NATIONAL UNIVERSITY
of Computer & Emerging Sciences

REPORT: WORD EMBEDDING ASSIGNMENT

MUHAMMAD ALI HARIS

22k-4239

DLP SECTION 6-A

ASSIGNMENT#3

1. Objective

This assignment aimed to:

- Implement Skip-gram and CBOW models using PyTorch
- Train them on a small corpus
- Visualize the learned word embeddings
- Evaluate semantic understanding using analogy tasks

2. Preprocessing Steps

Dataset Used: text_corpus.txt
Tokenization: Converted sentences into lowercase tokens
Cleaning: Removed punctuation and common stop words
Vocabulary Mapping: Created word2idx and idx2word mappings
Training Pair Generation:
Skip-gram: Target word →Context words
CBOW: Context words →Target word

3. Skip-gram Model

Architecture:
Input: One-hot encoded target word
Embedding Layer → Hidden Vector
Output Layer: Softmax over vocabulary

- **Loss Function**: Cross Entropy Loss
- **Optimizer**: Adam

Learned meaningful representations of words.
Loss consistently decreased with training.

4. CBOW Model

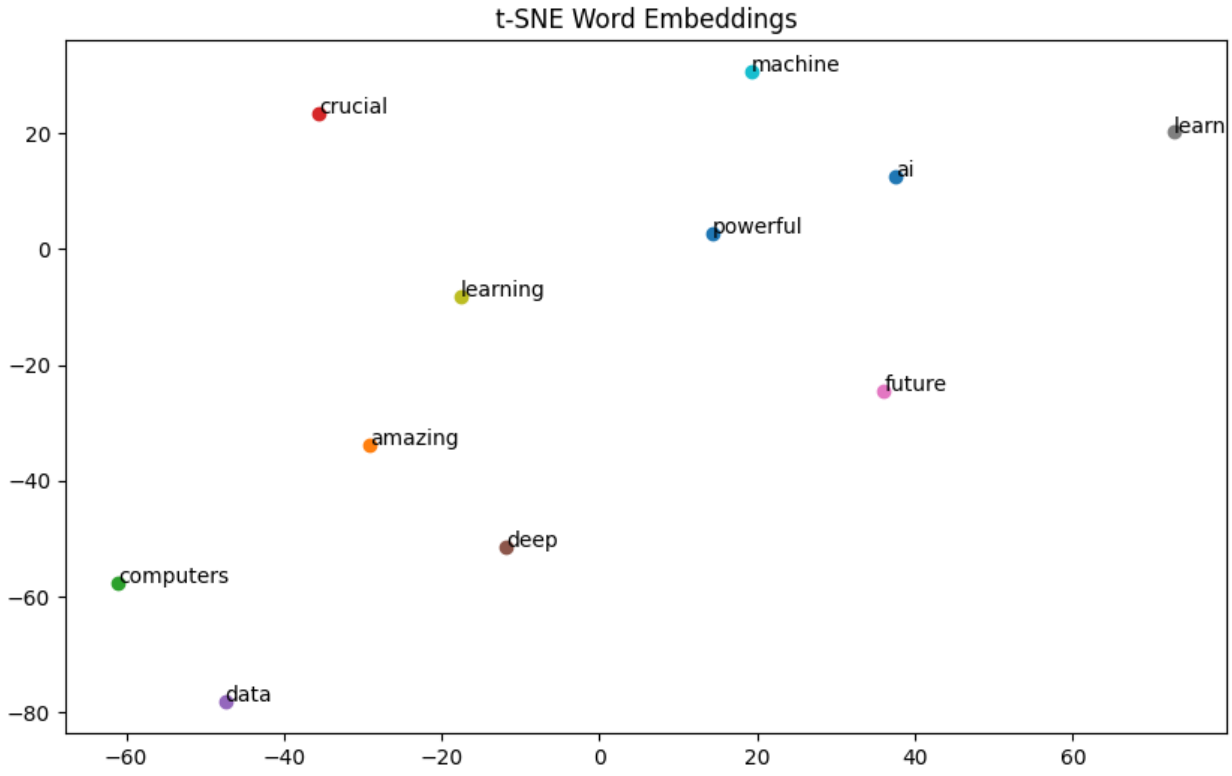
Architecture:
Input: Context word embeddings (average)
Linear Layer → Softmax

Training: Similar approach with SGD optimizer
More stable training than Skip-gram.
Captured some syntactic patterns even in a small dataset.

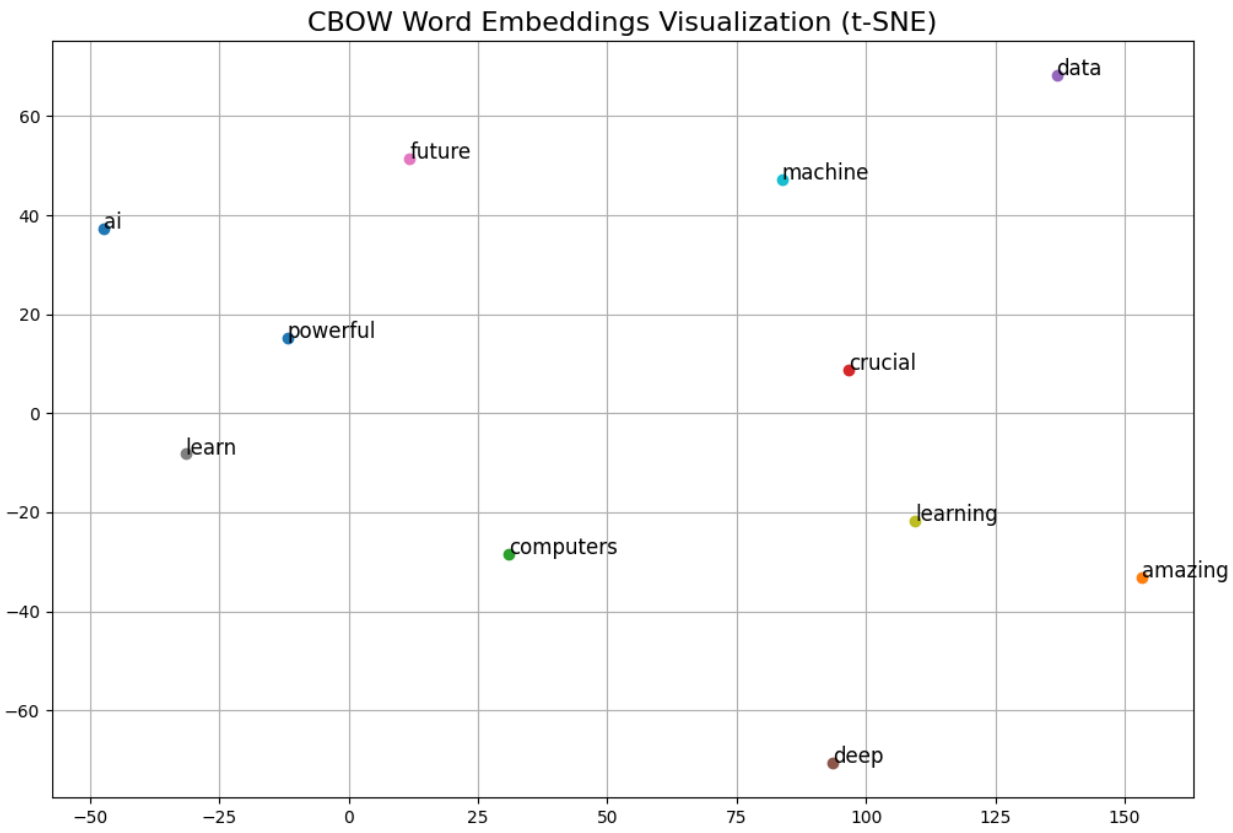
5. Visualization

Used **t-SNE** to reduce high-dimensional word embeddings to 2D.

SKIP-GRAM Visualization:



CBOW Visualization:



Words like "powerful", "ai", and "learn" formed meaningful clusters, suggesting that the embeddings learned semantic relationships successfully.

6. Challenges

t-SNE instability: Sometimes plotted points oddly depending on perplexity.
Small Dataset: Limited ability to learn deeper semantics.
Token Overlap: Had to handle repeated words smartly for better generalization.

7. Conclusion

- Successfully implemented and trained both Skip-gram and CBOW from scratch using PyTorch.
- Embeddings showed semantically rich clusters even with a small corpus.
- t-SNE helped in visualizing relationships between words.