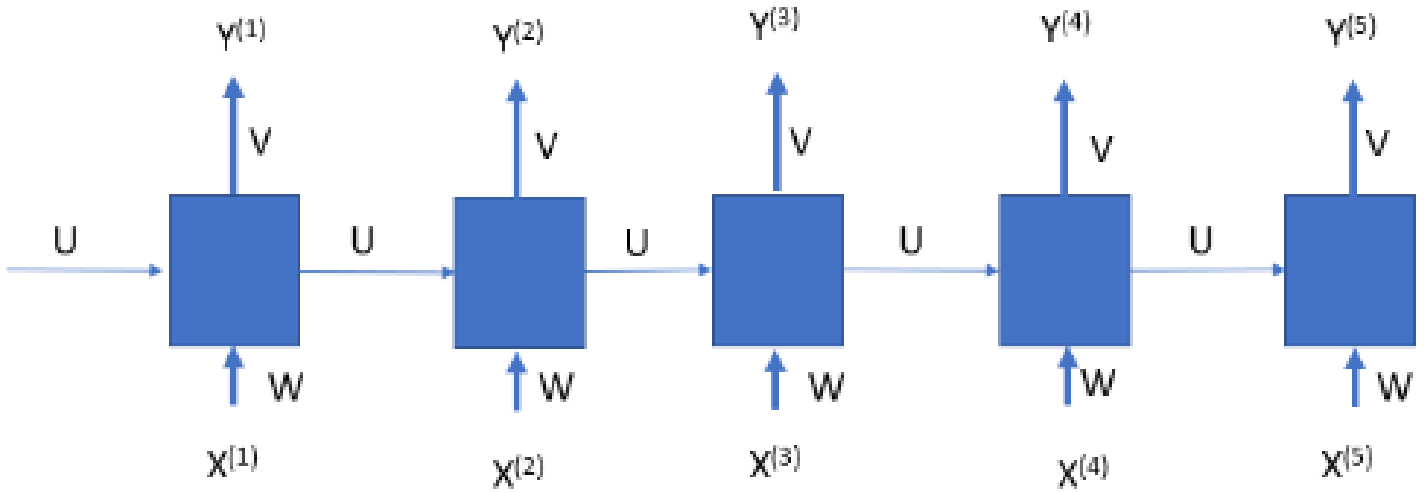
Implementation of RNN - Keras



The Keras RNN API has two types of RNN supports:

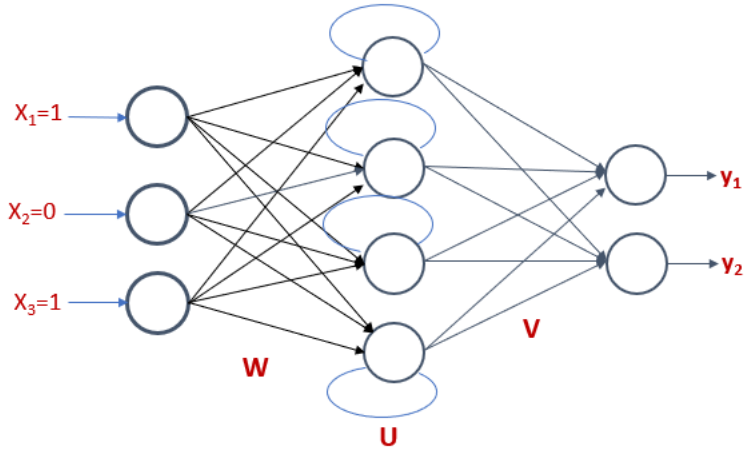
- Ease of use (the built-in API):
- ▪ **keras.layers.SimpleRNN**
- ▪ **keras.layers.LSTM**
- ▪ **keras.layers.GRU layers**
- Ease of customization: **keras.layers.RNN**

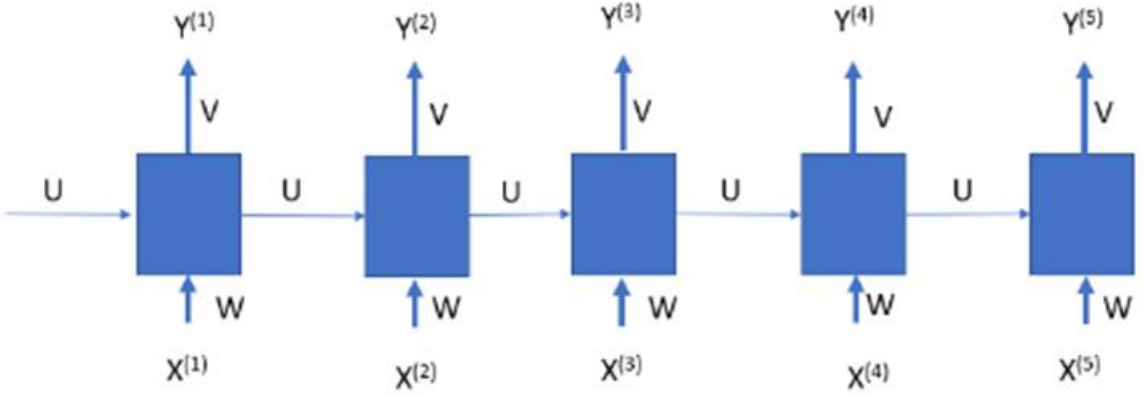
Implementation of RNN - Keras

```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=True, use_bias=False))
model.add(Dense(2, use_bias=False))
model.summary()
```

Implementation of RNN - Keras

```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=True, use_bias=False))
model.add(Dense(2, use_bias=False))
model.summary()
```





Implementation of RNN - Keras

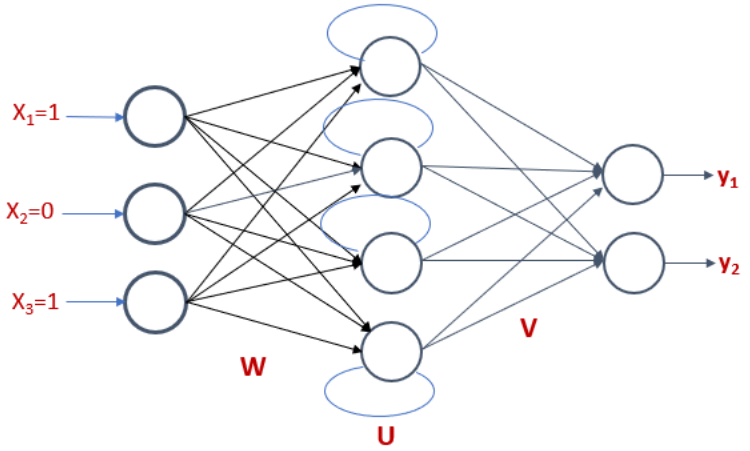
```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=True, use_bias=False))
model.add(Dense(2, use_bias=False))
model.summary()
```

```
for x in model.layers[0].weights:
    print(x.shape)
```

```
(3, 128)
(128, 128)
```

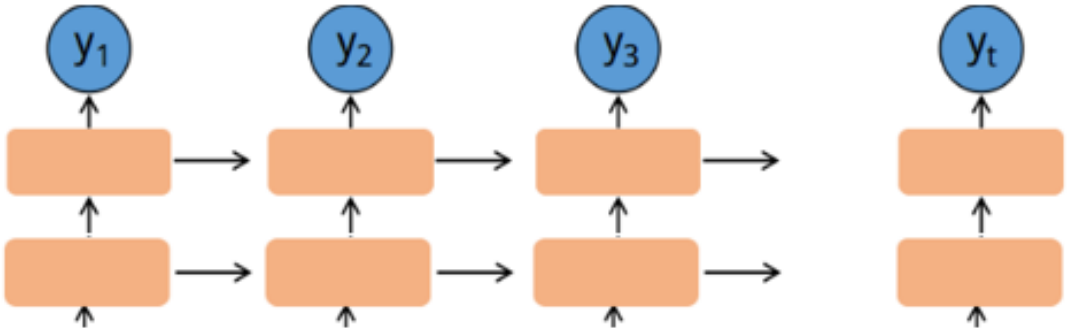
```
for x in model.layers[1].weights:
    print(x.shape)
```

```
(128, 2)
```



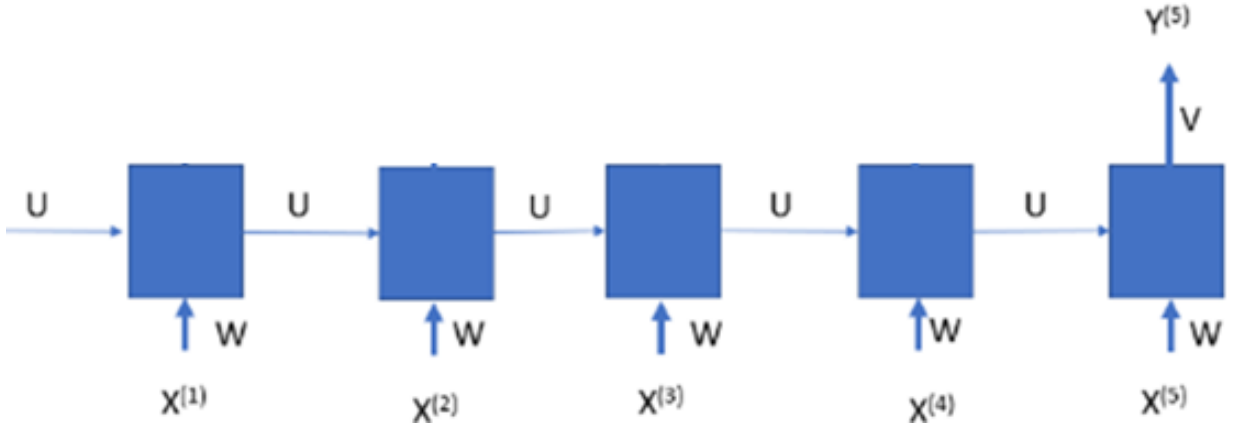
Implementation of RNN - Keras

```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=True, use_bias=False))
model.add(SimpleRNN(64, return_sequences=True, use_bias=False))
model.add(Dense(2, use_bias=False))
model.summary()
```



Implementation of RNN - Keras

```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=False, use_bias=False))
model.add(Dense(2, use_bias=False))
model.summary()
```



Implementation of RNN - Keras

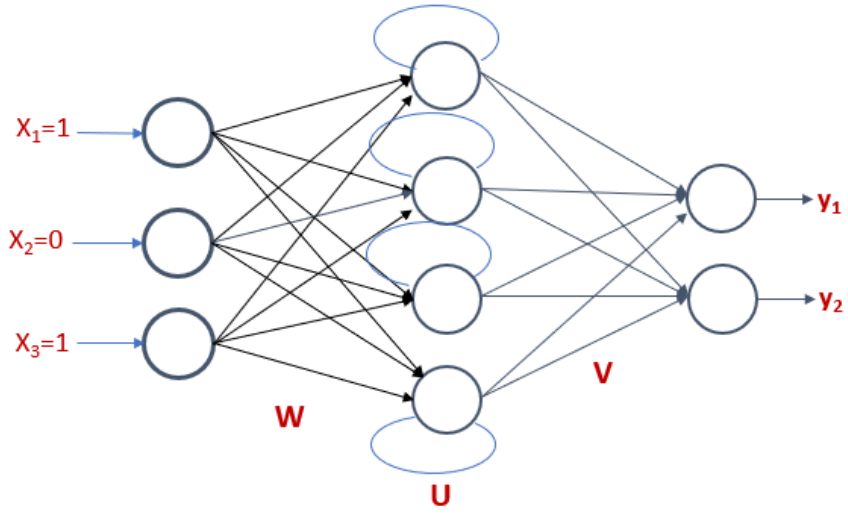
```
from keras.models import Sequential
from keras.layers import SimpleRNN
from keras.layers import Dense

model = Sequential()
model.add(SimpleRNN(128,input_shape = (5,3), return_sequences=True, use_bias=True))
model.add(Dense(2, use_bias=True))
model.summary()
```

```
for x in model.layers[0].weights:
    print(x.shape)
```

```
(3, 128)
(128, 128)
(128,)
```

```
for x in model.layers[1].weights:
    print(x.shape)
```

```
(128, 2)
(2,)
```



Implementation of LSTM - Keras

```
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense

model = Sequential()
model.add(LSTM(128,input_shape = (5,10), return_sequences=True, use_bias=False))
model.add(Dense(2))
model.summary()
```

Implementation of GRU - Keras

```
from keras.models import Sequential
from keras.layers import GRU
from keras.layers import Dense

model = Sequential()
model.add(GRU(128,input_shape = (5,10), return_sequences=True, use_bias=False))
model.add(Dense(2))
model.summary()
```