

## TensorFlow skill testing project: Classification in TensorFlow

**Objective:** Create and run your own machine learning algorithms in TensorFlow on a real-world dataset. You'll be examining the HAPT dataset (<http://bit.ly/skill-test-data-1>) and using it to predict people's posture given smartphone accelerometer data.

**Architecture:** Your code will consist of three modules. This document contains the specification for the first module.

---

### Module 1: `data_preprocessing.py`

You have complete freedom to design this module any way that makes sense to you. Just a few pointers:

- 1) This module should convert your example inputs and labels into a form that can be fed into a neural network (or some other learning algorithm) conveniently. To do this, you'll want to normalize your inputs such that they have **zero mean** and **unit variance**.
- 2) For example, imagine that you're trying to predict academic achievement from a set of inputs like `[PARENT_INCOME, NUMBER_OF_SIBLINGS, AGE]`. To preprocess `PARENT_INCOME` to zero mean, you would first subtract the average `PARENT_INCOME` in your dataset from the `PARENT_INCOME` entry for each example in your dataset. Then, to normalize `PARENT_INCOME` to have unit variance, you would divide this result by the by the standard deviation of `PARENT_INCOME` in the dataset. Then you would do the same thing for the `NUMBER_OF_SIBLINGS` and `AGE` categories.
- 3) Be sure to split your data into two components: a training set (which will be fed to the algorithm to train it) and a validation set (which the algorithm won't use to learn, but rather will just be used to check its performance on new data).

**Hint (for future reference):** If you have sparse data, consider using L1 or L2 normalization instead. See this Quora question: <http://bit.ly/quora-link-1> .