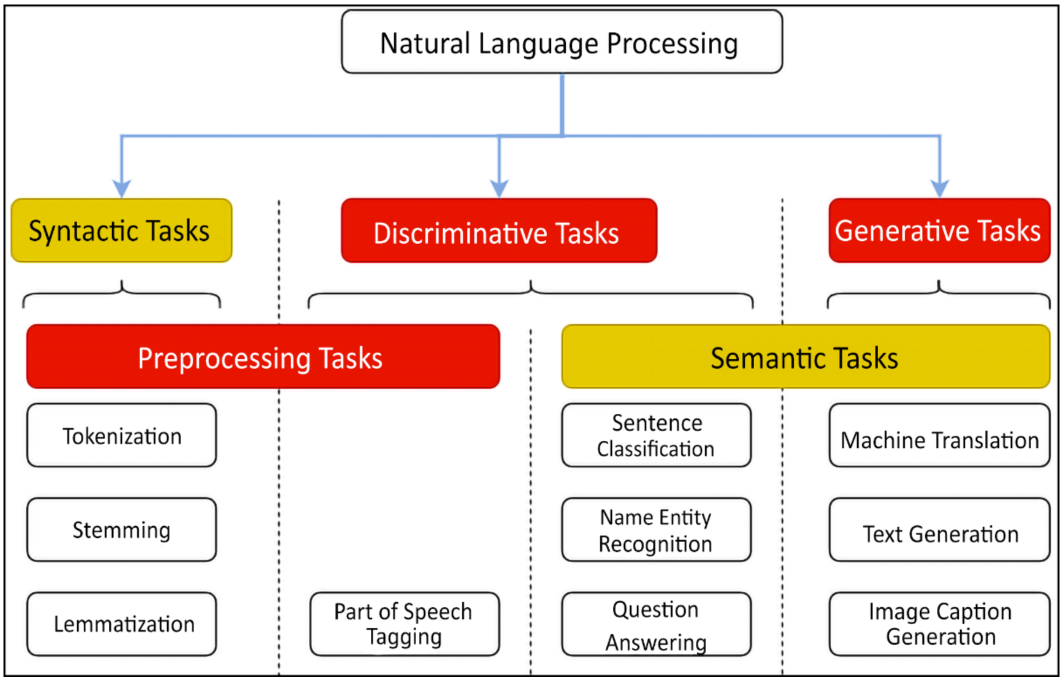
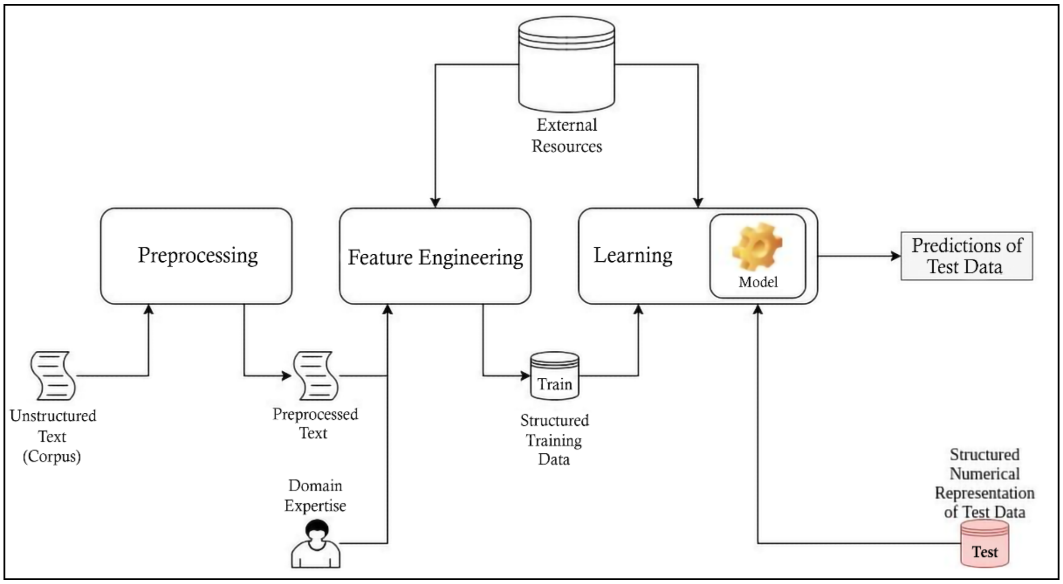
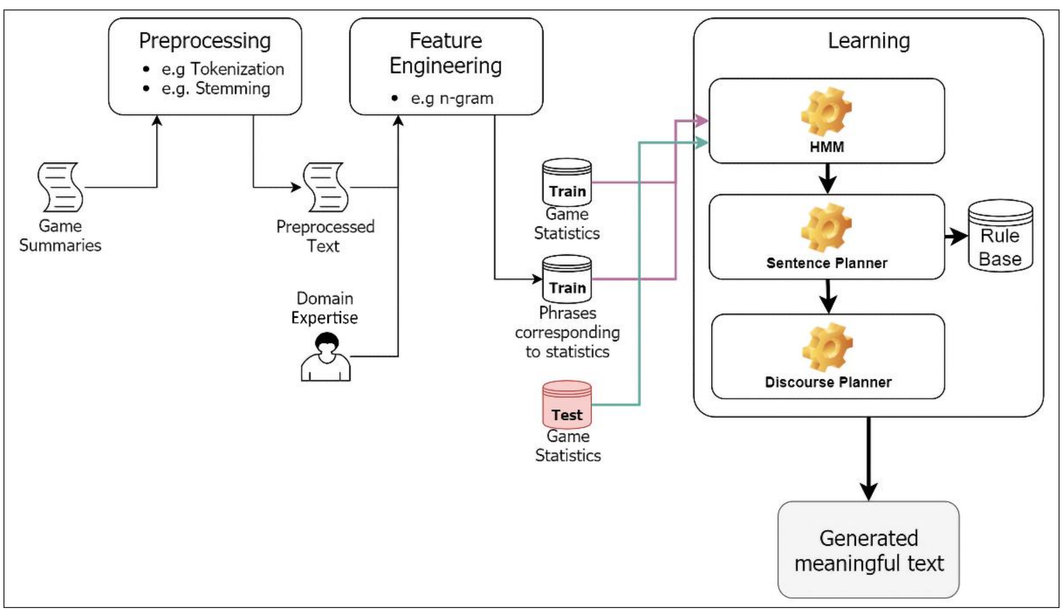
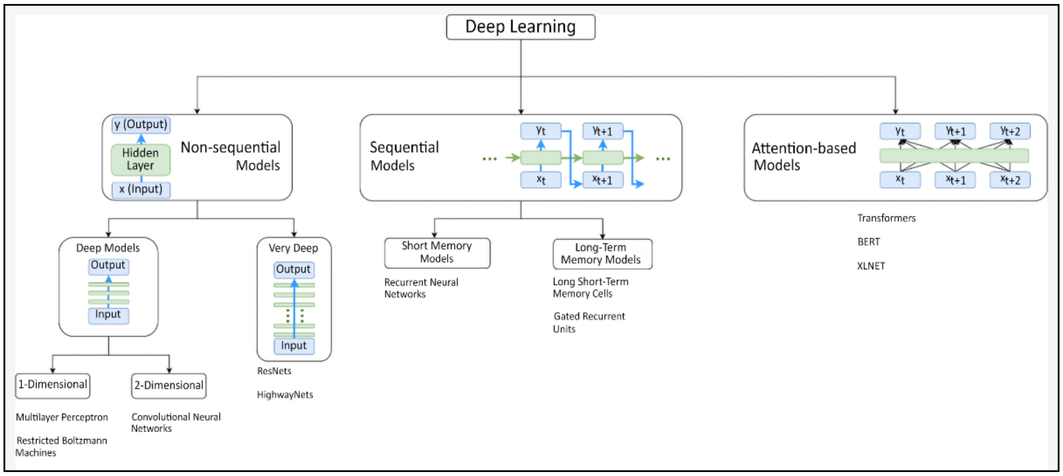
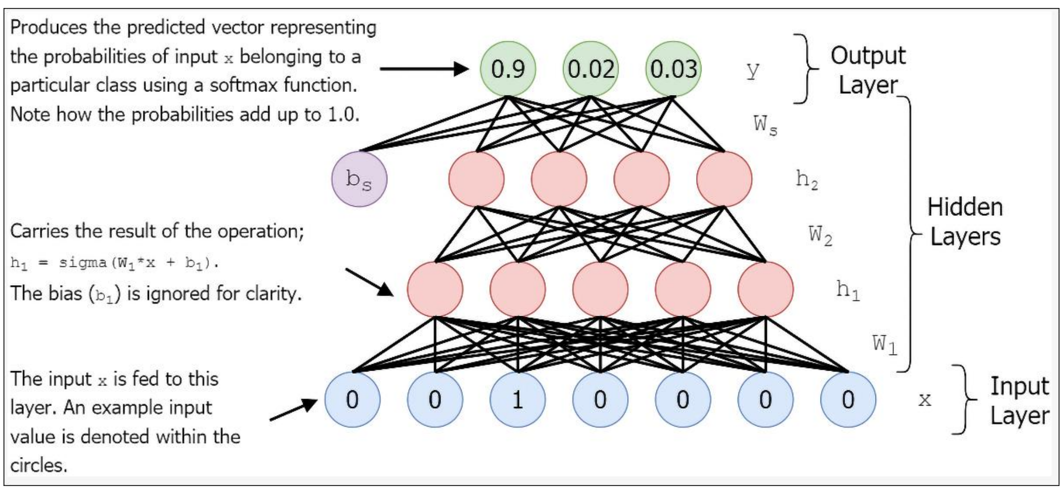
Chapter 1: Introduction to Natural Language Processing




External Resources
Preprocessing
Feature Engineering
Learning
Model
Predictions of Test Data
Unstructured Text (Corpus)
Preprocessed Text
Train
Structured Training Data
Domain Expertise
Structured Numerical Representation of Test Data
Test

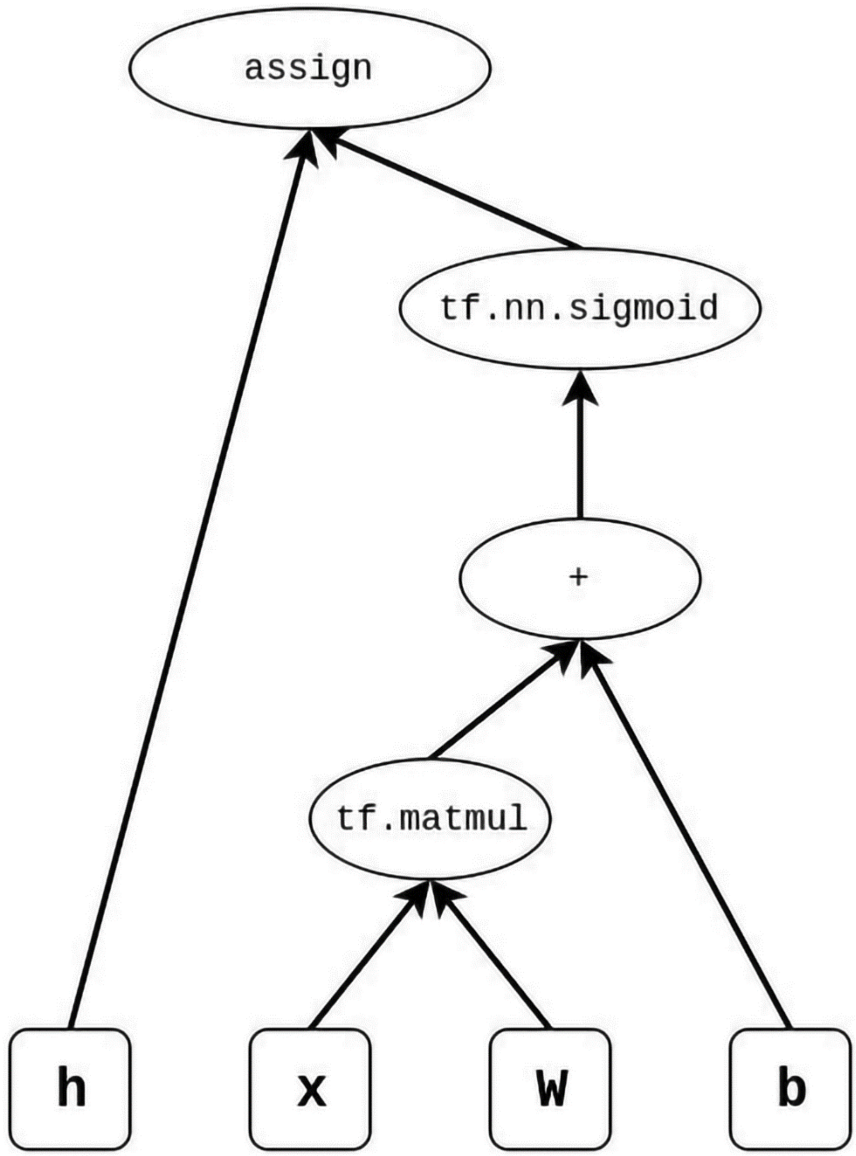

Preprocessing
e.g Tokenization
e.g. Stemming
Feature Engineering
e.g n-gram
Learning
HMM
Sentence Planner
Discourse Planner
Rule Base
Game Summaries
Preprocessed Text
Domain Expertise
Train
Game Statistics
Train
Phrases corresponding to statistics
Test
Game Statistics
Generated meaningful text

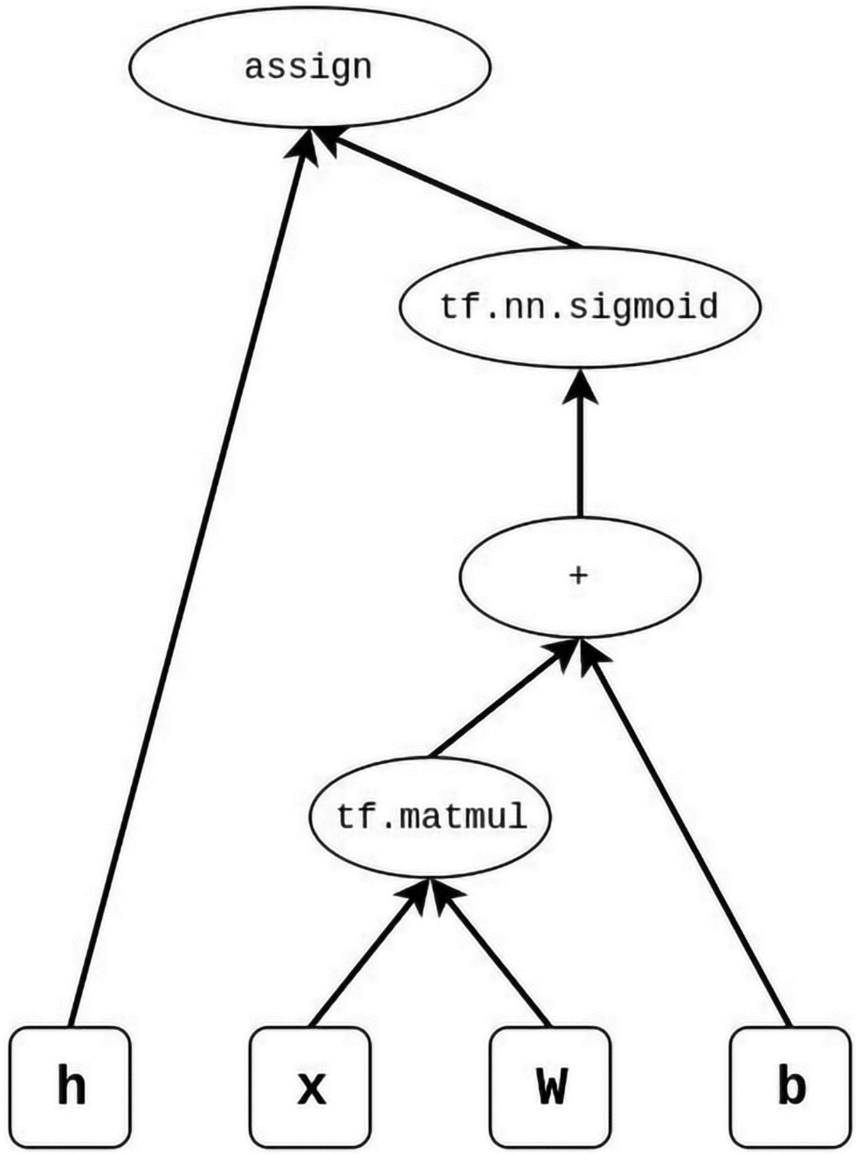

Deep Learning
y (Output)
Hidden Layer
x (Input)
Non-sequential Models
Sequential Models
Attention-based Models
Transformers
BERT
XLNET
Deep Models
Output
Input
Very Deep
Output
Input
Short Memory Models
Long-Term Memory Models
Recurrent Neural Networks
Long Short-Term Memory Cells
Gated Recurrent Units
ResNets
HighwayNets
1-Dimensional
2-Dimensional
Multilayer Perceptron
Restricted Boltzmann Machines
Convolutional Neural Networks

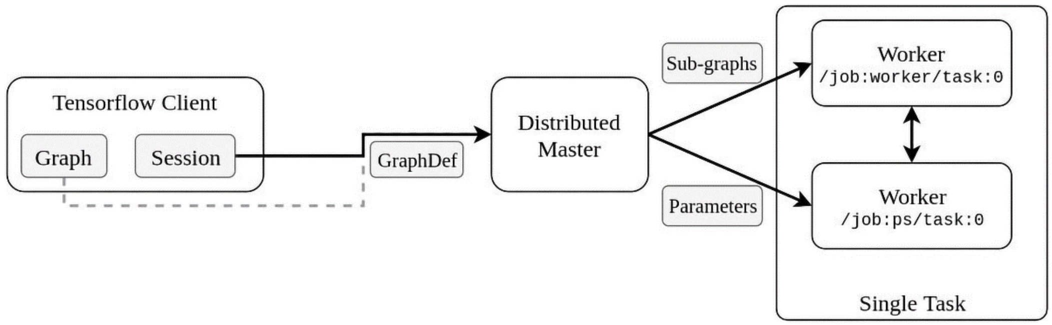

Produces the predicted vector representing the probabilities of input x belonging to a particular class using a softmax function. Note how the probabilities add up to 1.0.
0.9
0.02
0.03
y
Output Layer
Ws
bs
h2
W2
Hidden Layers
Carries the result of the operation;
h1 = sigma(W1*x + b1).
The bias (b1) is ignored for clarity.
h1
W1
The input x is fed to this layer. An example input value is denoted within the circles.
0
0
1
0
0
0
0
x
Input Layer

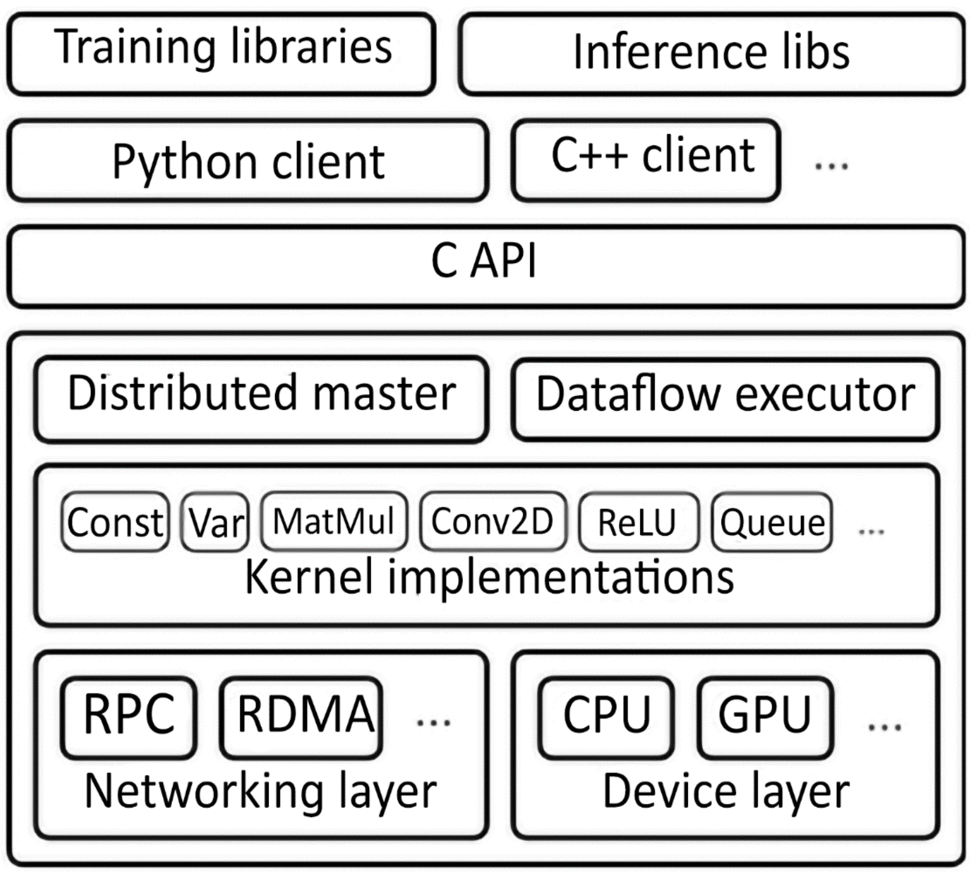

jupyter
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Name
Last Modified

Chapter 2: Understanding TensorFlow 2

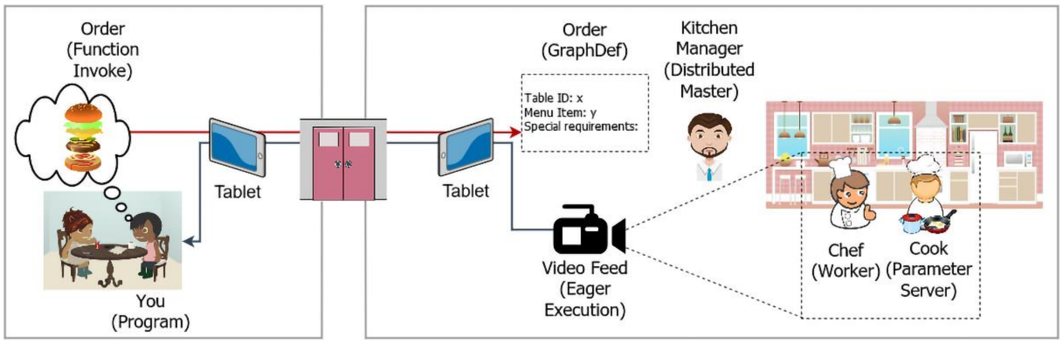


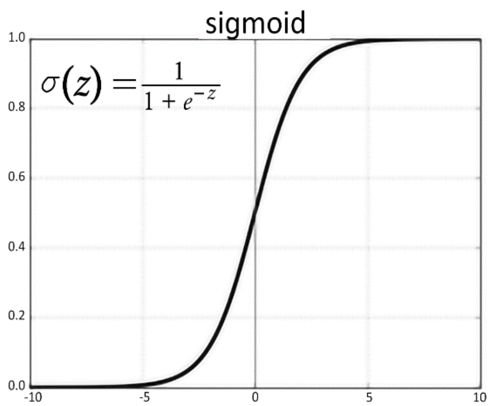

assign
tf.nn.sigmoid
+
tf.matmul
h
x
W
b

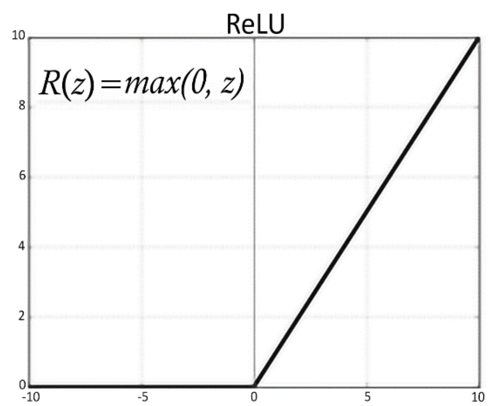

Tensorflow Client
Graph
Session
GraphDef
Distributed
Master
Sub-graphs
Parameters
Worker
/job:worker/task:0
Worker
/job:ps/task:0
Single Task

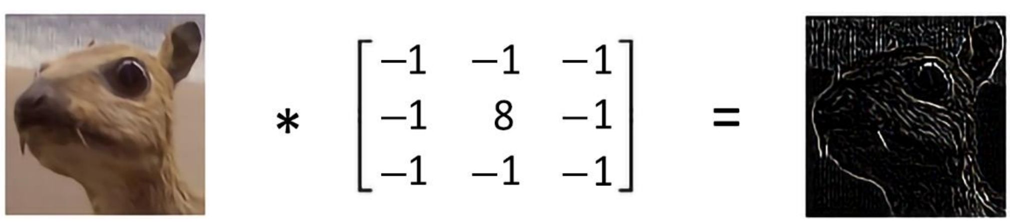

Training libraries
Inference libs
Python client
C++ client
...
C API
Distributed master
Dataflow executor
Const
Var
MatMul
Conv2D
ReLU
Queue
...
Kernel implementations
RPC
RDMA
...
Networking layer
CPU
GPU
...
Device layer

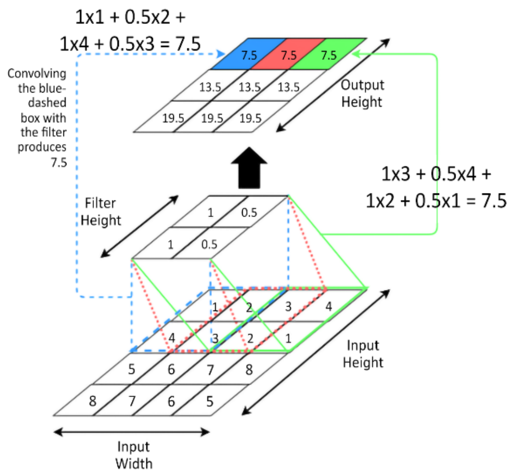
Cafe Le TensorFlow 2

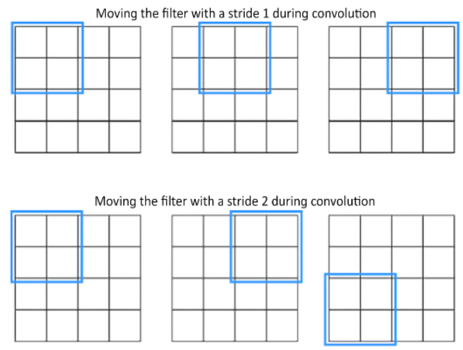


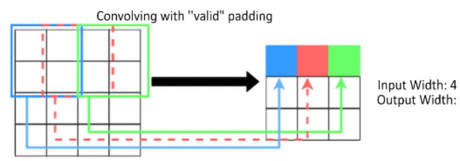


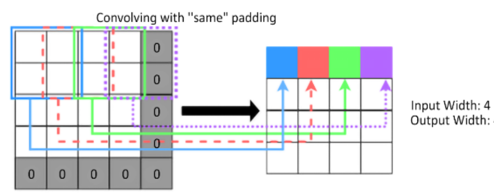


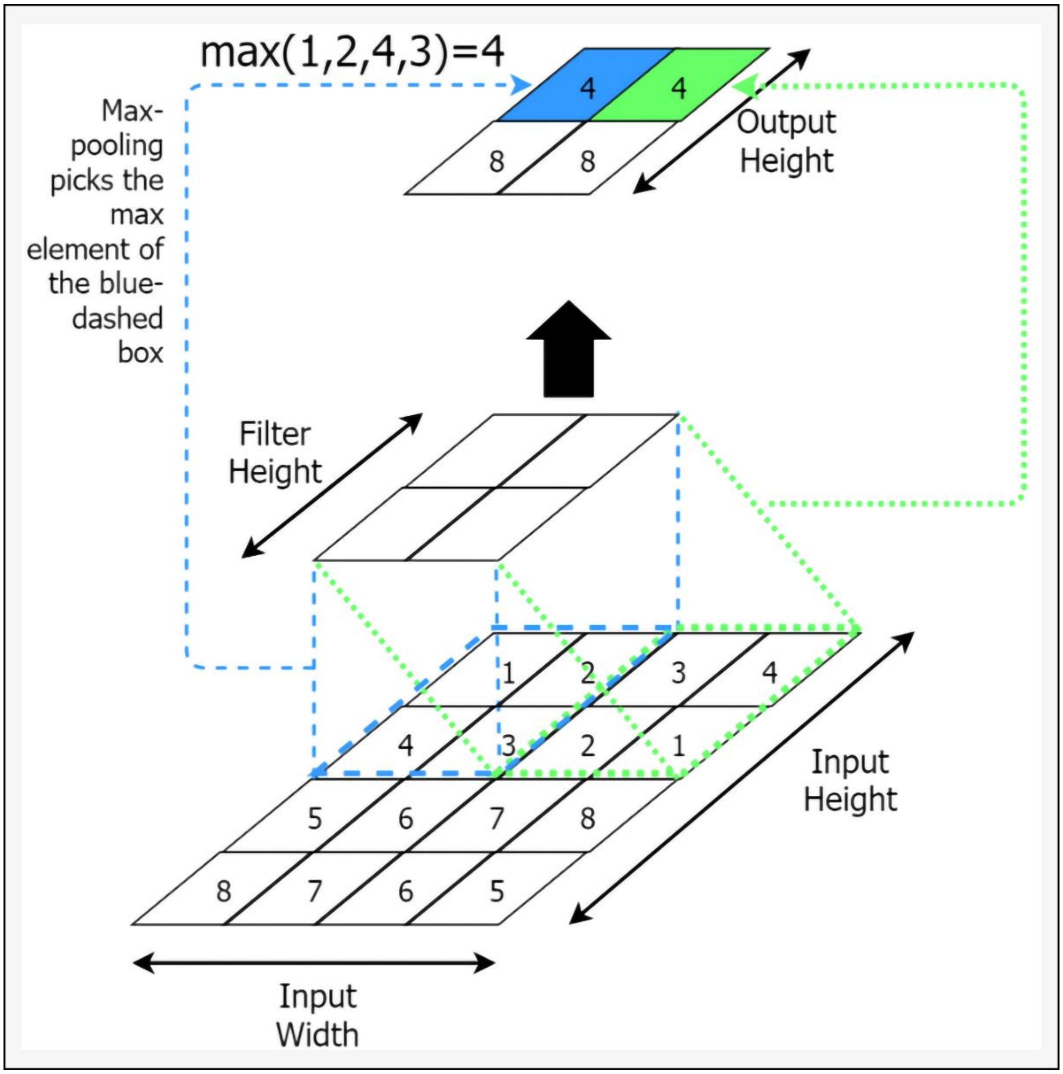


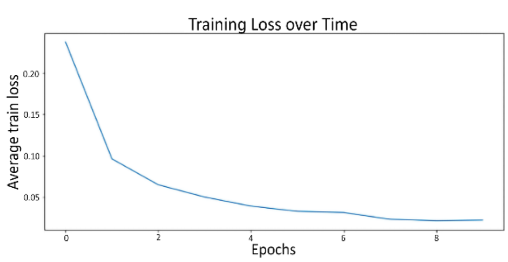

1x1 + 0.5x2 +
1x4 + 0.5x3 = 7.5
Convolving the blue-dashed box with the filter produces 7.5
7.5
7.5
7.5
13.5
13.5
13.5
19.5
19.5
19.5
Output Height
1x3 + 0.5x4 +
1x2 + 0.5x1 = 7.5
Filter Height
1
0.5
1
0.5
1
2
3
4
4
3
2
1
5
6
7
8
8
7
6
5
Input Height
Input Width

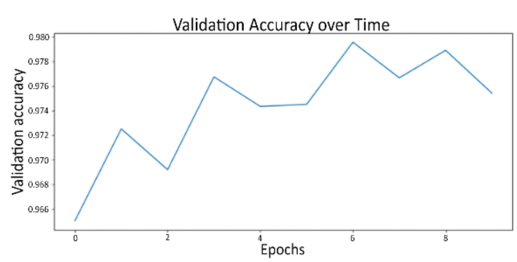

Moving the filter with a stride 1 during convolution
Moving the filter with a stride 2 during convolution

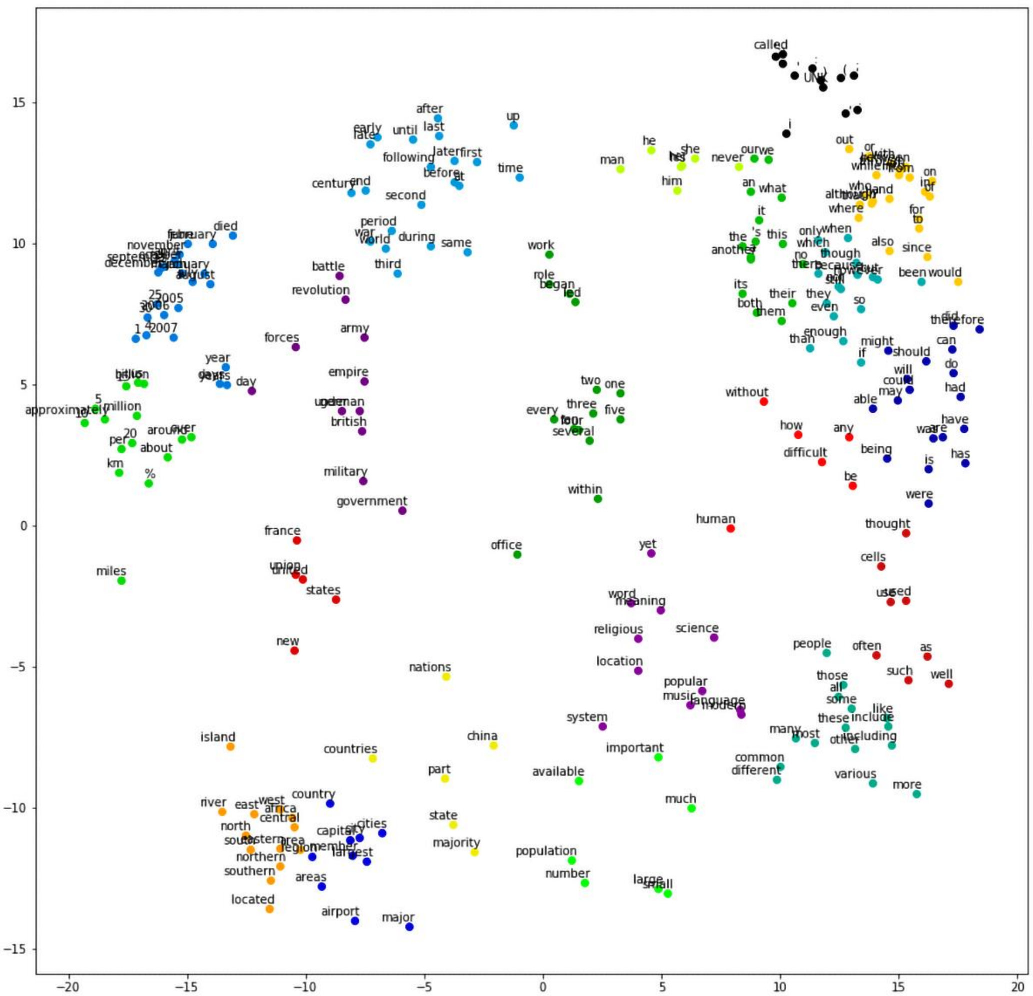

Convolving with "valid" padding
Input Width: 4
Output Width: 3


Convolving with "same" padding
0
0
0
0
0
0
0
0
0
Input Width: 4
Output Width: 4

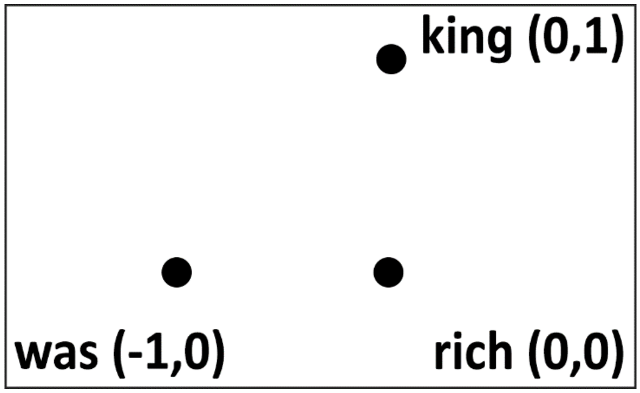

max(1,2,4,3)=4
Max-pooling picks the max element of the blue-dashed box
4
4
8
8
Output Height
Filter Height
1
2
3
4
4
3
2
1
5
6
7
8
8
7
6
5
Input Height
Input Width

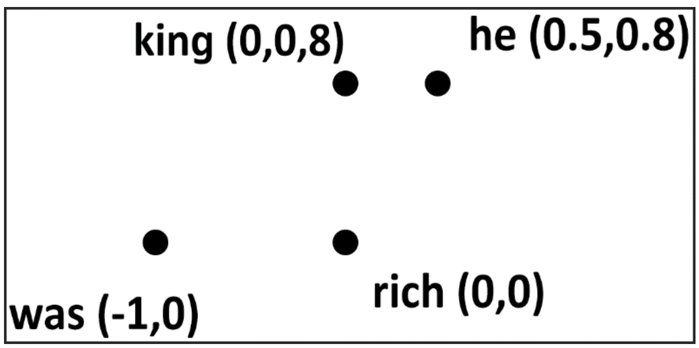

Training Loss over Time
Average train loss
0.20
0.15
0.10
0.05
0
2
4
6
8
Epochs

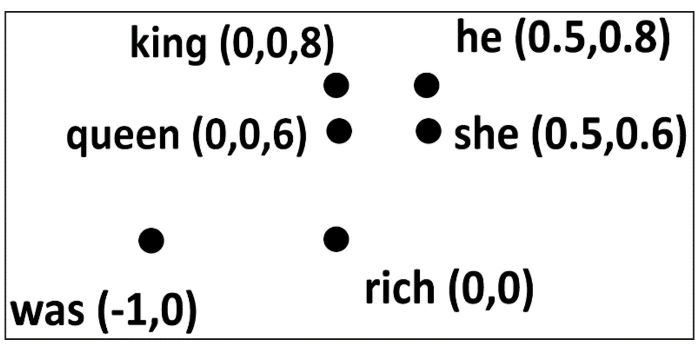

Validation Accuracy over Time
Validation accuracy
0.980
0.978
0.976
0.974
0.972
0.970
0.968
0.966
0
2
4
6
8
Epochs

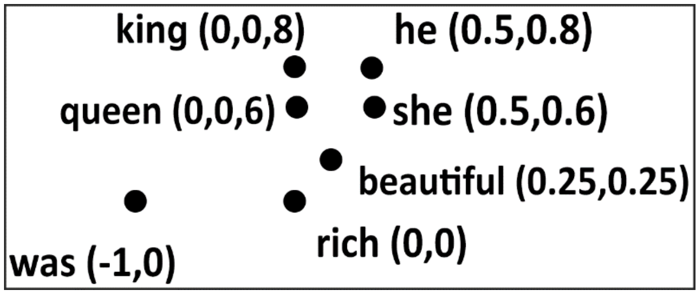


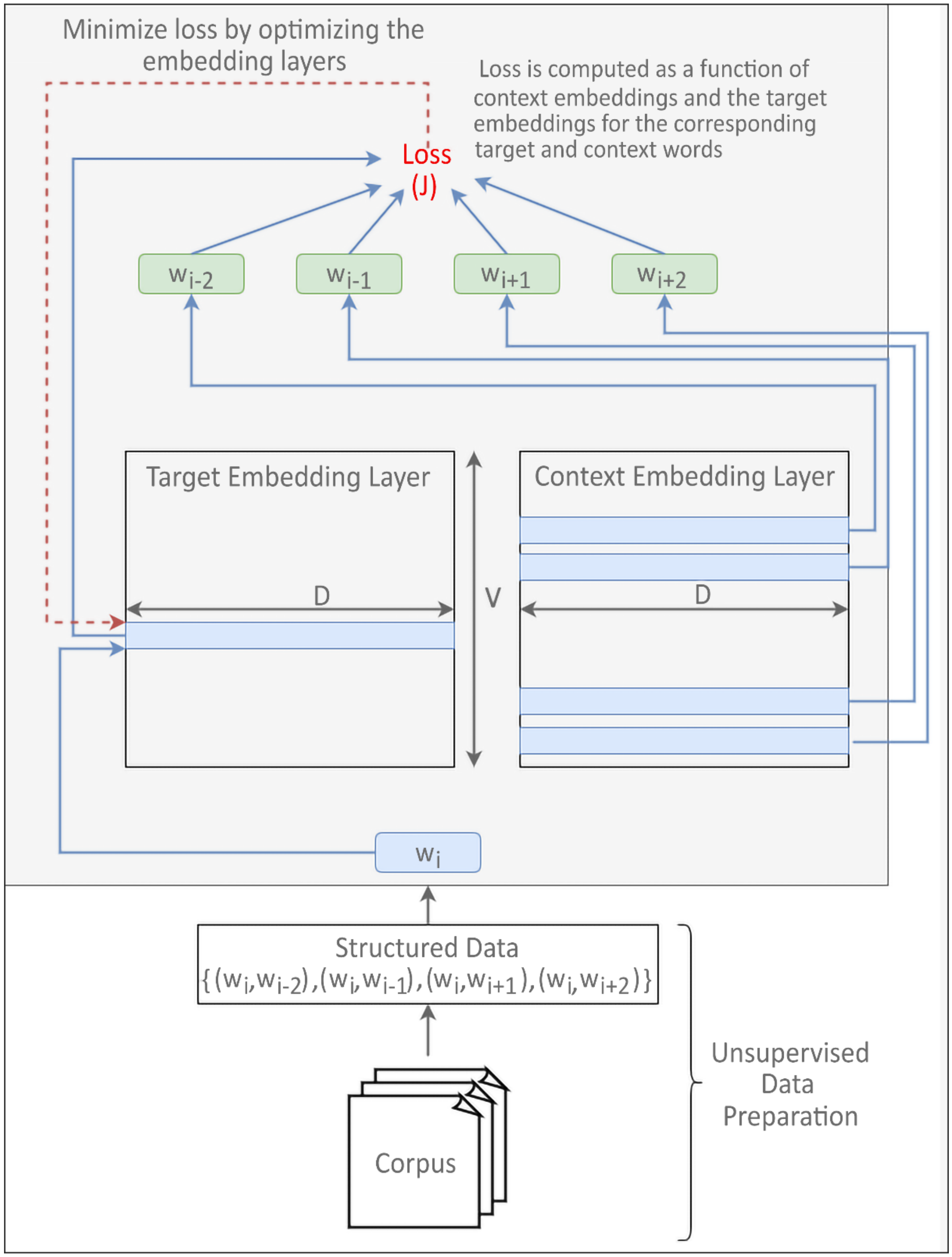
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15
20
called
i
out
he
she
man
never
him
what
it
the
this
another
only
which
though
also
since
for
to
on
where
been
would
its
both
their
them
they
even
so
than
enough
might
should
can
do
had
will
could
may
able
have
are
has
is
being
were
without
how
any
difficult
be
thought
cells
used
human
after
last
early
until
up
first
following
before
time
century
end
second
period
war
world
during
same
third
battle
revolution
forces
army
empire
british
military
government
february
november
december
died
august
25
2005
2007
year
day
million
approximately
20
around
about
per
km
%
miles
work
role
began
two
one
three
five
every
several
within
office
yet
word
meaning
religious
science
location
popular
music
language
system
important
available
much
large
small
population
number
france
united
states
new
nations
china
countries
part
state
majority
island
river
east
west
africa
central
country
north
northern
southern
located
capital
cities
largest
areas
airport
major
people
often
as
such
well
those
all
some
like
these
include
many
most
other
including
common
different
various
more


king (0,1)
was (-1,0)
rich (0,0)

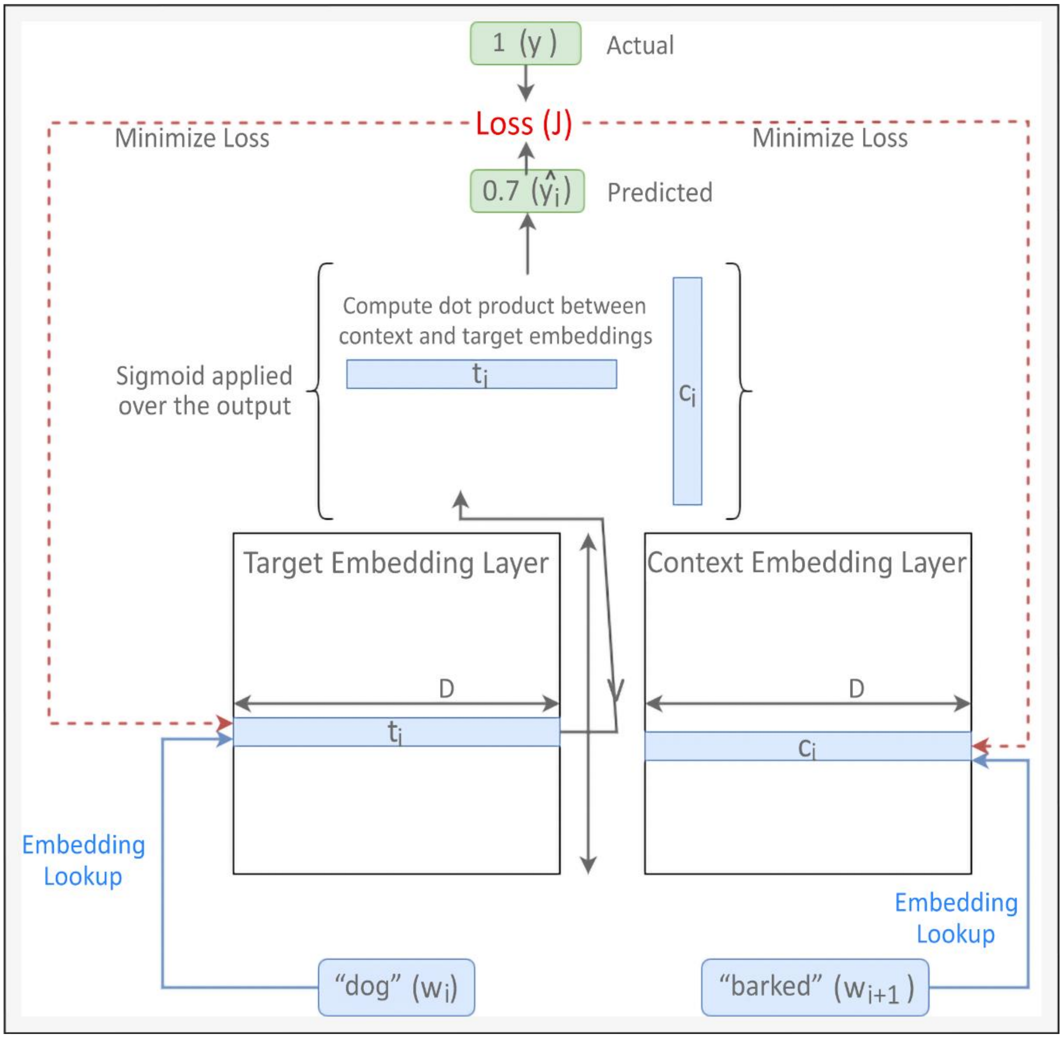

king (0,0,8)
he (0.5,0.8)
was (-1,0)
rich (0,0)

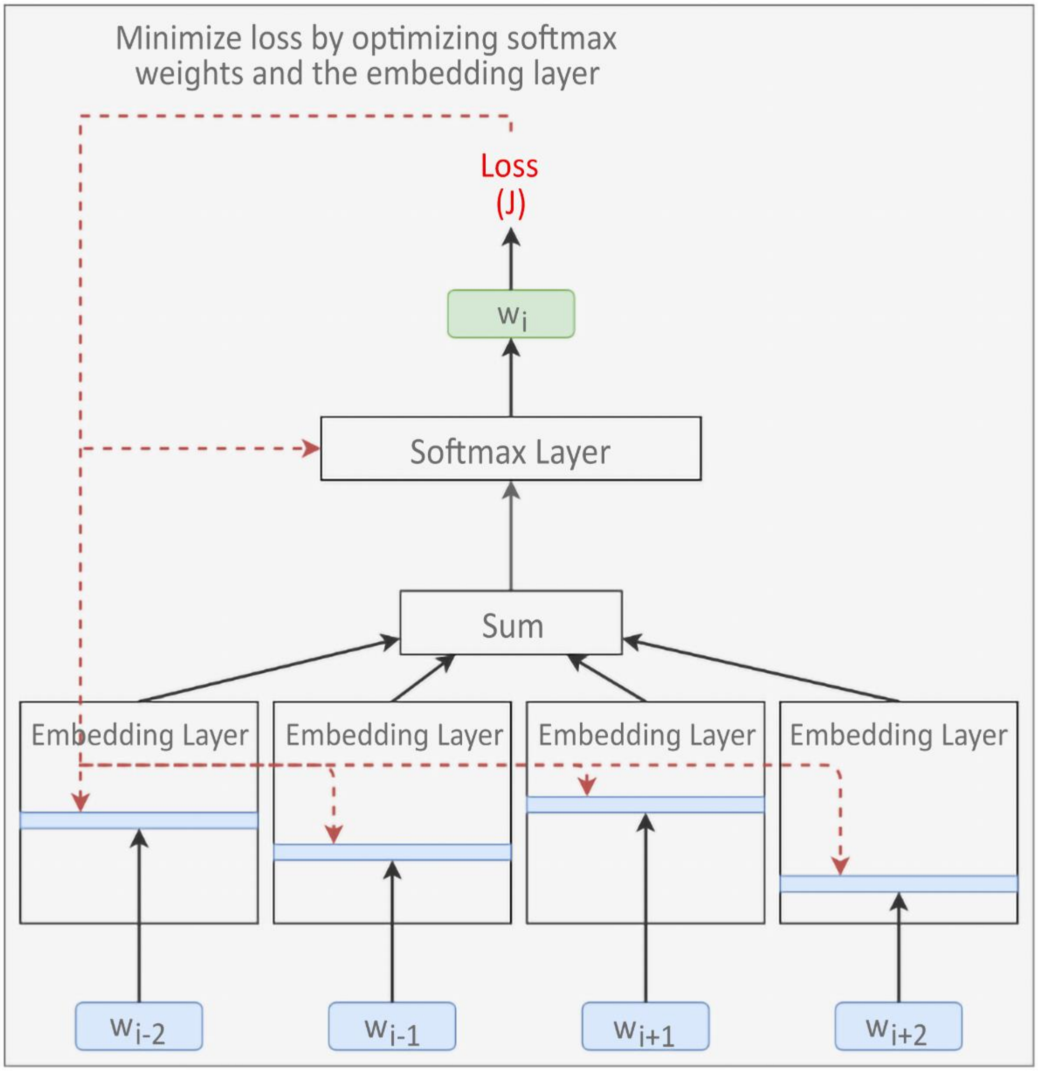

king (0,0,8)
he (0.5,0.8)
queen (0,0,6)
she (0.5,0.6)
was (-1,0)
rich (0,0)

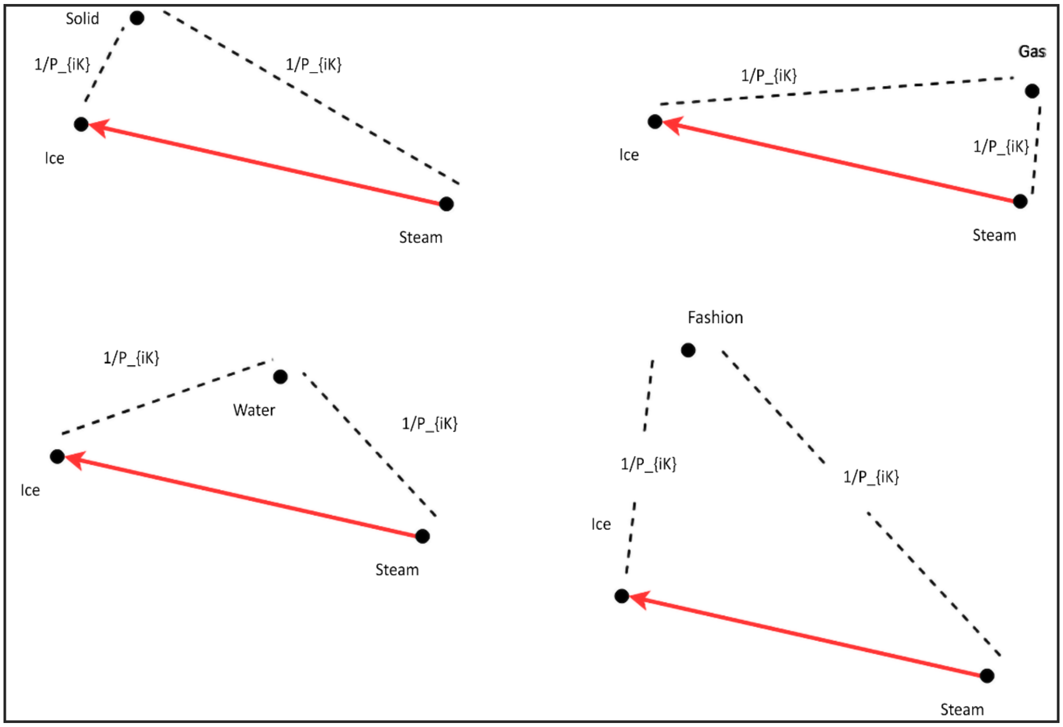

king (0,0,8)
he (0.5,0.8)
queen (0,0,6)
she (0.5,0.6)
beautiful (0.25,0.25)
rich (0,0)
was (-1,0)



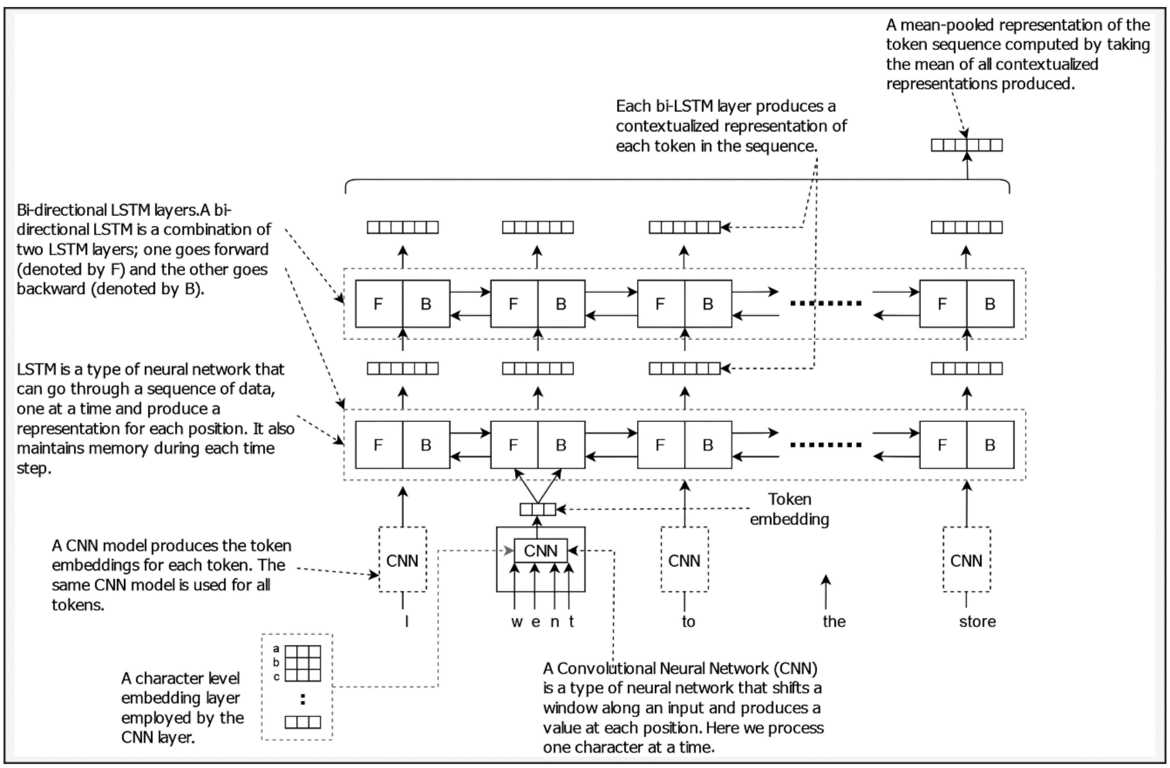
Minimize loss by optimizing the embedding layers
Loss is computed as a function of context embeddings and the target embeddings for the corresponding target and context words
Loss (J)
w_{i-2}
w_{i-1}
w_{i+1}
w_{i+2}
Target Embedding Layer
Context Embedding Layer
D
V
D
w_i
Structured Data
$\{(w_i,w_{i-2}),(w_i,w_{i-1}),(w_i,w_{i+1}),(w_i,w_{i+2})\}$
Unsupervised Data Preparation
Corpus


1 (y)
Actual
Loss (J)
Minimize Loss
Minimize Loss
0.7 ($\hat{y}_i$)
Predicted
Compute dot product between context and target embeddings
Sigmoid applied over the output
t_i
c_j
Target Embedding Layer
Context Embedding Layer
D
D
t_i
c_j
Embedding Lookup
Embedding Lookup
"dog" (w_i)
"barked" (w_{i+1})


Minimize loss by optimizing softmax weights and the embedding layer
Loss (J)
w_i
Softmax Layer
Sum
Embedding Layer
Embedding Layer
Embedding Layer
Embedding Layer
w_{i-2}
w_{i-1}
w_{i+1}
w_{i+2}


Solid
1/P_{iK}
1/P_{iK}
Ice
Steam
Gas
1/P_{iK}
1/P_{iK}
Ice
Steam
1/P_{iK}
Water
1/P_{iK}
Ice
Steam
Fashion
1/P_{iK}
1/P_{iK}
Ice
Steam

$$J = \sum_{i,j=1}^{V} \underbrace{f(X_{ij})}_{\text{Sample Weights}} \Big(\underbrace{w_i^T w_j + b_i + b_j}_{\text{Prediction}} - \underbrace{\log(X_{ij})}_{\text{Target}}\Big)^2$$

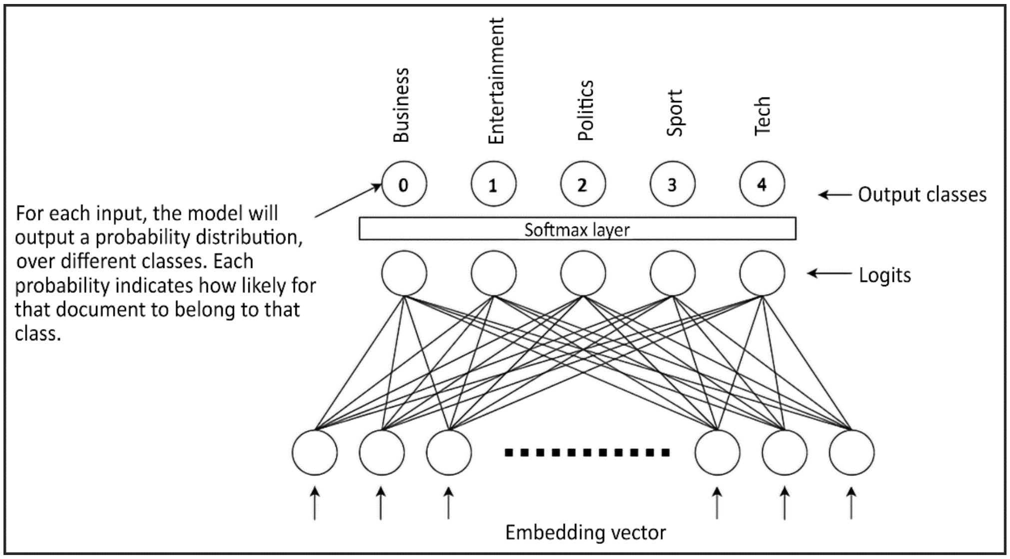


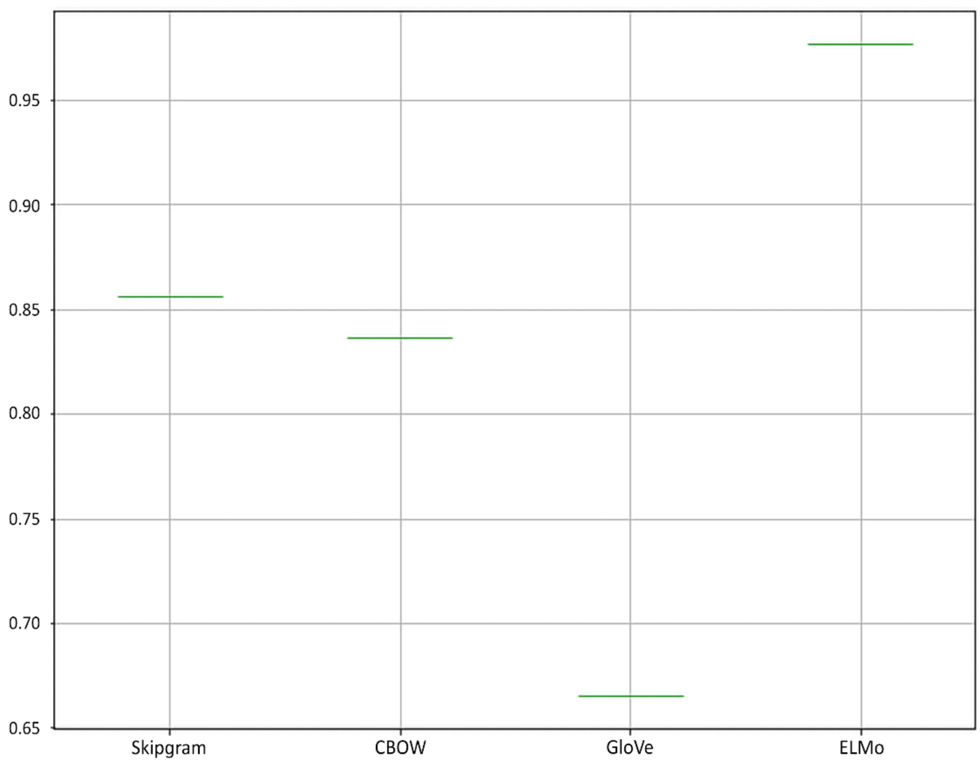
	0	1	2	3	4	5	6	7	8	9	...	118	119	120	
NaN	-1.440415	1.375493	-1.205786	-1.326926	-1.336768	-1.205858	-1.366250	-1.036717	-1.365316	-0.775372	...	1.205486	1.138118	-1.385370	-1.264
	-0.290653	0.574326	0.026581	-0.329045	-0.171542	0.224572	0.071370	0.146927	-0.041791	0.086741	...	0.259992	0.278289	-0.879627	-0.351
the	-0.058531	0.310743	0.232947	-0.338871	-0.334953	0.003350	-0.061267	0.257632	0.031830	-0.587573	...	0.484254	-0.228630	-0.044630	-0.305
to	-0.276640	0.025122	-0.024782	-0.624656	-0.016077	0.046590	-1.094414	-0.848296	-0.327094	-1.684406	...	-0.261233	-0.106222	-0.139904	-0.411
of	-0.281321	-0.034827	-0.068643	-0.210236	0.122180	0.021023	-0.258712	0.238127	-0.408491	-1.221611	...	0.083432	0.030410	-0.390671	-0.030
and	-0.207726	0.023091	0.186237	-0.209033	-0.121335	0.271208	-0.033538	0.390482	0.217749	-1.416989	...	-0.351555	0.300373	-0.019044	-0.298
a	-0.452312	0.305802	-0.962776	-0.884619	0.180468	-0.150204	-0.387859	0.362963	-0.194890	0.303527	...	-0.297423	0.582571	-0.498461	-0.074
in	0.264359	0.130696	0.374273	0.023650	-0.385127	0.085398	0.314853	-1.038579	-0.981065	0.308113	...	-0.258953	0.010786	-0.503573	0.038
for	-0.866129	0.162434	-0.183391	0.079462	-0.439157	-0.600472	-0.600996	-1.004945	-0.321327	-1.096564	...	0.153067	0.338188	-0.180635	0.506
is	-0.004946	0.531260	-0.119664	-0.702005	-0.368837	-0.918419	0.164347	-0.694515	-0.470691	0.242414	...	0.780400	-0.017688	-0.529367	0.020

10 rows × 128 columns

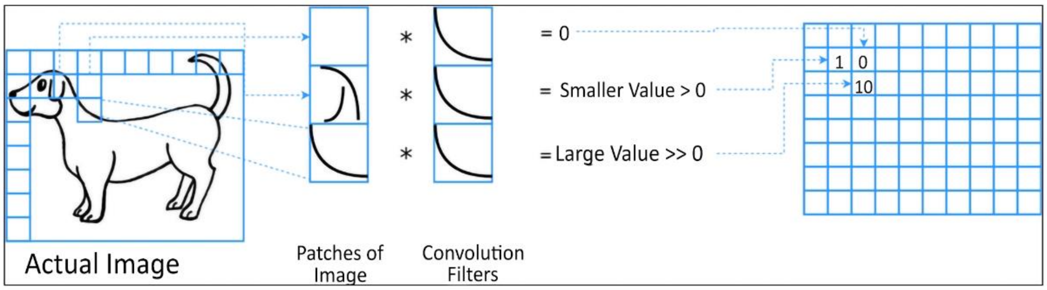
	0	1	2	3	4	5	6	7	8	9	...	1014	1015	
data/bbc/tech/272.txt	0.144291	0.015962	-0.151633	0.096555	-0.015913	0.075896	-0.033353	-0.118644	-0.192030	-0.124919	...	-0.418126	0.204618	0.
data/bbc/tech/127.txt	0.022870	-0.142899	-0.017096	-0.084165	0.320108	0.424914	-0.043930	0.257134	-0.215543	-0.046845	...	-0.219130	0.264653	0.
data/bbc/tech/370.txt	0.207623	0.058697	-0.008874	-0.088409	0.193419	0.046109	-0.107221	0.199647	-0.167632	0.003790	...	-0.054829	0.225892	0.
data/bbc/tech/329.txt	0.022106	0.060943	-0.127390	-0.100214	0.184243	-0.077529	-0.157470	-0.042993	-0.204254	-0.021419	...	-0.337353	0.153419	0.
data/bbc/tech/240.txt	0.259128	-0.108082	0.076262	-0.080416	0.183988	0.329807	0.156697	0.495652	-0.104913	-0.120077	...	-0.218093	0.236378	0.
data/bbc/tech/379.txt	0.071111	-0.112660	0.038746	-0.084503	0.207438	0.231360	-0.015819	0.235174	-0.238940	0.030840	...	-0.330608	0.267756	0.
data/bbc/tech/339.txt	0.080515	0.216040	-0.090722	0.118778	0.336153	0.223334	0.075926	0.472938	-0.046222	0.124439	...	-0.240382	0.015074	-0.
data/bbc/tech/046.txt	0.019151	-0.029814	-0.046270	-0.139506	0.255929	0.230742	-0.011093	0.426066	0.112122	-0.130358	...	-0.285940	0.280919	-0.
data/bbc/tech/140.txt	0.379357	-0.174718	-0.062910	-0.017047	-0.034842	0.260822	0.129207	0.358961	-0.040298	-0.027103	...	-0.144901	0.096271	0.
data/bbc/tech/349.txt	0.241237	-0.013553	0.033077	-0.159186	0.367290	0.414783	0.176808	0.556002	-0.273304	0.108183	...	-0.235471	0.265600	-0.

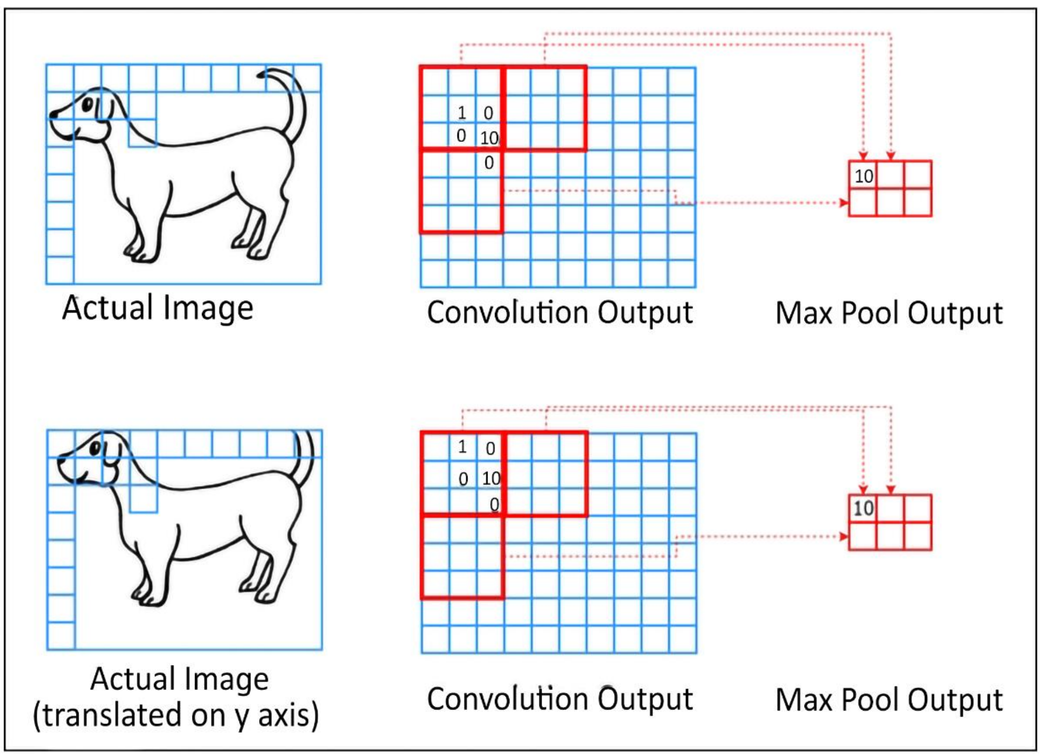
10 rows × 1024 columns

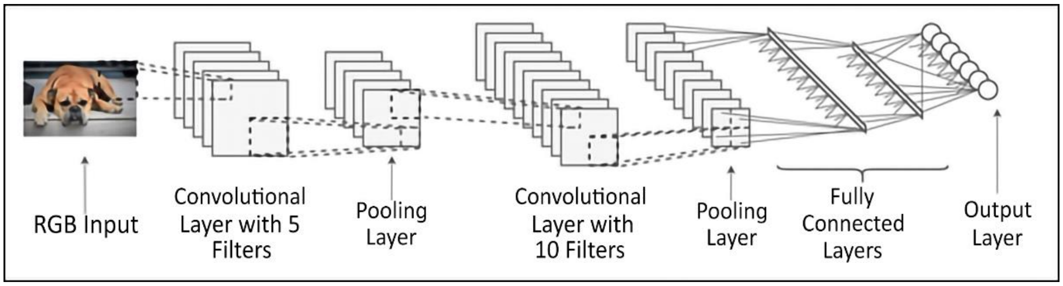

Business
Entertainment
Politics
Sport
Tech
0
1
2
3
4
Output classes
For each input, the model will output a probability distribution, over different classes. Each probability indicates how likely for that document to belong to that class.
Softmax layer
Logits
Embedding vector

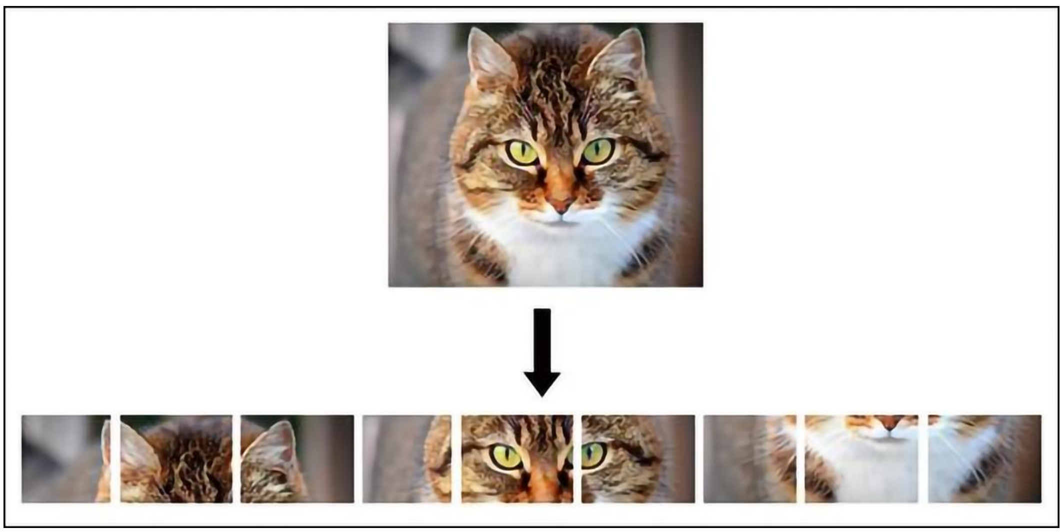

0.95
0.90
0.85
0.80
0.75
0.70
0.65
Skipgram
CBOW
GloVe
ELMo

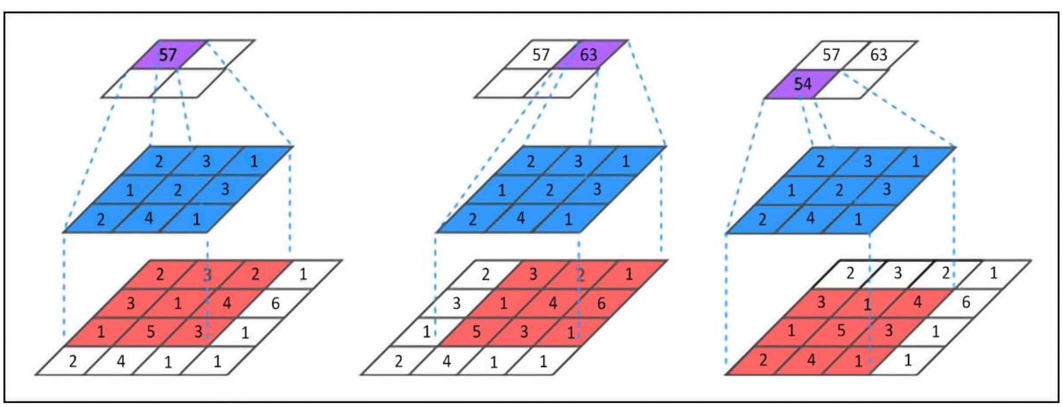
Chapter 5: Sentence Classification with Convolutional Neural Networks

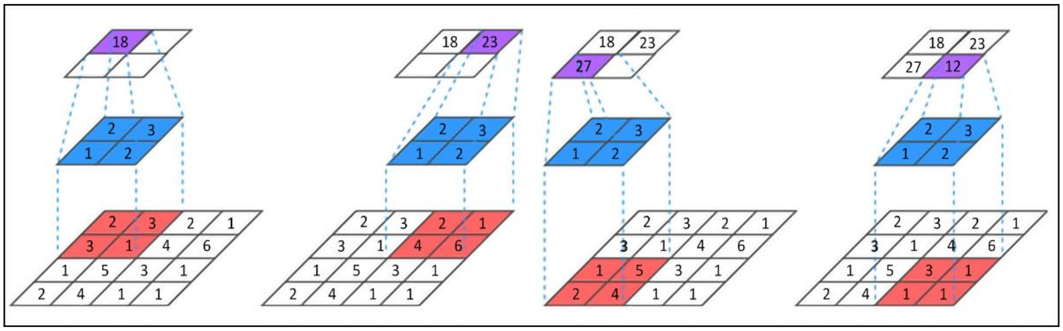


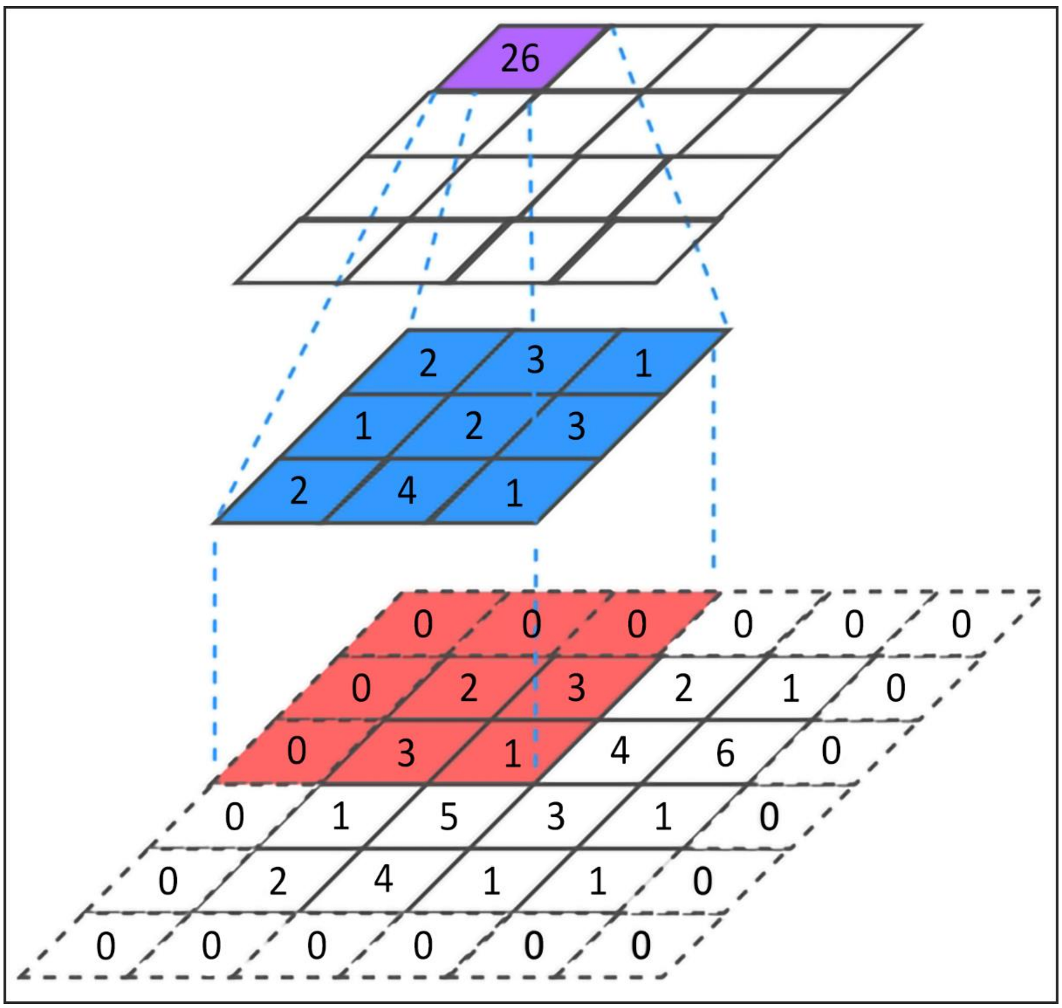


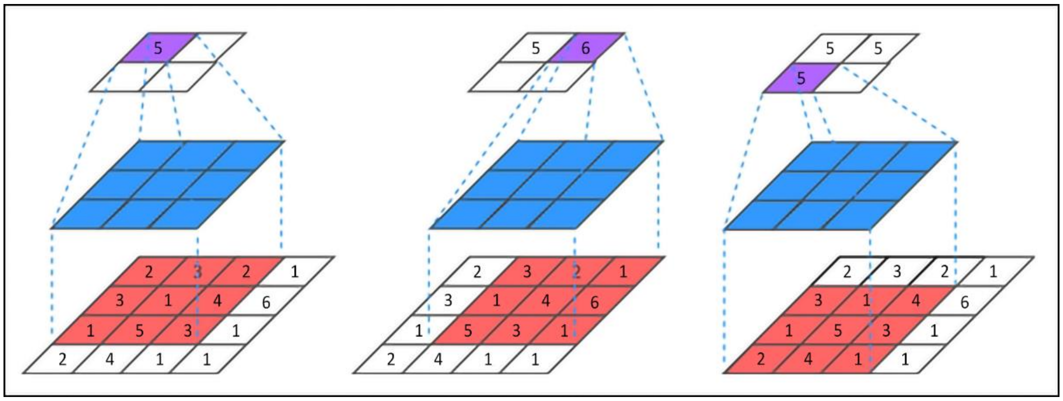

RGB Input
Convolutional Layer with 5 Filters
Pooling Layer
Convolutional Layer with 10 Filters
Pooling Layer
Fully Connected Layers
Output Layer

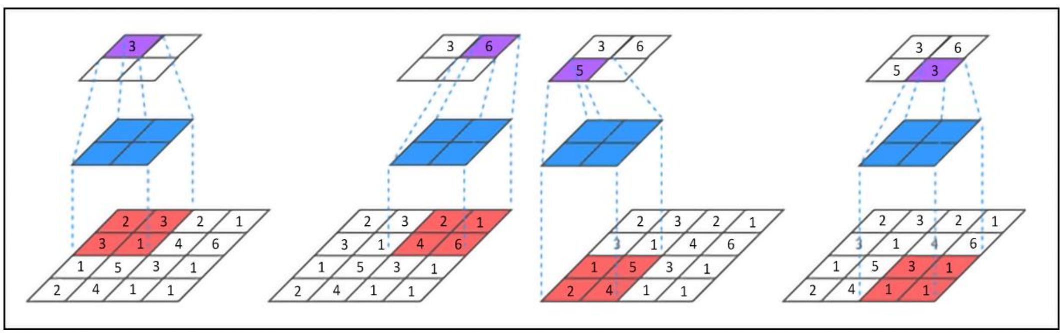


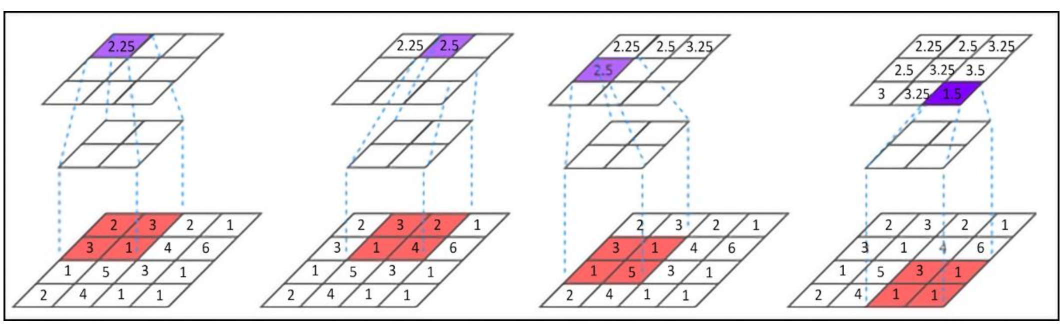

57
2 3 1
1 2 3
2 4 1
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1
57 63
2 3 1
1 2 3
2 4 1
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1
57 63
54
2 3 1
1 2 3
2 4 1
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1

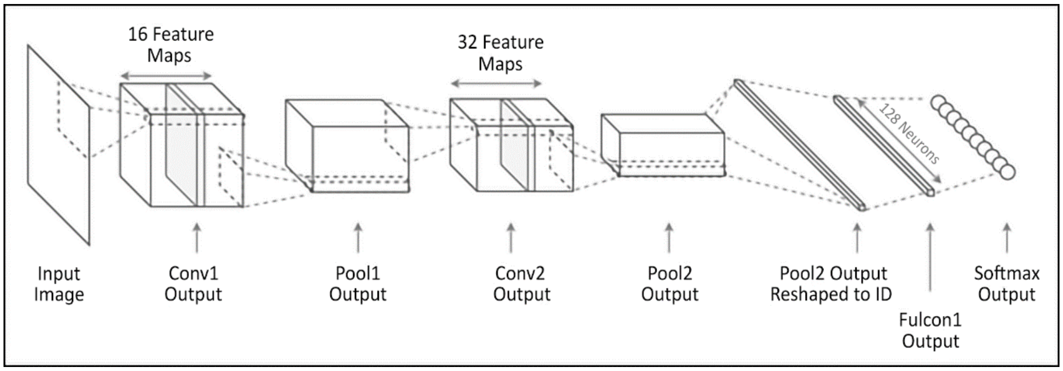

18
2 3
1 2
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1
18 23
2 3
1 2
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1
18 23
27
2 3
1 2
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1
18 23
27 12
2 3
1 2
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1

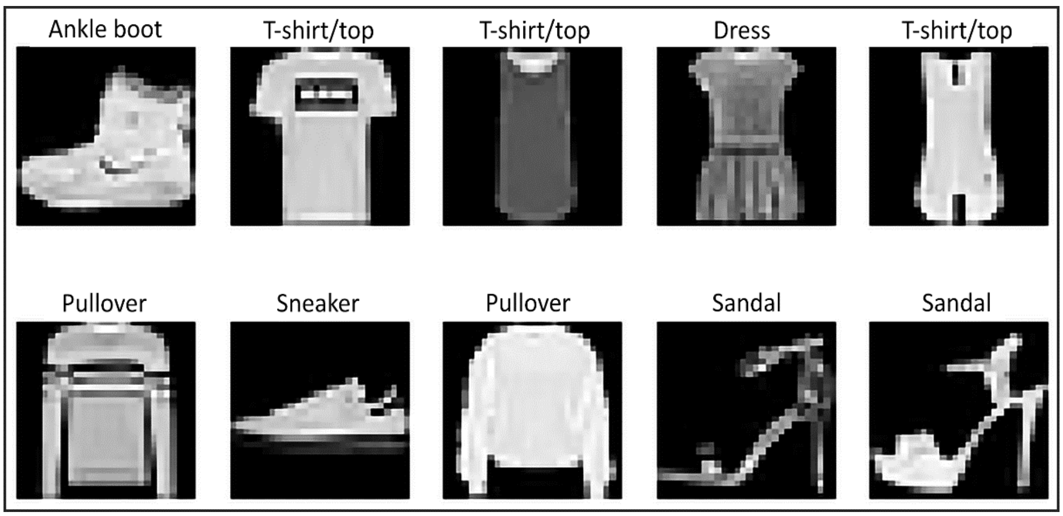

26
2 3 1
1 2 3
2 4 1
0 0 0 0 0 0
0 2 3 2 1 0
0 3 1 4 6 0
0 1 5 3 1 0
0 2 4 1 1 0
0 0 0 0 0 0

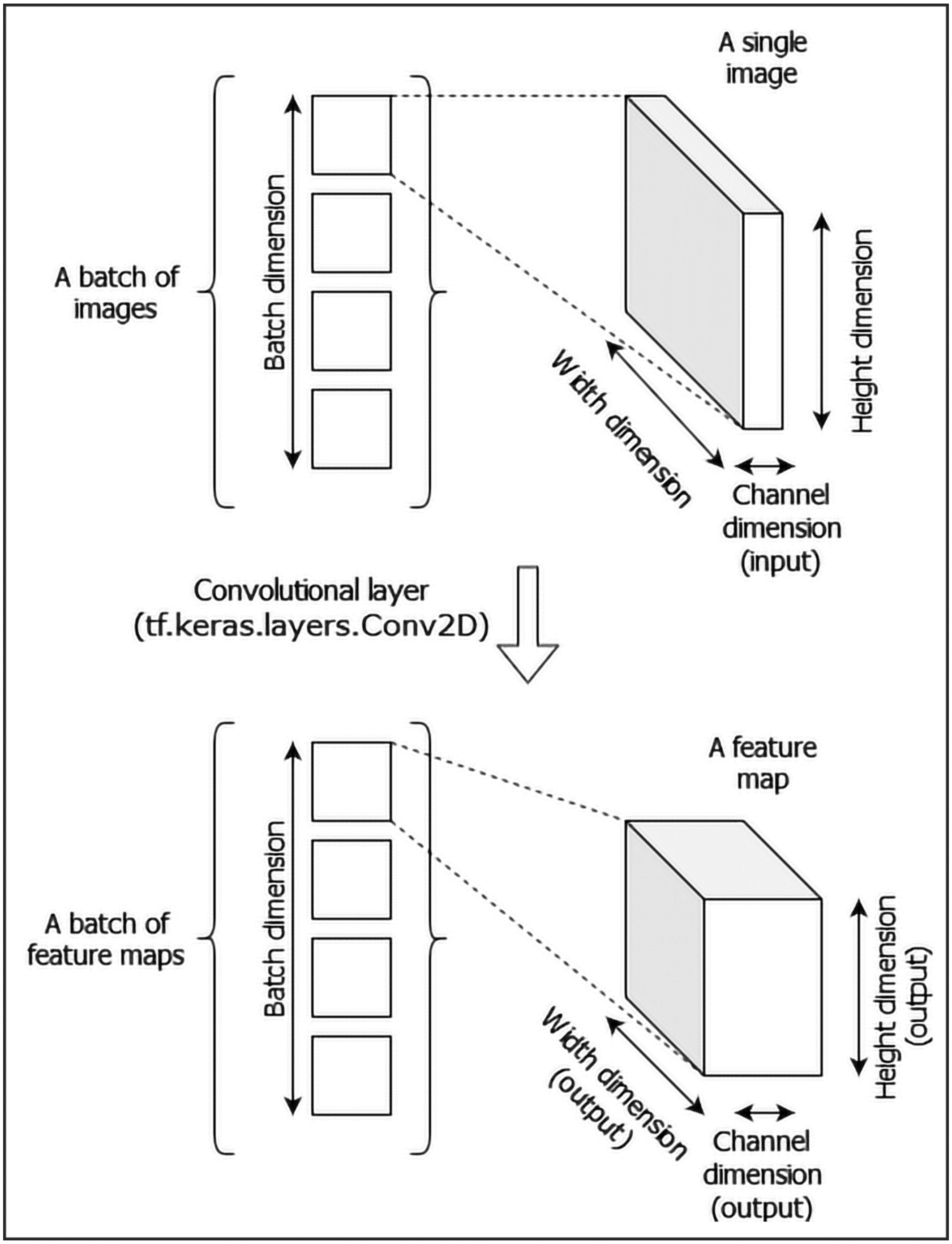

5
5 6
5 5
5
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1

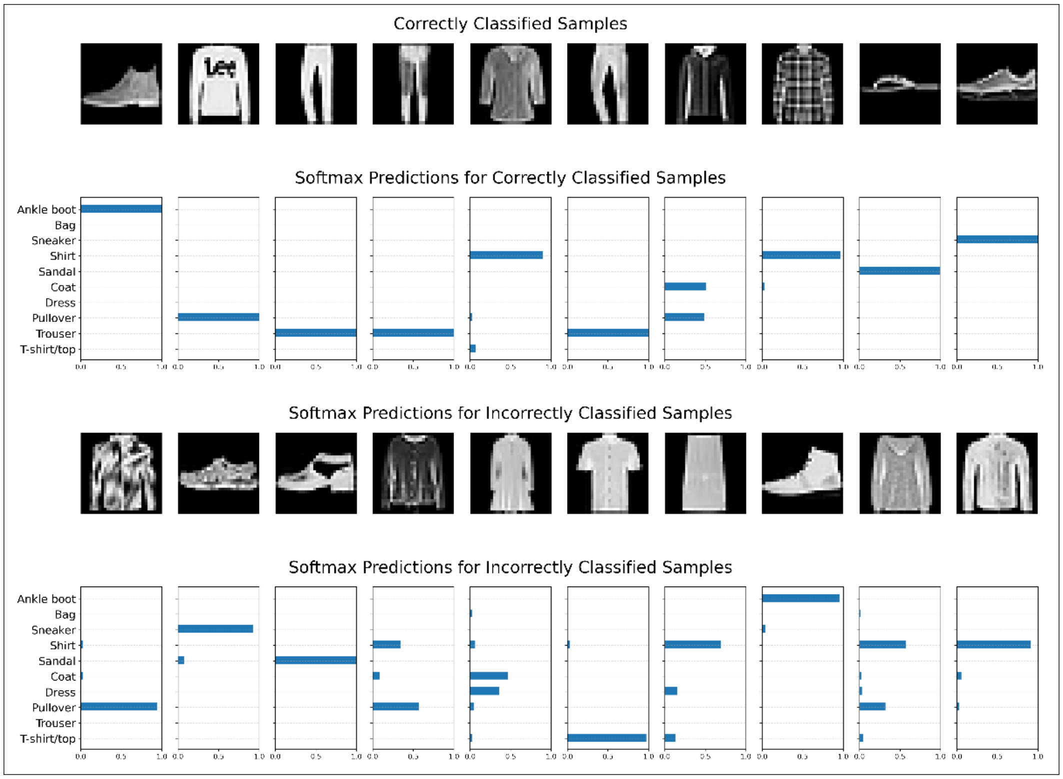

3
3 6
3 6
5
3 6
5 3
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1


2.25
2.25 2.5
2.25 2.5 3.25
2.5
2.25 2.5 3.25
2.5 3.25 3.5
3 3.25 1.5
2 3 2 1
3 1 4 6
1 5 3 1
2 4 1 1

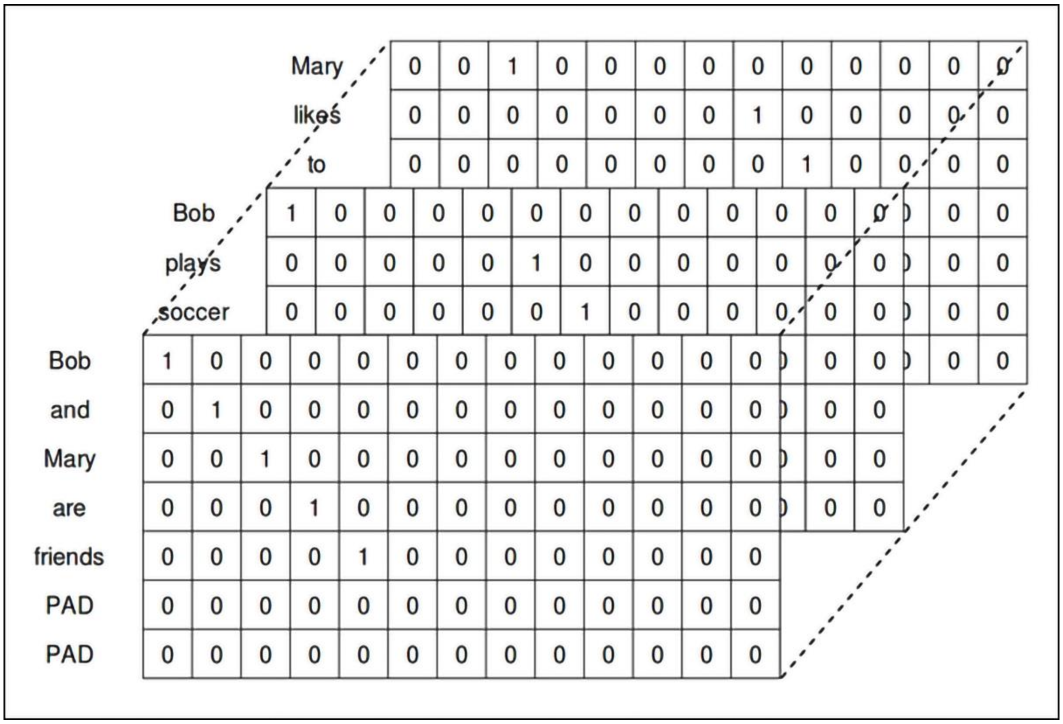

16 Feature Maps
32 Feature Maps
128 Neurons
Input Image
Conv1 Output
Pool1 Output
Conv2 Output
Pool2 Output
Pool2 Output Reshaped to ID
Fulcon1 Output
Softmax Output


Ankle boot
T-shirt/top
T-shirt/top
Dress
T-shirt/top
Pullover
Sneaker
Pullover
Sandal
Sandal


A single image
A batch of images
Batch dimension
Height dimension
Width dimension
Channel dimension (input)
Convolutional layer
(tf.keras.layers.Conv2D)
A feature map
A batch of feature maps
Batch dimension
Height dimension (output)
Width dimension (output)
Channel dimension (output)

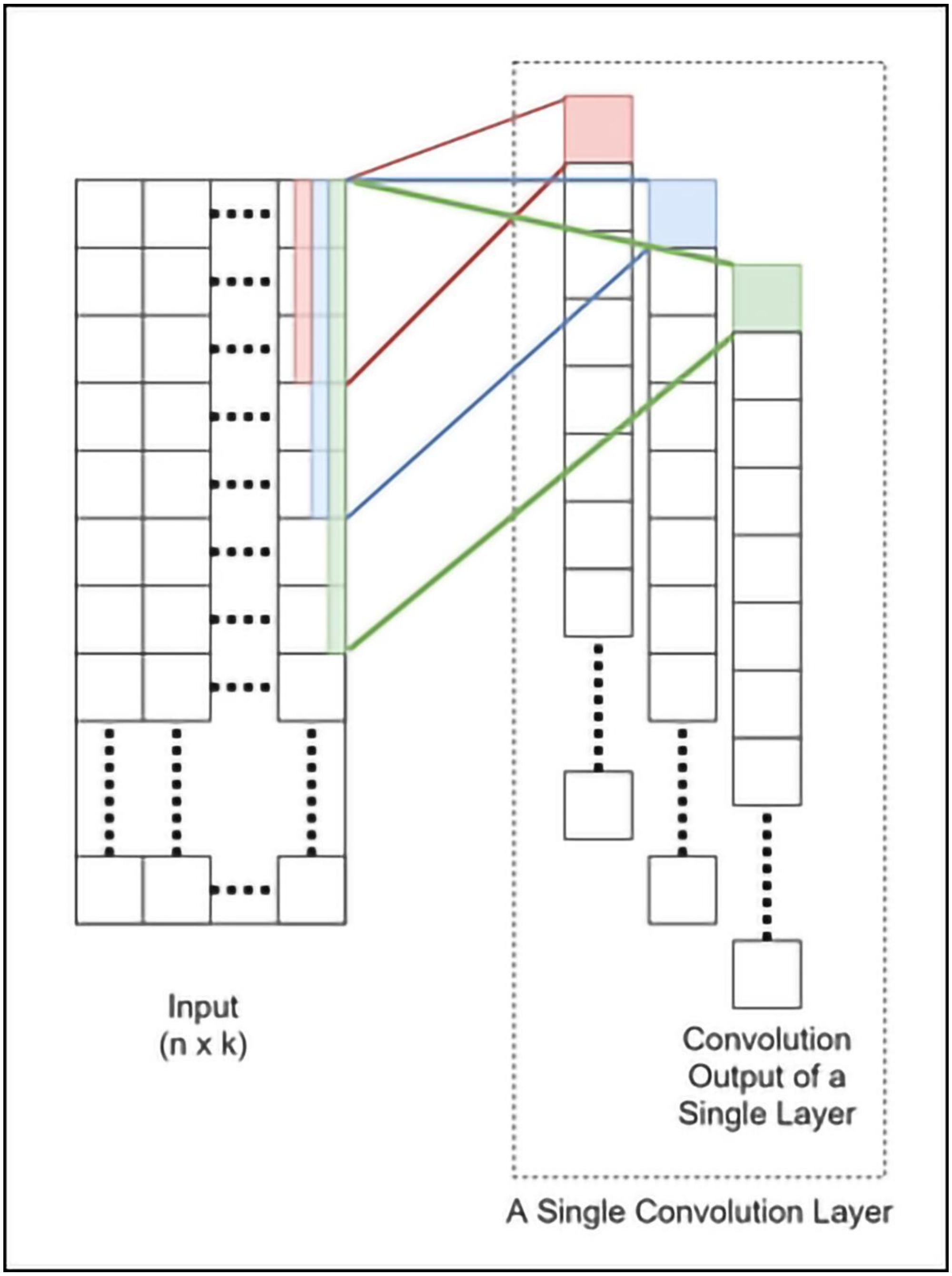


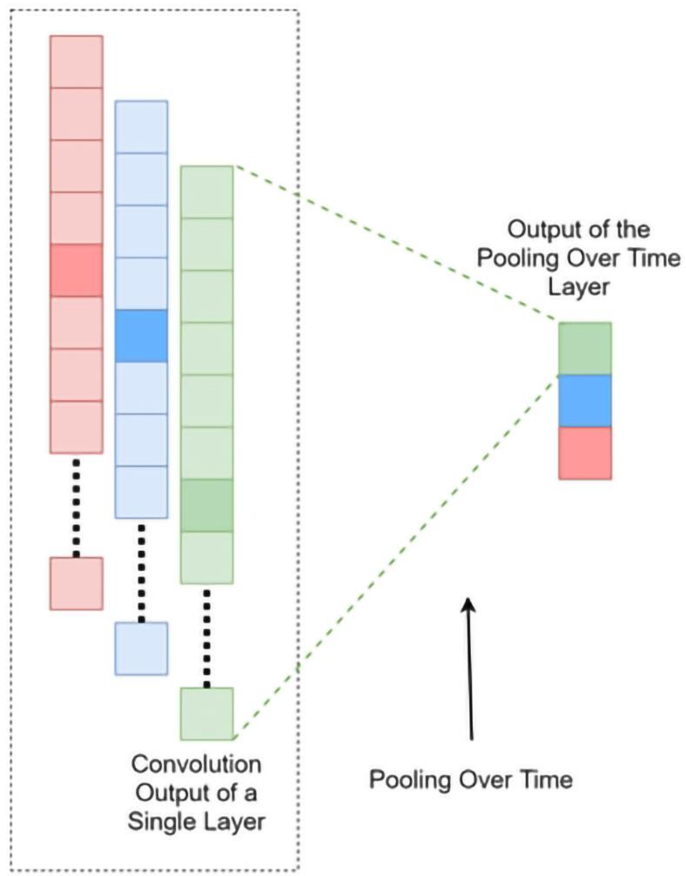
Correctly Classified Samples
Softmax Predictions for Correctly Classified Samples
Ankle boot
Bag
Sneaker
Shirt
Sandal
Coat
Dress
Pullover
Trouser
T-shirt/top
Softmax Predictions for Incorrectly Classified Samples
Softmax Predictions for Incorrectly Classified Samples


Mary
likes
to
Bob
plays
soccer
Bob
and
Mary
are
friends
PAD
PAD

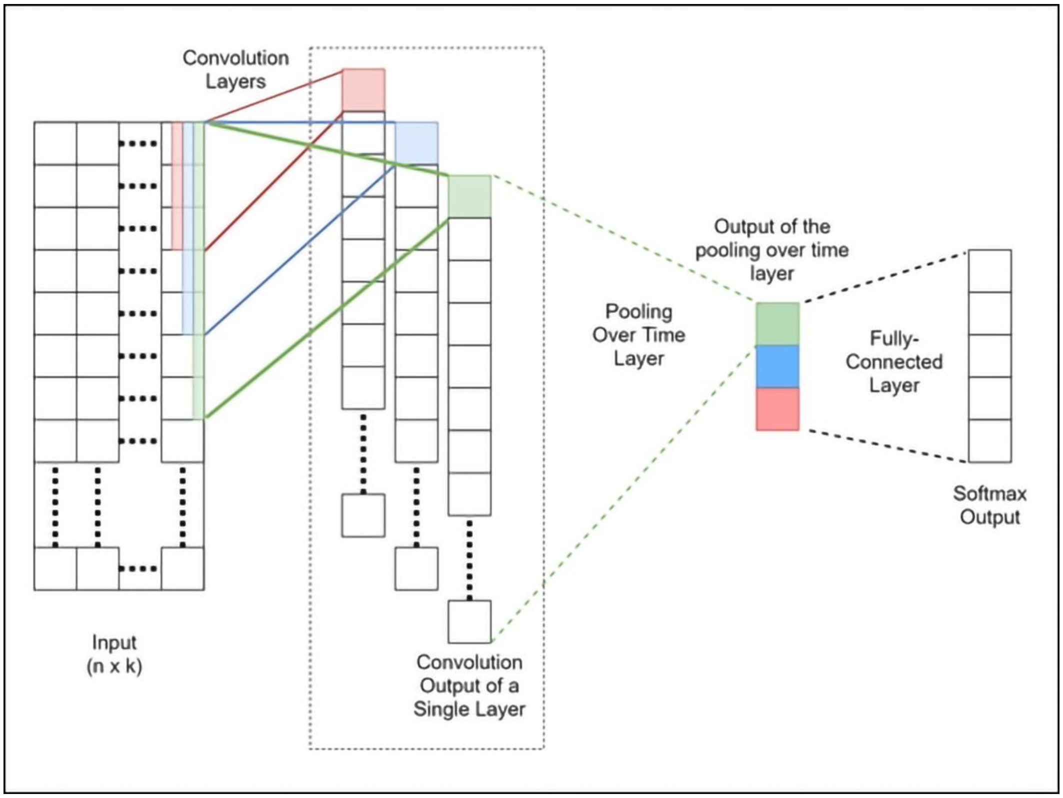
	question	category	sub_category
0	how did serfdom develop in and then leave russ...	DESC	manner
1	what films featured the character popeye doyle ?	ENTY	cremat
2	how can i find a list of celebrities ' real na...	DESC	manner
3	what fowl grabs the spotlight after the chines...	ENTY	animal
4	what is the full form of .com ?	ABBR	exp
5	what contemptible scoundrel stole the cork fro...	HUM	ind
6	what team did baseball 's st. louis browns bec...	HUM	gr
7	what is the oldest profession ?	HUM	title
8	what are liver enzymes ?	DESC	def
9	name the scar-faced bounty hunter of the old w...	HUM	ind

	question	category	sub_category
4343	how can i get started in writing for television ?	0	manner
2318	what were the achievements of richard nixon ?	1	other
2808	what was the alternate to vhs ?	1	other
3217	what country would you visit to ski in the dol...	2	country
3966	what country imposed the berlin blockade in 19...	2	country
5189	where on the web is adventours tours from sydn...	2	other
1675	what u.s. state ends with a g ?	2	state
1408	what country was sir edmund hillary born in ?	2	country
4916	what was the name of the u.s. navy gunboat in ...	1	veh
730	what 19th-century writer had a country estate ...	3	ind

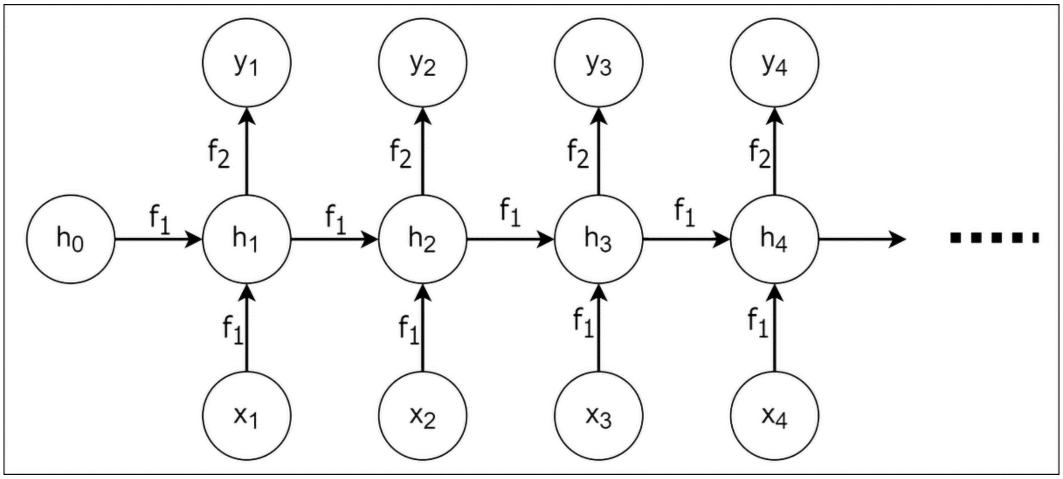

Input
(n x k)
Convolution
Output of a
Single Layer
A Single Convolution Layer

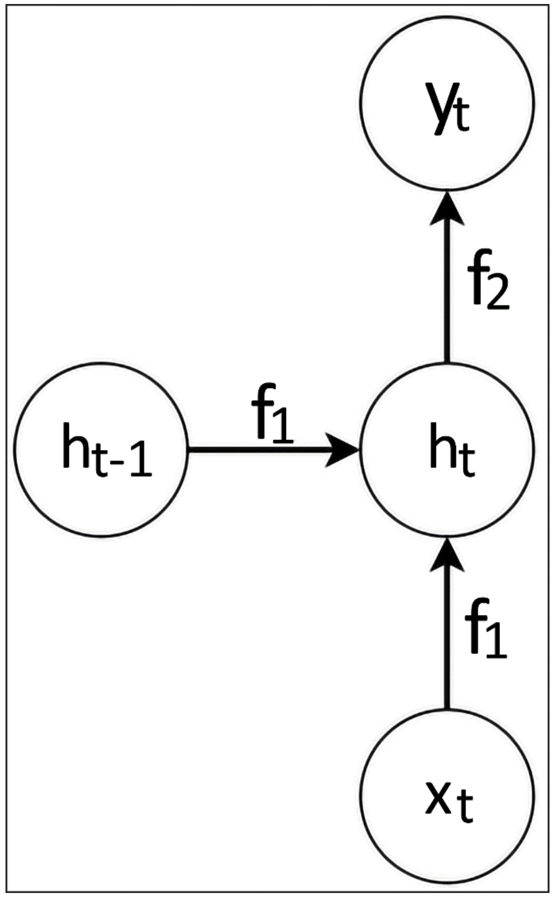


*We denote the maximum value of a parallel convolution layer with a darker color

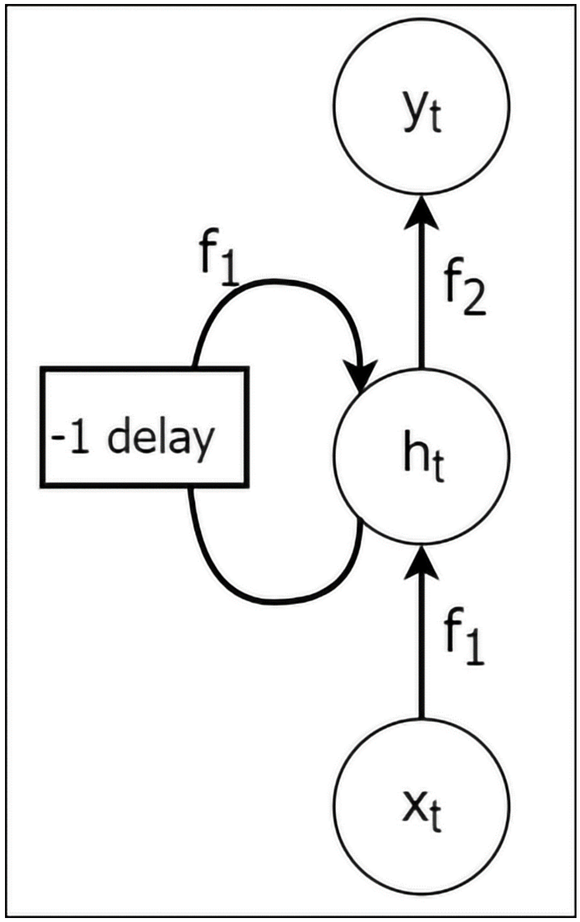

Convolution
Layers
Input
(n x k)
Convolution
Output of a
Single Layer
Pooling
Over Time
Layer
Output of the
pooling over time
layer
Fully-
Connected
Layer
Softmax
Output

Chapter 6: Recurrent Neural Networks

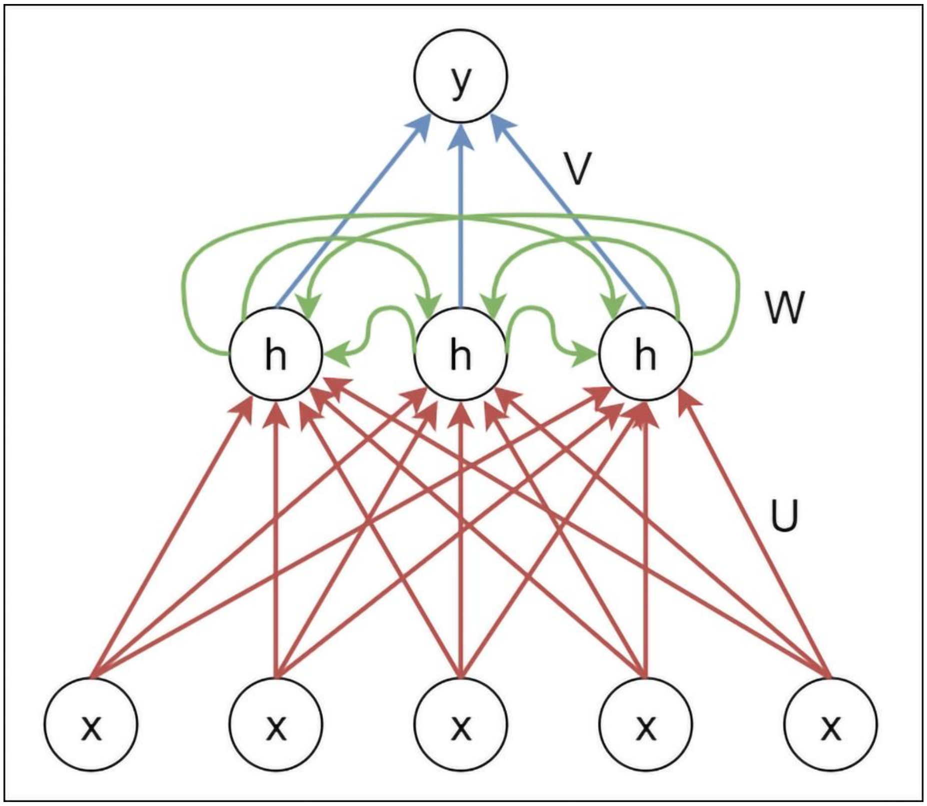


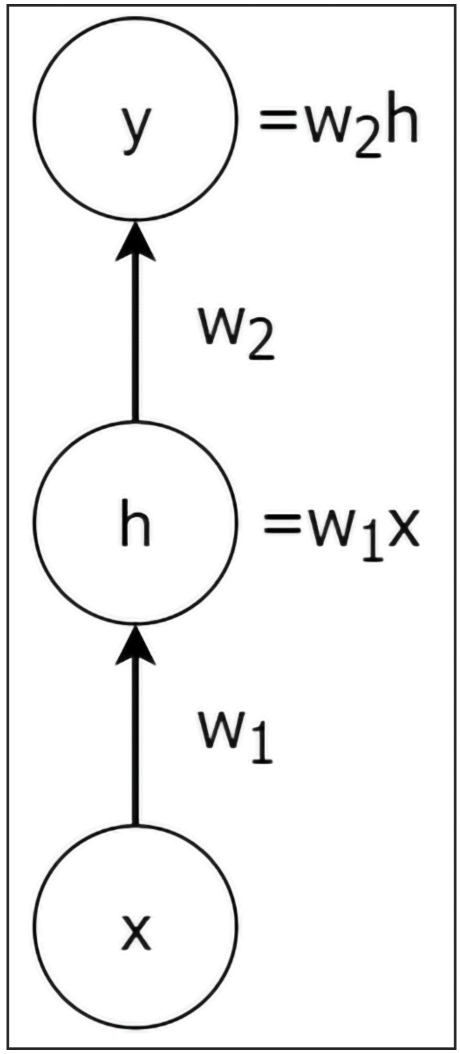


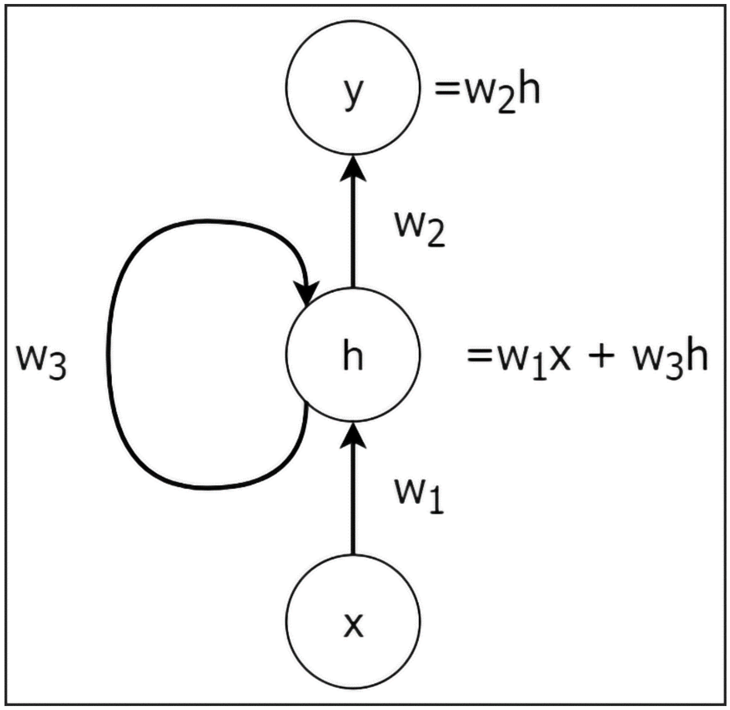
y_t
f_2
h_{t-1}
f_1
h_t
f_1
x_t

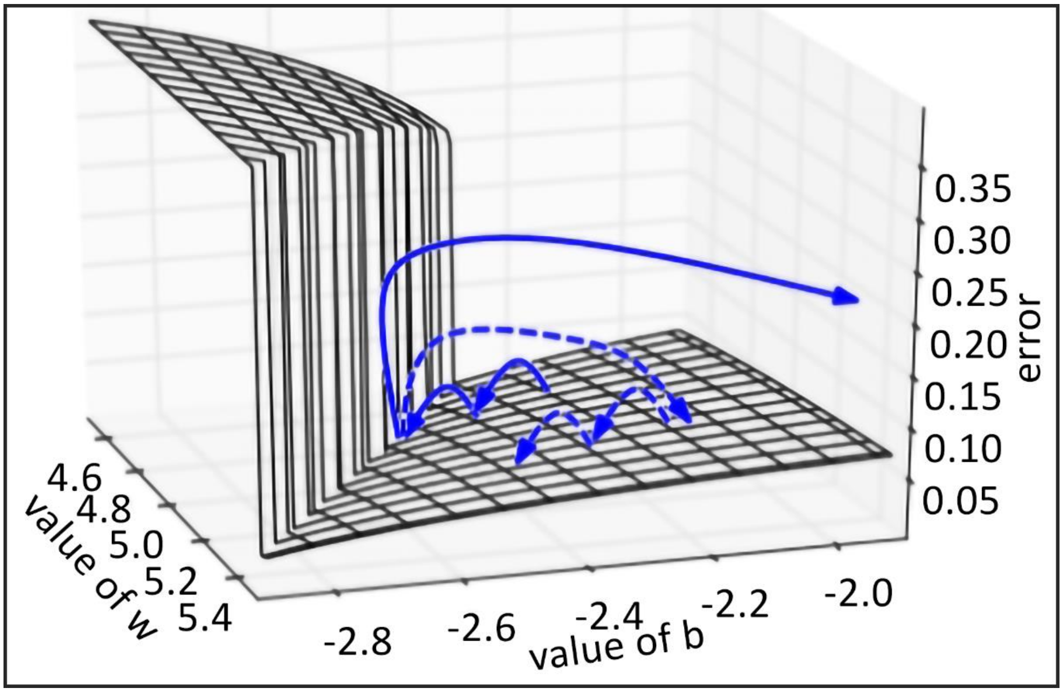


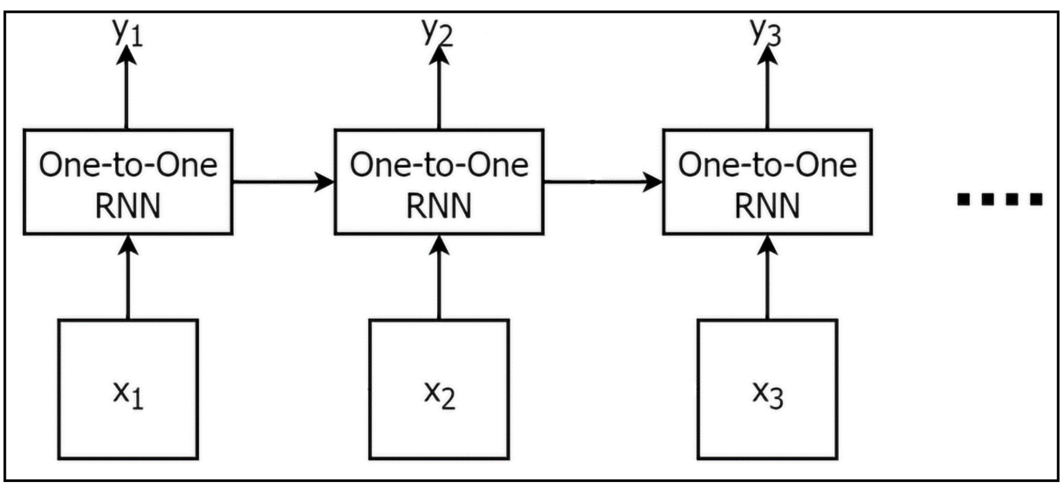
y_t
f_1
f_2
-1 delay
h_t
f_1
x_t

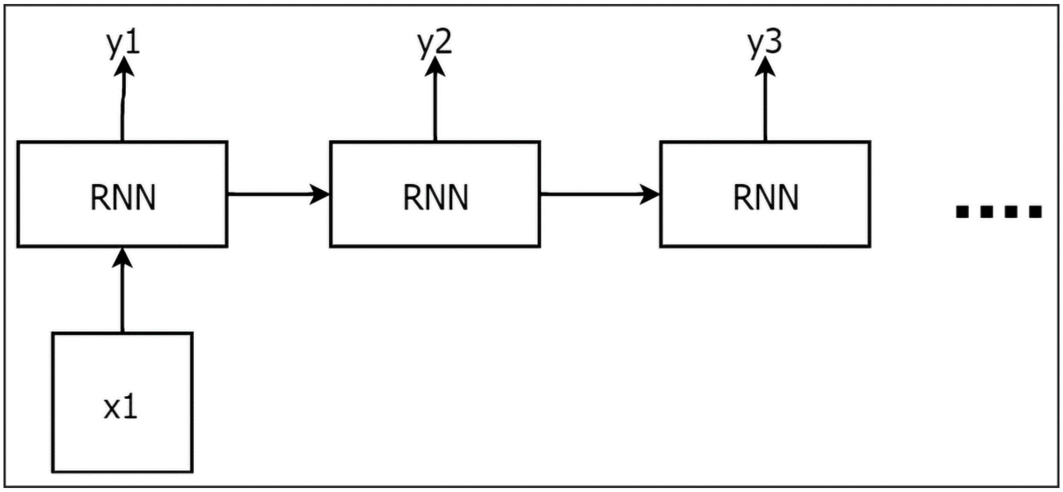

y
V
W
h
h
h
U
x
x
x
x
x

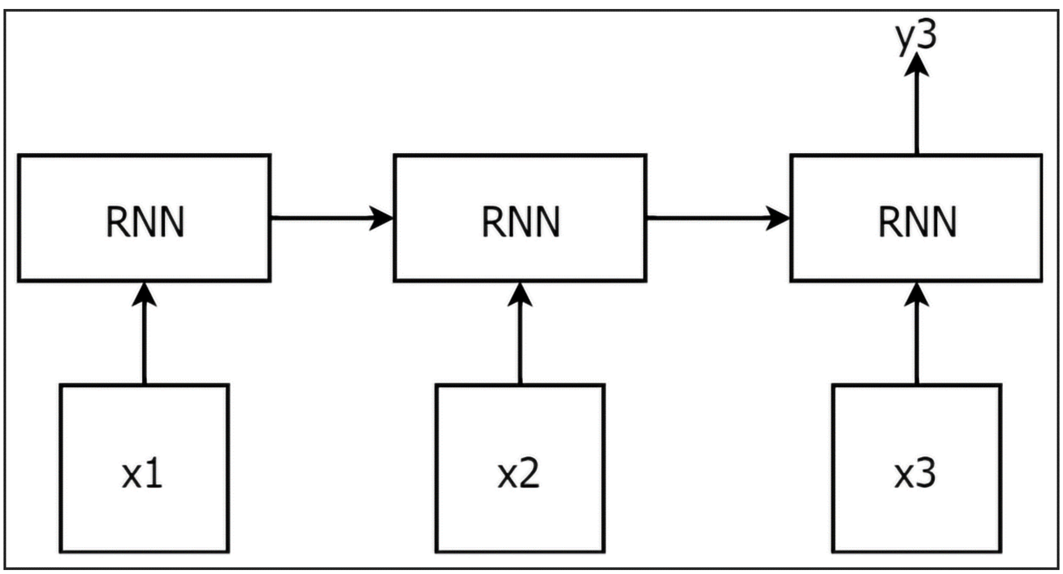

y
=w2h
w2
h
=w1x
w1
x

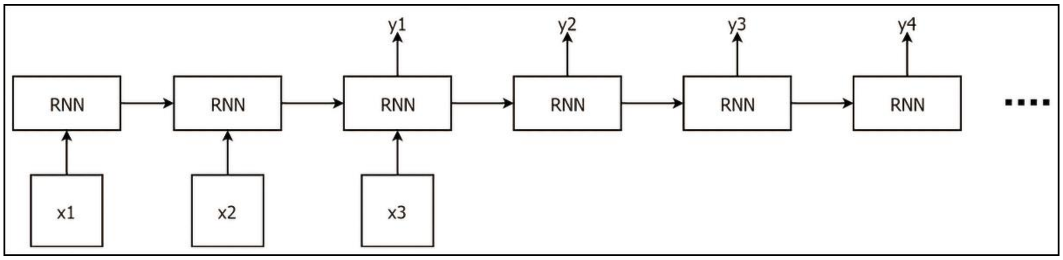

y
=w2h
w2
w3
h
=w1x + w3h
w1
x

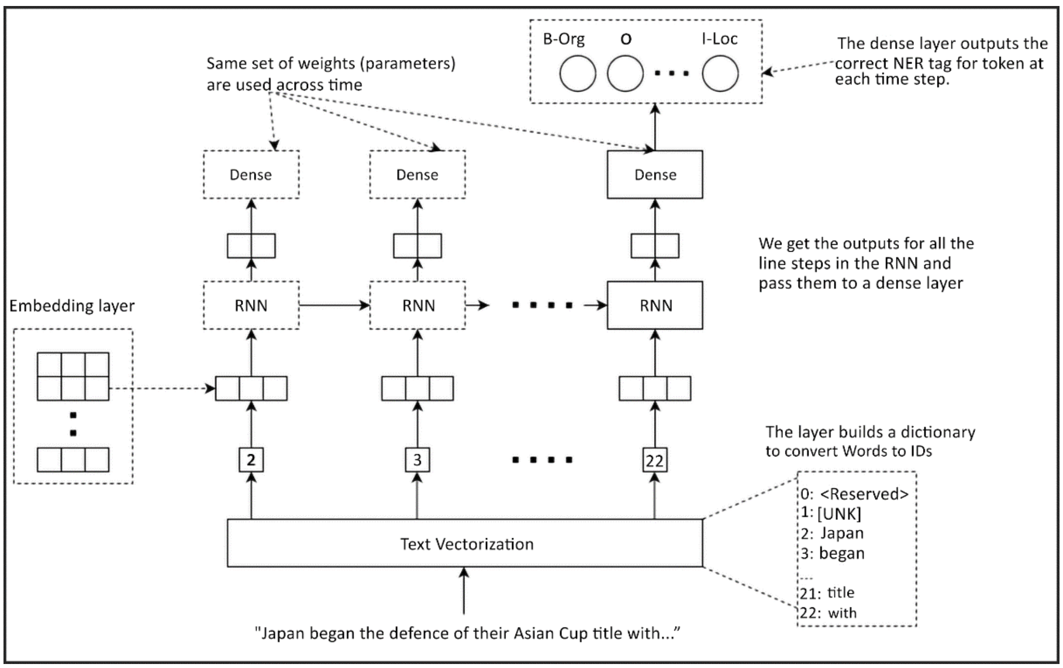

0.35
0.30
0.25
0.20
0.15
0.10
0.05
error
4.6
4.8
5.0
5.2
5.4
value of w
-2.8
-2.6
-2.4
-2.2
-2.0
value of b

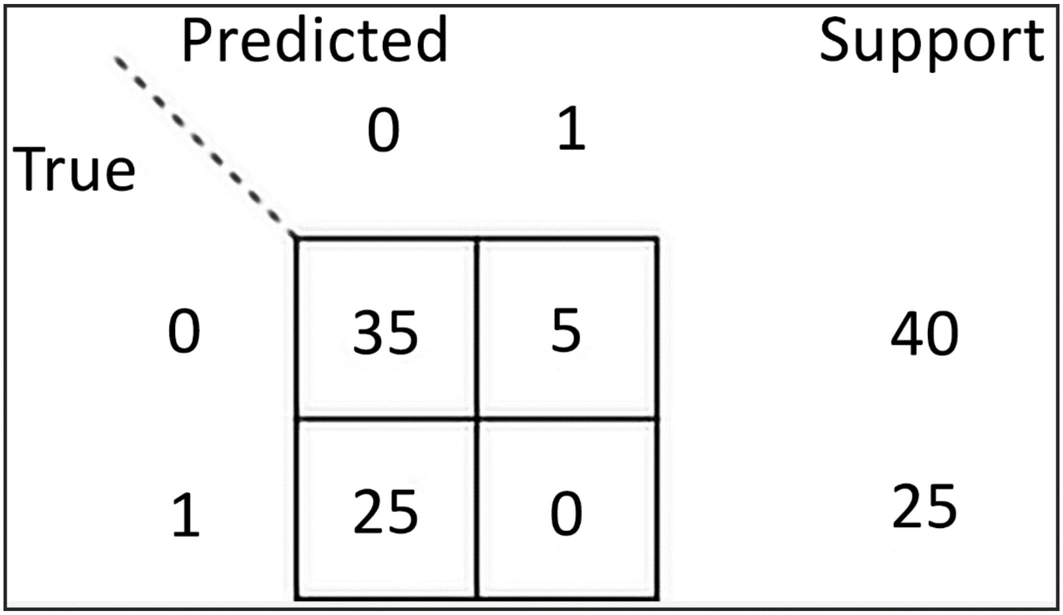

y1
y2
y3
One-to-One RNN
One-to-One RNN
One-to-One RNN
x1
x2
x3

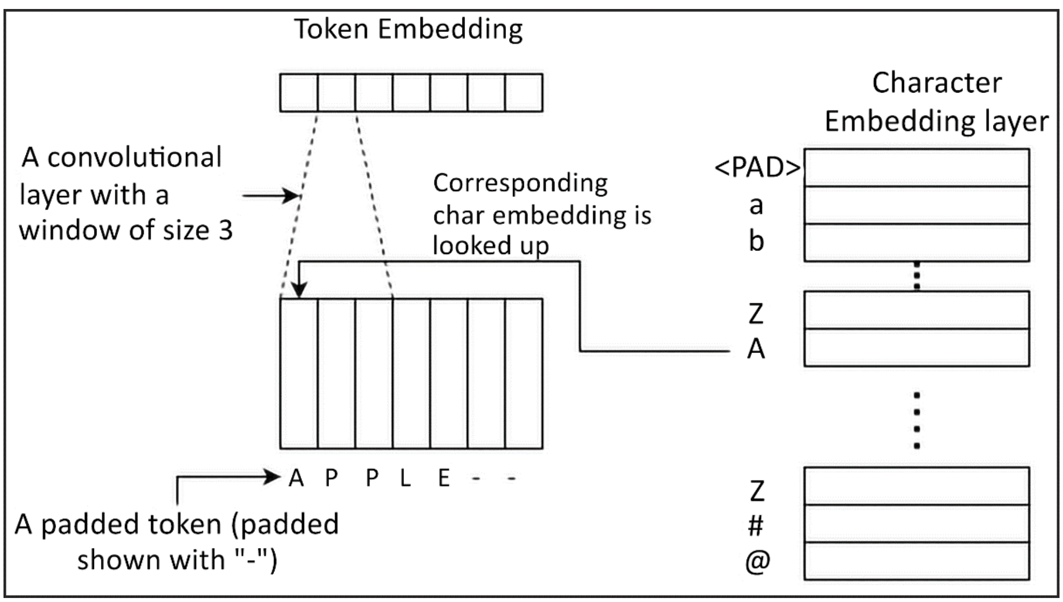

y1
y2
y3
RNN
RNN
RNN
x1


y3
RNN
RNN
RNN
x1
x2
x3

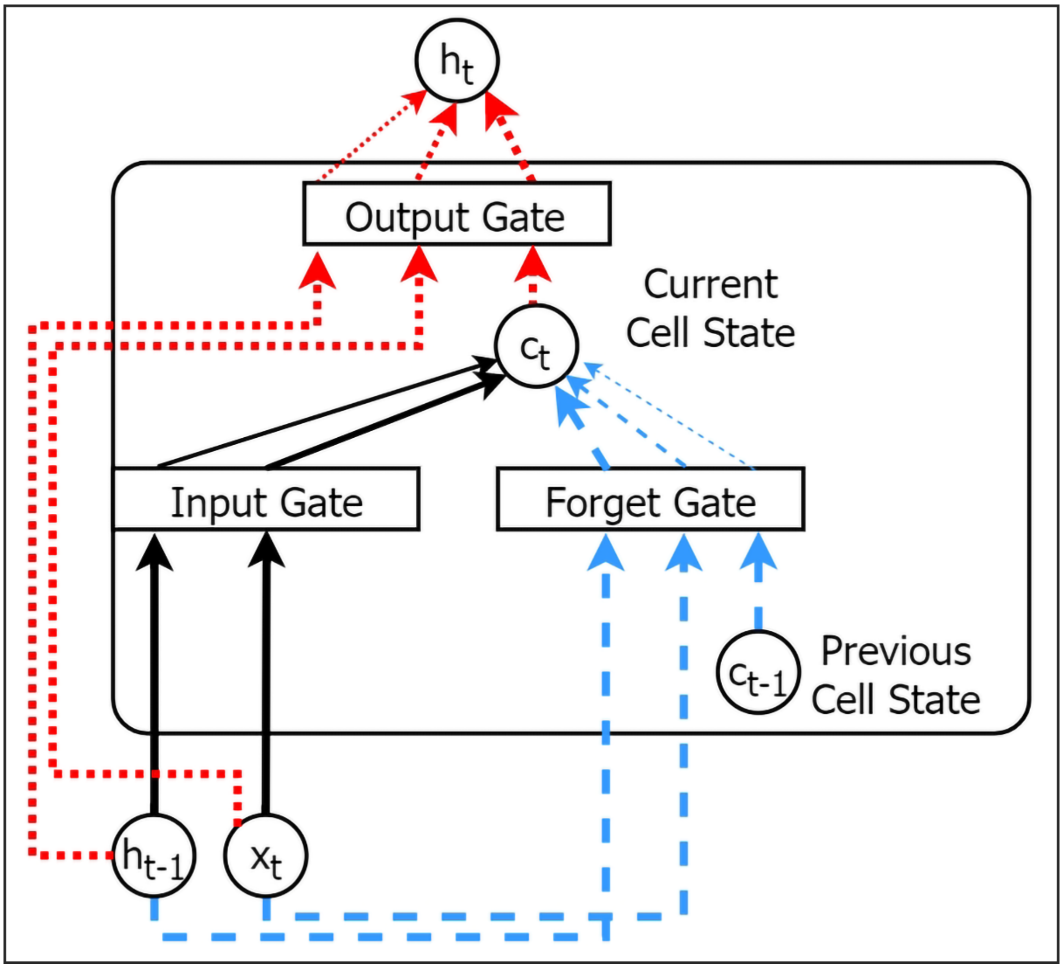

y1
y2
y3
y4
RNN
RNN
RNN
RNN
RNN
RNN
x1
x2
x3

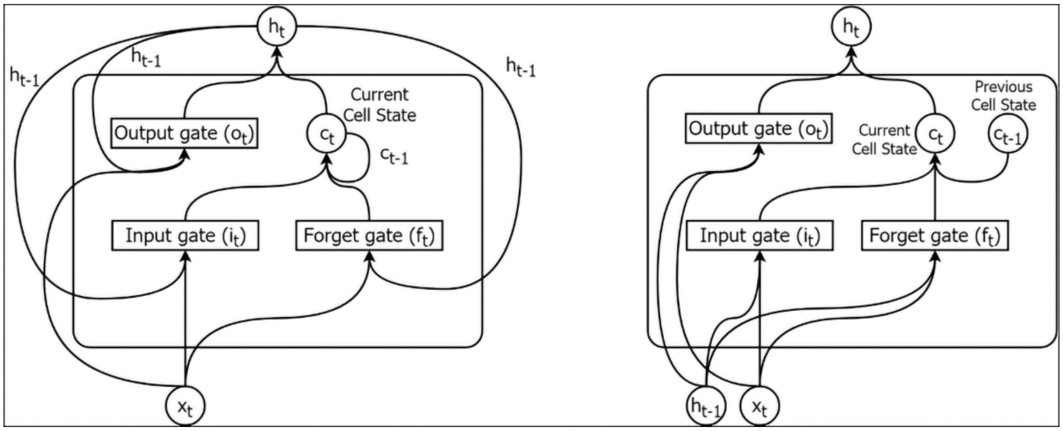

B-Org
O
I-Loc
The dense layer outputs the correct NER tag for token at each time step.
Same set of weights (parameters) are used across time
Dense
Dense
Dense
We get the outputs for all the line steps in the RNN and pass them to a dense layer
Embedding layer
RNN
RNN
RNN
2
3
22
The layer builds a dictionary to convert Words to IDs
0: <Reserved>
1: [UNK]
2: Japan
3: began
...
21: title
22: with
Text Vectorization
"Japan began the defence of their Asian Cup title with..."


Predicted
Support
True
0
1
0
35
5
40
1
25
0
25

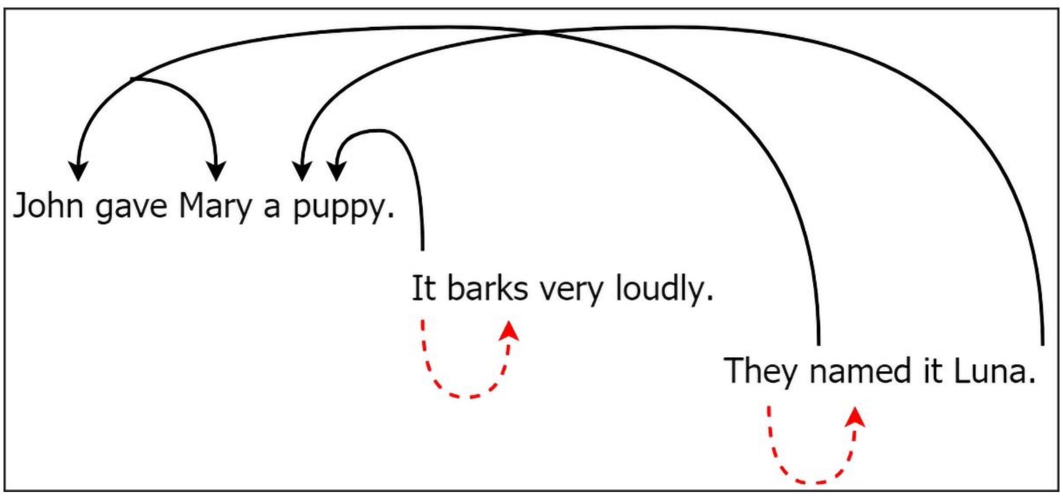

Token Embedding
Character
Embedding layer
A convolutional
layer with a
window of size 3
Corresponding
char embedding is
looked up
<PAD>
a
b
Z
A
A P P L E - -
A padded token (padded
shown with "-")
Z
#
@

Chapter 7: Understanding Long Short-Term Memory Networks

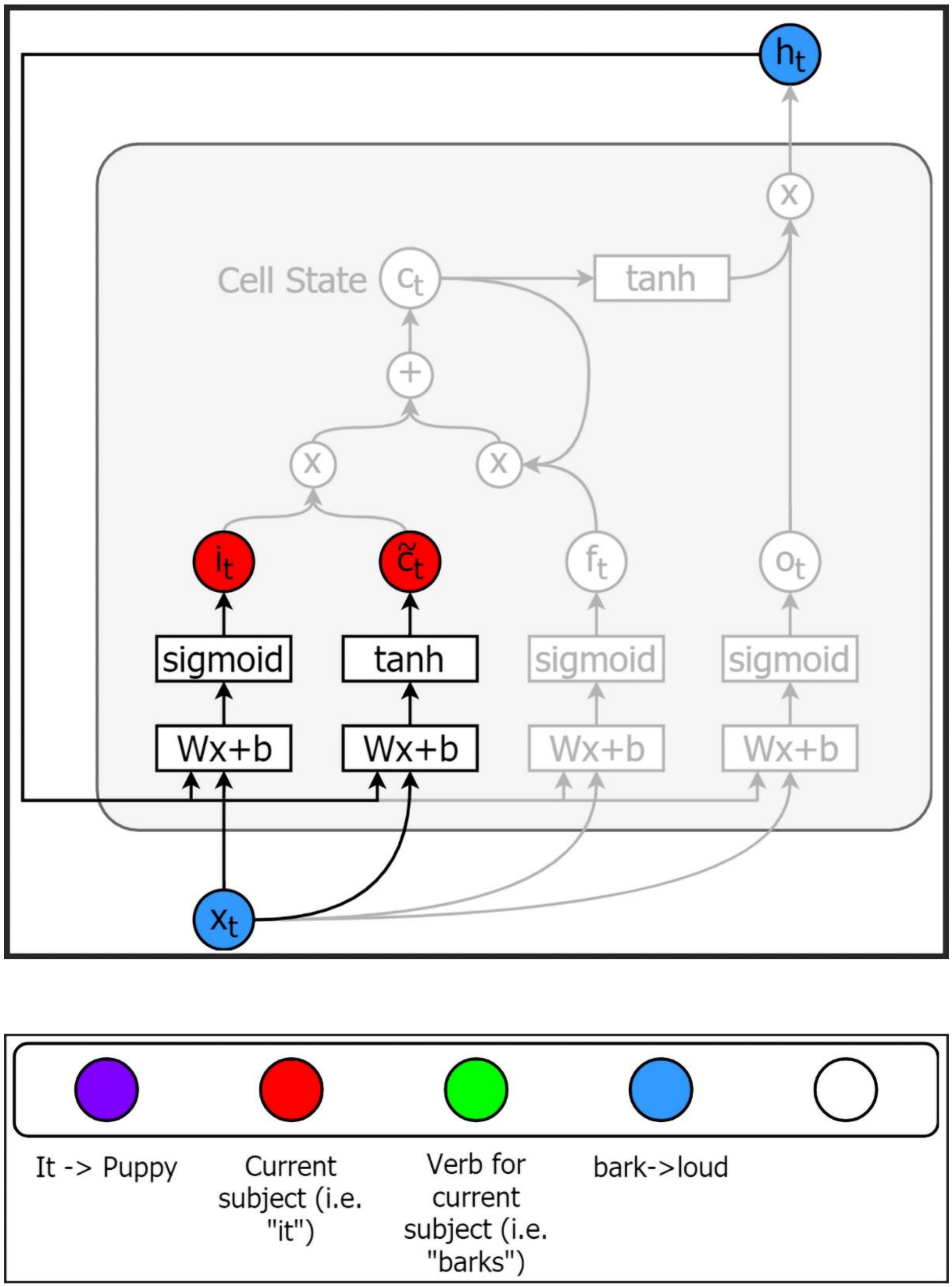




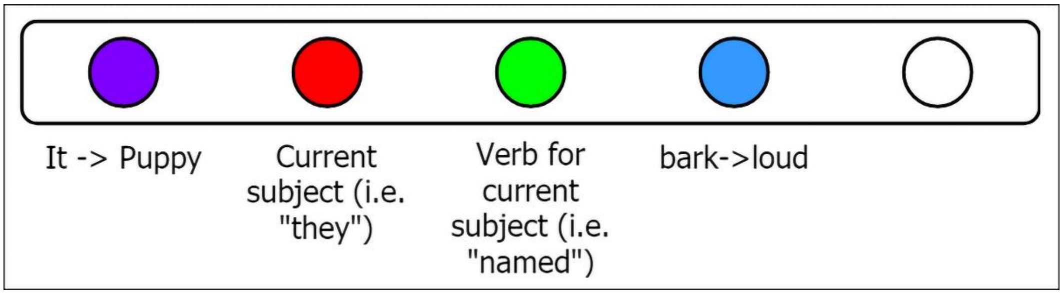
h_t
h_{t-1}
h_{t-1}
h_{t-1}
Current Cell State
Output gate (o_t)
c_t
c_{t-1}
Input gate (i_t)
Forget gate (f_t)
x_t
h_t
Previous Cell State
Output gate (o_t)
Current Cell State
c_t
c_{t-1}
Input gate (i_t)
Forget gate (f_t)
h_{t-1}
x_t

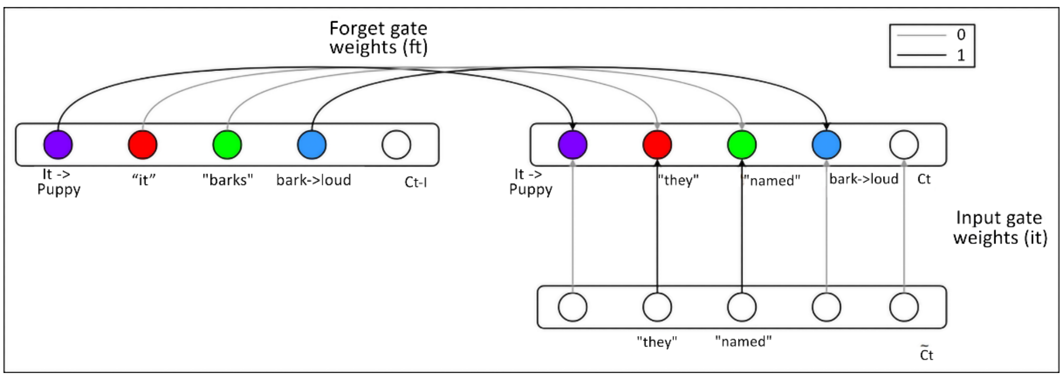


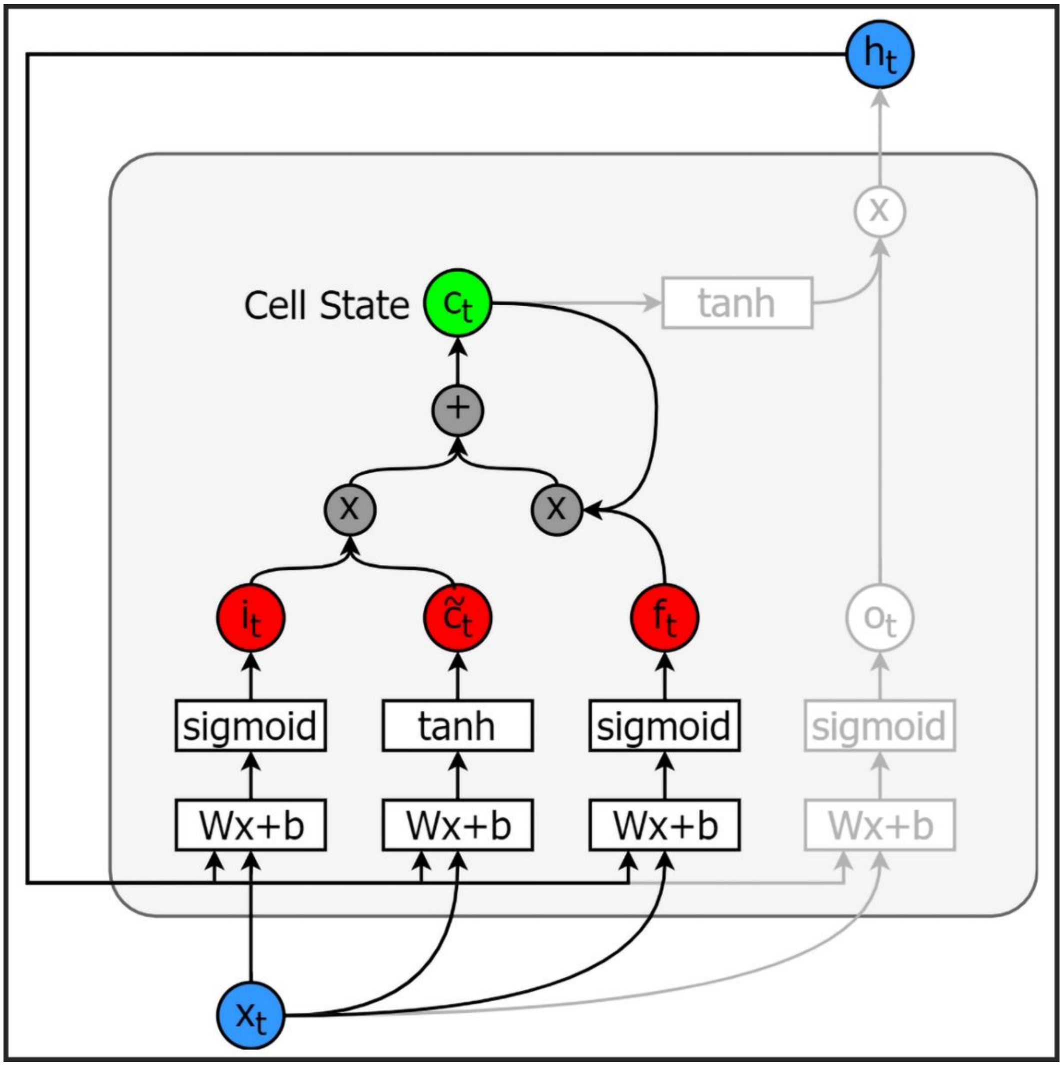
John gave Mary a puppy.
It barks very loudly.
They named it Luna.

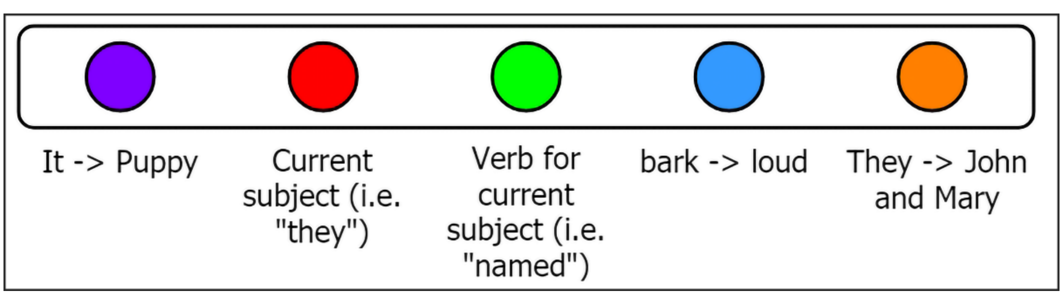


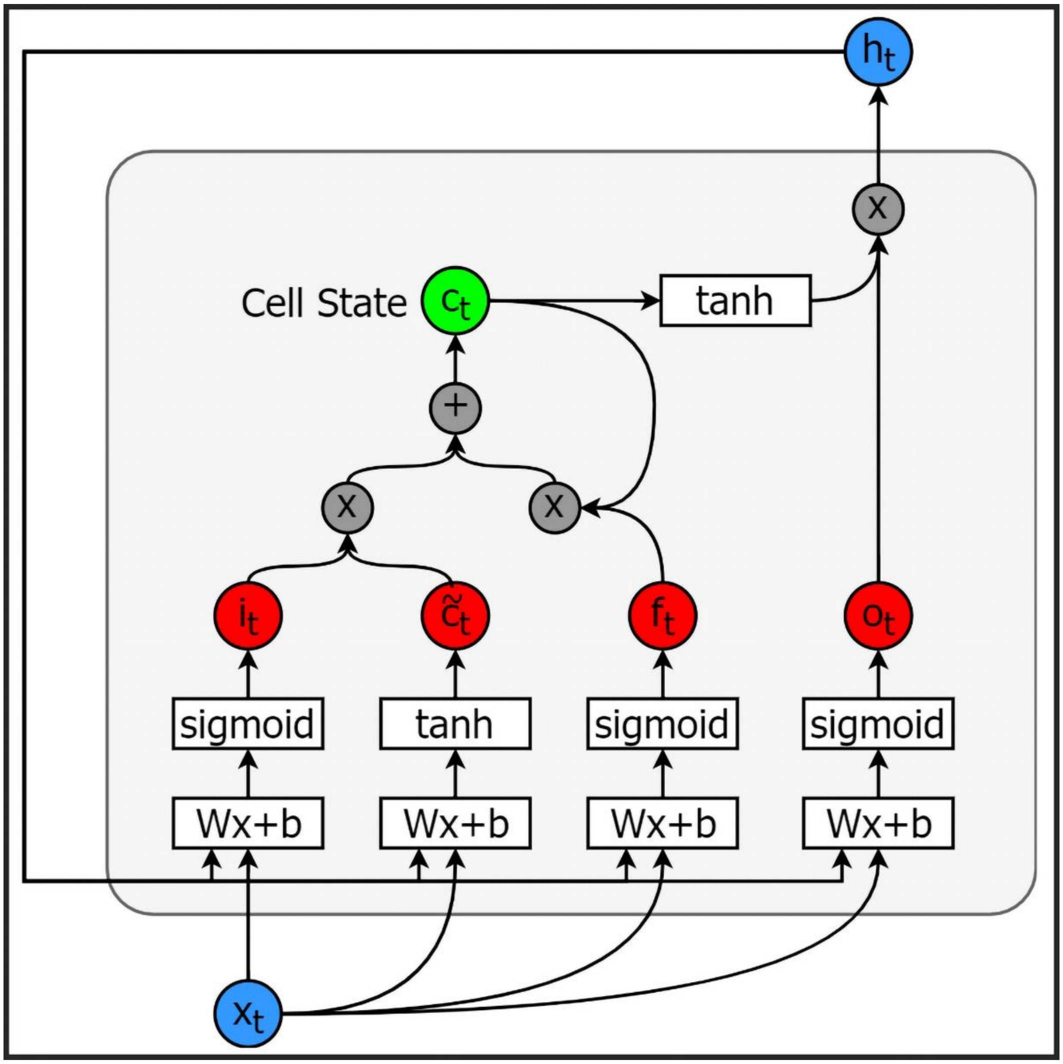
h_t
Cell State
c_t
tanh
x
+
x
x
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t
It -> Puppy
Current subject (i.e. "it")
Verb for current subject (i.e. "barks")
bark->loud


It -> Puppy
Current subject (i.e. "they")
Verb for current subject (i.e. "named")
bark->loud

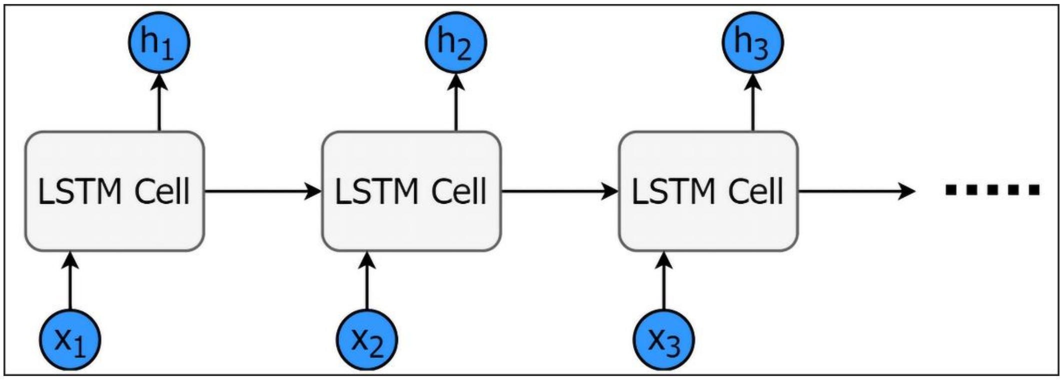

Forget gate weights (ft)
0
1
It -> Puppy
"it"
"barks"
bark->loud
Ct-l
It -> Puppy
"they"
"named"
bark->loud
Ct
Input gate weights (it)
"they"
"named"
C̃t

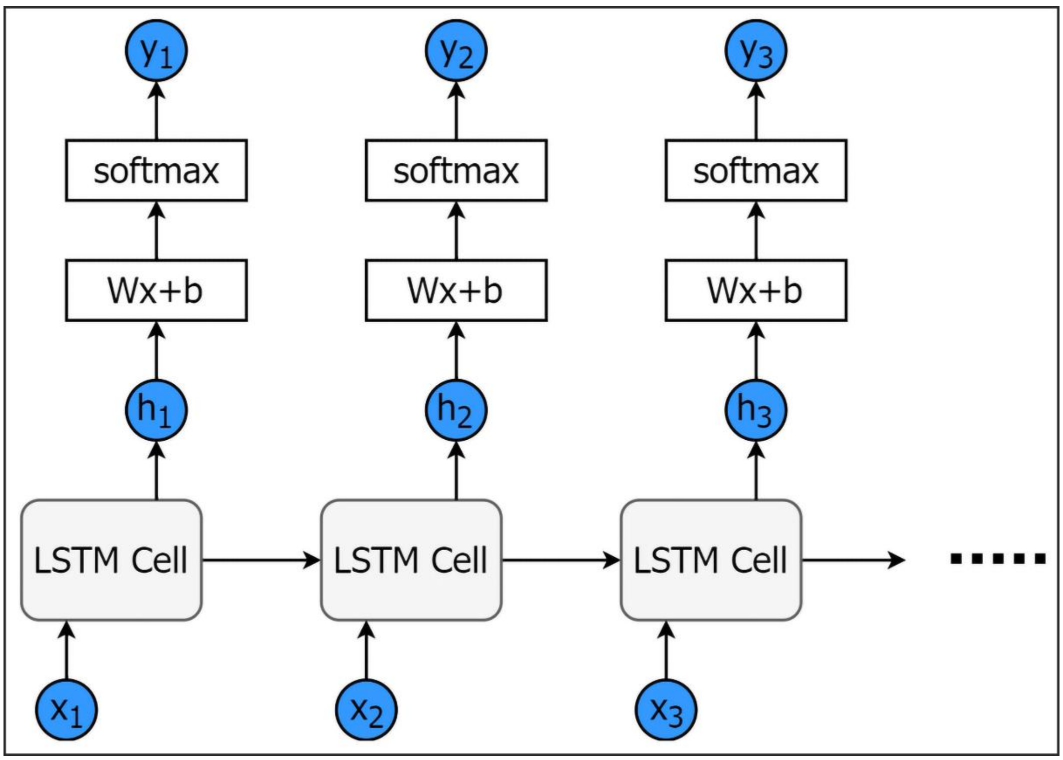

h_t
x
tanh
Cell State
c_t
+
x
x
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t

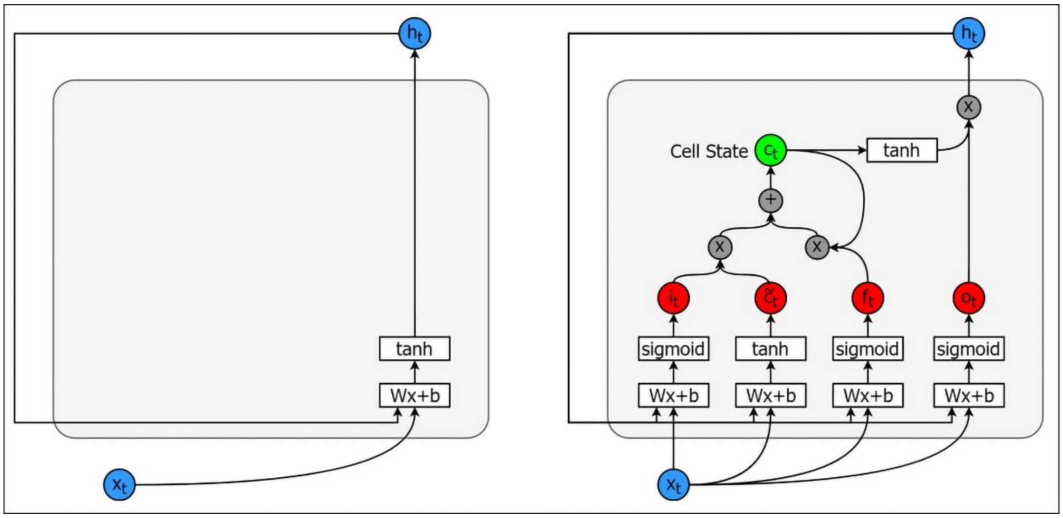

It -> Puppy
Current subject (i.e. "they")
Verb for current subject (i.e. "named")
bark -> loud
They -> John and Mary



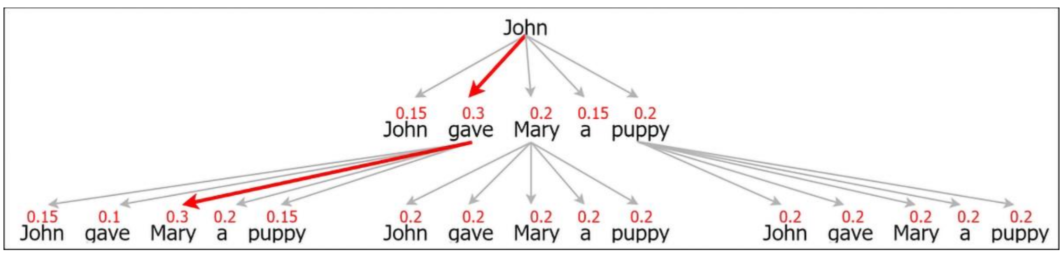
h_t
X
Cell State
c_t
tanh
+
X
X
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t


h1
h2
h3
LSTM Cell
LSTM Cell
LSTM Cell
x1
x2
x3

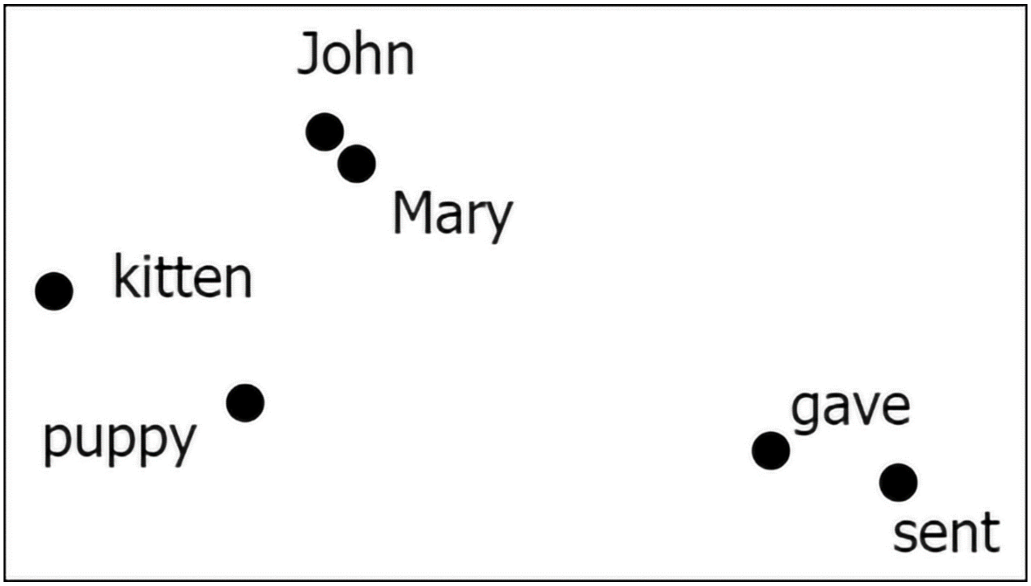

y1
y2
y3
softmax
softmax
softmax
Wx+b
Wx+b
Wx+b
h1
h2
h3
LSTM Cell
LSTM Cell
LSTM Cell
x1
x2
x3

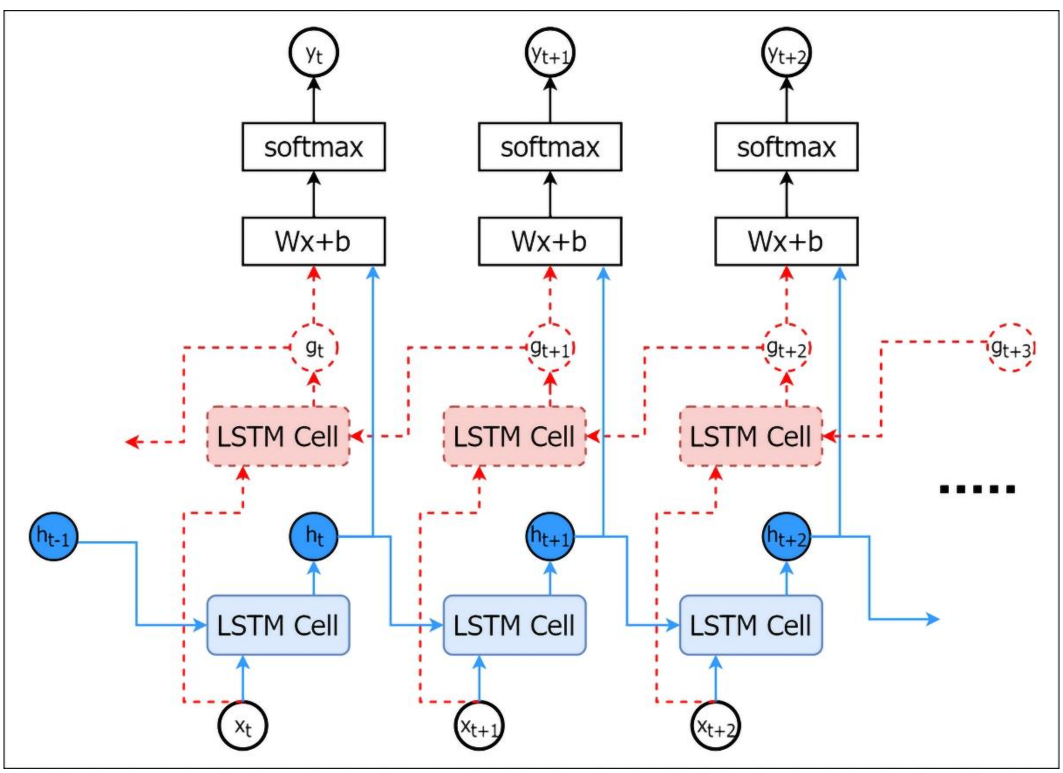


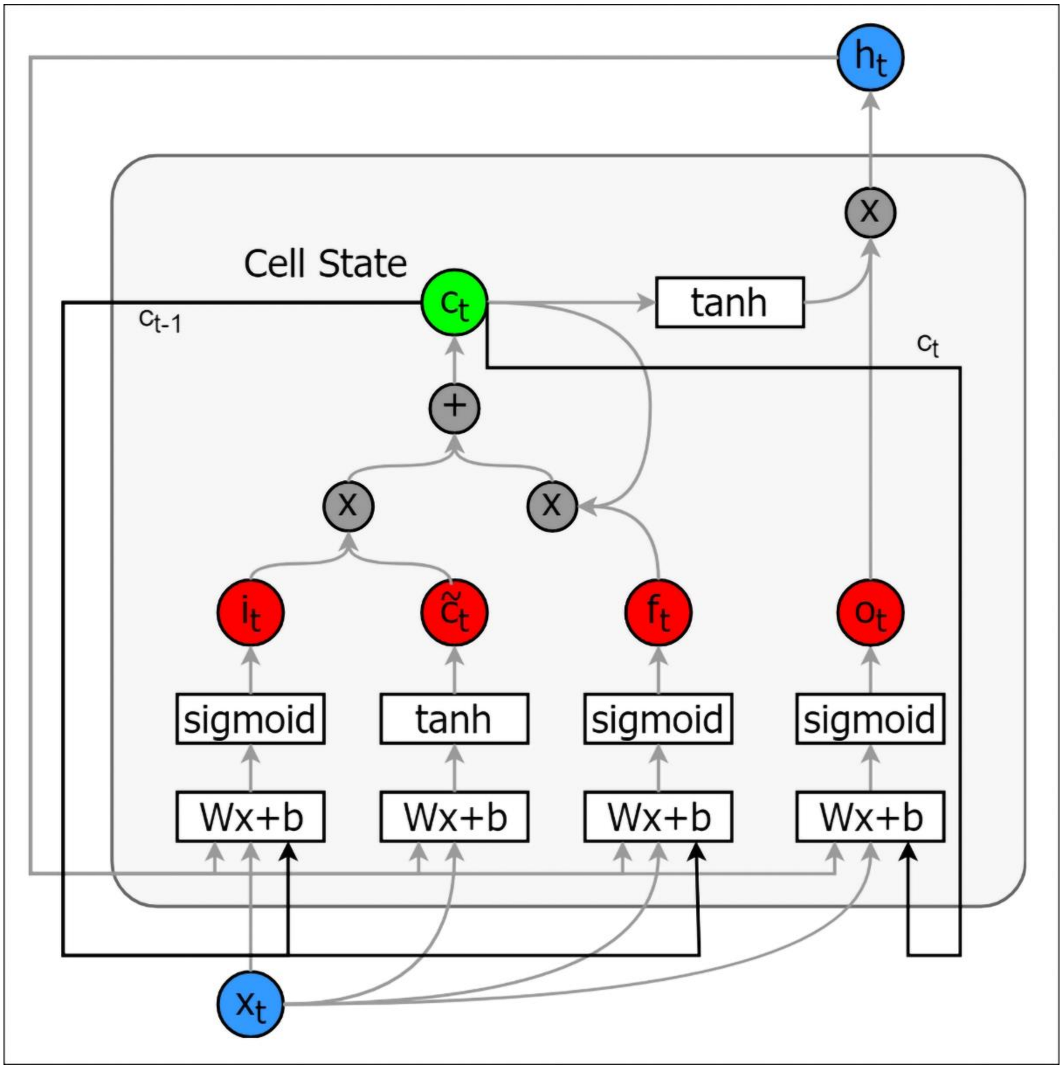
h_t
tanh
Wx+b
x_t
h_t
x
Cell State
c_t
tanh
+
x
x
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t



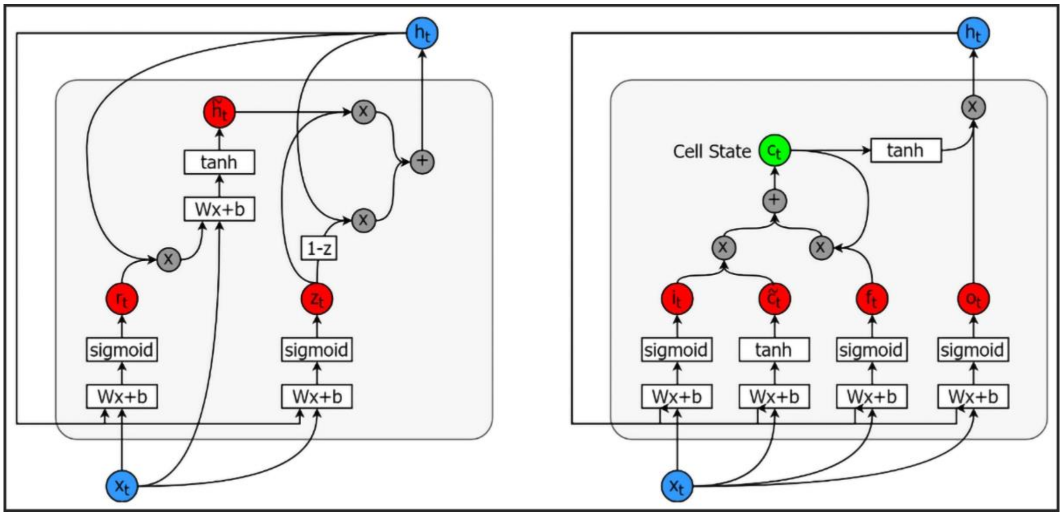
John
0.15 0.3 0.2 0.15 0.2
John gave Mary a puppy
0.15 0.1 0.3 0.2 0.15
John gave Mary a puppy
0.2 0.2 0.2 0.2 0.2
John gave Mary a puppy
0.2 0.2 0.2 0.2 0.2
John gave Mary a puppy


John
Mary
kitten
puppy
gave
sent


y_t
y_{t+1}
y_{t+2}
softmax
softmax
softmax
Wx+b
Wx+b
Wx+b
g_t
g_{t+1}
g_{t+2}
g_{t+3}
LSTM Cell
LSTM Cell
LSTM Cell
h_{t-1}
h_t
h_{t+1}
h_{t+2}
LSTM Cell
LSTM Cell
LSTM Cell
x_t
x_{t+1}
x_{t+2}

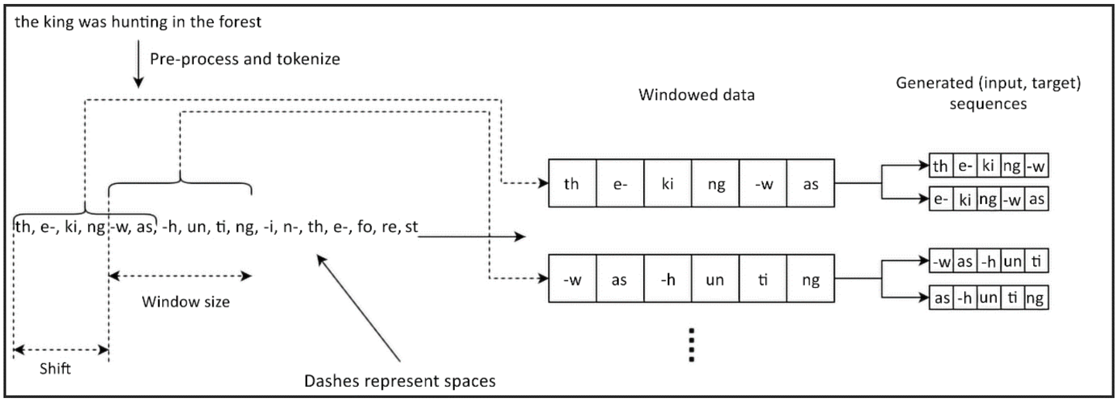


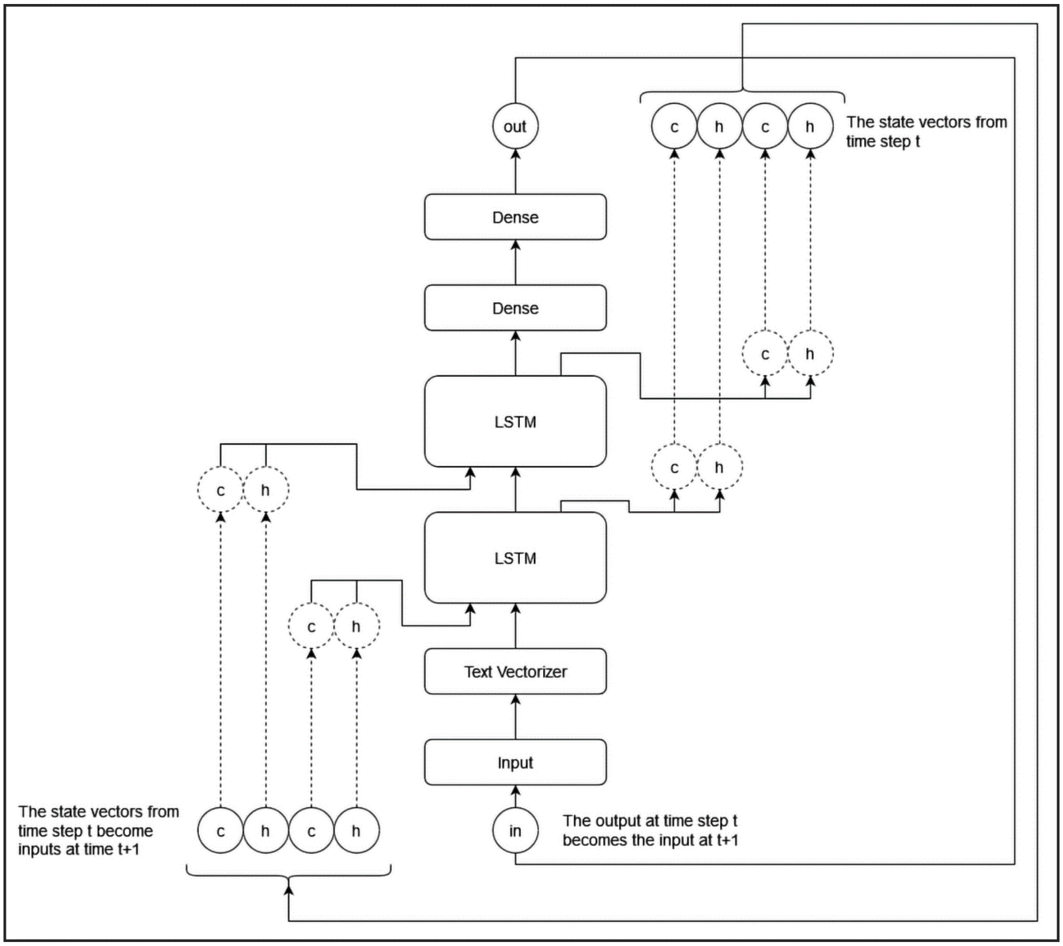
h_t
X
Cell State
c_t
tanh
c_{t-1}
c_t
+
X
X
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t



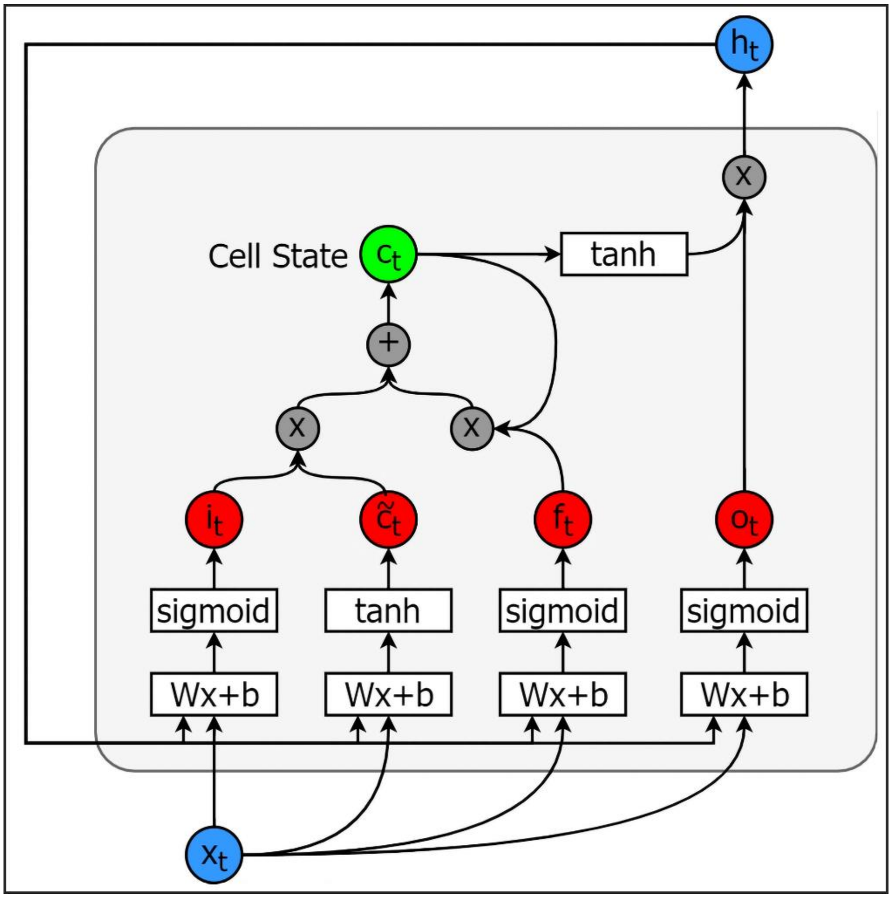
h_t
$\tilde{h}_t$
tanh
Wx+b
1-z
r_t
z_t
sigmoid
Wx+b
x_t
h_t
Cell State
c_t
tanh
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
Wx+b
x_t

Chapter 8: Applications of LSTM – Generating Text

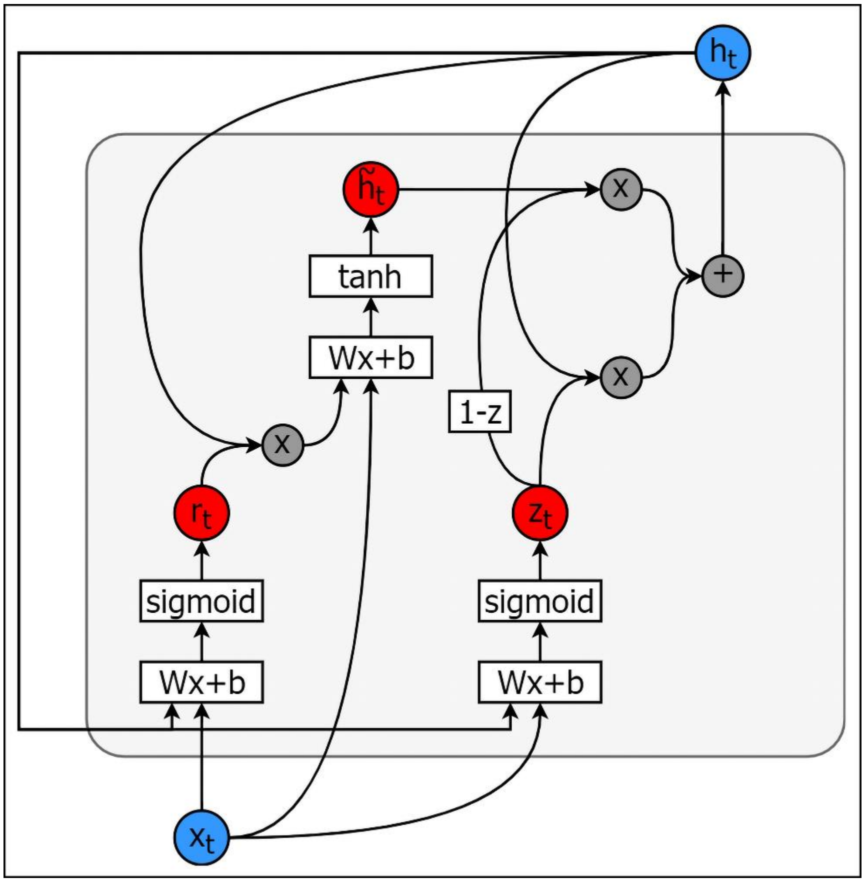




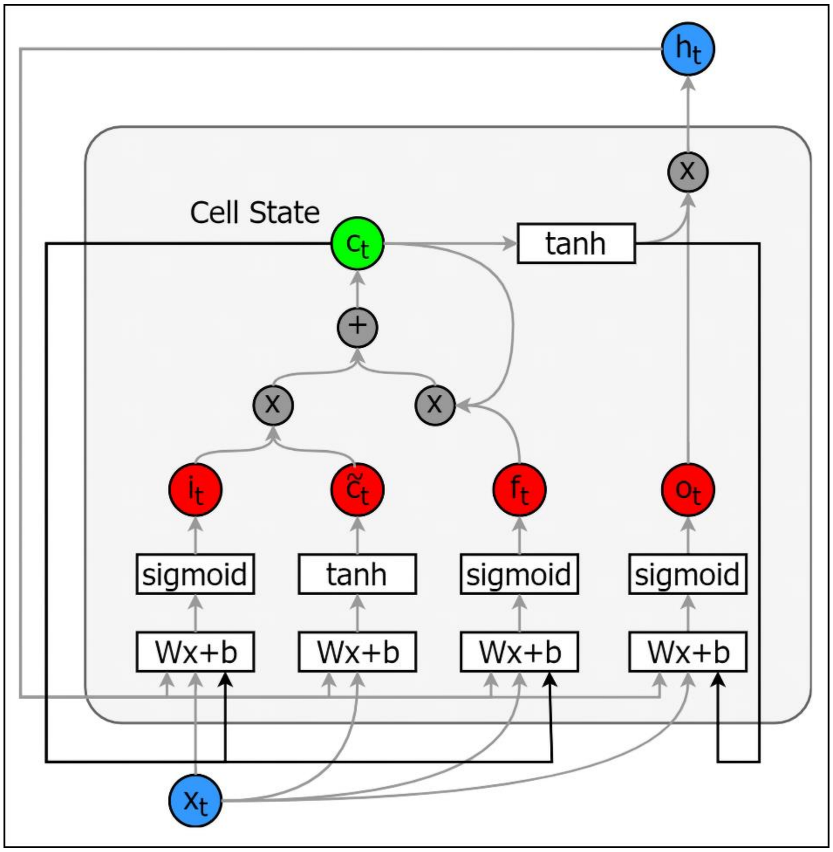
out
Dense
Dense
LSTM
LSTM
Text Vectorizer
Input
in
c
h
c
h
The state vectors from time step t
The state vectors from time step t become inputs at time t+1
The output at time step t becomes the input at t+1



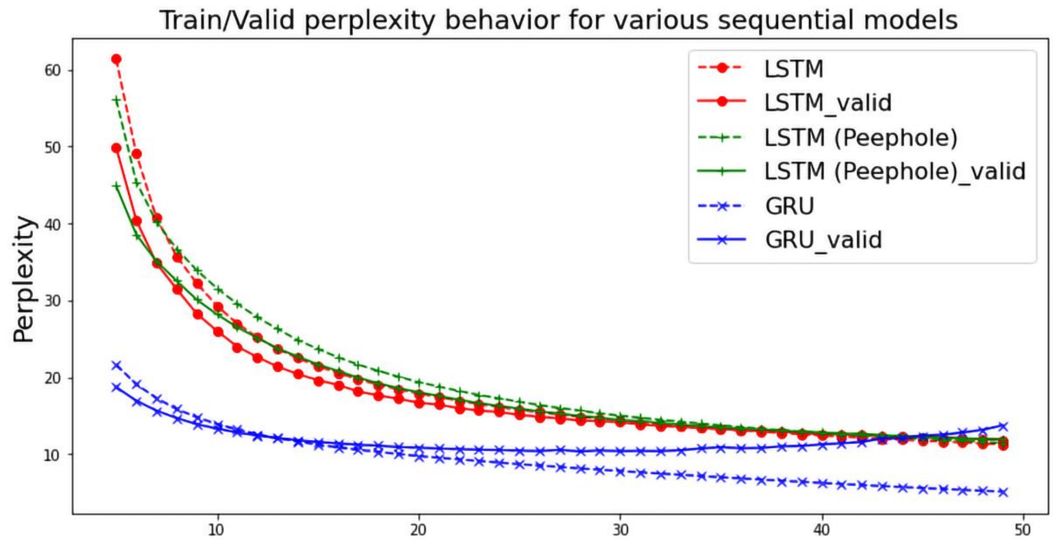
h_t
X
Cell State
c_t
tanh
+
X
X
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t

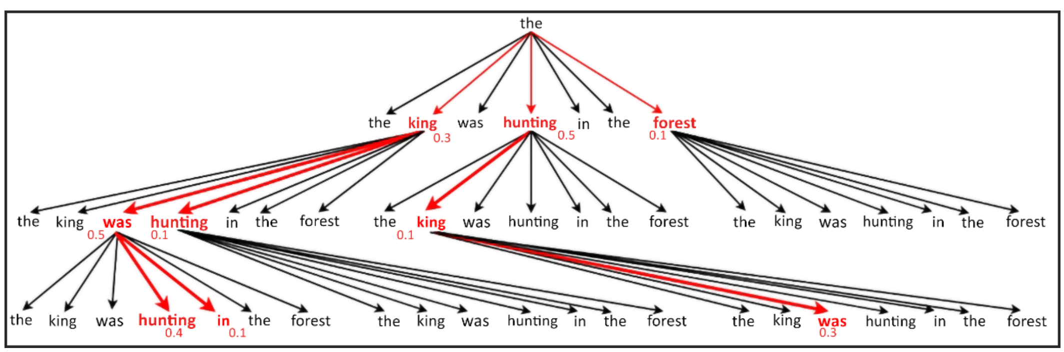


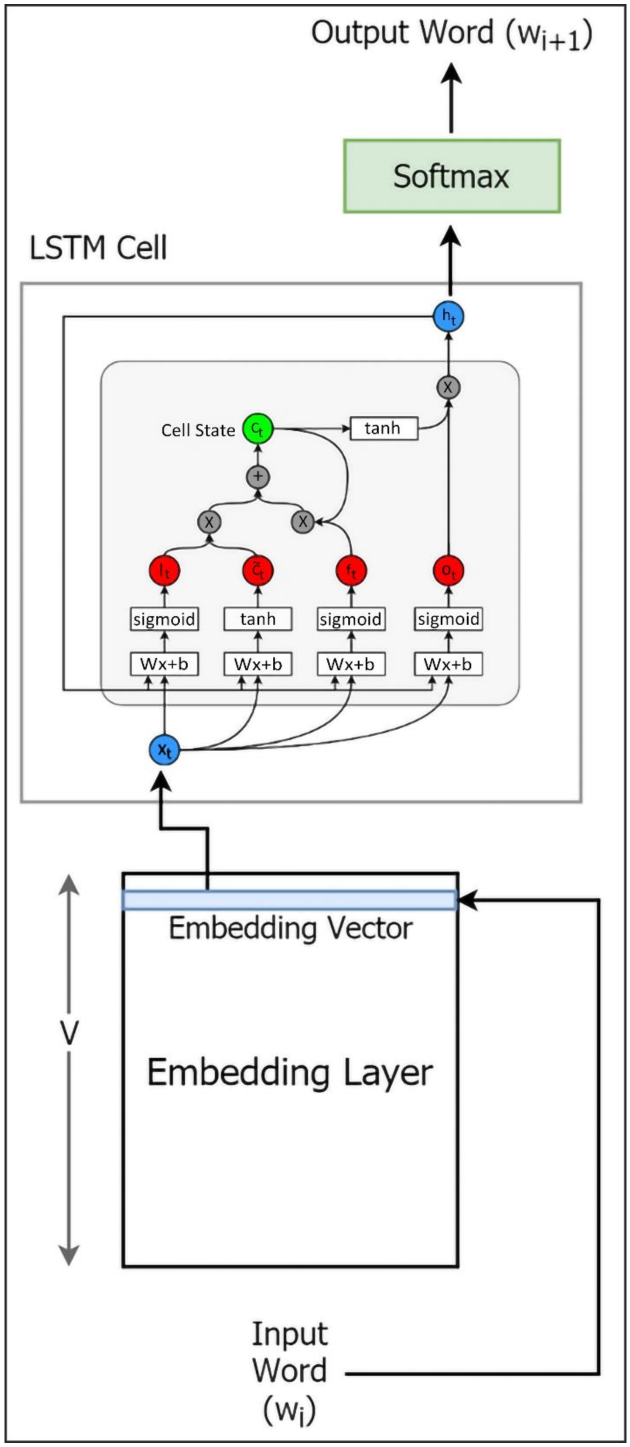
h_t
$\tilde{h}_t$
x
tanh
+
Wx+b
x
1-z
x
r_t
z_t
sigmoid
sigmoid
Wx+b
Wx+b
x_t



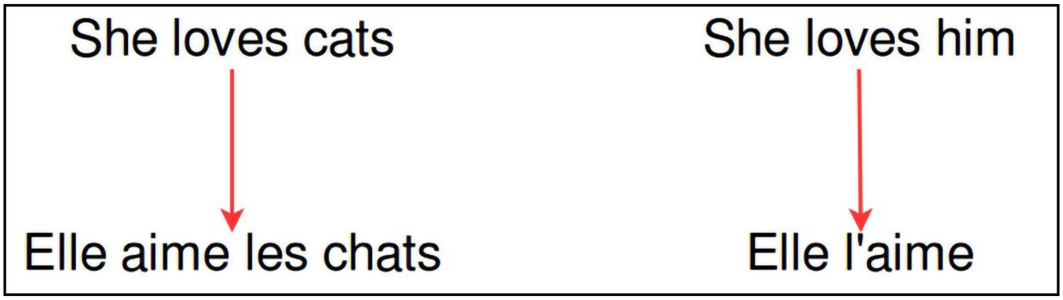
h_t
X
Cell State
c_t
tanh
+
X
X
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t

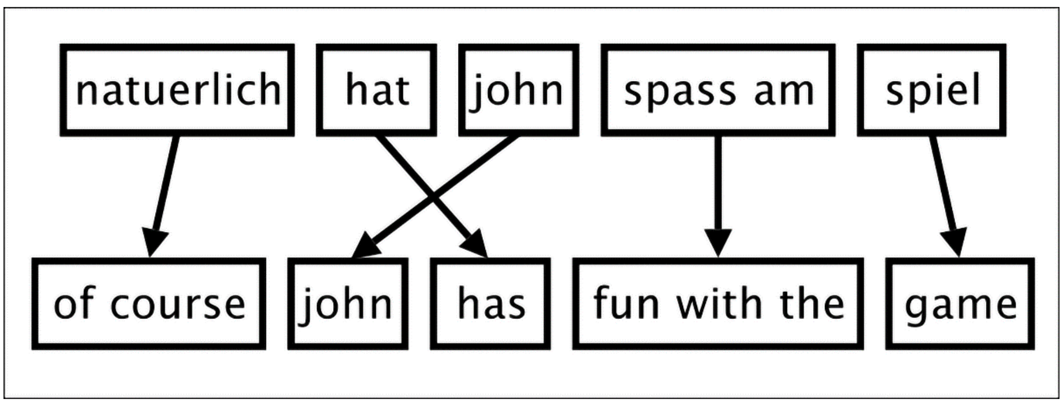

Train/Valid perplexity behavior for various sequential models
Perplexity
60
50
40
30
20
10
10
20
30
40
50
LSTM
LSTM_valid
LSTM (Peephole)
LSTM (Peephole)_valid
GRU
GRU_valid

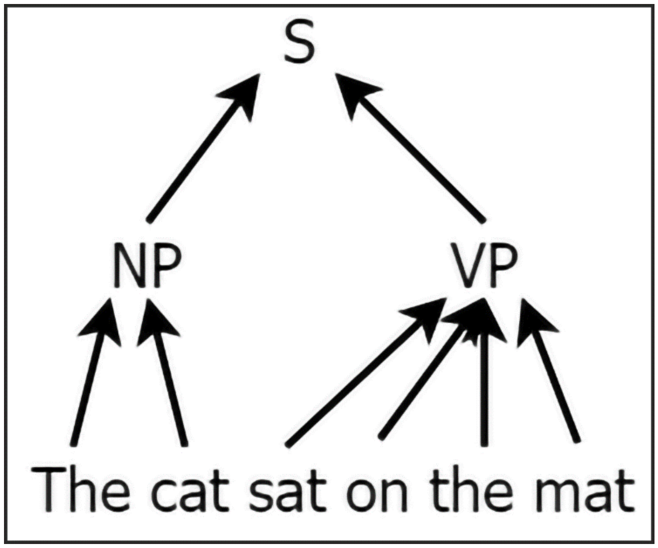

the
the king 0.3 was hunting 0.5 in the forest 0.1
the king was 0.5 hunting 0.1 in the forest
the king 0.1 was hunting in the forest
the king was hunting in the forest
the king was hunting 0.4 in 0.1 the forest
the king was hunting in the forest
the king was 0.3 hunting in the forest

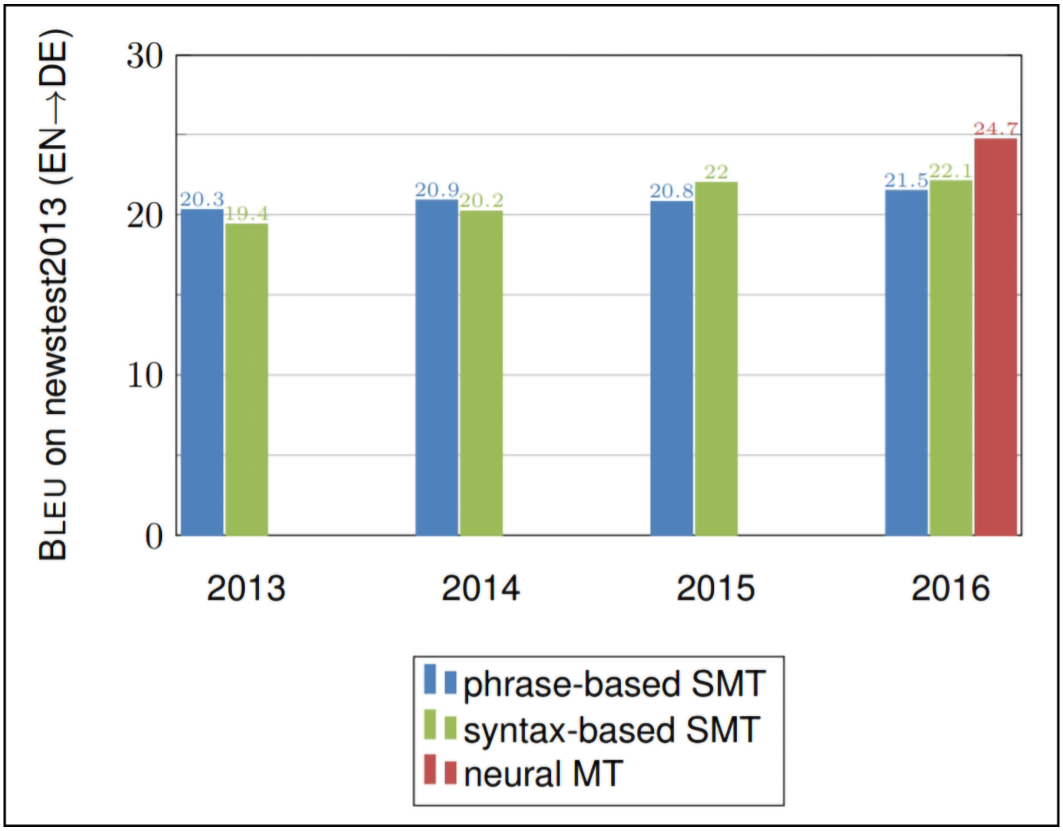

Output Word (w_{i+1})
Softmax
LSTM Cell
h_t
Cell State
c_t
tanh
i_t
$\tilde{c}_t$
f_t
o_t
sigmoid
tanh
sigmoid
sigmoid
Wx+b
Wx+b
Wx+b
Wx+b
x_t
Embedding Vector
V
Embedding Layer
Input Word (w_i)

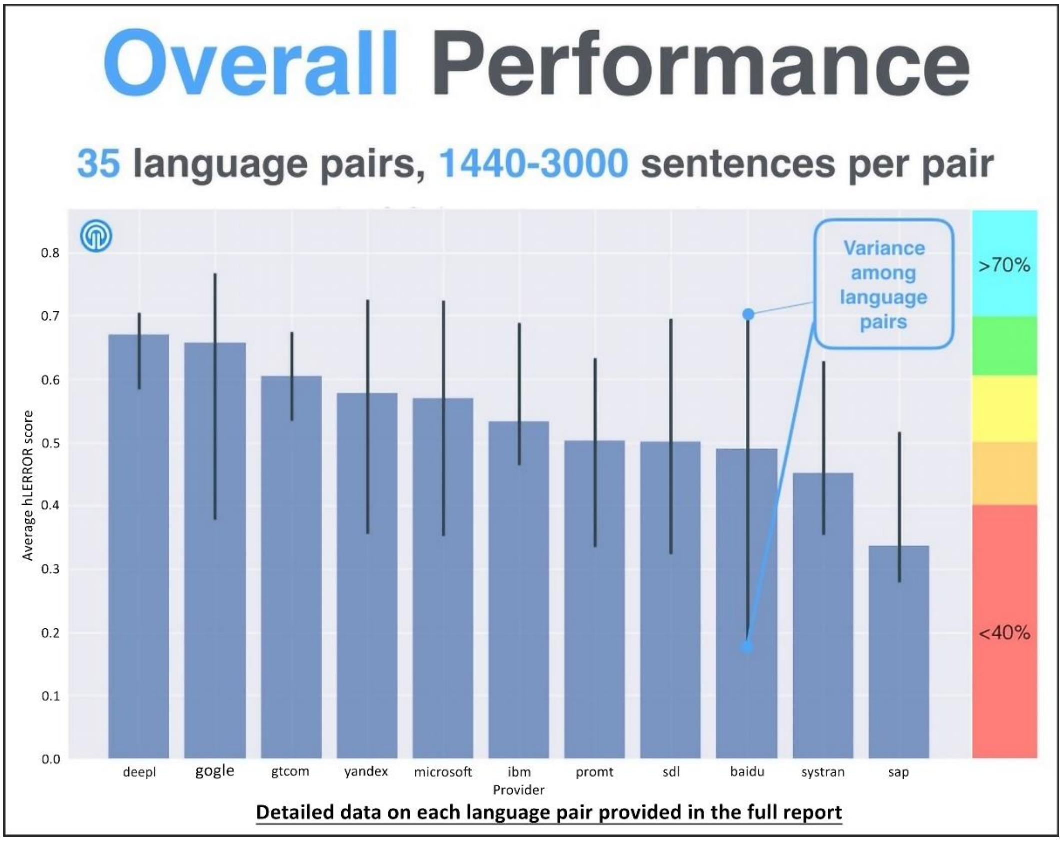
Chapter 9: Sequence-to-Sequence Learning – Neural Machine Translation

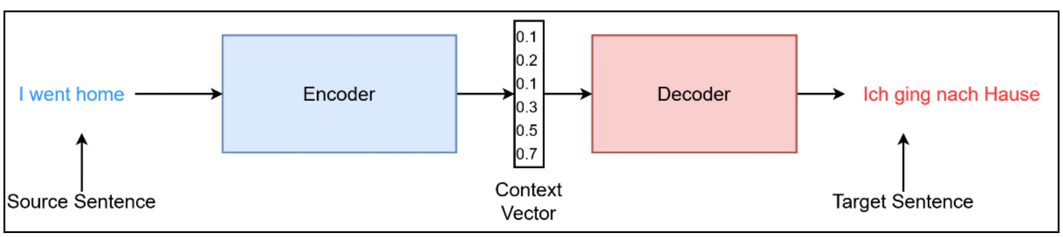


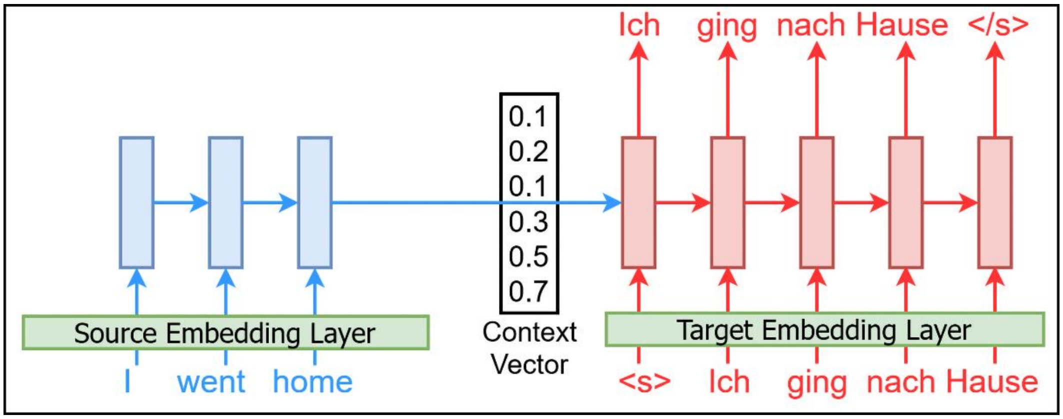


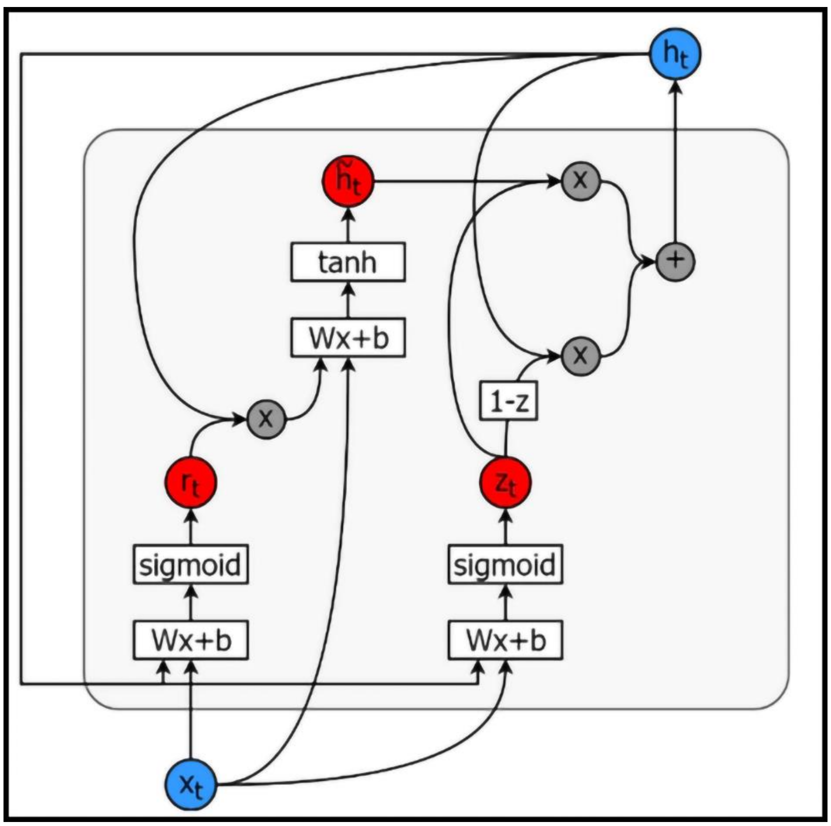

S
NP
VP
The cat sat on the mat

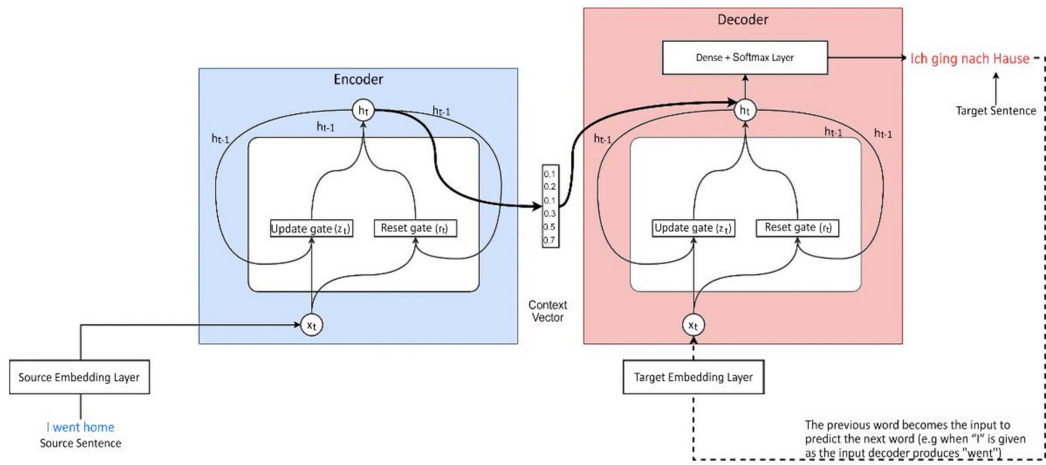

BLEU on newstest2013 (EN→DE)
30
20
10
0
20.3
19.4
20.9
20.2
20.8
22
21.5
22.1
24.7
2013
2014
2015
2016
phrase-based SMT
syntax-based SMT
neural MT

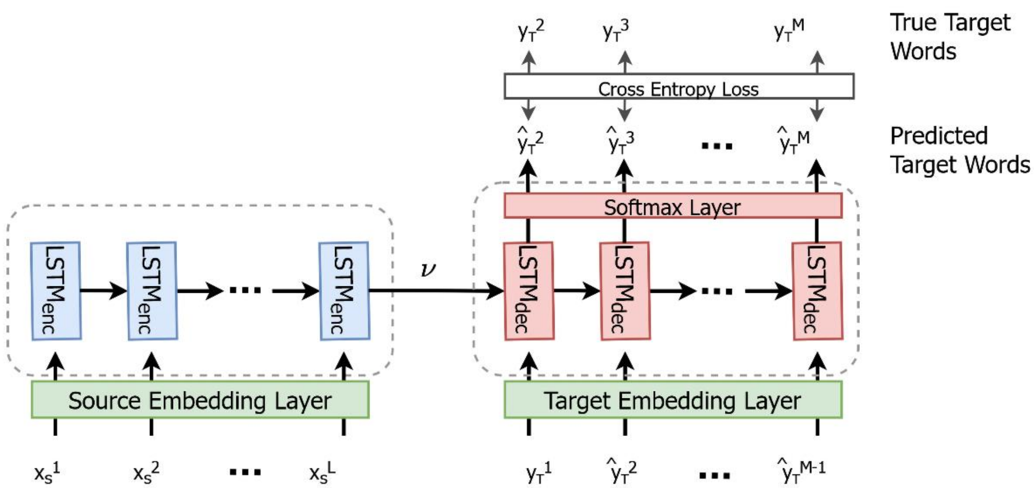

Overall Performance
35 language pairs, 1440-3000 sentences per pair
Variance among language pairs
>70%
<40%
Average hLERROR score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
deepl
gogle
gtcom
yandex
microsoft
ibm
promt
sdl
baidu
systran
sap
Provider
Detailed data on each language pair provided in the full report

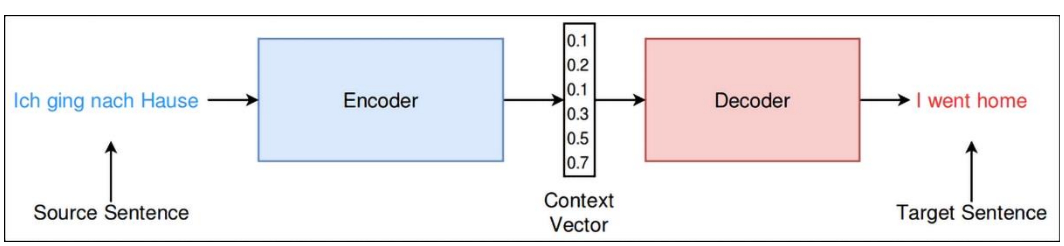

I went home
Source Sentence
Encoder
0.1
0.2
0.1
0.3
0.5
0.7
Context Vector
Decoder
Ich ging nach Hause
Target Sentence

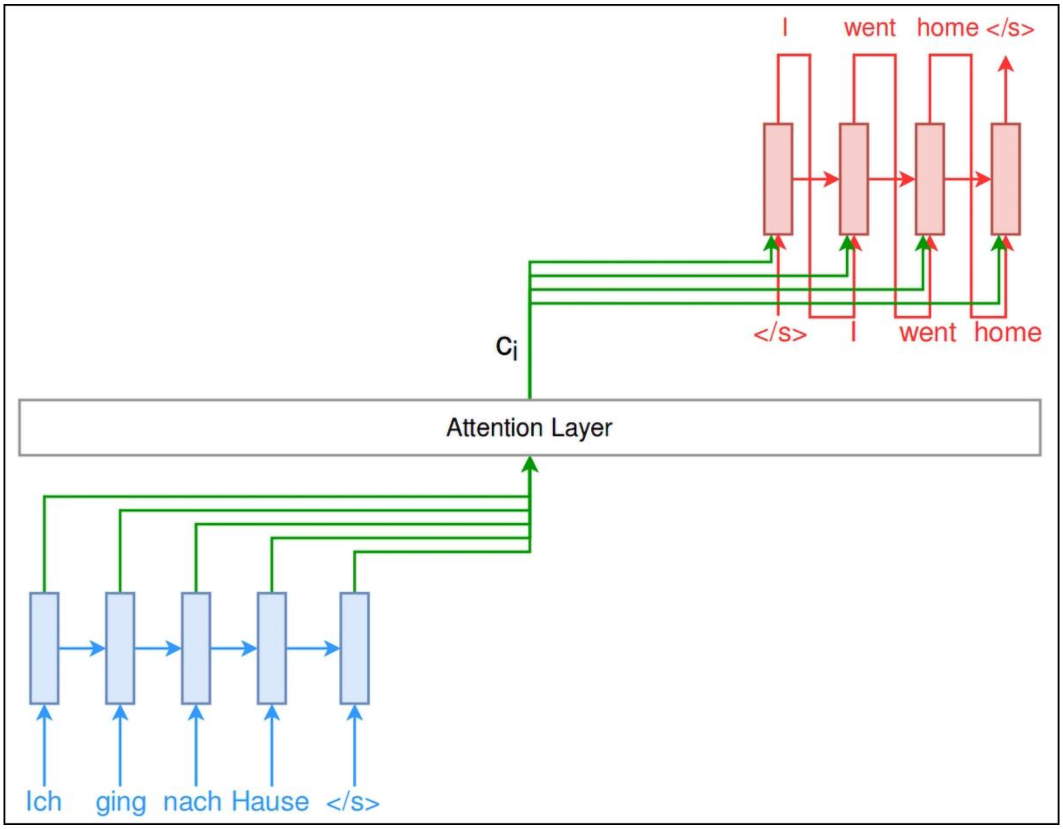

Ich
ging
nach
Hause
</s>
0.1
0.2
0.1
0.3
0.5
0.7
Source Embedding Layer
Context
Vector
Target Embedding Layer
I
went
home
<s>
Ich
ging
nach
Hause

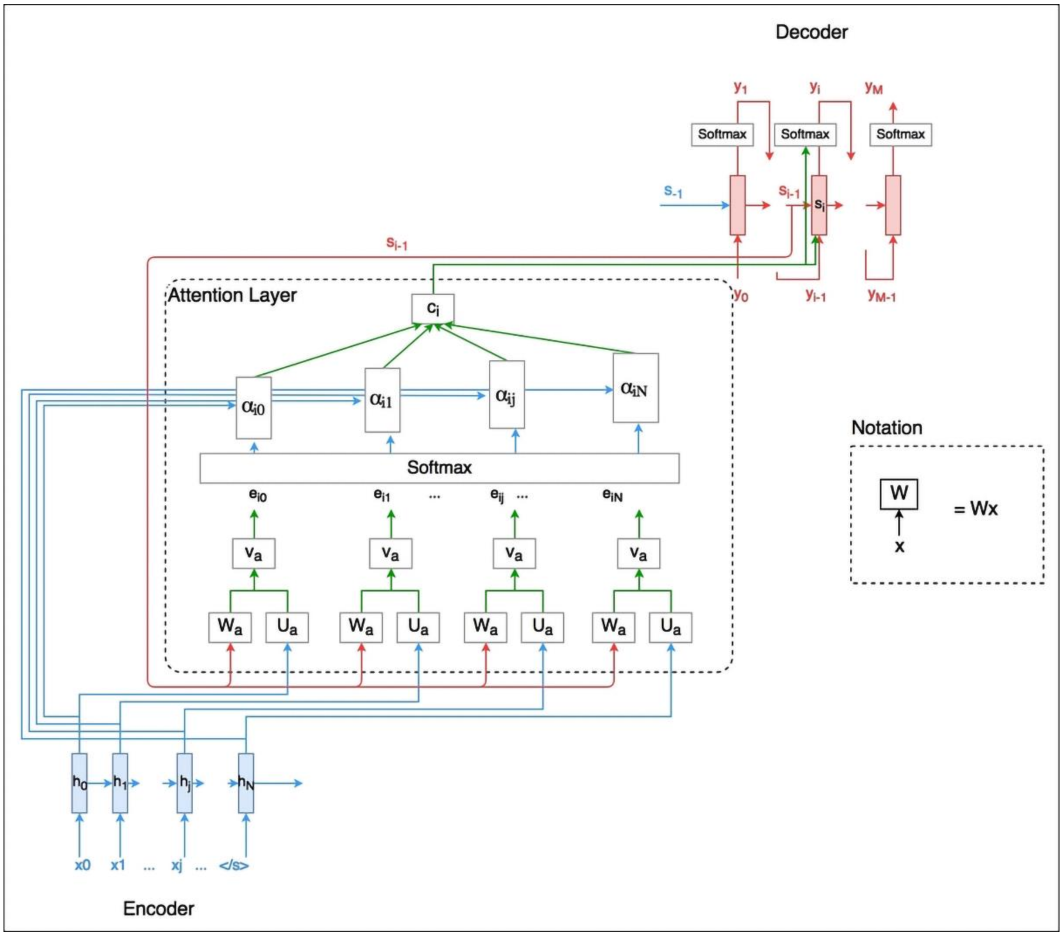

h_t
$\tilde{h}_t$
x
tanh
+
Wx+b
x
1-z
x
r_t
z_t
sigmoid
sigmoid
Wx+b
Wx+b
x_t

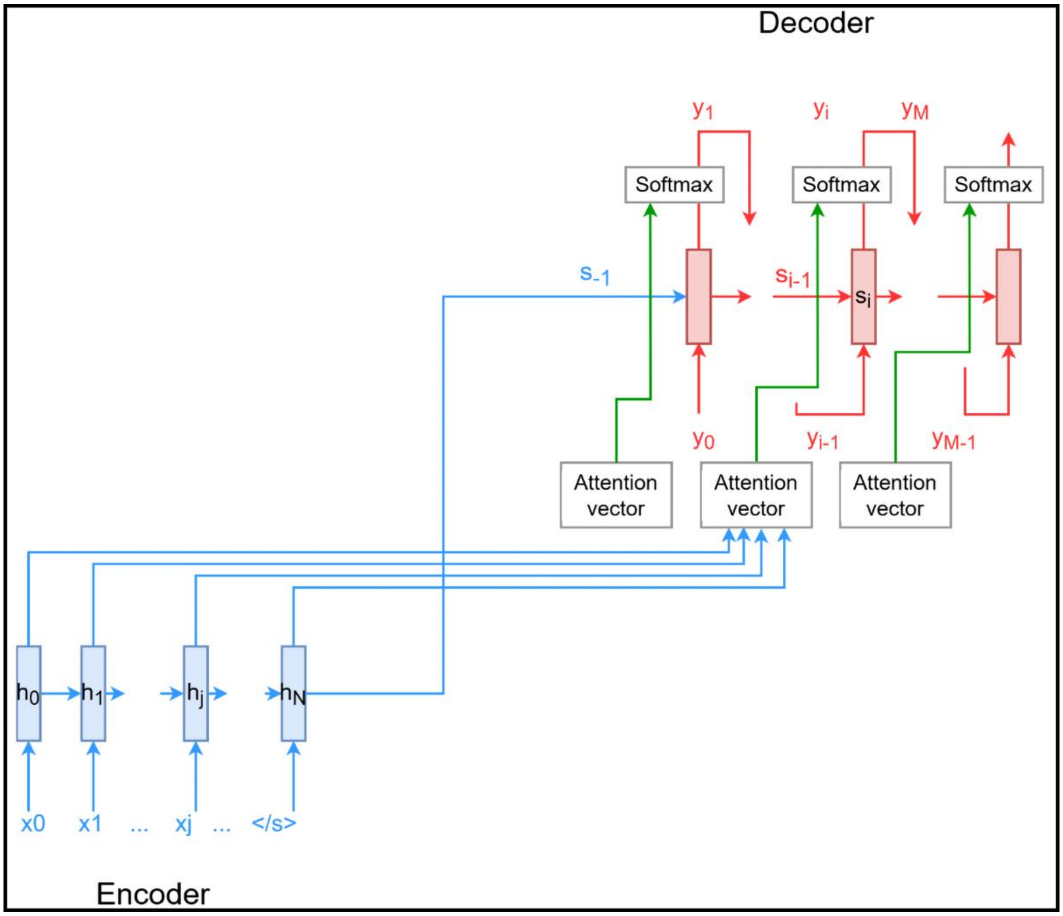

Encoder
Decoder
Dense + Softmax Layer
Ich ging nach Hause
Target Sentence
Update gate (z_t)
Reset gate (r_t)
h_t
h_{t-1}
x_t
0.1
0.2
0.1
0.3
0.5
0.7
Context Vector
Source Embedding Layer
Target Embedding Layer
I went home
Source Sentence
The previous word becomes the input to predict the next word (e.g when "I" is given as the input decoder produces "went")

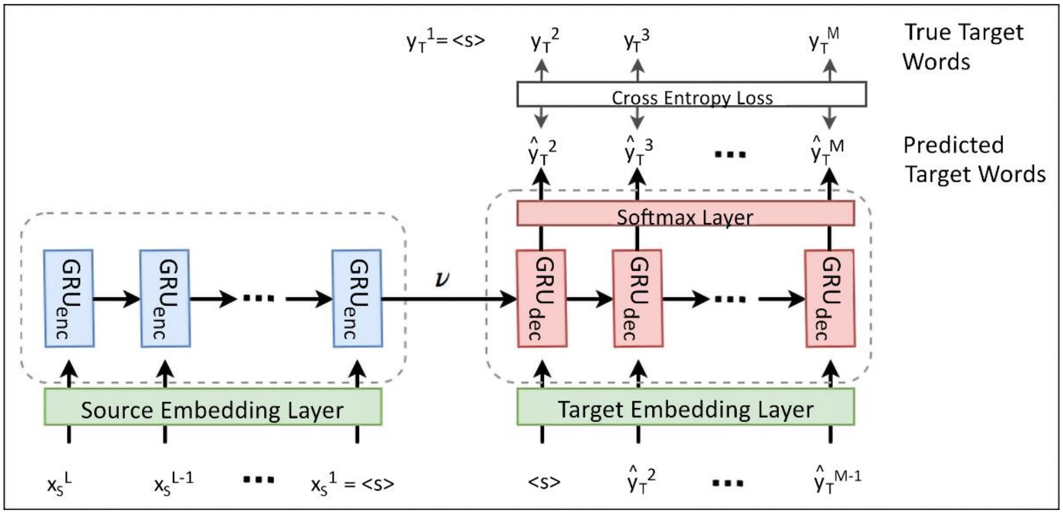

True Target Words
Predicted Target Words
y_T^2 y_T^3 y_T^M
Cross Entropy Loss
$\hat{y}_T^2$ $\hat{y}_T^3$ ... $\hat{y}_T^M$
Softmax Layer
$LSTM_{enc}$
$LSTM_{dec}$
ν
Source Embedding Layer
Target Embedding Layer
x_S^1 x_S^2 ... x_S^L
y_T^1 $\hat{y}_T^2$ ... $\hat{y}_T^{M-1}$


Ich ging nach Hause
Encoder
0.1
0.2
0.1
0.3
0.5
0.7
Decoder
I went home
Source Sentence
Context Vector
Target Sentence

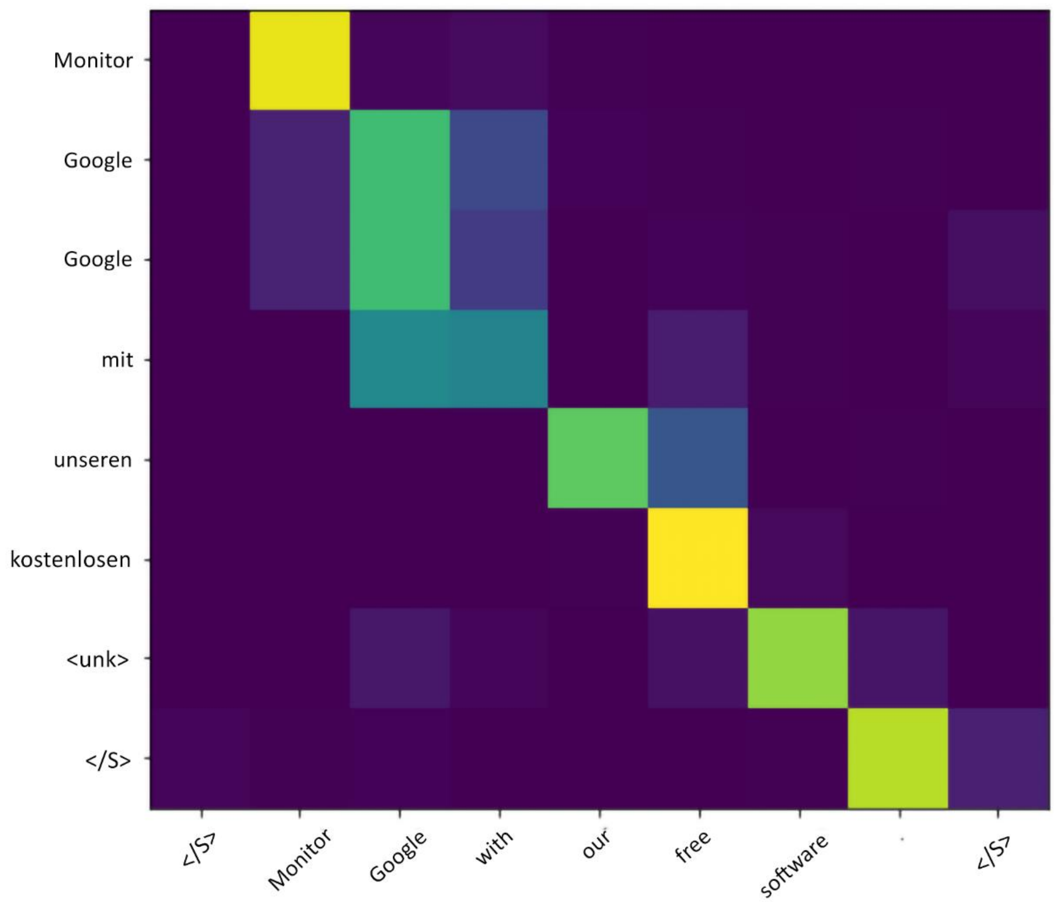

I went home </s>
</s> I went home
c_i
Attention Layer
Ich ging nach Hause </s>


Decoder
y_1
y_i
y_M
Softmax
Softmax
Softmax
s_{-1}
s_{i-1}
s_i
s_{i-1}
y_0
y_{i-1}
y_{M-1}
Attention Layer
c_i
α_{i0}
α_{i1}
α_{ij}
α_{iN}
Softmax
e_{i0}
e_{i1}
...
e_{ij}
...
e_{iN}
v_a
v_a
v_a
v_a
W_a
U_a
W_a
U_a
W_a
U_a
W_a
U_a
Notation
W
x
= Wx
h_0
h_1
h_j
h_N
x0
x1
...
xj
...
</s>
Encoder

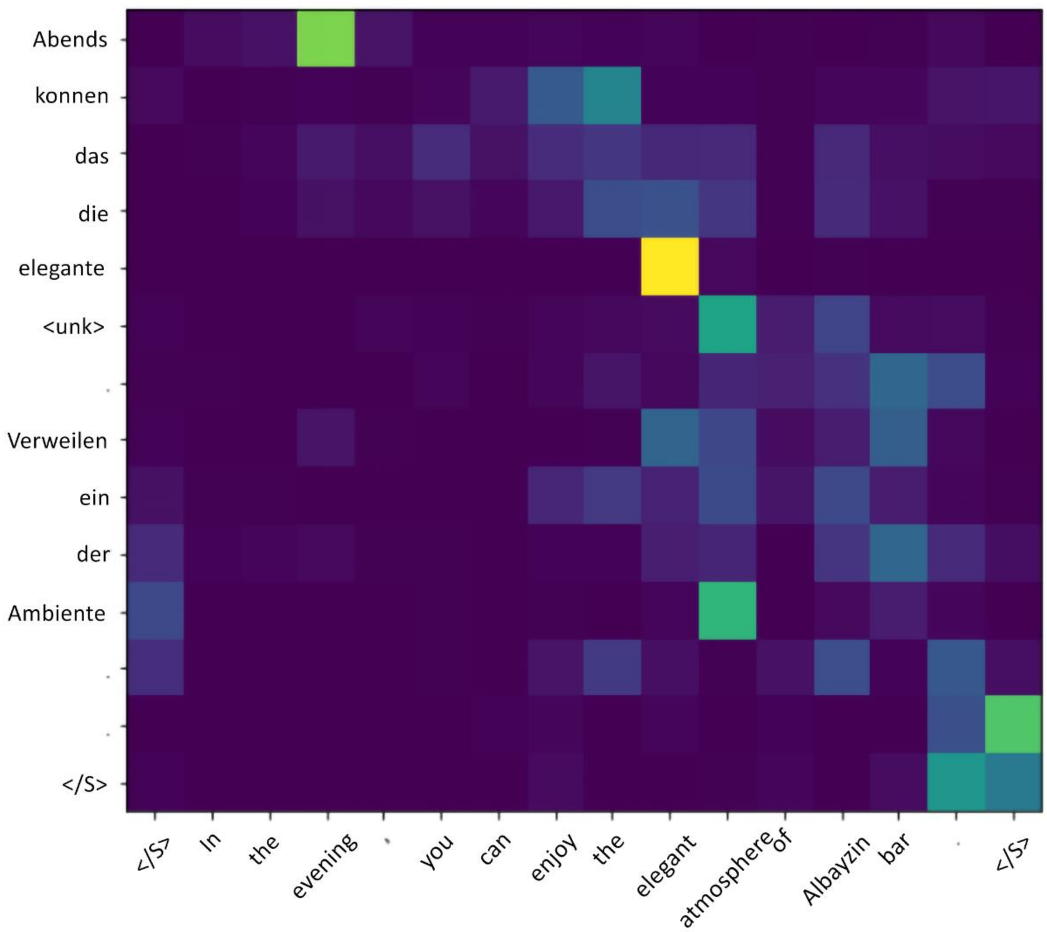

Decoder
y1
yi
yM
Softmax
Softmax
Softmax
s-1
si-1
si
y0
yi-1
yM-1
Attention vector
Attention vector
Attention vector
h0
h1
hj
hN
x0
x1
...
xj
...
</s>
Encoder

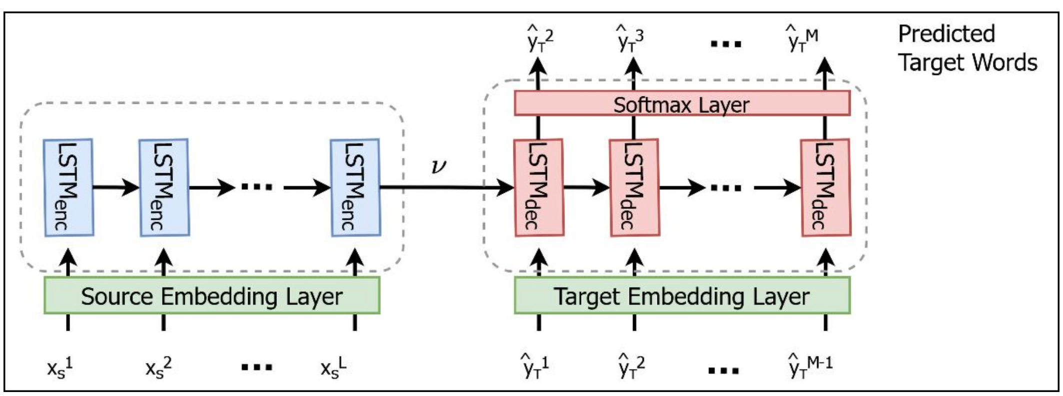


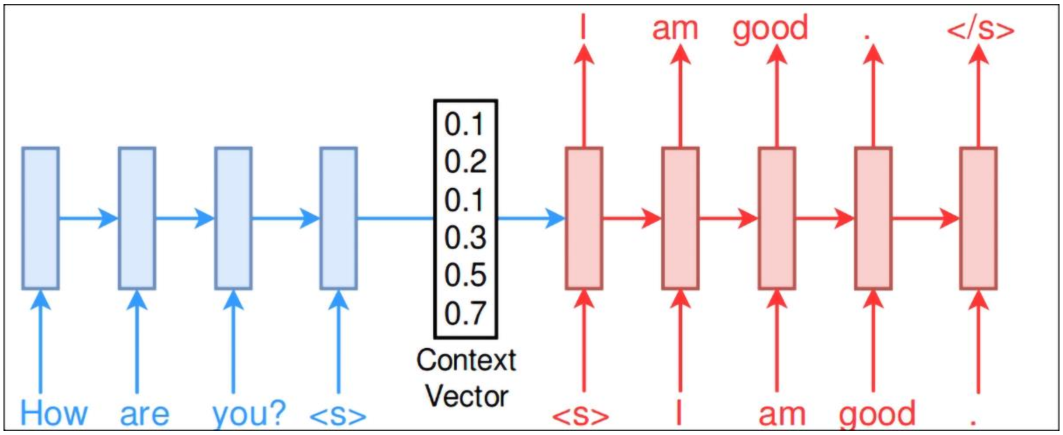
y_T^1= <s>
y_T^2
y_T^3
y_T^M
True Target Words
Cross Entropy Loss
$\hat{y}_T^2$
$\hat{y}_T^3$
$\hat{y}_T^M$
Predicted Target Words
Softmax Layer
GRU_{enc}
GRU_{enc}
GRU_{enc}
ν
GRU_{dec}
GRU_{dec}
GRU_{dec}
Source Embedding Layer
Target Embedding Layer
x_S^L
x_S^{L-1}
x_S^1 = <s>
<s>
$\hat{y}_T^2$
$\hat{y}_T^{M-1}$

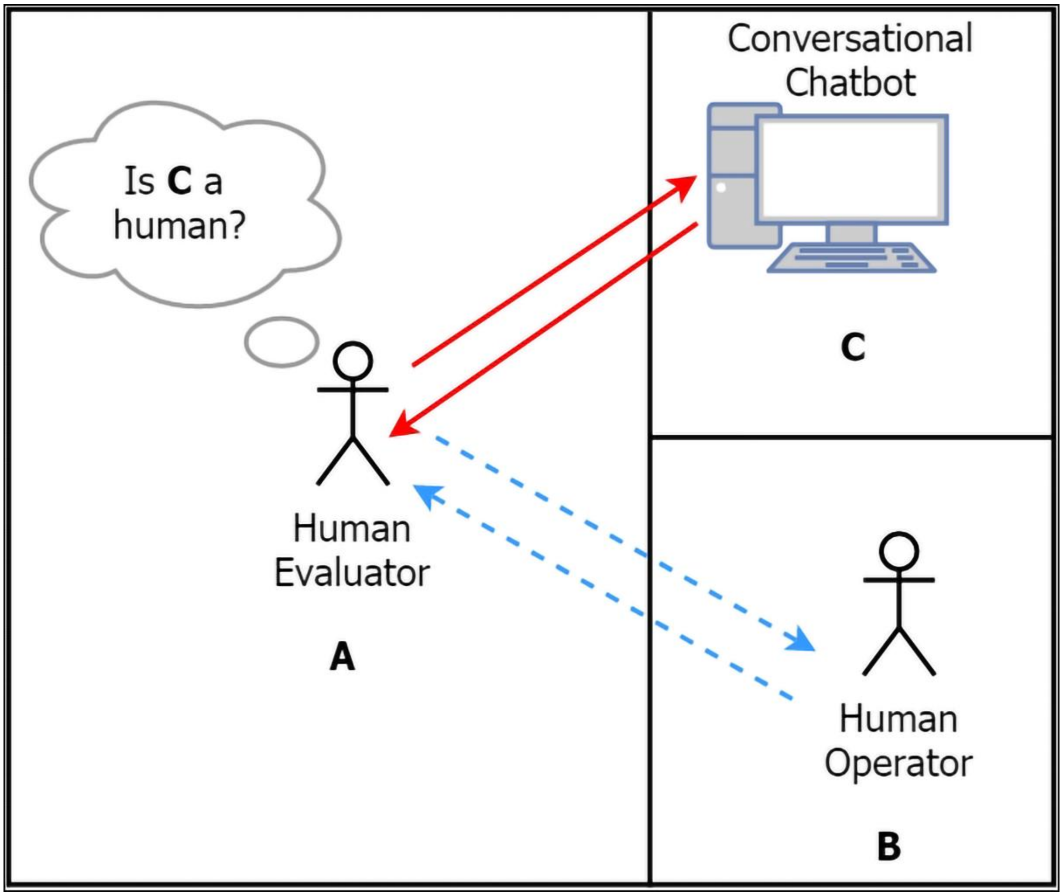


Monitor
Google
Google
mit
unseren
kostenlosen
<unk>
</S>
</S>
Monitor
Google
with
our
free
software
.
</S>

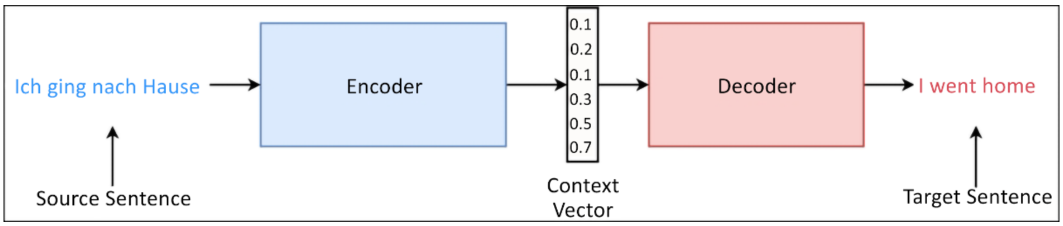

Abends
konnen
das
die
elegante
<unk>
Verweilen
ein
der
Ambiente
</S>
</S>
In
the
evening
you
can
enjoy
the
elegant
atmosphere
of
Albayzin
bar
</S>

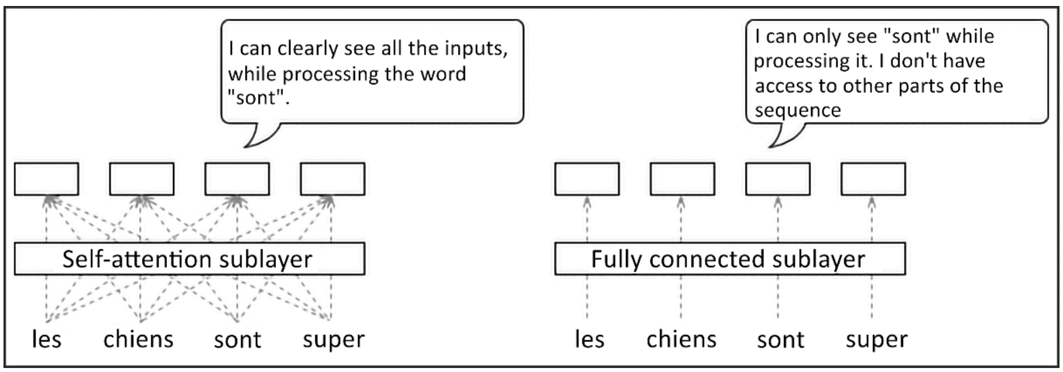

$\hat{y}_T^2$
$\hat{y}_T^3$
...
$\hat{y}_T^M$
Predicted Target Words
Softmax Layer
LSTM_{enc}
ν
LSTM_{dec}
Source Embedding Layer
Target Embedding Layer
x_S^1
x_S^2
...
x_S^L
$\hat{y}_T^1$
$\hat{y}_T^2$
...
$\hat{y}_T^{M-1}$

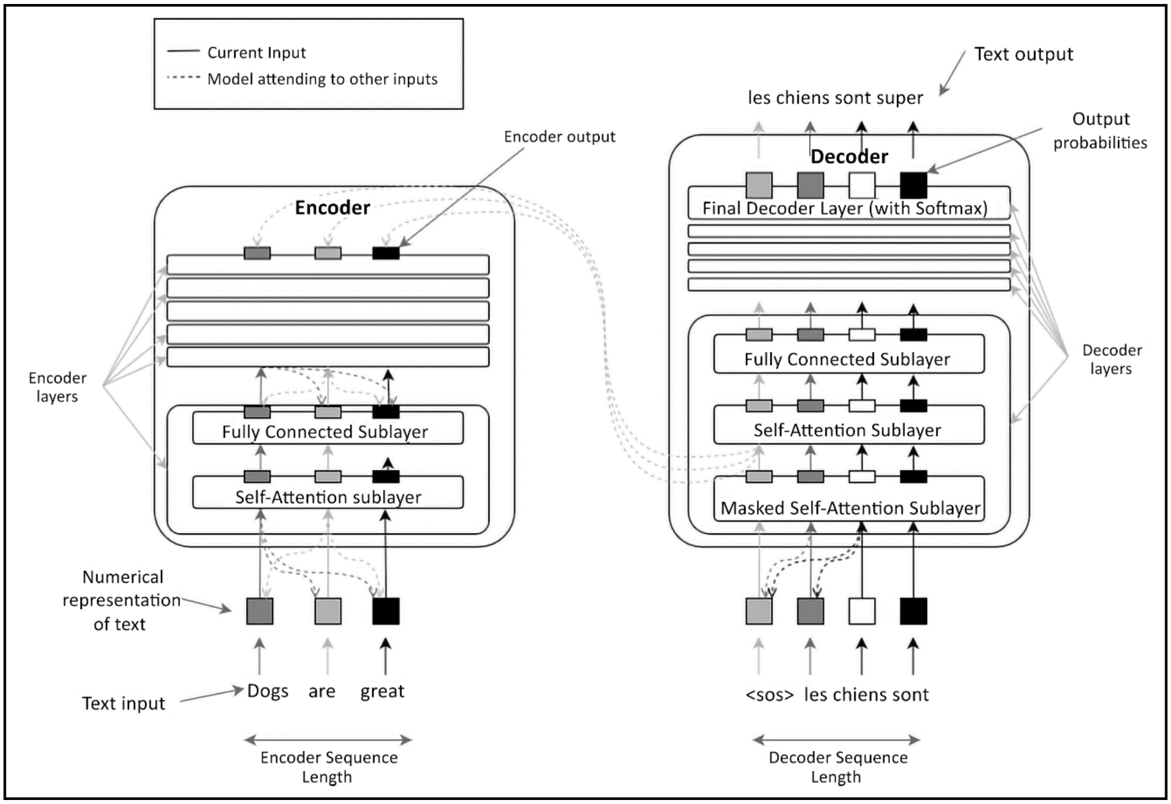

I
am
good
.
</s>
0.1
0.2
0.1
0.3
0.5
0.7
Context
Vector
How
are
you?
<s>
<s>
I
am
good
.


Conversational Chatbot
Is **C** a human?
C
Human Evaluator
A
Human Operator
B

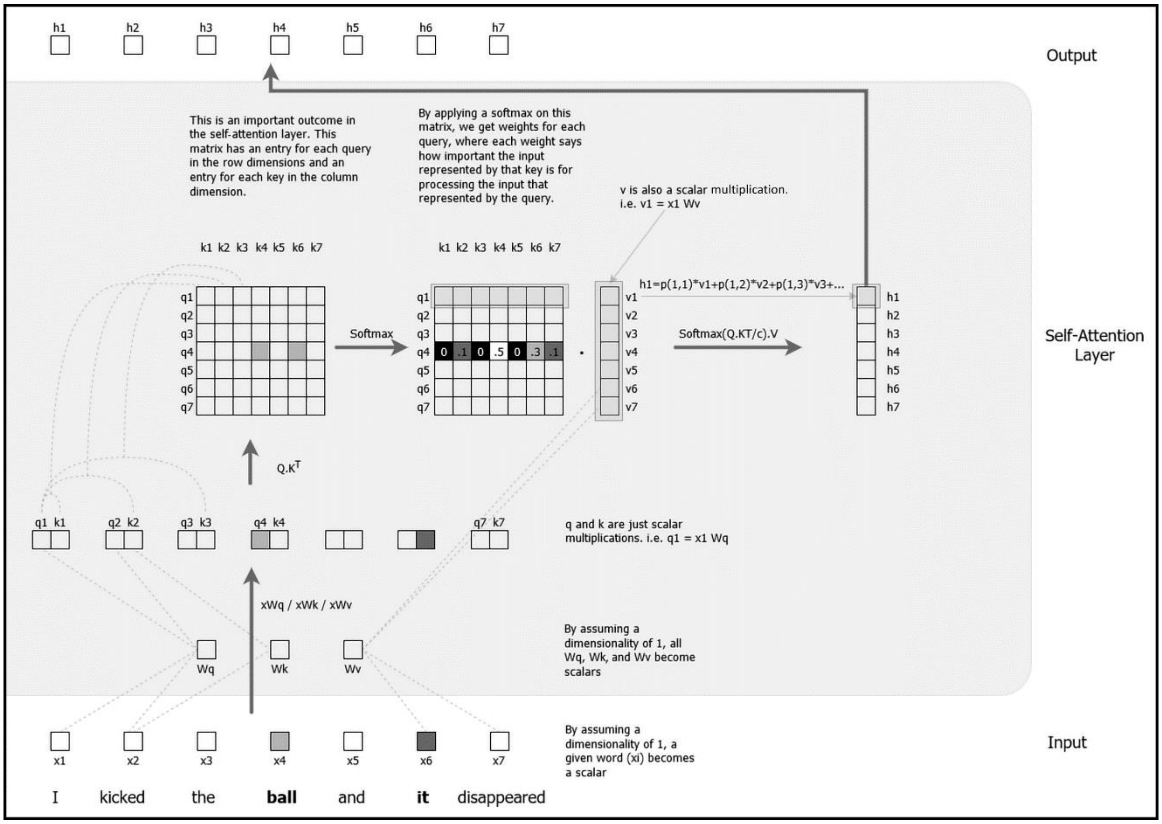
Chapter 10: Transformers



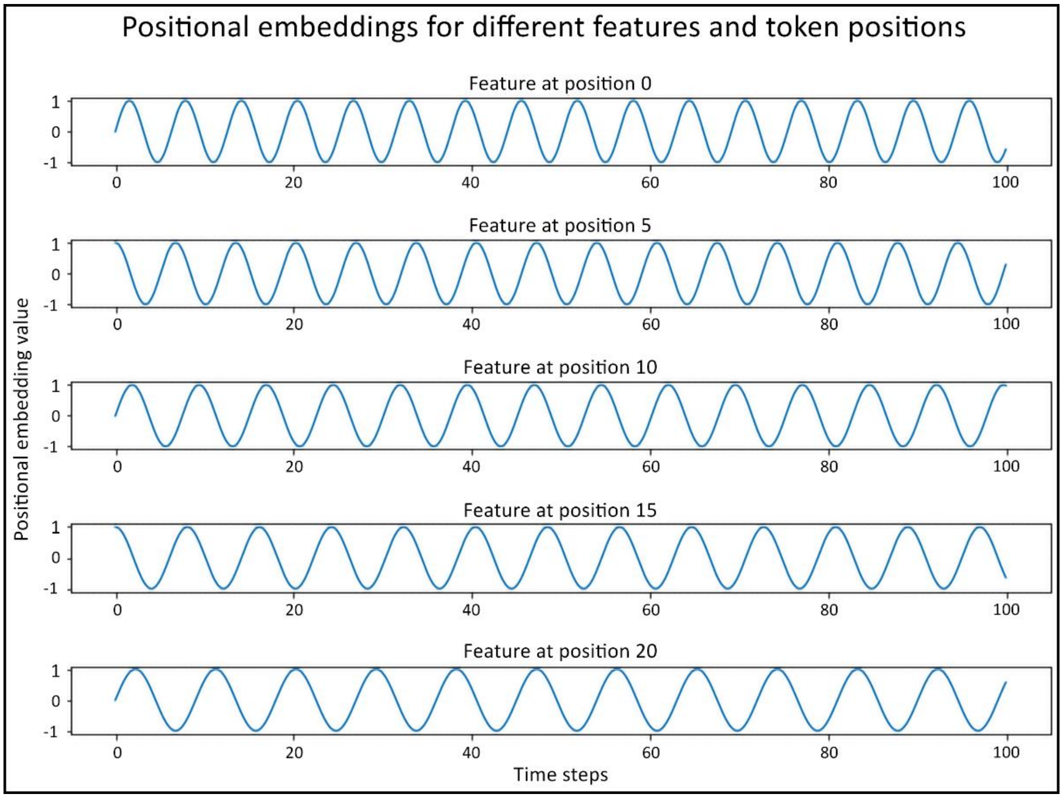


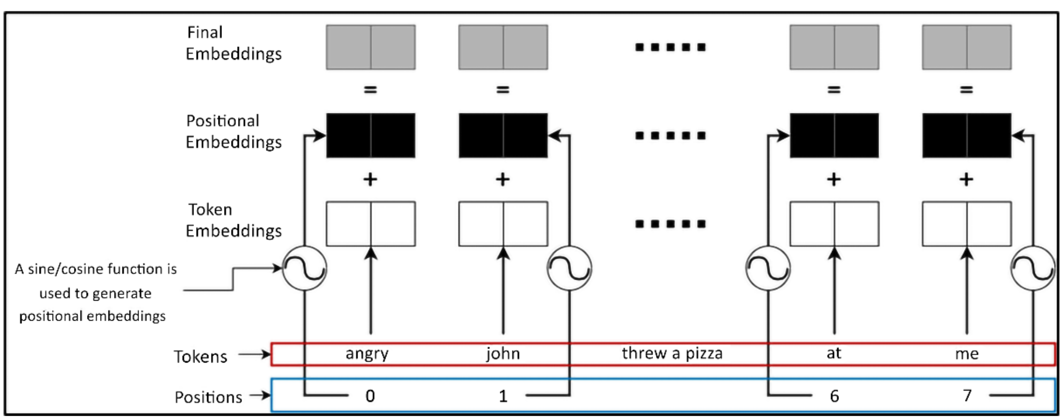


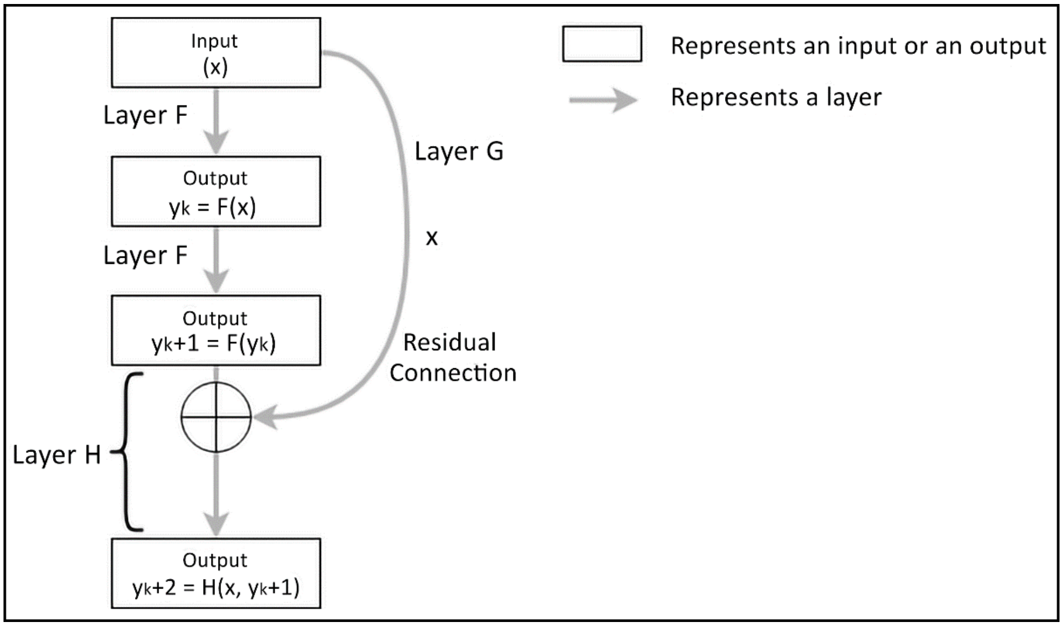
Current Input
Model attending to other inputs
Text output
les chiens sont super
Output probabilities
Encoder output
Decoder
Encoder
Final Decoder Layer (with Softmax)
Fully Connected Sublayer
Self-Attention Sublayer
Masked Self-Attention Sublayer
Decoder layers
Encoder layers
Fully Connected Sublayer
Self-Attention sublayer
Numerical representation of text
Text input
Dogs are great
<sos> les chiens sont
Encoder Sequence Length
Decoder Sequence Length

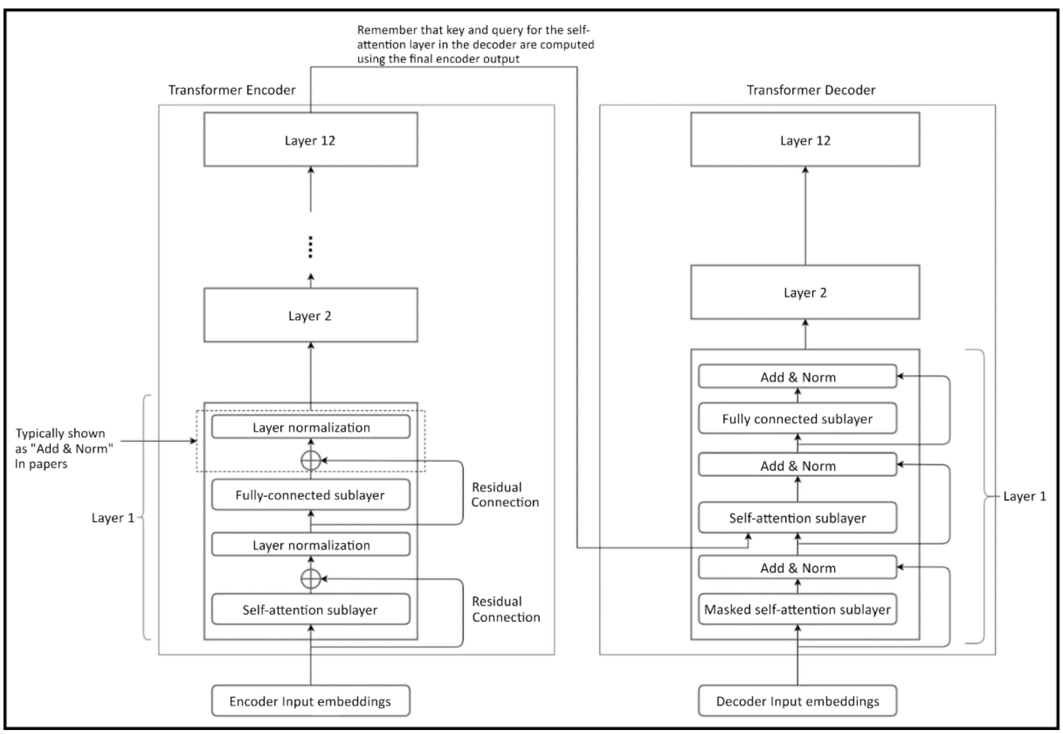


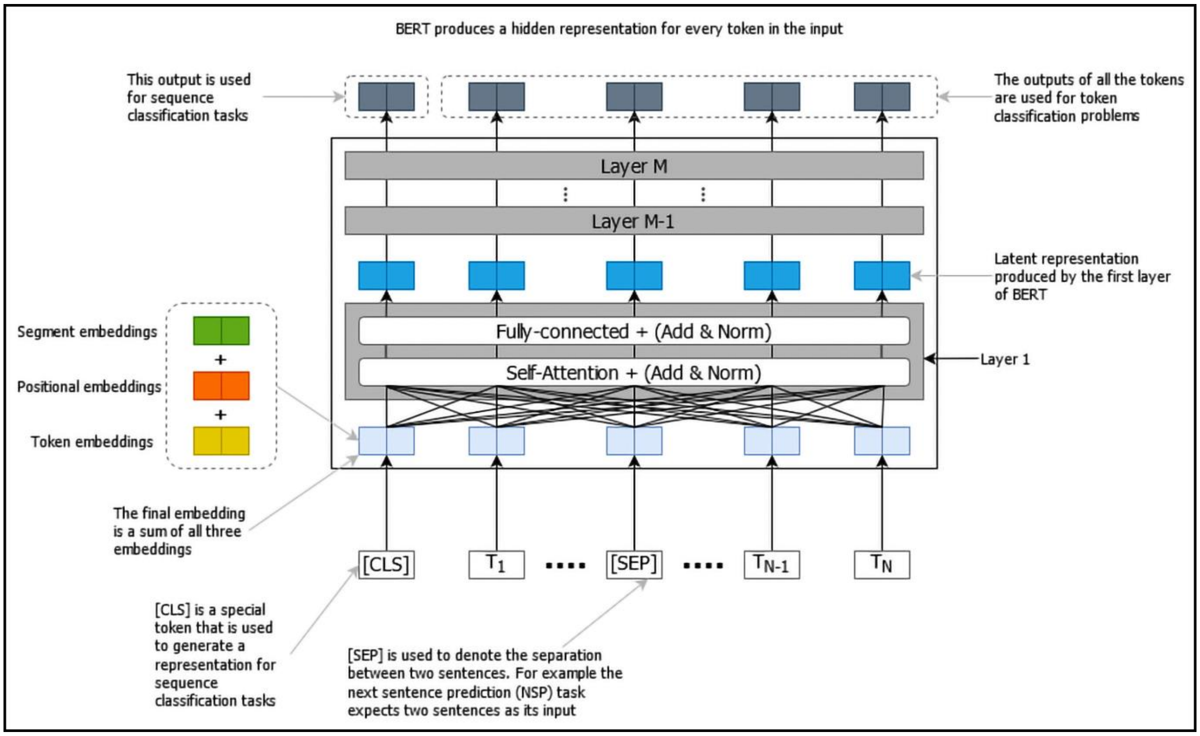
h1 h2 h3 h4 h5 h6 h7
Output
This is an important outcome in the self-attention layer. This matrix has an entry for each query in the row dimensions and an entry for each key in the column dimension.
By applying a softmax on this matrix, we get weights for each query, where each weight says how important the input represented by that key is for processing the input that represented by the query.
v is also a scalar multiplication. i.e. v1 = x1 Wv
k1 k2 k3 k4 k5 k6 k7
q1 q2 q3 q4 q5 q6 q7
Softmax
0 .1 0 .5 0 .3 .1
v1 v2 v3 v4 v5 v6 v7
h1=p(1,1)*v1+p(1,2)*v2+p(1,3)*v3+...
Softmax(Q.KT/c).V
Self-Attention Layer
Q.K^T
q1 k1 q2 k2 q3 k3 q4 k4 q7 k7
q and k are just scalar multiplications. i.e. q1 = x1 Wq
xWq / xWk / xWv
Wq Wk Wv
By assuming a dimensionality of 1, all Wq, Wk, and Wv become scalars
x1 x2 x3 x4 x5 x6 x7
By assuming a dimensionality of 1, a given word (xi) becomes a scalar
Input
I kicked the ball and it disappeared

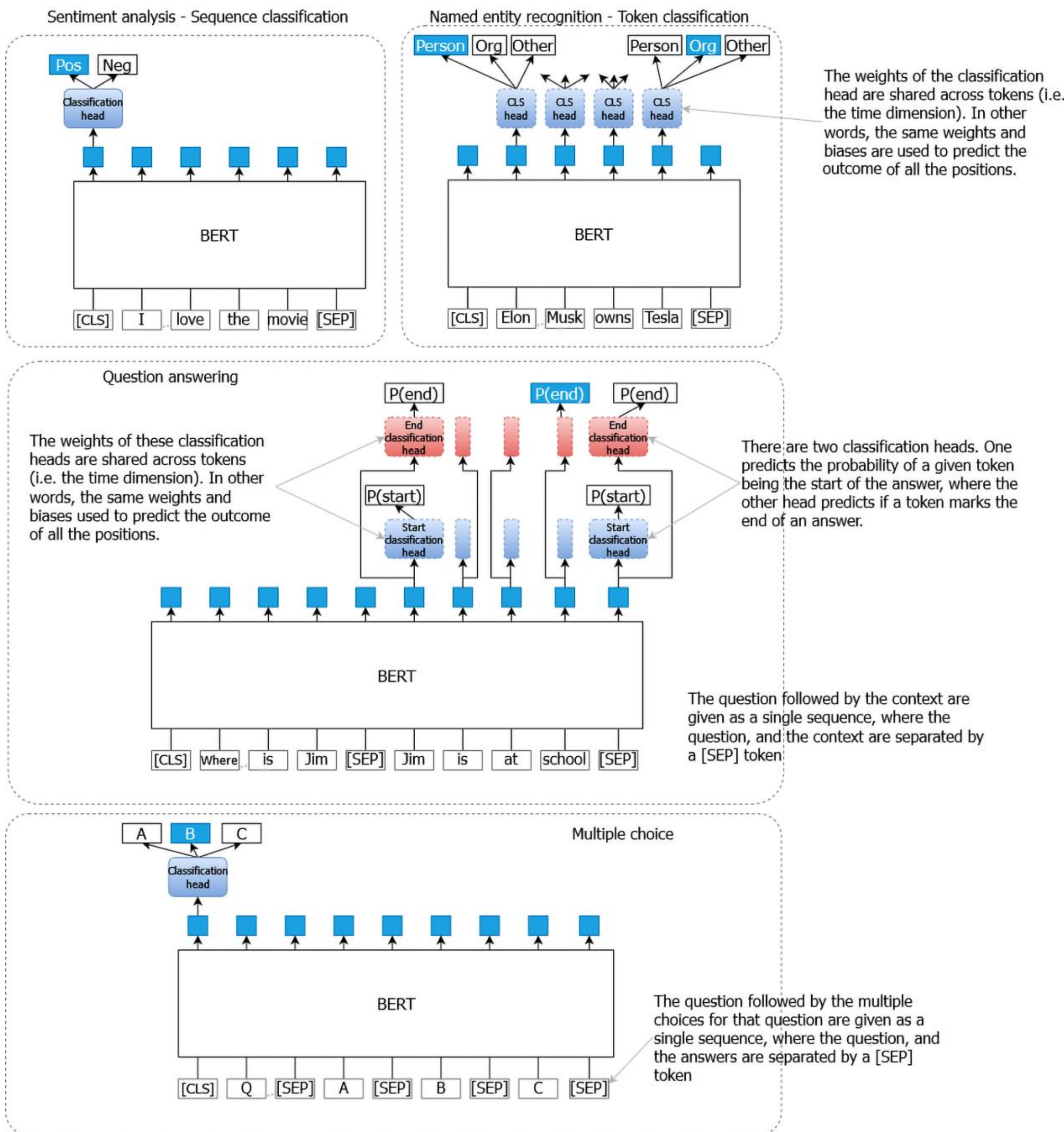

Positional embeddings for different features and token positions
Feature at position 0
Feature at position 5
Feature at position 10
Feature at position 15
Feature at position 20
Positional embedding value
Time steps
1
0
-1
0
20
40
60
80
100


Final Embeddings
Positional Embeddings
Token Embeddings
A sine/cosine function is used to generate positional embeddings
Tokens
angry
john
threw a pizza
at
me
Positions
0
1
6
7

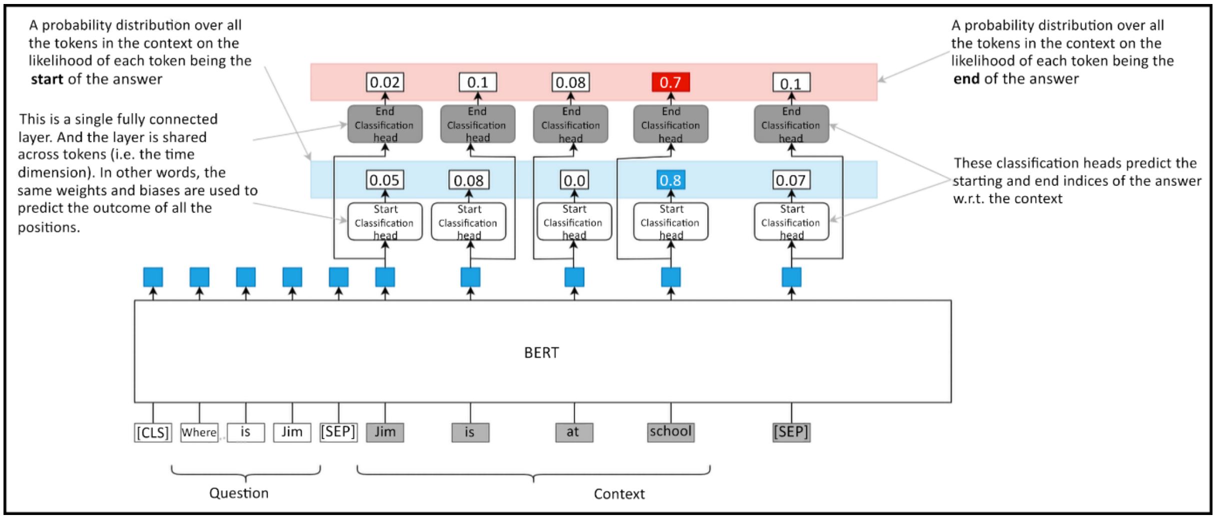

Input
(x)
Layer F
Output
yk = F(x)
Layer F
Output
yk+1 = F(yk)
Layer G
x
Residual
Connection
Layer H
Output
yk+2 = H(x, yk+1)
Represents an input or an output
Represents a layer


Remember that key and query for the self-attention layer in the decoder are computed using the final encoder output
Transformer Encoder
Layer 12
Layer 2
Typically shown as "Add & Norm" In papers
Layer 1
Layer normalization
Fully-connected sublayer
Layer normalization
Self-attention sublayer
Residual Connection
Residual Connection
Encoder Input embeddings
Transformer Decoder
Layer 12
Layer 2
Add & Norm
Fully connected sublayer
Add & Norm
Self-attention sublayer
Add & Norm
Masked self-attention sublayer
Layer 1
Decoder Input embeddings

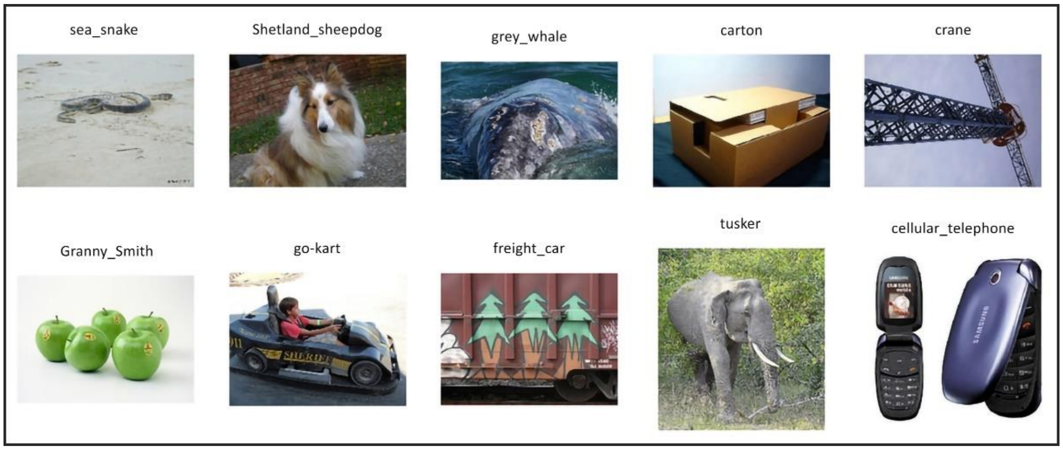

BERT produces a hidden representation for every token in the input
This output is used for sequence classification tasks
The outputs of all the tokens are used for token classification problems
Layer M
Layer M-1
Latent representation produced by the first layer of BERT
Segment embeddings
+
Positional embeddings
+
Token embeddings
Fully-connected + (Add & Norm)
Self-Attention + (Add & Norm)
Layer 1
The final embedding is a sum of all three embeddings
[CLS]
T_1
[SEP]
T_{N-1}
T_N
[CLS] is a special token that is used to generate a representation for sequence classification tasks
[SEP] is used to denote the separation between two sentences. For example the next sentence prediction (NSP) task expects two sentences as its input



Sentiment analysis - Sequence classification
Pos
Neg
Classification head
BERT
[CLS]
I
love
the
movie
[SEP]
Named entity recognition - Token classification
Person
Org
Other
Person
Org
Other
CLS head
CLS head
CLS head
CLS head
The weights of the classification head are shared across tokens (i.e. the time dimension). In other words, the same weights and biases are used to predict the outcome of all the positions.
BERT
[CLS]
Elon
Musk
owns
Tesla
[SEP]
Question answering
P(end)
P(end)
P(end)
End classification head
End classification head
The weights of these classification heads are shared across tokens (i.e. the time dimension). In other words, the same weights and biases used to predict the outcome of all the positions.
P(start)
P(start)
Start classification head
Start classification head
There are two classification heads. One predicts the probability of a given token being the start of the answer, where the other head predicts if a token marks the end of an answer.
BERT
The question followed by the context are given as a single sequence, where the question, and the context are separated by a [SEP] token
[CLS]
Where
is
Jim
[SEP]
Jim
is
at
school
[SEP]
Multiple choice
A
B
C
Classification head
BERT
The question followed by the multiple choices for that question are given as a single sequence, where the question, and the answers are separated by a [SEP] token
[CLS]
Q
[SEP]
A
[SEP]
B
[SEP]
C
[SEP]


A probability distribution over all the tokens in the context on the likelihood of each token being the **start** of the answer
A probability distribution over all the tokens in the context on the likelihood of each token being the **end** of the answer
This is a single fully connected layer. And the layer is shared across tokens (i.e. the time dimension). In other words, the same weights and biases are used to predict the outcome of all the positions.
These classification heads predict the starting and end indices of the answer w.r.t. the context
0.02
0.1
0.08
0.7
0.1
End Classification head
0.05
0.08
0.0
0.8
0.07
Start Classification head
BERT
[CLS]
Where
is
Jim
[SEP]
Jim
is
at
school
[SEP]
Question
Context

Chapter 11: Image Captioning with Transformers



A woman stands in the dining area at the table.

A room with chairs, a table, and a woman in it.

A woman standing in a kitchen by a window

A person standing at a table in a room.

A living area with a television and a table

A big burly grizzly bear is show with grass in the background.

The large brown bear has a black nose.

Closeup of a brown bear sitting in a grassy area.

A large bear that is sitting on grass.

A close up picture of a brown bear's face.

Bedroom scene with a bookcase, blue comforter and window.

A bedroom with a bookshelf full of books.

This room has a bed with blue sheets and a large bookcase

A bed and a mirror in a small room.

a bed room with a neatly made bed a window and a book shelf

A stop sign is mounted upside-down on it's post.

A stop sign that is hanging upside down.

An upside down stop sign by the road.

a stop sign put upside down on a metal pole

A stop sign installed upside down on a street corner

Three teddy bears, each a different color, snuggling together.

Three stuffed animals are sitting on a bed.

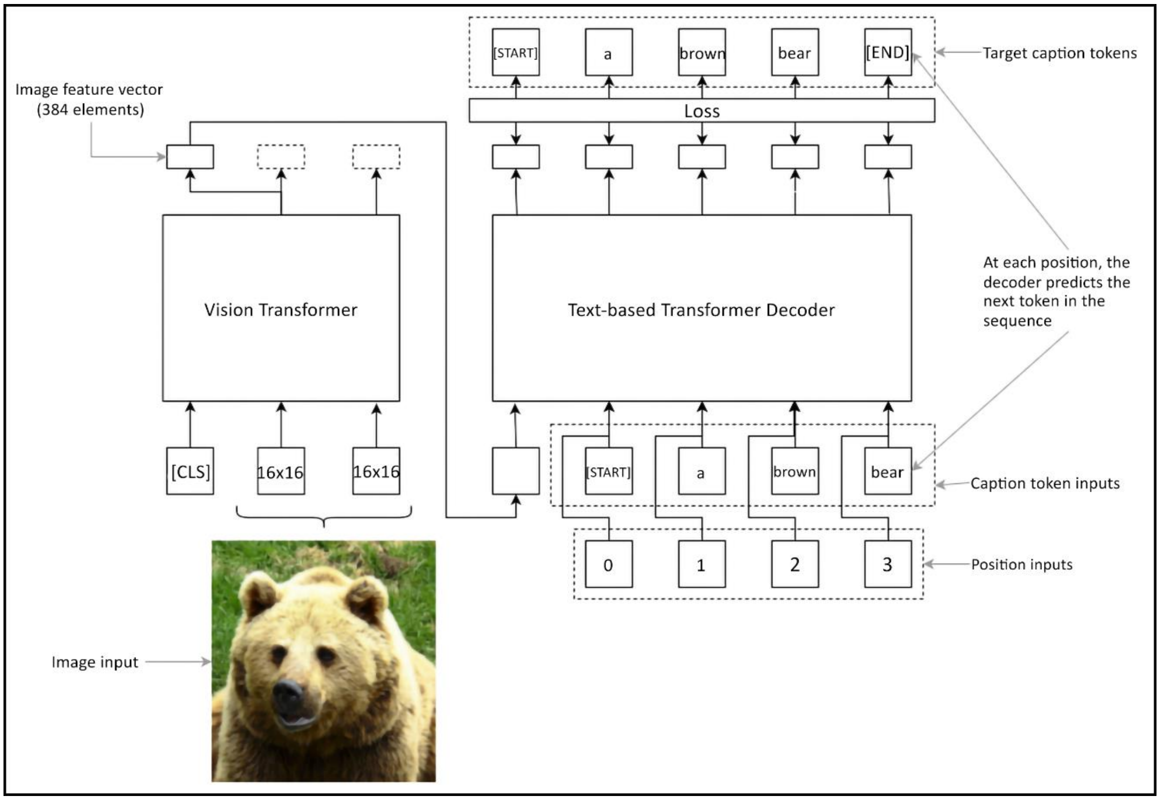
three teddy bears giving each other a hug

A group of three stuffed animal teddy bears.

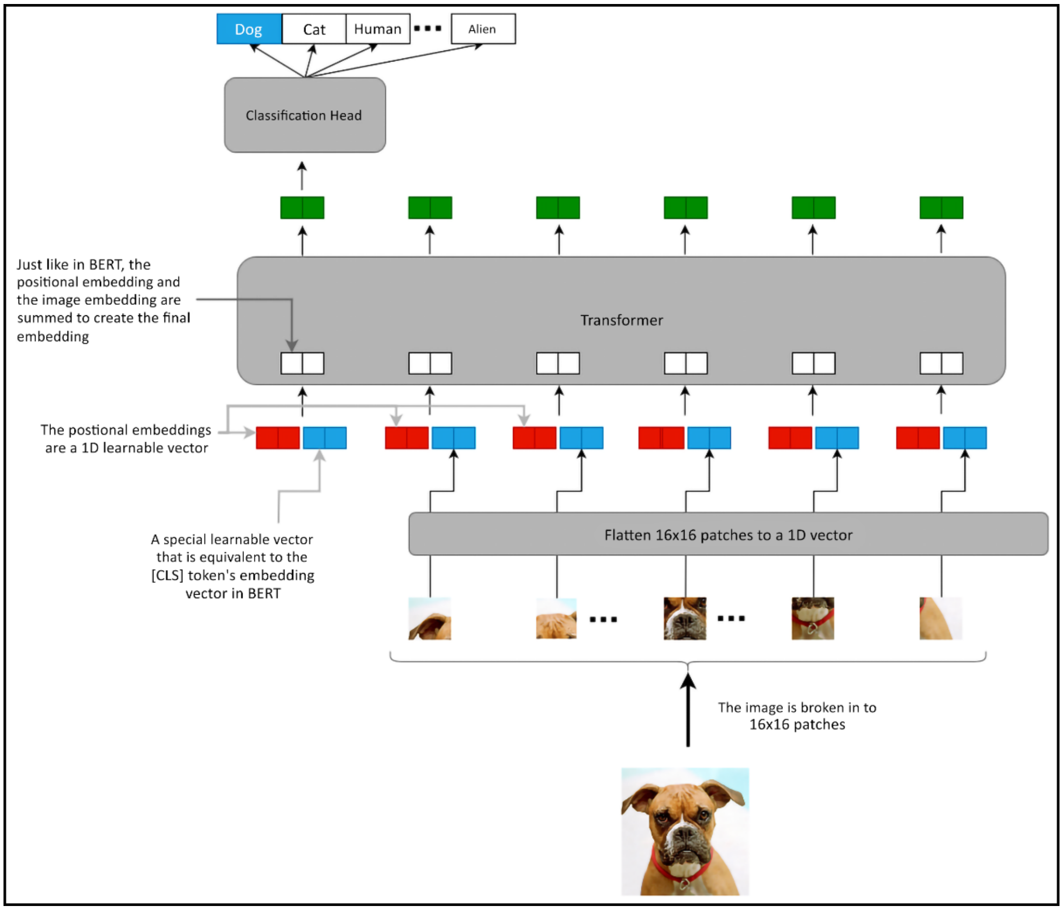
Three stuffed bears hugging and sitting on a blue pillow

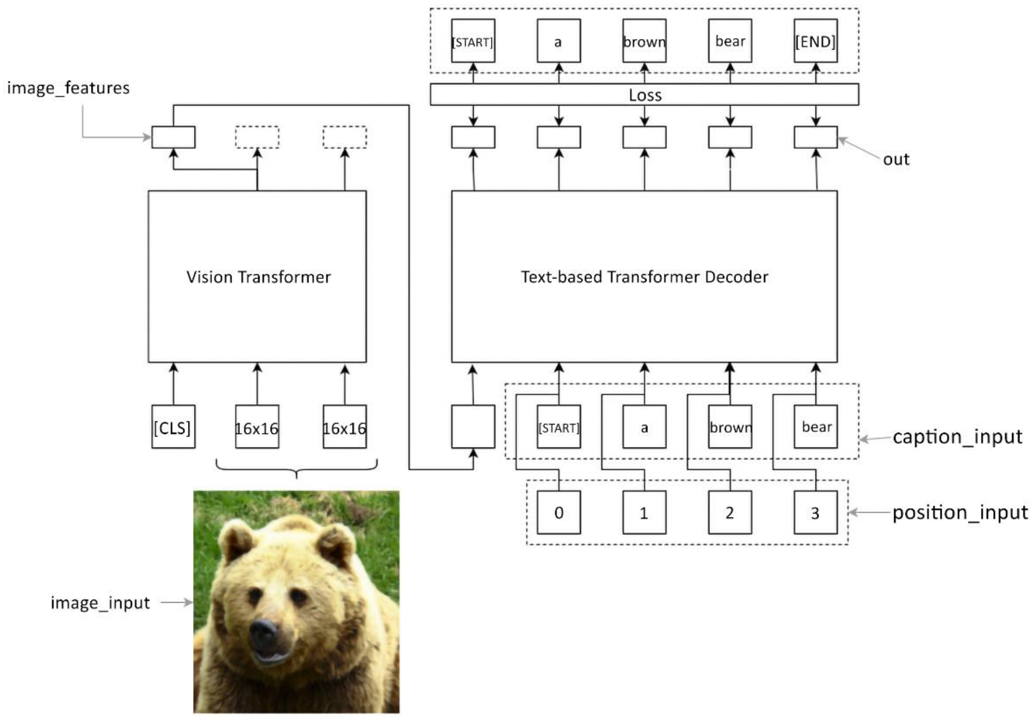
	image_id	id	caption	image_filepath	preprocessed_caption
0	318556	48	A very clean and well decorated empty bathroom	data\train2014\train2014\COCO_train2014_000000...	[START] a very clean and well decorated empty ...
1	116100	67	A panoramic view of a kitchen and all of its a...	data\train2014\train2014\COCO_train2014_000000...	[START] a panoramic view of a kitchen and all ...
2	318556	126	A blue and white bathroom with butterfly theme...	data\train2014\train2014\COCO_train2014_000000...	[START] a blue and white bathroom with butterf...
3	116100	148	A panoramic photo of a kitchen and dining room	data\train2014\train2014\COCO_train2014_000000...	[START] a panoramic photo of a kitchen and din...
4	379340	173	A graffiti-ed stop sign across the street from...	data\train2014\train2014\COCO_train2014_000000...	[START] a graffitied stop sign across the stre...

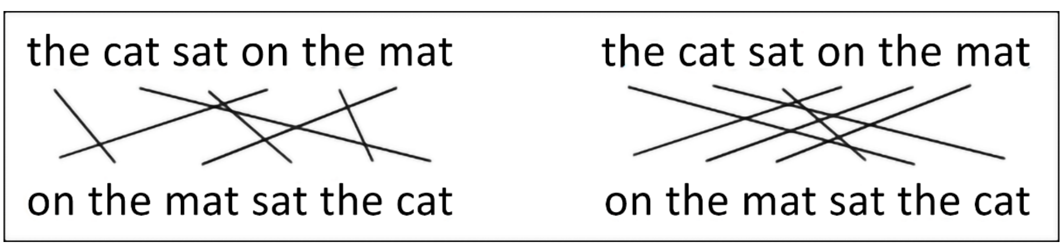
	0	1
count	808.000000	808.000000
mean	571.698020	499.007426
std	95.377281	100.333224
min	332.000000	182.000000
25%	480.000000	427.000000
50%	640.000000	480.000000
75%	640.000000	640.000000
max	640.000000	640.000000

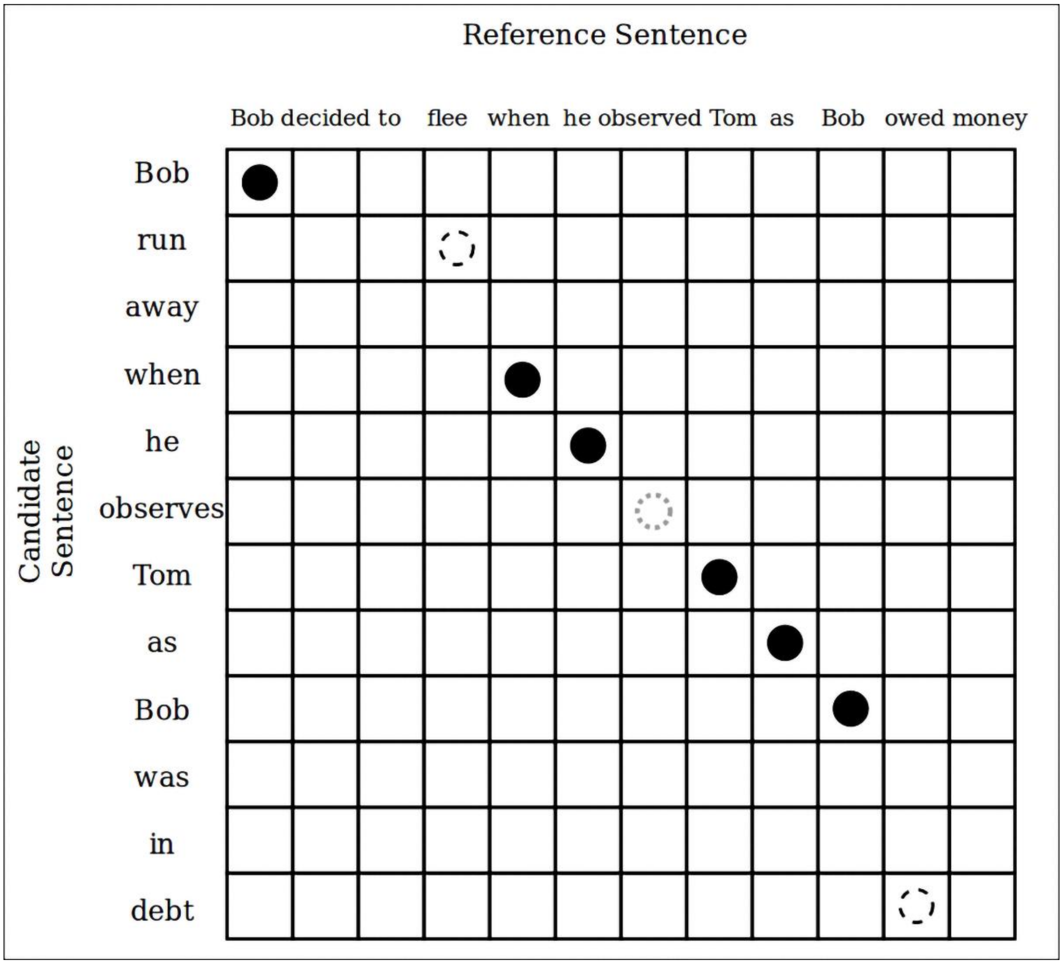


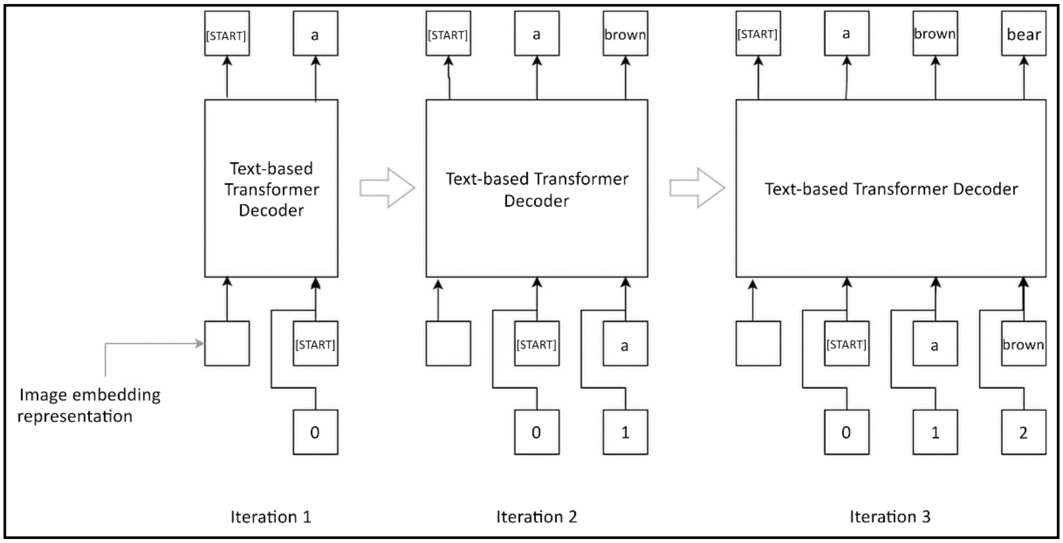
[START]
a
brown
bear
[END]
Target caption tokens
Image feature vector
(384 elements)
Loss
Vision Transformer
Text-based Transformer Decoder
At each position, the decoder predicts the next token in the sequence
[CLS]
16x16
16x16
[START]
a
brown
bear
Caption token inputs
0
1
2
3
Position inputs
Image input


Dog
Cat
Human
Alien
Classification Head
Just like in BERT, the positional embedding and the image embedding are summed to create the final embedding
Transformer
The postional embeddings are a 1D learnable vector
A special learnable vector that is equivalent to the [CLS] token's embedding vector in BERT
Flatten 16x16 patches to a 1D vector
The image is broken in to 16x16 patches


[START]
a
brown
bear
[END]
Loss
image_features
out
Vision Transformer
Text-based Transformer Decoder
[CLS]
16x16
16x16
[START]
a
brown
bear
caption_input
0
1
2
3
position_input
image_input


the cat sat on the mat
on the mat sat the cat
the cat sat on the mat
on the mat sat the cat


Reference Sentence
Bob decided to flee when he observed Tom as Bob owed money
Candidate Sentence
Bob
run
away
when
he
observes
Tom
as
Bob
was
in
debt


[START]
a
[START]
a
brown
[START]
a
brown
bear
Text-based Transformer Decoder
Text-based Transformer Decoder
Text-based Transformer Decoder
[START]
[START]
a
[START]
a
brown
Image embedding representation
0
0
1
0
1
2
Iteration 1
Iteration 2
Iteration 3

TRUE: [START] cars parked along a street with a street light that is green [END]

PRED: [START] a car stopped on a city street with cars [END]

TRUE: [START] several containers of food and beverages on a wooden table outdoors [END]

PRED: [START] a table and a plates on a table [END]

TRUE: [START] a few street lights on the side of the road [END]

PRED: [START] a traffic light on a pole in the middle of a city [END]

TRUE: [START] a person riding the horse jumping over a wooden obstacle [END]

PRED: [START] a person on a horse in a field [END]

TRUE: [START] a blue cake features a small island on top and sharks swimming on the bottom of the cake [END]

PRED: [START] a cake with a blue and white frosting on it [END]

TRUE: [START] a plastic dish with the food sectioned off [END]

PRED: [START] a tray of food and vegetables and a sandwich [END]

TRUE: [START] three images of the same man hitting a tennis ball [END]

PRED: [START] a man standing on a tennis court holding a racquet [END]

TRUE: [START] two people who are swimming in the water on surfboards [END]

PRED: [START] a dog in a body of water in the water [END]

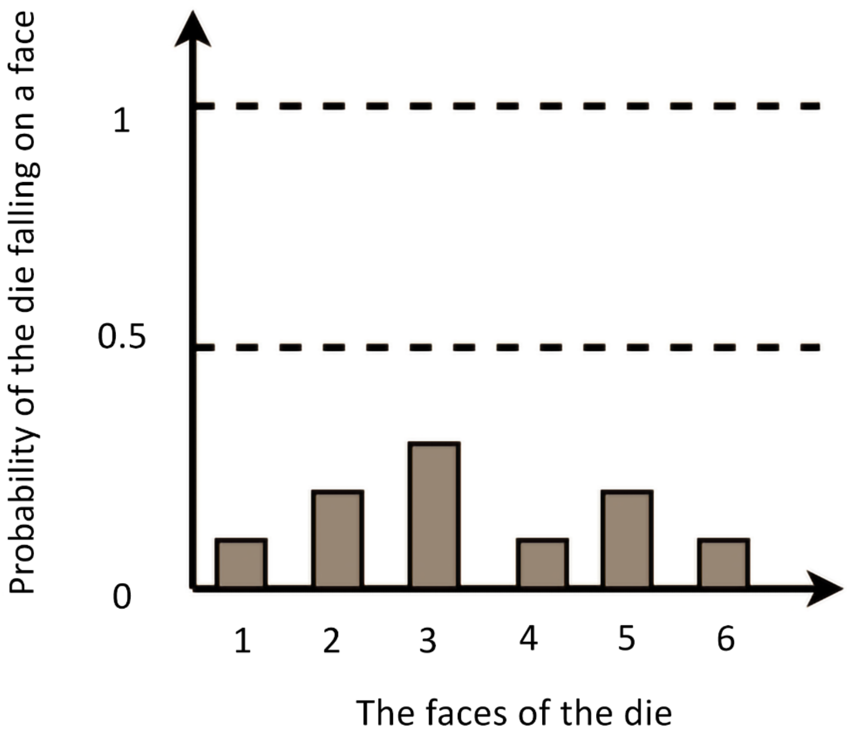
TRUE: [START] a person on a surf board riding a wave [END]

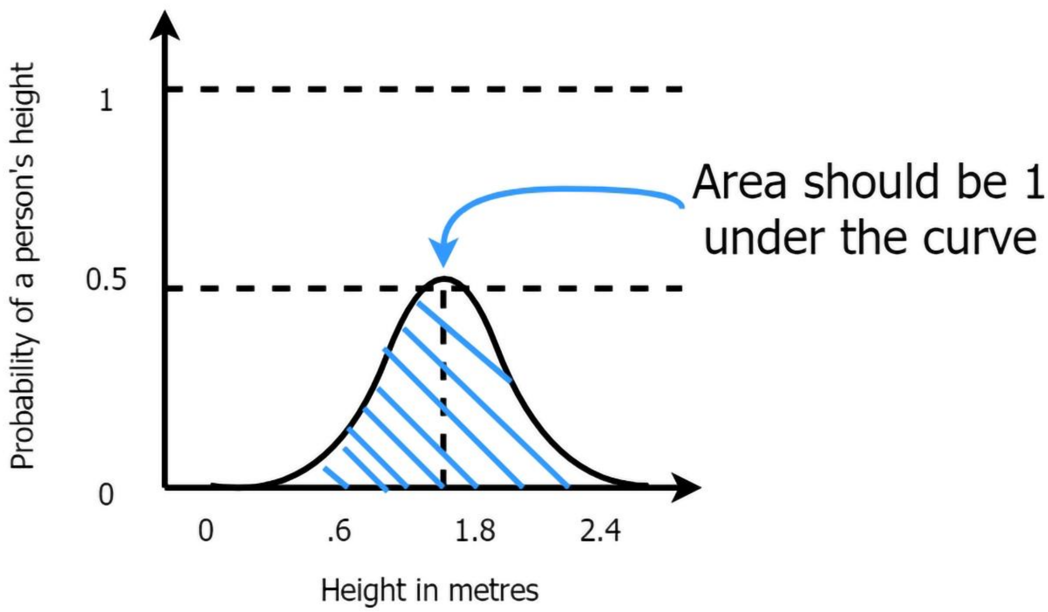
PRED: [START] a man on a surfboard in the ocean [END]

TRUE: [START] an individual is taken in this very picture [END]

PRED: [START] a group of people standing in a terminal [END]

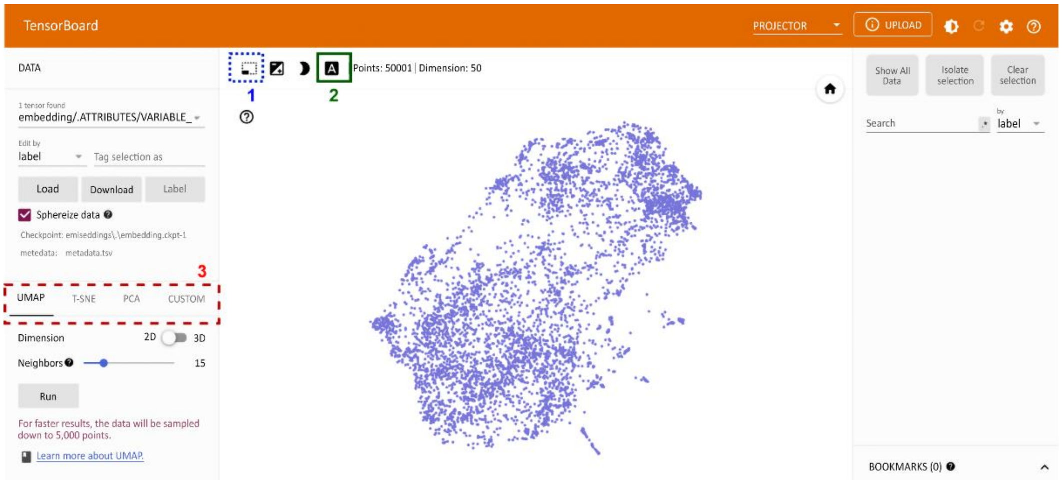
Appendix A: Mathematical Foundations and Advanced TensorFlow

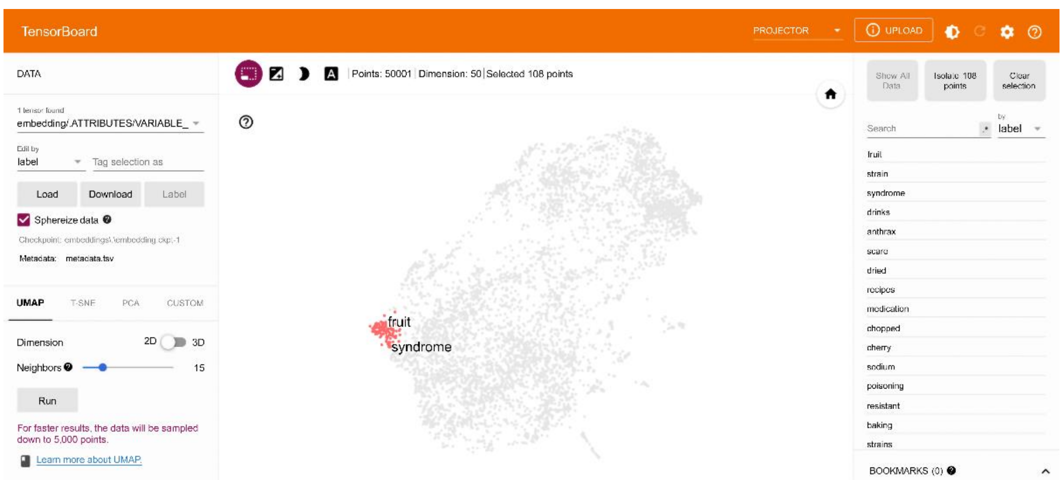




	0	1	2	3	4	5	6	7	8	9	...	40	41	42	43	44
the	0.418000	0.249680	-0.41242	0.121700	0.345270	-0.044457	-0.49688	-0.178620	-0.000660	-0.656600	...	-0.298710	-0.157490	-0.347580	-0.045637	-0.442510
,	0.013441	0.236820	-0.16899	0.409510	0.638120	0.477090	-0.42852	-0.556410	-0.364000	-0.239380	...	-0.080262	0.630030	0.321110	-0.467650	0.227860
.	0.151640	0.301770	-0.16763	0.176840	0.317190	0.339730	-0.43478	-0.310860	-0.449990	-0.294860	...	-0.000064	0.068987	0.087939	-0.102850	-0.139310
of	0.708530	0.570880	-0.47160	0.180480	0.544490	0.726030	0.18157	-0.523930	0.103810	-0.175660	...	-0.347270	0.284830	0.075693	-0.062178	-0.389880
to	0.680470	-0.039263	0.30186	-0.177920	0.429620	0.032246	-0.41376	0.132280	-0.298470	-0.085253	...	-0.094375	0.018324	0.210480	-0.030880	-0.197220
and	0.268180	0.143460	-0.27877	0.016257	0.113840	0.699230	-0.51332	-0.473680	-0.330750	-0.138340	...	-0.069043	0.368850	0.251680	-0.245170	0.253810
in	0.330420	0.249950	-0.60874	0.109230	0.036372	0.151000	-0.55083	-0.074239	-0.092307	-0.328210	...	-0.486090	-0.008027	0.031184	-0.365760	-0.426990
a	0.217050	0.465150	-0.46757	0.100820	1.013500	0.748450	-0.53104	-0.262560	0.168120	0.131820	...	0.138130	0.369730	-0.642890	0.024142	-0.039315
"	0.257690	0.456290	-0.76974	-0.376790	0.592720	-0.063527	0.20545	-0.573850	-0.290090	-0.136620	...	0.030498	-0.395430	-0.385150	-1.000200	0.087599
's	0.237270	0.404780	-0.20547	0.588050	0.655330	0.328670	-0.81964	-0.232360	0.274280	0.242650	...	-0.123420	0.659610	-0.518020	-0.829950	-0.082739

10 rows × 50 columns


TensorBoard
PROJECTOR
UPLOAD
DATA
1 tensor found
embedding/.ATTRIBUTES/VARIABLE_
Edit by
label
Tag selection as
Load
Download
Label
Sphereize data
Checkpoint: embeddings\.\embedding.ckpt-1
metadata: metadata.tsv
3
UMAP
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Neighbors
15
Run
For faster results, the data will be sampled down to 5,000 points.
Learn more about UMAP.
1
2
Points: 50001 | Dimension: 50
Show All Data
Isolate selection
Clear selection
Search
by
label
BOOKMARKS (0)


TensorBoard
PROJECTOR
UPLOAD
DATA
1 tensor found
embedding/.ATTRIBUTES/VARIABLE_
Edit by
label
Tag selection as
Load
Download
Label
Sphereize data
Metadata: metadata.tsv
UMAP
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Neighbors
15
Run
For faster results, the data will be sampled down to 5,000 points.
Learn more about UMAP.
Points: 50001 | Dimension: 50 | Selected 108 points
fruit
syndrome
Show All Data
Isolate 108 points
Clear selection
Search
by
label
fruit
strain
syndrome
drinks
anthrax
scare
dried
recipes
medication
chopped
cherry
sodium
poisoning
resistant
baking
strains
BOOKMARKS (0)


TensorBoard
PROJECTOR
UPLOAD
DATA
1 tensor found
embedding/.ATTRIBUTES/VARIABLE_
Edit by
label
Tag selection as
Load
Download
Label
Sphereize data
Checkpoint: embeddings\.\embedding.ckpt-1
Metadata: metadata.tsv
UMAP
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Neighbors
15
Run
For faster results, the data will be sampled down to 5,000 points.
Learn more about UMAP.
Points: 50001
Dimension: 50
chiseled
1
sauvignon
pollen
ethanol
cherry
microbe
potted
insecticides
scooping
jugs
apricots
faucet
rinsed
tastings
stinking
reagent
leach
clump
zoomed
braided
bead
earthy
zipper
anesthesia
hues
Show All Data
Isolate 108 points
Clear selection
Search
by
label
BOOKMARKS (0)