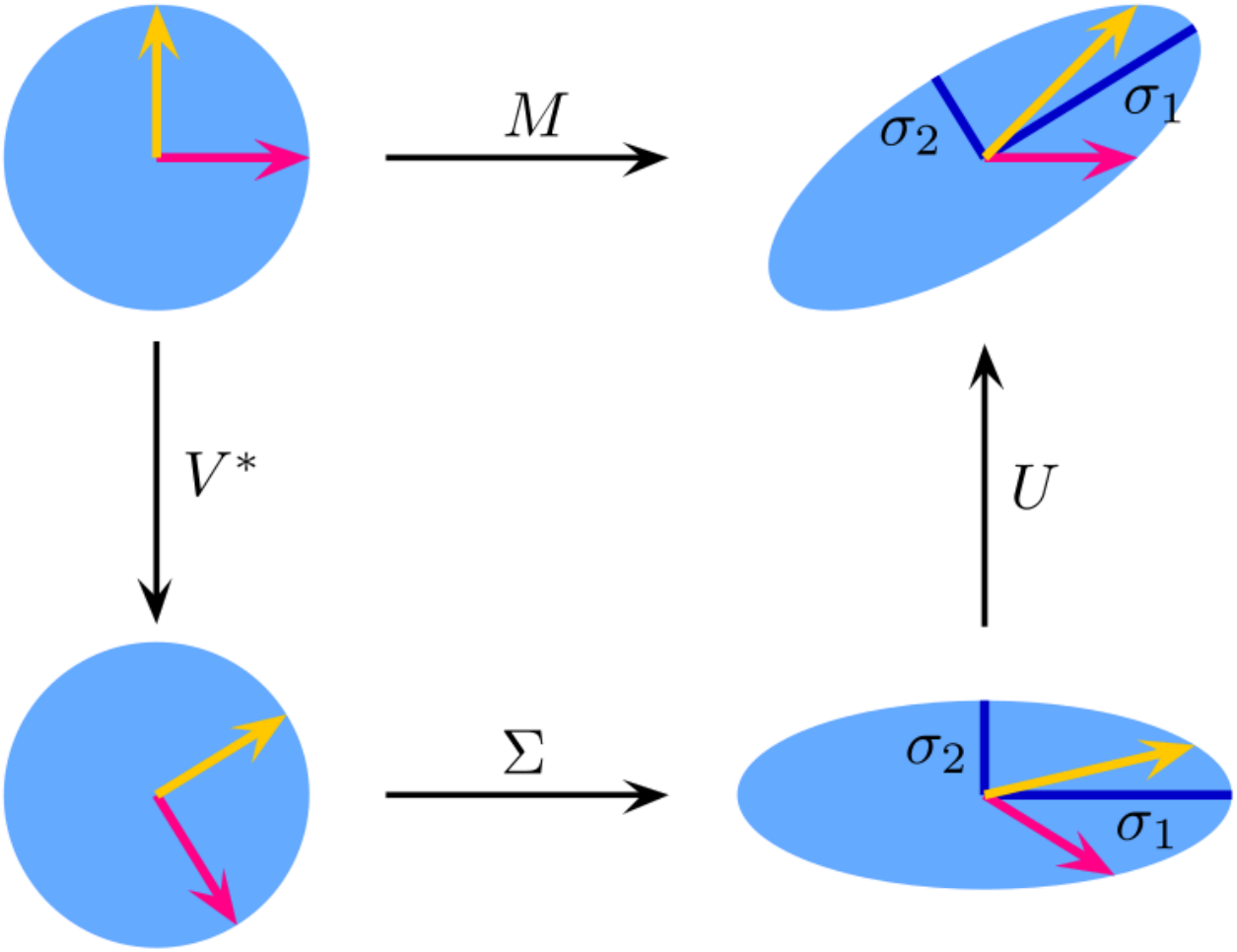
UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA



Treinamento Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

Gabriel van der Schmidt

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

GABRIEL VAN DER SCHMIDT

Treino Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): GABRIEL VAN DER SCHMIDT

Título do trabalho: Treino Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Gabriel Van Der Schmidt**, **Discente**, em 20/01/2025, às 10:46, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 20/01/2025, às 13:24, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089571** e o código CRC **D1907FC5**.

Referência: Processo nº 23070.001559/2025-95 SEI nº 5089571

GABRIEL VAN DER SCHMIDT

Treino Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2025

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

SCHMIDT, GABRIEL VAN DER
Treino Eficiente de Transformers [manuscrito] : Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque / GABRIEL VAN DER SCHMIDT. - 2025.
58 f.

Orientador: Prof. Dr. Fernando Marques Federson.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. eficiência computacional. 3. aproximação de baixo ranque. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

GABRIEL VAN DER SCHMIDT

Treino Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 17 de dezembro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Me. Lucas Araújo Pereira
(INF-UFG)

GABRIEL VAN DER SCHMIDT

Treino Eficiente de Transformers

Acelerando o Fine Tuning de Transformers com Aproximações de Baixo Ranque

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Eficiência Computacional em Transformers**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Computational Efficiency in Transformers**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

Minha Jornada

Gabriel van der Schmidt

Especialista em: Eficiência Computacional em Transformers

MINHA JORNADA

Nome: Gabriel van der Schmidt

Especialidade: Eficiência Computacional em Transformers

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha Jornada começou na **Semana 1** com pesquisas sobre a história e os fundamentos de tradução de voz, tema inicialmente escolhido para a minha especialização. Esta escolha, porém, havia sido feita visando o meu posicionamento no mercado, não necessariamente sendo algo pelo qual eu tivesse interesse genuíno. Por isso, passei a **Semana 2** procurando tópicos de interesse, a partir dos quais decidi explorar formas de otimizar Transformers. Com a **Semana 3**, veio a decisão final de abordar otimização de Transformers pelo viés de eficiência computacional, especificamente durante a fase de treino. A escolha de **Eficiência Computacional em Transformers** como tema da Minha Jornada teve forte influência das experiências vividas nas disciplinas de Aprendizado de Máquina Supervisionado e Redes Neurais Profundas. No **Apêndice 1**, encontram-se resumos e observações sobre os estudos realizados ao longo destas Semanas iniciais.

Definida a área de concentração, durante a **Semana 4**, pesquisei técnicas para o aumento da eficiência computacional de Transformers. Dentre as técnicas vistas, decidi

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.

explorar Aproximações de Baixo Ranque (Low-Rank Approximations, LRA) por ser um viés indutivo aparentemente bem alinhado com a natureza da arquitetura Transformer, apesar de sua simplicidade. O **Apêndice 2** contém os artigos relevantes para a pesquisa desta Semana. Dentre eles, gostaria de ressaltar o trabalho de Winata et al (2020)[2], com o qual eu já havia tido contato durante as Semanas iniciais, e o qual é em grande parte responsável pelas escolhas tanto da técnica de LRA quanto do tema de eficiência computacional.

Durante a **Semana 5**, fiz o levantamento de bibliotecas, frameworks, e outros recursos de software capazes de auxiliar a integração de LRA em modelos existentes, através do qual descobri que não existem bibliotecas que implementem LRA nativamente. Com isso, iniciei a **Semana 6** realizando testes preliminares para superar esta barreira inicial. Outra limitação encontrada ao longo desta Semana é referente aos recursos limitados de hardware disponíveis, dado o grande custo computacional do treino de Transformers. Mais detalhes sobre o trabalho desenvolvido ao longo destas Semanas se encontram no **Apêndice 3**.

Continuei com esses testes preliminares ao longo da **Semana 7**, durante a qual eu também busquei tratar a limitação de hardware encontrada. A solução encontrada foi optar pelo fine-tuning (ajuste fino) de modelos pré-treinados, ao invés de fazer o pré-treino completo dos modelos. Isso permitiu reduzir massivamente o custo computacional dos testes seguintes, mas mantendo a generalidade dos resultados obtidos. Assim, na **Semana 8**, foram definidos os objetivos e os métodos dos experimentos a serem realizados. Notavelmente, foram incluídas duas tarefas de Processamento de Linguagem Natural (Natural Language Processing, NLP) e duas tarefas de Visão Computacional (Computer Vision, CV) para demonstrar a independência de domínio da técnica empregada. O planejamento dos experimentos encontra-se no **Apêndice 4**.

As **Semanas 9 e 10** foram dedicadas à execução dos experimentos. O código utilizado, bem como os resultados obtidos, se encontram no **Apêndice 5**. Devido a problemas com o ambiente de execução e com a implementação dos fine-tunings, tive que

[2] Winata et al. **Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer**. 2020. Disponível em <https://arxiv.org/abs/1910.13923>

reduzir o escopo a apenas uma tarefa de NLP e uma tarefa de CV. Porém, ambas as tarefas testadas apresentaram bons resultados: foi possível reduzir o tempo de fine-tuning de 7% a 8% com menos de 2% de perda de performance quando comparado ao modelo sem alterações. Esses resultados são bastante promissores, pois demonstram que LRA não apenas aumenta eficiência computacional do modelo com pouco impacto na performance, mas que também pode ser aplicada a qualquer arquitetura Transformer independentemente do domínio de aplicação. A literatura também sugere que ganhos similares podem ser atingidos no pré-treino de modelos, e que LRA não é mutuamente exclusiva com outras técnicas de otimização de Transformers.

Ao longo de Minha Jornada, tive a oportunidade de viver o que creio ser a essência de Ciência de Dados: a descoberta de estruturas e princípios intrínsecos a sistemas complexos. Mesmo Transformers, famosos por sua habilidade de trocarem vieses indutivos explícitos por estrutura derivada de dados, podem se beneficiar de hipóteses seletas – não sobre o domínio dos dados, mas sobre a sua própria arquitetura. Estudar e refletir sobre as hipóteses da arquitetura Transformers ao longo da Residência foi uma experiência de aprendizado significativa, pois o questionamento de modelos estabelecidos é com frequência um catalisador de inovação no campo de Inteligência Artificial.

APÊNDICE 1

Termo de Aceite de Entrega (Semana 1)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- **Brainstorming preliminar com ChatGPT** (Brainstorming com ChatGPT), com revisão de conceitos conhecidos e a descoberta de novos: Dynamic inference, destilação de conhecimento vs. treino de modelo menor, **ESPnet**, etc.
- **Estudo de reviews da área de speech translation** de diferentes épocas Reviews de Speech Translation :
  - **Survey of Current Speech Translation Research** (2003): Abordagens muito reduzidas em escopo; pipelines altamente customizados para certa abordagem, sem interoperabilidade e com alto viés indutivo; recorrência de modelos baseados em exemplos, máquinas de estado finito e HMMs.
  - **Speech-to-Speech Translation: A Review** (2015): Cita apenas duas aplicações já abordadas no artigo anterior: MASTOR da IBM e VERBMOBIL. Certamente não é representativo do estado da arte da época.
  - **Recent Advances in Direct Speech-to-text Translation** (2023): Transformers dominam o estado-da-arte, e permitiram o surgimento de abordagens ponta-a-ponta (E2E) não vistas anteriormente. Focam em três principais problemas atuais:
    - Modeling burden: dificuldade de abordagens E2E em simultaneamente modelar mapeamentos intermodais e interlinguais.
    - Escassez de dados: Data augmentation e pseudo-anotação.
    - Problemas de aplicação: Tempo-real, tradução de entidades nomeadas, code-changing, etc.
- **Leitura** do capítulo de Introdução (2) de Introduction to Speech Processing
- **Encontro com o professor Arlindo para discussão de uma ideia de projeto final.** Principal takeaway: Para que o projeto consiga passar por um processo com começo, meio e fim dentro da residência, há duas abordagens viáveis:
  - Focar nos conceitos/ferramentas e buscar um escopo pequeno onde intervenções ou contribuições podem ser feitas.
  - Focar na aplicação final desejada e buscar ferramentas/modelos prontos para resolver o problema, fazendo pequenas adaptações para adequá-lo aos seus requisitos específicos.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Continuar com o estudo dos livros encontrados e de artigos da área, especificamente:
  - Continuação da leitura de Introduction to Speech Processing, e
  - Início da leitura de Speech and Language Processing.
- Estudar o toolkit ESPnet e tentar realizar experimentos preliminares com ele.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Eu ainda não tomei uma decisão entre as duas "abordagens viáveis". Portanto, penso que uma boa alternativa por enquanto seja "criar oportunidades" para ambas e ver qual mais naturalmente se encaixa comigo e com o propósito da residência. Por isso os objetivos de leitura (busca de gaps de pesquisa) e de experimentação (viabilizar a solução pensada).

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega (Semana 2)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 25 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

1) Exploração de artigos nas áreas de **low-rank matrix fatorization** para DNNs, **dynamic inference** em modelos de língua/fala, e **code-switching** em ASR/ST (Resumo de Artigos):

Low-rank matrix factorization:

- **LOW-RANK MATRIX FACTORIZATION FOR DEEP NEURAL NETWORK TRAINING WITH HIGH-DIMENSIONAL OUTPUT TARGETS (2013):** Treinam uma DNN com matrizes de rank baixo na camada final, conseguem alta redução no número de parâmetros e tempo de treinamento total do modelo.
- **Language model compression with weighted low-rank factorization (2022):** Aplicam a fatoração de matrizes após o modelo ser treinado para acelerar a inferência. Alternativa à destilação de conhecimento.
- **Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer (2020):** Utilizam a fatoração dos pesos de treinamento como o primeiro artigo, mas aplicado a transformers. Conseguem reduções de tamanho e de tempo de treinamento similares.

Dynamic inference:

- **A flexible BERT model enabling width- and depth-dynamic inference (2024):** Aplicam inferência dinâmica em profundidade (early exit off-ramps) e em largura (subredes de diferentes larguras).

Code-switching:

- **CODE-SWITCHING WITHOUT SWITCHING: LANGUAGE AGNOSTIC END-TO-END SPEECH TRANSLATION (2022):** Propõem um E2E para lidar com ST contendo CS. A abordagem se baseia principalmente na combinação de datasets de ASR/ST e data augmentation.
- **Data Augmentation for End-to-end Code-switching Speech Recognition (2020):** As principais técnicas propostas são: 1) audio splicing; 2) word translation; 3) word insertion; e 4) a combinação das técnicas anteriores junto com SpecAugment.

2) Leitura parcial do **Capítulo 3 (Basic Representations) de** Introduction to Speech Processing**.**

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Exploração mais aprofundada de técnicas de compressão de modelos/aceleração de treinamento como low-rank matrix factorization.
- Alinhamento com os professores (concordância com a disciplina, viabilidade técnica, etc).

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

LEONARDO ALVES: Go!

Termo de Aceite de Entrega (Semana 3)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

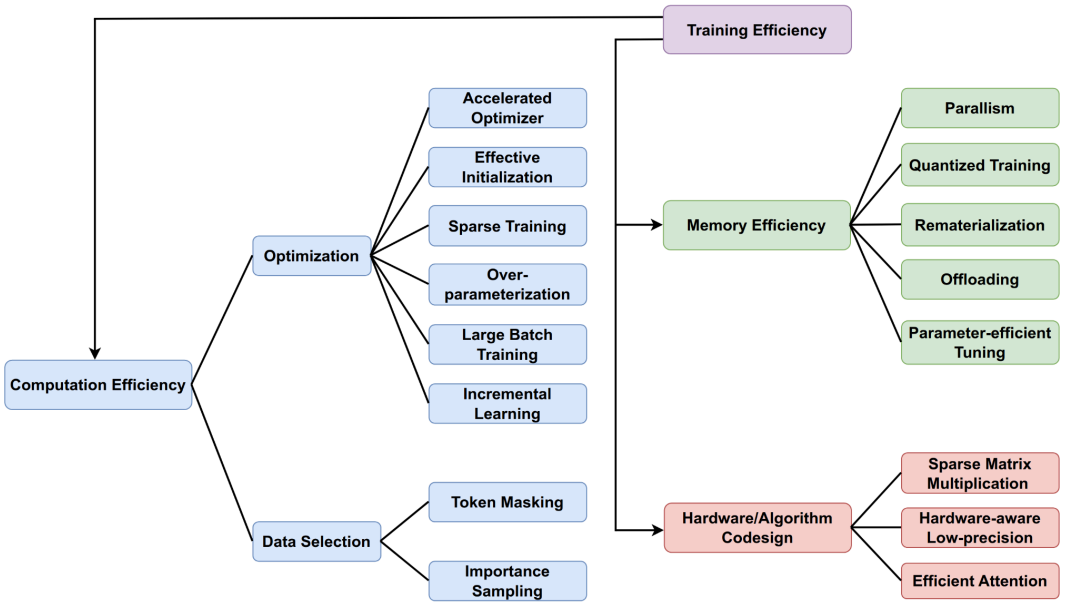
Data da Reunião ("gate") de aprovação: 3 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

1) Leitura do artigo [2302.01107] A Survey on Efficient Training of Transformers **(2023)**:



2) Leitura do artigo [2402.05964v1] A Survey on Transformer Compression **(2024)**:

Conceito	Técnica	Ferramenta
Quantization	Post-training Quant., Quant. Aware Training	GitHub - pytorch/ao
Knowledge Distillation	Logits-based KD, Hint-based KD,	GitHub - SforAiDl/KD_Lib

	API-based KD	GitHub - knowledge-distillation-pytorch
Pruning	Structured/Context Pruning	GitHub - VainF/Torch-Pruning
Efficient Architecture	Transformer Variants, Non-transformer Architectures	GitHub - Jamie-Stirling/RetNet

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Aprofundar a pesquisa acerca dos frameworks disponíveis para as técnicas levantadas, avaliando a sua usabilidade.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Brainstorming com ChatGPT.doc citado no Termo de Aceite de Entrega de 18 de setembro]

Brainstorming com ChatGPT

Gabriel van der Schmidt

A conversa completa está aqui. Segue um resumo dos pontos que eu achei mais interessantes/significativos desse chat.

Takeaway points:

- A maioria das abordagens modernas de speech translation utilizam uma dentre três abordagens principais:
    - Modelos cascata, onde o modelo ponta-a-ponta é formado por modelos treinados em tarefas específicas (audio speech recognition, machine translation, text-to-speech, e voice conversion). Vantagens incluem a facilidade de implementação com o uso de modelos pré-treinados e a possibilidade de trocar componentes por outros modelos. Porém, a sua principal desvantagem é a propagação de erros de um modelo para as etapas seguintes (por exemplo, um erro na transcrição leva a um erro de tradução, e assim por diante).
    - Modelos end-to-end usam um único modelo para realizar todas as tarefas no processo de tradução voz-para-voz, potencialmente acelerando o tempo de inferência e minimizando o problema de propagação de erros. Em contrapartida, há a perda da modularidade e interpretabilidade dos estágios do modelo.
    - Modelos multimodais, que combinam características de áudio e texto para refinar a qualidade da tradução. Porém, o LLM falha em citar a complexidade extra na arquitetura e no treinamento de tais modelos.
- Low-Rank Factorization aplicado a modelos de fala. O LLM confunde sua definição com LoRA, que embora partindo do mesmo princípio, é levemente diferente em sua implementação. A curiosidade me levou a pesquisar mais sobre o conceito:
    - Sainath et al (2013) utilizam a mesma hipótese de matrizes de pesos terem ranque baixo como em LoRA, mas ao invés de utilizar os adaptadores apenas no fine tuning (mantendo os pesos originais da rede congelados), neste trabalho eles são parte integral da arquitetura, substituindo as grandes matrizes

de pesos que compunham as camadas finais em redes neurais profundas da época.
    - Hsu et al (2022) utilizam a fatoração das matrizes de pesos para compressão de modelos, acelerando o tempo de inferência. Os autores também propõem uma variação do algoritmo de fatoração (SVD) para melhor preservar a performance do modelo não comprimido nas tarefas alvo, conseguindo reduzir o número de parâmetros até de modelos compactos com pouca perda de desempenho.
- Dynamic inference é um termo que eu desconhecia, e se refere ao processo de "pular" camadas ou realizar early stopping durante a inferência com base no nível de confiança no modelo em sua resposta preliminar, podendo acelerar a inferência conforme a complexidade dos dados de entrada. Pesquisando sobre o assunto, li partes do trabalho de Hu, Meinel e Yang (2024) , que usam classificadores intermediários para implementar o early stopping. Este trabalho implementa também inferência dinâmica em largura, mas o funcionamento exato deste mecanismo ainda não me é claro.
- Destilação de conhecimento como forma de redução da latência de inferência. O LLM fez um bom trabalho em explicar as diferentes vantagens e desvantagens de destilação versus o treino de um modelo compacto com a ausência do modelo-professor maior, que são:
    - Realizar o treino de um modelo menor diretamente sobre os dados resulta em um pipeline mais simples e mais rápido de treinar, porém correndo o risco deste modelo não ser capaz de encontrar certos padrões complexos que um modelo maior seria capaz de modelar. Indicado quando há escassez de recursos ou de tempo, ou quando um modelo menor já adequadamente captura a complexidade dos dados.
    - A destilação de conhecimento, por sua vez, facilita o trabalho de aprendizado do modelo menor ao usar as probabilidades de classes preditas pelo modelo maior (soft labels) como alvos, pois estas revelam relações entre classes descobertas pelo modelo maior com maior capacidade de representação. Isso, porém, vem com o custo adicional de treinar primeiro o modelo-professor, sem haver a garantia de que o modelo-aluno terá desempenho comparável ao professor. Útil para reduzir a latência e os requisitos de hardware necessários para a inferência, mas ainda preservando o máximo de acurácia do modelo maior.
- ESPnet, citado pelo LLM, é um toolkit cobrindo várias tarefas dentro de processamento de voz, possibilitando a construção tanto de pipelines cascata quanto

de ponta-a-ponta. Parece ser uma ferramenta interessante com a qual quero experimentar no futuro.

[Reviews de Speech Translation.doc citado no Termo de Aceite de Entrega de 18 de setembro]

Reviews de Speech Translation

Gabriel van der Schmidt

Para entender melhor o contexto e a evolução de tradução de fala, estudei três reviews, cada um de uma década diferente – 2003, 2015, e 2023. Esses artigos funcionam como cápsulas do tempo, preservando a memória do ápice do avanço tecnológico da época nesta tarefa. Seguem resumos contendo os pontos principais do artigos estudados:

Survey of Current Speech Translation Research† (2003)

Na época, não haviam aplicações comerciais de tradução de fala, os poucos exemplos existentes vindo principalmente de programas com fomento governamental.

Os exemplos existentes também eram bastante limitados em escopo, geralmente limitados a um certo domínio ou até a conjunto específico de frases (por exemplo, *Phraselator*, que possuía uma tabela de consulta de frases já traduzidas). Essa redução de escopo era necessária devido à baixa capacidade de aprendizado dos modelos da época.

As abordagens variam muito de uma para outra, sendo visível a grande quantidade de viés indutivo utilizado e a alta customização de cada pipeline, embora a sequência geral de passos ASR→MT→TTS já era vista numa escala macro do modelo (afinal, cada um desses passos era, ele próprio, composto por diversos subcomponentes). Outra semelhança encontrada é a recorrência de modelos baseados em exemplos, máquinas de estado finito e HMMs.

Por fim, apesar das imensas limitações de hardware da época, é interessante ver que já haviam esforços para gerar modelos genéricos capazes de facilmente adicionar novas línguas ao seu vocabulário de tradução, geralmente utilizando interlíngua como um meio-termo para unificar as diferentes línguas traduzidas.

Speech-to-Speech Translation: A Review (2015)

Este é um review significativamente mais curto que os demais, tratando apenas de duas aplicações que já haviam sido citadas no review anterior: MASTOR da IBM e VERBMOBIL. Esses modelos certamente não eram mais o estado-da-arte da época, afinal nesse meio tempo aplicações comerciais de tradução de voz já haviam sido desenvolvidas, notavelmente Google Tradutor.

[2306.11646] Recent Advances in Direct Speech-to-text Translation (2023)

Neste review, o estado da arte é dominado por transformers e suas variantes. Pela primeira vez também se vê abordagens ponta-a-ponta (E2E), ao invés do pipelines extremamente fragmentados do primeiro review. Mesmo abordagens que ainda usam arquiteturas cascata são mais interoperáveis devido ao uso ubíquo de transformers: modelos de ASR e MT podem ser tratados como partes independentes e substituídos conforme necessário com facilidade.

Muitos desafios ainda permanecem na área. Este artigo levanta três principais: modeling burden; escassez de dados; e problemas de aplicação.

Modeling burden se refere à dificuldade de abordagens E2E em simultaneamente modelar mapeamentos intermodais e interlinguais. Várias variações de transformers foram propostas para suavizar este problema, incluindo o uso de aprendizado auto-supervisionado e modelos multi-tarefa. Outro problema relacionado é o uso de modelagem não auto-regressiva em arquiteturas cascata, o que removeu a vantagem em latência de inferência que modelos E2E tinha sobre estas.

Escassez de dados pode ser mitigada aumentando a quantidade de dados utilizada ou encontrando formas de extrair mais conhecimento dos dados existentes.

O aumento de dados pode ser feito usando técnicas comuns de data augmentation (especialmente SpecAugment), mas também utilizando modelos treinados para realizar pseudo-anotação, como o uso de modelos de ML para transformar datasets de ASR em SL, ou o uso de TTS para expandir a quantidade de áudio disponível.

Para se extrair mais conhecimento do dados disponíveis, técnicas como o design de estratégias de pré-treino para aumentar a robustez e generalidade dos modelos treinados; o uso de destilação de conhecimento para a geração de representações de conhecimento mais compactas; e o uso de treino multilingue para incentivar o reuso de estruturas comuns a diversas línguas, aumentando a robustez do sistema especialmente em línguas com poucos dados.

Por fim, problemas específicos da aplicação prática de ST incluem requisitos de tradução em tempo-real; a necessidade de segmentação de fala devido à incapacidade de modelos atuais lidarem bem com trechos longos de áudio; o reconhecimento de entidades nomeadas e como traduzi-las; mudança de código, onde falantes intercalam trechos de línguas diferentes, especialmente em fala espontânea; e vieses de gênero presentes nos dados de entrada.

Algumas das tendências futuras apontadas pelos autores incluem a exploração do uso da capacidade generativa de LLMs em tarefas de ST, e a popularização de soluções

multimodais, permitindo a extrapolação para outras tarefas como tradução áudio para áudio e tradução de vídeos.

[Resumo de Artigos.doc citado no Termo de Aceite de Entrega de 25 de setembro]

Resumos de Artigos

Gabriel van der Schmidt

Seguindo as orientações da semana passada, comecei a afunilar o meu escopo de interesse para a Residência. Seguindo as minhas pesquisas anteriores, decidi explorar três técnicas/problemas específicos da área de processamento de voz que me pareceram mais interessantes: 1) fatoração de matrizes de rank baixo para acelerar o tempo de inferência e/ou de treinamento; 2) inferência dinâmica para acelerar o tempo de inferência; e 3) o problema de code-switching em tarefas de ASR e ML.

Low-rank Matrix Factorization:

- **LOW-RANK MATRIX FACTORIZATION FOR DEEP NEURAL NETWORK TRAINING WITH HIGH-DIMENSIONAL OUTPUT TARGETS (2013):** Assumem a hipótese de matrizes de pesos terem ranque baixo, como em LoRA, mas ao invés de utilizar os adaptadores apenas no fine tuning (mantendo os pesos originais da rede congelados), neste trabalho eles são parte integral da arquitetura, substituindo as grandes matrizes de pesos que compunham as camadas finais em redes neurais profundas da época. Os experimentos dos autores mostram que essa hipótese só é válida para as camadas finais do modelo, pois a fatorização de camadas escondidas levou a perdas consideráveis de desempenho.
- **[2207.00112] Language model compression with weighted low-rank factorization (2022):** Utilizam a fatoração das matrizes de pesos para compressão de modelos apenas após o modelo ser treinado, para acelerar o tempo de inferência. Os autores também propõem uma variação do algoritmo de fatoração (Fisher-Weighted SVD) que otimiza as matrizes de baixo rank segundo seu impacto na métrica alvo, ao invés de somente considerar o erro de reconstrução da matriz de pesos original. Isso melhor preserva a performance do modelo comprimido nas tarefas alvo, conseguindo reduzir o número de parâmetros até de modelos compactos com pouca perda de desempenho.
- **[1910.13923] Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer (2020):** Implementam a técnica de treinamento com fatoração de matrizes para transformers, aplicando essa fatoração em todas as projeções lineares do mecanismo de auto-atenção. Eles interpretam a multiplicação sucessiva das duas matrizes de baixo rank como um processo de encoding-decoding

para um espaço latente comprimido, tal qual acontece em autoencoders. Seguindo essa abordagem, os autores conseguem reduzir significativamente a quantidade de parâmetros do modelo treinado e consequentemente o seu tempo de treinamento.

Dynamic Inference:

- **A flexible BERT model enabling width- and depth-dynamic inference (2024):** Introduzem o conceito de neural grafting, que tem o objetivo de reativar as cabeças de atenção inativas em tarefas downstream. Para inferência dinâmica em profundidade, eles utilizam cabeças de classificação auxiliares em pontos diferentes do modelo, podendo ser ativadas ou não a depender de sua confiança na classificação. A inferência dinâmica em largura, por sua vez, é feita ao se treinar sub redes com diferentes quantidades de cabeças de atenção (= largura) e durante a inferência determinar qual a largura necessária para lidar com a entrada obtida.

Code-Switching (CS):

- **CODE-SWITCHING WITHOUT SWITCHING: LANGUAGE AGNOSTIC END-TO-END SPEECH TRANSLATION (2022):** Os autores focam em inter-sentença code-switching (onde a mudança de código ocorre a nível de frases completas e não de palavras individuais). Trabalhos anteriores utilizaram diversas abordagens diferentes, incluindo a separação do modelo acústico e do modelo de linguagem (este último treinado com texto contendo code-switching artificialmente introduzido), o uso de modelos multi-tarefa para a detecção da língua de cada token além do reconhecimento texto falado, ou o uso de dados contendo transcrições, traduções, e anotações da língua de origem de cada token. Este trabalho, por sua vez, trata tarefas de ASR bilíngue e ST como uma única tarefa de ST, combinando datasets de ambas tarefas para o treino de um modelo E2E de tradução inglês/alemão (com CS) para alemão. Eles também utilizam como estratégia de data augmentation a concatenação de frases de línguas diferentes como entrada no modelo. Eles obtêm um modelo com maior acurácia e menor latência do que abordagens tradicionais com o uso identificação de língua seguido de modelos de linguagem distintos, porém não qualitativamente observaram melhorias em casos de CS intra-sentença.
- **[2011.02160] Data Augmentation for End-to-end Code-switching Speech Recognition (2020):** Visto que datasets contendo CS são limitados em tamanho e quantidade de línguas cobertas, os autores propõem técnicas de data augmentation para compensar pela falta de dados, todas mostrando potencial para a melhoria do desempenho de CS-ASR. Eles propõem o uso de:

- 1) **audio splicing** de dados com CS, onde palavras/trechos de CS são identificados, recortados, e então substituídos em outras falas contendo CS, assim gerando diferentes variações de uma mesma fala onde o conteúdo dos trechos contendo CS são variados;
- 2) **tradução de palavras**, onde palavras de uma transcrição monolíngue são selecionadas aleatoriamente e individualmente traduzidas, o texto alterado sendo então processado por um modelo de TTS;
- 3) **inserção de palavras**, onde palavras de outra língua são aleatoriamente introduzidas em uma transcrição monolíngue e então processadas por um modelo de TTS; e
- 4) **a combinação** das técnicas anteriores, incluindo o uso de SpecAugment, obtendo os melhores resultados dentre os testes realizados.

APÊNDICE 2

Termo de Aceite de Entrega (Semana 4)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Afunilamento do escopo de pesquisa (Levantamento de Artigos - Arquiteturas Eficientes):

- **Transformers eficientes:**
  - Data Efficiency;
  - Memory Efficiency;
  - **Computational Efficiency:**
    - Hiperparametrização eficaz;
    - Otimizadores eficientes;
    - **Arquiteturas eficientes:**
      - **1º – Low-rank approximation (LRA)**
      - 2º – Sparse training
      - 3º – MatMul-Free attention
    - etc;

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- **Levantar frameworks** que permitam/facilitem a **aplicação de técnicas de LRA** em modelos existentes, e **avaliar a sua usabilidade** (features, documentação, ease-of-use, etc).
- **Preparação** para experimentação, ex.:
  - O quão fácil/difícil é fazer intervenções em modelos pré-treinados?
  - O quão fácil/difícil é criar variações de modelos pré-treinados?
  - O quão fácil/difícil é integrar modelos existentes com os frameworks levantados?

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

[Levantamento de Artigos - Arquiteturas Eficientes.doc citado no Termo de Aceite de Entrega de 09 de outubro]

Levantamento de Artigos - Arquiteturas Eficientes

Levantamento de artigos propondo alterações na arquitetura base de Transformers, focando em melhorias na eficiência computacional do treinamento (maior prioridade) e da inferência (menor prioridade).

Resumo do Levantamento

Artigo	Área	Proposta	Ganhos	Generalizável?
[1]	NLP/QA	Implementação de depth- e width-wise dynamic inference	CE (inf.)	Talvez
[2]	NLP-CV	Inicializar modelos maiores a partir de modelos menores pré-treinados	CE (train), DE	Sim
[3]	NLP/LM	Nova arquitetura baseada em mecanismos de retenção	CE, ME, DE	Talvez
[4]	ASP/ASR	Substituição de projeções lineares dentro de MHA por encoder/decoder de rank baixo	CE, ME	Sim
[5]	CV	Redução adaptativa do rank das matrizes de pesos ao longo do treinamento	ME	Sim
[6]	CV	Enquadramento da seleção de rank para LRA como um problema de 1-shot NAS	CE, ME	Sim, sob certas circunstâncias

CE = Computational Efficiency
ME = Memory Efficiency
DE = Data Efficiency

Natural Language Processing

1) A flexible BERT model enabling width- and depth-dynamic inference (2024) – NLP-QA – 2 citações Google Scholar

Intervenções: Implementam dynamic inference em largura e profundidade com seleção (gating) de subredes e early-exit off-ramps, respectivamente.

Resultados: Conseguem aumento no F1-score em downstream tasks com redução de 45% da computação média necessária para inferência.

Ganhos: CE (inferência).

Generalizável? Possivelmente, mas com adaptações.

Possíveis interferências: N/A

2) [2303.00980] Learning to Grow Pretrained Models for Efficient Transformer Training (2023) – NLP/CV – 53 citações Google Scholar

Intervenções: Propõem a inicialização de pesos de modelos grandes a partir de modelos menores pré-treinados, expandindo o modelo existente em profundidade e em largura.

Resultados: Conseguem redução de até 50% do tempo de treinamento em modelos de NLP e CV sem perda de performance quando comparada .

Ganhos: CE (treino) e DE.

Generalizável? Sim (drop-in replacement).

Possíveis interferências: Testar os limites desta técnica em termos de quão grande pode ser o salto entre o modelo existente e o inicializado.

3) [2307.08621] Retentive Network: A Successor to Transformer for Large Language Models (2023) – NLP – 261 citações Google Scholar

Intervenções: Nova arquitetura baseada em mecanismos de retenção, uma variação da auto-atenção de transformers.

Resultados: Obtêm ganhos significativos nos tempos de treino e inferência e no uso de memória. Conseguem também maior escalabilidade que transformers em sequências muito longas.

Ganhos: CE, ME, DE.

Generalizável? Possivelmente, observando a necessidade de adaptações similares a transformers em outros domínios.

Possíveis interferências: Testar se métodos de otimização de transformers podem ser adaptados para esta arquitetura, dadas as similaridades entre os mecanismos de auto-atenção e retenção.

Audio and Speech Processing

4) [1910.13923] Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer (2020) – ASP-ASR – 83 citações Google Scholar

Intervenções: Propõem a substituição das camadas de projeção linear $m \times n$ dentro das cabeças de atenção por duas camadas $m \times r$ e $r \times n$ consecutivas, interpretadas como um processo de encoding e decoding de/para um espaço latente de menor dimensionalidade.

Resultados: Conseguem speedup de até 1.35x durante inferência e 1.10x durante treinamento sem perda de desempenho mesmo com taxas de compressão beirando os 60%.

Ganhos: CE, ME (treino e inferência).

Generalizável? Sim (drop-in replacement)

Possíveis interferências: Aplicar o processo de low-rank encoding/decoding no cálculo da matriz de atenção (encoding antes da multiplicação de K, V, e Q; decoding após), já que aí é onde está boa parte da complexidade computacional ($O(n^2)$) do treino/inf. de MHA.

Computer Vision

5) [2401.08505] Harnessing Orthogonality to Train Low-Rank Neural Networks (2024) – CV – 1 citação Google Scholar

Intervenções: Propõem a exploração da ortogonalidade das bases das matrizes de pesos para adaptativamente reduzir o rank dessas matrizes durante o treinamento. É uma alteração no ciclo de treinamento que causa alterações na arquitetura final.

Resultados: Conseguem reduções significativas na quantidade total de parâmetros da rede (entre 20% e 50%), mas o tempo de treino é apenas levemente menor do que o baseline.

Ganhos: ME, e CE (pouco durante treino).

Generalizável? Sim (drop-in replacement, mas potencialmente incompatível com outros ciclos de treino).

Possíveis interferências: Aplicar as métricas utilizadas para adaptativamente reduzir o rank das matrizes de pesos com outras técnicas de low-rank factorization que resultam em ganhos mais significativos em eficiência computacional.

6) FLORA: Fine-Grained Low-Rank Architecture Search for Vision Transformer (2024) – CV – 5 citações Google Scholar

Intervenções: Enquadramento da seleção de rank para low-rank approximation como um problema de 1-shot neural architecture search. Isso é feito através do treinamento e seleção de sub-redes candidatas com diferentes ranks.

Resultados: Obtêm reduções significativas (45%~55%) na quantidade de FLOPS necessários para inferência.

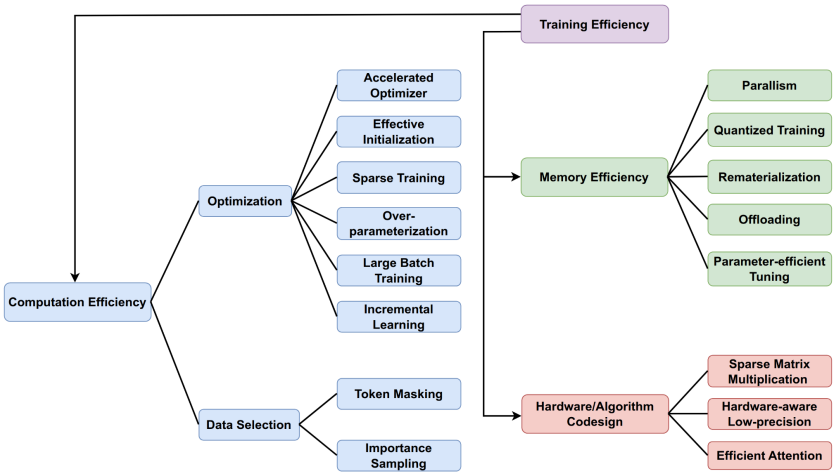
Ganhos: ME e CE (inf.).

Generalizável? Sim, observando a necessidade do treinamento de sub/super-rede(s).

Possíveis interferências: Buscar mecanismos de inicialização dos candidatos, por exemplo, encontrando bons ranks iniciais para o começo do treino.

Reviews

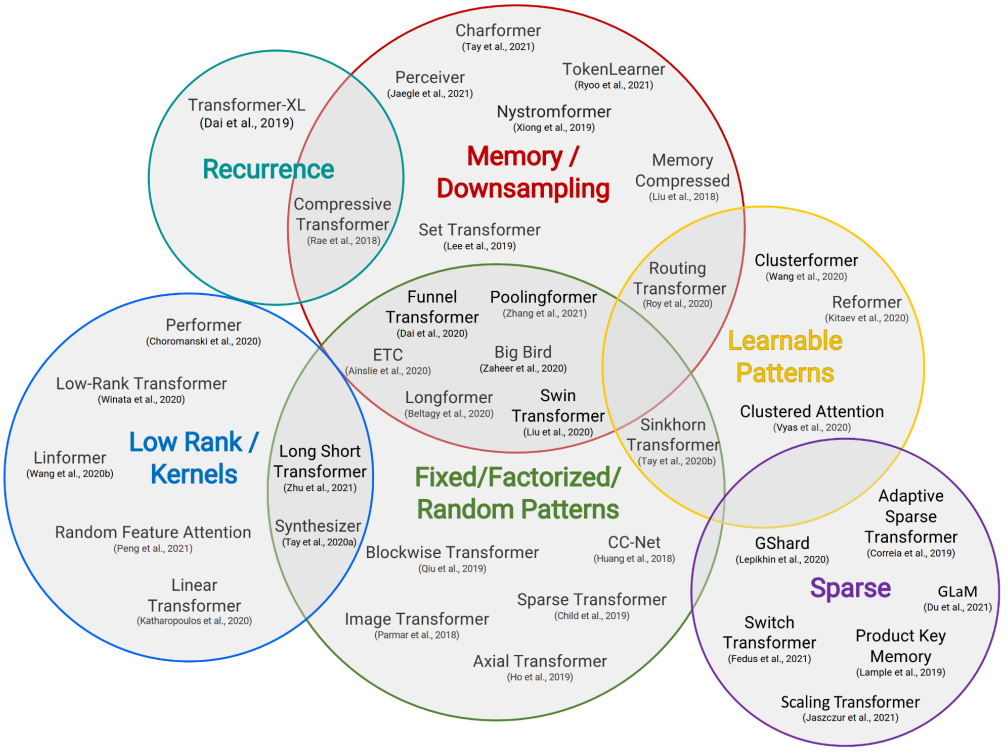
[2302.01107] A Survey on Efficient Training of Transformers



[2402.05964v1] A Survey on Transformer Compression

Category	Sub-category	Method
Quantization	NLP	SmoothQuant [18] OmniQuant [19] QLoRA [20]
	CV	PTQ-ViT [21] FQ-ViT [22] OFQ [23]
Knowledge Distillation	NLP	DistilBERT [24] MiniLM [25] Lion [7]
	CV	DeiT [26] TinyViT [27] ManifoldKD [28]
Pruning	NLP	LLM Pruner [29] Sheared LLaMA [30] Dynamic Context Pruning [31]
	CV	ViT-Slim [32] Patch Sliming [33] X-pruner [34]
Efficient Architecture	NLP	PaLM [35] RetNet [36] Reformer [37]
	CV	Swin [38] MetaFormer [39] MLP-Mixer [40]

[2009.06732] Efficient Transformers: A Survey



APÊNDICE 3

Termo de Aceite de Entrega (Semana 5)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 17 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Levantamento de Ferramentas de Otimização

Não existem frameworks para LRA durante treinamento (a maioria foca em quantização, LoRA, treinamento esparso, etc).

Das ferramentas levantadas, os experimentos terão que ser feitos com:
- **Hugging Face** para API de treinamento e modelos pré-treinados;
- **PyTorch** vanilla para modificações estruturais nos modelos; e
- (Potencialmente) **TorchAO** como referência para substituição de camadas.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejamento de Experimentos:
- **Estágio 1:** Reproduzir a arquitetura de Winata et al (low-rank "autoencoders") e realizar testes preliminares com modelos de CV, NLP e ASP.
- **Estágio 2:** Buscar métricas que possam guiar a escolha do ranque das aproximações:
  - Medidas de esparsidade, ou da contribuição de pesos na tarefa final (Fisher, gradientes, etc).

Busca por Recursos para Experimentos subsequentes:
- CEIA e Parque Tecnológico; ou
- AWS, Google Cloud, etc.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Termo de Aceite de Entrega (Semana 6)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 30 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Experimentos RES - LRA tests.ipynb **:**

- **Estágio 1:**
  - **Substituição de camadas** lineares por low-rank encoder-decoder (**BERT**, **ViT**). **Sucesso!**
  - Teste com recipes de **pré-treino/fine-tuning**. **WIP.**
    - Problema com download de datasets.
  - Teste com **transformers pequenos** (9M parâmetros) para xadrez google-deepmind/searchless_chess. **Fail**.
    - Problema para executar a ferramenta.
    - Dataset muito grande (>1TB).
    - Implementado com JAX/Haiku, e não PyTorch/HF.
- **Estágio 2 (WIP):**
  - Encontrada **função para decomposição** de matrizes com SVD, com teste preliminar para **substituição de pesos** com o seu resultado.
  - Pesquisa preliminar de **métricas de esparsidade/rank** de matrizes. **Poucos resultados**:
    - Esparsidade (espera zeros, não valores pequenos),
    - Singular values do SVD, ou explained variance do PCA.
    - Fast methods for estimating the Numerical rank of large matrices

Busca por Recursos Computacionais:

- **Parque Tecnológico:** 1 Tesla K40 e 1 Tesla V100.
- **Google Cloud:** Apenas para professores e (pós) doutorandos.
- **AWS:** Até **US$ 5.000** em créditos, a depender de aprovação.
- **Microsoft Azure:** Aparentemente restrito apenas aos EUA.
- **Open Datacenter:** Aguardando resposta.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Experimentos:

- **Estágio 1: Rodar as recipes** de pré-treino/fine-tuning para **validar as modificações** feitas.
- **Estágio 2:** Usar **singular values** para **estimar rank** de camadas em modelos pré-treinados.

- **Estágio 3:** Buscar **formas alternativas** de validar as intervenções nos transformers com os times de NLP/ASP**.**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

LEONARDO ALVES: Em análise!

[Levantamento de Ferramentas de Otimização.doc citado no Termo de Aceite de Entrega de 17 de outubro]

Levantamento de Ferramentas de Otimização de Arquiteturas Transformers

Lista de features, vantagens, limitações, etc., focando em aplicações para LRA.

Do material que pesquisei, encontrei várias bibliotecas para quantização e LoRA, algumas para treinamento esparso, e nenhuma com suporte nativo para LRA. Seguem algumas das bibliotecas mais promissoras para o projeto.

Torch Architecture Optimization

Features: Quantização (pós-treino, q. aware training) [**ME**], treinamento esparso (substituição de camadas lineares) [**ME**, CE], otimizadores eficientes (variações quantizadas de AdamW) [**ME**], integração com Hugging Face.

Possibilidades/Vantagens de Uso: TorchAO não possui ferramentas para o uso de LRA nativamente, mas eu creio que o código de substituição de camadas para treinamento esparso pode ser aproveitado para inserir os low-rank autoencoders dentro de modelos existentes. Seria também uma forma interessante de entrega de artefatos produzidos via PRs.

Limitações: Documentação escassa e exemplos/tutoriais mais escassos ainda, API em desenvolvimento e propensa a bugs, testes iniciais seguindo receitas do repositório não tiveram sucesso.

PyTorch vanilla

Features: Treinamento com tensores esparsos [ME, CE], algoritmos para decomposição de matrizes (SVD, PCA, etc.), modelos pré-treinados (poucos transformers), integração com Hugging Face.

Possibilidades/Vantagens de Uso: Substituição direta de camadas neurais de modelos existentes, com maior controle de operações de baixo nível. Documentação abundante, comunidade extensa, e API madura.

Limitações: Necessidade de operações de "baixo nível" comparado com outras bibliotecas/frameworks. Pouco suporte da comunidade para este tipo de aplicação.

Hugging Face

Features: Quantização (via frameworks suportados), paralelismo em dispositivos (via frameworks suportados), transformers pré-treinados e integração com diversas bibliotecas.

Possibilidades/Vantagens de Uso: A API de alto nível pode ser usada para o treinamento de transformers em diversas aplicações, enquanto alterações estruturais nos modelos podem ser feitas com PyTorch, permitindo manter a maior complexidade do código a ser desenvolvido apenas onde ela for necessária.

Limitações: Não encontrei suporte nativo para aplicações de LRA, apenas aplicações relacionadas a LoRA.

Provável Caminho a ser Seguido

As intervenções terão que ser feitas usando PyTorch (potencialmente seguindo o modelo de TorchAO), e o treinamento dos transformers em si feito com a API e os modelos da Hugging Face.

O planejamento dos experimentos, até o momento, segue a seguinte ordem:

Estágio 1: Reproduzir a arquitetura de Winata et al e realizar testes preliminares com modelos de CV, NLP e ASP.

Estágio 2: Buscar métricas que possam guiar a escolha do ranque das aproximações.

Estágio 3: Integrar métricas levantadas para escolha adaptativa de ranque durante treinamento (requer máquinas potentes).

APÊNDICE 4

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega (Semana 7)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Experimentos ∞ RES - LRA tests.ipynb **:**

- **Estágio 1:** Funcionamento das fases forward/backward validado**.**
- **Estágio 2:**
  - SVD usado para **inicializar as camadas** LRA inseridas.
  - Exploração de **relações entre SVD, rank e performance** (WIP).
- **Estágio 3:**
  - Sugestão de "tarefas proxy":
    - **Fine tuning** com/sem inicialização.
  - (Pré) Treinamento completo como resposta definitiva.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Experimentos:

- **Estágio 2:** Avaliar perda de performance conforme a redução de rank (tentar **estabelecer parâmetro de corte** para compactação do modelo).
- **Estágio 3:**
  - **Aprimorar** pipeline de **fine tuning** (buscar melhores datasets, hiperparâmetros, etc).
  - **Rodar** fine tuning em modelos **com e sem LRA**.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega (Semana 8)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 14 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Releitura de Artigos sobre LRA Releitura de Artigos de LRA

- **1º artigo (Low-rank Transformer):** Descoberta de pequeno erro de implementação.
- **2º artigo (Harnessing Orthogonality):**
  - SVD(W) = U . Σ . V^T
  - U . V^T estabiliza ao longo do treino, pode ser tratado como constante.
  - **Insights:**
    - **Ajustar apenas Σ**, ao invés de U e (Σ.V^T). Testar ambos.
    - **Critério de redução de rank**: $\Sigma_i > \max(\Sigma) . \beta$
  - Propulate, framework que pode ser usado para pesquisa de arquitetura neural.

Definição de tarefas de fine tuning:

- **NLP:**
  - **Question answering:** SQuAD.
  - **Named Entity Recognition*:** CONLL 2003, GUM-3.1.0, etc.
- **CV:**
  - **Classification:** ImageNet, CIFAR-100.
  - **Object Detection:** COCO, Pascal VOC, ImageNet.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

1) **Executar experimentos** de fine tuning nas tarefas selecionadas.
2) **Avaliar o efeito de β** para buscar **valores ótimos** de redução de rank.
3) **Comparar resultados** otimizando **U** e **(Σ.V^T)** versus otimizar apenas **Σ**.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

[Releitura de Artigos de LRA.doc citado no Termo de Aceite de Entrega de 14 de novembro]

Releitura de Artigos de LRA

[1910.13923] Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer

Percebi um pequeno erro na minha implementação deste trabalho, o LED que substitui a projeção linear de *query* inclui uma *skip connection* que não foi refletida na minha implementação.

Os autores descrevem também uma camada *Low-Rank Feed-forward*, mas a explicação de seu uso me é confusa, não entendo onde ela é utilizada na rede que eles treinaram. Tentarei entrar em contato com os autores para sanar esta dúvida. A rede usada segue uma arquitetura transformers própria, não baseada em um modelo anterior, utilizando apenas parte da VGG como extrator de características, mas não está explícito se essa rede é pré-treinada e congelada, pré-treinada e ajustada junto com o modelo, ou treinada do zero junto com o modelo. O regime de treinamento segue o processo padrão seguido por outros trabalhos, as únicas intervenções feitas foram na arquitetura do modelo.

[2401.08505] Harnessing Orthogonality to Train Low-Rank Neural Networks

Passei um tempo tentando compreender a matemática por trás deste trabalho. Ainda há muito que não entendo completamente, mas o TL;DR é que eles observam que o "componente ortogonal" $U.V^T$ dos pesos de cada camada estabiliza ao longo do treinamento. Assim, após um certo número de passos de treino, das três matrizes resultantes da decomposição SVD – U, Σ, e V^T – é suficiente o ajuste apenas de Σ (as outras duas matrizes continuam presentes como parâmetros não treináveis).

Os autores descrevem o procedimento que eles usam para iterativamente reduzir o rank de Σ conforme avança o treinamento – basicamente esperar certo tempo até as bases estabilizarem, congelar U e V^T, treinar apenas Σ, e esporadicamente reduzir o rank de Σ e atualizar U e V^T com a decomposição SVD de Σ. Como eu não quero interferir no loop de treino, esse procedimento não me interessa muito, mas entendê-lo me levou a dois insights importantes:

- **Critério de redução de rank** utilizado durante a compressão iterativa do modelo. Os autores utilizam um hiperparâmetro β para determinar o novo rank da camada, onde os valores singulares menores que β vezes o maior valor

singular em Σ são removidos. Posso utilizar esta métrica para determinar o rank de cada camada estaticamente, tendo mais flexibilidade do que a metodologia do primeiro trabalho, mas ainda podendo usar o loop de treinamento padrão da Hugging Face.

- **Treinamento mais eficiente em parâmetros**. Minha abordagem anterior era ajustar tanto o encoder LRA quanto o decoder LRA, ajustando U e ($\Sigma.V^T$). Para fine tuning (ou pré-treino, após certo tempo decorrido) eu posso utilizar a técnica desses autores para reduzir o número de parâmetros treinados. A questão é se a adição de Σ como uma camada linear distinta acrescenta overhead computacional suficiente para compensar a redução de overhead produzida pela remoção da necessidade de ajustar U e V^T. Pode ser algo a se testar.

Além disso, outra descoberta interessante a partir da releitura deste artigo é a existência do pacote Populate, que os autores utilizaram para pesquisa de arquitetura neural em seu trabalho. Talvez eu possa utilizá-lo para facilitar a escolha de hiperparâmetros, removendo uma incógnita dos meus experimentos.

APÊNDICE 5

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega (Semana 9)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 27 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Experimentos: RES - LRA fine tunings.ipynb

Question Answering (SQuAD v1.1):

- BERT baseline: **F1 = 88.28** **t = 46:42** 435.6MB
- LED-BERT, β=0.2: F1 = 27.74 t = 46:39 423.1MB **Seedup = 1.0001**
- LED-BERT, β=0.3: F1 = 27.27 t = 44:59 394.3MB Seedup = 1.0381
- LED-BERT, β=0.4: F1 = 26.95 **t = 42:10** **376.6MB** **Seedup = 1.1075**
- USV-BERT, β=0.2: **F1 = 25.80** t = 44:38 **439.8MB** Seedup = 1.0463
- USV-BERT, β=0.3: ERRO
- USV-BERT, β=0.4: ERRO

Image Classification (CIFAR-100):

- ViT baseline: Rodando…

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Executar os testes que faltam: USV-BERT, (LED/USV/Base)-Vit.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Os resultados ficaram muito abaixo do esperado.

ACEITE DA ENTREGA:

LEONARDO ANTÔNIO ALVES: Em análise!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Termo de Aceite de Entrega (Semana 10)

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 5 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Gabriel van der Schmidt

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Erro causado por "shapes" de pesos incorretos + problemas com hardware. Experimentos executados em nuvem.

Experimentos: RES - LRA fine tunings - Colab.ipynb

Testes concluídos:
- **Question answering:** LED e USV.
- **Image classification:** LED.
  - ViT-USV: estimativa de 6 a 7 horas até conclusão.

Resultados: RES - Resultados dos Testes
Tanto LED quanto USV tiveram os melhores resultados com **β=0.3**:
- **LED-BERT**: F1 = 87.51 (88.62) Speedup = 1.07135
- **USV-BERT**: F1 = 86.29 (88.62) Speedup = 1.07464
- **LED-ViT**: acc = 0.8836(0.8929) Speedup = 1.08013

β=0.2 obteve pouco ou nenhum speedup (>1%).
β=0.4 obteve maior speedup (~10%), mas com grande perda de performance, exceto **ViT-LED**.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

[RES - Resultados dos Testes.doc citado nos Termos de Aceite de Entrega de 27 de novembro e 05 de dezembro]

Resultados dos Testes

Resultados dos Testes

SU = Seedup CR = Model compression rate (total/trainable)

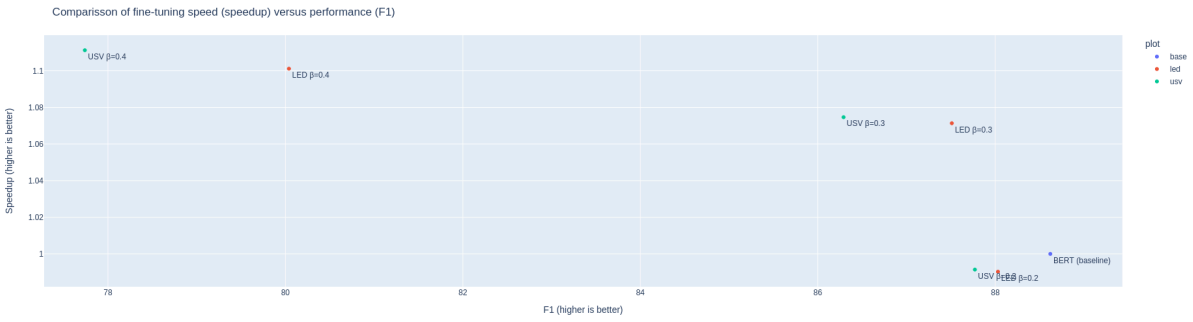
Question Answering (SQuAD v1.1)**:**

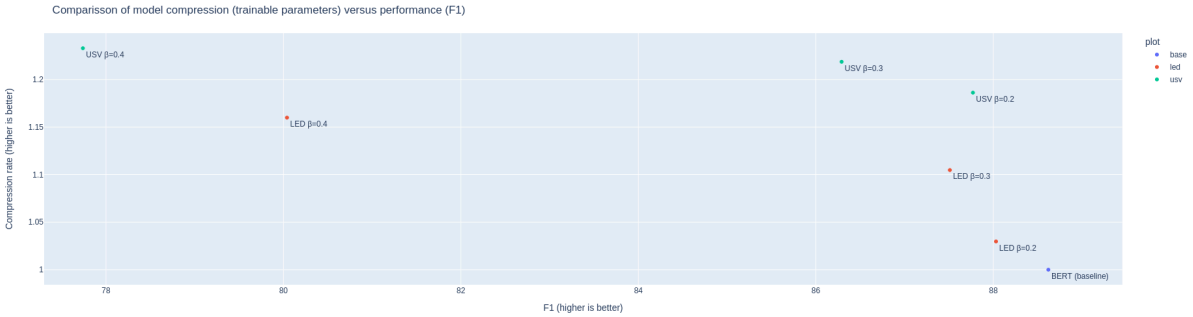
- BERT baseline: **F1 = 88.62** **t = 2:13:29** #params = 108.8M
- LED-BERT, β=0.2: F1 = 88.03 SU = 0.99029 CR = 1.02964
- **LED-BERT, β=0.3: F1 = 87.51 SU = 1.07135 CR = 1.10473**
- LED-BERT, β=0.4: F1 = 80.04 SU = 1.10113 CR = 1.15989
- USV-BERT, β=0.2: F1 = 87.77 SU = 0.99142 CR = 0.99057 / 1.18616
- **USV-BERT, β=0.3:** F1 = 86.29 SU = 1.07464 **CR = 1.08566 / 1.21855**
- USV-BERT, β=0.4: F1 = 77.74 **SU = 1.11124** CR = 1.15140 / 1.23287

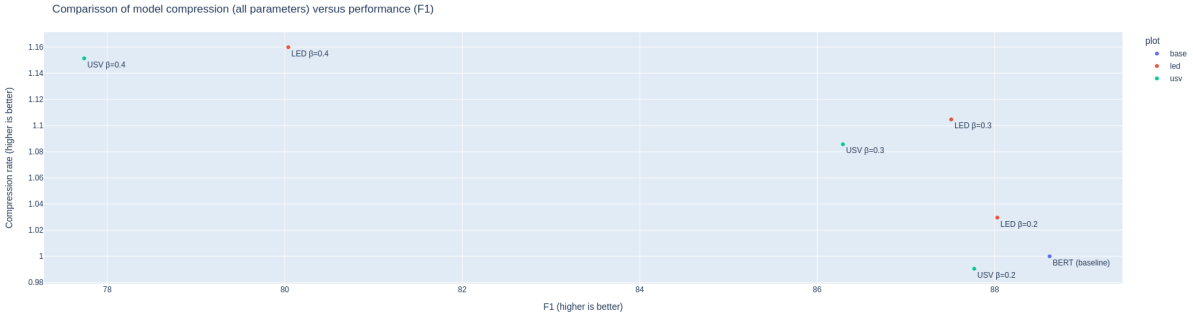
Image Classification (CIFAR-100):

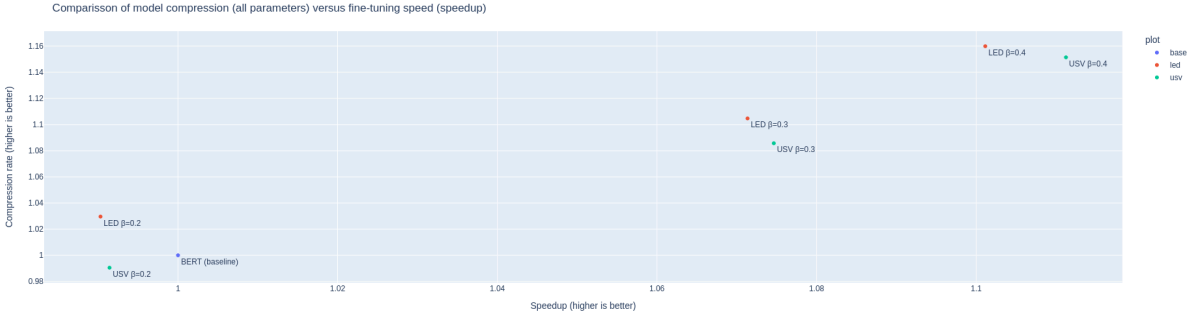
- ViT baseline: **acc = 0.8929** **t = 2:12:43** #params = 85.8M
- LED-ViT, β=0.2: acc = 0.8831 SU = 1.01771 CR = 1.06941
- LED-ViT, β=0.3: acc = 0.8836 SU = 1.08013 CR = 1.15820
- **LED-ViT, β=0.4: acc = 0.8753 SU = 1.12298 CR = 1.22618**
- USV-ViT, β=0.2: acc = 0,8986 SU = 1,05321 CR = 1,02659 / 1,26354
- **USV-ViT, β=0.3: acc = 0,8990 SU = 1,11570 CR = 1,13829 / 1,30289**
- USV-ViT, β=0.4: acc = 0,8886 SU = 1,13234 CR = 1,21833 / 1,31983

BERT

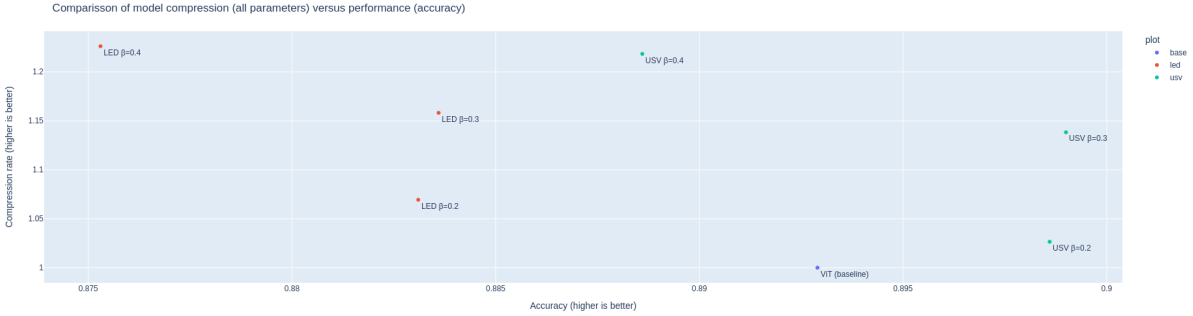


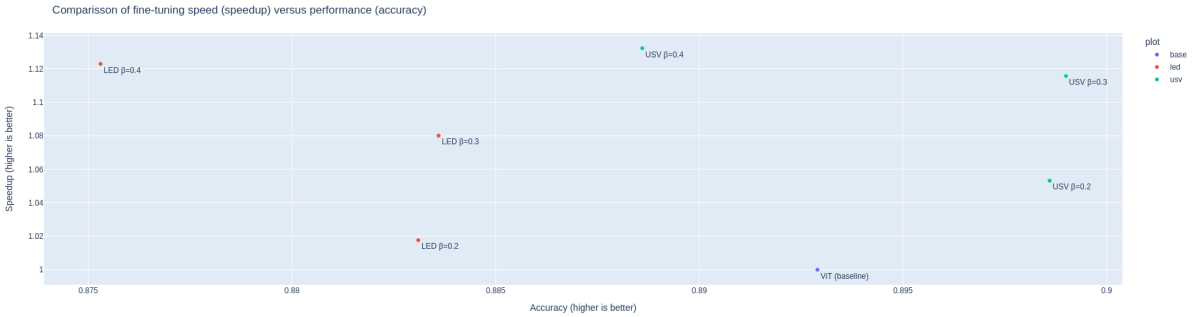

Comparisson of model compression (trainable parameters) versus performance (F1)
Compression rate (higher is better)
F1 (higher is better)
USV β=0.4
USV β=0.3
USV β=0.2
LED β=0.4
LED β=0.3
LED β=0.2
BERT (baseline)
plot
base
led
usv

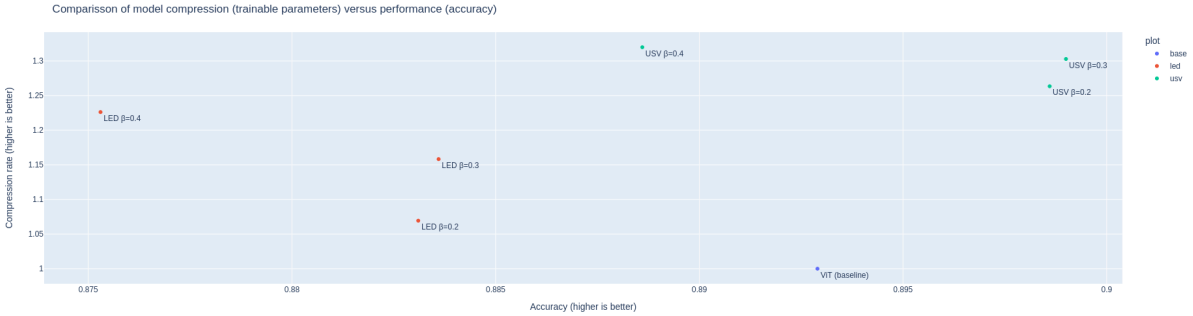

Comparisson of model compression (all parameters) versus performance (F1)
Compression rate (higher is better)
F1 (higher is better)
LED β=0.4
USV β=0.4
LED β=0.3
USV β=0.3
LED β=0.2
BERT (baseline)
USV β=0.2
plot
base
led
usv


Comparisson of model compression (all parameters) versus fine-tuning speed (speedup)
Compression rate (higher is better)
Speedup (higher is better)
LED β=0.4
USV β=0.4
LED β=0.3
USV β=0.3
LED β=0.2
BERT (baseline)
USV β=0.2
plot
base
led
usv

ViT


Comparisson of model compression (all parameters) versus performance (accuracy)
Compression rate (higher is better)
Accuracy (higher is better)
LED β=0.4
USV β=0.4
LED β=0.3
USV β=0.3
LED β=0.2
USV β=0.2
ViT (baseline)
plot
base
led
usv


Comparisson of fine-tuning speed (speedup) versus performance (accuracy)
Speedup (higher is better)
Accuracy (higher is better)
plot
base
led
usv
LED β=0.4
LED β=0.3
LED β=0.2
USV β=0.4
USV β=0.3
USV β=0.2
ViT (baseline)


Comparisson of model compression (trainable parameters) versus performance (accuracy)
Compression rate (higher is better)
Accuracy (higher is better)
plot
base
led
usv
LED β=0.4
LED β=0.3
LED β=0.2
USV β=0.4
USV β=0.3
USV β=0.2
ViT (baseline)

Tipos de LRA Testados

Para maior clareza das hipóteses testadas, reservo esta seção para debater os dois tipos de LRA testados: low-rank encoder-decoder (referido como LER nos Termos de Aceite) e low-rank encoder-sigma-decoder (referido como USV nos Termos de Aceite).

LER – Low-rank encoder-decoder

LER busca decompor uma matriz de pesos $W_{m\times n}$ em duas matrizes de pesos $E_{m\times r}$ e $D_{r\times n}$, onde m, n e r são as dimensões de entrada da camada, de saída, e o ranque da aproximação, respectivamente. $E_{m\times r}$ e $D_{r\times n}$ são, respectivamente, o encoder que codifica informações de entrada em um espaço latente de baixo ranque, e o decoder que decodifica representações no espaço latente.

Os pesos de $E_{m\times r}$ e $D_{r\times n}$ são inicializados a partir da decomposição SVD (single value decomposition) dos pesos de $W_{m\times n}$, onde $E_{m\times r} = U_{m\times r}$ e $D_{r\times n} = \Sigma_{r\times r} \times V_{r\times n}$. Ambos encoder e decoder são otimizados durante o fine-tuning.

Este LRA é baseado no trabalho **[1910.13923] Lightweight and Efficient End-to-End Speech Recognition Using Low-Rank Transformer**.

USV – Low-rank encoder-sigma-decoder

Da mesma forma, USV também busca decompor os pesos $W_{m\times n}$ via SVD. Porém, ao invés de multiplicar $\Sigma_{r\times r}$ e $V_{r\times n}$ para compor o decoder, estas duas matrizes são mantidas separadas, resultando em um LRA de três camadas: $E_{m\times r} = U_{m\times r}$, $\Sigma_{r\times r}$ e $D_{r\times n} = V_{r\times n}$. Os pesos do encoder e do decoder são congelados e apenas os pesos de $\Sigma_{r\times r}$ são atualizados durante o fine-tuning.

Este LRA é baseado no trabalho **[2401.08505] Harnessing Orthogonality to Train Low-Rank Neural Networks**.