Otto Product Classification using TensorFlow

This project is designed to use Machine Learning, in order to learn the product classifications of Otto, shared on Kaggle, and predict classifications for the test data.

Using TensorFlow

TensorFlow has been used to learn the model for the training data and then use it to evaluate the test data.

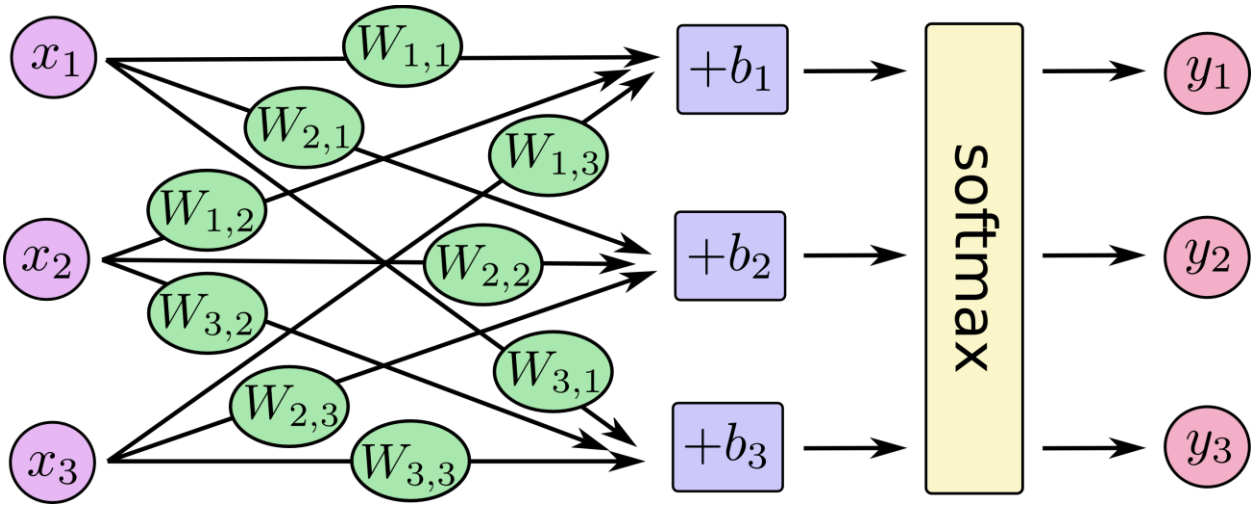
TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. (TensorFlow)

Softmax Regression

The Otto dataset comprises of features that need to be classified into one of the nine classes. Therefore, softmax regression model is used to assign the weights to each class for a given feature set.

A softmax regression has two steps: first we add up the evidence of our input being in certain classes, and then we convert that evidence into probabilities.

You can picture our softmax regression as looking something like the following, although with a lot more xs. For each output, we compute a weighted sum of the xs, add a bias, and then apply softmax.



Vectorizing the above operation:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

In summary: **y = Softmax(Wx + b)**

(Image source: TensorFlow)

Execution

OttoSoftmax.py script file contains the learning and evaluation logic, while the DatasetUtils provides the data loading utilities.

The model is currently made to learn over 100 iterations, however, it can be adjustsed to achieve the desired accuracy and/or efficiency.

The current learning operation trains the model to ~75% accuracy. This is not a particularly spectacular result however, other, more sophisticated models can be explored and experimented with to identify the one that best fits the given dataset.

Some examples of other models and learning techniques:

- ➔ XGBoost
- ➔ Ensemble Weights
- ➔ Calibration
- ➔ Deep Learning (Neural Nets)
- ➔ Tree-Based classification
- ➔ KNN

Here is the snapshot of the final output:

```
Training step: 91
Training step: 92
Training step: 93
Training step: 94
Training step: 95
Training step: 96
Training step: 97
Training step: 98
Training step: 99
Accuracy on training data:
0.746857
Classifying test data:
[9 1 8 ..., 4 4 4]
Press any key to continue . . .
```