

IFT 603-712 : Devoir 4

Travail par équipe de 2 ou 3

Remettez votre solution aux numéros 1 et 2 en format pdf ou manuscript (et scanné) via **turninweb**. Même chose pour le code.

1- [1.5 points] Nous avons vu dans le cours que l'opération Softmax est représentée par l'équation mathématique suivante :

$$y_{\vec{w}_i}(\vec{x}) = \frac{e^{a_i}}{\sum_c e^{a_i}}$$

où a_i est la sortie du i -ème neurone de sortie et que la fonction de perte entropie croisée (*cross-entropy*) est

$$E_D(W) = - \sum_{n=1}^N \sum_{k=1}^K t_{kn} \ln y_{\vec{w}_i}(\vec{x})$$

où t_{kn} est la cible du n -ième élément de la base de données d'entraînement. Donner l'équation du gradient de la fonction de perte par rapport à a_i ainsi que toutes les étapes mathématiques pour arriver à ce résultat.

2- [1.5 points] Soit $k_1(\mathbf{x}, \mathbf{x}')$ un noyau valide, prouvez la propriété 6.16 du livre de Bishop stipulant que $\exp(k_1(\mathbf{x}, \mathbf{x}'))$ est également un noyau valide.

3- [1 point] Démontrer le lien qu'il y a entre la descente de gradient de type *momentum*, et les formules pour calculer la position, la vitesse et l'accélération d'un objet en mouvement.

4- [6 points] Programmez des réseaux de neurones à 1 et 2 couches afin d'avoir un classifieur linéaire et un classifieur non linéaire telle qu'illustré au chapitre 7 du cours. Pour ce faire, vous devez implanter une entropie croisée (*cross-entropy*) avec une couche *Softmax* à la fin du réseau. Le classifieur non linéaire exige l'implantation d'un réseau avec une couche d'entrée, une couche cachée et une couche de sortie. Le code est contenu dans le fichier **devoir4.zip** via le site web du cours.

Les algorithmes doivent être implémentés à l'intérieur des fichiers pythons **linear_classifier.py** et **two_layer_classifier.py** qui contiennent déjà une ébauche et des mentions **TODO** aux endroits où vous devez ajouter du code. L'exécution des fonctions et des classes associées à ces fichiers se fait via le notebook **Devoir4.ipynb**. Comme vous le verrez, ce notebook contient bon nombre de « sanity checks » mentionnés tout au long du cours.

Note 1 : bien que vide, le code du notebook fonctionne déjà. Pour vous en convaincre, vous n'avez qu'à taper la commande suivante dans un terminal :

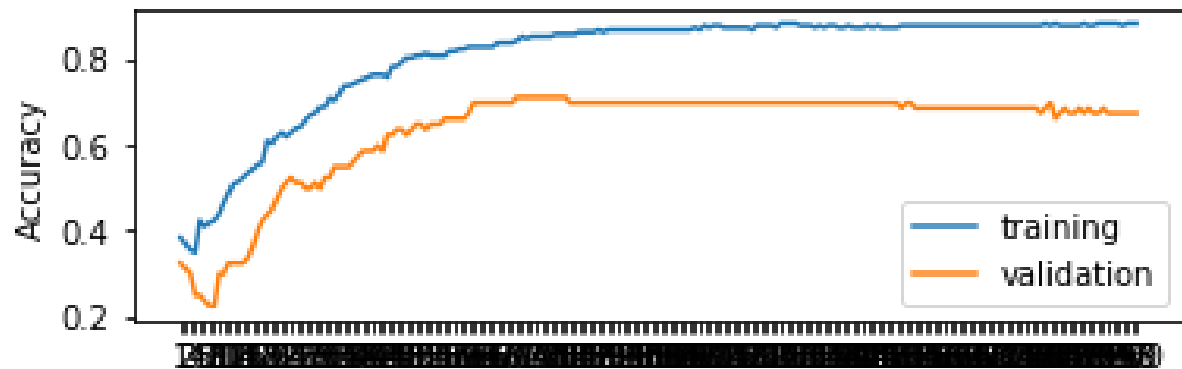
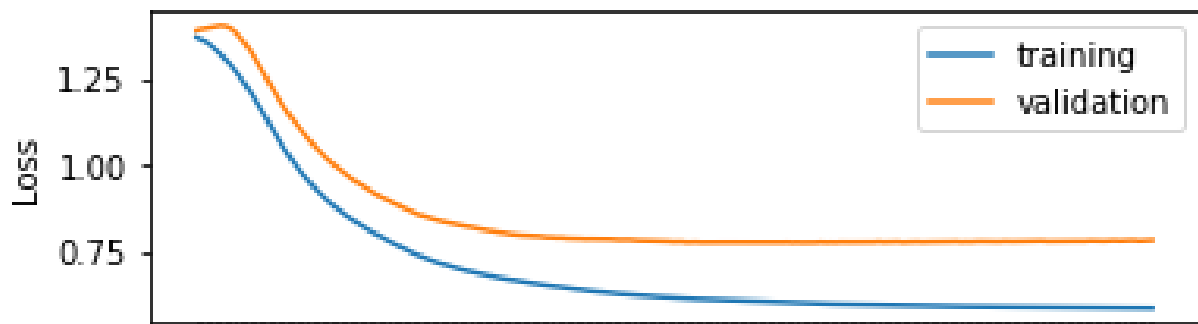
```
jupyter notebook
```

Et de sélectionner le fichier **Devoir4.ipynb**.

Note 2 : le code des devoirs (ainsi que des notebooks) a été testé avec python 3.5 sous Linux.

Note 3 : il est recommandé de rédiger son code dans un ide tel spyder ou pycharm.

Note 4 : voici de quoi devrait avoir l'air votre solution pour le modèle non linéaire appliqué aux données « Ncircles » 4 classes :



Frontières de décision

