Embeddings

RNN use krte tym important chiz yeah hoti hai ki

Textual Data ko Hum numbers mai convert karde ya Vectors mai convert krde Coz RNN Model Humara Numbers ya Vectors pe kaam krta hai

There are 2 Approaches

1.) Integer Encoding
2.) Embeddings

Integer Encoding Working:

1.) We Form Vocabulary
    Vocabulary means, Pura document mai Kitne Unique Words hai
2.) Har Unique word ko Integer Value Provide karenge.
3.) Har Unique Word ko Replace kardenge Integers se
4.)

Embedding

In NLP, Word Embedding is a term used for representation of words for text analysis, typically in the form of a real-valued vector that encodes meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning.

Embeddings, in a simple way, can be thought of as a way to represent words or entities in a lower-dimensional space.

It's a way to convert categorical variables, like words, into continuous vectors of real numbers. The key idea is to capture semantic relationships between words based on their context in a way that is useful for machine learning models.

Here's an analogy to help with the intuition:

Imagine you are trying to describe fruits in terms of their taste and texture. You decide to represent each fruit with two numbers: one for sweetness and one for juiciness.

- **Apple:** (Sweetness: 7, Juiciness: 5)
- **Orange:** (Sweetness: 8, Juiciness: 9)
- **Banana:** (Sweetness: 6, Juiciness: 8)

In this simplified example, you've created a 2-dimensional "fruit space" where each fruit is represented by a point. Similar fruits are closer together in this space. For instance, apples and bananas are closer to each other compared to apples and oranges.

Now, in the context of natural language processing and embeddings:

- **Word Embeddings:**
  - Words are represented as vectors in a multi-dimensional space.
  - The dimensions of this space capture certain semantic aspects.
  - Words with similar meanings or contexts are close together in this space.
- **Example:**
  - Word: "king" might be represented as (Gender: Male, Royalty: High, Power: High).
  - Word: "queen" might be represented as (Gender: Female, Royalty: High, Power: High).

- Word: "dog" might be represented as (Pet: Yes, Intelligence: Low, Size: Small).

The goal is that these embeddings capture meaningful relationships between words. For example, in the word embedding space, the vector for "king" minus the vector for "man" plus the vector for "woman" might be close to the vector for "queen."

These embeddings are often learned from large amounts of data using techniques like Word2Vec, GloVe, or embeddings layers in neural networks. The idea is that the model learns to place words in the embedding space based on their co-occurrence patterns in the data, allowing it to capture semantic relationships.

Word Embeddings able to capture Semantic Meanings

Word2Vec & Glovc are Word Embedding Techniques

DL Mai while in training Process we can put a Embedding Layer

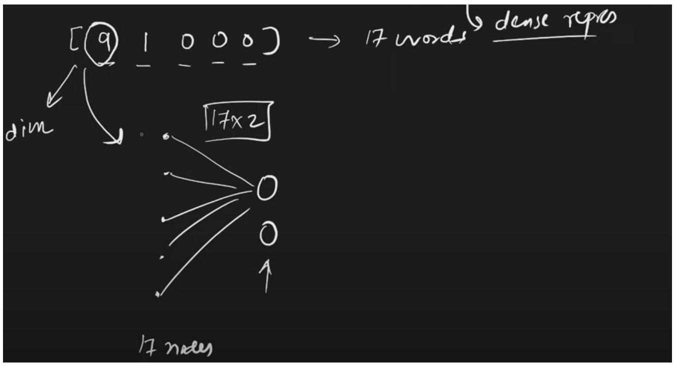
At start only Before starting Training,

In short we get better result from this.

In Keras Applying Embedding is Easy, we just have to make sure ki

Jo data input horaha Embedding layer mai that must be Integer Encoding.

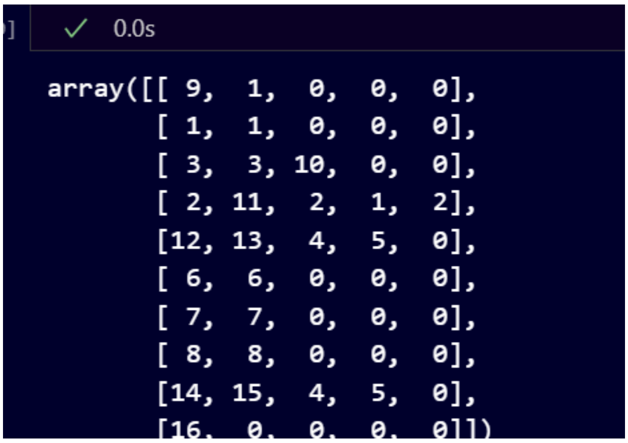
```
model = Sequential()
model.add(Embedding(17,output_dim=2,input_length=5))
model.summary()
```



Jo 17 Words fix kiya hai usko hum 2 Number mai Summarize krrhe kindoff yeah hai output_dim

Example:

Text ko Integer Encoding kiya Humne

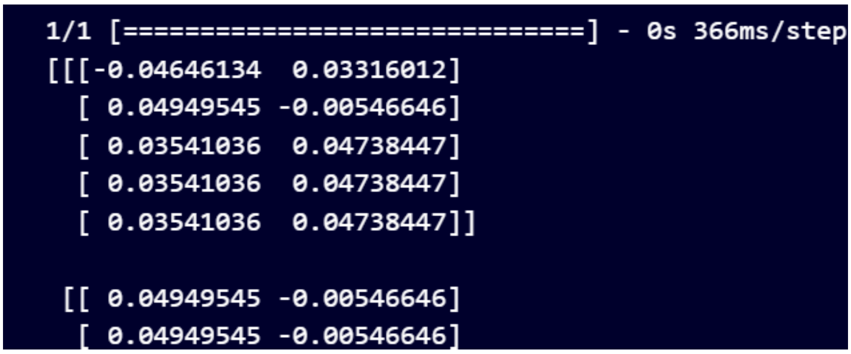


```
array([[ 9,  1,  0,  0,  0],
       [ 1,  1,  0,  0,  0],
       [ 3,  3, 10,  0,  0],
       [ 2, 11,  2,  1,  2],
       [12, 13,  4,  5,  0],
       [ 6,  6,  0,  0,  0],
       [ 7,  7,  0,  0,  0],
       [ 8,  8,  0,  0,  0],
       [14, 15,  4,  5,  0],
       [16,  0,  0,  0,  0]])
```

Then vo Jo Integer Encoded text tha usko Embedding layer kesath daala

```
model.add(Embedding(17,output_dim=2,input_length=5))
```

Then this is output , you can see , saare 5 max inputs ko 2 Dimension ke outputs mai represent krrha hai



```
1/1 [==============================] - 0s 366ms/step
[[[-0.04646134  0.03316012]
  [ 0.04949545 -0.00546646]
  [ 0.03541036  0.04738447]
  [ 0.03541036  0.04738447]
  [ 0.03541036  0.04738447]]

 [[ 0.04949545 -0.00546646]
  [ 0.04949545 -0.00546646]
```

Pretrained Embeddings bhi use karskte hoo lyk Word2Vec & Glov But

Model Kesath Embeddings karte waqt Zyaada fayda hota hai as compare to using pretrained Embeddings , As jo Embeddings jo generate hoti vo aapke data ke hisabse hoti