Neural Nets with PyTorch

Building the model

- PyTorch provides two approaches to building the structure of your model:
- Sequential Model: Allows you to build a model by stacking layers in a sequential order.
- Subclasses : Builds from python class that inherits from nn.Module and define your model architecture in the __init__ and forward methods.

PyTorch- Sequential Model

The Sequential function (similar to Keras) and initialized model object:

```
from torch import nn
model = nn.Sequential()
```

PyTorch- Sequential Model

```
from torch import nn

      nn.Sequential(
        nn.Linear(3,4),
        # model.add(Dense(units=4, input_dim=3))

        nn.Sigmoid(),
        # model.add(Activation(sigmoid'))

        nn.Linear(4, 4),
        # model.add(Dense(units=4))

        nn.Sigmoid(),
        # model.add(Activation(sigmoid'))

        nn.Linear(4, 3),
        # model.add(Dense(units=3))

        nn.Softmax()
        # model.add(Activation('softmax'))
      )
```