M2 MIAGE IDA – Deep learning
TD3: Réseaux de neurones récurrents

Objectifs :

- Implémenter des réseaux de neurones récurrents
- Prédire la suite de séries temporelles
- Utiliser les RNN dans le cadre du traitement automatique du langage

Exercice 1 : Séries temporelles

L'objectif de cet exercice est de prédire les cotations d'une action en utilisant des réseaux récurrents.

1. Télécharger un dataset contenant l'historique des cotations d'une action de votre choix (votre entreprise en alternance, facebook, google, CAC40, etc.). Vous trouverez ce qu'il faut ici : https://www.nasdaq.com. Il est conseillé de prendre la plus grande période possible pour avoir le dataset le plus grand possible.
2. Charger le fichier CSV en python et explorer le dataset. Proposer une ou plusieurs représentations graphiques permettant de visualiser les données.
3. Sélectionner la ou les colonnes permettant la prédiction les cotations de l'action sélectionnée.
4. Préparer un dataset d'entrainement et de test pour l'apprentissage. En particulier, le dataset doit être constitué de séquences de valeurs (en entrée) permettant de prédire la (ou les) valeur(s) suivante(s) de cette séquence (target).
5. Proposer et comparer quelques architectures neuronales à partir de couches LSTM et GRU. Sélectionner celle permettant d'obtenir les meilleurs résultats. Dans cette question, la prédiction doit être faite à très court terme. Vous pouvez aussi utiliser des couches convolutionnel 1D pour évaluer l'impact de la mémoire long terme.
6. Explorer la capacité de prédiction de ce système sur des plus long terme (prédiction à plusieurs jours).

Exercice 2 : Génération de texte car2car

L'objectif de cet exercice est de créer un programme permettant de générer automatiquement la fin d'une phrase. Pour cela, nous allons l'entrainer à prédire caractère par caractère (car2car) un livre fourni. Dans cet exercice, nous utiliserons une architecture basée sur les LSTM. Le réseau sera entrainé sur un livre, Alice au pays de merveille, que vous pouvez télécharger sur l'espace de cours.

1. Télécharger le fichier alice.txt depuis l'espace de cours. Charger ce fichier en python et afficher son contenu.
2. Depuis la chaîne de caractères précédemment chargée, générer la liste de caractères uniques présents dans le livre. Écrire une méthode permettant de transformer chaque caratère en entier et inversement (chaque entier en caractère). Ceci est nécessaire car les réseaux de neurones ne travaillent pas sur des caractères mais sur des nombres !

3. Préparer un dataset contenant des séquences de 50 caractères (paramétrable) de longueur en entrée et le caractère suivant cette séquence en target.
4. Transformer le dataset en séquence « one hot encodées »
5. Proposer et tester différentes architecture LSTM entrainées sur le dataset précédemment généré.
6. Tester la meilleure architecture sur une chaine de caractères qu'il n'a jamais vu, soit du dataset de test, soit que vous proposerez vous. Attention à bien utiliser votre convertisseur caractère/entier pour pouvoir chiffrer et déchiffrer les informations du réseau.

Exercice 3 : Traduction de texte automatique en seq2seq

L'objectif de cet exercice est de développer un traducteur automatique anglais-français. Pour cela, nous utiliserons des couches LSTM organiser pour créer un réseau de type seq2seq. Ainsi, le réseau devra, à partir d'une entrée sous la forme une séquence de caractères en anglais, prédire une séquence de caractères en français. Le réseau sera entrainé sur un dictionnaire anglais-français.

1. Télécharger, charger en python et afficher le fichier fra.txt
2. Créer un dataset contenant l'ensemble des séquences en anglais comme entrées et l'ensemble des séquences en français en cible. Attention, pour que le réseau sache quand commencer et finir à prédire, les séquences de sorties doivent commencer par une tabulation ('\t') et se terminer par un retour chariot ('\n').
3. Trouver les caractères et leurs nombres contenus dans les séquences d'entrées et de sorties.
4. Comme dans l'exercice 2, créer des fonctions permettant de transformer les entrées et les sorties en chaines d'entiers.
5. Transformer le dataset de chaînes de caractères en un dataset « one hot encoded ».
6. Construire un réseau de neurones ayant l'architecture suivante :
    a. Une couche LSTM, l'encodeur, ayant une couche d'entrées de la taille de la plus grande séquence des chaines en anglais. Cette couche devra avoir 256 états internes.
    b. Une couche LSTM, le décodeur, ayant en entrées les états internes (h et c) de l'encodeur. L'état interne initial de cette couche doit être l'état interne de l'encodeur.
    c. Une couche dense permettant de choisir le caractère (one hot encodé).

8/ Compiler et entrainer le modèle sur le dataset précédemment produit.

9/ Une fois l'entrainement réalisé :

a. Créer un nouveau modèle d'encodeur contenant la couche d'entrées et les états internes de l'encodeur.
b. Créer un nouveau modèle de décodeur contenant la couche LSTM et la couche dense du réseau précédemment entrainé.
c. Créer une fonction de traduction qui :
    - Passe la séquence one hot encodé à l'encodeur et récupère les états internes.
    - Génère une séquence contenant une tabulation '\t' (symbole de démarrage de la prédiction).
    - Prédit les caractères de la séquence jusqu'à l'obtention d'un retour chariot '\n' (symbole de fin de traduction).
    - Affiche la traduction de façon lisible.