

TP - PyTorch & Deep Learning

TP1 - Régression linéaire en Pytorch

Objectifs du TP

L'objectif de ce TP est de mettre en pratique les bases de Pytorch

Vous implémenterez ici une régression linéaire sur un jeu de données: le Boston Housing dataset, qui regroupe les prix de logements à Boston.

Comme indiqué dans le premier TD, vous rendrez un script python, pas de notebook hormis pour une éventuelle exploration des données (ou très bonne raison).

Contrairement à la première année, vous serez ici en autonomie: pas de scripts et réponses pré-remplies, vous partez d'une page vierge et d'un fichier .csv.

Les éventuelles questions qui seront posées sont principalement pour vous faire réfléchir, je n'attends pas de réponse écrite, sauf si je le précise. N'hésitez pas à vous servir de manière libérale de la documentation Pytorch et du polycopié de ce matin, les deux vous seront très utiles.

N'oubliez pas de commenter de manière libérale votre code: j'accorde autant d'importance au fait que le code marche qu'à la réflexion que vous avez face aux divers problèmes que vous rencontrez.

Chargement et Exploration des Données

1. Chargez le jeu de données depuis un fichier CSV nommé `house_price_regression_dataset.csv`.
2. Nettoyez le jeu de données (si nécessaire).
4. Affichez les premières lignes du jeu de données pour confirmer qu'il a été correctement chargé.

Préparation des données

1. Séparez 'Square_Footage' en variable de prédiction et 'House_Price' en variable à prédire.
2. Convertissez ces variables en tenseurs pytorch.
3. Divisez le jeu de données en ensembles d'entraînement et de test, avec 80% des données pour l'entraînement et 20% pour le test.

Mise en Place des Dataloaders

1. Créez des DataLoaders pour les ensembles d'entraînement et de test en utilisant une taille de lot de 32.

Définir un Modèle de Régression Linéaire Simple

1. Définissez un modèle de régression linéaire prenant en entrée la variable de prédiction 'Square_Footage'. Vous utiliserez la fonction `torch.nn.Parameter()` pour définir le(s) paramètre(s) à entraîner. De combien de paramètres avez vous besoin ?
Indice: La formule d'une régression linéaire est $Y = a * x + b$
2. Imprimez les paramètres initiaux du modèle, y compris le poids et le biais.



Implémenter la Boucle d'Entraînement

1. Spécifiez la fonction de perte et l'optimiseur pour le modèle.
2. Implémentez une boucle d'entraînement qui fonctionne pendant 200 époques.
3. Enregistrez et tracez les pertes d'entraînement et de test au fil des époques pour visualiser la performance du modèle.
4. Imprimez les coefficients du modèle à la fin de l'entraînement.



Évaluer le Modèle

1. Prédisez les prix des maisons pour l'ensemble de test et comparez ces prédictions avec les valeurs réelles.
2. Créez un nuage de points pour visualiser la relation entre les prix des maisons prédits et réels. Rajoutez une ligne idéale d'égalité (où $x = y$).



Régression Multilinéaire : Utilisation de Caractéristiques Multiples

1. Étendez le jeu de données pour inclure d'autres caractéristiques afin de prédire les prix des maisons.
2. Mettez à jour le modèle de régression linéaire pour gérer ces caractéristiques multiples.
3. Réentraîner le modèle et visualisez les résultats comme précédemment.



Implémenter une Régularisation Lasso

1. Appliquez une régularisation Lasso au modèle en utilisant le paramètre de décroissance (souvent nommé λ) que vous ajouterez à la perte calculée à chaque batch.
2. Affichez la contribution à la perte de la pénalité lasso en pourcentage, et son évolution au cours du temps.