

Statistique en Grande Dimension et Apprentissage - TP "Réseaux de Neurones"

Il existe principalement deux architectures pour l'apprentissage de modèles de deep learning : Tensor Flow développé par Google Brain et Pytorch, développé par Meta. Nous utiliserons ici Tensor Flow accompagné de l'API Keras. On commencera donc par importer Tensor Flow et les modules de Keras utilisés (**Sequential**, **Dense**, **Dropout**...). On trouvera de nombreux exemples à ce lien.

- Exercice 1** (Perceptron à une couche cachée). 1. Charger les données spam sur Moodle.
2. Fabriquer un échantillon d'entraînement et un échantillon test.
 3. Construire un modèle de type avec une couche cachée avec la fonction d'activation **Relu** et une sortie avec la fonction d'activation ψ la fonction sigmoïde.
 4. Que signifient les différents paramètres du modèle ? Affichez les caractéristiques de ce modèle à l'aide de **summary**.
 5. Entraînez le modèle sur l'échantillon d'entraînement (cette étape est divisée en deux parties : **compile** et **fit**, voir cette page et comprendre les paramètres en particulier epochs, batch_size, optimizer).
 6. Calculez l'erreur test.

- Exercice 2** (Perceptron Multicouches). 1. Charger les données Mnist_test sur Moodle (uniquement la base test pour limiter le temps de calcul).
2. Rediviser cet échantillon en un échantillon d'entraînement et un échantillon test.
 3. Construire un perceptron à une couche cachée en utilisant en fonction d'activation ψ la fonction "softmax". Paramétrer le "dropout" (paramètre d'élagage).
 4. Que signifient les différents paramètres du modèle (layers, units,...) ? Affichez les caractéristiques de ce modèle.
 5. Entraînez le modèle sur l'échantillon d'entraînement (comprendre les paramètres epochs, batch_size).
 6. Calculez l'erreur test.
 7. Construire un modèle à deux couches cachées et refaites la même procédure.
 8. Testez l'erreur de prédiction sur les données spam avec un réseau à une couche cachée en réglant les hyperparamètres.
 9. Comment pouvez-vous sauvegarder votre modèle ? (voir sur ce sujet cette page)

Exercice 3 (Réseau de neurones convolutif/Réseaux de neurones récurrents). .

1. Utiliser ce tutoriel afin de tester les modèles de réseaux de neurones convolutifs sur la base MNIST.
2. Tester les architectures LSTM pour les réseaux de neurones récurrents.
3. Tester les architectures convolutionnelles pour l'analyse du signal (exemple)

I