* **INPUT:**

"I love transformers"

* **TOKENIZATION**

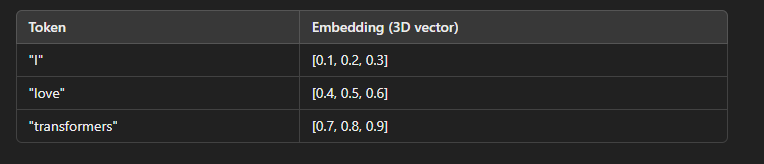
I = ###

Love =###

Transformers =###

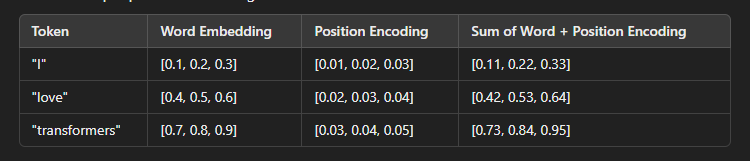
Unique id is given to each word

* **WORD EMBEDDING**

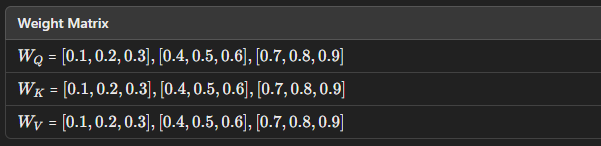


* **POSITION ENCODING**

**Q K V**



* **ENCODER LAYER**



**-LETS CALCULATE WEIGHT FOR “I”**

**-COMPUTE Q FOR I**





**-COMPUTE K FOR I**







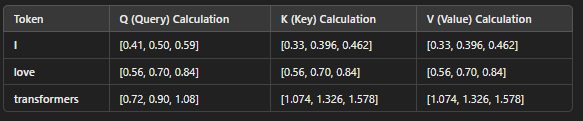
**-COMPUTE V FOR I**







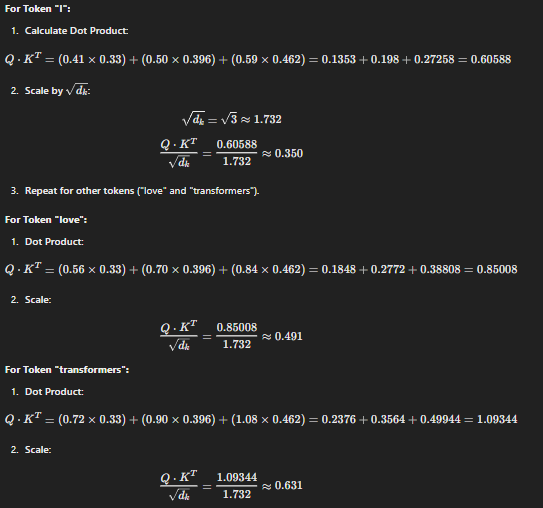
**-REPEAT THE STEPS FOR “LOVE”AND “NLP”**



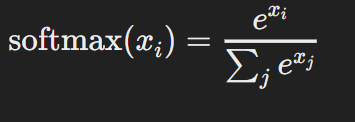
**-ATTENTION SCORE**

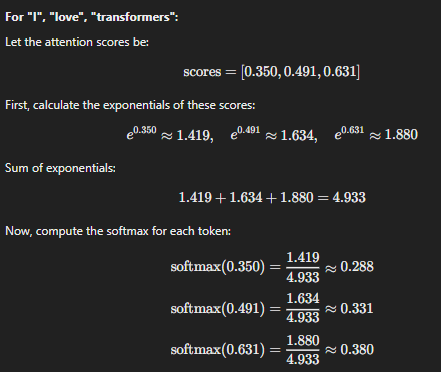
--

Calculating dot product



**-APPLY SOFTMAX**





**-Compute Weighted Sum (Output Calculation)**

Softmax **x** vector I





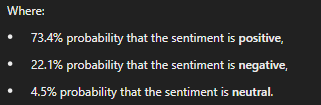
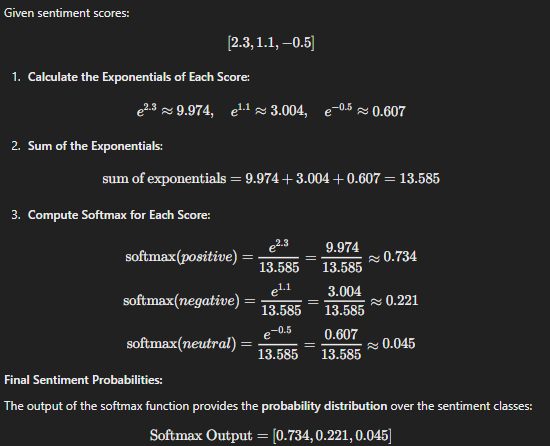


**-These weights are used as input for sentiment analysis model….**

Let’s assume that the model produces the following output scores

**Sentiment Scores=[2.3,1.1,−0.5]**

**-Apply softmax to this output again**



The above output has 3 label…

Sentiment Analysis

**Packages**

* **Import torch:**

used in training loops and also helps to manage GPU

* **from torch.utils.data import Dataset, DataLoader, random\_split:**

**Dataset**:it allows easy indexing and load sample dataset's ,sentiment dataset class derived from this

**Dataloader:** used to load data in batch during training and evaluation.

**Random\_split:**  splits dataset into training and validation

* **from transformers import BertTokenizer, BertForSequenceClassification, AdamW, get\_scheduler**

**Transformer:**

**To over come the limitations of RNN transformers were introduced, drawbacks includes:**

**-Sequential processing**

**-Slow training**

**-Vanishing gradient problem:** the neural network is very deep ,as the gradient signal travel back through many layers it can become smaller,this makes it difficult for earlier layers to learn effectively.

**-BertTokenizer:**

Converts text into numerical tokens tht can be given to model

**-BertForSequenceClassificaton:**

Model used for classification task

**-AdamW:**

adam optimization for weighdecay …reduce loss funciton using L2 lasso

**-get\_scheduler:**

creates learning rate scheduler, it adjust the learning rate during training

* **From transformers import pipeline**

We are using pipeline to load the saved model and tokenizer for further analysis

**bert-base-uncased**

Case insensitivity

Pretrained on bookcorpus and english wikipedia