Assignment 3 iNLP - Report

HyperParams Used:

svd-word-vectors.pt:

vector_size = 100

skip-gram-word-vectors.pt:

vector_size=100, negative_sampling = 5, learning_rate=0.01, epochs=10

svd-classification-model.pt & skip-gram-classification-model.pt

embedding_dim = 100 ,hidden_dim = 256, learning_rate =.001, epochs =3,batch_size =32

SVD evaluation metrics

window-size = 2

```
Train Metrics:
Accuracy: 0.8754
Precision: 0.8798
Recall: 0.8754
F1 Score: 0.8749
Confusion Matrix:
[[20755   879   959  1451]
 [  361 23016    82   483]
 [ 1033   511 18490  4055]
 [  771   369  1010 21775]]

Test Metrics:
Accuracy: 0.8571
Precision: 0.8614
Recall: 0.8571
F1 Score: 0.8563
Confusion Matrix:
[[1605   83   80  132]
 [  40 1812   11   37]
 [ 101   48 1403  348]
 [  57   44  105 1694]]
```

window- size = 1 & 5 respectively

```
Epoch [1/3], Loss: 0.7044
Epoch [1/3], Validation Loss: 0.5363, Validation Accuracy: 0.8085
Epoch [2/3], Loss: 0.4729
Epoch [2/3], Validation Loss: 0.4377, Validation Accuracy: 0.8455
Epoch [3/3], Loss: 0.4176
Epoch [3/3], Validation Loss: 0.4274, Validation Accuracy: 0.8522

Window Size: 1
Test Metrics:
Accuracy: 0.8543
Precision: 0.8549
Recall: 0.8543
F1 Score: 0.8539
Confusion Matrix:
[[1636   77   91   96]
 [  57 1780   25   38]
 [ 128   32 1459  281]
 [  93   41  148 1618]]
Epoch [1/3], Loss: 0.4957
Epoch [1/3], Validation Loss: 0.4187, Validation Accuracy: 0.8577
Epoch [2/3], Loss: 0.3823
Epoch [2/3], Validation Loss: 0.3623, Validation Accuracy: 0.8731
Epoch [3/3], Loss: 0.3494
Epoch [3/3], Validation Loss: 0.3549, Validation Accuracy: 0.8779

Window Size: 5
Test Metrics:
Accuracy: 0.8700
Precision: 0.8703
Recall: 0.8700
F1 Score: 0.8700
Confusion Matrix:
[[1635   72  123   70]
 [  46 1798   30   26]
 [  67   30 1595  208]
 [  79   30  207 1584]]
```

Observation on window sizes

- A higher window size (5) attributes to better metrics overall, captures more semantic info , since it is a news classification dataset , identifying the topic class is easier with broader context.
- but taking too high window sizes can also be problematic because:
  - A larger window size captures more context, which can help the model understand broader usage patterns. However, this doesn't always lead to better performance because if the window is too large, it might start to include noise, capturing relationships between words that are not meaningful.

Skip-Gram evaluation metrics

window-size = 2

```
Accuracy: 0.8934
Precision: 0.8942
Recall: 0.8934
F1 Score: 0.8930
Confusion Matrix:
[[21064   853  1043  1084]
 [  202 23433   128   179]
 [  782   377 19923  3007]
 [  757   414  1407 21347]]
Accuracy: 0.8836
Precision: 0.8845
Recall: 0.8836
F1 Score: 0.8832
Confusion Matrix:
[[1652   82   82   84]
 [  19 1847   18   16]
 [  60   31 1547  262]
 [  60   43  128 1669]]
```

window- size = 1

```
Epoch 1/3, Loss: 0.7108
Epoch 2/3, Loss: 0.4280
Epoch 3/3, Loss: 0.3888
Results for window size 1:
Accuracy: 0.8704, Precision: 0.8721, Recall: 0.8704, F1 Score: 0.8704
Confusion Matrix:
[[1616   76  105  103]
 [  40 1800   19   41]
 [  74   22 1528  276]
 [  66   37  126 1671]]
```

window-size = 5

```
Results for window size 5:
Accuracy: 0.8883, Precision: 0.8891, Recall: 0.8883, F1 Score: 0.8881
Confusion Matrix:
[[1651   71   90   88]
 [  13 1846   24   17]
 [  59   21 1583  237]
 [  58   30  141 1671]]
```

window sizes -reasoning

First test was done on a window size of 2 (randomly chosen) on which a good accuracy was observed on both. so a lesser window size and a greater window size was taken to observe whether the model accuracy would increase or decrease.

Observation on window sizes

- Here also, a higher window size (5) attributes to better metrics overall, because capturing more semantics info lets us classify the news better , since we have more context.
- but as discussed befoer taking too high window sizes can also be problematic because:
  - introduction of noise risk is still present here

Analysis on SVD vs Skip Gram

Since Skip gram performed slightly better, below is the reasoning:

- **Contextual Sensitivity:** Skip-Gram is designed to predict context words given a target word, making it highly sensitive to the context in which words are used. This can lead to a better understanding of word meanings in different news categories, improving classification accuracy.
- **Handling of Polysemy:** Skip-Gram's ability to capture multiple meanings of the same word based on different contexts can be particularly beneficial in news text, where words can have different meanings based on the article's topic or category.
- **Richer Semantic Relationships:** Skip-Gram embeddings tend to capture a wide range of semantic relationships, which is valuable in distinguishing between various types of news articles that may use similar vocabulary but in different contexts or with different connotations.
- **Flexibility in Window Size:** The ability to adjust the window size in Skip-Gram allows for fine-tuning based on the specific requirements of the news classification task. A larger window can capture broader semantic contexts, while a smaller window focuses on syntactic relationships, offering flexibility to optimize performance.

Shortcomings of Skip-Gram

- **Computational Intensity:** Training a Skip-Gram model can be computationally expensive and time-consuming, especially for large vocabularies and datasets, due to the nature of predicting context words for each target word.
- **Memory Usage:** Skip-Gram models, particularly those with large contexts or vocabularies, can require significant amounts of memory, making them less feasible on limited hardware.
- **Risk of Overfitting:** Skip-Gram models have the potential to overfit, especially with smaller datasets, where the model might learn peculiarities of the training data rather than generalizable linguistic patterns.
- **Handling of High Frequency Words:** Without proper handling techniques like subsampling, high-frequency words can dominate the training process, leading to less informative embeddings for these words.
- **Context Window Size Sensitivity:** The choice of context window size can significantly affect the model's performance, and finding the optimal size requires experimentation, which can be cumbersome.
- **Diminishing Returns with Rare Words:** While Skip-Gram is better at handling rare words than some other models, it still faces challenges in learning high-quality embeddings for

words with very few occurrences.

Shortcomings of SVD-Based Embeddings

- **High Initial Computational Cost:** Constructing the co-occurrence matrix and performing SVD on large corpora can be computationally demanding and require significant initial processing time.
- **Memory Requirements for Co-occurrence Matrix:** Storing and processing the co-occurrence matrix, especially for large vocabularies and window sizes, can consume a large amount of memory.
- **Static Nature:** Once generated, SVD-based embeddings do not easily accommodate updates or additions to the dataset without re-computation, making them less flexible for dynamic datasets.
- **Sparse Representation Issues:** Although reducing dimensionality, the SVD process can still result in relatively sparse vectors compared to other embedding methods, potentially affecting downstream task performance.
- **Limited Contextual Sensitivity:** Because they are based on global co-occurrence statistics, SVD-based embeddings may not capture the nuances of word usage in specific contexts as effectively as context-predictive models like Skip-Gram.