# Introduction

An overview of the multilayer perceptron neural network and deep learning in TensorFlow.

In the **Intro to Deep Learning** section, you will learn about one of the most essential neural networks used in deep learning, the multilayer perceptron.

A. Multilayer perceptron

The multilayer perceptron (MLP) is used for a variety of tasks, such as stock analysis, spam detection, and election voting predictions. In the following chapters you will learn how to code your own MLP and apply it to the task of classifying 2-D points in the Cartesian plane.

You will also learn the basics of creating a *computation graph*, i.e. the structure of a neural network. The structure of a neural network can be viewed in layers of *neurons*:
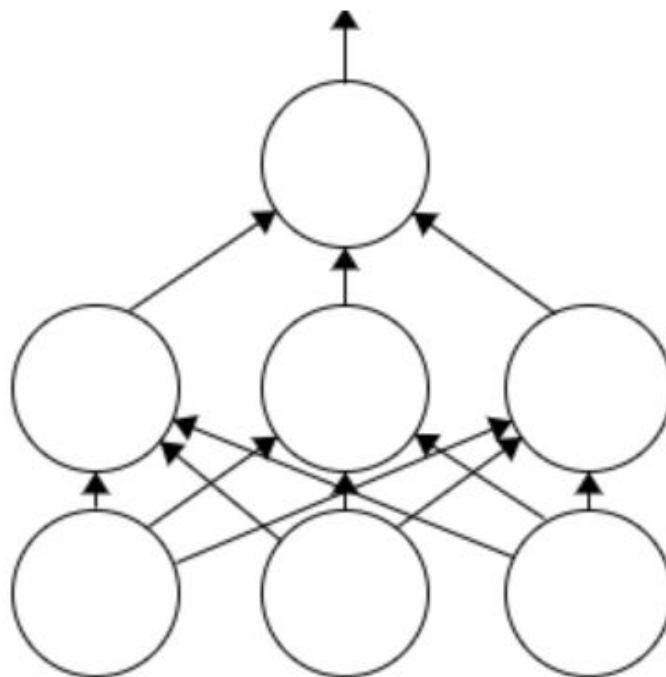

Multilayer perceptron

In the diagram, the circles represent neurons and the *connections* are the arrows going from neurons in one layer to the next. There are three layers in this diagram's neural network:

- *Input layer*: The first (bottom) layer.
- *Output layer*: The last (top) layer.
- *Hidden layer(s)*: The layer(s) between the input and output layers. In the diagram, there is 1.

The number of hidden layers represents how "deep" a model is, and you'll see the power of adding hidden layers to a model.

When diagramming the computation graph, it is common to eschew the neurons and connections:

Output

Output Layer

Hidden Layer

Input Layer
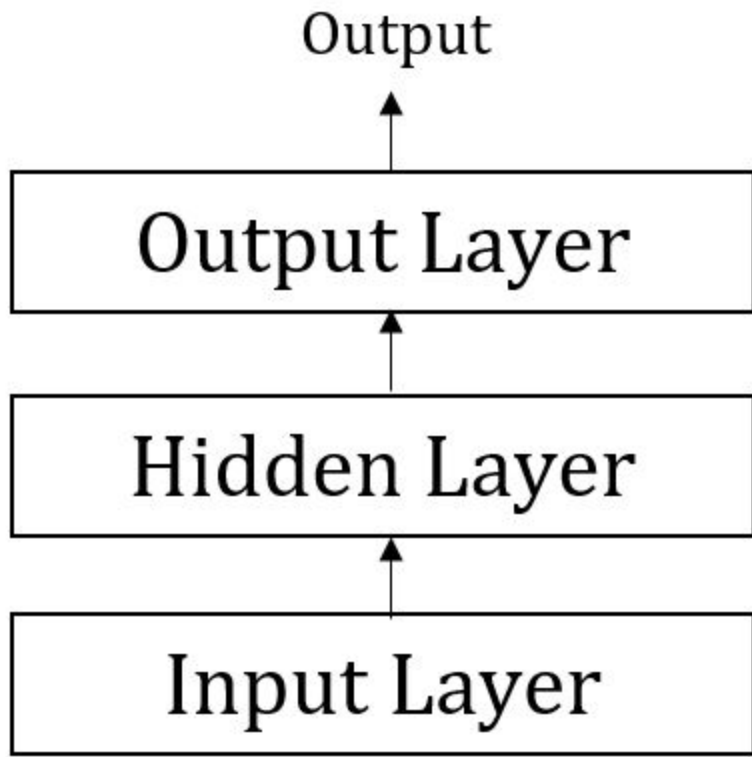
This makes it easier to view computation graphs of complex neural networks with dozens of layers.

B. TensorFlow

To code our neural network model, we will be using TensorFlow, one of the most popular deep learning frameworks. The name for TensorFlow is derived from *tensors*, which are basically multidimensional (i.e. generalized) vectors/matrices. When writing the code, it may be easier to think of anything with numeric values as being a tensor.

In TensorFlow, we first create the computation graph structure (which we will do in chapters 2-5), and then train and evaluate the model with input data and labels.

Something quick to note is that TensorFlow is normally imported via the statement `import tensorflow as tf`. This provides a shorthand way to call the module.