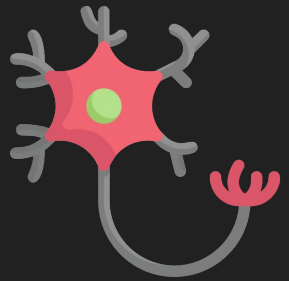
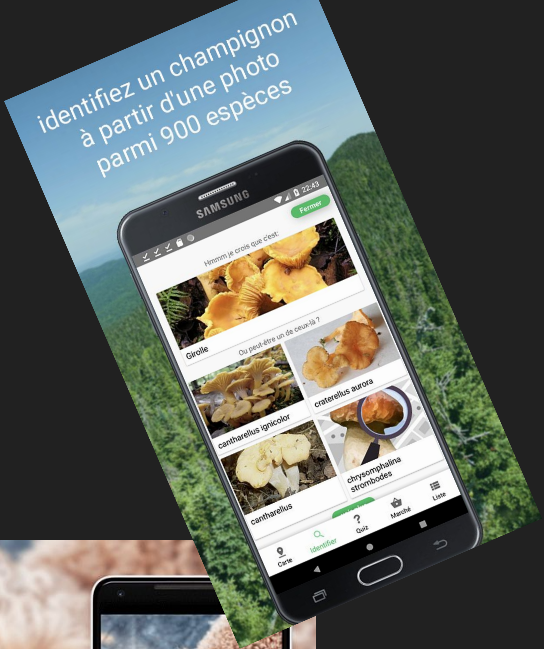
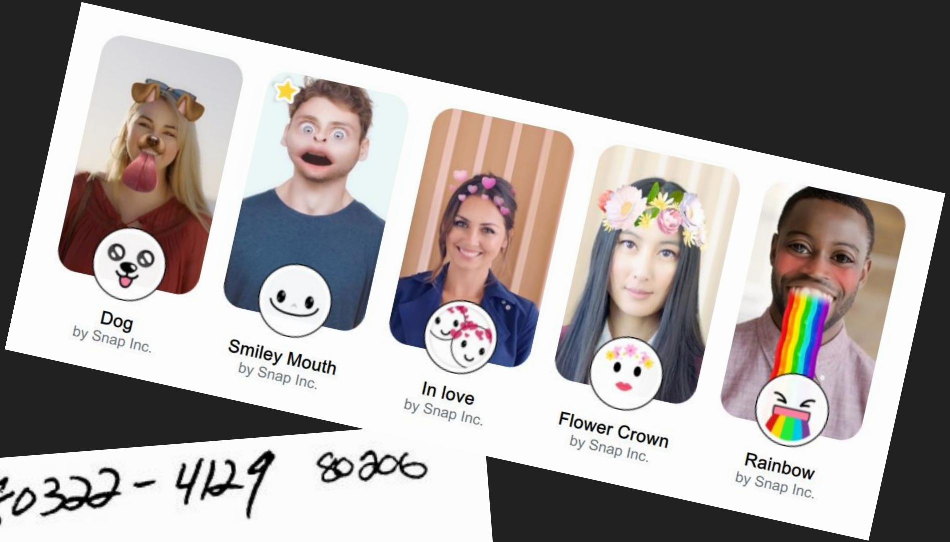
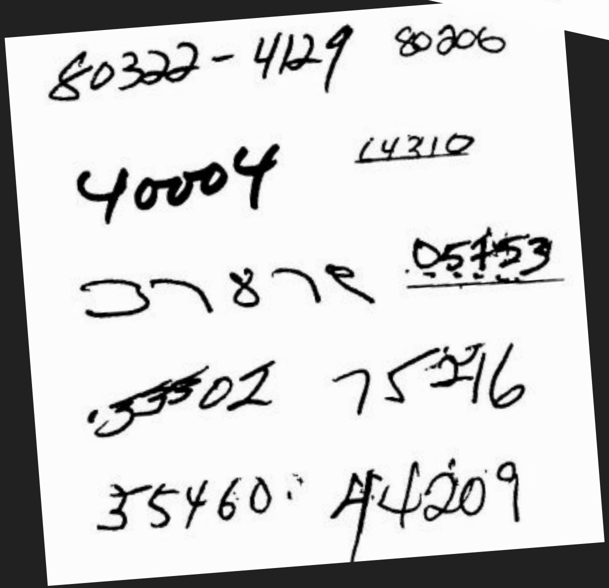
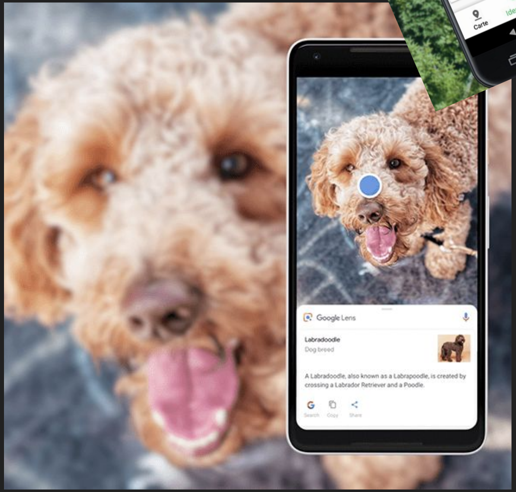
Connecte tes neurones !

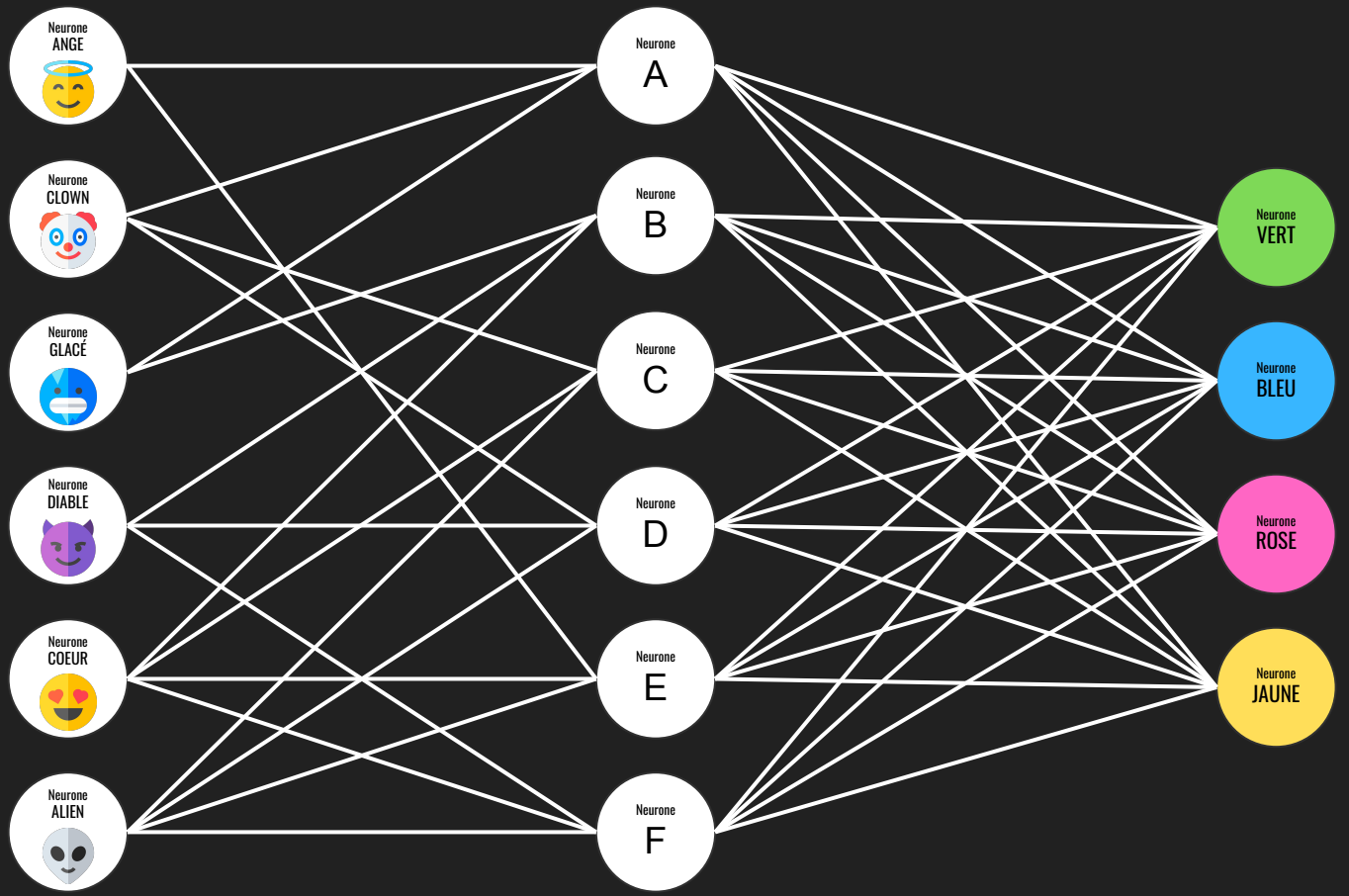


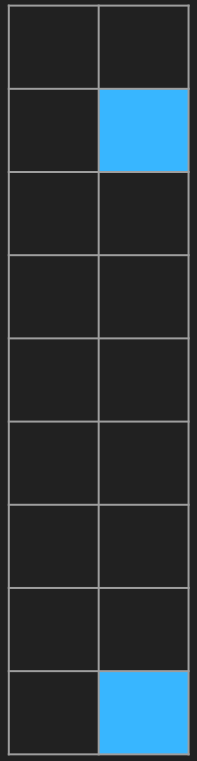

identifiez un champignon
à partir d'une photo
parmi 900 espèces
SAMSUNG
Fermer
Hmmm je crois que c'est:
Girolle
Ou peut-être un de ceux-là ?
craterellus aurora
cantharellus ignicolor
chrysomphalina strombodes
cantharellus
Carte
Identifier
Quiz
Marché
Liste


Dog
by Snap Inc.
Smiley Mouth
by Snap Inc.
In love
by Snap Inc.
Flower Crown
by Snap Inc.
Rainbow
by Snap Inc.

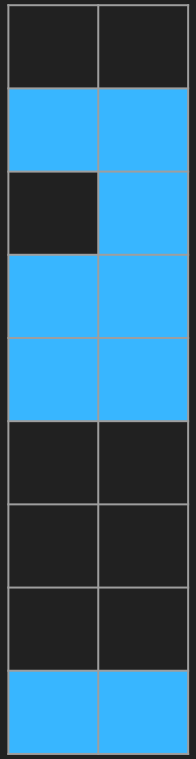

80322-4129
80206
40004
14310
37878
05153
35502
75216
35460
44209


Google Lens
Labradoodle
Dog breed
A Labradoodle, also known as a Labrapoodle, is created by crossing a Labrador Retriever and a Poodle.

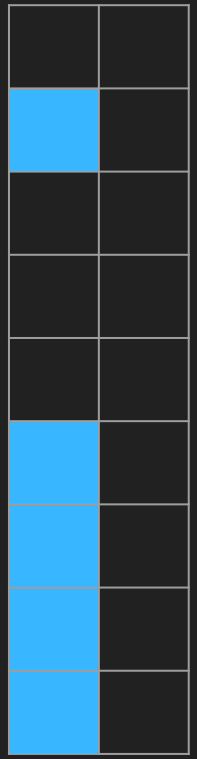

Neurone ANGE
Neurone CLOWN
Neurone GLACÉ
Neurone DIABLE
Neurone COEUR
Neurone ALIEN
Neurone A
Neurone B
Neurone C
Neurone D
Neurone E
Neurone F
Neurone VERT
Neurone BLEU
Neurone ROSE
Neurone JAUNE



Neurone 😇

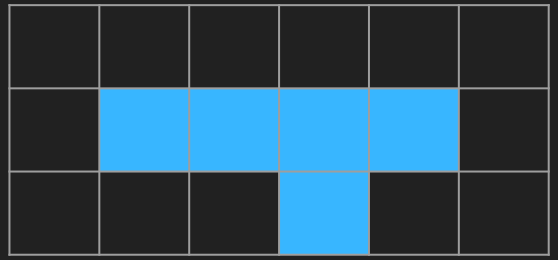


Neurone 🤡

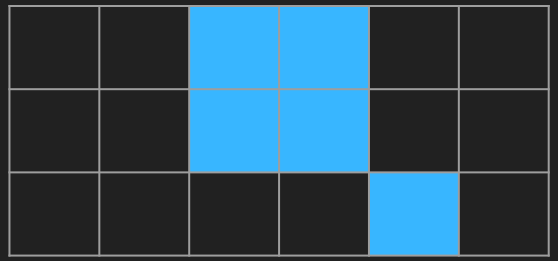


Neurone 🥶

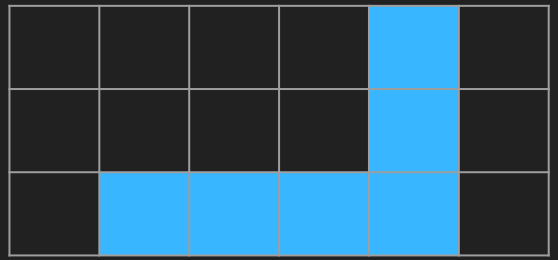
Neurone 😈

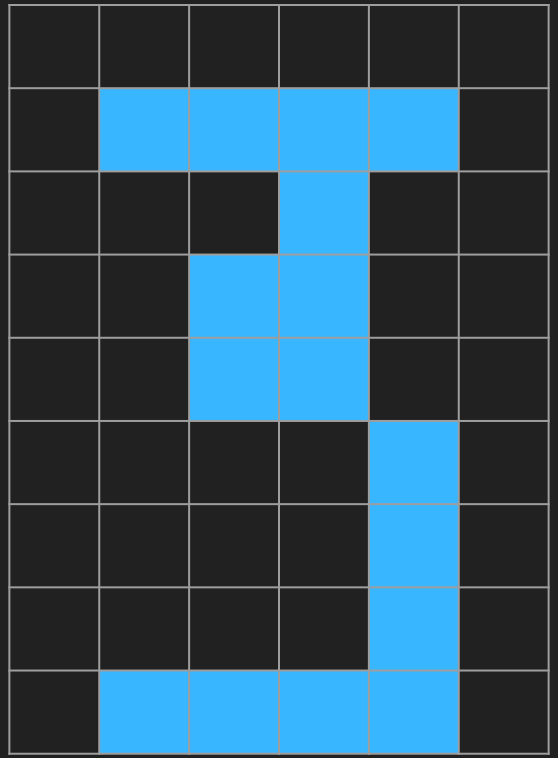


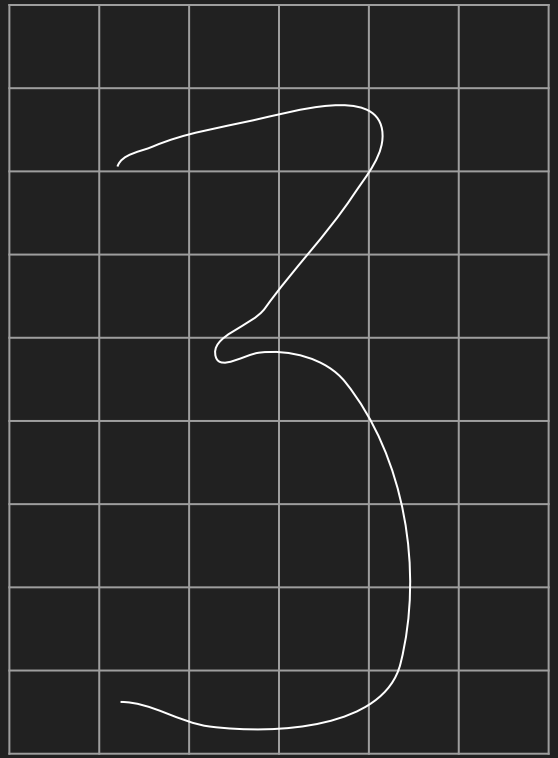
Neurone 😍

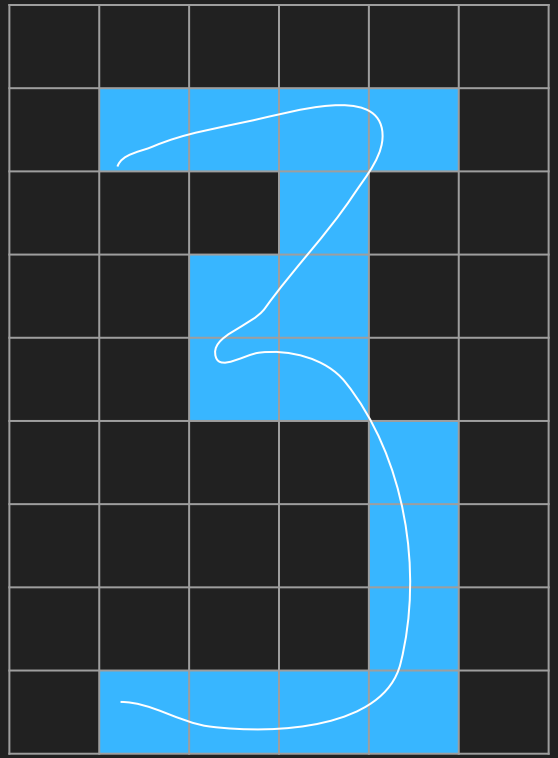


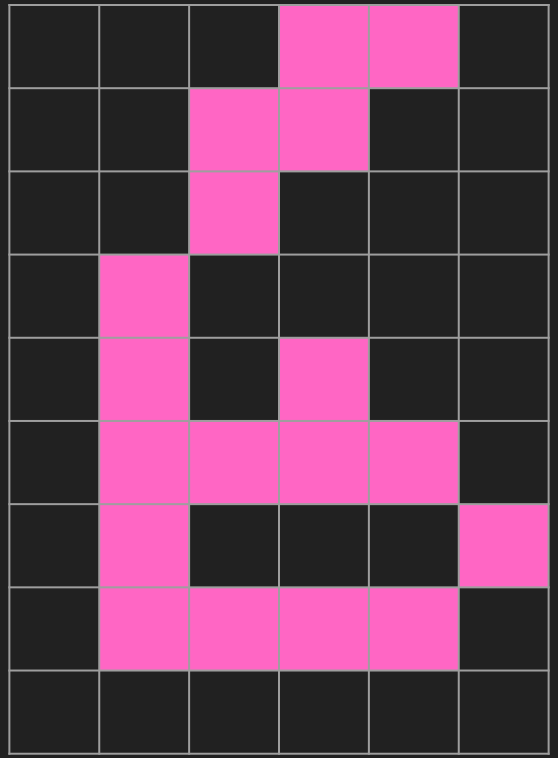
Neurone 👽











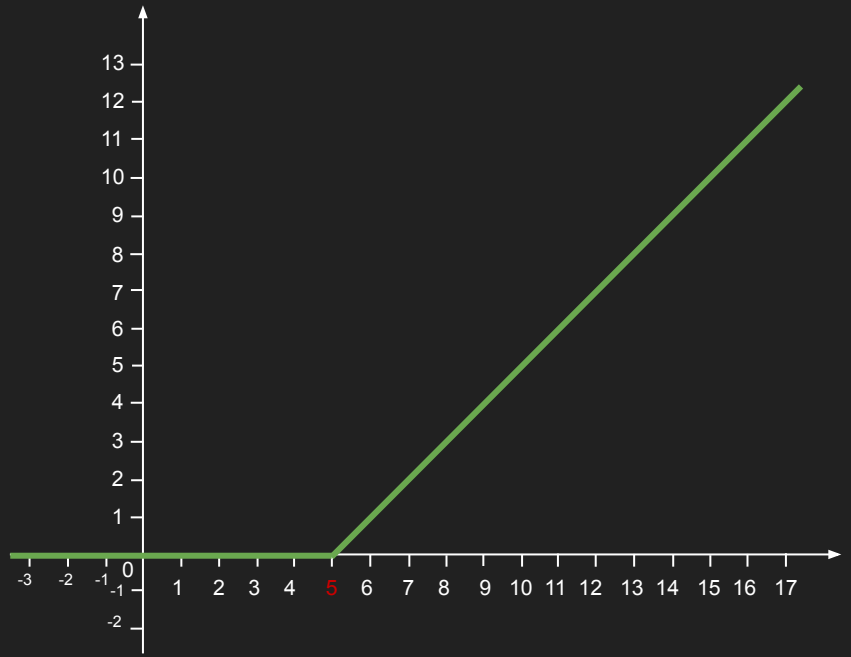
Rôle du neurone F

1 FAIS LE CALCUL SUIVANT :

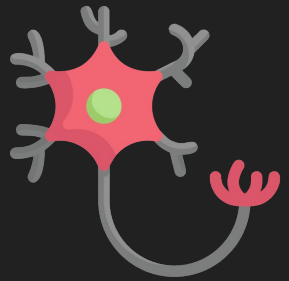
😈 + 👽 - 😍 = ?

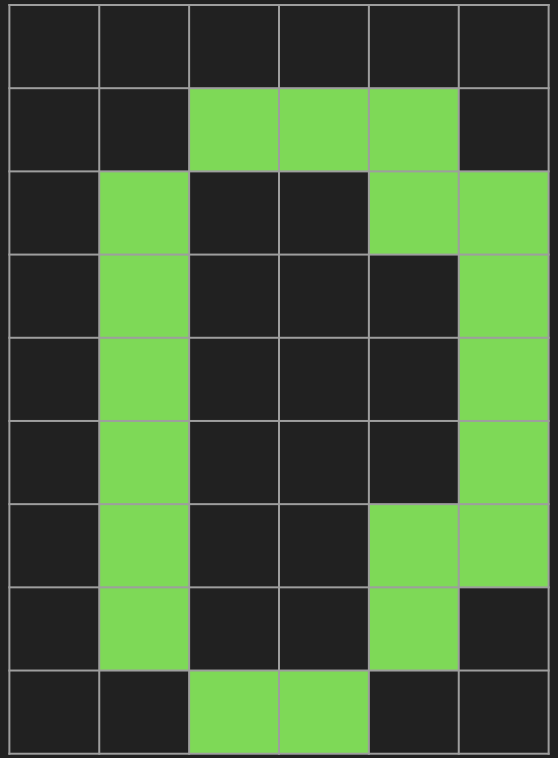
2 TROUVE LA RÉPONSE À MONTRER :

Résultat du calcul 😈+👽-😍	1	2	3	4	5	6	7	8	9
Réponse à montrer	0	0	0	0	0	1	2	3	4



Connecte tes neurones !





	Résultat du calcul	Sortie du neurone
Neurone A		
Neurone B		
Neurone C		
Neurone D		
Neurone E		
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B		
Neurone C		
Neurone D		
Neurone E		
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C		
Neurone D		
Neurone E		
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C	21	4
Neurone D		
Neurone E		
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C	21	4
Neurone D	5	0
Neurone E		
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C	21	4
Neurone D	5	0
Neurone E	19	3
Neurone F		

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C	21	4
Neurone D	5	0
Neurone E	19	3
Neurone F	7	2

	Résultat du calcul	Sortie du neurone
Neurone A	7	0
Neurone B	3	0
Neurone C	21	4
Neurone D	5	0
Neurone E	19	3
Neurone F	7	2

Envoyé à la couche suivante

	Résultat du calcul	Sortie du neurone
Neurone vert		
Neurone bleu		
Neurone rose		
Neurone jaune		

	Résultat du calcul	Sortie du neurone
Neurone vert	9	3
Neurone bleu		
Neurone rose		
Neurone jaune		

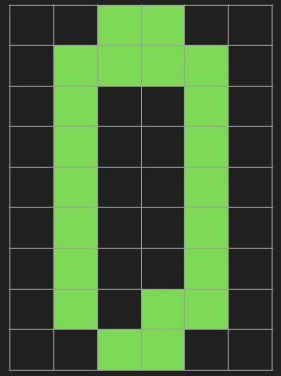
	Résultat du calcul	Sortie du neurone
Neurone vert	9	3
Neurone bleu	- 7	0
Neurone rose		
Neurone jaune		

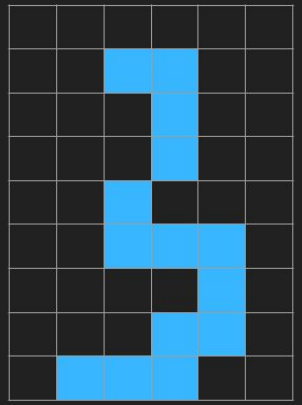
	Résultat du calcul	Sortie du neurone
Neurone vert	9	3
Neurone bleu	- 7	0
Neurone rose	-3	0
Neurone jaune		

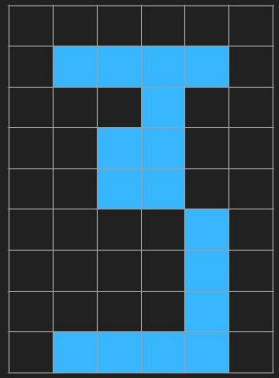
	Résultat du calcul	Sortie du neurone
Neurone vert	9	3
Neurone bleu	- 7	0
Neurone rose	-3	0
Neurone jaune	1	0

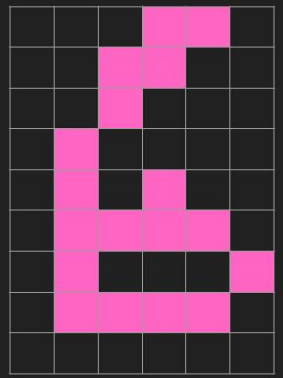
	Résultat du calcul	Sortie du neurone
Neurone vert	9	3
Neurone bleu	- 7	0
Neurone rose	-3	0
Neurone jaune	1	0

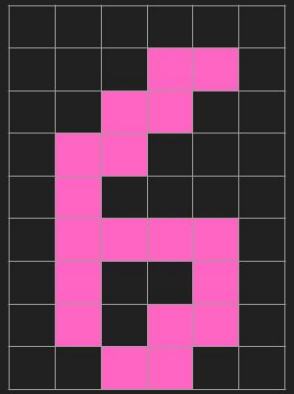
C'EST UN 0 !

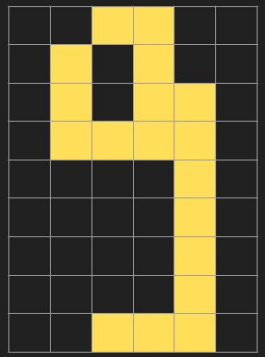


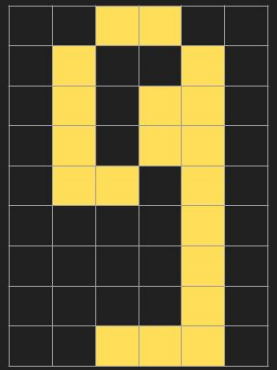


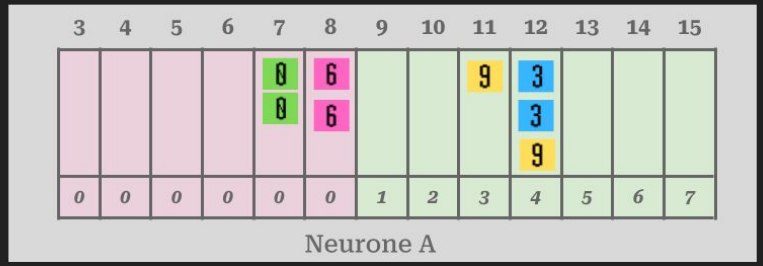


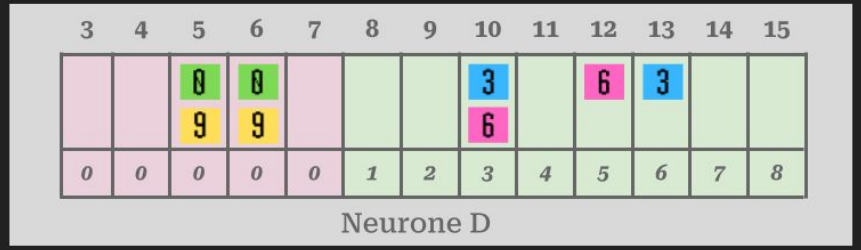


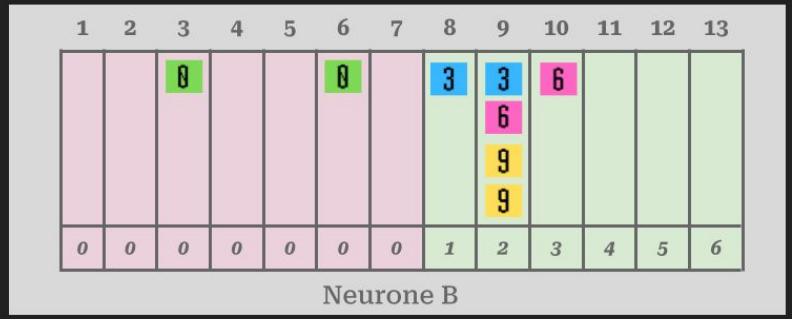


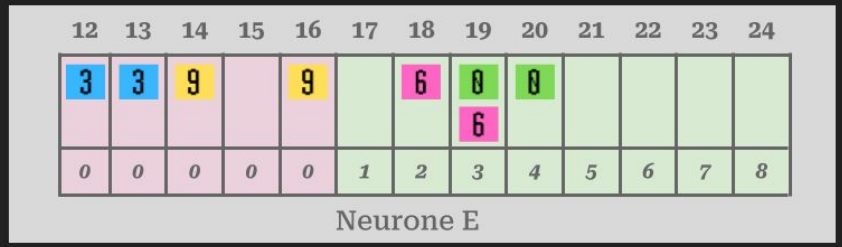


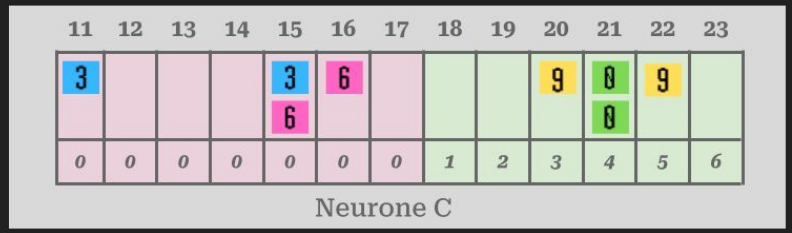


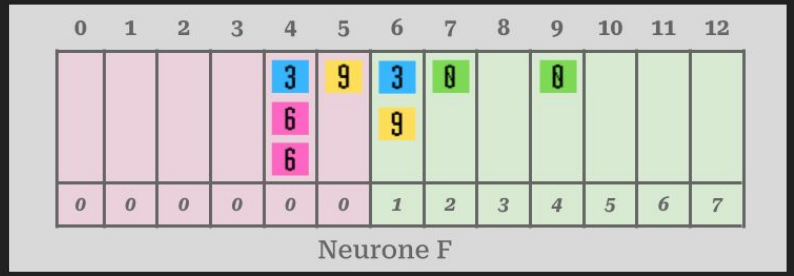

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone A

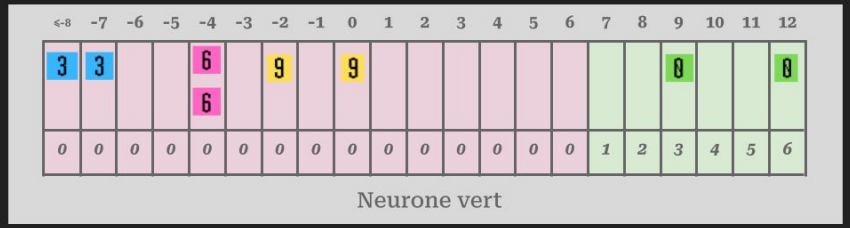

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone D


1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone B

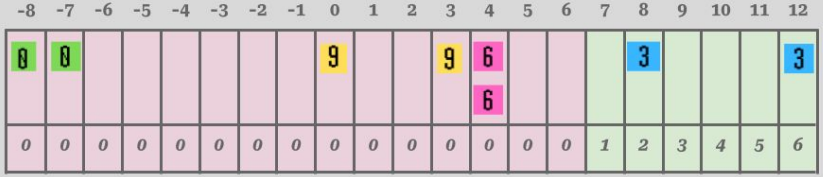

12 13 14 15 16 17 18 19 20 21 22 23 24
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone E


11 12 13 14 15 16 17 18 19 20 21 22 23
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone C

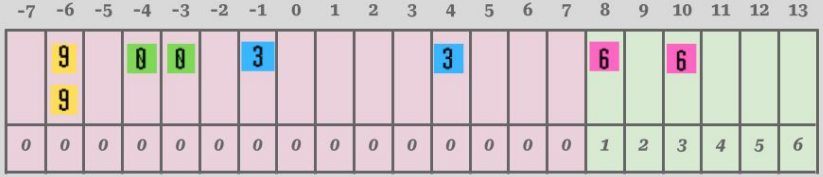

0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone F



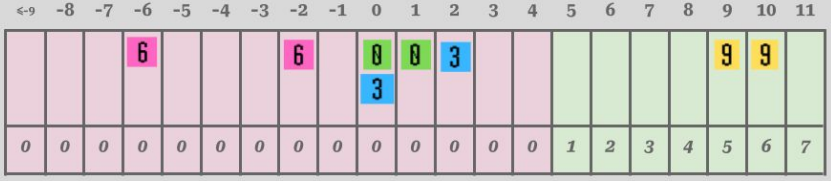
Neurone vert



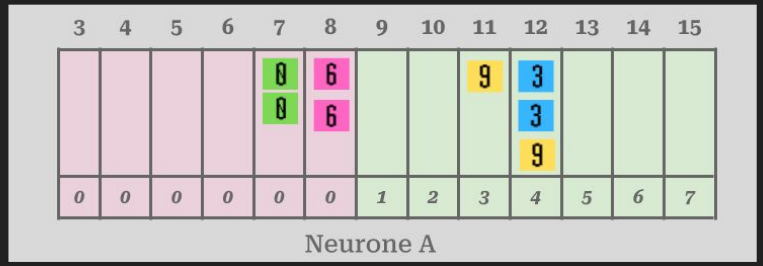
Neurone bleu

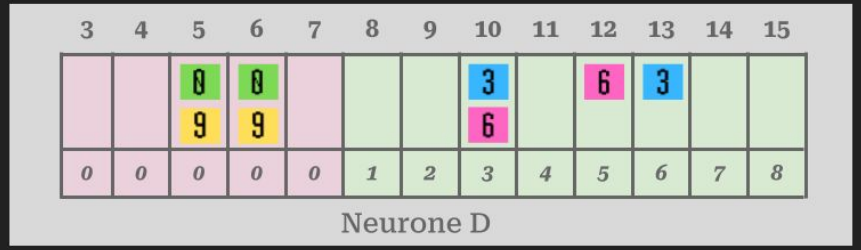


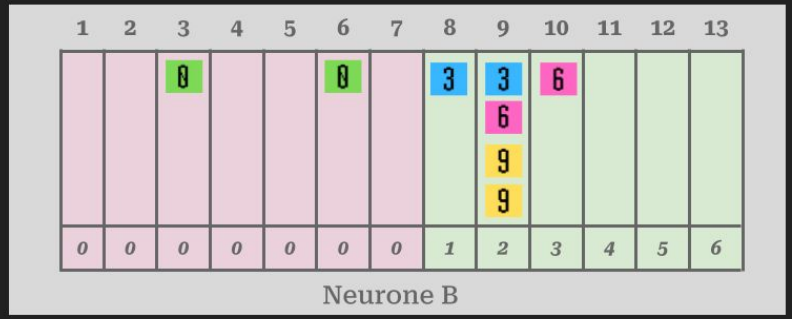
Neurone rose

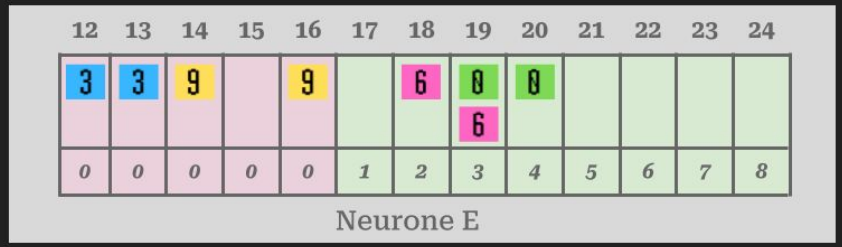


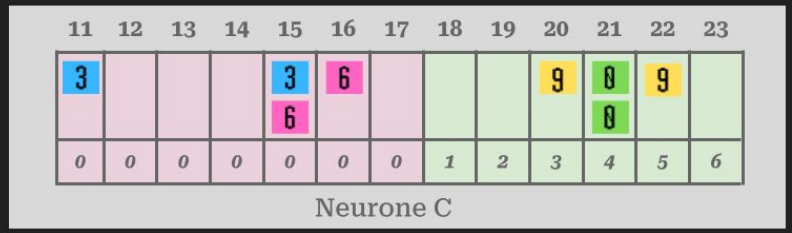
Neurone jaune

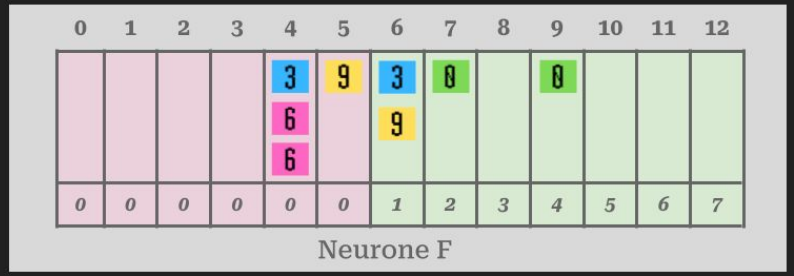

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone A


3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone D

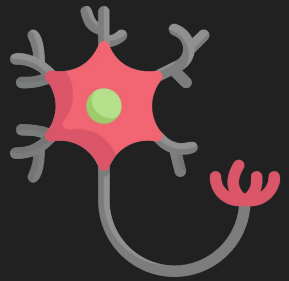

1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone B

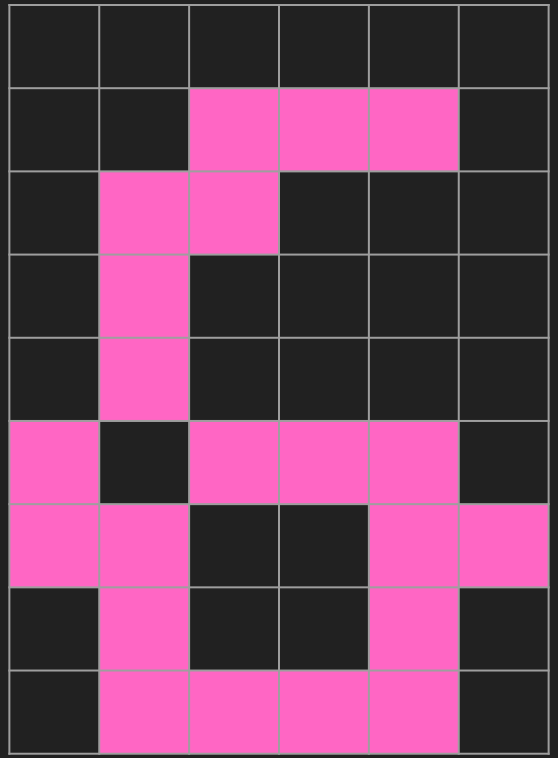

12 13 14 15 16 17 18 19 20 21 22 23 24
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone E

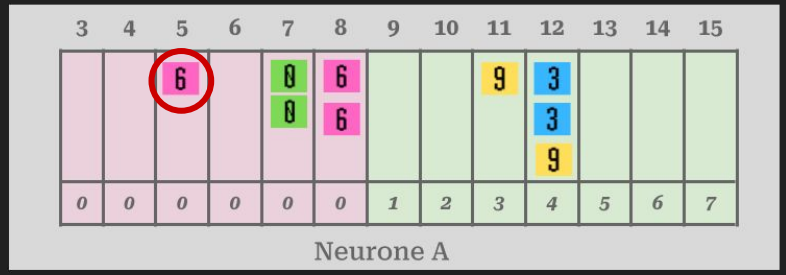

11 12 13 14 15 16 17 18 19 20 21 22 23
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone C


0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone F

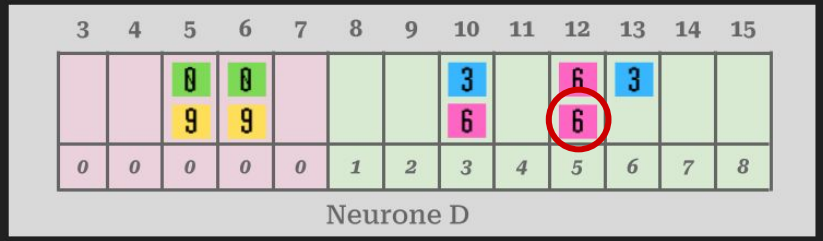
Connecte tes neurones !



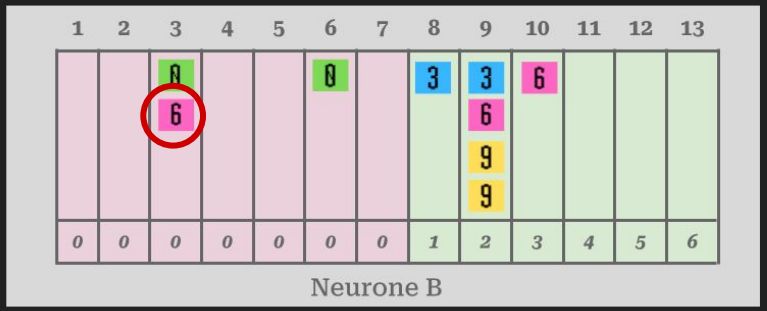




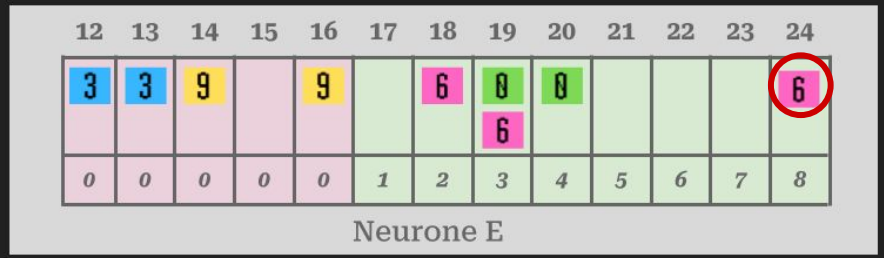
Neurone A



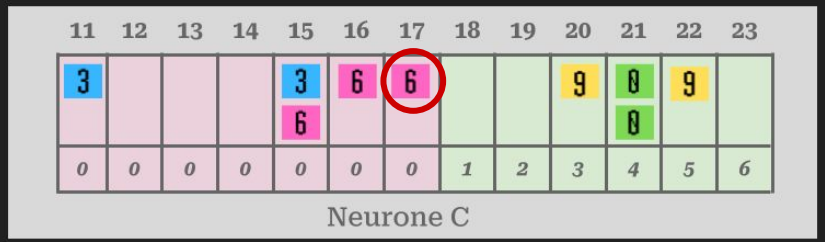
Neurone D



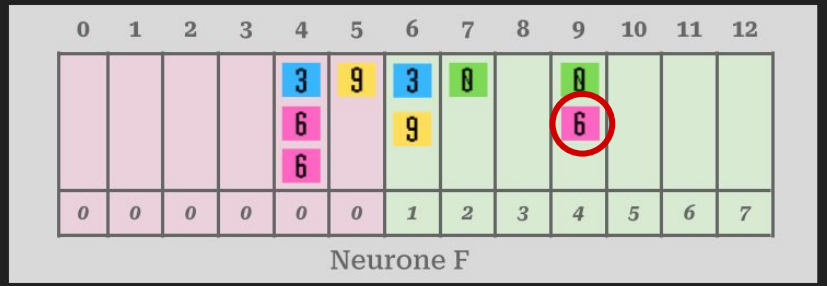
Neurone B



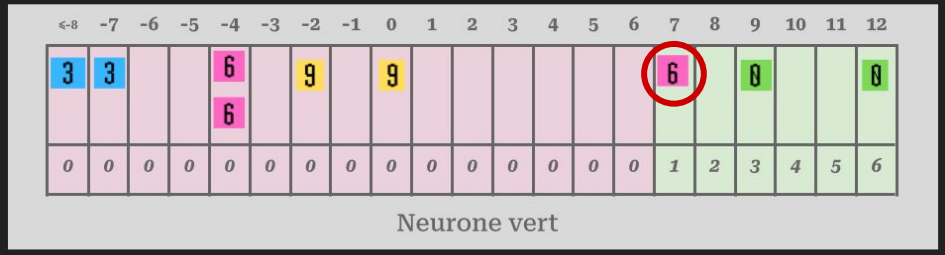
Neurone E



Neurone C

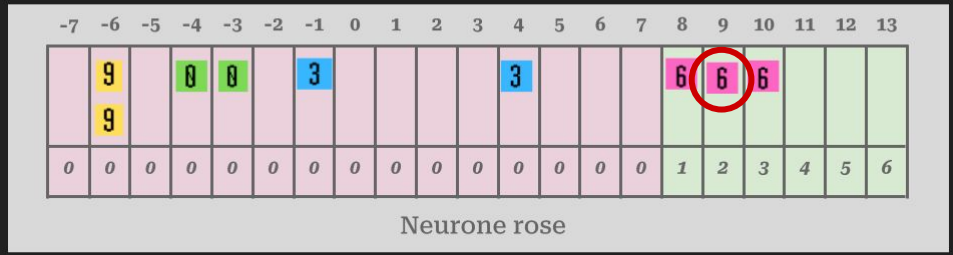


Neurone F

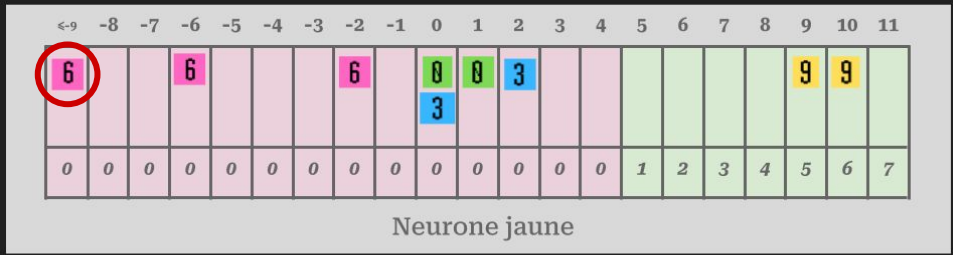


Neurone vert

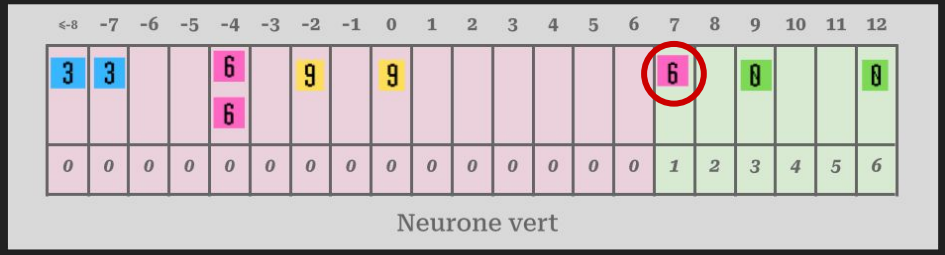
Neurone bleu



Neurone rose

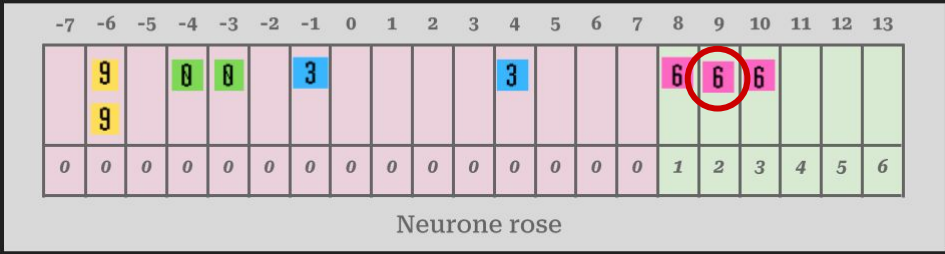


Neurone jaune

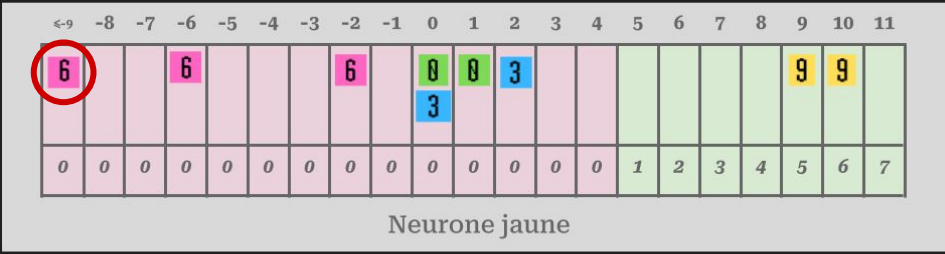


Neurone vert

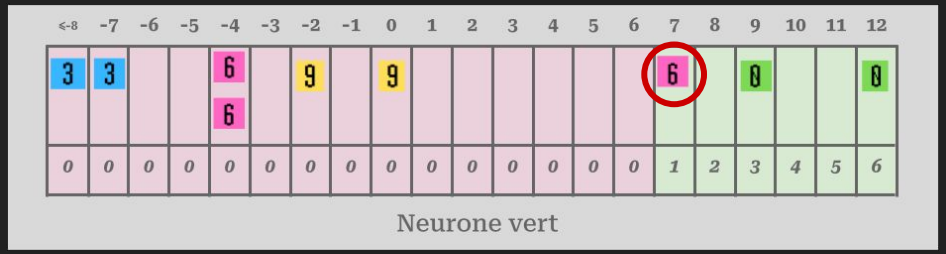
Neurone bleu

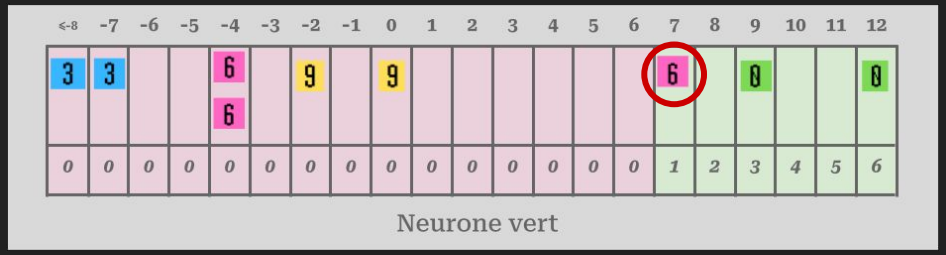


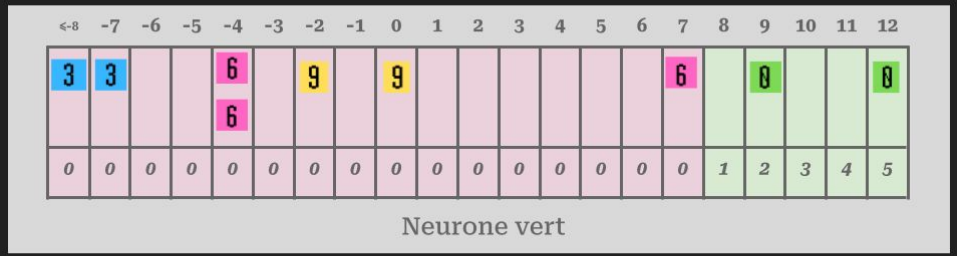
Neurone rose

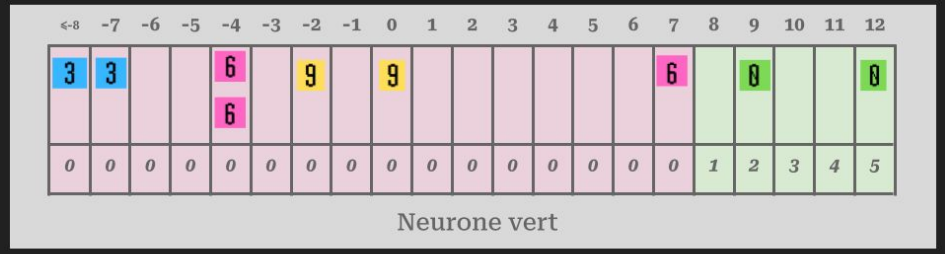


Neurone jaune


<-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
3 3 6 6 9 9 6 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone vert

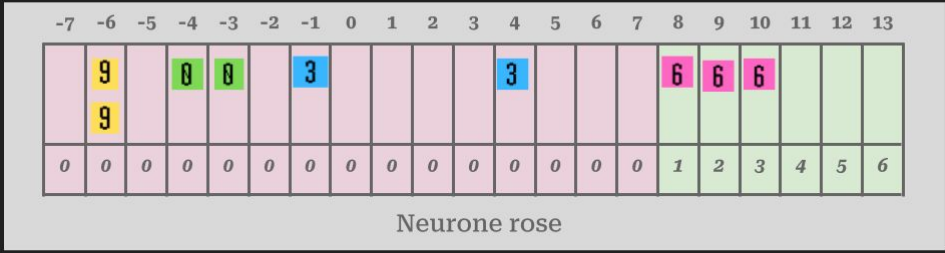

<-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
3 3 6 6 9 9 6 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone vert


<-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
3 3 6 6 9 9 6 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5
Neurone vert

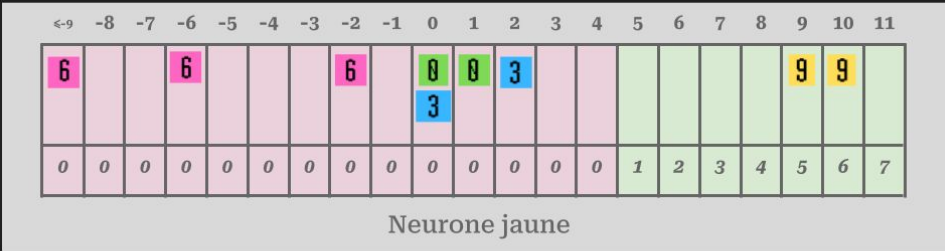


Neurone vert

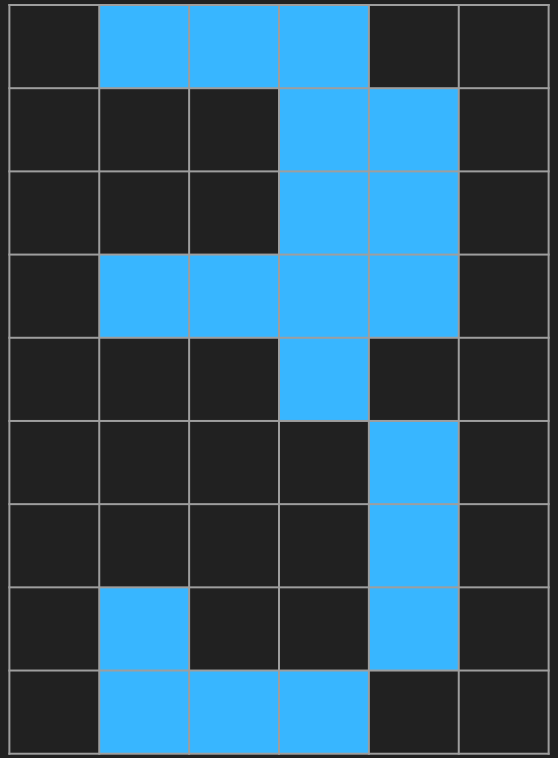
Neurone bleu

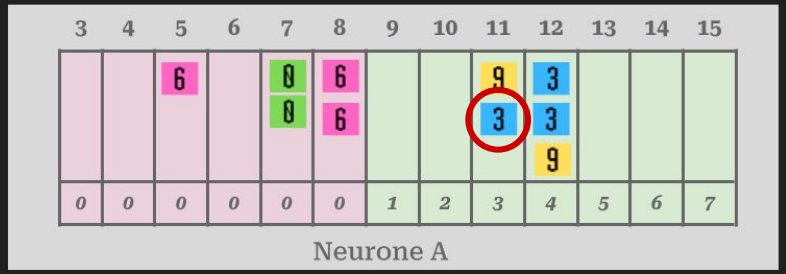


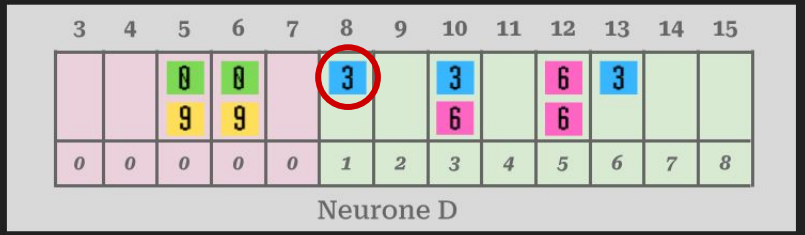
Neurone rose

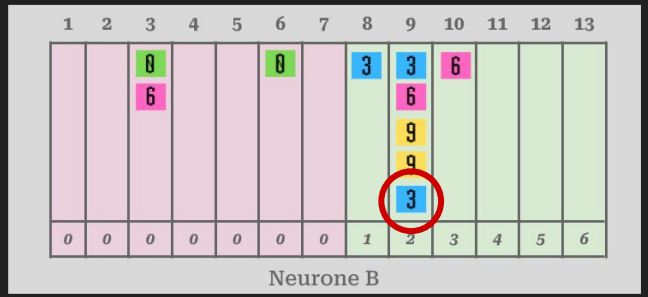


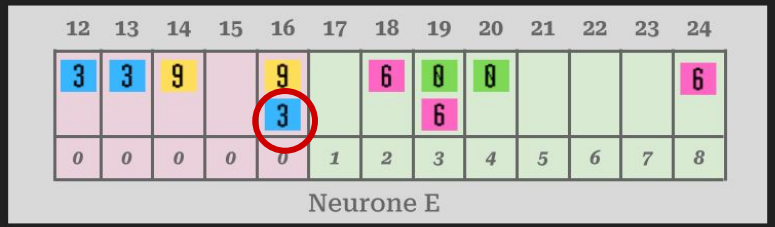
Neurone jaune

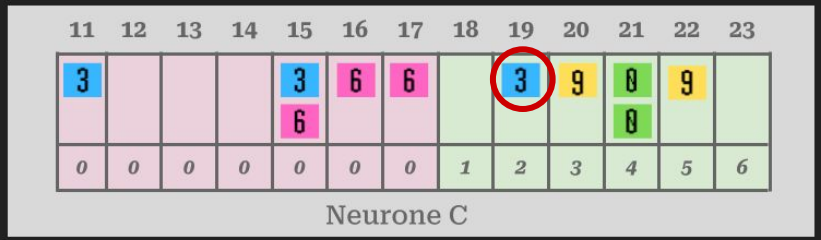


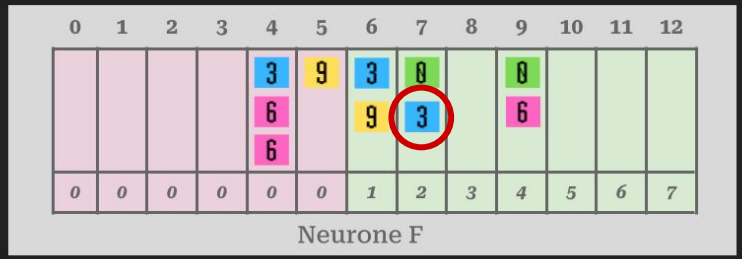

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone A

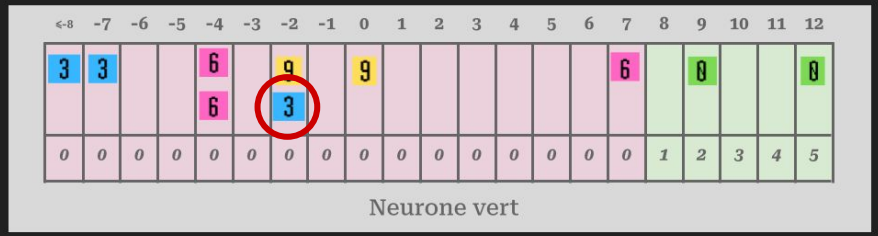

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone D

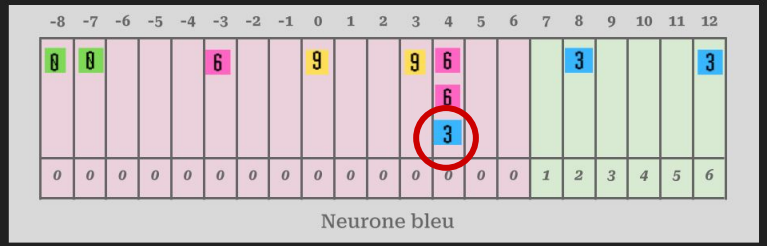

1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone B

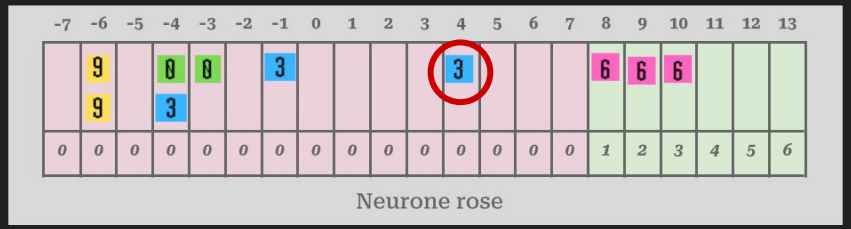

12 13 14 15 16 17 18 19 20 21 22 23 24
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone E

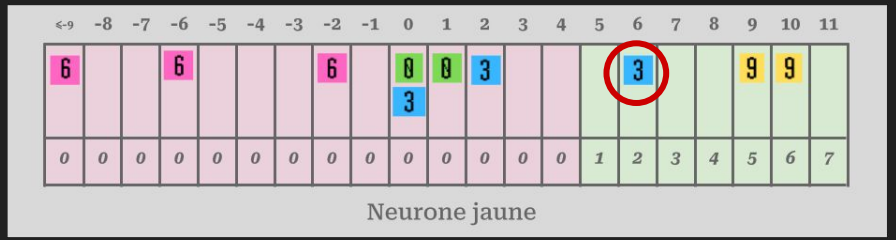

11 12 13 14 15 16 17 18 19 20 21 22 23
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone C

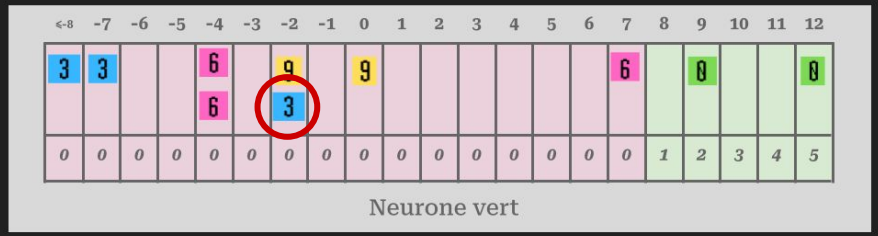

0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone F

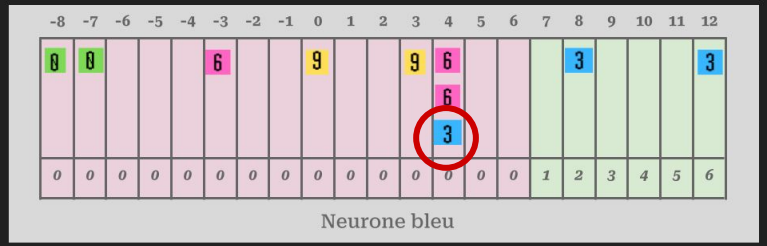

Neurone vert

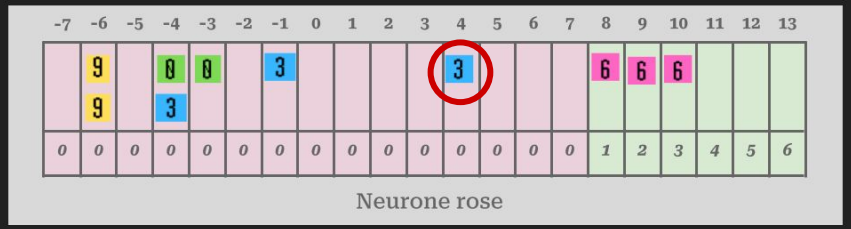

Neurone bleu

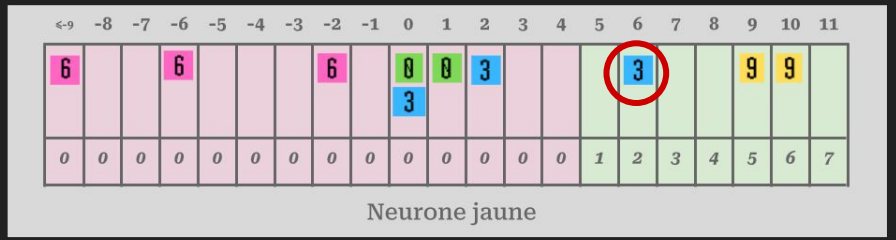

Neurone rose

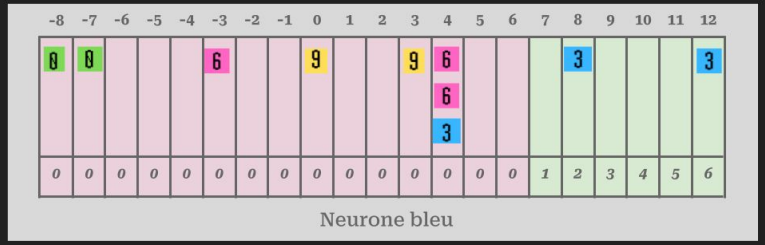

Neurone jaune


Neurone vert

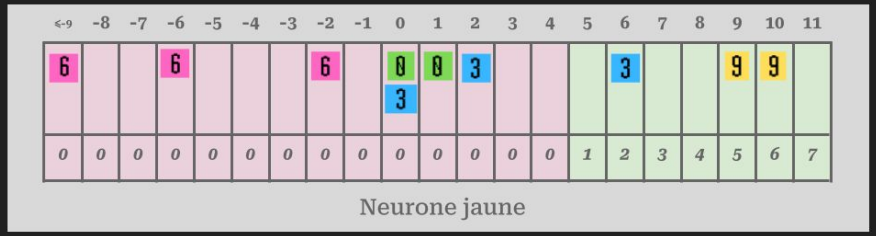

Neurone bleu


Neurone rose

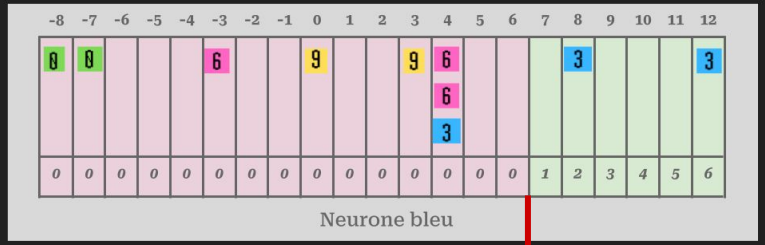

Neurone jaune

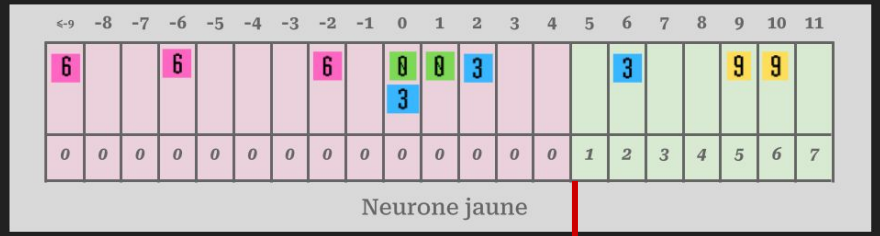


Neurone bleu



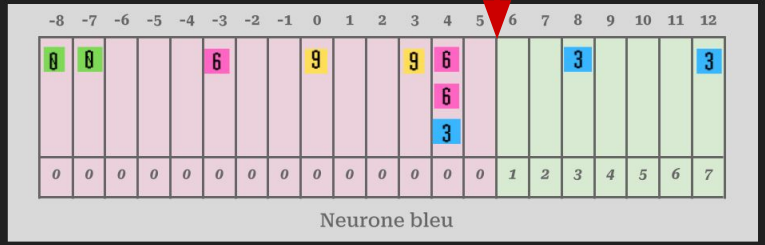
Neurone jaune

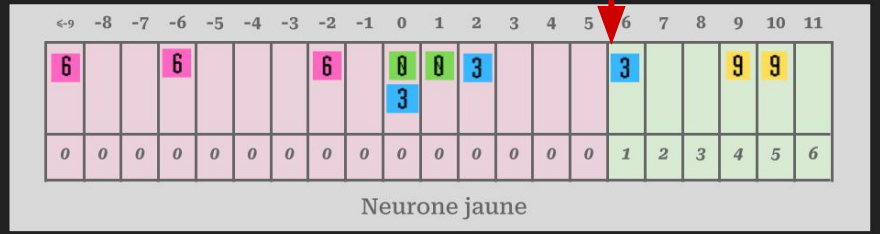

-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 6 9 9 6 6 3 3 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone bleu

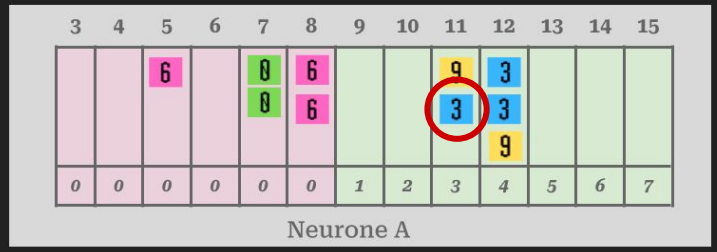

<-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
6 6 6 0 0 3 3 3 9 9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone jaune

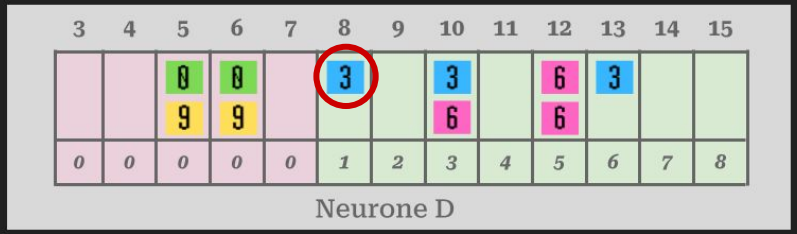
-1

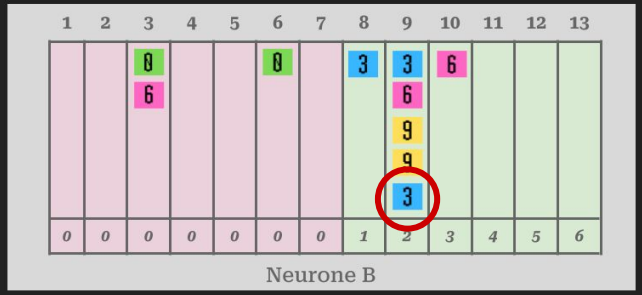
+1

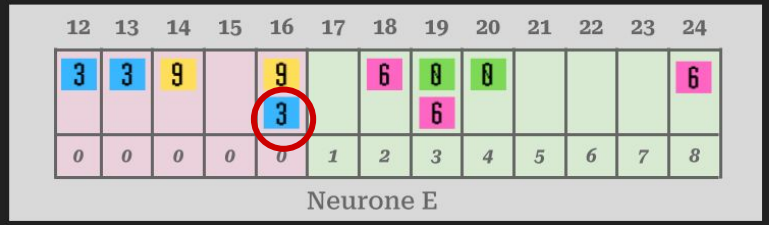

-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 6 9 9 6 6 3 3 3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone bleu

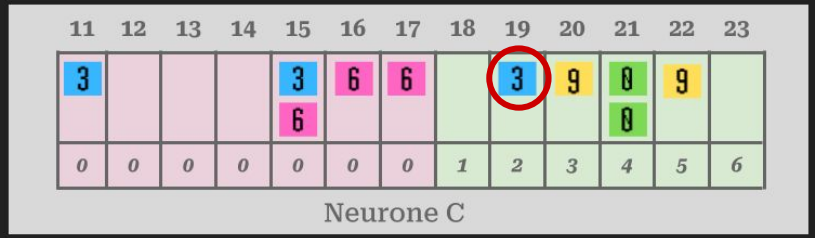

<-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
6 6 6 0 0 3 3 3 9 9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone jaune

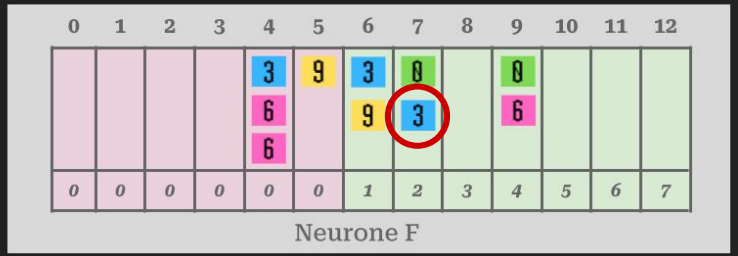

3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone A


3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone D

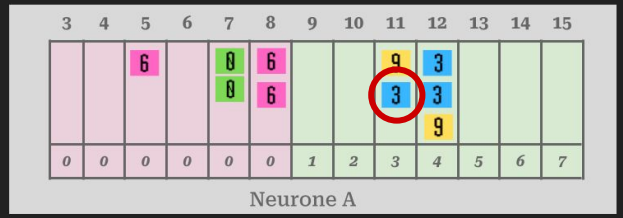

1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone B

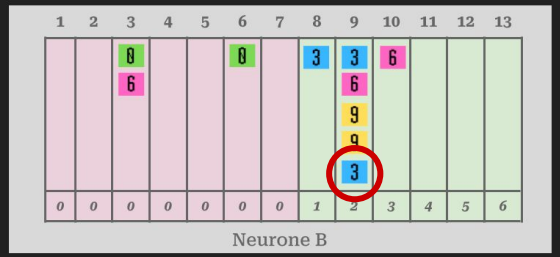

12 13 14 15 16 17 18 19 20 21 22 23 24
0 0 0 0 0 1 2 3 4 5 6 7 8
Neurone E

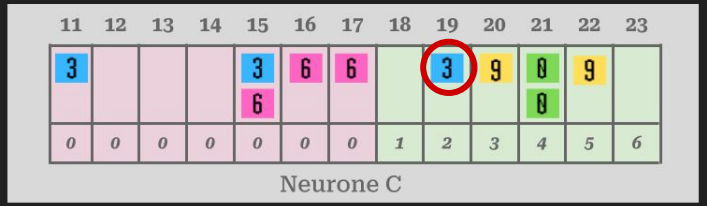

11 12 13 14 15 16 17 18 19 20 21 22 23
0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone C


0 1 2 3 4 5 6 7 8 9 10 11 12
0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone F

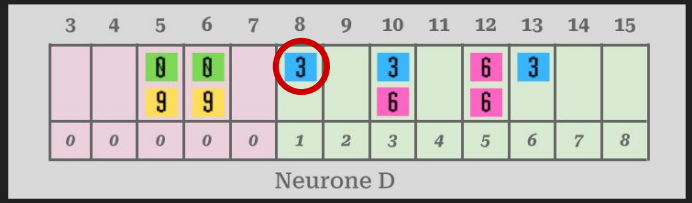
Neurone bleu : A + B + D - C - E

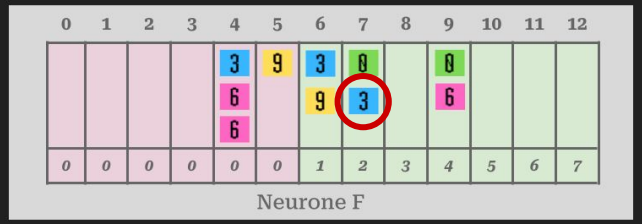






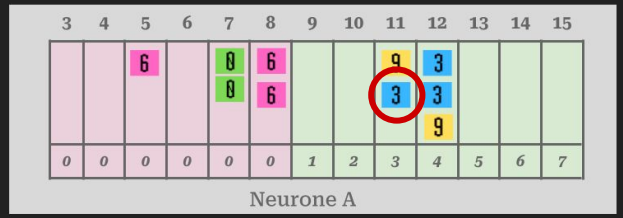
Neurone jaune : A + B + C - D - E



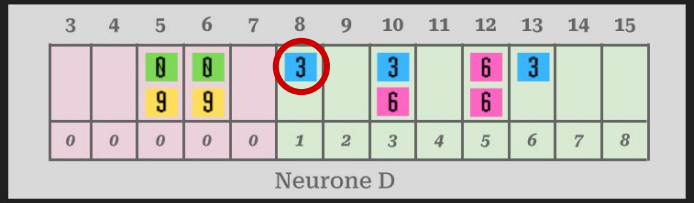


Neurone bleu : A + B + D - C - E

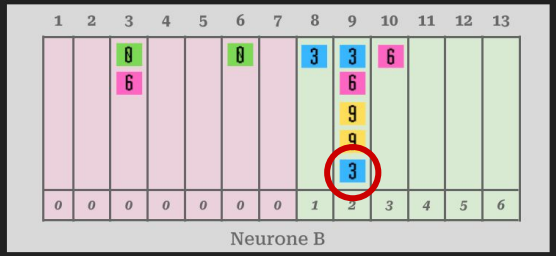
Neurone jaune : A + B + C - D - E



Neurone A



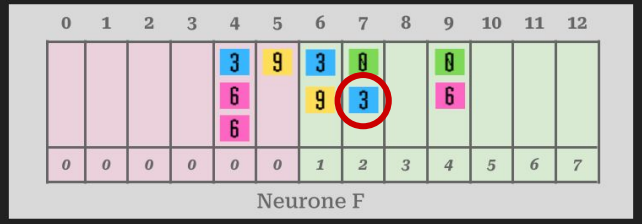
Neurone D



Neurone B

Neurone E

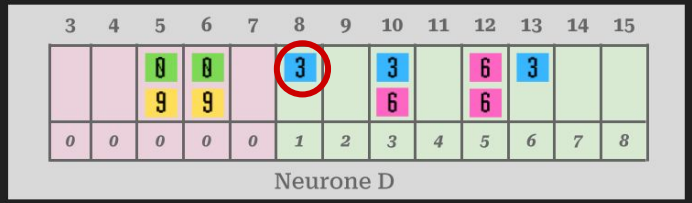
Neurone C



Neurone F

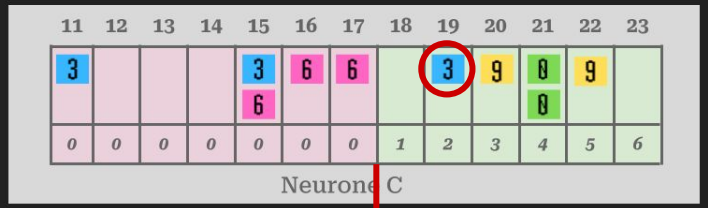
Neurone bleu : A + B + D - C - E

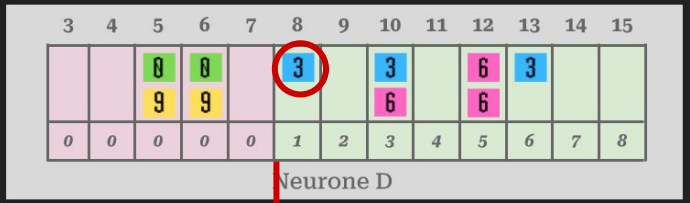
Neurone jaune : A + B + C - D - E



Neurone bleu : A + B + D - C - E

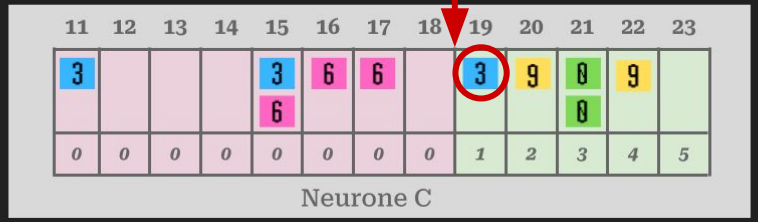
Neurone jaune : A + B + C - D - E

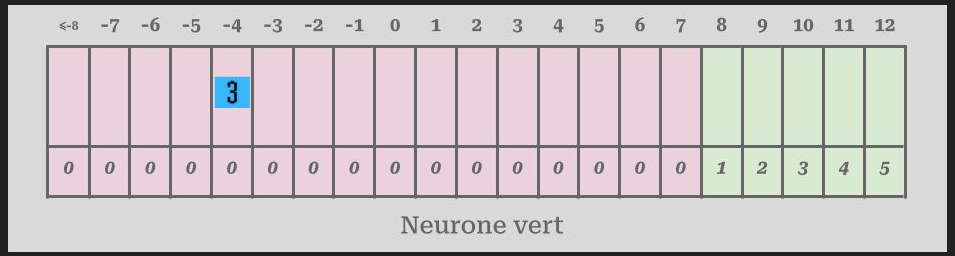


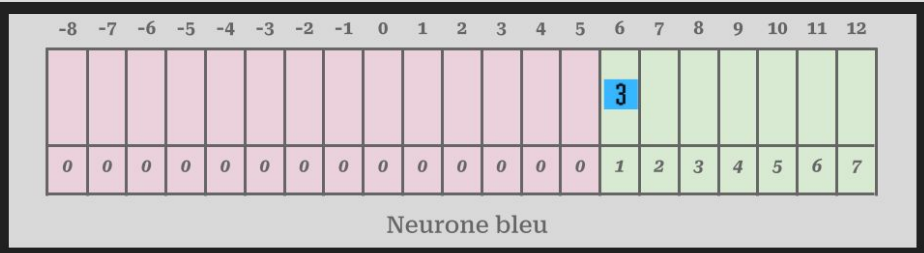


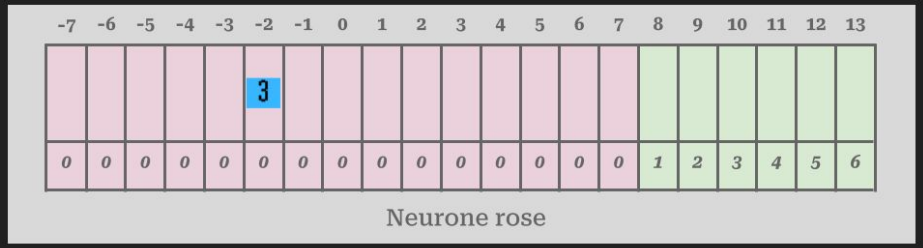
+1

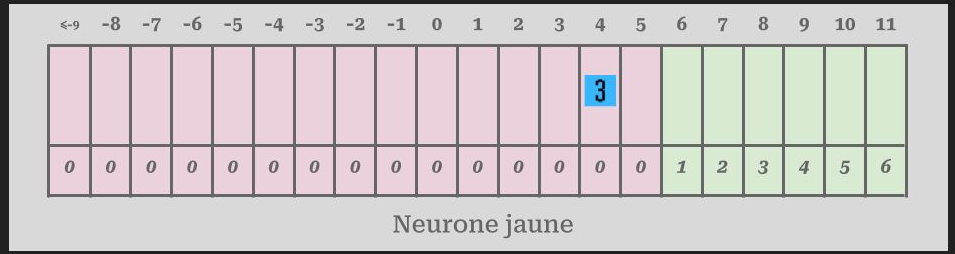
-1



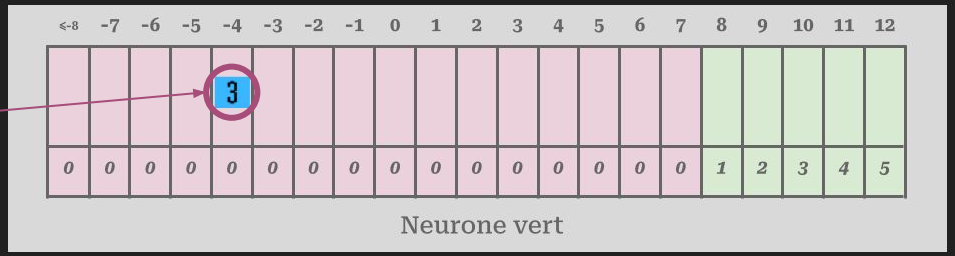

≤-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5
Neurone vert

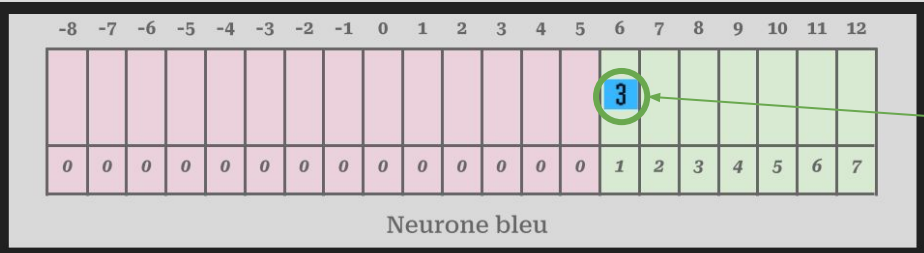

-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12
3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7
Neurone bleu


-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13
3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone rose


≤-9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 3 4 5 6
Neurone jaune

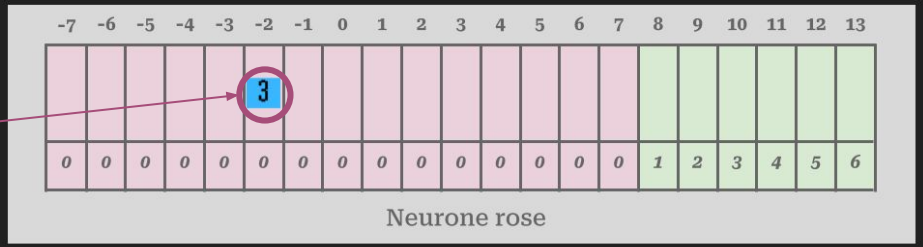
Ce n'est pas un 0



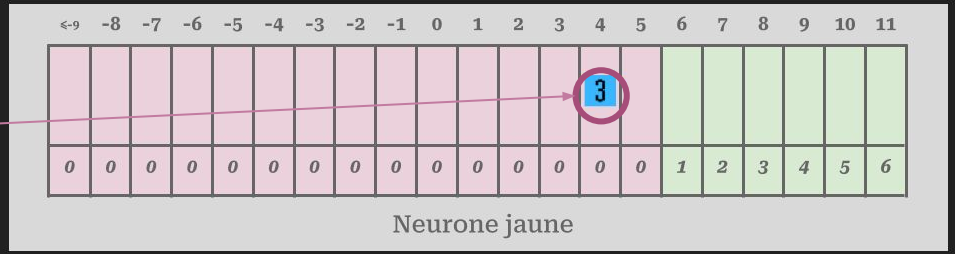


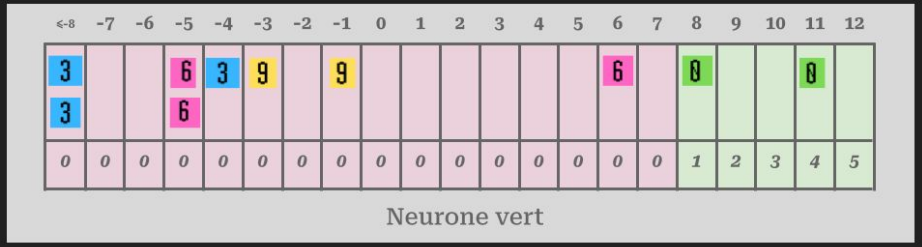
C'est un 3

Ce n'est pas un 6



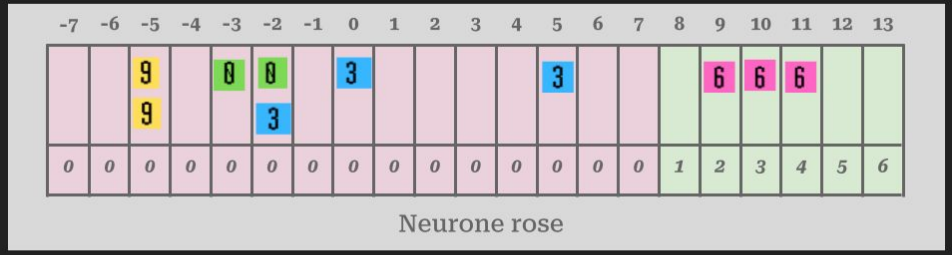
Ce n'est pas un 9



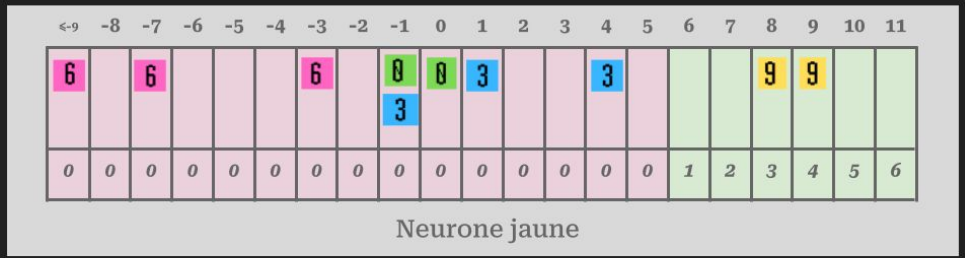


Neurone vert

Neurone bleu

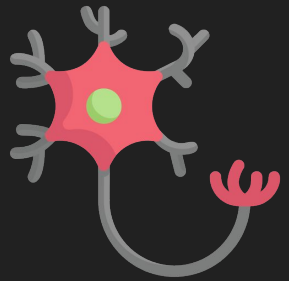


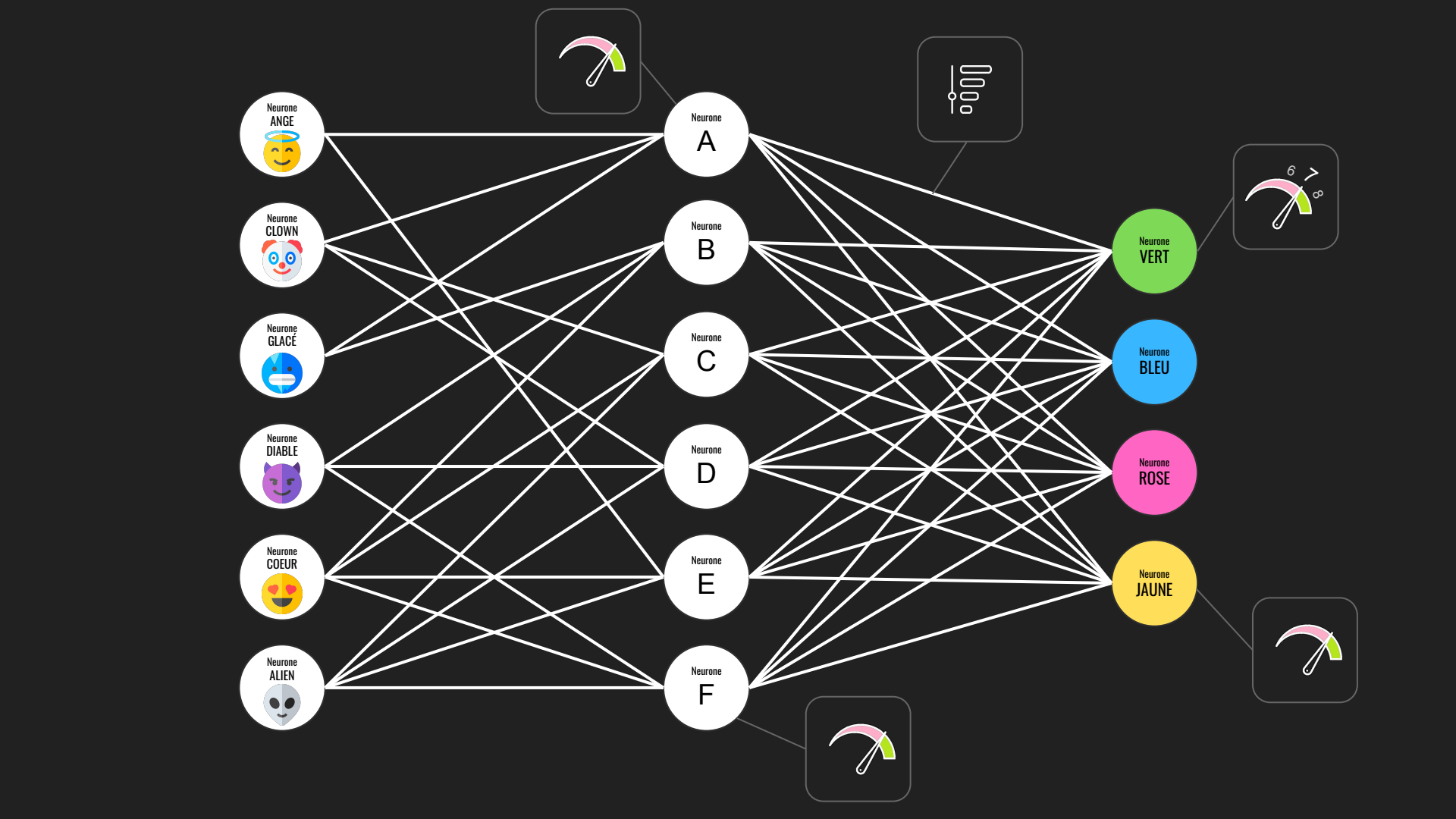
Neurone rose



Neurone jaune

Connecte tes neurones !




Neurone ANGE
Neurone CLOWN
Neurone GLACÉ
Neurone DIABLE
Neurone COEUR
Neurone ALIEN
Neurone A
Neurone B
Neurone C
Neurone D
Neurone E
Neurone F
Neurone VERT
Neurone BLEU
Neurone ROSE
Neurone JAUNE

Le réseau et son apprentissage en Python

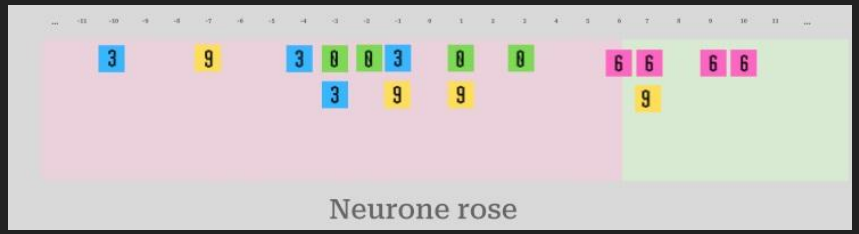
Le réseau et son apprentissage en Python

```python
import numpy as np



sortie_cachee=np.empty((16,6),dtype='float64')
sortie_finale=np.empty((16,4),dtype='float64')
seuil_cache=np.empty((6,),dtype='float64')
seuil_final=np.empty((4,),dtype='float64')
erreur=np.empty((16,6),dtype='float64')

################ INITIALISATION DES POIDS ET DES SEUILS ################

W=np.empty((6,6),dtype='float64')
W[0]=[-1,1,1,0,0,0]
W[1]=[1,1,0,0,0,-1]
W[2]=[0,0,1,1,1,0]
W[3]=[0,1,0,-1,0,1]
W[4]=[1,0,0,0,1,1]
W[5]=[0,0,0,1,-1,1]

masque=np.empty((6,6))

masque[0]=[1,1,1,0,0,0]
masque[1]=[1,1,0,0,0,1]
masque[2]=[0,0,1,1,1,0]
masque[3]=[0,1,0,1,0,1]
masque[4]=[1,0,0,0,1,1]
masque[5]=[0,0,0,1,1,1]

seuil_cache=np.array([8,7,18,6,16,5],dtype='float64')

P=np.empty((4,6),dtype='float64')
P[0]=[-1,-1,1,-1,1,1]
P[1]=[1,1,-1,1,-1,0]
P[2]=[-1,1,-1,1,1,-1]
P[3]=[1,1,1,-1,-1,0]

seuil_final=np.array([7,5,6,5],dtype='float64')

################ LES DONNÉES ################################

test=np.empty((16,6),dtype='float64')

test[0]=[7,7,7,8,6,7]
test[1]=[6,4,9,6,6,7]
test[2]=[7,4,7,6,6,6]
test[3]=[4,5,8,5,7,5]
test[4]=[2,9,5,5,5,6]
test[5]=[1,10,3,3,5,6]
test[6]=[4,9,6,7,6,6]
test[7]=[2,8,8,7,5,6]
test[8]=[5,9,4,5,7,6]
test[9]=[5,9,4,4,7,7]
test[10]=[8,7,6,5,6,10]
test[11]=[4,9,6,4,8,7]
test[12]=[3,8,7,7,6,5]
test[13]=[4,7,8,7,7,5]
test[14]=[4,7,7,5,6,7]
test[15]=[5,6,7,6,7,5]

label = np.array([0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3])
```

```python

################ PARAMETRES DE L'APPRENTISSAGE ############################

pas = 0.0001 #le pas dans le flot gradient
coef = 0.7 #coef dans la fonction de coût d'erreur qui est une fonction exponentielle

################ L'APPRENTISSAGE PAR RETROPROPAGATION DU GRADIENT #################

# fonction de coût
def cout(y,l,a):
    s = np.where(np.arange(4)[np.newaxis,:]==l[:,np.newaxis],-1,1)
    return np.exp(a*s*y)

# dérivée de la fonction de coût
def dercout(y,l,a):
    s = np.where(np.arange(4)[np.newaxis,:]==l[:,np.newaxis],-1,1)
    return a*s*np.exp(a*s*y)

for epoque in range(100001):
    ###calcul des sorties (passage forward)
    # calcul de l'activité de la couche cachée par produit scalaire
    # (i.e. somme pondérée) avec les entrées et ...
    sortie_cachee = np.dot(test,W.T)-seuil_cache
    # ... passage par la fonction d'activation ReLU
    sortie_cachee = np.maximum(0,sortie_cachee)
    # calcul de la sortie du réseau par produit scalaire avec la couche cachée
    sortie_finale = np.dot(sortie_cachee,P.T)-seuil_final
    # calcul de l'erreur du réseau
    erreur = cout(sortie_finale,label,coef)
    ### fin du calcul des sorties ####
    #### ajustement des seuils et des poids #####
    #### ajustement des seuils finaux ####
    # gradient de la couche de sortie
    grad = dercout(sortie_finale,label,coef)
    # mise à jour du seuil de la couche de sortie
    seuil_final += 10*pas*np.sum(grad,axis=0)
    # mise à jour des poids de la couche de sortie
    P -= pas*np.dot(grad.T,sortie_cachee)
    # rétropropagation du gradient à la couche cachée
    grad_cache = np.dot(grad,P)*(sortie_cachee!=0) #sortie_cachee!=0 (dérivée de ReLU)
    # mise à jour du seuil de la couche cachée
    seuil_cache += pas*np.sum(grad_cache,axis=0)
    # mise à jour des poids de la couche cachée
    W -= pas*np.dot(grad_cache.T,test)
    W *= masque # on force les connexions inexistantes à 0


####calcul des sorties au final (passage forward)
sortie_cachee = np.dot(test,W.T)-seuil_cache[np.newaxis,:]
sortie_cachee = np.maximum(0,sortie_cachee)
sortie_finale = np.dot(sortie_cachee,P.T)-seuil_final[np.newaxis,:]
erreur = cout(sortie_finale,label,coef)
```

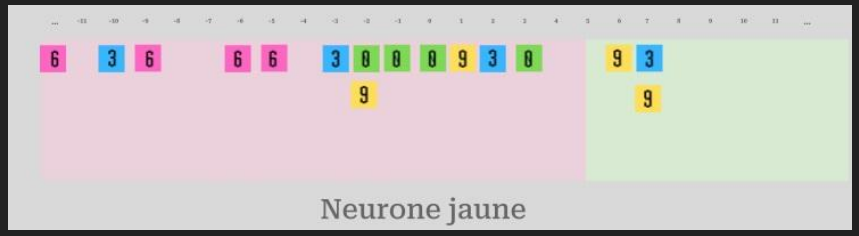
Avec quelques exemples en plus

Neurone vert



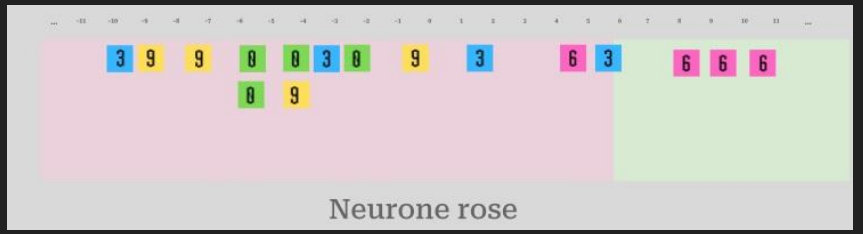
Neurone rose

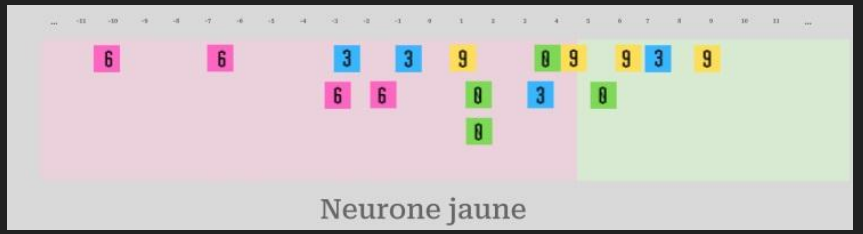
Neurone bleu



Neurone jaune

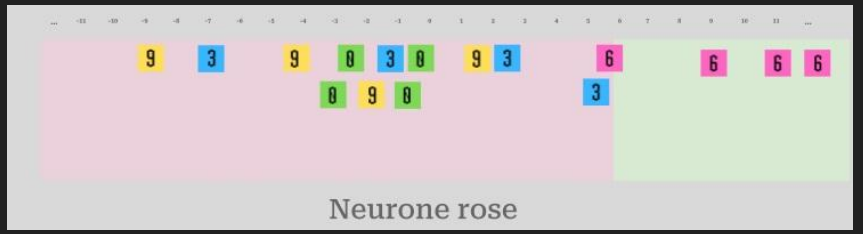
Après 100 itérations





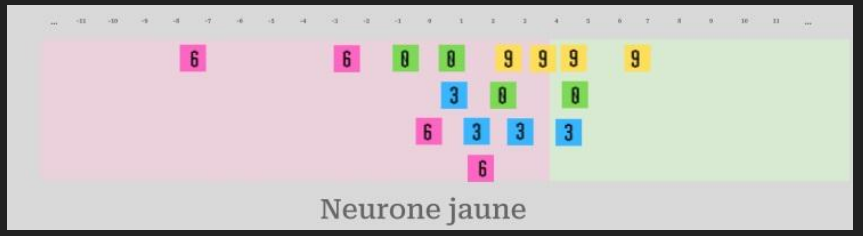
Après 1000 itérations

Neurone vert



Neurone rose

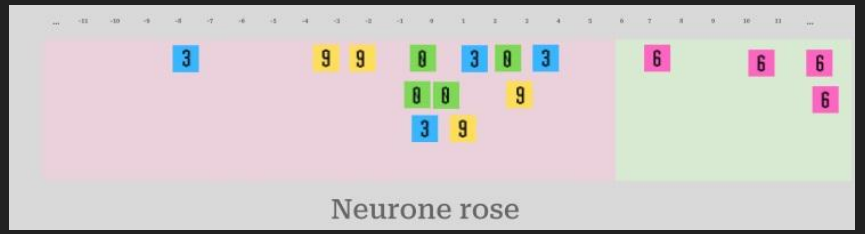
Neurone bleu



Neurone jaune

Après 5000 itérations

Neurone vert

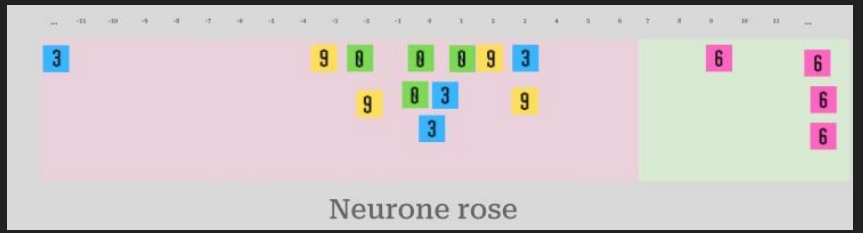


Neurone rose

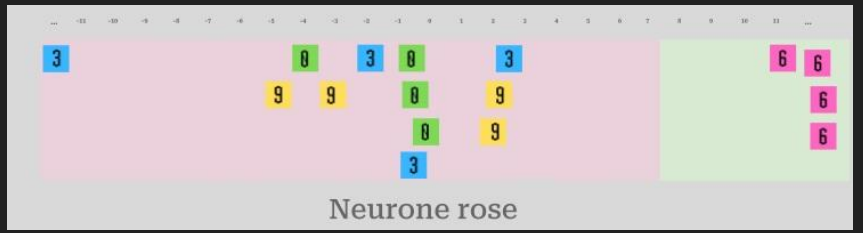
Neurone bleu

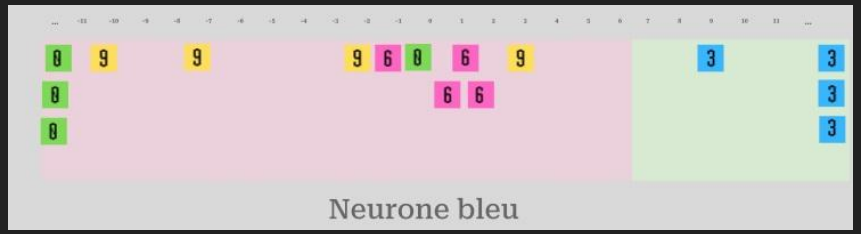
Neurone jaune

Après 20000 itérations

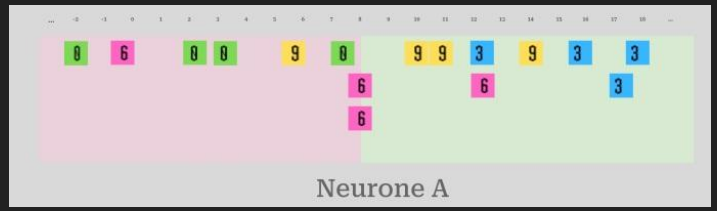


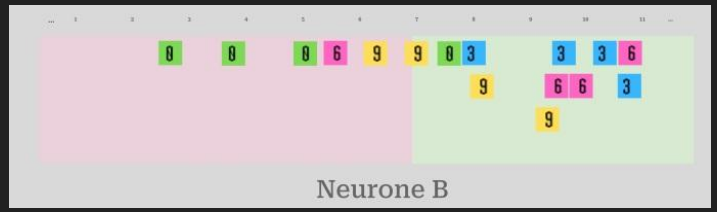
Après 50000 itérations

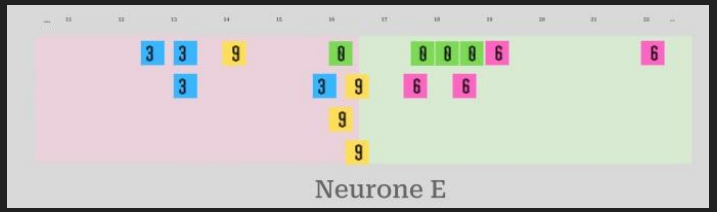


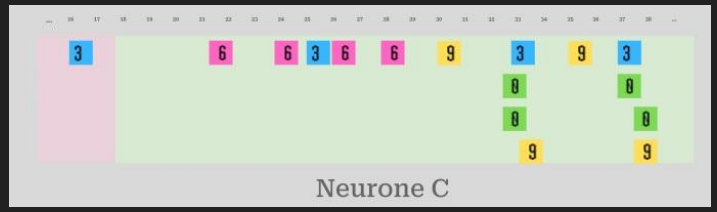


Après 50000 itérations









Neurone bleu

2 A + 1,2 B - 1,1 C + 0,5 D - 2,7 E + 3 F

Neurone E

0,7 😇 + 1,2 😍 + 👽