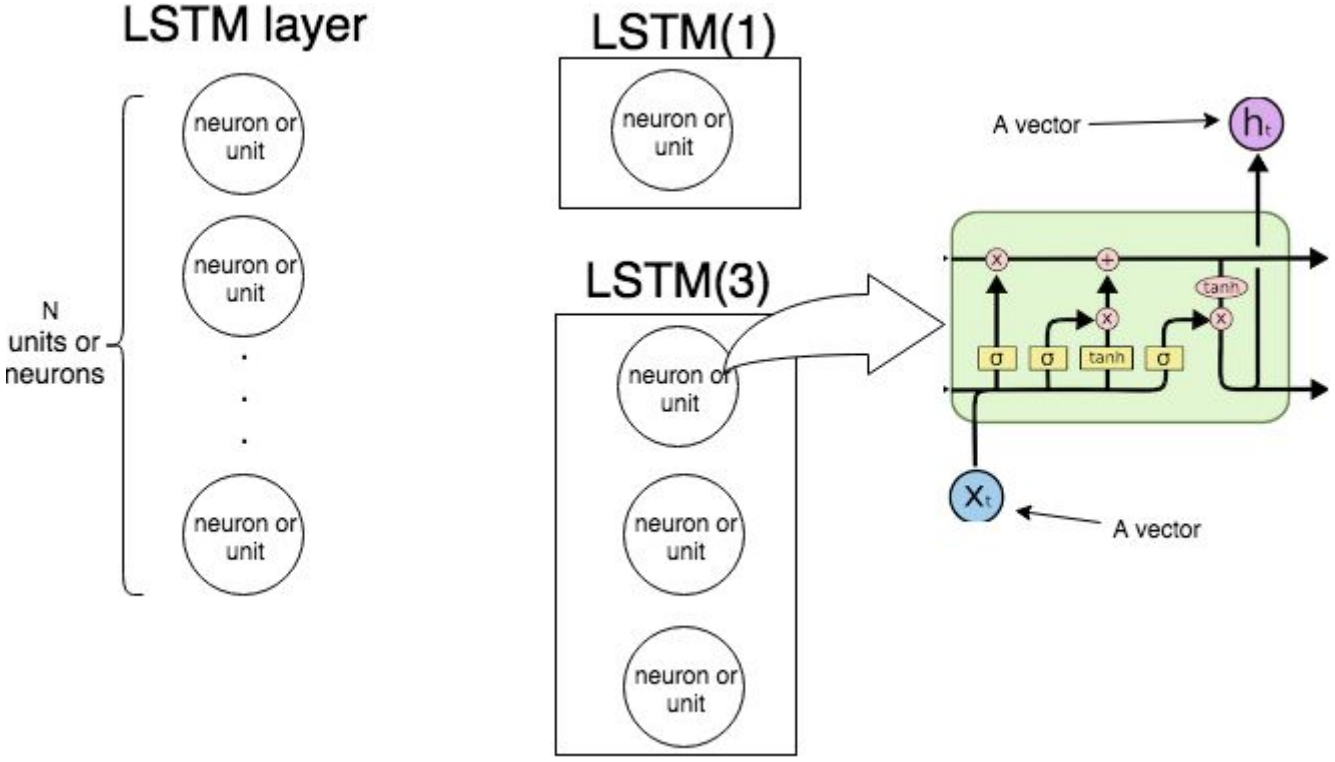
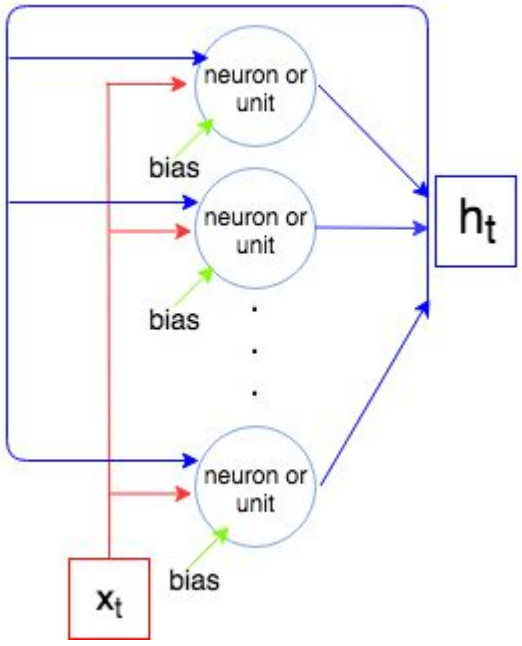
In Keras `LSTM(n)` means "create an LSTM layer consisting of LSTM units. The following picture demonstrates what layer and unit (or neuron) are, and the rightmost image shows the internal structure of a single LSTM unit.



The following picture shows how the whole LSTM layer operates.



As we know an LSTM layer processes a sequence, i.e, $x_1,\ldots,x_N$x1,…,xN. At each step tt the layer (each neuron) takes the input $x_t$xt, output from previous step h_{t-1}ht−1, and bias bb, and outputs a vector $h_t$ht. Coordinates of $h_t$ht are outputs of the

neurons/units, and hence the size of the vector h_t is equal to the number of units/neurons. This process continues until x_N.

Now let's compute the number of parameters for `LSTM(1)` and `LSTM(3)` and compare it with what Keras shows when we call `model.summary()`.

Let inp be the size of the vector x_t and out be the size of the vector h_t (this is also the number of neurons/units). Each neuron/unit takes input vector, output from the previous step, and a bias which makes $inp+out+1$ parameters (weights). But we have out number of neurons and so we have $out \times (inp+out+1)$ parameters. Finally each unit has 4 weights (see the rightmost image, yellow boxes) and we have the following formula for the number of parameters:

$$4out(inp+out+1)$$

Let's compare with what Keras outputs.

Example 1.

```
t1 = Input(shape=(1, 1))
t2 = LSTM(1)(t1)
model = Model(inputs=t1, outputs=t2)
print(model.summary())
```

```
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 1, 1)              0
_________________________________________________________________
lstm_2 (LSTM)                (None, 1)                 12
=================================================================
Total params: 12
Trainable params: 12
Non-trainable params: 0
_________________________________________________________________
```

Number of units is 1, the size of input vector is 1, so $4 \times 1 \times (1+1+1)=12$.

Example 2.

```
input_t = Input((4, 2))
output_t = LSTM(3)(input_t)
model = Model(inputs=input_t, outputs=output_t)
print(model.summary())
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         (None, 4, 2)              0
_________________________________________________________________
lstm_6 (LSTM)                (None, 3)                 72
=================================================================
Total params: 72
Trainable params: 72
Non-trainable params: 0
```

Number of units is 3, the size of the input vector is 2, so $4 \times 3 \times (2+3+1)=72$