

Deep Learning

# Project: IMDB sentiment classification

By Grégoire Ouerdane

29/05/2025

# Problem statement

- Goal: build a model to classify IMDB movie reviews as either positive or negative;
- Sentiment analysis helps in the understanding of public opinion;
- There are multiple applications in marketing, recommendation systems, content moderations, etc.

# Dataset description

- **Dataset:** IMDB movie reviews (from TorchText);
- 25 000 training reviews;
- 25 000 testing reviews;
- Binary labels: 0 (negative), 1 (positive).
- **Preprocessing:**
  - Tokenization with BERT tokenizer;
  - Padding to fixed sequence length;
  - created DataLoaders for batching.

# Model 1: LSTM

- **Architecture:**
  - Embedding layer (pretrained BERT embeddings);
  - LSTM layer;
  - Fully connected output layer.
- **Training setup:**
  - Loss: CrossEntropyLoss;
  - Optimizer: Adam;
  - Epochs: 40;
  - Device: GPU.

# Model 2: Transformer encoder

- **Architecture:**
  - Embedding layer (pretrained BERT embeddings);
  - TransformerEncoder block (PyTorch's nn.TransformerEncoder);
  - Fully connected output layer.
- **Training setup:**
  - Loss: CrossEntropyLoss;
  - Optimizer: Adam;
  - Epochs: 10;
  - Device: GPU.

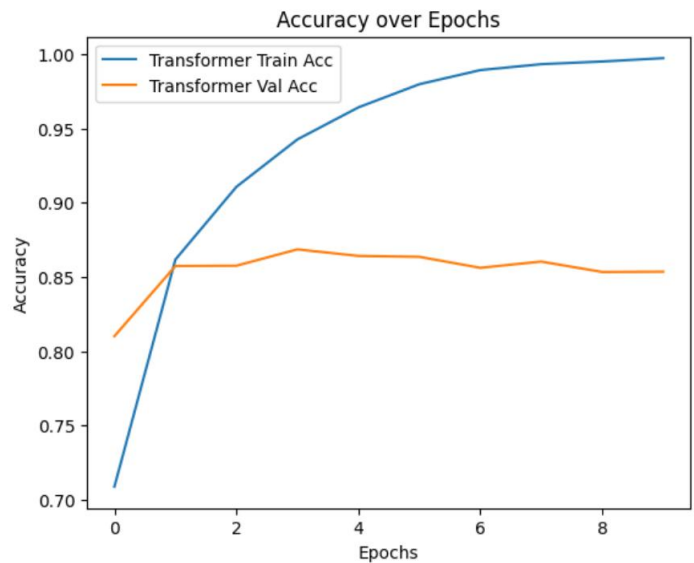
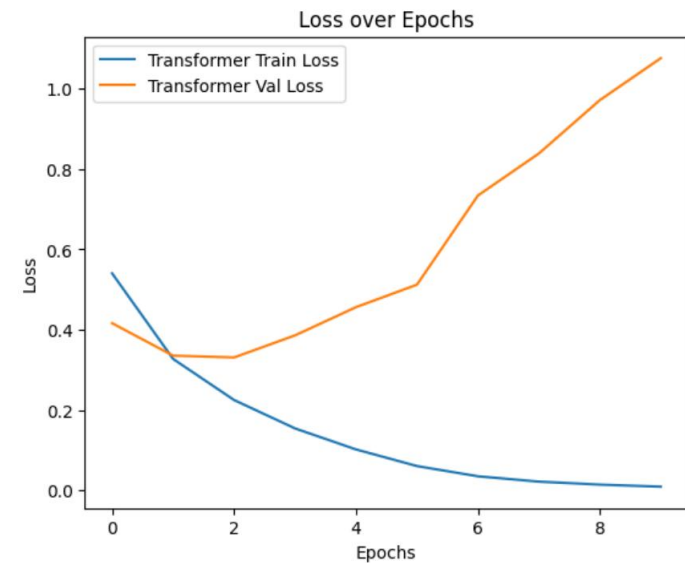
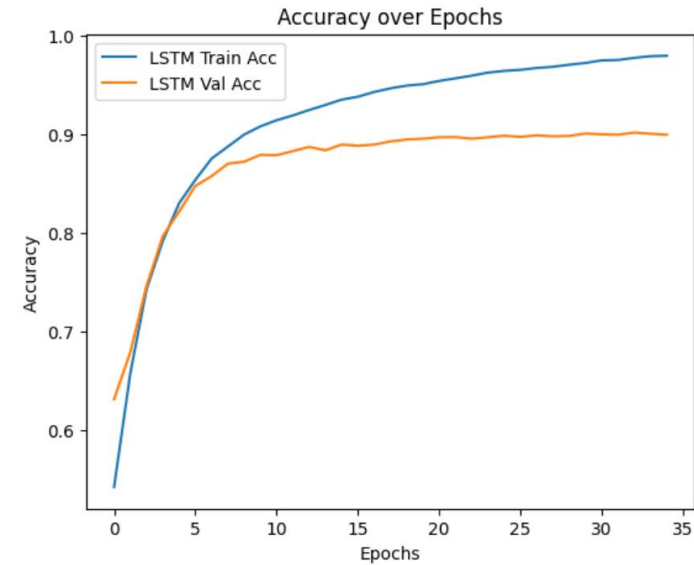
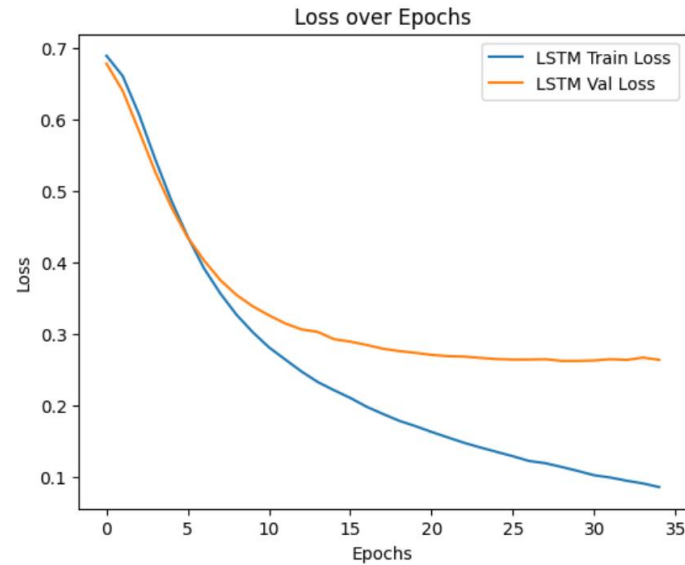
# Training details

## Setup:

- Loss: CrossEntropyLoss;
- Optimizer: Adam;
- Epochs: 40 (LSTM) or 10 (Transformer);
- Batch size: 16;
- Evaluation metric: accuracy;
- Device: GPU.

LSTM converged quickly and accurately, while Transformer required more tuning for better stabilization.

# Training and validation curves



# Final results table

## Model performance comparison:

	Model	Test Accuracy	Train Accuracy (last epoch)	Validation Accuracy (last epoch)
0	LSTM	0.89012	0.97955	0.8994
1	Transformer	0.83144	0.99725	0.8536

## Observations:

- LSTM outperformed the Transformer model;
- Maybe because of a simpler architecture better suited for small datasets.



# Challenges and fixes

During the implementation of models I faced the following challenges:

- Ensuring consistent data preprocessing accross models;
- Managing input dimensions for Transformer layers;
- Model convergence (Transformer required more tuning and LSTM required more epochs);
- Debugging.

# Conclusion

- Successfully implemented two deep learning models for sentiment classification;
- Obtained additional experience with LSTM and Transformers;
- We saw that model choice significantly affects results.
- In the future I plan:
  - Try BERT fine-tuning;
  - Use larger datasets.

Thank you !