Projet "Du Machine Learning au Deep Learning (RNN & variantes avancées / Autoencoders)"

Master 2 : Data Science, UIE

8 février 2025

Informations & Consignes sur l'évaluation

- **Type d'évaluation :** Projet
- **Date de rendu :** 10 Mars 2025
- **Année universitaire :** 2024 - 2025
- **Consignes :** Projet à faire en binôme dans un Notebook, **sans plagiat** entre binômes ou ailleurs, ni sur internet non plus (citez vos sources) et sans copier-coller de ChatGPT ou un autre ChatLLM (ce sont des outils utiles, sachez les utiliser ... sans oublier de les citer également en cas d'utilisation). **Mentionnez les noms des binômes dans le nom du fichier Notebook et dans le mail d'envoi.**
- **Enseignant :** Dr DIABATÉ Modibo (modibo.diabate.pro@gmail.com)

Partie I : Prédiction multivariée de données météo (Delhi)

Dans cette partie, on s'intéresse aux données climatiques quotidiennes (`meantemp` : température moyenne (`°C`), `humidity` : humidité moyenne de l'air (`%`), `wind_speed` : vitesse moyenne du vent (`km/h`) et `meanpressure` : pression atmosphérique moyenne (`hPa`)) recueillies du 1er janvier 2013 au 1er janvier 2017 à New Delhi et contenues dans le fichier `DailyDelhiClimateTrain.csv`. Notre objectif ici est de prédire, sur les jours suivants, plusieurs variables climatiques en même temps (prédiction multivariée) impliquant une variable `étiquette` (`target`) multidimensionnelle.

1. Séparez les données climatiques quotidiennes de New Delhi (contenant les 4 colonnes `meantemp`, `humidity`, `wind_speed` et `meanpressure`) en données d'entrainement (environ 70%) et données de test (environ 30%).
2. Créez une fonction `create_Xseq_and_y` pour qu'elle donne en sortie des séquences d'entrée (caractéristiques "X") et les sorties à prédire correspondantes (étiquettes y) permettant d'utiliser les données climatiques journalières (`meantemp`, `humidity`, `wind_speed`, `meanpressure`) des 20 jours précédents pour prédire celles du 21ème jour. Appliquez la fonction à vos données d'entrainement et de test.
3. Choisissez, en justifiant votre choix, un modèle de type RNN parmi le RNN classique, le LSTM, le GRU et le BiLSTM et mettez en place, à l'aide de la bibliothèque Python Ten-

sorFlow, une architecture de réseau de neurones pour ce modèle (architecture à améliorer éventuellement).

4. Entrainez le modèle mis en place à l'aide des données d'entrainement précédemment préparées et évaluez la qualité de l'entrainement.
5. Prédisez les données de test avec le modèle entrainez
6. Calculez l'erreur de prédiction et commentez
7. Proposez un graphique pour chaque variable climatique (`meantemp`, `humidity`, `wind_speed` et `meanpressure`) permettant de représenter en même temps et dans différentes couleurs : ses vraies données d'entrainement, ses vraies données de test et les prédictions du modèle RNN. Commentez !
8. Ré-entrainez votre modèle sur l'ensemble des données climatiques de New Delhi (du 1er janvier 2013 au 1er janvier 2017 avec l'ensemble des 4 variables climatiques) et prédisez les variables climatiques du 02 janvier 2017 au 12 janvier 2017. Représentez graphiquement ces résultats de prédictions et commentez.

Partie II : Génération de texte avec des modèles "RNN"

Dans cette partie, l'objectif est d'entrainer et de valider un modèle de type RNN pour la génération de texte tout en identifiant d'éventuelles limites (étudiées théoriquement pendant le cours ou non).

1. Identifiez un texte original, idéalement en lien avec le Mali et/ou l'Afrique, qui servira de données d'entrainement de votre modèle de génération de texte et faites les pré-traitements nécessaires sur ce texte afin qu'il puisse être utilisé comme données d'entrainement d'un modèle de langue de type RNN.
2. Mettez en place, à l'aide de la bibliothèque Python PyTorch, un modèle LSTM qui sera entrainé sur les données précédemment préparées. Vous avez le choix dans l'élaboration de l'architecture du modèle (nombre de couches / neurones, méthode optimisation, métriques d'évaluation, ...).
3. Faites de même pour un modèle BiLSTM, toujours avec PyTorch.
4. Entrainez vos modèles en ayant la liberté d'inclure ou non les bonnes options d'optimisation et d'évaluation de la qualité de vos entrainements et des modèles qui en découleront.
5. Inférence : utilisez vos modèles entrainés pour générer la suite de trois différentes phrases (pour chaque modèle) que vous initiez avec seulement deux ou trois mots.
6. Sur la base de ces résultats d'entrainement et d'inférence, lequel de vos deux modèles entrainés présente les meilleures performances ? Justifiez.
7. Quelles sont les limites, selon vous, de la solution mise en place en terme de : qualité des données (au sens large) ; modèle (LSTM/GRU vs BiLSTM/BiGRU vs Alternatives) ; architecture choisie (choix du nombre / position des couches, nombres de neurones, fonctions d'activation, ...) ; ...
8. Connaissez vous une solution alternative à la majeure partie de ces limitations ? Si oui, décrivez là en quelques mots.

Partie III : Synthèse de texte à l'aide d'un auto-encoder "RNN"

1. Entrainez, sur une partie des données d'entrainement de la Partie II, un auto-encoder LSTM ou BiLSTM (au choix) pour le résumer de texte et testez le sur l'autre partie des données de la Partie II gardées comme données de test. Commentez.