# École Polytechnique de Montréal
## Département Génie Informatique et Génie Logiciel
## INF8460 – Traitement automatique de la langue naturelle

## TP3 INF8460
## Automne 2022

### Objectifs d'apprentissage

- Coder un modèle Skipgram de base
- Se familiariser avec la librairie d'apprentissage profond Pytorch
- Produire des plongements lexicaux de documents et faire du clustering
- Visualiser les plongements lexicaux avec PCA

### Logiciels

Ce TP utilise la librairie NLTK, ScikitLearn(nom abrégé en sklearn), numpy, matplotlib et Jupyter notebook. Ces librairies ainsi que toutes les librairies de base provenant de python sont permises.

Il est conseillé d'installer Anaconda avec Python 3.10.6

### Corpus et code

Jeu de données : le jeu de données contient des textes du HuffPost dans l'attribut *headline* ainsi que leur catégorie.

Le zip du TP contient :
- Un jeu de données : train.csv et test.csv
- Le squelette du notebook qui doit être complété

### Travail à faire

### 1 Implémentation d'un modèle Skipgram avec Pytorch (20 points)

Dans cet exercice, vous allez implémenter un modèle Skipgram afin d'obtenir une représentation vectorielle de chaque mot, aussi appelé word embedding en anglais.

**1.1 (***3 points***)** Transformez chaque liste de jetons de la colonne "headline" en une séquence d'indices et stockez le résultat dans une nouvelle colonne "sequence".

**1.2 (***7 points)* Complétez la classe SkipGramDataLoader qui génère des exemples d'entraînement.

Implémentez les fonctions suivantes:
1. __init__

Complétez le constructeur et construisez l'ensemble d'entrainement comme indiqué plus haut

2. __getitem__(self, idx)

   Cette fonction retourne l'exemple à l'indice voulu idx provenant de l'ensemble d'entrainement. Le format voulu est sous la forme: tuple(jeton d'entrée, jeton prédit)
3. __len__(self)

   Cette fonction retourne la taille de l'ensemble d'entrainement

**1.3 (***5 points***)** Implémentez maintenant l'architecture du Skipgram

Implémentez les fonctions suivantes de la classe SkipGramSoftMax:

1. __init__

```
 Complétez le constructeur en initialisant aléatoirement les m
atrices de poids.
```

2. forward(self, batch_inputs: List[int])

   param: batch_inputs: List[int]

   return: valeurs softmax avec shape=(len(batch_inputs), len(vocab))

```
 forward est la méthode de la propagation vers l'avant, qui vo
us permet de retouner une distribution de probabilité sur le v
ocabulaire.



 Cette fonction prends en entrée un batch d'indices (voir exem
ple ci-dessus),

 puis calcule et retourne les valeurs log_softmax correspondan
tes.
```

3. embed(self, indices)

   param: indices: List[int]

   return: représentations vectorielles avec shape=(len(indices), embed_size)

```
Cette fonction prend en entrée un batch de d'indices, et retou
rne les représentations vectorielles (projection/plongements l
exicaux). Ceux-ci sont obtenus en calculant le vecteur h dans
la figure ci-dessus.
```

**1.4 (***5 points***)** Entrainez le modèle skipgram

Complétez le processus d'entraînement. Les étapes à suivre sont définies ci-bas. **Voir le tutoriel pour plus d'aide.**

1. Initialiser les gradients à zéro

2. Passer les batchs_inputs dans le skipgram afin d'obtenir les valeurs de sortie.
3. Passer les sorties du réseau et les labels dans la fonction de perte (loss function)
4. Calculer les valeurs de gradients avec la fonction backward()
5. Mettre à jour les poids avec l'optimiseur

## 2 Clustering K-MEANS avec représentation vectorielle de documents (20 points)

Dans cet exercice, l'objectif est de regrouper d'une manière non-supervisée les documents (headline) afin de les classifier, les performances des différentes représentations illustreront l'efficacité de word2vec.

Nous allons d'abord construire 3 représentations vectorielles différentes qui seront comparées avec l'algorithme de clustering.

**2.1 (***3 points***)** Encodage de documents avec Tfidf + LSA

**2.2 (***3 points***)** Encodage de documents avec votre modèle Skip-gram

**2.3 (***2 points***)** Encodage de documents avec les plongements word2vec pré-entrainés de Gensim

**2.4 (***6 points***)** Clustering avec K-Means

**2.5 (***1 points***)** Exécuter la fonction cluster_and_eval sur les 3 représentations obtenues en 2.1, 2.2 et 2.3

**2.6 (***5 points***)** Discussion (4 lignes maximum par réponse)

2.6.1 Si vos implémentations fonctionnent, vous devriez remarquer que le Skipgram avec softmax (2.2) est le modèle obtenant les meilleurs résultats. Pourquoi selon vous, le Skipgram avec softmax (2.2) arrive à des meilleurs résultats que le skipgram avec negative sampling (2.3) ?
2.6.2 Pourquoi selon vous le modèle skipgram fonctionne aussi bien dans ce contexte de clustering?
2.6.3 Pour un exemple positif (batch_size=1), combien de paramètres sont mis à jour pour:
   a) le skipgram avec softmax
   b) BONUS: le skipgram avec negative sampling (5 negative samples):

## 3 Visualisation des représentations vectorielles (10 points)

Ici les représentations vectorielles seront projetées dans une dimension inférieure, cela nous permet de mieux comprendre la composition des clusters créés par l'algorithme K-Means.

Projetez les représentations de documents provenant du skipgram softmax (2.2) dans un espace 2D à l'aide de PCA. Le résultat devrait être un nuage de points ayant 3 couleurs (1 par catégorie), chaque point du nuage est un exemple et a la couleur de sa catégorie. Par exemple, si on définit la catégorie Business comme étant la catégorie avec la couleur bleue, on affiche tous les points de cette catégorie en bleu. Assurez-vous que la couleur de chaque catégorie soit bien identifiée.

Utilisez l'ensemble d'entrainement pour faire cette visualisation.

## LIVRABLES

Vous devez remettre sur Moodle un zip contenant les fichiers suivants :

**1-** *Le code* : Vous devez compléter le squelette inf8460_tp3.ipynb sous le nom equipe_i_inf8460_TP3.ipynb (i = votre numéro d'équipe). Indiquez vos noms et matricules au début du

notebook. Ce notebook reprend les différentes questions, et doit contenir les fonctionnalités requises avec des commentaires appropriés. Le code doit être exécutable sans erreur et accompagné de commentaires appropriés de manière à expliquer les différentes fonctions. Les critères de qualité tels que la lisibilité du code et des commentaires sont importants. Tout votre code et vos résultats doivent être exécutables et reproductibles ;

**2-** Un fichier html représentant votre notebook complètement exécuté sous format html

**3-** Un document *contributions.txt :* Décrivez brièvement le pourcentage de contribution et la contribution de chaque membre de l'équipe. Tous les membres sont censés contribuer au développement. Bien que chaque membre puisse effectuer différentes tâches, vous devez vous efforcer d'obtenir une répartition égale du travail.

## EVALUATION

Votre TP sera évalué selon les critères suivants :

1. Exécution correcte du code
2. Performance attendue du modèle
3. Organisation du notebook
4. Qualité du code (noms significatifs, structure, performance, gestion d'exception, etc.)
5. Commentaires clairs et informatifs
6. Réponses correctes aux questions de réflexion