

Que faire avec un neurone ?

Collonville Thomas

Version 0.1.0-SNAPSHOT, 20/11/2018

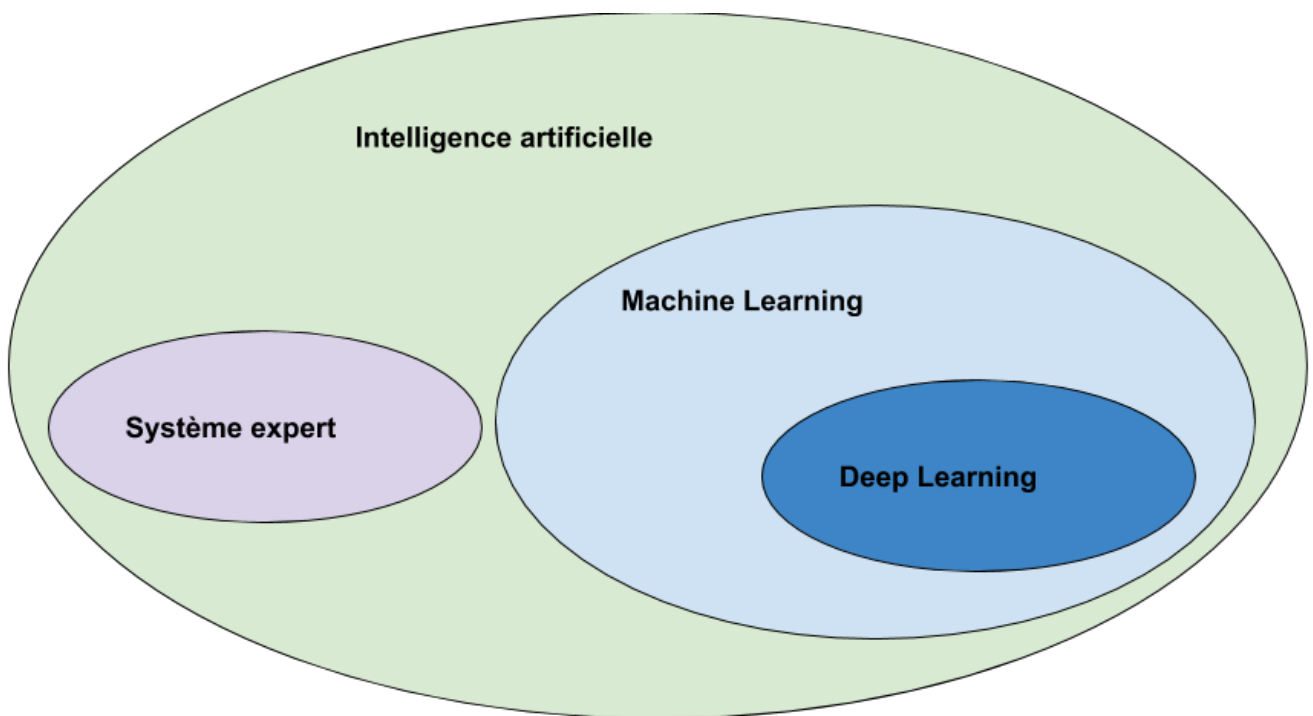
Plan

1. Sommaire	1
2. Qu'est ce que l'IA	1
3. Historique	1
3.1. Historique	2
4. Les modeles	2
4.1. Non-supervisé	2
4.2. Supervisé	2
5. Le neurone	2
5.1. Le neurone biologique	2
5.2. Constitution	3
5.3. Quelques nombres	3
5.4. Utilité	3
5.5. Le neurone formel	3
5.6. Le neurone formel	4
5.7. La fonction d'activation	4
5.8. Lineaire	4
5.9. Sigmoire	4
5.10. Limiteur	5
6. Processus	5
7. Exemples	5
7.1. Probleme de tri	6
7.2. Les données	6
7.3. Les données	6
7.4. Les données	6
7.5. La classification lineaire	7
7.6. La classification sigmoide	7
7.7. La regression lineaire	7
8. Conclusion	8

1. Sommaire

- Qu'est ce que l'IA
- Historique
- Le neurone
- Processus de traitement
- Exemple

2. Qu'est ce que l'IA



3. Historique

- 1940 : Alan TURING : Machine de Turing
- 1943 : Warren McCULLOCH & Walter PITTS Modèle formel de neurone.
- 1949 : Donald HEBB : Mémoire associative, premières règles d'apprentissage.
- 1960 : Franck ROSENBLATT et Bernard WIDROW, Perceptron et Adaline.
- 1980 : Stephen GROSSBERG et Teuvo KOHONEN Auto-organisation des réseaux et adaptation

3.1. Historique

- 1986 : Paul Smolenski : Machine de BOLTZMANN
- 1997: Deep Blue
- 2011 : Watson
- 2014 : LeCun Deep Learning
- 2015 : Alpha Go
- 2018 : Alexa

4. Les modeles

4.1. Non-supervisé

- Foret aléatoire
- Clustering et réduction de dimension
- SVD
- Analyse par composante principale (a noyau ou non)
- Règles

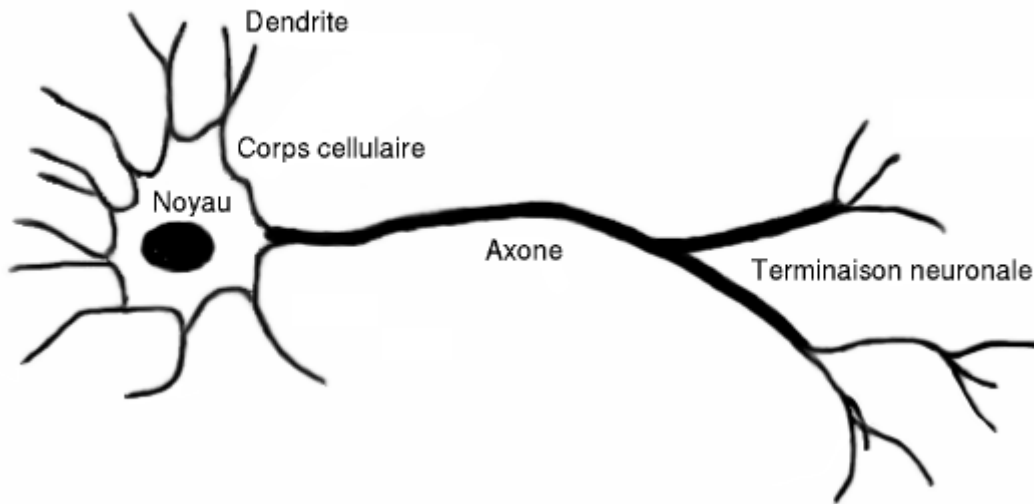
4.2. Supervisé

- Régression linéaire ou logistique
- Descente de gradient
- Régression polynomiale
- Séparation a vaste marge (SVM)
- Arbre de descision
- Classification linéaire ou non linéaire
- Réseau de neurone
- Naïve bayes

5. Le neurone

- Outil mathématique
- Formalisé par McCULLOCH et PITTS

5.1. Le neurone biologique



5.2. Constitution

- d'un noyau : le cœur de la cellule neuronale
- de dendrites permettant d'agréger les informations entrantes venant des synapses
- d'axones fournissant la réponse neuronale
- de synapses : interconnexion entre les axones et les dendrites permettant le transfert de l'influx nerveux

5.3. Quelques nombres

- 100 Milliards de neurones
- 10000 Synapses par neurone
- 10^{15} Synapses dans le cerveau humain

5.4. Utilité

- Mémoire et persistance des données dans le temps
- Réflexion, élaboration des idées, associer des concepts et des stratégies
- Sens, Analyse des données, traitements des sons, des images, du touché
- Construction d'une réponse moteur, l'équilibre, l'orientation, la marche, dextérité

5.5. Le neurone formel

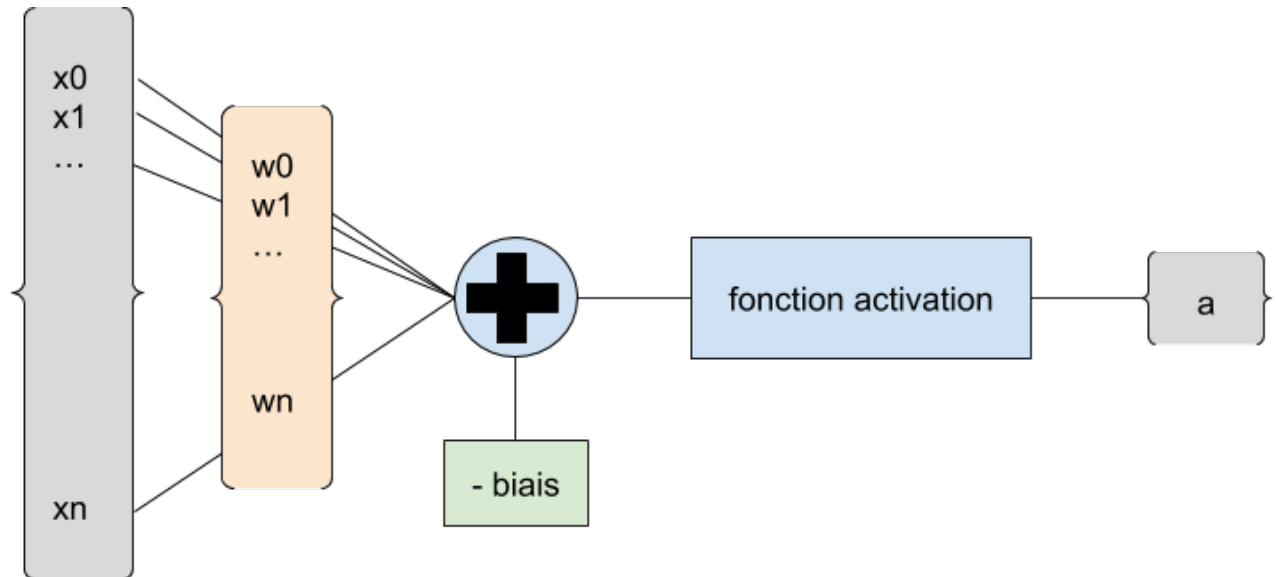
$$a = f(\sum_i^n (x_i \cdot w_i) - b)$$

$$a = f(W^T \cdot X - b)$$

- a la sortie du neurone
- x_i , le signal d'entrée

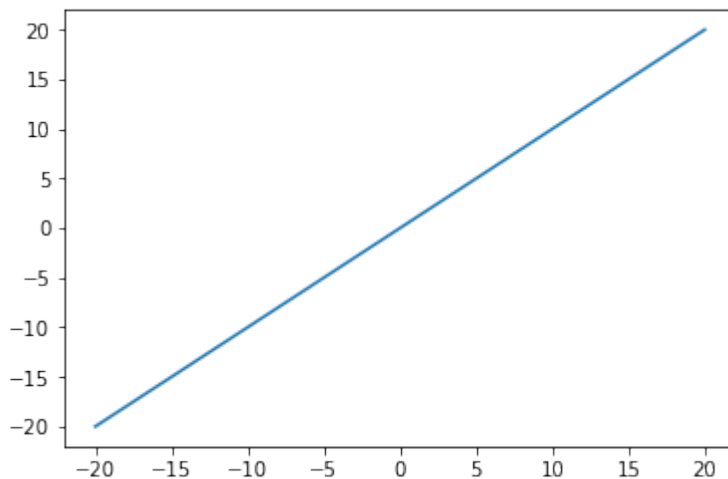
- w_i , le poid de ponderation
- biais, une constante de pondération
- f , la fonction d'activation

5.6. Le neurone formel

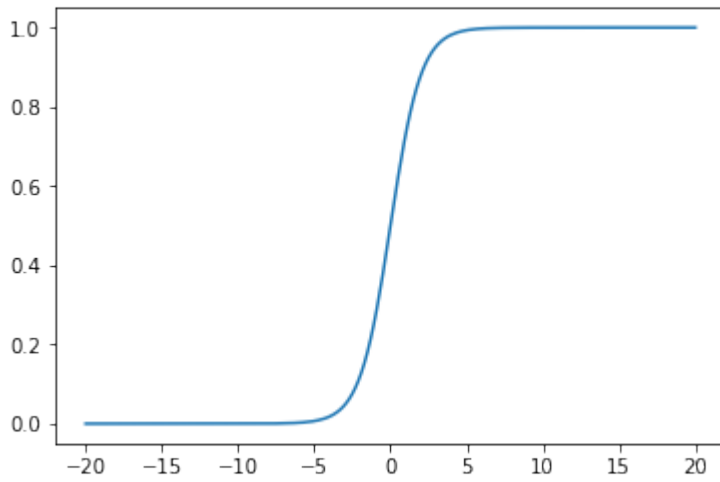


5.7. La fonction d'activation

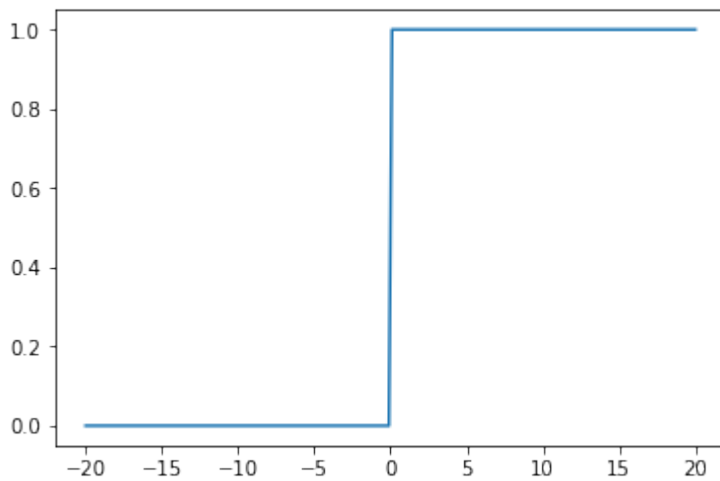
5.8. Lineaire



5.9. Sigmoire



5.10. Limiteur

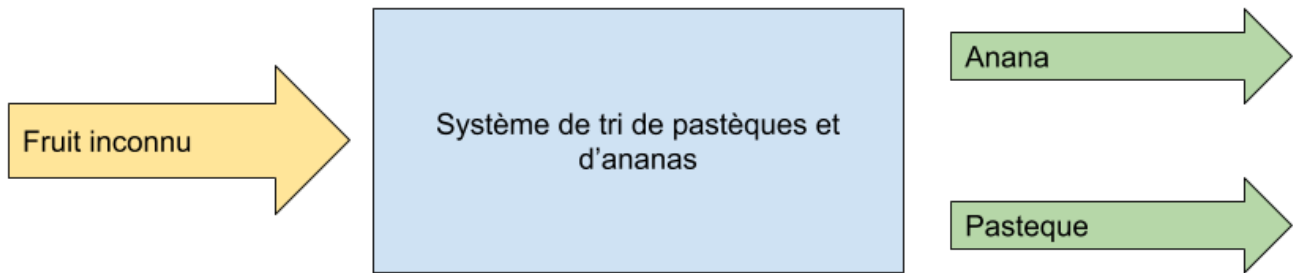


6. Processus

- Analyse du probleme
 - Nettoyage des données
 - Visualisation des données
 - Jeux de test
 - Jeux d'entraînement
- Definition d'un modele
- Apprentissage
- Mesure d'efficacité
- Mise en exploitation

7. Exemples

7.1. Probleme de tri



- a rugosité → 0 lisse a 1 rudeux
- la couleur → 0 bleu a 1 rouge
- la forme → 0 rond a 1 alongé
- le poid → 0 (20gr) à 1 (2000gr)

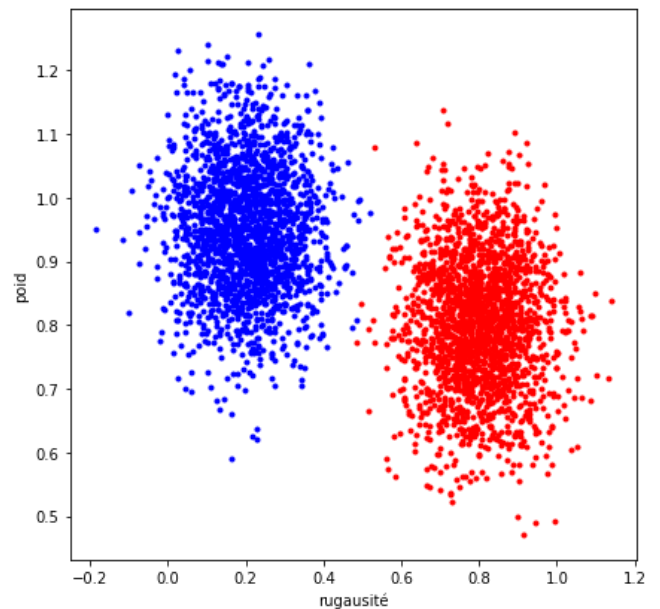
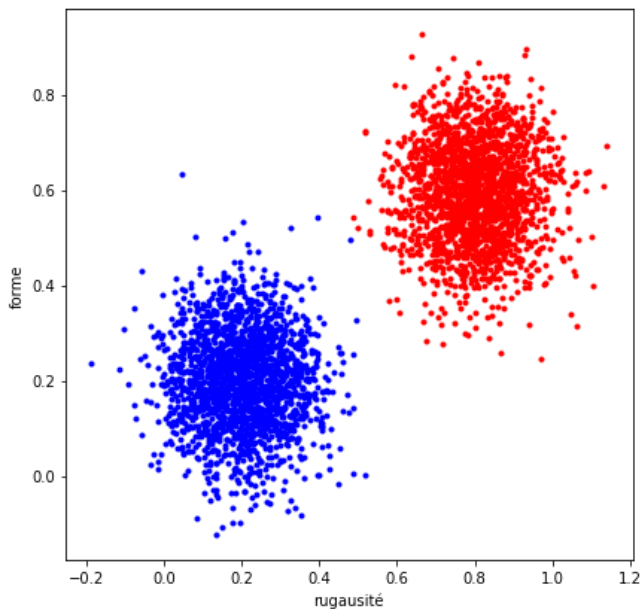
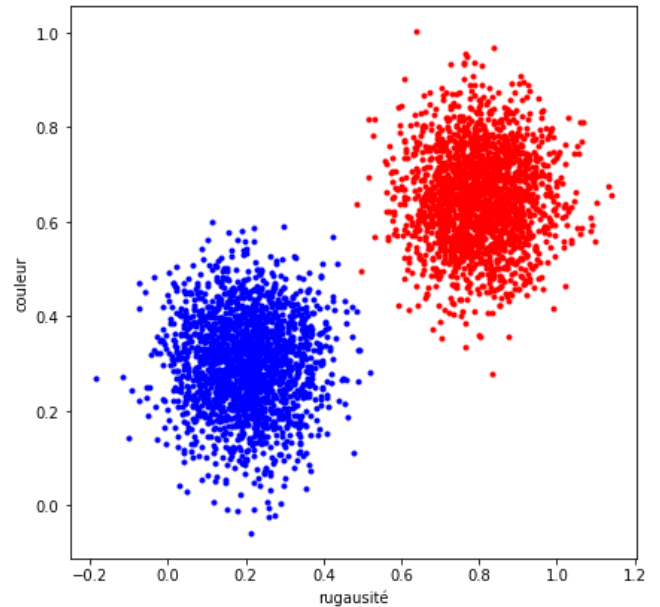
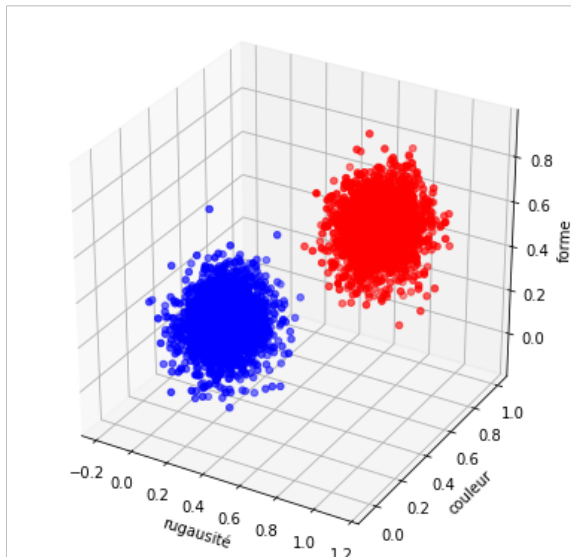
7.2. Les données

```
def generateSet(prototype,nbrEchantillon,coef):  
    rand_value=np.random.randn(len(prototype),len(prototype[0]))/coef  
    #print(rand_value)  
    rand_set=prototype+rand_value  
    if nbrEchantillon == 0 :  
        return prototype  
    else:  
        return np.concatenate((rand_set,generateSet(prototype,nbrEchantillon-1,coef)))
```

7.3. Les données

```
pasteque=np.array([[0.2, 0.3, 0.2, 0.95]])  
anana=np.array([[0.8, 0.65, 0.6, 0.8]])  
pasteques=generateSet(pasteque,1999,4)# -> pour separer les ensembles  
ananas=generateSet(anana,1999,4)
```

7.4. Les données



7.5. La classification lineaire

```
def neuroneLim(entre,W,biais):
    a=np.dot(entre,W.T)-biais
    #print("a neurone:",a)
    if a > 0:
        return 1
    return 0
```

7.6. La classification sigmoïde

7.7. La regression lineaire

8. Conclusion