Deep Learning for NLP - Lab 1

Marine Vieillard

16/11/2025

Sentiment analysis is an important field of natural language processing. In this project, we aim to determine whether a movie review expresses a positive or negative opinion. We compare two approaches: a Bag-of-Words (BoW) classifier, which represents texts based on the words they contain, and a Convolutional Neural Network (CNN), which has the ability to capture local structures and take greater advantage of the structure of language.

The dataset contains 600,000 reviews with 300,000 positive and 300,000 negative texts. After cleaning the data and processing it, we apply both models, train each model and evaluate them. This project thus highlights the differences between a traditional method and a more advanced neural model for sentiment classification.

1 Bag of word model

1.1 Model description

The Bag-of-Words model relies on a simple representation of text: each sentence is converted into a set of words, independently of their order or grammatical structure. In a variant such as the Continuous Bag-of-Words (CBOW) model, each word is projected into a vector space using an embedding layer, and the vectors corresponding to the words in a sentence are averaged to obtain a global representation of the text. This approach captures denser information than simple word counting, while preserving the idea of BoW: the sentence is represented by a single vector summarizing its lexical content. Even though this kind of model cannot take advantage of word order or syntactic relationships, it is still useful for basic classification tasks.

1.2 Model implementation

The implementation uses a class called `CBOW_classifier` derived from `nn.Module`. The model starts with an embedding layer that changes the index of each word into a dense vector of a set size. To represent a sentence, the embeddings of all its words are combined by computing their average, which is the core of the CBOW principle.Then, this representation goes through a perceptron, which can be either a single linear layer or a small network with an optional hidden layer that is turned on with ReLU. The model outputs a logit, that is an unnormalized value later converted into a probability using a sigmoid function.

To train the model, we use the `BCEWithLogitsLoss` loss function, which combines a sigmoid and Binary Cross-Entropy into one stable operation. It is more robust than `BCELoss`, which would require applying a sigmoid manually before the loss and could give `NaN` values when logits get too big or too small.

The evaluation of the model will be completed using the accuracy metric, which quantifies the number of accurate predictions across the size of the dataset. This metric is particularly appropriate for our task, since we have a balanced binary classification problem. Accuracy would therefore provide a simple and direct interpretation: the higher it is, the better the model distinguishes between positive and negative reviews. Accuracy on the validation set is calculated at each training epoch to monitor the model's generalization capability.

At last, a confusion matrix is plotted to make a more detailed analysis of the model errors for both the training and validation sets. It shows the correct positive predictions, the correct negative predictions, and any possible misclassifications.

1.3 Results and analysis

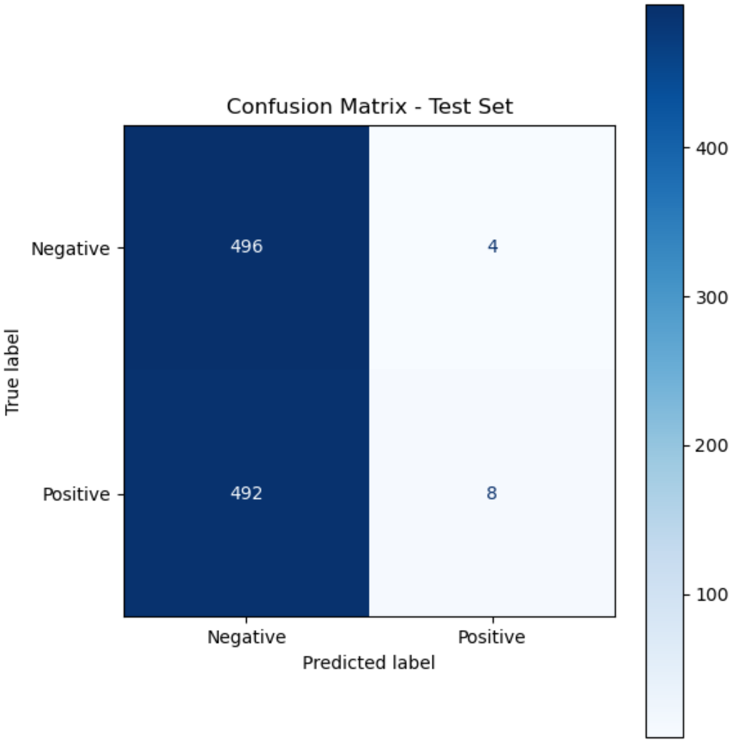

Figure 1: Confusion matrix - Bag of word model

The evaluation of the CBOW model on the test set (figure 1) reveals very limited performance. According to the confusion matrix, the model correctly classifies 496 out of 500 examples that are actually negative. Only 8 of the 500 positive reviews are correctly identified, whereas 492 are mistakenly predicted to be negative. This extreme imbalance in predictions indicates that the model almost systematically predicts the “negative” class, which leads to an overall accuracy of about 50.4%. This value may seem close to random chance. It reflects the model’s inability to learn a meaningful decision boundary between the two classes. The analysis shows that the CBOW model does not extract enough discriminative information in this configuration to differentiate between positive and negative reviews.

2 CNN model

2.1 Model description

The convolutional neural network (CNN) used for sentiment analysis is designed to capture local n-gram patterns in text, allowing it to detect short discriminative phrases that correlate with positive or negative sentiment. The model first maps each token to a dense vector through an embedding layer, enriching the representation of the input sequence. A ReLU activation is used to add non-linearity after a 1-D convolutional layer with a fixed window size slides over the embedded sentence and extracts local features. A max-pooling operation aggregates the most essential feature from each filter across the sentence, resulting in a fixed-size representation regardless of input length. Finally, a fully connected layer transforms these features into a single logit for binary classification. Because it can identify important phrases like ”very good” or ”not worth” without depending on global sentence structure, this architecture is especially useful for sentiment analysis.

2.2 Model implementation

Several design choices were taken to accommodate variable-length text inputs in the implementation. The vocabulary expands with the inclusion of special tokens (PAD, BOS, and EOS), which automatically embellish each sentence with beginning and end-of-sequence markers. Sentences are padded dynamically within each batch to match the longest sequence, enabling efficient batching while preserving model flexibility. The embedding layer maps each token to a vector of size 128 that is transposed to the expected input shape of the convolutional layer. The local tri-gram features are then extracted from a 1-D convolution with 100 filters and a window size of 3 before being max-pooled into a compressed representation of the sentence. The dropout layer is used to regularize the model, and the output sentiment logit is computed by a final linear layer. The forward method organizes all feature preprocessing (padded sequences, tensor batching) to ensure the model can accept raw token index sequences directly.

2.3 Results and analysis

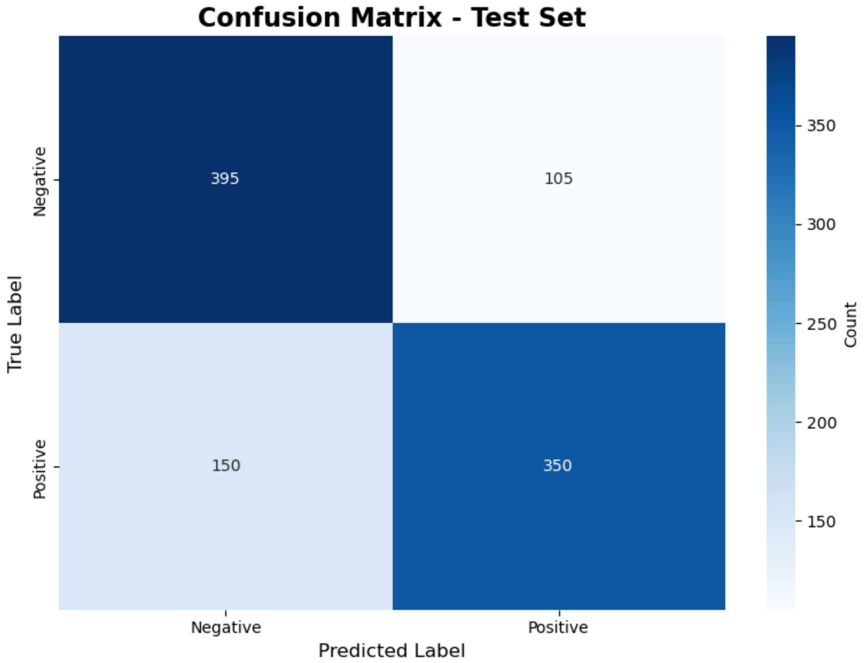

Figure 2: Confusion matrix - CNN model

The evaluation of the CNN on the test set (figure 2) demonstrates a clear improvement over the Bag-of-Words model. According to the confusion matrix, 350 out of 500 positive reviews and 395 out of 500 negative reviews are correctly classified by the model. Misclassifications are more balanced than in the CBOW model: the CNN incorrectly predicts 105 negative reviews as positive and 150 positive reviews as negative. This distribution reflects a model that captures meaningful linguistic patterns rather than collapsing into a single dominant prediction strategy.

The resulting accuracy of the CNN is 74.5%, which is significantly higher than the CBOW model's near-chance performance of 50.4%. The CNN clearly succeeds in learning discriminative features from the data, even though it still shows some asymmetry, being marginally better at detecting negative sentiment than positive. This distinction demonstrates how convolutional filters can capture local sentiment frequency representations. Overall, the CNN provides a much more robust and meaningful separation between positive and negative reviews, confirming the advantage of neural architectures for sentiment classification.

2.4 Hyperparameters optimization

To improve the performance of the initial CNN model, a Random Search procedure was applied to explore a broader set of hyperparameter configurations. Random Search consists in sampling combinations of hyperparameters, such as embedding dimension, number of filters, window size, learning rate, and dropout probability, from predefined ranges. Unlike grid search, which evaluates all possible combinations, random search enables the exploration of a wide and diverse subset of the hyperparameter space at significantly lower computational cost.

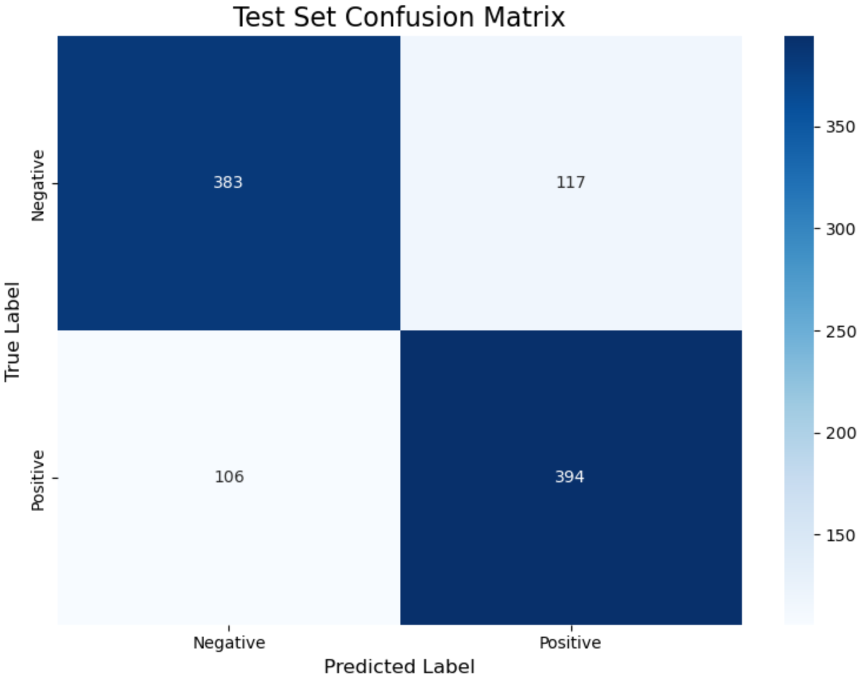

Figure 3: Confusion matrix - CNN model with hyperparameters tuning

The improved model demonstrated an advantage over the original CNN. The updated confusion matrix (figure 3) shows it accurately identifies 383 negative reviews and 394 positive reviews, while misclassifying 117 negatives and 106 positives. We now have an accuracy of 77.7% compared to the original CNN. Although the new model still performs adequately on negative reviews, the optimized version enables better identification of positive sentiment. This advancement diminishes the class imbalance and leads to a better capacity to capture sentiment-specific patterns.

Overall, the random search–optimized CNN offers a more balanced and accurate classification, demonstrating that hyperparameter tuning can improve model generalization and mitigate limitations of the original architecture.

3 Conclusion

This project explored two distinct approaches to sentiment analysis : a Bag-of-Words classifier and a Convolutional Neural Network. The comparison clearly highlighted the limitations of simple lexical representations: the BoW model achieved an accuracy close to random guessing. Its confusion matrix revealed a strong bias toward predicting negative reviews, suggesting that it was unable to identify significant discriminative patterns.

On the other hand, the CNN showed a much stronger capacity to extract local textual features and model the structure of language, resulting in significantly higher accuracy and more balanced predictions. Some improvements were achieved with hyperparameter optimization using Random Search, which allowed the model to explore many configurations and converge toward a more robust architecture.

Overall, this work shows that neural architectures outperform traditional methods for sentiment classification tasks. To further improve performance, future extensions might investigate pretrained language models, like transformers, or try deeper convolutional stacks and larger embedding dimensions.