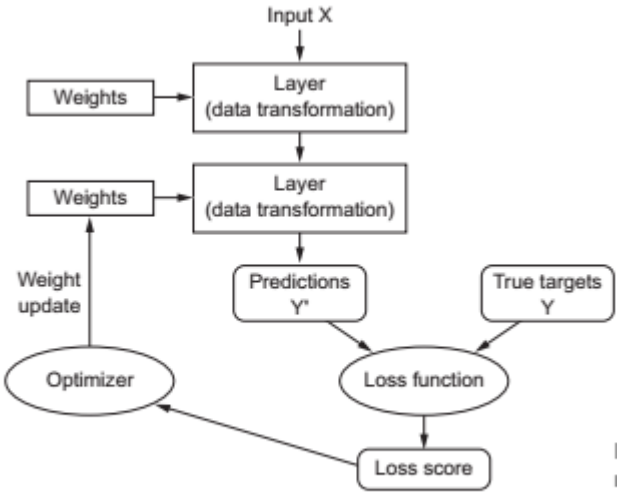
Chapter 3: Getting started with neural networks

A neural network is made of components which are

1) *Layers*, which are combined into a *network* (or *model*)
2) The *input data* and corresponding *targets*
3) The *loss function*, which defines the feedback signal used for learning
4) The *optimizer*, which determines how learning proceeds

A simple diagram of how the components are combines is shown below



Layers are the data processing module of a neural network. Layers can be stateless or stateful depending on what one is trying to do. There are different types of layer depending on the input tensor on is feeding into it. For 2D tensor input a Densely connected layer is used. For a 3D tensor input recurrent layers such as LSTM layer is used. For a 4D tensor input a 2D Convolution layers Conv2D are used .

```
from keras import models
from keras import layers
model = models.Sequential()
model.add(layers.Dense(32, input_shape=(784,)))
model.add(layers.Dense(32))
```

The first layer in the above needs an input of vector of dimension of 784 and will output 32 nodes.The output of one layer is the input of the next layer that is why input_shape parameter is not included in the second layer.A layer can also have an activation parameter depending on the layer and what the output is about.

Layers will combine together to form network. The combination architecture may be in different forms

1)Two-branch networks

2) Multi head networks
3) Inception block

Loss Function (Objective function): The quantity that compute the error between what the network gives and the expected value. There are different type of loss function depending on the type of problem on is trying to solve

Optimizer: This helps to update the trainable parameters of the network.

Three examples in the notebooks help solidify the different components of a neural network, the data preprocessing involved, and the common issue that arise when training and validating a data(overfitting).