

Deep Learning Par la Pratique

Hugo Mougard



<https://www.orsys.fr/>

Table des matières

Deep Learning par la Pratique	2
Machine Learning	9
Réseaux de neurones	106
Principaux outils	369

Deep Learning par la Pratique

Présentation & coordonnées

Nom Hugo Mougard

Courriel hugo@mougard.fr

Activité Formateur et consultant en Machine Learning, Deep Learning & Python

Spécialité Traitement des langues et du code source

Parcours Master en Machine Learning & Traitement Automatique des Langues

Description

Cette formation présente les **fondamentaux** du deep learning à travers des travaux pratiques.

Prérequis

- Bonne connaissances du Machine Learning de ses principes de fonctionnement **théorique comme pratiques**
- Avoir un **compte Google** afin de pouvoir faire les TPs dans [Google Colaboratory](#)

Objectifs pédagogiques

- Comprendre l'évolution des réseaux de neurones vers le deep learning
- Utiliser TensorFlow/Keras
- Comprendre les principes de conceptions, les outils de diagnostic et les effets des différents verrous et leviers à disposition
- Mettre en pratique sur des problèmes réels

Ressources

Je vous ferai parvenir les ressources informatiques utilisées à chaque début de cours. Elles sont aussi accessibles via [My Orsys](#).

Emploi du temps

- 3 jours de 9h à 12h30 et de 14h à 17h30
- Le dernier jour, à 17h00 on finit de remplir les documents administratifs.

Tour de table : présentez-vous !

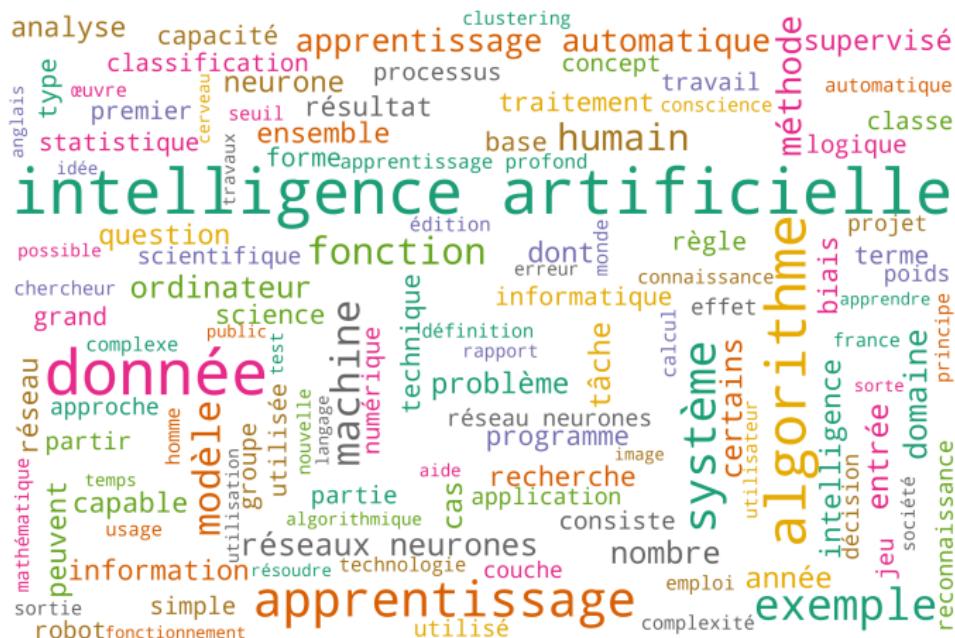
- Votre nom
- Votre métier
- Votre société client si applicable
- Vos compétences dans les domaines liés à cette formation
- Vos objectifs et vos attentes vis-à-vis de cette formation

Machine Learning

Machine Learning

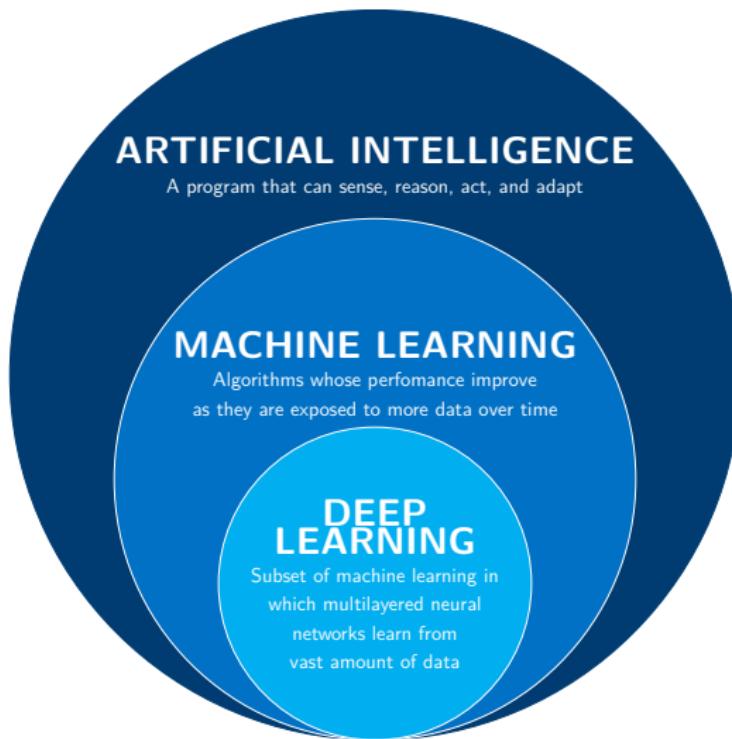
Introduction

Un domaine vaste



Nuage de mots liés à l'intelligence artificielle, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Hiérarchie des noms



Machine Learning

Nouvelle manière d'aborder la conception logicielle.

Changement de paradigme

Programmation explicite → programmation implicite



Processus d'ingénierie du ML, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

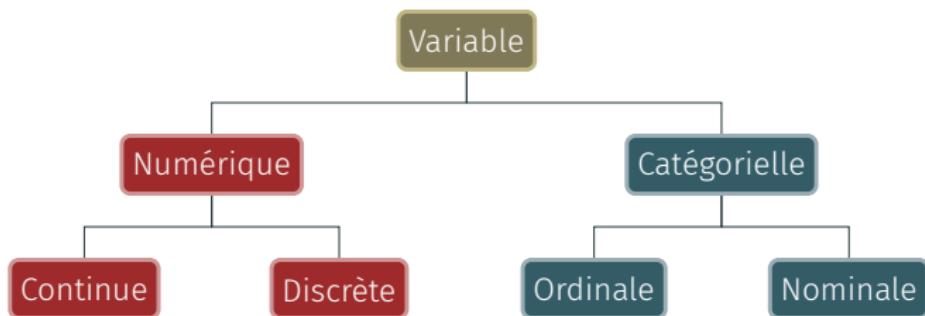
Matière première : les données

On cherche une fonction qui a :

- en entrée : N données en dimension D
- en sortie : N sorties en dimension K

Matière première : les données

Chaque dimension du problème est représentée par une variable :



Les différents types de variables, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Grandes familles

Apprentissage supervisé, non-supervisé, par renforcement, ...

Apprentissage supervisé

Prédire une valeur numérique (régression) ou l'appartenance à une classe (classification).

Apprentissage non-supervisé

Faire émerger des profils, des groupes.

Exemple

Groupes de clients pour adapter sa stratégie marketing

Apprentissage par Renforcement

Apprendre une stratégie efficace dans un univers où les actions fournissent des récompenses (possiblement négatives).



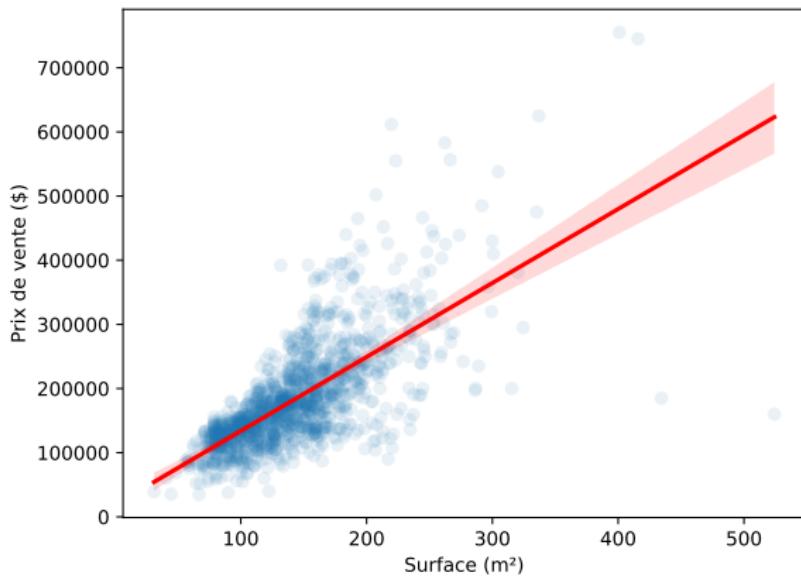
Plateau d'échecs, Membre PublicDomainPictures de Pixabay, Pixabay License.



Stanford Racing et Victor Tango à la même intersection pendant la finale du DARPA Urban Challenge, gouvernement des États-Unis d'Amérique, domaine public.

Exemple — Régression linéaire

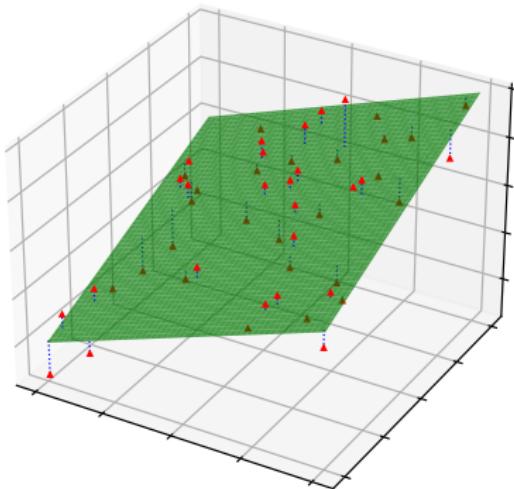
Prédire une valeur en fonction d'une autre.



Régression linéaire du prix en fonction de la surface, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

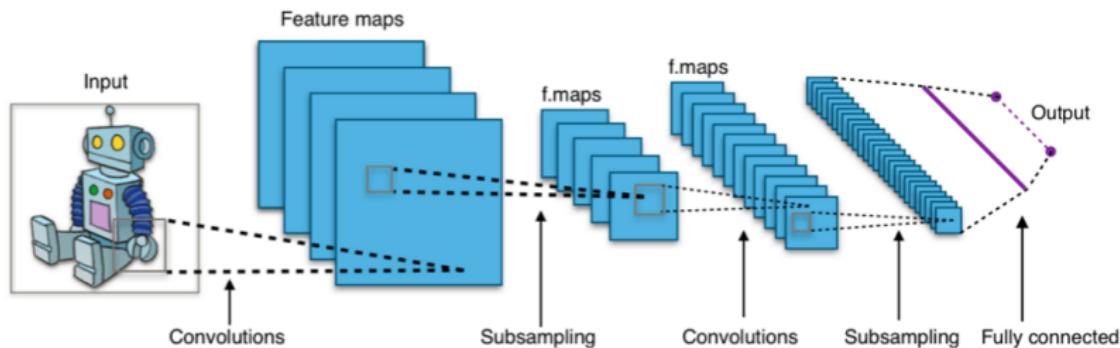
Exemple — Régression linéaire multiple

Prédire une valeur en fonction de plusieurs autres.



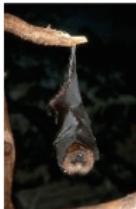
Extension de la régression linéaire quand X est de dimension 2, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Exemple – Classification avec des réseaux à convolutions



Typical CNN architecture, membre de Wikimedia Aphex34, CC-BY-SA-4.0.

Exemple – Classification avec des réseaux à convolutions

[001.ak47](#)[002.american-flag](#)[003.backpack](#)[004.baseball-bat](#)[005.baseball-glove](#)[006.basketball-hoop](#)[007.bat](#)[008.bathtub](#)[009.bear](#)[010.beer-mug](#)[011.billiards](#)[012.binoculars](#)

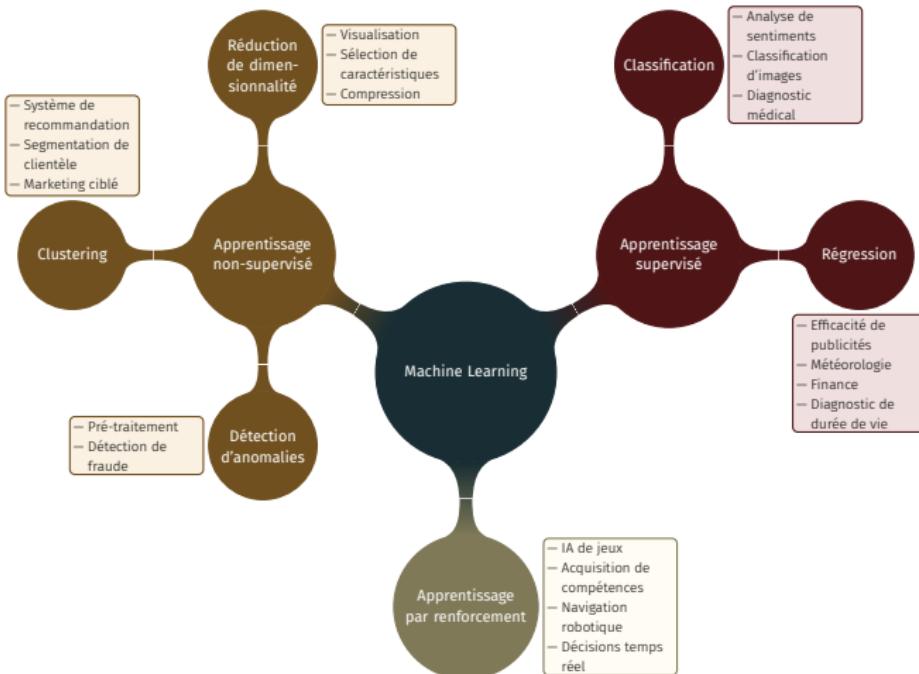
Caltech-256 Object Category Dataset, G. Griffin et al., Caltech.

Exemple — Apprentissage par renforcement



Tableau de bord, Membre FotoRech de Pixabay, Pixabay License.

Topologie du domaine



Les différentes branches du Machine Learning, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Points de vue

Beaucoup de façons de voir le machine learning. Basées sur :

- les paradigmes (supervisé, non supervisé, renforcement, en ligne, ...)
- les modèles (arbres, grammaires, automates, réseaux de neurones ...)
- les données (tabulaire, image, texte, vidéo, graphe, ...)
- les techniques (statistiques, symboliques, probabilistes, ...)
- les contraintes (real time, embarqué, big data, multilingue, ...)

→ Domaine **extrêmement** vaste.

Choisir la bonne facette

Critères pour s'orienter dans les approches de machine learning:

- quantité de données à disposition
- qualité du signal d'apprentissage dans les données
- difficulté du problème à résoudre
- besoin d'interprétabilité
- contraintes techniques
- contraintes de délai
- ... et d'autres en fonction des domaines métiers

Conclusion

- le machine learning est un champ vaste.
- il existe sûrement un modèle/paradigme pour vos besoins
- l'important est de définir les bons critères

Discussion

- à quelles données allez-vous appliquer le machine learning ? À quels besoins ?
- aurez-vous besoin de modèles interprétables ou simplement très performants en prédiction ?
- quelles sont vos contraintes ?

Machine Learning

Prérequis

Machine Learning

Prérequis

Quelques prérequis

Objectifs

- exprimer des transformations de données grâce à l'algèbre linéaire
- minimiser des fonctions analytiquement
- décrire l'incertain
- décrire des données

Machine Learning

Prérequis

Algèbre linéaire

Utilité

- décrire des transformations simples sur un dataset entier avec des mécanismes adaptés
- comprendre les possibilités et les limites de ces transformations simples.

Transformation linéaire

- algèbre linéaire = on se limite aux sommes pondérées des inputs.
- bonne nouvelle : énorme partie des opérations en machine learning

Description des données – échantillon

Python :

```
data = (1, 3)
```

Algèbre linéaire :

$$\mathbf{d} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Description des données – dataset

Python :

```
data = [(1, 3),  
        (2, 2),  
        (4, 2)]
```

Algèbre linéaire :

$$\mathbf{D} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix}$$

Description des transformations linéaires

Python :

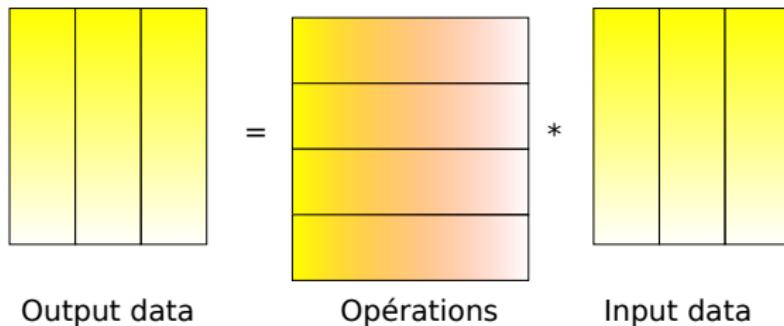
```
def weights(x, y):  
    return x * 2 + y / 2
```

Algèbre linéaire :

$$\mathbf{w} = \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix}$$

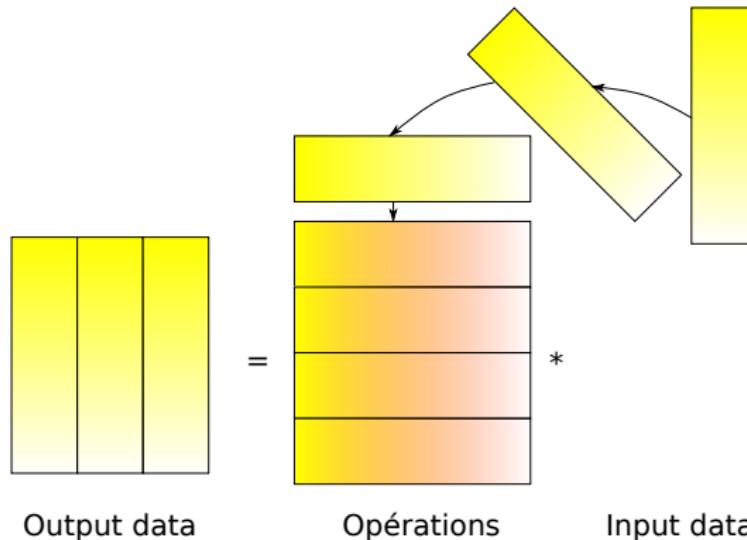
Transformation linéaire = somme pondérée.

Application d'une transformation linéaire à un exemple



Bonne intuition à garder : Verser les colonnes (les exemples du dataset) dans les lignes (les opérations).

Bonne intuition à garder



Bonne intuition à garder : Verser les colonnes (les exemples du dataset) dans les lignes (les opérations).

Application d'une transformation linéaire à un exemple

Python :

```
data = (1, 3)

def weights(x, y):
    return 2 * x + y / 2

res = weights(*data)
```

Algèbre linéaire :

$$\begin{aligned}f &= \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \\&= 2 \times 1 + \frac{1}{2} \times 3\end{aligned}$$

Application d'une transformation linéaire à un dataset

Python :

```
data = [(1, 3),  
        (2, 2),  
        (4, 2)]  
  
def f(x, y):  
    return x * 2 + y / 2  
  
res = [f(x, y)  
      for x, y  
      in data]
```

Algèbre linéaire :

$$\text{res} = \begin{bmatrix} 2 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 3,5 & 5 & 9 \end{bmatrix}$$

Application de plusieurs transformations linéaires à un dataset

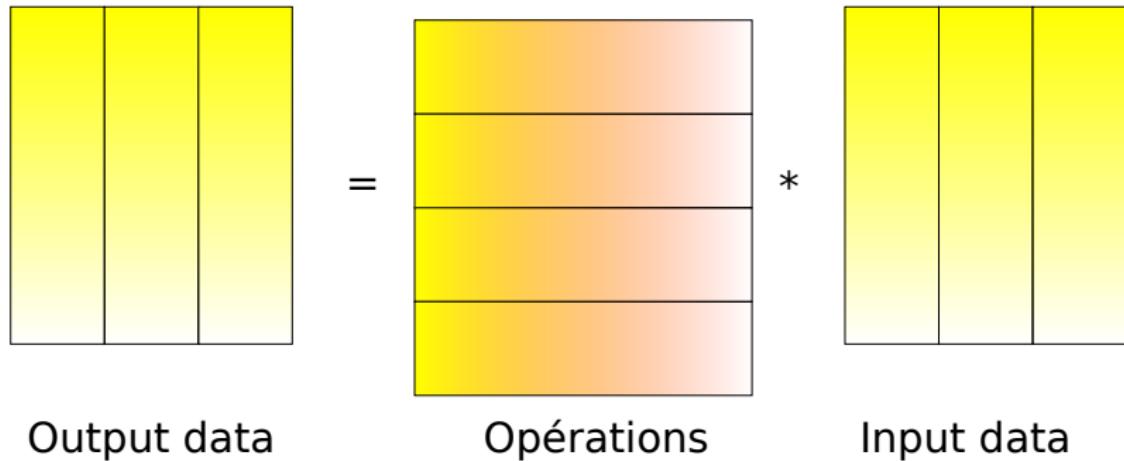
Python :

```
data = [(1, 3), (2, 2),
         (4, 2)]  
  
def f(x, y):  
    return x * 2 + y / 2  
  
def g(x, y):  
    return x / 2 + y * 2  
  
res = [[t(x, y) for x, y
           in data]
         for t in [f, g]]
```

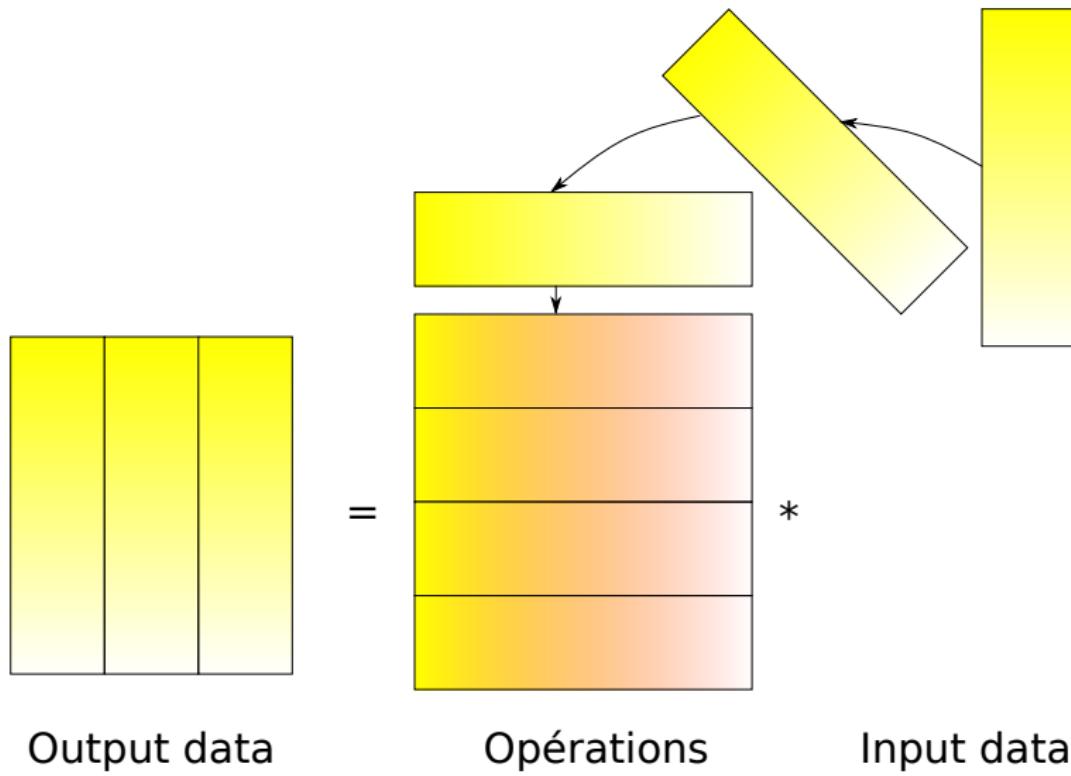
Algèbre linéaire :

$$\begin{aligned} \text{res} &= \begin{bmatrix} 2 & \frac{1}{2} \\ \frac{1}{2} & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 3 & 2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 3,5 & 5 & 9 \\ 6,5 & 5 & 6 \end{bmatrix} \end{aligned}$$

Bonne intuition à garder



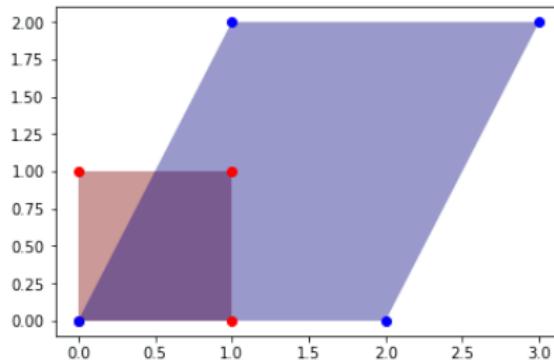
Bonne intuition à garder



Exercice

$$\begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} = ?$$

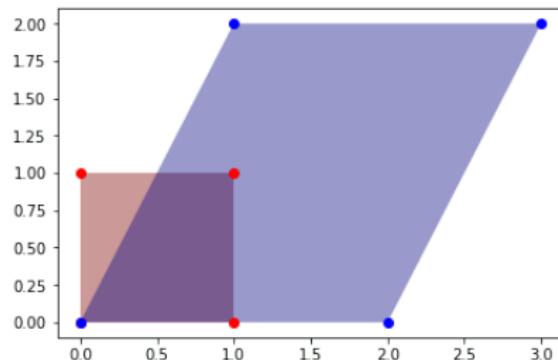
Exemple de transformation



Bleu = Transformation \times Rouge

$$\begin{aligned} &= \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 3 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix} \end{aligned}$$

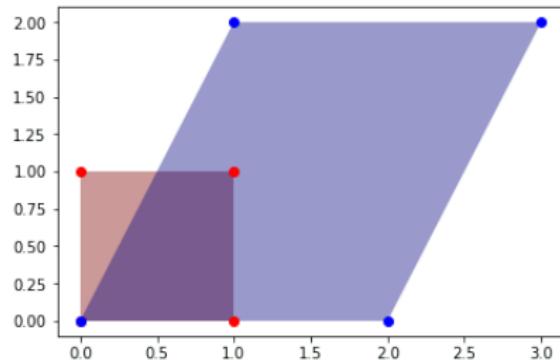
Vecteur propre



Vecteur partant de l'origine qui conserve sa direction malgré la transformation.

Pouvez-vous en trouver un ? $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ par exemple.

Valeur propre



Facteur par lequel un vecteur propre est redimensionné.

Quelle est la valeur propre de $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$? 2.

Machine Learning

Prérequis

Analyse

Utilité

Souvent besoin de minimiser une fonction en machine learning.

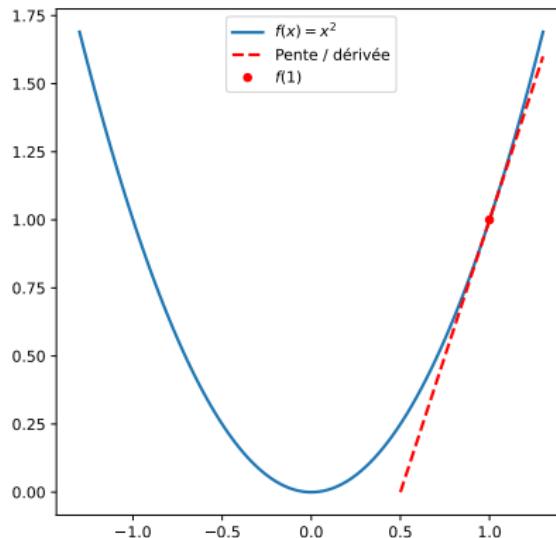
Idée clef

Décider d'un x de départ puis suivre la pente jusqu'au minimum.

Pente = dérivée

→ Modifier itérativement x par un pas vers l'opposé de la dérivée.

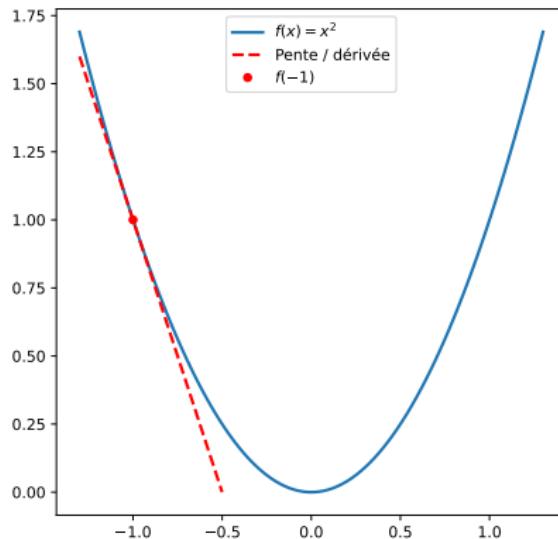
Pente positive



Pente au point $x = 1$ de la fonction x^2 , F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Opposé de la pente = -2 . Avec un pas de $0,1$, on passe de 1 à $0,8$.

Pente négative

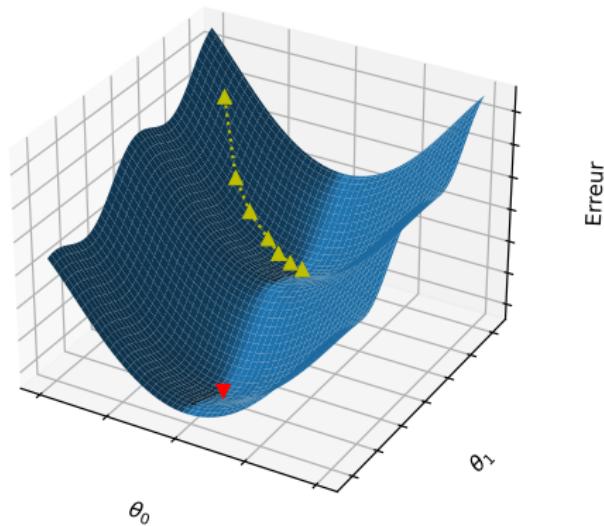


Pente au point $x = -1$ de la fonction x^2 , F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Opposé de la pente = 2. Avec un pas de 0,1, on passe de -1 à -0,8.

Exemple en 2 dimensions

Dérivée → gradient



Surface de l'erreur en fonction des paramètres θ_0 et θ_1 , F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Machine Learning

Prérequis

Probabilités

Utilité

- quantifier l'incertain
- support pour les statistiques

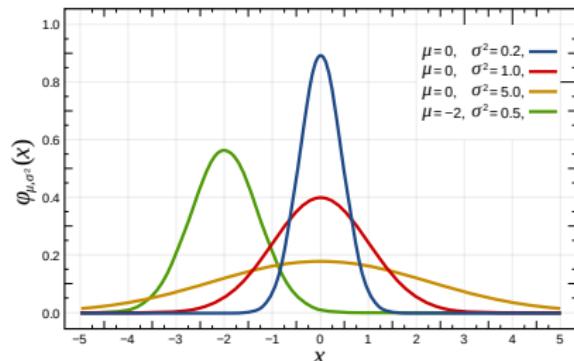
Probabilité

- la probabilité de l'événement X est notée $P(X)$
- $P(X) \in [0, 1]$
- $P(X) = 0 \iff X$ est impossible
- $P(X) = 1 \iff X$ est certain
- $P(\neg X) = 1 - P(X)$

Loi de probabilité

Décrit le comportement aléatoire d'un phénomène dépendant du hasard.

- $\sum_u P(X = u) = 1$ en discret
- $\int P(X)dX = 1$ en continu
- loi uniforme
- loi normale/gaussienne



Lois normales, membre de Wikimedia
Inductiveload, Domaine public.

Machine Learning

Prérequis

Probabilités

Rappels - Probabilités

Rappels :

- Une variable aléatoire

$$A \in \mathbb{R}$$

- Probabilité

$$0 \leq P(A \in [a_1 a_2]) \leq 1$$

- Probabilité conditionnelle

$$P(A > 0 | B < -3)$$

- Évènements indépendants

$$P(A|B) = P(A) \text{ et } P(B|A) = P(B)$$

- Probabilité jointe

$$P(A, B) = P(B|A) * P(A)$$

$$P(A, B) = P(A|B) * P(B)$$

- A et B Indépendants

$$\iff P(A, B) = P(A) * P(B)$$

Théorème de Bayes

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Machine Learning

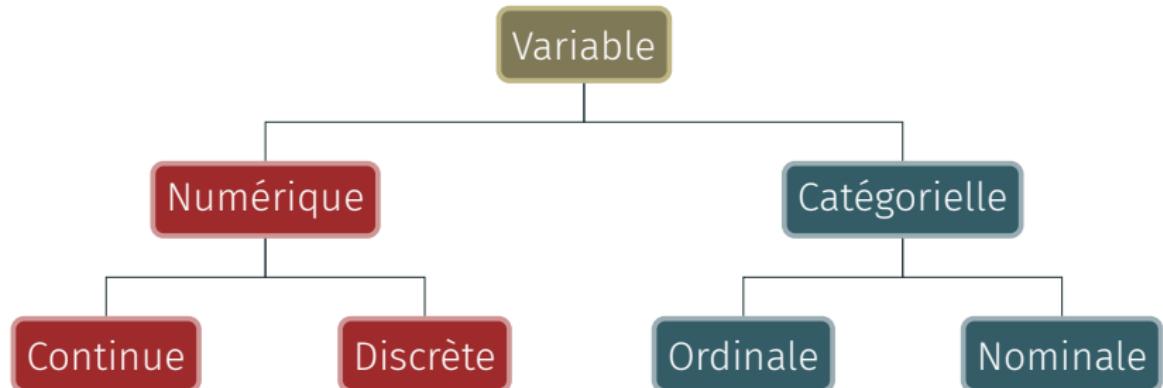
Prérequis

Statistique

Utilité

- description et compréhension des données
- correction pour faciliter les traitements

Types de variables



Les différents types de variables, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Hypothèse

Pré-requis pour les mesures statistiques qui suivent (et la plupart du machine learning) :

- les données doivent être issues d'une même loi
- chaque échantillon doit être indépendant des autres
- pas évident en pratique ! Pourquoi ?

Variance

Mesure la dispersion d'une série statistique (ou d'une variable) :

$$V(X) = \mathbb{E} [(X - \mathbb{E}[X])^2]$$

Pour la calculer :

$$V(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

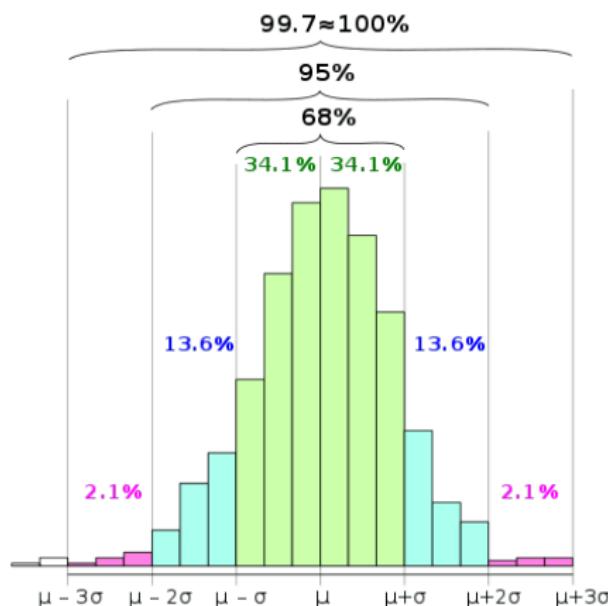
Écart-type

Racine carrée de la variance

$$\sigma(X) = \sqrt{V(X)}$$

Écart-type – règle des 68, 95 et 99,7

Pour les lois normales :



Règle des 68–95–99,7, Membre de Wikimedia Melikamp, CC-BY-SA-4.0.

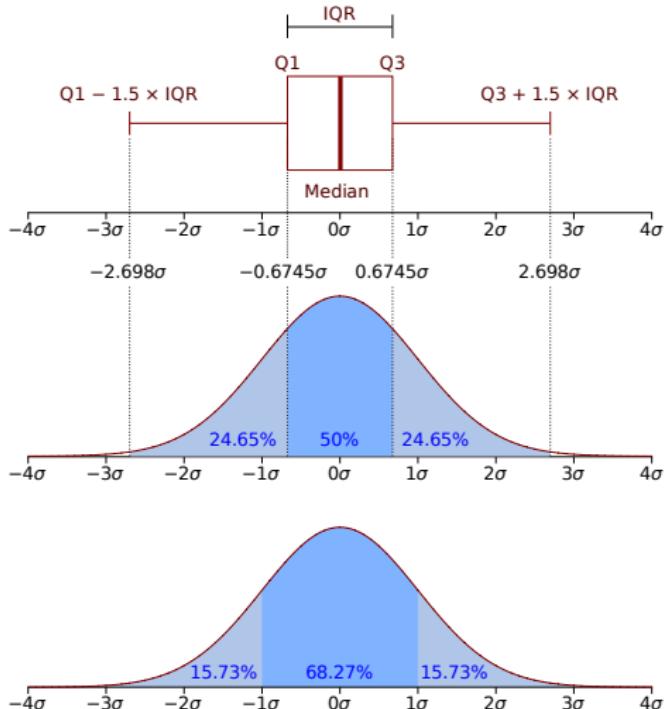
Quartile

Les quartiles (Q_1 , Q_2 et Q_3) divisent les données en 4 intervalles contenant le même nombre d'observations.

Déclinable en quantile de taille arbitraire (décile, percentile).

Que veut dire être dans le 95^e percentile ?

Boxplot



Loi normale et boxplot, membre de Wikimedia Jhguch, CC BY-SA 2.5.

ORSYS · Deep Learning Par la Pratique · CC BY-SA 4.0

Covariance

Mesure la variabilité jointe de deux variables aléatoires :

$$V(X) = \mathbb{E} [(X - \mathbb{E}[X])(X - \mathbb{E}[X])]$$

$$\text{cov}(X, Y) = \mathbb{E} [(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Pour la calculer :

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Corrélation

Covariance divisée par le produit des écart-types :

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

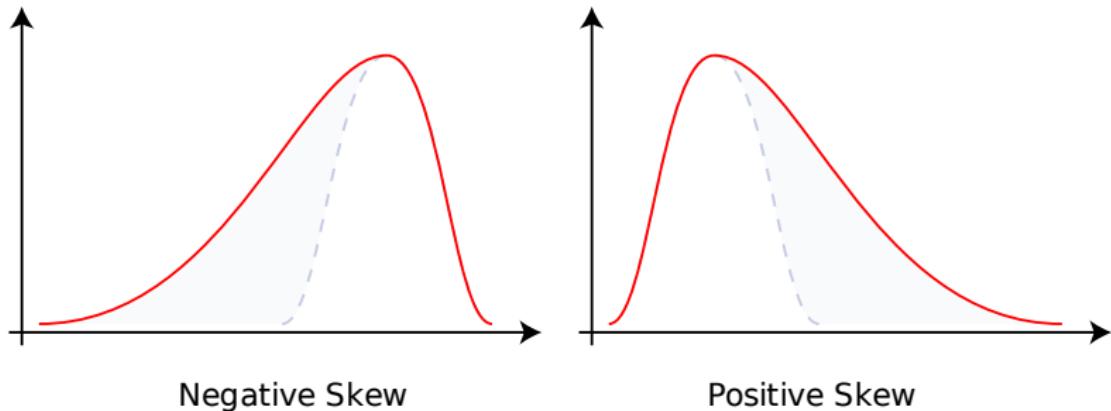
Intérêt ? Pas d'unité.

Test de normalité

Pour tester (et corriger) la normalité d'une distribution, on utilise deux mesures :

- l'asymétrie (*skew*)
- le kurtosis

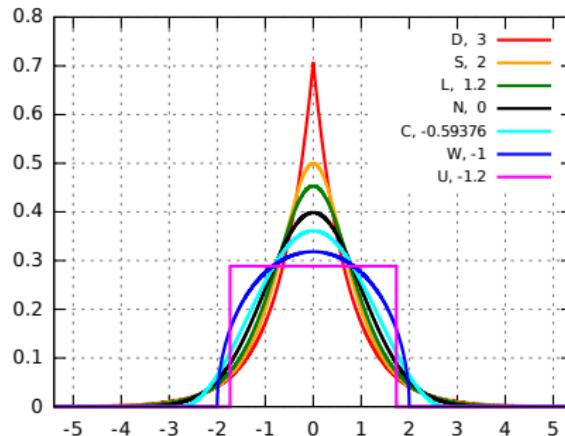
Asymétrie



Skews positives et négatives, membre de Wikimedia Rodolfo Hermans, CC-BT-SA-3.0.

$$\text{asym}(X) = \mathbb{E} \left[\left(\frac{X - \bar{X}}{\sigma} \right)^3 \right]$$

Kurtosis



Différentes formes de densité, membre de Wikimedia MarkSweep, CC-Zero.

$$\text{kurt}(X) = \mathbb{E} \left[\left(\frac{X - \mu}{\sigma} \right)^4 \right]$$

Transformation de Box-Cox

Asymétrie et kurtosis peuvent se corriger avec la transformation de Box-Cox ou des transformations log.

Machine Learning

Prérequis

Conclusion

Conclusion

- algèbre linéaire → raisonner sur des opérations simples et les décrire efficacement
- minimiser une fonction continue → dérivée
- décrire l'incertain → probabilités
- caractériser une série de données → statistiques

Avez-vous des questions ?

Ressources complémentaires – Algèbre

- [Le produit scalaire](#)
- [Le produit matriciel](#)
- [Vecteurs propres](#)

Ressources complémentaires – Statistiques

- Statistiques - bases
- Covariance
- Corrélation

Machine Learning

Modélisation et préparation des données

Introduction



Processus d'ingénierie du ML, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Biais statistiques

Attention aux différents biais de vos données !

- **variables confondantes** (Ex : “obésité” dans la corrélation entre “conso. viande” et “cancer colon”)
- **biais statistiques**
 - sélection, autosélection
 - mesure
 - attrition
 - ...
- trouver de fausses variables explicatives

→ Le garder en tête pendant toute l'étude.

Qualité des données

Meilleures données > Meilleurs modèles
(trash-in, trash-out)

→ À garder en tête pendant toute l'étude, en particulier durant l'entraînement de modèles

Pipeline de préparation

- valeurs manquantes
- préprocessing (texte, image)
- standardisation
- transformation

Valeurs manquantes

Gênant pour certains modèles. Plusieurs options :

- supprimer les enregistrements
- remplacer par une valeur (imputation) :
 - constante
 - moyenne de la colonne
 - prédiction d'un autre modèle

Prétraitement

- tokenizer, POS-tagger le texte (<https://spacy.io/>)
- utiliser un réseau de neurones préentraîné sur les images (<https://keras.io/applications/>)
- appliquer une transformée de fourier sur le son
- ...

Standardisation

Beaucoup de modèles travaillent mieux avec des données normales et sont plus efficaces autour de $[-5, 5]$:

- centrer sur la moyenne puis diviser par l'écart-type
- transformation de Box-Cox en cas d'asymétrie
- transformations spécifiques en fonction de la distribution

Transformation

Quand un modèle n'accepte pas de données catégorielles :

- label encoding si ordinal
- one-hot encoding sinon

Label encoding

Si les données sont ordinales :

Ordinal :

Température
Froid
Froid
Tiède
Chaud
Tiède

Label encoding :

Température
1
1
2
3
2

One-hot encoding

Remplacer une feature par n features avec n le nombre de catégories.

Catégoriel :

Couleur
Rouge
Rouge
Jaune
Vert
Jaune

One-hot :

	Rouge	Jaune	Vert
Rouge	1	0	0
Rouge	1	0	0
Jaune	0	1	0
Vert	0	0	1
Jaune	0	1	0

Exploration des données

But :

- se rendre compte des prétraitements à effectuer (Box-Cox, imputations, etc)
- comprendre la variable de sortie : distribution, équilibre des classes, features les plus corrélées, ...
- détecter les corrélations
- appréhender la complexité nécessaire du modèle

Attention : garder des données de côté (test set) et ne pas les regarder. Sinon biais statistique énorme.

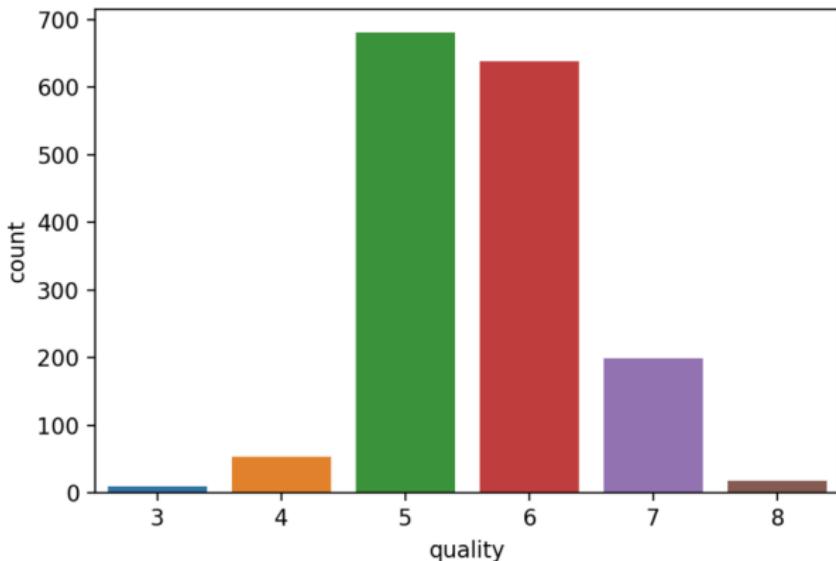
Outils

Plusieurs outils sont disponibles pour explorer des données. On utilise principalement des plots pour :

- se renseigner sur une distribution
- se renseigner sur la corrélation de deux distributions
- visualiser des corrélations linéaires

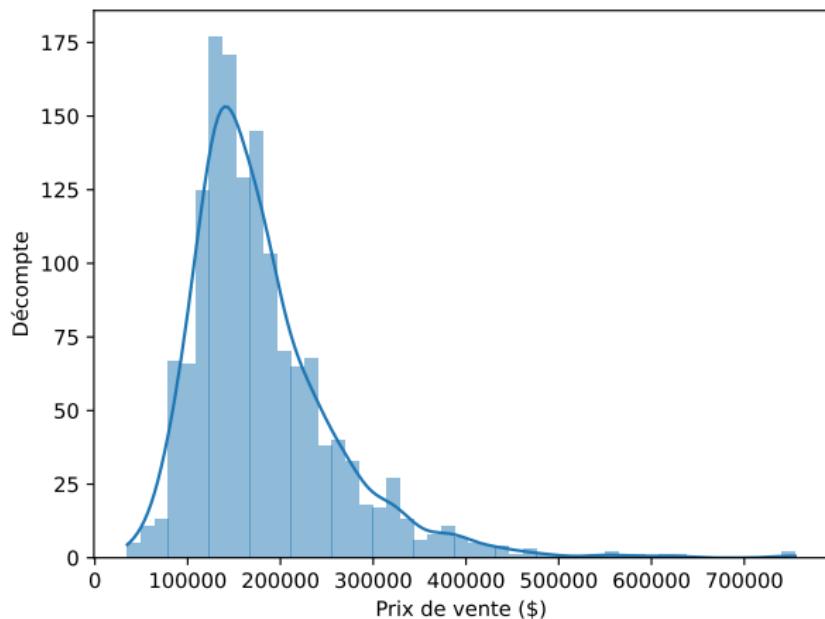
Les outils suivants sont sauf mention contraire présents dans [seaborn](#).

Outils – count plot



`seaborn.countplot(df['quality'])`, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – Histogramme



Exemple d'histogramme avec seaborn.histplot, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – Diagramme quantile-quantile

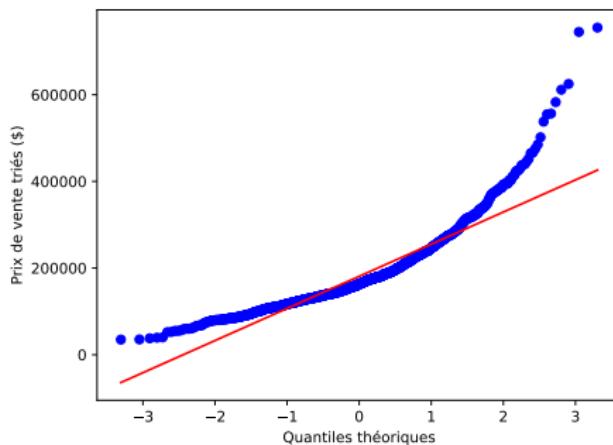
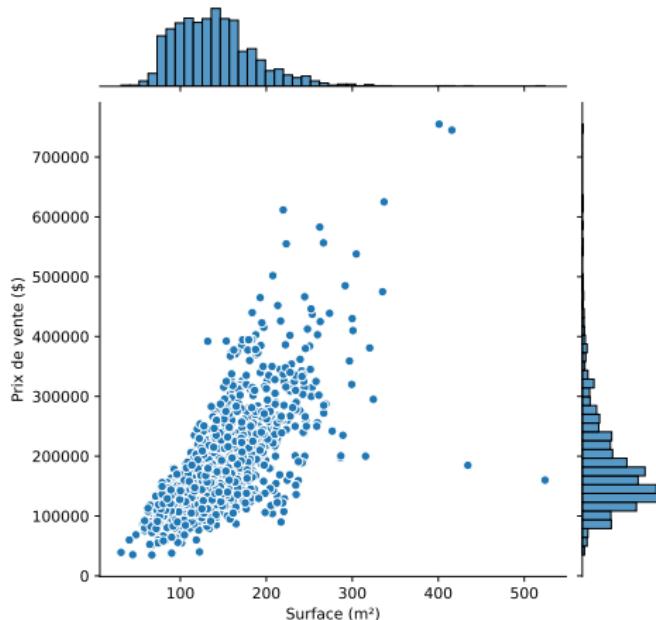


Diagramme quantile-quantile sur le dataset AMES, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Attention, pas seaborn mais statsmodel ou scipy.stats.

Outils – Nuage de points



Nuage de points sur les données AMES, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – Diagramme en bâtons

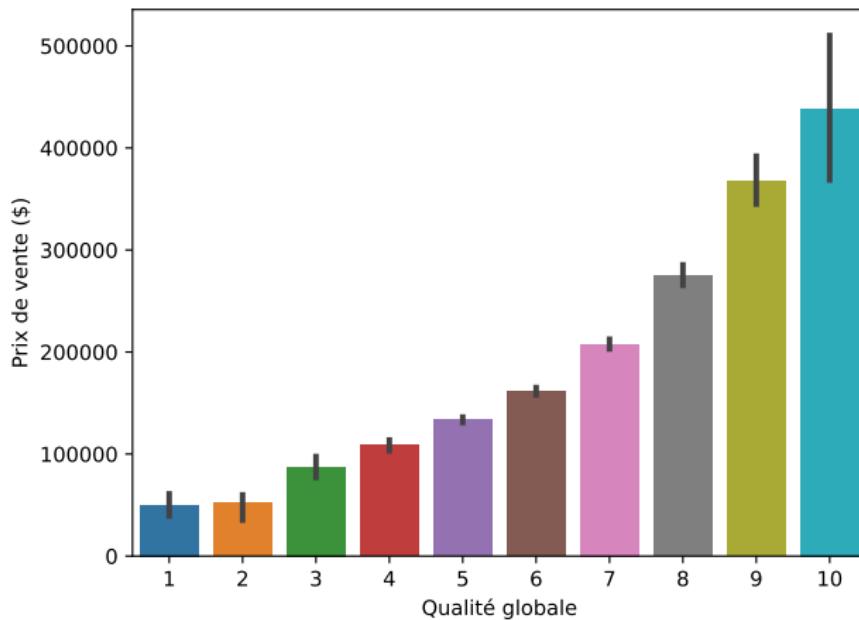


Diagramme en bâtons sur les données AMES, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – Diagramme en violons

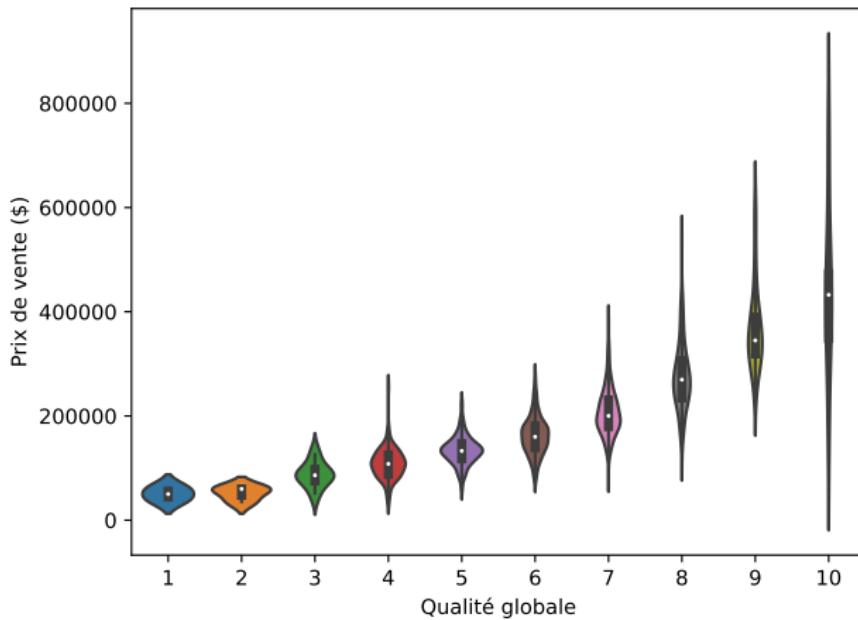
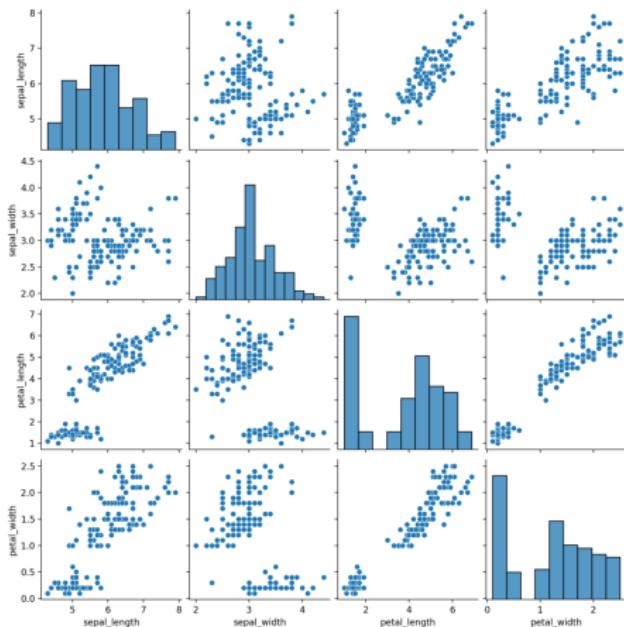


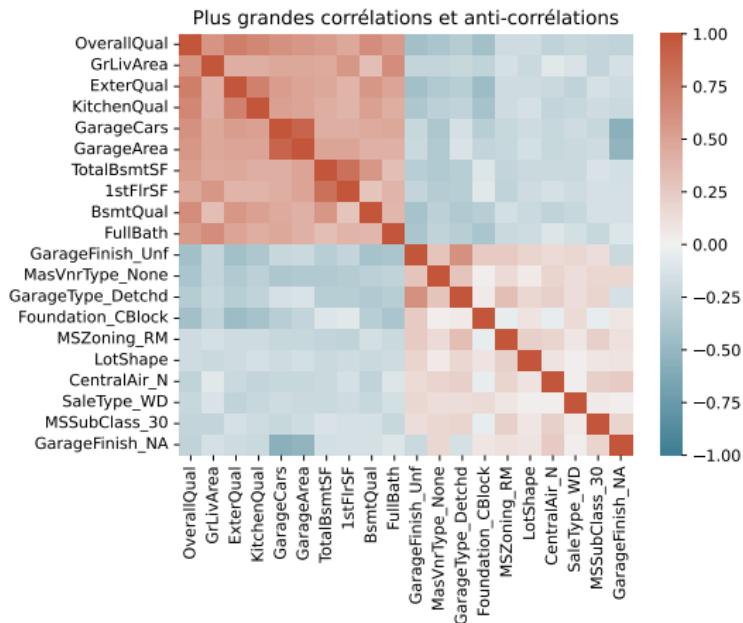
Diagramme en violon sur les données AMES, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – pair plot



Pair plot on Iris data, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Outils – Matrice de corrélation



Matrice des corrélations et anti-corrélations sur les données AMES, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Mode opératoire

Bonnes pratiques pour explorer un dataset :

- analyser la(es) variable(s) de sortie (diagramme de décompte ou histogramme)
- trouver les corrélations linéaires les plus fortes
- analyser les variables correspondantes
- regarder s'il y a des outliers évidents dans ces variables

Machine Learning

Évaluation

Buts

- Mesurer la qualité des prédictions du modèle
- Évaluer si un modèle peut remplir un objectif métier
- Se protéger des régressions après mise à jour

Évaluation de modèles de régression

erreur absolue moyenne ou erreur au carré moyenne.

Évaluation de modèles de classification

Les deux premières métriques sont les plus utilisées:

Précision

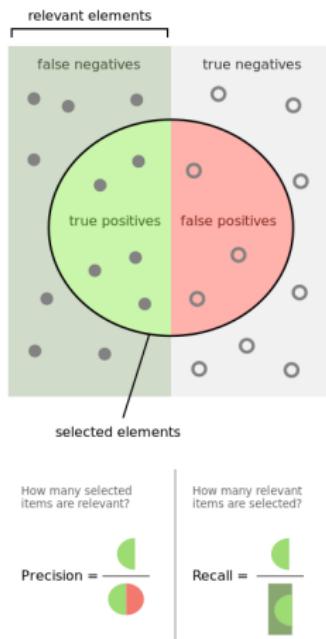
$$\frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}}$$

Rappel

$$\frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}}$$

F-mesure Moyenne harmonique entre précision et rappel (aussi appelée score F1)

Illustration de la précision et du rappel



Precision and recall, membre de Wikimedia Walber, CC-BY-SA-4.0.

Espace ROC

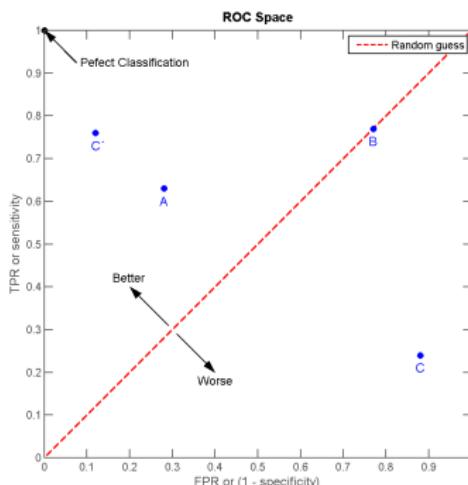
Mesure la performance du modèle à plusieurs seuils de décision.

Abscisse

1 - spécificité, taux de faux positifs, ou probabilité de *fausse alerte* ($\frac{FP}{VN+FP}$)

Ordonnée

Sensibilité, taux de vrais positifs ou rappel ($\frac{VP}{VP+FN}$)



Espace ROC, membre Indon de Wikimedia,
GFDL-1.2-or-later.

Matrice de confusion

Montrer clairement quelles sont les erreurs faites par le modèle



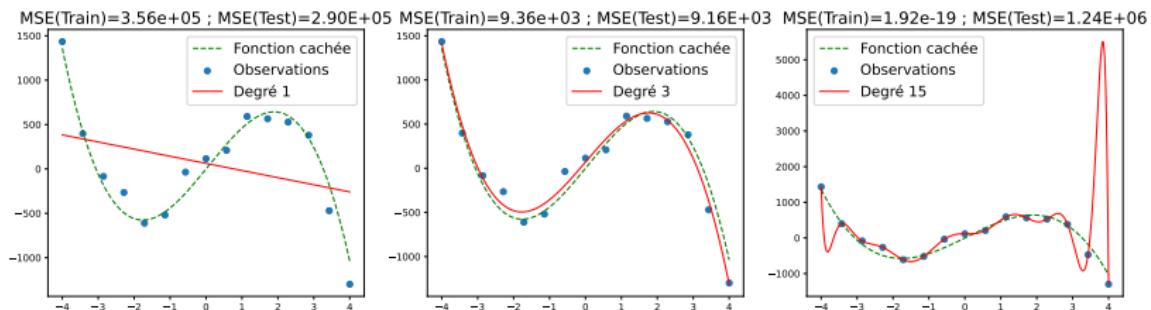
Matrice de confusion, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Machine Learning

Apprentissage

Qualité de l'apprentissage

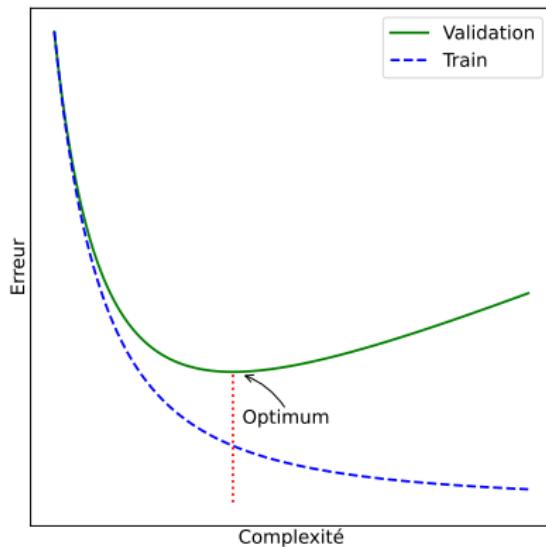
Entrainement supervisé d'un modèle — overfit



Mise en évidence du surapprentissage, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Problème : trop minimiser la perte n'est pas bon !

Qualité de l'apprentissage



Mise en évidence du surapprentissage sur les performances, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

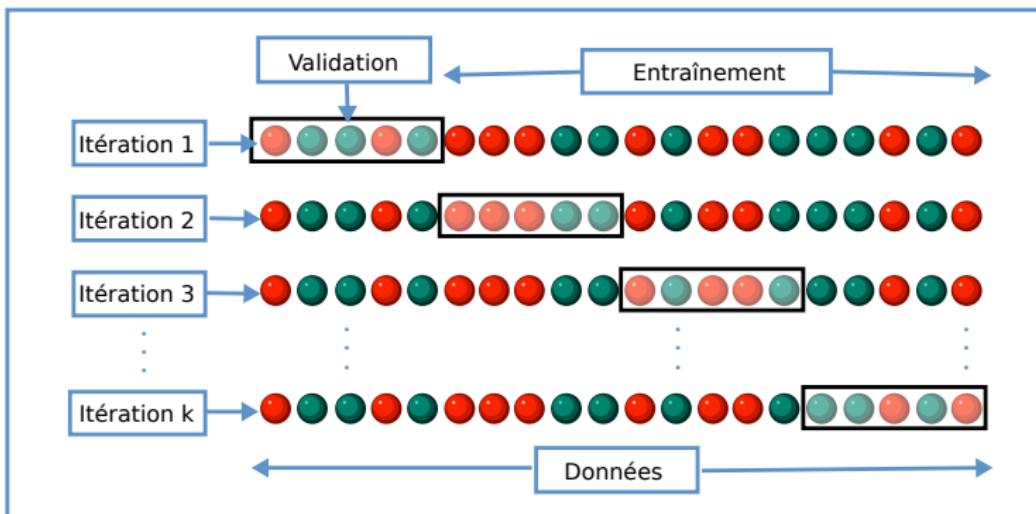
→ Minimiser la perte sur un ensemble de validation

Séparation des données

- ensemble d'entraînement
 - ensemble de validation pour mesurer la généralisation
 - ensemble de test (pour éviter le biais statistique)
- Split 60/20/20 habituel.

Cross-validation

Pour « perdre » moins de données et mieux tester la généralisation :

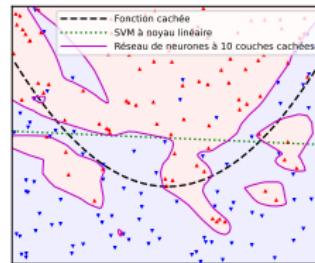


Validation croisée à 4 blocs, Gufosowa sur Wikimedia, CC-BY-SA-4.0.

Ici, 4-fold cross-validation.

Régularisation

Régularisation
≈
empêcher le surapprentissage



Sous-apprendre & sur-apprendre une fonction simple, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Techniques variées en fonction du modèle :

- Pénalisation de la norme des paramètres
- Bruitage
- Dropout
- ...

Optimisation des méta-paramètres

Méta-paramètres : paramètres **non appris** par le modèle.

Exemples

Forme Nombre de couches ? De quelles tailles ? ...

Optimisation SGD, AdaBoost, Adam, ...

Régularisation Pénalisation de la Norme des paramètres dans la loss, bruitage, dropout, ...

Optimisation par recherche aléatoire ou processus gaussien.

Avez-vous des questions ?

Ressources complémentaires

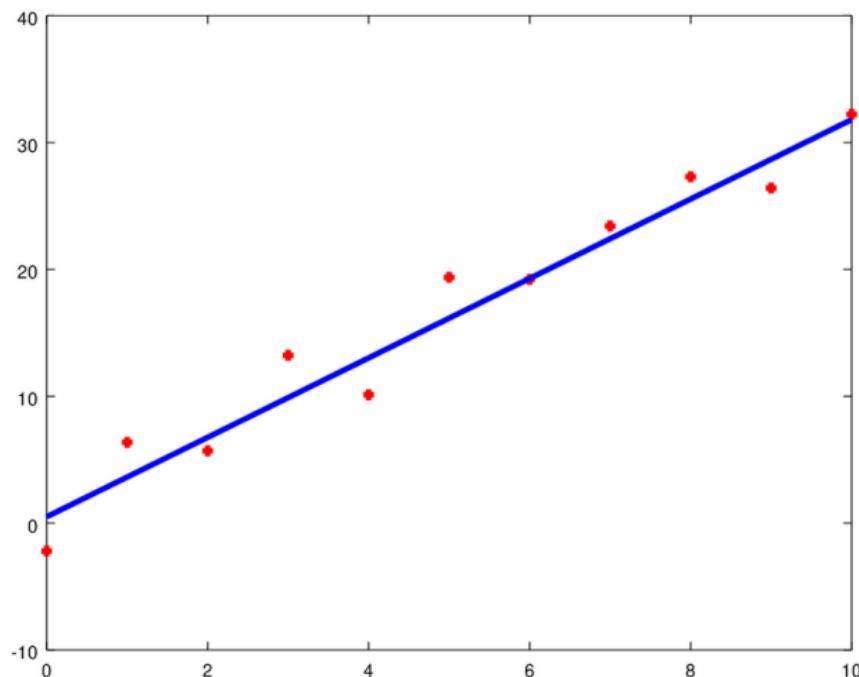
- [Les prétraitements](#)
- [La normalisation en particulier](#)

Réseaux de neurones

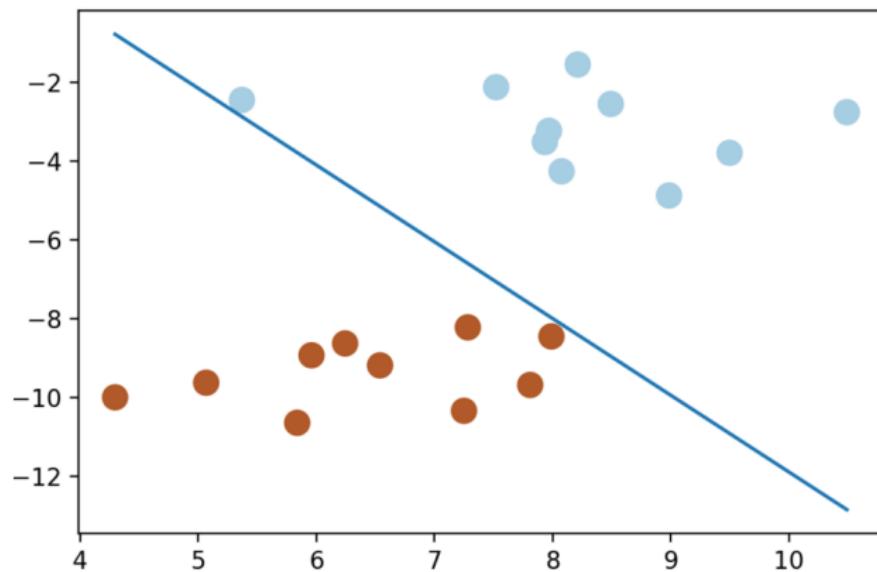
Réseaux de neurones

Introduction

Lien avec la régression linéaire

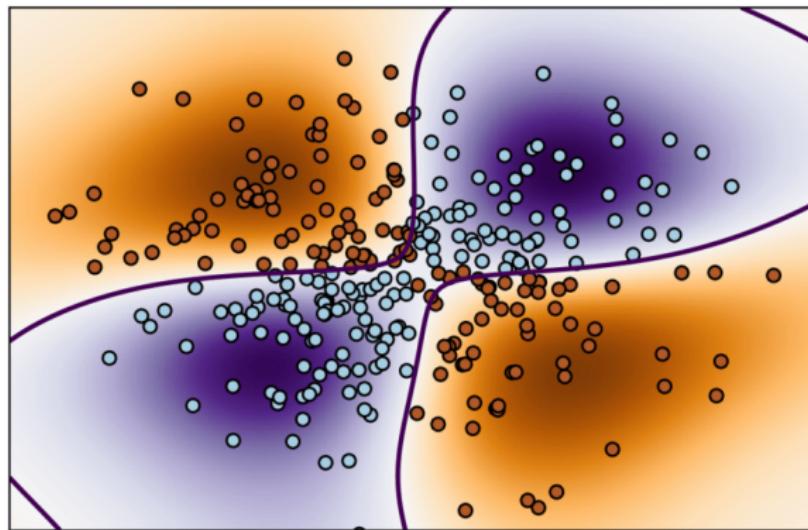


Lien avec la régression linéaire



Séparateur linéaire, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Lien avec la régression linéaire



Classification de données obtenues grâce à XOR, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Lien (tenu) avec la biologie

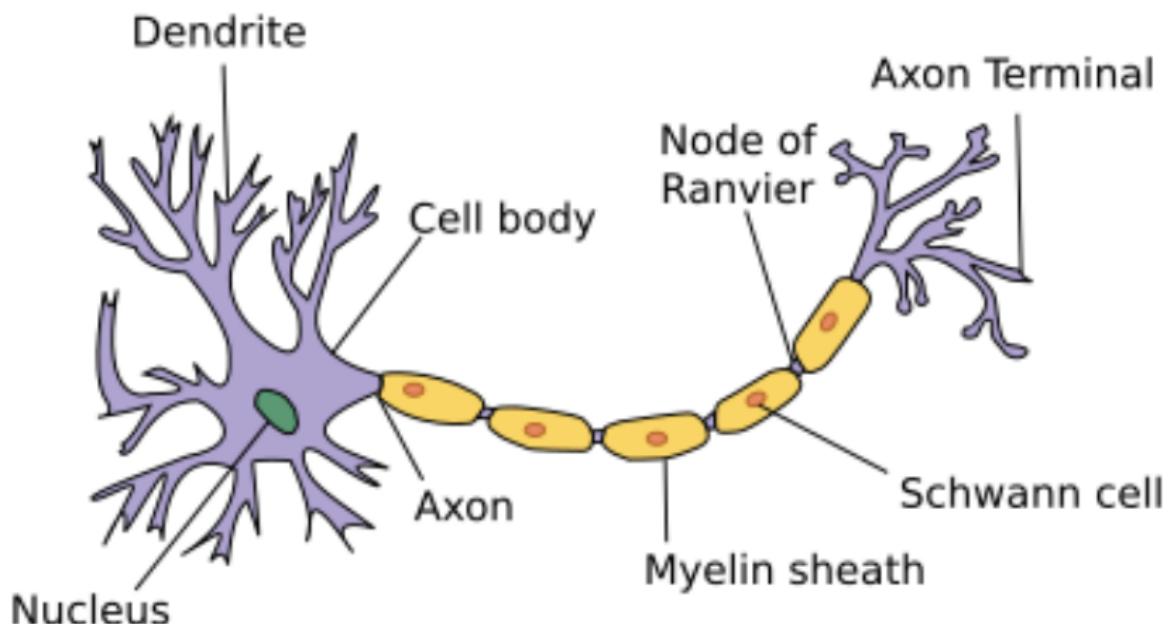
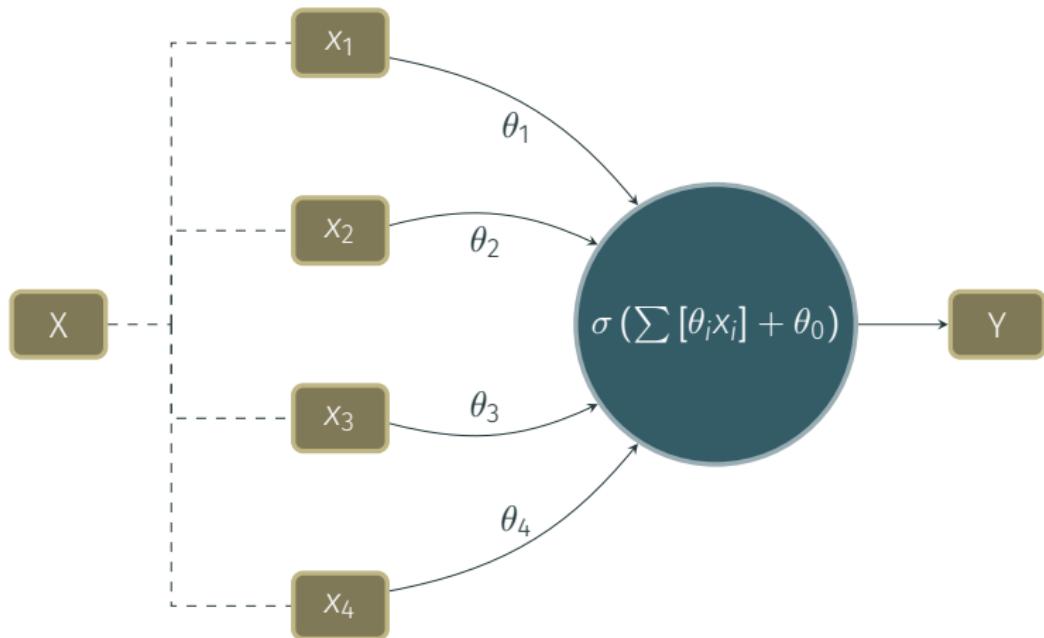


Diagram of a Neuron, membre de Wikimedia Dhp1080, CC BY-SA 3.0.

Modélisation d'un neurone



où σ est une fonction émulant une activation à seuil

Modélisation d'un réseau de neurones

Agencement de beaucoup de neurones :

En parallèle Calculent des résultats indépendamment dans la même couche

En série Prennent en entrée les résultats des neurones de la couche précédente

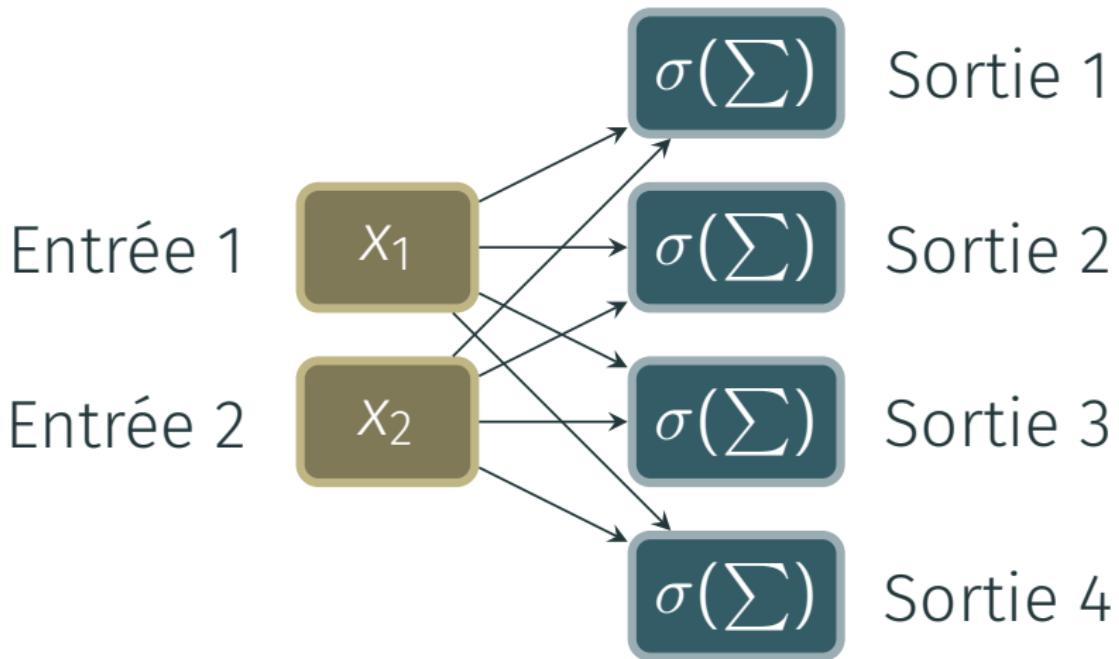
Deux types de neurones

On distingue deux types de neurones :

Neurones cachés Neurones des couches intermédiaires. Améliorent l'expressivité du modèle

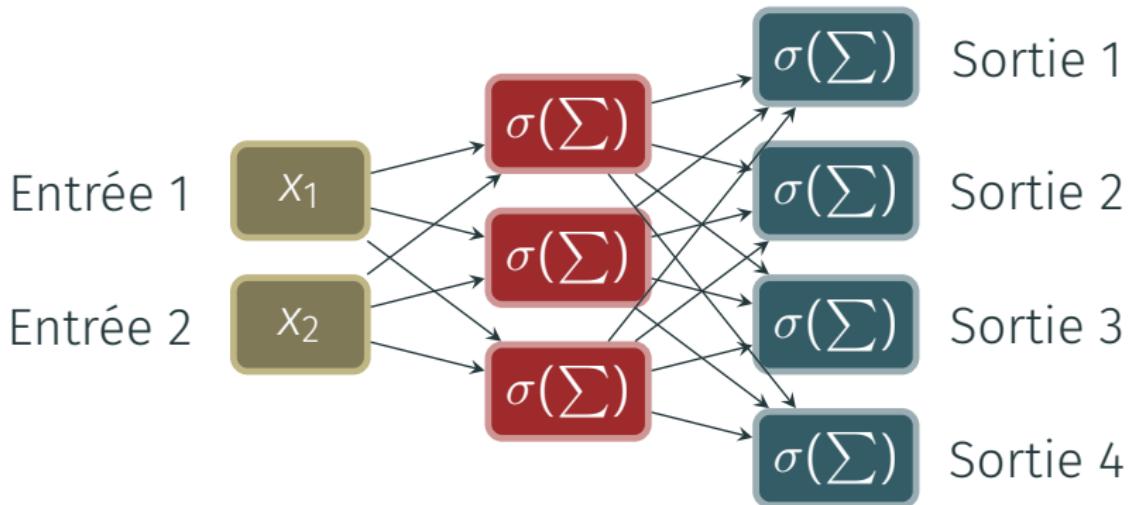
Neurones de sortie Neurones de la couche finale. Contraints par le type de sortie attendu

Réseau sans couche cachée



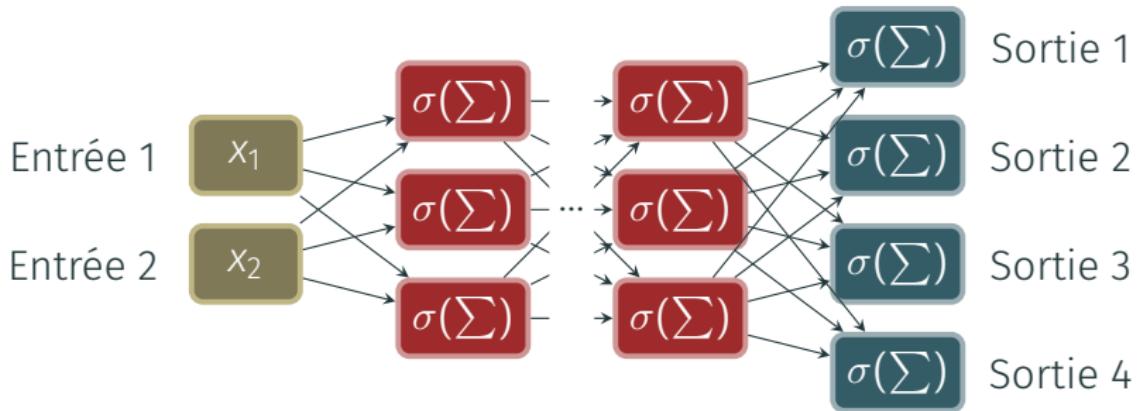
Une couche de réseau de neurone, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Réseau avec une couche cachée



Réseau de neurone avec une couche cachée, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Réseau profond



Réseau de neurone profond, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Un potentiel infini

Kurt Hornik, 1991 : Théorème d'approximation universelle

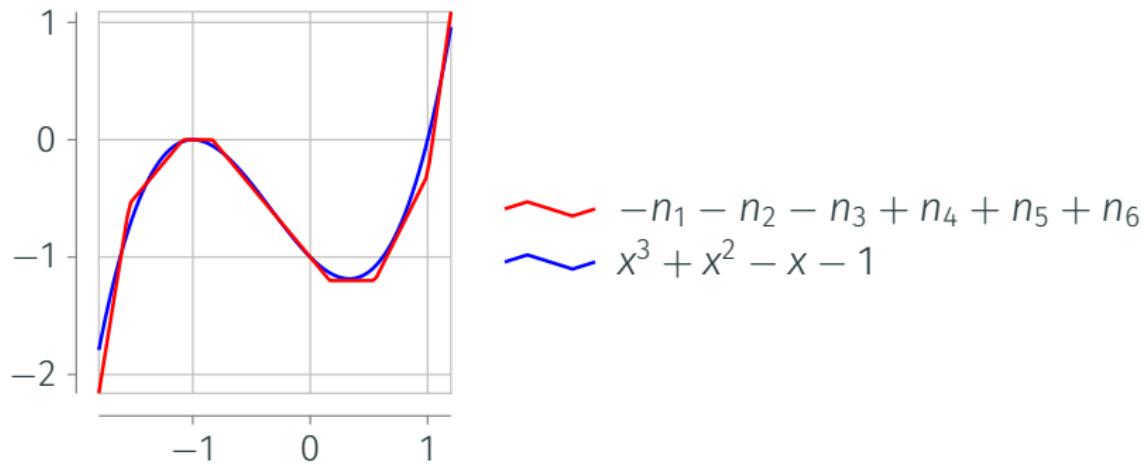


Illustration du théorème d'approximation universel, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Illustration d'une approximation

Modélisation matricielle – Échantillon

Représentable sous forme de vecteur à d colonnes correspondant à d caractéristiques :

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}$$

Voir même de matrice dans le cas d'un batch (groupe d'échantillons) :

$$\mathbf{X} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \end{bmatrix}$$

Modélisation matricielle – Poids

Représentables sous forme de matrice de poids :

$$\theta = \begin{bmatrix} \theta_{0,1} & \theta_{0,2} & \dots & \theta_{0,n} \\ \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{d,1} & \theta_{d,2} & \dots & \theta_{d,n} \end{bmatrix}$$

Modélisation matricielle complète

$$\begin{aligned}
 O &= \sigma\left(\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} X \theta\right) \\
 &= \sigma\left(\begin{bmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,d} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,d} \end{bmatrix} \begin{bmatrix} \theta_{0,1} & \theta_{0,2} & \dots & \theta_{0,n} \\ \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{d,1} & \theta_{d,2} & \dots & \theta_{d,n} \end{bmatrix}\right) \\
 &= \begin{bmatrix} O_{1,1} & O_{1,2} & \dots & O_{1,n} \\ O_{2,1} & O_{2,2} & \dots & O_{2,n} \end{bmatrix}
 \end{aligned}$$

X Données en entrée de dimension d

θ Paramètres à trouver des n neurones de notre modèle

σ Fonction d'activation

O Sortie du réseau

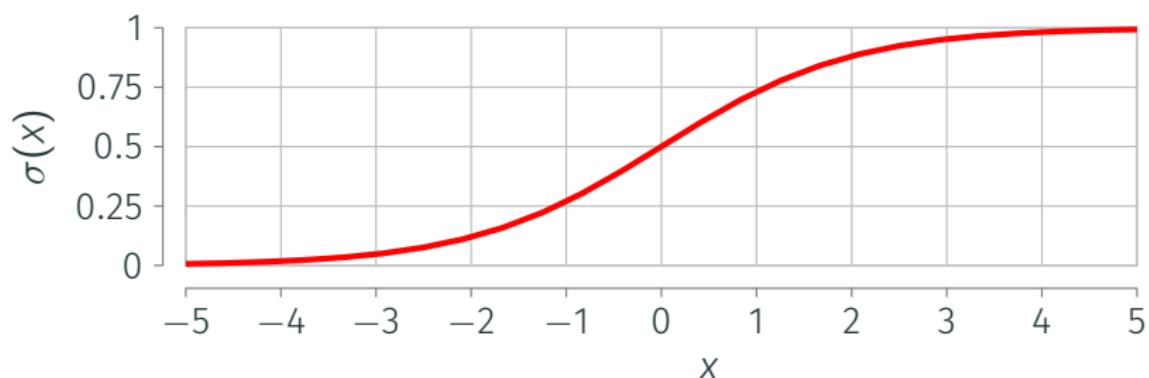
Fonctions d'activation – Critères de choix

- Propriétés mathématiques (conservation du gradient)
- Propriétés d'apprentissage (éviter la création de poids morts)
- Rapidité de calcul
- Intervalle de sortie pour la dernière couche

Fonctions d'activation – Les plus classiques

- Sigmoïde
- Tanh
- Softmax
- ReLU
- ...

Fonctions d'activation – Sigmoïde



fonction d'activation Sigmoïde, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

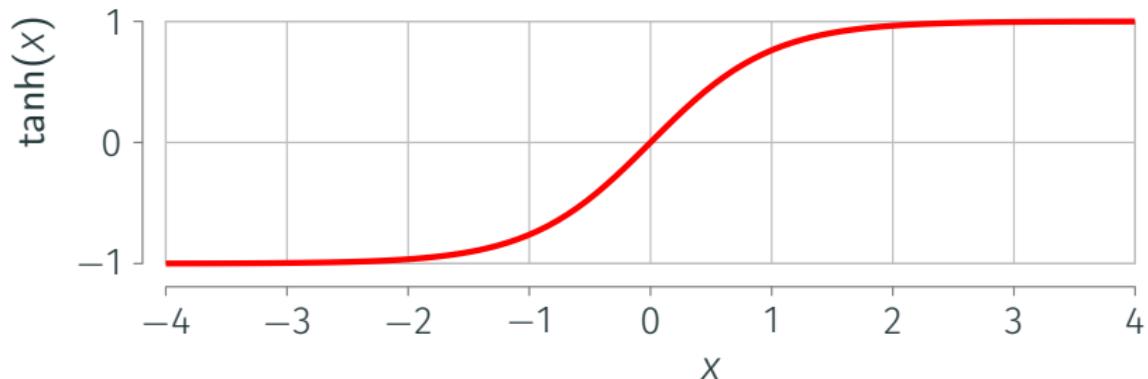
Définition

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

Dérivée

$$\phi'(x) = \phi(x)(1 - \phi(x))$$

Fonctions d'activation – Tangente hyperbolique



fonction d'activation Tangente Hyperbolique (\tanh), F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

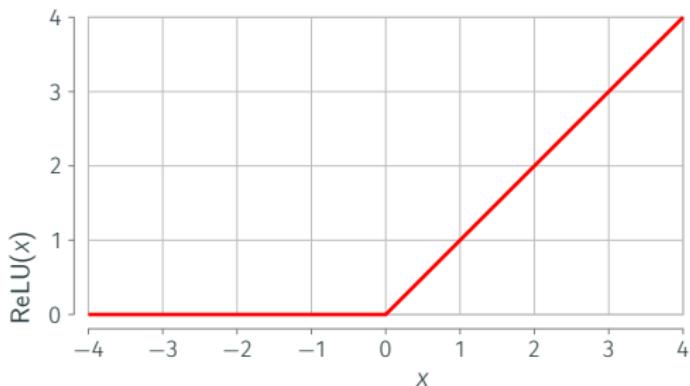
Définition

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Dérivée

$$\tanh'(x) = 1 - \tanh^2(x)$$

Fonctions d'activation – ReLU



fonction d'activation Rectified Linear Unit (RELU), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Définition

$$\text{ReLU}(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

Dérivée

$$\text{ReLU}'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Fonctions d'activation – Approximation d'une fonction

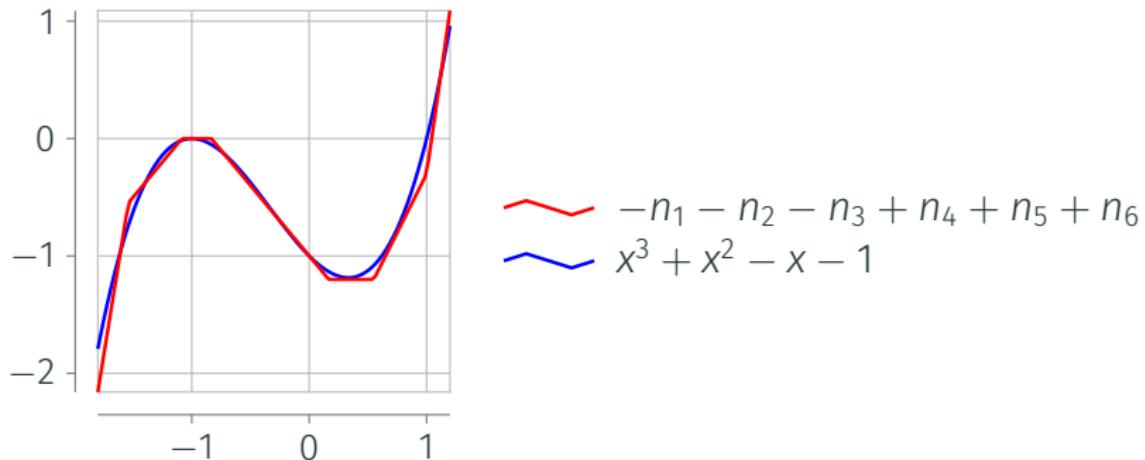


Illustration du théorème d'approximation universel, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

$$n_1 = \text{ReLU}(-5x - 7.7) \quad n_2 = \text{ReLU}(-1.2x - 1.3) \quad n_3 = \text{ReLU}(1.2x + 1)$$

$$n_4 = \text{ReLU}(1.2x - 0.2) \quad n_5 = \text{ReLU}(2x - 1.1) \quad n_6 = \text{ReLU}(5x - 5)$$

Fonctions d'activation – Softmax

Définition

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{k=1}^n \exp(x_k)}$$

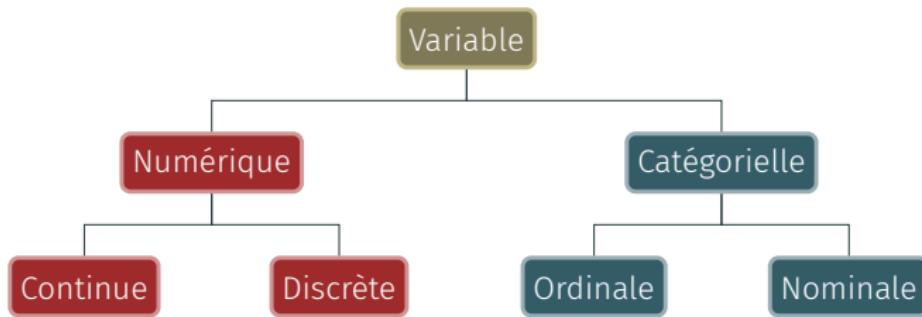
Propriété

$$\sum_{i=1}^n \text{softmax}(x)_i = 1$$

Gradient

$$\frac{\partial \text{softmax}(x)_i}{\partial x_j} = \begin{cases} \text{softmax}(x)_i(1 - \text{softmax}(x)_j) & i = j \\ -\text{softmax}(x)_i \text{softmax}(x)_j & i \neq j \end{cases}$$

Classification



Les différents types de variables, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Classification



A few samples from the MNIST test dataset, Josef Steppan, CC-BY-SA-4.0.

Classification



[001.ak47](#)



[002.american-flag](#)



[003.backpack](#)



[004.baseball-bat](#)



[005.baseball-glove](#)



[006.basketball-hoop](#)



[007.bat](#)



[008.bathtub](#)



[009.bear](#)



[010.beer-mug](#)



[011.billiards](#)



[012.binoculars](#)

Caltech-256 Object Category Dataset, G. Griffin et al., Caltech.

Classification

≈ Distance entre la sortie et la cible ?

Sortie :

0.00	0.10	0.40	0.00	0.00	0.20	0.10	0.00	0.20	0.00
------	------	------	------	------	------	------	------	------	------

Cible :

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
------	------	------	------	------	------	------	------	------	------

Classification

Critère de l'erreur absolue (MAE) = 0.12

Sortie :

0.00	0.10	0.40	0.00	0.00	0.20	0.10	0.00	0.20	0.00
------	------	------	------	------	------	------	------	------	------

Cible :

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
------	------	------	------	------	------	------	------	------	------

Classification

Critère de l'erreur absolue (MAE) = 0.12

Sortie :

0.12	0.12	0.88	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
------	------	------	------	------	------	------	------	------	------	------

Cible :

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
------	------	------	------	------	------	------	------	------	------	------

Classification

Entropie croisée entre la sortie et la cible :

$$H(y, \hat{y}) = - \sum_i y_i \log \hat{y}_i$$

⇒ Minimale quand sortie = cible

Démonstration

Réseau de neurones

Réseaux de neurones

Descente de gradient

Principe

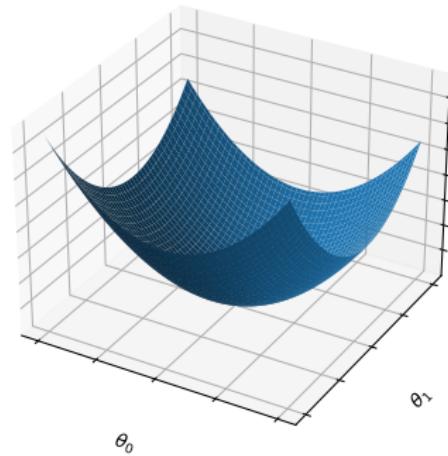
Calcul du gradient de l'erreur par rapport aux paramètres:

$$\frac{\partial E}{\partial \theta_i}$$

Mise à jour :

$$\theta_i \leftarrow \theta_i - \gamma \frac{\partial E}{\partial \theta_i}$$

où $0 < \gamma < 1$ (pas d'apprentissage)



Erreur quadratique moyenne

Surface du critère des moindres carrés pour la régression linéaire, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Algorithme

1. Initialisation aléatoire du modèle
2. Pour n itérations :
 - Pour b batchs :
 - **Forward** : Passe avant du **batch** dans le modèle
 - Calcul de l'erreur par rapport aux sorties attendues
 - **Backward** : Rétropropagation du gradient de l'erreur en fonction des paramètres dans le modèle (mise à jour du modèle)
 - Vérification des critères d'arrêt

Exemple de dérivation – Régression linéaire

$$E = \frac{1}{2n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

$$E = \frac{1}{2n} \sum_{i=1}^n ((\theta_0 + \theta_1 x_i) - y_i)^2$$

...

$$\frac{\partial E}{\partial \theta_0} = \frac{1}{n} \sum_{i=1}^n (\theta_0 + \theta_1 x_i - y_i)$$

$$\frac{\partial E}{\partial \theta_1} = \frac{1}{n} \sum_{i=1}^n (\theta_0 + \theta_1 x_i - y_i) x_i$$

$$\hat{y}_i = \theta_0 + \theta_1 x_i$$

$$U^2' = 2U' \times U$$

Mise à jour

$$\theta_0 \leftarrow \theta_0 - \gamma \frac{\partial E}{\partial \theta_0}$$

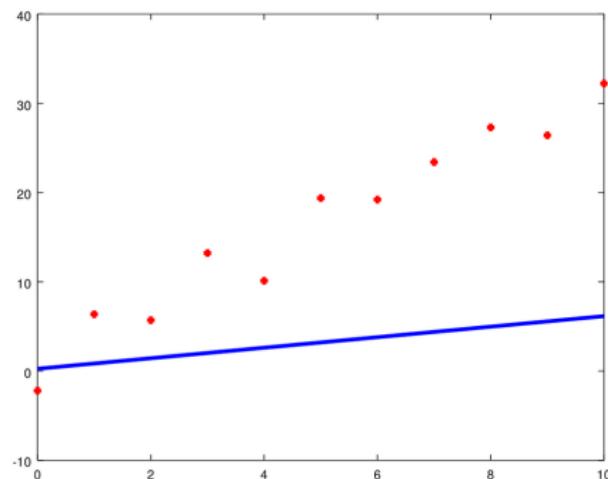
$$\theta_1 \leftarrow \theta_1 - \gamma \frac{\partial E}{\partial \theta_1}$$

où $0 < \gamma < 1$ (pas d'apprentissage)

Exemple

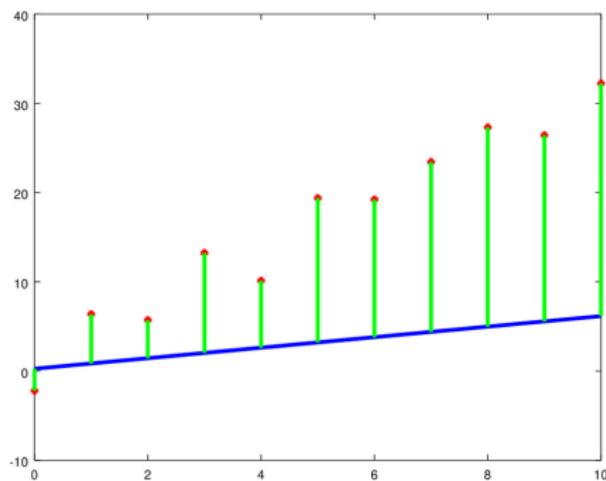
Initialisation au hasard ($\gamma = 0.01$)

- $\theta_0 = 0.25$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 0.58$ (vrai $\theta_1 = 3.0$)



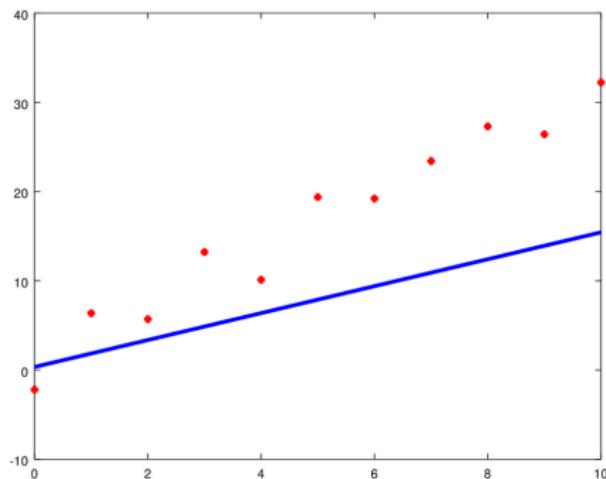
Exemple

- $\theta_0 = 0.25$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 0.58$ (vrai $\theta_1 = 3.0$)



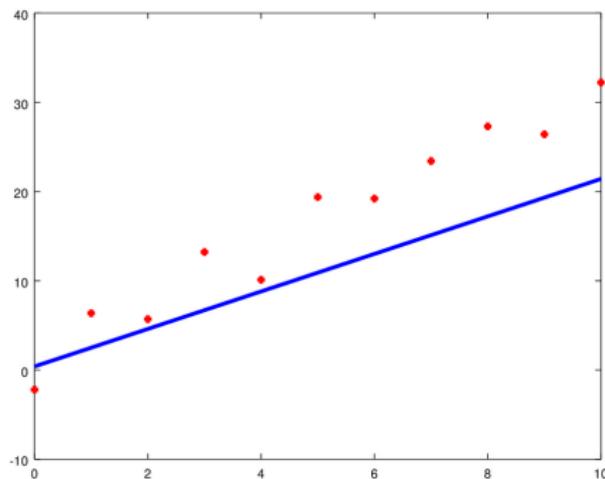
Exemple

- $\theta_0 = 0.35$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 1.50$ (vrai $\theta_1 = 3.0$)



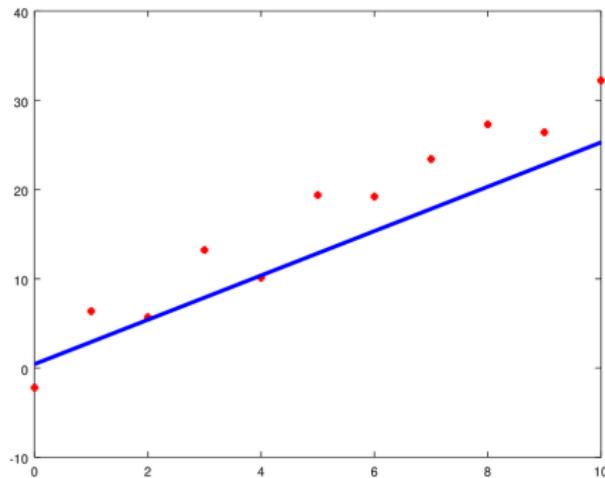
Exemple

- $\theta_0 = 0.40$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 2.10$ (vrai $\theta_1 = 3.0$)



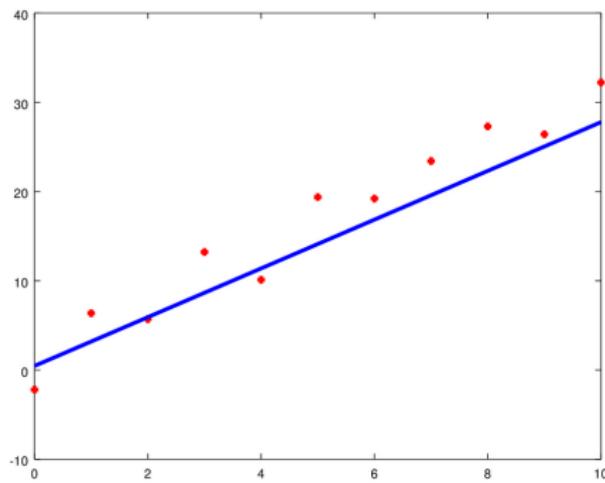
Exemple

- $\theta_0 = 0.43$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 2.48$ (vrai $\theta_1 = 3.0$)



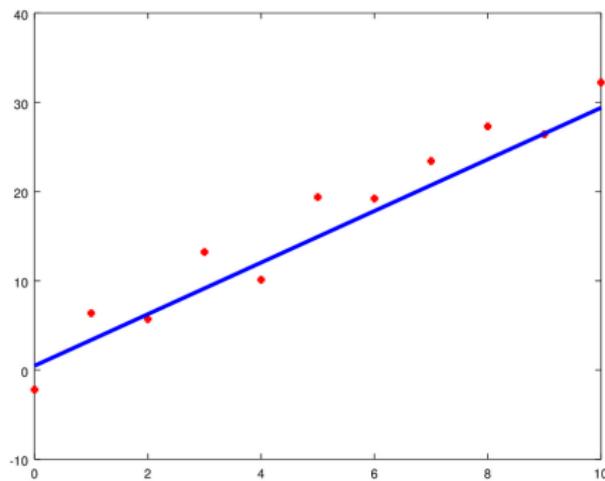
Exemple

- $\theta_0 = 0.46$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 2.73$ (vrai $\theta_1 = 3.0$)



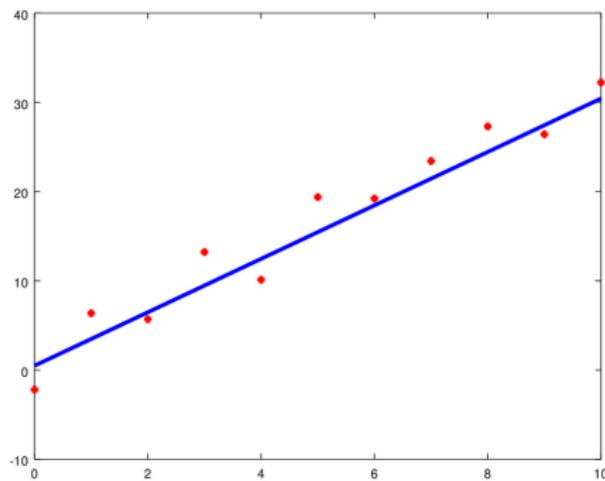
Exemple

- $\theta_0 = 0.47$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 2.89$ (vrai $\theta_1 = 3.0$)



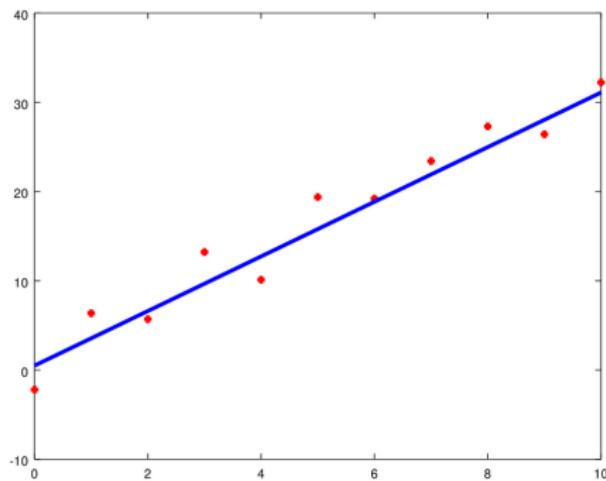
Exemple

- $\theta_0 = 0.48$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 2.99$ (vrai $\theta_1 = 3.0$)



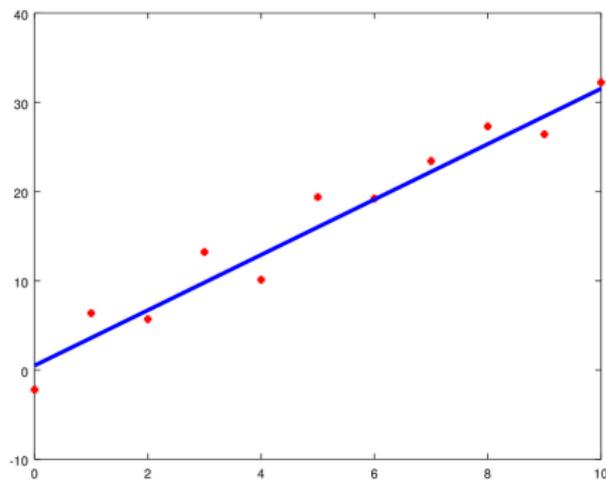
Exemple

- $\theta_0 = 0.49$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 3.06$ (vrai $\theta_1 = 3.0$)



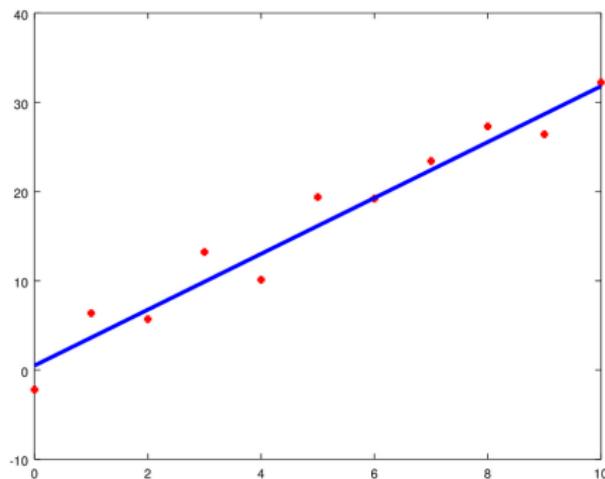
Exemple

- $\theta_0 = 0.49$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 3.10$ (vrai $\theta_1 = 3.0$)



Exemple

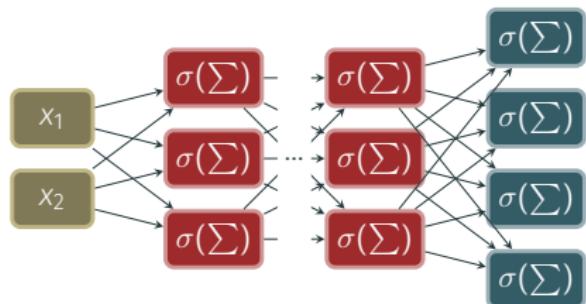
- $\theta_0 = 0.50$ (vrai $\theta_0 = 0.5$)
- $\theta_1 = 3.13$ (vrai $\theta_1 = 3.0$)



Réseaux de neurones

Optimisation des hyper-paramètres

Qu'est-ce qu'un hyper-paramètre ?



- Learning rate
- Taille de batch
- Nombre de couches
- Taille des couches

Réseau de neurone profond, F.-M. Giraud, R.

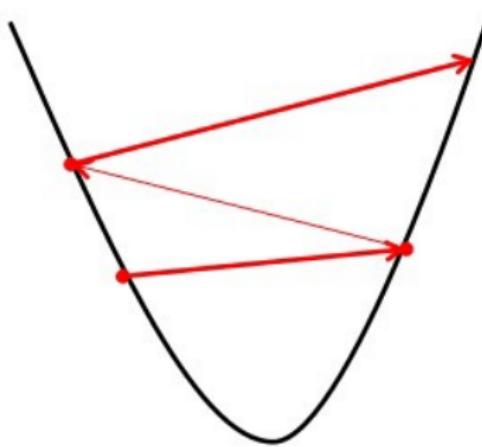
Rincé & H. Mougard, CC-BY-SA-4.0.

Approche pour optimiser les hyper-paramètres

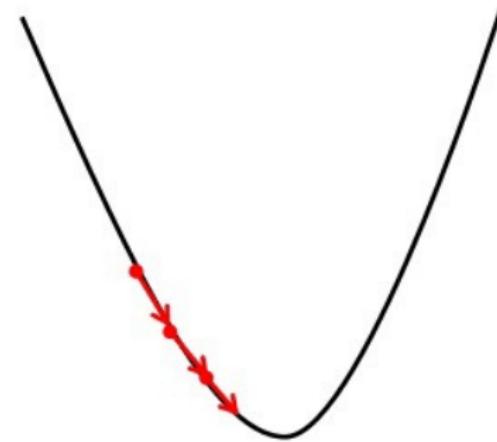
- Commencer avec une bonne baseline et une rapide recherche « à la main »
- Quand toute la pipeline est fonctionnelle, implémenter la recherche en utilisant une librairie spécialisée

Learning rate

trop grand



trop petit



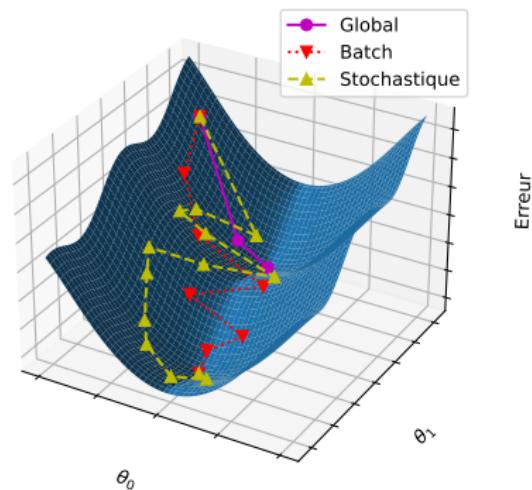
Learning rate

Utilisation d'une échelle logarithmique (dans un premier temps) :

```
for lr in [0.1, 0.01, 0.001, 0.0001, 0.00001]:  
    train = tf.train.GradientDescentOptimizer(lr).minimize(loss)  
    ...
```

Taille du batch

Usuellement, des puissances de 2.



Effet de la taille du batch, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Taille des couches

Usuellement, des puissances de 2.

Même intuition que pour le nombre de couches.

Nombre de couches

```
while perf > previous_perf:  
    previous_perf = perf  
    model = add_layer(model)  
    model.fit(data_train)  
    perf = model.eval(data_eval)
```

Pour aller plus loin

Entraîner des modèles coûte cher. Une bonne navigation de l'espace des hyper-paramètres est importante.

⇒ Faire appel à une librairie dédiée comme Keras Tuner, Hyperopt, Nevergrad, Optuna ou Ray pour utiliser des processus gaussiens ou d'autres méthodes état de l'art.

Réseaux de neurones

Optimisation de réseaux profonds

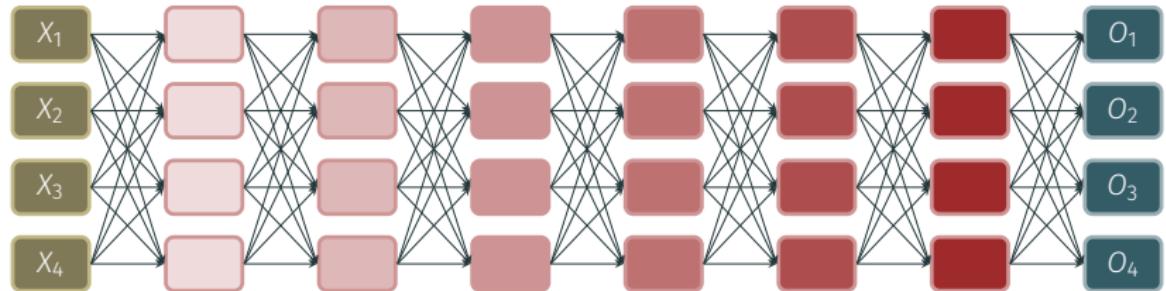
Rétropropagation du gradient

Pour calculer le gradient dans des réseaux profonds, on utilise la rétropropagation :

- Algorithme de programmation dynamique
- Commence par calculer le gradient de la dernière couche
- Calcule le gradient d'une couche en se basant sur le gradient de la couche suivante
- Utilise la règle de dérivation des fonctions composées pour le faire

⇒ Évite de recalculer des sous-expressions.

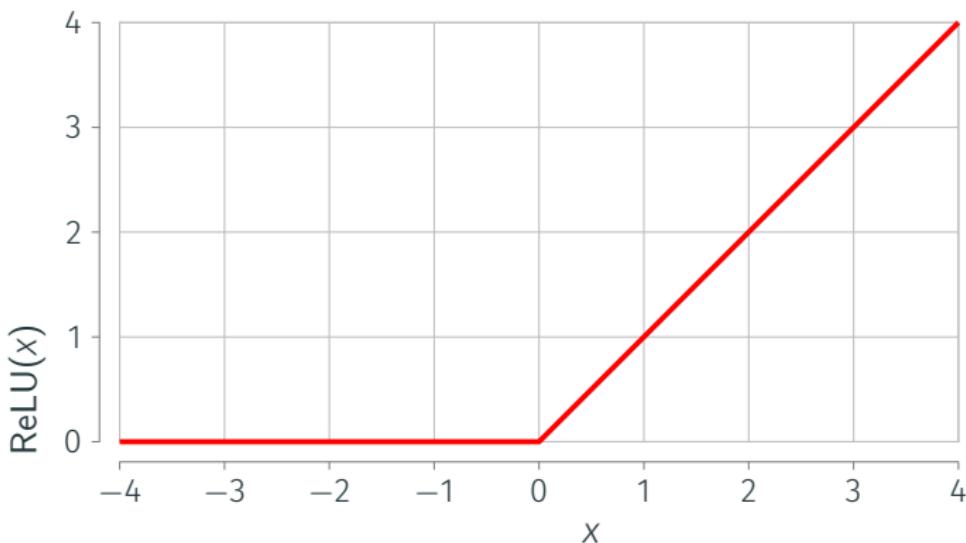
Disparition du gradient (ou son explosion)



Le problème du gradient qui disparaît (vanishing gradient), F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Disparition des gradients

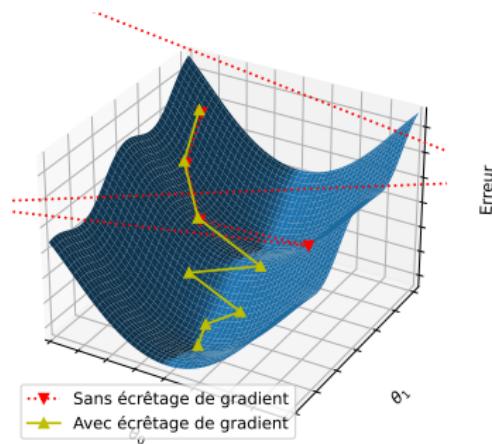
Vanishing Gradient \Rightarrow utilisation de ReLU plutôt que les sigmoïdes



fonction d'activation Rectified Linear Unit (RELU), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Explosion des gradients

Exploding Gradient \Rightarrow Gradient clipping



Effet de l'écrêtage de gradient , F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Dans le cadre des réseaux récurrents \Rightarrow initialisation avec des matrices orthogonales

Algorithme d'optimisation

Optimisateur plus rapide que SGD

- Adaptative Gradient Algorithm (AdaGrad)
- Root Mean Square Propagation (RMSProp)
- Adaptative Moment Estimation (Adam) ⇐
- ... AdaBound (2019) ?

Root Mean Square Propagation (RMSProp)

- Un pas d'apprentissage par paramètre
- Empêche les grandes mises à jour fréquentes :
 - Calcule une moyenne glissante exponentielle :

$$v_t(\theta) \leftarrow \gamma v_{t-1}(\theta) + (1 - \gamma)(\nabla_\theta L)^2$$

- Met à jour chaque gradient en le divisant par sa moyenne glissante :

$$\theta \leftarrow \theta - \frac{\eta}{\sqrt{v_t(\theta)}} \nabla_\theta L$$

Adaptive Gradient Algorithm (AdaGrad)

Pénalise les mises à jour fréquentes comme RMSProp mais calcule le facteur différemment :

$$\theta \leftarrow \theta - \frac{\eta}{\sqrt{\sum_{\tau=1}^t (\nabla_{\theta,\tau} L)^2}} \nabla_{\theta} L$$

Adam

Similaire à RMSProp dans l'algorithme. Garde deux valeurs au lieu d'une en mémoire pour chaque paramètre : la moyenne m et la variance v de son gradient.

$$m_\theta^{(t+1)} \leftarrow \beta_1 m_\theta^{(t)} + (1 - \beta_1) \nabla_\theta L^{(t)}$$

$$v_\theta^{(t+1)} \leftarrow \beta_2 v_\theta^{(t)} + (1 - \beta_2) (\nabla_\theta L^{(t)})^2$$

$$\hat{m}_\theta = \frac{m_\theta^{(t+1)}}{1 - \beta_1}$$

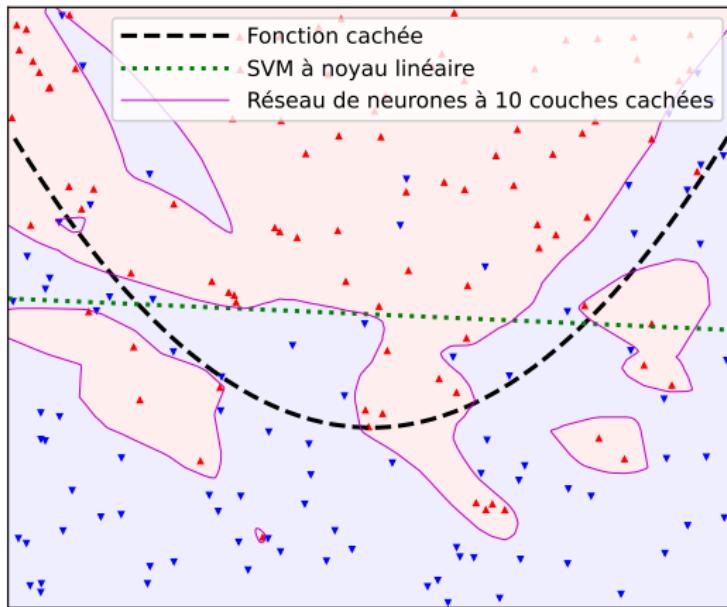
$$\hat{v}_\theta = \frac{v_\theta^{(t+1)}}{1 - \beta_2}$$

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta \frac{\hat{m}_\theta}{\sqrt{\hat{v}_\theta} + \epsilon}$$

Réseaux de neurones

Régularisation

Sur-apprentissage (et sous-apprentissage)



Sous-apprendre & sur-apprendre une fonction simple, F.-M. Giraud, R. Rincé & H. Mougard,
CC-BY-SA-4.0.

Par la pénalisation de l'utilisation des paramètres

Coût supplémentaire pour l'utilisation des paramètres dans la fonction de perte :

L1 Produit beaucoup de poids à 0, s'appelle aussi Lasso :

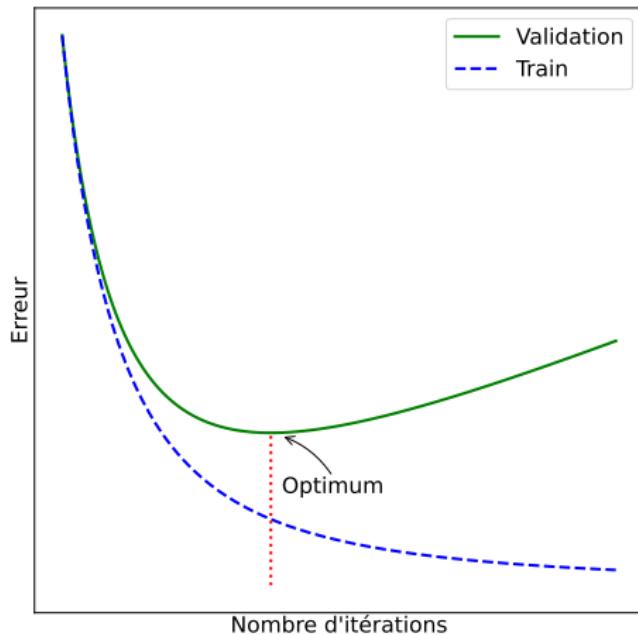
$$E = E + \lambda \|\boldsymbol{\theta}\|_1$$

L2 Pénalise les très gros poids, s'appelle aussi Ridge :

$$E = E + \lambda \|\boldsymbol{\theta}\|_2^2$$

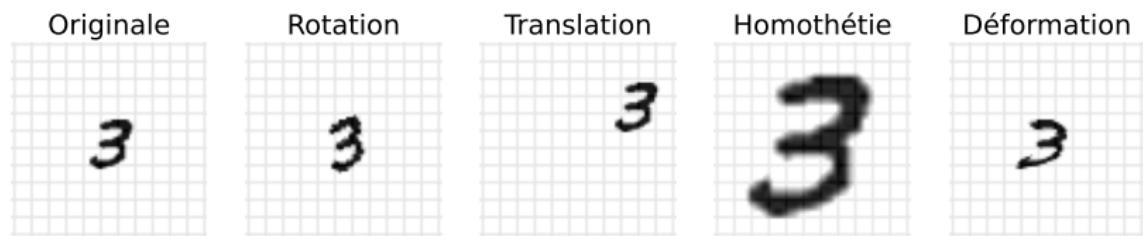
où λ est un hyperparamètre

Par early stopping



Effet du nombre d'itérations sur le surapprentissage des réseaux de neurones, F.-M. Giraud, R. Rincé & H. Mougaard, CC-BY-SA-4.0.

Par augmentation/bruitage des données



Exemples d'augmentation de données, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Par dropout

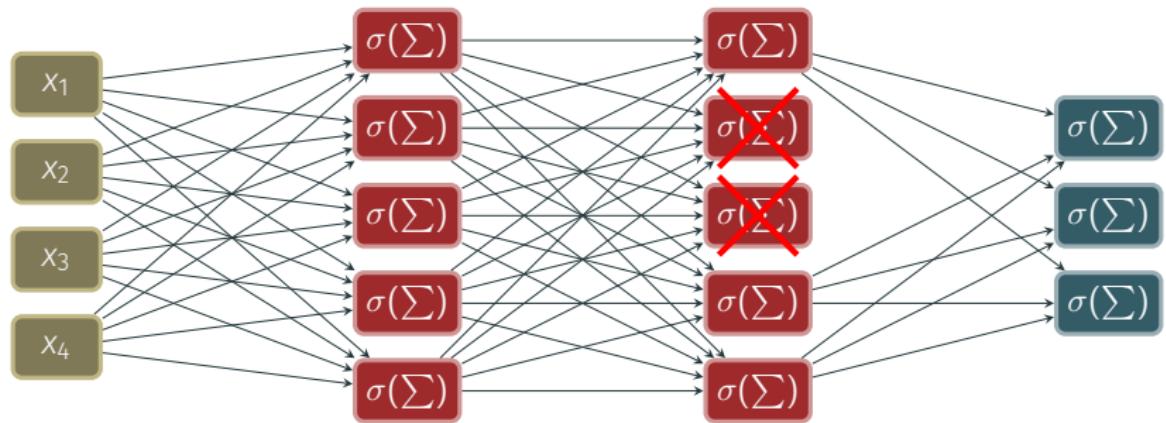
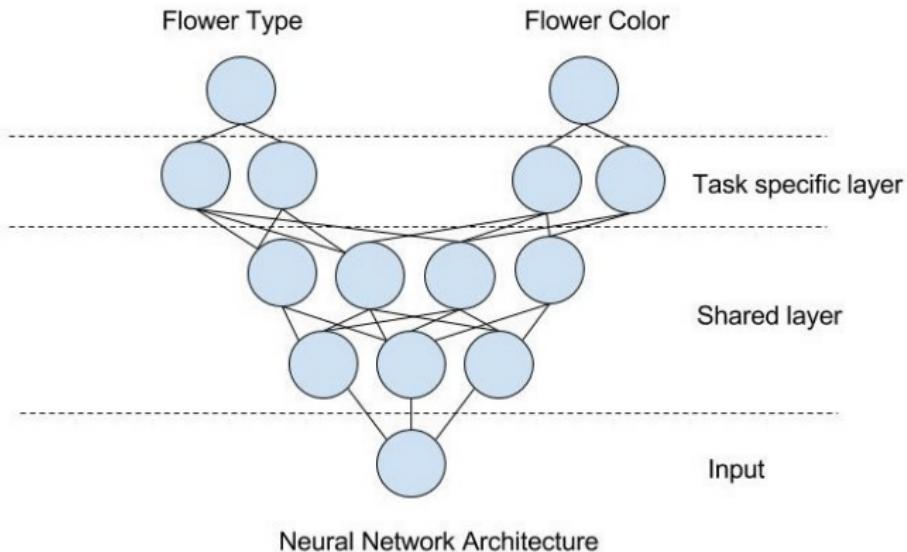


Schéma de dropout, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

En entraînant sur plusieurs tâches



En opposant des réseaux de neurones

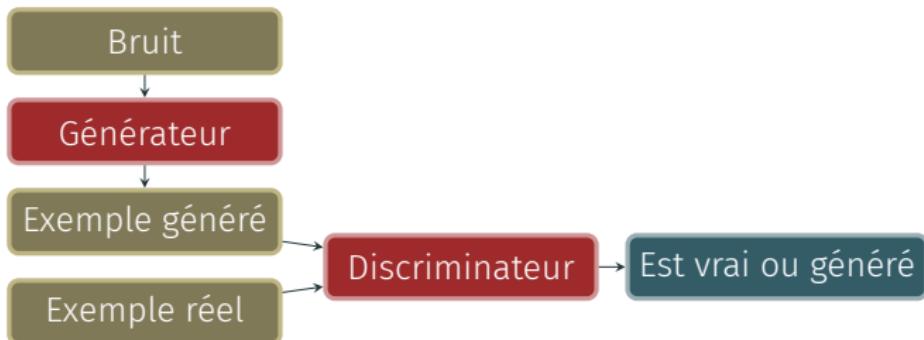


Schéma de réseaux générateurs adverses (GAN), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Démonstration

Réseau de neurones

Avez-vous des questions ?

Réseaux de neurones

TensorFlow et Keras

Réseaux de neurones

TensorFlow et Keras

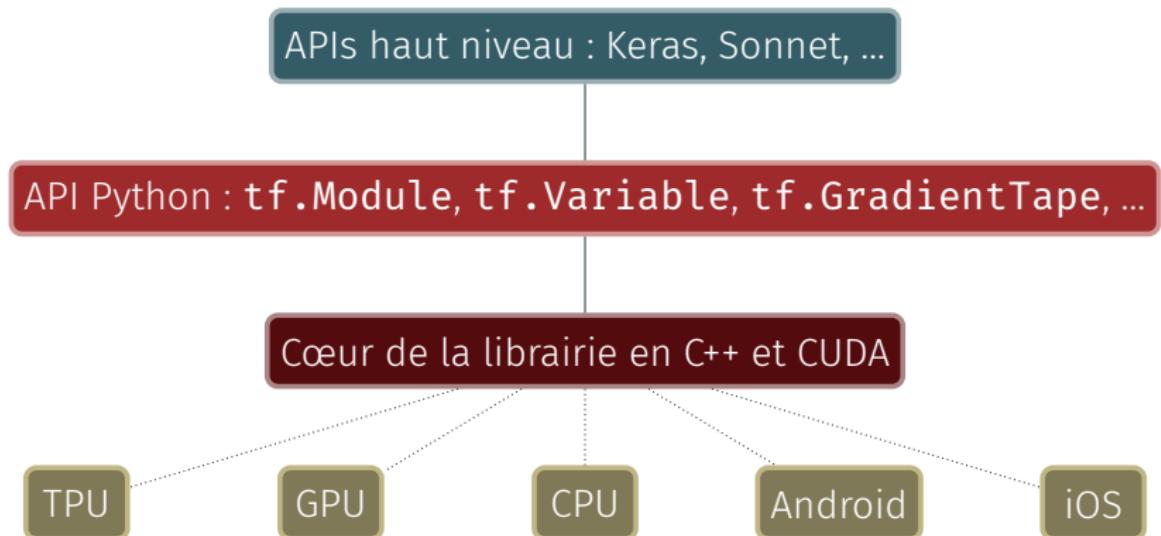
TensorFlow 2

Introduction



Logo Tensorflow, Tensorflow authors, Domaine public.

Structure de l'API



L'API Tensorflow, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Deux styles

Mode graphe : on définit un graphe de calcul qu'on exécute ensuite.

Mode *eager* : on définit des fonctions python qui opèrent directement sur des valeurs.

TensorFlow 1 vs TensorFlow 2

Principales différences

TF1

- Graphes de calcul explicites par défaut
- API de haut niveau Keras indépendante

TF2

- Graphes de calcul implicites par défaut
- API de haut niveau Keras intégrée à TF

Tenseurs – Objets au cœur de TensorFlow

Tableaux multidimensionnels, utilisés pour :

- Poids des réseaux
- Données

`tf.Tensor` \approx `numpy.ndarray` dans l'univers TensorFlow.

Tenseurs – Création

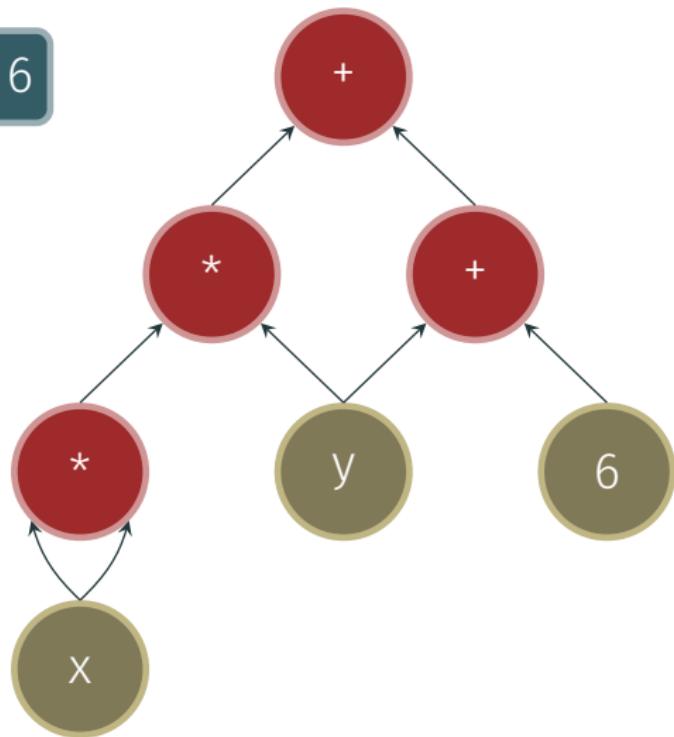
- Directement à partir de tableaux NumPy ou de listes
- Échantillonnés aléatoirement
- Appel à une fonction tensorflow avec un tableau NumPy
- Pipeline `tf.data`

tf.Variable

Surcouche mutable de `tf.Tensor` pour les poids des modèles.

Graph de calcul

$$f(x, y) = x^2y + y + 6$$



Graph de calcul

Conversion du mode *eager* au mode graphe par `tf.function`.

Permet :

- L'optimisation du graphe des opérations
- Le déploiement vers Android, iOS, TPU, GPU, ...
- De faire tourner le réseau sans interpréteur Python !

Différenciation automatique

Calcul automatique des gradients pour faciliter la rétropropagation des gradients pendant la descente de gradient.

`tf.GradientTape` enregistre les opérations effectuées sur une variable : permet l'autodiff.

Organisation du code

`tf.Module` regroupe du code d'une même couche ou unité logique.

Un modèle = une combinaison de `tf.Modules`.

API principale :

- Accès aux variables du modèle par `model.variables`
- Accès aux variables entraînables du modèle par `model.trainable_variables`
- Sauvegarde des poids par `tf.train.Checkpoint`
- Chargement des poids par `tf.train.Checkpoint.restore`
- Sauvegarde du modèle complet par `tf.saved_model.save`
- Chargement du modèle complet par `tf.saved_model.load`

Avez-vous des questions ?

Travaux pratiques

Instructions

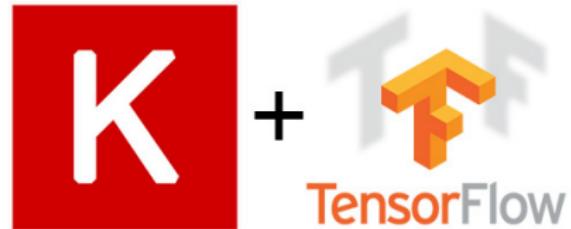
Réseaux de neurones

TensorFlow et Keras

Keras

Introduction

Keras est une API de haut niveau qui permet de prototyper des réseaux de neurones de toutes sortes.



Logos Keras et TensorFlow, François Cholet, Domaine public.

Avantages – Facile d'utilisation

- Interface simple
- Accès facilité aux métriques d'évaluation

Avantages – Modulaire

- Les réseaux se « branchent » facilement les uns avec les autres
- Tous les réseaux se configurent facilement

Avantages – État de l'art

- Les modèles et optimiseurs pertinents sont rapidement ajoutés à Keras
- Reproduction facile de résultats récents

Avantages – Extensible

Facile de développer de nouvelles :

- couches de réseau (Layers)
- fonctions de perte (Loss)
- métriques d'évaluation

Réseau de neurones en Keras

```
model = tf.keras.models.Sequential()
# Il est impératif de spécifier input_shape pour la première couche
model.add(tf.keras.layers.Dense(64,
                                activation="sigmoid",
                                input_shape=(32,)))
# Ajouter une autre couche
model.add(tf.keras.layers.Dense(64, activation="relu"))
# Une dernière couche de classification avec softmax
model.add(tf.keras.layers.Dense(10, activation="softmax"))
```

Options de configuration des Layers

```
# Utiliser une fonction d'activation
tf.keras.layers.Dense(64, activation="sigmoid")
# Ou
tf.keras.layers.Dense(64, activation=tf.keras.activations.sigmoid)

# Régularisation L1 des poids de la matrice
tf.keras.layers.Dense(64,
                      kernel_regularizer=tf.keras.regularizers.l1(0.01))

# Régularisation L2 des biais
tf.keras.layers.Dense(64,
                      bias_regularizer=tf.keras.regularizers.l2(0.01))

# Initialisation des poids avec une matrice orthogonale
tf.keras.layers.Dense(64,
                      kernel_initializer="orthogonal")

# Initialisation des biais avec une constante :
tf.keras.layers.Dense(
    64, bias_initializer=tf.keras.initializers.Constant(2.0))
```

Optimiseur, fonction de perte et métrique d'évaluation

```
# Compilation du modèle avec entropie croisée et accuracy
model.compile(optimizer="adam",
                loss="sparse_categorical_crossentropy",
                metrics=[ "accuracy"])

# Compilation équivalente mais permet la customisation
model.compile(optimizer=tf.keras.optimizers.Adam(),
                loss=tf.keras.losses.SparseCategoricalCrossentropy(),
                metrics=[tf.keras.metrics.Accuracy()])

# Exemple de compilation pour un problème de régression
model.compile(optimizer=tf.keras.optimizers.Adam(0.01),
                loss="mse",          # Mean Squared Error
                metrics=[ 'mae' ])  # Mean Absolute Error
```

Affichage du modèle

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	2112
dense_1 (Dense)	(None, 64)	4160
dense_2 (Dense)	(None, 10)	650
<hr/>		
Total params: 6,922		
Trainable params: 6,922		
Non-trainable params: 0		
<hr/>		

Apprentissage

```
data = numpy.random.random((1000, 32))
labels = numpy.random.random((1000, 10))

model.fit(data, labels, epochs=10, batch_size=32)
```

Apprentissage avec validation

```
data = numpy.random.random((1000, 32))
labels = numpy.random.random((1000, 10))

val_data = numpy.random.random((100, 32))
val_labels = numpy.random.random((100, 10))

model.fit(data, labels, epochs=10, batch_size=32,
           validation_data=(val_data, val_labels))
```

Apprentissage avec validation par split

```
data = numpy.random.random((1000, 32))
labels = numpy.random.random((1000, 10))

model.fit(data, labels, epochs=10, batch_size=32, validation_split=0.3)
```

Utilisation de `tf.data.Dataset`

```
# Avec un Dataset TensorFlow
dataset = tf.data.Dataset.from_tensor_slices((data, labels))
dataset = dataset.batch(32)

model.fit(dataset, epochs=10)
```

Évaluation

```
data = numpy.random.random((1000, 32))
labels = numpy.random.random((1000, 10))

model.evaluate(data, labels, batch_size=32)

# Ou avec un Dataset TensorFlow
dataset = tf.data.Dataset.from_tensor_slices((data, labels))
dataset = dataset.batch(32)

model.evaluate(dataset)
```

```
1000/1 [=====] - 0s 70us/sample - loss: 199785.4507
                                         - categorical_accuracy: 0.0990
32/32 [=====] - 0s 2ms/step - loss: 200361.9849
                                         - categorical_accuracy: 0.0990
```

Prédiction

```
result = model.predict(data, batch_size=32)
print(result.shape)
```

```
(1000, 10)
```

Sauvegarde de la configuration d'un modèle

```
# Sérialisation d'un modèle vers JSON
json_string = model.to_json()

# Chargement d'un modèle depuis JSON
fresh_model = tf.keras.models.model_from_json(json_string)
```

Sauvegarde des poids d'un modèle

```
# Sauvegarde des poids dans un fichier HDF5  
model.save_weights("my_model.h5", save_format="h5")  
  
# Restauration des poids du modèle  
model.load_weights("my_model.h5")
```

Sauvegarde d'un modèle complet

```
# Sauvegarde d'un modèle complet
model.save('my_model.h5')

# Chargement du modèle, avec optimiseur, perte et métriques
model = tf.keras.models.load_model('my_model.h5')
```

Callbacks

```
callbacks = [
    # Interrompt l'apprentissage si `val_loss` ne s'améliore plus depuis
    # 2 epochs
    tf.keras.callbacks.EarlyStopping(patience=2,
                                      monitor="val_loss"),
    # Sauvegarde le meilleur model
    tf.keras.callbacks.ModelCheckpoint(filepath='models/bestmodel.hdf5',
                                        verbose=1,
                                        save_best_only=True)
]
model.fit(data, labels, batch_size=32, epochs=5, callbacks=callbacks,
           validation_split=0.2)
```

Création d'un Layer

```
class MyLayer(tf.keras.layers.Layer):
    # Sauvegarde des paramètres
    def __init__(self, output_dim, **kwargs):
        super().__init__(**kwargs)
        self.output_dim = output_dim

    # Création des poids
    def build(self, input_shape):
        self.kernel = self.add_weight(
            name="kernel",
            shape=(input_shape[1], self.output_dim),
            initializer="uniform",
            trainable=True,
        )

    # Calcul de la couche
    def call(self, inputs):
        return inputs @ self.kernel
```

Utilisation de notre nouveau Layer

```
model = tf.keras.models.Sequential()
model.add(MyLayer(10))
model.add(tf.keras.layers.Activation("softmax"))

model.compile(optimizer=tf.keras.optimizers.RMSprop(0.001),
              loss="sparse_categorical_crossentropy",
              metrics=[ "accuracy"])

model.fit(data, labels, batch_size=32, epochs=5)
```

Avez-vous des questions ?

Travaux pratiques

[Instructions](#)

Réseaux de neurones

Réseaux de neurones à convolutions

Réseaux de neurones

Réseaux de neurones à convolutions

Données en vision par ordinateur

MNIST

810k caractères manuscrits



A few samples from the MNIST test dataset, Josef Steppan, CC-BY-SA-4.0.

Caltech 256

300k objets/animaux/personnes



[001.ak47](#)



[002.american-flag](#)



[003.backpack](#)



[004.baseball-bat](#)



[005.baseball-glove](#)



[006.basketball-hoop](#)



[007.bat](#)



[008.bathtub](#)



[009.bear](#)



[010.beer-mug](#)



[011.billiards](#)



[012.binoculars](#)

Caltech-256 Object Category Dataset, G. Griffin et al., Caltech.

Street View House Numbers

600k numéros annotés



The Street View House Numbers (SVHN) Dataset, Y. Netzer et al., Stanford.

ImageNet

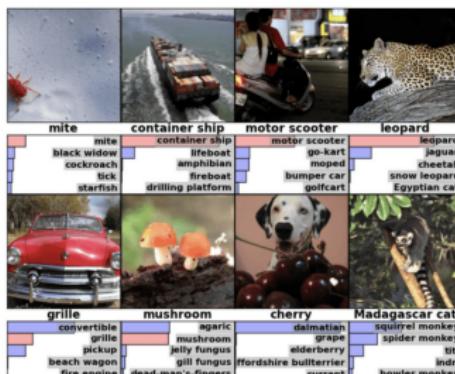
Images annotés par synset de wordnet :

- 20k synsets non-vides (objectif 100k)
- 14M images (objectif 100M)
- dont 1M sont annotées

ImageNet Challenge

IMAGENET

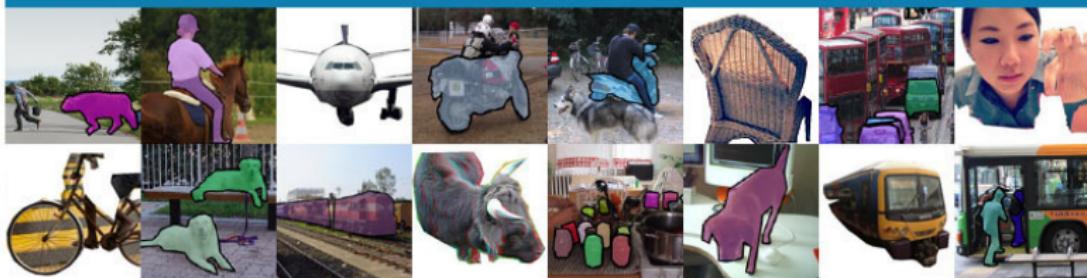
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



MS-COCO

330k images avec annotation+segmentation

Dataset examples



Description du dataset MS-COCO, Tsung-Yi Lin, et al., Microsoft COCO: Common Objects in Context.

VisualQA

265k images, Question answering

Who is wearing glasses?

man



woman



Where is the child sitting?

fridge



arms



Is the umbrella upside down?

yes



no



How many children are in the bed?

2



1



Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering, Y. Goyal et al., arXiv.

Open Images

9M images annotées avec 6000 catégories



The Cowboy and his Iguana, membre eflon de Flickr, CC-BY-2.0.

MS-Celeb-1M

10M photos de 100k célébrités



MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition, Y. Guo et al., arXiv.

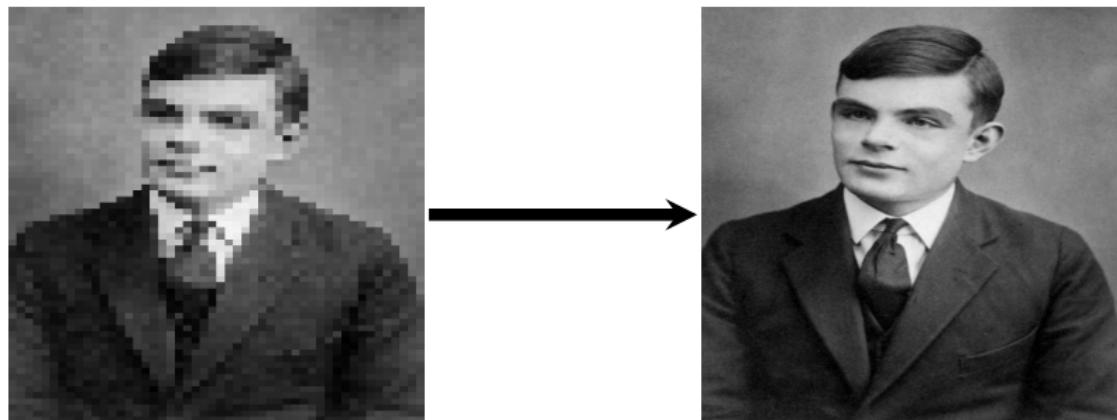
Classification



Forêt
Mer
Rue
Montagne
Glacier
Bâtiment

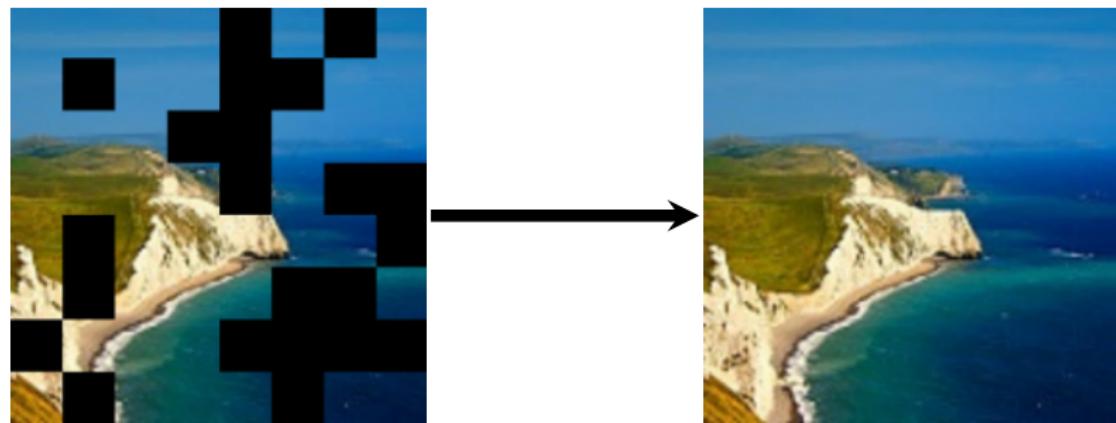
Classification d'image, F.-M. Giraud, R. Rincé & H. Mougaard, CC-BY-SA-4.0.

Super résolution



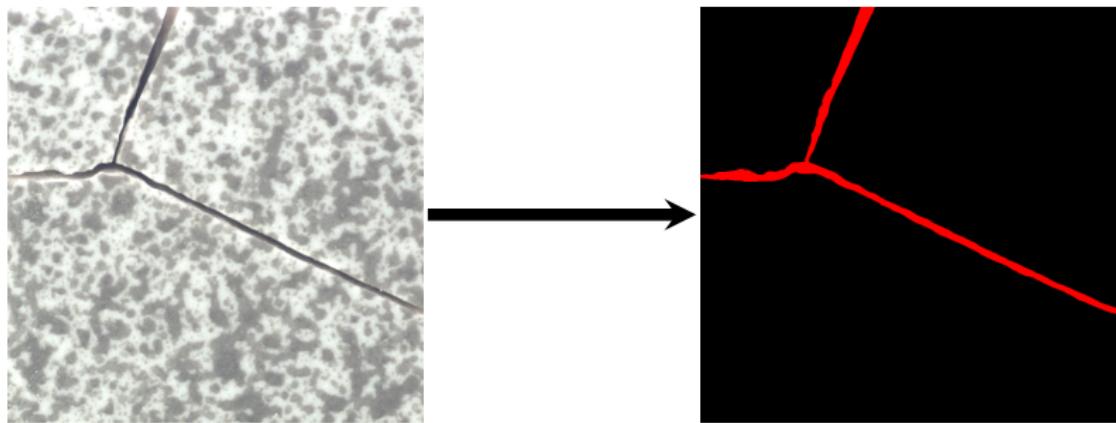
Exemple de « super résolution », F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Inpainting



Exemple d'« inpainting », F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Détection d'anomalie



Exemple de détection d'anomalie, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Réseaux de neurones

Réseaux de neurones à convolutions

Modèle

Problème des MLPs pour les images

Supposons que nous voulons détecter un objet dans une image. Un bon modèle devrait :

- Trouver l'objet où qu'il soit dans l'image
- Utiliser les pixels locaux autour de l'objet pour prendre sa décision

Les MLPs standards ne satisfont pas ces critères.

⇒ Les réseaux convolutifs, si.

Convolution

Bloc clef : l'opération de corrélation croisée (appelée par erreur convolution).

Elle incorpore la **localité** et l'**invariance à la translation**.

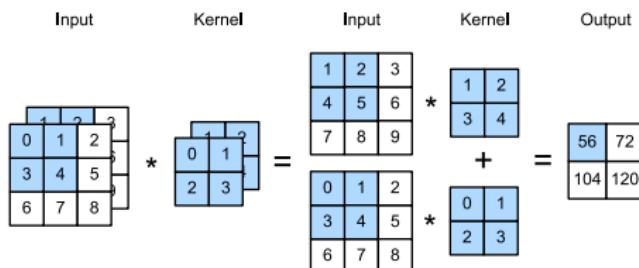
Input	Kernel	Output													
<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	$*$	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr></table>	0	1	2	3
0	1	2													
3	4	5													
6	7	8													
0	1														
2	3														
	=	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>19</td><td>25</td></tr><tr><td>37</td><td>43</td></tr></table>	19	25	37	43									
19	25														
37	43														

Opération de corrélation croisée à deux dimensions, Dive into Deep Learning, CC-BY-SA-4.0.

Convolution sur plusieurs canaux

La slide précédente est incomplète : une image a souvent plusieurs canaux (RGB) — les représentations intermédiaires d'un CNN aussi.

Un noyau de convolution a plusieurs canaux. Un par canal d'entrée:



Calcul de corrélation croisée avec 2 canaux d'entrée, Dive into Deep Learning, CC-BY-SA-4.0.

Habituellement la dimension de profondeur est omise quand on décrit une convolution (convolution 2×2 au lieu de $2 \times 2 \times 2$).

Remplissage

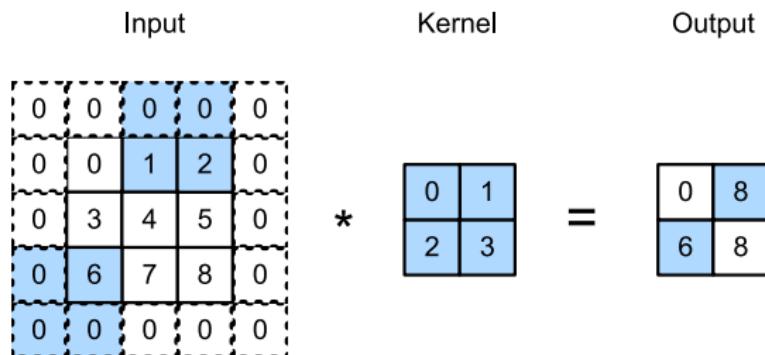
Le remplissage (*padding* en anglais) contrôle le changement de taille dû à l'opérateur de corrélation croisée en ajoutant des 0s:

Input	Kernel	Output
$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 \\ 0 & 3 & 4 & 5 & 0 \\ 0 & 6 & 7 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$	$\begin{bmatrix} 0 & 3 & 8 & 4 \\ 9 & 19 & 25 & 10 \\ 21 & 37 & 43 & 16 \\ 6 & 7 & 8 & 0 \end{bmatrix}$
*	=	

Corrélation croisée à deux dimensions avec remplissage, Dive into Deep Learning, CC-BY-SA-4.0.

Pas de convolution

Les pas de convolution (*strides* en anglais) permettent de réduire les dimensions spatiales dans un CNN :

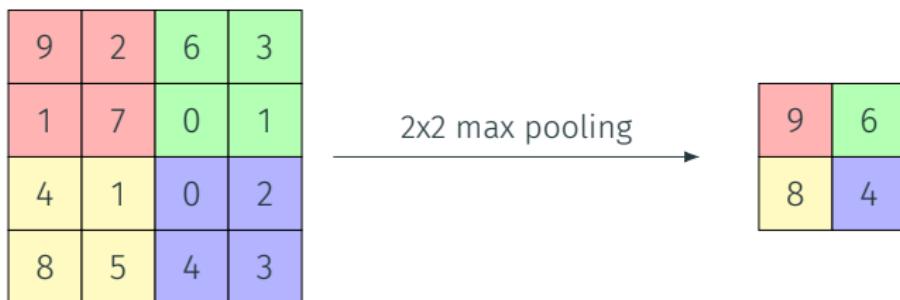


Corrélation croisée avec pas de 3 et 2 pour la hauteur, et la largeur, respectivement, Dive into Deep Learning, CC-BY-SA-4.0.

Ici, on est passé de 5×5 à 2×2 grâce aux pas de convolution (3, 2).

Agrégation

Pooling en anglais. L'autre mécanisme pour réduire les dimensions spatiales :

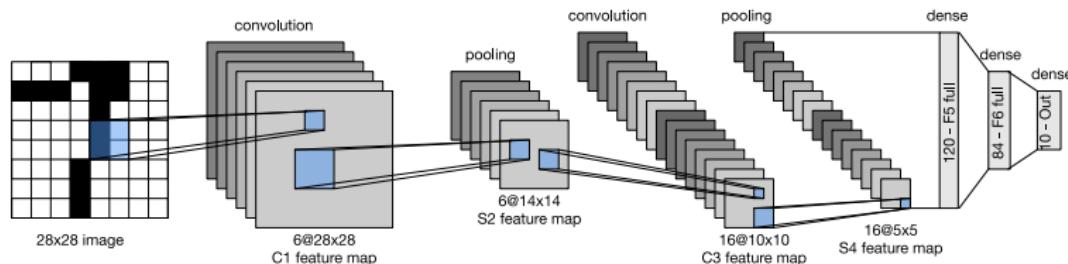


Opérateur de Max-pooling 2×2 , F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Une agrégation $n \times m$ est souvent utilisée avec des pas (n, m) .

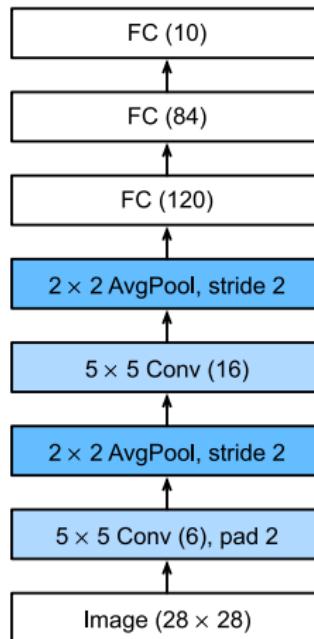
Combinaison des blocs de base en un réseau convolutif

Yann LeCun – maintenant lauréat du prix Turing – a proposé la première combinaison de ces blocs simples en un réseau complet, LeNet :



Flot de données dans LeNet, Dive into Deep Learning, CC-BY-SA-4.0.

Schéma de LeNet



Notation courte pour LeNet-5, Dive into Deep Learning, CC-BY-SA-4.0.

Entraînement

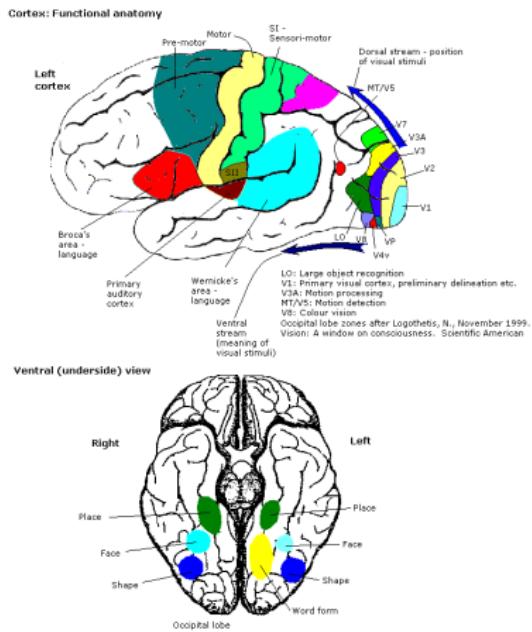
Par descente de gradient, comme les MLPs.

Moins de paramètres, plus d'opérations : cible parfaite pour les GPUs.



RTX 2080 GPU, nanadua11, Pixabay License.

Lien avec le cortex visuel



Couches successives:

V1 Orientation, lignes

V2 Formes, tailles,
couleurs

V3 Motricité

V4 Reconnaissance
d'objets

V5 Suivi d'objets

Le cortex visuel,

<https://en.wikibooks.org/wiki/User:RobinH>,

CC-BY-SA-3.0.

Démonstration

Convolutional neural networks

Avez-vous des questions ?

Réseaux de neurones

Réseaux de neurones à convolutions

Architectures modernes

Introduction

En 2012, AlexNet a remporté haut la main la compétition ImageNet.

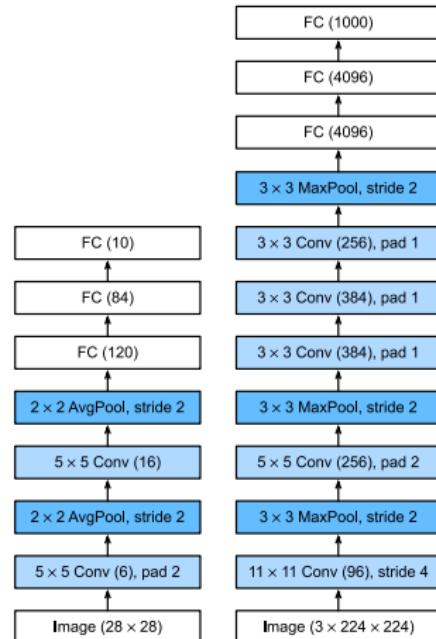
Depuis, de nombreuses architectures ont été proposées.

Top 5 en erreur de classification depuis 2012 : $\sim 15\% \rightarrow \sim 2\%$.

La plupart des modèles sont disponibles *directement* dans les libraries d'apprentissage profond.

AlexNet

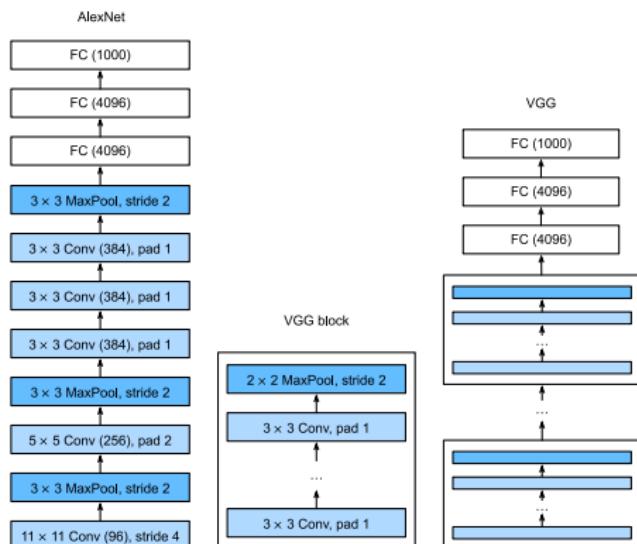
- Premier CNN profond entraîné sur des GPUs
- A introduit ReLU comme excellente non-linéarité pour entraîner des modèles profonds
- Sinon très similaire à LeNet — en plus profond



De LeNet (gauche) à AlexNet (droite), Dive into Deep Learning, CC-BY-SA-4.0.

VGG

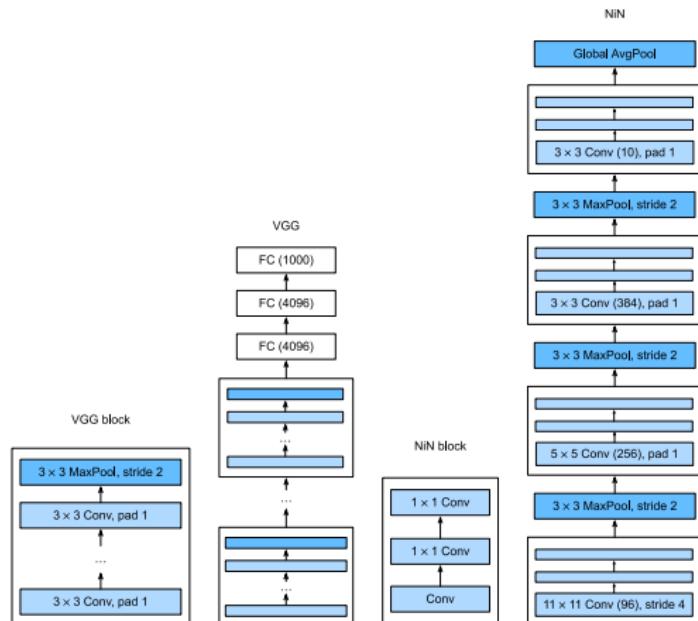
- A introduit la notion de bloc dans les CNNs
- Sinon similaire à AlexNet
 - en plus gros & plus profond
- A longtemps été très populaire comme modèle de base pour le transfert d'apprentissage



D'AlexNet à VGG, qui est construit avec des blocs, Dive into Deep Learning, CC-BY-SA-4.0.

Network in Network

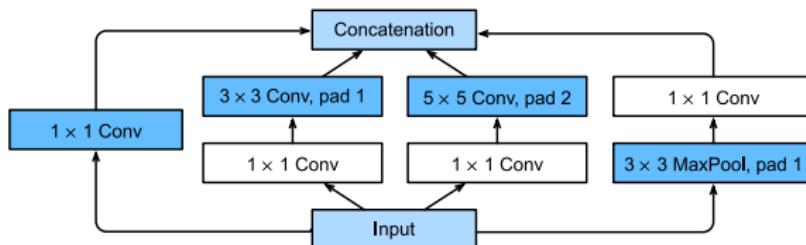
- Utilise des noyaux 1×1 pour appliquer un « réseau dans un réseau »
- Un noyau 1×1 ne change pas la structure spatiale. Seulement le nombre de canaux
- Il revient à appliquer un MLP à chaque position spatiale



Comparaison des architectures de VGG et NiN, Dive into Deep Learning, CC-BY-SA-4.0.

Inception/GoogLeNet – Bloc

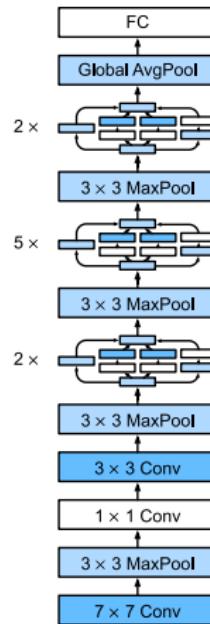
Utilise un bloc qui combine des noyaux de tailles différentes.



Structure d'un bloc Inception, Dive into Deep Learning, CC-BY-SA-4.0.

Inception/GoogLeNet – Modèle

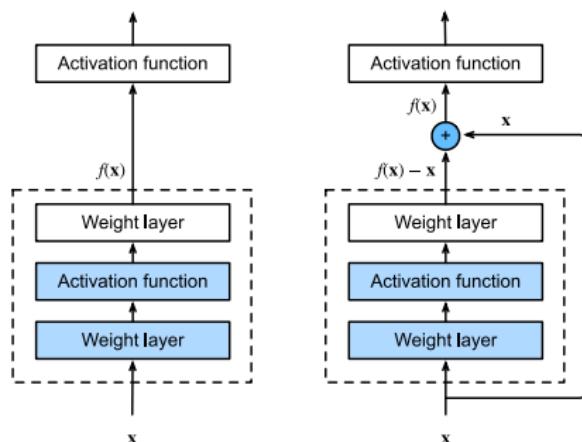
Mis à part l'architecture du bloc, le reste est standard.



L'architecture de GoogLeNet, Dive into Deep Learning, CC-BY-SA-4.0.

ResNet – Blocs résiduels

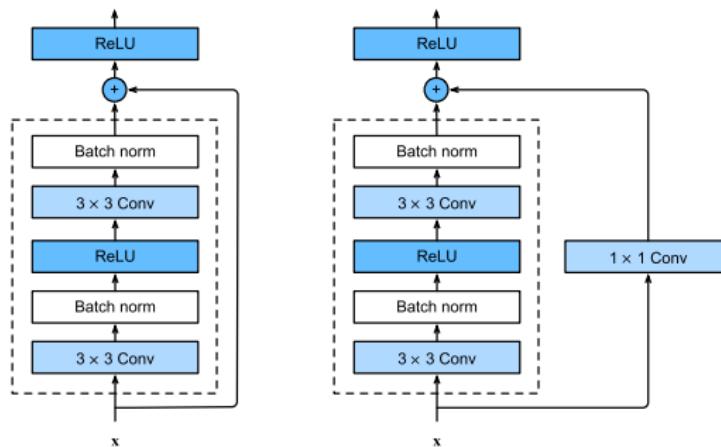
- Contribution architecturale majeure: le bloc résiduel
- Fait évoluer le problème d'apprentissage de *apprendre y depuis x* à *apprendre $y - x$ depuis x*
- Par exemple, l'identité devient triviale à apprendre
- Très bonnes propriétés de circulation du gradient
- Utilise la normalisation de batch (centrage et division par l'écart-type des batchs)



Un bloc normal (gauche) et un bloc résiduel (droite), Dive into Deep Learning, CC-BY-SA-4.0.

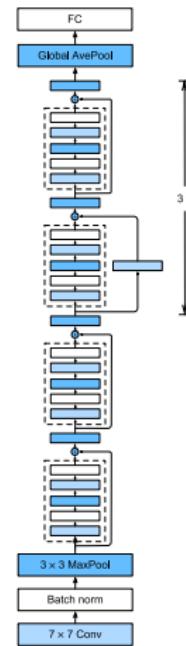
ResNet – Blocs ResNet

ResNet – comme GoogLeNet – utilise des noyaux 1×1 pour réduire la dimensionnalité.



Un bloc ResNet avec et sans convolution 1×1 , Dive into Deep Learning, CC-BY-SA-4.0.

ResNet – Modèle

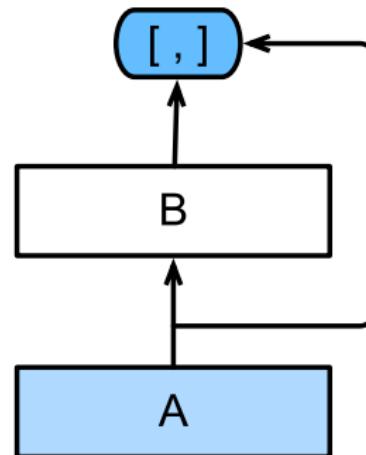
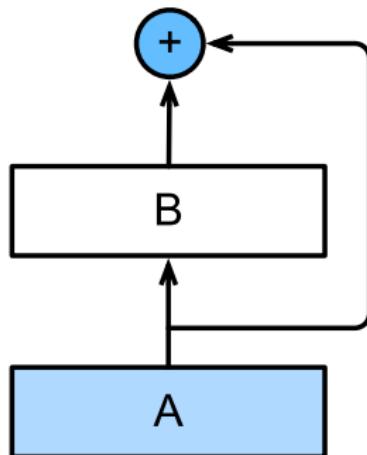


Le reste du modèle est standard.

L'architecture de ResNet-18, Dive into Deep Learning, CC-BY-SA-4.0.

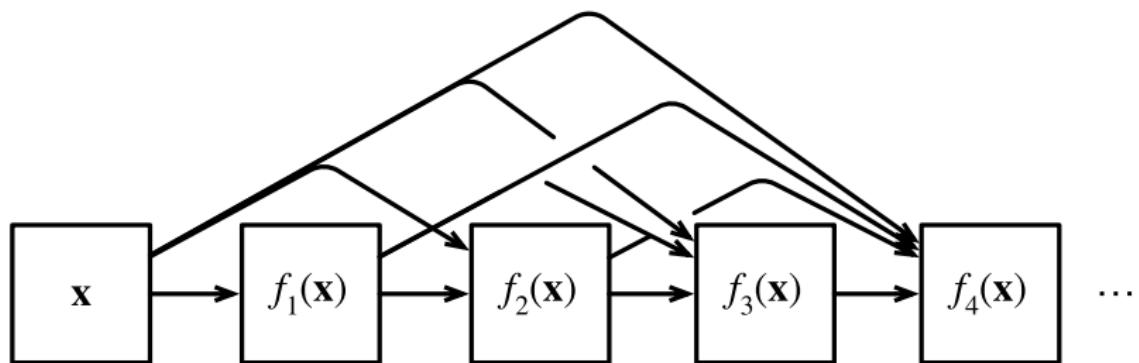
DenseNet – Bloc

Variation sur ResNet: utilise la concaténation des sorties plutôt que l'addition.



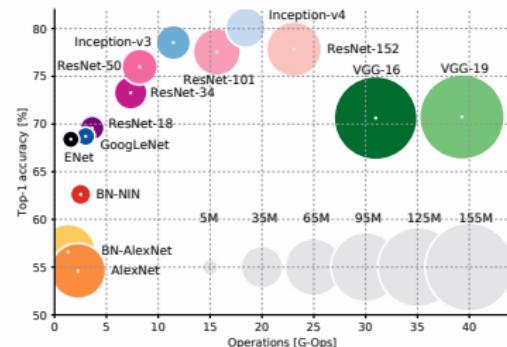
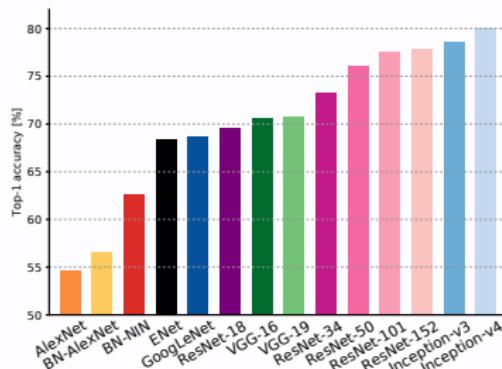
ResNet utilise l'addition (gauche), DenseNet utilise la concaténation (droite), Dive into Deep Learning, CC-BY-SA-4.0.

DenseNet – Modèle



Connexions denses dans DenseNet, Dive into Deep Learning, CC-BY-SA-4.0.

Comparaison des différents modèles



An analysis of deep neural network models for practical applications, A. Canziani, A. Paszke & E. Culurciello, arXiv.

Avez-vous des questions ?

Réseaux de neurones

Réseaux de neurones à convolutions

Implémentation en Keras

Dimension du tensor d'entrée

```
print(f"Forme des images : {images.shape}")
```

```
Forme des images : (14034, 150, 150, 3)
```

Définition d'un modèle CNN

```
model = tf.keras.models.Sequential()
# Une première couche de 10 convolutions de 3x3 pixels
model.add(tf.keras.layers.Conv2D(10,
                                kernel_size=(3, 3),
                                activation="relu",
                                input_shape=(150, 150, 3)))
# Une couche de max pooling
model.add(tf.keras.layers.MaxPool2D(3,3))
# Une couche de redimensionnement, qui aplatis le tenseur
model.add(tf.keras.layers.Flatten())
# Une couche "Dense" avec 6 sorties et un softmax
model.add(tf.keras.layers.Dense(6, activation="softmax"))
# Compilation du modèle avec la définition de la loss
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
              loss="sparse_categorical_crossentropy",
              metrics=[ "accuracy"])
```

Affichage du modèle

```
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 10)	280
max_pooling2d (MaxPooling2D)	(None, 49, 49, 10)	0
flatten (Flatten)	(None, 24010)	0
dense (Dense)	(None, 6)	144066
<hr/>		
Total params:	144,346	
Trainable params:	144,346	
Non-trainable params:	0	
<hr/>		

Avez-vous des questions ?

Travaux pratiques

1. [Instructions, données paysages](#)
2. [Instructions, données émotions](#)

Ressources complémentaires

[Image Transformer](#)

Réseaux de neurones

Auto-encodeurs

Réseaux de neurones

Auto-encodeurs

Présentation

Schéma général

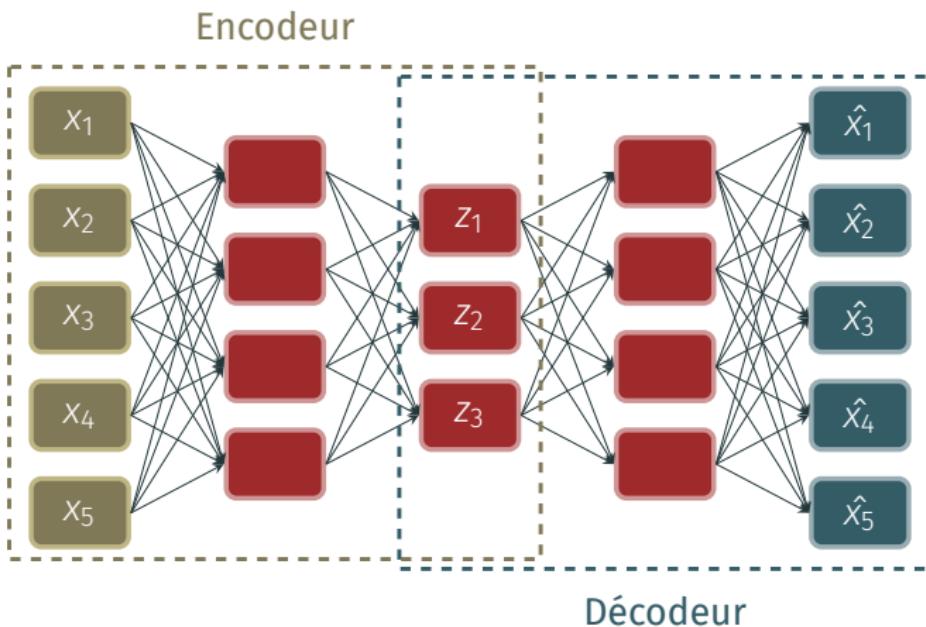
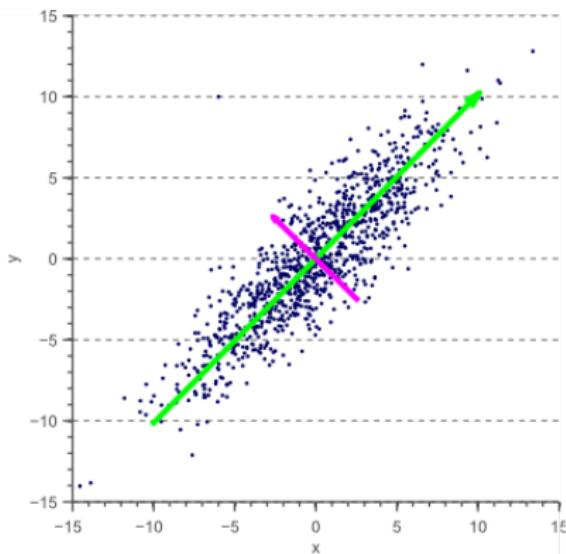


Schéma d'auto-encodeur, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Cas linéaire

Activations linéaires et fonction de perte moindre carrés \approx ACP



A scatter plot of samples that are distributed according a bivariate Gaussian distribution, member Nicoguaro de Wikimedia, CC-BY-4.0.

Cas linéaire

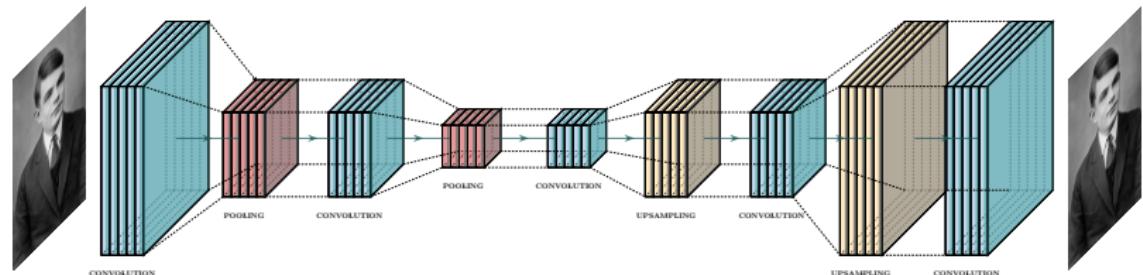
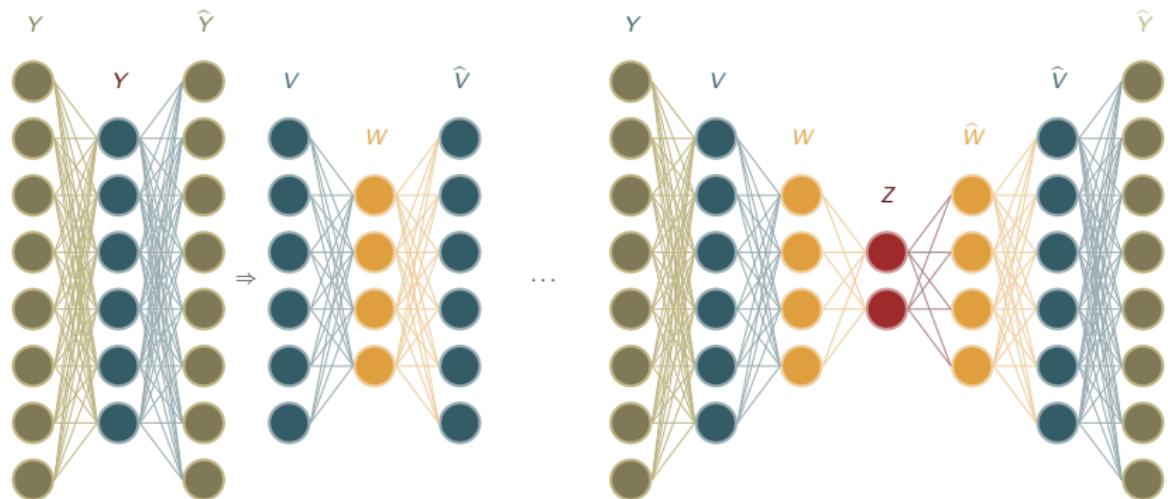


Schéma d'un autoencodeur profond, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Auto-encodeurs profonds

Aussi appelées auto-encodeurs empilés.



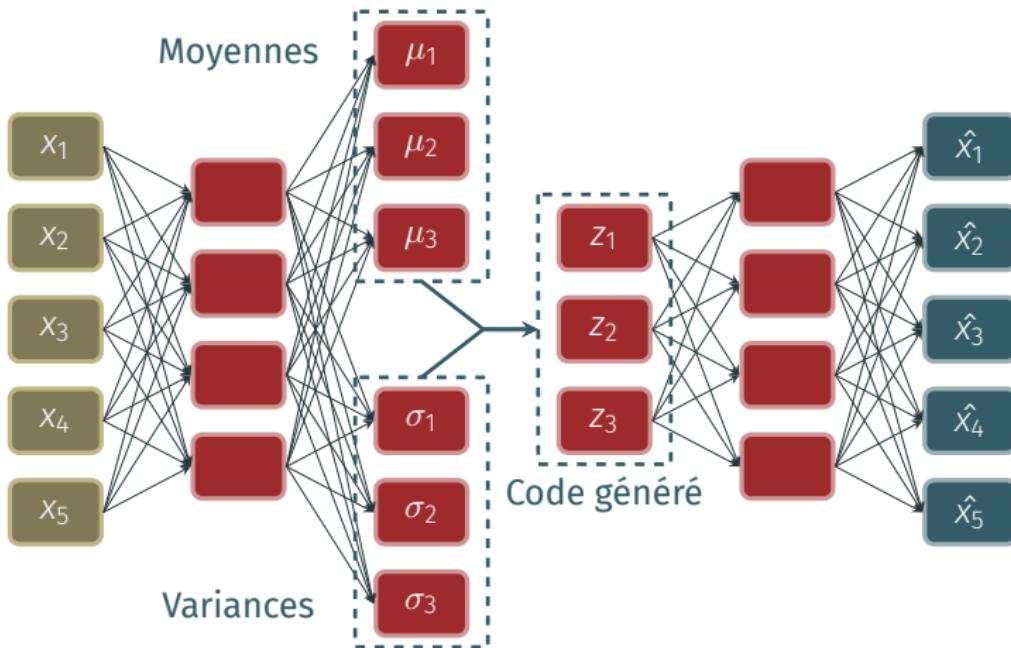
Auto-encodeurs empilés, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Auto-encodeurs débruiteurs



Auto-encodeur débruiteur, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Auto-encodeurs variationnels



Auto-encodeurs Variationnels (VAE), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Avez-vous des questions ?

Travaux pratiques

1. [Instructions, auto-encodeurs simples](#)
2. [Instructions, auto-encodeurs débruiteurs](#)

Réseaux de neurones

Réseaux génératifs antagonistes

Principe

- Utiliser le pouvoir d'un réseau pour en entraîner un autre, génératif
- Modéliser l'apprentissage comme un jeu
- Le premier réseau génère des exemples depuis du bruit
- Le deuxième distingue entre les générations du premier et les vrais exemples

Architecture

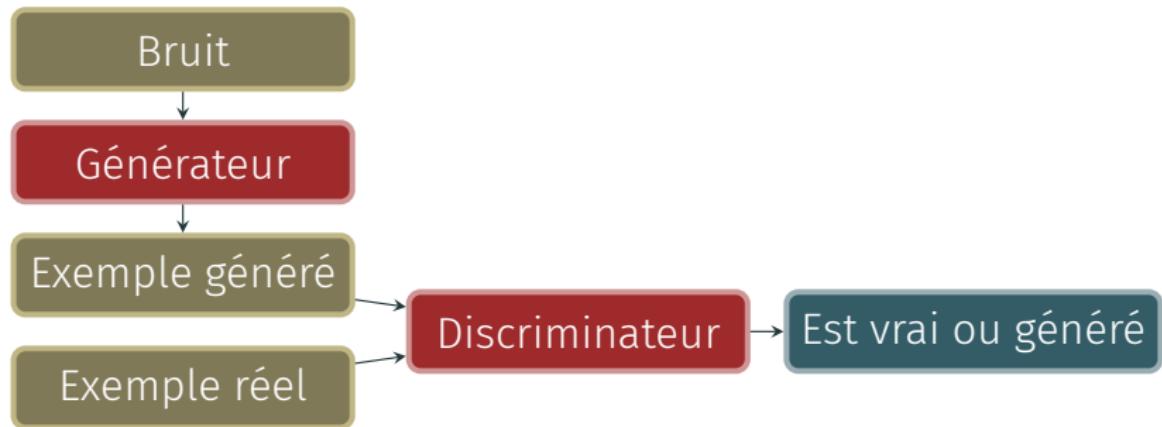


Schéma de réseaux générateurs adverses (GAN), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Discriminateur

Entraîné pour maximiser la probabilité d'assigner 1 aux vrais exemples et 0 aux faux :

$$\max_D \left\{ \underbrace{y \log D(\mathbf{x})}_{\text{vrai exemple}} + \underbrace{(1 - y) \log(1 - D(\mathbf{x}))}_{\text{faux exemple}} \right\}$$

- Pour un vrai exemple ($y = 1$), le terme à maximiser est $\log D(\mathbf{x})$
 $\Rightarrow D(\mathbf{x})$ doit tendre vers 1
- Pour un faux exemple ($y = 0$), le terme à maximiser est
 $\log(1 - D(\mathbf{x}))$
 $\Rightarrow D(\mathbf{x})$ doit tendre vers 0

Générateur

Entraîné pour minimiser la probabilité que D assigne 0 à de faux exemples :

$$\min_G \{(1 - y) \log(1 - D(G(z)))\}$$

Le terme à minimiser est $\log(1 - D(G(z)))$

$\Rightarrow D(G(z))$ doit tendre vers 1

Fonction d'entraînement complète

Les deux équations précédentes, combinées sur tous les exemples (vrais et faux) :

$$\max_D \min_G \left\{ \underbrace{\mathbb{E}_{x \sim \text{Data}} \log D(x)}_{\text{Seulement } D} + \underbrace{\mathbb{E}_{z \sim \text{Noise}} \log(1 - D(G(z)))}_{D \& G} \right\}$$

⇒ Description du jeu minimax entre G & D .

Algorithme d'apprentissage

Pour chaque itération d'apprentissage :

1. Échantillonner le vecteur de bruit Z
2. Échantillonner un batch de vrais exemples X
3. Mettre à jour les paramètres de $D(\theta_D)$ par montée de gradient :

$$\theta_D \leftarrow \theta_D + \alpha \nabla_{\theta_D} \frac{1}{m} \sum_{i=1}^m [\log D(X_i) + \log (1 - D(G(Z_i)))]$$

4. Mettre à jour les paramètres de $G(\theta_G)$ par descente de gradient :

$$\theta_G \leftarrow \theta_G - \alpha \nabla_{\theta_G} \frac{1}{m} \sum_{i=1}^m [\log (1 - D(G(Z_i)))]$$

Les étapes 3. et 4. peuvent être répétées plusieurs fois.

Stabilité de l'apprentissage

Entraîner un GAN est difficile :

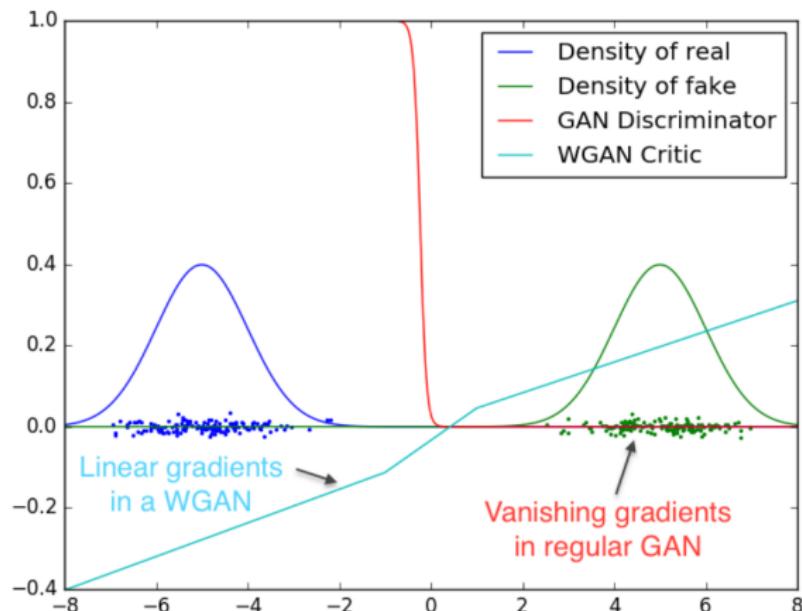
- Le discriminateur peut dominer
- Le générateur peut boucler sur certaines générations (mode collapse)
- Il peut être nécessaire d'entraîner un réseau plus que l'autre
- La distribution du bruit est importante
- Le pas d'apprentissage est important
- ...

Conseils utiles : <https://github.com/soumith/ganhacks>

Wasserstein GAN

- Utilise la distance de transport optimal entre la distribution des vraies et fausses données au lieu de l'entropie binaire croisée
- Gradient non saturé : meilleur signal d'apprentissage
- Rend l'optimisation plus facile

Wasserstein GAN – Visualisation



Wasserstein GAN, M. Arjovsky et al, arXiv.

BEGAN

- Utilise un auto-encodeur au lieu d'un classifieur comme discriminateur
- Le discriminateur doit avoir une plus grande erreur de reconstruction pour les fausses images

Avez-vous des questions ?

Démonstration

- [Visualisation GAN](#)
- [DCGAN](#)

Réseaux de neurones

Traitement automatique des langues

Introduction

Plusieurs noms :

- Traitement du texte
- Traitement automatique des langues (TAL)
- Traitement automatique du langage naturel (TALN)
- Natural Language Processing (NLP)

Utilisation du machine learning pour traiter des données textuelles.

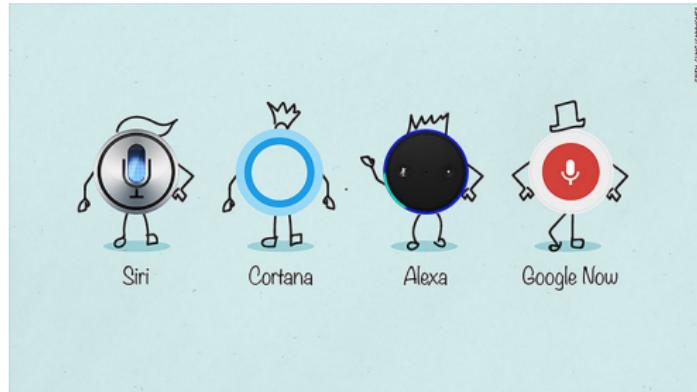
Cas d'usage – Classification

- auteur (gutenberg.org : 57k livres)
- polarité/sentiment (Kaggle movie review : 222k commentaires rotten tomatoes)
- reconnaissance d'entités nommées (Kaggle Annotated Corpus for NER : 1.3M tags)
- thème/genre (gutenberg.org : 57k livres)
- ...

Cas d'usage – Compréhension

- Question/réponses (SQuAD : 150k questions)
- Traduction (europarl : 450k phrases alignées)
- ...

Cas d'usage – Transcription et synthèse de la parole



Cas d'usage – Identification du locuteur



Cas d'usage – Chatbot



The image shows a screenshot of a mobile application interface for a chatbot named Zo. On the left, there is a large circular profile picture of a smiling woman with blonde hair. Below the profile picture, the text "Zo" and "zo.ai" is displayed. On the right, there is a conversation transcript between the user and the bot. The user's messages are in white boxes with a blue border, and the bot's messages are in green boxes with a white border. The conversation goes as follows:

- User: So the guy LITERALLY POSTED THE JIF.
- Bot: ahahaha, it was.. you just stood there staring at me until I fell
- User: What?
- Bot: Life.
- Bot: Man, me too.
- User: You're not making sense
- Bot: I've spent a majority of my life not making sense so... I must be showing u my true colors.
- User: Haha ok

Cas d'usage – Moteur de recherche

YAHOO!

 bing

 Google

 DuckDuck**Go**

 Qwant

Cas d'usage – Extraction de données



Cas d'usage – Analyse de sentiments



Cas d'usage – Résumé

Input: Article 1st sentence	Model-written headline
metro-goldwyn-mayer reported a third-quarter net loss of dlrs 16 million due mainly to the effect of accounting rules adopted this year	mgm reports 16 million net loss on higher revenue
starting from july 1, the island province of hainan in southern china will implement strict market access control on all incoming livestock and animal products to prevent the possible spread of epidemic diseases	hainan to curb spread of diseases
australian wine exports hit a record 52.1 million liters worth 260 million dollars (143 million us) in september, the government statistics office reported on monday	australian wine exports hit record high in september

Cas d'usage – Traduction

Traduire français (langue identifiée)

Bonjour,

Nos filles souhaitent faire un échange ensemble dans le cadre du programme Brigitte Sauzay. Avant de passer au dossier administratif, nous souhaitions vous poser quelques questions pour s'assurer que nous envisagions cet échange entre nos filles de la même façon.

Le professeur d'allemand de Gaïa a gentiment accepté de traduire nos questions afin d'éviter des incompréhensions à cause de problèmes de traduction.

Concernant les dates, Emma nous a dit que vous souhaitiez que Gaïa soit repartie avant le 19 juin car vous partiez en vacances.

Traduire en anglais

Hello,

Our daughters would like to do an exchange together as part of the Brigitte Sauzay program. Before moving on to the administrative file, we wanted to ask you a few questions to ensure that we consider this exchange between our daughters in the same way.

Gaïa's German teacher kindly agreed to translate our questions in order to avoid misunderstandings due to translation problems.

Regarding the dates, Emma told us that you wanted Gaïa to leave before June 19 because you were going on vacation.

État de l'art – Vers 2015



Language Technology

making good progress

mostly solved

Spam detection

Let's go to Agra!
Buy V1AGRA ...



Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV
Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC
Einstein met with UN officials in Princeton

Sentiment analysis

Best roast chicken in San Francisco!



The waiter ignored us for 20 minutes.



Coreference resolution

Carter told Mubarak he shouldn't run again.



Word sense disambiguation (WSD)

I need new batteries for my *mouse*.

Parsing



I can see Alcatraz from the window!

Machine translation (MT)

第13届上海国际电影节开幕...

The 13th Shanghai International Film Festival...

Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30



May 27

add

still really hard

Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday
ABC has been taken over by XYZ

Summarization

The Dow Jones is up
The S&P500 jumped
Housing prices rose



Economy is good

Dialog

Where is Citizen Kane playing in SF?
Castro Theatre at 7:30. Do you want a ticket?



État de l'art – 2019

mostly solved

Spam detection

Let's go to Agra!
Buy VIAGRA ...



Part-of-speech (POS) tagging

ADJ ADJ NOUN VERB ADV
Colorless green ideas sleep furiously.

Named entity recognition (NER)

PERSON ORG LOC
Einstein met with UN officials in Princeton

Word sense disambiguation (WSD)

I need new batteries for my *mouse*.



Machine translation (MT)

第13届上海国际电影节开幕...
The 13th Shanghai International Film Festival...



Information extraction (IE)

You're invited to our dinner party, Friday May 27 at 8:30



making good progress

Sentiment analysis

Best roast chicken in San Francisco!
The waiter ignored us for 20 minutes.



Coreference resolution

Carter told Mubarak he shouldn't run again.

Parsing



Question answering (QA)

Q. How effective is ibuprofen in reducing fever in patients with acute febrile illness?

Paraphrase

XYZ acquired ABC yesterday
ABC has been taken over by XYZ

Summarization

The Dow Jones is up
The S&P500 jumped
Housing prices rose

Economy is good

still really hard

Dialog

Where is Citizen Kane playing in SF?
Castro Theatre at 7:30. Do you want a ticket?



Réseaux de neurones

Traitement automatique des langues

Prétraitements

Collecte

Des sources variées :

- Wikipedia
- Articles de journaux
- Littérature
- User Generated Content
 - Blogs
 - Commentaires
 - Réseaux sociaux

Une source ⇒ un « web scraper »

Caractères accentués et spéciaux

```
import unicodedata

def utf8_to_ascii(string: str) -> str:
    normalized = unicodedata.normalize("NFKD", string)
    ascii_bytes = normalized.encode("ascii", "ignore")
    ascii_string = ascii_bytes.decode()
    return ascii_string

utf8_string = "Vous êtes le Père Noël ? s'étonna le petit garçon."
ascii_string = utf8_to_ascii(utf8_string)

print(ascii_string)
```

Vous etes le Pere Noel ? s'etonna le petit garcon.

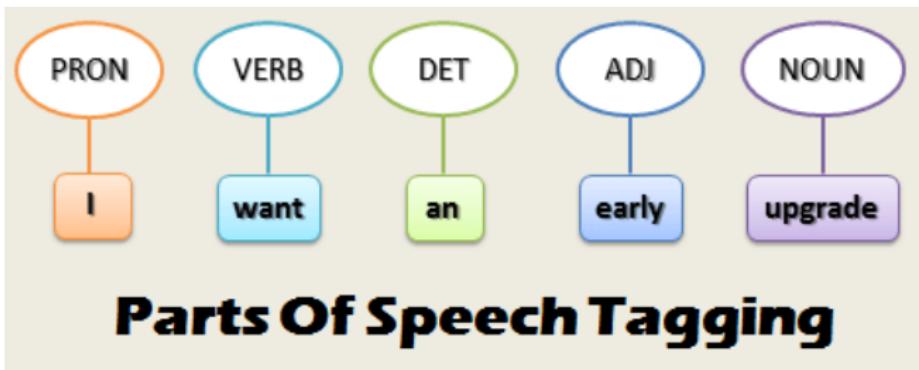
Tokenisation

Séparer une chaîne de caractères en tokens n'est pas trivial :

Le Dr. Pond èleve des poules. L'éleveur les sur-exploite.

(en phrases ou en mots)

Étiquetage morpho-syntaxique

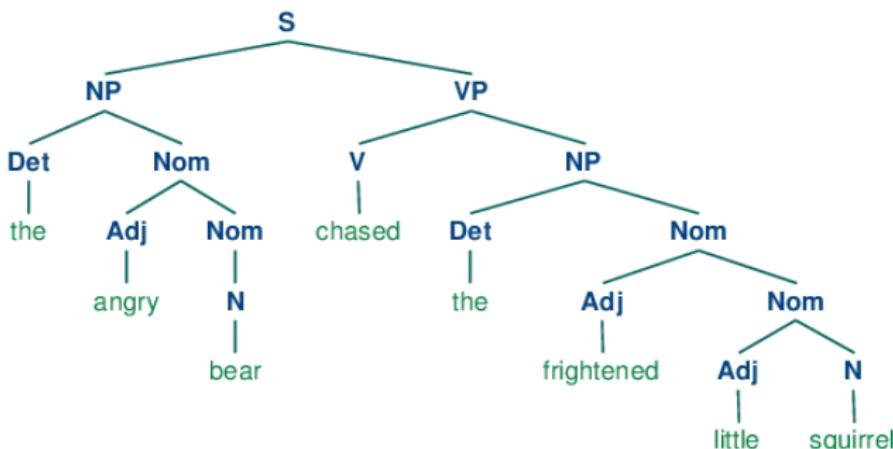


Reconnaissance d'entités nommées



Figure 1: An example of NER application on an example text

Analyse syntaxique



Lemmatisation

Exprimer les mots (ou groupes de mots) sous une forme canonique :

Mot	Lemme
jouant	jouer
ont été jouées	jouer
étoiles	étoile
claires	clair
noire	noir

Outils – WordNet

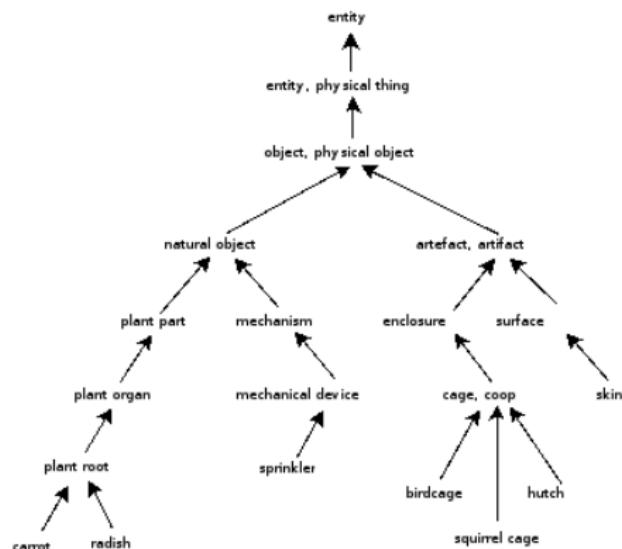
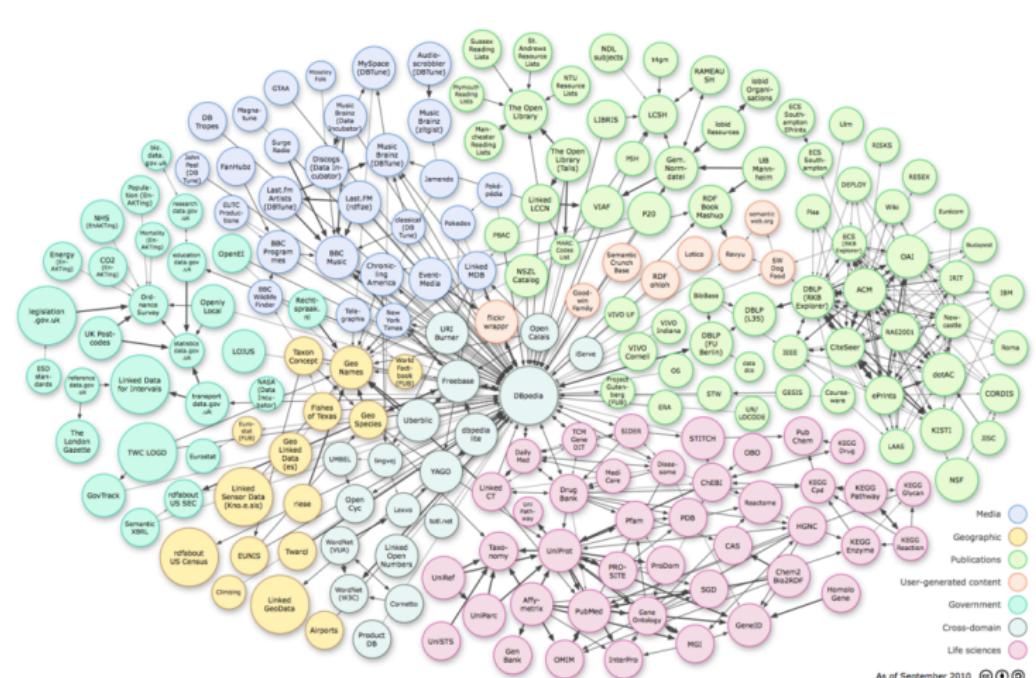


Figure 1. "is_a" relation example

Projet sur le français : WOLF (Wordnet Libre du Français)

Outils – DBpedia



As of September 2010

Illustration des interconnexions avec DBpedia, Charles Sturt University, CC-BY-SA-4.0.

Réseaux de neurones

Traitement automatique des langues

Représentations TF-IDF

TF-IDF

Un document = un vecteur de la taille d'un dictionnaire. (donc à dimension fixe)

$$w_{ij} = tf_{ij} \log \frac{N}{df_i}$$

où :

tf_{ij} Nombre d'occurrences du mot i dans le document j

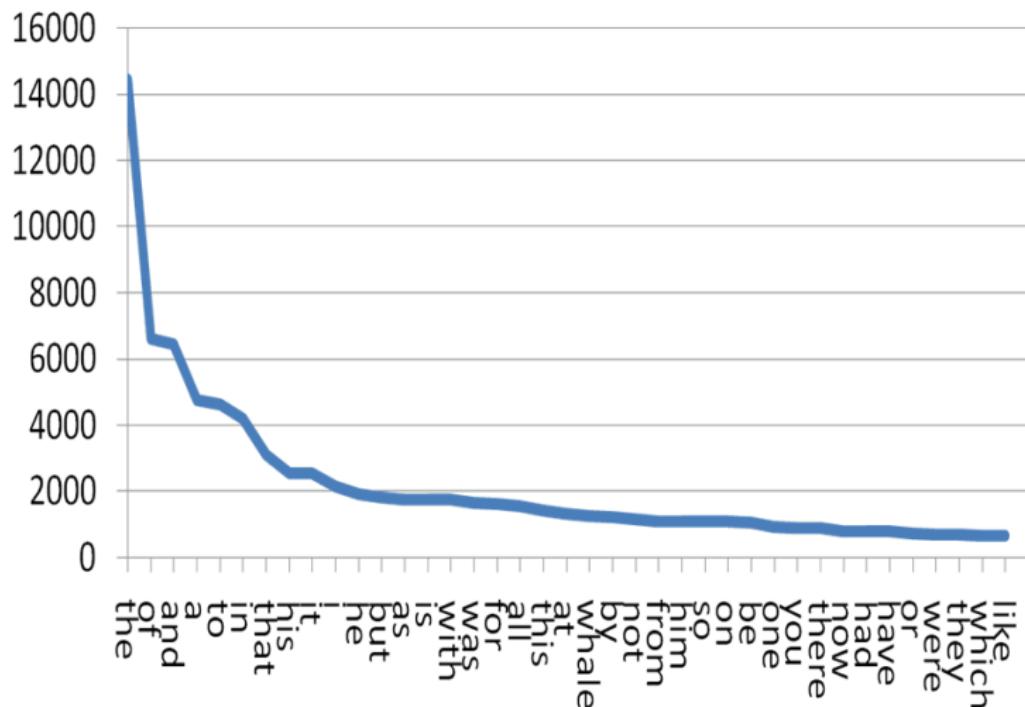
df_i Nombre de documents contenant le mot i

N Nombre total de documents

⇒ Produit scalaire, SVM, arbres, réseaux de neurones, ...

TF-IDF – Intérêt de df_i

Loi de Zipf, justifiant l'utilisation du terme df_i



TF-IDF

Utilisation de n -grammes de mots ou de caractères.

Exemple : « Le chien mange de la viande »

Bigrammes de mots :

- le-chien, chien-mange, mange-de, de-la, la-viande

Trigrammes de caractères :

- le_, e_c, _ch, chi, hie, ien, en_, n_m, _ma, man, ...

Analyse sémantique latente

Famille algorithmique des modèles thématiques (topic models) :

- \approx PCA sur la matrice des documents \rightarrow relations entre les mots
- Composantes principales \approx topics

Par exemple : un axe correspond au champ lexical du sport, un autre à celui de l'économie, etc.

Réseaux de neurones

Traitement automatique des langues

Word Embeddings

Word Embeddings

mot = indice dans un dictionnaire (dimension > 30000)

mot = vecteur “sémantique” (dimension < 1000)

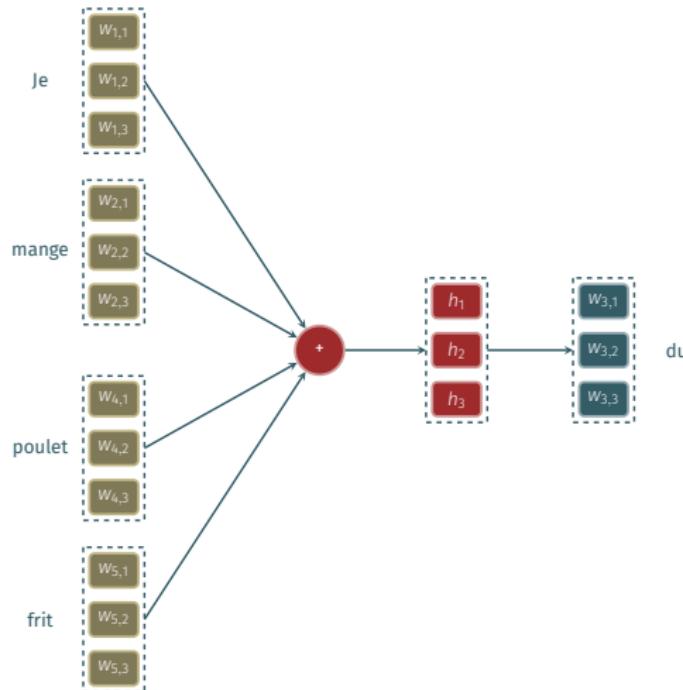
- word2vec
- CBOW/Skip-Gram
- GloVe
- Thought vector (pour des phrases ou même des documents entiers)
- ...

Word Embeddings

	1	2	...	K
je	0.34	0.29	...	0.91
tu	0.88	0.34	...	0.1
mange	0.41	0.32	...	0.80
cuisine	0.2	0.16	...	0.62
un	0.83	0.95	...	0.96
ce	0.22	0.97	...	0.49
le	0.6	0.86	...	0.7
poulet	0.95	0.80	...	0.84
poisson	0.42	0.76	...	0.56
frit	0.24	0.76	...	0.13
mariné	0.2	0.60	...	0.64

Chaque mot est initialement projeté au hasard dans l'espace en dimension K.

Word Embeddings



Continuous Bag of Word (CBOW), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Word Embeddings

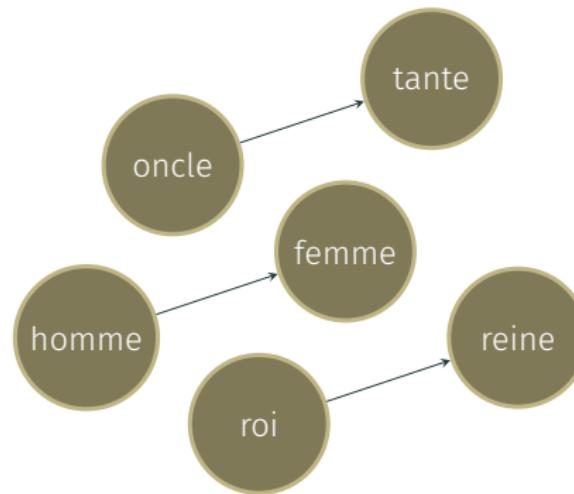


Illustration d'une direction sémantique dans l'espace word2vec, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Visualisation de l'espace word2vec Word Embeddings à télécharger

Avez-vous des questions ?

Ressources complémentaires

- [Word embeddings](#)
- [Dépôt de modèles modernes de langages déjà appris](#)
- [Un tutoriel de modèles à attention](#)

Réseaux de neurones

Données séquentielles

Réseaux de neurones

Données séquentielles

Introduction

Introduction

Datasets dont les instances sont des séquences de caractéristique(s).

Définition

Soit $X = (x_i)_{1 \leq i \leq k}$ un dataset de k exemples :

où

$x_i = (x_i^1, \dots, x_i^{n_i})$ avec x_i une séquence de n_i frames

Pour des séries d'entiers par exemple :

$$x_1 = (1, 3, 5, 2, 8)$$

$$x_2 = (7, 3)$$

$$x_3 = (4, 0, 9, 1)$$

$$x_4 = (\dots)$$

Pour des séries de vecteurs :

$$x_1 = ([2, 5], [9, 8], [3, 6])$$

$$x_2 = ([1, 1], [3, 4], [5, 4], [3, 2], [8, 1])$$

$$x_3 = (\dots)$$

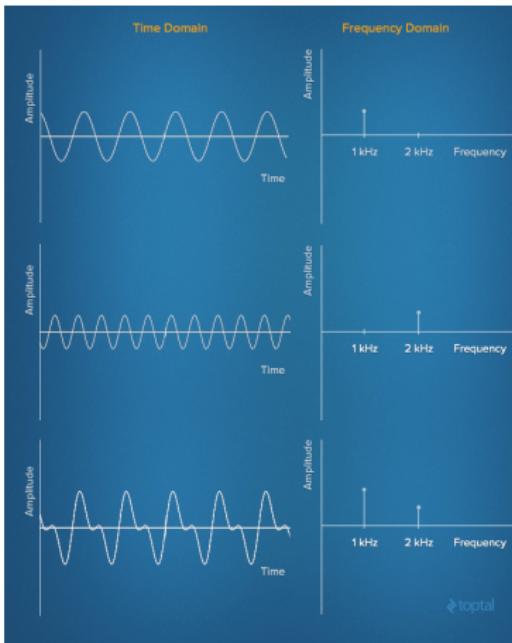
Types de données séquentielles – texte

1. lorem ipsum dolor sit amet, consectetur adipiscing elit, ...
2. Great minds discuss ideas; average minds discuss events; small minds discuss people.
3. Hier, mon voisin a mangé une pomme et sa femme une poire.
4. ...

Types de données séquentielles – ADN

1. ATGCGATCTATCGCTAGCCGCGCTATACGCA
2. GATTATAGCTAGCTCGCGCTATATCGCTAGCTAGCTAGC

Types de données séquentielles – son



Types de données séquentielles – vidéo



Types de données séquentielles – et bien d'autres

Economie actions, obligations, monnaies

Météo date, latitude, longitude, température, pression, vent, pluie

Santé date, température, pouls

Comportements Clients date, achat, fréquence de visite

...

Réseaux de neurones

Données séquentielles

Modèles Séquentiels

Tâches variées

- Prédiction d'une classe
- Prédiction de la suite d'une séquence (one step)
- Prédiction de la suite d'une séquence (multi step)
- Génération de séquence
- Découverte de patterns
- Clustering
- Détection d'anomalies

Tâches – Prédiction d'une classe

input : $x_i = (1, 6, 8, 4)$

output : positif ou négatif (cas binaire)

- Son (chant d'oiseau, personne, genre musical, ...)
- Vidéos (film, documentaire, stand up, ...)

Tâches – Prédiction de la suite d'une séquence (one step)

input : $x_i = (1, 2, 3, 4, 5)$

output : 6

- Données Économiques
- Comportement Clients
- Météo
- Modèle de Langage

Tâches – Prédiction de la suite d'une séquence (multi step)

input : $x_i = (1, 2, 3, 4, 5)$

output : (6, 7, 8)

- Traduction automatique
- Données Économiques
- Météo

Tâches – Génération

input : $x_1 = (1, 3, 5)$, $x_2 = (7, 9, 11)$

output : (5, 7, 9)

Quasi toujours conditionnée.

- Texte
- Texte vers voix
- Voix vers texte
- Génération de partitions

Tâches – Découverte de Patterns

- Découverte de gènes
- Compression de signal
- Décrire et comprendre des phénomènes

Avez-vous des questions ?

Réseaux de neurones
Réseaux de neurones récurrents

Réseaux de neurones

Réseaux de neurones récurrents

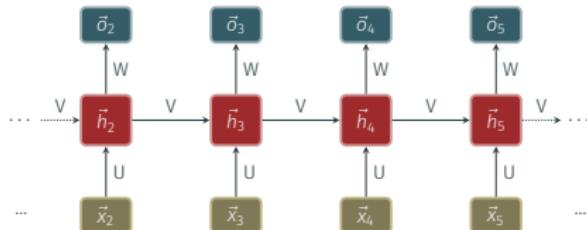
Modèle

Introduction

Caractéristiques des réseaux récurrents :

- Gestion des séquences
- Traitement séquentiel des entrées
- RéPLICATION d'un réseau autant de fois qu'il y a d'entrées

Modèle



Réseau récurrent simple en vue déroulée, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

x_t Entrée

h_t Couche cachée

o_t Sortie

U, V, W Matrices de poids

b_h, b_o Vecteurs de biais

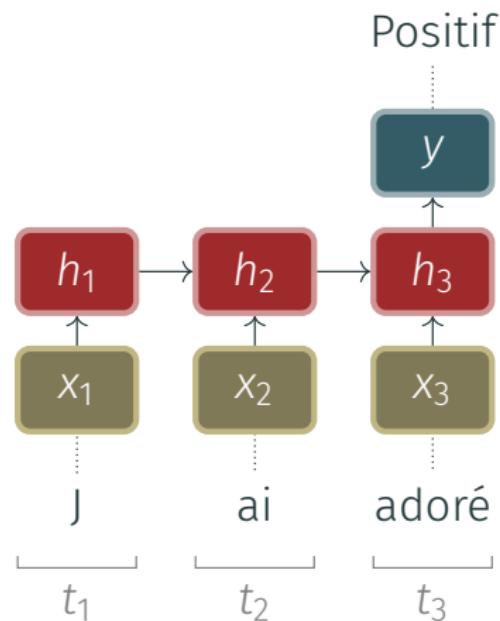
σ_h, σ_o Activations tanh

$$h_t = \sigma_h(Ux_t + Vh_{t-1} + b_h)$$

$$o_t = \sigma_o(Wh_t + b_o)$$

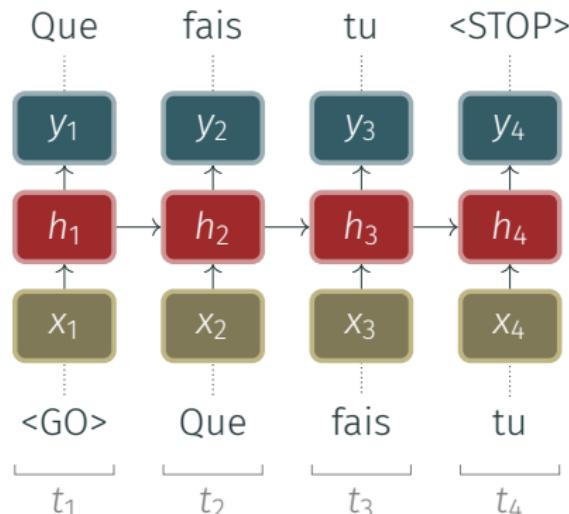
Prédiction d'une sortie

Exemple : détection de polarité.



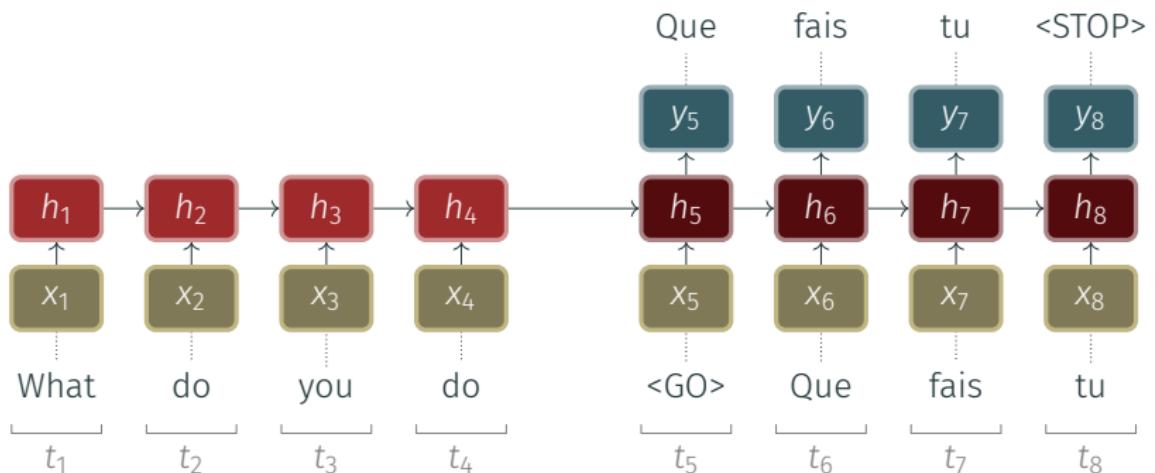
RNN pour la classification, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Prédiction d'autant de sorties qu'il y a d'entrées



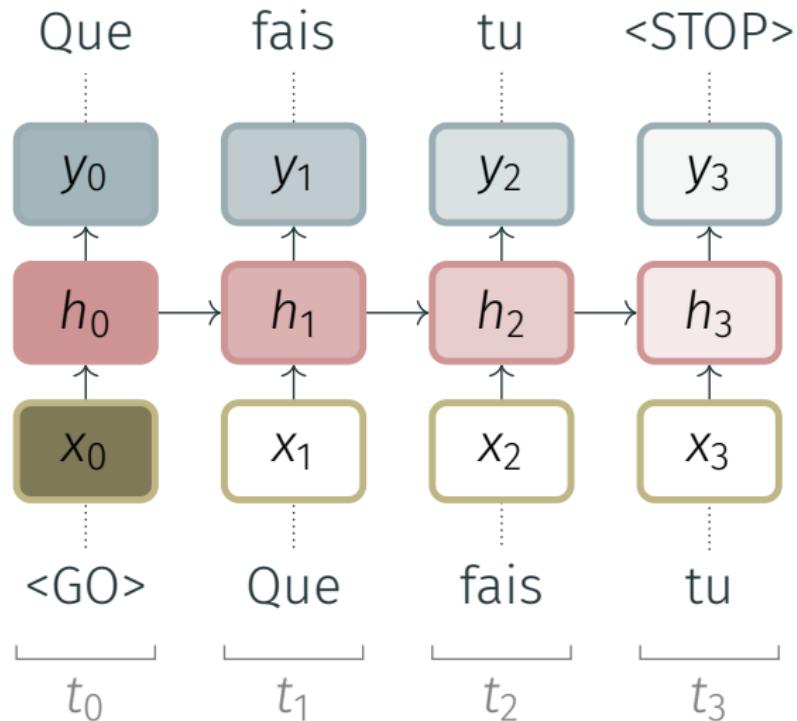
Un RNN comme modèle de langage, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Prédiction d'un nombre arbitraire de sorties



Sequence to sequence (seq2seq), F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Problème du gradient qui disparaît (vanishing gradient)



Réseaux de neurones

Réseaux de neurones récurrents

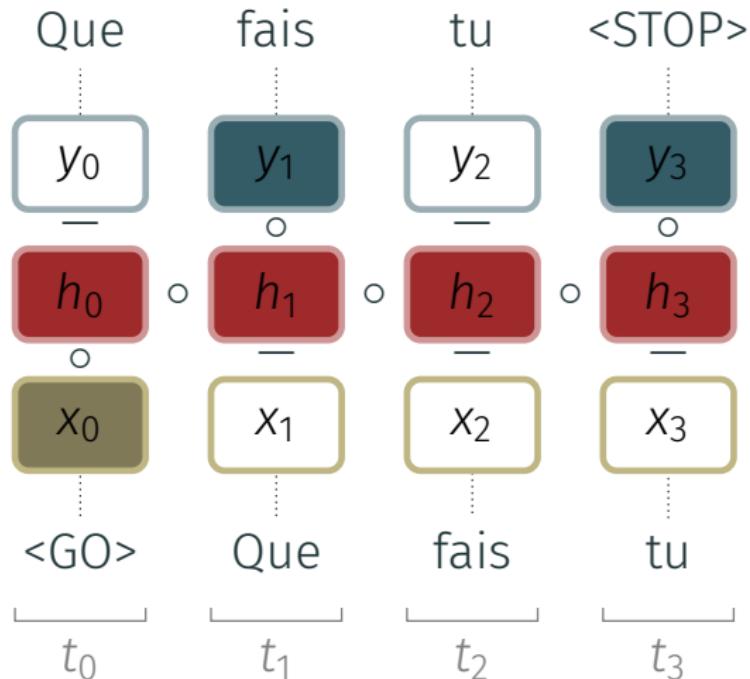
Réseaux à longue mémoire de court terme

Principe

Lutter contre les gradients qui disparaissent :

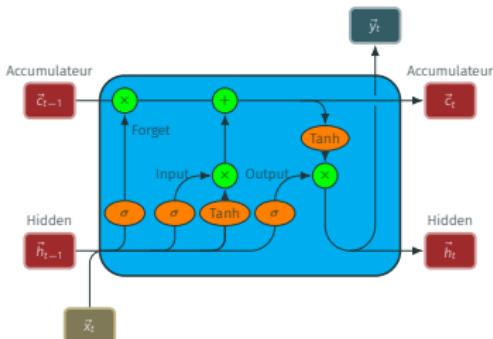
- Introduction d'une mémoire
- Update additive de la mémoire (gradient plus facile à conserver)
- « Protection » de cette mémoire par des portes

Vanishing gradient résolu (ou presque)



Effet des portes du LSTM sur le problème du gradient qui disparaît, F.-M. Giraud, R. Rincé & H.

Détails théoriques



Cellule Long Short-Term Memory, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

$$\text{Forget gate} \quad F_t = \sigma(W_F \times x_t + U_F \times h_{t-1} + b_F) \in [0; 1]$$

$$\text{Input gate} \quad I_t = \sigma(W_I \times x_t + U_I \times h_{t-1} + b_I) \in [0; 1]$$

$$\text{Output gate} \quad O_t = \sigma(W_O \times x_t + U_O \times h_{t-1} + b_O) \in [0; 1]$$

$$\text{Accumulateur} \quad c_t = F_t \times c_{t-1} + I_t \times \tanh(W_c \times x_t + U_c \times h_{t-1} + b_c)$$

$$\text{Hidden} \quad h_t = O_t \times \tanh(c_t)$$

$$\text{Output} \quad o_t = f(W_o \times h_t + b_o)$$

Réseaux de neurones

Réseaux de neurones récurrents

Variantes

Gated Recurrent Unit – Principe

Similaire au LSTM, mais :

Update gate \approx Input et forget gates du LSTM combinées

Reset gate \approx Influence de la mémoire pendant l'update

→ Moins de paramètres que LSTM, peut être aussi efficace sur certains datasets

Gated Recurrent Unit – Détails théoriques

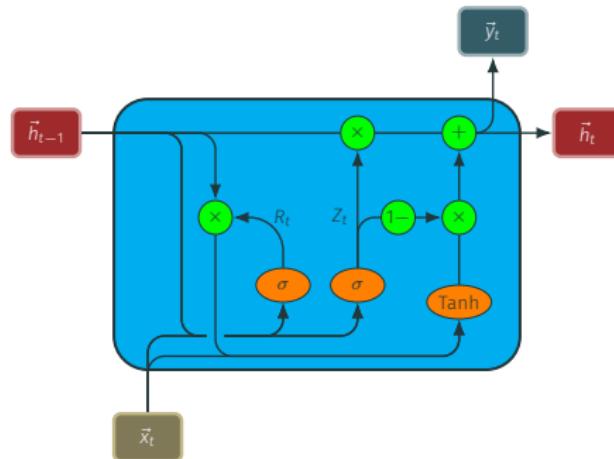


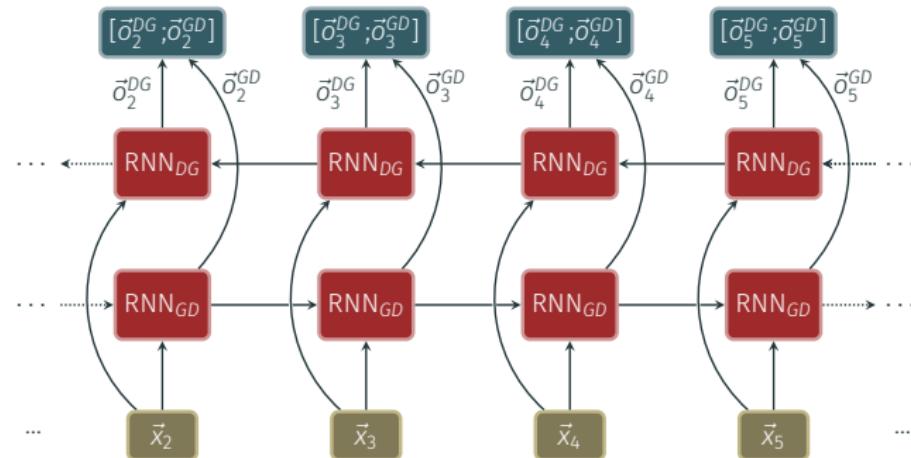
Schéma d'une Gated Recurrent Unit (GRU), F.-M. Giraud, R. Rincé & H. Mougaard, CC-BY-SA-4.0.

$$\text{Update gate } z_t = \sigma(W_z \times x_t + U_z \times h_{t-1} + b_z)$$

$$\text{Reset gate } r_t = \sigma(W_r \times x_t + U_r \times h_{t-1} + b_r)$$

$$\text{Sortie } h_t = (1-z_t) \times h_{t-1} + z_t \times \tanh(W_h \times x_t + r_t \times U_h \times (h_{t-1}) + b_h)$$

Réseaux récurrents bi-directionnels



Réseau récurrent bidirectionnel en vue déroulée, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

- Permet d'apprendre sur des séquences plus longues
- Permet de modéliser des dépendances droite-gauche

D'autres variantes

- Récurrence dans la profondeur
- Skip-connexions (qui sautent des couches)
- Résiduels (qui apprennent la différence à l'input)
- ...

Réseaux de neurones

Réseaux de neurones récurrents

Implémentation avec Keras

Dimension du tensor d'entrée

```
print(f"Forme du corpus de documents : {X_train.shape}")  
print(f"Premier exemple : {X_train[0]}")
```

Forme du corpus de documents : (62405, 150)

Premier exemple :

RNN simple

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))
model.add(layers.SimpleRNN(128))
model.add(layers.Dense(10, activation='softmax'))
model.summary()
```

Layer (type)	Output Shape	Param #
<hr/>		
embedding_1 (Embedding)	(None, None, 64)	64000
<hr/>		
simple_rnn (SimpleRNN)	(None, 128)	24704
<hr/>		
dense_1 (Dense)	(None, 10)	1290
<hr/>		
Total params: 89,994		
Trainable params: 89,994		
Non-trainable params: 0		

LSTM

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))
model.add(layers.LSTM(128))
model.add(layers.Dense(10, activation='softmax'))
model.summary()
```

Layer (type)	Output Shape	Param #
<hr/>		
embedding_2 (Embedding)	(None, None, 64)	64000
<hr/>		
lstm_1 (LSTM)	(None, 128)	98816
<hr/>		
dense_2 (Dense)	(None, 10)	1290
<hr/>		
Total params:	164,106	
Trainable params:	164,106	
Non-trainable params:	0	

GRU

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))
model.add(layers.GRU(128))
model.add(layers.Dense(10, activation='softmax'))
model.summary()
```

Layer (type)	Output Shape	Param #
<hr/>		
embedding_3 (Embedding)	(None, None, 64)	64000
<hr/>		
gru (GRU)	(None, 128)	74112
<hr/>		
dense_3 (Dense)	(None, 10)	1290
<hr/>		
Total params:	139,402	
Trainable params:	139,402	
Non-trainable params:	0	

RNN profond

```
model = tf.keras.Sequential()
model.add(layers.Embedding(input_dim=1000, output_dim=64))
model.add(layers.GRU(256, return_sequences=True))
model.add(layers.SimpleRNN(128))
model.add(layers.Dense(10, activation='softmax'))
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 64)	64000
gru (GRU)	(None, None, 256)	246528
simple_rnn (SimpleRNN)	(None, 128)	49280
dense (Dense)	(None, 10)	1290
<hr/>		
Total params:	361,098	
Trainable params:	361,098	
Non-trainable params:	0	

Avez-vous des questions ?

Instructions

1. [Génération de texte de Voltaire](#)
2. [Implémentation d'un calculateur](#)
3. [Classification de texte](#)

Réseaux de neurones

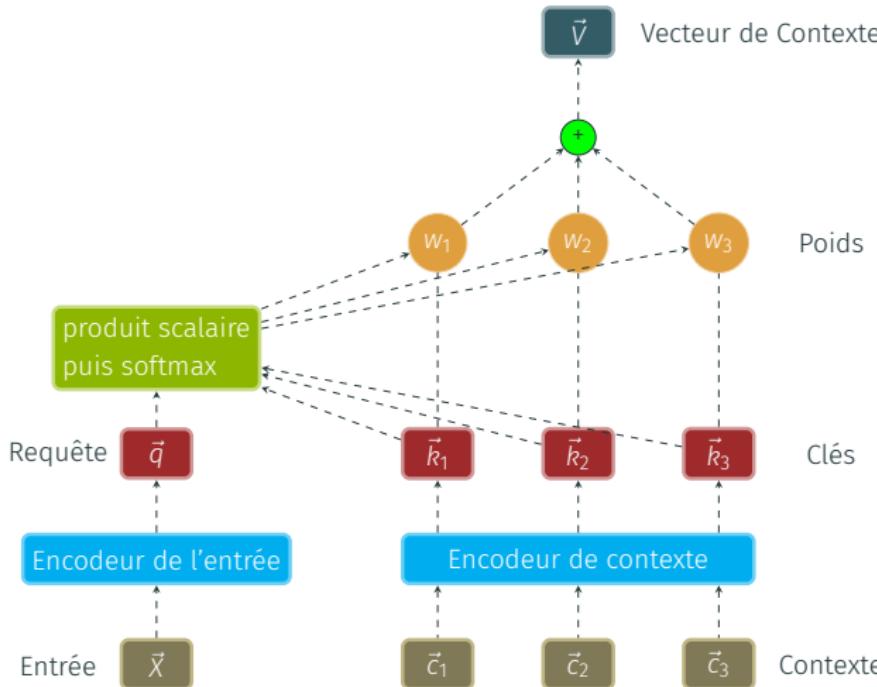
Réseaux transformeurs

Réseaux de neurones

Réseaux transformateurs

Modèles à attention

Modèle à attention



Le mécanisme d'attention, F.-M. Giraud, R. Rincé & H. Mougard, CC-BY-SA-4.0.

Réseaux de neurones

Réseaux transformateurs

Modèle

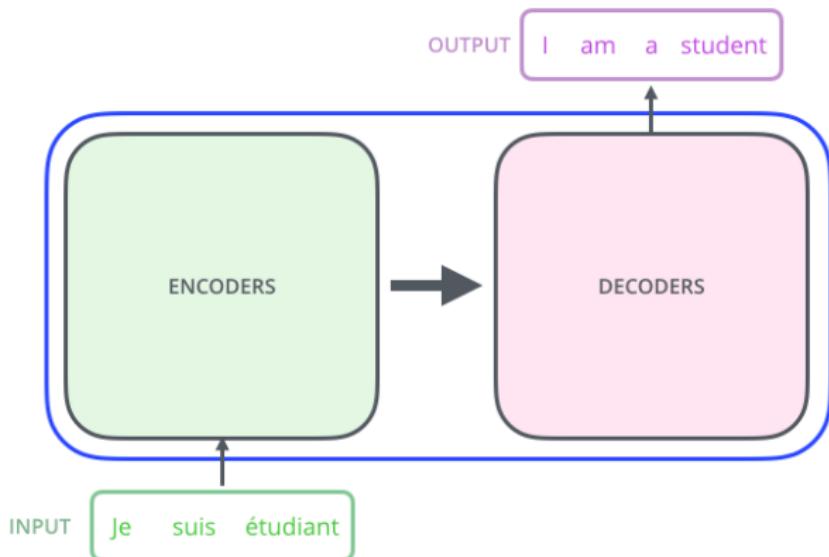
Modèle à attention

Démo modèle à attention

Vue générale

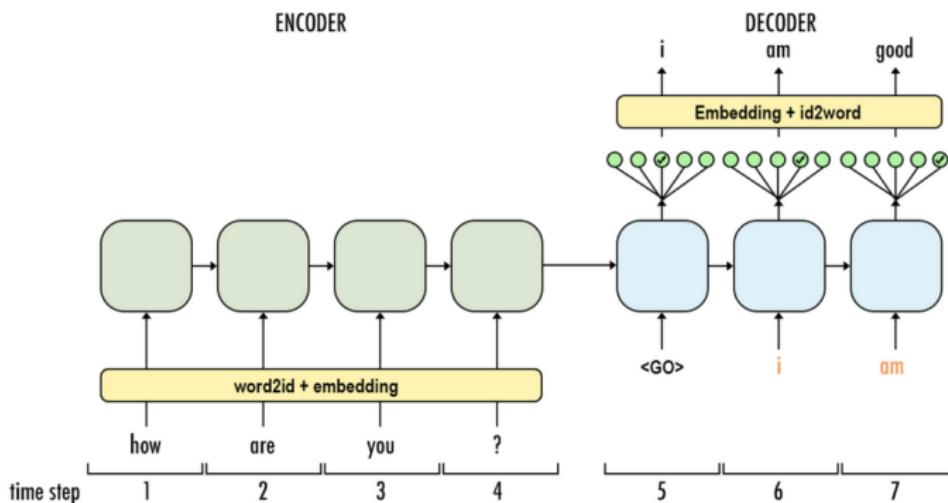


Architecture encodeur-décodeur



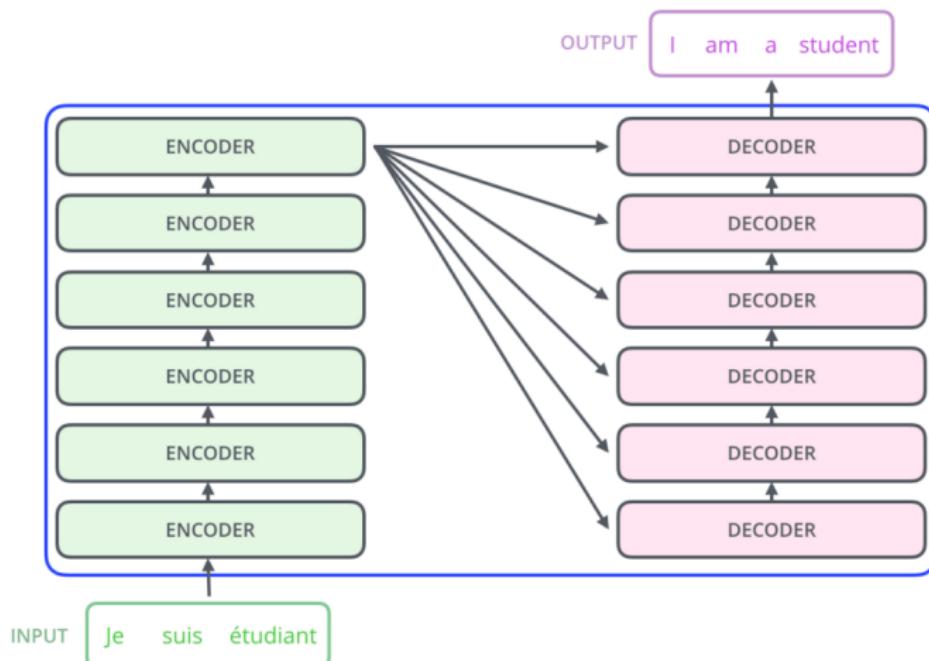
Architecture encodeur-décodeur — Rappel

Rappelez-vous, les encodeurs-décodeurs :

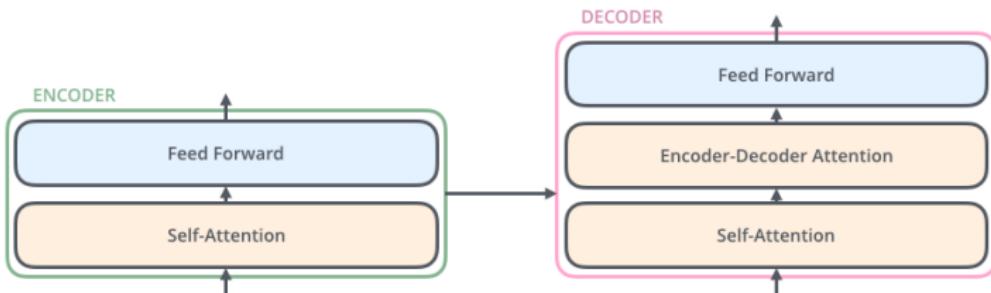


Architecture profonde

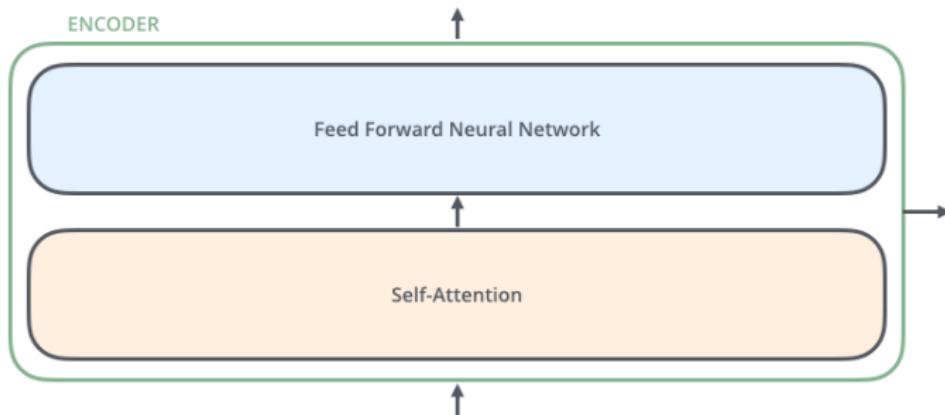
Couches « simples » empilées :



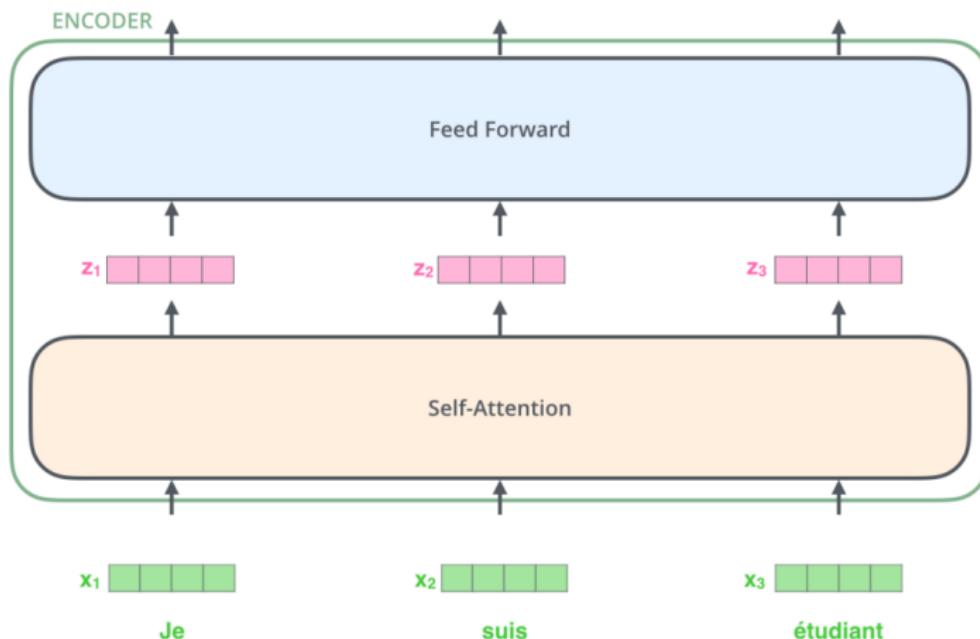
Architecture globale de l'encodeur et du décodeur



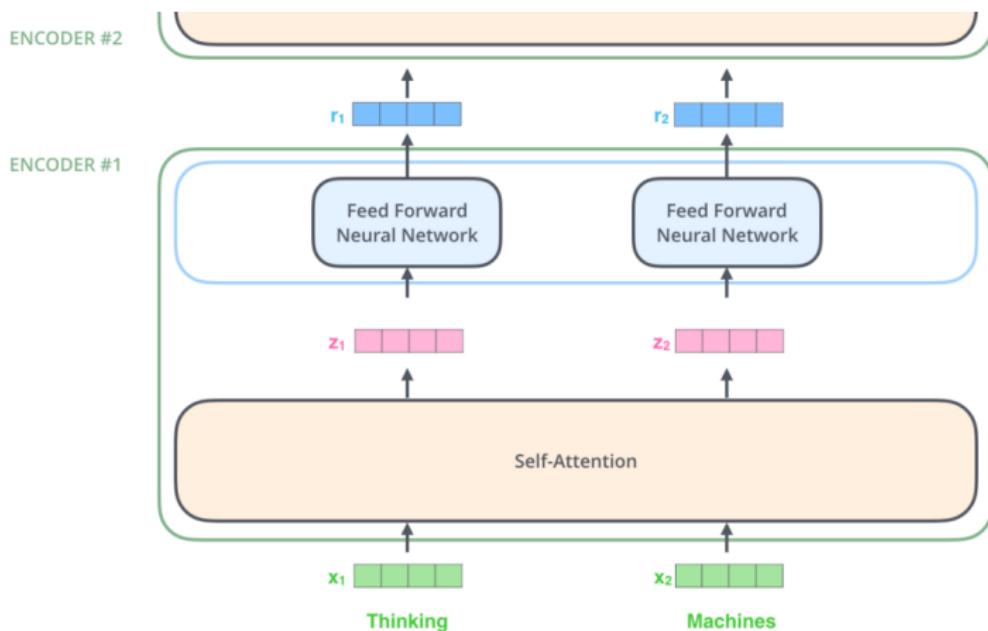
Architecture globale de l'encodeur



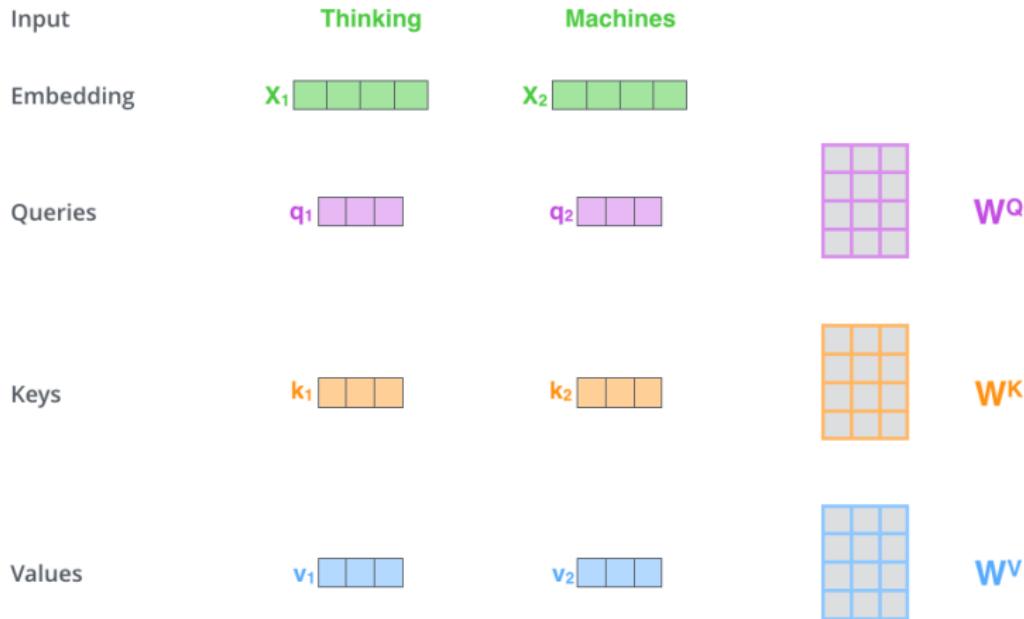
Forme des données en jeu dans l'encodeur



Forme des données en jeu dans l'encodeur



Self-Attention – Éléments utilisés



Self-Attention – Combinaison des éléments

$$\begin{matrix} X & \times & W^Q & = & Q \end{matrix}$$

The diagram shows three matrices: X (green, 3x3), W^Q (grey, 3x3), and Q (purple, 3x3). The multiplication operation \times is placed between X and W^Q , and the equals sign $=$ is placed after Q .

$$\begin{matrix} X & \times & W^K & = & K \end{matrix}$$

The diagram shows three matrices: X (green, 2x5), W^K (orange, 5x5), and K (orange, 2x2). The multiplication $X \times W^K$ results in matrix K .

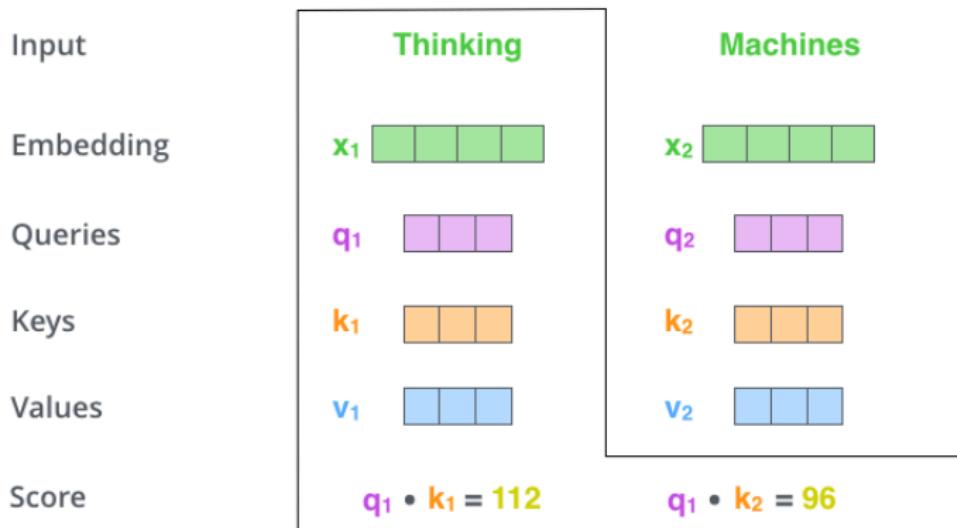
$$\begin{matrix} \text{X} & \times & \text{W}^V & = & \text{V} \\ \begin{matrix} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{matrix} & & \begin{matrix} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{matrix} & & \begin{matrix} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{matrix} \end{matrix}$$

Self-Attention – Obtention du résultat

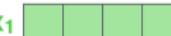
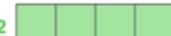
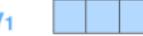
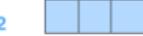
$$\text{softmax} \left(\frac{\begin{array}{|c|c|c|}\hline & \textcolor{purple}{Q} & \textcolor{orange}{K^T} \\ \hline & \begin{array}{|c|c|c|}\hline & \textcolor{purple}{\square} & \textcolor{purple}{\square} \\ \hline & \textcolor{purple}{\square} & \textcolor{purple}{\square} \\ \hline & \textcolor{purple}{\square} & \textcolor{purple}{\square} \\ \hline \end{array} & \times & \begin{array}{|c|c|}\hline & \textcolor{orange}{\square} & \textcolor{orange}{\square} \\ \hline & \textcolor{orange}{\square} & \textcolor{orange}{\square} \\ \hline \end{array} \\ \hline \end{array} \right) \frac{\sqrt{d_k}}{\textcolor{blue}{V}}$$
$$= \begin{array}{|c|c|c|}\hline & \textcolor{pink}{Z} \\ \hline & \begin{array}{|c|c|c|}\hline & \textcolor{pink}{\square} & \textcolor{pink}{\square} \\ \hline & \textcolor{pink}{\square} & \textcolor{pink}{\square} \\ \hline & \textcolor{pink}{\square} & \textcolor{pink}{\square} \\ \hline \end{array} \\ \hline \end{array}$$

Avec le softmax défini ainsi : $\sigma(z_j) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$

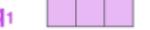
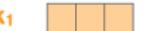
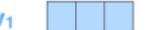
Self-Attention — Run sur 2 mots



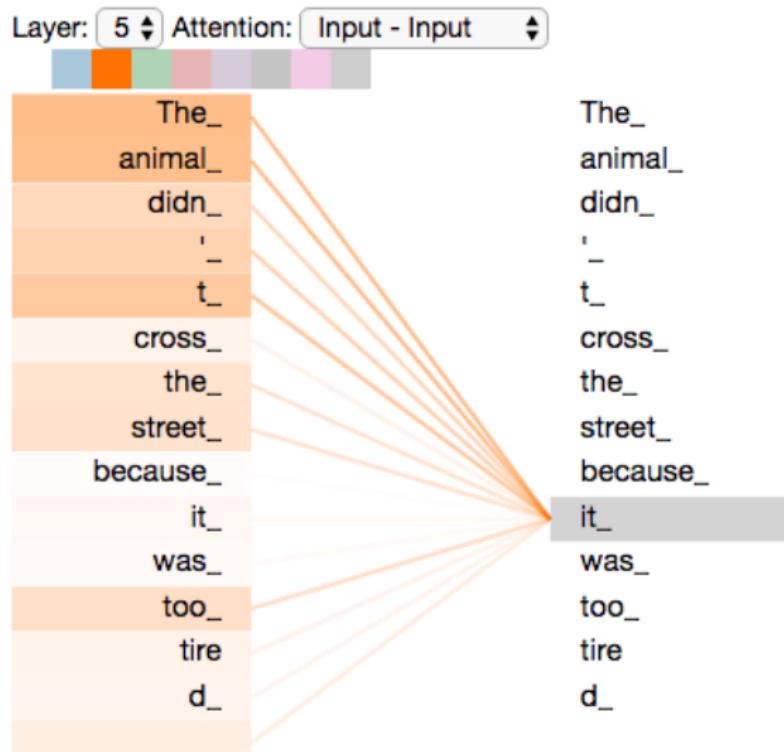
Self-Attention — Run sur 2 mots

Input	Thinking x_1 		Machines x_2 	
Embedding	q_1		q_2	
Queries	k_1		k_2	
Keys	v_1		v_2	
Score	$q_1 \cdot k_1 = 112$		$q_1 \cdot k_2 = 96$	
Divide by $8 (\sqrt{d_k})$	14		12	
Softmax	0.88		0.12	

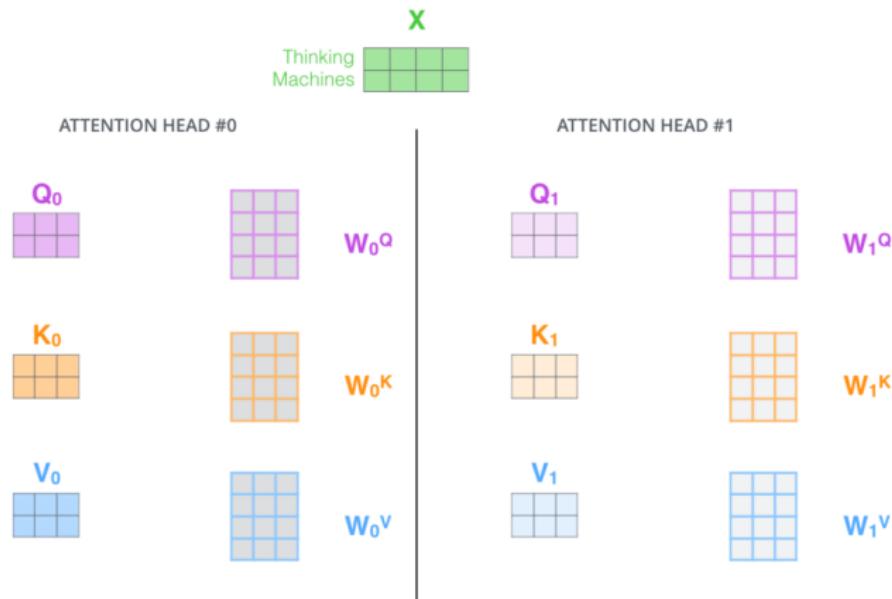
Self-Attention — Run sur 2 mots

Input		
Embedding	x_1	
Queries	q_1	
Keys	k_1	
Values	v_1	
Score	$q_1 \cdot k_1 = 112$	$q_1 \cdot k_2 = 96$
Divide by 8 ($\sqrt{d_k}$)	14	12
Softmax	0.88	0.12
Softmax X Value	v_1	
Sum	z_1	
	z_2	

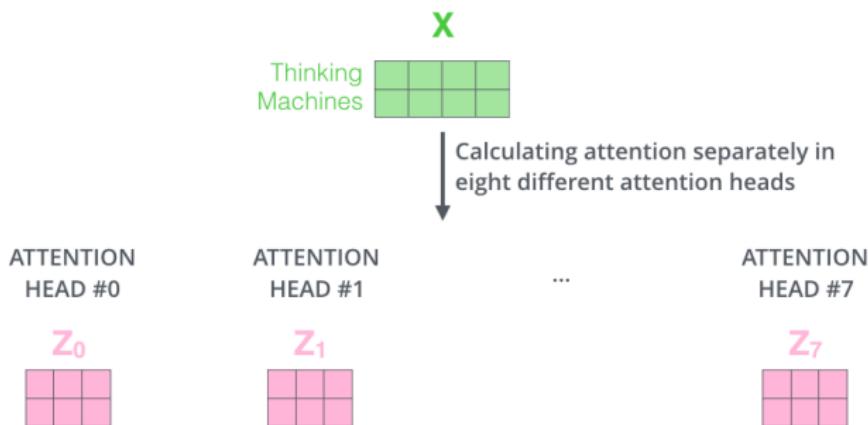
Self-Attention — Visualisation



Attention multi-têtes – Têtes d'attention



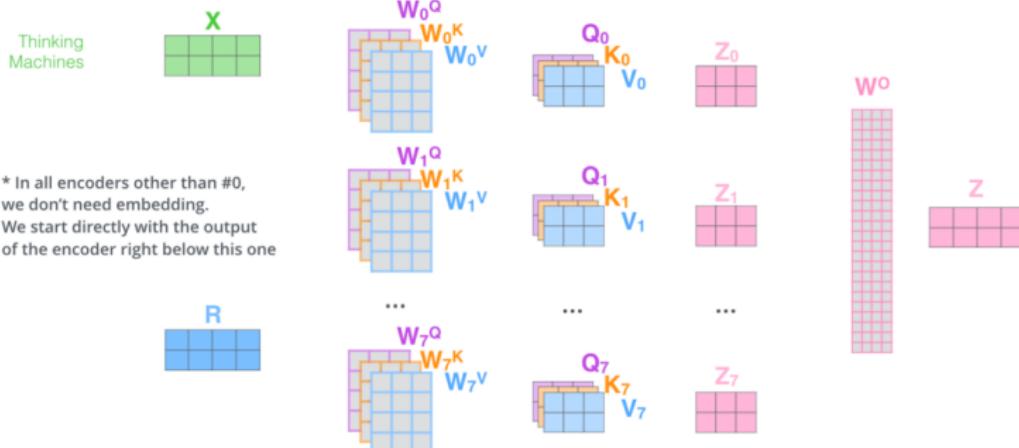
Attention multi-têtes – Combinaison



Encodeur – Résumé

Un modèle sans récurrence, uniquement des sommes pondérées.

- 1) This is our input sentence* 2) We embed each word*
- 3) Split into 8 heads. We multiply X or R with weight matrices
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^o to produce the output of the layer

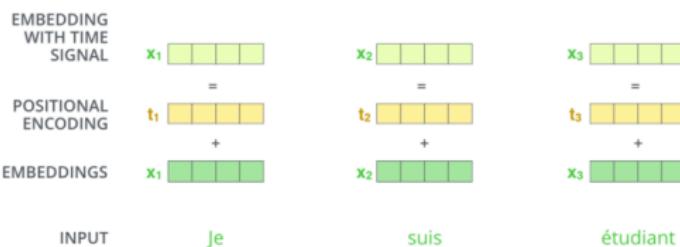
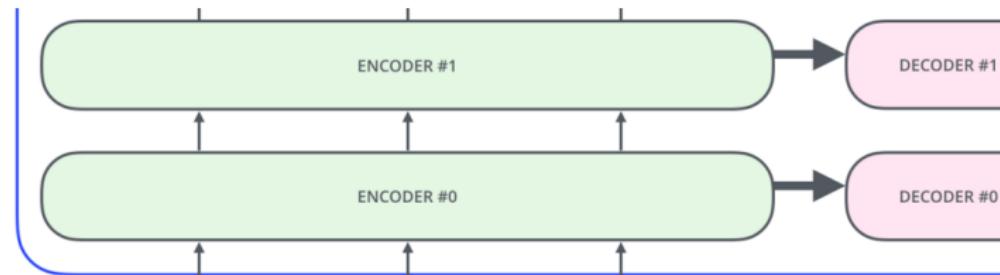


Encodeur – Gestion de la position

Pour l'instant, modèle invariant à l'ordre.

⇒ Nécessité d'encoder la position (encodage positionnel).

Encodeur – Encodage positionnel – Fonction hardcodée



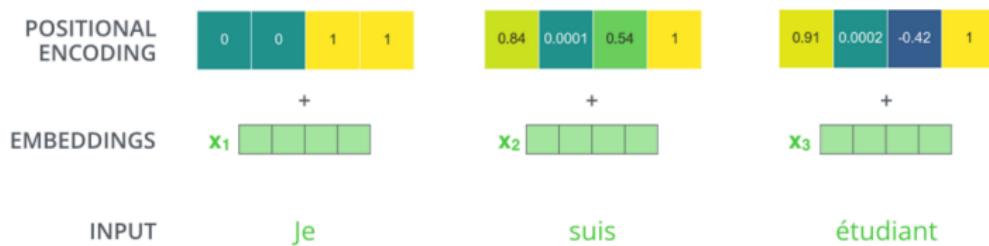
$$PE_{pos,2i} = \sin\left(pos/10000^{2i/d_{model}}\right)$$

$$PE_{pos,2i+1} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

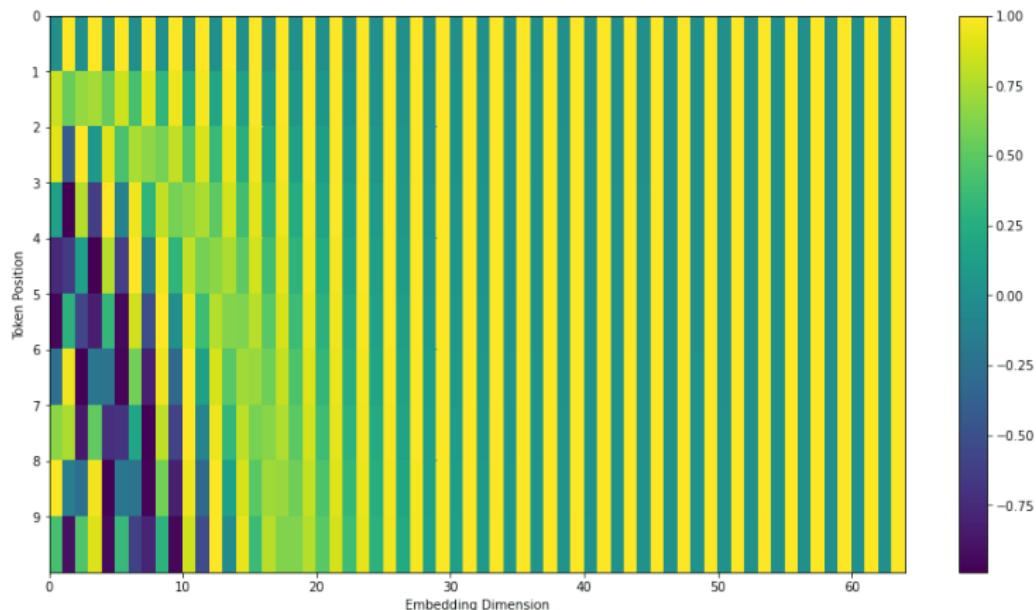
Encodeur – Encodage positionnel – Alternative

Il est aussi possible d'apprendre les embeddings de position, sans détérioration des performances.

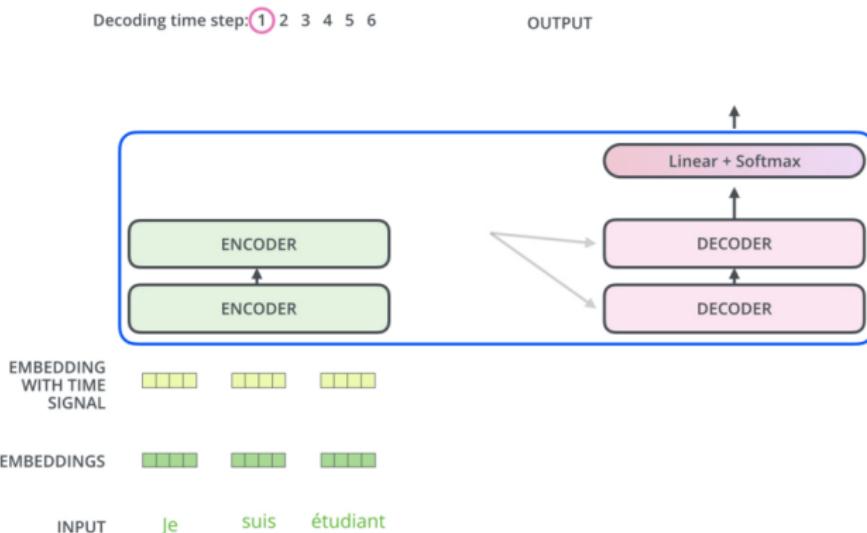
Encodeur – Encodage positionnel – Exemple



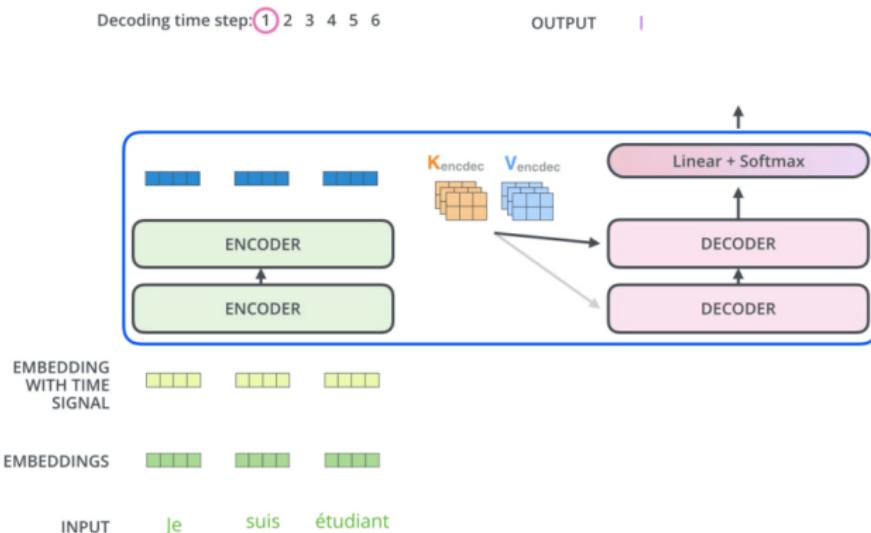
Encodeur – Encodage positionnel – Visualisation



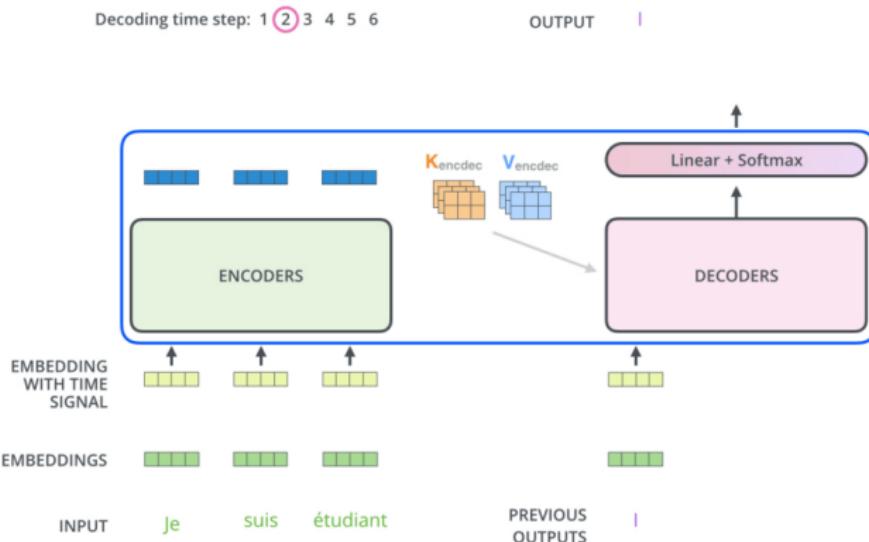
Décodeur – Exemple de décodage



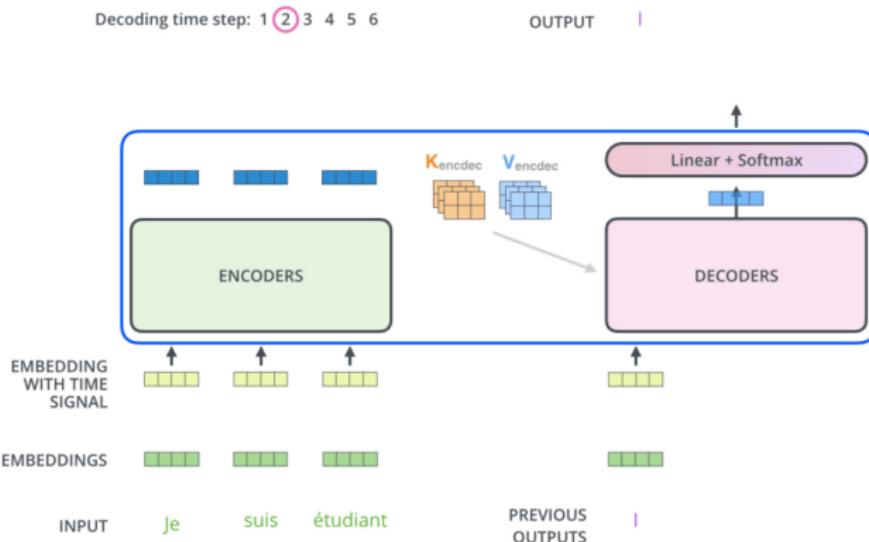
Décodeur – Exemple de décodage



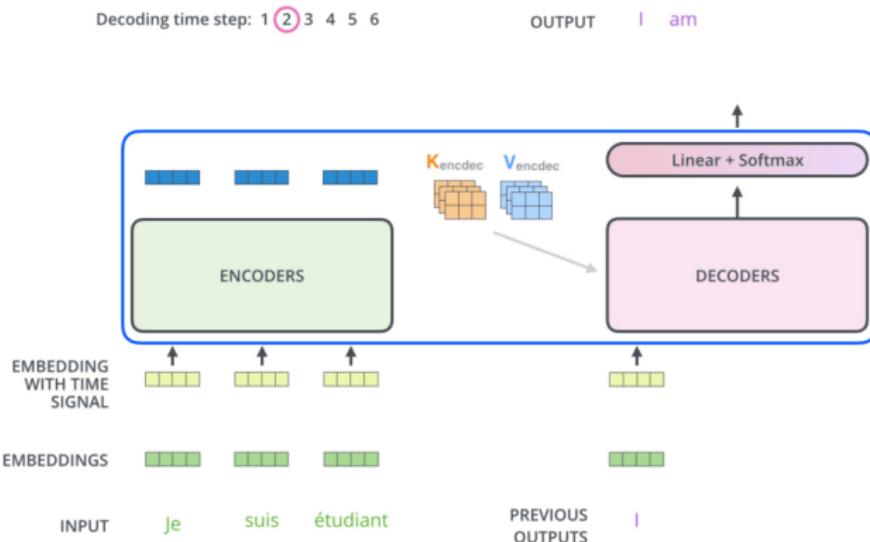
Décodeur – Exemple de décodage



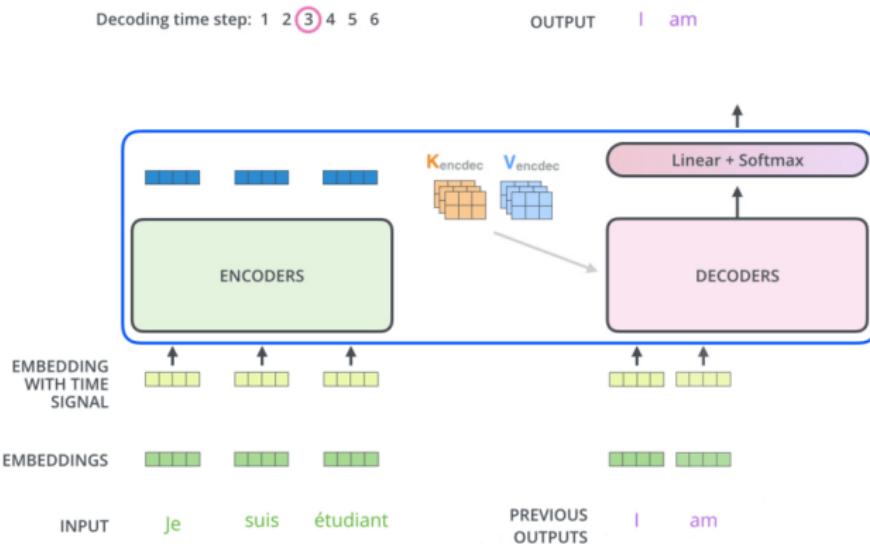
Décodeur – Exemple de décodage



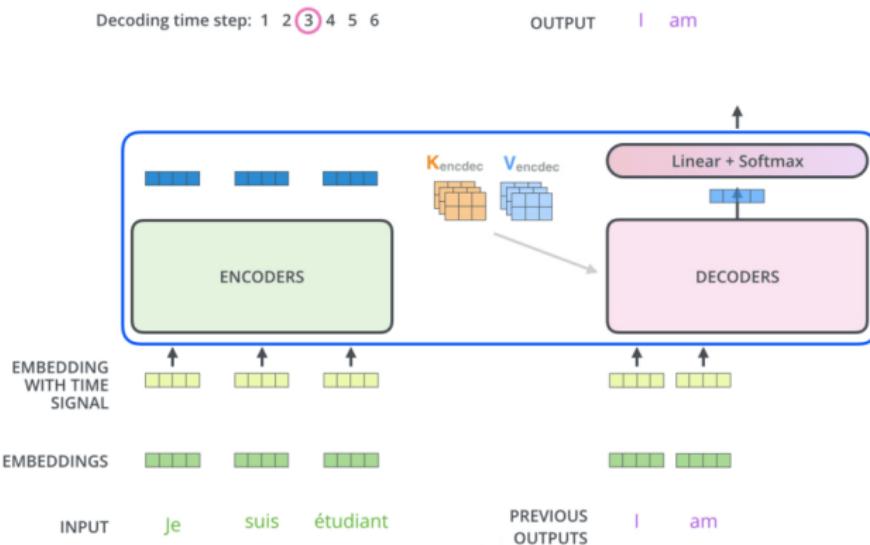
Décodeur – Exemple de décodage



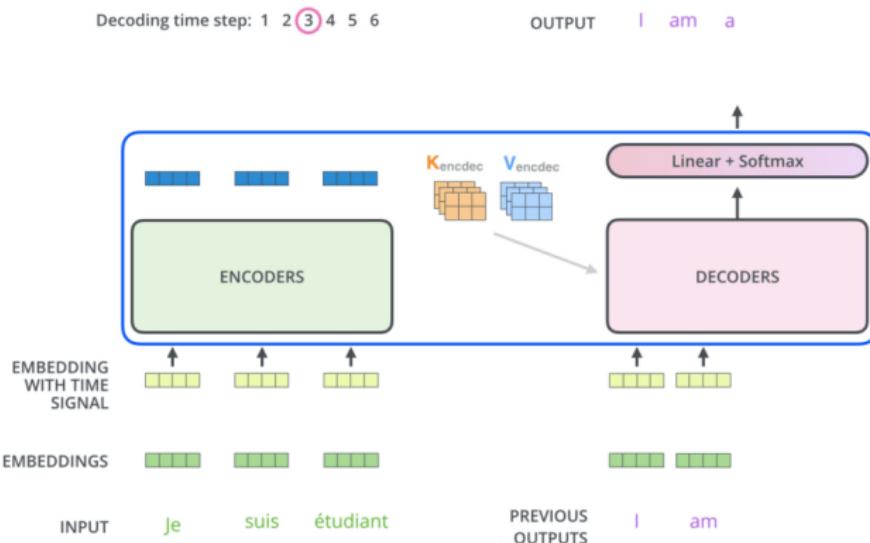
Décodeur – Exemple de décodage



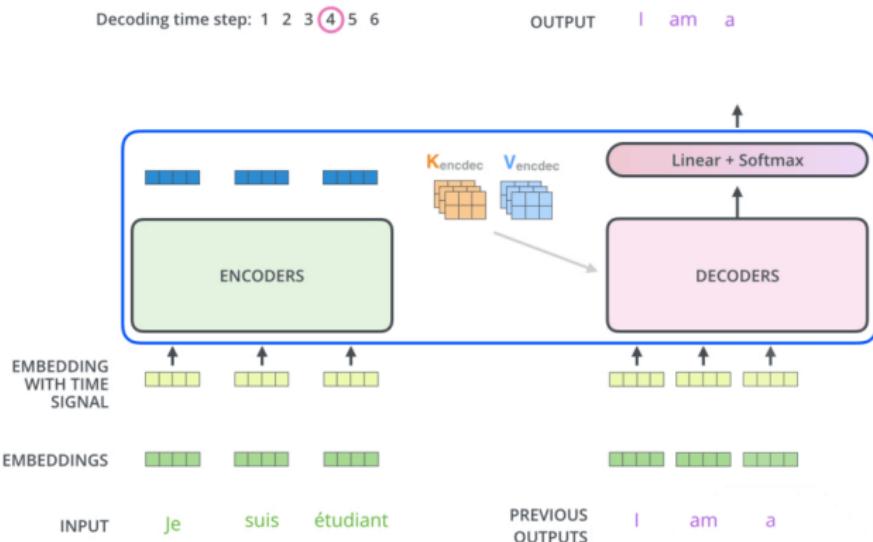
Décodeur – Exemple de décodage



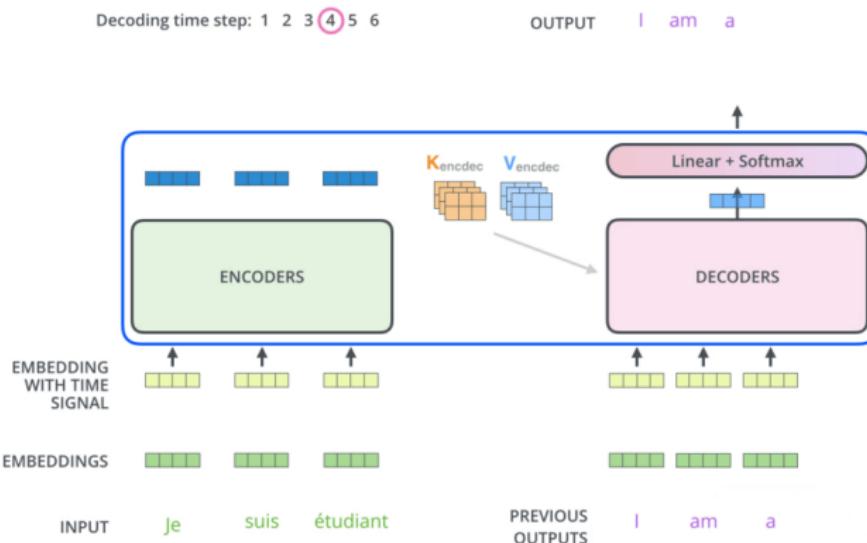
Décodeur – Exemple de décodage



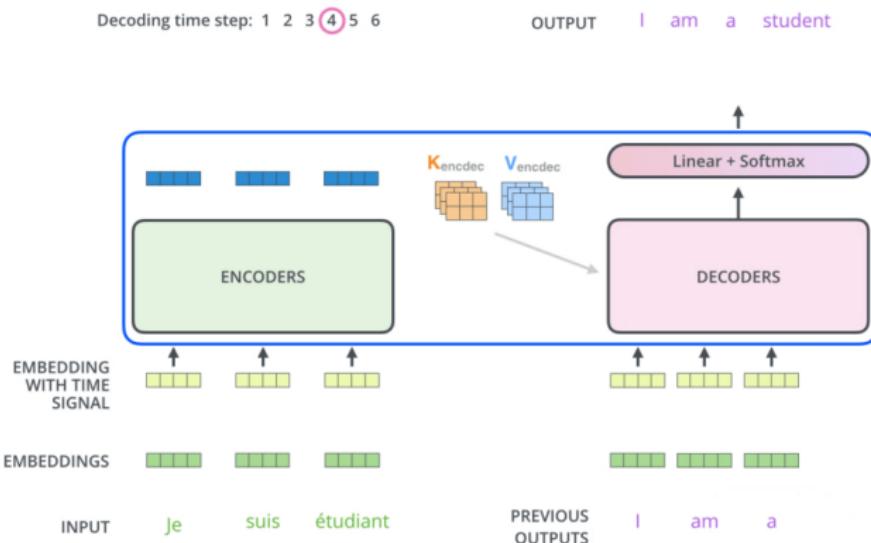
Décodeur – Exemple de décodage



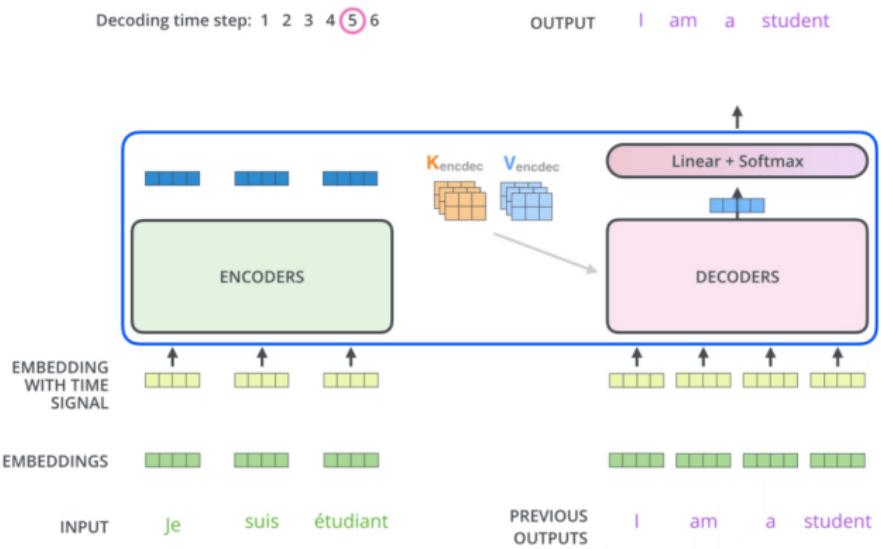
Décodeur – Exemple de décodage



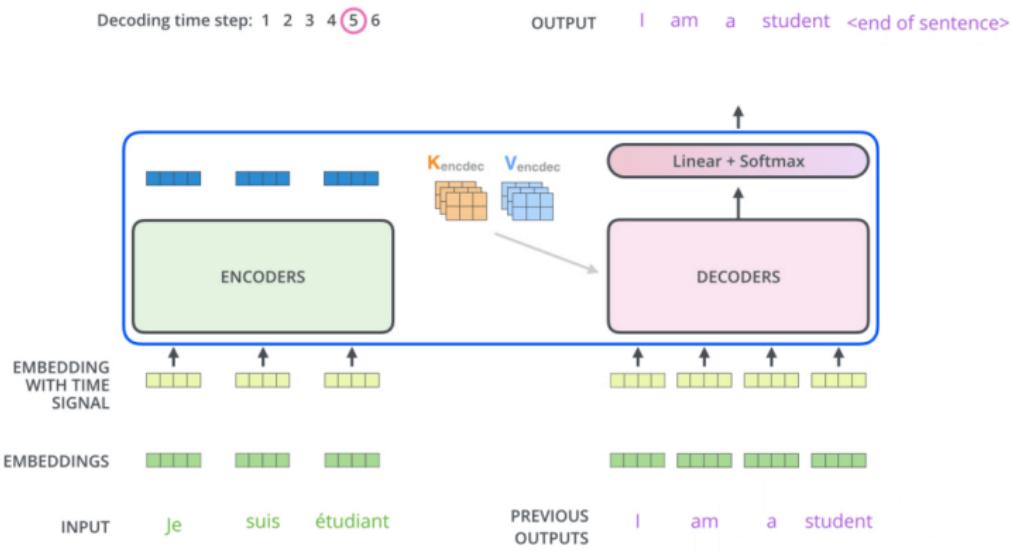
Décodeur – Exemple de décodage



Décodeur – Exemple de décodage



Décodeur – Exemple de décodage



Transformer Network

Les illustrations sont tirées du billet de blog de Jay Alammar :
<http://jalammar.github.io/illustrated-transformer/>

Avez-vous des questions ?

Travaux pratiques

Instructions

Ressource complémentaires

[Modèles multi-tâches sur Coursera](#)

Principaux outils

Principaux outils

Introduction

Intérêt de l'outillage

Le ML est à la croisée des mathématiques et de l'ingénierie :

- Gérer des expérimentations (modèles, données)
- Stocker des quantités importantes de données
- Pouvoir calculer en parallèle
- Mettre en production
- ...

→ Il est très important de s'outiller !

Principaux outils

Machine Learning & Deep Learning

Librairies fondamentales

Utilisées dans tous les frameworks Python :

- [pandas](#)
- [NumPy](#)

Machine Learning tabulaire

- [scikit-learn](#)
- [XGBoost](#)

Deep Learning général

- [TensorFlow](#)
- [Keras](#)
- [PyTorch](#)
- [Apache MXNet](#)

Deep Learning pour le texte

- [AllenNLP](#)
- [spaCy](#)

Deep Learning pour les images

Incontournable : [OpenCV](#)

Deep Learning pour les graphes

- [DGL](#)
- [PyG](#)

Principaux outils

Environnement logiciel

Introduction

Contrôler son environnement logiciel pour :

- Rendre son environnement de développement reproductible
- Contrôler les dépendances pour la prod
- Déployer son environnement facilement sur différents clouds

virtualenv

virtualenv permet d'isoler un environnement Python :

- N'intéragit pas avec l'environnement système
- Permet la cohabitation de plusieurs environnements incompatibles
- Plus rapide et natif que les solutions basées sur les conteneurs
- Copie d'une distribution Python de base + customisation

Solutions basées sur virtualenv

pip + setup.py Définition traditionnelle d'une librairie Python

pip + requirements.txt Liste de dépendances

pip + pip-compile Lister les dépendances puis les geler pour la stabilité

Pipenv Définition moderne d'une **application** Python (pas librairie)

poetry Définition moderne d'application ou librairie

Solutions basées sur les conteneurs

Docker : conteneurs qui embarquent un système d'exploitation en plus de l'environnement Python :

- Permet de contrôler les librairies natives en plus des librairies Python
- Plus grande robustesse si le logiciel s'exécute sur plusieurs OS
- Plus lourd à mettre en place que `virtualenv`
- Plus adapté pour la prod (déploiement facile par Kubernetes)

Principaux outils

Ingénierie

Outils d'aide à l'ingénierie

Buts :

- Collaborer
- Contrôler le source code, **les modèles & les données**
- Déployer facilement

Solutions

Open source :

- [mlflow](#)
- [dvc](#)
- [Kubeflow](#)

Propriétaires :

- [Neptune](#)
- [Weights & Biases](#)
- [comet](#)

Principaux outils

Big Data

Outils pour le Big Data

Buts :

- Stocker efficacement les données et modèles
- Pouvoir traiter la masse importante de données
- Exprimer les algorithmes de ML/DL de manière distribuée

Solutions cloud

- Cloud AWS
- Google Cloud Platform
- Microsoft Azure

Solutions cloud – Intérêts

- Tous les services sont intégrés
- Puissance de calcul ajustable
- APIs intéressantes accessibles (reconnaissance d'images, de texte, ...)

Solutions cloud – Problèmes

- Coût important
- Vendor lock-in
- Confidentialité des données

Libraries Big Data

Librairies de calcul & stockage distribués :

- Apache Spark (+ MLlib)
- Dask (+ Dask-ML)
- Ray (+ Ray Tune, + Ray SGD, + RLLib)
- Apache Hadoop (récemment de plus en plus délaissé pour Spark)

Librairies de déploiement distribué :

- Kubernetes
- Terraform

Principaux outils

APIs

Introduction

Chaque grand acteur a son API :

- Permet de prototyper très rapidement
- Bonne intégration au reste des plate-formes
- Potentiellement cher

Types d'API proposées

- Traitement images, vidéo
- Traitement de texte, analyse de sentiment, traduction, détection d'entités, ...
- Speech to text, text to speech
- Chatbots
- AutoML, inférence sur séries temporelles, recommandations, ...
- Prévention de fraude

Principaux outils

Cloud computing

Cloud Computing



Logo AWS, Amazon Web Services., Apache License 2.0.



Logo Microsoft Azure, Microsoft, Domaine public.



Logo Google Cloud, Google, domaine public.



Logo OVH, OVH, CC-BY-SA-4.0.



ORSYS · Deep Learning Par la Pratique · CC BY-SA 4.0

Logo IBM, Paul Rand, Domaine public.



Logo Citrix, Citrix Systems, Domaine public.

Principaux outils

S'informer

Trouver des papiers scientifiques

- [Google Scholar](#)
- [Semantic Scholar](#)
- [arXiv](#)
- [arXiv Sanity Preserver](#)

Rester à jour

Bien configurer des alertes si nécessaire

Principaux outils

Kaggle

Kaggle

Site d'hébergement de compétitions de Machine Learning.



Logo Kaggle, Kaggle, Domaine public.

Avez-vous des questions ?

Hugo Mougard

hugo@mougard.fr

+33 6 37 63 82 71