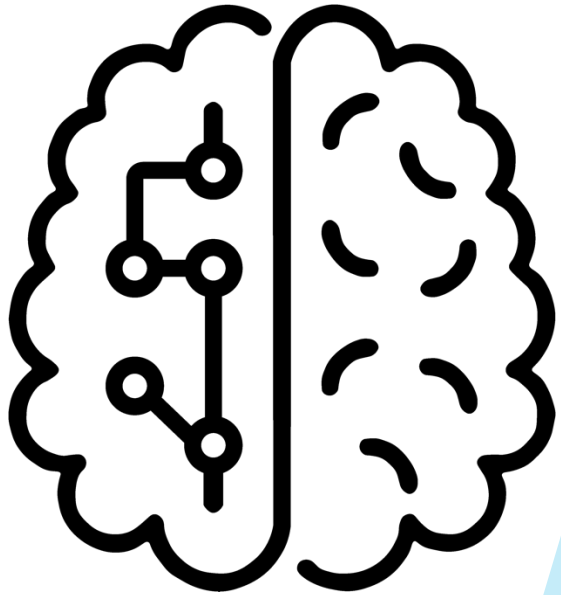
Word embeddings

AIForYou – Morgan Gautherot



Les matrices creuses

V	D_1	D_2	D_3	D_4	…	D_n
mot_1	0	$tfidf$	0	0	…	0
mot_2	0	0	0	$tfidf$	…	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
mot_m	0	$tfidf$	0	0	0	0

V	D_1	D_2	D_3	D_4	…	D_n
mot_1	0	1	0	0	…	0
mot_2	0	0	0	1	…	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
mot_m	0	1	0	0	0	0

Word embeddings

- Représentent les mots et les documents par des vecteurs
- Est une représentation qui capture le sens du mot ou du document

Proximité des mots

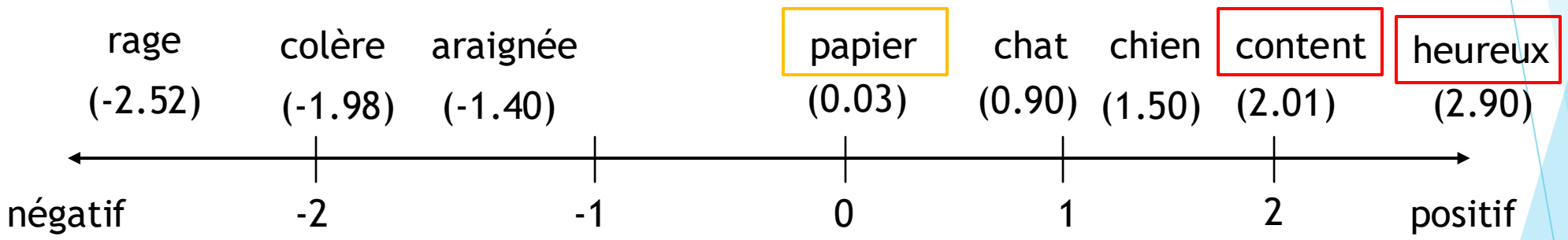
Content = Papier = Heureux

$$\text{Content} = [\ 0, \dots, 0, 0, \dots, 1, \dots, 0, \dots, 0]$$

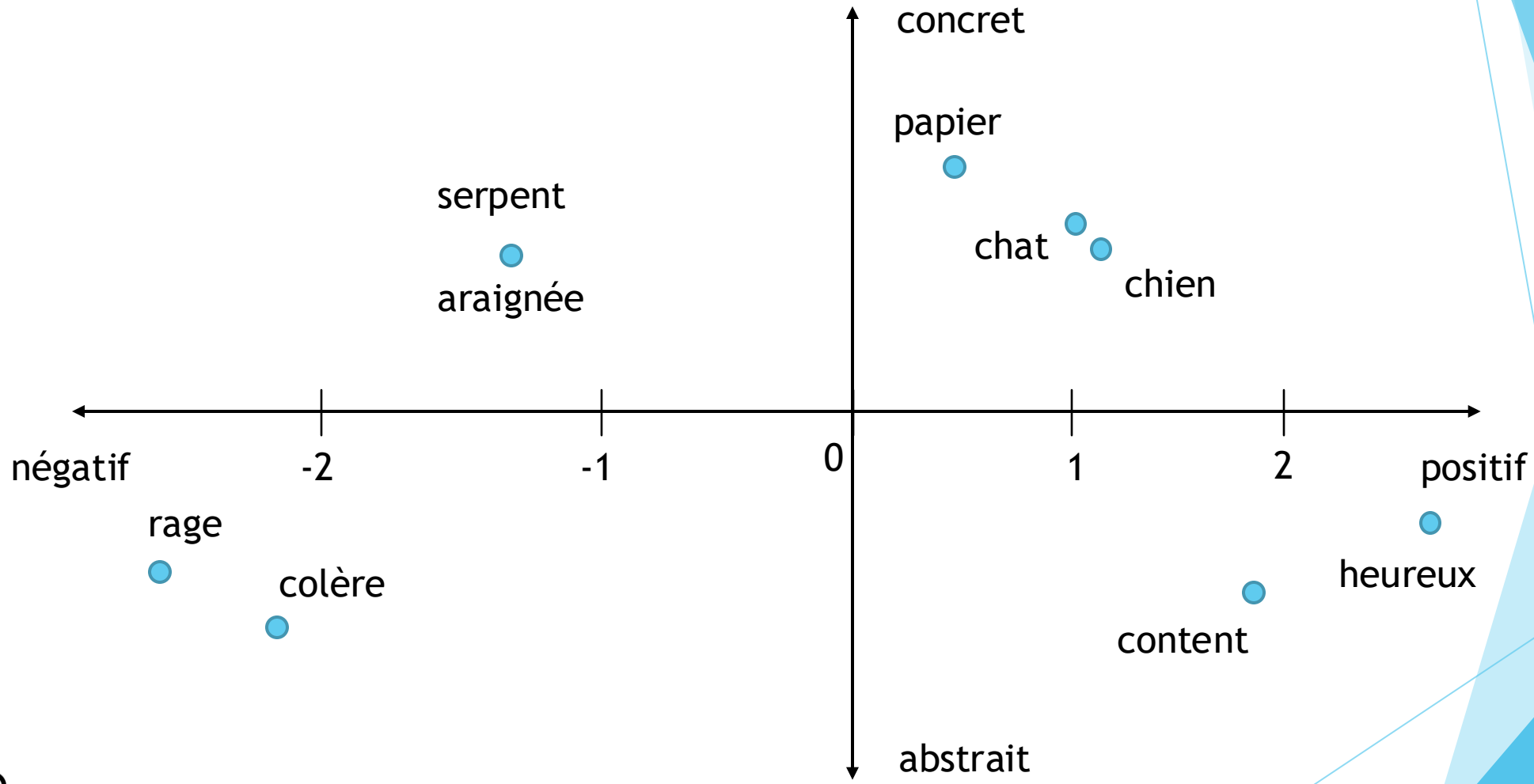
$$\text{Papier} = [\ 0, \dots, 0, 0, \dots, 0, \dots, 1, \dots, 0, \dots, 0]$$

$$\text{Heureux} = [\ 0, \dots, 1, 0, \dots, 0, \dots, 0, \dots, 0, \dots, 0]$$

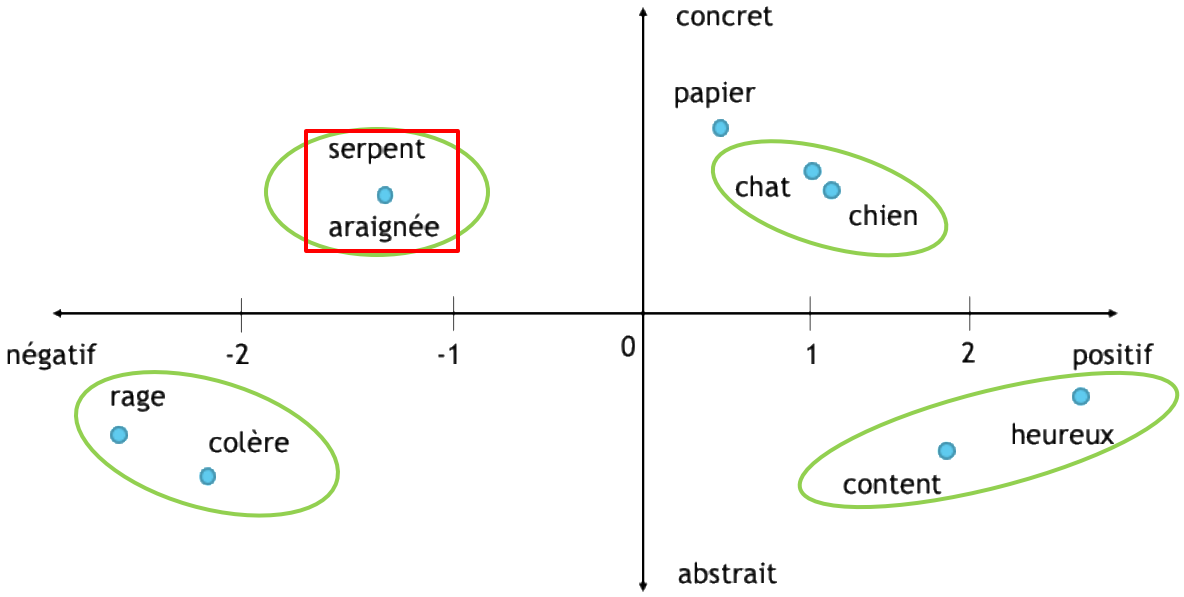
Représentation sous forme vectorielle



Représentation sous forme vectorielle



Représentation sous forme vectorielle



sens

petite dimension

précision

Mots	x_1	x_2
araignée	-1.40	0.41
…	…	…
content	2.01	-0.32
…	…	…
serpent	-1.40	0.41
…	…	…

Word2vec (Google 2013)

- Continuous bag-of-words (CBOW)
  - « J’aime lire le ??????? en buvant mon café »
- Continuous skip-gram / skip-gram with negaive sampling (SGNS)
  - « ???? ?????? journal ????? ????? »

GloVe (Stanford 2014)

- Factorisation du logarithme de la matrice de co-occurrence du corpus.

FastText (Facebook, 2016)

- Prend en compte la structure des mots en les représentants par des n-gram
- Permet d'utiliser des mots non vue pendant l'entraînement (OOV, out-of-vocabulary)
- Se base sur les lettres qui composent le mots pour créer l'embedding

Exemple :

Chaton ≈ Chat

Technique plus avancées d'embeddings

- Ces modèles ont des techniques de word embeddings qui change le vecteur d'un même mot suivant le contexte. (ex : orange, adjectif ou nom ?)

- BERT (Google, 2018)
- ELMo (Allen institute for AI, 2018)
- GPT-2 (OpenAI, 2018)

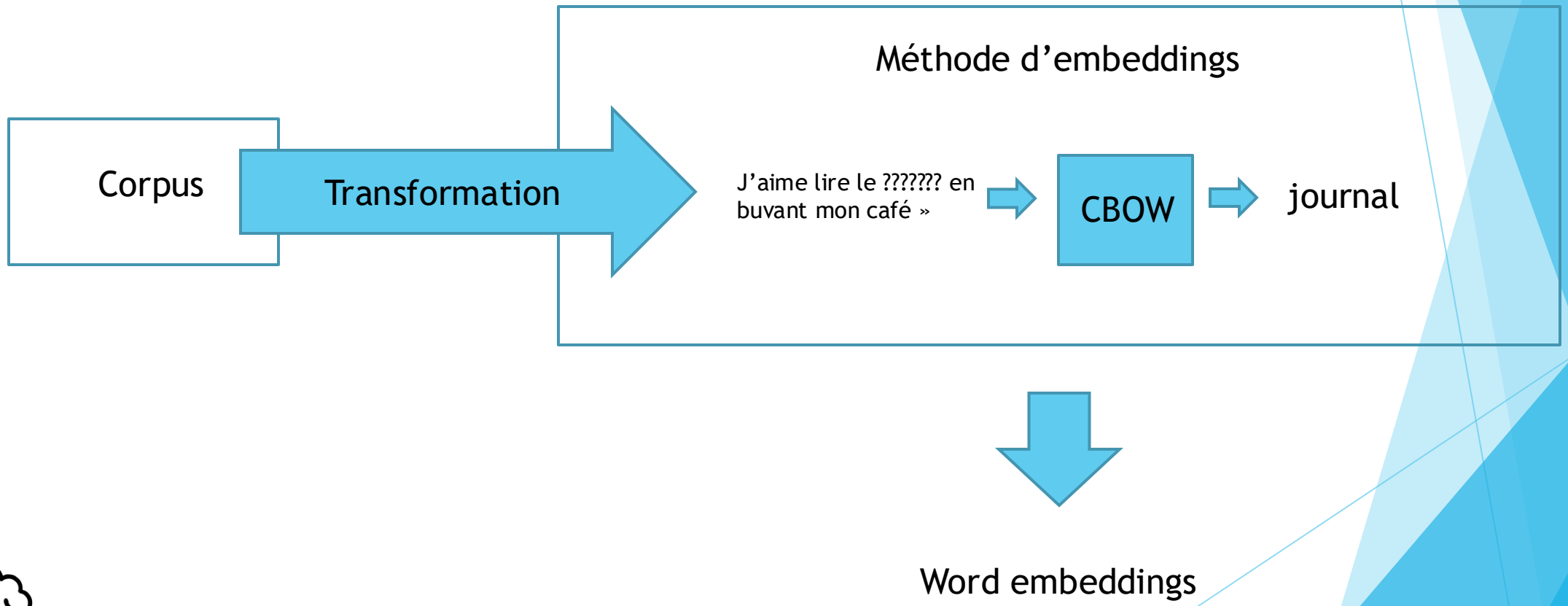
Des versions pré-entraîné de ces Modèles sont disponible sur internet

Continuous bag-of-words

J’aime lire le ??????? en buvant mon café »

Self learning = unsupervised learning + supervised learning

Continuous bag-of-words word embeddings



Distance entre deux mots

La signification des mots sera déterminé d'après leur contexte

J'aime lire un journal en buvant mon café »

J'aime lire un livre en buvant mon café »

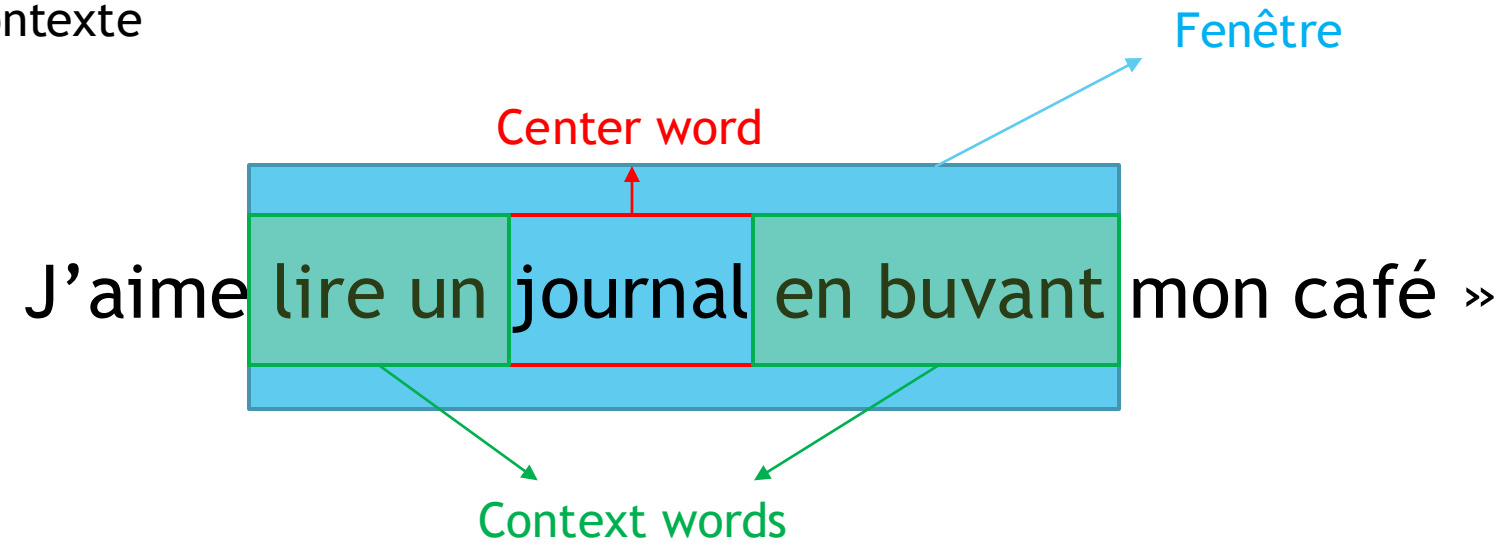
Livre ≈ Journal

Créer une observation du jeu d’entraînement

C = 2

Demi taille du contexte

Taille de la fenêtre = 5

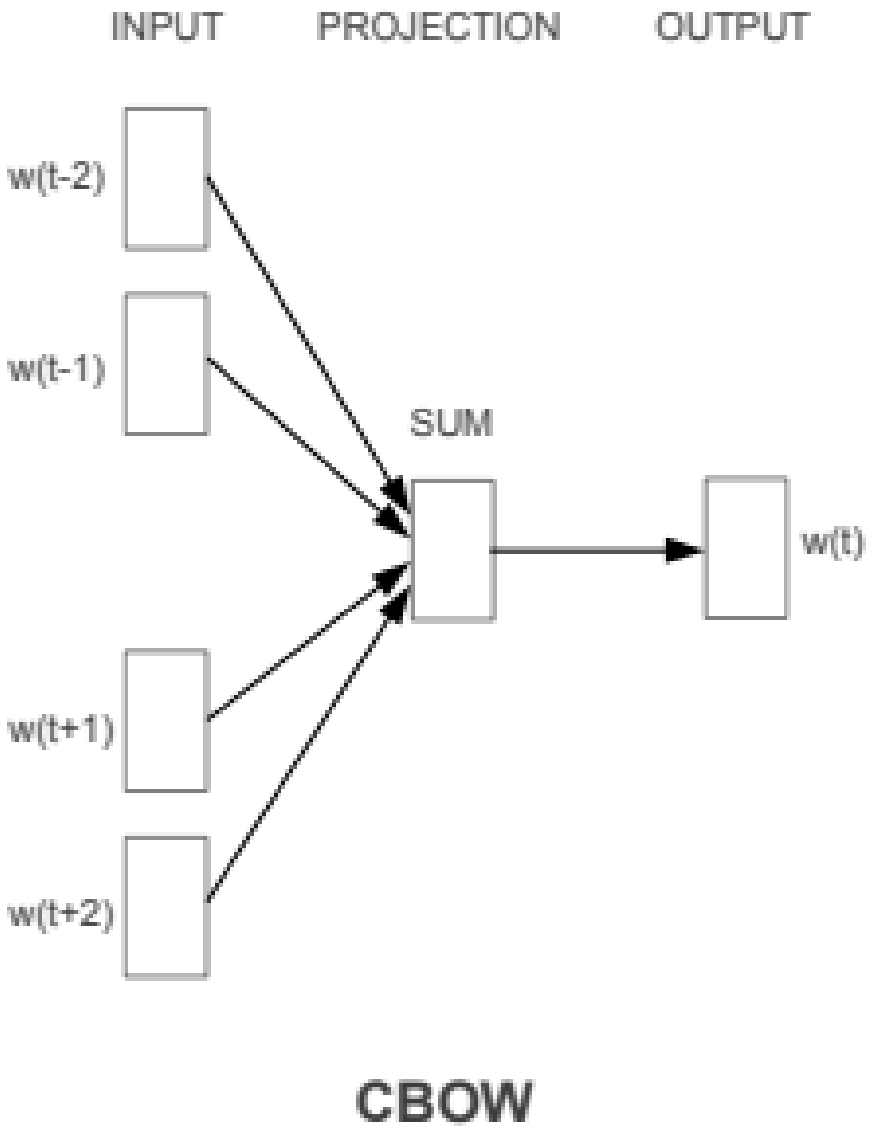


Extraire les mots

C = 2

Context word 1 (x_1)	Context word 2 (x_2)	Context word 3 (x_3)	Context word 4 (x_4)	Center word (y)
lire	un	en	buvant	journal
journal	en	mon	café	buvant
…	…	…	…	…

CBOW



Source : Efficient Estimation of Word Representations in Vector Space, Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean

Pré-traitement du corpus

- Tout mettre en minuscule
- Soit supprimer la ponctuation, soit tout remplacer par « . »
- Soit supprimer tout les nombres, soit tout remplacer par <NUMBER>
- Supprimer les caractères spéciaux $ * € …

Center word en vecteur

journal

$$V = \begin{bmatrix} arachide \\ \vdots \\ journal \\ \vdots \\ voiture \\ \vdots \\ zumba \end{bmatrix} \qquad w_{journal} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Context words en vecteur

Lire, un, en, buvant

$$V = \begin{bmatrix} arachide \\ \vdots \\ buvant \\ \vdots \\ en \\ \vdots \\ lire \\ \vdots \\ un \\ \vdots \\ zumba \end{bmatrix} \left[\begin{matrix} \text{lire} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{matrix} + \begin{matrix} \text{un} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \end{matrix} + \begin{matrix} \text{en} \\ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{matrix} + \begin{matrix} \text{buvant} \\ \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{matrix} \right] /4 = \begin{matrix} \text{Context vecteur} \\ \begin{bmatrix} 0 \\ \vdots \\ 0.25 \\ \vdots \\ 0.25 \\ \vdots \\ 0.25 \\ \vdots \\ 0.25 \\ \vdots \\ 0 \end{bmatrix} \end{matrix}$$

Données d'entraînement

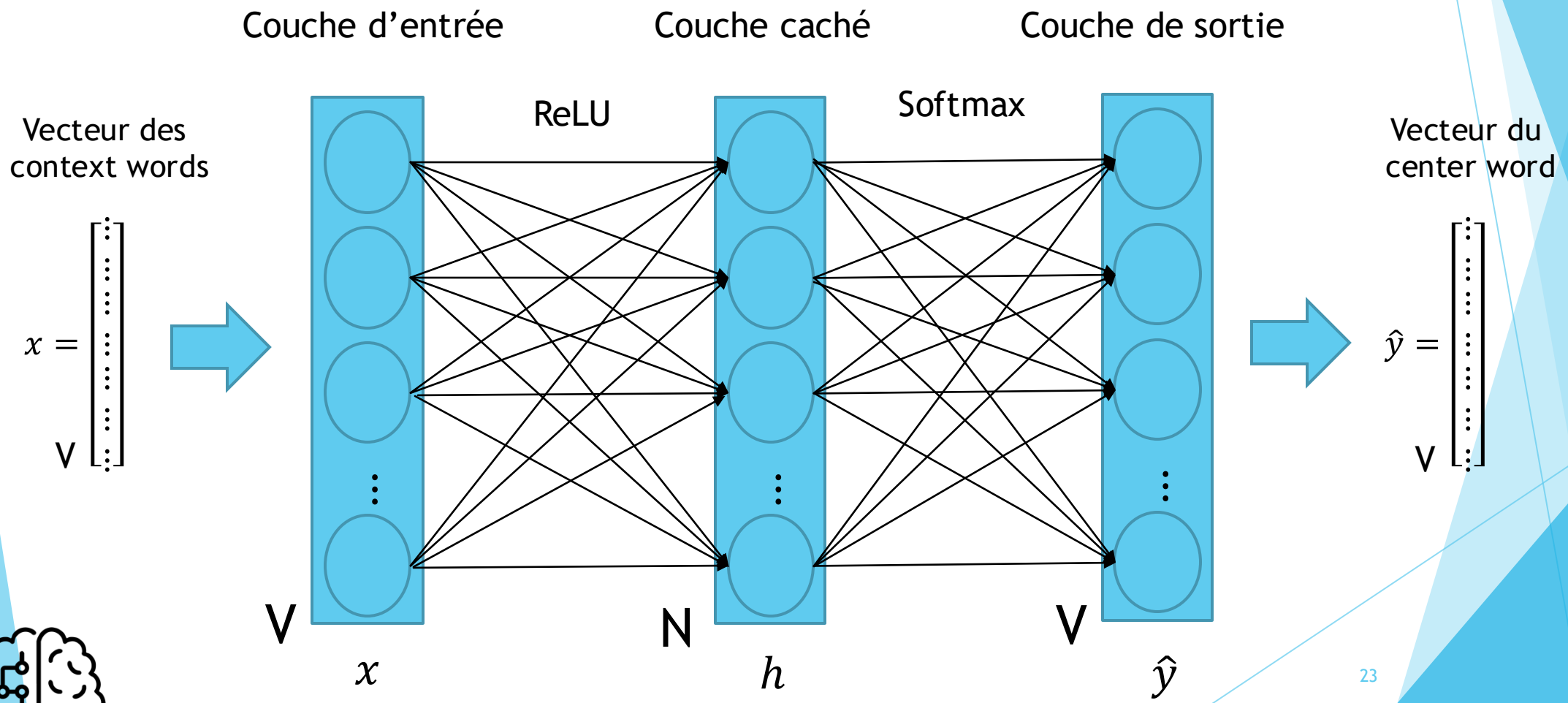
$$V = \begin{bmatrix} arachide \\ \vdots \\ buvant \\ \vdots \\ en \\ \vdots \\ journal \\ \vdots \\ lire \\ \vdots \\ un \\ \vdots \\ zumba \end{bmatrix} \quad x = \begin{bmatrix} 0 \\ \vdots \\ 0.25 \\ \vdots \\ 0.25 \\ \vdots \\ 0 \\ \vdots \\ 0.25 \\ \vdots \\ 0.25 \\ \vdots \\ 0 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Préparation des données TP 5

Objectifs :

- Pré-traiter les données textuelles
- Extraire les éléments de contexte des mots centraux
- Transformer les mots en vecteurs

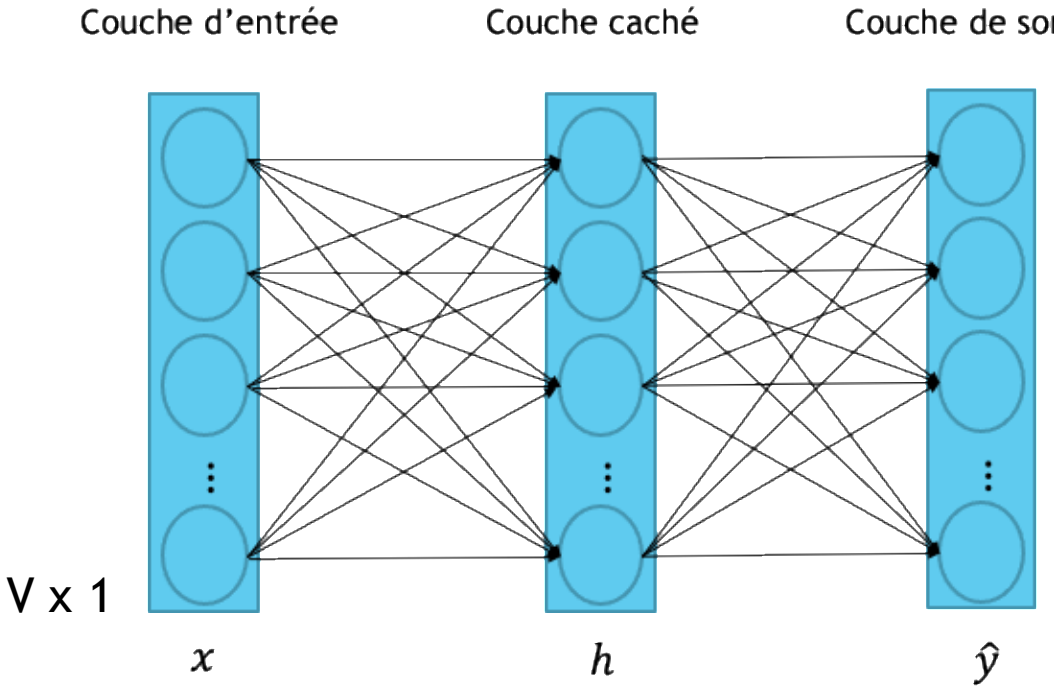
L’architecture du modèle CBOW



Dimension de l'architecture

Avec un seul exemple

$x =$



$z_1 = w_1 x + b_1$ N x 1

$h = ReLU(z_1)$ N x 1

$z_2 = w_2 h + b_2$ V x 1

$\hat{y} = softmax(z_2)$ V x 1

w_1 N x V

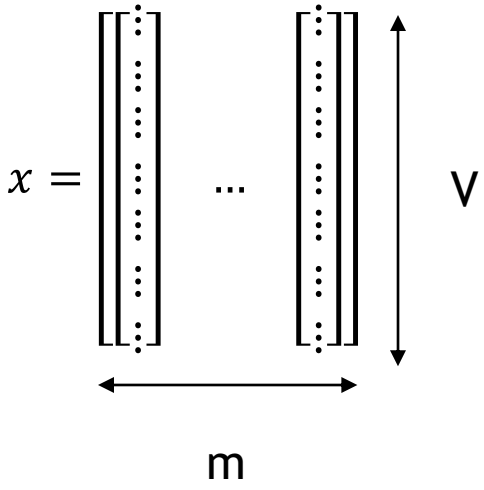
b_1 N x 1

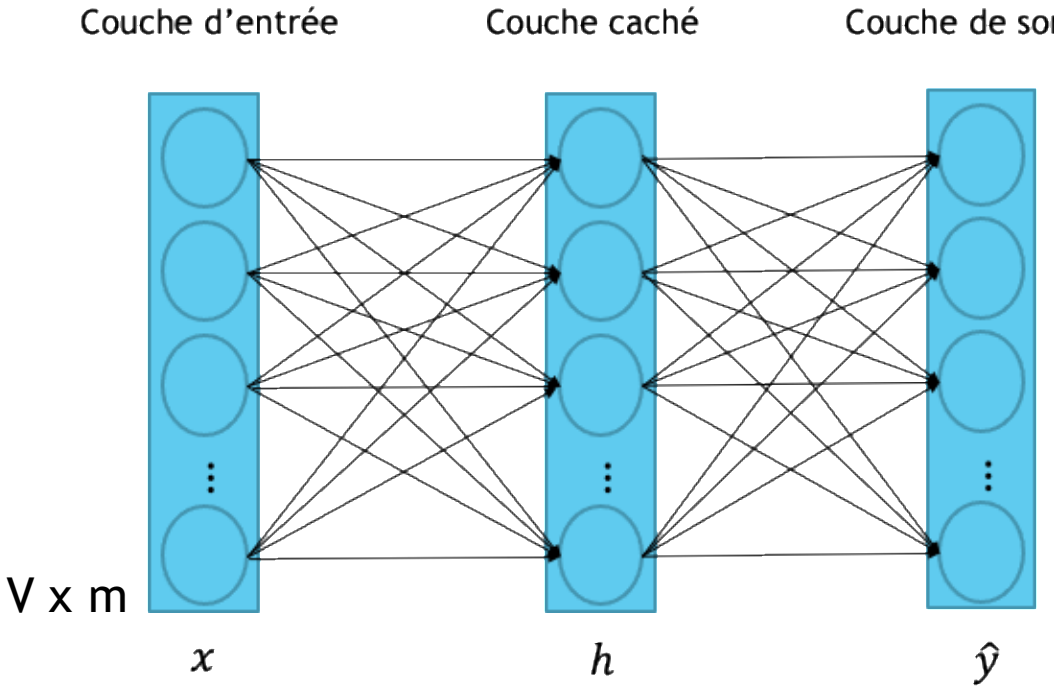
w_2 V x N

b_2 V x 1

Dimension de l'architecture

Avec *m* exemples





$z_1 = w_1 x + b_1$ N x m

$h = ReLU(z_1)$ N x m

$z_2 = w_2 h + b_2$ V x m

$\hat{y} = softmax(z_2)$ V x m

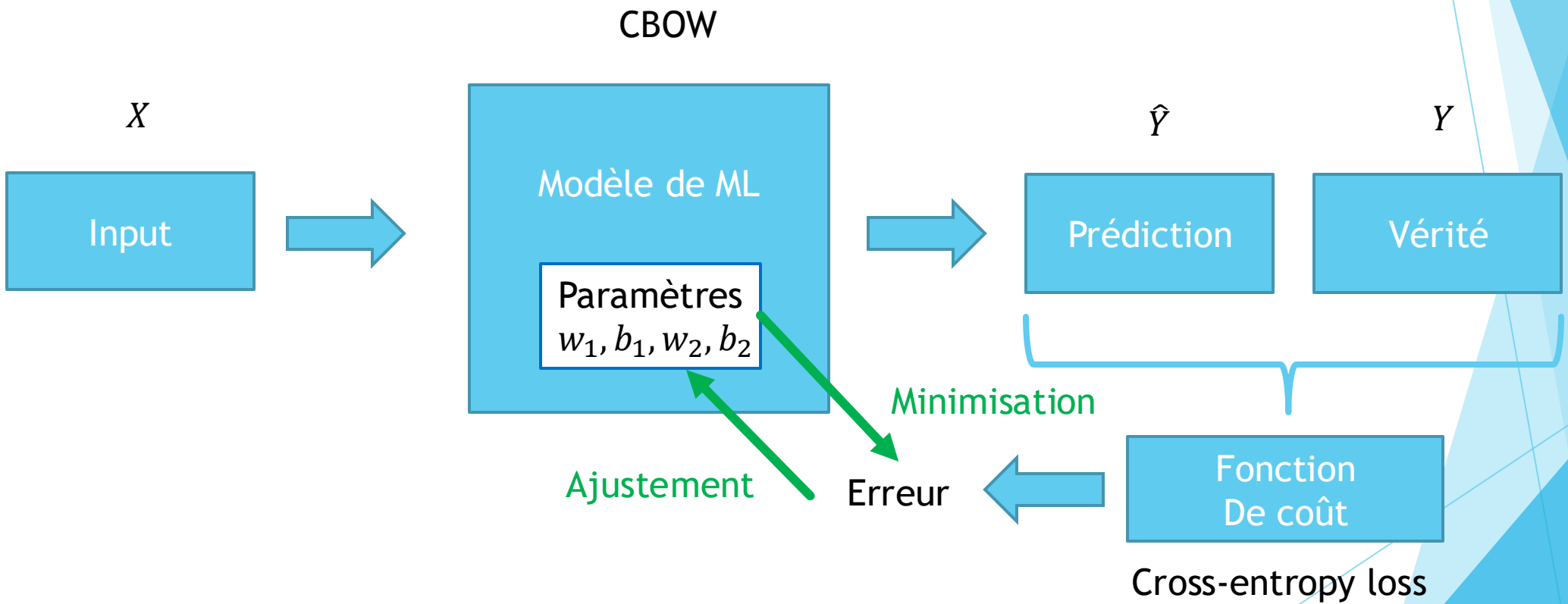
w_1 N x V

b_1 N x m

w_2 V x N

b_2 V x m

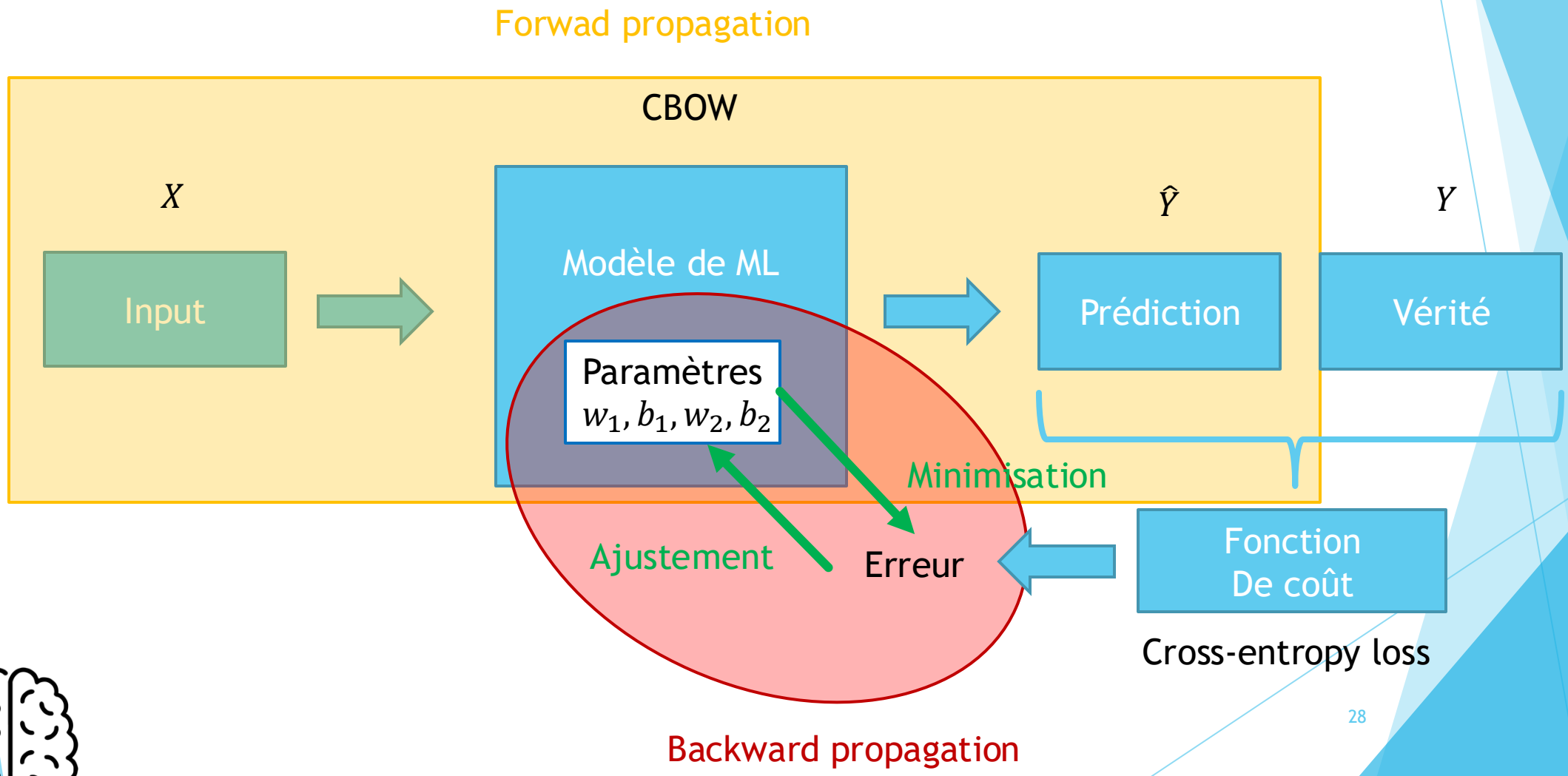
Entraînement du CBOW



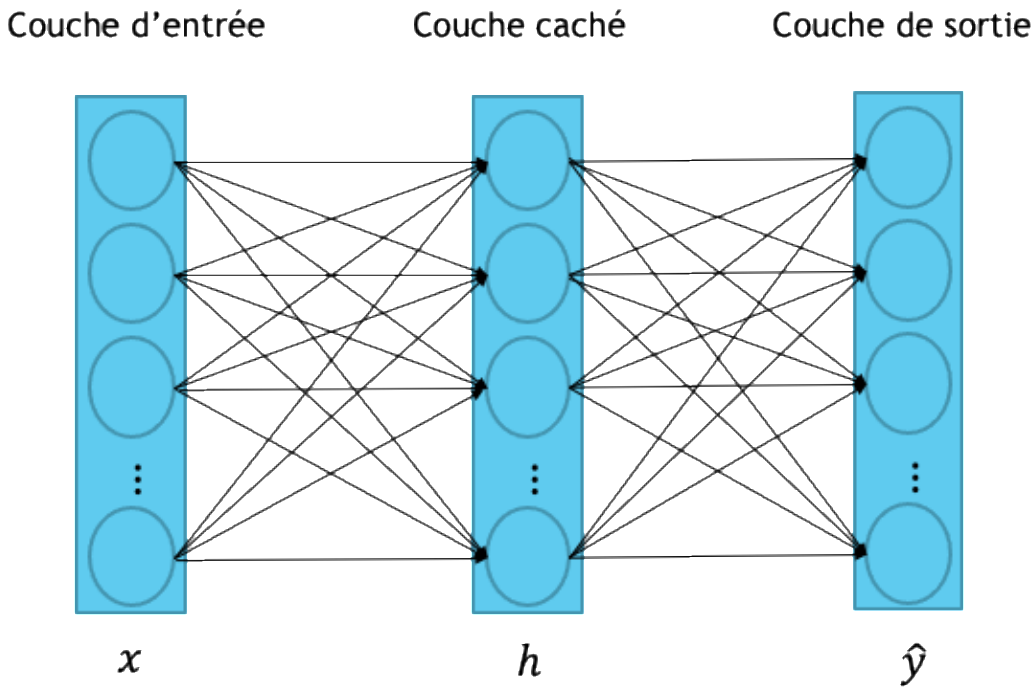
Cross entropy loss

$$J = -\sum_{k=1}^{V} y_k \log \hat{y}_k$$

Forward et backward propagation



Extraire la partie word embeddings option 1



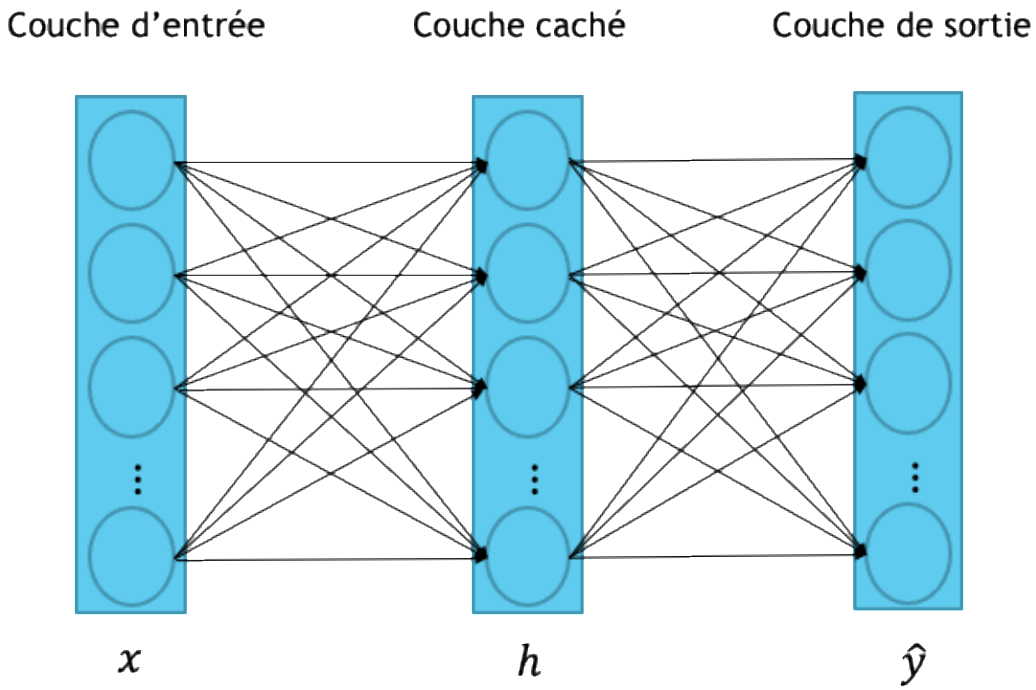
w_1 N x V

$$w_1 = [[w^{(1)}] \quad \dots \quad [w^{(V)}]]$$

N

V

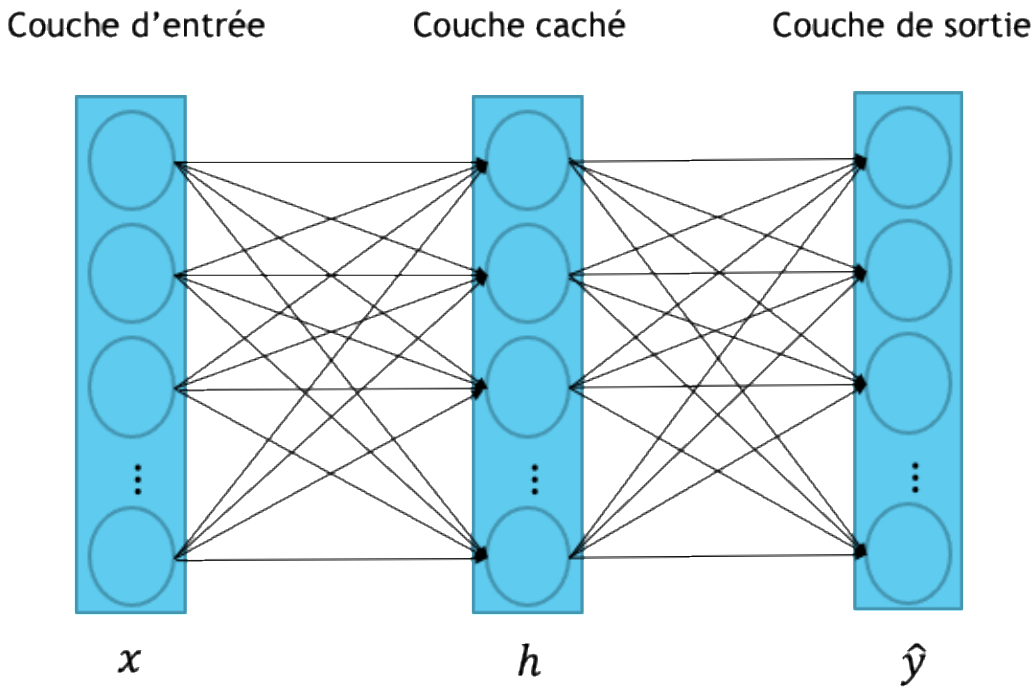
Extraire la partie word embeddings option 2



w_2 V x N

$$w_2 = \begin{bmatrix} [w^{(1)}] \\ \vdots \\ [w^{(V)}] \end{bmatrix} \quad \text{V}, \quad \text{N}$$

Extraire la partie word embeddings option 3



$$w_3 = 0.5(w_1 + {w_2}^T) = [[{w_3}^{(1)}] \quad \dots \quad [{w_3}^{(V)}]]$$

N

V

Implémenter le CBOW TP 6

Objectifs :

- Initialiser les poids du modèle
- Implémenter la fonction softmax
- Implémenter la forward propagation
- Implémenter la la backward propagation
- Implémenter le gradient descent

Entraîner un algorithme CBOW TP 7

Objectifs :

- Utiliser les fonctions du TP 5 pour préparer le jeu de données
- Initialiser une architecture de CBOW (avec Keras)
- Entraîner l'algorithme CBOW (avec Keras)
- Extraire le word embeddings pour les trois options