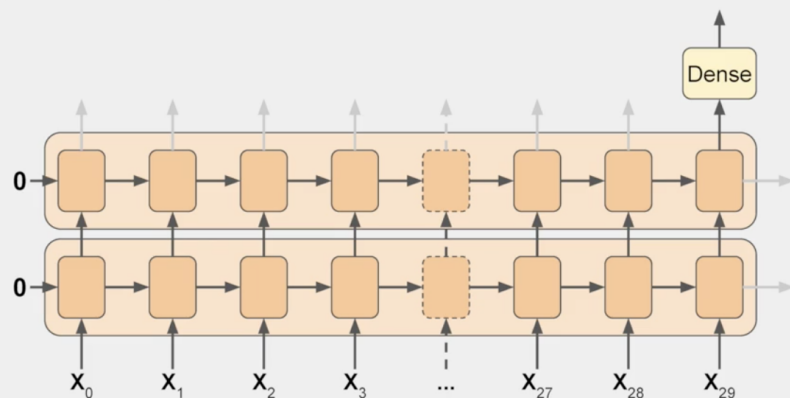


Outputting a sequence

Consider the following architecture. The first RNN layer feeds another, so we need to return the entire sequence. The second layer feeds a single Dense network, so we don't need to return sequence (default).

- TensorFlow assumes that the first dimension is the batch size. It assumes that I can have any size at all, so I don't need to define it.
- The second dimension is the length of the sequence, and is defined as None (in the red box below) because we can have sequences of any length.
- The last dimension is 1, because we are using a univariate time series.

```
model = keras.models.Sequential([  
    keras.layers.SimpleRNN(20, return_sequences=True,  
                             input_shape=[None, 1]),  
    keras.layers.SimpleRNN(20),  
    keras.layers.Dense(1)  
])
```

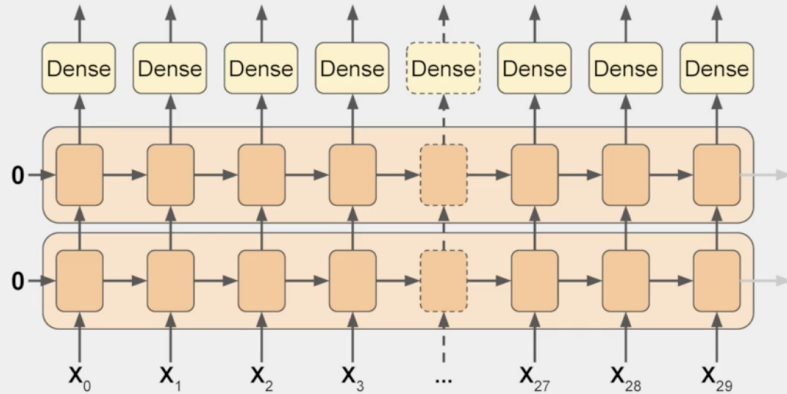


If we set both RNN layers to return the entire sequence, Keras uses the same Dense layer independently for each time step. It does not have a separate or unique Dense layer defined for each time step.

```

model = keras.models.Sequential([
    keras.layers.SimpleRNN(20, return_sequences=True,
                           input_shape=[None, 1]),
    keras.layers.SimpleRNN(20, return_sequences=True),
    keras.layers.Dense(1)
])

```



This is known as a sequence to sequence RNN. It takes a batch of sequences as input and outputs a batch of sequences of the same length. The dimensionality may not be the same as it depends on the number of neurons (features) in the memory cell.