Introduction au Traitement Automatique des Langues

6 – Le niveau Sémantique : partie 2

Introduction au traitement automatique des langues

Contenu de la matière :

1) Introduction Générale
2) Les applications du TAL
3) Les niveaux de traitement - Traitements de «bas niveau»
4) Les niveaux de traitement - Le niveau lexical
5) Les niveaux de traitement - Le niveau syntaxique
6) **Les niveaux de traitement - Le niveau sémantique**
7) Les niveaux de traitement - Le niveau pragmatique

Plan du cours

1. **Définitions** : Sémantique et Analyse sémantique
2. **Concepts de base**: relations sémantiques, connotation, similarité, proximité, semantic frames, vector semantics & embeddings, mesures de similarité (cosinus).
3. **Représentation vectorielle et techniques** : Term-Document Matrix (Count Vectorizer, Bag-of-Words, N-grams), Term-Term (Co-Occurrence) Matrix, TF-IDF, Word Embeddings, Word2Vec, CBoW, Skip Gram.

Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Types:

Traditional Techniques
Frequency-based or Statistical based vectorization approach

Ex : One-Hot, N-grams, BoW, TF-IDF, PMI, Count Vectorizer, co-occurrence matrix, etc.

New Age Techniques
Prediction / Neural Network based vectorization approach

Ex : Word2Vec, CBoW, Skip Gram, Glove, FastText, ELMo, BERT, XLNet, etc.

Fig. Evolution of NLP Models

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

- Prediction-based technique. Pre-trained model from Google.
- Prediction-based car donnent les probabilités aux mots. Capable de réaliser des tâches d'opérations algébriques comme: King - man +woman = Queen.
- Word2Vec est un outil fourni par Google implémentant deux méthodes de création de Word Embeddings : **CBOW** : Continuous Bag-of-Words et **Skip-gram** : Continuous Skip-gram
- Représentation des mots : embeddings, vecteurs courts et denses.
- Encoder les mots sur un petit vecteur (de dimension de 50-1000) en se basant sur le contexte (en utilisant un encodeur-décodeur) et non pas sur la taille du vocabulaire.
- Et les vecteurs sont denses : pas de comptes nuls (=0, sparse), les valeurs seront des nombres à valeur réelle qui peuvent être négatifs.

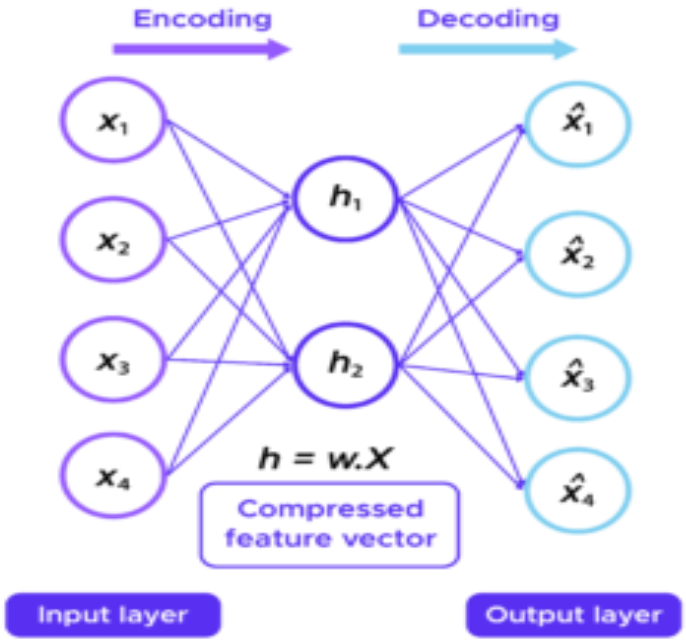
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

- Il existe deux variantes du Word2vec, les deux utilisent un réseau de neurones à 3 couches (1 couche d'entrée, 1 couche cachée, 1 couche de sortie)

- Ces modèles fonctionnent en utilisant le contexte. Cela implique que pour apprendre l'embeddings, il regarde les mots proches. => Window size.



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

- **CBOW** : **Le modèle est nourri par le contexte**, et prédit le mot cible. Le résultat de la couche cachée est la nouvelle représentation du mot ($h1$, ..., hN).

- **Skip Gram** : **Le modèle est nourri par le mot cible**, et prédit les mots du contexte. Le résultat de la couche cachée est la nouvelle représentation du mot ($h1$, ..., hN).

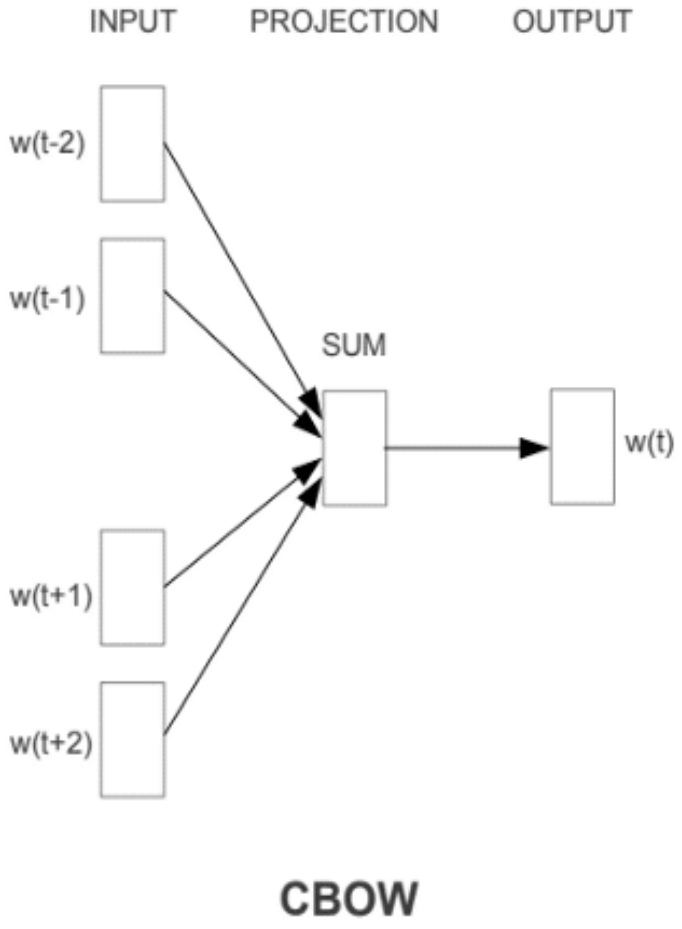
- Skip-Gram fonctionne bien avec un petit volume de données d'entraînement et peut mieux représenter des mots ou des phrases rares.

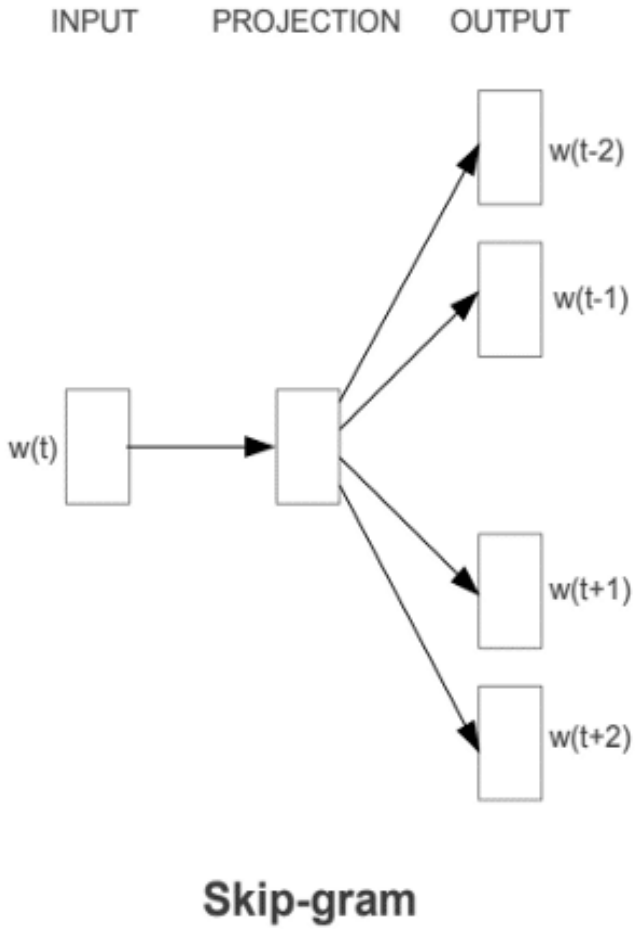
- CBOW s'entraîne plus rapidement que Skip-Gram et peut mieux représenter des mots plus fréquents, ce qui signifie qu'il donne une précision légèrement meilleure pour les mots fréquents.

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec





Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

Etapes:

- **Préparation des données** : définir le corpus en tokenisant le texte.
- **Générer les données d'entraînement** : créer un vocabulaire de mots, un encodage à chaud (one-hot encoding) pour les mots, un index de mots.
- **Modèle d'entraînement** :
  - Passez les mots encodés comme entrée au réseau de neurones (forward propagation),
  - Calculez le taux d'erreur en calculant la perte (loss),
  - Et ajustez les poids à l'aide de la backpropagation.
- **Sortie** : en utilisant le modèle entrainé précédemment, on calcule le vecteur de mots (embeddings) et on trouve les mots similaires.

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

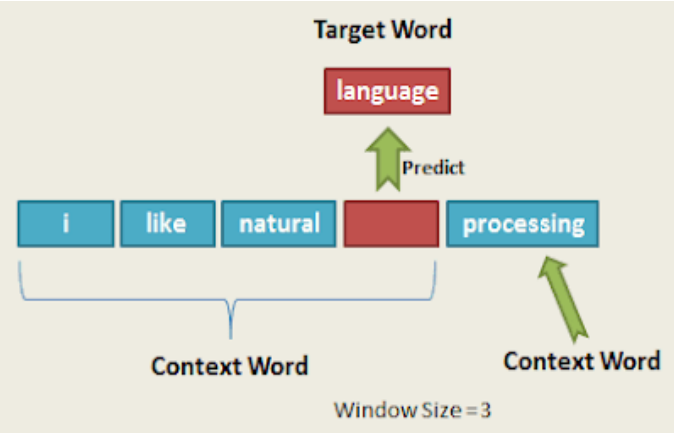
Word2Vec – CBOW

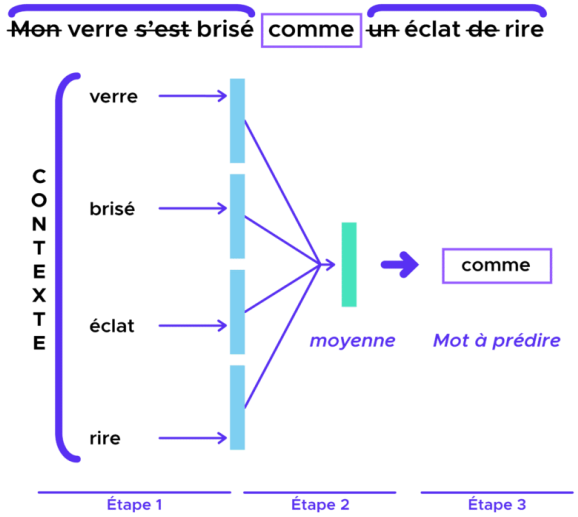
CBOW : Tente de prédire le mot cible à partir de son contexte (mots voisins).

Etapes:

- **Préparation des données** : définir le corpus en tokenisant le texte.
- Exemple – Texte : *i like natural language processing*

=> Tokens : [“i”, “like”, “natural”, “language”, “processing”]





Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Générer les données d'entraînement** :
- Unique vocabulary (without duplicate) : [“i”, “like”, “natural”, “language”, “processing”]

Training Example	Context Word	Target Word
#1	i	like
#2	like	natural
#3	natural	language
#4	language	processing

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Générer les données d'entraînement** :
- Unique vocabulary (without duplicate) : [“i”, “like”, “natural”, “language”, “processing”]

Training Example	Context Word	Target Word
#1	(i, natural)	like
#2	(like, language)	natural
#3	(natural, processing)	language
#4	(language)	processing

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Générer les données d'entraînement** :
- Convertir les mots vers leur one-hot encoding

	i	like	natural	language	processing
i	1	0	0	0	0
like	0	1	0	0	0
natural	0	0	1	0	0
language	0	0	0	1	0
processing	0	0	0	0	1

Training Example	Context Word	Target Word
#1	i	like
#2	like	natural
#3	natural	language
#4	language	processing

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

Training dataset final, où le contexte encodé est les X (input) et le mot cible encodé est le Y (output).

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Générer les données d'entraînement** :
- Convertir les mots vers leur one-hot encoding

	i	like	natural	language	processing
i	1	0	0	0	0
like	0	1	0	0	0
natural	0	0	1	0	0
language	0	0	0	1	0
processing	0	0	0	0	1

Training Example	Context Word	Target Word
#1	(i, natural)	like
#2	(like, language)	natural
#3	(natural, processing)	language
#4	(language)	processing

Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Générer les données d'entraînement** :
- Contexte de plusieurs mots doit être converti en un seul mot/vecteur => moy

Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]

Training Example	Encoded Context Word	Mean Context Word	Encoded Target Word
#1	([1,0,0,0,0],[0,0,1,0,0])	[0.5,0,0.5,0,0]	[0,1,0,0,0]
#2	([0,1,0,0,0],[0,0,0,1,0])	[0,0.5,0,0.5,0]	[0,0,1,0,0]
#3	([0,0,1,0,0],[0,0,0,0,1])	[0,0,0.5,0,0.5]	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,1,0]	[0,0,0,0,1]

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Entraînement**: apprendre les poids **W et W'**

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

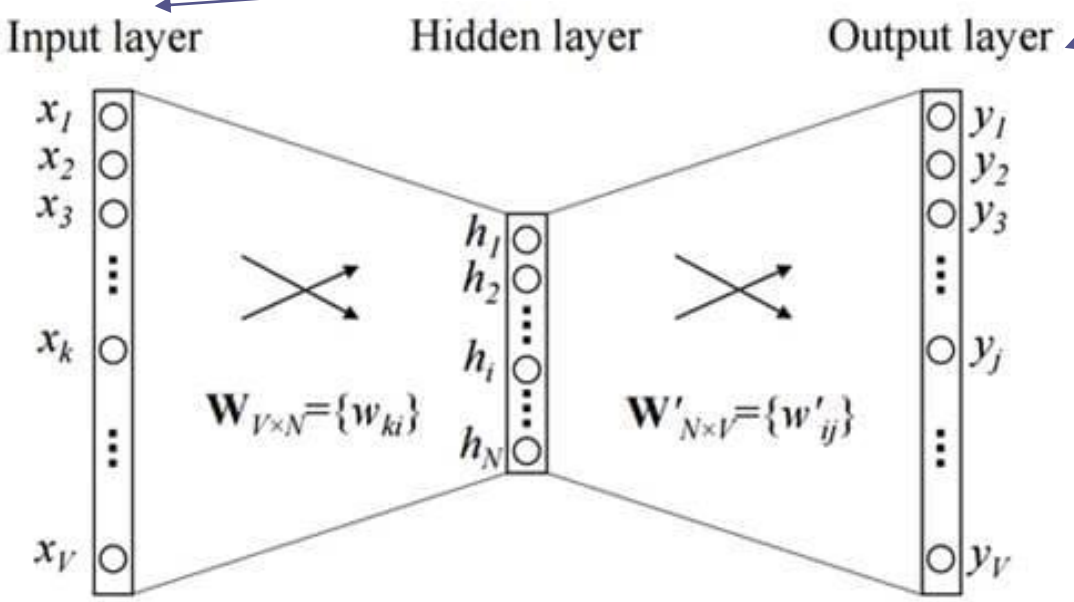


Figure 1: A simple CBOW model with only one word in the context

Représentation vectorielle

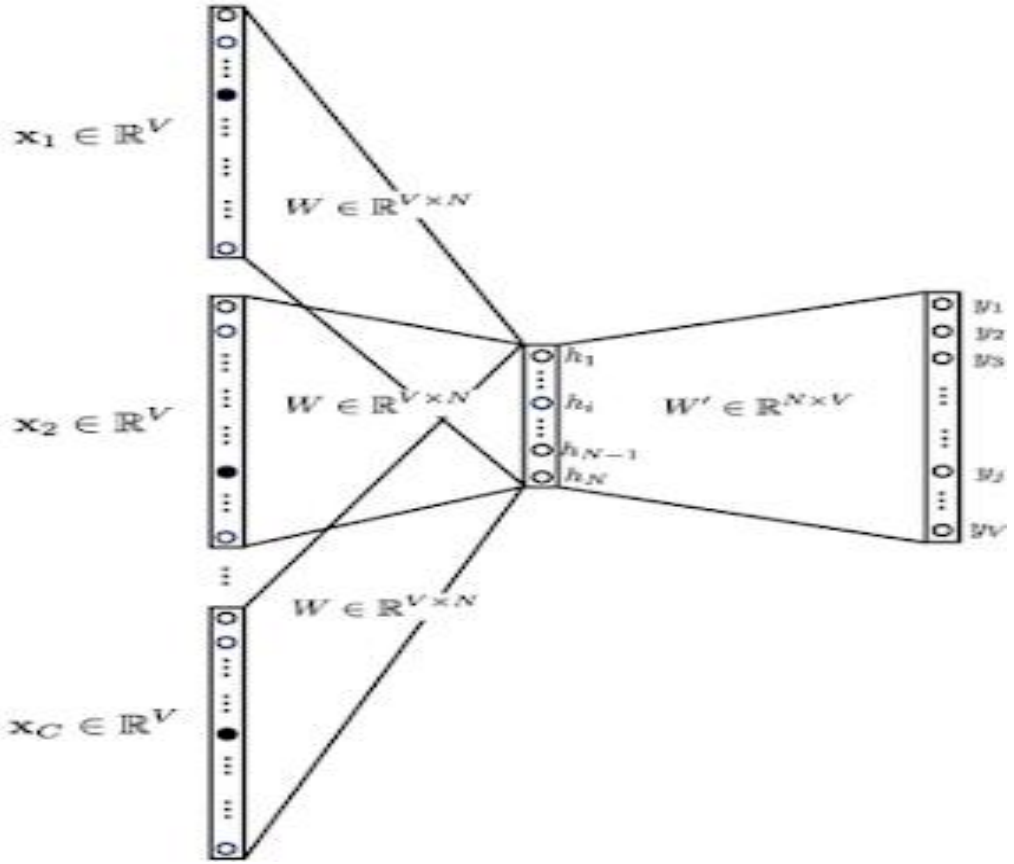
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Modèle d'entraînement** :

Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]



Représentation vectorielle

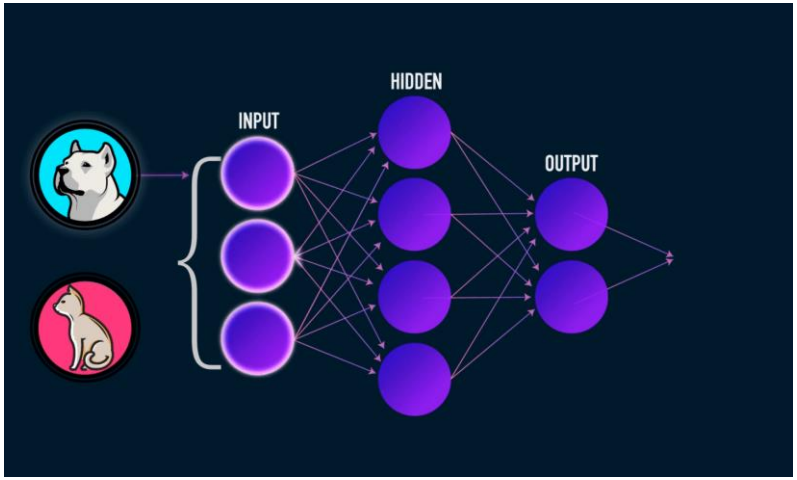
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation
  - Weight tuning using backward pass - backpropagation

Repeat – plusieurs itérations



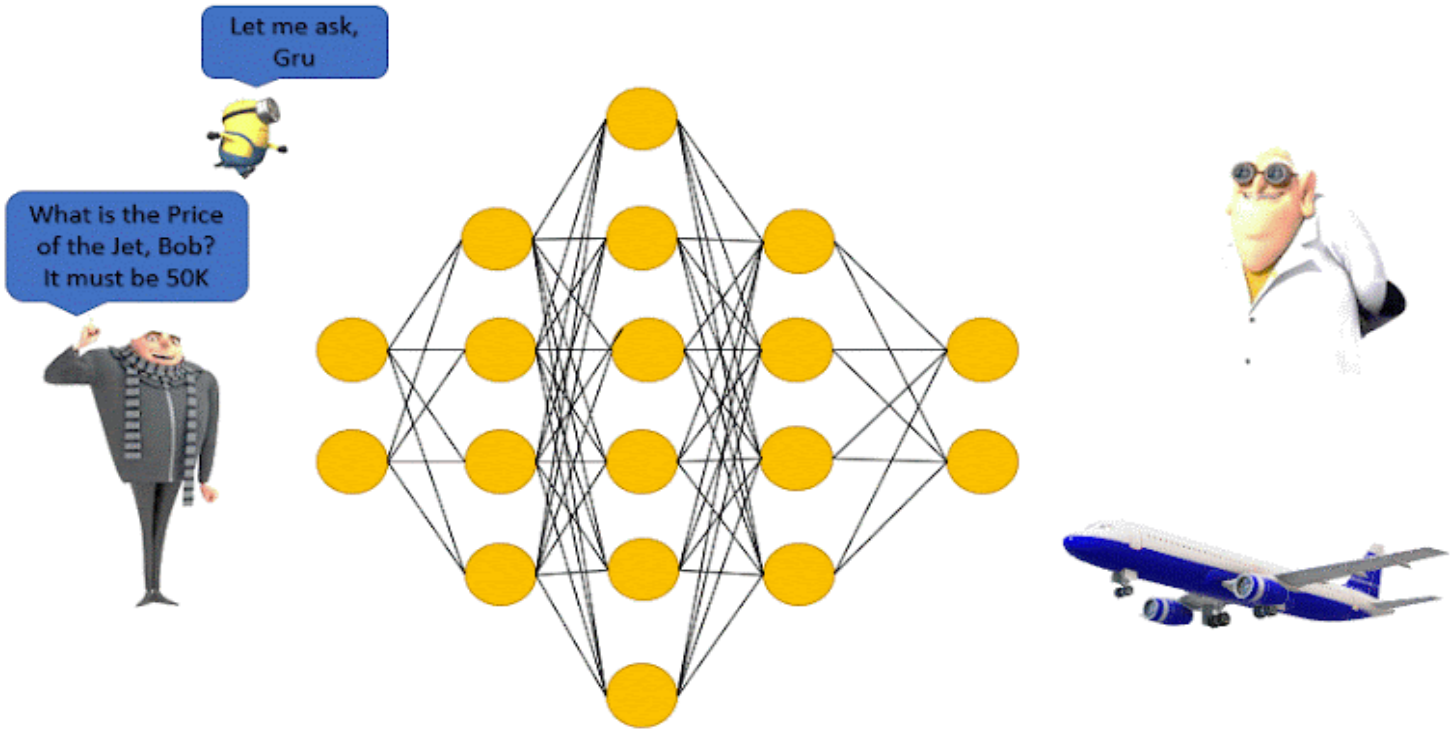
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes



Représentation vectorielle

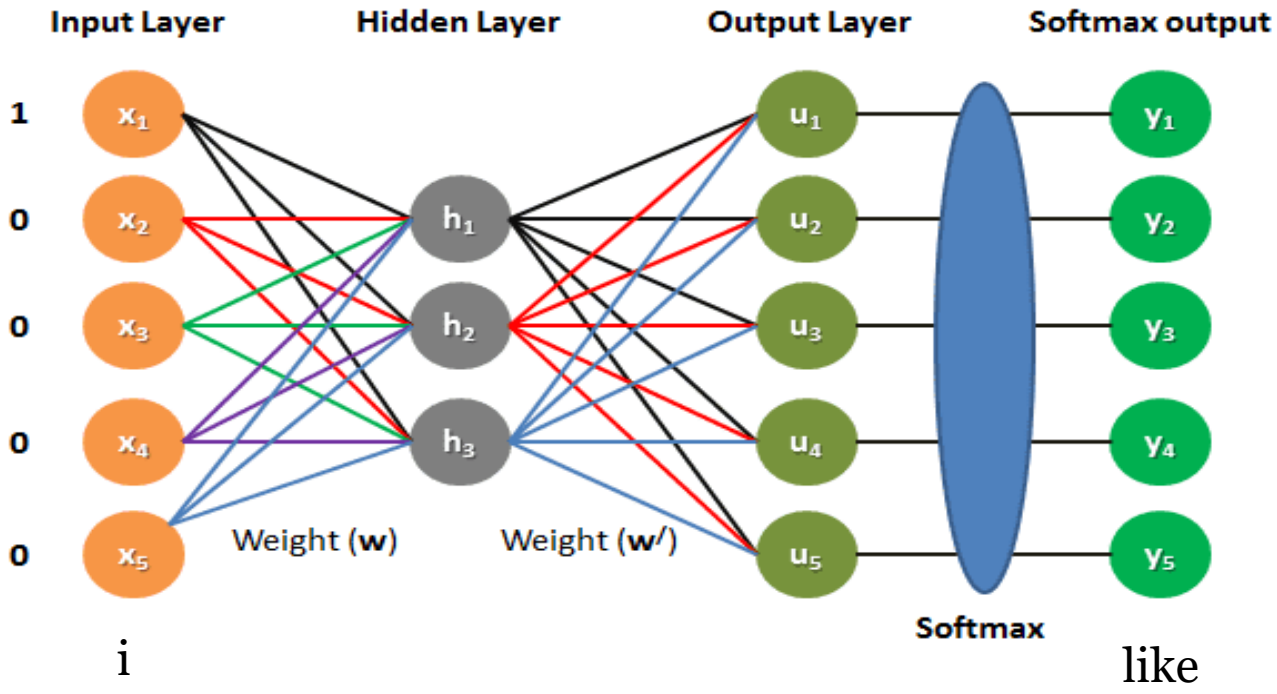
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Architecture

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



First training data point: The context word is "**i**" and the target word is "**like**".

Représentation vectorielle

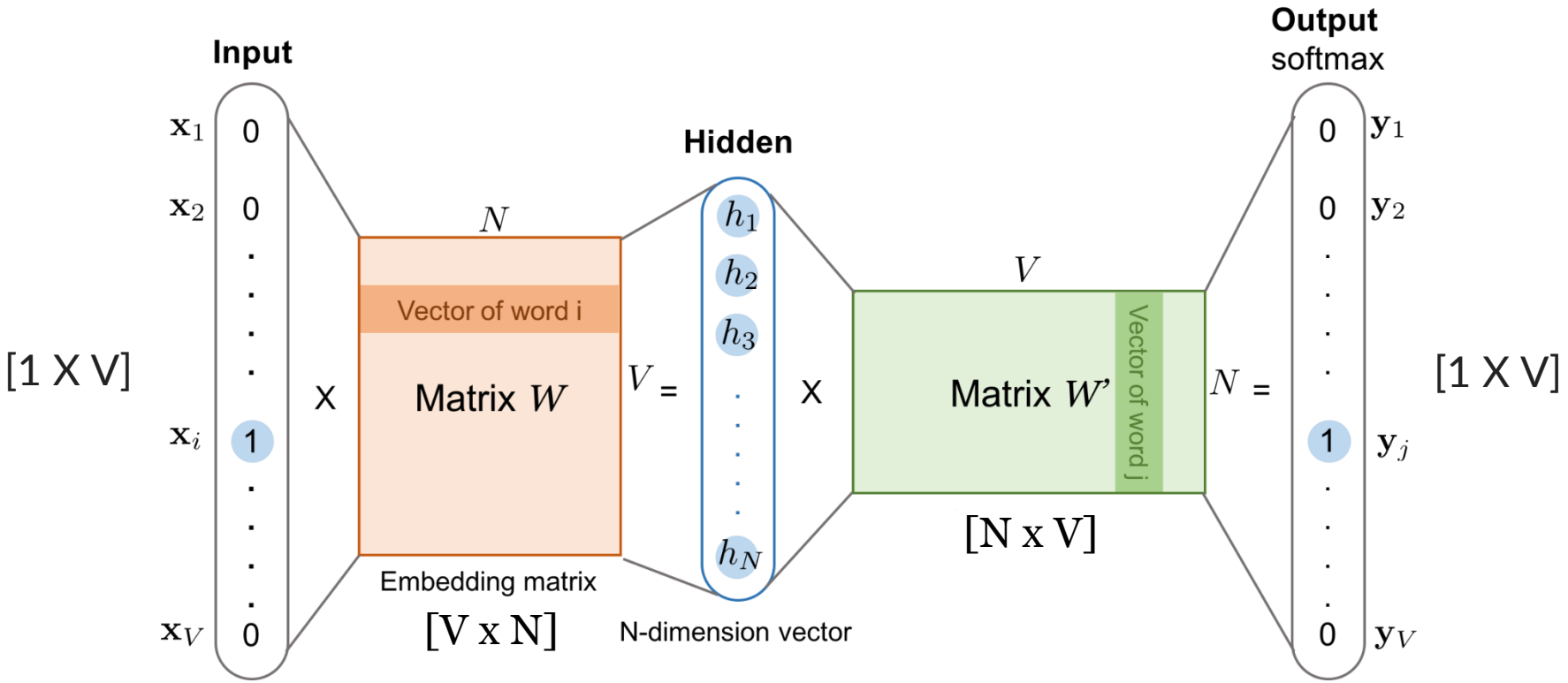
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Architecture

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

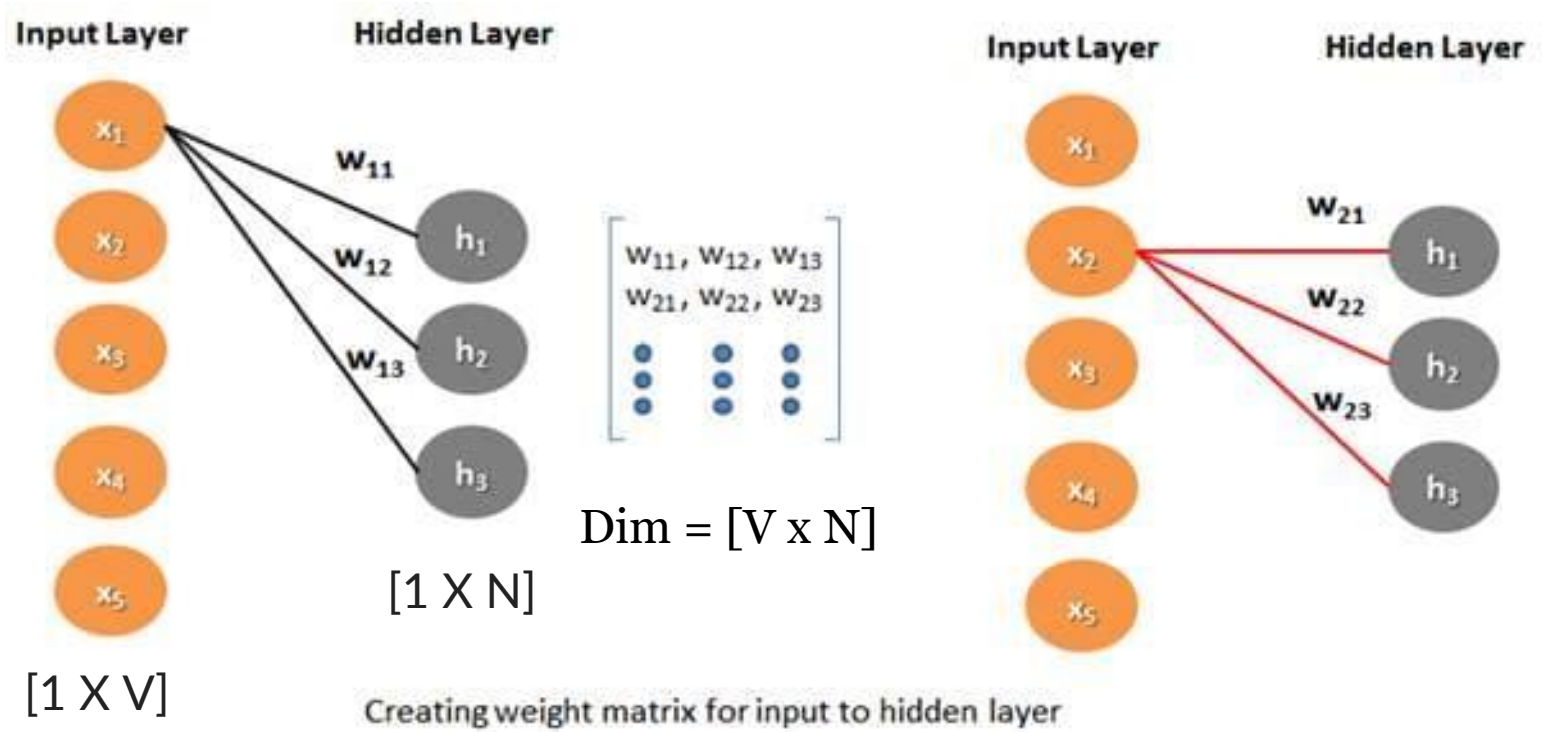
Word2Vec – CBOW : Single Word Model

Etapes:

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

- **Modèle d'entraînement** : Matrice des poids **W**

  La dimension du vecteur d'un mot sera = au nombre de nœuds cachés. Ses valeurs = poids appris.



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

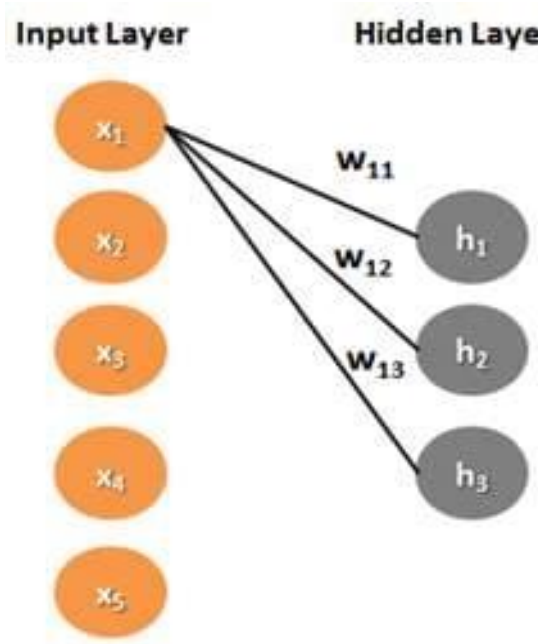
Word2Vec – CBOW : Single Word Model

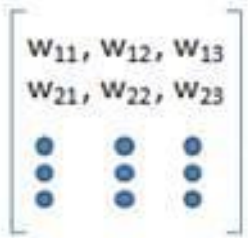
Etapes:

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

- **Modèle d'entraînement** : Matrice des poids **W**

  La dimension du vecteur d'un mot sera = au nombre de nœuds cachés. Ses valeurs = poids appris.





$$w = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

Dim = [V x N]

Représentation vectorielle

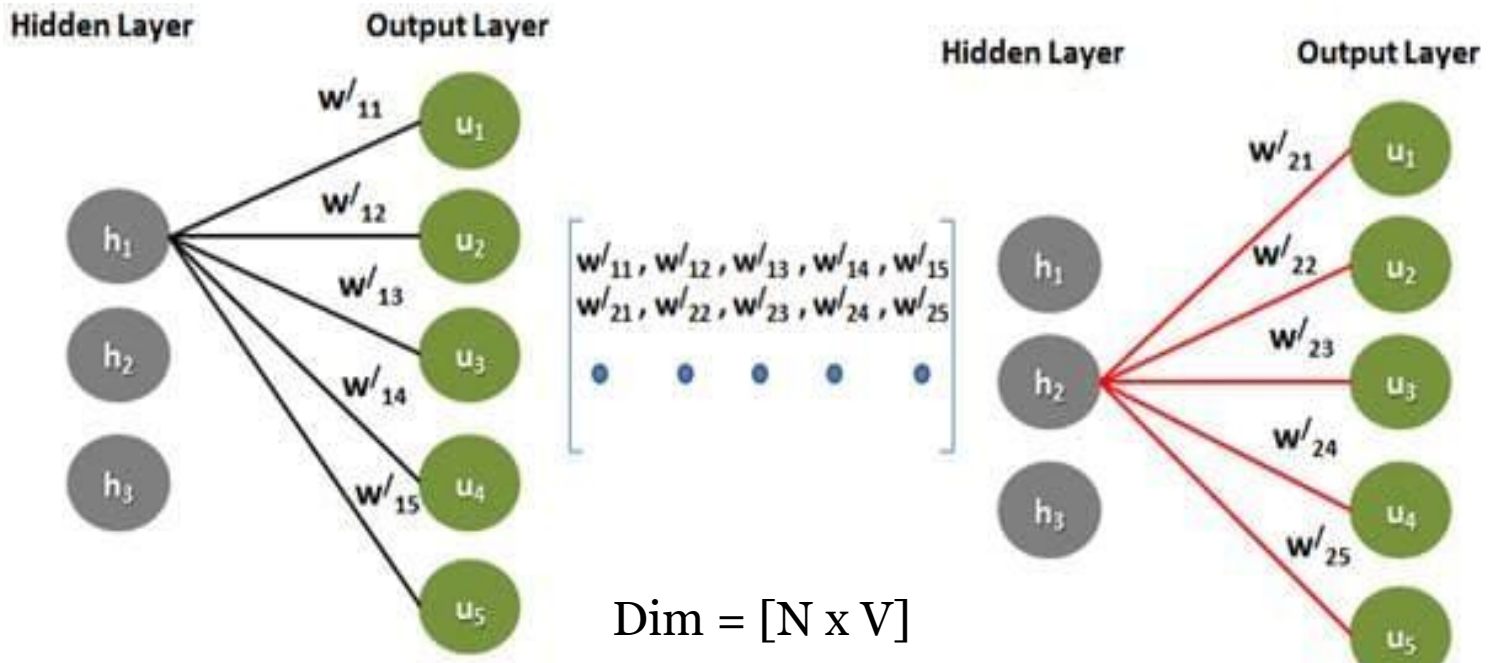
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Matrice des poids **W'**

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



Creating weight matrix for hidden to output layer

Représentation vectorielle

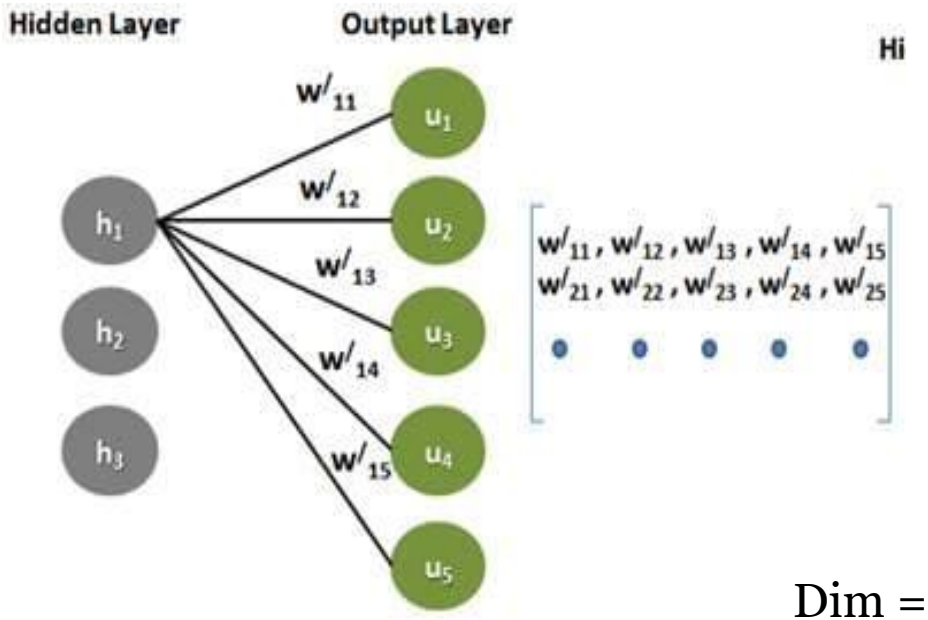
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Matrice des poids **W'**

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



$$W' = \begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} & w'_{15} \\ w'_{21} & w'_{22} & w'_{23} & w'_{24} & w'_{25} \\ w'_{31} & w'_{32} & w'_{33} & w'_{34} & w'_{35} \end{bmatrix}$$

Dim = [N x V]

Représentation vectorielle

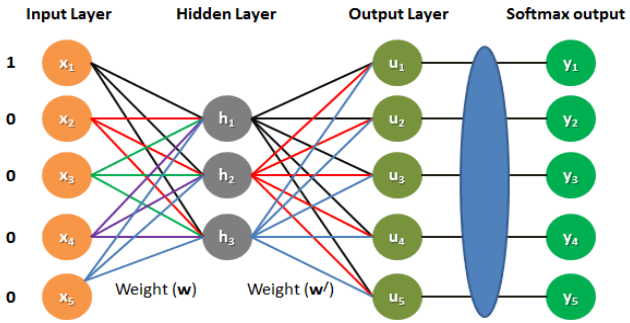
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : CBOW Vectorized Form

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \times \begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} & w'_{15} \\ w'_{21} & w'_{22} & w'_{23} & w'_{24} & w'_{25} \\ w'_{31} & w'_{32} & w'_{33} & w'_{34} & w'_{35} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} \text{Softmax} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

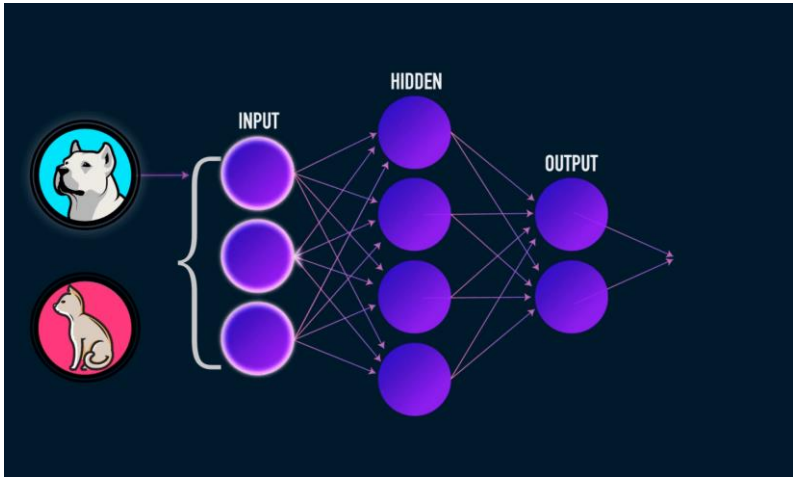
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation
  - Weight tuning using backward pass - backpropagation



Représentation vectorielle

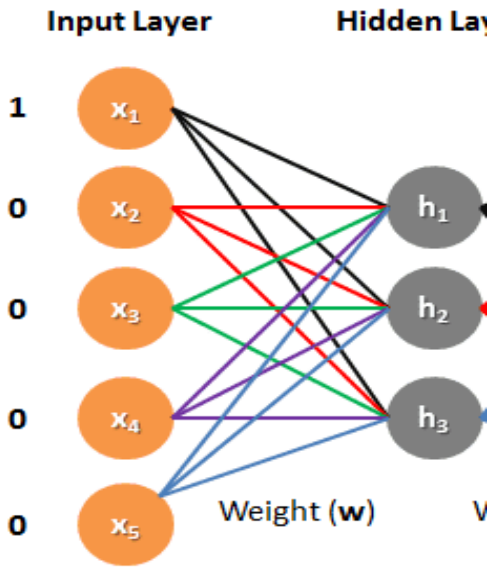
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Forward Propagation - **W**

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



x_1	x_2	x_3	x_4	x_5

x

w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}
w_{41}	w_{42}	w_{43}
w_{51}	w_{52}	w_{53}

=

h_1	h_2	h_3

Hiddel Laver matrix calculation

$$h_1 = w_{11}x_1 + w_{21}x_2 + w_{31}x_3 + w_{41}x_4 + w_{51}x_5$$

$$h_2 = w_{12}x_1 + w_{22}x_2 + w_{32}x_3 + w_{42}x_4 + w_{52}x_5$$

$$h_3 = w_{13}x_1 + w_{23}x_2 + w_{33}x_3 + w_{43}x_4 + w_{53}x_5$$

Représentation vectorielle

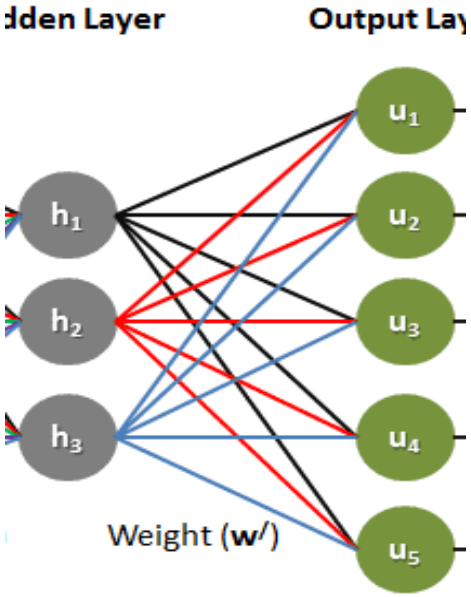
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement** : Forward Propagation – **W'**

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]



h_1	h_2	h_3	x	w'_{11}	w'_{12}	w'_{13}	w'_{14}	w'_{15}	=	u_1	u_2	u_3	u_4	u_5
				w'_{21}	w'_{22}	w'_{23}	w'_{24}	w'_{25}						
				w'_{31}	w'_{32}	w'_{33}	w'_{34}	w'_{35}						

Output Layer matrix calculation

$$u_1 = w'_{11}h_1 + w'_{21}h_2 + w'_{31}h_3$$
$$u_2 = w'_{12}h_1 + w'_{22}h_2 + w'_{32}h_3$$
$$u_3 = w'_{13}h_1 + w'_{23}h_2 + w'_{33}h_3$$
$$u_4 = w'_{14}h_1 + w'_{24}h_2 + w'_{34}h_3$$
$$u_5 = w'_{15}h_1 + w'_{25}h_2 + w'_{35}h_3$$

Représentation vectorielle

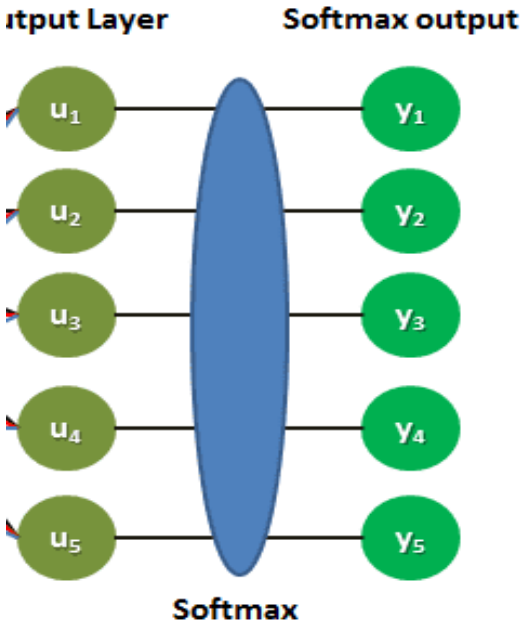
New Age Techniques
Prediction / Neural Network based vectorization approach

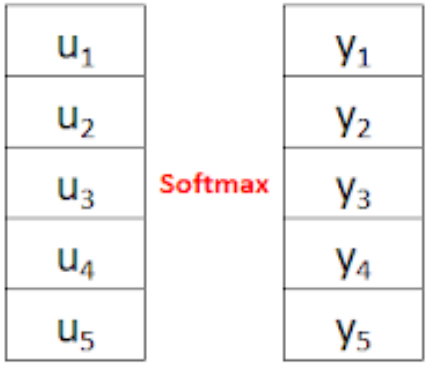
Word2Vec – CBOW : Single Word Model

Etapes:

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

- **Modèle d'entraînement** : Forward Propagation – **W'**





$y_1 = \text{Softmax}(u_1)$
$y_2 = \text{Softmax}(u_2)$
$y_3 = \text{Softmax}(u_3)$
$y_4 = \text{Softmax}(u_4)$
$y_5 = \text{Softmax}(u_5)$

Softmax calcule la probabilité pour chaque classe possible. La fonction Softmax utilise l'exponentiel afin d'obtenir la sortie de softmax dans une plage comprise entre 0 et 1.

Représentation vectorielle

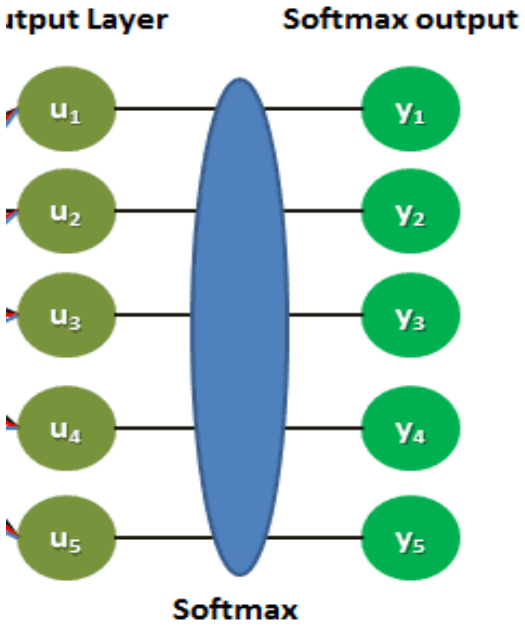
New Age Techniques
Prediction / Neural Network based vectorization approach

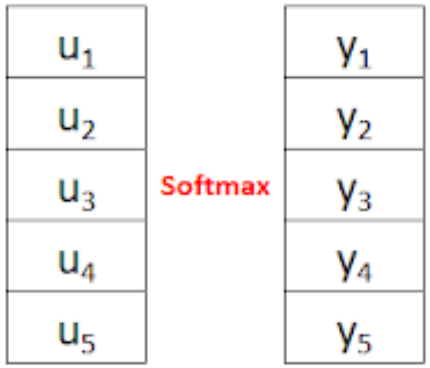
Word2Vec – CBOW : Single Word Model

Etapes:

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

- **Modèle d'entraînement** : Forward Propagation – **W'**





$y_1 = \text{Softmax}(u_1)$
$y_2 = \text{Softmax}(u_2)$
$y_3 = \text{Softmax}(u_3)$
$y_4 = \text{Softmax}(u_4)$
$y_5 = \text{Softmax}(u_5)$

$$y_j = \frac{e^j}{\sum_{j=1}^{V} e^j}$$

$$y_1 = \frac{e^{u_1}}{\left(e^{u_1} + e^{u_2} + e^{u_3} + e^{u_4} + e^{u_5}\right)}$$

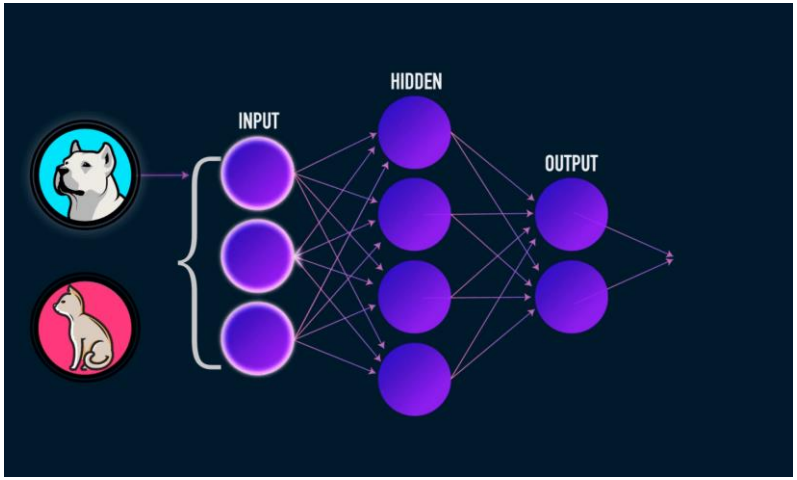
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation – Loss function
  - Weight tuning using backward pass - backpropagation



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Modèle d'entraînement**: Error Calculation, for **Y**

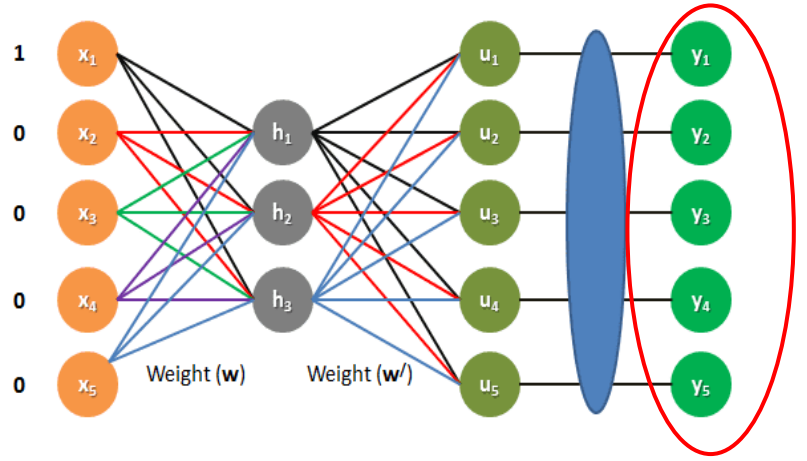
Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

$$E = -u_{j^*} + log \sum_{j=1}^{V} e^{u_j}$$

Actual value (target) **Vs**. Predicted value (output)

j* est l'index du mot cible (target) dans l'output layer

$$E(y2) = -u_2 + log\left(e^{u_1} + e^{u_2} + e^{u_3} + e^{u_4} + e^{u_5}\right)$$



Plus la fonction de cout (loss) est faible, meilleures sont les performances du réseau de neurones.
- Le but est de minimiser l'erreur E (la perte).

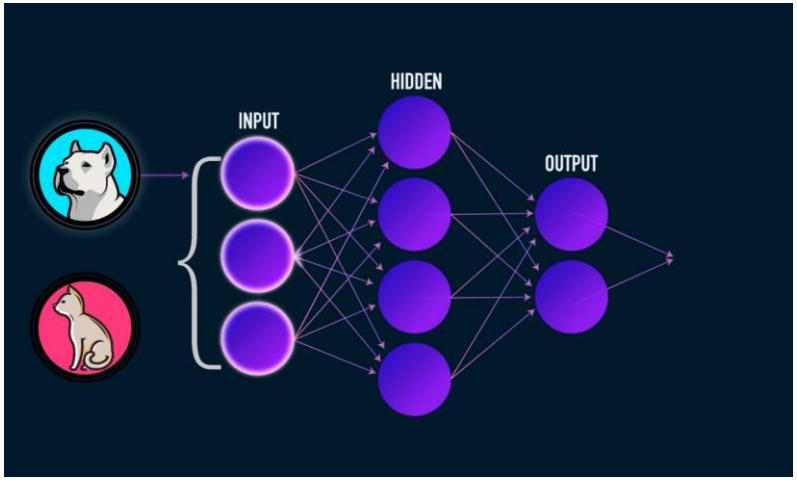
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation – Loss function
  - Weight tuning using backward pass - backpropagation



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

Etapes:

- **Modèle d'entraînement**: Backpropagation
- Trouver les poids optimaux d'un réseau de neurones en les ajustant : ceux qui minimisent la fonction de perte (loss).
- Apprendre les poids W et W'.
- La manière standard de trouver ces valeurs est d'appliquer l'algorithme de descente de gradient (gradient descent), ce qui implique de trouver les dérivées de la fonction de perte par rapport aux poids.
- Afin d'appliquer cet algorithme et mettre à jour les matrices des poids W et W', on doit trouver les dérivées (derivatives) :

$$\partial E/\partial W \quad \text{et} \quad \partial E/\partial W'$$

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

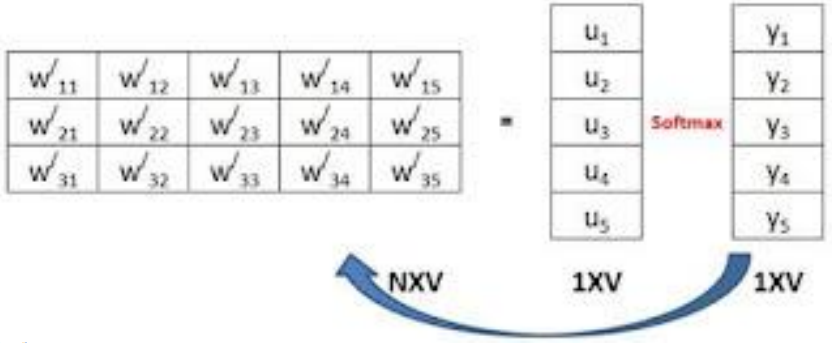
Etapes:

- **Modèle d'entraînement**: Backpropagation

1 - Calculer le gradient pour chaque poids de **W'**: (w'_{11} , w'_{12}, w'_{13} …. w'_{15} … w'_{35})

Ex : Gradient of E with respect to w'_{11}:

$$\frac{dE}{dw'} == e * h$$



$$\frac{dE(y_1)}{dw'_{11}} = \frac{dE(y_1)}{du_1}, \frac{du_1}{dw'_{11}} = e_1 h_1$$

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

Etapes:

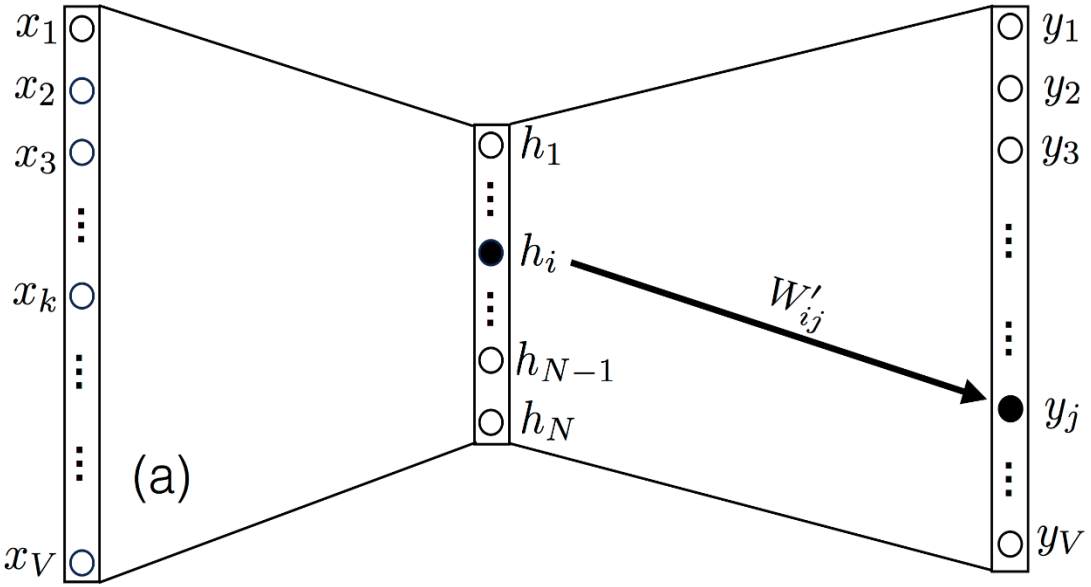
- **Modèle d'entraînement**: Backpropagation

1 - Calculer le gradient pour chaque poids de **W'**: (w'_{11} , w'_{12}, w'_{13} …. w'_{15} … w'_{35})

Ex : Gradient of E with respect to w'_{11}:

$$\frac{dE}{dw'} == e * h$$

$$\frac{dE(y_1)}{dw'_{11}} = \frac{dE(y_1)}{du_1}, \frac{du_1}{dw'_{11}} = e_1 h_1$$



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

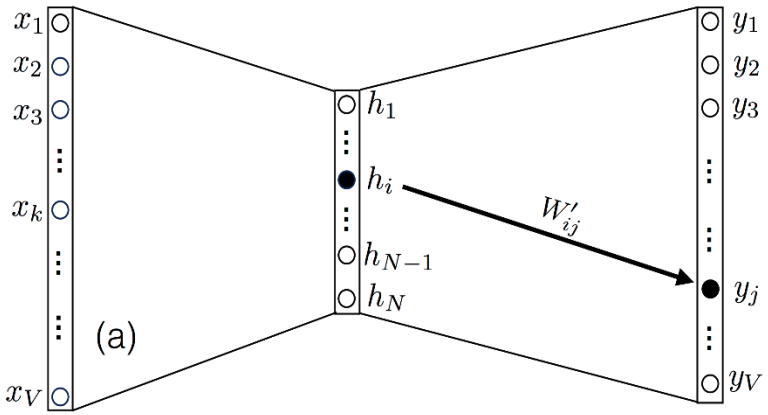
- **Modèle d'entraînement**: Backpropagation

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

2 – Mettre à jour tous les poids de **W'**: (w'_{11} , w'_{12}, w'_{13} …. w'_{15} … w'_{35})

Ex : Update w'_{11}:

$$new(w'_{11}) = w'_{11} - \frac{dE(y_1)}{w'_{11}} = (w'_{11} - e_1 h_1)$$



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

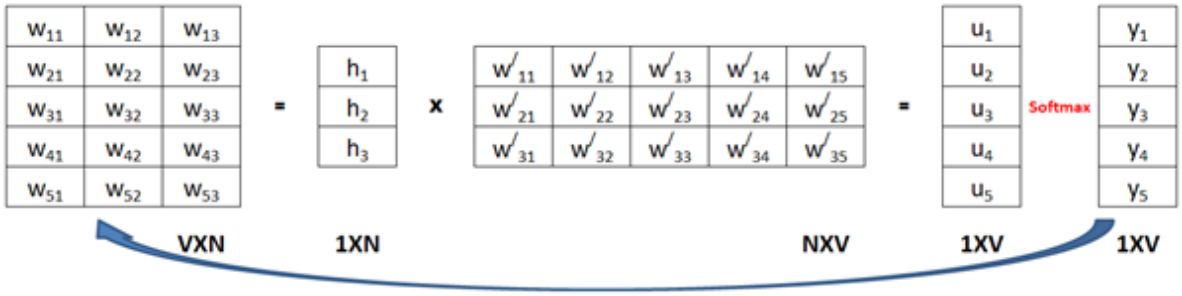
Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

Etapes:

- **Modèle d'entraînement**: Backpropagation

1 - Calculer le gradient pour chaque poids de **W**: (w_{11} , w_{12}, w_{13} ….)

Ex : Gradient of E with respect to w_{11}:



$$\frac{dE}{dw_{11}} = \frac{dE}{dh_1}, \frac{dh_1}{dw_{11}}$$

$$= (ew'_{11} + ew'_{12} + ew'_{13} + ew'_{14} + ew'_{15}) * x$$

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

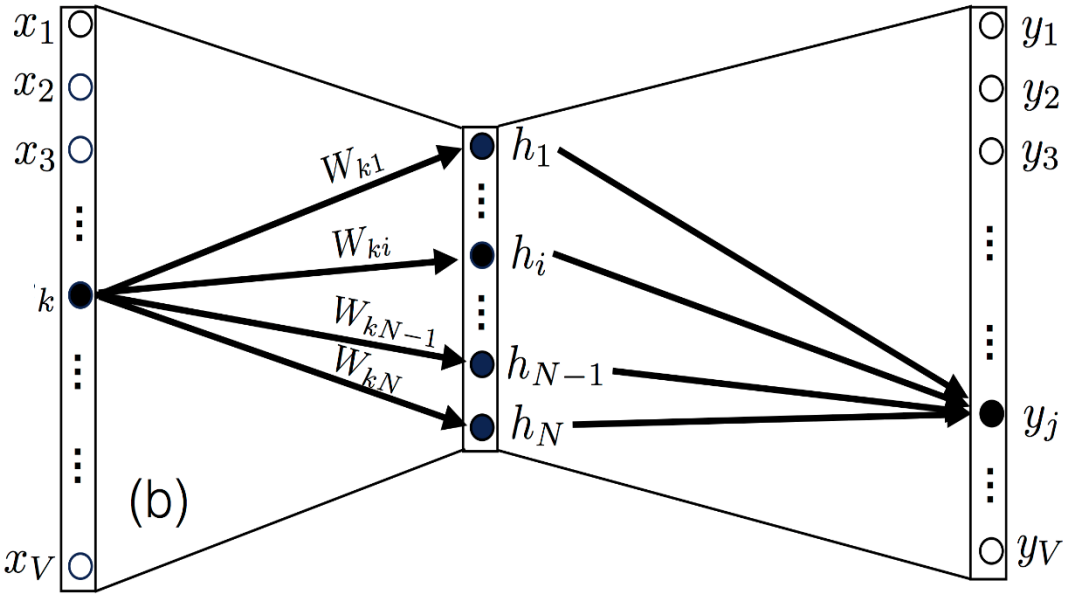
Etapes:

- **Modèle d'entraînement**: Backpropagation

1 - Calculer le gradient pour chaque poids de **W**: (w_{11} , w_{12}, w_{13} ….)

Ex : Gradient of E with respect to w_{11}:

$$\frac{dE}{dw_{11}} = \frac{dE}{dh_1}, \frac{dh_1}{dw_{11}}$$
$$= (ew'_{11} + ew'_{12} + ew'_{13} + ew'_{14} + ew'_{15}) * x$$



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

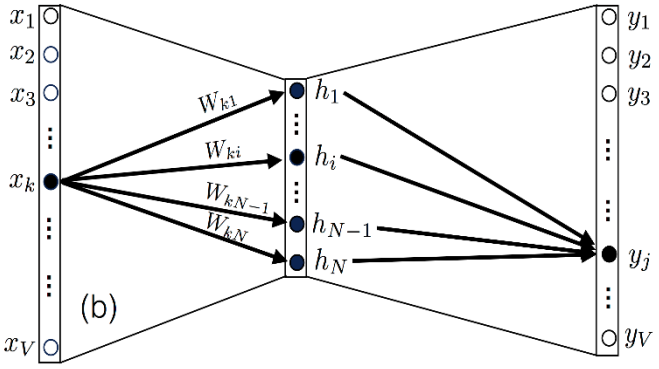
- **Modèle d'entraînement**: Backpropagation

Training Example	Encoded Context Word	Encoded Target Word
#1	[1,0,0,0,0]	[0,1,0,0,0]
#2	[0,1,0,0,0]	[0,0,1,0,0]
#3	[0,0,1,0,0]	[0,0,0,1,0]
#4	[0,0,0,1,0]	[0,0,0,0,1]

2 – Mettre à jour tous les poids de **W**: (w_{11} , w_{12}, w_{13} ….)

Ex : Update w_{11}:

$$new(w_{11}) = w_{11} - \frac{dE}{dw_{11}}$$
$$= w_{11} - (ew'_{11} + ew'_{12} + ew'_{13} + ew'_{14} + ew'_{15}) * x$$



Représentation vectorielle

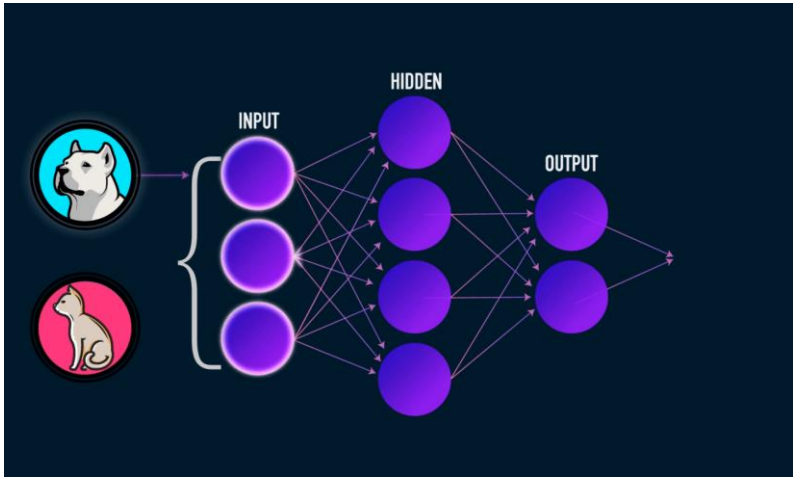
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation
  - Weight tuning using backward pass - backpropagation

Repeat – plusieurs itérations /epochs



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

Etapes:

- **Préparation des données** : définir le corpus en tokenisant le texte.
- **Générer les données d'entraînement** : créer un vocabulaire de mots, un encodage à chaud (one-hot encoding) pour les mots, un index de mots.
- **Modèle d'entraînement** :
  - Passez les mots encodés comme entrée au réseau de neurones (forward propagation),
  - Calculez le taux d'erreur en calculant la perte (loss),
  - Et ajustez les poids à l'aide de la backpropagation.
- **Sortie** : en utilisant le modèle entrainé précédemment, on calcule le vecteur de mots (embeddings) et on trouve les mots similaires.

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Single Word Model

Etapes:

- **Sortie** : en utilisant le modèle entrainé précédemment, on calcule le vecteur de mots (**embeddings**) et on trouve les mots similaires. À partir de W ou W'
- Exemple – Texte : *i like natural language processing*
- *Word2vce embedding (word vector) du mot "i" est : <w11, w12, w13>*

$$w = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

"i"	w11	w12	w13
"like"	w21	w22	w23
"natural"	w31	w32	w33
"language"	w41	w42	w43
"processing"	w51	w52	w53

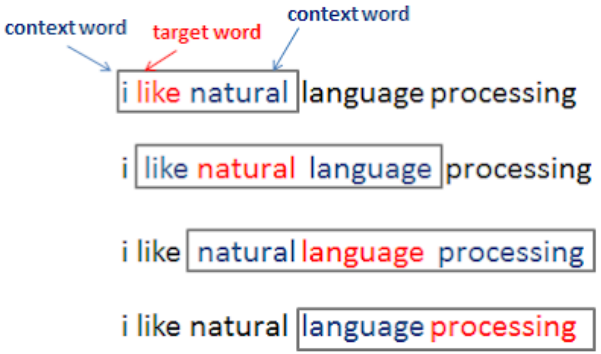
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

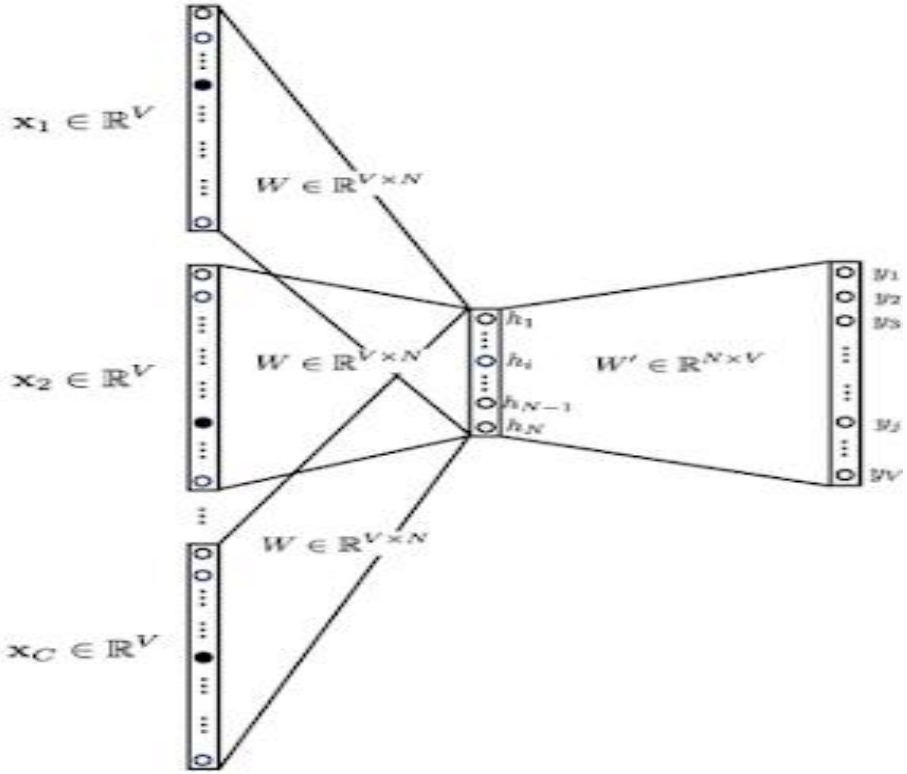
Word2Vec – CBOW : Multi Word Model

Etapes:

- **Modèle d'entraînement** : Architecture



Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]



Représentation vectorielle

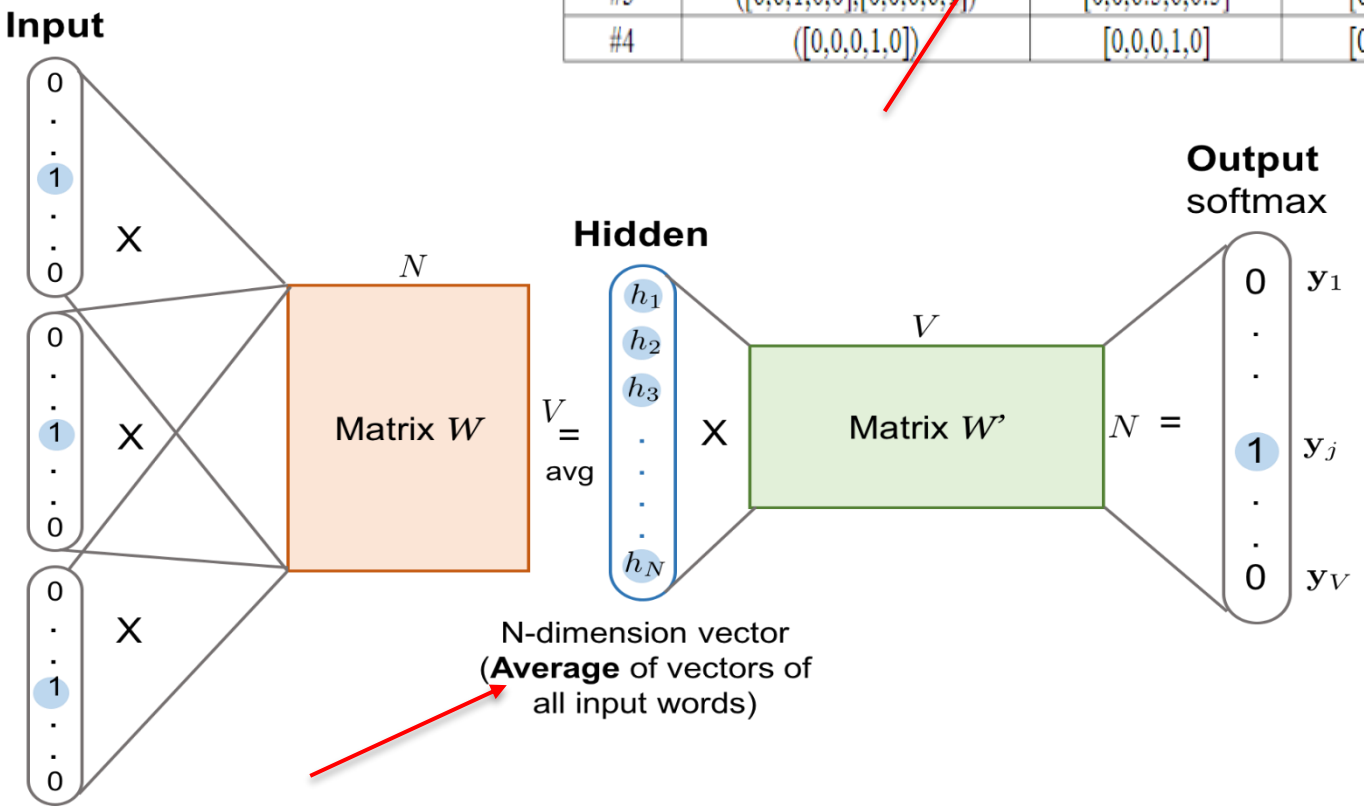
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Modèle d'entraînement** :

Training Example	Encoded Context Word	Mean Context Word	Encoded Target Word
#1	([1,0,0,0,0],[0,0,1,0,0])	[0.5,0,0.5,0,0]	[0,1,0,0,0]
#2	([0,1,0,0,0],[0,0,0,1,0])	[0,0.5,0,0.5,0]	[0,0,1,0,0]
#3	([0,0,1,0,0],[0,0,0,0,1])	[0,0,0.5,0,0.5]	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,1,0]	[0,0,0,0,1]



Représentation vectorielle

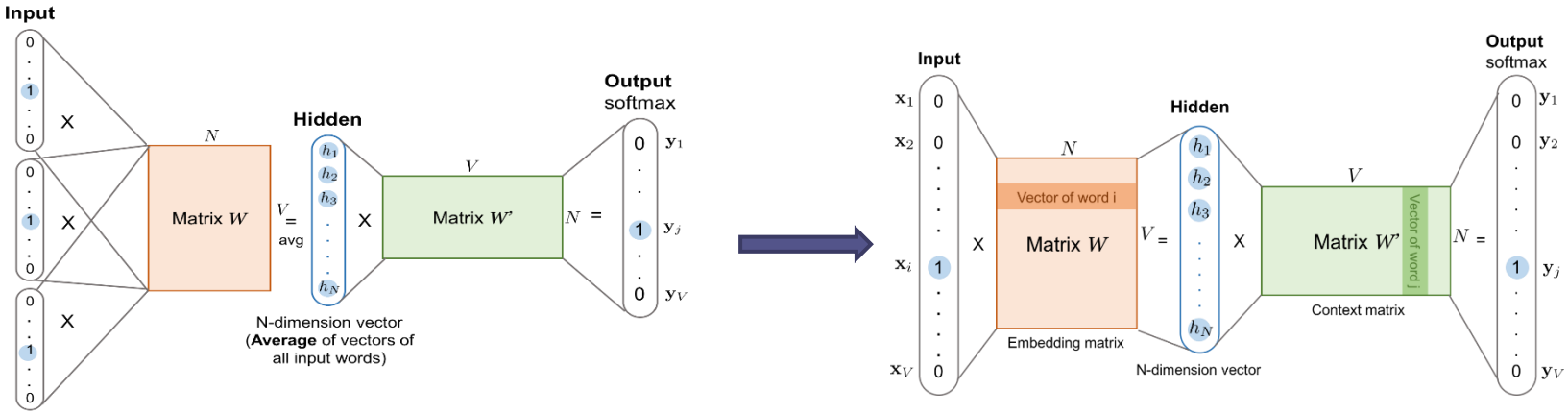
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – CBOW : Multi Word Model

Etapes:

- **Modèle d'entraînement** : **pareil qu'avec Single Word Model**
  - Create model Architecture
  - Forward Propagation
  - Error Calculation
  - Weight tuning using backward pass - backpropagation

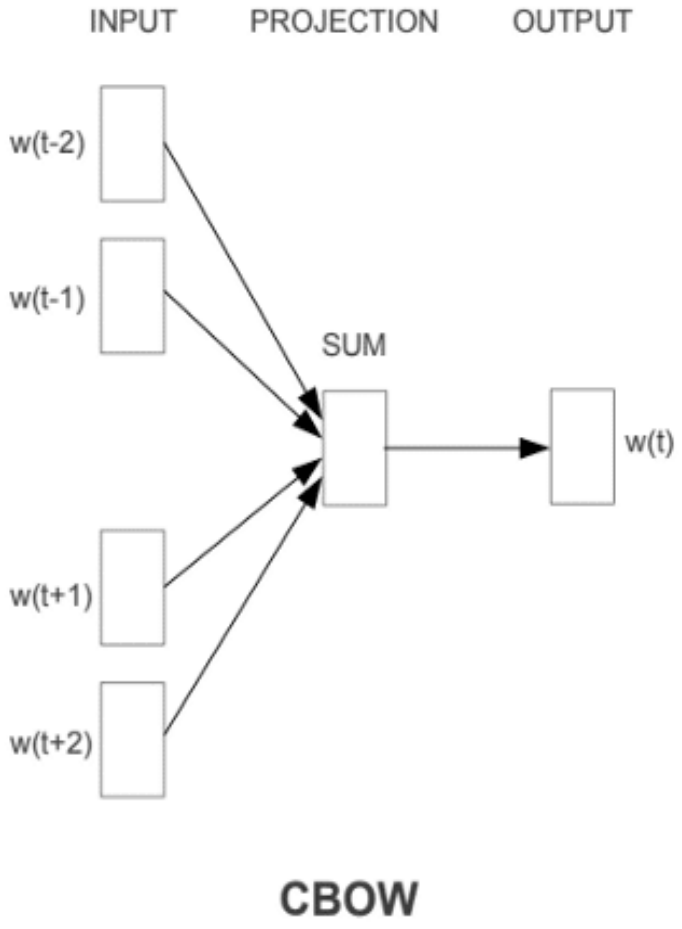
Repeat – plusieurs itérations /epochs

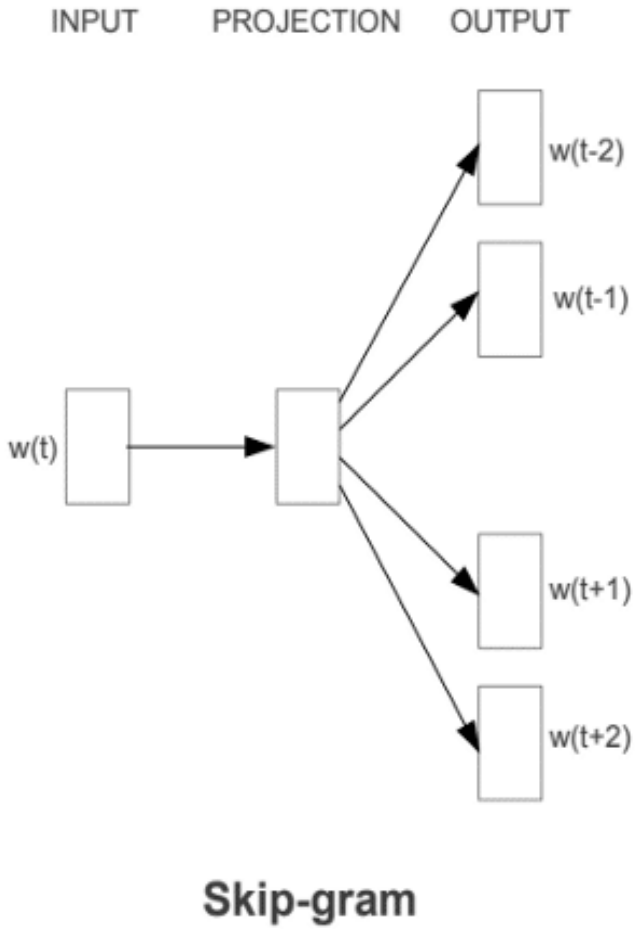


Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec





Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec

Etapes:

- **Préparation des données** : définir le corpus en tokenisant le texte.
- **Générer les données d'entraînement** : créer un vocabulaire de mots, un encodage à chaud (one-hot encoding) pour les mots, un index de mots.
- **Modèle d'entraînement** :
  - Passez les mots encodés comme entrée au réseau de neurones (forward propagation),
  - Calculez le taux d'erreur en calculant la perte (loss),
  - Et ajustez les poids à l'aide de la backpropagation.
- **Sortie** : en utilisant le modèle entrainé précédemment, on calcule le vecteur de mots (embeddings) et on trouve les mots similaires.

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

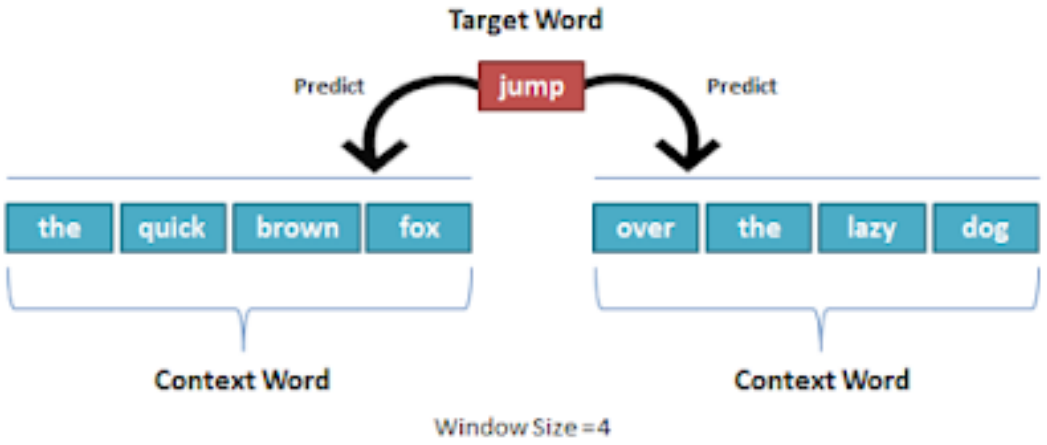
Word2Vec – Skip-Gram

Skpi-Gram : Tente de prédire le contexte à partir du mot cible.

Etapes:

- **Préparation des données** : définir le corpus en tokenisant le texte.
- Exemple – Texte : *i like natural language processing*

=> Tokens : [“i”, “like”, “natural”, “language”, “processing”]



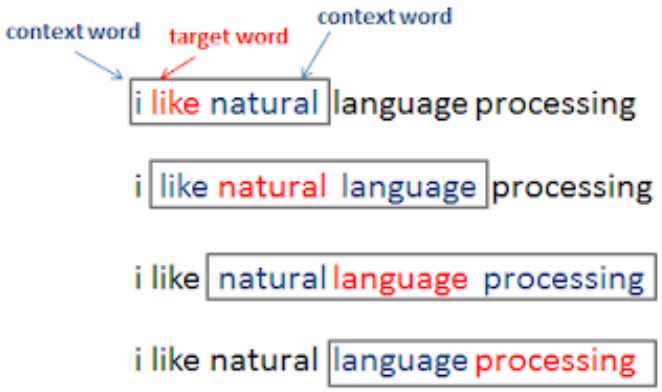
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Générer les données d'entraînement** : window-size = 1



Training Example	Context Word	Target Word
#1	(i, natural)	like
#2	(like, language)	natural
#3	(natural, processing)	language
#4	(language)	processing

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Générer les données d'entraînement** :
- Convertir les mots vers leur one-hot encoding

	i	like	natural	language	processing
i	1	0	0	0	0
like	0	1	0	0	0
natural	0	0	1	0	0
language	0	0	0	1	0
processing	0	0	0	0	1

Training Example	Context Word	Target Word
#1	(i, natural)	like
#2	(like, language)	natural
#3	(natural, processing)	language
#4	(language)	processing

Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Générer les données d'entraînement** :
- Contexte de plusieurs mots doit être converti en un seul mot/vecteur => moy

Training Example	Encoded Context Word	Encoded Target Word
#1	([1,0,0,0,0], [0,0,1,0,0])	[0,1,0,0,0]
#2	([0,1,0,0,0], [0,0,0,1,0])	[0,0,1,0,0]
#3	([0,0,1,0,0], [0,0,0,0,1])	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,0,1]

Training Example	Encoded Context Word	Mean Context Word	Encoded Target Word
#1	([1,0,0,0,0],[0,0,1,0,0])	[0.5,0,0.5,0,0]	[0,1,0,0,0]
#2	([0,1,0,0,0],[0,0,0,1,0])	[0,0.5,0,0.5,0]	[0,0,1,0,0]
#3	([0,0,1,0,0],[0,0,0,0,1])	[0,0,0.5,0,0.5]	[0,0,0,1,0]
#4	([0,0,0,1,0])	[0,0,0,1,0]	[0,0,0,0,1]

Représentation vectorielle

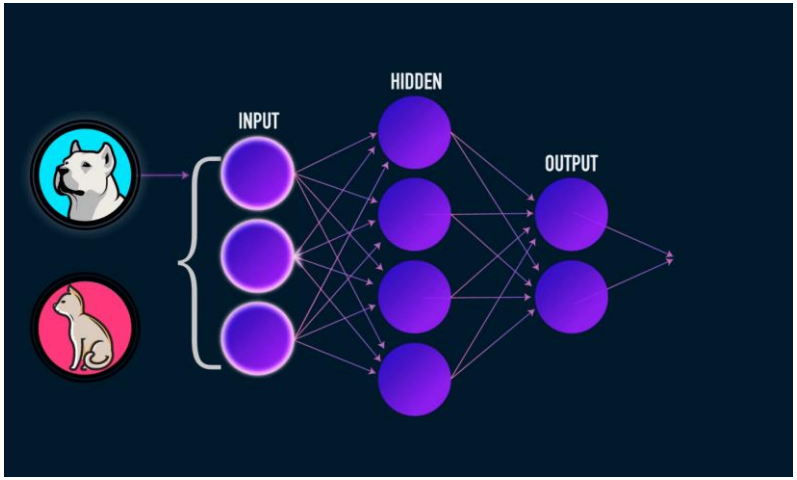
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation
  - Error Calculation
  - Weight tuning using backward pass - backpropagation

Repeat – plusieurs itérations



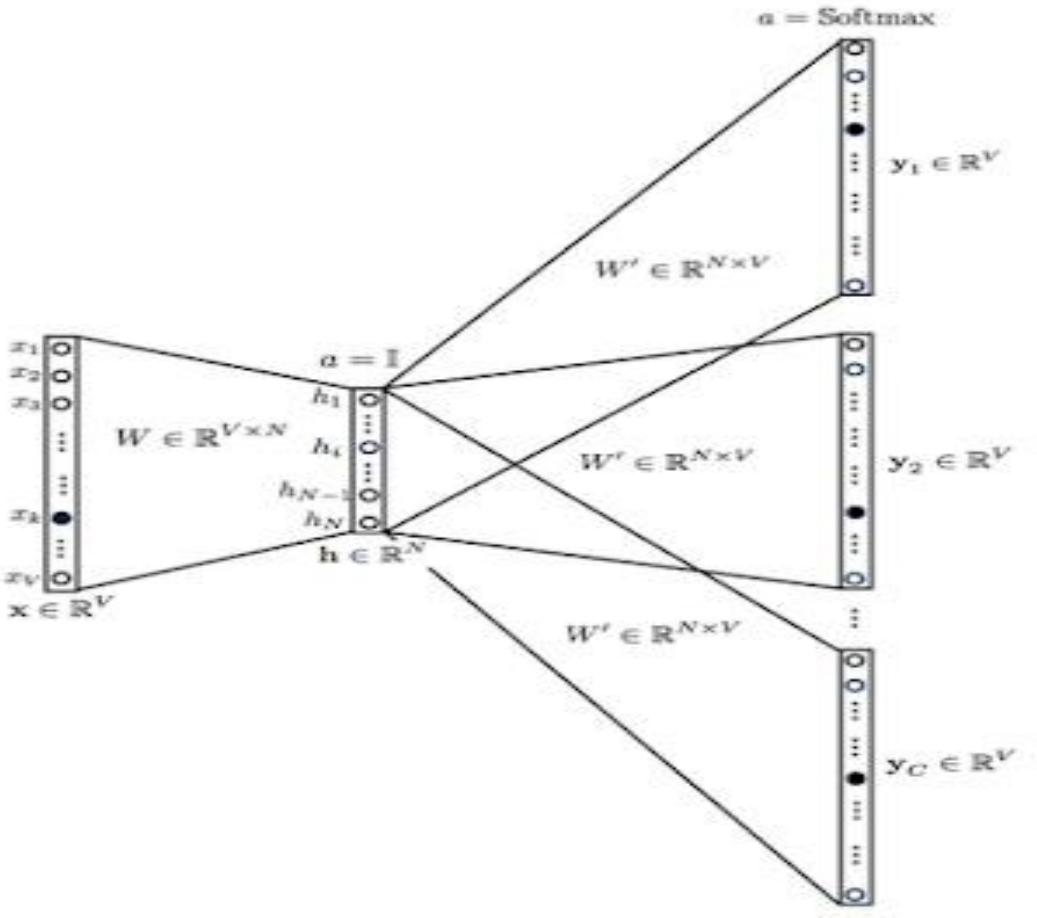
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Modèle d'Entraînement**:



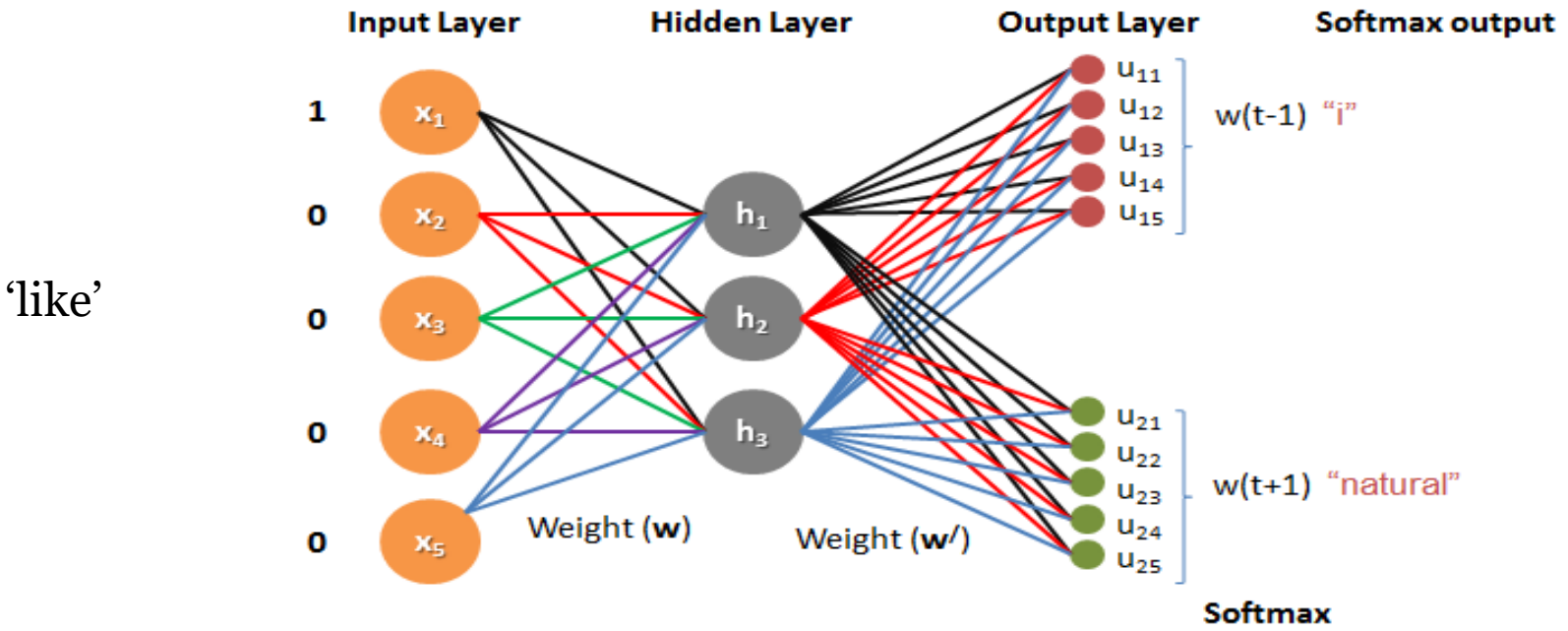
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Modèle d'entraînement** : Architecture - **window-size = 1**



'like'

First training data point: The context words are "**i**" and "**natural**" and the target word is "**like**".

Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Modèle d'entraînement** : Matrice des poids **W et W'**

$$w = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

Weight matrix for input to hidden layer

$$W' = \begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} & w'_{15} \\ w'_{21} & w'_{22} & w'_{23} & w'_{24} & w'_{25} \\ w'_{31} & w'_{32} & w'_{33} & w'_{34} & w'_{35} \end{bmatrix}$$

Weight matrix for hidden to output layer

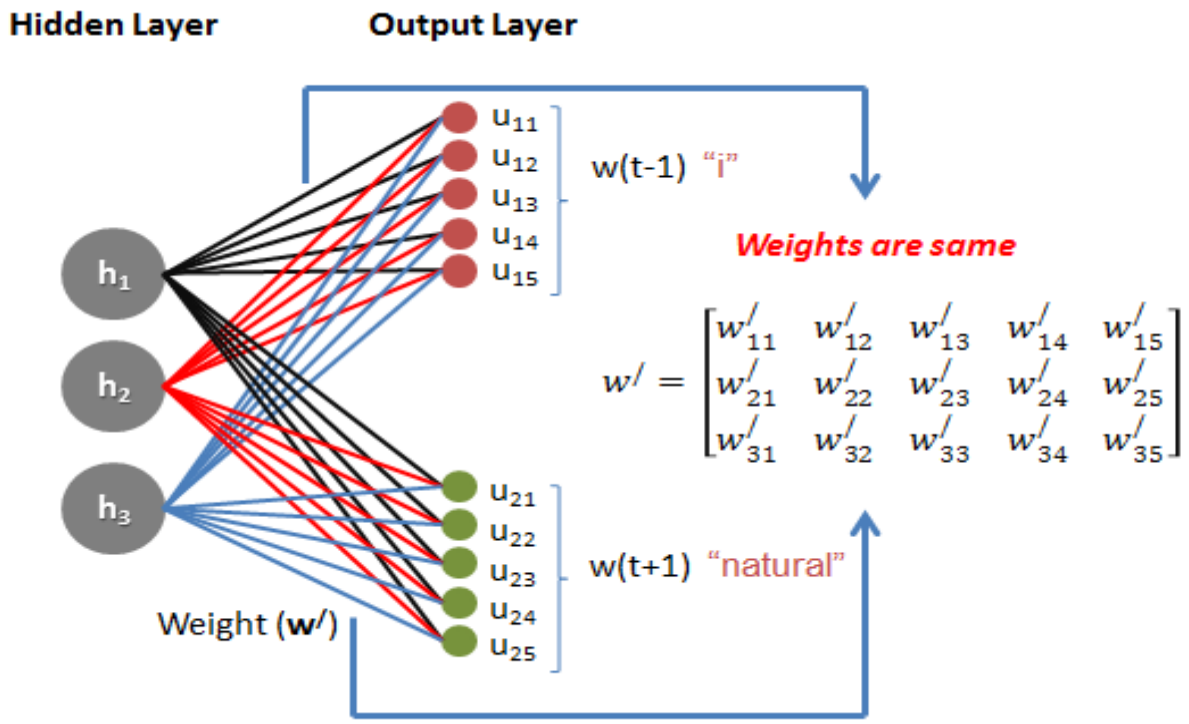
Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

Etapes:

- **Modèle d'entraînement** : Matrice des poids **W et W'**



$$w' = \begin{bmatrix} w'_{11} & w'_{12} & w'_{13} & w'_{14} & w'_{15} \\ w'_{21} & w'_{22} & w'_{23} & w'_{24} & w'_{25} \\ w'_{31} & w'_{32} & w'_{33} & w'_{34} & w'_{35} \end{bmatrix}$$

Les poids de chaque couche cachée vers couche de sortie sont les mêmes.

Représentation vectorielle

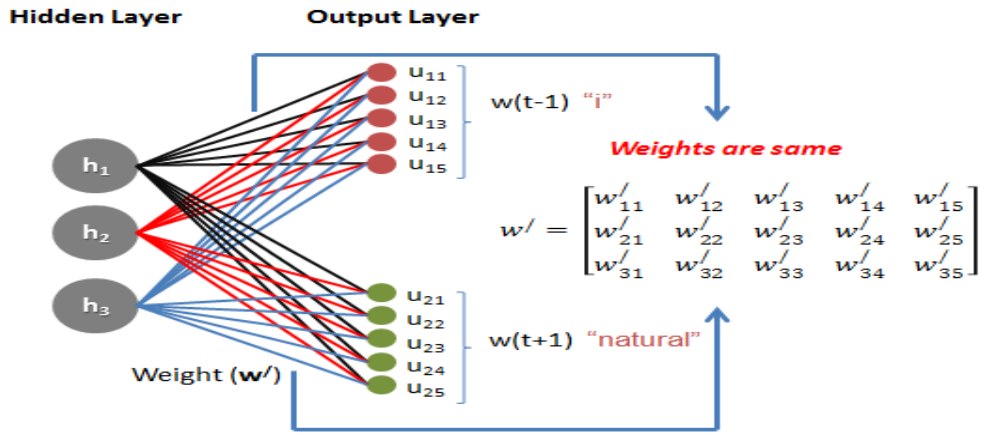
New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram

Etapes:

- **Modèle d'entraînement** : Entrainer un réseau de neurones : étapes
  - Create model Architecture
  - Forward Propagation – pareil que CBOW
  - **Error Calculation – la somme**
  - Backpropagation – pareil que CBOW

Repeat – plusieurs itérations



Représentation vectorielle

New Age Techniques
Prediction / Neural Network based vectorization approach

Word2Vec – Skip-Gram : Multi Word Model

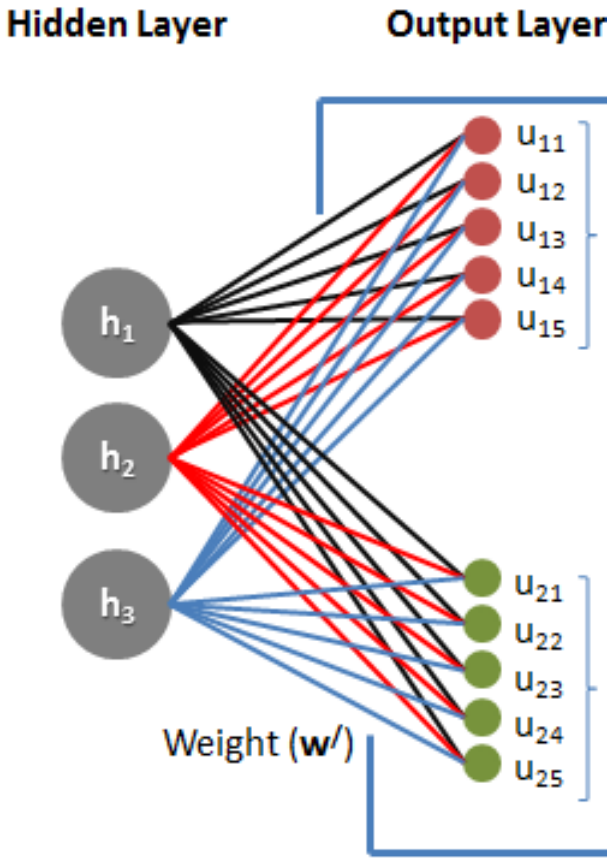
Etapes:

- **Modèle d'entraînement**: Error Calculation

C : the total number of context window

V: Vocabulary size

$$E = -\sum_{c=1}^{C} u_{c,j^*} + \sum_{c=1}^{C} \log \sum_{j=1}^{V} e^{u_{c,j}}$$



Références

Speech and Language Processing - Livre de Dan Jurafsk - https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Article - Xin Rong, word2vec Parameter Learning Explained https://arxiv.org/pdf/1411.2738.pdf

Anindya Naskar, Word2Vec, https://thinkinfi.com/continuous-bag-of-words-cbow-single-word-model-how-it-works/

Claudio Bellei – Word2Vec, http://www.claudiobellei.com/2018/01/06/backprop-word2vec/

Cours - ARIES Abdelkrime - Le traitement automatique du langage naturel. https://github.com/projeduc/ESI_2CS_TALN

Articles - Step by Step Guide to Master NLP, by CHIRAG GOYAL - https://www.analyticsvidhya.com/blog/