens(); log.info(tokens.toString());

```
Output
```

[To, boldly], [boldly, go], [go, where],..... [To, boldly, go], [boldly, go, where] .....

```
```

Word2Vec

- Model for word embeddings
- Vector Space
- Each word in Corpus => Vector in Vector Space
- Relative location of word in vector space denotes relationship
    - Boy->Man Girl->Woman

Word2Vec

- Model for word embeddings
- Vector Space
- Each word in Corpus => Vector in Vector Space

Word2Vec - Generating the Vector Space

- Neural Network trained to return word probabilities of a moving window
  - Given word "Paris", out of the corpus of words predict probility of each word occuring within say 5 words of the word "Paris"
- One hot Vector, size of every word in the corpus
- all 0's except for 1 representing the word
- See Demo https://ronxin.github.io/wevi/
- See example in intellij
- ALlows you to do word math
  - King - Man + Woman = (?) Queen

One-hot encoding

- Vector, the size of the vocabulary, all 0's except for 1

Text as Sequence of Characters

Text can be treated as sequence of characters, and neural network can be trained to answer the question. Given input character X predict the next character, and repeat.

Recurrent Neural Networks and Sequence Data

- Recurrent Neural Networks have the capacity to recognize dependencies in time series data
- Breaking a text corpus into a series of single characters allows the network to learn dependencies such as the most common letter after a "Q" is a "U", when a quote has been opened it should eventually be closed.
- In the Lab you will train a neural network to write weather forecasts.

GloVE