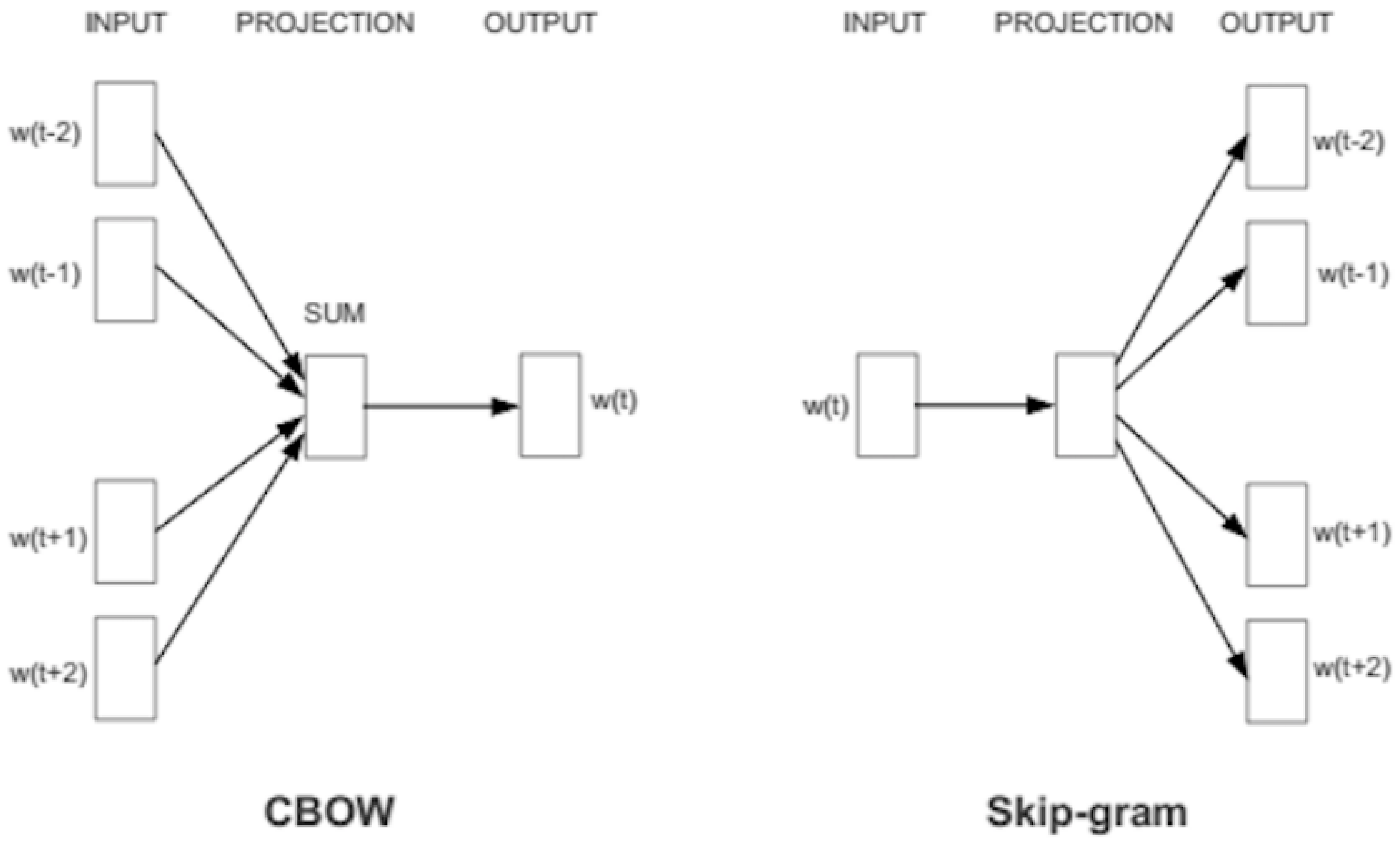
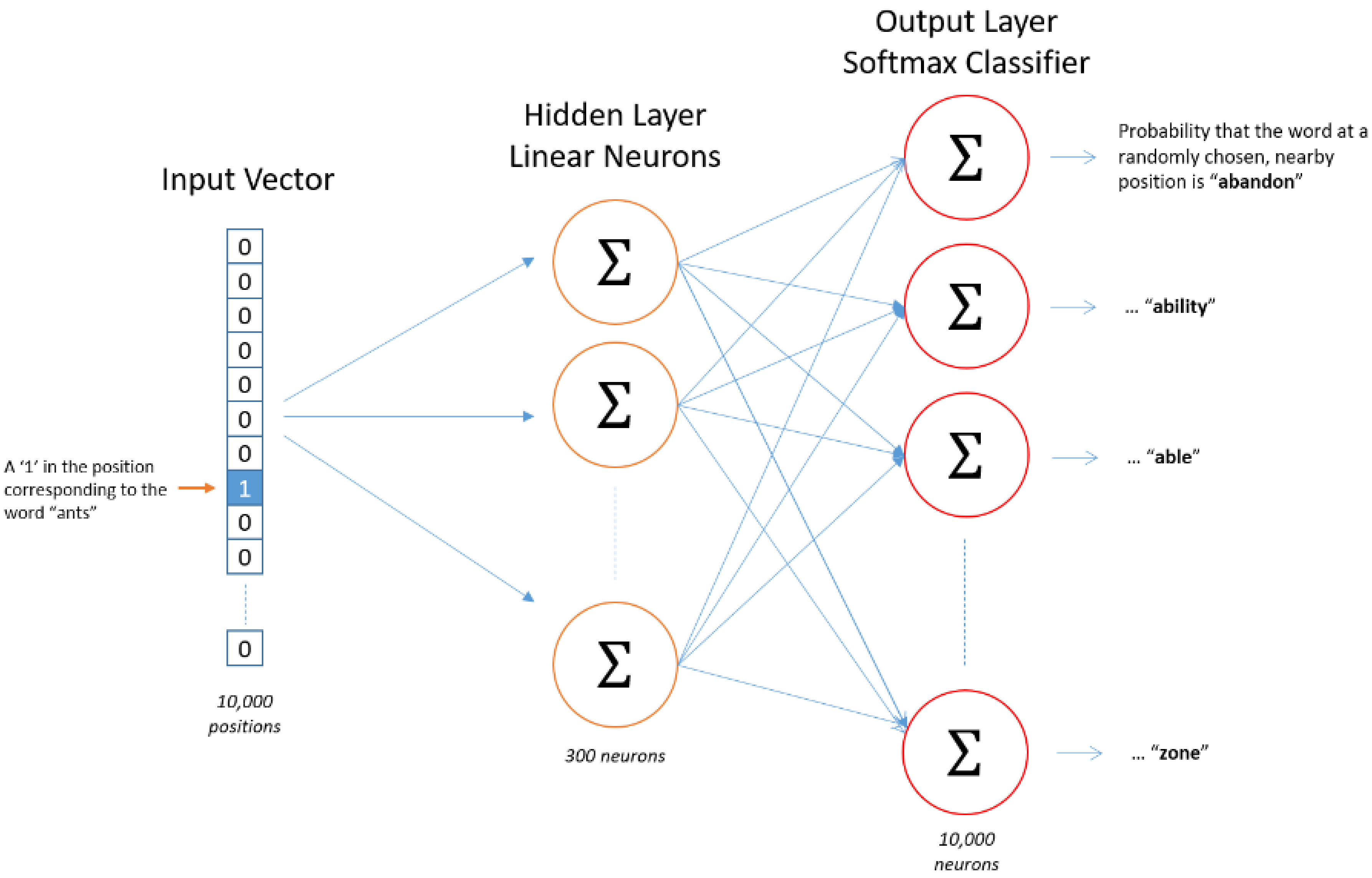
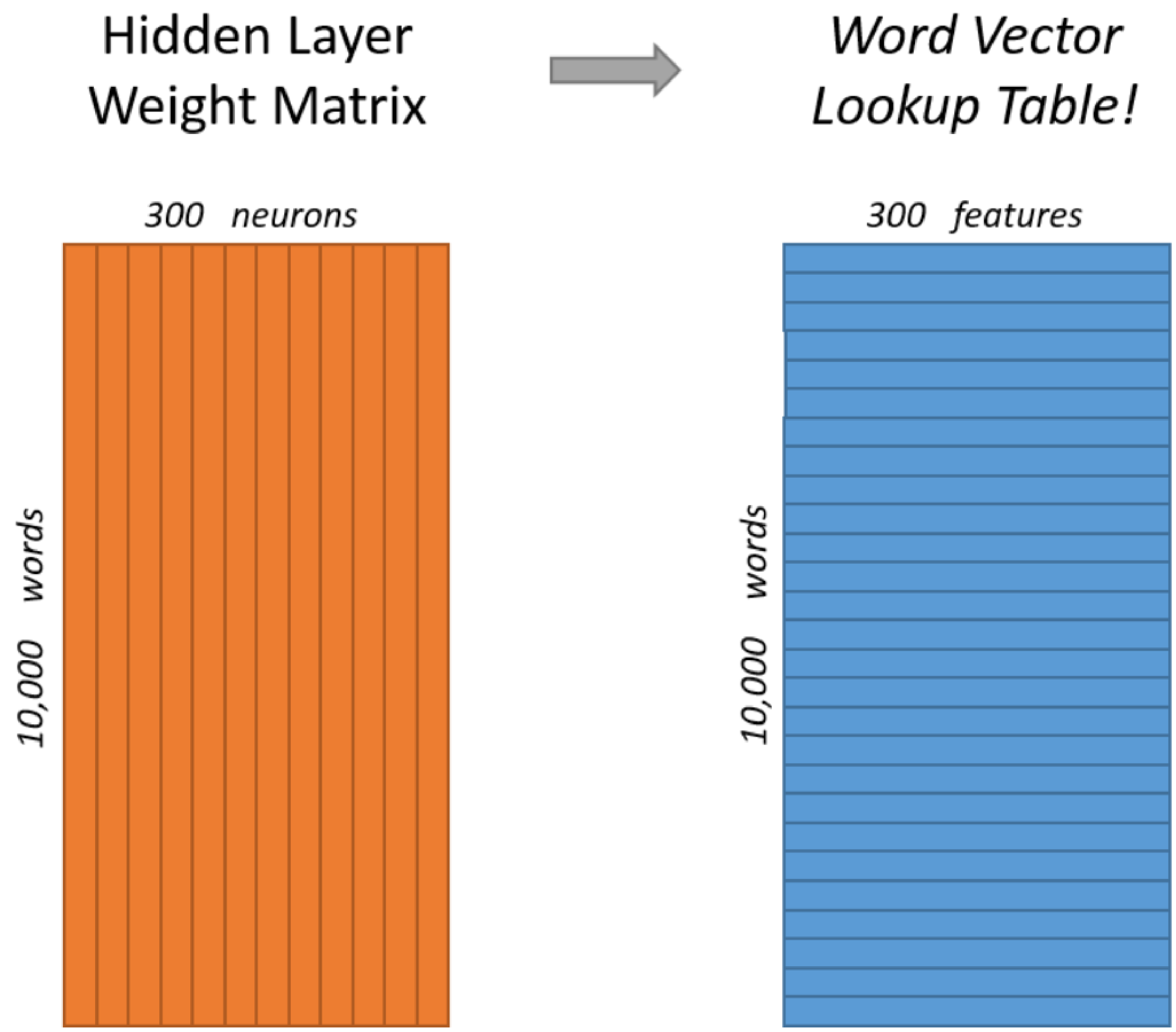
Word2Vec


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram



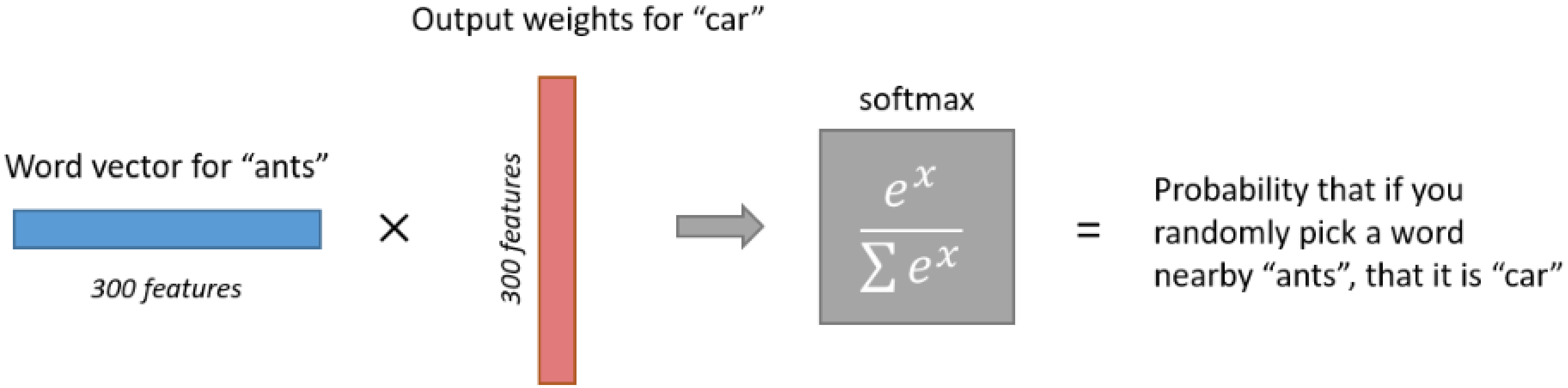
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
Hidden Layer
Linear Neurons
Σ
300 neurons
Output Layer
Softmax Classifier
Σ
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
10,000 neurons

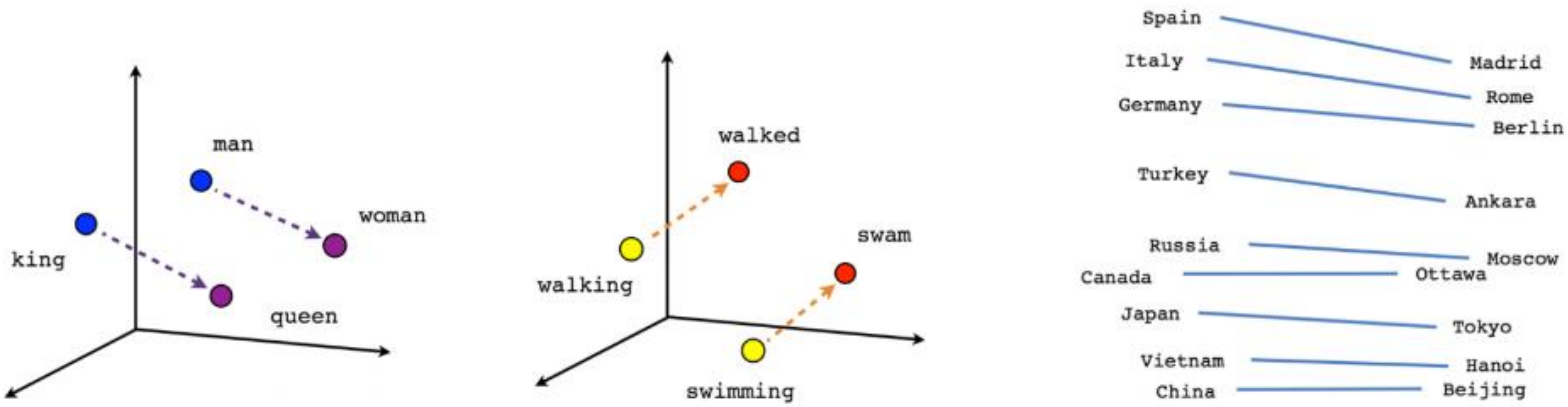

Hidden Layer
Weight Matrix
Word Vector
Lookup Table!
300 neurons
10,000 words
300 features
10,000 words

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Softmax activation function

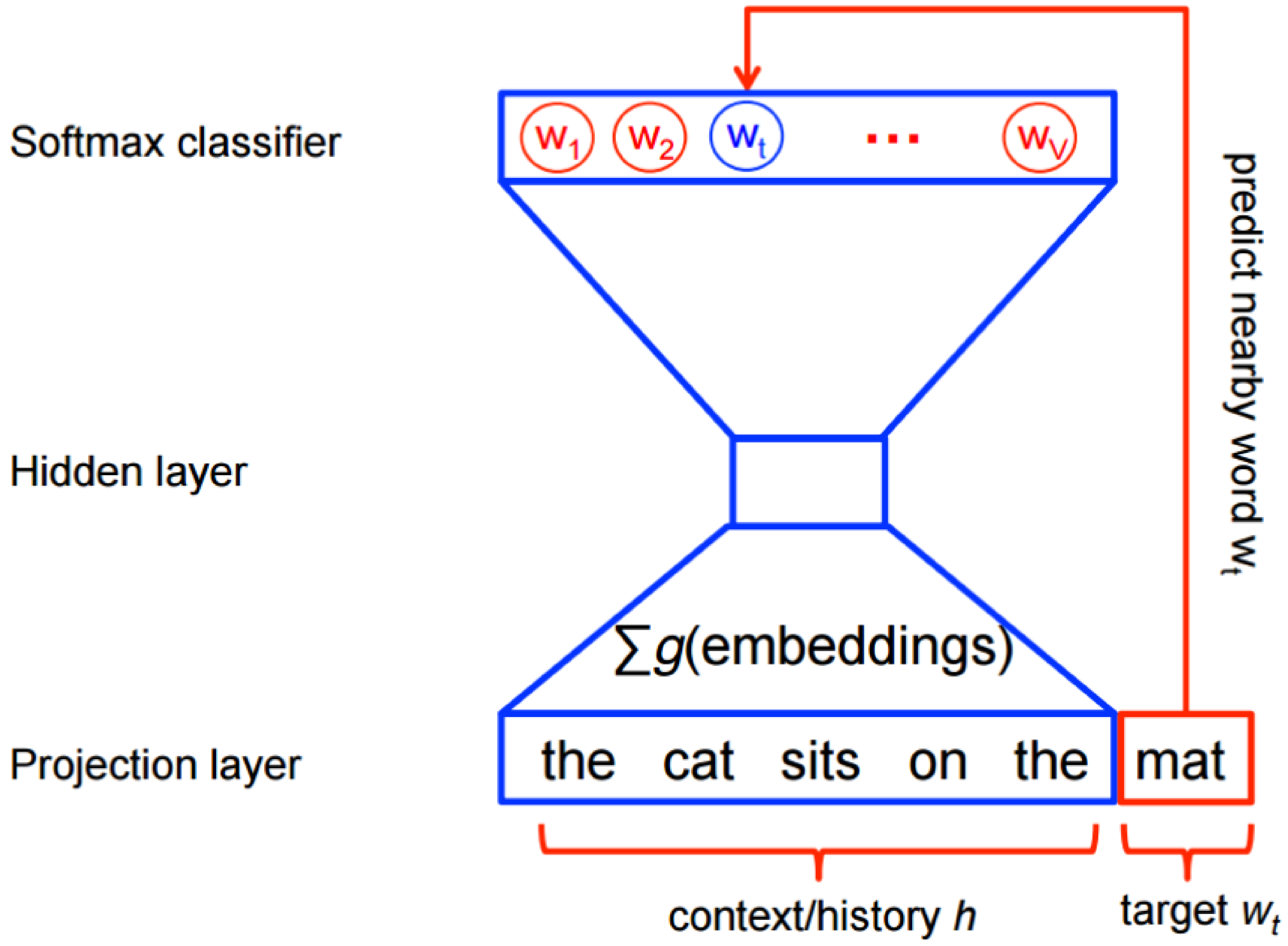




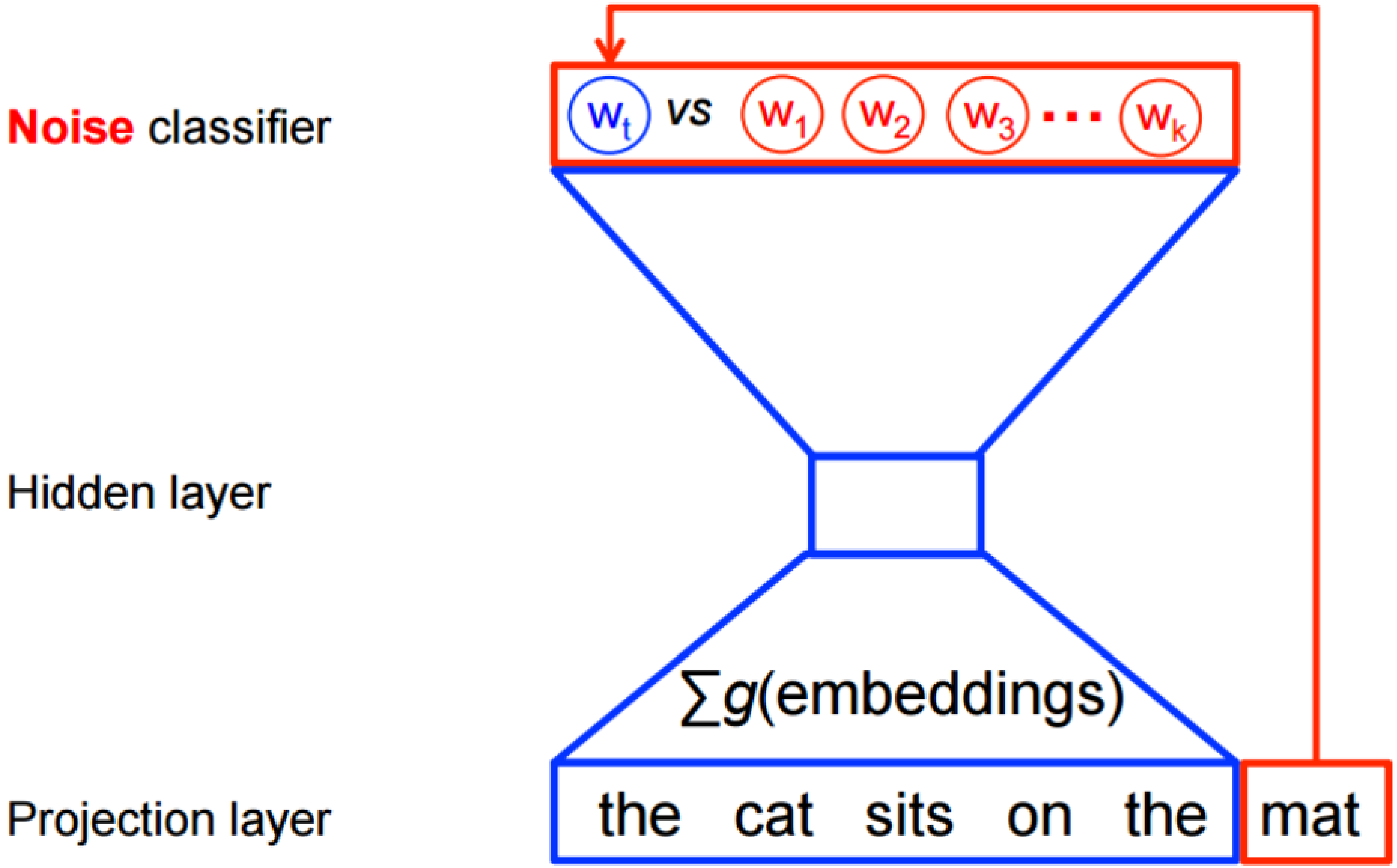
Male-Female

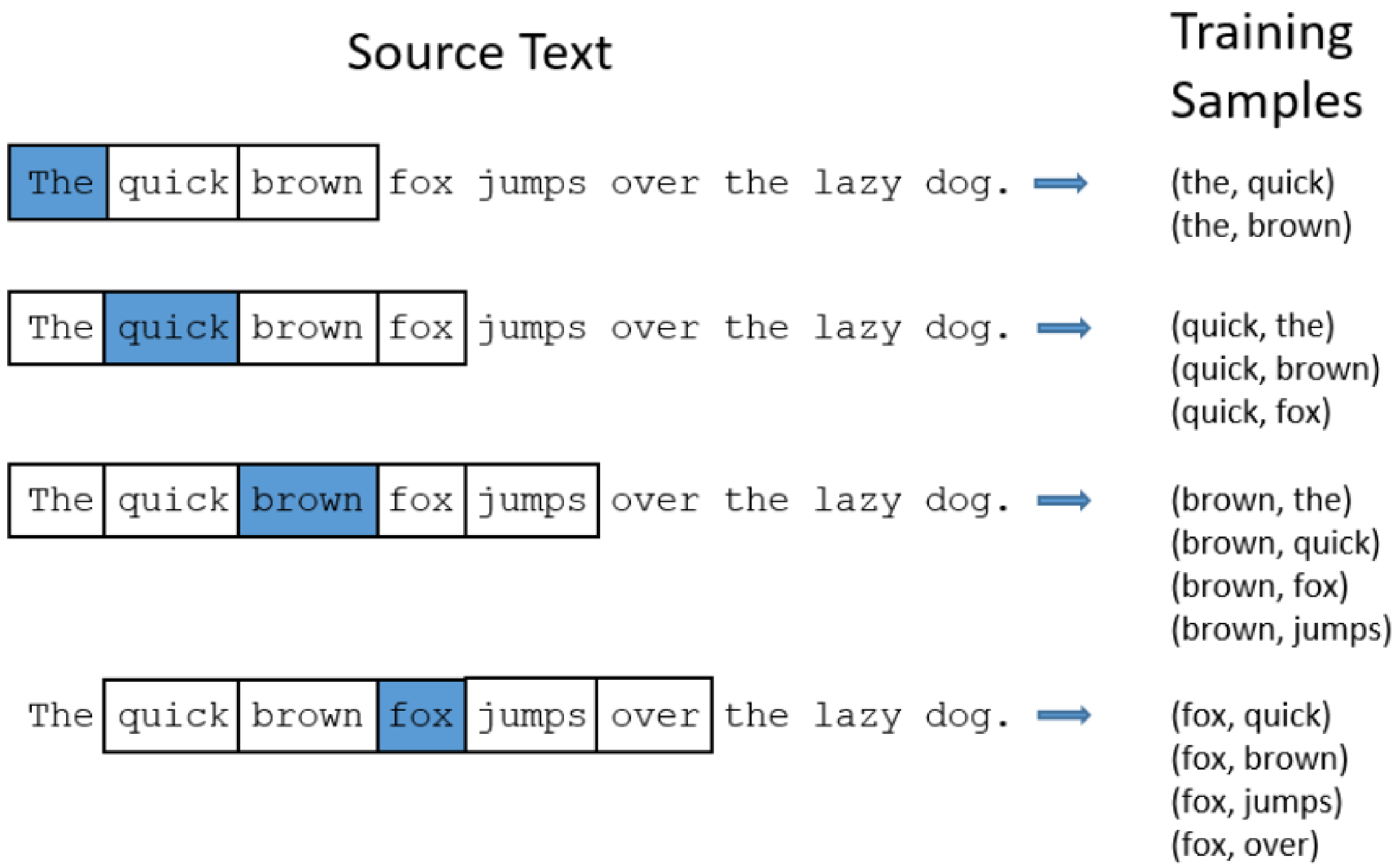
Verb tense

Country-Capital



Softmax classifier
Hidden layer
Projection layer
w1 w2 wt ... wV
∑g(embeddings)
the cat sits on the
mat
context/history h
target wt
predict nearby word wt


Noise classifier
Hidden layer
Projection layer
w_t vs w_1 w_2 w_3 ... w_k
∑g(embeddings)
the cat sits on the
mat


Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

GloVe

X_{ij} tabulate the number of times word j occurs in the context of word i.

$X_i = \sum_k X_{ik}$

$P_{ij} = P(j|i) = X_{ij}/X_i$

$w \in \mathbb{R}^d$ are word vectors

probe word

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

co-relations between the word wi and wj

co-occurrence probabilities for the word wj and wk

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Very small or large:

solid is related to ice but not steam, or
gas is related to steam but not ice

close to 1:

water is highly related to ice and steam, or
fashion is not related to ice or steam.

$w_i{}^T \tilde{w}_k$ relate to (high probability if they are similar)

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

$w_j{}^T \tilde{w}_k$

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

co-occurrence count for word w*i* and w*k*


Love in Venice
Normandy
Dark night
Detective Bob
4 1 4 2
1 5 ? ?
5 ? 4 ?

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & ? & ? & \dots & r_{1n} \\ r_{21} & r_{22} & ? & r_{24} & ? & \dots & ? \\ & & \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{m1} & ? & r_{m3} & ? & r_{m5} & \dots & r_{mn} \end{bmatrix} \approx \begin{bmatrix} z_{11} & z_{12} & z_{13} & \dots & z_{1k} \\ z_{21} & z_{22} & z_{23} & \dots & z_{2k} \\ & & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & \dots & z_{mk} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1k} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2k} \\ & & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nk} \end{bmatrix}^T$$

$$J(W, Z) = \sum_i \sum_j (W_j^T z_i - r_{ij})^2 + \frac{\lambda_1}{2} \|W\|_f^2 + \frac{\lambda_2}{2} \|Z\|_f^2$$

$$\underline{w_i^T \tilde{w}_k} + b_i + \tilde{b}_k = \log(X_{ik})$$

measures the similarity of the hidden factors between both words to predict co-occurrence count

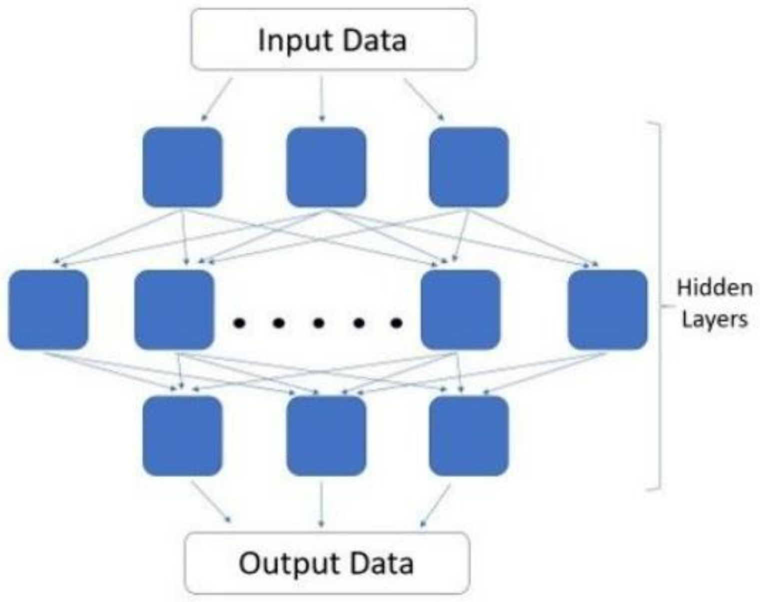
$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

100 3/4

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

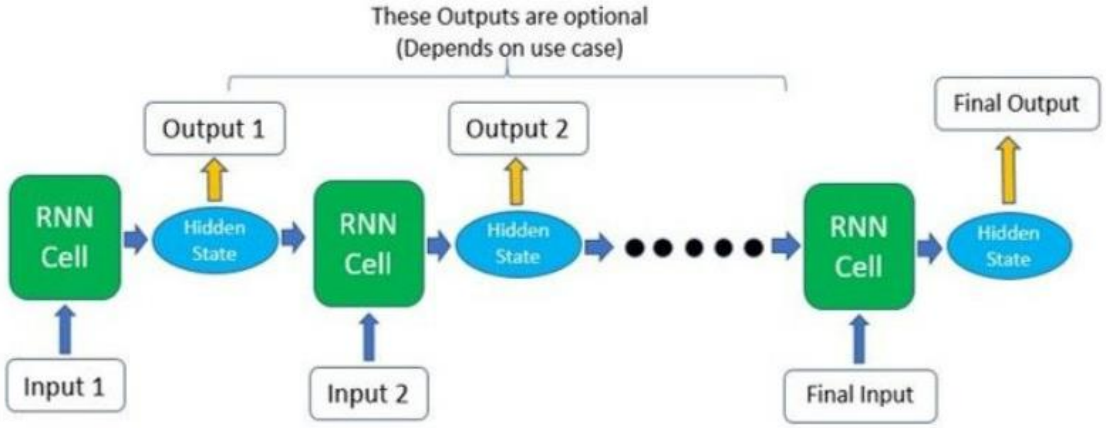
Recurrent Neural Networks (RNN)

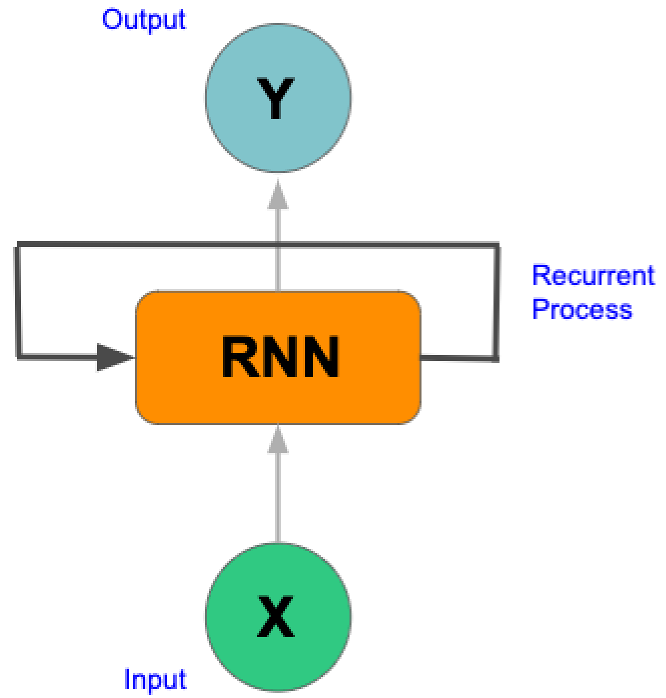
Traditional Feed-Forward Network

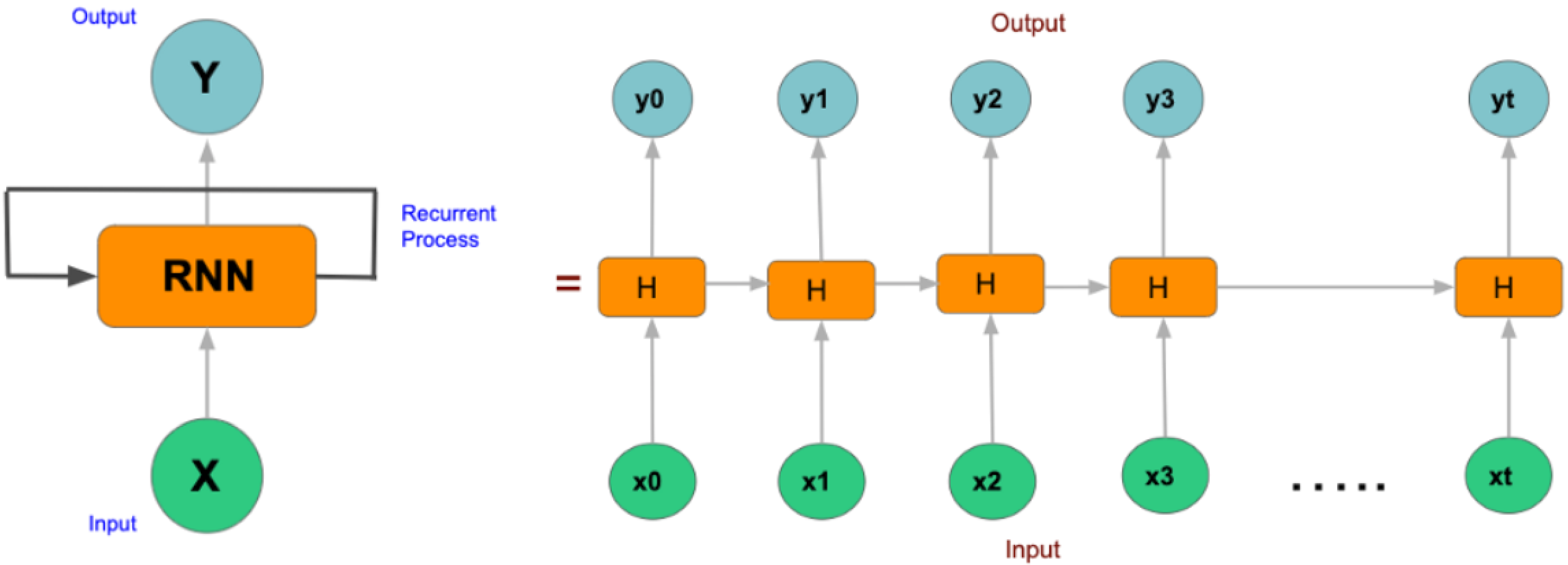


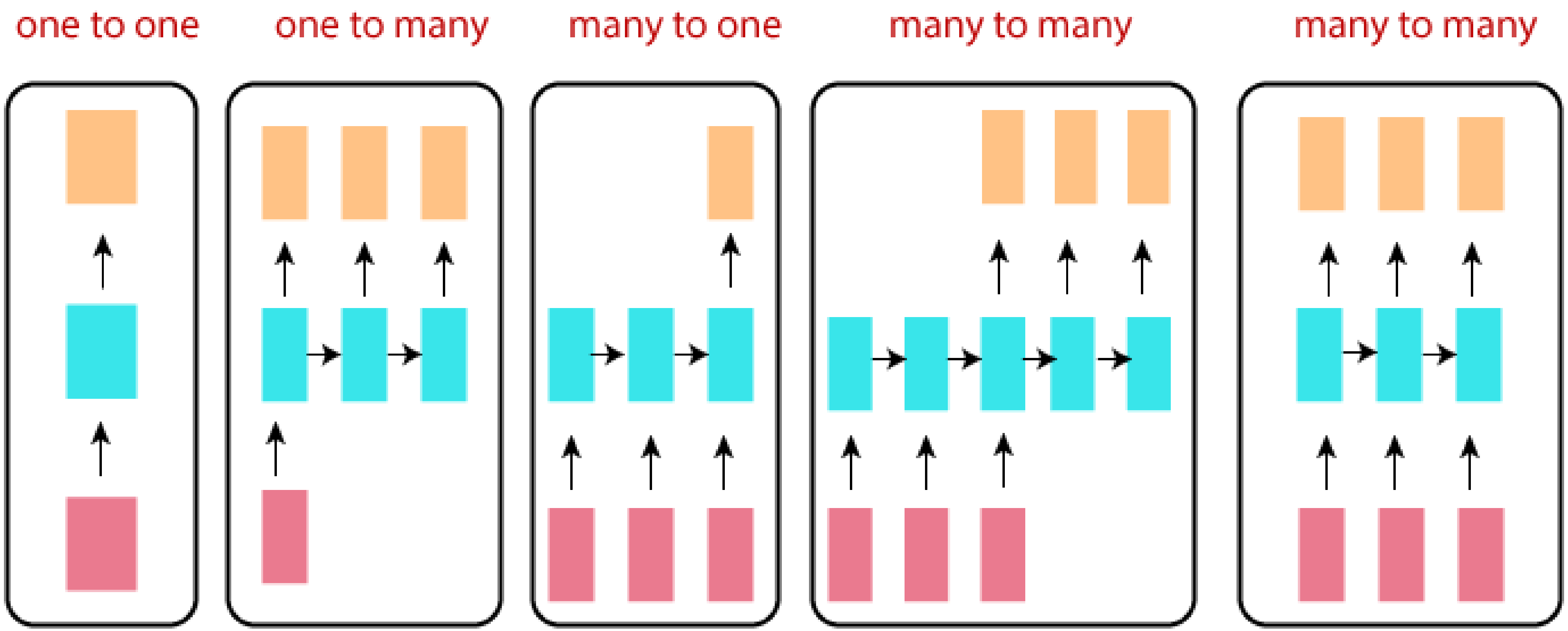
VS

Recurrent Neural Network

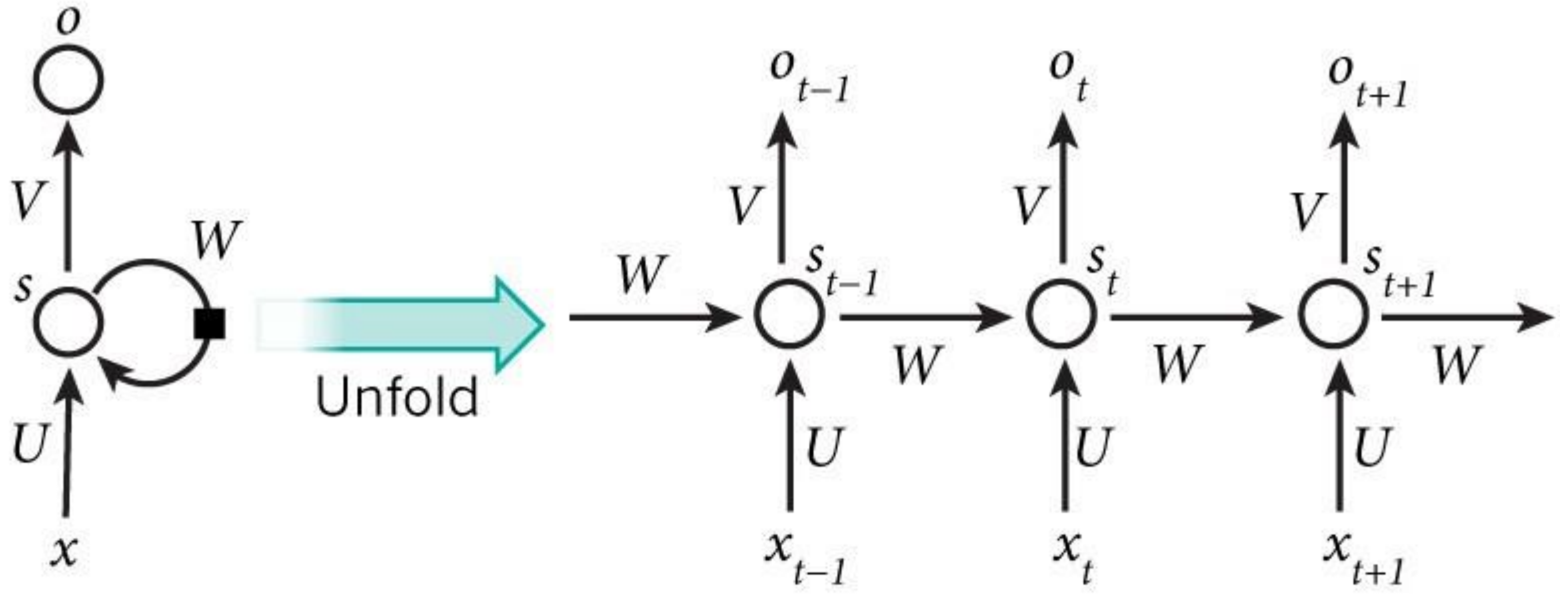



Output
Y
Recurrent
Process
RNN
X
Input


Output
Y
Recurrent
Process
RNN
X
Input
=
Output
y0
y1
y2
y3
yt
H
H
H
H
H
x0
x1
x2
x3
. . . . .
xt
Input


one to one
one to many
many to one
many to many
many to many

Recurrent Neural Networks (RNN)



$$h_t = f(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

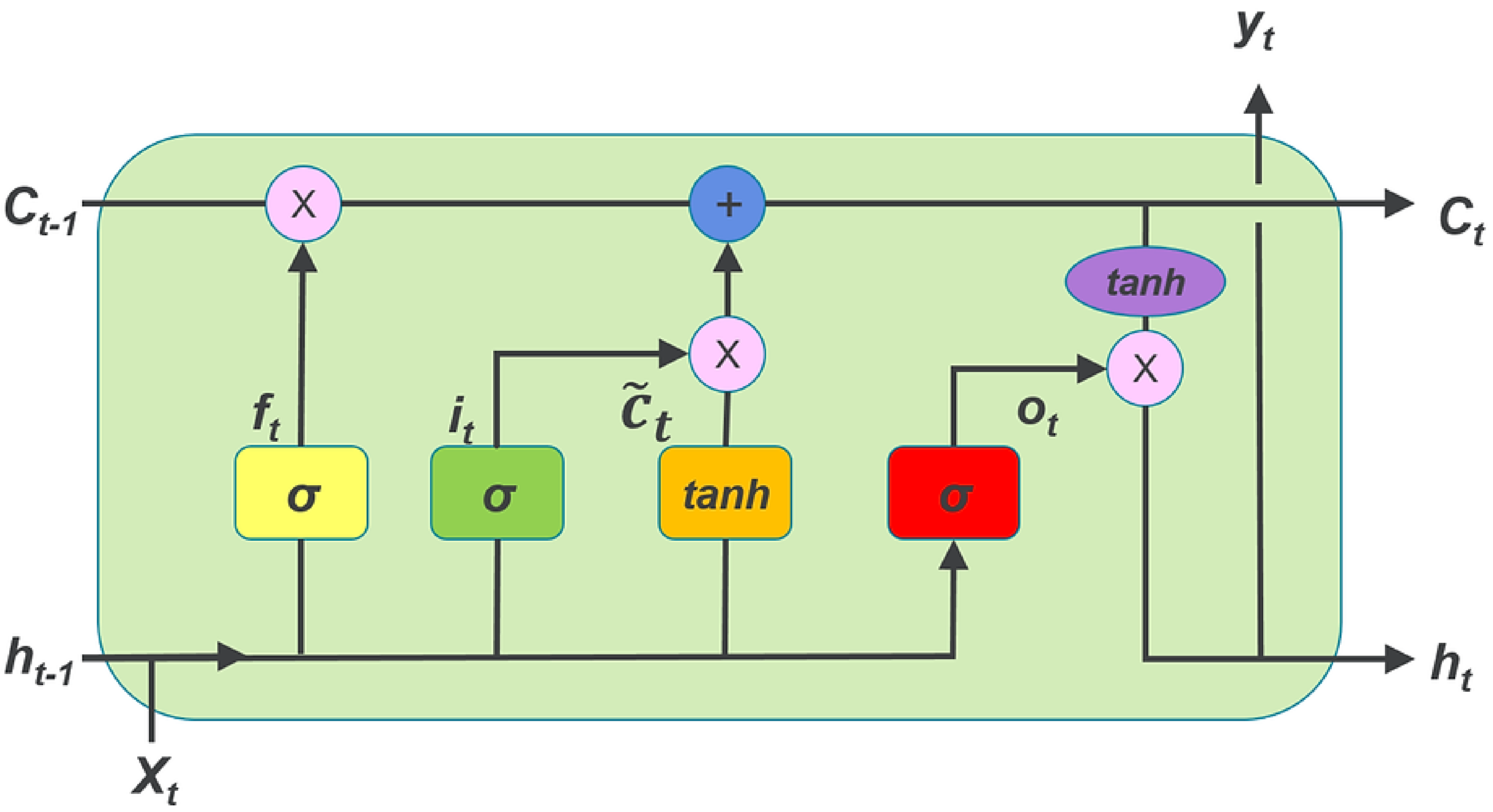
The longest river on earth is _____ .

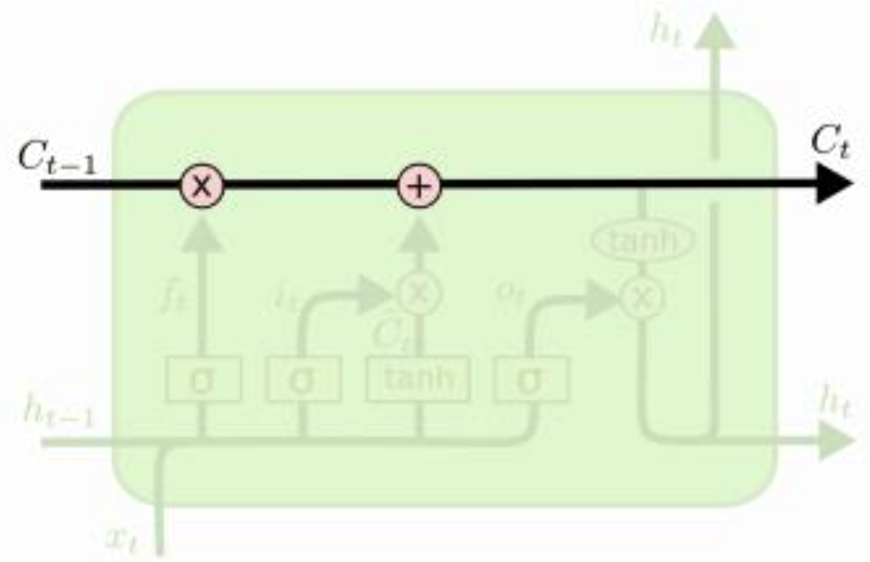
The longest river on earth is Nile.

The man who ate my pizza has purple hair.

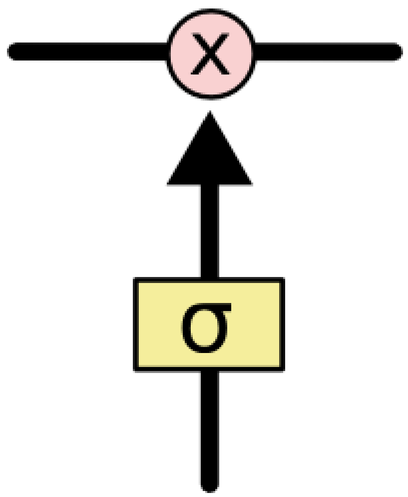
$$\partial E/\partial W = \partial E/\partial y_3 * \partial y_3/\partial h_3 * \partial h_3/\partial y_2 * \partial y_2/\partial h_1 \,..$$

LSTM





h_t
C_{t-1}
C_t
tanh
f_t
i_t
$\tilde{C}_t$
o_t
σ
σ
tanh
σ
h_{t-1}
h_t
x_t


x
σ

$$
\begin{aligned}
i_t &\rightarrow \textit{represents input gate.}\\
f_t &\rightarrow \textit{represents forget gate.}\\
o_t &\rightarrow \textit{represents output gate.}\\
\sigma &\rightarrow \textit{represents sigmoid function.}\\
w_x &\rightarrow \textit{weight for the respective gate}(x)\textit{ neurons.}\\
h_{t-1} &\rightarrow \textit{output of the previous lstm block}(\textit{at timestamp } t-1).\\
x_t &\rightarrow \textit{input at current timestamp.}\\
b_x &\rightarrow \textit{biases for the respective gates}(x).
\end{aligned}
$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma\left(W_o \ [\, h_{t-1}, x_t] \ + \ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$