Embedding Columns

Learn about the embedding feature columns for the ML model's input layer.

Chapter Goals:

- Process the embedding feature columns used for the machine learning model's input layer

A. Feature embeddings

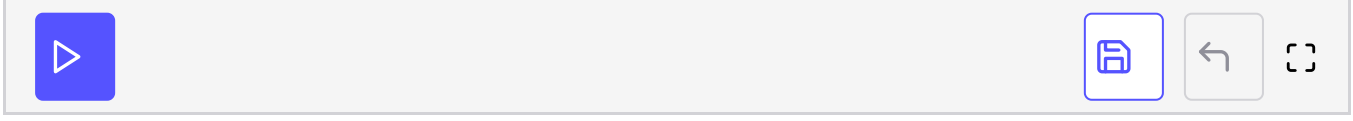
The other feature column for categorical features is embedding feature columns. For our dataset, the embedding features are `'Store'` and `'Dept'`. Both these features have many more categories than the indicator features, so using a one-hot vector representation wouldn't necessarily be the most efficient or useful.

Furthermore, there are connections between categories for both the `'Store'` and `'Dept'` features. For example, we may see similar sales trends between stores in the same region or between departments that sell complementary items (e.g. milk and cereal).

These are trends that can be encapsulated by an embedding feature column, which maps each category to a real-number vector of a specified dimension. We can set the vector dimension to anything, but a good rule of thumb is to set it equal to the 4^{th} root of the number of categories.

```
import tensorflow as tf

stores = list(range(1, 46))
stores_col = tf.feature_column.categorical_column_with_vocabulary_list(
    'StoreID', stores, dtype=tf.int64)
embedding_dim = int(45**0.25)  # 4th root
feature_col = tf.feature_column.embedding_column(
    stores_col, embedding_dim)
```



Creating an embedding feature column for the 'StoreID' feature. There are 45 stores in the dataset.

Creating an embedding feature column for the 'Store' feature. There are 45 stores in the dataset, labeled from 1-45.

For more on word embedding, check out our course Image Recognition on educative.

Time to Code!

In this chapter you'll be creating the embedding feature columns for the dataset by completing the `add_embedding_columns` function. We've already filled the function with skeleton code that iterates through the embedding features in the dataset.

Each embedding feature column is built from a vocabulary list. The vocabulary list comes from the unique values of the feature in the `final_dataset` DataFrame.

Set `vocab_list` equal to the unique values in `final_dataset[feature_name]`, cast as a list.

Using the vocabulary list, we'll create the categorical column for the feature. Each of our dataset's embedding features has datatype `tf.int64`.

Set `vocab_col` equal to `tf.feature_column.categorical_column_with_vocabulary_list` with `feature_name` and `vocab_list` as the required arguments, as well as `tf.int64` for the `dtype` keyword argument.

Following the general rule of thumb, we set the embedding dimension as the 4^{th} root of the size of the vocabulary list.

Set `embedding_dim` equal to the length of `vocab_list` raised to the power of `0.25` and cast as an integer.

```
import tensorflow as tf

# Add the embedding feature columns to the list of feature columns
def add_embedding_columns(final_dataset, feature_columns):
    embedding_features = ['Store', 'Dept']
    for feature_name in embedding_features:
        # CODE HERE
        pass
```