

Projet Annuel : Machine learning

Présentation de la problématique :

Le but de ce projet est de créer une intelligence artificielle capable de classer une page selon son type de d'ouvrage. Les classes nous avons choisi sont manga, comics et roman.

Nous avons entraîné notre modèle pour classer une page de romans, comics ou manga.

Nous chargeons une image comme entrée

Nous sommes passé par plusieurs étapes pour réaliser le projet.

Dans un premier temps il nous a fallu constituer un dataset sur lequel notre modèle allait s'entraîner.

Dans un deuxième temps nous avons travaillé sur le pré-traitement de l'image. Ensuite nous avons créé un modèle adapté à notre cas : une classification à plus de 2 classes. Enfin nous avons enregistré notre modèle et nous l'avons testé.

I- Constitution du dataset

On a choisi de partir sur environ 2000 occurrences pour chaque classe. On a récupéré les mangas, les comics, et les romans sous format PDF pour ensuite les convertir en format JPEG. Il n'y pas eu une automatisation de la récupération des images avec un script. Nous avons choisi les ouvrages et avons fait le dataset à la main.

II- Pré-traitement de l'image

Il est très facile pour l'humain de faire la différence entre un roman et un comics. Notre cerveau le fait de manière innée, on voit qu'il n'y a pas de dessin sur une page de roman et qu'il y en a sur une page de comics.

Les machines ne peuvent pas traiter directement une image mais elles peuvent traiter des tableaux de valeurs de matrice etc..

C'est pour cela que nous avons mis en place un pré-traitement de l'image.

1ere phase : on a chargé grâce à la librairie Pillow

2eme phase : nous avons transformé l'image en niveau de gris et l'avons converti en Numpy Array

Nous avons décidé de transformer notre image en nuance de gris pour avoir des numpy array de dimensions uniformes.

Une image en couleurs converti en np array n'est de même dimensions qu'un np array d'une image en noir et blanc.

Une fois le pré-traitement de l'image fait nous pouvons traiter les données avec un modèle.

III- Création et entraînement du modèle

Nous sommes partie sur un modèle MLP « MultiLayer Perceptron », Perceptron MultiCouche en français. Nous avons choisi l'algorithme d'apprentissage : la descente de gradient stochastique pour entraîner notre modèle.

Après la constitution de notre dataset un peu plus tôt nous avons séparé le dataset en 2 parties. La 1ère partie représentant 2/3 du dataset d'origine a servi à l'entraînement du modèle. L'autre partie a servi à tester le modèle précédemment entraîné.

Voici les résultats :

- Training set accuracy : 0,991970
- Test set accuracy : 0,985979
- Loss : 0,03188009
- Nombre d'époques : 305

IV- Test du modèle

À la fin de l'entraînement nous voulions tester le modèle, nous voulions pouvoir charger l'image d'une page sur internet et faire prédire la classe de cette image par le modèle.

Pour cela nous avons enregistré notre modèle et nous l'avons chargé dans FastAPI.

Démo :

V- Les difficultés rencontrées

La première difficulté rencontrée était d'éditer notre propre librairie de machine learning. Nous avons eu le choix entre plusieurs langages pour faire cette librairie. Nous avons choisi le C++.

Ne maîtrisant pas le langage et la configuration à mettre en place pour build le projet. Nous n'avons pas su implémenter notre librairie pour qu'elle soit complètement fonctionnelle.

Pour pouvoir faire la présentation d'aujourd'hui nous avons utilisé la librairie Scikit-learn.

Nous nous sommes confrontés à d'autres difficultés lors de la phase d'entraînement. Nous n'arrivions pas à obtenir de bons résultats d'accuracy pendant l'entraînement.

Plusieurs hypothèses se dessinaient :

- 1- La taille du dataset pas assez importante
- 2- Le nombre de neurones ou le nombre de couche de cache de neurones n'est pas adapté à notre problème.
- 3- Les données du dataset n'est pas assez normalisées.

La bonne hypothèse était la 3eme.

*montrer la fonction process data : img-size X/255.**2

Desc : de base nous avons choisi d'avoir un resize de l'image de 28 pixel par 28. La transformation suite au resize ne permettait pas d'avoir de bon resultat. En modifiant ce parametre nous avons réussi à maximiser le resultat.

Nous avons aussi eu a normaliser les données la valeur d'un pixel allant de 0 a 255 notre fonction de loss n'arrivé à minimiser l'erreur. En divisant la valeur du pixel nous avons constaté de meilleur résultat.

VI- Conclusion

Il est plus facile de faire la difference entre une page de romans et les autres autres classe de page, c'est un peu plus compliqué de différencier la page de comics et de manga et nous pensons que c'est pour cela que nous n'avons pas un score d'accuracy parfait.

Il aurait été très intéressant de faire la classification sur un seul type d'ouvrage par exemple le roman et d'avoir des classes en fonction du genre.

Par Exemple genre narratif , epistolaire ou théâtral.