INTELLIGENCE ARTIFICIELLE

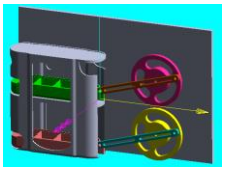
- Paramétrage d'une chaîne d'acquisition
- Algorithme, programme, langage de programmation
- Notions d'intelligence artificielle

ACTIVITE PRATIQUE

Durée : 4h

Objectifs de l'activité :

- Etablir le programme permettant la reconnaissance faciale
- Simuler son fonctionnement,
- Valider son fonctionnement réel et simulé



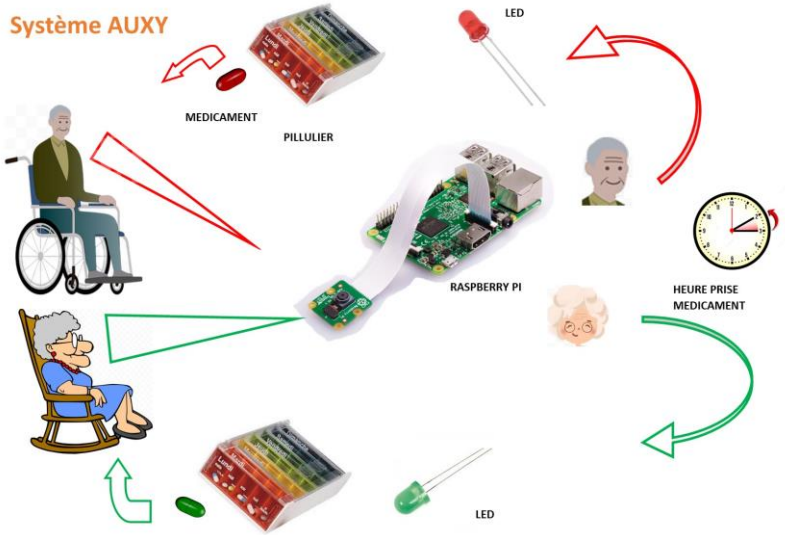
1. PRESENTATION

De nos jours, le nombre de personnes âgées ne cessent de croître. En France, en 2018, L'INSEE recense plus de six millions de personnes de plus de 75 ans et légèrement plus de deux millions ayant plus de 85 ans. Dans ce contexte, l'ouverture d'établissements d'hébergements pour personnes âgées et dépendantes progressent. Cependant, certains retraités souhaitent et ont le loisir de rester à domicile. Souvent, ils font l'objet d'un suivi médical par une infirmière ou une aide à domicile. Afin de rendre un minimum autonome ces personnes et surtout encadrer la prise journalière de leurs médicaments, de nombreux piluliers ont vu le jour.

Le but de l'activité proposée est l'étude de l'ouverture d'un pilulier d'un couple de retraités. Pour cela, on va faire appel à la reconnaissance faciale, en fonction de la personne reconnue (soit mamie, soit papi), le tiroir associé s'ouvrira afin d'éviter une confusion dans la prise de médicaments. Bien évidemment, le système étudié présente des failles, par exemple, il n'assure pas la prise certaine des médicaments par la personne reconnue. En fait, ce qui est recherché, c'est l'atout que peut représenter l'intelligence artificielle dans de telles situations.

L'étude comporte la reconnaissance faciale et l'ouverture du bon casier (matérialisé ici par une LED). Ce TP est une reprise d'une partie d'un projet de classe de terminale.

SYNOPTIQUE :



2. ANALYSE DU BESOIN

CAS D'UTILISATION

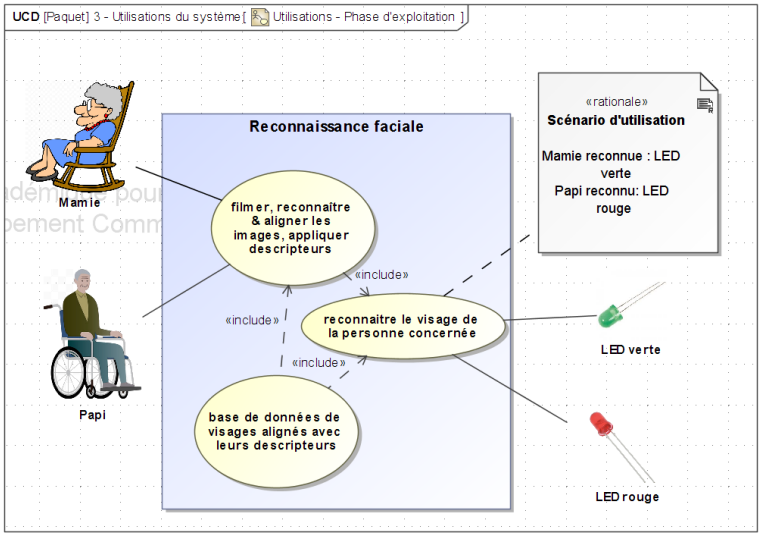


DIAGRAMME DES EXIGENCES

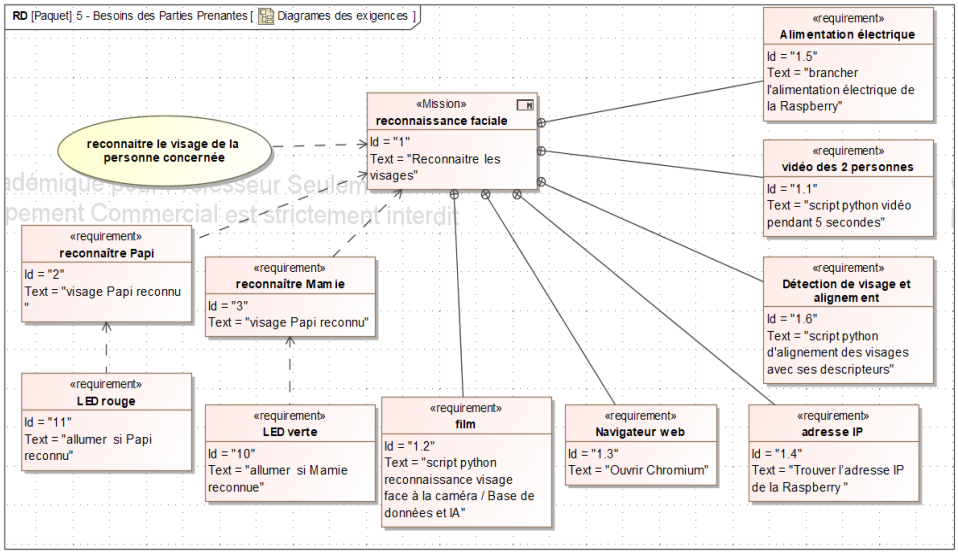
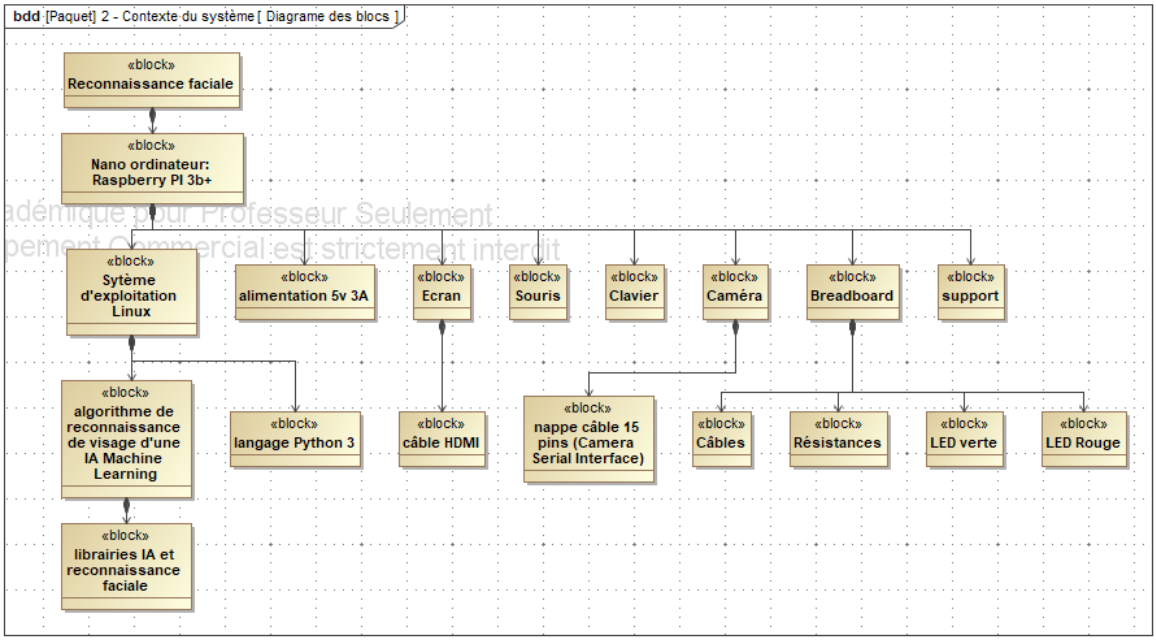


DIAGRAMME DE DEFINITIONS DES BLOCS



3. IDENTIFICATION DE LA BONNE PERSONNE : APPROCHE DE LA MACHINE LEARNING

Exigences : **Identifier** la bonne personne (papi ou mamie)

Problématique **:** Comment sont traitées par le nano ordinateur Raspberry PI, les informations issues de la caméra ?

Notre système de reconnaissance faciale est un exemple d'intelligence artificielle, une Machine Learning. La reconnaissance faciale est assurée par des programmes en python installés sur le système d'exploitation basé sur linux sur le Raspberry Pi (Annexe 1 Guidance Raspberry).

Expérimenter la reconnaissance faciale :

a. Enregistrer des vidéos de 5 secondes pour chaque personne à détecter :

- **Ouvrir** le LXTerminal dans la barre du haut de votre écran Raspberry
- **Taper** dans le terminal : cd /home/pi/Documents/3DFaceRecognitionOnRaspberryPI-master
- **Faire** entrée (le nom du dossier doit alors s'afficher en bleu) :
  ~ /Documents/3DFaceRecognitionOnRaspberryPI-master $

- **Taper** dans le terminal la commande suivante :
  python save_videos_from_webcam.py --name_of_person nom_de_la_personne
  (Exemple: python save_videos_from_webcam.py --name_of person PAPI)
- **Faire** entrée

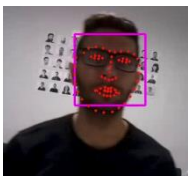
La caméra se met en route sans pour autant montrer le film : **se mettre** face à la caméra.

Arrêter au bout de 5 secondes en tapant CTRL C (attention de ne pas faire trop long)

Un film est sauvegardé dans le dossier **/home/pi/Documents/3DFaceRecognitionOnRaspberryPI-master/Database**.

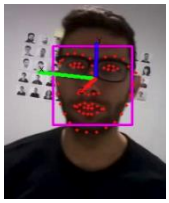
b. Détection de visage et alignement :

Principes :



Extraire des images du film

Localiser le visage du fond,



Détecter certains points sur le visage (points sur le contour des yeux, du nez, de la bouche…). On a utilisé pour cela un algorithme de Machine Learning existant permettant d'atteindre cet objectif en 1 milliseconde seulement. Il est basé sur l'apprentissage automatique (ou statistique) de l'intelligence artificielle par arbre de décision.

Déterminer l'orientation du visage

Procéder à une étape d'alignement du visage pour faciliter la tâche de reconnaissance

Générer la vue frontale

Éliminer les facteurs d'orientation et de luminosité, en temps réel sur un système embarqué comme le Raspberry PI : seules les images dont l'alignement a réussi seront envoyées à l'algorithme de reconnaissance.

On utilise pour cela le programme « python face_recognition_HAAR.py ». D'autres détecteurs sont plus efficaces que les descripteurs de "HAAR", mais nécessitent une puissance de calcul plus importante.

TRAVAIL à FAIRE

Les images reconnues comme correctement alignées parviennent à la dernière étape qui consiste à diviser l'image alignée en une grille de blocs. Un descripteur sera appliqué sur chaque bloc et l'image alignée sera représentée par la concaténation des différents descripteurs. Le descripteur de la nouvelle image sera alors comparé aux différents descripteurs de la base de données pour reconnaître la personne.

Expérimenter :

- **Lancer** le LXTerminal
- **Taper** dans le terminal : python face_recognition_HAAR.py
- **Faire** entrée
- **Attendre**

c. Lancer la reconnaissance faciale

Taper l'adresse IP de la Raspberry, par exemple python main.py --ip "127.0.0.1" -o 8000
Faire entrée
Ouvrir le navigateur chromium dans la Raspberry (ou sur ordinateur sur même réseau internet)
Taper et ouvrir l'URL par exemple **http:// 127.0.0.1:8000**

d. Tester la validité de l'IA

Se placer devant la caméra
Expérimenter et **valider** la reconnaissance faciale pour vos visages
Vérifier que le nom de la personne s'affiche au-dessus du rectangle entourant le visage.
Arrêter le programme en tapant dans le terminal CRTL C

e. Conclusion

Vous avez expérimenté une intelligence artificielle basée sur un algorithme de reconnaissance de visage implémentée sur un nano ordinateur. Cet algorithme permet d'identifier les visages en traitant les images à une cadence de 8~17 images par secondes.
Comparer les écarts entre les résultats de votre expérience et les niveaux attendus de ce système de reconnaissance faciale :

Conclure sur la pertinence du système

Conclure sur la pertinence du protocole expérimental

Conclure sur la validité de la solution au regard du cahier des charges

4. APPLICATION DE L'IA

Exigences : **Identifier** la bonne personne (élève1, élève2, ……..) et allumer la LED

Problématique : Comment programmer pour contrôler les LED en fonction de l'heure avec la Raspberry ?

Il faut modifier le programme principal main.py de la reconnaissance faciale. **Modifier** l'extrait du programme en python ci-dessous avec les noms des élèves à reconnaitre par l'IA et le créneau horaire de l'activité :

```
73.  if("papi" in identities) :
74.      print("Papi est ici")
75.      GPIO.setmode(GPIO.BCM) # BCM = emplacements GPIO
76.      GPIO.setup(5, GPIO.OUT) # LED Rouge sur GPIO 5 Pin n°29
77.      GPIO.setup(27, GPIO.OUT) # LED Verte sur GPIO 27 Pin n°13
78.      GPIO.output(5, True)
79.      GPIO.output(27, False)
80.      time.sleep(0.5)
81.      GPIO.cleanup()
82.  elif("mamie" in identities) :
83.      print("Mamie est ici")
84.      GPIO.setmode(GPIO.BCM) # BCM = emplacements GPIO
85.      GPIO.setup(5, GPIO.OUT) # LED Rouge sur GPIO 5 Pin n°29
86.      GPIO.setup(27, GPIO.OUT) # LED Verte sur GPIO 27 Pin n°13
87.      GPIO.output(5, False)
88.      GPIO.output(27, True)
89.      time.sleep(0.5)
90.      GPIO.cleanup()
```

Implanter votre programme dans le Raspberry :

Ouvrir le programme main.py fourni par le professeur sur clef USB (formatée en FAT 32) sur un poste travail classique

Modifier le code (espaces blancs en début de ligne, essentiel en python). **Enregistrer** vos modifications sur la clef USB

Insérer votre clef dans le Raspberry, puis ouvrir dans le gestionnaire de fichiers, puis valider.

Copier-coller, remplacer votre programme dans la Raspberry, dans le dossier **/home/pi/Documents/3DFaceRecognitionOnRaspberryPI-master, en suivant les consignes du fichier 1 guidance Raspberry**

Expérimenter le système IA avec reconnaissance faciale avec votre programme, suivre les instructions suivantes :

Ouvrir le LXTerminal dans la barre du haut de votre écran Raspberry

Taper dans le terminal : cd /home/pi/Documents/3DFaceRecognitionOnRaspberryPI-master

Faire entrée (le nom du dossier doit alors s'afficher en bleu) :

~ /Documents/3DFaceRecognitionOnRaspberryPI-master $

Taper l'adresse IP de la Raspberry, par exemple python main.py --ip "127.0.0.1" -o 8000
Faire entrée
Ouvrir le navigateur chromium dans la Raspberry (ou sur ordinateur sur même réseau internet)
Taper et ouvrir l'URL par exemple **http:// 127.0.0.1:8000**
Valider le fonctionnement du système.

Arrêter le programme en tapant Ctrl C dans le terminal.

Conclure sur cette action liée à une Intelligence artificielle avec reconnaissance faciale.

5. APPROCHE DE L'INTELLIGENCE ARTIFICIELLE

Notre système de reconnaissance faciale est un exemple de machine Learning. Yann Le Cun, chercheur en Intelligence Artificielle, définit l'IA comme un « ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux.

5.1 Situations d'usage de l'IA

Les domaines d'applications de l'IA sont de plus en plus nombreux, avec certains très controversés. Parmi les plus courants, on trouve notamment :

- La médecine, avec une aide au diagnostic de maladie et de préconisations médicamenteuses ;
- Le militaire, avec des systèmes d'aide à la décision et de commandement ;
- Le contrôle d'accès, avec des systèmes de reconnaissance faciale par exemple ; …

5.2 Les différents types d'IA

L'intelligence artificielle se décompose actuellement, en 2 entités imbriquées, le Machine Learning, et l'Apprentissage profond (Deep Learning)

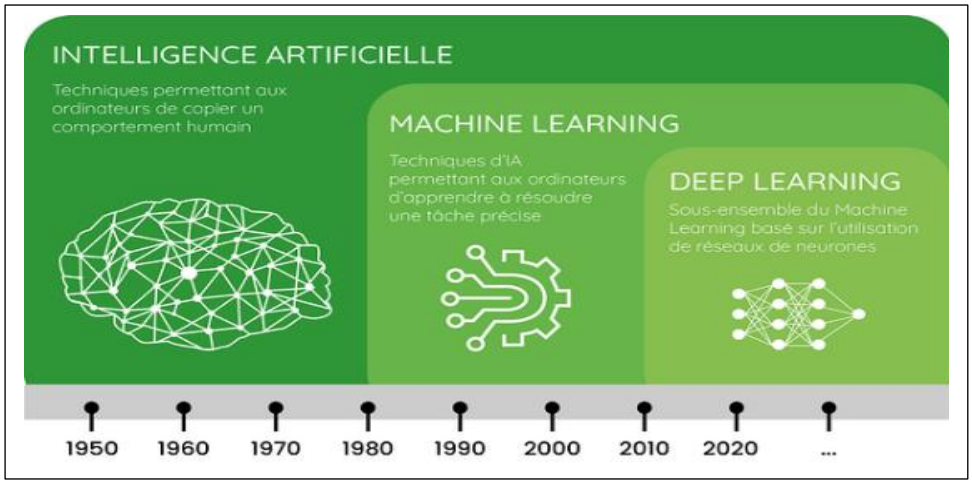


FIGURE 1 – l'IA est apparue dès les années 1950

L'Intelligence Artificielle mobilise une grande quantité d'informations classifiées, issues généralement d'une base de données. Le but de l'intelligence artificielle ici, est de prédire la classe la plus probable, après une phase d'apprentissage en 2 phases :

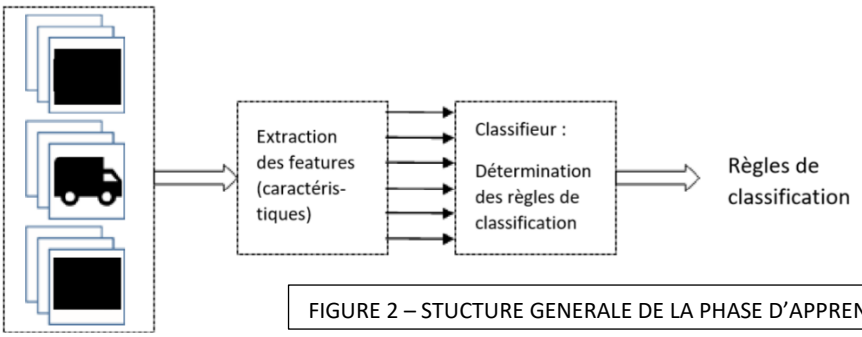


FIGURE 2 – STUCTURE GENERALE DE LA PHASE D'APPRENTISSAGE

Base de données classifiées

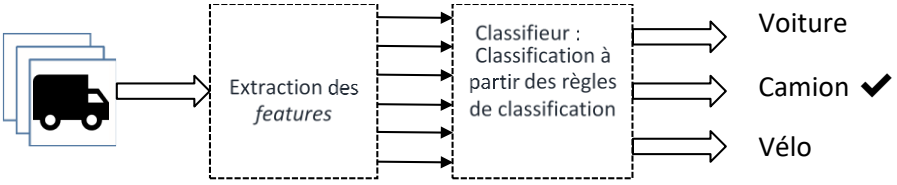


FIGURE 3 – STRUCTURE GENERALE DE LA PHASE D'UTILISATION

5.3 Machine Learning

Dans le cas du Machine Learning, le classifieur est basé sur des algorithmes d'analyse statistique des features (caractéristiques) extraits à partir de la donnée à identifier.
Un inconvénient du Machine Learning est le choix pertinent des features. En effet, certains features peuvent engendrer un biais dans la classification, et donc amener à une mauvaise prédiction de classe.

5.4 Deep Learning (apprentissage profond)

Dans le cas du Deep Learning, le classifieur est basé sur l'utilisation de réseaux de neurones, à l'image du cerveau humain.

FIGURE 4- NEURONE BIOLOGIQUE

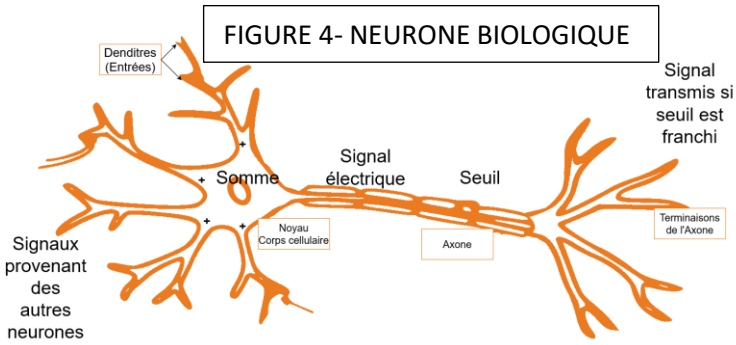
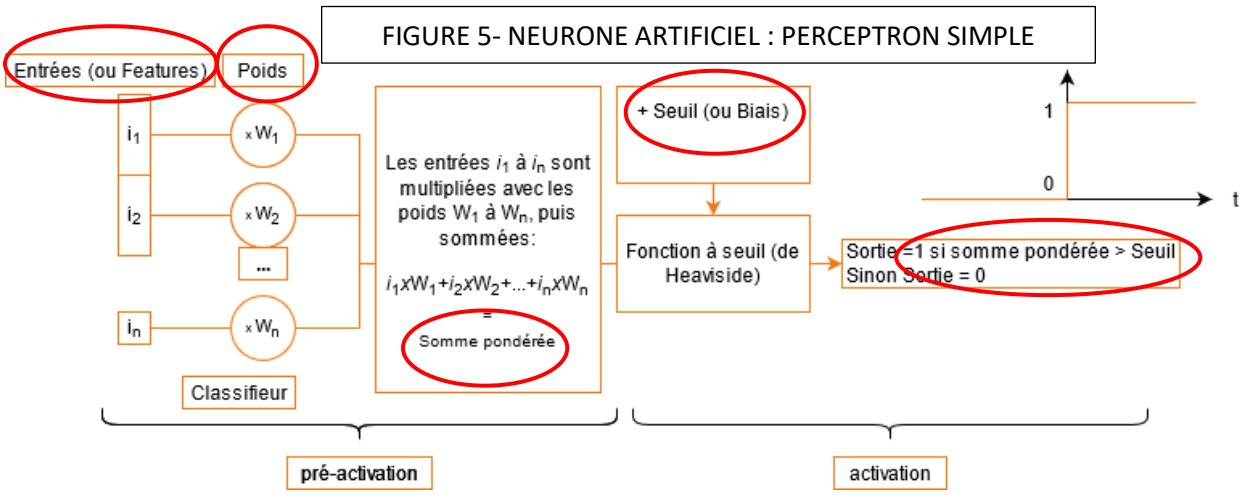


FIGURE 5- NEURONE ARTIFICIEL : PERCEPTRON SIMPLE



Lors de la phase d'apprentissage, le classifieur va adapter les liaisons entre les neurones (par l'intermédiaire des valeurs associées aux poids w_i et du biais) en les optimisant. Si un feature n'est pas très pertinent, son poids sera très faible à l'issue de la classification lors de l'apprentissage, tandis que si un feature est très pertinent, son poids sera relativement important. Les algorithmes usuels de Deep Learning sont :

- Les réseaux de neurones artificiels (ANN) ;
- Les réseaux de neurones convolutifs (CNN), très utilisés dans le traitement d'images ;
- Les réseaux de neurones récurrents, très utilisés pour l'analyse de texte.

6. COMMENT CODER UN PERCEPTRON SIMPLE : UN NEURONE ARTIFICIEL

Les parties de pré-activation et d'activation de ce perceptron ont été codée en Python :

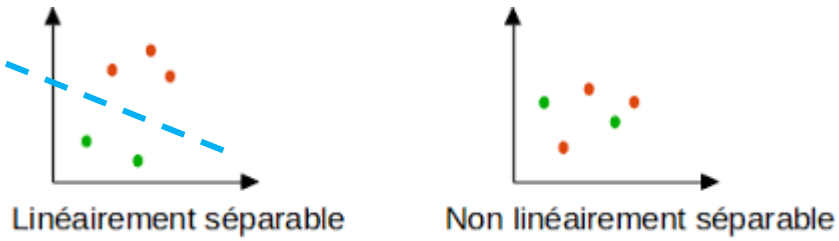
Identifier à partir des lignes de code ci-dessous, le nom de la fonction1 à l'aide de la figure 5

```
1.  def fonction1(features:list, poids:list, biais:int):      #Les arguments de la fonction sont des listes de feature, poids et un entier biais
2.      poids_somme = 0                                      # initialisation
3.      for feature in features:                             # boucle qui va chercher chaque feature dans la liste features
4.          for weight in weights:                           # boucle qui va chercher chaque weight dans la liste weights
5.              poids_somme += feature * poids               # somme pondérée des produits feature*poids
6.      poids_somme += biais                                 # somme pondérée des produits + biais
7.      return poids_somme                                   # La fonction retourne : somme pondérée des produits feature*poids
```

Identifier à partir des lignes de code ci-dessous, le nom de la fonction2 à l'aide de la figure 5

```
1.  def fonction2(valeur:int):      #L'argument de la fonction est valeur = somme pondérée (un entier)
2.      seuil = 0
3.                                  #  etat du neurone : sortie y
4.      est_actif = bool            # est_actif est une valeur booléenne (vrai ou faux)
5.      if valeur >= seuil:
6.          est_actif = True
7.      else:
8.          est_actif = False
9.      return int(est_actif)       # Quelle valeur retourne la fonction :  état de la sortie du neurone ?
```

D'un ensemble de données, le perceptron (reproduisant le neurone) calcule les coefficients a et b de la droite (y = somme pondérée des produits + biais ou y=ax + b) et donne en sortie un état actif ou inactif en fonction d'un seuil. Cette droite peut séparer des données linéairement séparables (par une droite).

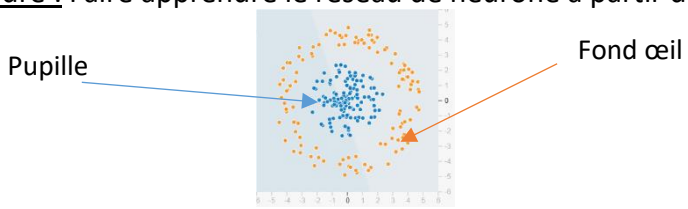


Imaginer que ces populations de points représentent une partie de visage, par exemple un sourcil et la peau.

On a donc vu la modélisation d'un perceptron en python. Le perceptron est l'un des tout premiers algorithmes de Machine Learning, et le réseau de neurones artificiels le plus simple. En mettant plusieurs perceptrons ensembles, on crée un réseau de neurones, un modèle qui va permettre de faire des prédictions à partir d'un ensemble de données initiales. Ces calculs sont possibles sur votre propre ordinateur qui peut faire tourner des programmes en python.

7. SIMULATION D'UN RESEAU DE NEURONES

Problème à résoudre : Faire apprendre le réseau de neurone à partir des données :

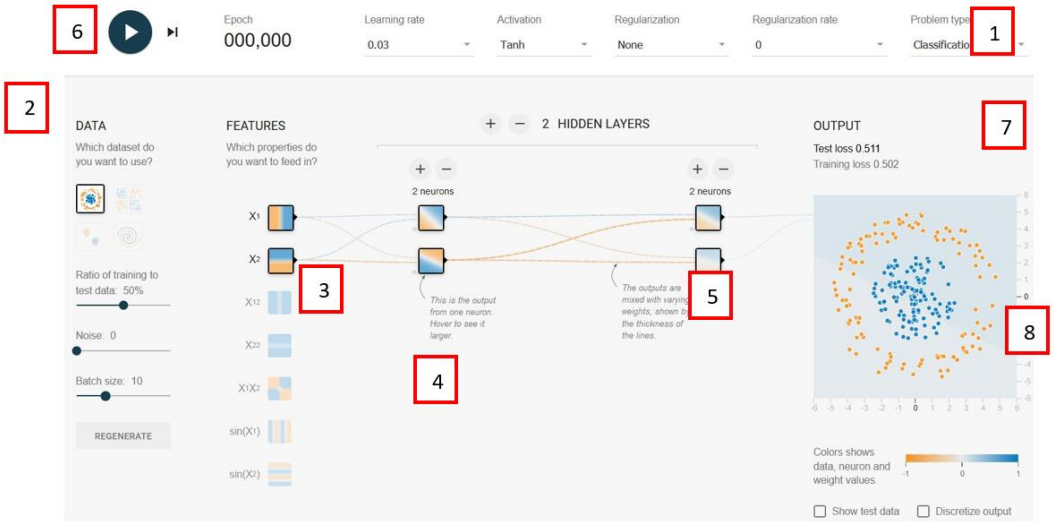


Qu'est-ce qu'un réseau neuronal ? Il s'obtient en accumulant plusieurs couches de neurones (Hydden Layers : couches cachées de neurones). Elles se transforment à travers toutes les couches et la dernière couche donne en sortie une prédiction sur ces données, par exemple détecter s'il y a une pupille dans un œil. Le réseau de neurone constitue ainsi une fonction paramétrée par ces nombreux coefficients (on parle de « poids ») et c'est le choix de ces poids qui définit le traitement effectué.

Simuler avec le logiciel en ligne TensorFlow (cliquer ici): permet de constituer des réseaux de neurones artificiels et de tester leurs réponses pour différents types de problèmes et ce, sur différents types de données

Choisir (1) Classification dans Problem type, (2) des populations de points en cercle dans DATA, **faire varier** (3) les features (propriétés des entrées), le nombre de de neurones et de couches neuronales (4) et (5), puis **simuler** (6) jusqu'à stabilisation en temps (7) et modèle de sortie (8). **Conclure** sur votre configuration du réseau neuronal pour résoudre le problème initial

Jouer avec les neurones de la machine



8. EXPERIMENTER L'APPRENTISSAGE D'UN RESEAU DE NEURONES

Expérimenter comment un réseau de neurones peut apprendre à reconnaître les dessins en cliquant sur « C'est parti » dans https://quickdraw.withgoogle.com/#

9. ALLER PLUS LOIN SUR L'IA

9.1 Voir "c'est quoi l'intelligence artificielle ? " https://www.youtube.com/watch?v=yQLmgw3rClM

Justifier en quoi le système de reconnaissance faciale décrit en début d'activité est une intelligence artificielle

9.2 Voir la vidéo Qui a peur de l'IA ? https://www.youtube.com/watch?v=mkfXdzA6B_c

Communiquer à l'écrit et avec le numérique

Décrire en quoi cela vous questionne