TPF-03: Aprimoramentos do Modelo DistilBERT

Henrique Alves
Disciplina de Redes Neurais
Universidade Federal de Viçosa – Campus Florestal
Florestal, MG
henrique.a.campos@ufv.br

1 Introdução

A etapa anterior deste projeto, denominada TPF-02, consistiu na reprodução fiel dos experimentos de *fine-tuning* do modelo DistilBERT na tarefa de análise de sentimentos, utilizando o conjunto de dados SST-2. Naquela fase, estabeleceu-se uma linha de base de desempenho com uma acurácia de 90.37%, validando a eficácia do modelo destilado em manter uma performance competitiva em relação ao seu professor, o BERT, mesmo com uma arquitetura reduzida.

O presente relatório, referente à etapa TPF-03, tem como objetivo propor, implementar e avaliar modificações estruturais e procedimentais visando a otimização do modelo. O foco central desta investigação não se limita apenas à maximização da métrica de acurácia, mas também à análise crítica da eficiência computacional. Para tanto, foram introduzidas técnicas de *Transfer Learning* através do congelamento de camadas e estratégias avançadas de otimização com agendamento não linear da taxa de aprendizado. Este estudo busca compreender como tais alterações impactam a convergência do treinamento e a capacidade de generalização do modelo em cenários de recursos limitados.

2 Modificações e Justificativas

Para superar os resultados e as limitações observadas na etapa anterior, foram implementadas duas modificações significativas na configuração original do treinamento. A seguir, detalham-se as alterações e o embasamento teórico para cada escolha.

2.1 Modificação 1: Congelamento Parcial da Arquitetura (Backbone Freezing)

Na abordagem original reproduzida no TPF-02, todos os pesos do modelo, desde a camada de entrada até o classificador final, foram atualizados durante o processo de *fine-tuning*. Nesta nova etapa, optou-se por congelar os parâmetros dos *embeddings* e das quatro primeiras camadas do Transformer, correspondentes aos índices 0, 1, 2 e 3. Desta forma, apenas as duas últimas camadas, índices 4 e 5, juntamente com a cabeça de classificação, permaneceram treináveis.

A justificativa para esta abordagem reside na hierarquia de aprendizado dos modelos de linguagem baseados em Transformers. As camadas iniciais tendem a capturar características linguísticas mais gerais e fundamentais, como sintaxe e morfologia, que são amplamente transferíveis e não necessitam de ajustes drásticos para a tarefa de análise de sentimento. Ao congelar essas camadas, reduz-se significativamente o número de parâmetros sujeitos ao cálculo de gradiente, o que previne o fenômeno do esquecimento catastrófico e simula um cenário realista de implantação em dispositivos com restrição de memória e processamento.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2.2 Modificação 2: Agendador de Taxa de Aprendizado Cosseno

O agendador de taxa de aprendizado linear, utilizado anteriormente, foi substituído por um agendador do tipo *Cosine Annealing with Warmup*. A configuração incluiu uma fase de aquecimento inicial correspondente a 10% dos passos totais de treinamento.

A escolha desta técnica deve-se às limitações do decaimento linear, que pode reduzir a taxa de aprendizado de forma abrupta ou ineficiente ao se aproximar de mínimos locais complexos na superfície de erro. O agendador cosseno reduz a taxa de aprendizado seguindo uma curva suave e periódica. Isso permite que o modelo refine seus pesos de maneira mais delicada nas etapas finais do treinamento, potencializando a convergência e evitando que o otimizador fique preso em pontos de sela, uma característica comum em otimizações de redes neurais profundas.

2.3 Disponibilidade de Código e Reprodutibilidade

Para garantir a total reprodutibilidade deste estudo, todos os códigos-fonte, juntamente com os *logs* de execução e instruções detalhadas de configuração do ambiente, estão disponíveis publicamente no repositório GitHub do projeto.

O repositório pode ser acessado através do seguinte endereço:

`https://github.com/alveshenriique/redes-neurais`

3 Configuração Experimental e Resultados

O treinamento foi executado no ambiente Google Colab. Diferente da etapa anterior, onde limitações técnicas forçaram o uso de CPU, este experimento utilizou uma GPU Tesla T4, garantindo a viabilidade computacional das modificações propostas. O modelo foi treinado por 4 épocas, uma a mais que o padrão anterior, para permitir que o ciclo do agendador cosseno se completasse adequadamente. O tamanho do lote, ou *batch size*, foi mantido em 16 amostras.

3.1 Resultados Obtidos

A Tabela 1 apresenta a evolução detalhada das métricas de perda e acurácia ao longo das quatro épocas de treinamento.

Tabela 1: Histórico de treinamento do modelo aprimorado (TPF-03).

Época	Training Loss	Validation Loss	Accuracy
1	0.267900	**0.290824**	0.892202
2	0.142900	0.324195	0.888761
3	0.163000	0.382278	0.894495
4	0.168800	0.390543	0.895642

O mecanismo de parada antecipada e salvamento de modelos, configurado através do parâmetro `load_best_model_at_end`, selecionou automaticamente o modelo resultante da **Época 1**. Esta seleção ocorreu porque a primeira época apresentou a menor perda de validação, com valor de 0.2908. Consequentemente, a acurácia final reportada para o modelo otimizado foi de **89.22%**.

3.2 Comparação: Abordagem Original vs. Aprimorada

A Tabela 2 estabelece um comparativo direto entre a reprodução fiel realizada no TPF-02 e a versão modificada proposta no TPF-03, evidenciando os impactos das alterações na performance e na eficiência.

Tabela 2: Comparativo de Desempenho e Eficiência entre as etapas.

Métrica	**TPF-02 (Original)**	**TPF-03 (Modificado)**
Acurácia Final	90.37%	89.22%
Validation Loss	0.3102	0.2908
Tempo de Treinamento	$\sim$11 horas (CPU)	$\sim$9 minutos (GPU)
Parâmetros Treináveis	100% (Total)	$\sim$35% (Parcial)

4 Discussão e Conclusões

4.1 Análise do Desempenho Preditivo

A implementação do congelamento das camadas iniciais resultou em uma redução marginal na acurácia final, quantificada em 1.15 pontos percentuais. Este comportamento era teoricamente esperado. Ao impedir que as camadas inferiores do Transformer se adaptem às especificidades léxicas e sintáticas do dataset SST-2, limita-se a plasticidade total do modelo. No entanto, o fato de o modelo alcançar 89.22% de acurácia com apenas uma época de treinamento efetivo nas camadas superiores demonstra que as representações vetoriais pré-treinadas do DistilBERT são extremamente robustas e generalistas.

4.2 Eficiência Computacional e Convergência

O impacto mais expressivo das modificações foi observado na eficiência do processo. A combinação do uso de hardware acelerado com a redução do grafo de computação, devido ao congelamento de camadas, reduziu o tempo de treinamento de horas para minutos. Além disso, observa-se que a perda de validação do modelo modificado na melhor época (0.2908) foi inferior à do modelo original (0.3102). Isso sugere que o agendador cosseno auxiliou o modelo a encontrar um mínimo local de perda mais favorável nas etapas iniciais, mesmo operando com um número reduzido de parâmetros livres.

4.3 Conclusão

O experimento conduzido no TPF-03 ilustrou um clássico compromisso da engenharia de aprendizado de máquina: a troca de uma fração mínima de desempenho preditivo por ganhos substanciais em eficiência e estabilidade. A rápida degradação da perda de validação observada nas épocas 2, 3 e 4, que subiu de 0.29 para 0.39, indica que o modelo modificado tende ao sobreajuste mais rapidamente, pois as poucas camadas treináveis tentam memorizar os dados de treino de forma agressiva.

Conclui-se que a estratégia de congelamento parcial, aliada a um agendamento de taxa de aprendizado otimizado, é uma abordagem viável e recomendada para cenários onde o custo computacional é um fator crítico, mantendo a qualidade do modelo em níveis competitivos com a abordagem de treinamento completo.

Referências

Referências

[1] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108. Disponível em: `https://arxiv.org/abs/1910.01108`

[2] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.

[3] Hugging Face. *Transformers Documentation*. Disponível em: `https://huggingface.co/docs/transformers`. Acesso em: 19 nov. 2025.

[4] Loshchilov, I., & Hutter, F. (2016). *SGDR: Stochastic Gradient Descent with Warm Restarts*. arXiv preprint arXiv:1608.03983. (Referência para o Cosine Annealing).

[5] Hugging Face. *Datasets Documentation*. Disponível em: `https://huggingface.co/docs/datasets`. Acesso em: 19 nov. 2025.