UNIVERSITÉ CÔTE D'AZUR

Bases de l'IA

Réseaux de neurones, Word embeddings, Deep Learning… Quelques éclaircissements

Elena CABRIO

elena.cabrio@univ-cotedazur.fr

Plan pour cette séance

- Principes et fonctionnement des réseaux de neurones "classiques"
- Les méthodes neuronales d’analyse distributionnelle – Word Embeddings
- Principes techniques et épistémologiques de l'apprentissage profond (deep learning)

Remerciements: Bénédicte PIERREJEAN & Ludovic TANGUY

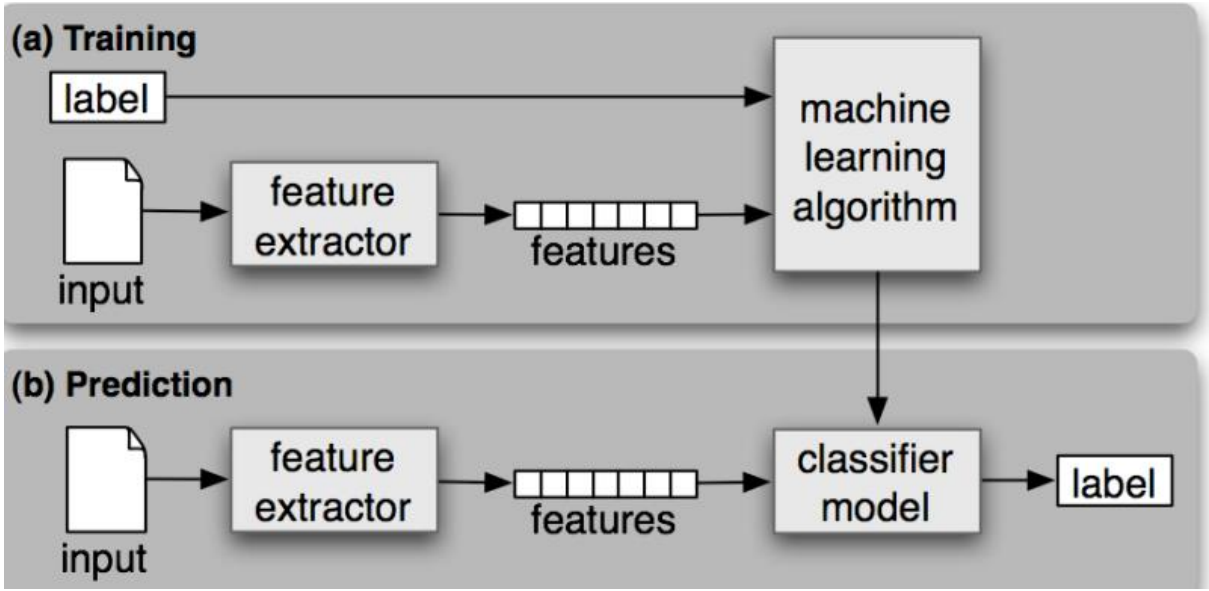
Rappel:
Une tâche centrale, la classification

- "Most of science can be viewed as attempts to find useful ways to categorize phenomena." (Solomonoff 1957)
- Classer = attribuer une catégorie (dans une liste fermée) à un objet
  - Tagger des mots, filtrer des spams, reconnaître un caractère manuscrit, transcrire la parole, etc.
- Autres tâches connexes :
  - Régression : attribuer un score numérique à une entité
  - Clustering : proposer des catégories sans les avoir prédéfinies (regrouper des objets similaires)

Rappel:
Classification par apprentissage supervisé

- Réunir un ensemble de cas déjà résolus
  - E.g. faire étiqueter les mots d'un texte à la main
- Expliciter/calculer les caractéristiques supposées pertinentes d'un objet
  - Taille d'un mot, forme (chaîne de début, de fin), position, catégories des mots voisins, etc.
- Appliquer une méthode d'apprentissage qui va rechercher automatiquement ce qui relie les caractéristiques d'un objet à sa catégorie
  - Et représenter ces liens dans un modèle prédictif
- Appliquer ce modèle aux nouveaux cas à traiter

Rappel:
Classification par apprentissage supervisé



Réseaux de neurones : un peu d'histoire (1)

UNIVERSITÉ CÔTE D'AZUR

- Réseaux de neurones = ensemble d'algorithmes ayant pris pour modèle le cerveau humain
- McCulloch & Pitts (1943) : premier modèle conceptuel du réseau de neurones
- Donald Hebb (1949) : The Organization of Behavior
  - Chaque fois qu'une connexion entre neurones est utilisée -> renforcée
  - Concept fondamental de l'apprentissage humain
- Années 1950 : avancements technologiques, possible de simuler réseau de neurones

Réseaux de neurones : un peu d'histoire (2)

UNIVERSITÉ CÔTE D'AZUR

- Widrow & Hoff (1959) : premiers modèles développés utilisant des réseaux de neurones
- Manque de financements, laissé un peu de côté
- Premiers succès : promesses non tenues, questions philosophiques (quels effets de ces nouvelles « machines pensantes »)
- Réseaux de neurones :
  un peu d'histoire (1)Regain d'intérêt pour les réseaux de neurones dans les années 80

Réseaux de neurones : un peu d'histoire (3)

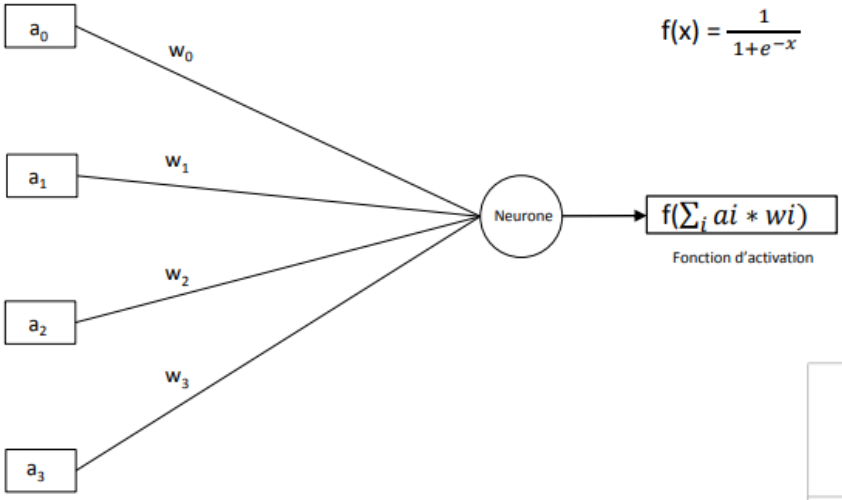
UNIVERSITÉ CÔTE D'AZUR

- Abandon dans les années 90 :
  - Problèmes de vitesse d'apprentissage (notamment par rapport aux SVM)
  - Dépendant des développements du matériel informatique
- Retour en grâce dans les années 2000 :
  - Machines plus rapides, calcul parallèle
  - Succès en traitement d'image, de parole, puis en TAL général
  - Et bien sûr le Deep Learning

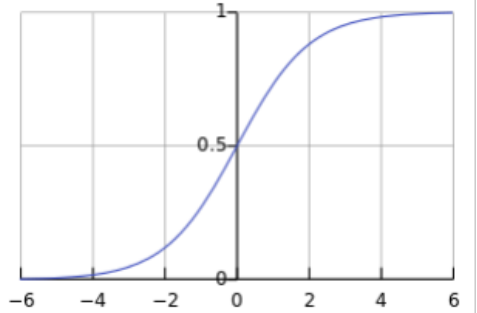
Neurone : définition

- Unité qui prend n entrées et produit une seule sortie
- Le neurone en action a lui-même une valeur d'activation
  - Activé ou non (0/1 modèle binaire)
  - Valeur numérique quelconque (modèle continu)
- L'activation d'un neurone dépend :
  - Des valeurs en entrée (a_i)
  - Des poids de chaque connexion (w_i)
- Activation du neurone :
  - Somme des produits entrée*poids
  - Fonction d'activation

Neurone: illustration



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



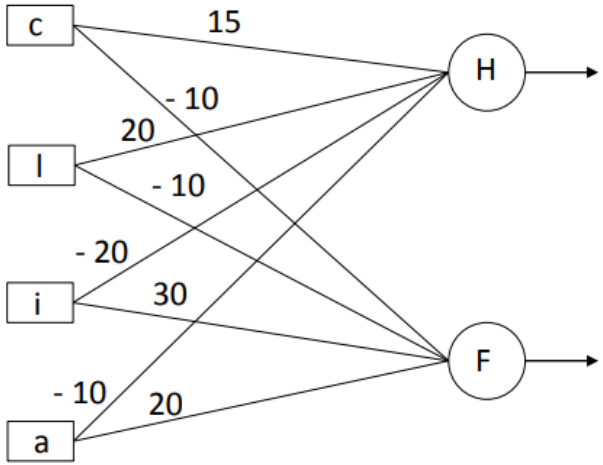
https://en.wikipedia.org/wiki/Sigmoid_functior

Witten & Frank (2005)

Perceptron

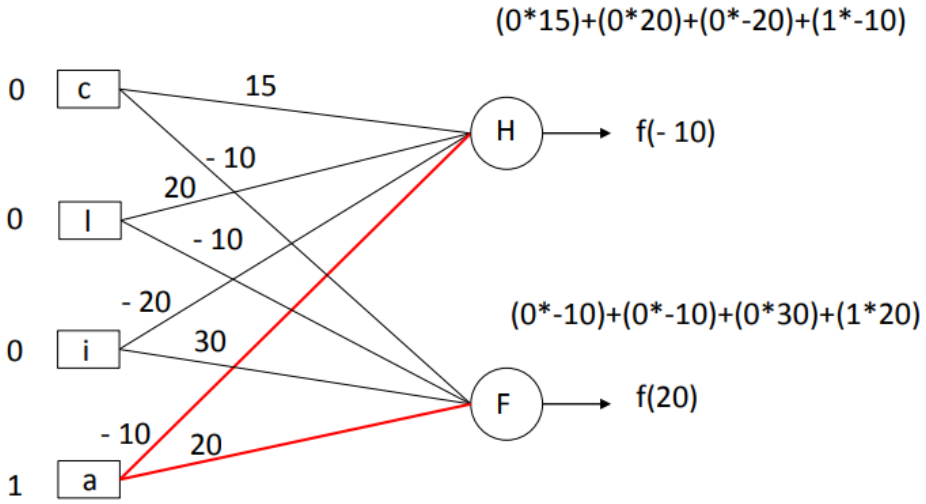
- Réseau de neurones le plus simple
- Deux couches :
  - Entrée : un point d'entrée pour chaque attribut
  - Sortie : un neurone par valeur possible
- Tous les nœuds d'entrée connectés à couche de sortie
- Poids pour chaque connexion nœud entrée/nœud sortie
- Equivalent à régression linéaire/logistique

Perceptron mono-couche: exemple (1)



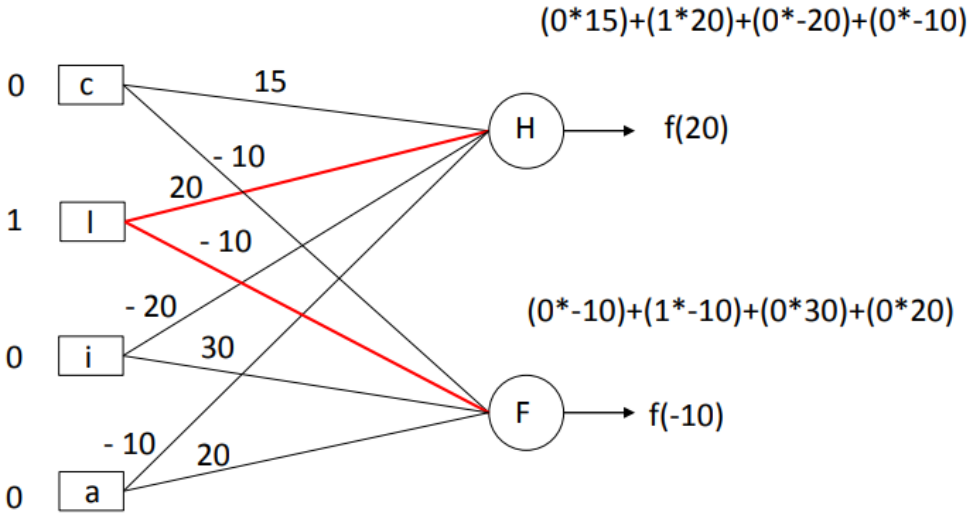
Witten & Frank (2005)

Perceptron mono-couche: exemple (2)



Witten & Frank (2005)

Perceptron mono-couche: exemple (3)



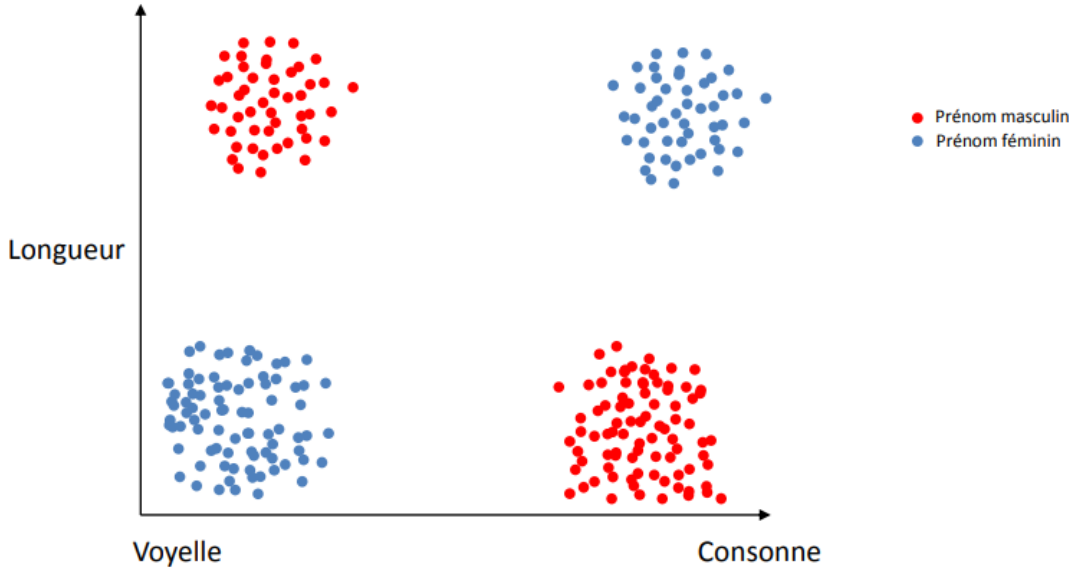
Witten & Frank (2005)

Perceptron multi-couches : structure

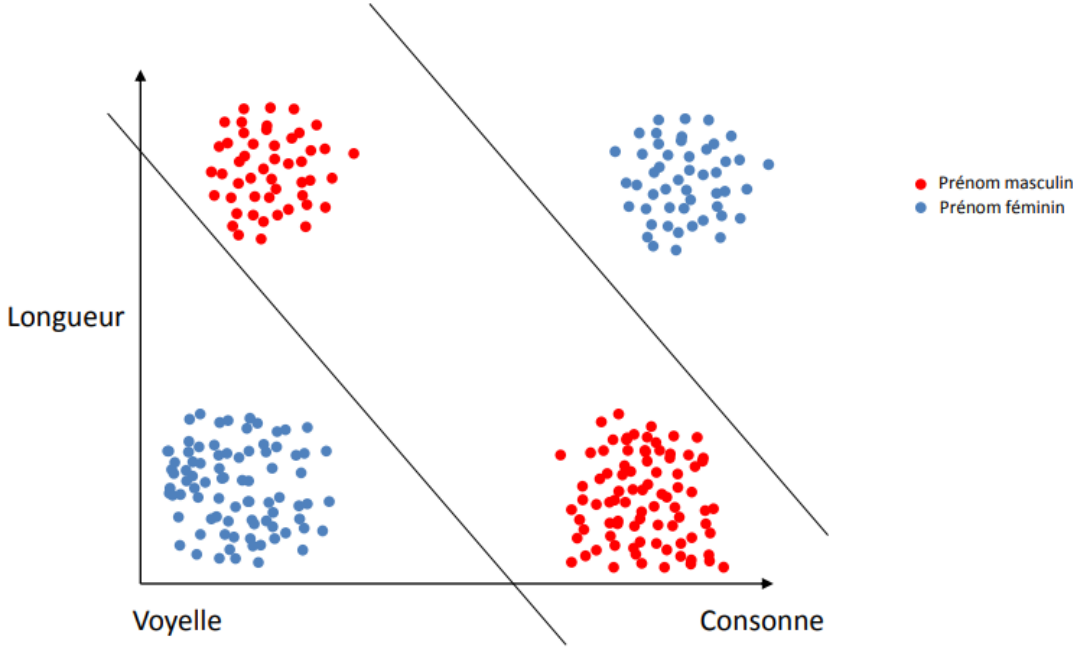
UNIVERSITÉ CÔTE D'AZUR

- Ensemble d'unités entrées/sorties connectées
- Poids associé à chaque connexion
- Couche d'entrée :
  - Un point d'entrée pour chaque attribut
- Couche cachée :
  - Pas de connexion directe à l'environnement (input/output)
  - Sorties peuvent être transmises à une autre couche cachée
  - Dernière couche cachée : sorties données à la couche de sortie
- Couche de sortie
  - Prédiction du réseau de neurones
- Permet de séparer ce qui n'est pas séparable linéairement

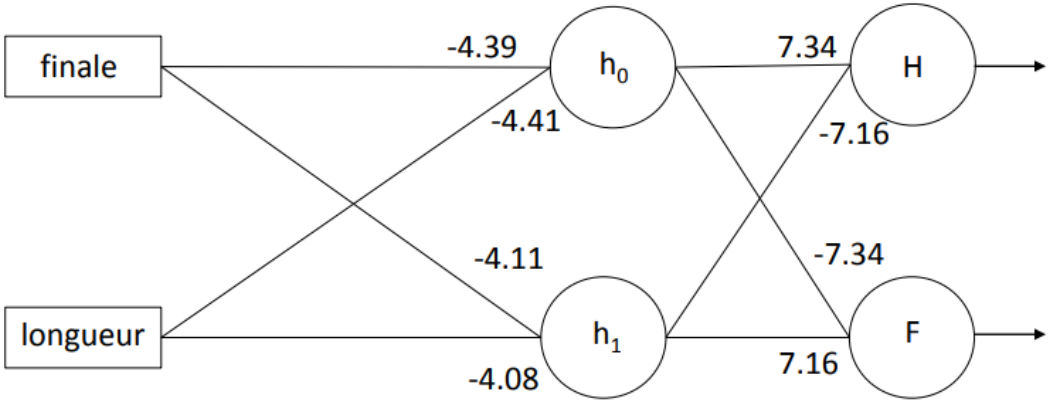
Perceptron multi-couches : exemple



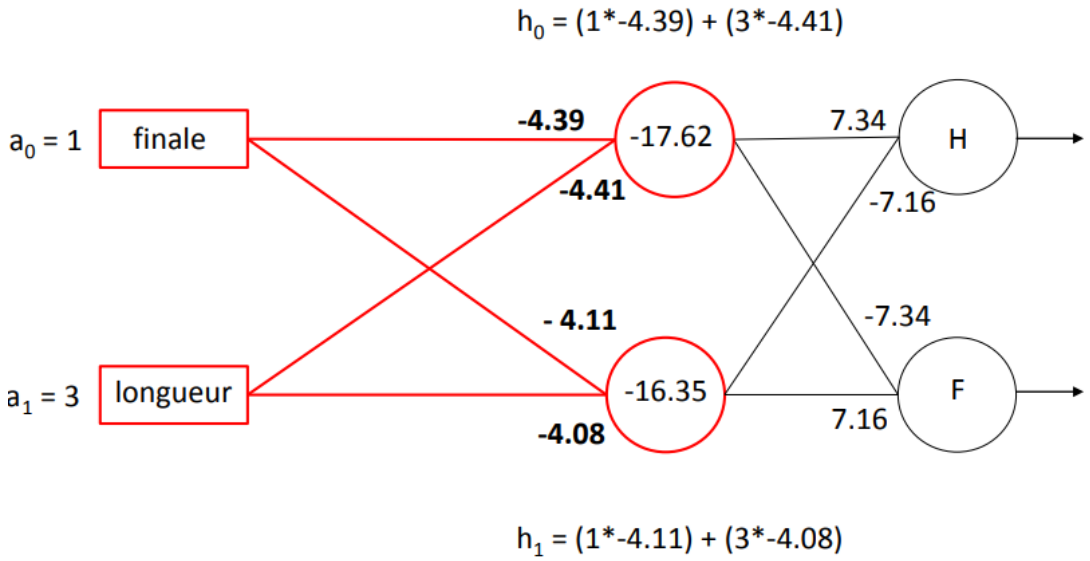
Perceptron multi-couches : exemple



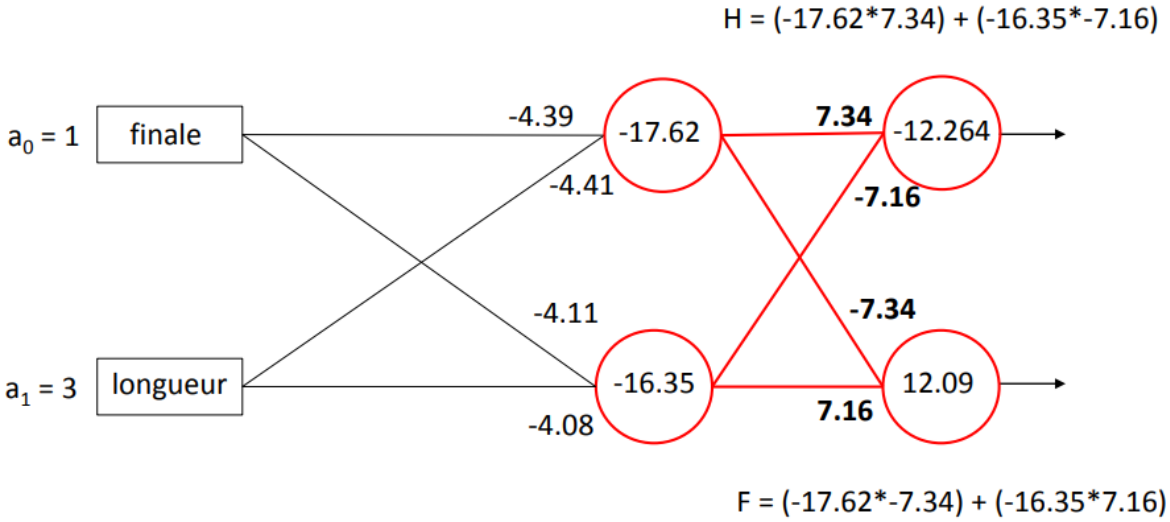
Perceptron multi-couches : exemple



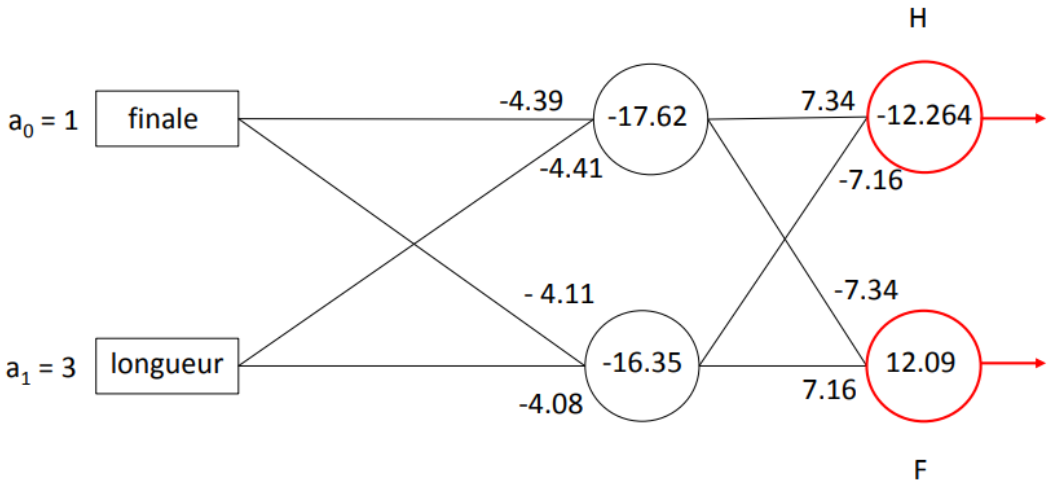
Perceptron multi-couches : exemple



Perceptron multi-couches : exemple



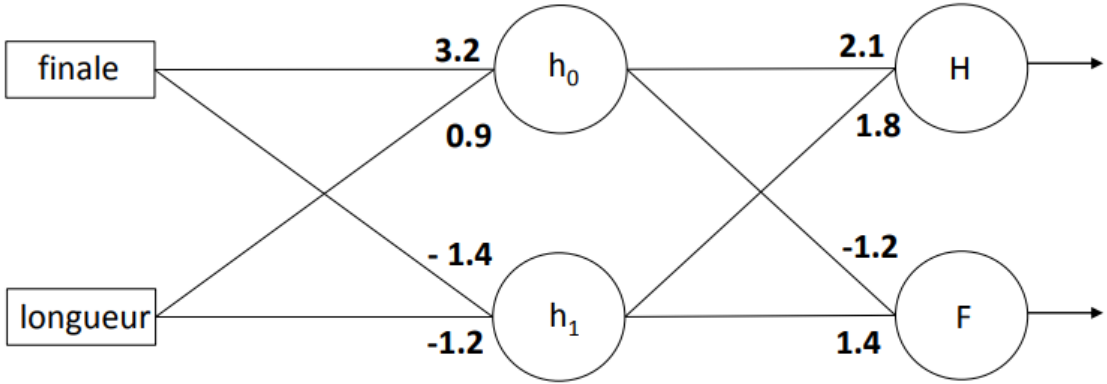
Perceptron multi-couches : exemple



Apprentissage (1)

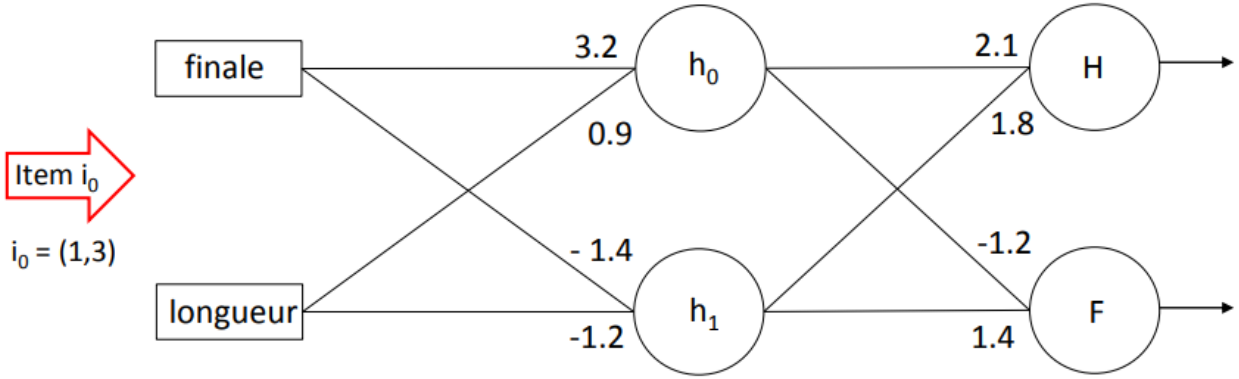
- But : déterminer les poids pour chaque connexion du réseau
  - En se basant sur les données d'apprentissage (entrées et sorties connues)
- Algorithme de base : rétro-propagation
  - Comparaison entre prédiction donnée par le réseau et résultat attendu
  - Ajustement des poids pour diminuer le taux d'erreur (de la couche de sortie à la couche d'entrée)

Apprentissage (2)



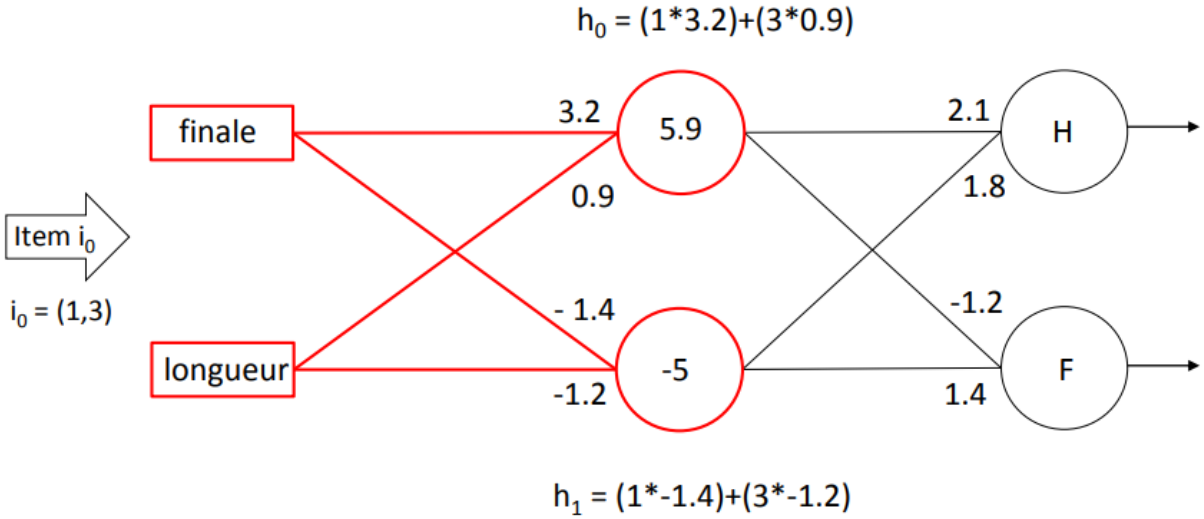
- Poids définis de manière aléatoire

Apprentissage (2)

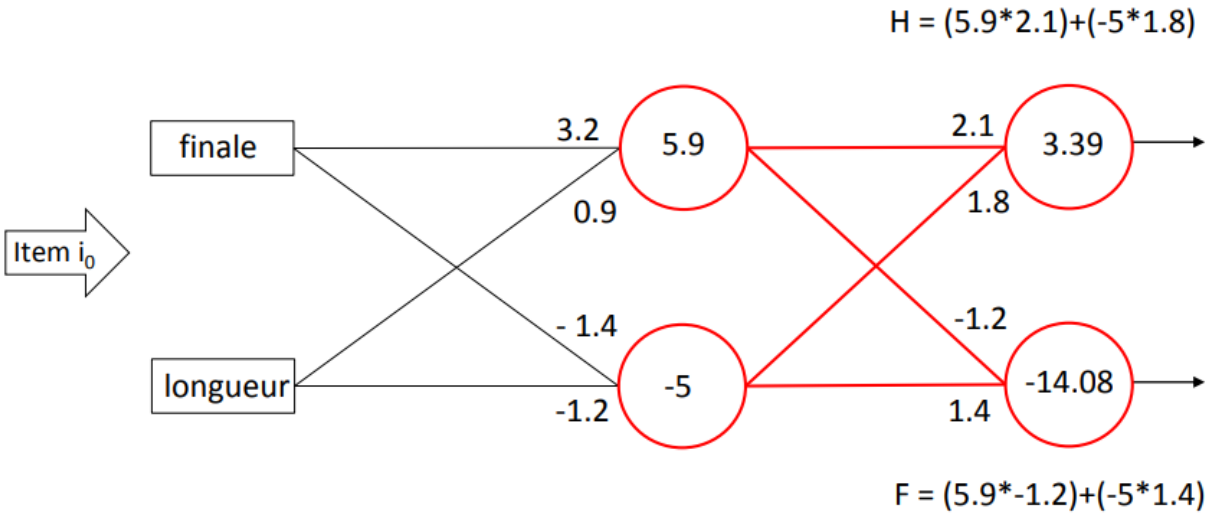


- Item donné en entrée

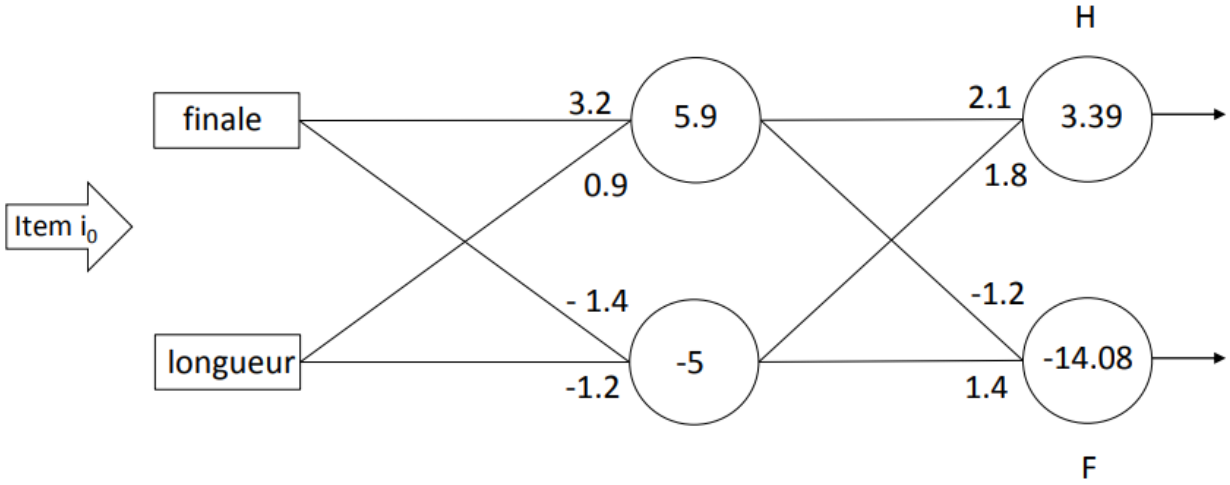
Apprentissage (2)



Apprentissage (2)



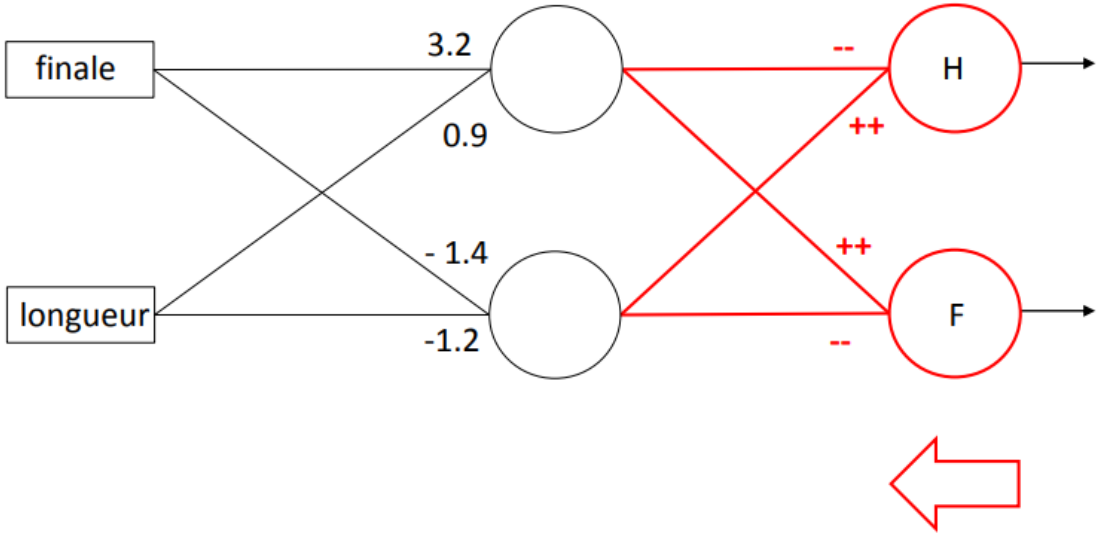
Apprentissage (2)



Résultat obtenu ⇨ (1,3) = H

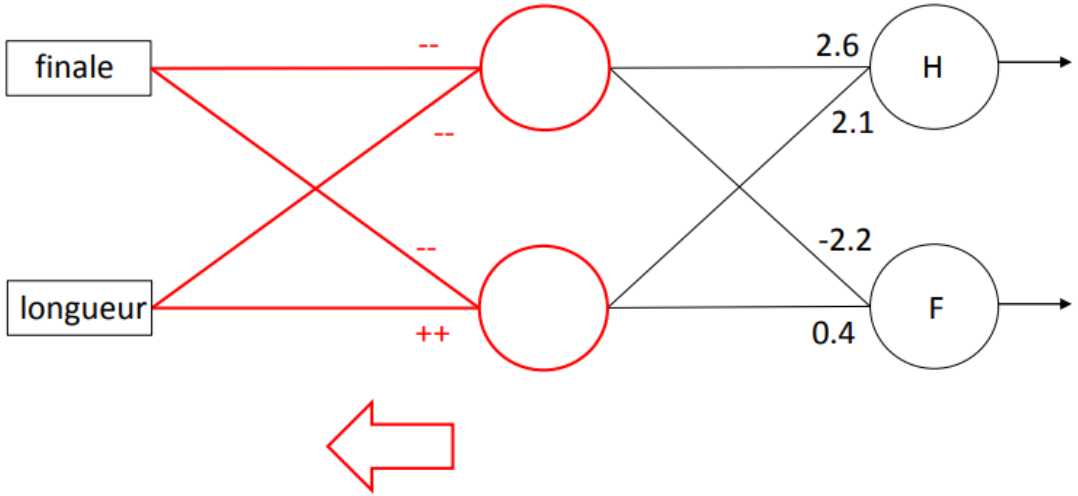
Valeur attendue ⇨ (1,3) = F

Apprentissage (2)



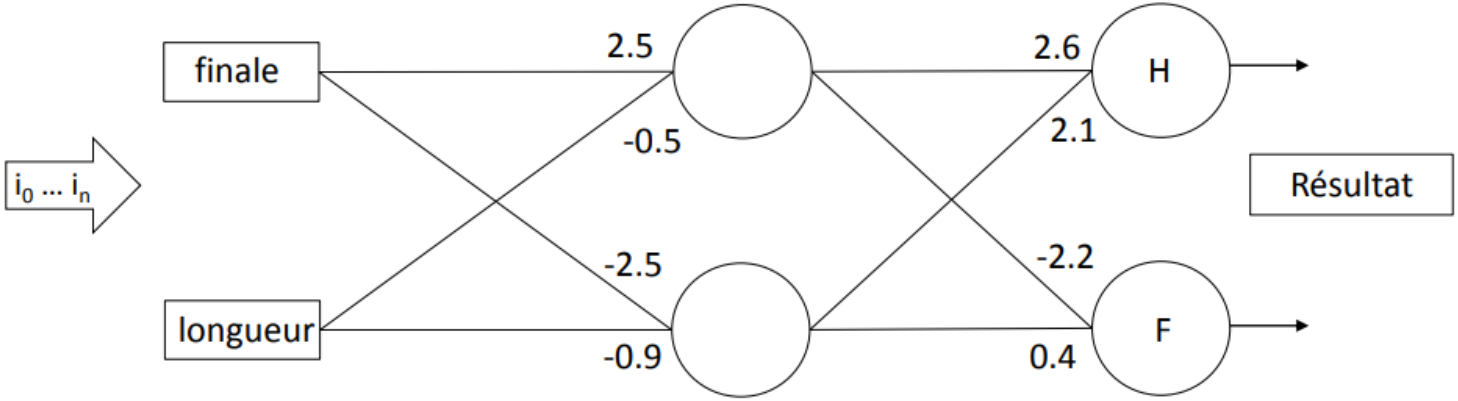
- Ajustement des poids

Apprentissage (2)



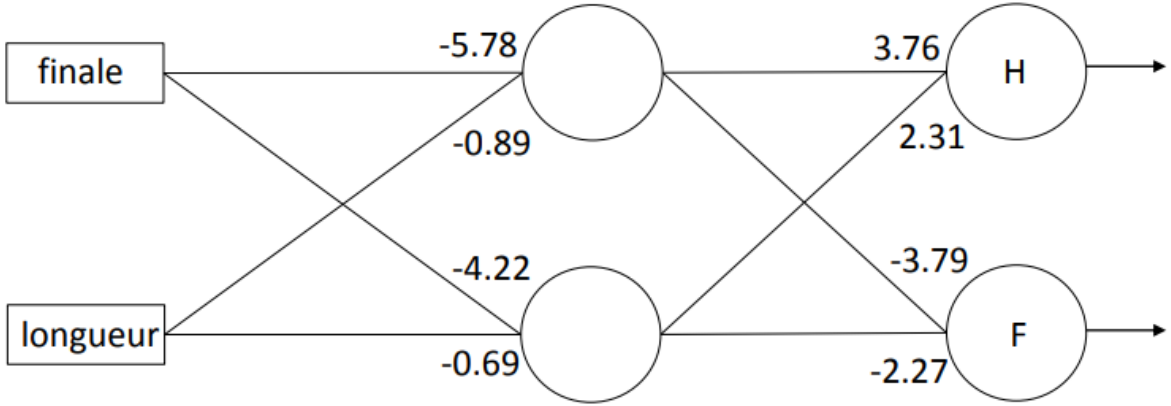
- Ajustement des poids

Apprentissage (2)



- Processus répété avec tous les items ($i_1 \dots i_n$) avec les nouveaux poids
- Stabilisation

Apprentissage (2)



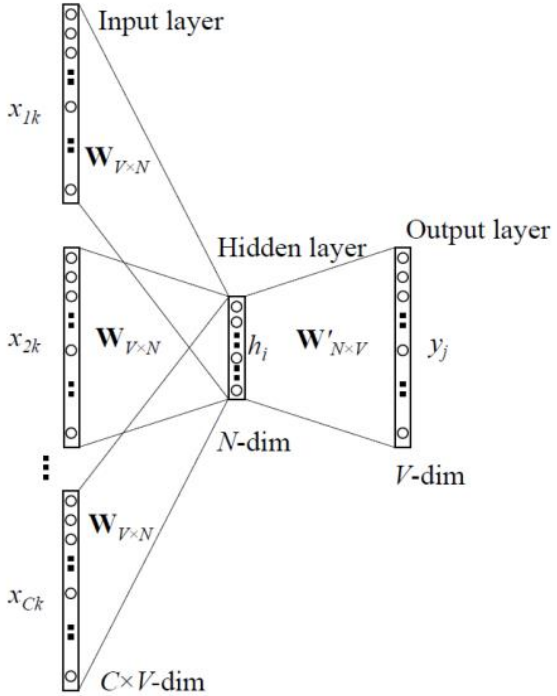
Les réseaux de neurones pour la sémantique distributionnelle

UNIVERSITÉ CÔTE D'AZUR

- Comment transformer l'analyse distributionnelle en un problème de classification supervisée ?
- Solution :
  - En entrée : une description du contexte d'un mot
  - En sortie : le mot qui apparaît dans ce contexte
- (ou l'inverse, en fait c'est pareil)
  - Le système va "apprendre" que certains mots sont prédits par les mêmes contextes
- Résultat
  - C'est l'état du modèle prédictif qui est intéressant, pas sa capacité à prédire
  - Comme c'est un réseau de neurones, on récupère à la fin les pondérations de chaque mot avec la couche cachée

Word2vec (vu dernière séance)

- Modèle le plus connu (Mikolov et al. 2013)
  - Pas le plus récent (cf. Bengio et al. 2003)
  - Mais un code disponible et très efficace
- Deux approches :
  - **CBOW (continuous bag of words)**
  - Skipgram



Déroulement

UNIVERSITÉ CÔTE D'AZUR

- Représentations des mots
  - Couches d'entrée et de sortie
  - Principe "1 hot" :
    - Soit V la taille du vocabulaire retenu
    - chaque mot est représenté par un vecteur de dimension V
    - Ce vecteur contient des 0 partout, sauf 1 pour la dimension correspondant à ce mot
  - Couche cachée de taille N arbitraire (e.g. 200, 500, 1000…)
- Balayage du corpus : extraction de situations
  - X mots de contexte -> mot-cible
  - Apprentissage des poids liant la couche de sortie à la couche cachée
- Au final, une matrice de taille VxN
  - Chaque mot de V est représenté par un vecteur de taille N
  - Les mots similaires sont activés par les mêmes neurones de la couche cachée et ont donc des vecteurs similaires

Les word embeddings

- On appelle word embeddings les représentations vectorielles des mots ainsi constituées
  - Matrice qui associe chacun des V mots aux N neurones de la couche cachée
- Intérêts :
  - Représentation compacte (N << V)
  - Par des vecteurs denses (pas de zéro)
  - "Renfermant" le comportement distributionnel des mots

Utilisation

- Usage direct pour la sémantique distributionnelle
  - Similarité (type cosinus) entre les embeddings
  - Compositionnalité des sens par addition des vecteurs
- Représentation vectorielle du lexique : word embeddings vs 1-hot
  - Continu et pas discret
  - Comparable
  - De dimension réduite
- Apparemment avantageux pour toute tâche de TAL par apprentissage qui nécessite une représentation du lexique
  - traduction, tagging, parsing, reconnaissance d'entités nommées, traitement de la parole, RI, EI, etc

UNIVERSITÉ CÔTE D'AZUR

Le Deep Learning

- L'apprentissage profond est avant tout l'utilisation de réseaux de neurones avec de nombreuses couches cachées
- L'idée était présente dès le début (années 1960), mais
  - la puissance de calcul ne suivait pas
  - certains problèmes théoriques persistaient concernant la rétro-propagation à travers ces couches multiples

UNIVERSITÉ CÔTE D'AZUR

Actuellement

- On peut construire des réseaux énormes
  - GoogleNet, pour la classification d'images (2014), possède 27 couches (dont la taille varie de quelques dizaines de neurones à plus de 1000)
- Qui tournent généralement sur des machines parallèles
  - E.g. AlphaGo (qui a battu les champions au jeu de go) utilise 1202 CPUs et 176 GPUs

Pourquoi toutes ces couches ?

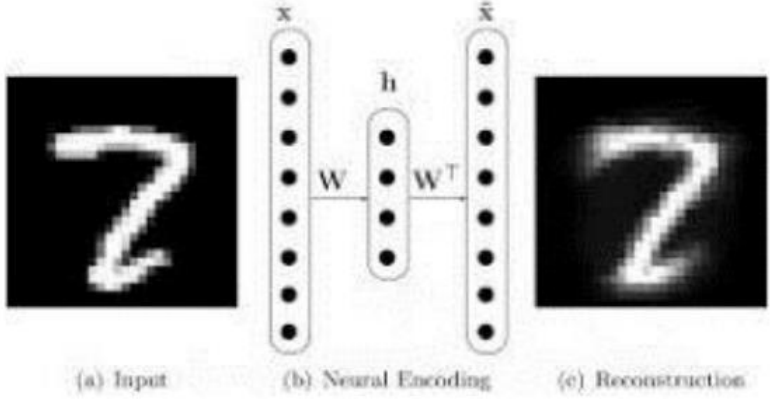
- Le principe est que chaque couche intermédiaire
  - Représente des régularités observées dans les couches inférieures
  - Améliore le traitement des couches supérieures
- Chaque couche cachée permet donc d'abstraire des informations
  - Comme le font les traits descriptifs définis pour représenter les entrées dans un système classique

Apprentissage non supervisé

- Les couches inférieures (côté input) d'un réseau profond sont généralement entraînées de façon non supervisée
  - i.e., sans recevoir d'information sur la nature des objets traités
  - Elles ne servent qu'à représenter des régularités observées dans les données
- Plusieurs techniques, dont la plus simple est l'auto-encodage

Auto-encodeur

- Réseau de neurones entraîné avec une couche de sortie qui est une copie conforme de la couche d'entrée
  - Son seul but est d'apprendre une (ou plusieurs) couches cachées qui stabilisent la redondance entre les caractéristiques des données
  - Une fois apprises, ces couches sont insérées dans le réseau profond



Intérêt des phases non supervisée

- Les données brutes non catégorisées sont plus faciles à trouver en grand nombre
- L'auto-encodage permet de réduire les dimensions des vecteurs d'entrée
  - En identifiant la redondance entre les features
    - En construisant des représentations synthétiques
    - Sans perdre d'information puisqu'on apprend au réseau à reproduire l'entrée

Le deep learning classique

Un réseau de neurones à plusieurs couches cachées et généralement d'architecture complexe

- Dont les couches de bas niveau ont été configurées sans supervision
  - Sur un très grand nombre de données brutes
- Dont les couches de haut niveau ont été configurées avec supervision
  - Sur un moins grand nombre de données annotées

Variante : Réseaux convolutifs

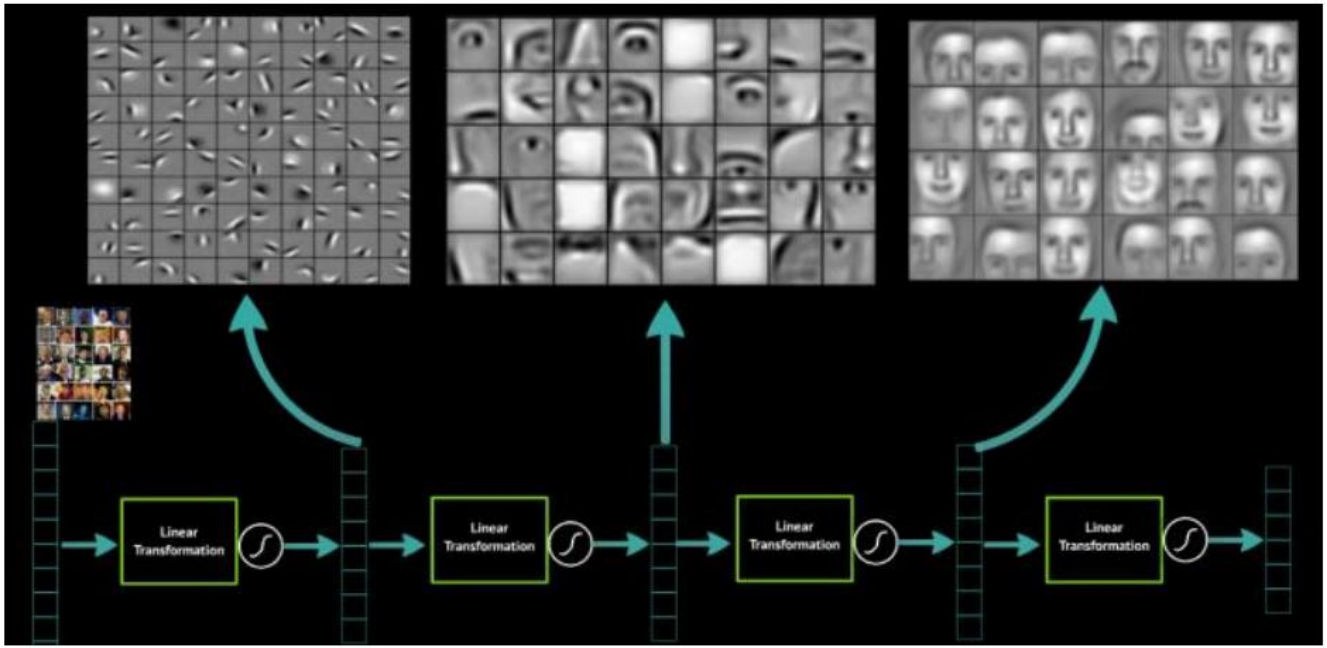
- Conçus initialement pour le traitement d'image, en s'inspirant des connaissances sur le cortex visuel
- Les couches intermédiaires ne traitent qu'une partie des données d'entrée
  - E.g. juste une partie d'une image à traiter
  - Avec un recouvrement entre elles, et un partage des paramètres
- Au final, les couches du milieu se concentrent sur certaines caractéristiques, et ce sont les couches supérieures qui synthétisent leurs résultats

UNIVERSITÉ CÔTE D'AZUR

Exemple : Histoire de chats

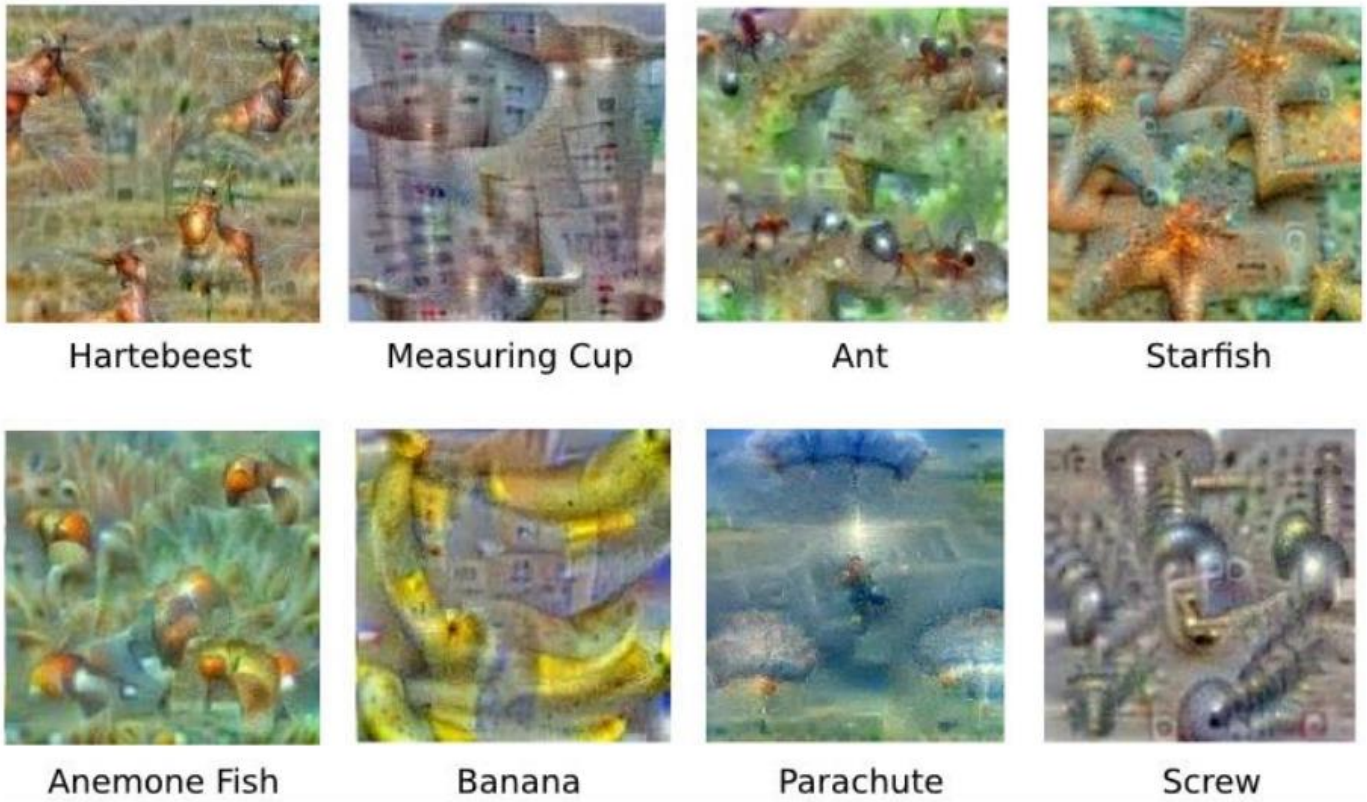
- Réseau de neurone pour la classification d'images
  - Pré-apprentissage sur des millions de photos de toute nature
  - Apprentissage supervisé sur quelques cas
  - Capable de « conceptualiser » très rapidement la notion de chat
- Et même capable de représenter une photo « prototypique » de chat
  - Il suffit d'activer le réseau à l'envers (feed-backward)
  - En activant le neurone de sortie correspondant au chat
  - Et on obtient les pixels de la couche d'entrée

Le point de départ : classification d'images



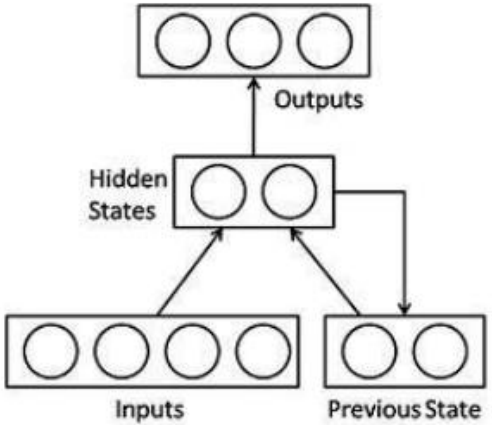
Emprunté à datarobot.org

Réseaux de neurones utilisés à l'envers (backward propagation)



Variante : Réseaux récurrents

- Avec des connexions qui forment des cycles dans le réseau
  - L'état interne du réseau après l'entrée n° i-1 est disponible lorsque l'entrée n° i est traitée
- Permet de gérer des modèles de langues
  - Probabilité d'avoir un mot/car après avoir trouvé les n mots/car précédents



Exemple : Génération de texte

- Modèles très utilisés pour la génération de texte
  - Corpus de départ pour entraîner le modèle ; quelques caractères en entrée puis le réseau « prédit » la suite

La philosophie du deep learning

- Objectif : faire TOUT apprendre par le réseau
  - A partir d'observations non supervisées des données – « Comme le font les humains » à qui on n'explique pas tout en détails
- Et donc se débarrasser de ces traits compliqués qu'on utilise pour l'apprentissage classique
  - Considérés comme une perte de temps et induisant des biais
  - E.g. préférer des n-grammes de caractères à des catégories syntaxiques ou sémantiques prédéfinies

La philosophie du deep learning

- Plus de feature engineering
  - Qui était pourtant une porte d'entrée des connaissances linguistiques dans les systèmes statistiques
- Plus de boîte grise
  - Très compliqué de voir ce qui se passe au milieu du système
    - Même d'expliquer pourquoi deux mots sont similaires dans des embeddings
- Un coup d'entrée très élevé
  - Besoin de puissance de calcul et de très grosses masses de données pour l'apprentissage non-supervisé