

Deep Learning

Master Recherche AIC LRI

14:00-16:00

Feb. 7th, 2017

Alexandre Allauzen, Michèle Sebag, Thomas Schmitt

Documents are allowed. Computers are not allowed.

Read the exam document before you start. Any answer must be justified. Write the answers to Part I and Part II on different sheets.

Il est conseillé de lire tout le texte avant de commencer. Les réponses doivent être justifiées. Rédigez les parties I et II sur des copies différentes.

1 Partie I. Questions de cours (10 points)

Q. 1.1 *List several activation functions, graphically indicate how they vary and discuss their strengths and weaknesses in the context of neural nets.*

Citez plusieurs fonctions d'activation, dessinez leur courbe $y=f(x)$ et discutez leurs forces et leurs faiblesses dans le cadre de l'apprentissage des réseaux neuronaux.

Q. 1.2 *Define a deep neural net. When should one use a deep NN: never, always, it depends (on what) ?*
Définissez ce qu'est un réseau neuronal profond. Quand doit-on de préférence apprendre un réseau neuronal : jamais, toujours, ça dépend (de quoi) ?

Q. 1.3 *Briefly explain what is the bias and variance of a hypothesis space. When the number of NN layers increases, does the bias increase ? decrease ? It depends (on what ?). Same questions for the variance.* Rappelez brièvement les notions de biais et de variance relatives à un espace d'hypothèses. Quand le nombre de couches d'un réseau neuronal augmente, le biais : augmente ? décroît ? ça dépend (de quoi ?). Mêmes questions pour la variance.

Q. 1.4 *Should one pre-process the dataset before training a neural net ? Why ? How ? What is the impact on the NN initialization ?*

Doit-on pré-traiter les données avant l'entraînement d'un réseau profond ? Pourquoi ? Comment ? Quel est l'impact sur l'initialisation des poids du réseau ?

Q. 1.5 *Let*

$$\mathcal{E} = \{(x_i, y_i), x_i \in \mathbb{R}, y_i \in \mathbb{R}, i = 1 \dots N\}$$

be a training set. Is it possible without training to have a neural net h such that $h(x_i) = y_i$ for $i = 1 \dots n$? (intuition, consider radius-based activation functions ($g(x) = e^{-x^2}$)).

A partir d'un ensemble d'entraînement comme ci-dessus, comment construire sans entraînement un réseau neuronal tel que $h(x_i) = y_i$ pour $i = 1 \dots n$? (Intuition, considérez des fonctions d'activation radius-based, $g(x) = e^{-x^2}$).

Q. 1.6 *How to prevent overfitting when learning a neural net ? a deep neural net ? How is this question related to the previous question ?*

Comment prévenir le surapprentissage quand on apprend un réseau neuronal ? un réseau neuronal profond ? Quelle est la relation entre cette question et celle qui précède ?

Q. 1.7 *A limitation of (deep) neural nets is to build non-understandable models. Why is this a limitation ? How to explain the output of a NN ? (Indication, think of explaining a k-nearest-neighbor classifier).*

Une limitation des réseaux neuronaux (profonds) est de ne pas être intelligible. Pourquoi est-ce une limitation ? Comment “expliquer” la sortie d’un NN ? (Indication: comment expliquer la sortie d’un k-plus proche voisin ?)

Q. 1.8 *In case of missing values in your dataset, which heuristics would you use to train a NN ?*

Si votre dataset comprend des données manquantes, comment vous y prendriez-vous pour entraîner un NN ?

Q. 1.9 *Several papers based on deep NNs have been presented in the last course. Cite one (or more); discuss a result that you found surprising or that you did not understand (explain what you did not understand).*

Plusieurs articles portant sur les réseaux profonds ont été présentés lors du dernier cours. Citez-en un (ou plusieurs); discutez un résultat que vous avez trouvé étonnant, ou un résultat que vous n’avez pas compris. Expliquez ce que vous n’avez pas compris.

2 Algorithme d’apprentissage (10 points)

Considérons un réseau de neurones feed-forward avec une couche cachée avec les notations suivantes: le vecteur d’entrée est $\mathbf{x}^{(1)}$; la couche cachée est représentée par le vecteur $\mathbf{y}^{(1)} = f^{(1)}(\mathbf{W}^{(1)}\mathbf{x}^{(1)})$; la sortie est $\mathbf{y}^{(2)} = f^{(2)}(\mathbf{W}^{(2)}\mathbf{x}^{(2)})$, avec $\mathbf{x}^{(2)} = \mathbf{y}^{(1)}$. La fonction d’activation ($f^{(1)}$) de la couche cachée est la tangente hyperbolique. La tâche considérée est de la classification binaire, donc la couche de sortie n’a qu’un seul neurone avec une fonction d’activation logistique:

$$f^{(2)}(a^{(2)} = \mathbf{W}^{(2)}\mathbf{x}^{(2)}) = \sigma(a) = \frac{1}{1 + e^{-a}}$$

La sortie peut directement s’interpréter comme la probabilité que $\mathbf{x}^{(1)}$ appartienne à la classe $c = 1$. La fonction objectif est le maximum de vraisemblance, soit pour un exemple:

$$l(\boldsymbol{\theta}, \mathbf{x}, c) = -(c \log(\mathbf{y}^{(2)}) + (1 - c) \log(1 - \mathbf{y}^{(2)})),$$

où \mathbf{x} est l’exemple d’apprentissage, c la classe de référence à prédire ($c = 0$ or 1) et $\boldsymbol{\theta}$ regroupe tous les paramètres du modèle. Notons que la couche de sortie n’a qu’un neurone et donc que $\mathbf{W}^{(2)}$ est une matrice ligne.

1. écrire la formule de mise à jour des paramètres de la couche de sortie, $w_j^{(2)}$ qui correspond à l’élément j de $\mathbf{W}^{(2)}$. Pour cela, vous pouvez si vous le souhaitez suivre les étapes suivantes:
 - Exprimer la valeur du neurone de sortie $y^{(2)}$ en fonction de $\mathbf{x}^{(2)}$ et $\mathbf{W}^{(2)}$.
 - Partant de la fonction objectif calculer sa dérivée par rapport à $w_j^{(2)}$.
 - Donner et interpréter la formule de mise à jour.
2. Faire de même avec la couche cachée $w_{kj}^{(1)}$. Vous pouvez pour cela adapter la démarche précédente aux spécificités de la couche cachée².
3. Décrire de manière plus globale l’algorithme d’apprentissage de ce réseau (définir la fonction objectif sur l’ensemble des données d’apprentissage et comment l’optimiser, ...).

Pour ces questions la clarté de la rédaction joue un rôle important.

¹La dérivée de la fonction logistique σ est $\sigma'(a) = \sigma(a)(1 - \sigma(a))$.

²La dérivée de la fonction tangente hyperbolique \tanh est $\tanh'(a) = 1 - \tanh^2(a)$