SKIP-GRAM WORD2VEC WITH NEGATIVE SAMPLING ON WIKIPEDIA

---

1. INTRODUCTION

Word embeddings are dense vector representations of words that capture semantic and syntactic relationships from large text corpora. Traditional one-hot representations fail to encode similarity between words, whereas distributed representations learned by neural models overcome this limitation.

In this assignment, we implement Skip-gram with Negative Sampling (SGNS) from scratch and train it on a Wikipedia corpus (enwik8). The learned embeddings are evaluated through cosine similarity comparison with pretrained Word2Vec vectors, word analogy tasks, and bias detection as discussed in Chapter 5.

---

2. DATASET DESCRIPTION

The dataset used is enwik8, a cleaned Wikipedia text corpus.

- Source: Matt Mahoney's Large Text Compression Benchmark
- Size: ~34 MB (compressed)
- Content: English Wikipedia articles

Preprocessing Steps

- Converted text to lowercase
- Removed punctuation, numbers, and special characters
- Tokenized text into words
- Removed rare words using a minimum frequency threshold

After preprocessing, the corpus contained approximately 14 million tokens with a vocabulary size of ~72,000 words.

---

3. SKIP-GRAM WITH NEGATIVE SAMPLING

3.1 Skip-gram Model

The Skip-gram model learns word representations by predicting surrounding context words given a center word. For a center word w_c, the objective is to maximize the probability of observing its context words w_o.

Unlike CBOW, Skip-gram focuses on learning high-quality representations for infrequent words.

3.2 Negative Sampling

Computing a full softmax over the entire vocabulary is computationally expensive. Negative Sampling addresses this by reformulating the objective as a binary classification task:

- Positive samples: actual (center, context) pairs
- Negative samples: randomly sampled noise words

The loss function is defined as:

$$\mathcal{L} = -\log\ \sigma(v_c \cdot v_o) - \sum_{i=1}^{k} \log\ \sigma(-v_c \cdot v_{n_i})$$

where:

- $v_c$is the center word vector
- $v_o$is the context word vector
- $v_{n_i}$are negative samples
- $\sigma$is the sigmoid function

3.3 Model Architecture

The model consists of:

- Input embedding matrix (center words)
- Output embedding matrix (context words)

Both embeddings are learned during training. The implementation follows the original Word2Vec formulation by Mikolov et al. (2013).

4. TRAINING SETUP

Hyperparameters Used

Parameter	Value
Embedding Dimension	100
Window Size	5
Negative Samples	5
Minimum Word Count	5
Epochs	3
Optimizer	Adam

Mini-batch stochastic gradient descent was used to train the model efficiently.

---

5. EVALUATION

5.1 Cosine Similarity Comparison

Cosine similarity was used to measure semantic similarity between word pairs. The similarity scores obtained from our trained embeddings were compared with pretrained Gensim Word2Vec (Google News) vectors.

Despite being trained on a much smaller corpus, our embeddings showed similar semantic trends, validating the correctness of the implementation.

5.2 Word Analogy Task

Word analogies were evaluated using vector arithmetic of the form:

$$\text{vec}(b) - \text{vec}(a) + \text{vec}(c)$$

Examples tested:

- king − man + woman ≈ queen
- paris − france + italy ≈ rome

The model successfully captured several linear semantic relationships, demonstrating a key property of Word2Vec embeddings.

---

6. BIAS DETECTION IN WORD EMBEDDINGS

Word embeddings are known to reflect social biases present in training data. To analyze this, a gender direction was computed using word pairs such as:

- man − woman
- he − she
- boy − girl

Profession words were projected onto this gender direction using cosine similarity.

Observations

- Words like *engineer* and *programmer* showed male bias
- Words like *nurse* and *assistant* showed female bias

This confirms that distributional word embeddings can encode societal biases, as discussed in Chapter 5.

7. DISCUSSION

The results demonstrate that Skip-gram with Negative Sampling is capable of learning meaningful semantic representations even

from a relatively small corpus. While pretrained embeddings trained on massive datasets achieve higher accuracy, the trends observed in similarity, analogy, and bias tasks closely align with expected Word2Vec behavior.

---

8. CONCLUSION

In this work, Skip-gram with Negative Sampling was successfully implemented from scratch and trained on Wikipedia text. The learned embeddings were evaluated using cosine similarity, word analogy tasks, and bias detection. The results confirm the effectiveness of the SGNS approach and highlight important ethical considerations related to bias in learned representations.