

Luminobellule : Vers un programme Python

Objectif : Réaliser un programme Python simulant l'étude du luminobellule.

I. Les données

On introduit une liste de listes représentant nos 40 cartes :

```
data = [[4.2, 2.6, 1], [4.4, 3.25, 1], [3.8, 3.3, 1], [4.7, 2.8, 1], [4.6, 3.45, 1],
        [3.5, 3.8, 1], [4.5, 3.2, 1], [3.9, 3.15, 1], [4.7, 2.9, 1], [4.1, 2.9, 1],
        [4.9, 3.0, 1], [4.1, 3.35, 1], [4.1, 2.7, 1], [4.3, 3.35, 1], [3.8, 3.35, 1],
        [4.6, 2.85, 1], [3.75, 3.0, 1], [4.5, 2.55, 1], [3.9, 2.65, 1], [4.5, 1.4, 1],
        [3.2, 2.9, 0], [3.1, 3.5, 0], [3.0, 2.9, 0], [3.1, 2.25, 0], [3.8, 1.8, 0],
        [3.1, 2.85, 0], [3.6, 2.6, 0], [3.0, 2.3, 0], [3.1, 2.4, 0], [3.7, 2.2, 0],
        [3.0, 2.4, 0], [3.2, 3.3, 0], [3, 3.9, 0], [3.1, 2.85, 0], [3.2, 2.2, 0],
        [3.4, 2.4, 0], [3.4, 2.65, 0], [3.1, 2.05, 0], [3.5, 2.55, 0], [3.5, 2.35, 0]]
```

On peut ensuite accéder à une carte ou à une donnée sur une carte comme ceci :

```
carte = data[0]
print(carte)
print(carte[0])
```

```
[4.2, 2.6, 1]
```

```
4.2
```

II. L'entraînement

Compléter en Python ou en Français le programme d'entraînement suivant :

```
import random
```

```
#Calcul pour prediction :
```

```
def prediction(X, Y, w1, w2, b):
```

```
#Ajustement des poids en fonction de l'erreur :
```

```
def ajust(pas, X, Y, w1, w2, b, E):
```

```
# Initialisation du pas et des poids :
```

```
pas = 0.1
```

```
w1 = 5
```

```
w2 = 2
```

```
b = -14
```

```
#====Entrainement:====
```

```
#Selection de 20 cartes au hasard:
```

```
select = random.sample(data, 20)
```

```
for carte in select:
```

```
    X = carte[0]
```

```
    Y = carte[1]
```

```
    color = carte[2]
```

```
    predict = prediction(X, Y, w1, w2, b)
```

```
    #Calcul de l'erreur:
```

```
    E = color - predict
```

```
    w1, w2, b = ajust(pas, X, Y, w1, w2, b, E)
```

```
print(w1, w2, b)
```