

Fondements de l’informatique. Examen

Durée: 3h

Sujet proposé par Olivier Bournez
Version 6

Les 4 parties sont indépendantes, et peuvent être traitées dans un ordre quelconque. On pourra admettre le résultat d’une question pour passer aux questions suivantes. On pourra utiliser tous les résultats et les théorèmes démontrés ou énoncés en cours ou en petite classe ou dans le polycopié sans chercher à les redémontrer.

Il est possible d’avoir la note maximale sans répondre à toutes les questions. La difficulté des questions n’est pas une fonction linéaire ni croissante de leur numérotation.

La **qualité et clarté de votre argumentation et de votre rédaction** sera une partie importante de votre évaluation.

1 Décidabilité

On considère des machines de Turing qui travaillent sur un alphabet qui contient au moins les chiffres $0, 1, \dots, 9$, et les lettres de l’alphabet latin, et le symbole $_$.

Question 1. Est-ce que l’on peut décider si une machine de Turing M accepte le mot INF412 et rejette le mot CSC_41012_EP ?

Question 2. Est-ce que l’on peut décider si une machine de Turing M termine et refuse sur le mot CSC_41012_EP en 41012 étapes, mais accepte et termine toujours à sa 412ième étape sur toute autre entrée ?

Question 3. On se donne une machine de Turing M qui termine sur toutes ses entrées. Peut-on décider si M termine en temps polynomial sur toutes ses entrées ?

Question 4. Le problème de décision de la question précédente est-il récursivement énumérable ?

2 Réseaux de neurones à seuil

On note $\mathbf{B}$ pour $\mathbf{B} = \{0, 1\}$: l’intuition est que 0 représente la valeur logique *faux*, et que 1 représente la valeur logique *vrai*. On considère des réseaux de neurones dont la fonction de transition est la *fonction de Heaviside*, c’est-à-dire la fonction $\mathcal{H}(x)$ qui vaut 1 pour $x \geq 0$, 0 sinon.

Un *neurone* à n entrées est décrit par des paramètres $w_1, w_2, \dots, w_n$ et $h \in \mathbb{Q}$. Chacun des w_i est appelé un *poids*, et h est appelé le *seuil* du neurone. On dit que les poids sont unitaires¹ si $w_1, w_2, \dots, w_n$ restent dans $\{-1, 0, 1\}$. Un tel neurone calcule une fonction de $\mathbb{Q}^n$ dans $\mathbb{Q}$ de la façon suivante : sur les entrées $x_1, x_2, \dots, x_n$, il prend la valeur $\mathcal{H}(w_1x_1 + w_2x_2 + \dots + w_nx_n - h)$. Cette valeur s’appelle sa *valeur d’activation*. On dit qu’une fonction $F : \mathbf{B}^n \rightarrow \mathbf{B}$ est une *fonction booléenne linéaire à seuil* si elle correspond à la fonction calculée par un neurone à seuil : dit autrement, il existe des rationnels $w_1, \dots, w_n$ et h tels que pour tout $\mathbf{x} = (x_1, x_2, \dots, x_n)$, on a $F(\mathbf{x}) = 1$ si et seulement $\sum_{i=1}^n w_i x_i \geq h$.

1. On n’impose rien sur le seuil. Il peut être quelconque.

Par exemple, la fonction ET logique (à n -arguments) que l'on peut nommer $\text{AND}_n(x_1, \dots, x_n)$, qui vaut 1 si et seulement si chacun des x_i vaut 1, est une fonction booléenne linéaire à seuil (et même à poids unitaires) car $\text{AND}_n(x_1, \dots, x_n) = 1$ si et seulement si $1x_1 + 1x_2 + \dots + 1x_n \geq n$.

Question 5. Montrer que le OU logique (à n -arguments) OR_n , la NEGATION logique NOT (à un argument) sont également des fonctions booléennes linéaires à seuil.

On obtient un réseau de neurones en combinant des neurones en couches comme on s'y attend².

2.1 Sans fixer l'architecture

Question 6. Montrer que le problème de décision suivant est NP-complet : on se donne un réseau de neurones avec 1 sortie. S'il possède n entrées, on veut savoir si cette sortie est 1 sur au moins un élément de $\mathbf{B}^n$.

Dans la question précédente, on ne cherchait pas à fixer l'architecture. On va montrer que cela reste vrai, même si l'on fixe l'architecture.

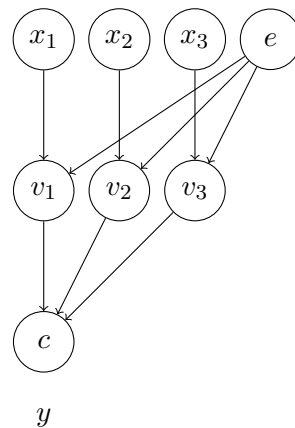
2.2 En fixant l'architecture

Considérons un réseau de neurones à seuil à n entrées et m sorties. Une tâche est un élément de $\mathbf{B}^n \times \mathbf{B}^m$. On dit que le réseau est compatible avec la tâche $(x_1, \dots, x_n, y_1, \dots, y_m)$ si l'on obtient la sortie $(y_1, \dots, y_m)$ quand on évalue le réseau sur l'entrée $(x_1, \dots, x_n)$.

On s'intéresse au problème de décision suivant APPRENTISSAGE : étant donnée une architecture (un graphe) d'un réseau de neurones et une liste finie de tâches, déterminer si l'on peut trouver des poids unitaires et des seuils pour cette architecture qui la rende compatible avec tous les tâches de la liste.

Question 7. Montrer que le problème est dans la classe NP.

Question 8. On considère le réseau à seuil à 4 entrées (à savoir x_1, x_2, x_3, e) et 1 sortie (à savoir y) qui se représente graphiquement de la façon suivante :



2. Si cela aide, formellement : l'architecture d'un réseau est donné par un graphe $G = (V, E)$, où les sommets V se partitionnent en $V_0, V_1, \dots, V_d$, et avec chaque arête (u, v) de la forme $u \in V_k, v \in V_{k+1}$ pour un certain entier k . Les n sommets de V_0 sont les entrées. Les m sommets de V_d sont les sorties. On associe à chaque sommet n de V_k , pour $1 \leq k < d$, un neurone, avec autant d'entrées que son degré entrant. Un tel réseau calcule une fonction de $\mathbf{B}^n$ dans $\mathbf{B}^m$ en propageant les valeurs des entrées vers les sorties : si on note F_n pour la fonction associée au neurone/sommet n , sa valeur est l'image par F_n des valeurs d'activation de ses entrées, tout cela calculé récursivement des entrées vers les sorties. d est appelé la profondeur du réseau de neurones.

On considère les 8 tâches que l'on obtient en faisant varier $x_1, x_2, x_3 \in \mathbf{B}$, et en ayant fixé $e = 0$, en prenant y qui vaut 1 sauf dans l'unique cas $x_1 = 1, x_2 = 0, x_3 = 1$ où y vaut 0.

Prouver que cette architecture est compatible avec ces 8 tâches.

Question 9. On suppose qu'on a fixé les seuils et poids de l'architecture de la question précédente de manière à obtenir un réseau de neurones qui, sur les 8 tâches de cette dernière question, produit les sorties attendues. Notons $f_i(x)$ la valeur d'activation du nœud v_i sur l'entrée $(x, 0)$ (c'est-à-dire, celle où $x_i = x$ et $e = 0$). Montrer que f_i est soit la fonction identité, soit la fonction $x \mapsto \text{NOT}(x)$.

Notons $\overline{x_i}$ pour $f_i(x_i)$.

Question 10. On suppose toujours donné un réseau de neurones comme à la question précédente.

Montrer que la valeur d'activation de c , notée y , s'exprime sous la forme $y = \text{NOT}(\overline{x_1}) \vee \overline{x_2} \vee \text{NOT}(\overline{x_3})$.

Lorsque x est une variable, on note $x[0]$ pour x , et $x[1]$ pour sa négation. Si on pose $d_i = 0$ lorsque f_i est la fonction identité, et $d_i = 1$ lorsque f_i est la fonction $\text{NOT}(x)$, on a donc $\overline{x_i} = x_i[d_i]$.

Question 11. On se donne une formule du calcul propositionnel $\phi(x_1, \dots, x_n)$ avec les variables $x_1, \dots, x_n$. Montrer que les deux assertions suivantes sont équivalentes :

- la formule $\phi(x_1, \dots, x_n)$ est satisfiable ;
- on peut choisir $d_1, d_2, \dots, d_n \in \{0, 1\}$, tels que si l'on regarde la formule $\phi'(x_1, \dots, x_n, e)$ avec $n+1$ variables, donnée³ par $\phi'(x_1, \dots, x_n, e) = \phi(x_1[d_1], \dots, x_n[d_n])$ pour tout $x_1, \dots, x_n, e \in \mathbf{B}$, on a que

$$\phi'(0, \dots, 0, 1)$$

est satisfaite.

Question 12. Montrer que le problème APPRENTISSAGE, même limité aux réseaux de neurones de profondeur⁴ 2, est NP-complet. Même si dans les exemples précédents, les réseaux de neurones n'avaient qu'une sortie, rappelons que les réseaux de neurones utilisés peuvent avoir plusieurs sorties.

3 Sous-shifts et motifs interdits

Une configuration est une fonction de $\mathbb{Z}^2$ dans $\{0, 1\}$: une configuration c colore chaque point du plan soit par 0 soit par 1. Un sous-ensemble fini $\mathbb{U} \subset \mathbb{Z}^2$ est appelé une fenêtre. Un motif est une fonction de $\mathbb{U}$ dans $\{0, 1\}$, où $\mathbb{U}$ est une fenêtre, que l'on appelle le support du motif. Etant donnée une configuration c , on écrit $c_{\mathbb{U}}$ pour le motif que l'on voit dans la fenêtre $\mathbb{U}$: formellement, c'est la restriction de la fonction c à $\mathbb{U}$. On dit que ce motif est dans la configuration.

Etant donné $(d_1, d_2) \in \mathbb{Z}^2$, la fonction décalage $\sigma^{(d_1, d_2)}$ est la fonction qui envoie la configuration c sur la configuration c' avec $c'(i, j) = c(i - d_1, j - d_2)$. Elle envoie un motif m sur un motif m' avec $m'(i, j) = m(i - d_1, j - d_2)$: son support $\mathbb{U}'$ est donné par l'ensemble des (i, j) tels que $(i - d_1, j - d_2) \in \mathbb{U}$.

On se fixe un ensemble de motifs $\mathcal{F}$. On s'autorise à ce que cet ensemble soit fini ou infini.

On note $T(\mathcal{F})$ pour l'ensemble des configurations où l'on s'interdit les motifs de $\mathcal{F}$: c'est l'ensemble des configurations c privé de tous les décalages des motifs de $\mathcal{F}$.

Question 13. Montrer qu'il existe une théorie $\mathcal{T}$ du calcul propositionnel qui possède un modèle si et seulement si $T(\mathcal{F})$ est non-vide.

3. Nous confirmons que la valeur de ϕ' ne dépend pas de son argument e .

4. Voir la note en bas de page de la page précédente pour une définition formelle de la profondeur. Par exemple, le réseau de neurones de la figure de la question 8 est de profondeur 2.

On dit qu'une suite $(c^i)_{i \in \mathbb{N}}$ de configurations converge vers la configuration c si pour tout $(n_1, n_2) \in \mathbb{Z}^2$ il y a un rang $k \in \mathbb{N}$ tel que pour tout $i \geq k$ on a $c^i(n_1, n_2) = c(n_1, n_2)$: autrement dit, une suite converge si à chaque position, à partir d'un moment, un seul élément de $\{0, 1\}$ apparaît dans cette suite à cette position. On dit qu'un ensemble de configurations C est *fermé* si pour toute suite à valeur dans C qui converge vers une certaine limite, est telle que cette limite est dans C .

Un ensemble de configurations C est stable par décalage, si pour tout $c \in C$, pour toute valeur possible de (d_1, d_2) , on a que $\sigma^{(d_1, d_2)}(c) \in C$.

On appelle *sous-shift* un ensemble de configurations qui est fermé et qui est stable par décalage.

Question 14. Prouver que $T(\mathcal{F})$ est un sous-shift.

C'est même une caractérisation des sous-shifts :

Question 15. Montrer que pour tout sous-shift T , on peut trouver un ensemble de motifs $\mathcal{F}$ tel que $T = T(\mathcal{F})$.

Question 16. On se fixe un ensemble de motifs $\mathcal{F}$.

Montrer que $T(\mathcal{F})$ est non-vide si et seulement si pour tout entier n , on peut trouver un motif de support $\mathbb{U}_n = [-n, n]^2$ qui ne contient aucun motif de $\mathcal{F}$.

4 Systèmes de Post

Un système⁵ de Post est défini par un alphabet fini Σ , et un nombre fini de règles de réécriture de la forme $g \rightarrow d$, où g, d sont des mots de Σ^* .

Sur un mot $w \in \Sigma^*$ sur l'alphabet Σ , on applique l'instruction suivante :

1. pour chaque règle $g \rightarrow d$, si le mot commence par le mot g , on efface ce préfixe, et on ajoute le mot d à sa fin : autrement dit, si le mot est de la forme $w = gw'$, il devient $w'd$.

On fait cela tant que c'est possible. Si ce processus mène à un mot auquel plus aucune règle ne s'applique, on dit que le mot initial w est accepté. Notons qu'il est possible que sur un mot donné il y ait plusieurs possibilités d'évolution. Il est aussi possible de considérer les règles $g \rightarrow d$ dans n'importe quel ordre.

Question 17. Montrer que l'ensemble des mots acceptés par un système de Post est récursivement énumérable.

On dira qu'on transforme un mot w en un mot w' s'il y a un nombre fini d'étapes tel qu'en partant du mot w on fini par atteindre le mot w' .

Question 18. On considère un mot de la forme $I_j 1^{n_1} K_j 1^{n_2}$: c'est-à-dire avec le symbole I_j , puis $n_1 \geq 0$ fois le symbole 1, puis le symbole K_j , puis $n_2 \geq 0$ fois le symbole 1.

Proposer des règles qui transforment ce mot en un mot similaire, mais où on a ajouté 1 à n_1 , remplacé I_j par I_{j+1} , et remplacé K_j par K_{j+1} .

Question 19. Que fait le système de Post avec les règles $1 \rightarrow 1$, $I_j K_j \rightarrow I_{j'} K_{j'}$, $I_j 1 \rightarrow I_{j+1}$, $K_j \rightarrow K_{j+1}$ sur un mot de la forme $I_j 1^{n_1} K_j 1^{n_2}$?

Question 20. Démontrer qu'il est indécidable de déterminer si un système de Post accepte un mot donné.

5. C'est en fait ce que l'on appelle un système de post en forme normale. On ne considère que des systèmes de Post en forme normale dans ce sujet.