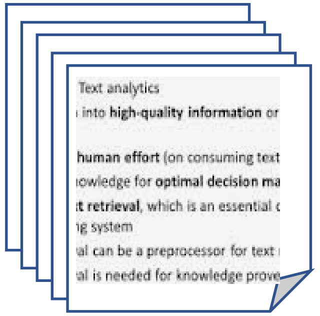
Word Embedding

Text Analysis



Text Analysis

Various Applications

- Text Recommendation
  - Information Retrieval
- Text Classification
  - Sentiment analysis
  - Hate/Communal speech detection
  - Fake news detection
- NLP
  - POS
  - NER
  - Machine transliteration
  - Machine Translation
- Image/Video Tagging

Text Analysis

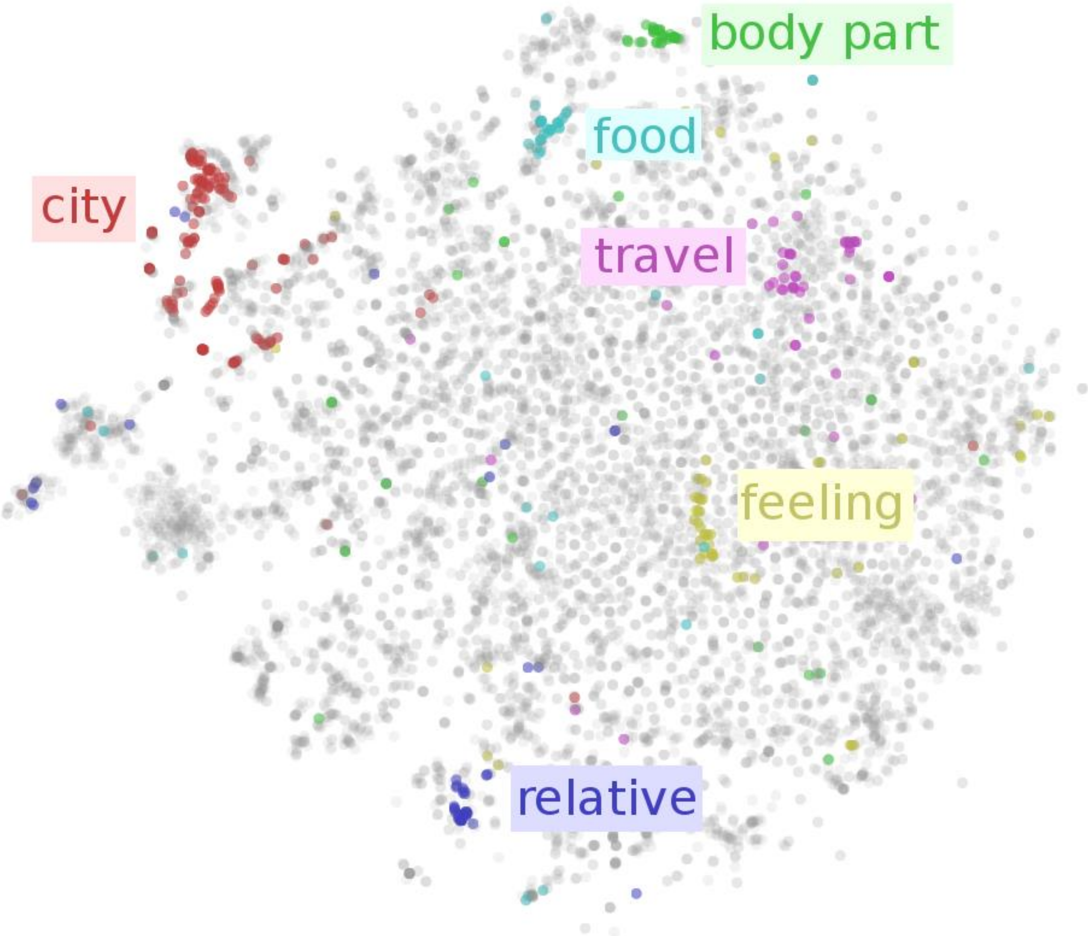
We need effective representation of :

- Words
- Sentences
- Documents

Various Applications

- Text Recommendation
  - Information Retrieval
- Text Classification
  - Sentiment analysis
  - Hate/Communal speech detection
  - Fake news detection
- NLP
  - POS
  - NER
  - Machine transliteration
  - Machine Translation
- Image/Video Tagging

Text Analysis



We need effective representation of :

- Words
- Sentences
- Document

Traditional Form of Text Representation

$$\begin{array}{c|cccccccccc|}
 & t_1 & t_2 & t_3 & & & & & & & t_m \\
\hline
d_1 & 1 & 0 & 1 & 1 & \ldots\ldots.\ldots. & 0 & 0 & 0 & 1 & \\
d_2 & 1 & 1 & 0 & 0 & \ldots\ldots.\ldots. & 0 & 1 & 1 & 0 & \\
d_3 & & & & & & & & & & \\
d_4 & & & & & & & & & & \\
 & & & & & & & & & & \\
d_n & 0 & 0 & 1 & 0 & \ldots\ldots.\ldots. & 0 & 1 & 0 & 0 & \\
\hline
\end{array}$$

$n_x m$

Document and Term Matrix

Possible Terms are

- Unigram
- Bigram
- Tri-gram

Traditional Form of Text Representation

$$
\begin{array}{c|cccccccccc|}
 & t_1 & t_2 & t_3 & & & & & & & t_1 \\
\hline
d_1 & 1 & 0 & 1 & 1 & \ldots\ldots.\ldots. & 0 & 0 & 0 & 1 \\
d_2 & 1 & 1 & 0 & 0 & \ldots\ldots.\ldots. & 0 & 1 & 1 & 0 \\
d_3 & & & & & & & & & \\
d_4 & & & & & & & & & \\
 & & & & & & & & & \\
d_n & 0 & 0 & 1 & 0 & \ldots\ldots.\ldots. & 0 & 1 & 0 & 0 \\
\hline
\end{array}
$$

A document is represented by term features

Document and Term Matrix

Traditional Form of Text Representation

$$
\begin{array}{c|ccccccccc}
 & t_1 & t_2 & t_3 & & & & & & t_1 \\
\hline
d_1 & 1 & 0 & 1 & 1 & \ldots\ldots . \ldots. & 0 & 0 & 0 & 1 \\
d_2 & 1 & 1 & 0 & 0 & \ldots\ldots . \ldots. & 0 & 1 & 1 & 0 \\
d_3 & & & & & & & & & \\
d_4 & & & & & & & & & \\
d_n & 0 & 0 & 1 & 0 & \ldots\ldots . \ldots. & 0 & 1 & 0 & 0
\end{array}
$$

A term is represented by document features

Document and Term Matrix

Traditional Form of Text Representation

$$
\begin{array}{c|cccccccccc|}
 & t_1 & t_2 & t_3 & & & & & & & t_1 \\
\hline
d_1 & 1 & 0 & 1 & 1 & \ldots\ldots & . \ldots. & 0 & 0 & 0 & 1 \\
d_2 & 1 & 1 & 0 & 0 & \ldots\ldots & . \ldots. & 0 & 1 & 1 & 0 \\
d_3 & & & & & & & & & & \\
d_4 & & & & & & & & & & \\
 & & & & & & & & & & \\
d_n & 0 & 0 & 1 & 0 & \ldots\ldots & . \ldots. & 0 & 1 & 0 & 0 \\
\hline
\end{array}
$$

Document and Term Matrix

Find the most similar documents

- Estimate similarity between the documents

$$Cos\theta(d_i, d_j) = \frac{\bar{d}_i . \bar{d}_j}{\|\bar{d}_i\| . \|\bar{d}_j\|}$$

Find the most similar words

- Estimate similarity between the documents

Traditional Form of Text Representation

$$
\begin{array}{c|ccccccccc|}
 & t_1 & t_2 & t_3 & & & & & & t_1 \\
\hline
d_1 & 1 & 0 & 1 & 1 & \ldots\ldots.\;\ldots. & 0 & 0 & 0 & 1 \\
d_2 & 1 & 1 & 0 & 0 & \ldots\ldots.\;\ldots. & 0 & 1 & 1 & 0 \\
d_3 & & & & & & & & & \\
d_4 & & & & & & & & & \\
d_n & 0 & 0 & 1 & 0 & \ldots\ldots.\;\ldots. & 0 & 1 & 0 & 0 \\
\hline
\end{array}
$$

Document and Term Matrix

Issues with document-term representations

- **Large feature set**
- **Many of the features may not be useful**
- **Sparse**
- **Curse of dimensionality**

What is word embedding?

Word embedding is a dense representation of words in the form of numeric vectors in lower dimensional space.

What is word embedding?

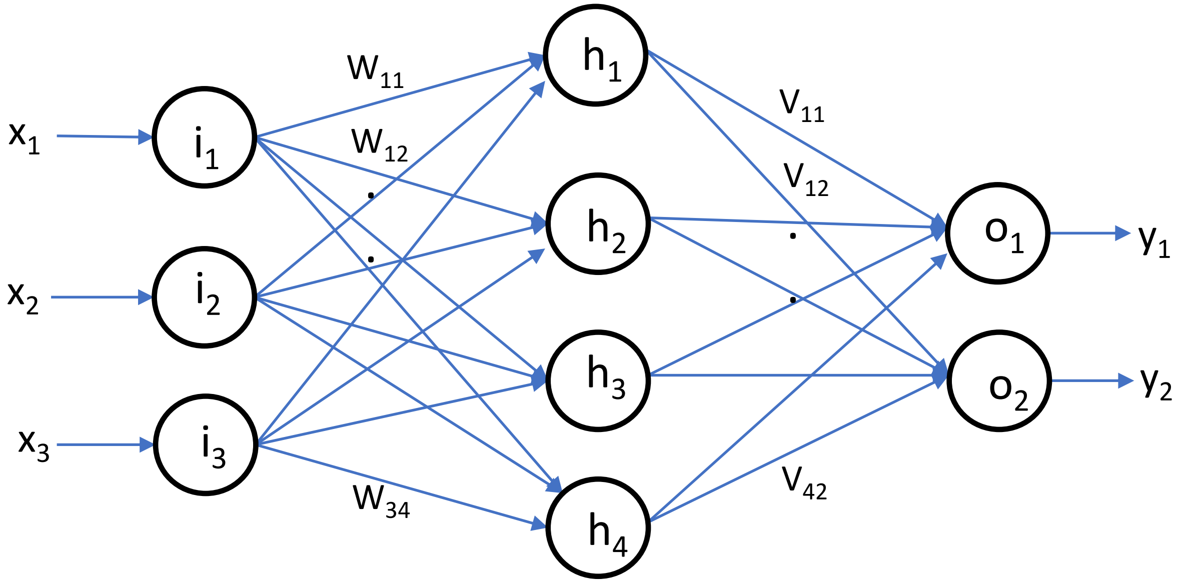
Word embedding is a dense representation of words in the form of numeric vectors in lower dimensional space.

Earlier Methods

- Principal component Analysis (PCA)
- Singular Value Decomposition (SVD) – Latent Semantic Indexing
- Latent Dirichlet Allocation

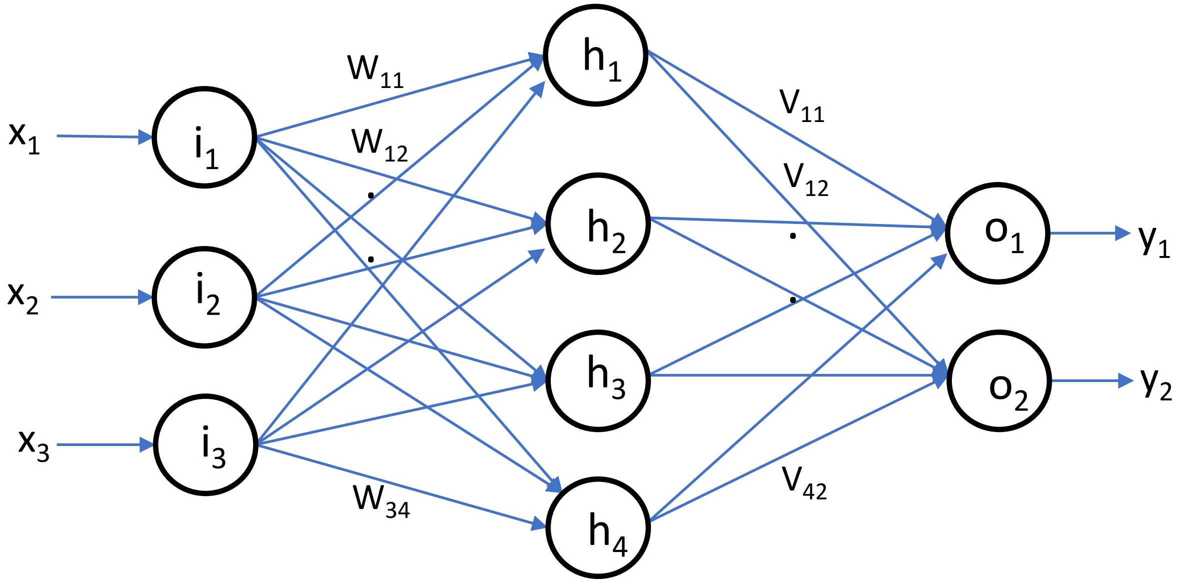
Recent Methods

- Word2Vec (Skipgram/CBOW)
- Glove
- FastText
- BERT



This model can be used for various tasks

- Classification
- Regression
- Probability estimate
- Representation
- ....



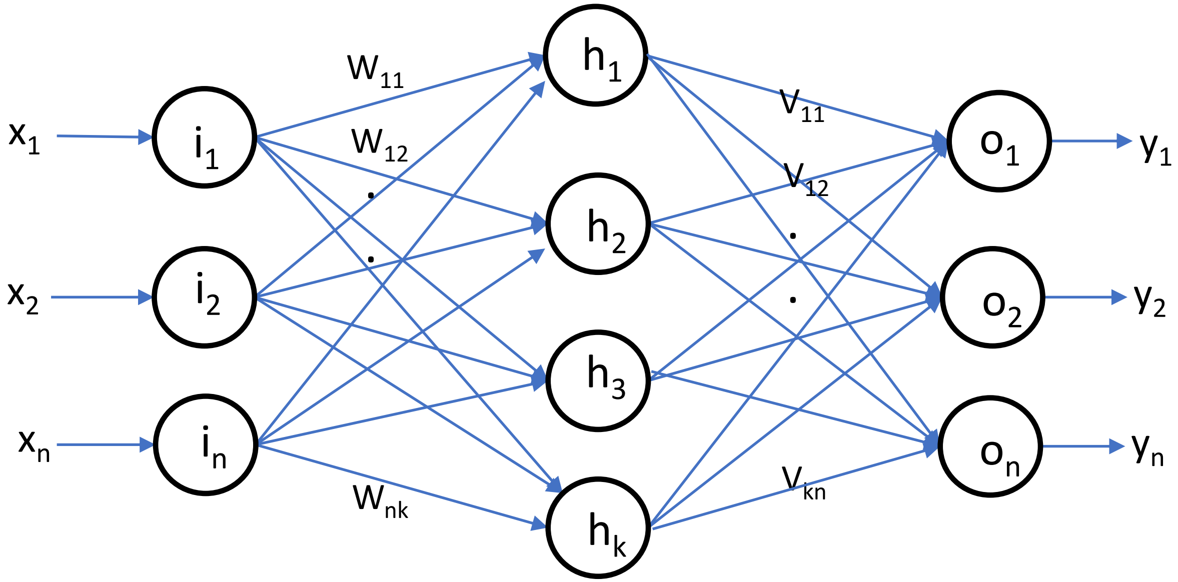
This model can be used for various tasks

- Classification
- Regression
- **Probability estimate**
- Representation
- ….

Example Sentence : The man sat on the floor

Training samples : (The, man), (man, sat), (sat, on), (on, the),……..

Pr(man | The) = ?



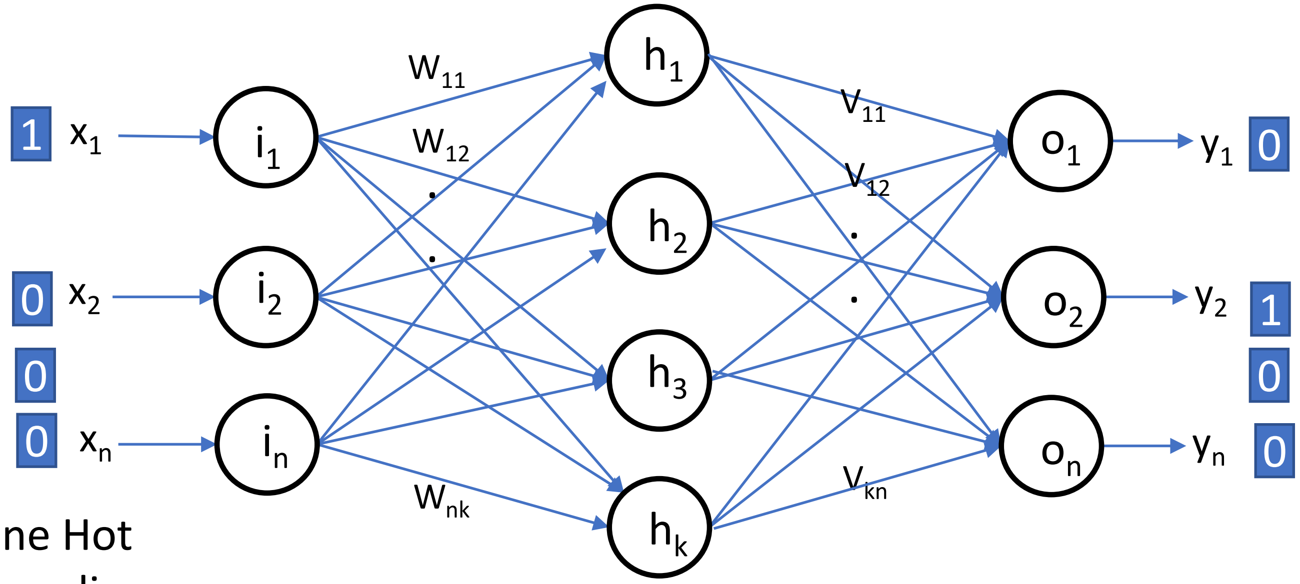
This model can be used for various tasks

- Classification
- Regression
- **Probability estimate**
- Representation
- ….

Example Sentence : The man sat on the floor

Training samples : (The, man), (man, sat), (sat, on), (on, the),……..

Pr(man | The) = ?



One Hot Encoding

This model can be used for various tasks

- Classification
- Regression
- **Probability estimate**
- Representation
- ....

Example Sentence : The man sat on the floor

Training samples : (The, man), (man, sat), (sat, on), (on, the),........

Pr(man | The) = ?

Word2vec – Word Embedding

- Objective : Predict the target word given the words in the context

- Example Sentence : The man sat on the floor

- Target Word: man
- Window Size: 2
- Context Words: The, sat, on

- Training samples : (man, the), (man, sat), (man, on)

Word2vec – Word Embedding

The man sat on the floor

Training samples : (the, man), (the, sat)

The man sat on the floor

Training samples : (man, the), (man, sat)
(man, on)

The man sat on the floor

Training samples : (sat, the), (sat, man)
(sat, on), (sat, the)

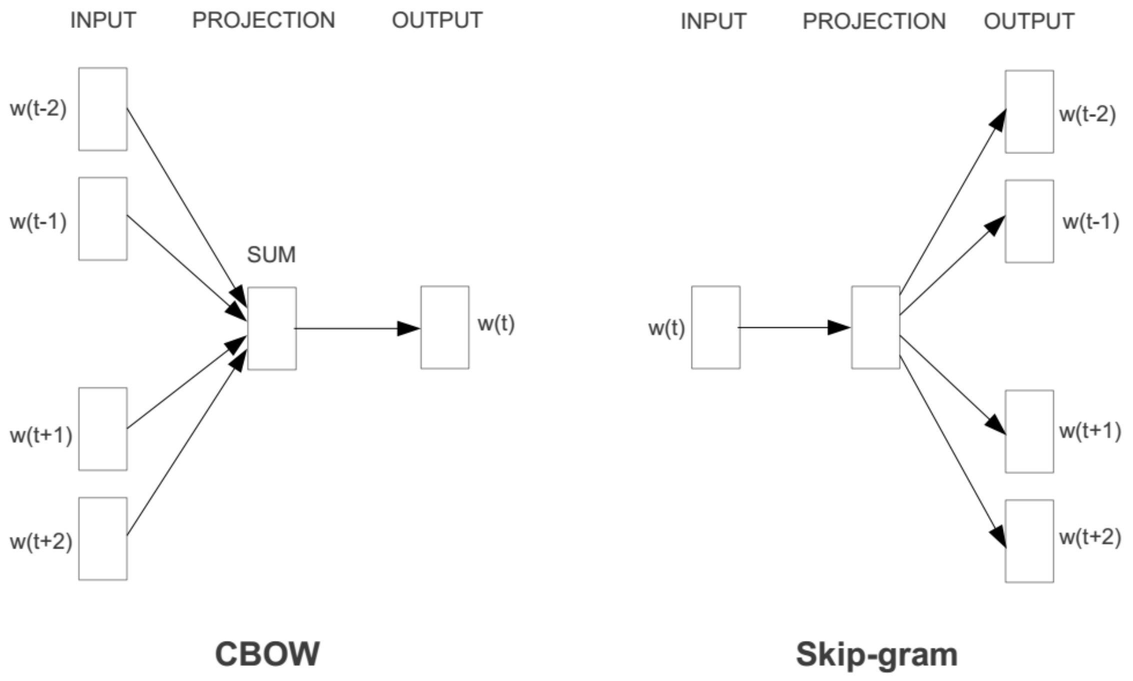
The man sat on the floor

Training samples : (on, man), (on, sat)
(on, the), (on, floor)

word2vec

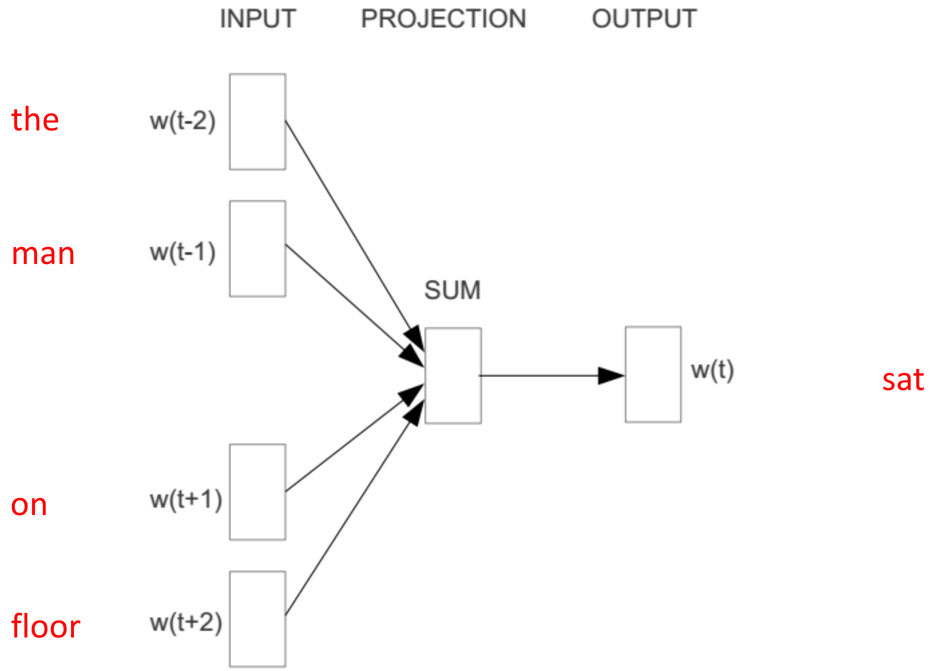
2 basic neural network models:

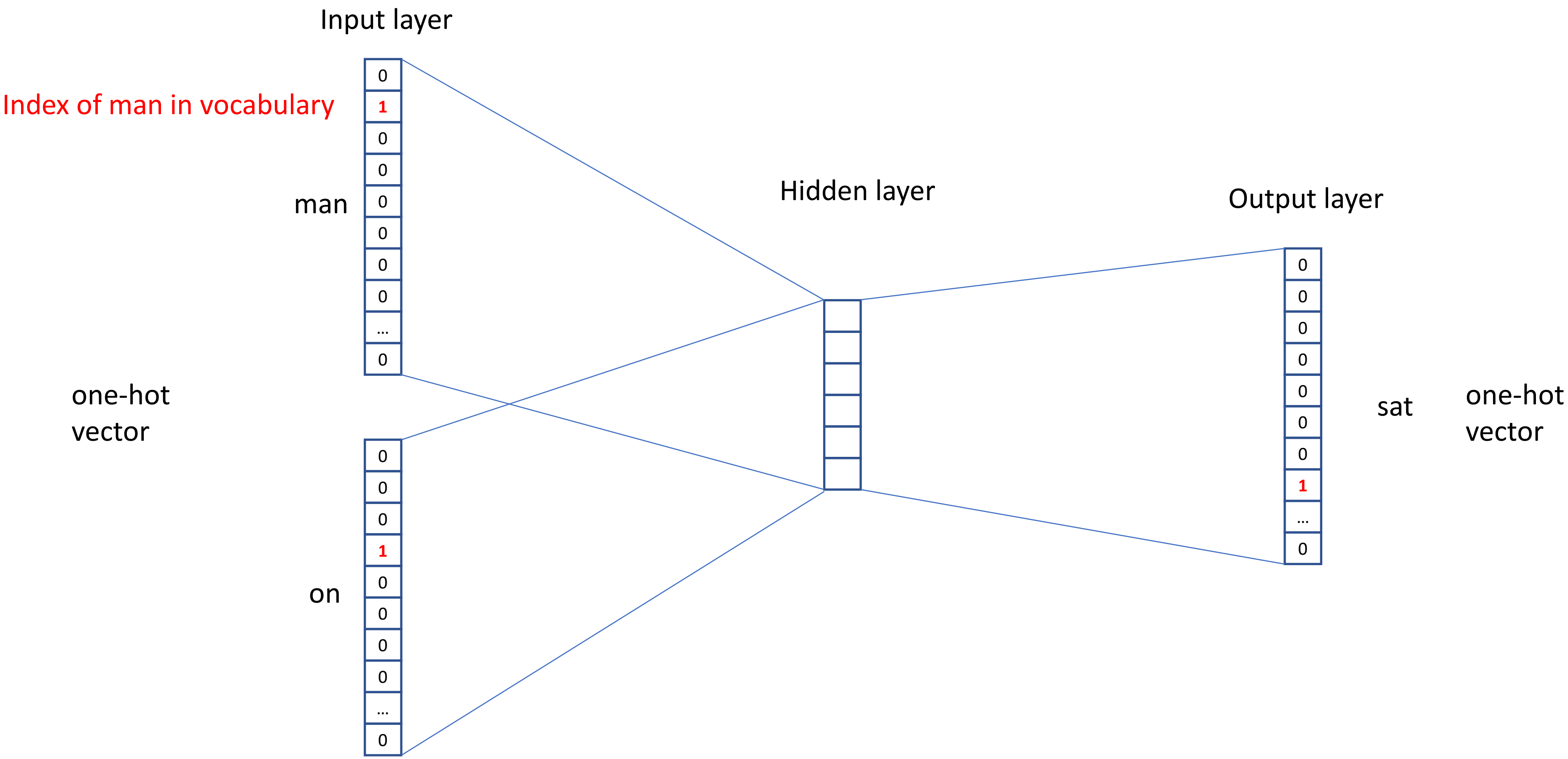
- **Continuous Bag of Word (CBOW)**: Pr(target|context)
- **Skip-gram (SG):** Pr(context|target).

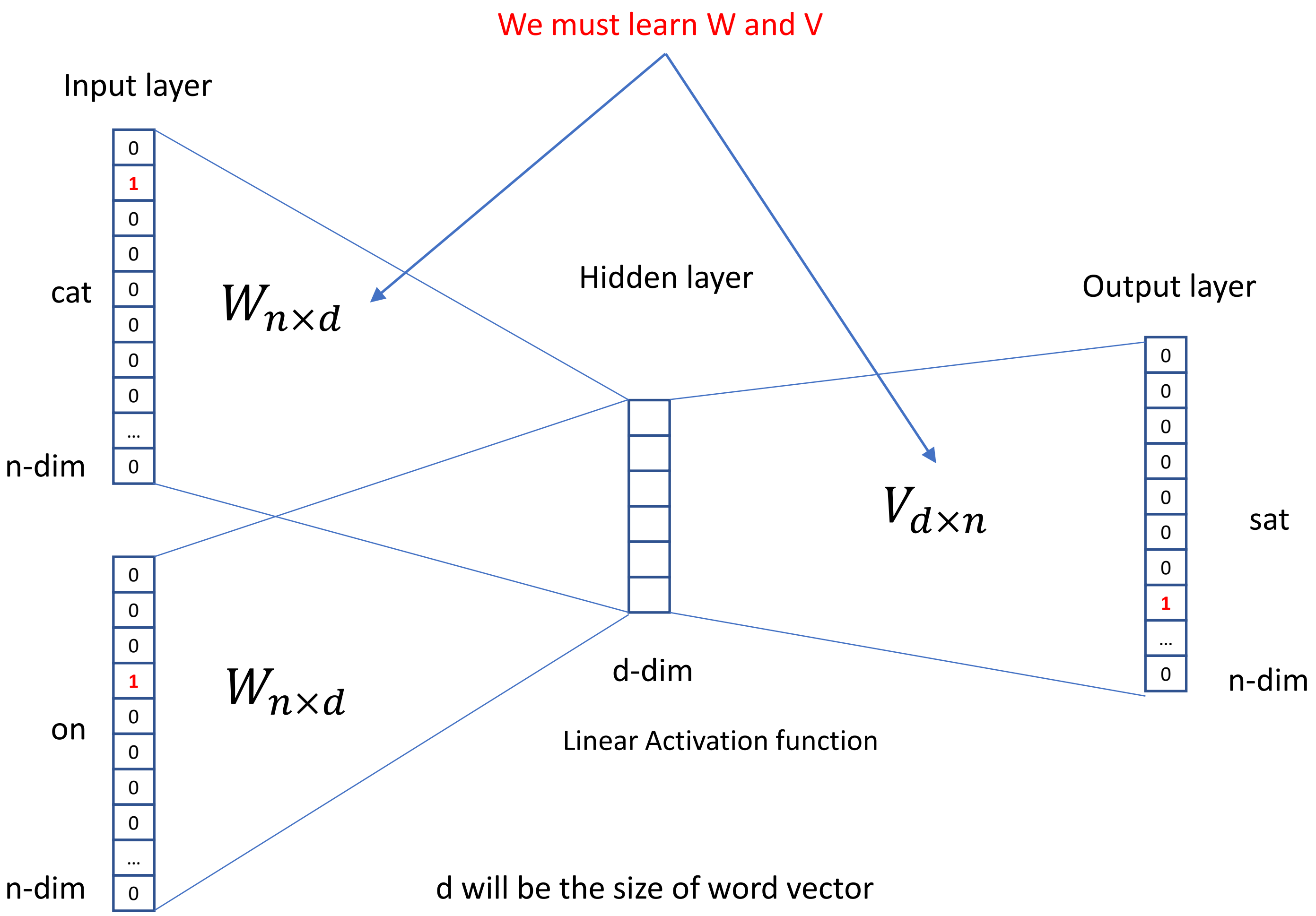


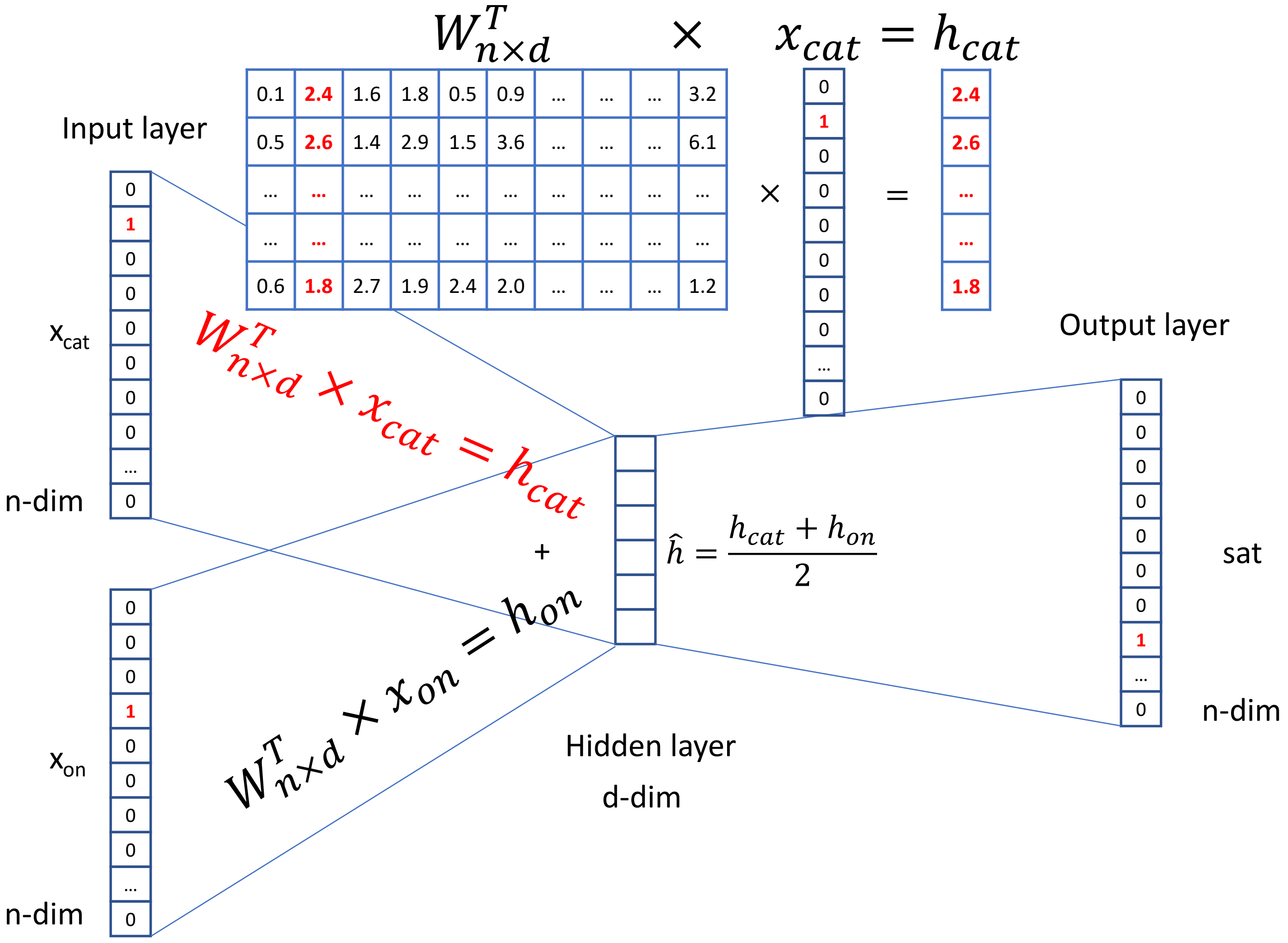
Word2vec – Continuous Bag of Word

- E.g. “The man sat on floor”
  - Window size = 2



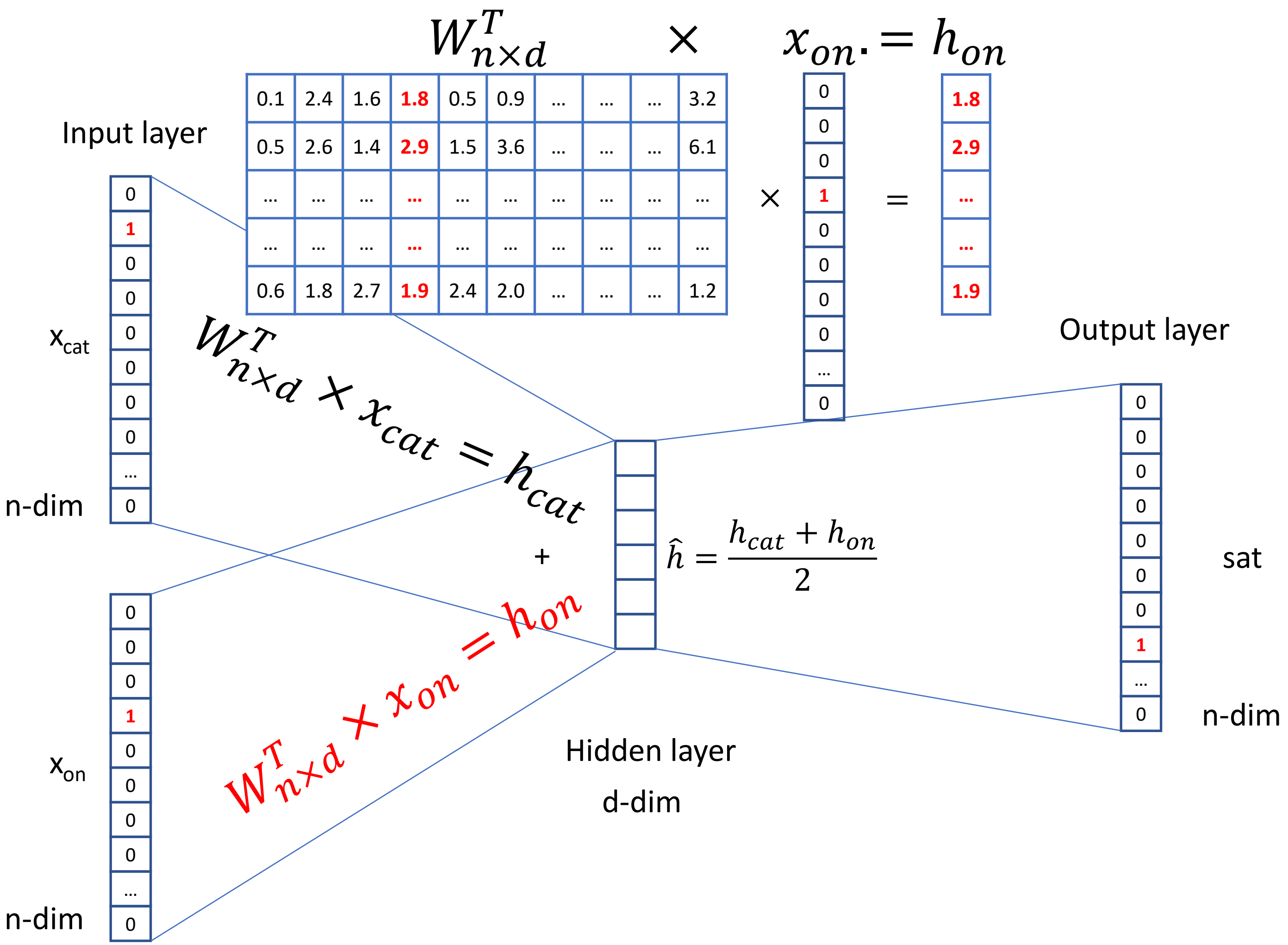

Input layer
Index of man in vocabulary
man
0
1
0
0
0
0
0
0
...
0
one-hot
vector
on
0
0
0
1
0
0
0
0
...
0
Hidden layer
Output layer
0
0
0
0
0
0
0
1
...
0
sat
one-hot
vector


We must learn W and V
Input layer
cat
n-dim
$W_{n \times d}$
Hidden layer
Output layer
$V_{d \times n}$
sat
n-dim
on
n-dim
$W_{n \times d}$
d-dim
Linear Activation function
d will be the size of word vector
0
1
0
0
0
0
0
0
...
0
0
0
0
1
0
0
0
0
...
0
0
0
0
0
0
0
0
1
...
0

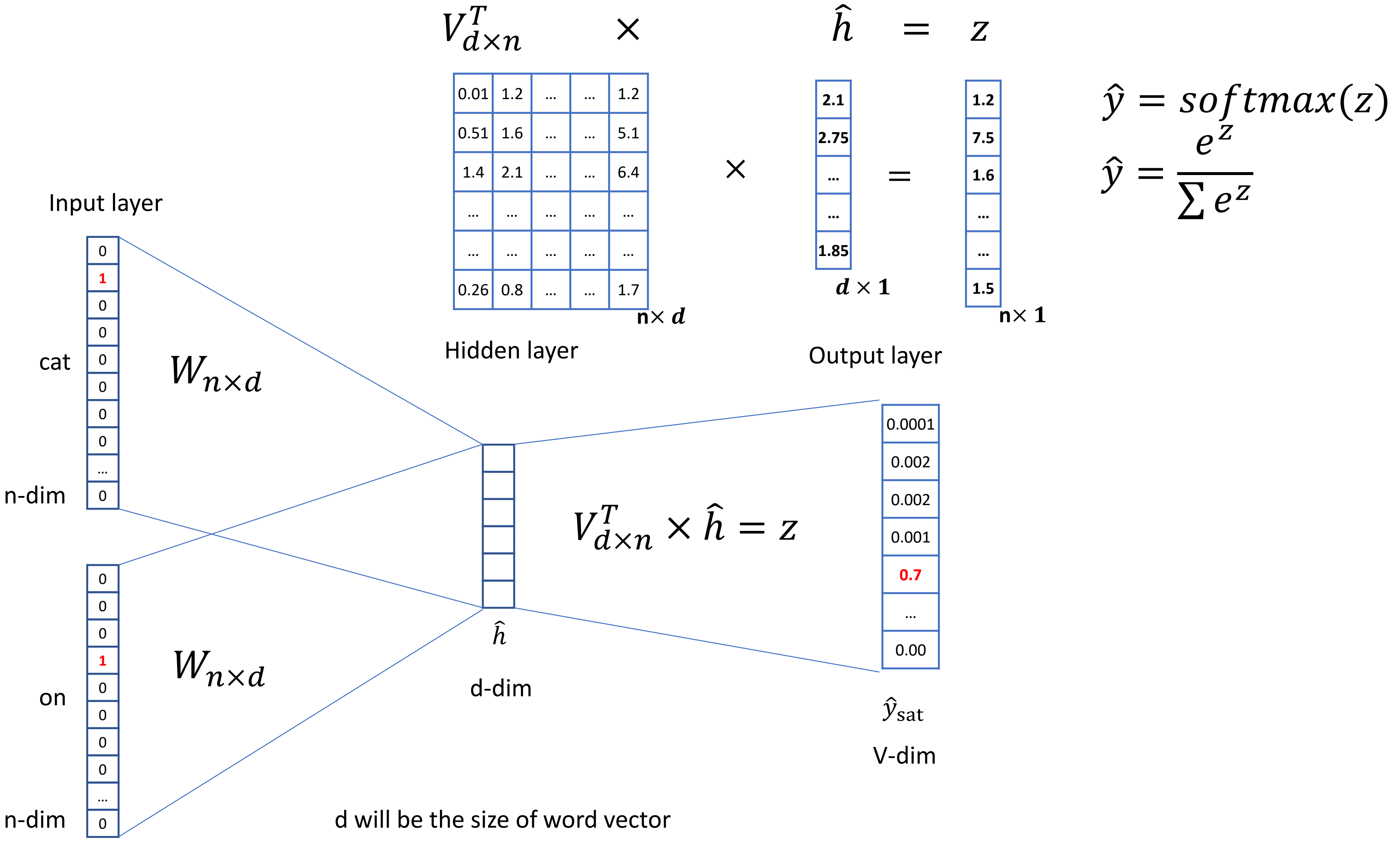


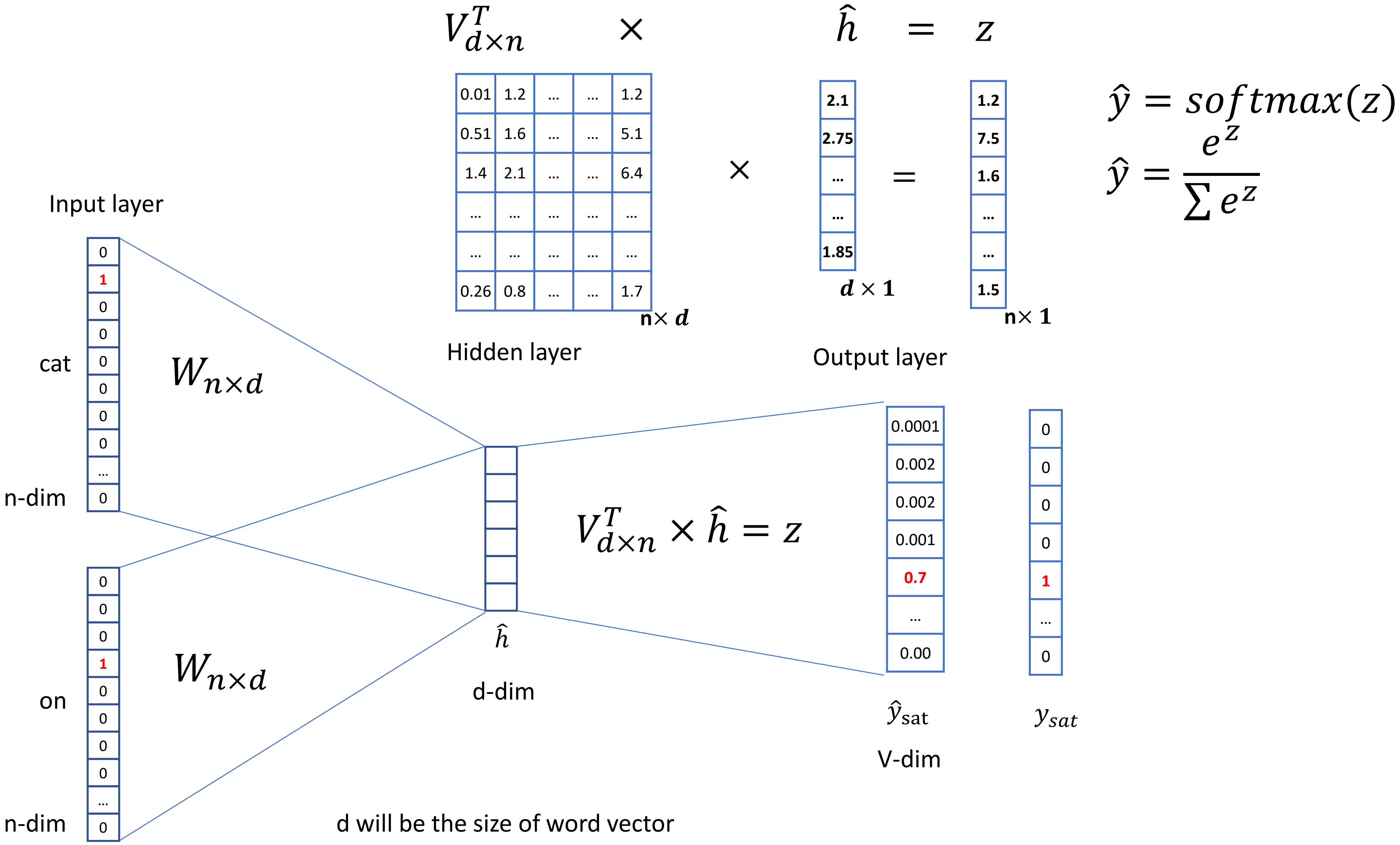
0.1	2.4	1.6	1.8	0.5	0.9	...	...	...	3.2
0.5	2.6	1.4	2.9	1.5	3.6	...	...	...	6.1
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
0.6	1.8	2.7	1.9	2.4	2.0	...	...	...	1.2

$\times$ [0, 1, 0, 0, 0, 0, 0, 0, ..., 0] $=$ [2.4, 2.6, ..., ..., 1.8]

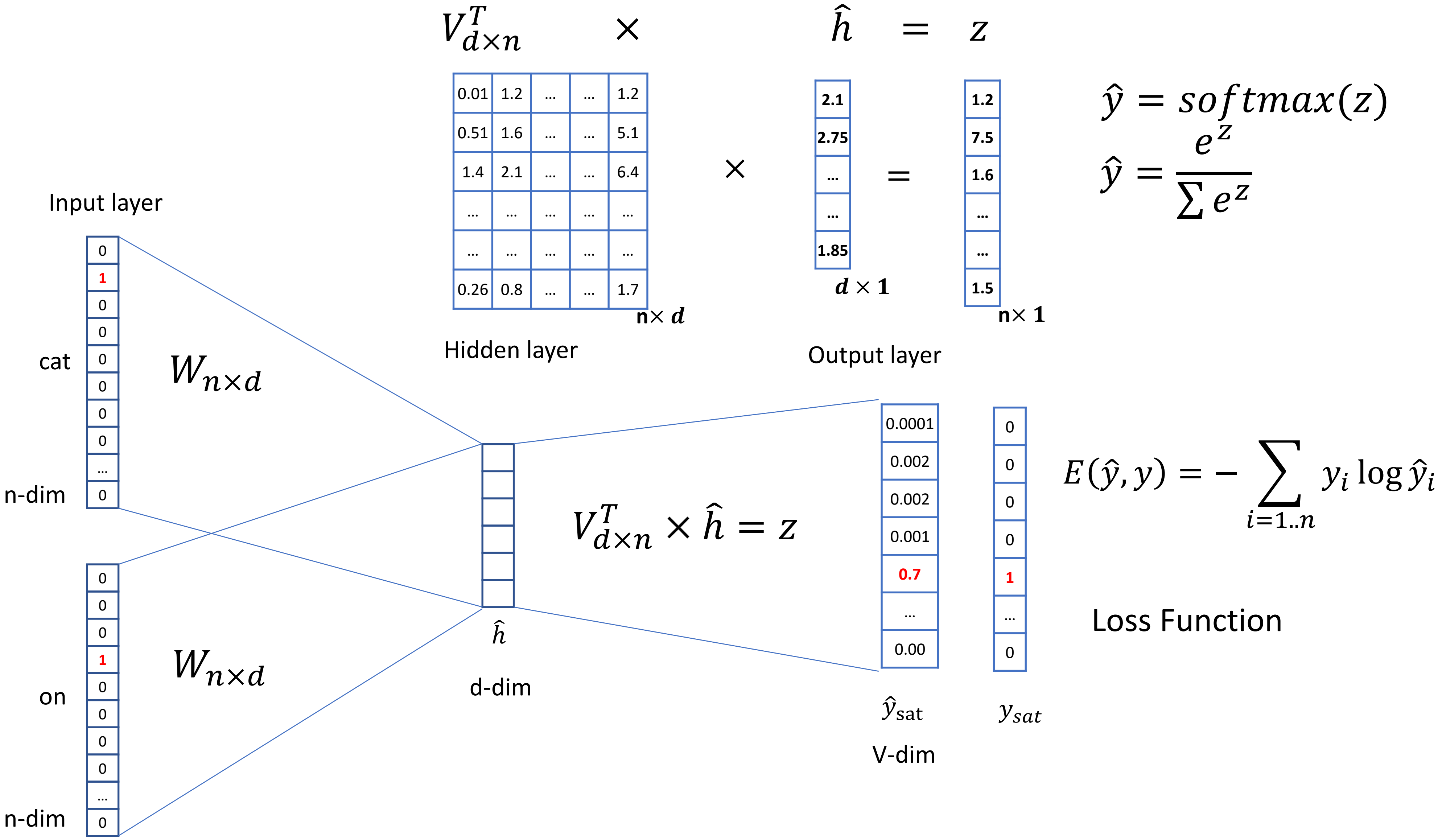


$W^T_{n\times d}$ × $x_{on}.= h_{on}$
Input layer
0.1 2.4 1.6 1.8 0.5 0.9 … … … 3.2
0.5 2.6 1.4 2.9 1.5 3.6 … … … 6.1
0.6 1.8 2.7 1.9 2.4 2.0 … … … 1.2
0 0 0 1 0 0 0 0 … 0
1.8 2.9 … … 1.9
x_{cat}
n-dim
$W^T_{n\times d} \times x_{cat} = h_{cat}$
+
$W^T_{n\times d} \times x_{on} = h_{on}$
x_{on}
n-dim
$\hat{h} = \frac{h_{cat} + h_{on}}{2}$
Hidden layer
d-dim
Output layer
sat
n-dim

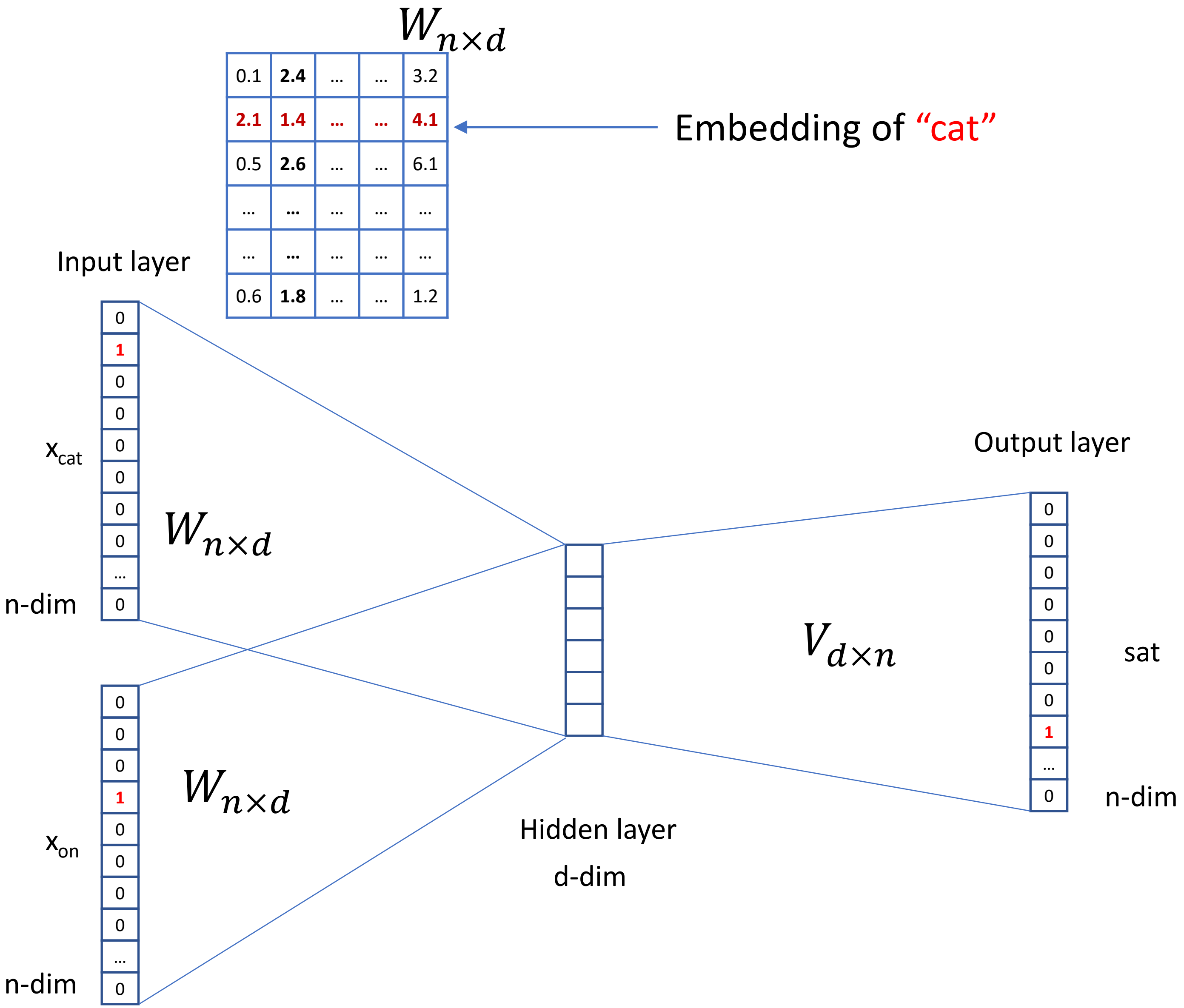

$V^T_{d\times n} \times \hat{h} = z$
0.01 1.2 … … 1.2
0.51 1.6 … … 5.1
1.4 2.1 … … 6.4
0.26 0.8 … … 1.7
n× d
Hidden layer
2.1
2.75
1.85
d × 1
1.2
7.5
1.6
1.5
n× 1
Output layer
$\hat{y} = softmax(z)$
$\hat{y} = \frac{e^z}{\sum e^z}$
Input layer
cat
n-dim
on
n-dim
$W_{n\times d}$
$W_{n\times d}$
$V^T_{d\times n} \times \hat{h} = z$
$\hat{h}$
d-dim
0.0001
0.002
0.002
0.001
0.7
0.00
$\hat{y}_{\text{sat}}$
V-dim
d will be the size of word vector



$V^T_{d\times n} \times \hat{h} = z$
0.01 1.2 ... ... 1.2
0.51 1.6 ... ... 5.1
1.4 2.1 ... ... 6.4
0.26 0.8 ... ... 1.7
n× d
2.1 2.75 ... ... 1.85
$d \times 1$
1.2 7.5 1.6 ... ... 1.5
n× 1
$\hat{y} = softmax(z)$
$\hat{y} = \frac{e^z}{\sum e^z}$
Hidden layer
Output layer
Input layer
cat
n-dim
on
n-dim
$W_{n\times d}$
$W_{n\times d}$
$\hat{h}$
d-dim
$V^T_{d\times n} \times \hat{h} = z$
0.0001 0.002 0.002 0.001 0.7 ... 0.00
0 0 0 0 1 ... 0
$\hat{y}_{sat}$
y_{sat}
V-dim
d will be the size of word vector



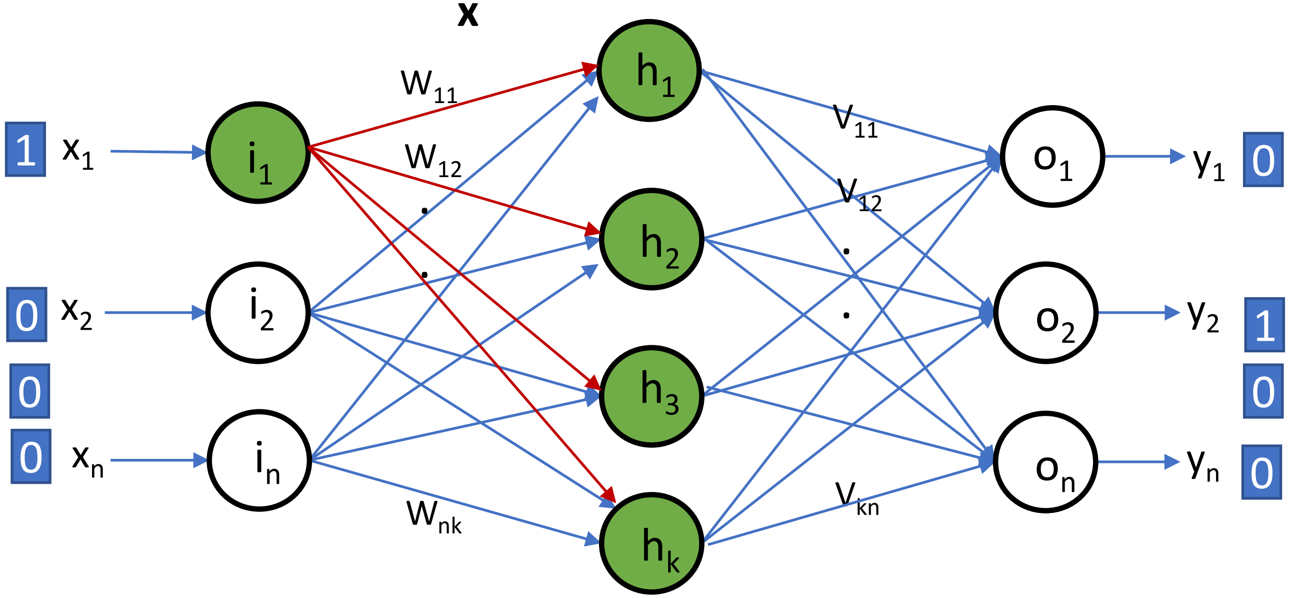
$V^T_{d\times n} \times \hat{h} = z$
Hidden layer
$n\times d$
Output layer
$d \times 1$
$n\times 1$
$\hat{y} = softmax(z)$
$\hat{y} = \frac{e^z}{\sum e^z}$
Input layer
cat
n-dim
on
n-dim
$W_{n\times d}$
$W_{n\times d}$
$\hat{h}$
d-dim
$V^T_{d\times n} \times \hat{h} = z$
$\hat{y}_{sat}$
y_{sat}
V-dim
$E(\hat{y}, y) = -\sum_{i=1..n} y_i \log \hat{y}_i$
Loss Function



We can consider either W or V as the word's representation. Or even take the average.

$$\underbrace{\begin{bmatrix} 0.1 & 2.4 & 1.6 & 1.8 & 0.5 & 0.9 & \dots & \dots & \dots & 3.2 \\ 0.5 & 2.6 & 1.4 & 2.9 & 1.5 & 3.6 & \dots & \dots & \dots & 6.1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.6 & 1.8 & 2.7 & 1.9 & 2.4 & 2.0 & \dots & \dots & \dots & 1.2 \end{bmatrix}_{k\mathrm{x}n}}_{W^T} \cdot \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 0.1 \\ 0.5 \\ \dots \\ \dots \\ 0.6 \end{bmatrix}}_{\mathbf{h}}$$

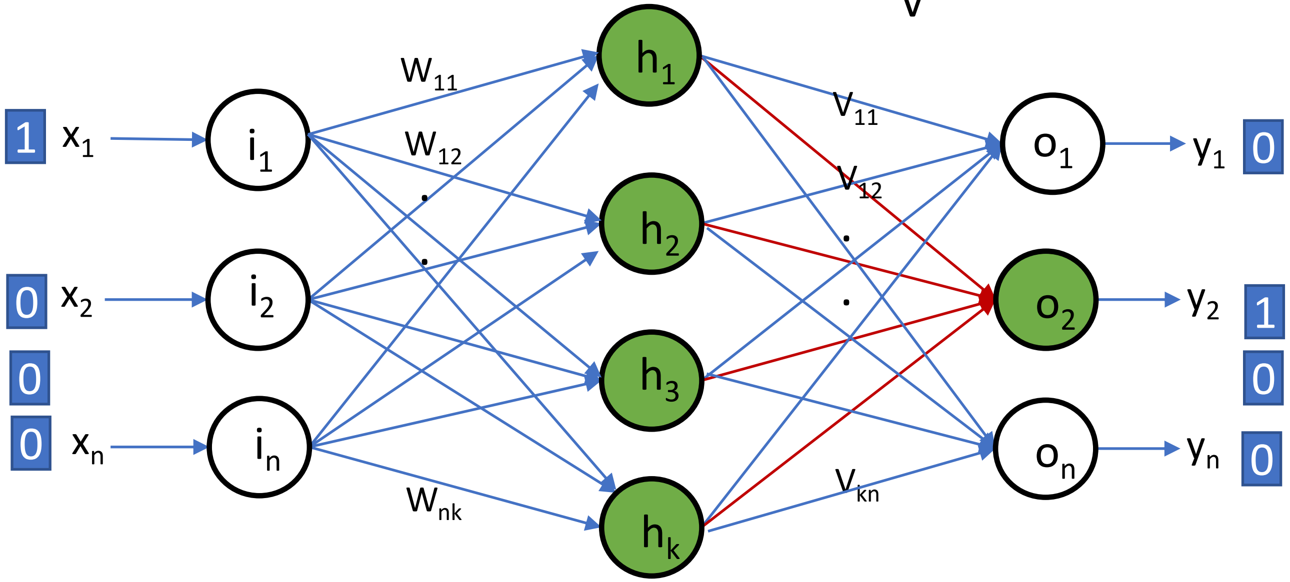
Embedding of X_1

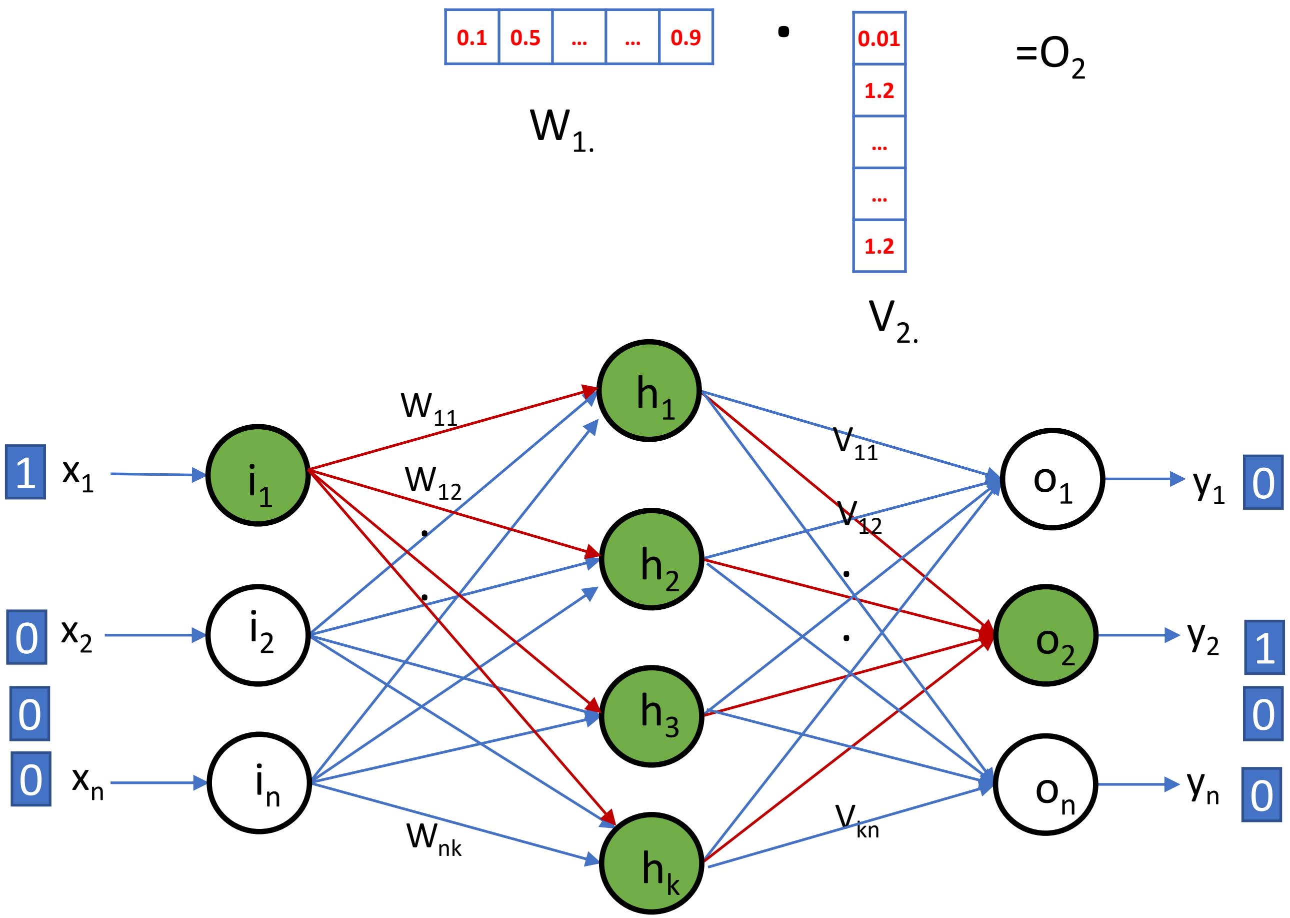


0.1
0.5
...
...
0.6

0.01	0.51	1.4	...	...	...	3.2
1.2	1.6	2.1	...	...	...	6.1
...	...	...	...	...	...	...
...	...	...	...	...	...	...
1.2	5.1	6.4	...	...	...	1.2

V




0.1
0.5
...
...
0.9
W1.
0.01
1.2
...
...
1.2
V2.
=O2
1
x1
0
x2
0
0
xn
i1
i2
in
W11
W12
Wnk
h1
h2
h3
hk
V11
V12
Vkn
o1
o2
on
y1
y2
yn
0
1
0
0