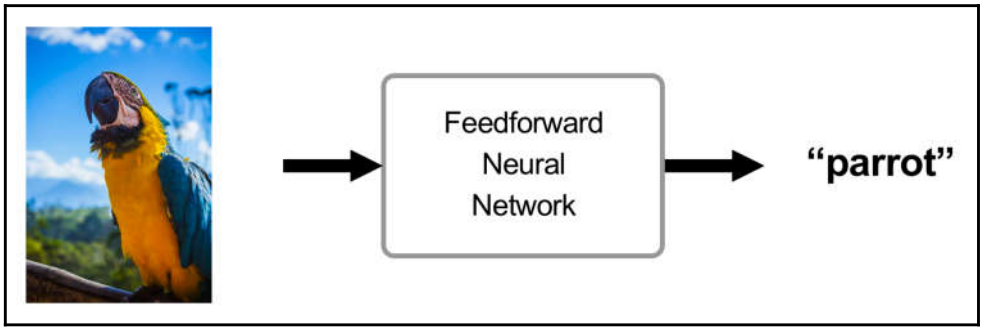
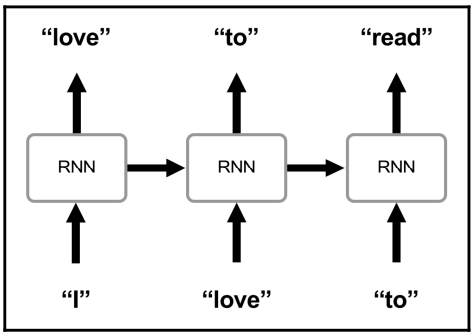
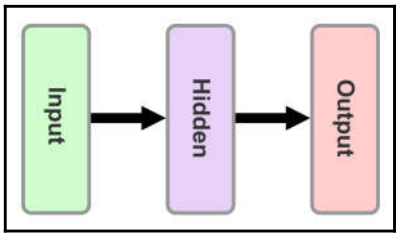
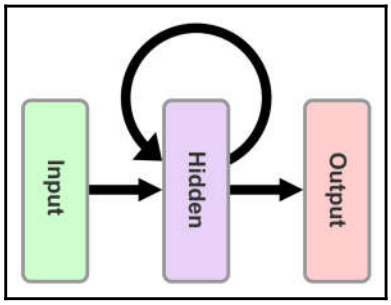
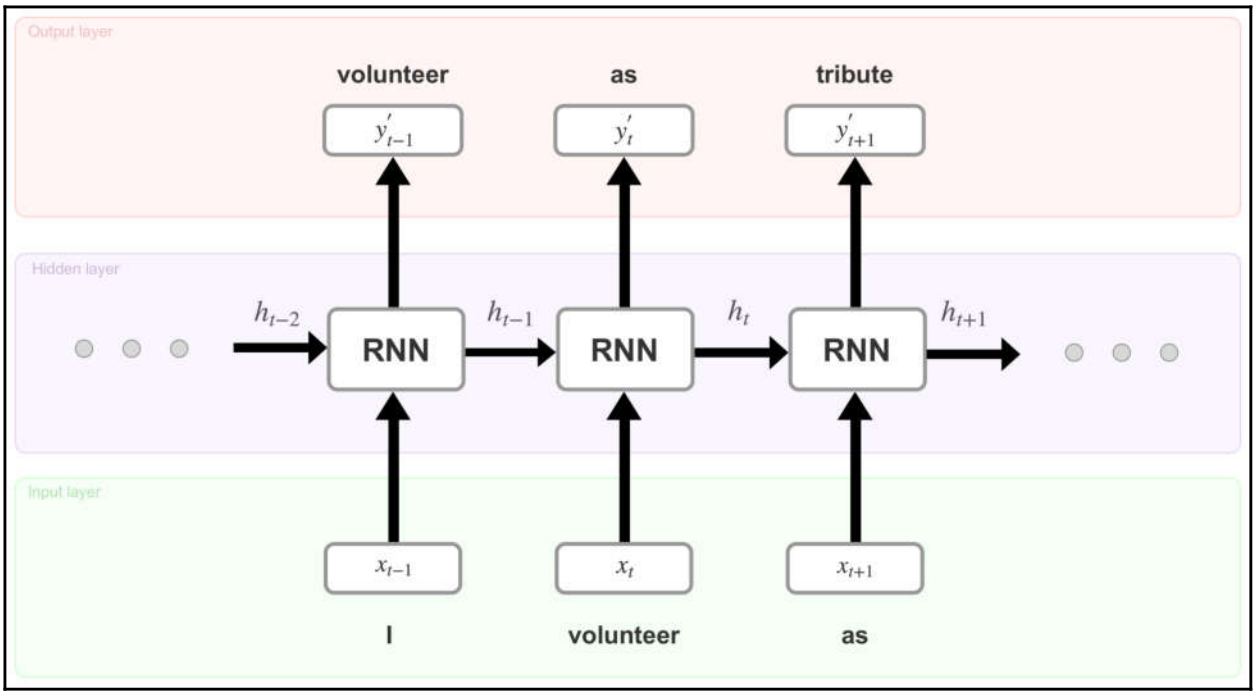
Chapter 1:
Introducing Recurrent Neural Networks

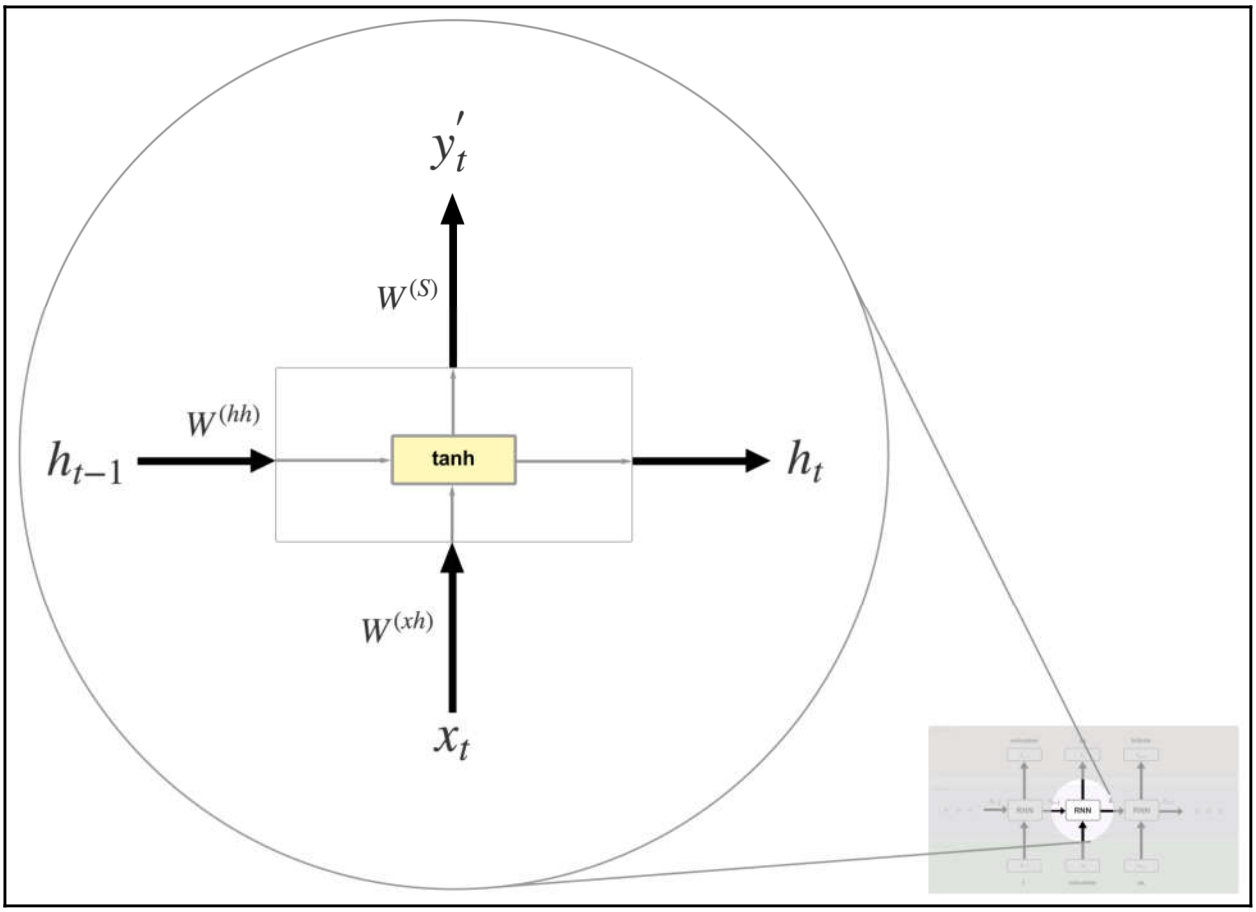









Output layer
volunteer
as
tribute
y'_{t-1}
y'_t
y'_{t+1}
Hidden layer
h_{t-2}
RNN
h_{t-1}
RNN
h_t
RNN
h_{t+1}
Input layer
x_{t-1}
x_t
x_{t+1}
I
volunteer
as



$$h_t = tanh(W^{(hh)} * h_{t-1} + W^{(hx)} * x_t + b^h)$$

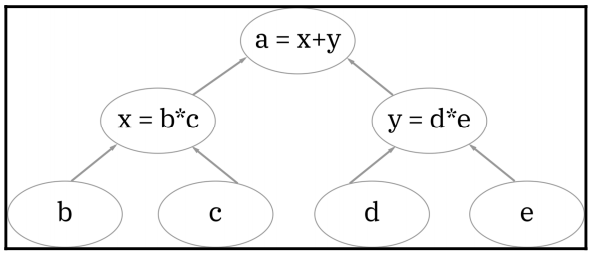
$$y_t' = softmax(W^{(S)} * h_t + b^S)$$

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

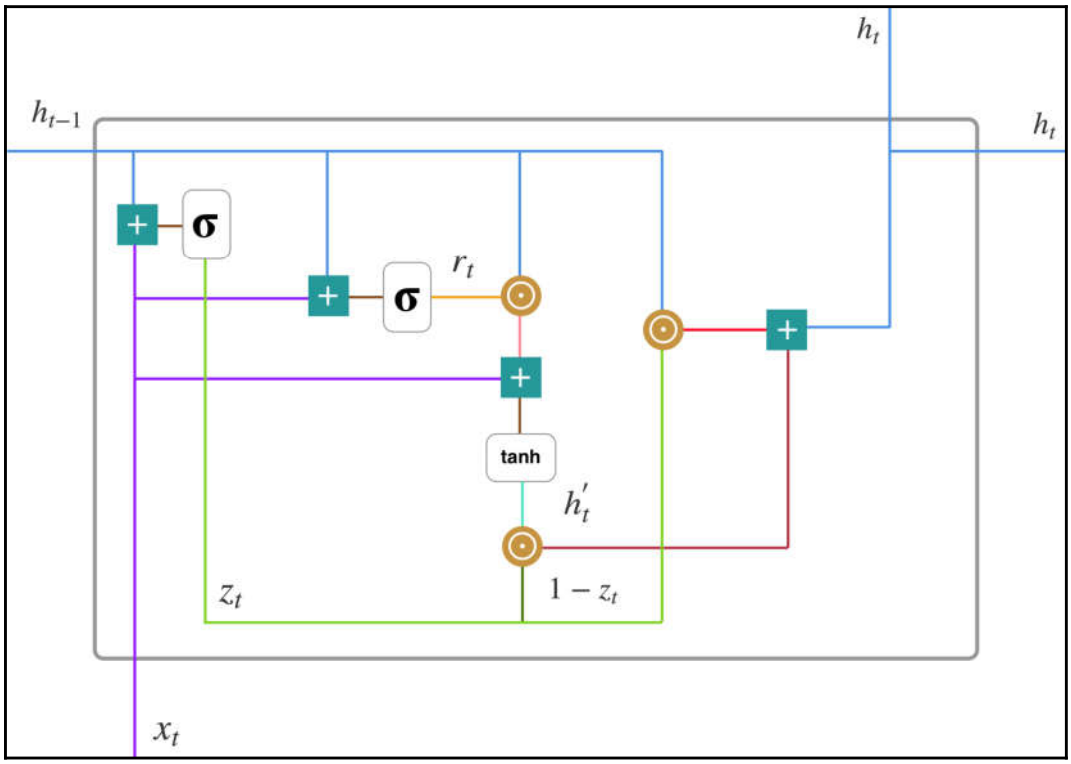
$$softmax(z)_i = \frac{e^{z_i}}{e^{z_1} + e^{z_2} + \cdots + e^{z_K}}$$

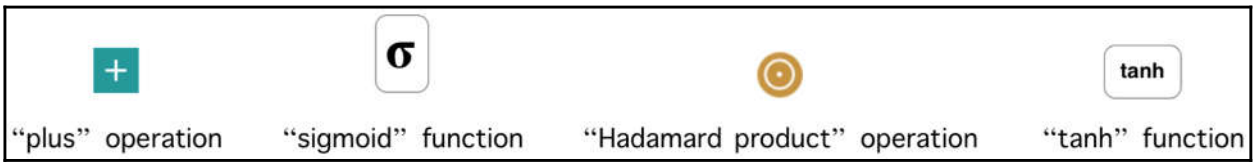
$$J(y, y') = -\sum y_i * log(y_i')$$

Chapter 2:
Building Your First RNN with TensorFlow



Chapter 3:
Generating Your Own Book Chapter



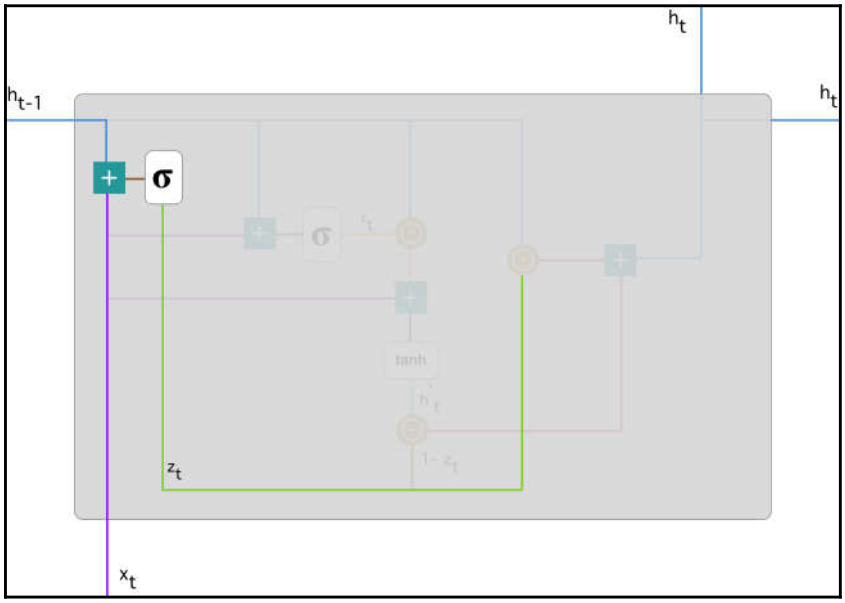


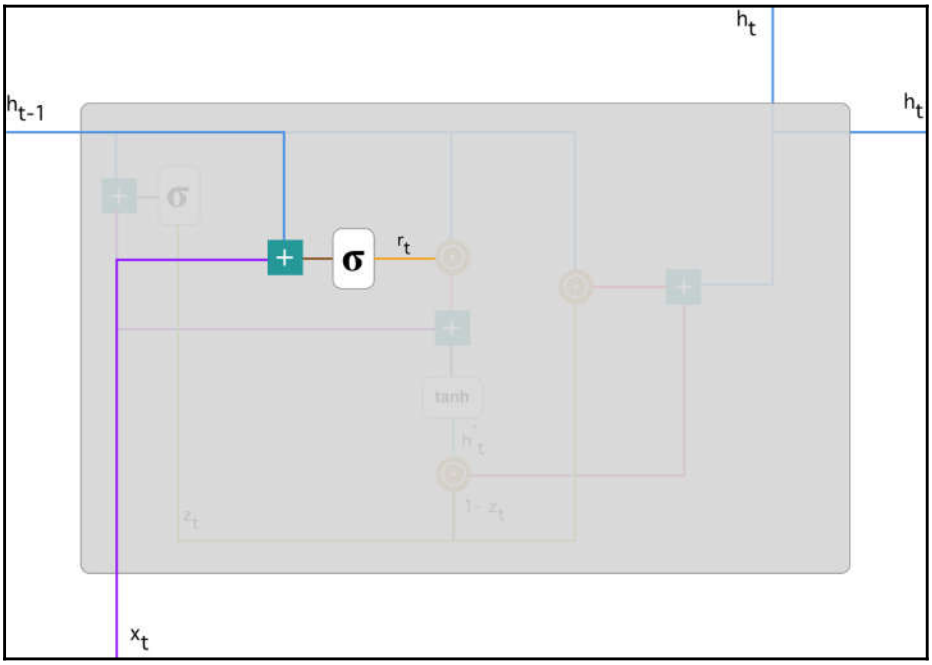
$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1})$$

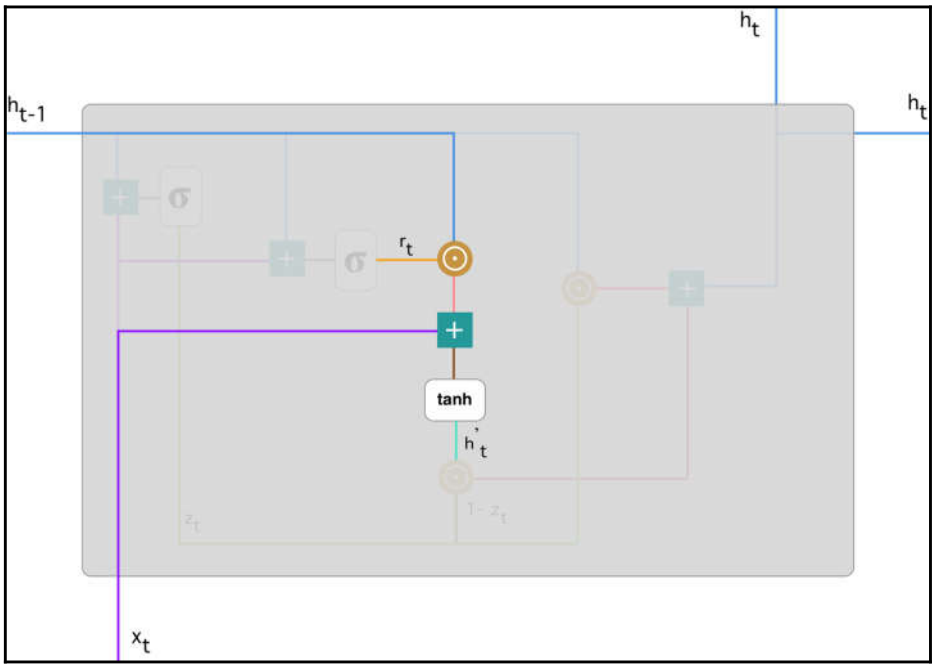
$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1})$$

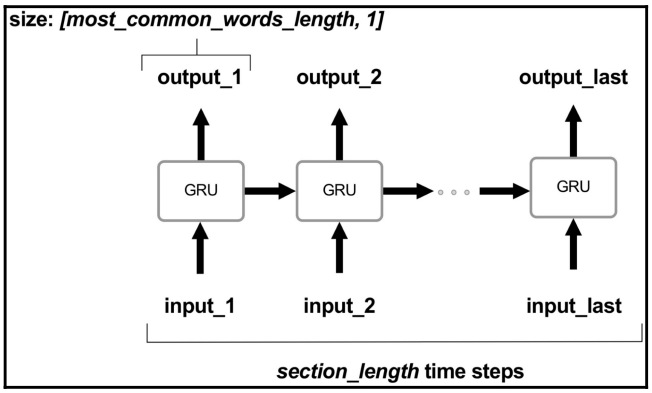
$$h_t^{'} = tanh(W x_t + r_t \odot U h_{t-1})$$

$$\boxed{h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t^{'}}$$

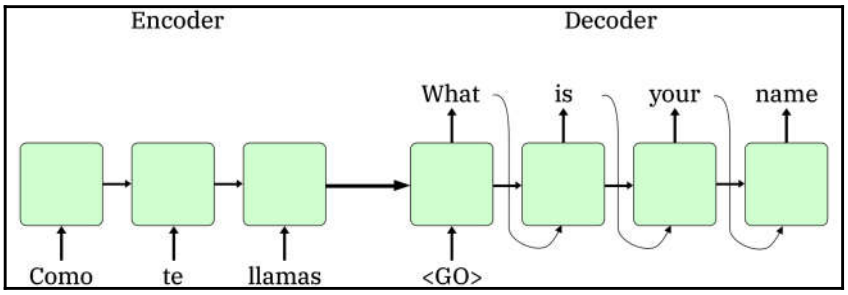


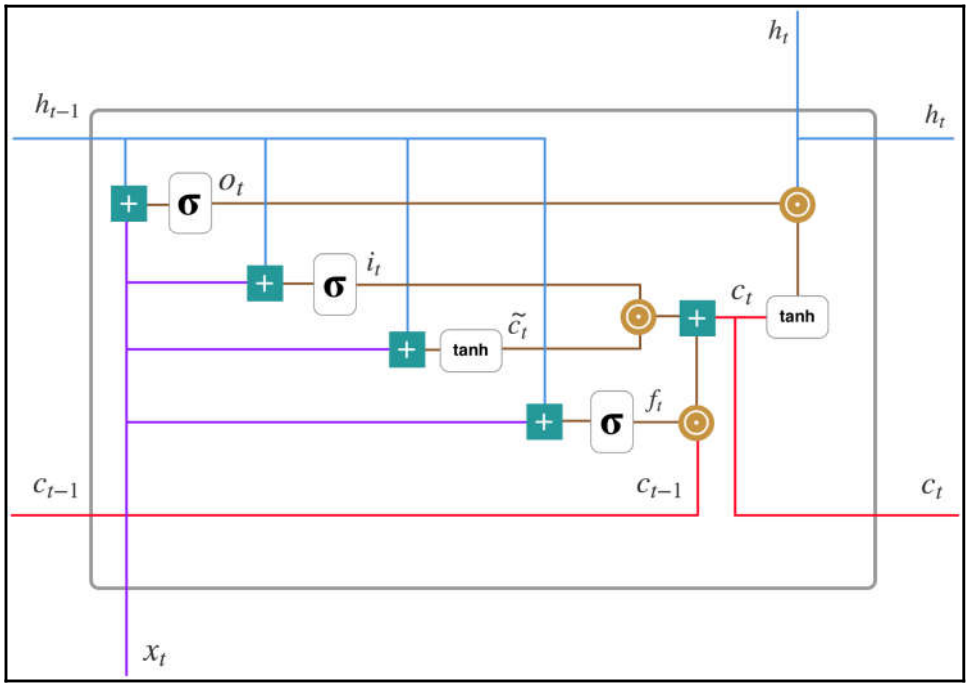



h_t
h_{t-1}
h_t
σ
σ
r_t
tanh
h'_t
z_t
$1-z_t$
x_t


size: [most_common_words_length, 1]
output_1
output_2
output_last
GRU
GRU
GRU
input_1
input_2
input_last
section_length time steps

Chapter 4: Creating a Spanish-to-English Translator





$$o_t = \sigma(W^{(o)} x_t + U^{(o)} h_{t-1})$$

$$i_t = \sigma(W^{(i)} x_t + U^{(i)} h_{t-1})$$

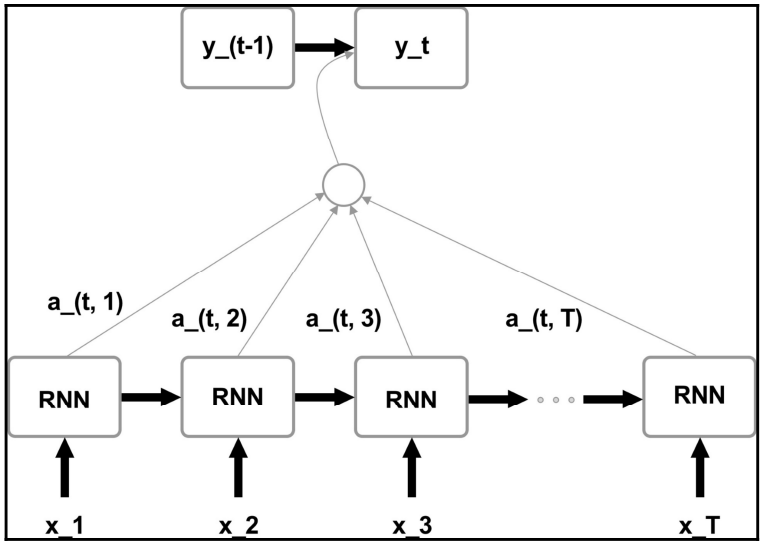
$$\boxed{\tilde{c}_t = \tanh\left(W^{(c)} x_t + U^{(c)} h_{t-1}\right)}$$

$$\boxed{f_t = \sigma(W^{(f)} x_t + U^{(f)} h_{t-1})}$$

$$\boxed{o_t, i_t, \tilde{c}_t, f_t}$$

$$\boxed{c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t}$$

$$\boxed{h_t = o_t \circ tanh(c_t)}$$

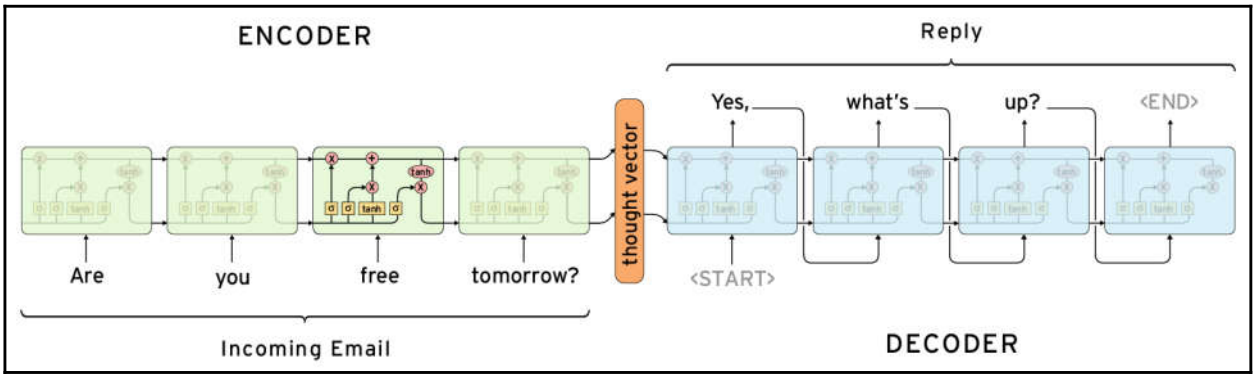


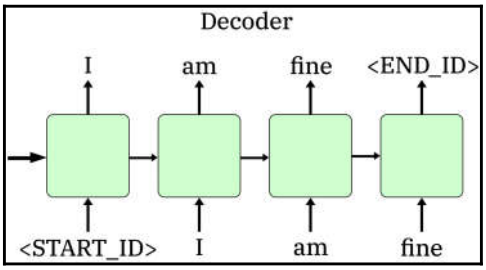
$$\boxed{x_1, x_2, x_3, \ldots, x_t}$$

$$\boxed{\ldots, y_{t-1}, y_t, \ldots}$$

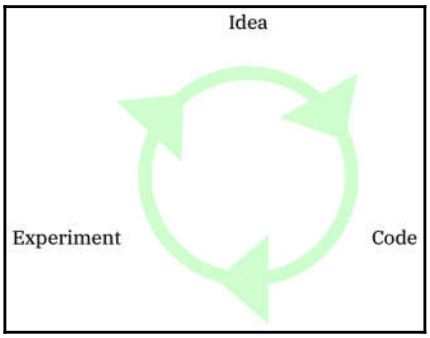
$$\boxed{a_{t,1}, a_{t,2}, a_{t,3}, \ldots, a_{t,T}}$$

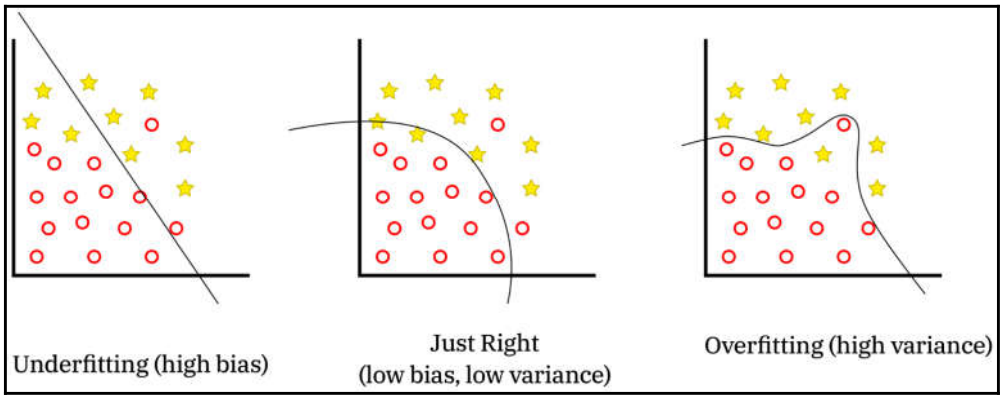
Chapter 5: Building Your Personal Assistant

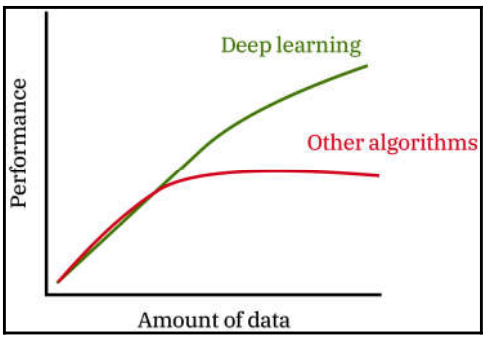


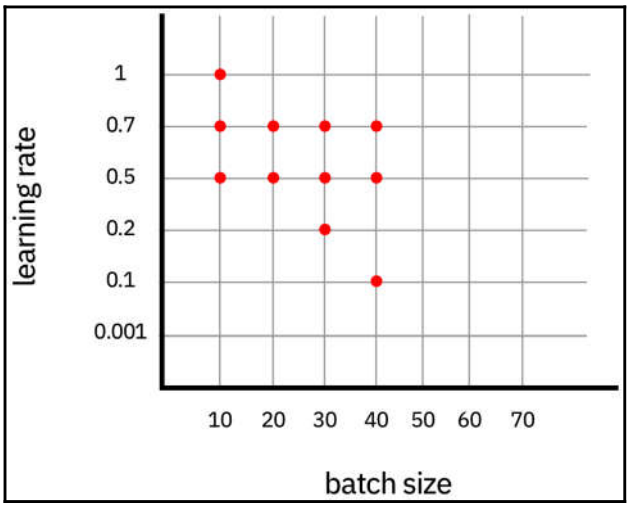


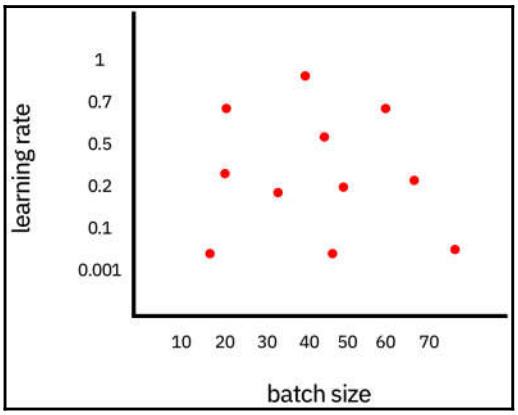
Chapter 6:
Improving Your RNN Performance

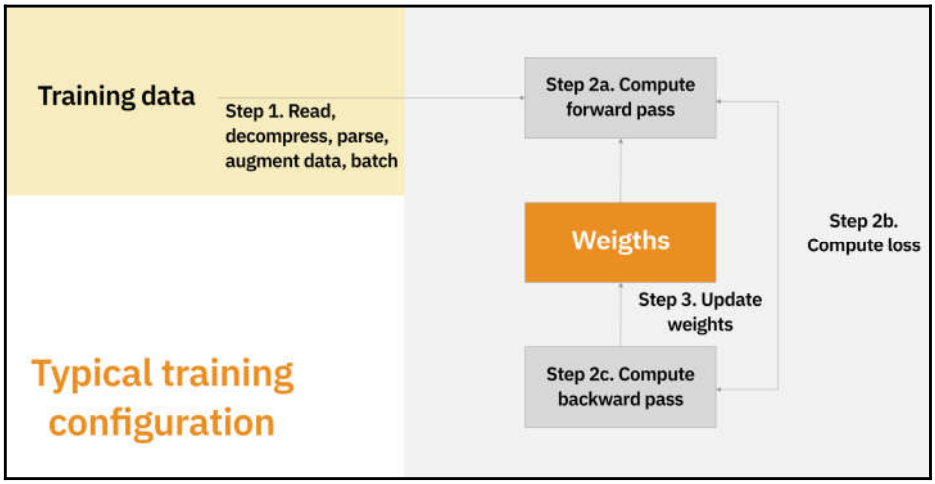







learning rate
1
0.7
0.5
0.2
0.1
0.001
10 20 30 40 50 60 70
batch size


learning rate
1
0.7
0.5
0.2
0.1
0.001
10 20 30 40 50 60 70
batch size


Training data
Step 1. Read, decompress, parse, augment data, batch
Step 2a. Compute forward pass
Weigths
Step 2b. Compute loss
Step 3. Update weights
Step 2c. Compute backward pass
Typical training configuration