Words Embedding

IA Frameworks

Brendan Guillouet - 02/11/2020

Institut National des Sciences Appliquées de Toulouse

Table of contents

FRAMEWORKS

ML Python Libraries

scikit learn
Keras
gensim
surprise
Gym
TensorFlow
NLTK

Python Environment

PC
jupyter
ANACONDA

Viz' Python Libraries

matplotlib
Version 3.1.1
seaborn
plotly

Framework & Tool

docker
Google Cloud Platform
git

Introduction

Motivations

Problem of vectorisation's method: No relation between words!

What would be the *perfect* features space?

	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97

How to build this representation ?

Word2Vec

Word2Vec - Mikolov et al. [2013a]

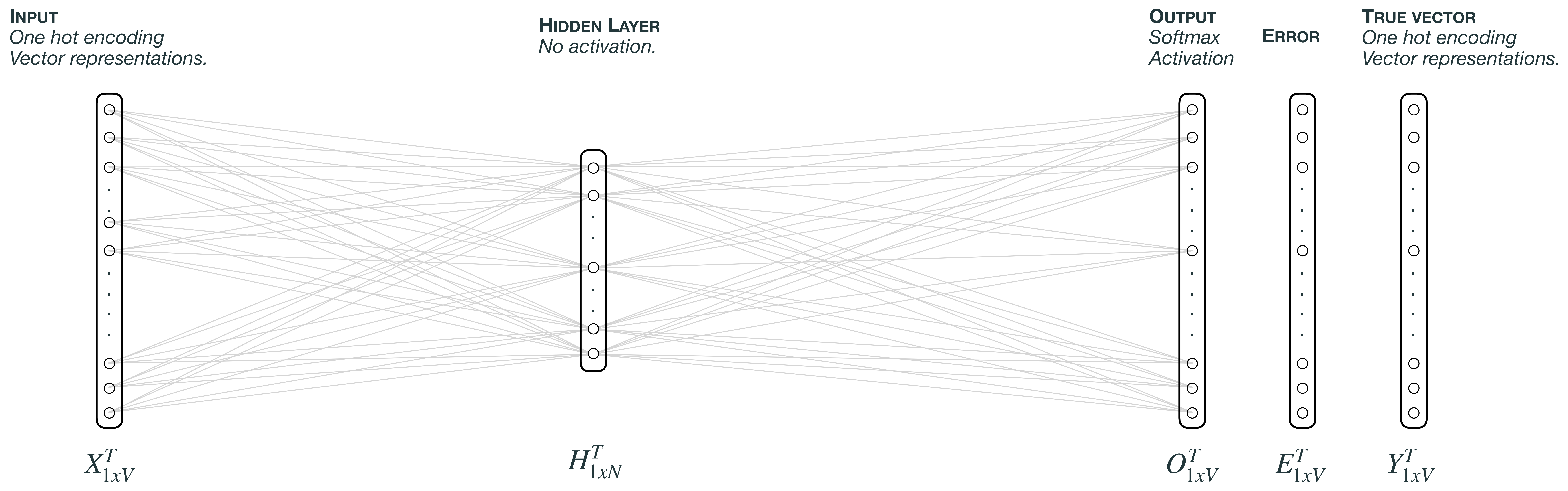
“ la langue française a des règles de grammaire compliquées ” window=1

context target context

Word2Vec - Mikolov et al. [2013a]

“ la langue française a des règles de grammaire compliquées ” window=1

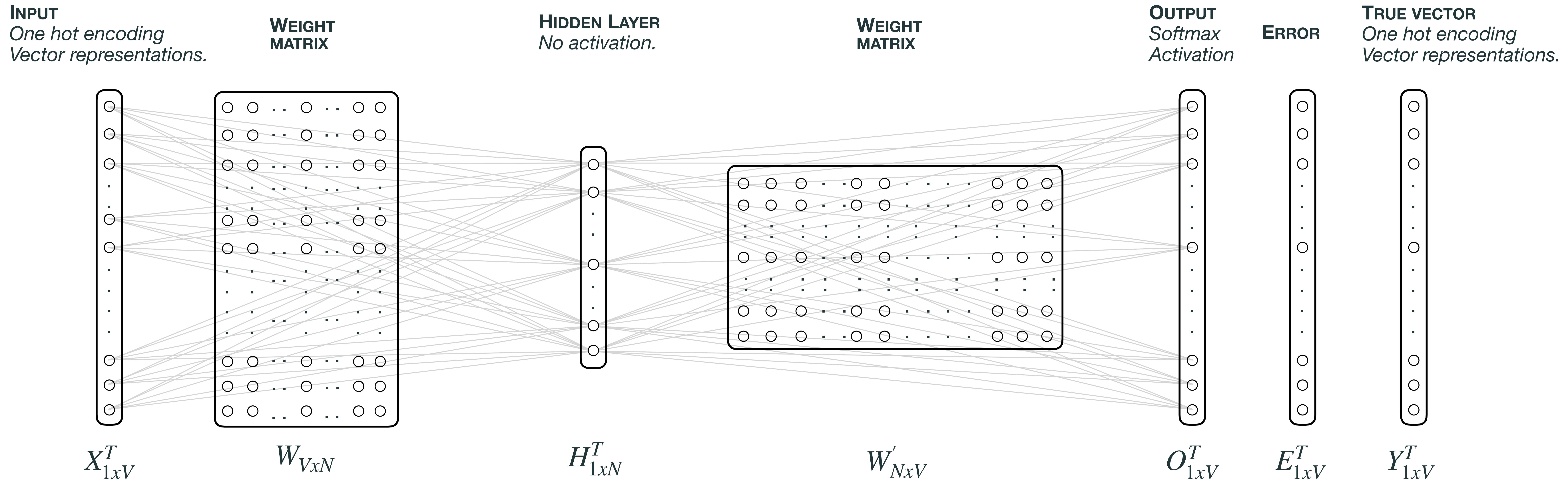
context target context



Word2Vec - Mikolov et al. [2013a]

“ la langue française a des règles de grammaire compliquées ” window=1

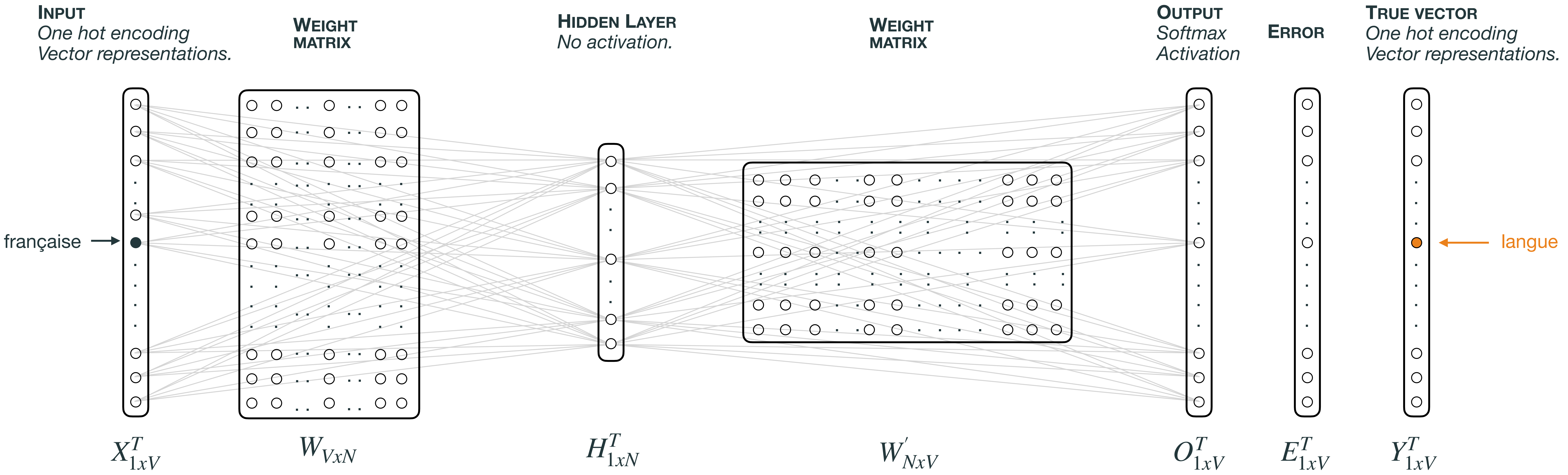
context target context



Word2Vec - Mikolov et al. [2013a]

“ la langue française a des règles de grammaire compliquées ” window=1

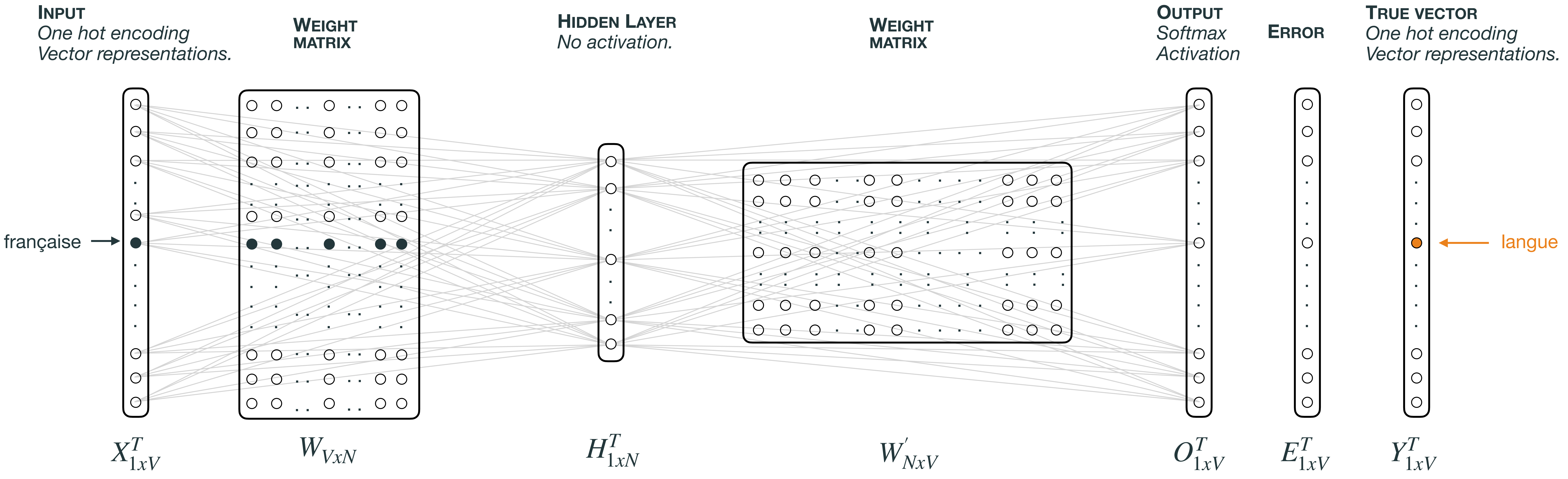
context target context



WORD2VEC - MIKOLOV ET AL. [2013A]

“ la langue française a des règles de grammaire compliquées ” window=1

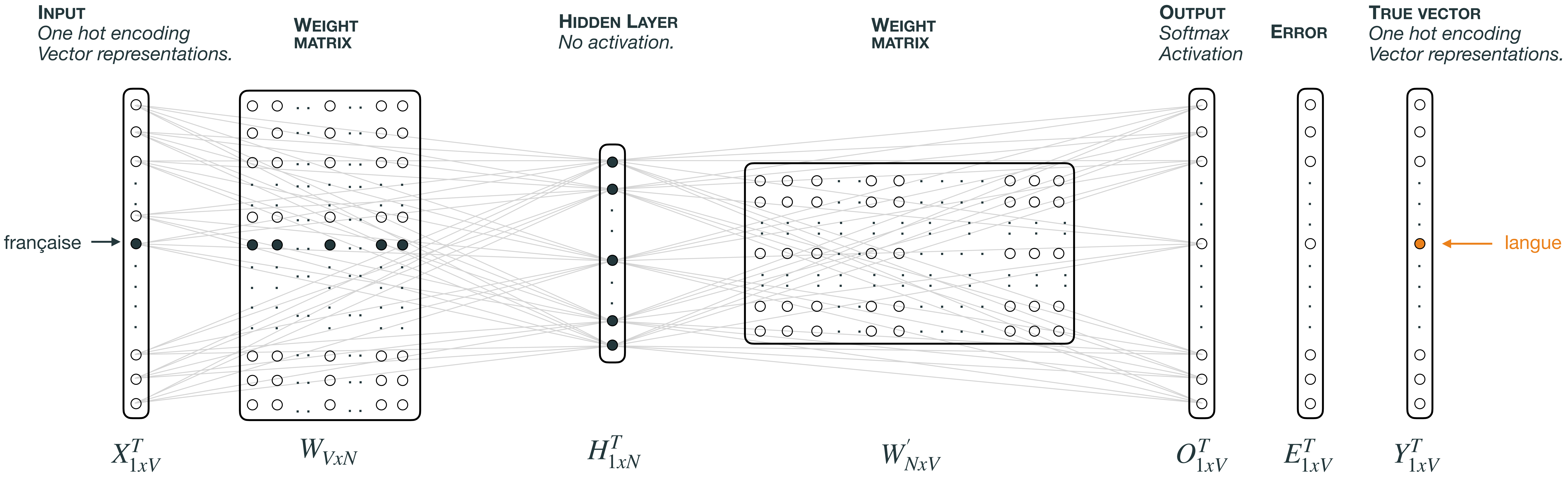
context target context



WORD2VEC - MIKOLOV ET AL. [2013A]

“ la langue française a des règles de grammaire compliquées ” window=1

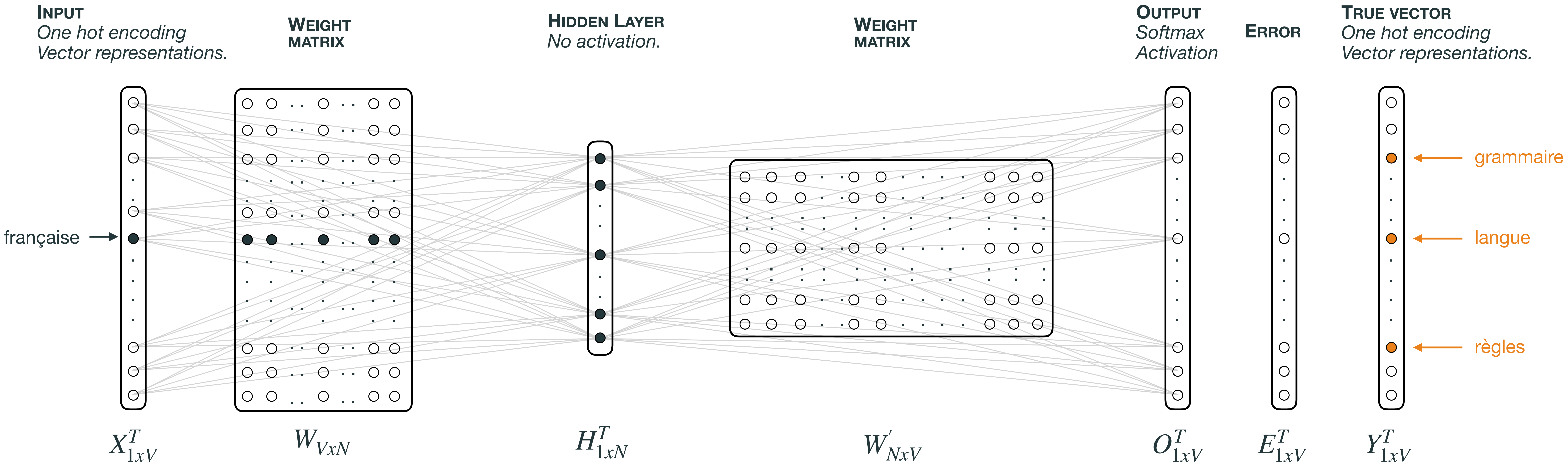
context target context



Word2Vec - Mikolov et al. [2013a]

“ la langue française a des règles de grammaire compliquées ” window=5

context target context



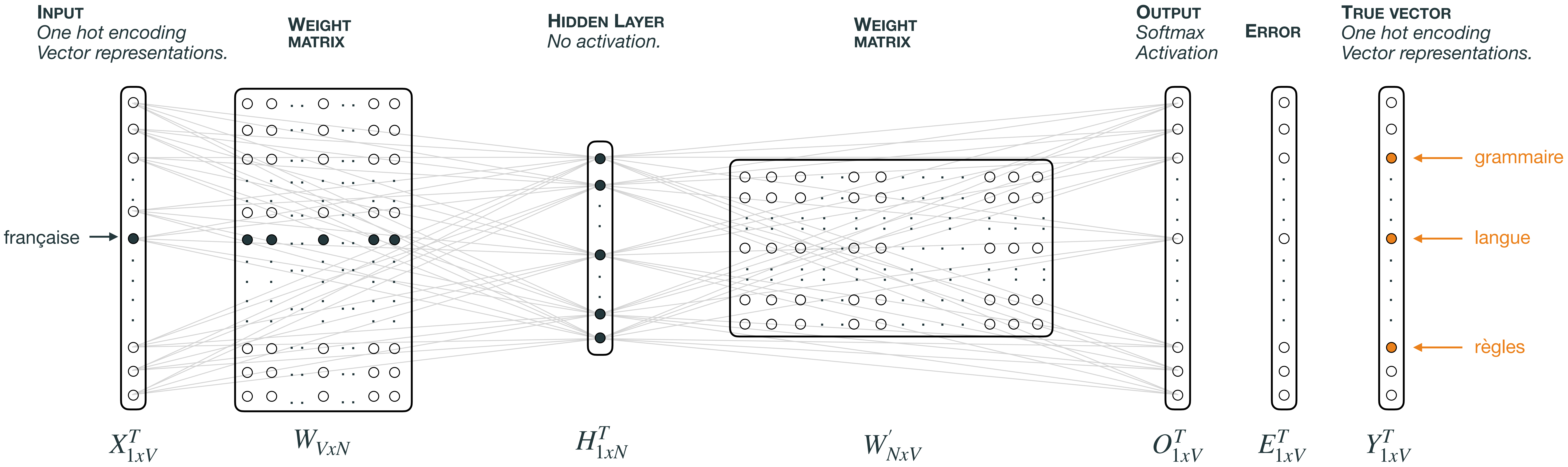
Word2Vec - Mikolov et al. [2013a]

" la langue française a des règles de grammaire compliquées " window=5

context — target — context

" la langue allemande a des règles de grammaire compliquées " window=5

context — target — context



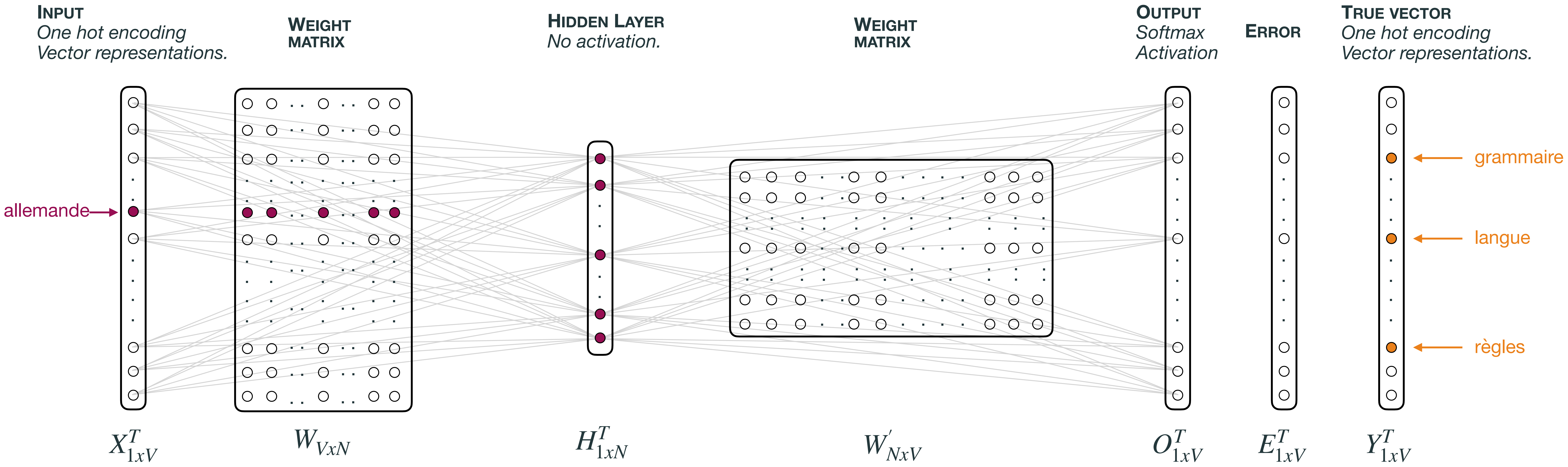
Word2Vec - Mikolov et al. [2013a]

“ la langue française a des règles de grammaire compliquées ” window=5

context — target — context

“ la langue allemande a des règles de grammaire compliquées ” window=5

context — target — context



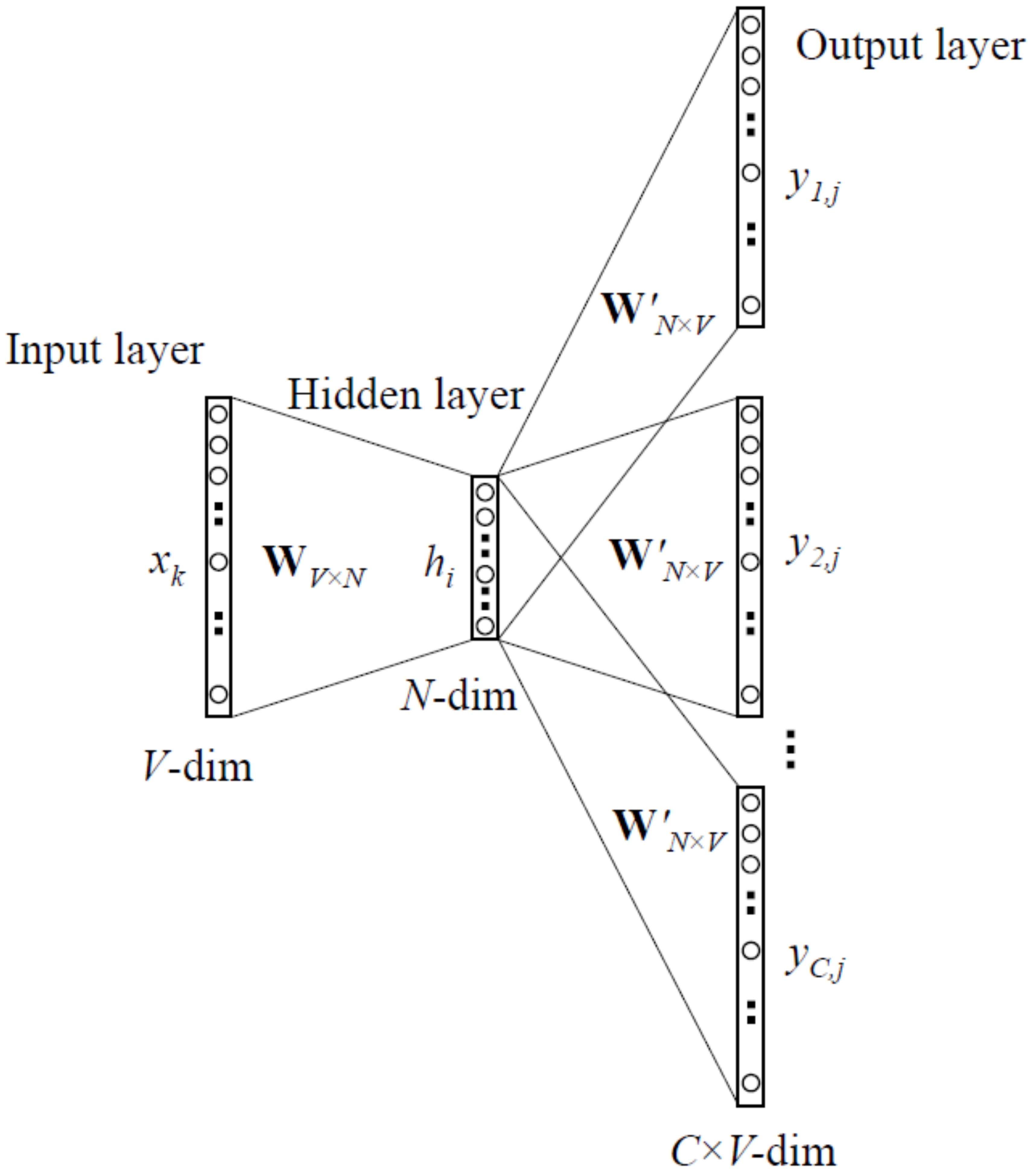
Word2Vec - Generalties

- **No** activation function (or linear activation) on the hidden layer.
- The **loss function** can be either cross-entropy or log likelihood of a word knowing the context.
- Activation function of output layer is softmax function.

- 2 versions:
    - Continuous Bag Of Word (CBOW)
    - Skip-gram

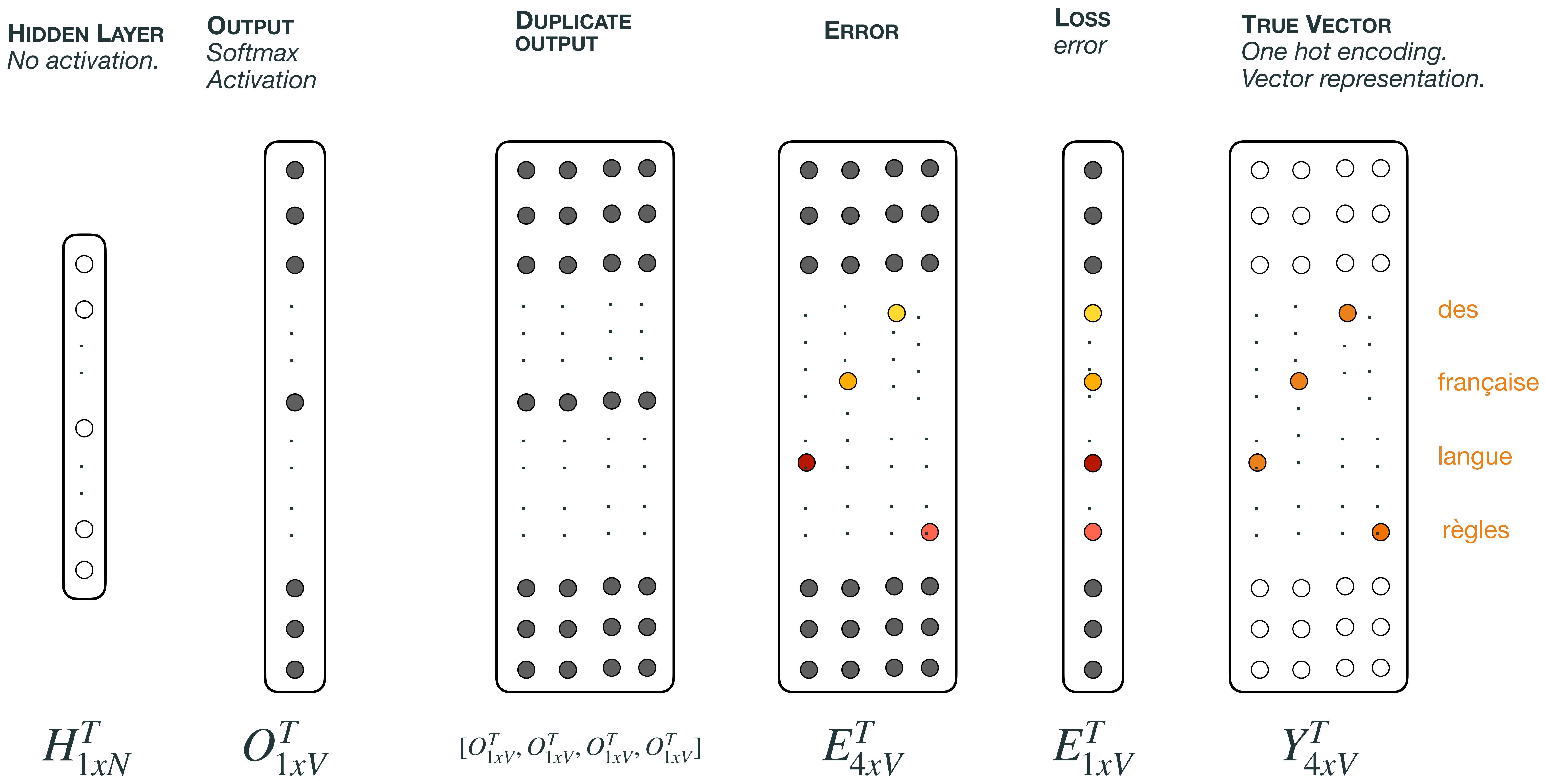
Word2Vec - Skip-Gram

Input	Output
la	langue, française
langue	la, française, a
française	la, langue, a, des
a	langue, française, des, règles
des	française, a, règles, de
règles	a, des, de, grammaire
de	des, règles, grammaire, compliqués
grammaire	règles, de, compliquées
compliqués	de, grammaire



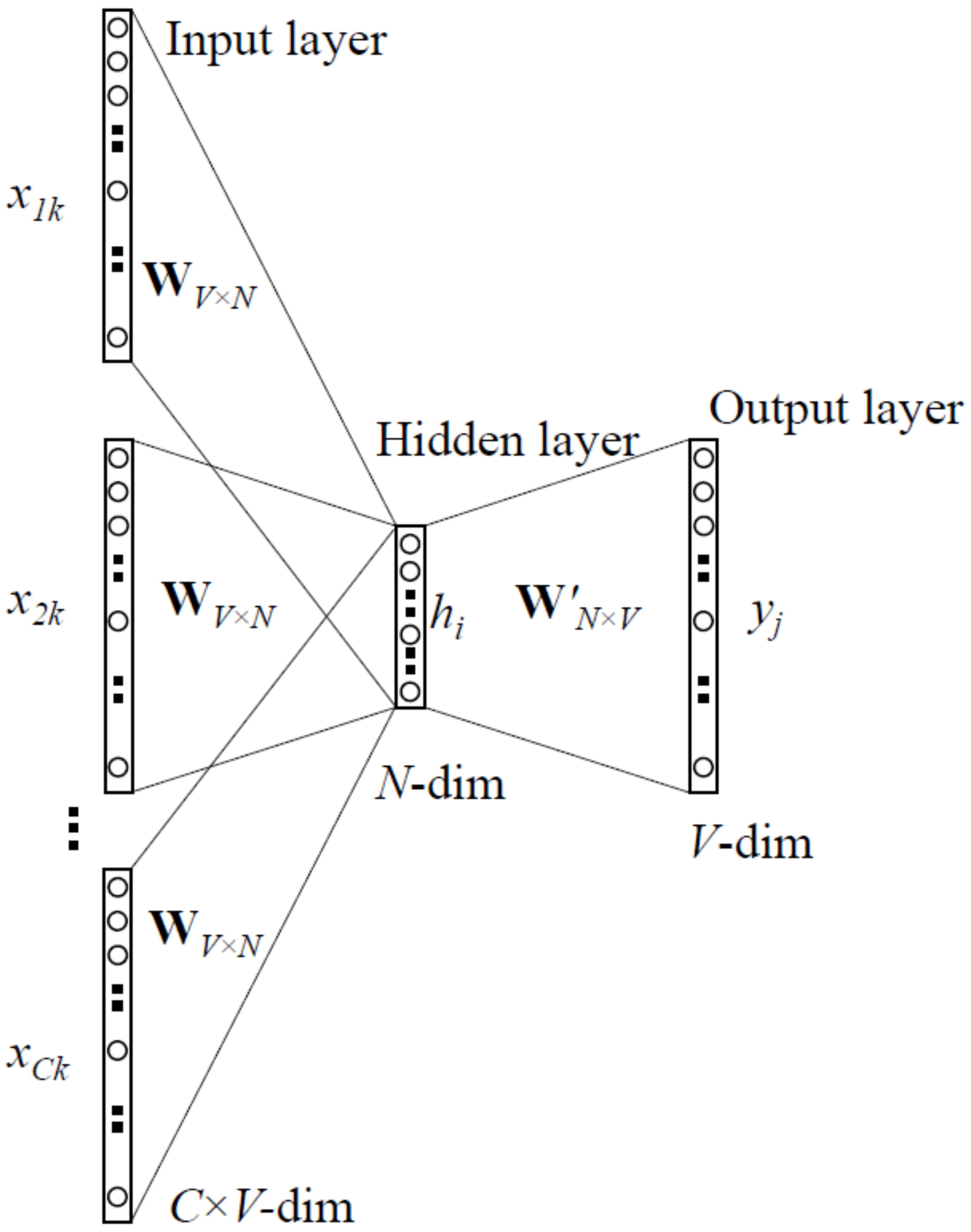
Word2Vec - Skip-Gram

Previous layer are the same.

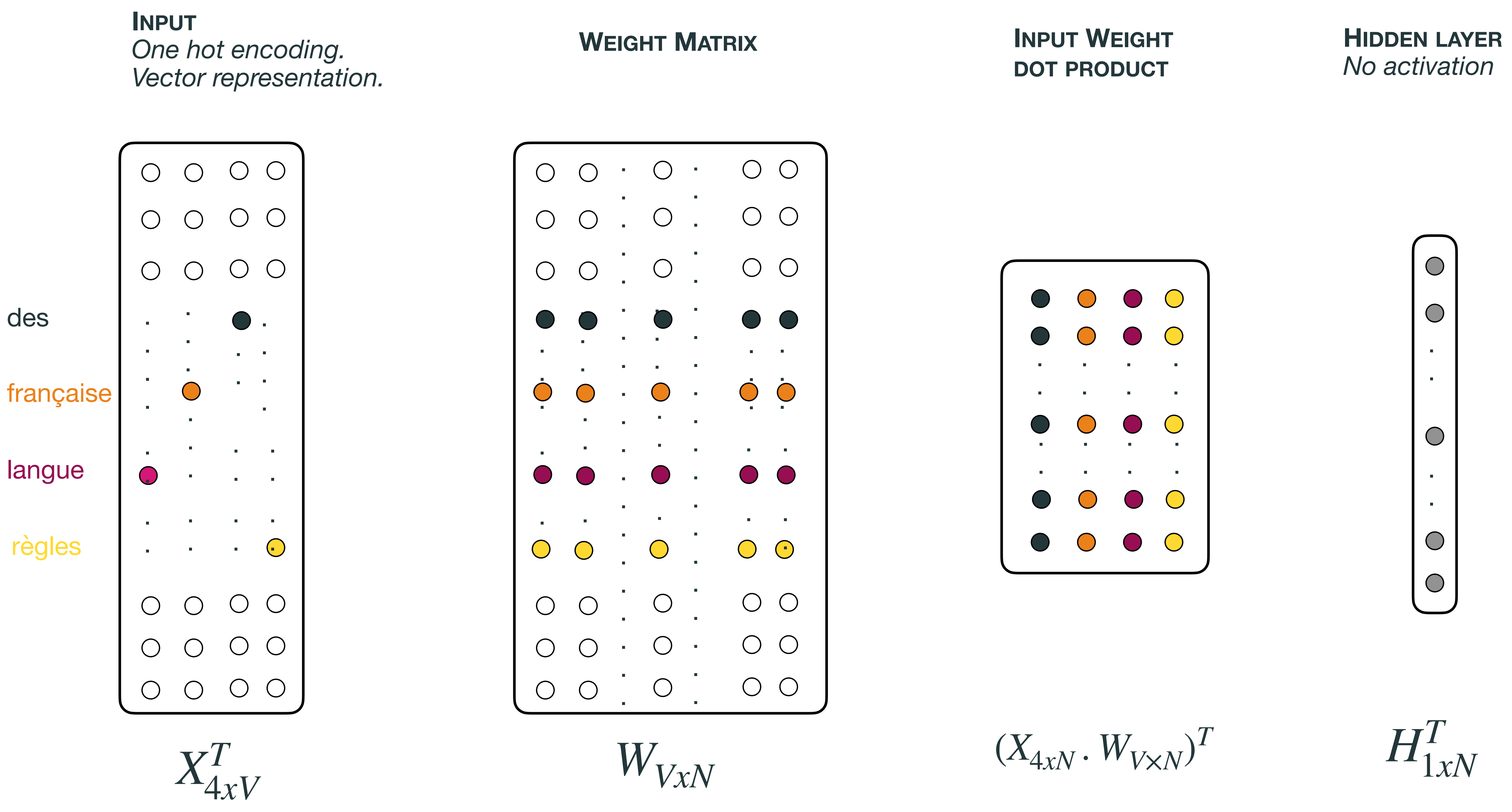


Word2Vec - Continuous Bag of Words (CBOW)

Input	Output
langue, française	la
la, française, a	langue
la, langue, a, des	française
langue, française, des, règles	a
française, a, règles, de	des
a, des, de, grammaire	règles
des, règles, grammaire,	de
règles, de, compliquées	grammaire
de, grammaire	compliqués



Word2Vec - Continuous Bag of Words (CBOW)



Other layers are the same.

Negative Sampling - Mikolov and al. [2013b]

- Default activation function: Softmax

$$P(Y_j/X_i) = \frac{exp(W_{i,:} \, . \, W'_{:,j})}{\sum_{k=1}^{V} exp(W_{k,:} \, . \, W'_{:,j})}$$

Problem ->each neurons is updated at each iteration.

- Negative sampling activation function:

$$P(T = 1/Y_j, X_i) = \frac{1}{1 - exp(W_{i,:} \cdot W'_{:,j})}$$

Limited number of neurons updated at each iteration.

Input	output	target
française	langue	1
française	mobylette	0
française	caramel	0
française	pudding	0
française	bateau	0

Properties

	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97

$$e_{king} - e_{man} + e_{woman} = e_{pred} \approx e_{queen}$$

$$\begin{bmatrix} -0.95 \\ 0.93 \\ 0.7 \\ 0.02 \end{bmatrix} \quad \begin{bmatrix} -1 \\ 0.01 \\ 0.03 \\ 0.04 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0.02 \\ 0.02 \\ 0.01 \end{bmatrix} \quad \begin{bmatrix} 1.05 \\ 0.94 \\ 0.69 \\ -0.01 \end{bmatrix} \quad \begin{bmatrix} 0.97 \\ 0.95 \\ 0.69 \\ 0.01 \end{bmatrix}$$

FastText

FastText - Mikolov and al. [2016]

- FastText is an extension of Word2Vec proposed by the same authors.
- The algorithm is the same.
- Words are not word but **subwords of n characters**. *Example:*
  - n=2. Apple = ap, pp, pl, le
  - n=3. Apple = app, ppl, pple.
  - n=4. Apple = appl, pple.

FastText - Nikolov and al. [2016]

- It's possible to compose different level of subwords for one training. *(n_min=2, n_max=4)*
- Embedding of a word is the embedding of all its subwords.
- An embedding of a word with few occurence will be improved.
- An embedding of a word can exist event if it's not in the dataset but similar word are.
- Works ok for words embedding. Not for constructed language. (Playstation and Xbox will never been the same if never seen in the same context)
- On gensim with parameters *min_n* and *max_n* equal to 0 fastest is equivalent to word2vec

GLOVE

Glove - Pennington et al. [2014]

- Glove stands for **Glo**bal **Ve**ctor.
- Word2Vec is build on **local** properties of the words, Glove on **global** properties.
- It uses co-occurence matrix, example:

“ la langue française a des règles de grammaire compliquées ” window=1

“ la langue allemande a des règles de grammaire compliquées ” window=1

	A	Allemande	Compliquées	De	Des	Française	Grammaire	La	Langue	Règles
A	0	**1**	0	0	**2**	**1**	0	0	0	0
Allemande	**1**	0	0	0	0	0	0	0	**1**	0
Compliquées	0	0	0	0	0	0	**2**	0	0	0
De	0	0	0	0	0	0	**2**	0	0	**2**
Des	**2**	0	0	0	0	0	0	0	0	**2**
Française	**1**	0	0	0	0	0	0	0	**1**	0
Grammaire	0	0	**2**	**2**	0	0	0	0	0	0
La	0	0	0	0	0	0	0	0	**2**	0
Langue	0	**1**	0	0	0	**1**	0	**2**	0	0
Règles	0	0	0	**2**	**2**	0	0	0	0	0

Glove - Co-occurence matrix

How useful is this matrix ?

	A	Alleman	Compli	De	Des	Français	Gramm	La	Langue	Règles
A	0	**1**	0	0	**2**	**1**	0	0	0	0
Alleman	**1**	0	0	0	0	0	0	0	**1**	0
Compli	0	0	0	0	0	0	**2**	0	0	0
De	0	0	0	0	0	0	**2**	0	0	**2**
Des	**2**	0	0	0	0	0	0	0	0	**2**
Français	**1**	0	0	0	0	0	0	0	**1**	0
Gramm	0	0	**2**	**2**	0	0	0	0	0	0
La	0	0	0	0	0	0	0	0	**2**	0
Langue	0	**1**	0	0	0	**1**	0	**2**	0	0
Règles	0	0	0	**2**	**2**	0	0	0	0	0

- We have all the statistics between words for all the dataset !
- $P(j\,|\,i)$ is the probability that j appears in context of I

Notations

X_{ij} = # j appears in the context of j, example:

$$P(j\,|\,i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}}$$

Example

$$X_{compliquees,grammaire} = 2, P(compliquee/grammaire) = 1/2$$

$$X_{allemande,langue} = 1, P(allemande/langue) = 1/4$$

Glove - Co-occurence matrix

How to use this matrix ? (Example of original paper)

	$k =$ solid	$k =$ gas	$k =$ water	$k =$ (random)
$P(k\,\vert\,\text{ice})$	high	low	high	low
$P(k\,\vert\,\text{steam})$	low	high	high	low
$\frac{P(k\,\vert\,\text{ice})}{P(k\,\vert\,\text{steam})}$	>1	<1	~1	~1

- Very useful properties.
- But vector of the co-occurence matrix is too big.
- Solution: Build word vectors that fulfil these properties!

Glove - Model

Find F so that

$$F(w_i, w_j, \hat{w}_k) = \frac{P(k \mid i)}{P(k \mid j)}$$

Where, w_i is the vector representation of word i.

There is an infinite solution for F, so let us define more desiderata for this function.

Vector spaces are linear structure. Difference is the most natural way to compare elements (here only w_i and w_j)

$$F(w_i - w_j, \hat{w}_k) = \frac{P(k \mid i)}{P(k \mid j)}$$

F can be quite complicated. In order to not losing the linear structure of the argument. Let's apply F on the dot product of the argument.

$$F((w_i - w_j)^T \cdot \hat{w}_k) = \frac{P(k \mid i)}{P(k \mid j)}$$

Glove - Model

$$F((w_i - w_j)^T \cdot \hat{w}_k) = \frac{P(k\,|\,i)}{P(k\,|\,j)}$$

For that let us require that F is an homomorphism between the groups $(\mathbb{R}, +)$ and $(\mathbb{R}_{>0}, \times)$, i.e.

$$F(w_i^T \cdot \hat{w}_k - w_j^T \cdot \hat{w}_k) = \frac{F(w_i^T \cdot \hat{w}_k)}{F(w_j^T \cdot \hat{w}_k)}$$

Hence:

$$F(w_i^T \cdot \hat{w}_k) = P(k\,|\,i) = \frac{X_{ik}}{X_i}$$

Which means that $F = e$ is a solution !

Glove - Model

The equation can the be re-written:

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

The equation is not symmetric because of $log(X_i)$. But it's independent of k so we can replace it by a bias b_i and adding a bias $\hat{b}_k$ to restore the symmetry

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

Hence, word vector can be built optimising the following cost function:

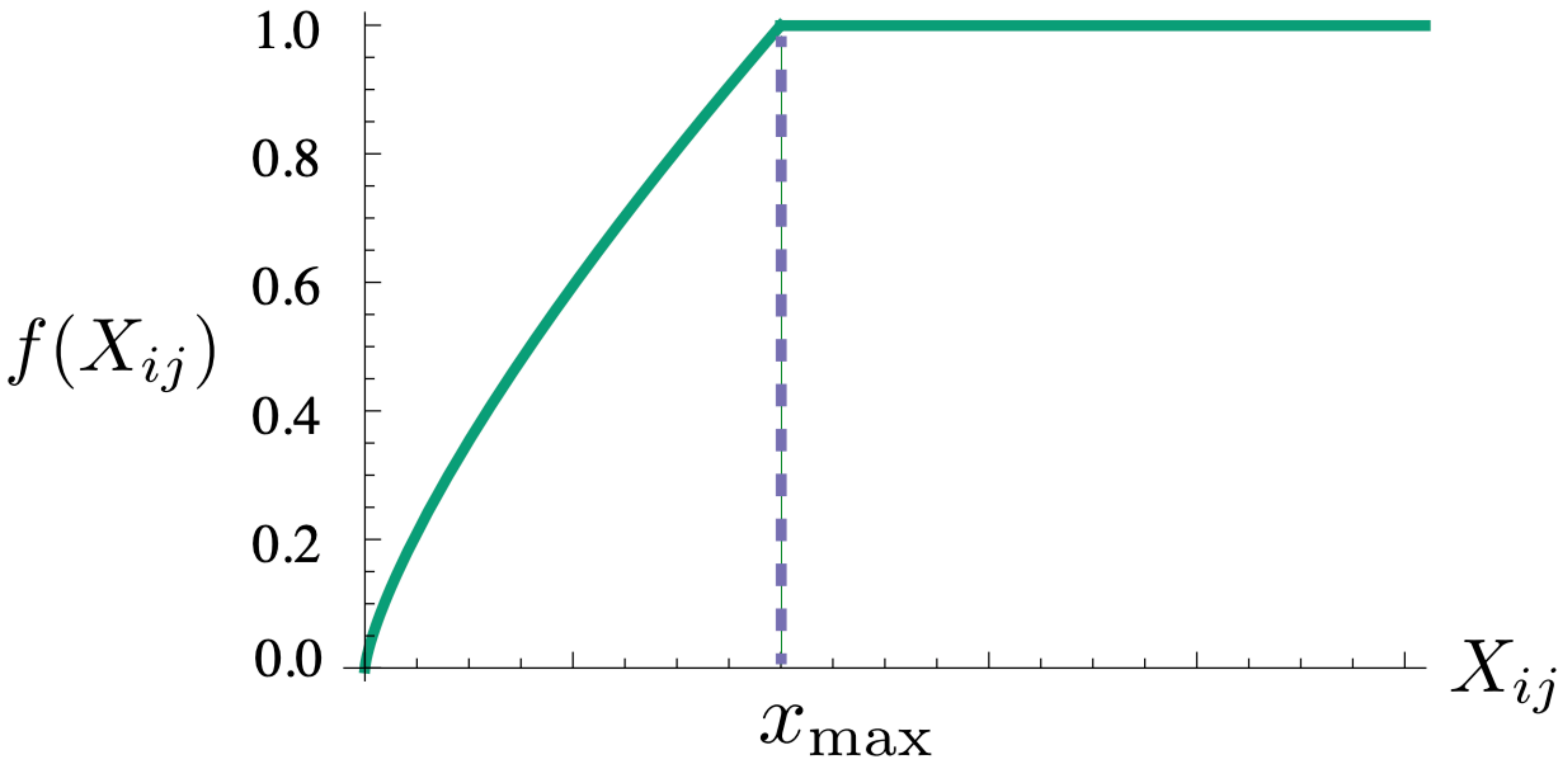
$$J = \sum (w_i^T \hat{w}_k + b_i + \hat{b}_k - log(X_{ik}))^2$$

GloVe - Model

One last problem: all co-occurences are weighted equally, even the rare one.
Let us add a weight function, depending of $X_{i,j}$.

$$J = \sum f(X_{i,j})(w_i^T \hat{w}_k + b_i + \hat{b}_k - log(X_{ik}))^2$$

Where $f = x/x_{max}^{\alpha}$ if $x > x_{max}$, 1 otherwise.



- Convex cost function -> *easy* to solve.
- Constant values are usually set to $x_{max} = 100,\ \alpha = 3/4$

Glove Library

- No python library.
- C library available on GitHub : https://github.com/stanfordnlp/GloVe
- For TP : we use this python wrapper : https://github.com/WenchenLi/GloVePyWrapper to train the model.
- Once the model is trained, it can be loaded with gensim using glove2word2vec api for exploration.

Few Labeled Dataset

Situation

You have a dataset of N_{total} rows but only $N_{labeled} < N_{total}$ are labeled.

Problem: How to use the vocabulary within non-labeled data?

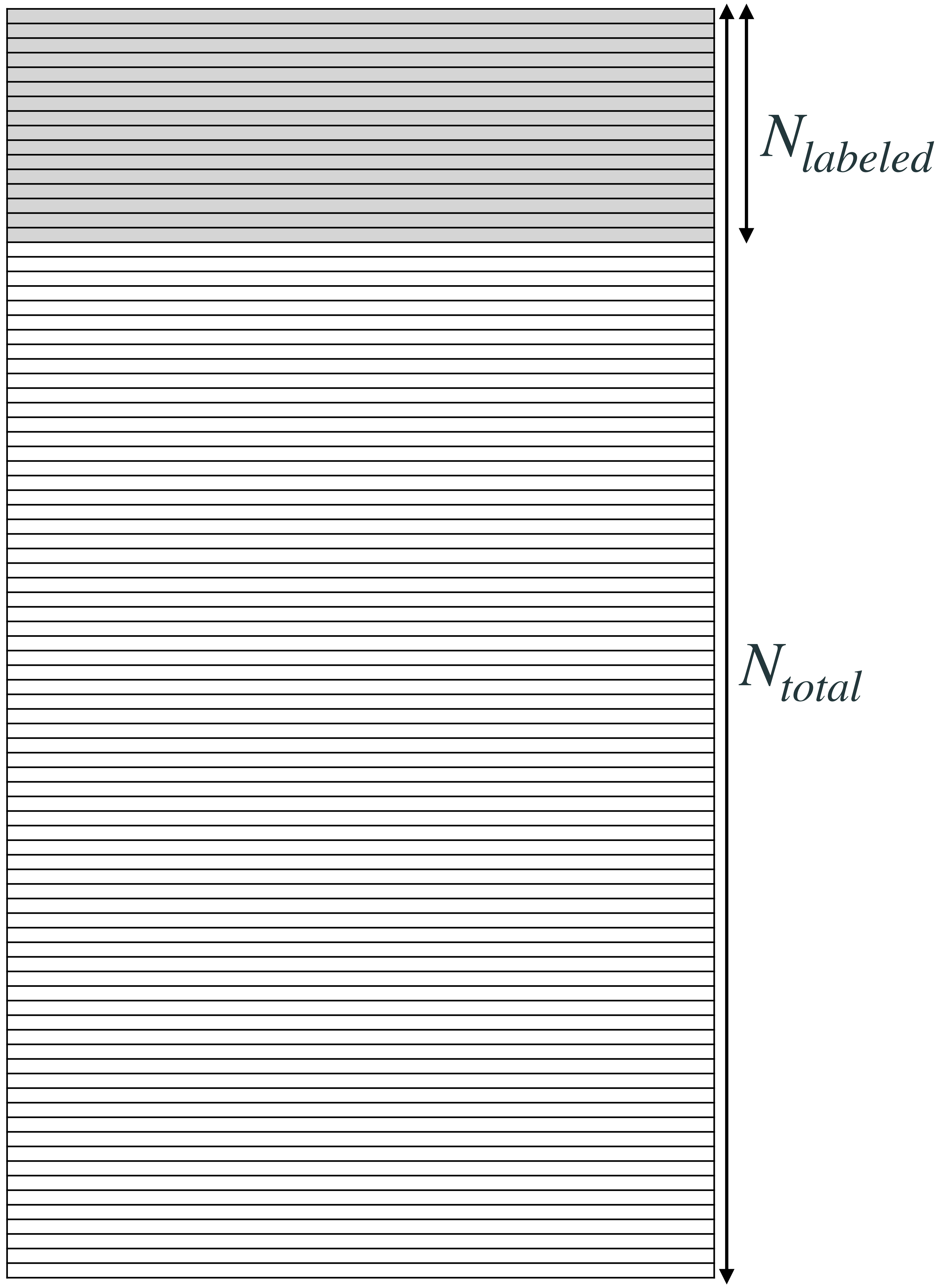
Labelling data is expensive, especially on constructed language or technical data.

Example: *Playstation* is never mentioned within the labeled dataset, but *xbox* is.

With vectorisation, *Playstation* won't be use to predict the category of a product description.

Solution:

- Manually Labeled Dataset with word *Playstation* within it.
- Use words embedding model.



Words Embedding for semi-supervised Learning

How can word embedding help in this situation?

- Learn the word embedding model on the N_{total} lines of the dataset.
- There are a high probability that words *playstation* and *xbox* have the same embedding.
- It's enough that *xbox* is within the $N_{labeled}$ dataset, so that *playstation* can be handle during prediction.
- You save time and money labeled a lot of lines!

TP

Objectives

- Train Word2Vec model using gensim library.
- Explore properties of Word2Vec model (similar word, word operation).
- Use Word2vec features using product classification on train and non labeled dataset.
- Check how it can help in few labeled dataset situation.
- Use FastText and Glove for comparison.
- Compare performance on product classification versus vectorisation methods.

References

Mikolov et al. [2013a], *Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.*

Mikolov et al. [2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Mikolov et al. [2016], *Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.*

Pennington et al. [2014] *Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).*