\*\*\*I have used Tensorflow notebook that uses tf.keras, a high-level API to build and train models in TensorFlow, and TensorFlow Hub, a library and platform for transfer learning. I have modified it a bit by and a dropout layer and Tensorboard for model visualization\*\*\*

#Details

Created using Swivel matrix factorization method.

Submatrix-wise Vector Embedding Learner (Swivel), a method for generating low-dimensional feature embeddings from a feature co-occurrence matrix. Swivel performs approximate factorization of the point-wise mutual information matrix via stochastic gradient descent. It uses a piecewise loss with special handling for unobserved co-occurrences, and thus makes use of all the information in the matrix. While this requires computation proportional to the size of the entire matrix, we make use of vectorized multiplication to process thousands of rows and columns at once to compute millions of predicted values. Furthermore, we partition the matrix into shards in order to parallelize the computation across many nodes. This approach results in more accurate embeddings than can be achieved with methods that consider only observed co-occurrences, and can scale to much larger corpora than can be handled with sampling methods

Reference- Cornell University. creator-Noam Shazeer, Ryan Doherty,

#Input

The module takes a batch of sentences in a 1-D tensor of strings as input.

#Preprocessing

The module preprocesses its input by splitting on spaces.

#Vocabulary

Vocabulary contains 20,000 tokens and 1 out of vocabulary bucket for unknown tokens.

#Sentence embeddings

Word embeddings are combined into sentence embedding using the sqrtn combiner

Model summary

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

keras\_layer (KerasLayer) (None, 20) 400020

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 16) 336

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 1) 17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 1) 0

=================================================================

Keras layer uses the tensorflow hub model to embed the sentences

**Loss Function And An Optimizer For Training.** Since this is a binary classification problem and the model outputs logits (a single-unit layer with a linear activation), we'll use the binary\_crossentropy loss function. Adam for optimization.

FINAL RESULTS

**loss: 0.1017 - accuracy: 0.9679 - val\_loss: 0.3145 - val\_accuracy: 0.8756**

For model visualization I have used TensorBoard which you can run on google colab.