

## TD2 : Reconnaissance d'image – FASCHION MNIST

### 1. Prétraitement des données

- Télécharger *fashion mnist* de la bibliothèque Keras.Datasets.
- Comment sont codées les données, de quel type sont-elles. Quelle est forme des données taille des données d'entrée et des targets pour le train et test.
- Comment se répartissent les targets au sein des deux jeux de données. Cette répartition est-elle cohérente pour les modèles d'apprentissage.
- Quels traitements doivent être appliqués à ces données pour un traitement par un NLP. Expliquez pourquoi faire ces prétraitements.
- Qu'aurait-il fallu faire si la répartition était déséquilibrée.
- Créer la liste des targets, correspondant aux données suivantes :

*T-shirts, Pantalons, Pulls, Robes, Manteaux, Sandales, Chemises, Baskets, Sacs, Bottines.*

- Afficher les neuf premières entrées avec le nom de leur target.
- Transformer vous données afin qu'elles puissent être traitées correctement par un réseau de neurones.

### 2. Crédation d'un premier modèle

- De quel type doit être la couche de sortie (nombre de neurones, fonction d'activation).
- On souhaite créer un premier model avec simplement deux couches cachées identiques de même taille (256 neurones) et avec des fonctions d'activation de type relu.
- Combien y-a-t-il de poids (paramètres) à caler. Détaillez les nombres de poids obtenus par couches et par neurones.
- Quel est selon vous les meilleurs paramètres pour la fonction compile. Donnez petite explication pour les trois paramètres (fonction de perte, fonction d'apprentissage, mesure de la qualité du réseau).

- Si l'on souhaite mettre à jour nos poids moins de 200 fois en ayant un pourcentage de validation de 20%, quelle est la taille des batchs.
- Entraînez le modèle sur 20 epochs, avec les paramètres précédents (batchs et pourcentage de validation).
- Affichez maintenant le taux d'erreur sur le test et la validation. Que constatez-vous, pensez-vous que pour notre réseau il soit judicieux de faire plus de 20 epochs.

### 3. Modification du modèle

- Ajoutez maintenant une troisième couche cachée identique aux précédentes, et effectuer un nouvel apprentissage.
- Affichez les taux d'erreurs sur les parties test et validation. Que constatez-vous par rapport au modèle à deux couches cachées. Est-il judicieux d'ajouter cette nouvelle couche.
- Reprenez le modèle à deux couches cachées et divisez le nombre de neurones par deux. Les résultats sont-ils dégradés.
- Sur la base d'un modèle à deux couches cachées avec 256 et 128 neurones, ajoutez des couches de dropout après chacune de ces deux couches avec un taux de désactivation de 40%. (model M)
- Effectuer l'apprentissage et affichez à nouveau les courbes des taux d'erreurs. Que peut-on en conclure.
- Augmentez le nombre de mise à jour des poids, en divisant par deux la taille de batchs. Quel impact cela a-t-il sur les taux d'erreurs.
- Si maintenant on utilise des fonctions d'activation de type sigmoïde ou elu sur les couches cachées, cela a-t-il un impact réel pour ce jeu de données.
- Compte tenu de tous ces tests quel bilan peut-on faire pour sur les réseaux de neurones (en général) au regard du nombre de couche, du nombre de neurones par couche, du nombre d'epochs et de batchs.

### 4. Analyse des résultats sur la partie test

- A partir du modèle M effectuer une prédiction sur les données de test. Évaluer les performances du modèle. Ces performances sont-elles

cohérentes par rapport aux résultats obtenus sur la partie validation au cours du *fit*.

- Transformer les résultats de la prédiction, de type one-hot encoding, en une valeur, entre 0 et 9, correspondant au numéro de l'habit. Enregistrez les prédictions et les valeurs des sorties test en dataframe, pour faciliter les traitements ultérieurs.
- Affichez pour chacun des objets de la base de données le nombre d'erreurs commis. Quel est l'objet le plus mal identifié.
- Afficher via la fonction heatmap de la bibliothèque seaborn la matrice de confusion entre les valeurs de teste et de prédiction. Vous remplacerez les nombres des abscisses x et des ordonnées y par les noms des objets de la liste targets (*T\_shirt*, ..., *Bottines*).
- Pour chemises classées en T-shirt affichez les 25 premières erreurs commises (5 images par 5). Ces affichages permettent-ils de comprendre les erreurs commises. Pensez-vous qu'en conservant la taille des images les résultats pourraient être améliorés.
- Effectuer la même chose pour les bottines classées en sandales. Pensez-vous que les résultats du modèle pourraient être améliorés en conservant la taille des images.