IA symbolique et neuronale

Bruno Bachimont
Costech, université de technologie de Compiègne
Alliance Sorbonne Université

Sommaire

- Les IA : caractérisations
- L'IA symbolique
  - Principes
  - Impact sur le traitement documentaire
- L'IA neuronale
  - Principes
  - Impact sur le traitement documentaire

L'IA

Une définition générale

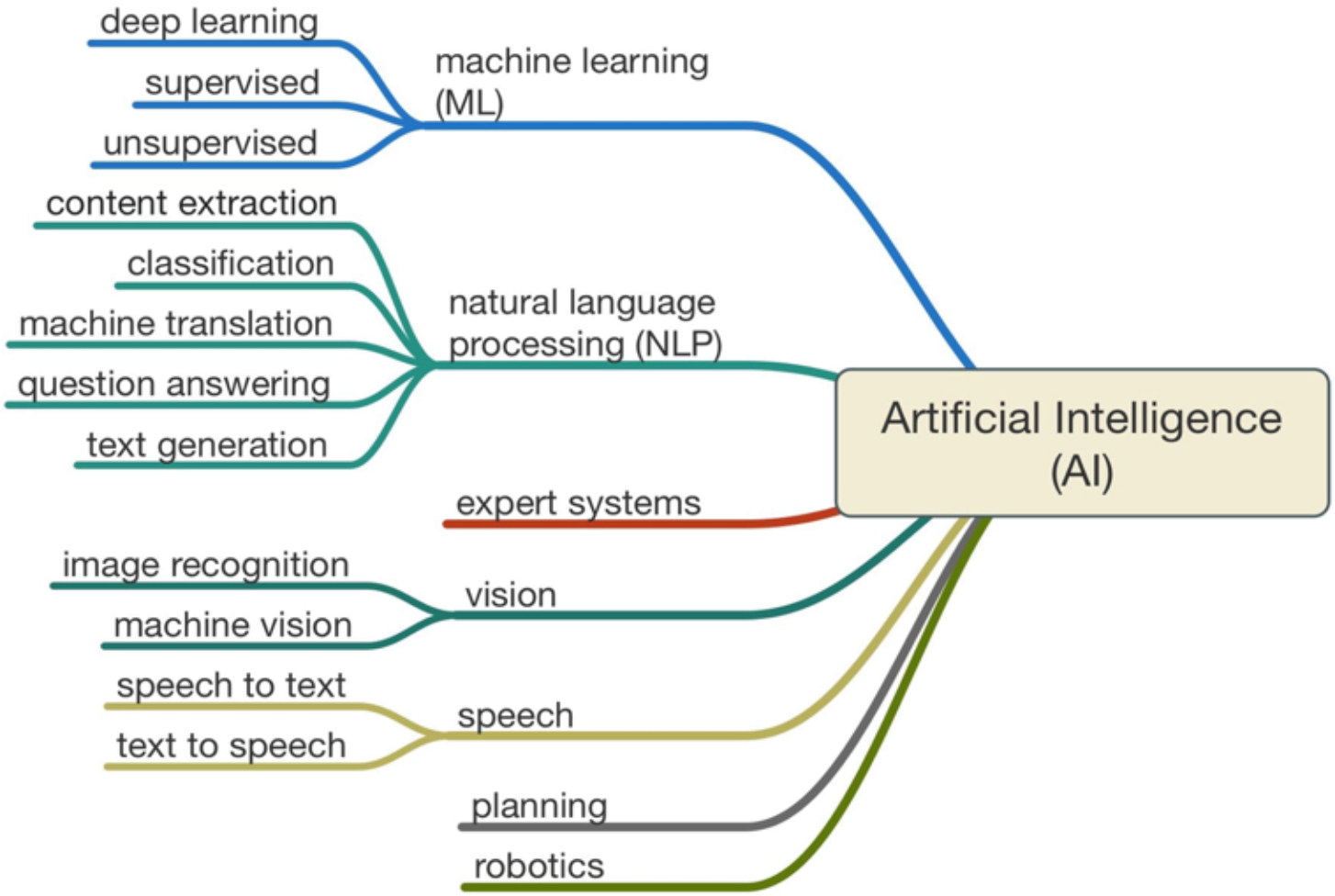
- Faire faire à la machine des tâches qui requiert de l'intelligence quand nous les faisons :
  - Cela ne signifie pas que la machine doive être intelligente pour le faire ni qu'elle le soit si elle le fait.
  - Exemples : diagnostic médical, planification, jouer aux échecs, etc.
- Plusieurs niveaux:
  - IA forte : les systèmes d'IA peuvent être conscient, avoir des états mentaux, avoir des sentiments ;
  - IA faible : les systèmes simulent ou effectuent par d'autres moyens des fonctions cognitives.
- Quelques phantasmes :
  - La singularité ;
  - La « guerre des intelligences » : l'enjeu est plutôt la guerre des bêtises ou des stupidités, la naturelle étant avantageusement remplacée par l'artificielle.

Quelques définitions de l''A (Conseil de l'Europe)

- IA: « Les systèmes d'IA sont des modèles algorithmiques qui accomplissent dans le monde des fonctions cognitives et perceptives autrefois réservées aux êtres humains, lesquels ont la faculté de penser, juger et raisonner ».

- Algorithme: Processus de calcul ou ensemble de règles qui sont exécutés pour résoudre un problème. Les algorithmes complexes sont généralement exécutés par des ordinateurs, mais un humain peut également suivre un processus algorithmique, par exemple faire une division ou une recherche dans le dictionnaire.
  - D. Leslie, C. Burr, M. Aitken, J. Cowls, M. Katell et M. Briggs, Intelligence artificielle, droits de l'homme, démocratie et État de droit. Guide introductif, Conseil de l'Europe, 2021. p.28

Plusieurs types d'IA - Classification par domaines



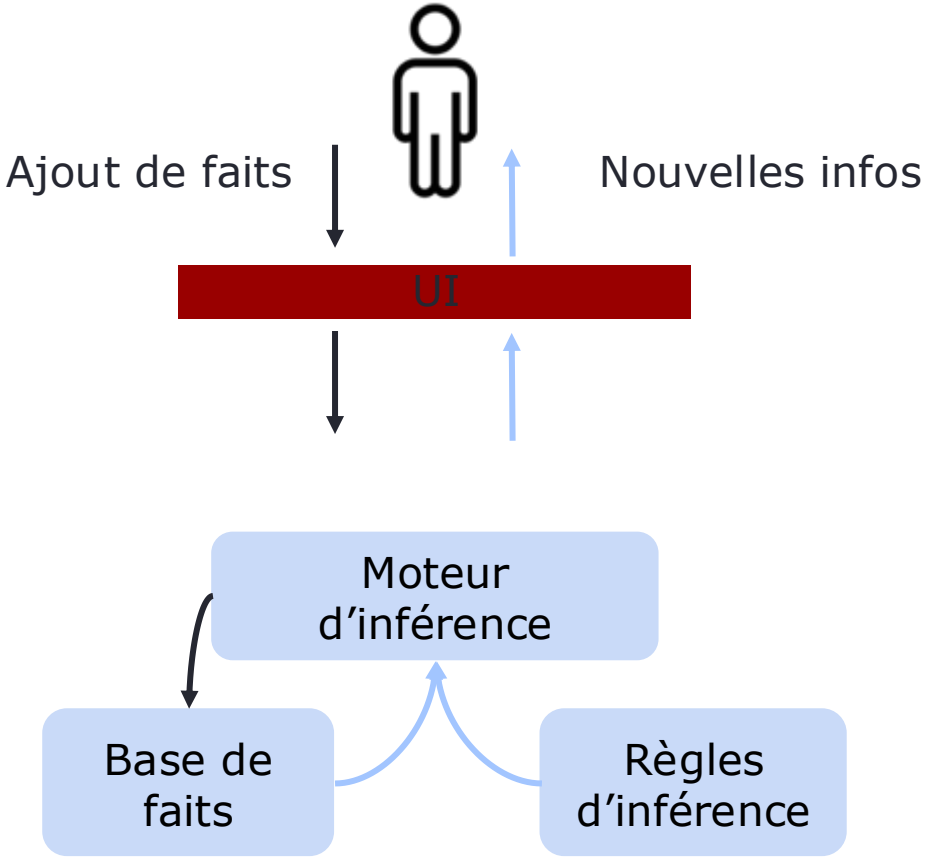
L'IA symbolique

Généralités

IA Symbolique : une définition

- Méthodes d'IA sur des connaissances explicites, souvent exprimées en langue naturelle, que l'on veut formaliser pour qu'elles soient exploitables par la machine.

IA Symbolique - Système expert



Exemple

Fait :
Age = 15

Règle :
SI age ≥ 18 ALORS statut=majeur

Info :
NON majeur

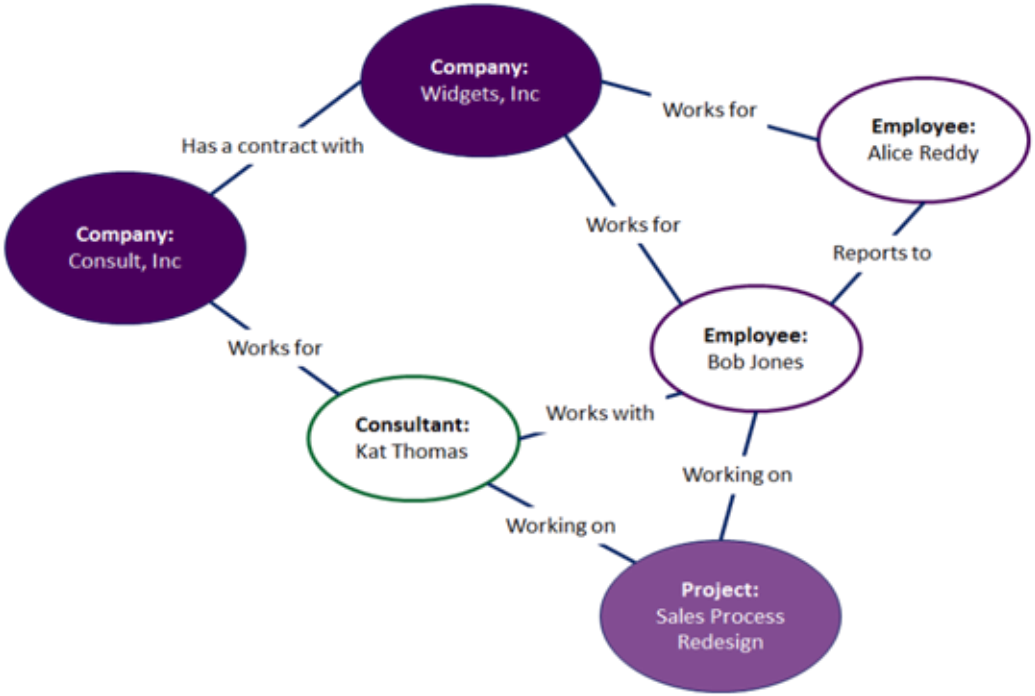
Utilisation

Formaliser des connaissances dans un domaine et les réexploiter

Application

Réalisation de diagnostics

IA Symbolique - Ontologies



Utilisation

Parcourir et interroger le graphe pour une question donnée

Application

Réaliser une taxonomie, un organigramme, un réseau de relations et en extraire des informations

IA Symbolique

Avantages

- Explicables
- Interprétables
- Souvent optimales
- Adaptées aux domaines où l'on a peu de données

Inconvénients

- Difficilement extensibles
- Restreintes au domaine ciblé
- Très sensibles aux données erronées ou mal formées

IA symbolique

Et le traitement documentaire

Le principe de l'IA symbolique

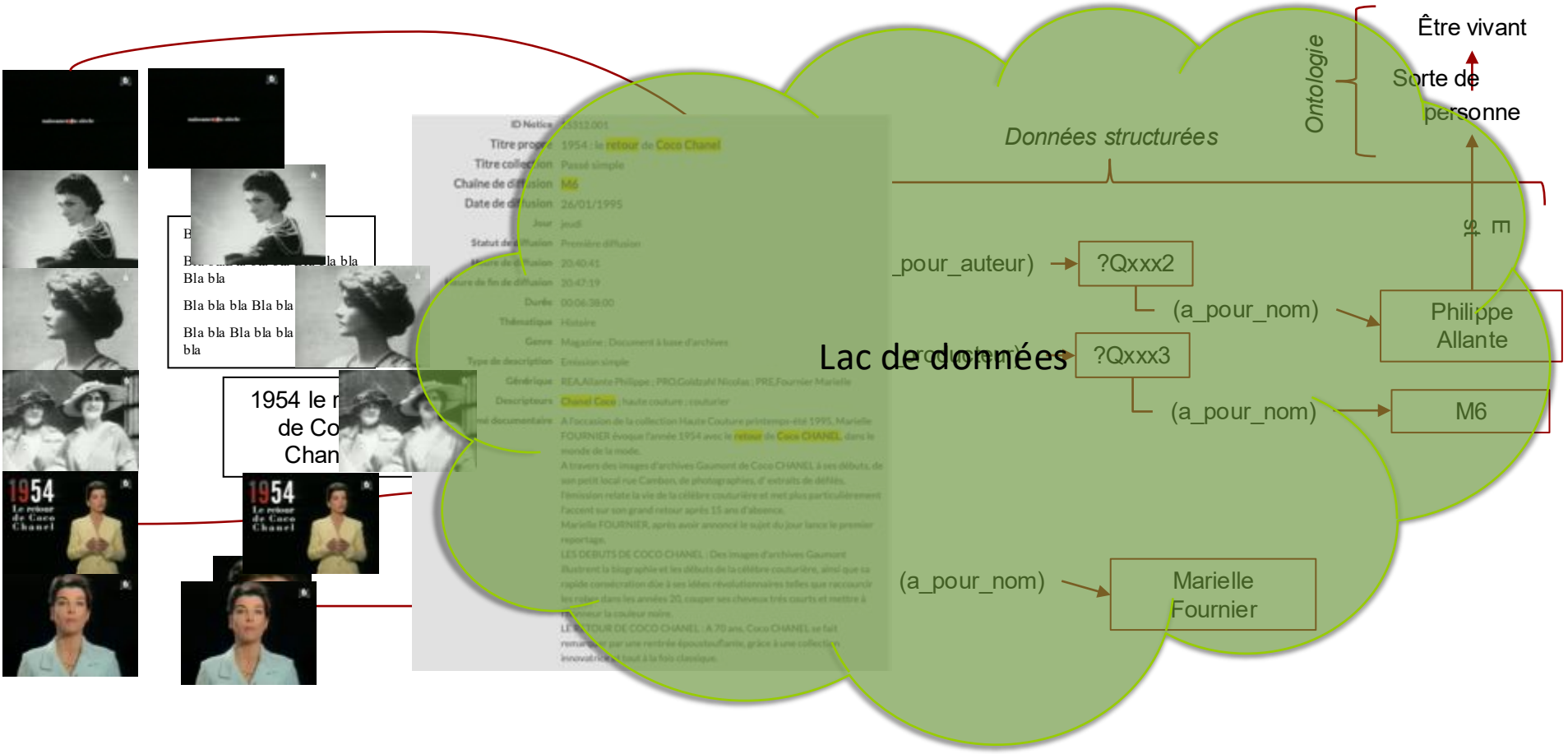
- On dispose de faits décrivant le monde :
  - Ces faits sont des ***données***
- On dispose de règles pour manipuler ces données et en tirer des inférences :
  - Ces règles sont des ***connaissances*** exploitées par les outils formels de raisonnement
- Depuis les années 2000, on distingue :
  - Les outils et formalismes pour exprimer les données et les faits et les règle pour les manipuler et en tirer les inférences :
    - Les ontologies
      - Les faits sont des propriétés ou relations déclarées dans l'ontologie : Paris est une ville, Paris est en France
  - Les outils et formalismes pour exploiter les inférences :
    - Ce sont les raisonneurs.
      - On déduit des faits précédents que Paris est une ville en France.
  - Les raisonneurs sont génériques, donc l'effort est mis sur les ontologies et les connaissances (faits + règles)
  - Une mise en œuvre bien connue :
    - Le Web sémantique appelé aussi Web des données (Web of data).

Une donnée est :

- Une assertion formatée et formalisée minimale :
  - Assertion :
    - C'est une affirmation à propos du monde, l'expression dans un langage d'un fait du monde ;
  - Formatée :
    - Elle respecte une syntaxe dont la machine sait faire quelque chose ;
  - Formalisée :
    - Elle respecte une syntaxe formelle lui donnant une sémantique, une signification associée ;
  - Minimale :
    - Ses parties ne sont pas des données.
- Exemples :
  - un triplet RDF
    - [La Joconde]-(a_pour_auteur)->[Léo]
  - Une cellule d'un tableau Excel
  - La ligne d'un fichier en CSV
  - Une série de tokens (caractères, pixels, etc) dans un document :
    - C'est l'affirmation de leur présence dans un contenu
    - C'est le contenu comme fait, comme la somme des faits / tokens dans lesquels on le décompose et l'analyse.

Les URI

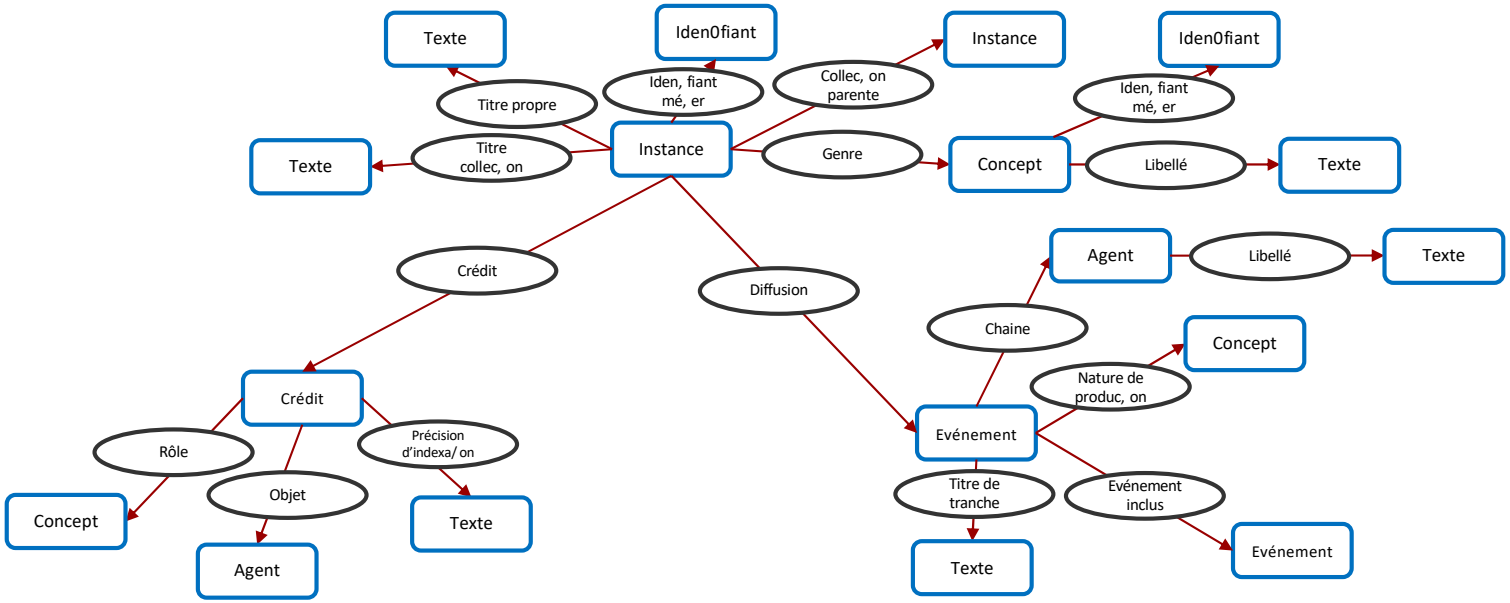
- Avoir des données ne suffit pas, il faut savoir de quoi on parle :
  - Les données sont des faits à propos d'entités ;
  - Pour savoir qu'on parle d'une chose donnée, ou que des faits renvoient à une même entité, on définit un identifiant unique des entités dont on peut/veut parler :
    - Les Uniform Resource Identifer (URI)
- Tout peut avoir une URI, que ce soit numérique, physique ou abstrait :
  - La ville de Paris a URI
  - Vous pouvez en avoir une, et moi aussi ;
  - Un document numérique ou un triplet RDF peuvent en voir une aussi.
- En résumé :
  - On sait de quoi on parle : les URI
  - On exprime ce qu'on veut en dire : les données
  - On infère ce qu'on peut en déduire : les règles et les inférences.



IA statistique (extraction / reconnaissance)

IA Symbolique (inférence / raisonnement)

Il faut un modèle cible….



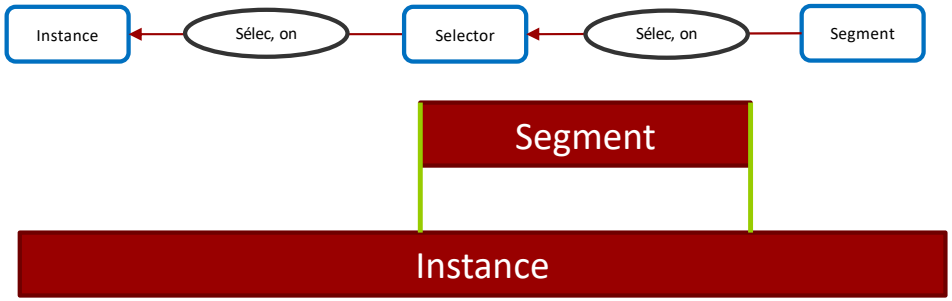
Pour repartir des notices

Identifiant de la notice extrait :	I05007967
Titre de l'extrait :	Jean Rouch sur le racisme
Identifiant de la notice integrale :	CPF86614340
Titre propre de l'intégrale :	Le racisme : 1ère partie
Titre collection de l'intégrale :	Faire Face
Société de progr. de l'intégrale :	RTF (Radio Télévision Française)
Canal de diffusion de l'intégrale :	1ere chaine
Date de diffusion de l'intégrale :	11/09/1961 00:00:00
Théque de l'intégrale :	CP (Vidéothèque production)
Durée de l'extrait :	00:13:44
Producteurs (Aff.) :	Société de production - Radiodiffusion Télévision Française (RTF) - Paris - 1961
Genre :	Documentaire ; Interview entretien ;
Thematique :	Sciences ; Société ;
Générique (Aff. Lig.) :	PAR Rouch, Jean ;
Descripteurs (Aff. Lig.) :	DET: racisme ; DET: société ; DET: interview ; DET: cinéma ; DET: anthropologie ; DET: Schweitzer, Albert ; DET: sexualité ;
Description de l'extrait :	L'ethnologue cinéaste Jean Rouch explique en quoi le racisme n'est pas fondé scientifiquement. Les caractères raciaux utilisés en anthropologie physique ; le métissage généralisé de l'humanité ; exemple d'un étudiant africain génie des mathématiques. Il donne son avis sur l'évolution du racisme, causé par le maintien de la suprématie d'une race sur une autre. Un extrait de son film "La pyramide humaine" ponctue ses propos. La survivance de stéréotypes racistes, dans tous les groupes humains ; le contre-exemple du docteur Albert Schweitzer, incapable selon Jean Rouch de s'intéresser à la culture où il vit (les Bantous) ; l'amitié de Jean Rouch avec ses amis noirs, fondés sur l'honnêteté. A partir d'extraits de "Chroniques d'un été", Jean Rouch appelle au métissage généralisé, citant Léopold Sedar Senghor. Il regrette le "racisme sexuel".
Corpus (Aff.) :	Corpus: Afrique et Europe : entre deux cultures - (Corpus INA > PERSONNALITE > ROUCH JEAN > Propos > Afrique et Europe : entre deux cultures)
Document dévolu INA :	Dévolu INA

Date de création :	07/01/2005
Date de modification :	30/09/2022
Documentaliste :	GER
Dernier intervenant :	EDC
Type de l'extrait :	Extrait Thématique
Type de fonds de l'intégrale :	Production
Couleur / NB :	NOIR ET BLANC
Muet/Sonore :	Document sonore
Indexation (info.) :	Atteint: Oui Validé: O Date: 21/08/2006

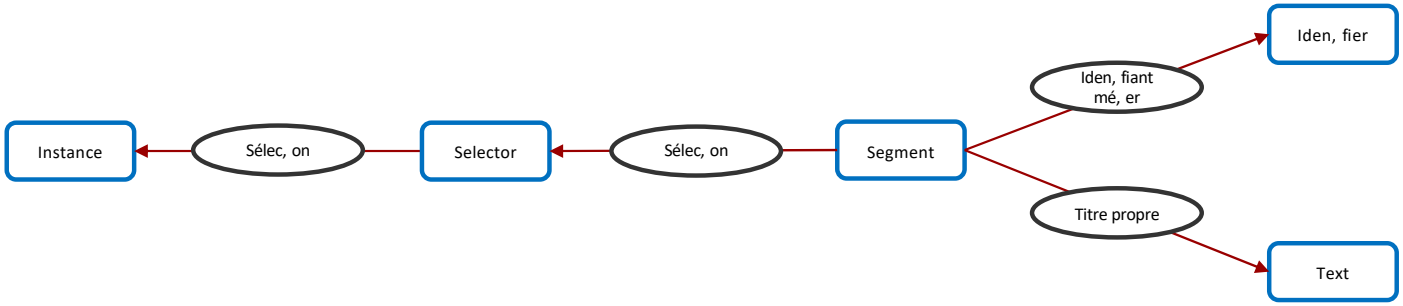
Et produire des données : exemple des segments et instances

La notice I05007967 désigne un **extrait (segment)** relié à une **intégrale (instance)**



Exemple : Texte et identifiant du segment

L'identifiant « source » I05007967 est porté par un **identifier du segment**
Le titre de l'extrait est porté par un **text du segment**



L'IA neuronale

Généralités

L'IA numérique

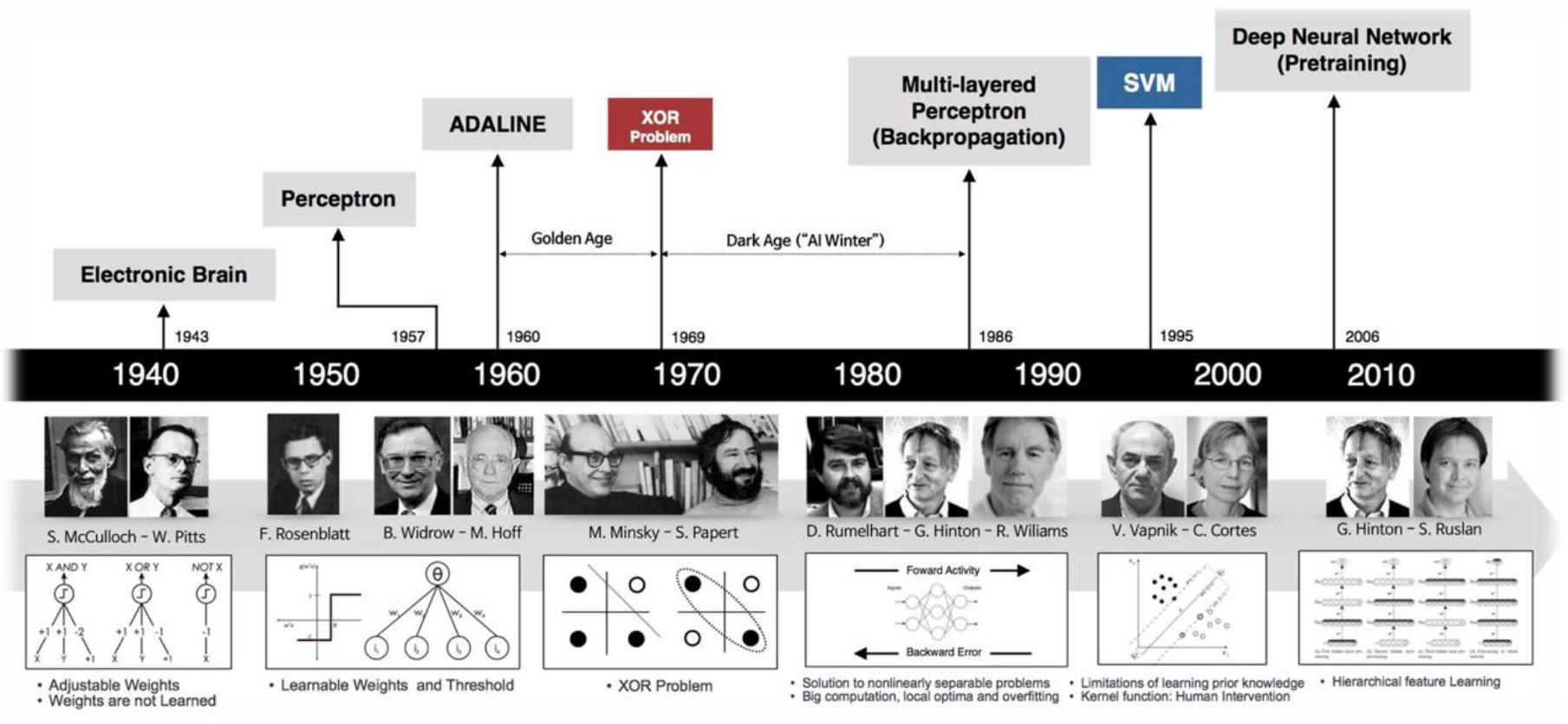
- Les connaissances nécessaires à la production du résultat ne sont pas connues ou ne sont pas explicites : on sait le faire, mais on ne sait pas dire comment on le fait.
- On utilise alors les données à traiter pour abstraire des règles à partir d'elles ; ces règles ne sont pas explicites mais elles sont encodées de manière à être utilisable par la machine.
- Trois approches :
  - Le machine learning : nom donnée aux méthodes mobilisant des outils statistiques / probabilistes pour inférer un modèle à partir des données ; le modèle obtenu n'est pas explicite mais on peut le comprendre car on connaît la méthode qui a permis de le construire.
  - Le deep learning : variante du précédent, mobilisant les réseaux de neurones, où le modèle construit est implicite et réparti dans les paramètres du réseau. Par ailleurs, la méthode d'apprentissage de ces paramètres ne permet pas de comprendre le modèle obtenu et d'expliquer les résultats qu'il produit.
  - L'IA générative : variante du précédent, où les outils développés ne servent pas à reconnaître une donnée, mais à en produire de nouvelles.

Données : 4 étapes

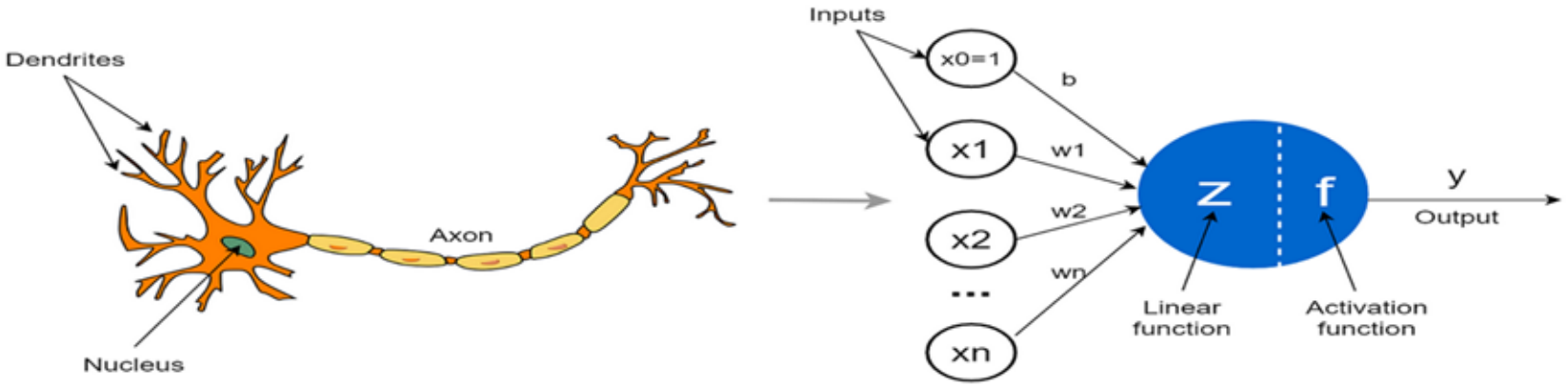
- Le calcul scientifique (1940 – 1950)
  - Ordinateur : number cruncher
- La transaction de gestion (1950 – 1980)
  - Ordinateur = symbolic processor
  - Symbole = nombres ou lettres
- L'ingénierie des contenus (1980 – 2000)
  - Tout contenu peut être codé et calculé
  - On passe de l'informatique au numérique :
    - E.g. Audiovisuel numérique
- Le traitement des data (2000 -)
  - Tout enregistrement devient annoté (linked data) ou intégré (big data) pour être porteur d'information.

Calcul : 4 étapes

- 1957 : Le Perceptron
  - Une architecture de neurones formels permet d'apprendre à classer / reconnaître des formes
- 1969 : Minsky montre les limitations de cette approche
  - La recherche se porte sur l'intelligence artificielle symbolique
- 1985 : Rétropropagation du gradient
  - Une architecture de neurones multicouches permet d'apprendre toutes sortes de population
- Années 2000 :
  - Intelligence artificielle statistique : massification des données, puissances de machines.


Deep Neural Network (Pretraining)
SVM
Multi-layered Perceptron (Backpropagation)
XOR Problem
ADALINE
Perceptron
Electronic Brain
Golden Age
Dark Age ("AI Winter")
1943
1957
1960
1969
1986
1995
2006
1940
1950
1960
1970
1980
1990
2000
2010
S. McCulloch - W. Pitts
F. Rosenblatt
B. Widrow - M. Hoff
M. Minsky - S. Papert
D. Rumelhart - G. Hinton - R. Wiliams
V. Vapnik - C. Cortes
G. Hinton - S. Ruslan
X AND Y
X OR Y
NOT X
Foward Activity
Backward Error
• Adjustable Weights
• Weights are not Learned
• Learnable Weights and Threshold
• XOR Problem
• Solution to nonlinearly separable problems
• Big computation, local optima and overfitting
• Limitations of learning prior knowledge
• Kernel function: Human Intervention
• Hierarchical feature Learning

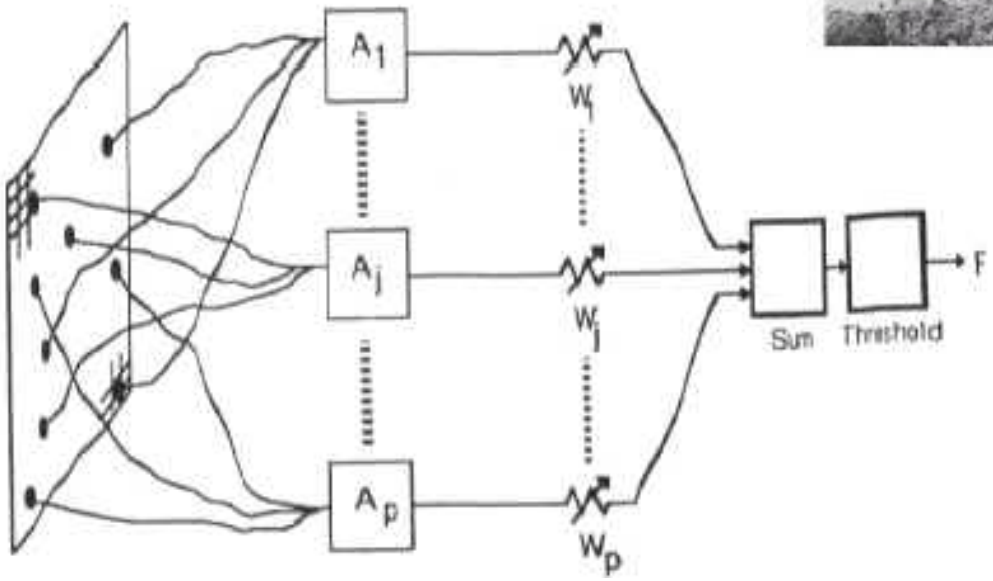
Une formalisation du neurone biologique



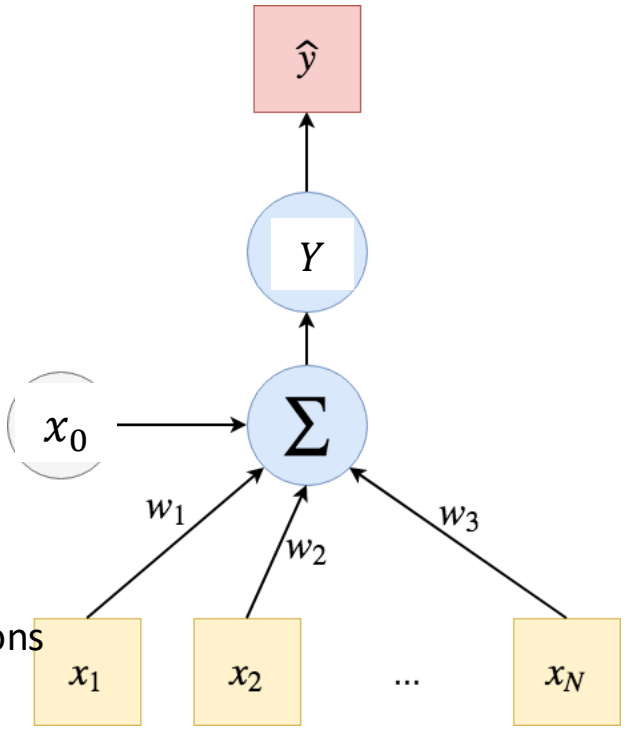
towardsdatascience.com

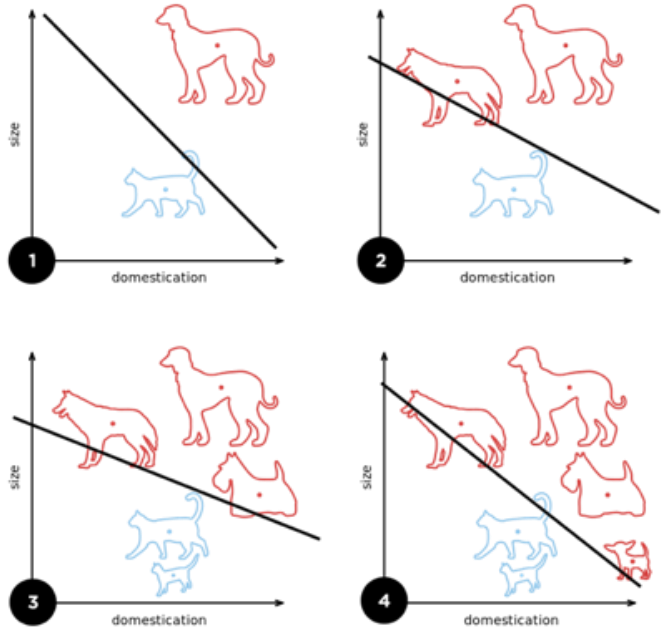
$$y = f\left(\sum x_i . w_i + b \right)$$

Perceptron de Rosenblatt



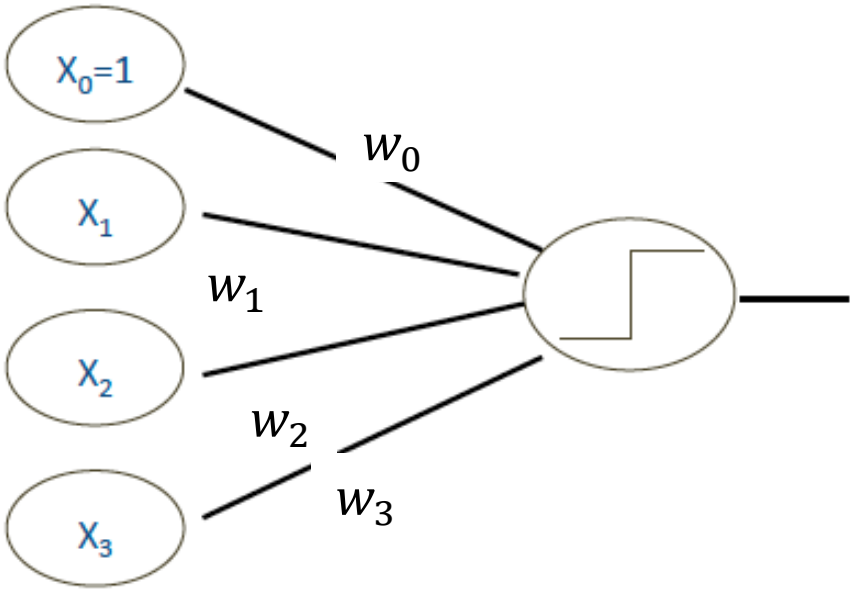
Perceptron : Principe





Avec :

- x_i les valeurs d'entrée,
- w_j les poids des connexions
- x_0 le biais
- Y la fonction de transfert
  - Si x>0 $Y(x) = 1$
  - Si x≤ 0 $Y(x) = 0$
- $\hat{y}$ le résultat $Y(x)$
- $\hat{y} = Y(x_0 + \sum_{i,j}(x_i * w_j))$



Comment calculer les poids synaptiques à partir d'un fichier de données (Y ; X1, X2, X3)

1) Quel critère optimiser ?

2) Comment procéder à l'optimisation ?

(1) Minimiser l'erreur de prédiction

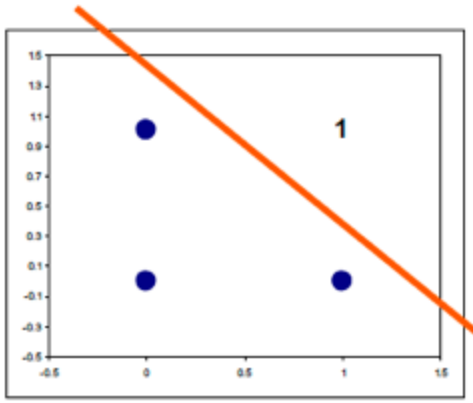
(2) Principe de l'incrémentalité (online

Fonction ET / AND à représenter

- Principales étapes :
  - Mélanger aléatoirement les observations
  - Initialiser aléatoirement les poids synaptiques
  - Faire passer les observations une à une
    - Calculer l'erreur de prédiction pour l'observation
    - Mettre à jour les poids synaptiques
  - Jusqu'à convergence du processus

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Données



Représentation dans le plan

Processus d'apprentissage

- Notations
  - $y = Y(z)$ le résultat du perceptron à partir de l'entrée z;
  - $D = \{(x_1, d_1), \dots, (x_s\ d_s)\}$: ensemble d'apprentissage avec $x_i$les valeurs d'entrée et les $d_i$les sorties attendues correspondantes ;
  - $x_{j,i}$est l'entrée i (sur la synapse i) de la donnée d'apprentissage j.
  - $x_{j,0} = 1$ C'est le biais dans le réseau, toutes les valeurs de l'entrée 0 sont fixées à 1 au départ.
  - $w_i$le i-ème poids reliant l'entrée i à la sortie.
  - $w_i(t)$ le poids i à l'itération t de l'apprentissage.
- Apprentissage :
  - Initialisation : on met les poids à 0.
  - Pour chaque exemple (x_j, d_j) faire :
    - Calculer la sortie du perceptron;
    - $y_j(t) = f[w_o(t)x_{j,0} + w_1(t)x_{j,1} + \cdots + w_n(t)x_{j,n}]$
    - Calculer le nouveau poids :
    - $\boldsymbol{w_i(t+1) = w_i(t) + r(d_j - y_j(t))x_{j,i}}$
    - N.B. r est le taux d'apprentissage, entre 0 et 1.
  - Recommencer sur tous les exemples jusqu'à ce que les poids n'évoluent plus, i.e. les erreurs $(d_j - y_j(t)$ sont nulles, i.e. il n'y en a plus.
  - Le processus aboutit si et seulement si les données sont linéairement séparables.

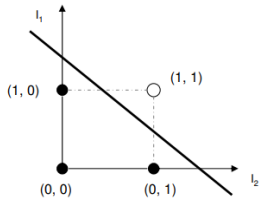
Exemple sur AND

- Initialisation :
  - $w_0 = 0, w_1 = 0, w_2 = 0$
  - $r = 0{,}1$
- On itère avec une donnée :
  - $x_O = 1$ (le biais), $x_1 = 1, x_2 = 1, d = 1$
  - On calcule la réponse du perceptron :
    - $0 \times 1 + 0 \times 1 + 0 \times 1 = 0$ donc $y = 0$
    - Or, d$-y = 1$, il y a une erreur à corriger, il faut modifier les poids :
      - $\Delta w_0 = 0{,}1 \times (1) \times 1 = 0{,}1$
      - $\Delta w_1 = 0{,}1 \times (1) \times 1 = 0{,}1$
      - $\Delta w_2 = 0{,}1 \times (1) \times 1 = 0{,}1$
      - Tous les poids sont modifiés et deviennent $0{,}1$.
- On recommence avec les autres exemples :
  - $x_O = 1$ (le biais), $x_1 = 1, x_2 = 0, d = 0$
  - $x_O = 1$ (le biais), $x_1 = 0, x_2 = 1, d = 0$
  - $x_O = 1$ (le biais), $x_1 = 0, x_2 = 0, d = 0$
- On recommence sur tous les exemples jusqu'à ce que les poids ne bougent plus

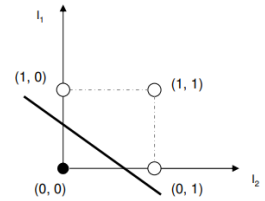
La frontière de 1960

- Minsky et Papert montrent en 1969 que le perceptron ne peut reconnaître que des populations linéairement séparables (en gros, par des droites dans un plan / surface, ou un plan dans un volume).
  - Notamment, la fonction XOR (ou exclusif) n'est pas représentable par un perceptron simple (sans couche cachée).

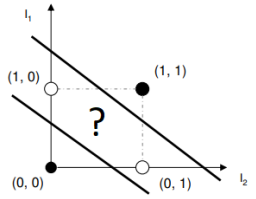
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

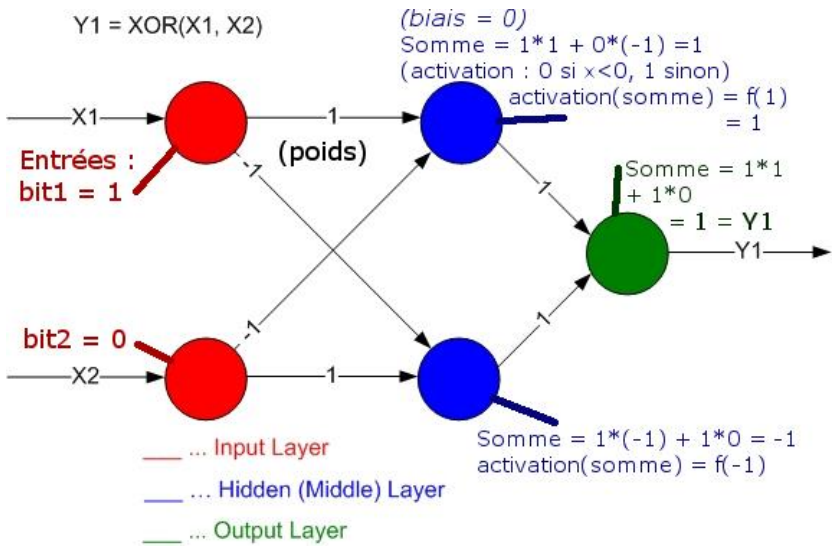


XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

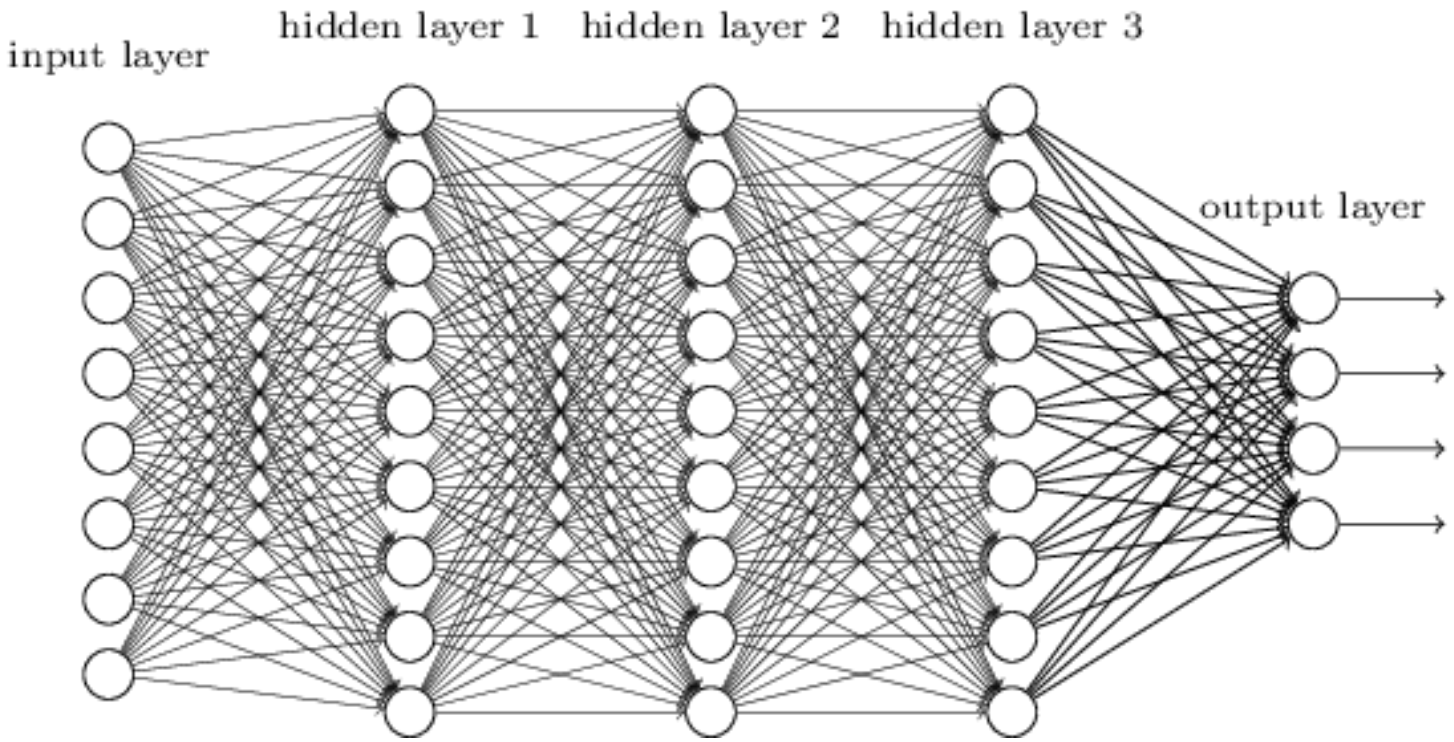


Solution

- introduire une ou plusieurs couches cachées :
  - La couche cachée permet de faire des calculs intermédiaires.
- Théorème d'approximation universelle :
  - Toute fonction réelle bornée est représentable par un perceptron avec une couche cachée
    - On est content en théorie, mais en pratique cela ne nous aide pas beaucoup, car la couche cachée peut être arbitrairement grande…



Donc, avoir beaucoup de couches cachées



Elaborer et utiliser un réseau à couches cachées

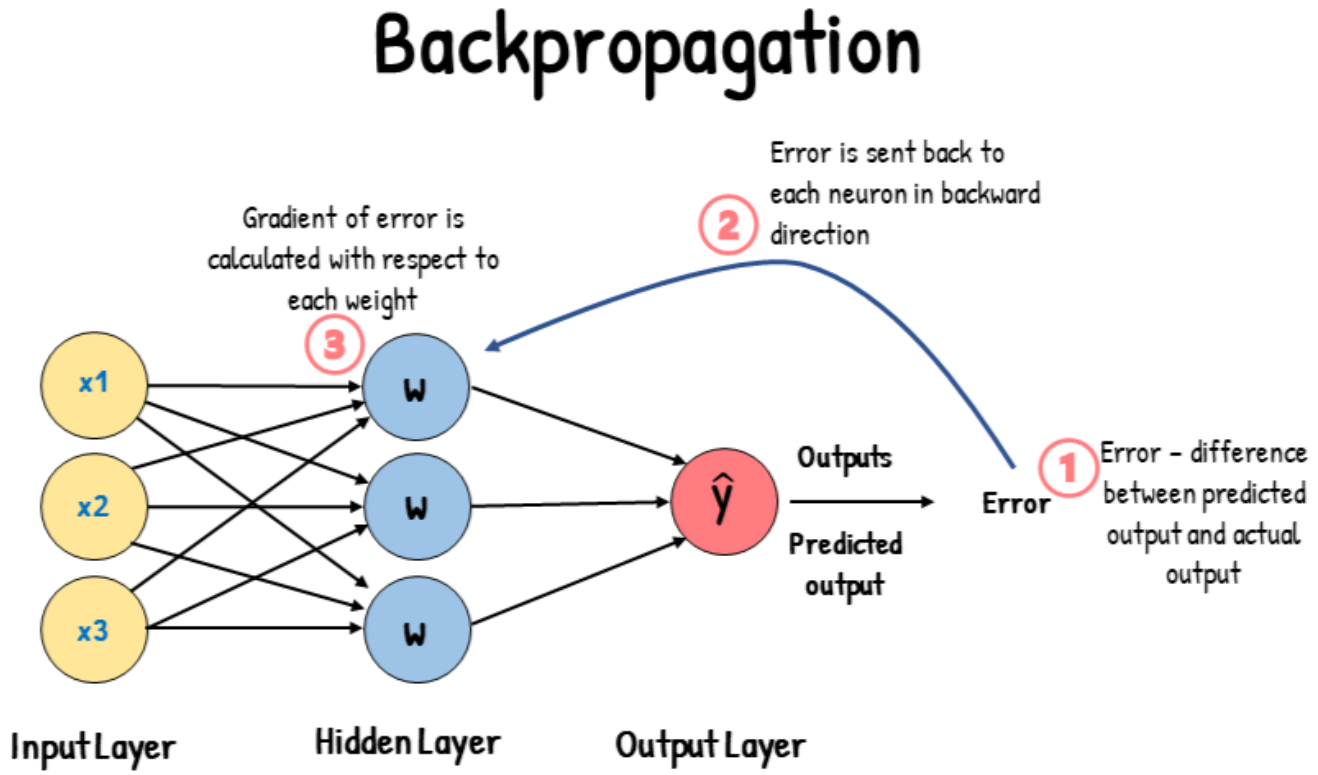
Apprentissage : apprendre les poids

- Définir l'architecture du réseau : nombre de couches, nombres de neurones par couches
- Disposer d'exemples :
  - X une entrée, pour Y la sortie attendue
- Initialiser les poids synaptiques de manière aléatoire ou tous à 0
- Pour tous les exemples, jusqu'à ce que le réseau réponde correctement pour chacun :
  - Calculer la réponse Y' pour l'exemple (X,Y)
  - Calculer l'erreur / l'écart et rectifier les poids en conséquence :
    - Algorithme de rétropropagation du gradient.

Exploitation : utiliser les poids obtenus

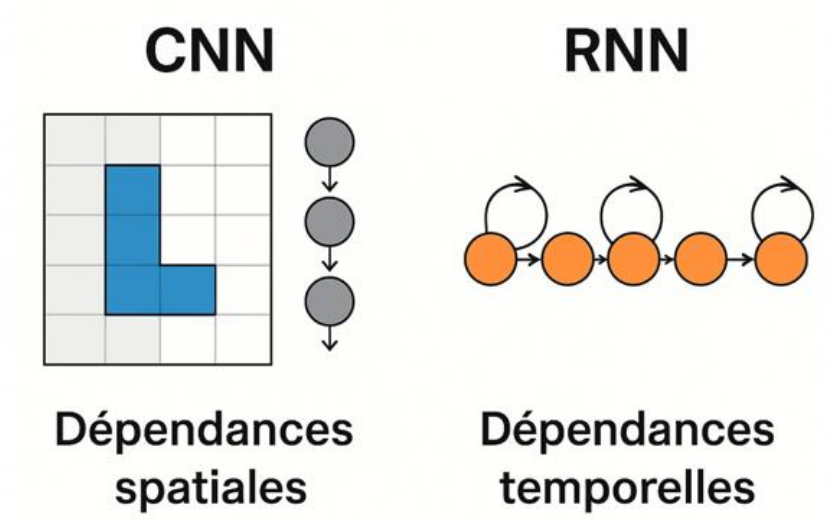
- Quand l'apprentissage est terminé, les poids synaptiques sont stabilisés : le réseau est un modèle appris à partir des données.
- On utilise ce modèle pour prédire la réponse sur de nouvelles données.
- On peut alors évaluer le réseau par rapport à ces nouvelles données.
- En fonction de la qualité de l'évaluation, le réseau peut être utilisé de manière opérationnelle.

Rétropropagation du gradient



Différents réseaux pour différents traitements

- CNN = Convolutionnal neural networks
  - Reconnaître des structures spatiales locales
    - Images, arêtes, cartes, lettres, etc.
  - dépendances locales, invariance spatiale.
- RNN = Recurrent neural networks
  - Reconnaître des structures temporelles pour traiter des series
    - Phrases, audio,
  - dépendances dans le temps, ordre séquentiel.



Catégorie d'apprentissage

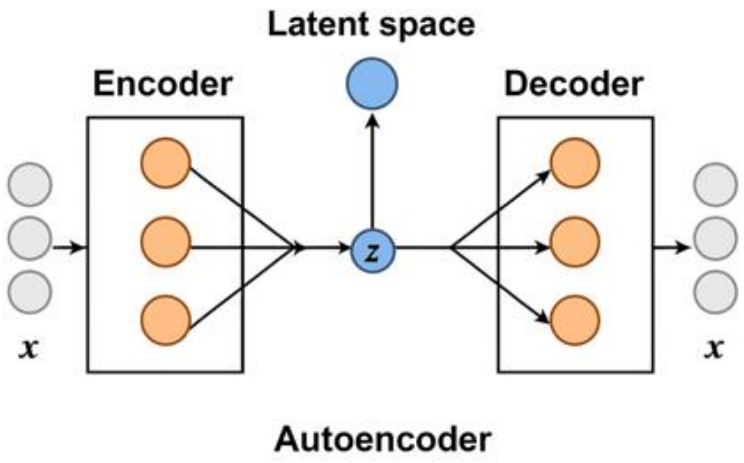
- Apprentissage supervisé :
  - On a des exemples étiquetés : une photo de chat (exemple) et on dit que c'est un chat (étiquette)
  - On veut apprendre à reconnaître sur une photo quelconque est une image de chat (l'image appartient à la classe des chats).
- Apprentissage non supervisé :
  - On a des exemples, mais ils ne sont pas classés ou catégories : des photos de chats et de chiens, mais on ne sait qui est chat et qui est chien.
  - On veut regrouper les images qui se ressemblent et trouver les catégories correspondantes : le groupe des images de chat et le groupe des images de chien
    - Nb : on a des groupes d'images, mais pas d'étiquette : on ne sait pas que les chats sont des chats, ni des chiens des chiens….

Apprentissage auto-supervisé

- Le principe est de fabriquer sa propre supervision à partir de données non étiquetées:
  - On crée une tâche artificielle à partir des données dont on connaît le résultat grâce à la donnée :
  - Exemples :
    - Donnée = « Le petit chat est mort » / tâche = deviner le mot « chat » dans la phrase à trou « le petit <> est mort ».
    - Donnée = image / tâche = reconstruire l'image à partir d'un trou pratiqué dans l'image (pixel mis à blanc)
    - Donnée = image
      - On crée des images supplémentaires à partir de cette image (rotation, recadrage, etc.). Ces images sont toutes des images de la même chose. Elles constituent les images d'une même classe. Les autres images sont alors des exemples négatifs de cette classe.
      - On fait cela pour toutes les images.
      - On apprend à reconnaître les images d'une même classe.

Un outil fondamental : l'autoencodage

- Principe :
  - Entraîner un réseau de neurone à reconstruire la donnée en entrée ;
  - Le réseau est sous contrainte:
    - dimensions plus petites que la donnée : le réseau doit réduire l'information. Pour cela, l'entraînement dégage des données les informations importantes et discriminantes.
    - Bruit ajouté à l'image pour retrouver l'image originale (cf infra modèle de diffusion)
  - Le résultat est donc un réseau qui construit un encodage réduit et « sémantique » des données.
- Étapes :
  - Encodeur : z = encodeur(x = donnée).
    - z est le vecteur latent codant l'entrée.
  - Décodeur : x' = decodeur(z)
    - x' est une approximation car reconstruction de x
    - Intéressant car permet de construire objet voisin de x, proche de lui car il est reconstruit à partir de l'encodage des traits pertinents de x.
  - x → [Encoder] → z → [Decoder] → x̂



IA neuronale

Impacts sur le traitement documentaire : un exemple.

IA et traitement des archives

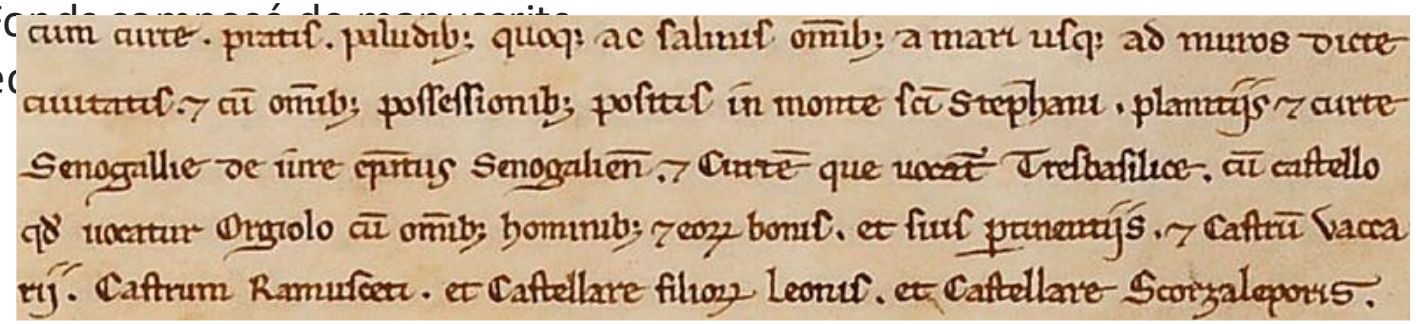
- L'enjeu est de reconnaître et de rendre manipulable l'information contenue dans les archives :
  - Outils de reconnaissances de formes fondés sur les IA neuronales (deep learning) ou statistique (machine learning)
  - Par exemple :
    - Reconnaissances des visages, des voix, scènes, des
    - Écritures manuscrites, etc.

Un exemple : inCodice Ratio

Des archives du Vatican

- **85km de rayonnages**, plus de 600 collections d'archives
- Des textes allant **du VIII au XXe siècle**
- Fo
é

Écriture caroline



Écriture caroline

- Claude Médiavilla
- Calligraphie
- Éditions de l'Imprimerie Nationale 1993

Une difficulté : la segmentation

- Le problème à traiter :
  - Une segmentation imprécise des caractères dans la minuscule caroline
- Une solution :
  - Considérer les mots comme une suite de segments et non comme une suite de caractères

Exemple du mot latin "anno" donnant lieu à de nombreuses interprétations par la ROC : “aiiiio”, “aimo”, “amio”, “aniio’, “aiino”, and “ainio”. (aucun mot latin).

Segmentation en tranches

Segmentation puzzle

Approche

- Entrainer le programme à reconnaitre l'épaisseur des traits de plume ;
- Les traits les plus fins sont généralement ceux liant les caractères les uns aux autres ;
- Utiliser les probabilités :
  - Un segment « iii » est probablement un « m » mal déchiffré.

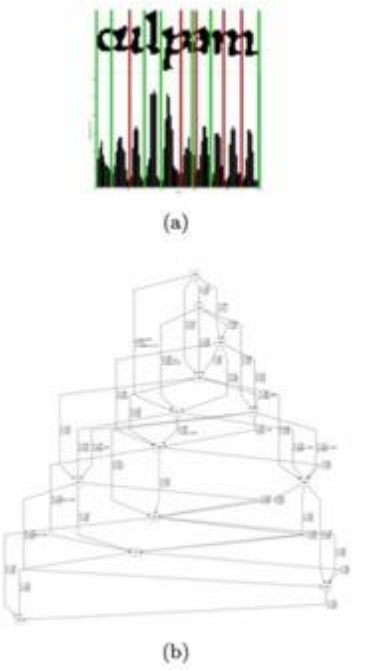


Fig. 3: Cut-points for the word "culpam" and corresponding lattice. We use green for actual character boundaries, and red otherwise.

path	prob
culpam	
cullum	$1 \cdot 10^{-4}$
culpam	$8 \cdot 10^{-7}$
culluni	$3 \cdot 10^{-7}$
criminis	
criminis	$8 \cdot 10^{-7}$
crinunis	$6 \cdot 10^{-8}$
crinuius	$4 \cdot 10^{-8}$
uiuscemod(i)	
uiufemod	$2 \cdot 10^{-2}$
uuifemod	$2 \cdot 10^{-3}$
uiiifemod	$5 \cdot 10^{-4}$

Table 1: Path probabilities for the words in Figure 4.

Un préalable : données d'apprentissage

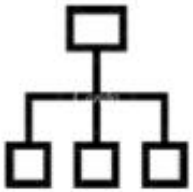
- Avant d'utiliser l'outil : collecte des données
- Application de crowdsourcing ;
- Participation des lycéens

Processus global

Image en haute résolution d'un manuscrit

Classification entre combinaison de traits reconnues et inconnues

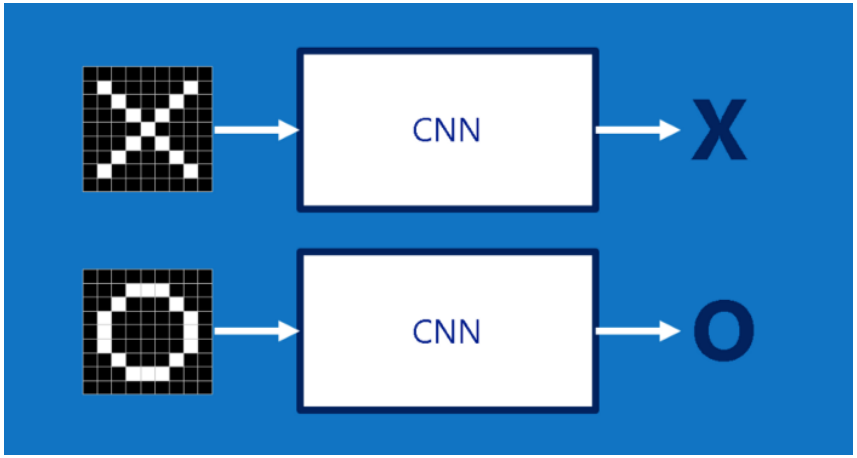
La page est segmentée par **crowdsourcing** en mots, les mots sont segmentés en traits

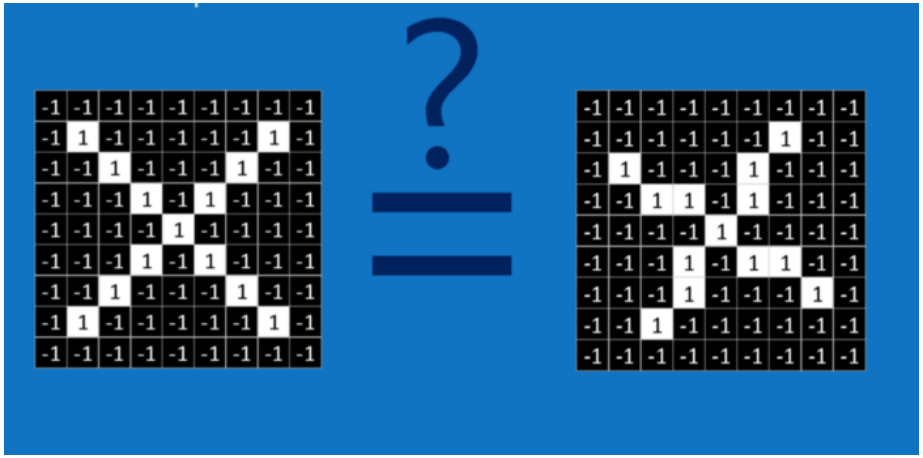


Classement des probabilités de résultats

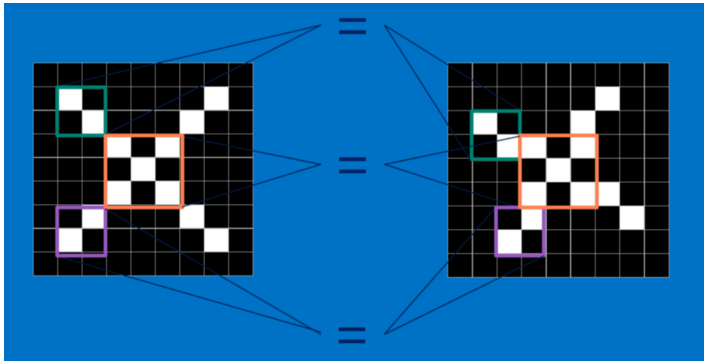
Convolutional Neural Networks

Le problème





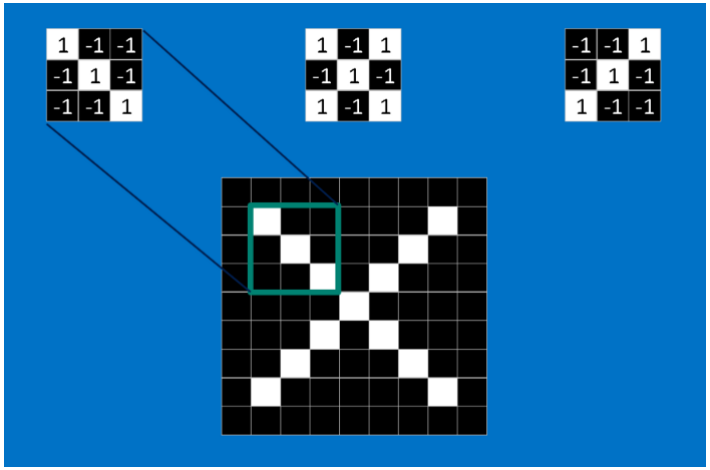
Convolution Principe = Extraire les caractéristiques



- Principe:
  - Décomposer l'image en différents fragments ;
  - Analyse chaque fragment pour voir s'il correspond à une caractéristique recherchée, un motif composant l'image globale.
- L'idée est donc de s'intéresser aux sous-images dans l'image plutôt qu'à l'image globale.

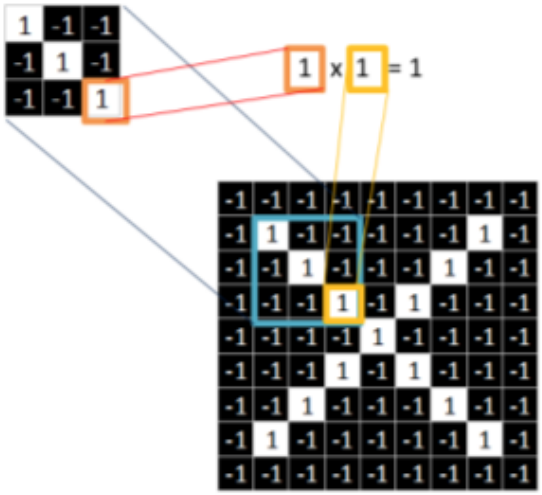
Convolution Enjeu = les motifs recherchés

- Principe :
  - Repérer les motifs structurants composant l'image.
- Dans l'exemple :
  - Un motif pour les branches « descendantes » du X (imagette à gauche)
  - Un motif pour le croisement des deux diagonales au milieu (imagette du milieu)
  - Un motif pour la recherche des branches « montantes » (imagette à droite)
- Enjeu :
  - Trouver une méthode pour comparer ces motifs au contenu effectif de l'image soumise au réseau.



La convolution : la méthode

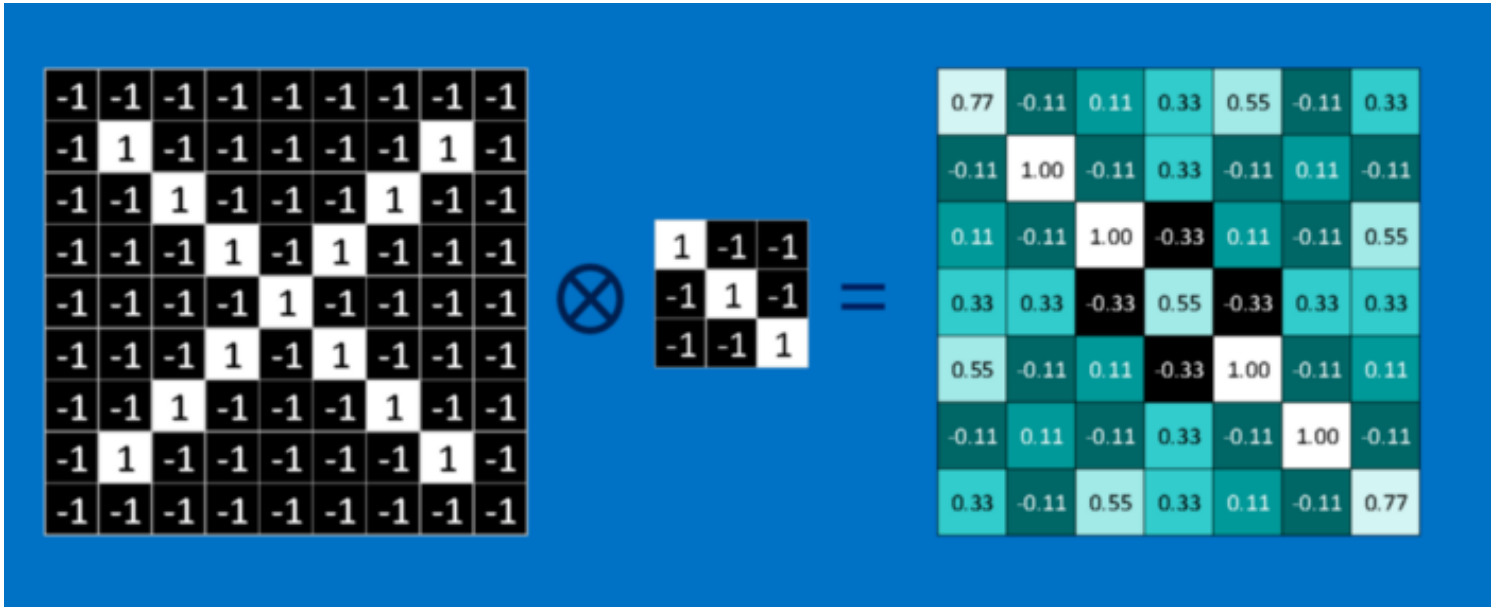
- Principe :
  - On prend une partie (fenêtre) de l'image à analyser de même taille que le motif d'exploration
  - Cette fenêtre commence en haut à gauche et on la fait glisser sur toutes les cases de l'image soumise
  - Pour cette partie :
    - Les valeurs des pixels sont multipliées un à un, additionnées et on divise par le nombre de pixel du motif :
      - Ici : (1 + 1 +1 +1 +1 +1 +1+ 1+1)/9 = 1
      - Si toutes les valeurs coïncident : on a 1
        - L'imagette est le positif (image conforme) du motif
      - Si toutes les valeurs sont différentes : on a -1.
        - L'imagette est le négatif (image inversée) du motif.
    - On recommence pour toutes les cases de l'image.
  - On recommence la convolution sur l'image obtenue par une convolution précédente.



1	1	1
1	1	1
1	1	1

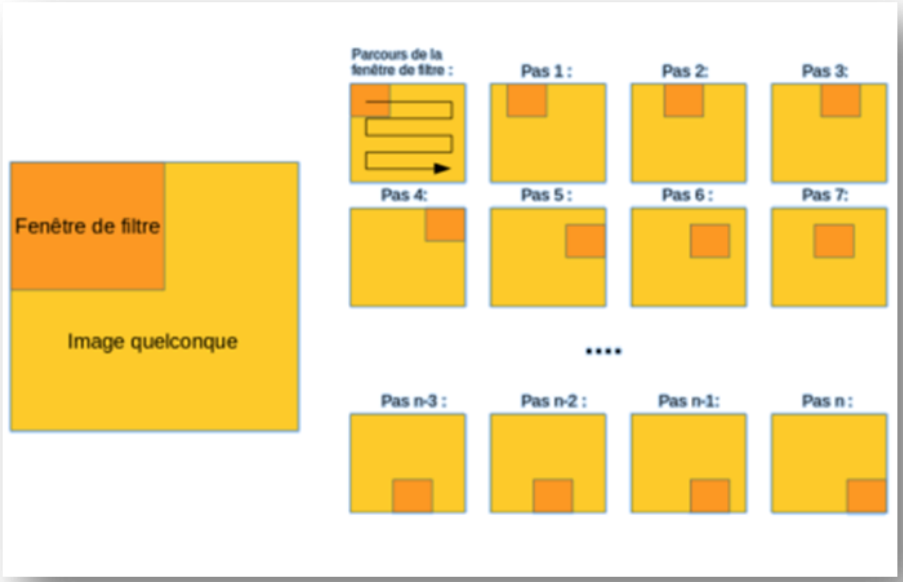
Convolution : résultat

- Calcul de la première cellule:
  - (-1 + 1 +1 +1 +1 +1 +1+ 1+1)/9 = 0,77



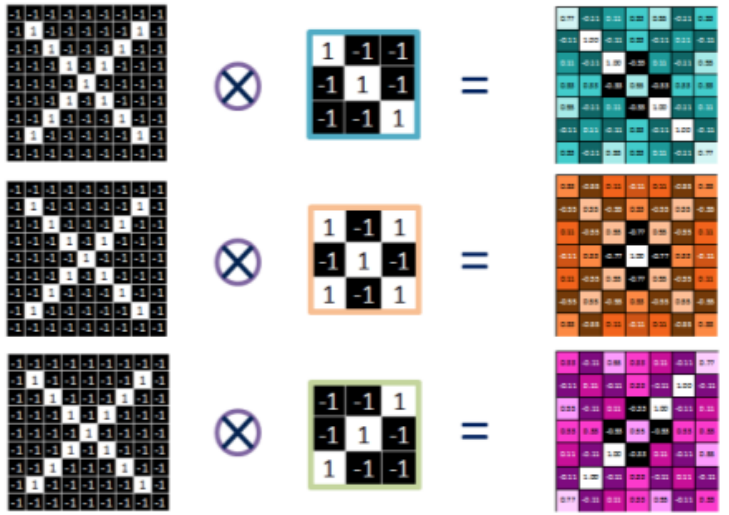
Convolution : les réglages

- Trois hyperparamètres :
  - Profondeur de la couche :
    - nombre de noyaux de convolution ou nombre de motifs.
  - pas de contrôle du chevauchement des champs récepteurs
    - Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.
  - La marge (à 0) ou zero padding :
    - Principe : mettre des zéros à la frontière du volume d'entrée.
    - Intérêt : contrôle la dimension spatiale du volume de sortie.



Convolution : démultiplication sur les motifs

- On effectue la convolution pour chaque motif :
  - On obtient une matrice / image composée des « scores » mesurant la conformité de l'image au motif examiné.
- On a au final autant de matrices que de motifs.



Pour de vrai

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Operation	Filter	Convolved Image
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Pour se faire peur…

- On note X un tableau $n \times m$, f un filtre $n_f \times m_f$ avec $n_f \leq n, m_f \leq m$, la convolution de X par f pour un pixel de coordonnées $i, j, i \in [\![0, n-1]\!], j \in [\![0, m-1]\!]$ est définie par :

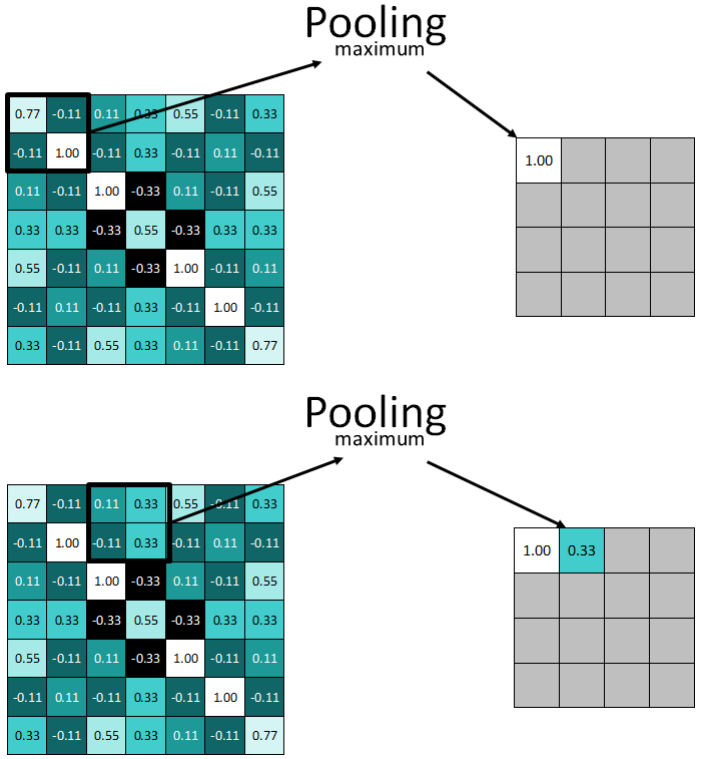
- $$(X * f)[i,j) = \sum_{k=-\frac{n_f}{2}}^{\frac{n_f}{2}} \sum_{l=-\frac{m_f}{2}}^{\frac{m_f}{2}} X[i+k, j+l] * f[k+\frac{n_f}{2}, l+\frac{m_f}{2}]$$

- Dans nos exemples : $n_f = m_f = 2$

- Remarque sur le padding :
  - Dans le cas ci-dessus, si i+k ou j+l sort de l'image X (c'est-à-dire, i+$k \geq n$ par exemple) alors on prend Xi+k,j+l=0, on parle de « zéro-padding ». D'autres paddings sont possibles (en reprenant la valeur du pixel le plus proche par exemple).
  - On peut aussi choisir de ne faire le calcul que sur les pixels i,j tels que i+k et j+l sont toujours dans l'image : il y aura alors réduction de la dimension de sortie

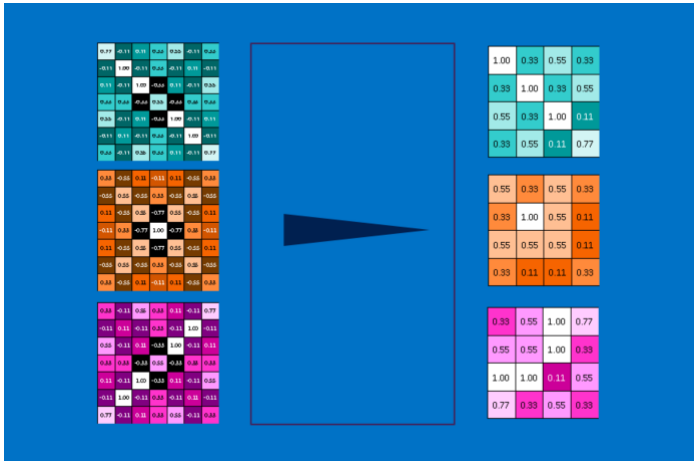
Pooling

- Enjeu
  - Réduire la taille de la matrice
- Principe
  - On prend une fenêtre, par exemple de taille 2 x 2, on la fait glisser de 2 en 2 (pas de chevauchement) sur la matrice : chaque bloc est remplacé par sa valeur maximale.
    - On prend parfois 3x3, avec un pas de 2 ce qui donne un chevauchement.
  - En le faisant sur toute la matrice, on en obtient une nouvelle 4 fois plus petite.



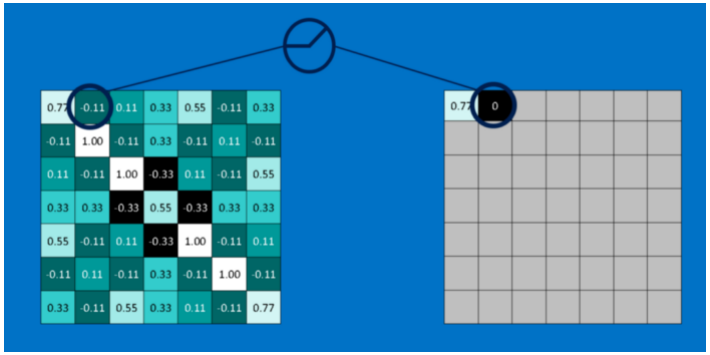
Pooling suite

- Le Pooling est effectué sur chaque matrice obtenue par convolution.
- Le pooling retient la valeur la plus importante, si bien qu'on sait que le motif a été reconnu dans l'image, mais on perd l'information de savoir où.
- L'intérêt du pooling est de réduire la taille des calculs qui sont très importants puisqu'on le fait un calcul matriciel pour chaque cellule de l'image initiale.



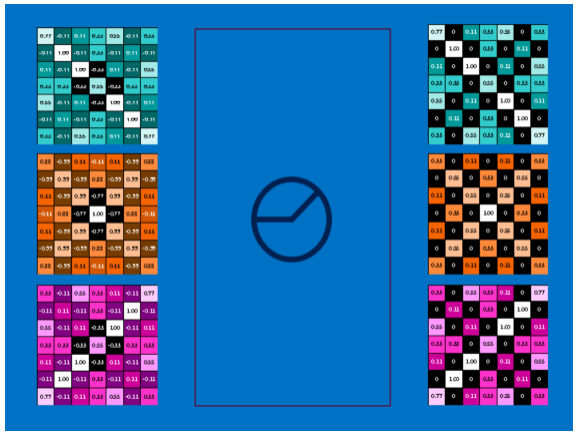
ReLu

- Unité rectifiée linéaire ou Rectified Linear Unit
- Principe :
  - Toutes les valeurs négatives sont remplacées par 0.
  - Cela permet d'assurer une bonne convergence des calculs.



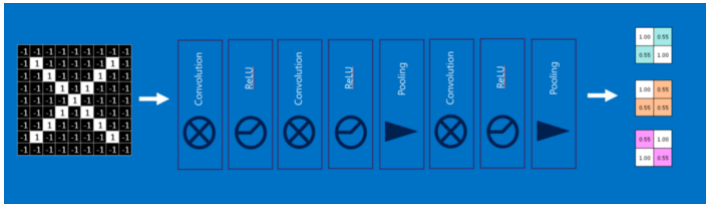
Au final

- Après
  - Convolution
  - Pooling
  - Relu
- On obtient
  - Autant de matrices que de motifs, mais réduites en taille et rectifiées.
  - On a la mémoire des motifs présents, mais sans savoir où, et uniquement en positif.



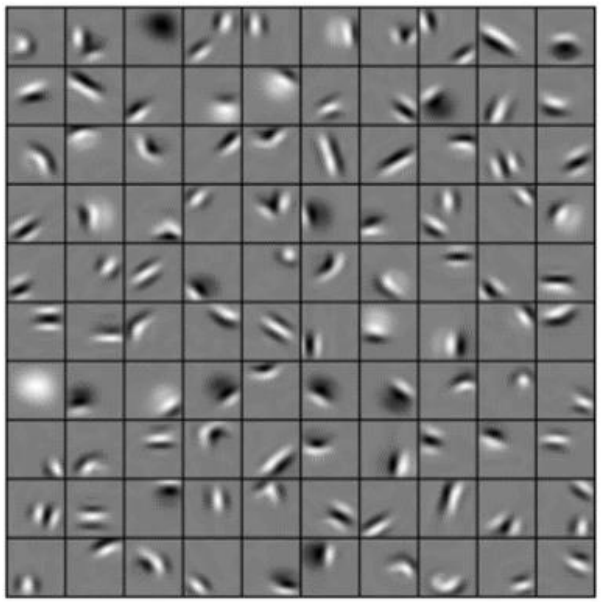
Passer au deep learning

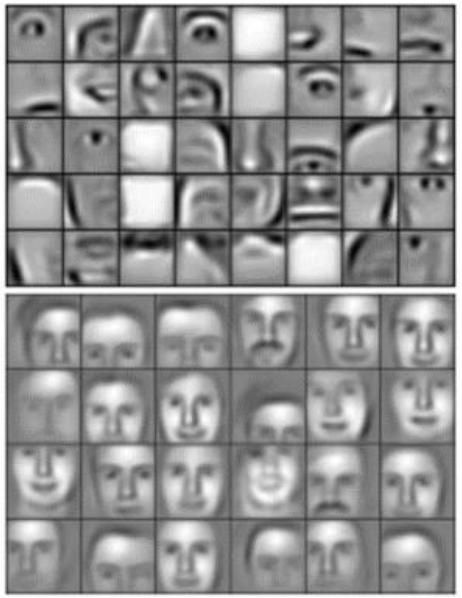
- Puisqu'on obtient après chaque étape une image (avec une taille et un contenu modifié), on peut composer les étapes de traitement dans l'ordre que l'on veut :
  - C'est la cuisine du deep learning : proposer une architecture permettant au mieux d'analyser les entrées
  - Essais / erreurs = ajustements empiriques.



Différentes caractéristiques

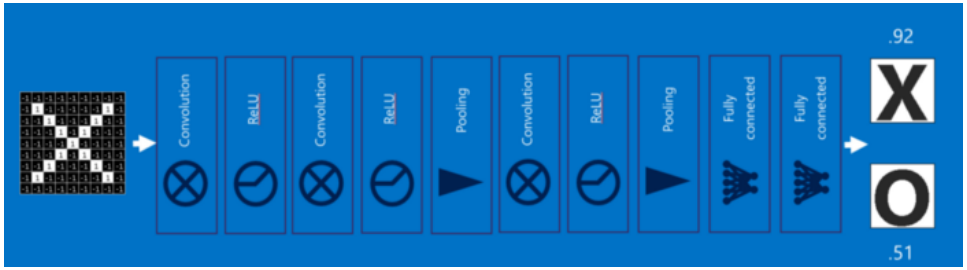
- Les couches hautes du CNN extraient des motifs a priori reconnaissables car ils correspondent à des caractéristiques de haut niveau (parties de visage ici).
- Les couches basses correspondent à des caractéristiques de bas niveaux qu'on ne sait pas forcément interpréter ou qui n'ont pas de lien évident avec l'image analysée : des points lumineux, des traits, etc.





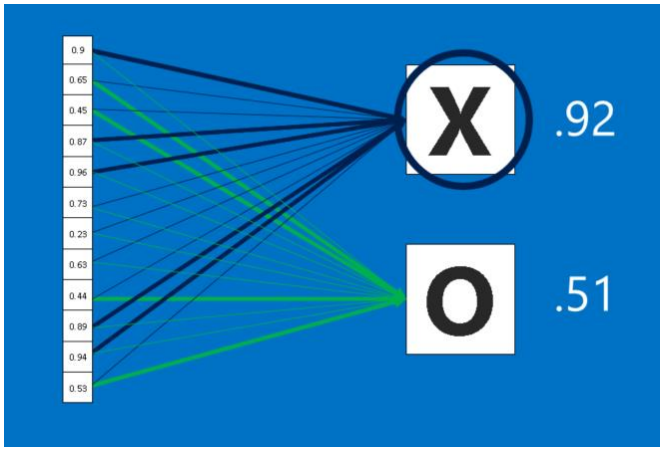
Les couches entièrement connectées

- On n'a pas encore décidé si on a reconnu un X ou un O
- On introduit pour cela :
  - Des couches entièrement connectées
  - Chaque neurone d'une couche reçoit un vote (valeur pondérée) des neurones de la couche précédente ;
  - Une fonction seuil permet de déterminer la valeur résultante de la dernière couche neurone en fonction des votes donnant une valeur pour chacun de ses neurones.



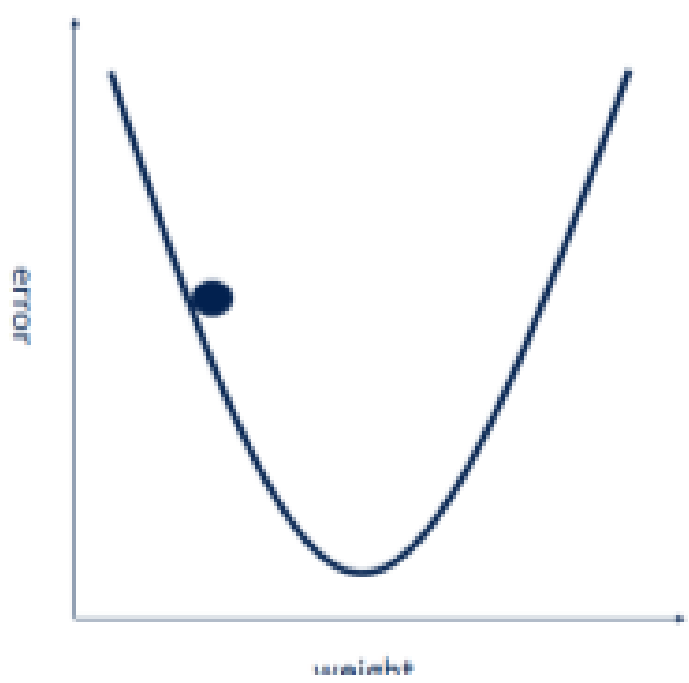
Enfin, l'apprentissage : la rétropropagation du gradient

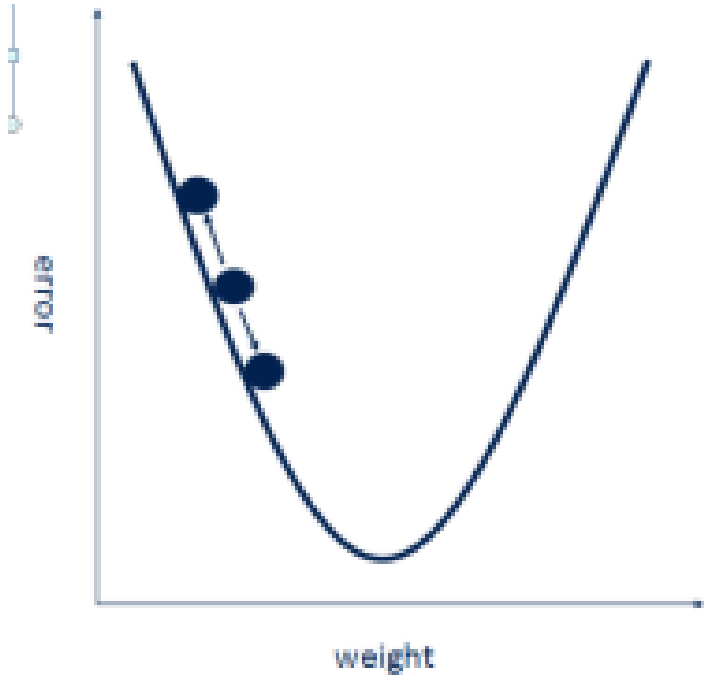
- On n'a pas encore défini:
  - D'où viennent les motifs d'analyse ?
  - Comment déterminer la valeur des poids entre les neurones de deux couches totalement connectées
- Méthode :
  - On part d'un ensemble d'images réelles classées : on sait lesquelles doivent être reconnues comme des X, et les autres comme des O.
  - On part du réseau avec les pixels des motifs et les poids des connections fixés de manière aléatoire.
  - On soumet une par une chaque image réelle : le réseau propose une réponse.
  - A partir de l'écart entre la réponse fournie et la réponse attendue, on rétropropage l'erreur sur les couches connectées pour ajuster les poids des connections et les motifs pour ajuster la valeur de leurs pixels.
- Conséquence :
  - La qualité de l'apprentissage dépend de la qualité et de la représentativité de la base d'images utilisées.
  - Le processus est gourmand en calcul : il a fallu attendre l'explosion des capacités de calcul pour avoir une exploitation effective des CNN.



	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49
		Total	0.57

Principe : minimiser un grandeur

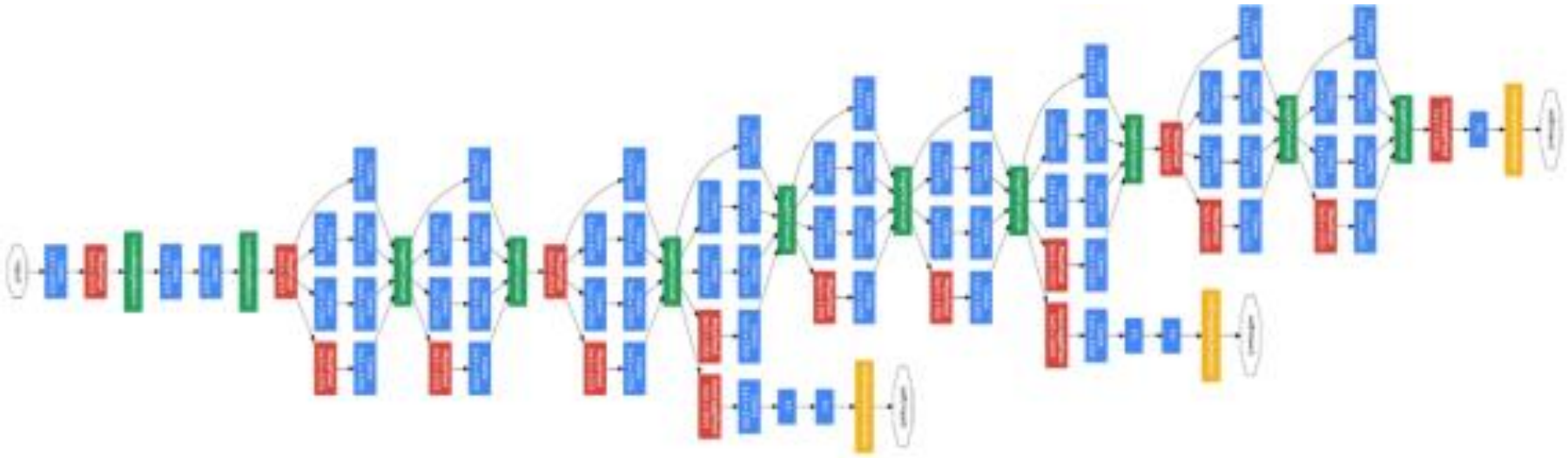




Top Chef

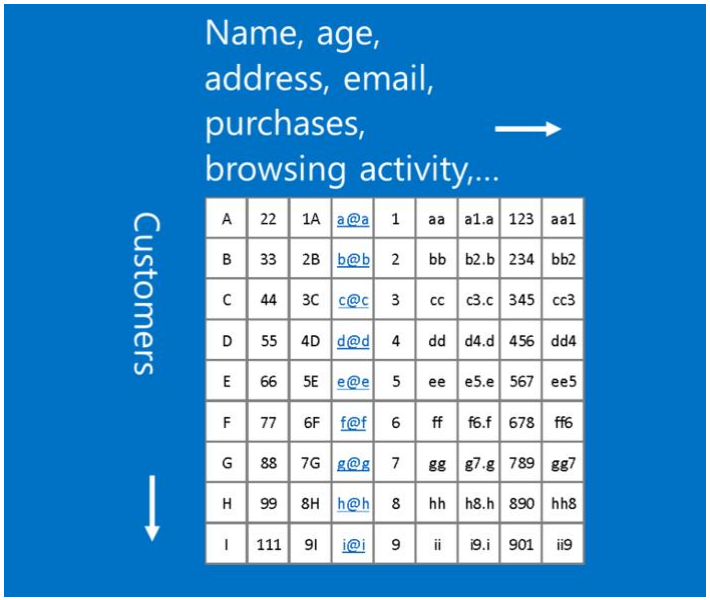
- Pour chaque couche convolutive, combien de caractéristiques doit on choisir? Combien de pixels doit-on considérer dans chaque caractéristique?
- Pour chaque couche de Pooling, quelle taille de fenêtre doit-on choisir? Quel pas?
- Pour chaque couche entièrement connectée supplémentaire, combien de neurones cachés doit on définir?
- Pour chaque couche convolutive, combien de caractéristiques doit on choisir? Combien de pixels doit-on considérer dans chaque caractéristique?
- Pour chaque couche de Pooling, quelle taille de fenêtre doit-on choisir? Quel pas?
- Pour chaque couche entièrement connectée supplémentaire, combien de neurones cachés doit on définir?
- Note : ces réglages sont appelés « hyperparamètres », car ils configurent l'architecture du réseau.

GoogleLeNet (2014)



Et au-delà des images ?

- Principe:
  - Coder le domaine traité sous forme de tableau, ce qui en fait un problème analogue au traitement d'images, où l'on veut retrouver des motifs caractéristiques de corrélations ou formes intéressantes.

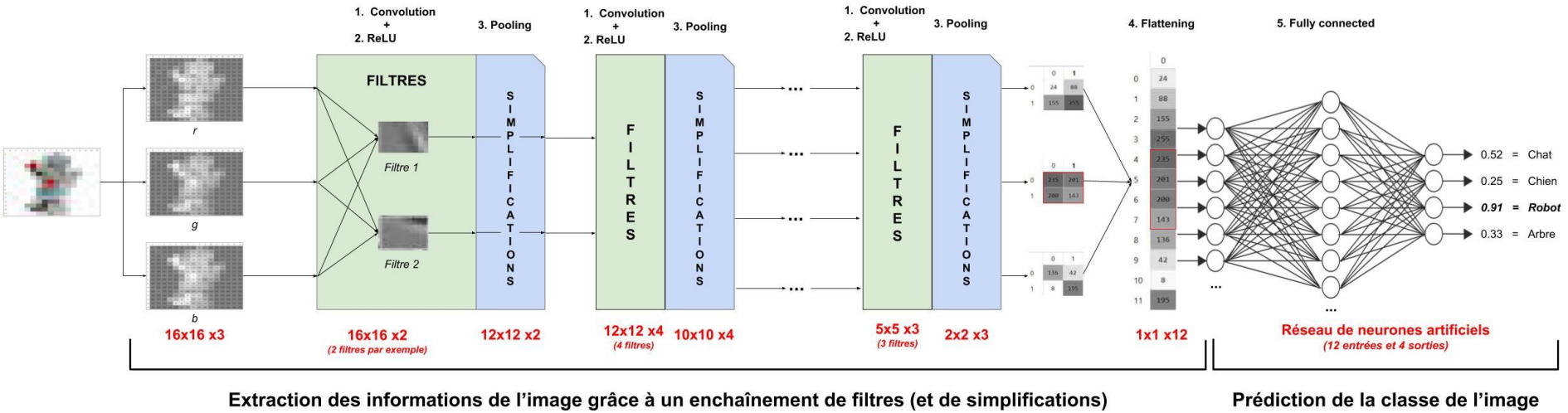


A	22	1A	a@a	1	aa	a1.a	123	aa1
B	33	2B	b@b	2	bb	b2.b	234	bb2
C	44	3C	c@c	3	cc	c3.c	345	cc3
D	55	4D	d@d	4	dd	d4.d	456	dd4
E	66	5E	e@e	5	ee	e5.e	567	ee5
F	77	6F	f@f	6	ff	f6.f	678	ff6
G	88	7G	g@g	7	gg	g7.g	789	gg7
H	99	8H	h@h	8	hh	h8.h	890	hh8
I	111	9I	i@i	9	ii	i9.i	901	ii9

Récapitulation-1

- 4 types de couches :
  - Convolution
    - Toujours la première couche, mais peut être utilisée plusieurs fois.
    - Permet d'extraire des caractéristiques.
  - Pooling
    - Réduction de la taille
    - Perte de l'information de position dans l'image
  - Relu
    - Relu(x) = max(0,x), mais d'autres fonctions sont possibles.
    - Normalisation pour l'efficacité des calculs
  - Couches entièrement connectées
    - Toujours la dernière couche du CNN.
    - Fonction d'arbitrage pour proposer un choix en sortie.

Récapitulation-2



Extraction des informations de l'image grâce à un enchaînement de filtres (et de simplifications)

Prédiction de la classe de l'image