Keras word embedding

2018 Data science spring

In this document ...

Word embeddings provide a **dense** representation of words and their relative meanings.

They are an improvement over sparse representations used in simpler bag of word model representations.

Word embeddings can be learned from text data and reused among projects. They can also be learned as part of fitting a neural network on text data.

You will learn ...

About word embeddings and that Keras supports word embeddings via the Embedding layer.

How to use a pre-trained word embedding in a neural network.

Outline

Word Embedding

Keras embedding layer

Pre-trained GloVe embedding

Word embedding

A word embedding is a class of approaches for representing words and documents using a **dense vector representation**.

It is an improvement over more the traditional **bag-of-word** model encoding schemes where large sparse vectors were used to represent each word or to score each word within a vector to represent an entire vocabulary. These representations were sparse because the vocabularies were vast and a given word or document would be represented by a large vector comprised **mostly of zero values**.

Instead, in an embedding, words are represented by dense vectors where a vector represents the projection of the word into a continuous vector space.

Word embedding

The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used.

The position of a word in the learned vector space is referred to as its embedding.

Two popular examples of methods of learning word embeddings from text include:

- **Word2Vec**
- **GloVe**

Keras embedding layer

Keras offers an Embedding layer that can be used for neural networks on text data.

It requires that the input data be **integer encoded**, so that each word is represented by a unique integer. This data preparation step can be performed using the Tokenizer API also provided with Keras.

Keras embedding layer

The Embedding layer is initialized with random weights and will learn an embedding for all of the words in the training dataset.

It is a flexible layer that can be used in a variety of ways, such as:

- It can be used alone to **learn a word embedding** that can be saved and used in another model later.
- It can be used as **part of a deep learning** model where the embedding is learned along with the model itself.
- It can be used to load a **pre-trained word embedding** model, a type of transfer learning.

Keras embedding layer

The Embedding layer is defined as the first hidden layer of a network. It must specify 3 arguments:

- **input_dim**: This is the **size of the vocabulary** in the text data. For example, if your data is integer encoded to values between 0-10, then the size of the vocabulary would be 11 words.
- **output_dim**: This is the **size of the vector space** in which words will be embedded. It defines the size of the output vectors from this layer for each word. For example, it could be 32 or 100 or even larger. Test different values for your problem.
- **input_length**: This is the **length of input sequences**, as you would define for any input layer of a Keras model. For example, if all of your input documents are comprised of 1000 words, this would be 1000.

Pre-trained GloVe embedding

The Keras Embedding layer can also use a word embedding learned elsewhere.

It is common in the field of Natural Language Processing to learn, save, and make freely available word embeddings.

For example, the researchers behind GloVe method provide a suite of pre-trained word embeddings on their website released under a public domain license. See:

- GloVe: Global Vectors for Word Representation

Pre-trained GloVe embedding

the 0.418 0.24968 -0.41242 0.1217 0.34527 -0.044457 -0.49688 -0.17862 -0.00066023 -0.6566 0.27843 -0.14767 -0.55677 0.14658 -0.0095095 0.011658 0.10204 -0.12792 -0.8443 -0.12181 -0.016801 -0.33279 -0.1552 -0.23131 -0.19181 -1.8823 -0.76746 0.099051 -0.42125 -0.19526 4.0071 -0.18594 -0.52287 -0.31681 0.00059213 0.0074449 0.17778 -0.15897 0.012041 -0.054223 -0.29871 -0.15749 -0.34758 -0.045637 -0.44251 0.18785 0.0027849 -0.18411 -0.11514 -0.78581

Pre-trained GloVe embedding

Now you will have to load the whole embedding into memory:

1. Declare a dictionary to store the word to word embedding mapping
2. Open file ‘glove.6B.300d.txt’
3. For each line, split the string. The first value will be the word and the others are the word vector.

Pre-trained GloVe embedding

Now you will have to create a embedding matrix for the word2index of the dataset:

1. Declare a numpy array of ((vocabulary size, embedding dimension)).
2. Use items() function in word2index dictionary to iterate through all word and their index.
3. Assign the word vector for each word in word2index into the embedding matrix.

Pre-trained GloVe embedding

Embedding layer

```
keras.layers.Embedding(input_dim,
     output_dim,
     embeddings_initializer = 'uniform',
     embeddings_regularizer = None,
     activity_regularizer = None,
     embeddings_constraint = None,
     mask_zero = False,
     input_length = None)
```

Reference

How to Use Word Embedding Layers for Deep Learning with Keras