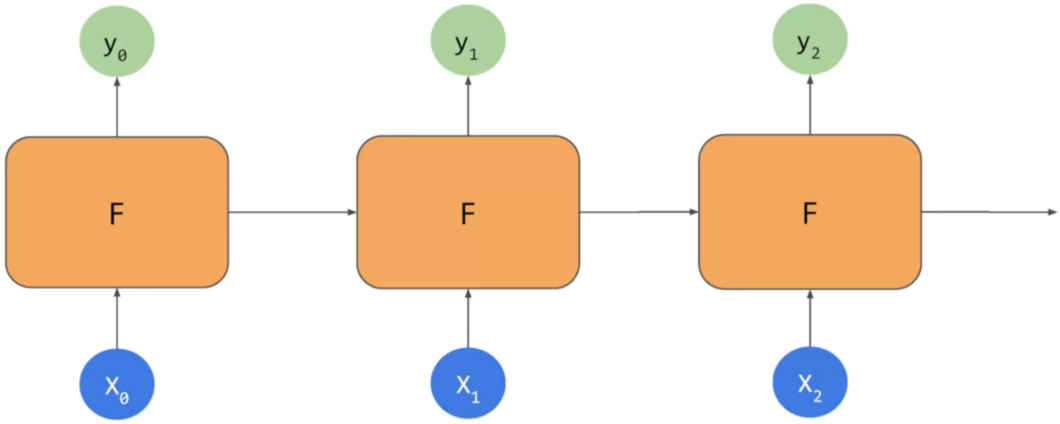
Week 3

- This week would apply RNNs into the time series neural network



RNN Diagram

- A difference compared to last week is that RNN inputs are 3 dimentional

```
shape=[batch_size, time_steps, dims]
```

- Sequence to vector RNNs ignores all the outputs except the last one (Default in Keras)
- Has to add `return_sequence=True` to add another layer

Lambda Layer

- Lambda layer allows arbitrary operations to effectively expand the functionality of Keras
- Example:

```
model = keras.models.Sequential([
  keras.layers.Lambda(lambda x: tf.expand_dims(x, axis=-1),
                      input_shape=[None]),
  keras.layers.SimpleRNN(20, return_sequences=True),
  keras.layers.SimpleRNN(20),
  keras.layers.Dense(1),
  keras.layers.Lambda(lambda x: x * 100.0)
])
```

Using RNN and Lambda to Create Neural Network

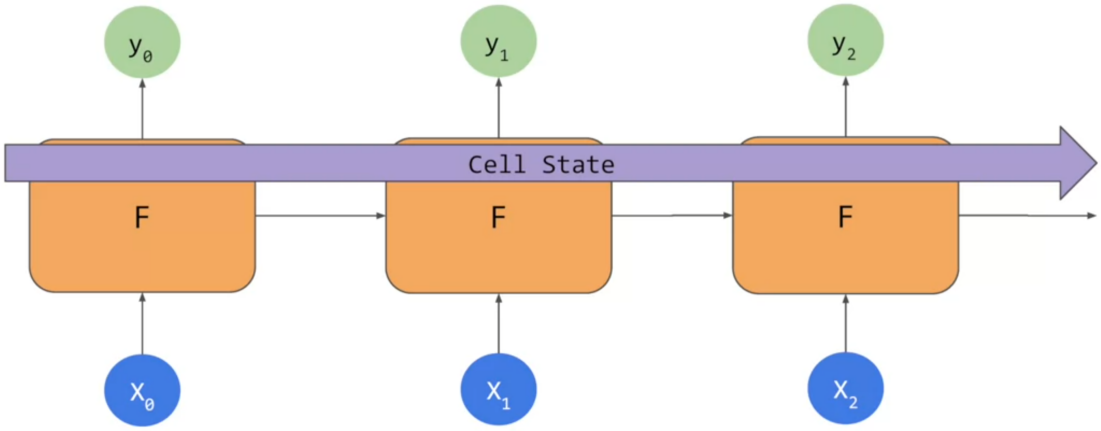
- The code below uses the techniques from above to create a neural network

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/a8f5b026-ec80-4787-a9fd-edb23fbf33fa/rnn_neural_network.ipynb

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2c4ec812-b1cb-4106-a1a3-7c3f28ba105f/rnn_neural_network.py

LSTM

- A better approach may be using LSTMs instead of RNNs to see the impact since there was a flat area while using RNNs
- LSTMs has cell states that passes through each cell so more information could be maintained
- Earlier data can have a better impact



LSTM Cells

- Code for neural network using LSTM cells

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1972d2be-213b-4731-a2d8-c5576537b4c4/lstm_neural_network.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/910fdd60-be70-4b63-8bf7-6c1e1ab49d17/lstm_neural_network.ipynb

Exercise

- Code for exercise in week 3

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d9a2bd27-6c6e-4508-8072-98cc320f4efb/exercise_1_w3.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b2eccbc3-b9d4-4719-a939-5e0f6b31459a/exercise_1_w3.ipynb