**Introduction to Dense Net with TensorFlow**

[TensorFlow](https://www.tensorflow.org/) offers both high- and low-level APIs for Deep Learning. Coding in TensorFlow is slightly different from other machine learning frameworks. You first need to define the variables and architectures. This is because the entire code is executed outside of Python with C++ and the python code itself is just a bunch of definitions.

The aim of this post is to replicate the previous Keras code into TensorFlow. Before writing code in TensorFlow, it is better to use high-level APIs like Keras to build the model (read [**Introduction to Dense Net with Keras**](https://www.mydatahack.com/introduction-to-dense-net-with-keras/)for a preparation).

**Steps**

1. **Data Preparation**

As in the [previous post](https://www.mydatahack.com/introduction-to-dense-net-with-keras/), we are importing the Iris dataset from a csv file. This step is the same as before.

You can also check the official TensorFlow documents about deep learning on Iris dataset [here](https://www.tensorflow.org/get_started/eager). The way the dataset is preprocessed is quite different from what I did and will be an interesting read.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

1. **Defining Variables and Models**

Before running the code, we need to define variables and models. Even for more complicated models (e.g. with added convolutional layers), you will do pretty much the same steps.

- Set hyperparameters

- Set layers

- Define placeholders

- Define layers

- Define architecture

- Define variable dictionary

- Build Model

- Define loss & optimizer

- Define evaluation metrics

Here is the code from the steps above.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

1. **Initialise and Run**

Once everything is set up, initialise variables and execute the code in session! After 1000 epochs, you will see the test accuracy of 96%.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]