C'est toi le neurone !

... une adaptation de Marie Duflot-Kremer
d'après une activité de Teaching London Computing
créée en mars 2019

dernière modification le 2 avril 2020

Le document que vous êtes en train de consulter n'est pas une référence très finalisée, ni un guide strict à suivre. Il regroupe des infos que vous pourriez trouver utiles si vous envisagez d'animer cette activité. Comme l'activité est inspirée de "the brain in a bag" disponible sur le site teachinglondoncomputing.org, elle est soumise à leur licence, à savoir CC-BY-NC-SA. Vous pouvez utiliser, imprimer ou modifier le matériel, dès lors que vous mentionnez la source (en l'occurence moi si vous utilisez ma variante, mais surtout Paul Curzon, Queen Mary University of London avec le soutien du maire de Londres pour le site Teaching London Computing : http://teachinglondoncomputing.org), si vous n'en faites aucun usage commercial, et si vous partagez votre travail sous la même licence. Vous pouvez en particulier trouver sur leur site la page de l'activité (en anglais). En ce qui concerne ma version de l'activité, vous pouvez suggérer des améliorations/enrichissements à `marie.duflot-kremer@loria.fr`. La page https://members.loria.fr/MDuflot permet de trouver (section médiation/activités) une liste d'autres activités pour faire découvrir différents aspects de l'informatique, réalisables en grande majorité sans ordinateur.

Étape 1 — Le contexte

Dans cette activité, on va découvrir comment fonctionne le cerveau, en envoyant des impulsions électriques (aussi appelées influx) entre les neurones. Les participants vont selon les cas soit incarner un neurone, soit en observer notre réseau de neurones pour découvrir la tâche réalisée par notre "cerveau". Dans un deuxième temps il est possible de voir comment un neurone peut "apprendre" le comportement à tenir pour que le réseau effectue une fonction donnée.

Étape 2 — Préparer le matériel

Dans la version originale de l'activité, il y a 7 neurones et du coup besoin de moins de matériel. Vous pouvez aller vous inspirer de leur site si vous voulez un plus petit réseau.

Pour ma part j'utilise :

- une bonne vingtaine de mètres de ficelle/cordelette/bolduc... (ou d'un peu plus d'une pelote de laine, un crochet et quelques heures à tuer)
- de 14 objets qui peuvent glisser le long de la cordelette. Cela peut être des morceaux de tuyau en plastique (au rayon plomberie), des anneaux à rideaux, des ronds de serviette ou même des (demi, pour avoir des tubes plus courts) tubes de rouleaux de papier toilette[1]. Bien veiller à ce que le diamètre de vos objets ne soit pas trop petit (pour la mise en place on devra faire passer plusieurs fois la cordelette dans le même objet).
- les cartes représentant les formes, à imprimer en deux exemplaires
- les cartes donnant les actions à faire par les neurones. Il y en a 15 (autant que de neurones) et sur chacune on écrit un message, en l'occurence voici la répartition :
  - je dis SNAP si je reçois deux influx (une fois)

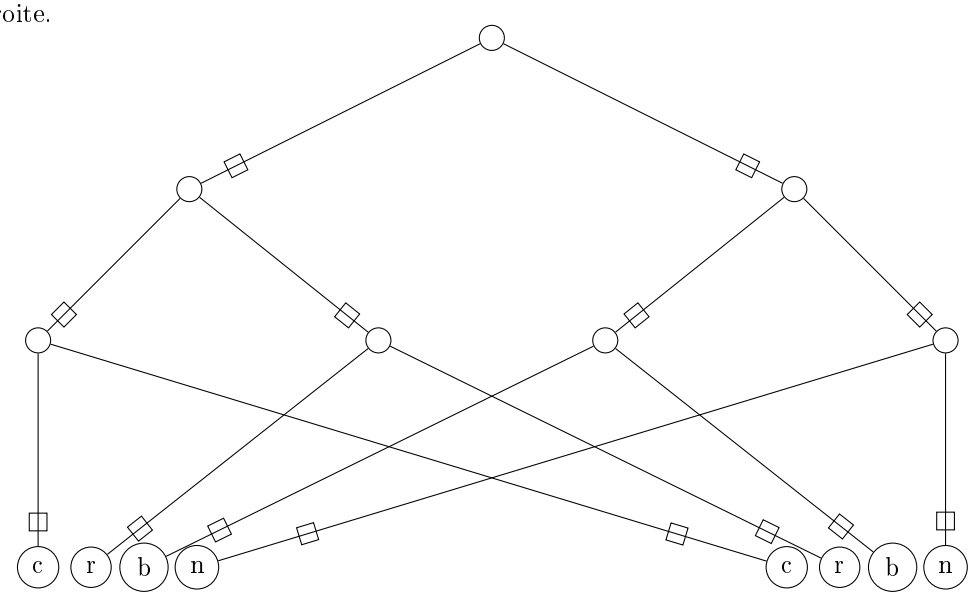
1. cela fonctionne bien si votre cordelette est bien lisse, sinon le carton accroche

— j'envoie un influx si je reçois au moins un influx (2)
— j'envoie un influx si je reçois deux influx (4)
— j'envoie un influx si je vois un carré (2)
— j'envoie un influx si je vois un rond (2)
— j'envoie un influx si la figure est noire (2)
— j'envoie un influx si la figure est blanche (2)

Réaliser les cartes des neurones (je les fais sur des fiches cartonnées pour plus de résistance).

En coupant et nouant votre cordelette, réaliser le réseau présenté ci-dessous : ici chaque cercle représente l'emplacement d'un neurone (et pour la plupart les endroit où vous allez devoir faire vos noeuds). Les croisements des fils en bas (sans cercle au croisement) montrent juste comment les fils seront disposés au final, mais il ne faut surtout pas y faire de noeuds (sinon vos influx nerveux ne vont pas passer ! ! !).

Les 8 neurones du bas sur le dessin jouent le rôle de récepteurs (ils sont reliés aux yeux si vous voulez). Comme vous le voyez, ils sont disposés en deux groupes, l'un à droite et l'autre à gauche. C'est fait exprès, pour que vous puissiez montrer une carte à tous ceux de gauche et une autre à tous ceux de droite.



Étape 3 — Mais que fait ce réseau ?

Prendre 15 volontaires, leur faire jouer les neurones en se mettant dans notre réseau un par cercle (donc aux croisements de ficelles et aux extrêmités). Glisser les anneaux le long du fil pour qu'il y en ait un pour chaque neurone sauf celui du haut, comme indiqué par les carrés sur le dessin. Puis distribuer les cartes action (ce qu'ils doivent faire) à chacun comme indiqué ci-dessous. Attention, les neurones doivent prendre connaissance de leur rôle sans le montrer aux autres.

Pour distribuer les cartes aux neurones, il faut donner les cartes mentionnant les carrés (resp. rond, blanc et noir) aux neurones portant la lettre c (resp. r, b et n) donc aux neurones de la ligne du bas sur le dessin. Pour les 4 neurones de la ligne d'au-dessus, donner les 4 cartes "si je reçois deux influx". Au rang d'au-dessus les cartes ("... si je reçois au moins un influx") et enfin la carte SNAP au neurone le plus en haut sur le dessin.

Vous allez ensuite utiliser les cartes représentant des images (carrés ou ronds), en prendre deux parmi les 2 x 8 = 16 que vous avez imprimées et en montrer une aux 4 récepteurs de gauche et une autre aux 4 récepteurs de droite. Chaque neurone va effectuer sa tâche (envoyer un influx ou pas, dire quelque chose ou non) en fonction de ce qu'il voit ou reçoit. Suivant les cartes, les neurones envoient ou pas leur influx, et dans certains cas le neurone de tête du réseau dit SNAP.

L'objectif pour le public, à savoir les personnes qui ne font pas partie du réseau est de comprendre ce qu'il se passe. Pour comprendre ce que font les récepteurs, c'est facile : après quelques expériences le public peut deviner qui réagit à quoi (quelle forme ou quelle couleur). Vous pouvez même confier au public les cartes avec les images et c'est à lui de choisir les deux cartes à montrer aux deux groupes de

récepteurs pour tester s'il a compris ce qu'il se passe. Ensuite il faut essayer de comprendre ce que font les neurones de la deuxième couche (ceux connectés aux récepteurs) puis de la troisième couche et enfin de comprendre dans quel cas le neurone de tête dit SNAP et dans quel cas non.

La réponse à cette question est la suivante : pour les récepteurs on sait ce qu'ils font. Pour les neurones de la deuxième couche, chacun est connecté à deux récepteurs qui détectent la même caractéristique (soit une couleur donnée, soit une forme donnée) mais l'un pour la première carte et l'autre pour la deuxième. Et ce neurone dit si les deux cartes ont toutes les deux la caractéristique en question. De gauche à droite ces neurones de la deuxième couche transmettent un influx si les deux cartes montrent un carré, un rond, une figure blanche ou une figure noire. On pourrait les appeler le neurone carré, le neurone rond, le neurone blanc et le neurone noir. Dans la troisième couche on a deux neurones, l'un en charge de la forme (il reçoit et donc renvoie un influx si soit on lui dit que les deux figures sont des ronds soit qu'elles sont toutes les deux des carrés) et l'autre en charge de la couleur (il envoie un influx si les deux figures sont de la même couleur). Enfin le neurone de tête crie SNAP si et la couleur et la forme sont identiques entre les deux cartes, et appartiennent aux valeurs prévues (rond, carré, blanc, noir). En d'autres mots, le neurone de tête crie SNAP si les deux cartes sont les mêmes.

Et si je montre aux neurones deux cartes identiques mais qui ne sont pas dans nos formes/couleurs prévues, par exemples deux triangles verts ? Eh bien dans ce cas nos neurones récepteurs ne vont pas retrouver la caractéristique qu'ils attendaient, et ne vons pas réagir. Le réseau est bien spécifique à une tâche et si on s'en détourne un tout petit peu, il ne sait plus faire ! !

Étape 4 — Compétition de réseaux

Si l'on a assez de matériel, ou éventuellement en faisant de plus petits réseaux (à 7 neurones en testant juste la couleur), on peut faire fonctionner deux réseaux en même temps, et voir quel réseau est le plus rapide (sans se tromper ! ! !). Si on veut faire cela il faut trouver un moyen de décrocher les éléments de notre réseau. Ici si on enlève le neurone de tête et les deux fils qui y sont connectés, on a deux petits réseaux à 7 neurones chacun, utilisables pour calculer des fonctions plus simples. Comme le montre le site cité en haut du document, avec 7 neurones on peut faire un jeu de SNAP juste sur les couleurs (4 récepteurs, 2 qui détectent le blanc, deux le noir) puis en deuxième couche un neurone pour le blanc un pour le noir, puis le neurone de tête qui dit SNAP s'il reçoit au moins un influx.

On peut alors faire des compétitions, en montrant les deux mêmes cartes aux deux réseaux en même temps, et en voyant quel réseau crie SNAP le plus vite (et uniquement quand c'est nécessaire).

Étape 5 — Notre réseau peut-il apprendre ?

L'intérêt d'un réseau de neurones en informatique c'est quand même sa capacité à apprendre à réaliser une fonction. Pour le moment notre réseau a été livré tout fonctionnel, chaque neurone sachant parfaitement ce qu'il doit faire, et on a juste essayé de comprendre ce qu'il faisait.

Il est possible de découvrir une version (très simplifiée, voir section 6) de l'apprentissage avec une extension assez simple de notre activité. L'idée, pour laisser entrevoir comment l'apprentissage fonctionne, c'est de prendre un réseau pour lequel : la plupart des neurones ont une règle établie, mais un ou deux ne reçoivent aucune instruction.

On peut prendre un réseau à 7 neurones (donc qui a la même forme que celui donné en annexe) mais dont le but est de détecter si deux cartes sont différentes (sur la forme ou sur la couleur).

On aura 4 récepteurs, deux pour chaque carte. Pour chaque carte il y aura un détecteur de carré et un détecteur de blanc. Les deux réponses des détecteurs de carrés (idem pour les deux réponses des détecteurs de blanc) vont arriver à un neurone à qui on ne va pas dire quoi faire. Les neurones de la deuxième couche sont donc ceux qui doivent apprendre. Le neurone au sommet de notre réseau dira SNAP s'il reçoit au moins un influx.

Les deux neurones de la deuxième couche vont avoir une feuille telle que donnée en annexe. Dans notre réseau, le neurone est relié à deux neurones de la couche précédente. Il peut donc recevoir 4 informations (où on suppose que à signifie pas d'influx et 1 signifie un influx) : 0 1, 1 0, 0 0 et 1 1. Au départ il ne sait pas comment réagir et donc il choisit par exemple de ne pas jamais envoyer d'influx (quelle que soit l'entrée). Sur la fiche cela consiste à noter en face de chaque entrée possible la valeur 0 (pas d'influx).

Pour la phase d'apprentissage, on va montrer des cartes aux récepteurs, laisser le réseau agir et au final dire si le neurone de tête a bien réagi (dire SNAP ou pas). S'il a bien réagi, les neurones qui apprennent se disent qu'ils ont probablement fait le bon choix, et ne pas changer leur comportement. S'ils se trompent, ils vont changer leur réaction à cette entrée.

Par exemple le neurone voit 0 et 1, sa carte lui dit de ne pas envoyer d'influx, et on lui dit que le réseau a bien agi. Il va du coup décider de ne rien changer. Si par contre on lui dit que le réseau s'est trompé, il va changer sur sa carte et en face de l'entrée 0 1 écrire 1 (= j'envoie un influx)

Et on continue ainsi, en faisant tester différentes combinaisons d'entrées pour que le réseau apprenne.

Questions :

1. Si on se débrouille bien, en combien d'essais peut-on apprendre le comportement des deux neurones ? Sur notre exemple, en deux coups (on envoie d'abord 01 et 01 aux deux neurones, ils n'envoient rien, c'est une erreur, ils changent, puis on leur envoie 10 et 10, pareil erreur, ils changent et après ne se tromplent plus jamais)
2. Avec cette méthode, le réseau va-t-il forcément arriver à un état où il connait la bonne réaction pour chaque entrée ? Non, si on choisit exprès les entrées qu'on lui montre, il peut soit ne jamais tester certains cas (et donc ne pas savoir y répondre) soit faire un cycle qui lui fait changer sans fin la valeur de la réponse (voir section suivante). Mais si on fait les choses soit pour aider notre réseau soit au hasard, alors le réseau va apprendre.

Étape 6 _____ Et hop, c'est de l'informatique !

Tout le monde a entendu parler de réseaux de neurones, d'intelligence artificielle, c'est même un sujet très à la mode. L'activité essaie de montrer un aperçu de ce que ça peut être. Seulement dans un vrai réseau de neurones, les messages transmis par les neurones ne sont pas binaires (vrai ou faux, influx ou pas d'influx) mais numériques (une certaine valeur) et chaque neurone va choisir de donner plus ou moins de poids à chacune des valeurs reçues, calculer une fonction à partir de ces valeurs et la renvoyer au neurone de la couche suivante. Lors de la phase d'apprentissage, on va comparer la valeur calculée par le réseau à la valeur qu'on attendait, calculer la différence entre les deux et renvoyer cette information aux neurones qui vont modifier un peu leur façon de réagir pour essayer de s'approcher du résultat souhaité. En principe plus un neurone est loin de la sortie du réseau, plus il a une responsabilité faible dans l'erreur, et moins il va modifier sa fonction (la réponse qu'il envoie en fonction des entrées qu'il reçoit). En effectuant pleiiiiin d'étapes de ce genre, la fonction calculée par chacun des neurones est alors sensée s'ajuster afin que le réseau soit plus proche du résultat attendu.

Ici c'est compliqué à réaliser car cette méthode d'apprentissage s'applique plutôt pour des réseaux numériques avec poids, mais la partie de l'activité décrite dans la section 5 s'en approche un peu.

Notre expérience d'apprendissage, elle fonctionne ? La bonne nouvelle est que, si à chaque étape d'apprentissage on tire les cartes au hasard, la probabilité que les neurones apprenants finissent par faire ce qu'il faut tous les deux est 1. Et une fois que les deux ont la bonne réaction pour chaque couple de valeurs d'entrée, ils ne vont plus jamais se tromper et plus jamais changer de comportement. La nouvelle un peu moins bonne est que le temps moyen (le nombre d'expériences qu'il faut en moyenne pour y arriver) est de 39,5.

Comme 39.5 est un peu lent pour une expérience menée avec des élèves, on peut essayer de le faire un peu pour expérimenter, puis de se poser et de réfléchir pourquoi ça marche. Tout d'abord à tout moment on peut regarder les tables de vérité des deux neurones apprenants. Ils ont chacun 4 entrées possibles (00 01, 10 et 11) et une réponse à donner pour chacune de ces cas. Il y a donc en tout 8 valeurs (4 pour chaque neurone) à "apprendre"

A l'issue d'une étape d'apprentissage, on a trois possibilités :

1. le réseau a donné la bonne réponse, auquel cas on ne touche à rien, et donc on n'a pas progressé mais pas régressé non plus : le nombre de valeurs correctes est resté stable
2. le réseau s'est trompé mais un seul des deux neurones avait tort. Comme on change la réponse pour les deux, dans ce cas après le changement on aura une valeur correcte de plus mais aussi une valeur fausse de plus, ce qui fait un bilan nul,

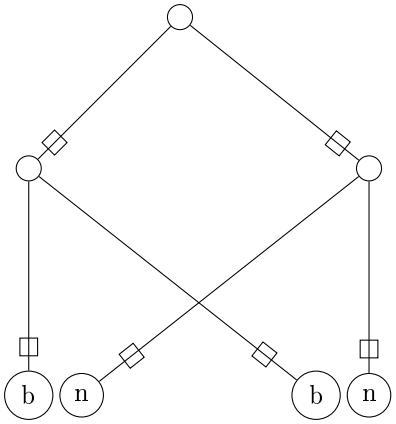


FIGURE 1 – Réseau à 7 neurones, à utiliser pour la compétition entre deux réseaux. Ici les neurones récepteurs (en bas) détectent la couleur (blanc ou noir, b ou n), ceux de la deuxième couche envoient un influx s'ils reçoivent deux influx, et le neurone de tête dit SNAP s'il reçoit un influx (il est impossible qu'il en reçoive deux car cela voudrait dire que les deux images sont toutes les deux noires ET toutes les deux blanches).

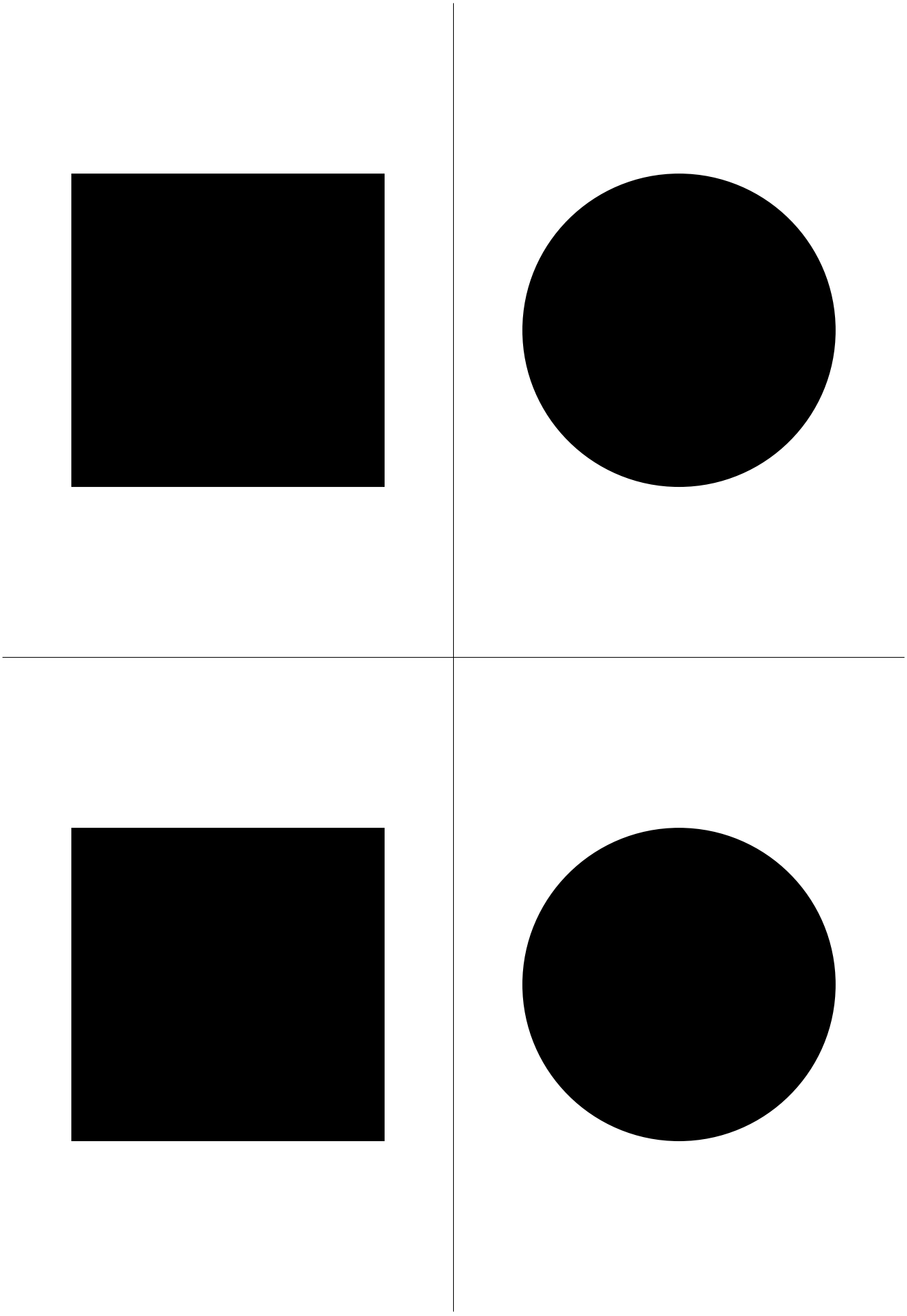
3. soit les deux neurones s'étaient trompés, et ils corrigent tous les deux leur réponse et on a deux valeurs correctes de plus qu'avant.

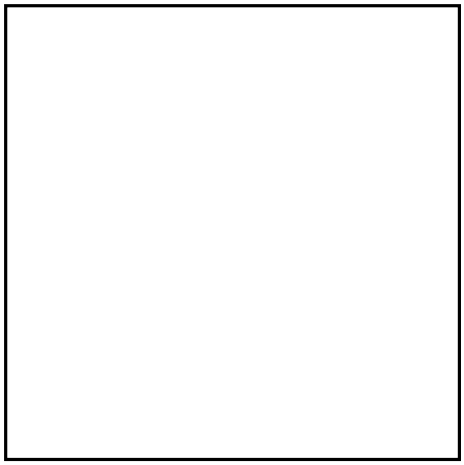
Dans tous les cas on ne peut régresser (avoir plus de valeurs fausses qu'avant), et on a une probabilité non nulle de progresser (plus de valeurs correctes qu'avant) [2]. Au final on va arriver à une situation où toutes les valeurs sont bonnes, où le réseau va donc donner la bonne réponse à chaque fois, et où les valeurs ne vont plus jamais changer.

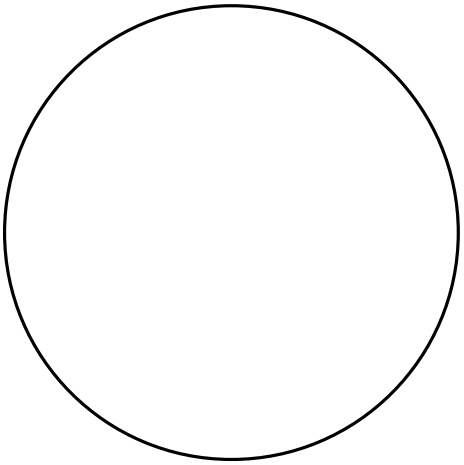
Dans un tel système, la rapidité d'apprentissage dépend des exemples que l'on montre à notre réseau (dans le meilleur des cas en deux coup le réseau peut avoir tout appris) mais aussi de la configuration initiale. Dans notre cas on a choisi qu'au départ nos neurones répondent "Faux" (ou 0, ou pas d'influx) tout le temps. Sur ce même exemple si on avait choisi qu'ils disent au départ "Vrai" tout le temps, le temps moyen d'apprentissage serait descendu à 28. Ces résultats ont été obtenus à l'aide du model-checker Prism. Il est même possible de construire un graphe donnant la probabilité d'apprentissage en fonction du nombre d'étapes d'apprentissage. Les graphes ont également été tracés à l'aide du model-checker PRISM mais en mode statistique cette fois, le système étant trop grand (avec une variable qui compte le temps et qui augmente nettement le nombre d'états) pour utiliser une méthode de vérification exacte.

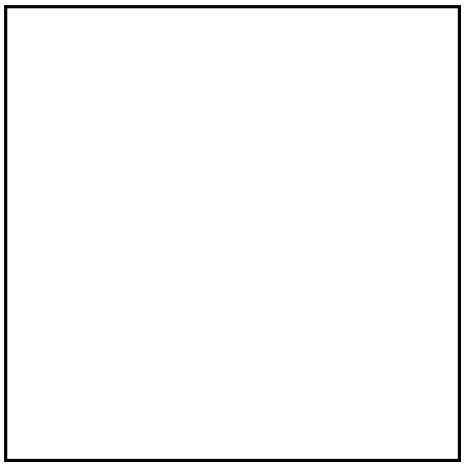
Étape 7 — Les annexes

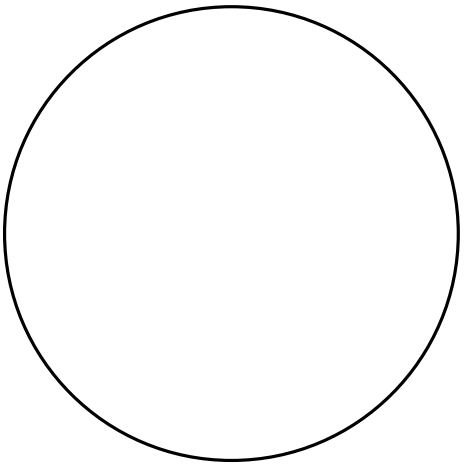
2. Attention, ce cas est très simplifié, et ne fonctionne que parce qu'au départ le nombre de valeurs erronnées est pair. S'il était impair, on finirait par arriver à une situation où on a une seule valeur erronnée, et l'apprentissage soit ne changerait aucune valeur (cas 1 ci-dessus), soit en changerait deux (cas 2 ci-dessus), corrigerait l'ancienne valeur erronnée mais changerait également une valeur anciennement correcte, annulant ainsi tout bénéfice.











Valeurs reçues :		Influx envoyé
gauche	droite	
0	1	oui - non
1	0	oui - non
1	1	oui - non
0	0	oui - non

Valeurs reçues :		Influx envoyé
gauche	droite	
0	1	oui - non
1	0	oui - non
1	1	oui - non
0	0	oui - non

FIGURE 2 – tables à donner aux neurones qui doivent apprendre. Ils peuvent marquer la réponse prévue pour une valeur d'entrée donnée en mettant un objet ou un bout de pâte collante sur la réponse choisie pour le moment (oui ou non