Machine Learning

Metz Numeric School

M2I

Florian AUBRY - 2024

Programme

- **Jour 1:**
  - Fondamentaux du Machine Learning
  - Pratique: Modèles supervisés
- **Jour 2**
  - Modèles avancés et optimisation
  - Pratique: Introduction aux modèles non supervisés

Programme

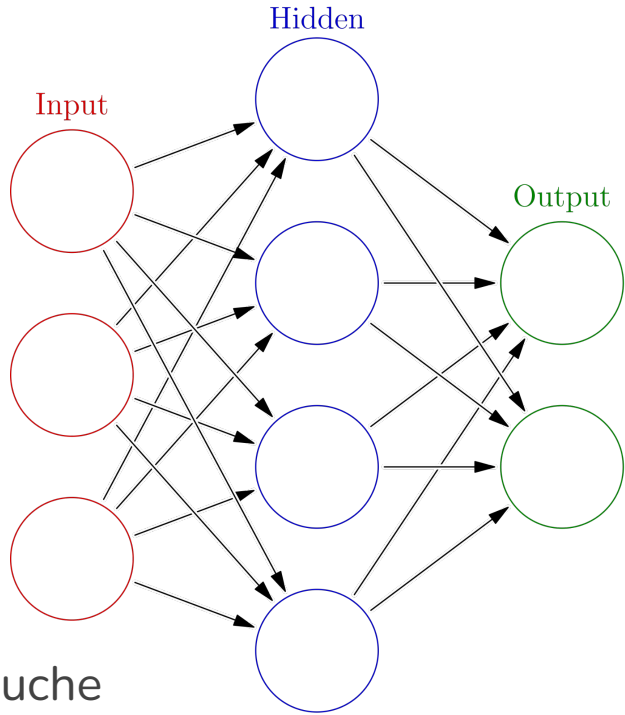
- **Jour 3**
  - Introduction au Deep Learning
  - Pratique: Application à l’analyse d’image et Traitement des langues
- **Jour 4**
  - Analyses avancées de données
  - Pratique: Combinaison de modèles en cas complexe

Sommaire

1. Réseaux Neuronaux
2. Application du ML à la NLP

Réseaux neuronaux

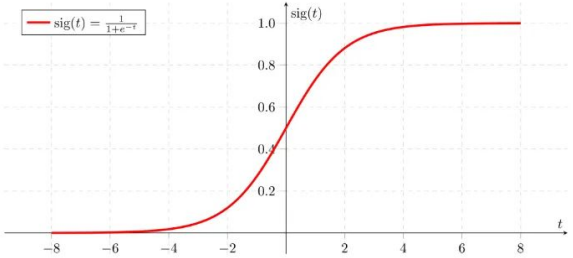
- Combinaison non linéaire de poids et paramètres
- Entrée du neurone
  - $e = w_1x_1 + \dots w_nx_n$
- Fonction d'activation interne pour la sortie
  - $s = f(e)$
- Chaque neurone prend en entrée la combinaison linéaire des résultats de la couche précédente

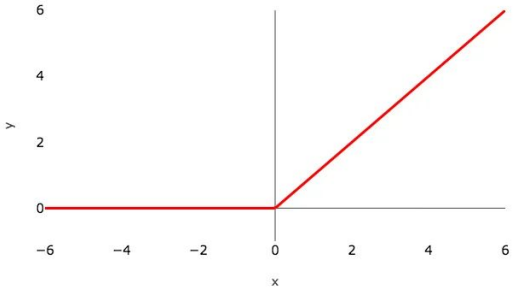


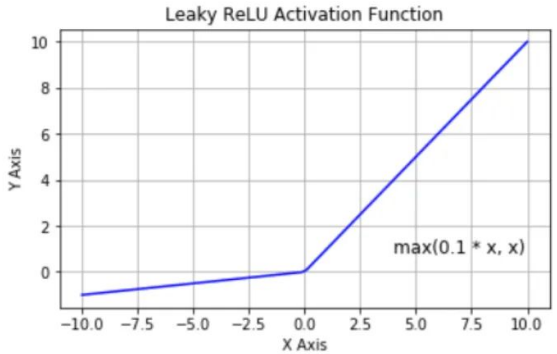
Réseaux neuronaux

Fonctions d’activations principales

- Sigmoïde: Sortie continue
- ReLU (+variantes): Très utile avec les CNN
- Leaky ReLU: Corrige le problème des valeurs négatives
- SotfMax



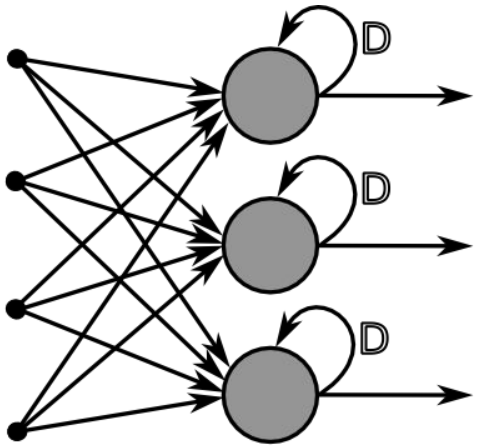




Réseaux neuronaux

Réseaux Récurrents (RNN)

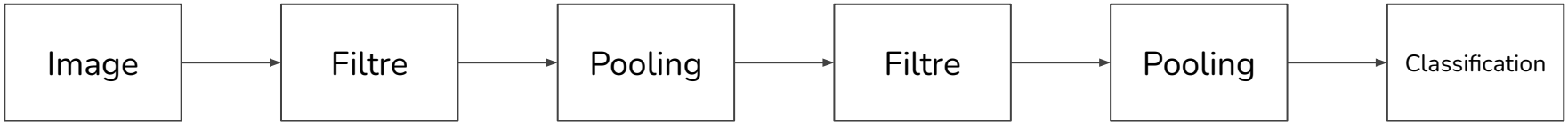
- “Mémoire” des entrées précédentes
- Traitement de données séquentielles
  - Séries temporelles
  - Texte, ...
- Problème: Perte de mémoire lointaine à l’entraînement
  - Introduction des LSTM (Long Short-Term Memory)
  - Introduction des GRU (Gated Recurrent Unit)



Réseaux neuronaux

Réseaux Convolutionnels (CNN)

- Alternance de:
  - filtres (convolutions) sur certaines parties de l'image
  - réduction de dimension (pooling)
- Couches finales:
  - réseau traditionnel pour la classification



Réseaux neuronaux

Réseaux Antagonistes Générateurs (GAN)

- Générateur:
  - Génère des données ressemblant aux exemples appris
- Discriminateur:
  - Détermine si les données sont générées ou réelles

Apprentissage:

- Le Générateur s’ajuste pour que le discriminateur ne fasse plus la différence
- Le Discriminateur s’ajuste pour mieux faire la différence

Réseaux neuronaux

Transformeurs

- Objectif:
  - Supprimer le problème de séquentialité et perte de mémoire des RNNs
- Mécanisme d’attention:
  - Permettre au modèle d’extraire les informations utiles
- Encodeur
  - Extraire les informations grâce à l’attention
- Décodeur
  - Génère une sortie séquentielle

Réseaux neuronaux

Transformeurs

- Couche FeedForward:
  - Traitement non linéaire de la sortie de l'encodeur
- Attention multi-tête:
  - Utilisation de plusieurs matrices d'attention pour gérer des perspectives complexes

ML et NLP

Machine Learning for Natural Language Processing (TAL)

- Prétraitements spécifiques:
  - Tokenization (phrase, mot, ...)
    - Isoler les éléments individuels du texte
  - Suppression des stop words (le, la, les, et, du, ...)
    - Enlever une information superflue
  - Lemmatisation:
    - Conserver la forme canonique petit**e**, petit**es** => petit
  - Stemmatisation
    - Extraire le radical du mot: coupur**e** => coupur

ML et NLP

```python
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

stopWords = set(stopwords.words('french')) # Stop Word list

text = "Le chien mange vite."
tokens = word_tokenize(text) # ['Le', 'chien', 'mange', 'vite', '.']

stemmer = PorterStemmer()
stemmer.stem("reference") # 'refer'

lemmatizer = WordNetLemmatizer()
lemmatizer.lemmatize("rocks")
```

ML et NLP

Machine Learning for Natural Language Processing (TAL)

- Uniformiser les vecteurs:
  - Phrases de différentes longueurs = vecteurs variables
  - Évaluer la répartition de taille
  - Choisir une taille fixe de vecteur
  - Tronquer les vecteurs trop longs
  - Padder: Ajouter des 0 à la fin du vecteur trop court

ML et NLP

Machine Learning for Natural Language Processing (TAL)

- Analyse fréquentielle TF-IDF
  - Évaluer l’importance d’un mot dans un document (TF - Term Frequency)
  - Mesure de l’importance du terme dans le document (IDF - Inverse Document Frequency)

ML et NLP

```python
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = ['This is the first document.', 'This is the second document.', 'and this is a thrid one', 'Is
this the first document?']
vectorizer = TfidfVectorizer()
X = vectorizer.fit(corpus)

vectorizer.get_feature_names_out()
# array(['and', 'document', 'first', 'is', 'one', 'second', 'the', 'this', 'thrid'], dtype=object)

X.transform(['I worked on my document']).toarray()
# array([[0., 1., 0., 0., 0., 0., 0., 0., 0.]])

X.transform(['I worked on my first document']).toarray()
# array([[0. , 0.62922751, 0.77722116, 0. , 0. ,0. , 0. , 0. , 0. ]])
```

ML et NLP

Rappel: Embeddings avec SpaCy

```
# python -m spacy download en_core_web_lg
import spacy

word_emb = spacy.load("en_core_web_lg")
sentences = [word_emb("The cat sat on the mat."), word_emb("The dog ate a croissant")]
sentence_list = [s.reshape(1,-1) for s in sentences] # Invert vector shapes in array
input_data = np.concatenate(sentence_list)
```

ML et NLP

Machine Learning for Natural Language Processing (TAL)

- **Part-Of-Speech Tagging** (Étiquetage-Morpho Syntaxique)
  - Analyser la grammaire d'une phrase
  - Fournir un tag correspondant à une catégorie grammaticale:
    - Nom, sujet, verbe, ...
  - Annotation plus large que la grammaire scolaire
    - Pour obtenir la liste:

```
import nltk
nltk.download('tagsets')
nltk.help.upenn_tagset()
```

ML et NLP

```
import nltk
nltk.download('punkt') # Tokenizer Model
text = word_tokenize("I ate an apple yesterday!")
nltk.pos_tag(text)

# [('I', 'PRP'), ('ate', 'VBP'), ('an', 'DT'), ('apple', 'NN'), ('yesterday', 'NN'), ('!', '.')]
```

Partie Pratique

Vous disposez d'un jeu de données représentant les revues d'utilisateurs Steam. L'objectif est de prédire si l'utilisateur va recommander ou non un jeu en fonction de son commentaire.

- Réalisez une étude préalable des données (nombre de mots, longueur des phrases, équilibrage des classes, ...)
- Effectuez une première phase de nettoyage manuelle des données ainsi que de normalisation (à vous de chercher différentes solutions)