Faiza Abdullah

Lab 06

ITAI 2376 Deep Learning in Artificial Intelligence

Professor: Patricia Mcmanus

REFLECTIVE JOURNAL: MODULE 2, LAB 5 - FINETUNING BERT

INTRODUCTION

This lab oriented with fine-tuning DistilBERT—a lightweight transformer model—to classify Amazon product review sentiments, where initial run trained for 10 epochs, while the revised run extended to 20 epochs to explore validation improvements, particularly in val_loss, as prompted by the lab's challenge. Across both, I learned to load and format data, leverage a pre-trained model, and train it for a specific task, building on prior labs' NLP foundations.

Learning Insights

- **Data Preparation:** Tokenizing with DistilBertTokenizerFast (In[9]) and creating ReviewDataset (In[10]) formatted text for BERT's input needs.
- **Pre-trained Model:** Loading DistilBERT (In[11]) and freezing all but the classifier layer (In[12]) showcased transfer learning efficiency.
- **Training Dynamics:**
    - ❖ Initial Run (10 Epochs): Validation accuracy rose from 0.620 to 0.835, val_loss dropped from 0.632 to 0.426.
    - ❖ Revised Run (20 Epochs): Validation accuracy peaked at 0.890, val_loss fell from 0.410 to 0.303, showing deeper optimization.
- **Prediction:** Outputs like Prediction: 1 for a positive review (In[35], first doc) validated model performance.

In real life, this powers review platforms as the lab ties preprocessing (Lab 1), embeddings (Lab 3), and sequence modeling (Lab 4) into a modern NLP pinnacle.

Most Impactful Learning Moments: The encoded sentence (Out[26], first doc)—"An excellent resource…" to [101, 2019, …, 102]—revealed BERT's tokenization, a leap from Lab 2's BoW. Training outputs were pivotal:

> **10 Epochs:** val_acc 0.835 and val_loss 0.426
>
> **20 Epochs:** Peak val_acc 0.890 and val_loss 0.303 showed extended potential, albeit fluctuations (e.g., 0.870 at epoch 12).

Challenges and Struggles

Conceptually, BERT's bidirectionality was abstract—how does it "look both ways"? Freezing layers (In[12]) confused me until I grasped its efficiency. Technically:

> **10 Epochs:** The first epoch's slow 48.159 seconds tested the setup.
>
> **20 Epochs:** Memory warnings and repeated outputs suggested resource strain or logging errors.

Problem Solving Strategies: I visualized BERT as reading a sentence fully before judging, linking to attention_mask (In[25]). For freezing, I traced named_parameters() to see classifier weights update. I mitigated runtime by trusting the 2000-data limit (In[7]) and noted batch size reduction tips (PAGE1), though not needed. I adopted a "decode-to-understand" tactic—e.g., reversing tokens (Out[28]) to confirm encoding. Comparing epoch outputs across runs became my metric for progress.

Personal Growth

This led to evolution from sequential (Lab 4) to contextual modeling, seeing BERT as a holistic reader. Confidence grew in handling large models, appreciating fine-tuning's balance of power and practicality. BERT's rapid accuracy jump (0.620 to 0.835 in 10 epochs, 0.890 in 20), the val_loss drop to 0.303 (PAGE11, first doc) versus 0.426 (PAGE9, second doc) surprised me. Academically, I think it can be applied to social media analysis, capturing nuanced sentiments. Professionally, it is ideal for customer feedback systems, offering precise insights.

Critical Reflection

I would test smaller batch sizes (PAGE1) to manage memory and add early stopping to halt at peak val_loss, avoiding 20-epoch overkill. Using the full dataset could reveal true potential, if time allows. Lab 5 crowns NLP progression—preprocessing (Lab 1), vectorization (Lab 2), embeddings (Lab 3), sequences (Lab 4)—with transformers, dominant in modern applications like chatbots or translation.

Research Time: Does full BERT outperform DistilBERT here? How do unfreezing layers affect val_loss with more data?

Effect of Changing num_epochs

Before (10 Epochs): val_loss 0.426, val_acc 0.835 (PAGE9, second doc)—solid but improvable.

After (20 Epochs): val_loss 0.303, val_acc 0.880 (PAGE11, first doc)—a 0.123 loss drop and 0.045 accuracy boost, peaking at 0.890 (epoch 11). Extended training reduced loss consistently (0.410 to 0.303), enhancing generalization, though accuracy fluctuated (0.835 to 0.890, then 0.880), suggesting diminishing returns or noise. The lower val_loss indicates better fit, but stability requires further tuning.

CONCLUSION

Lab 5's dual runs showcased transformers' prowess, with outputs proving fine-tuning's strength—val_loss falling from 0.426 to 0.303 and accuracy peaking at 0.890. The epoch increase revealed optimization limits, building my tuning instincts for future NLP endeavors.

REFERENCES

https://eagleonline.hccs.edu/courses/282423/files/70403051?module_item_id=19492478

https://eagleonline.hccs.edu/courses/282423/files/72470430?module_item_id=19492351

https://www.tensorflow.org/tfmodels/nlp/fine_tune_bert

https://www.restack.io/p/fine-tuning-answer-epochs-for-bert-cat-ai

https://discuss.huggingface.co/t/bert-fine-tuning-low-epochs/54869

https://mccormickml.com/2019/07/22/BERT-fine-tuning/

https://medium.com/towards-data-science/what-exactly-happens-when-we-fine-tune-bert-f5dc32885d76