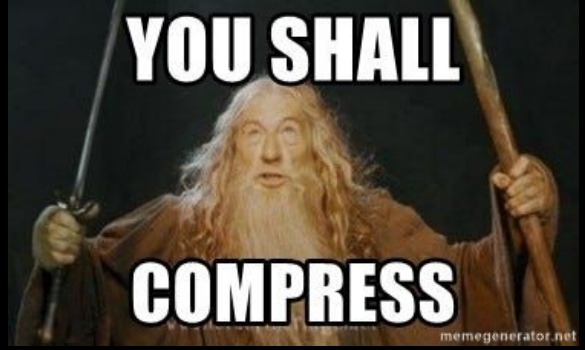


Compressão de Modelos

Paulo Mateus Luza Alves

DSBD

UFPR

Da primeira aula

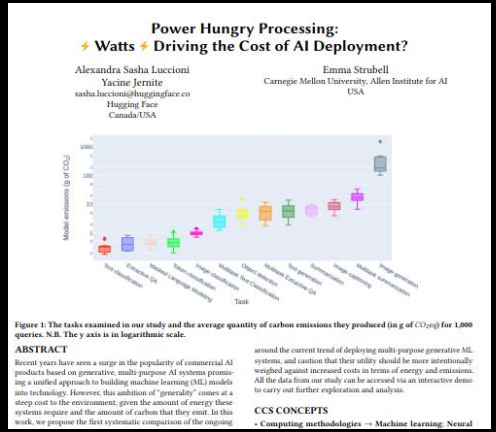
Na primeira aula, comentamos sobre os custos do Deep Learning.

Da primeira aula

Na primeira aula, comentamos sobre os custos do Deep Learning.

Custos de energia

- Cada 1.000 imagens geradas custam em média 2,907 kWh.
- Considerando o ChatGPT, o consumo diário seria equivalente a uma cidade com 160.000 Habitantes.



Power Hungry Processing:
⚡ Watts ⚡ Driving the Cost of AI Deployment?

Alexandra Sasha Luccioni
Yacine Jernite
sasha.luccioni@huggingface.co
Hugging Face
Canada/USA

Emma Strubell
Carnegie Mellon University, Allen Institute for AI
USA

Figure 1: The tasks examined in our study and the average quantity of carbon emissions they produced (in g of CO_2eq) for 1,000 queries. N.B. The y axis is in logarithmic scale.

ABSTRACT

Recent years have seen a surge in the popularity of commercial AI products based on generative, multi-purpose AI systems promising a unified approach to building machine learning (ML) models into technology. However, this ambition of "generality" comes at a steep cost to the environment, given the amount of energy these systems require and the amount of carbon that they emit. In this work, we propose the first systematic comparison of the ongoing around the current trend of deploying multi-purpose generative ML systems, and caution that their utility should be more intentionally weighed against increased costs in terms of energy and emissions. All the data from our study can be accessed via an interactive demo to carry out further exploration and analysis.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**. **Neural**

Luccioni, Sasha, Yacine Jernite, and Emma Strubell. **Power hungry processing: Watts driving the cost of ai deployment?** Proceedings of the 2024 ACM conference on fairness, accountability, and transparency. 2024.

Da primeira aula

Na primeira aula, comentamos sobre os custos do Deep Learning.

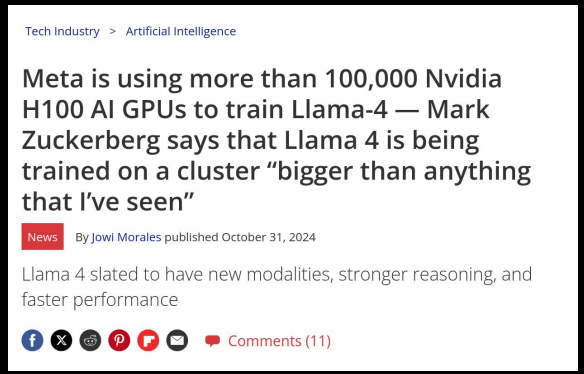
Custos de energia

- Cada 1.000 imagens geradas custam em média 2,907 kWh.
- Considerando o ChatGPT, o consumo diário seria equivalente a uma cidade com 160.000 Habitantes.

Custos de Hardware

- Poder de processamento elevado é necessário.
- Tanto para treinamento quanto inferência.

https://www.tomshardware.com/tech-industry/artificial-intelligence/meta-is-using-more-than-100-000-nvidia-h100-ai-gpus-to-train-llama-4-mark-zuckerberg-says-that-llama-4-is-being-trained-on-a-cluster-bigger-than-anything-that-ive-seen



Tech Industry > Artificial Intelligence

Meta is using more than 100,000 Nvidia H100 AI GPUs to train Llama-4 — Mark Zuckerberg says that Llama 4 is being trained on a cluster "bigger than anything that I've seen"

News By Jowi Morales published October 31, 2024

Llama 4 slated to have new modalities, stronger reasoning, and faster performance

Comments (11)

Qual o problema?

Os modelos estão cada vez maiores e mais complexos.

Conseguimos resultados melhores, porém quanto maior o modelo.

Qual o problema?

Os modelos estão cada vez maiores e mais complexos.

Conseguimos resultados melhores, porém quanto maior o modelo.

> Mais complexo e robusto o hardware precisa ser.

Qual o problema?

Os modelos estão cada vez maiores e mais complexos.

Conseguimos resultados melhores, porém quanto maior o modelo.

- Mais complexo e robusto o hardware precisa ser.
- Torna-se inviável o uso em dispositivos de ponta (Cidades Inteligentes).

Qual o problema?

Os modelos estão cada vez maiores e mais complexos.

Conseguimos resultados melhores, porém quanto maior o modelo.

- Mais complexo e robusto o hardware precisa ser.
- Torna-se inviável o uso em dispositivos de ponta (Cidades Inteligentes).
- Aplicações que necessitam de resposta em tempo real são prejudicadas.

Compressão de Modelos

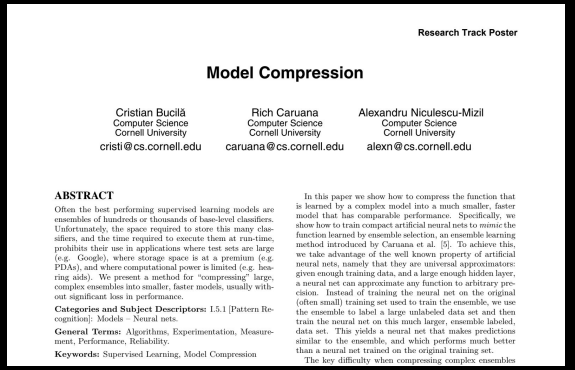
Conjunto de técnicas que visam “comprimir” os modelos.

- Reduzir o tamanho das redes.
- Diminuir o tempo de inferência.

Compressão de Modelos

Conjunto de técnicas que visam “comprimir” os modelos.

- Reduzir o tamanho das redes.
- Diminuir o tempo de inferência.



Research Track Poster

Model Compression

Cristian Bucilă
Computer Science
Cornell University
cristi@cs.cornell.edu

Rich Caruana
Computer Science
Cornell University
caruana@cs.cornell.edu

Alexandru Niculescu-Mizil
Computer Science
Cornell University
alexn@cs.cornell.edu

ABSTRACT

Often the best performing supervised learning models are ensembles of hundreds or thousands of base-level classifiers. Unfortunately, the space required to store this many classifiers, and the time required to execute them at run-time, prohibits their use in applications where test sets are large (e.g. Google), where storage space is at a premium (e.g. PDAs), and where computational power is limited (e.g. hearing aids). We present a method for “compressing” large, complex ensembles into smaller, faster models, usually without significant loss in performance.

Categories and Subject Descriptors: I.5.1 [Pattern Recognition]: Models – Neural nets.

General Terms: Algorithms, Experimentation, Measurement, Performance, Reliability.

Keywords: Supervised Learning, Model Compression

In this paper we show how to compress the function that is learned by a complex model into a much smaller, faster model that has comparable performance. Specifically, we show how to train compact artificial neural nets to *mimic* the function learned by ensemble selection, an ensemble learning method introduced by Caruana et al. [5]. To achieve this, we take advantage of the well known property of artificial neural nets, namely that they are universal approximators: given enough training data, and a large enough hidden layer, a neural net can approximate any function to arbitrary precision. Instead of training the neural net on the original (often small) training set used to train the ensemble, we use the ensemble to label a large unlabeled data set and then train the neural net on this much larger, ensemble labeled, data set. This yields a neural net that makes predictions similar to the ensemble, and which performs much better than a neural net trained on the original training set.

The key difficulty when compressing complex ensembles

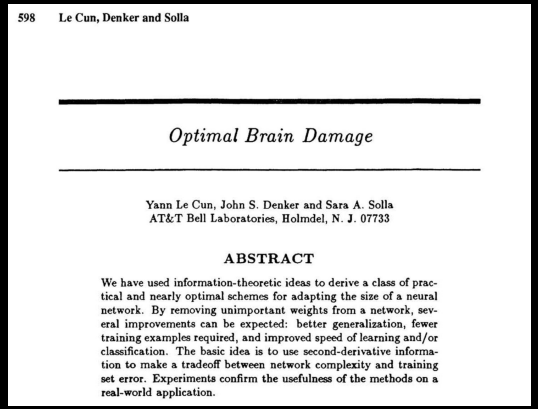
Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006.

O termo *model compression* foi utilizado a primeira vez em 2006, nomeando uma das técnicas atuais. Hoje é utilizado para representar o conjunto.

Compressão de Modelos

Conjunto de técnicas que visam “comprimir” os modelos.

- Reduzir o tamanho das redes.
- Diminuir o tempo de inferência.



598 Le Cun, Denker and Solla

Optimal Brain Damage

Yann Le Cun, John S. Denker and Sara A. Solla
AT&T Bell Laboratories, Holmdel, N. J. 07733

ABSTRACT

We have used information-theoretic ideas to derive a class of practical and nearly optimal schemes for adapting the size of a neural network. By removing unimportant weights from a network, several improvements can be expected: better generalization, fewer training examples required, and improved speed of learning and/or classification. The basic idea is to use second-derivative information to make a tradeoff between network complexity and training set error. Experiments confirm the usefulness of the methods on a real-world application.

LeCun, Yann, John Denker, and Sara Solla. "Optimal brain damage." Advances in neural information processing systems 2 (1989).

O termo *model compression* foi utilizado a primeira vez em 2006, nomeando uma das técnicas atuais. Hoje é utilizado para representar o conjunto.

Porém temos técnicas datadas desde 1989.

Compressão de Modelos

Como comentado, a compressão de modelos engloba um conjunto de técnicas.

Cada técnica possui sua peculiaridade, método de aplicação, etc.

Porém todas possuem o mesmo objetivo, “comprimir” o modelo.

Vamos estudar um pouco sobre algumas das técnicas, não são todas (e não tudo), que englobam a maior parte da pesquisa atual.

Quantização

Quantização é um método para **mapear** entradas, normalmente contínuas, para um conjunto discreto de valores.

Quantização

Quantização é um método para **mapear** entradas, normalmente contínuas, para um conjunto discreto de valores.

Técnica importante no tratamento de sinais digitais.

Busca mapear o sinal da forma mais precisa possível.

Quantização

Em redes neurais, a quantização é utilizada para mapear os parâmetros e/ou ativações, normalmente armazenadas em ponto flutuante (32 bits), para representações de menor precisão (inteiros).

Buscamos fazer esse processo minimizando o impacto na generalização/acurácia dos modelos.

Quantização

Em redes neurais, a quantização é utilizada para mapear os parâmetros e/ou ativações, normalmente armazenadas em ponto flutuante (32 bits), para representações de menor precisão (inteiros).

Buscamos fazer esse processo minimizando o impacto na generalização/acurácia dos modelos.

Diferente das pesquisas tradicionais, não buscamos um modelo quantizado equivalente ao original.

> Modelos podem cair em mínimos locais diferentes.

Quantização

Uma função de quantização comumente utilizada é:

$$Q(r) = Int(r/S) - Z$$

Onde Q é o operador de quantização, ***r*** o valor real de entrada, ***S*** é o fator de escala e ***Z*** o ponto zero. O fator de escala ***S*** pode ser calculado da seguinte forma:

$$S = \frac{\beta - \alpha}{2^b - 1}$$

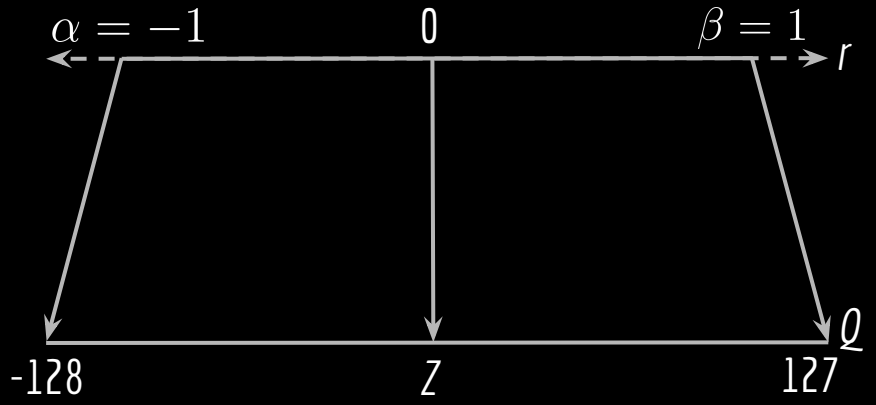
Onde $[\alpha, \beta]$ definem a faixa de corte dos valores reais e ***b*** é a quantidade de bits na quantização.

O processo de definir os valores de $[\alpha, \beta]$ é denominado **calibração.**

Quantização

$$Q(r) = Int(r/S) - Z$$

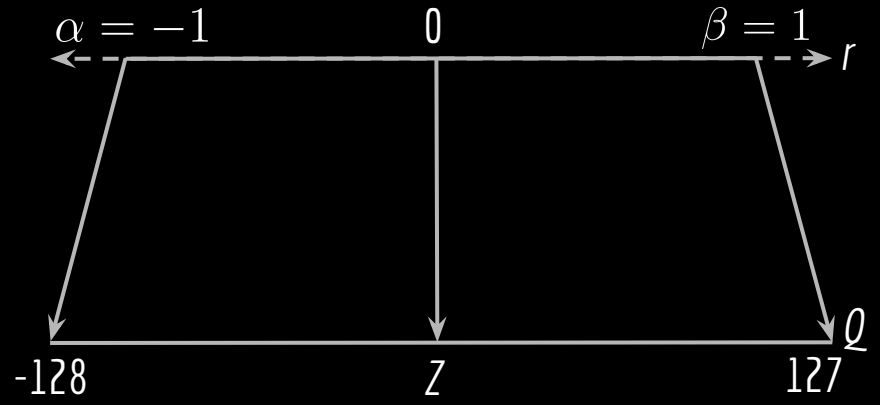
$$S = \frac{\beta - \alpha}{2^b - 1}$$



Quantização

$$Q(r) = Int(r/S) - Z$$

$$S = \frac{\beta - \alpha}{2^b - 1}$$

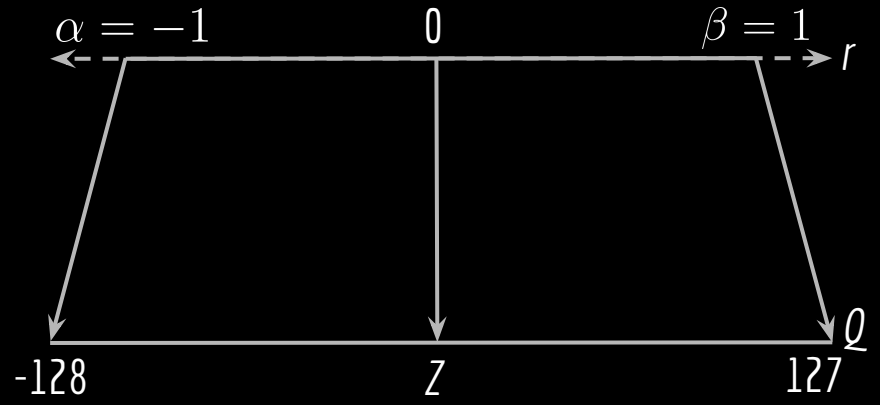


Estes são os conceitos básicos para a quantização **uniforme.**

Quantização

$$Q(r) = Int(r/S) - Z$$

$$S = \frac{\beta - \alpha}{2^b - 1}$$



Estes são os conceitos básicos para a quantização **uniforme.**

Dentro deste tipo de quantização temos várias subcategorias ou métodos diferentes de aplicação.

Existem outros tipos/categorias de quantização.

Pesquise.

Faça você mesmo

Execute o exemplo de quantização no Google Colab disponibilizado.

Pruning

Pruning é um método que consiste em **cortar** estruturas da rede neural para obter compressão.

Pruning

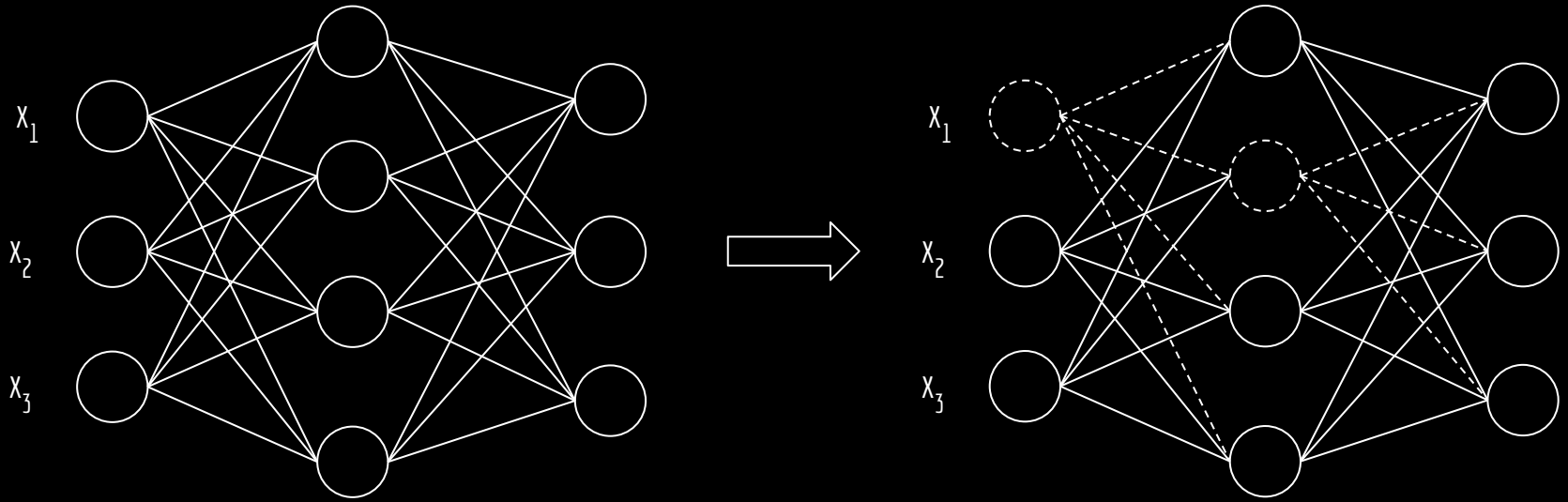
Pruning é um método que consiste em **cortar** estruturas da rede neural para obter compressão.

Estruturas podem ser **pesos**, **neurônios**, **filtros/canais**, etc.

Pruning

Pruning é um método que consiste em **cortar** estruturas da rede neural para obter compressão.

Estruturas podem ser **pesos**, **neurônios**, **filtros/canais**, etc.



Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **pesos,** em qualquer posição da rede, atingimos uma **aceleração específica.**

A remoção de pesos deixa a rede com uma estrutura **irregular**.

Necessita de hardware **específico** para obter o ganho de performance.

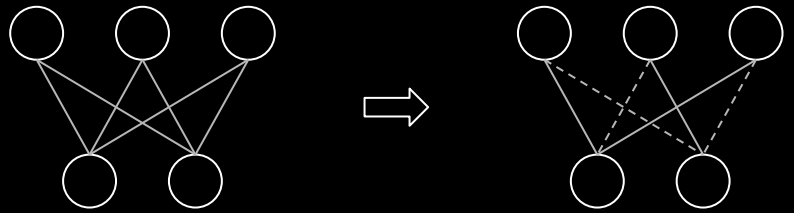
Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **pesos,** em qualquer posição da rede, atingimos uma **aceleração específica.**

A remoção de pesos deixa a rede com uma estrutura **irregular**.

Necessita de hardware **específico** para obter o ganho de performance.



Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **pesos,** em qualquer posição da rede, atingimos uma **aceleração específica.**

A remoção de pesos deixa a rede com uma estrutura **irregular**.

Necessita de hardware **específico** para obter o ganho de performance.

Normalmente é aplicado uma máscara [0, 1] que define os pesos ativos e cortados (como no dropout).

Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **pesos,** em qualquer posição da rede, atingimos uma **aceleração específica.**

A remoção de pesos deixa a rede com uma estrutura **irregular**.

Necessita de hardware **específico** para obter o ganho de performance.

Normalmente é aplicado uma máscara [0, 1] que define os pesos ativos e cortados (como no dropout).

As técnicas que cortam pesos pertencem ao conjunto de **Unstructured Pruning.**

Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **pesos,** em qualquer posição da rede, atingimos uma **aceleração específica.**

A remoção de pesos deixa a rede com uma estrutura **irregular.**

Necessita de hardware **específico** para obter o ganho de performance.

Normalmente é aplicado uma máscara [0, 1] que define os pesos ativos e cortados (como no dropout).

As técnicas que cortam pesos pertencem ao conjunto de **Unstructured Pruning.**

Existe um conjunto denominado **Semi-structured Pruning** que aplica regras ao corte de pesos.

Busca deixar a rede o mais regular possível.

Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **estruturas completas (neurônios, filtros, etc.)**, atingimos uma **aceleração universal.**

Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **estruturas completas (neurônios, filtros, etc.)**, atingimos uma **aceleração universal.**

A remoção das estruturas deixa a rede com uma estrutura **regular.**

Conseguimos reconstruir uma rede comprimida a partir das estruturas restantes.

Não necessita de hardware específico para obter o ganho de performance.

Aceleração universal *vs* específica

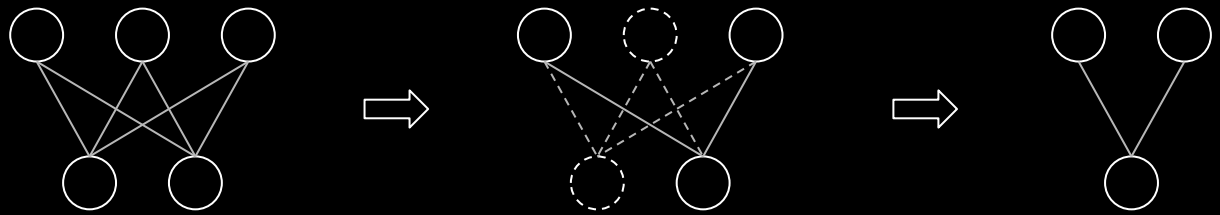
A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **estruturas completas (neurônios, filtros, etc.)**, atingimos uma **aceleração universal.**

A remoção das estruturas deixa a rede com uma estrutura **regular.**

Conseguimos reconstruir uma rede comprimida a partir das estruturas restantes.

Não necessita de hardware específico para obter o ganho de performance.



Aceleração universal *vs* específica

A escolha de qual estrutura que vai ser cortada também define qual tipo de **aceleração** estamos alcançando.

Ao cortar **estruturas completas (neurônios, filtros, etc.)**, atingimos uma **aceleração universal.**

A remoção das estruturas deixa a rede com uma estrutura **regular.**

Conseguimos reconstruir uma rede comprimida a partir das estruturas restantes.

Não necessita de hardware específico para obter o ganho de performance.

As técnicas que cortam estruturas completas pertencem ao conjunto de **Structured Pruning.**

Normalmente possuem degradação de acurácia maior que os métodos de Unstructured Pruning.

Outras “variáveis”

Em adição a qual estrutura cortar, ainda podemos definir outras “variáveis” como:

Outras “variáveis”

Em adição a qual estrutura cortar, ainda podemos definir outras “variáveis” como:

Quando realizar o corte.

Qual critério para realizar o corte.

O quanto cortar a rede.

Outras “variáveis”

Em adição a qual estrutura cortar, ainda podemos definir outras “variáveis” como:

Quando realizar o corte.

Qual critério para realizar o corte.

O quanto cortar a rede.

A definição destas variáveis constituem os mais diversos algoritmos de *pruning* existentes atualmente.

Faça você mesmo

Execute o exemplo de *pruning* no Google Colab disponibilizado.

Destilação de Conhecimento

Destilação de Conhecimento é um método que busca **transferir** o conhecimento de um modelo para outro.

Destilação de Conhecimento

Destilação de Conhecimento é um método que busca **transferir** o conhecimento de um modelo para outro.

Utiliza o *framework* **Estudante-Professor**.

Estudante é um modelo menor, guiado pelo Professor.

Professor é um modelo normalmente grande e custoso que guia o modelo Estudante.

Destilação de Conhecimento

Destilação de Conhecimento é um método que busca **transferir** o conhecimento de um modelo para outro.

Utiliza o *framework* **Estudante-Professor**.

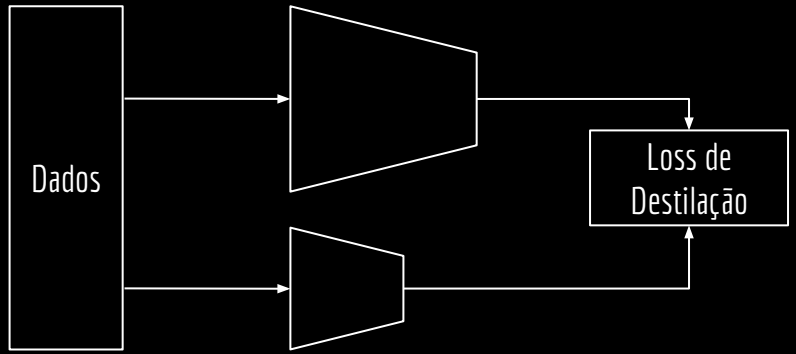
Estudante é um modelo menor, guiado pelo Professor.

Professor é um modelo normalmente grande e custoso que guia o modelo Estudante.

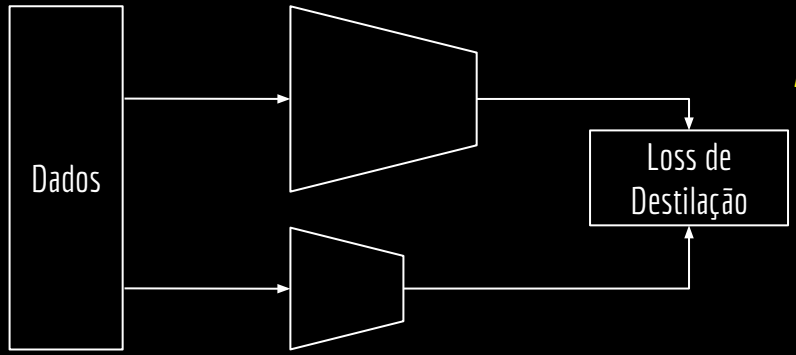
O objetivo é fazer o modelo Estudante **imitar** as saídas do modelo Professor.

*Considere que o trapézio é uma rede neural

Destilação de Conhecimento



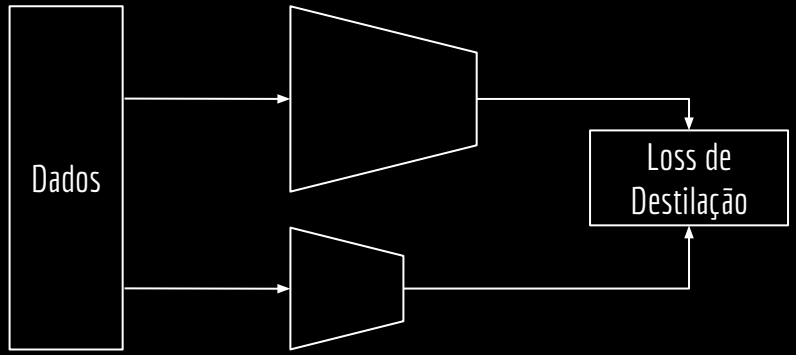
Destilação de Conhecimento



A mágica está na definição desta função de loss

A função de Loss define **como** vamos transferir o conhecimento do modelo Professor para o Estudante.

Destilação de Conhecimento



$$softmax(\hat{y}_i, \tau) = \frac{e^{(\hat{y}_i/\tau)}}{\sum_j e^{(\hat{y}_j/\tau)}}$$

A função de Loss define **como** vamos transferir o conhecimento do modelo Professor para o Estudante.

O método mais simples, porém eficaz, é o uso dos **logits** ou ***Soft-targets*** (probabilidades do modelo Professor).

Em suma, é aplicado uma *softmax* com um fator de temperatura que "espalha" as probabilidades.

As *soft-targets* contém o ***dark knowledge*** do modelo Professor.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Também existem técnicas baseadas em **Features** e **Relações.** Pesquise.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Também existem técnicas baseadas em **Features** e **Relações.** Pesquise.

Também podemos definir o paradigma para destilar o conhecimento.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Também existem técnicas baseadas em **Features** e **Relações.** Pesquise.

Também podemos definir o paradigma para destilar o conhecimento.

Destilação Offline - Treinamento do Professor e Estudante é sequencial.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Também existem técnicas baseadas em **Features** e **Relações.** Pesquise.

Também podemos definir o paradigma para destilar o conhecimento.

Destilação Offline - Treinamento do Professor e Estudante é sequencial.

Destilação Online - Treinamento do Professor e Estudante é paralelo.

Destilação de Conhecimento

O uso de logits não é a única forma de destilar o conhecimento.

As técnicas que os usam são denominadas **Destilação de Conhecimento Baseadas em Respostas**.

Também existem técnicas baseadas em **Features** e **Relações.** Pesquise.

Também podemos definir o paradigma para destilar o conhecimento.

Destilação Offline - Treinamento do Professor e Estudante é sequencial.

Destilação Online - Treinamento do Professor e Estudante é paralelo.

Auto-Destilação - O modelo Estudante é o seu próprio Professor.

Destilando para a classificação de vagas

Redução de 38M de parâmetros para 159K sem perda de acurácia.

Redução de 26x no tamanho do modelo no pior caso.

Modelo gerado compatível com hardwares de edge computing.

Paulo

Luiz

André

Paulo

Optimizing Parking Space Classification: Distilling Ensembles into Lightweight Classifiers

Paulo Luza Alves*, André Hochuli†, Luiz Eduardo de Oliveira*, Paulo Lisboa de Almeida*
* Departamento de Informática (DInf), Universidade Federal do Paraná, Curitiba, PR - Brazil
{paulomateus,luiz.oliveira,paulorla}@ufpr.br
† Programa de Pós-Graduação em Informática (PPGIa), Pontifícia Universidade Católica do Paraná, Curitiba, PR - Brazil
aghochuli@ppgia.pucpr.br

***Abstract*—When deploying large-scale machine learning models for smart city applications, such as image-based parking lot monitoring, data often must be sent to a central server to perform classification tasks. This is challenging for the city's infrastructure, where image-based applications require transmitting large volumes of data, necessitating complex network and hardware infrastructures to process the data. To address this issue in image-based parking space classification, we propose creating a robust ensemble of classifiers to serve as Teacher models. These Teacher models are distilled into lightweight and specialized Student models that can be deployed directly on edge devices. The knowledge is distilled to the Student models through pseudo-labeled samples generated by the Teacher model, which are utilized to fine-tune the Student models on the target scenario. Our results show that the Student models, with 26 times fewer parameters than the Teacher models, achieved an average accuracy of 96.6% on the target test datasets, surpassing the Teacher models, which attained an average accuracy of 95.3%.**

Fig. 1. Occupied (red) and empty (blue) parking spaces – PKLot [1].

I. Introduction

When deploying machine learning models that deal with image-related tasks, we often face the problem that these models may require specialized hardware, such as Graphics Processing Units (GPUs), to deal with massive amounts of inputs, often requiring the acquired images to be sent to a central server, which has the processing power required. In a smart city, where thousands of cameras may be deployed to feed the machine learning models, bottlenecks may be created in these servers and the smart city's network infrastructure.

We propose distilling parking space classification models small-scale deployment with 1,000 cameras. Suppose each camera sends a 1280×720 pixels JPEG compressed image to a central server every 30 seconds. In that case, it will result in a transfer of 35 gigabytes of data per hour to the server, not accounting for the network's overheads (estimated using the PKLot dataset [1]). This data transfer can approximately double for 1920×1080 pixel images.

The hypothesis that small specialist models can be used and deployed in the real world is supported by many works [1]–[4], although the authors of these lightweight approaches use real labels of the target datasets, creating scalability problems, since every time a new camera is deployed, humans need to label the images from the target parking lot to train the model.

We hypothesize that lightweight models that can be deployed in edge devices, fine-tuned using pseudo-labels, can

Alves, Hochuli, Oliveira, Almeida. Optimizing Parking Space Classification: Distilling Ensembles into Lightweight Classifiers. ICMLA 2024.

Faça você mesmo

Execute o exemplo de Destilação de Conhecimento no Google Colab disponibilizado.

Por que funciona?

Como é possível “comprimir” os modelos sem, ou com pouco, impacto na generalização/acurácia dos modelos?

Por que funciona?

Como é possível “comprimir” os modelos sem, ou com pouco, impacto na generalização/acurácia dos modelos?

Não temos uma teoria 100% comprovada ainda.

A hipótese mais forte atualmente se baseia no fato de que os modelos atuais são super-parametrizados.

Relacionado em como construímos e treinamos modelos.

Por que funciona?

Como é possível “comprimir” os modelos sem, ou com pouco, impacto na generalização/acurácia dos modelos?

Não temos uma teoria 100% comprovada ainda.

- A hipótese mais forte atualmente se baseia no fato de que os modelos atuais são super-parametrizados.
- Relacionado em como construímos e treinamos modelos.

As técnicas de *pruning* tem forte apoio nessa hipótese.

- Hipótese do Bilhete de Loteria (*The lottery Ticket Hypothesis*).
- Pesquise.

Juntando técnicas

Além de poder aplicar as técnicas isoladamente, é possível combinar mais de uma técnica para atingir melhores resultados de compressão.

Juntando técnicas

Além de poder aplicar as técnicas isoladamente, é possível combinar mais de uma técnica para atingir melhores resultados de compressão.

É muito comum combinações como Destilação de Conhecimento + Quantização / *Pruning*.

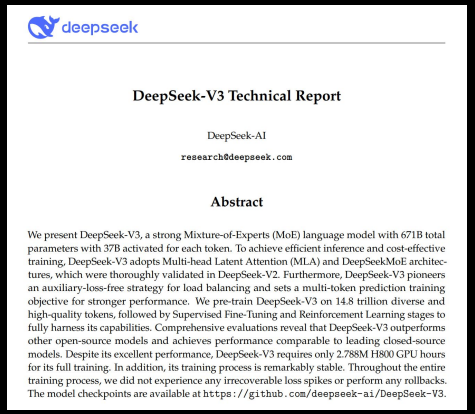
Juntando técnicas

Além de poder aplicar as técnicas isoladamente, é possível combinar mais de uma técnica para atingir melhores resultados de compressão.

- É muito comum combinações como Destilação de Conhecimento + Quantização / *Pruning*.

Um exemplo famoso de junção das técnicas é o *DeepSeek-V3*.

- Destilação de Conhecimento + Quantização durante o treinamento do modelo.


deepseek

DeepSeek-V3 Technical Report

DeepSeek-AI

research@deepseek.com

Abstract

We present DeepSeek-V3, a strong Mixture-of-Experts (MoE) language model with 671B total parameters with 37B activated for each token. To achieve efficient inference and cost-effective training, DeepSeek-V3 adopts Multi-head Latent Attention (MLA) and DeepSeekMoE architectures, which were thoroughly validated in DeepSeek-V2. Furthermore, DeepSeek-V3 pioneers an auxiliary-loss-free strategy for load balancing and sets a multi-token prediction training objective for stronger performance. We pre-train DeepSeek-V3 on 14.8 trillion diverse and high-quality tokens, followed by Supervised Fine-Tuning and Reinforcement Learning stages to fully harness its capabilities. Comprehensive evaluations reveal that DeepSeek-V3 outperforms other open-source models and achieves performance comparable to leading closed-source models. Despite its excellent performance, DeepSeek-V3 requires only 2.788M H800 GPU hours for its full training. In addition, its training process is remarkably stable. Throughout the entire training process, we did not experience any irrecoverable loss spikes or perform any rollbacks. The model checkpoints are available at https://github.com/deepseek-ai/DeepSeek-V3.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C. and Dai, D., 2024. **Deepseek-v3 technical report**. arXiv preprint arXiv:2412.19437.

Neural Architecture Search (NAS)

Utiliza um espaço de busca contendo estruturas/parâmetros conhecidos que compõem as redes neurais.

Filtros de convolução, blocos residuais, camadas de batch normalization, etc.

Neural Architecture Search (NAS)

Utiliza um espaço de busca contendo estruturas/parâmetros conhecidos que compõem as redes neurais.

- Filtros de convolução, blocos residuais, camadas de batch normalization, etc.

Aplica um **algoritmo de busca** neste espaço.

- Busca encontrar o melhor modelo dado um critério (quantidade de parâmetros, por exemplo).

Neural Architecture Search (NAS)

Utiliza um espaço de busca contendo estruturas/parâmetros conhecidos que compõem as redes neurais.

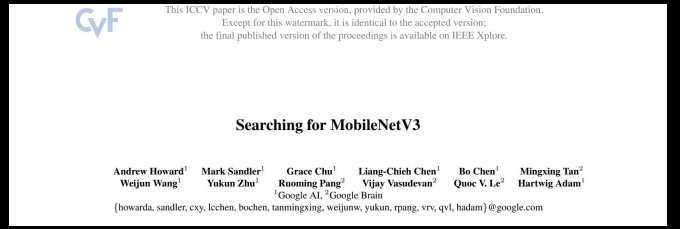
Filtros de convolução, blocos residuais, camadas de batch normalization, etc.

Aplica um **algoritmo de busca** neste espaço.

Busca encontrar o melhor modelo dado um critério (quantidade de parâmetros, por exemplo).

Um exemplo famoso de rede resultante do NAS (e que usamos algumas vezes) é a **MobileNetV3.**

Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. and Le, Q.V., 2019. **Searching for mobilenetv3**. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314-1324).



Searching for MobileNetV3

Andrew Howard[1] Mark Sandler[1] Grace Chu[1] Liang-Chieh Chen[1] Bo Chen[1] Mingxing Tan[2]
Weijun Wang[1] Yukun Zhu[1] Ruoming Pang[2] Vijay Vasudevan[2] Quoc V. Le[2] Hartwig Adam[1]
[1]Google AI, [2]Google Brain
{howarda, sandler, cxy, lcchen, bochen, tanmingxing, weijunw, yukun, rpang, vrv, qvl, hadam}@google.com

Exercícios

1. Modifique o exemplo de *Pruning* para usar outros métodos de corte e compare os resultados.
    a. Teste coisas como diferentes critérios para o *pruning*, técnicas de *structured pruning*, etc.
2. Modifique o exemplo de Destilação de Conhecimento (ou o de *pruning*) para executar os métodos em conjunto.
    a. Compare a acurácia do método de *pruning* isolado e quando aplicado com a Destilação de conhecimento.
    b. Os resultado melhoraram, pioraram?

Referências

Mansourian, A.M., Ahmadi, R., Ghafouri, M., Babaei, A.M., Golezani, E.B., Ghamchi, Z.Y., Ramezanian, V., Taherian, A., Dinashi, K., Miri, A. and Kasaei, S., 2025. **A Comprehensive Survey on Knowledge Distillation.** arXiv preprint arXiv:2503.12067.

Moslemi, A., Briskina, A., Dang, Z. and Li, J., 2024. **A survey on knowledge distillation: Recent advancements.** Machine Learning with Applications, 18, p.100605.

Cheng, H., Zhang, M. and Shi, J.Q., 2024. **A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations.** IEEE Transactions on Pattern Analysis and Machine Intelligence.

Wei, L., Ma, Z., Yang, C. and Yao, Q., 2024. **Advances in the neural network quantization: A comprehensive review.** Applied Sciences, 14(17), p.7445.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W. and Keutzer, K., 2022. **A survey of quantization methods for efficient neural network inference.** In Low-power computer vision (pp. 291-326). Chapman and Hall/CRC.

Licença