



Intelligence Artificielle et fonctions logiques

publié le 25/05/2022 - mis à jour le 16/06/2022

TraAM 2021 - 2022

Descriptif :

Scénario TraAM proposant la découverte du neurone artificiel et son utilisation pour la construction de fonctions logiques.

Sommaire :

- [Sommaire des TraAms "Intelligence Artificielle"](#)
- Caractéristiques du scénario
- Déroulement du scénario
- Réussites, obstacles et limites
- Prolongements

● [Sommaire des TraAms "Intelligence Artificielle"](#)

● [Caractéristiques du scénario](#)

○ [Thématique](#)

Découverte du neurone artificiel

○ [Niveau concerné](#)

Groupe de spécialité NSI et Spécialité Mathématiques ou option Mathématiques complémentaires.

○ [Problématique](#)

Est-il possible, à l'aide d'un unique neurone artificiel, de construire des fonctions logiques ? Si oui, quelles sont les limitations et comment les dépasser ?

Cette construction est-elle unique ?

○ [Contenu](#)

Étude de fonctions, continuité, fonction exponentielle et logarithme, système d'inéquations, programmation en Python : dictionnaires tuples et assertions.

○ [Objectifs](#)

Les objectifs sont :

- de démystifier l'IA et de comprendre le fonctionnement de l'élément de base (le neurone) et sa fonction,
- de montrer qu'un unique programme qui simule un neurone, peut réaliser différentes fonctions en changeant ses paramètres. Cela permet de limiter le temps de développement et modifie l'approche habituelle d'un programme pour un problème.

Remarques :

Dans le document donné aux élèves la formule de Z était $Z = \text{biais} \times w_0 + x_1 \times w_1 + x_2 \times w_2$ et il était indiqué en-dessous $\text{biais} = b_0$. Les élèves ont eu des difficultés à associer biais et b_0 , j'ai donc transformé la formule

$$Z = b_0 \times w_0 + x_1 \times w_1 + x_2 \times w_2$$

L'algorithme d'ajustement des poids ne sera pas abordé dans cette séquence, les élèves doivent trouver une solution du système d'inéquations par eux-mêmes.

La partie programmation est décomposée en deux parties :

- dans la première partie, je ne donne que le prototype de la fonction et ils doivent écrire la fonction et les tests nécessaires.
- dans la seconde je donne un fichier à compléter avec le dictionnaire des poids à définir, la fonction logique à compléter et les assertions pour vérifier le bon fonctionnement.

○ Nombre d'heures utilisées

Environ les trois quarts des élèves ont réussi en 2 heures à atteindre la dernière partie, deux ont tout fait. La séance suivante j'ai pris 15 minutes pour faire la synthèse et échanger sur leur perception de l'IA et ce qu'avait apporté la séquence.

○ Outils et ressources

- Un environnement de programmation Python, la seule bibliothèque utilisée est « math », donc n'importe quel environnement Python peut convenir.
- Une salle informatique pour tester les programmes en Python.

Tous les documents (fichiers .pdf et .py) :

-  [Énoncé du TD/TP : Introduction à l'IA](#) (PDF de 138.7 ko)
Intelligence Artificielle et fonctions logiques - TraAM 2021 - 2022 - Académie de Poitiers.
-  [Proposition de correction du TD : introduction à l'IA](#) (PDF de 111.1 ko)
Intelligence Artificielle et fonctions logiques - TraAM 2021 - 2022 - Académie de Poitiers.
-  [Programme Python introduction à l'IA : programme à compléter](#) (Zip de 370 octets)
Intelligence Artificielle et fonctions logiques - TraAM 2021 - 2022 - Académie de Poitiers.
-  [Programme Python introduction à l'IA : programme complété](#) (Zip de 631 octets)
Intelligence Artificielle et fonctions logiques - TraAM 2021 - 2022 - Académie de Poitiers.

○ Démarche de l'enseignant

En spécialité NSI, l'IA est un thème qui peut être abordé, j'ai cherché à construire une séquence qui permette de découvrir le fonctionnement du neurone et ses limitations. En première, on introduit la notion d'architecture machine et les différentes fonctions logiques. Dans un MOOC que j'ai partiellement suivi, on montrait le lien entre la logique et le neurone. C'est ce qui m'a conduit à construire cette séance en essayant d'y ajouter une partie mathématique abordable.

● Déroulement du scénario

La séance a eu lieu le 20/05 juste après les épreuves de spécialité, dans une salle informatique, la durée est de 110 minutes.

La séance précédente j'avais prévenu le groupe que nous allions travailler sur l'IA, je n'avais pas donné plus d'information.

Au début de la séance, je distribue le document de 3 pages : Introduction à l'IA.pdf.

Je rappelle rapidement le thème et leur demande de lire l'introduction et de répondre aux questions.

○ L'introduction :

La justification de la solution $f(z) = 0,5$ et la résolution de l'équation n'ont pas posé de problème. Tous les élèves ont réussi en prenant parfois des raccourcis ...

Un extrait de copie :

2a) f est continue sur \mathbb{R} car dérivable sur \mathbb{R} . De plus, elle est strictement croissante sur son intervalle de définition et $0 < 0,5 < 1$.
 Donc d'après le théorème des valeurs intermédiaires, l'équation corollaire du
 $f(y) = 0,5$ admet une unique solution sur \mathbb{R} .

2b) $f(y) = 0,5 \Leftrightarrow \frac{1}{1+e^{-y}} = 0,5 \Leftrightarrow 1 = 0,5 + 0,5e^{-y}$
 $\Leftrightarrow 1 = e^{-y} \Leftrightarrow \ln(1) = \ln(e^{-y}) \Leftrightarrow y = 0$
 0 est solution de l'équation $f(y) = 0,5$

Production élève : justification de la solution $f(z)=0,5$ et résolution de l'équation

Afin de m'assurer que les élèves comprennent ce qu'ils font, j'ai demandé de justifier le seuil de 0,5. La plupart ont écrit que c'était au milieu de l'intervalle $]0 ; 1[$ sans trop développer, sauf quelques rares copies :

c) La fonction logique admet en sortie unique ment deux états : Vrai ou Faux. On est définie sur \mathbb{R} avec comme $\lim_{z \rightarrow -\infty} f(z) = 0$ et $\lim_{z \rightarrow +\infty} f(z) = 1$.
 Il faut donc un seuil pour savoir si l'état est vrai ou faux. Pour qu'il y ait autant de chance qu'elle soit vraie que fausse on prend donc comme seuil $f(z) = 0,5$ car milieu de 0 et 1. Il n'était pas possible d'en choisir une autre valeur car sinon un état aurait plus de chance de se produire que l'autre.

Production élève : justification de la solution $f(z)=0,5$ et résolution de l'équation

Le tableau de cette première partie n'a pas posé de problème :

2/d)	S	Valeur de S	Valeur de Z
	Si S est vrai	$S \geq 0,5$	$Z \in [0, +\infty[$
	Si S est faux	$S < 0,5$	$Z \in]-\infty, 0[$

O La fonction logique ET

Je n'ai eu aucune question concernant la table de vérité de la fonction ET. (c'est la première séance après les épreuves de spécialités, cela pouvait être demandé au bac).

x_1	x_2	Etat logique de $S = x_1 \wedge x_2$	Valeur de Z	Valeur de S
0	0	0	< 0	$< 0,5$
1	0	0	< 0	$< 0,5$
0	1	0	< 0	$< 0,5$
1	1	1	≥ 0	$\geq 0,5$

Production élève : table de la vérité de la fonction ET

La résolution du système en a surpris plusieurs, ils ont su écrire le système mais n'ont pas osé prendre l'initiative de proposer des valeurs pour les poids. Je m'attendais à des valeurs de type 0, 1, 2, -1 etc...

Une copie m'a surpris :

f) $Z = -w_0 + x_1 \times w_1 + x_2 \times w_2$

c) pour $x_1 = 0$ et $x_2 = 0$:
 $Z < 0$
 $-w_0 + x_1 \times w_1 + x_2 \times w_2 < 0$
 $-w_0 < 0$

pour $x_1 = 0$ et $x_2 = 1$:
 $-w_0 + x_1 \times w_1 + x_2 \times w_2 < 0$
 $-w_0 + w_2 < 0$

pour $x_1 = 1$ et $x_2 = 0$:
 $-w_0 + x_1 \times w_1 + x_2 \times w_2 < 0$
 $-w_0 + w_1 < 0$

pour $x_1 = 1$ et $x_2 = 1$:
 $-w_0 + x_1 \times w_1 + x_2 \times w_2 < 0$
 $-w_0 + w_1 + w_2 > 0$

d) pour $x_1 = 0$ et $x_2 = 0$:
 $-w_0 < 0$ $w_0 = 33$
 $-33 < 0$

pour $x_1 = 0$ et $x_2 = 1$:
 $-w_0 + w_2 < 0$ $w_0 = 33$
 $-33 + 13 < 0$ $w_2 = 13$
 $-20 < 0$

pour $x_1 = 1$ et $x_2 = 0$:
 $-w_0 + w_1 < 0$ $w_0 = 33$
 $-33 + 29 < 0$ $w_1 = 29$
 $-4 < 0$

pour $x_1 = 1$ et $x_2 = 1$:
 $-w_0 + w_1 + w_2 > 0$
 $-33 + 29 + 13 > 0$
 $9 > 0$

Production élève : présentation des poids

Il a choisi $w_0 = 33$, $w_1 = 29$ et $w_2 = 13$. La comparaison des poids obtenus entre eux a permis de montrer le large choix possible.

○ L'implémentation en Python :

Je voulais qu'ils créent la totalité de la fonction sans être guidés. J'avais mis à disposition le fichier *neuronal_logic_el.py* au début de la séance pour la question 4. Certains se sont inspirés de ce fichier, peut-être faudrait-il ne le mettre à disposition que lorsqu'il est nécessaire.

Deux exemples d'implémentation :

```
### ET ###
w0_ET=33
w1_ET=29
w2_ET=13
def et(x1:int, x2:int):
    Z=-w0_ET+x1*w1_ET+x2*w2_ET
    if Z<0:
        return 0
    else:
        return 1

assert et(0,0)==0
assert et(1,0)==0
assert et(0,1)==0
assert et(1,1)==1
```

Exemple d'implémentation : fonction ET

```
def et(x1,x2):
    formule=(-1*2)+(x1*1)+(x2*1)
    formule2=1/(1+math.exp(-formule))
    return(formule2>=0.5)
```

Exemple d'implémentation : fonction ET

Dans la version de gauche la fonction d'activation n'a pas été prise en compte mais la fonction est correcte. J'ai eu quelques questions sur les tests nécessaires pour vérifier le bon fonctionnement. Souvent, ils ont fait le choix de tester la fonction dans la console Python.

○ La fonction logique OU

Les élèves ont procédé de la même façon que pour le ET de façon autonome avec succès, sauf un.

Un élève a choisi $w_0 = 1$, $w_1 = 1$ et $w_2 = 1$. Il a implémenté la fonction en Python qui fonctionne parfaitement. Dans le sujet, je demande ensuite les coefficients de la fonction NON OU. Il m'appelle et me dit : « la fonction NON OU c'est l'inverse de la fonction OU, donc il faut inverser les coefficients. Mais le programme Python ne fonctionne

pas ».

Il avait donc choisi l'opposé pour implémenter le NON OU : $w_0 = -1$, $w_1 = -1$ et $w_2 = -1$.

Initialement, j'ai choisi S vrai si $Z \geq 0$ et S faux si $Z < 0$, les niveaux ne sont donc pas symétriques et dans ce cas particulier cela complique un peu les choses.

Peut-être faudrait-il imposer S vrai si $Z > 0$ et S faux si $Z < 0$?

Exemples d'une implémentation des fonctions logiques OU et NON OU :

```
def ou(x1,x2):
    z=-1+x1*2+x2*2
    s = 1/(1+math.exp(-z))
    if s<0.5:
        return 0
    else:
        return 1
```

Programme en Python pour la fonction OU

```
def nonOu(x1,x2):
    z=1+x1*-2+x2*-2
    s = 1/(1+math.exp(-z))
    if s<0.5:
        return 0
    else:
        return 1
```

Programme en Python pour la fonction Non OU

Les élèves n'ont éprouvé aucune difficulté avec la partie 5.

○ Le OU exclusif partie 6

Les rares élèves qui ont abordé cette dernière partie n'ont pas trouvé de solution au système et m'ont demandé confirmation. Avec un peu d'aide, ils ont compris qu'il fallait plusieurs neurones.

Deux élèves ont eu le temps d'implémenter le OU exclusif, j'ai rappelé comment construire un XOR à l'aide de portes logiques et j'ai proposé deux schémas. Cela a donné l'implémentation suivante par les élèves :

```
def ou_exclusif(x1: int, x2: int) -> int:
    no_x1 = logic_function(weight["non ou"], x1, x1)
    no_x2 = logic_function(weight["non ou"], x2, x2)

    et_1 = logic_function(weight["et"], no_x1, x2)
    et_2 = logic_function(weight["et"], no_x1, x2)

    xor = logic_function(weight["ou"], et_1, et_2)
    return xor
```

Fonction Python programmant OU exclusif

```
def xor(w,x1,x2):
    a=logic_function(w["non ou"],x1,x2)
    h=logic_function(w["non ou"],x1,a)
    b=logic_function(w["non ou"],a,x2)
    return logic_function(w["ou"],h,b)
```

Fonction Python programmant le OU exclusif

● Réussites, obstacles et limites

Remarques :

- Il est possible de se passer du biais qui n'apporte rien ici, mais qui permet d'être fidèle au modèle historique.
- De même, la fonction d'activation n'est pas nécessaire comme l'a montré l'implémentation de certains élèves. Cela permet aussi de décliner le TP en classe de 1ère générale spécialité NSI et spécialité mathématiques.

Synthèse :

Nous avons passé 15 minutes, lors du cours suivant, à échanger sur le sujet de l'IA et du deep-learning. Dans l'ensemble, ils sont satisfaits et ont bien compris le rôle du neurone. J'ai eu beaucoup de questions sur comment déterminer les poids automatiquement.

● Prolongements

Étant donné que j'ai eu quelques questions concernant le calcul des poids, il pourrait être intéressant d'ajouter une

partie sur le calcul automatique des poids à l'aide d'un tableur ou en python. Cela permettrait de comprendre :

- la nécessité d'avoir de nombreux échantillons,
- l'initialisation des poids qui est en général choisie au hasard et qui conduit à une variété de solutions,
- les contraintes matérielles qu'impose le deep-learning.



**Académie
de Poitiers**

Avertissement : ce document est la reprise au format pdf d'un article proposé sur l'espace pédagogique de l'académie de Poitiers.

Il ne peut en aucun cas être proposé au téléchargement ou à la consultation depuis un autre site.