Word2Vec with TensorFlow

Department of Computer Science,
National Tsing Hua University, Taiwan

Deep Learning

Word embedding

- How to represent words in machine learning?

  My favorite pet is cat .

  - Treat words as discrete atomic symbols, and therefore 'cat' may be represented as '1' and 'pet' as '2.'

  - However, this embedding includes no semantics. Tasks learning from this representation causes models hard to train.

  - Better solutions?

Word embedding

- Intuition: Modeling relationships between words.
  - Semantics Similarity
    - CBOW (Continuous Bag-of-Words model),
    - Skip-Gram
  - Grammar / Syntax

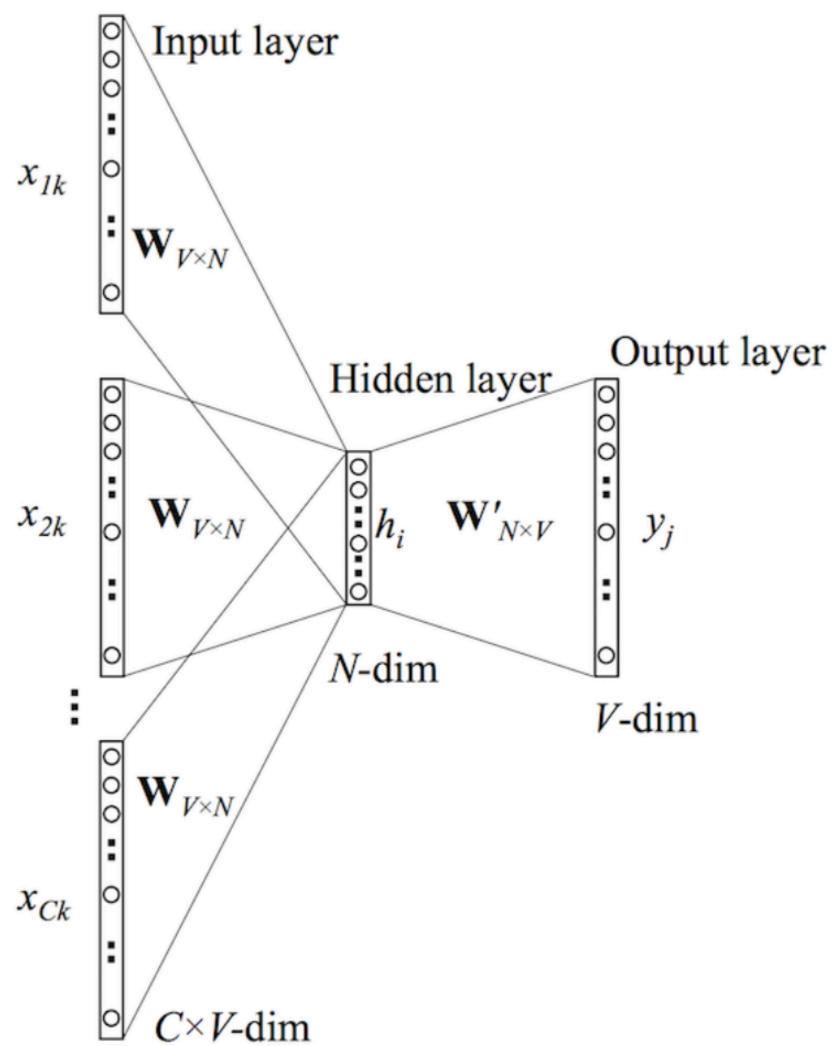
CBOW & Skip-Gram

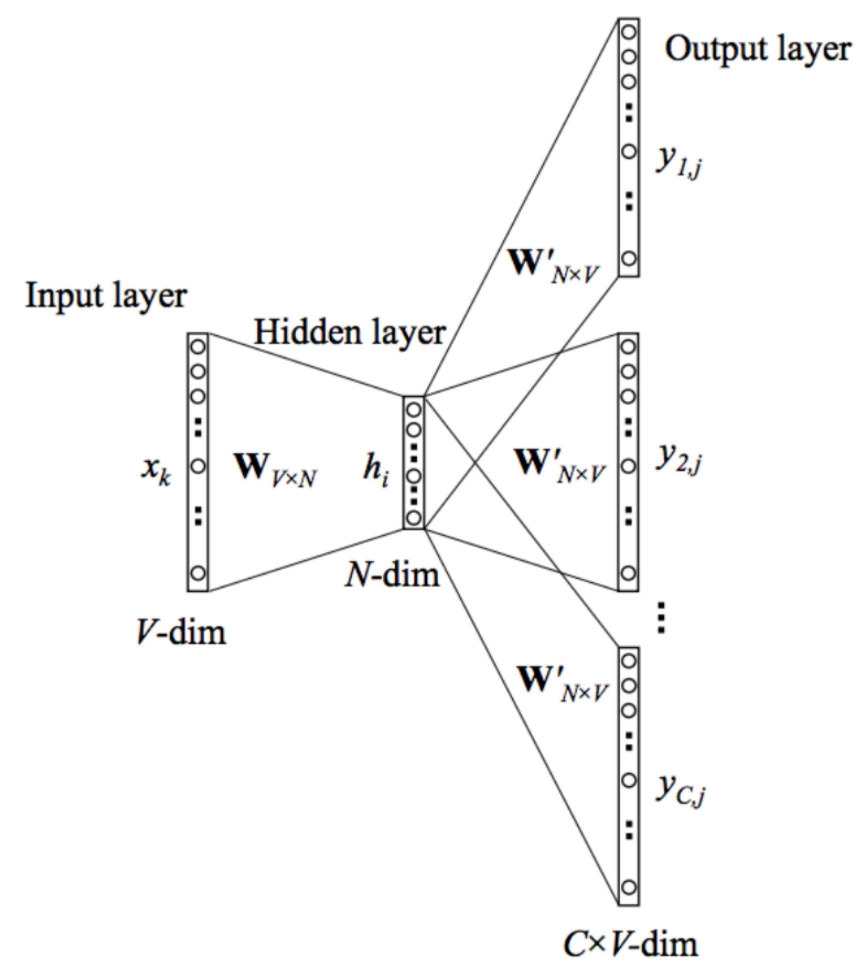
- Assumptions in CBOW & Skip-Gram:
  - Words with similar meaning / domain have the close contexts.

My favorite pet is ___. *cat, dog*

CBOW & Skip-Gram

- CBOW: Predicting the target words with context words.
- Skip-Gram: Predicting context words from the target words.





CBOW & Skip-Gram

- For example, given the sentence:

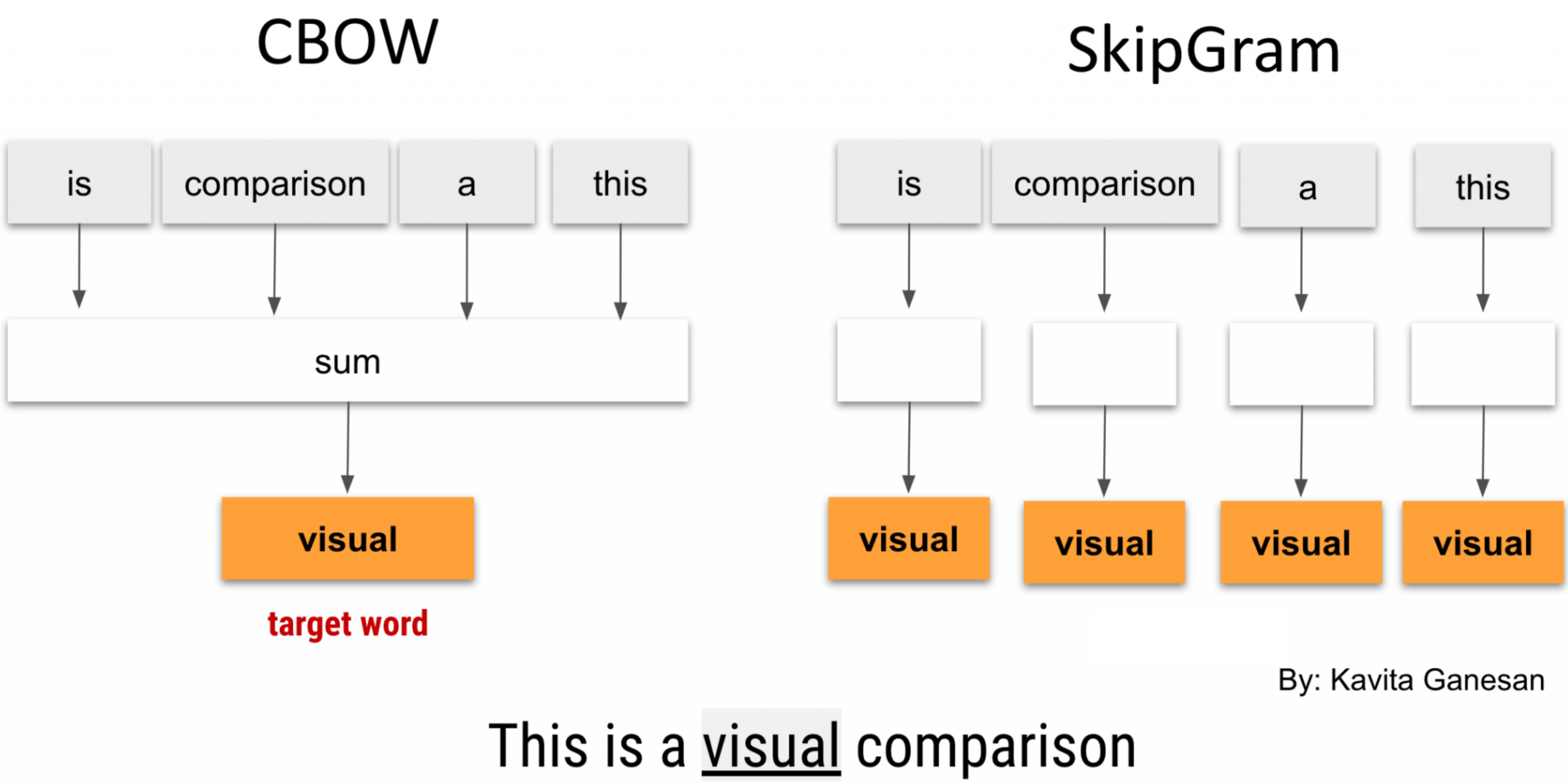
the quick brown fox jumped over the lazy dog.

- CBOW will be trained on the dataset:

```
([the, brown], quick), ([quick, fox], brown), …
```

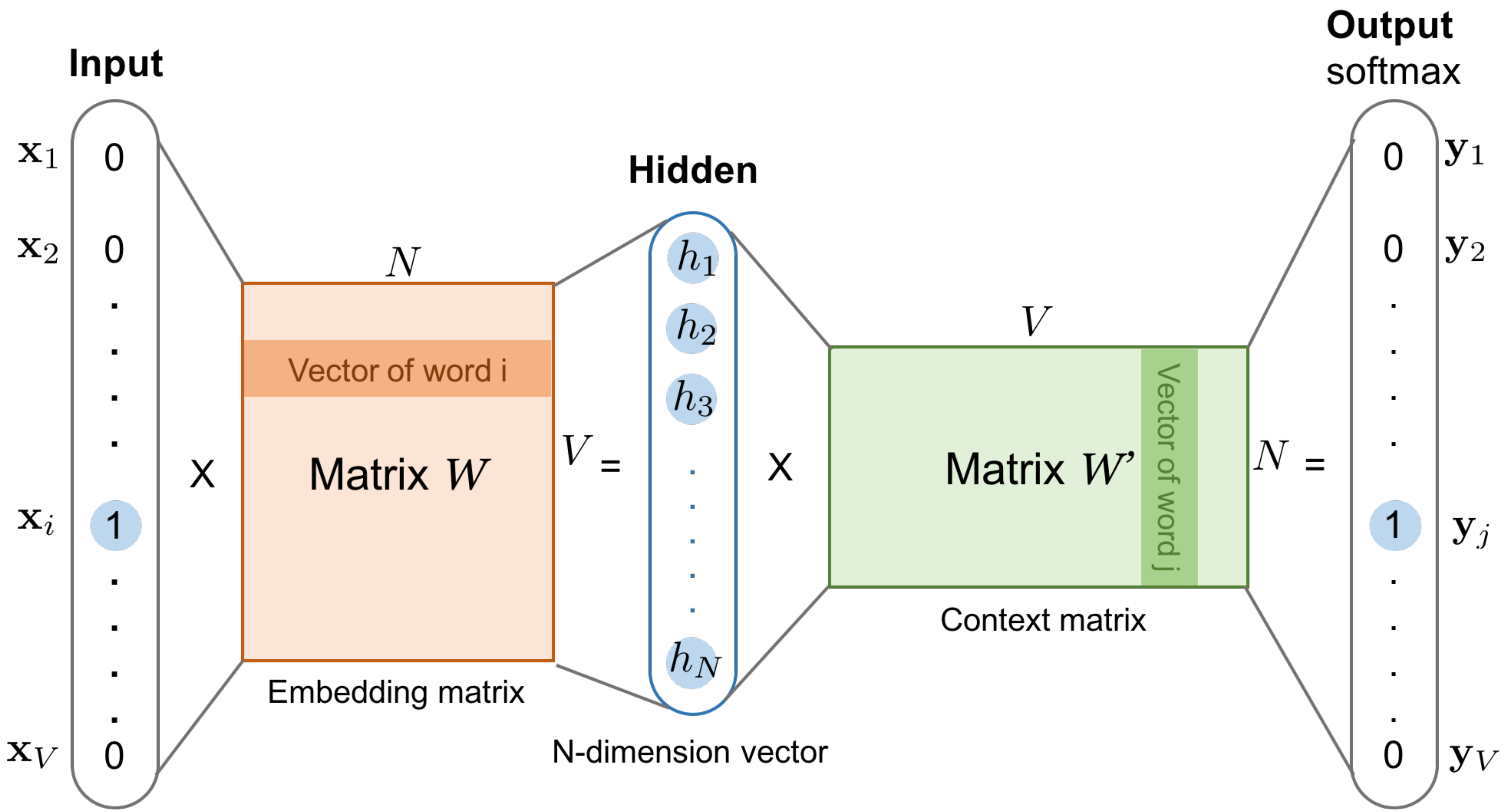
- Skip-Gram will be trained on the dataset:

```
(quick, the), (quick, brown), (brown, quick),
(brown, fox)…
```



Picture from https://kavita-ganesan.com/

Skip-Gram



Picture from https://lilianweng.github.io/

Softmax & Words classification

- Recap the softmax function:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- While training the embedding, we need to sum over the entire vocabulary in the denominator, which is costly.

$$p(w = i \,|\, \mathbf{x}) = \frac{\exp(h^\top v_{w_i'})}{\boxed{\sum_{j \in V} \exp(h^\top v_{w_j'})}}$$

$|V|$ **(vocabulary size) is thousands of millions**

Noise Contrastive Estimation

- Intuition: Estimate the parameters by learning to discriminate between the data w and some artificially generated noise $\tilde{w}$.

$$\textit{maximum likelihood estimation} \rightarrow \textit{noise contrastive estimation}$$

- Objective function:

$$\arg\max \ \log p(d=1 \mid w_t, x) + \sum_{\tilde{w}_i \sim Q}^{N} \log p(d=0 \mid \tilde{w}_i, x)$$

- Derive form:

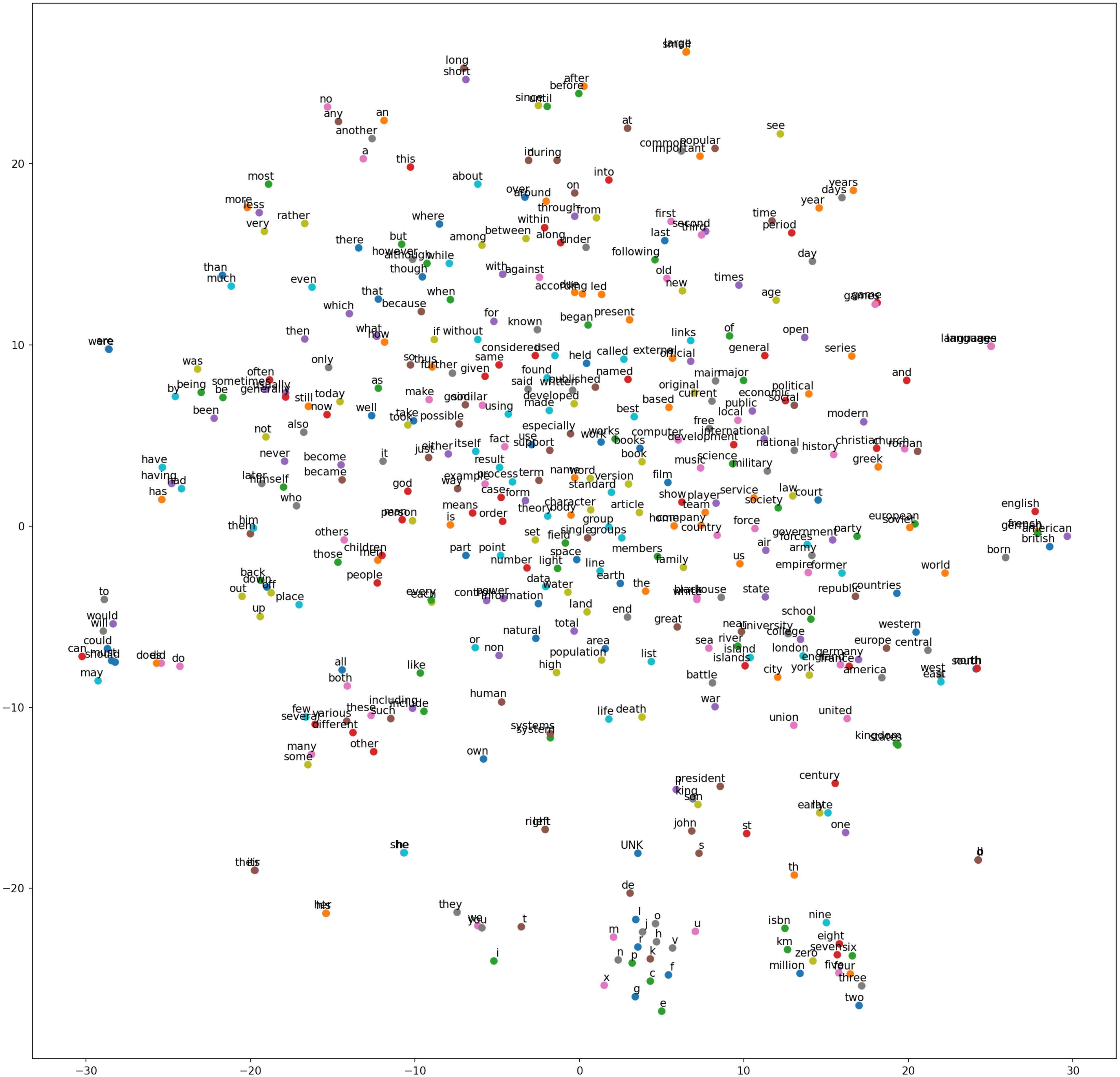
$$\arg\max \ \log \frac{p(w_t \mid x)}{p(w_t \mid x) + Nq(\tilde{w})} + \sum_{\tilde{w}_i \sim Q}^{N} \log \frac{Nq(\tilde{w}_i)}{p(w_t \mid x) + Nq(\tilde{w}_i)}$$

Noise Contrastive Estimation

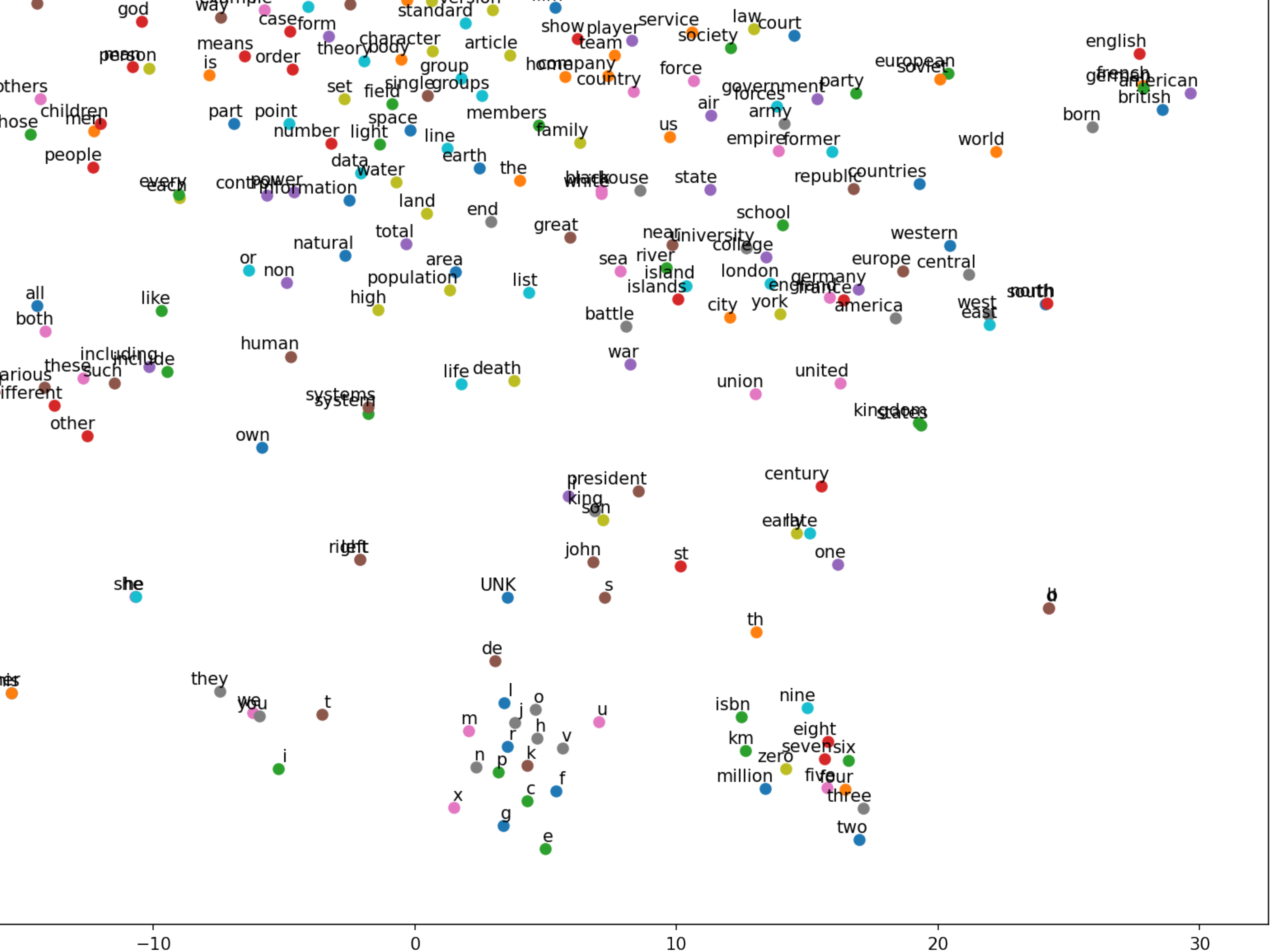
- Intuition: Estimate the parameters by learning to discriminate between the data w and some artificially generated noise $\tilde{w}$.

$$\textit{maximum likelihood estimation} \rightarrow \textit{noise contrastive estimation}$$

- After training: evaluate with softmax function.



20
10
0
−10
−20
−30
−20
−10
0
10
20
30
long
short
after
before
since
until
no
any
an
another
at
see
common
popular
important
a
this
during
most
about
into
over
around
on
years
days
year
more
less
very
rather
where
through
from
within
time
period
first
second
third
last
there
but
however
although
while
though
among
between
along
under
following
day
than
much
even
with
against
old
new
times
age
that
when
because
according
led
which
for
known
began
present
then
what
how
if
without
links
of
open
language
languages
were
are
was
only
so
thus
further
given
considered
used
same
held
called
external
official
general
series
and
by
being
be
often
sometimes
generally
as
found
said
published
written
named
main
major
original
current
political
economic
social
make
similar
developed
made
based
public
been
still
today
now
well
take
took
possible
using
best
local
modern
not
also
free
especially
works
work
books
computer
international
development
national
history
christian
church
roman
have
either
itself
fact
use
support
book
science
military
music
greek
never
become
just
it
result
having
had
later
himself
became
example
process
term
name
word
version
standard
film
law
court
has
god
way
case
form
show
player
team
service
society
who
character
article
english
him
them
person
reason
means
is
order
theory
body
group
company
european
soviet
french
german
american
british
others
set
field
single
groups
country
force
government
forces
party
children
men
part
point
members
air
army
born
those
number
light
space
line
family
us
empire
former
world
people
earth
data
water
the
white
house
state
republic
countries
back
down
off
out
place
to
up
every
each
power
information
land
end
school
great
total
near
university
college
western
would
will
natural
or
non
area
sea
river
island
europe
central
could
can
must
should
does
did
do
population
list
islands
london
england
germany
france
city
york
america
south
north
west
east
may
all
high
both
battle
like
human
war
including
include
few
several
various
these
such
different
systems
system
life
death
union
united
many
some
other
own
kingdom
states
president
king
son
century
late
early
right
left
john
st
one
she
he
UNK
s
their
its
th
de
his
her
they
we
you
t
l
o
u
isbn
nine
m
j
h
r
v
eight
km
seven
six
i
n
p
k
zero
f
million
five
four
c
three
x
g
two
e



god
case
form
standard
show
player
service
law
court
society
means
character
article
team
english
person
is
order
theory
body
group
company
european
soviet
force
country
others
set
field
single
groups
government
party
french
german
american
forces
british
air
army
hose
children
men
part
point
space
members
family
us
born
number
light
line
empire
former
world
people
data
earth
water
the
every
each
power
information
black
white
house
state
republic
countries
land
end
school
great
near
university
college
western
natural
total
or
non
area
sea
river
island
london
europe
central
population
list
islands
germany
england
france
north
south
all
like
high
city
york
america
west
east
both
battle
human
including
include
these
such
war
united
life
death
union
various
different
systems
system
kingdom
states
other
own
president
century
king
son
early
late
right
left
john
st
one
she
he
UNK
s
th
de
they
we
you
t
l
o
j
u
isbn
nine
eight
m
h
r
v
km
seven
six
n
k
zero
p
i
f
million
five
four
x
c
three
g
two
e
−10
0
10
20
30

Homework

- Functional API:

```
from tensorflow.keras import Model, Input

center_words = Input(shape=(), name='center_words', dtype='int32')
target_words = Input(shape=(1), name='target_words', dtype='int32')

embedding = embedding_lookup()(center_words)
loss = nce_loss()((embedding, target_words))

word2vec = Model(name='word2vec',
                 inputs=[center_words, target_words],
                 outputs=[loss])
```

Homework

- Model subclassing:

```
class ResNet(tf.keras.Model):

    def __init__(self):
        super(ResNet, self).__init__()
        self.block_1 = ResNetBlock()
        self.block_2 = ResNetBlock()
        self.global_pool = layers.GlobalAveragePooling2D()
        self.classifier = Dense(num_classes)

    def call(self, inputs):
        x = self.block_1(inputs)
        x = self.block_2(x)
        x = self.global_pool(x)
        return self.classifier(x)


resnet = ResNet()
```

Reference

- https://ruder.io/word-embeddings-softmax/index.html#noisecontrastiveestimation
- https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html
- https://aegis4048.github.io/optimize_computational_efficiency_of_skip-gram_with_negative_sampling
- http://proceedings.mlr.press/v9/gutmann10a/gutmann10a.pdf
- https://papers.nips.cc/paper/5165-learning-word-embeddings-efficiently-with-noise-contrastive-estimation.pdf
- https://www.tensorflow.org/api_docs/python/tf/nn/nce_loss