



AVIGNON  
UNIVERSITÉ

# Approches Neuronales – TP 3 Perceptron

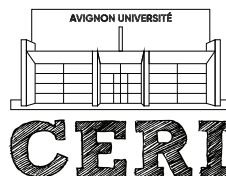
Sarra Bensafi

31 mars 2022

**Informatique**  
**Master1 Intelligence Artificielle**  
**UCE** Approches Neuronales

**Responsables**  
Juan-Manuel Torres

**UFR**  
**SCIENCES**  
**TECHNOLOGIES**  
**SANTÉ**



**CENTRE**  
**D'ENSEIGNEMENT**  
**ET DE RECHERCHE**  
**EN INFORMATIQUE**  
[ceri.univ-avignon.fr](http://ceri.univ-avignon.fr)

## Sommaire

Titre	1
Sommaire	2
1 Introduction	3
1.1 Données Utilisé . . . . .	3
2 Partie I	3
2.1 Calcul des erreurs( $E_a$ $E_g$ ) . . . . .	3
2.2 Calcul des stabilités et N+1 poids . . . . .	3
3 Partie II	4
3.1 Calcul des erreurs( $E_a$ $E_g$ ) . . . . .	4
3.2 Calcul des stabilités et N+1 poids . . . . .	4
Bibliographie	6

## 1 Introduction

Avec minimerror, le but est de trouver une séparation linéaire si elle existe mais aussi de trouver des surfaces localement discriminantes avec l'hyperparamètre Beta. ce dernier est modifié en ajoutant un recuit à chaque apprentissage

le Beta représente la contribution d'un exemple avec une stabilité à la fonction de coût, dans notre cas le bêta est petit 0.01 c'est-à-dire que la région autour de l'hyperplan est grand. La fonction de coût V suivante dépend d'un paramètre appelé température T qui est  $1/\beta$ .

$$\frac{\partial V(\gamma^\mu; \beta)}{\partial w_i} = -\frac{\beta}{4} \frac{\xi_i^\mu \tau^\mu}{\cosh^2(\frac{\beta \gamma^\mu}{2})}$$

**Figure 1.** Formule d'apprentissage Minimerror

### 1.1 Données Utilisé

Dans ce travail pratique j'ai testé l'algorithme minimerror avec 3 ensembles de données AND, OR et l'ensemble Sonar, je considérerais l'ensemble AND ET OR pour la première partie et l'ensemble Sonar pour la deuxième, mais dans le notebook python les résultats pour les trois ensembles de données sont fournis.

## 2 Partie I

Dans cette partie nous considérons une seule température T telle que  $T=1/\beta$ , le beta est de 0.01 c'est-à-dire que chaque exemple contribue au coût proportionnellement à sa stabilité car tous les exemples sont dans la région où le coût diminue linéairement.

### 2.1 Calcul des erreurs(Ea Eg)

L'erreur d'apprentissage Ea pour les 2 datasets :

- L'ensemble **AND** a  $E_a=0$  et  $E_g = 1.0$
- L'ensemble **OR** a  $E_a=0$  et  $E_g = 1.0$

### 2.2 Calcul des stabilités et N+1 poids

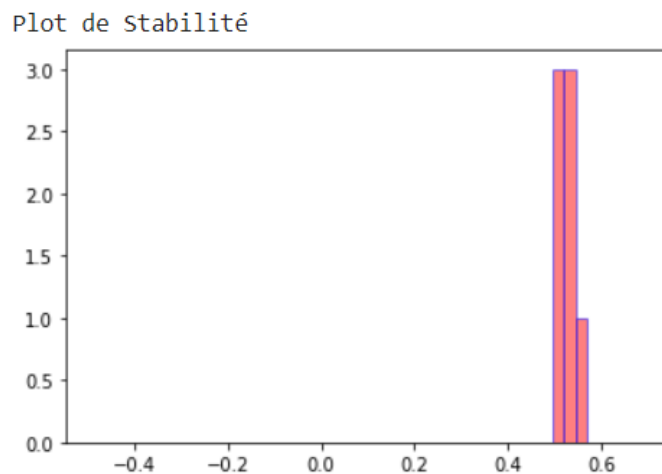
La stabilité pour les 2 datasets :

- Stabilité pour **AND** [[ 1.73205081 0.57735027 0.57735027 -0.57735027] [ 1.73205081 0.57735027 0.57735027 -0.57735027] [ 1.73205081 0.57735027 0.57735027 -0.57735027] [-1.73205081 -0.57735027 -0.57735027 0.57735027]]
- Stabilité pour **OR** [[ 0.57735027 -0.57735027 -0.57735027 -1.73205081] [ 0.57735027 -0.57735027 -0.57735027 -1.73205081] [ 0.57735027 -0.57735027 -0.57735027 -1.73205081] [-0.57735027 0.57735027 0.57735027 1.73205081]]

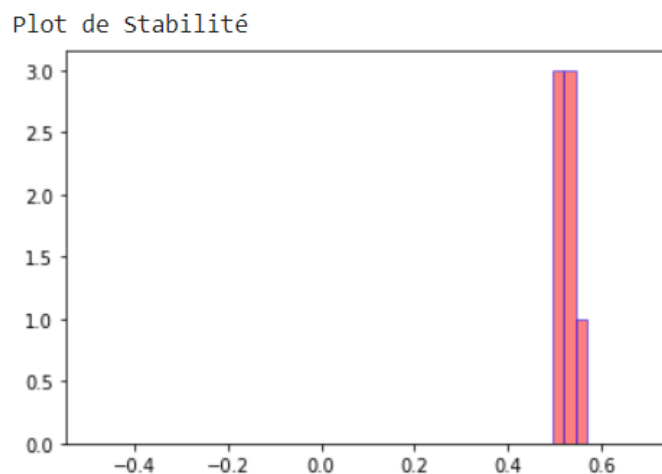
Le graphique de stabilité pour les 2 datasets :

Nous pouvons voir que les stabilités des deux graphique sont pour la plupart comprises entre 0,5 et 0,6 ce qui signifie que toutes les données sont bien classées et aucune partie des données est mal classées c'est-à-dire avec une stabilité négative.

- Le graphique de Stabilité pour **AND**



- Le graphique de Stabilité pour **OR**



les N+1 poids W du perceptron :

- Les Poids de **AND** :  $[-0.79266987 \ 0.79266987 \ 0.79266987]$
- Les Poids de **OR** :  $[2.82530634 \ 2.82530634 \ 2.82530634]$

### 3 Partie II

Dans cette partie nous considérons une température de deux  $\beta_{+}=0.01$  et  $\beta_{+} / \beta_{-} = 6$  puis  $\beta_{-}=0.001$ ,  $\beta_{+}$  est dédié pour les exemples de stabilité positive bien appris et  $\beta_{-}$  pour ceux avec une stabilité négative nous allons les augmenter au cours de l'apprentissage jusqu'à ce qu'ils deviennent assez grands pour que les exemples ne puissent pas contribuer au poids et donc au déplacement de l'hyperplan.

#### 3.1 Calcul des erreurs( $E_a$ $E_g$ )

- L'ensemble **Sonar** a  $E_a=3$  et  $E_g = 0.97$

#### 3.2 Calcul des stabilités et N+1 poids

La stabilité pour le dataset Sonar :

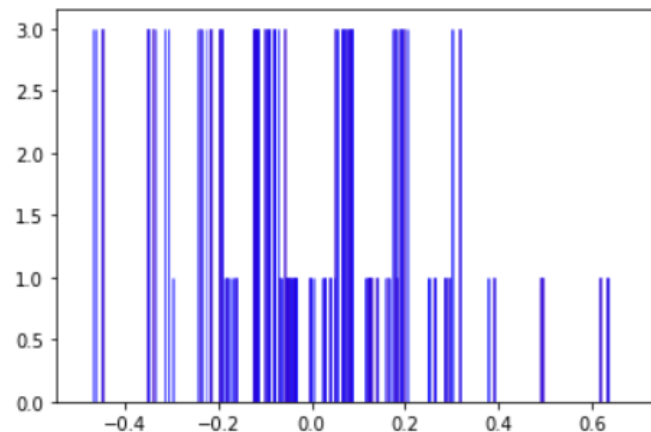
- Stabilité pour **Sonar** [[-0.05304748 -0.1235309 -0.16145586 -0.07194264 -0.04281963 -0.05526148 -0.11972178 -0.10141362 -0.17240494 -0.0600539 -0.21989211 -0.105686 -0.13258759 -0.05637807 -0.1008023 -0.19775607 -0.32514974 -0.280642 -0.32902819 -0.4908298 -0.32009512 -0.24714102 -0.55352546 -0.55803818 -0.13844269 -0.03819672 -0.07352346 -0.04518345 -0.09873309 -0.18226733 -0.2479677 -0.16564388 -0.28689432 -0.31140836 -0.24885414 -0.21825866 ..... -0.15648505 -0.06669513 -0.05703293 -0.1039376 -0.24595569 -0.21298669 -0.0357485 -0.22720802 -0.09751389 -0.09865845 -0.05147017 -0.05134141 -0.12442715 -0.0848115 -0.12005424 -0.08239524 -0.06932978 -0.06107405]]

Le graphique de stabilité pour le dataset Sonar :

Nous pouvons voir dans le graphique que les stabilités sont moins centrées et plus étalées avec des pics à plusieurs endroits mais une forte densité autour de 0.0, c'est-à-dire que de nombreux exemples sont proches de l'hyperplan, proches de la limite, mais dans notre cas, seuls 3 exemples ont été mal classés.

- Le graphique de Stabilité pour **Sonar**

Plot de stabilité



- Les Poids de **SONAR**

[-1.10984103e+02 1.28650980e+01 -1.71416929e+01 -1.10812062e+00 4.88030821e+01  
2.65064328e+01 2.25289503e+01 -3.24755760e+01 2.47232806e+01 8.73117468e+01  
1.03330084e+02 1.51598391e+02 1.16407203e+02 3.67393772e+01 -7.11130236e+01  
-6.13525299e+01 2.26001495e+01 2.37310241e+01 -6.73337646e+01 -3.66577898e+01  
9.99728775e+00 7.97996485e+01 6.55771467e+01 5.74829644e+01 1.77904941e+01 -  
1.02359440e+02 -9.07353311e+01 1.66738570e+01 6.47858584e+01 8.84121511e+00  
-4.79476680e+01 -4.89195147e+01 7.03471154e+01 -1.55328581e+00 -4.61889683e+01  
-2.95126784e+00 -9.39934350e+01 -9.26770447e+01 7.15546160e+00 1.06108640e+02  
-1.18434966e+01 5.92912588e+01 1.22102747e+01 1.13062243e+02 1.40898047e+02  
7.09857737e+01 7.64872312e+01 5.18180851e+01 2.14200742e+01 8.20074323e+00  
4.71137881e+00 2.28964646e+01 1.82727918e+01 7.38129835e+00 -1.37454710e+00  
-4.31646734e-01 4.32539427e+00 3.95994735e+00 5.91835844e+00 1.30975931e+01  
4.55260568e-02]

## Références

- [1] Juan-Manuel TORRES-MORENO (1992) *ΛT<sub>E</sub>X : Apprentissage et généralisation par des réseaux de neurones : étude de nouveaux algorithmes constructifs*