Darmstadt University of Applied Sciences

Computer Vision Seminar

Unearthing Surgical-Dino

Author
Tilman Seeßelberg

Reviewer
Prof. Dr. Andreas Weinmann, Darmstadt University of Applied Sciences
Vladyslav Polushko, Darmstadt University of Applied Sciences

Date of Submission: June 13, 2024

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

Abstract

Kurzzusammenfassung

Contents

List of Figures

List of Tables

List of abbreviations

MLP	Multi Layer Perceptron

Chapter 1
Introduction

Chapter 2 Theoretical Foundations

2.1 Transformers

Quellen für RNNs und LSTMs?

Transformers were first introduced in 2017 as successors to neural machine translation (NMT) algorithms such as long short-term memory (LSTM) and Recurrent Neural Networks (RNNs) [1]. The transformer architecture follows an encoder-decoder structure, where the encoder maps an input sequence to a continuous representation, and the decoder generates the output sequence from this representation.

Transformers ~~solve~~ several problems that previous state-of-the-art algorithms such as LSTM and RNNs have struggled with, including computational efficiency and context window size [2, p 609]. The context window refers to the length of the input sequence considered for predicting the next word or token. A small context window limits the ability to attend to words that are far apart, which can be problematic for understanding long texts or complete documents. The transformer model efficiently handles a much larger, theoretically infinite, context window using a mechanism called attention, which comes in two flavors: self-attention and cross-attention.

In general, a transformer is built from several different building blocks: the embedding layer, the positional encoding, the attention layers, and the feed-forward neural networks. → Bild

Loss erwähnen?

Before sequential data is input into the transformer model, it needs to be embedded. The embedding model is a dictionary that maps known small patterns such as words, syllables, or other snippets in sequences to so-called embedding vectors. This way, a sentence can be represented as a matrix where each column is an embedding vector of a word in the sentence. As the position of a word in a sentence can provide important context, a positional encoding is added as an additional pattern for the model to learn.

inductive bias ~?

Positional encoding is necessary because transformers lack the sequential ~~nature~~ of RNNs and CNNs, which inherently capture positional information. have The positional encoding is generated using the following functions, calculated for each element in the matrix. To every element in an even row index i, the result of function 2.1 is added, and to every element in an odd row index, the result of function 2.2 is added. pos is the actual column index, representing the position of the token in the input sequence. | Bild Matrix

s

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.2}$$

With the input sequence embedded and the positional encoding added, the sequence is ready to be fed into the attention layers.

Attention Layers, also known as **Alignment Models** in machine learning, are a type of trainable memory that can determine the relationship between an embedding vector and other embedding vectors given an arbitrarily long input sequence. This concept was first introduced in 2014 by Bahdanau et al. [3] as additive attention in the context of machine translation and refined by Luong et al. [4] n 2015 as multiplicative attention, which is more computationally efficient and used in transformers.

ref → Bild

importance

The attention layer outputs an $n \times n$ matrix where n is the number of tokens in the input sequence. Each element in that matrix represents an energy score, which can be interpreted as how much the word in the row is related to the word in the column, similar to a correlation matrix. ~~Usually for each word only the relation to previouse~~ words is considered, so the upper right triangle of the matrix is set to $-\infty$, later changed to 0 after applying a softmax function. The attention mechanism implemented in the original transformer paper ~~'Attention is All You Need'~~ [1] uses trainable weights in the form of three parallel linear layers, producing a query matrix Q, a key matrix K and a value matrix V. These matrices have the same dimensionality $d_k \times d_k$ where d_k is the dimensionality of the embedding vector. The attention is calculated using equation 2.3 by multiplying Q and K, resulting in a square matrix called the attention score. This score is then normalized to prevent exploding gradients before being multiplied with the value matrix V. The softmax function is applied to ensure the sum of the values in each row is 1.

Autoren

Bild

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

Multi-head attention allows the model to learn different types of relationships, such as grammatical structure or contextual changes in word meaning. Multiple attention heads in parallel, each learning independently, are concatenated and processed by a final linear layer. The entire process is illustrated in figure 2.1.

The primary advantages of transformers include the ability to handle long-range dependencies, significant parallelization leading to faster training times, and superior performance on ~~various tasks~~ like machine translation. Experiments have shown that transformers can be trained significantly faster than architectures based on recurrent or convolutional layers.

~~**WORK IN PROGRESS**~~ ||

by X et al. [ ]

how much

2.2 Vision Transformer

2.3 Vision Transformer

Um schreiben

One of the first applications of attention mechanisms in the computer vision field was developed by Xu et al. in 2015 [5], who used a CNN and a downstream RNN with an attention mechanism to generate image descriptions. However, Vision Transformers (ViTs) were only introduced in 2020 by Dosovitskiy et al. [6] for the task of image classification. ViTs are an evolution of transformers as discussed in 2.1.

The ViT, while reusing most parts of the original transformer, presents several key differences from the original transformer architecture. Unlike the original transformer which employs both an encoder and a decoder, the ViT only uses an encoder, as the task of image classification does not necessitate a decoder. Just like the transformer the ViT processes 1d tokens. These however are not generated using a dictionary like $\overset{\text{in a}}{\text{embeddign}}$ layer as the transformer, but simply by splitting the image into 16×16 patches and flattening them into a 1D tensor. These flattened patches undergo a learnable linear transformation and are treated as embedding vectors, like in the original architecture. Another significant difference lies in the positional embedding. While the original transformer uses fixed positional embedding, the ViT implements a learnable one. Additionally, a so called class token is ~~prepended~~ to the embedding vector matrix, ~~which is meant~~ to destil the information of the image into a single vector, which later is used for classification. Despite these differences, the ViT has demonstrated superior performance over state-of-the-art

Math

similar to

— Bild

Bsp(!

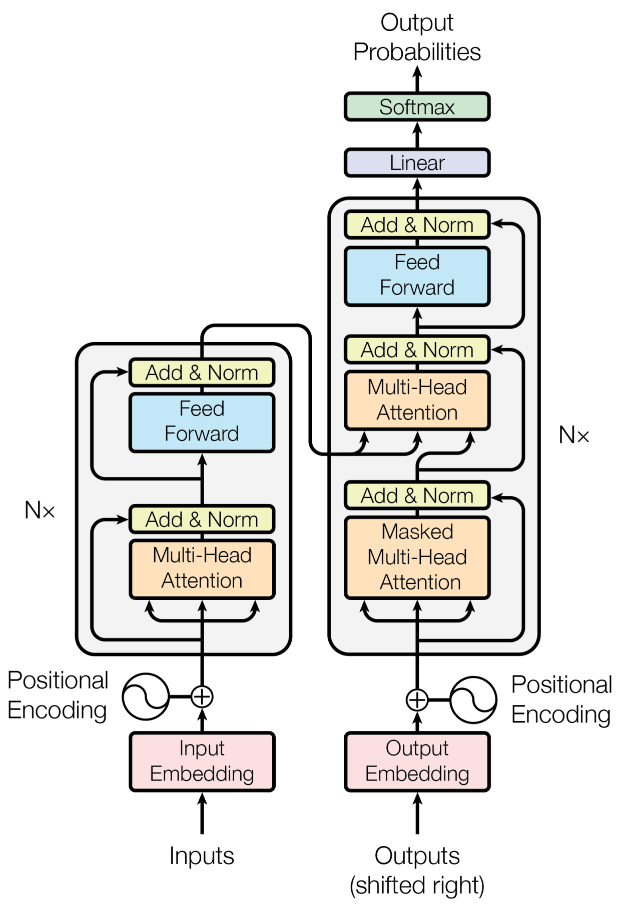

Figure 2.1: The transformer architecture (copied from the original paper [1])

Multi-head?

Concat?

Samples?

convolutional neural networks (CNNs) in several image classification tasks. However, it requires significantly more training data and computational resources, ~~as~~ unlike CNNs, transformers do not inherently leverage spatial information in images.

Paragraph?

ViTs involves several key steps and components:

- **Patch Embedding**: The input image $x \in \mathbb{R}^{H \times W \times C}$ devided into non-overlapping patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, with the original resolution (H, W), the color channels C, the path resulution (P, P) and $N = HW/P^2$ as the number of patches. Each patch is flattend and linearly embedded into a embedding vector.
- **Position Embedding**: To retain positional information, standard learnable 1D position embeddings are added to the patch embeddings as 2D position embeddings did not yield any advantage.
- **Transformer Encoder**: The embedded patches, along with positional embeddings, are fed into the transformer encoder. The encoder consists of alternating layers of multi-headed self-attention (MSA) and multi-layer perceptron (MLP) blocks. Each block applies layer normalization (LN) before and residual connections after every block.

The architecture of the Vision Transformer is mathematically described in the paper [1] as follows:

$$z_0 = [x_{\text{class}}; x_1^p E; x_2^p E; \cdots; x_N^p E] + E_{\text{pos}} \quad (2.4)$$

,
,
none

where $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is the trainable linear projection for patch embedding, and $E_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ are the position embeddings.

Aaaah

The transformer encoder operates on these embeddings through L attention heads:

$$z'_l = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1}, \quad l = 1 \dots L \tag{2.5}$$

?

$$z_l = \mathrm{MLP}(\mathrm{LN}(z'_l)) + z'_l, \quad l = 1 \dots L \tag{2.6}$$

where z_l represents the output of the l-th layer.

Finally, the output representation for classification is obtained from the embedded class token:

$$y = \mathrm{LN}(z_L^0) \tag{2.7}$$

?

The Vision Transformer (ViT) demonstrates that a pure transformer model applied directly to sequences of image patches can achieve state-of-the-art performance on image classification tasks. The success of ViTs underscores the importance of pre-training on large datasets and the flexibility of transformer architectures in processing different types of data beyond natural language.

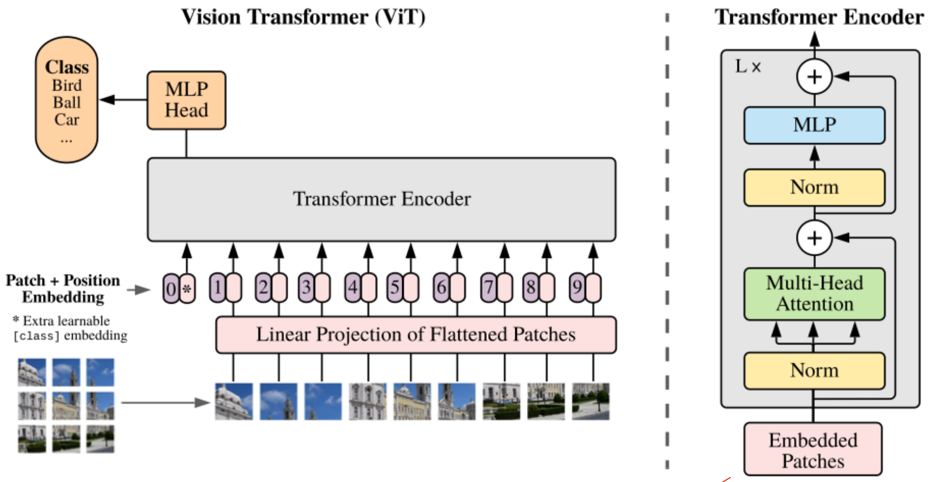

Figure 2.2: The Vision Transformer architecture (copied from [6]).

2.4 Dino and DinoV2

So far ViTs were mostly trained in a supervised manner and needed a lot of labeled data to achieve state-of-the-art performance. As labeled trainingdata is expensive and time consuming to generate, it is very desirable to train models in an unsupervised manner. Just a year after the introduction of the Vision Transformer, Caron et al. [7] introduced an approach to achieve exactly this called DINO (DIstilled NOt labeled), which is a framework for training in theory any network capable of ingesting image data using self-distillation without labels. However the paper Name in particular emphasizes the synergy of Dino with Vision Transformers as a good compromise between performance and computational cost. Dino is a usefull framework for tasks such as segmentation, depth estimation, and object detection, as it provides a good initialization for supervised fine-tuning. by name

As explained in [8] destillation in machine learning is a method facilitating the transfer of knowledge from one model to another using a teacher-student model approach. Often the teacher is a larger complex pretrained model which is used to train a much smaller stundent model. However, in the case of Dino the teacher is a ~~non-pretrained~~ model being trained in parallel with the student model. This process is called self-distillation, as no pretrained model is used to inject prexisting knowledge into the student model.

In a teacher stundent architecture the student tries to predict the output of the teacher model. In Dino the students weights get updated using the classical gradient decend approach, while the teacher model is updated using a exponential moving average of the student model weights. Caron et al. figured that in order to prevent a trivial solution where both teacher and student would start outputing the same output no regard to the input, it is enough to center the output of the teacher model to a mean of 0 and to apply a sharpened softmax function to the output by applying a temperatur parameter τ called temperature to the softmax function as shown in equation 2.8. It makes very large values even larger and very small values even smaller.

Architecture

Dino is trained using a classical student teacher network approach, though the teacher is not a pretrained model but a model trained in parallel to the student model. The Tracher model is largely the same as the student model, but the outputs follow a normalisation applying centering subtracting the mean and running the resulting outputs thorugh a shapened softmax function as shown in equation 2.8.

$$q_i = \frac{f_i}{|f_i|_2} \cdot \tau \tag{2.8}$$

- Attention Map right after training puts much attention on the main object of the image and less on the background. This already is a good start for segmentation - Dino is trained unsupervised and places, when trained, images in to a latent vector, where similar images are close to each other, similar to how similar words are embedded close to each other when training a embedding on them. This could already be used for a k-nearest neighbor approach to classify images.

2.4.1 Depth Estimation

2.4.2 Segmentation

2.4.3 Low Rank Adaption (LoRA)

Chapter 3
Unearthing Surgical-Dino

3.1 Improvements of Surgical-Dino

Emphasis Contribution of Surgical-Dino

Chapter 4
Discussion

Chapter 5
Future Work

Chapter 6
Conclusion

Appendix A
Code Tools

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762.

[2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems, Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, 2022, ISBN: 9781098125974.

[3] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014. DOI: 10.48550/ARXIV.1409.0473.

[4] M.-T. Luong, H. Pham, and C. D. Manning, *Effective approaches to attention-based neural machine translation*, 2015. DOI: 10.48550/ARXIV.1508.04025.

[5] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 2048–2057. [Online]. Available: https://proceedings.mlr.press/v37/xuc15.html.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2020. DOI: 10.48550/ARXIV.2010.11929.

[7] M. Caron, H. Touvron, I. Misra, *et al.*, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9650–9660.

[8] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, Jul. 2019, pp. 5142–5151. [Online]. Available: https://proceedings.mlr.press/v97/phuong19a.html.