Efficient Transformers applied to video classification

TFG

Oriol Martínez Pérez

Universitat de Barcelona

29th June 2023

Table of Contents

Table of Contents

Motivation

Huge growth on Machine Learning the during last years.

This growth has been highly influenced by the appearance of a specific model architecture: **the Transformer**

Based on **self-attention**, it provided a new approach which overcame some limitations at that moment and lead to better models

Why self-attention?

Before self attention, CNNs and RNNs were the standard procedure for NLP tasks.

CNNs

- Based on convolutional layers.
- Induced locality

RNNs

- Added sequentially
- Use recurrency

Problem

They can not handle long term dependencies and consider all input at once.

Self-attention overcomes this limitations.

Self-attention idea

Consider how strongly related each word is with all the others.

Example

For the sentence *Transformers are awesome, I love them.*

	Transformers	*are*	*awesome*	*I*	*love*	*them*
Transformers	1	0.7	0.6	0.2	0.3	0.8
are	0.7	1	0.8	0.2	0.1	0.6
awesome	0.6	0.8	1	0.2	0.3	0.3
I	0.2	0.2	0.2	1	0.8	0.6
love	0.3	0.1	0.3	0.2	1	0.8
them	0.8	0.6	0.3	0.6	0.8	1

Problem

Matrix with n^2 coefficients. $n =$ input length

Objectives

Two main objectives:

- Understand and mathematically build self-attention
- Present and experimentally compare some of the most relevant efficient self-attention mechanisms: **Nyströmformer, Linformer** and **Cosformer**.

Specifically, we focus on video classification since videos are a heavy input with long term dependencies.

Table of Contents

Kernels

Let $\mathcal{X} = x_1, \dots, x_n$ be any set.

Kernel

A kernel is a function
$$k : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$$
where $k(x, y)$ is a real number characterizing their pairwise similarity.

A kernel is **positive semidefinite** if its Gram matrix is positive semidefinite.

The first step is to send $\mathcal{X} \times \mathcal{X}$ to a suitable space where we can compute $k(x, y)$.

Let $\mathcal{H}$ be a linear space equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

Feature map

A **feature map** is a function ϕ

$$\phi : \mathcal{X} \longrightarrow \mathcal{H}$$

From now on, we consider $\mathcal{H} = \mathbb{R}^d$.
Dot product acts as similarity measure and we can consider it as $k(x, y)$.

Mathematical construction

First we define 3 different feature maps and apply them to all elements of $\mathcal{X}$:

$$W_Q : \mathcal{X} \longrightarrow \mathbb{R}^d \quad W_K : \mathcal{X} \longrightarrow \mathbb{R}^d \quad W_V : \mathcal{X} \longrightarrow \mathbb{R}^d$$
$$x_i \mapsto x_q^i \qquad\qquad x_i \mapsto x_k^i \qquad\qquad x_i \mapsto x_v^i$$

We obtain 3 $M_{n\times d}$ matrices: Q, K and V.

Q, K, V

These 3 matrices are the core of self-attention and are called **query, key** and **value** respectively.

Using Q, K we compute the normalized **scores** matrix:

$$\frac{QK^\top}{\sqrt{d}}$$

And using a Softmax normalization we transform them to probabilities:

$$\mathcal{S} = Softmax\left(\frac{QK^\top}{\sqrt{d}}\right)$$

Multiplying by V we obtain the self-attention formula

$$SelfAttention(Q, K, V) = Softmax\left(\frac{QK^\top}{\sqrt{d}}\right) V$$

Self-attention complexity

Operation	**Complexity**
$QK^\top$	$\mathcal{O}\left(dn^2\right)$
$\frac{QK^\top}{\sqrt{d}}$	$\mathcal{O}\left(n^2\right)$
$Softmax\left(\frac{QK^\top}{\sqrt{d}}\right)$	$\mathcal{O}\left(n + 2n^2\right)$
$Softmax\left(\frac{QK^\top}{\sqrt{d}}\right)V$	$\mathcal{O}\left(n^2 d\right)$

Considering $d << n$,

Self-attention complexity

$\mathcal{O}\left(n^2\right) \Rightarrow$ Prohibitive calculations and memory requirements when n is of the order of 10^3

Table of Contents

Categorization of efficient self-attention mechanisms

They can be grouped into 3 groups:

- **Sparse attention:** Reduce number of inputs
- **Attention reformulations:** compute $SelfAttention(Q, K, V)$ in a more efficient way through the rearrangement of operators
- **Low-rank methods:** Assume $Softmax\left(\frac{QK^\top}{\sqrt{d}}\right)$ has redundancies and an approximated version can be used

Attention Reformulations

Main idea

Compute $K^\top V$, $\mathcal{O}(nd^2)$, instead of $QK^\top$, $\mathcal{O}(dn^2)$

We find a function $\phi : M_{n\times d} \longrightarrow M_{n\times d}$ that modifies Q and K, such that

$$\mathcal{S} = Softmax\left(\frac{QK^\top}{\sqrt{d}}\right) \approx \phi(Q)\phi(K^\top) = \mathcal{S}'$$

Therefore,

$$SelfAttention(Q, K, V) = \mathcal{S}'V = \Big(\phi(Q)\phi(K^\top)\Big)\, V = \phi(Q)\,\Big(\phi(K^\top)V\Big)$$

Problem

We are not using Softmax, losing the probabilistic approach and the positive semidefinite property.

Cosformer

Sets $\phi = \mathrm{ReLU}$. Consider $Q' = \mathrm{ReLU}(Q) \quad K' = \mathrm{ReLU}(K)$. Alternative normalization to Softmax:

$$SelfAttention_i = \frac{\sum_{j=1}^{n} \left(Q_i' K_j'^{\top} \right) V_j}{\sum_{j=1}^{n} \left(Q_i' K_j'^{\top} \right)} \quad 1 \leq i \leq n$$

Using Ptolemy's theorem:

$$\mathcal{S}`_{ij} = Q_i' {K'}_j^{\top} \cos\left(\frac{\pi}{2} \times \frac{i-j}{\alpha} \right)$$

Therefore:

$$\mathcal{S}' = Q^{cos} K^{cos} + Q^{sin} K^{sin}$$

And we can compute

$$SelfAttention(Q, K, V) = Q^{cos} \left(K^{cos} V \right) + Q^{sin} \left(K^{sin} V \right)$$

Low-rank methods

Main idea

Find approximated version of *Softmax* $\left(\frac{QK^\top}{\sqrt{d}}\right)$, $\mathcal{S}'$, such that $S' \in M_{k\times d}$, with $k < n$. Therefore, the n^2 complexity term is reduced.

Existence of $\mathcal{S}'$ warrantied by the following theorem:

For any Q, K, $V \in M_{n\times d}$, for any column vector $v \in \mathbb{R}^n$ of V, there exists a low-rank matrix $\tilde{\mathcal{S}} \in M_{n\times n}$ such that

$$\Pr\left(\|\tilde{\mathcal{S}}v^\top - \mathcal{S}v^\top\| < \epsilon\|\mathcal{S}v^\top\|\right) > 1 - o(1)$$

and $rank(\tilde{\mathcal{S}}) = \Theta(\log n)$.

Linformer

Let E and F be two $M_{k \times n}$ learnable matrices. We define

$$K' = EK \quad V' = FV, \quad K', V' \in M_{k \times d}$$

It uses as the approximated version the matrix

$$\mathcal{S}' = Softmax\left(\frac{QK'^{\top}}{\sqrt{d}}\right)$$

And computes self-attention as follows:

$$SelfAttention(Q, K, V) = \mathcal{S}'V'$$

Complexity

Considering $k << n$, $\mathcal{O}(n)$

Nyströmformer

Select m rows of Q and K matrices, with $m < n$, called **landmarks.**
Let $\tilde{Q}$ and $\tilde{K}$ be the selected landmarks.
We consider the following matrix and its SVD decomposition:

$$\mathcal{S}' = Softmax\left(\frac{\tilde{Q}\tilde{K}^{\top}}{\sqrt{d}}\right)$$
$$= \left(Softmax\left(\frac{Q\tilde{K}^{\top}}{\sqrt{d}}\right)\right)\left(Softmax\left(\frac{\tilde{Q}\tilde{K}^{\top}}{\sqrt{d}}\right)\right)^{+}\left(Softmax\left(\frac{\tilde{Q}K^{\top}}{\sqrt{d}}\right)\right)$$

And multiply by V, $\mathcal{S}'V$.

Complexity

Taking into account the landmark computations and SVD decomposition, with $d << n$ and $m << n$, $\mathcal{O}(n)$

Table of Contents

Objective

We wanted to answer the following question:

Which self attention mechanism has the best performance/computational cost trade off for video classification?

To do so, we needed:

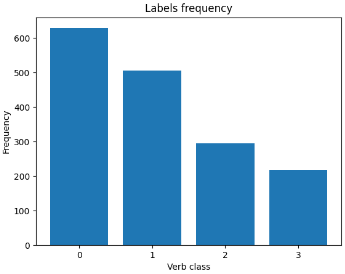
- Dataset
- Construct a transformer architecture
- Train and test the transformer with the three considered self attention mechanisms.

Dataset

Obtain from EpicKitchens-100. Consists of first-person video of kitchen activities.

	Hours	Videos	Clips	Verb Classes
Train	1.8	338	1648	4
Validation	0.3	168	291	4
Test	0.3	85	280	4

Table: Epic-Kitchens-100 modified dataset



Input embedding

Consider videos with RGB encoding, $x \in \mathbb{R}^{3 \times H \times W \times T}$. We need to obtain our tokens from it.

Figure: Spatio-temporal tokenization

We use a **sinusoidal positional encoding** to retain positional information.

Attention block

Self-attention is computed in parallel using **multi-head attention**:

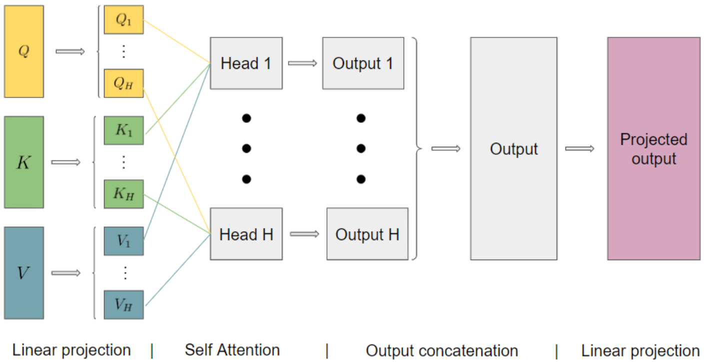


Figure: Multi-head attention

Transformer architecture

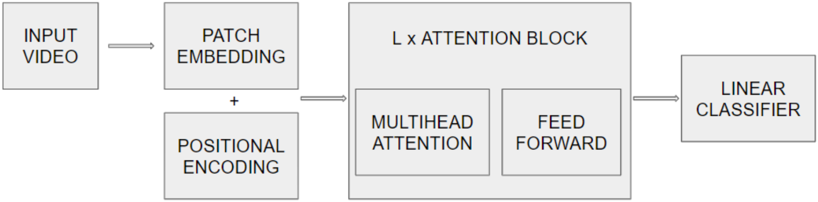


Figure: Transformer architecture

Training configuration

- Loss function: **Cross entropy loss**
- Optimizer: **Adam** (lr=10^{-5})
- 50 epochs with early stopping
- Early stop metric: **Average recall** (recall $= \frac{t_p}{t_p+f_n}$)

Clip resolution	Frames/clip	Batch size	Attn. heads	Attn. Blocks
112×112	100	4	4	2

Table: Model and dataset final configuration

Table of Contents

Training loss evolution

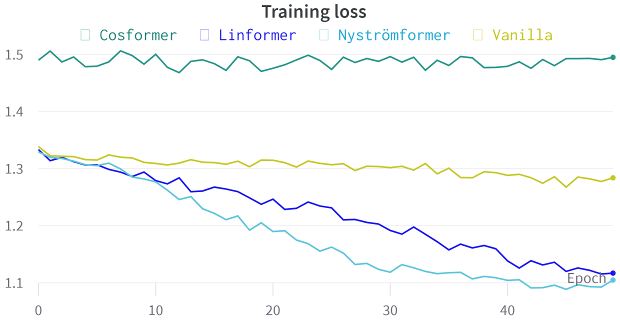


Figure: Training set loss

We can appreciate how Cosformer fails to learn.

Validation set evolution

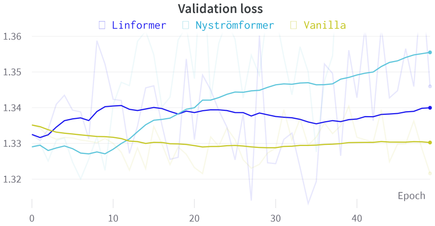


Figure: Training Loss

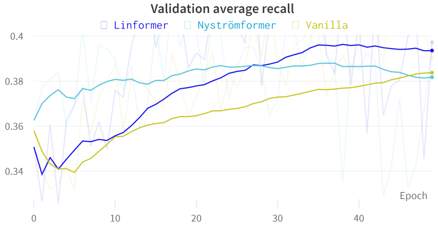


Figure: Validation Loss

- Nyströmformer overfits fast.
- Vanilla learns steadily and slowly.

Test set results

Matthews coefficient is our main criteria

Model	Accuracy	Avg. recall	Matthews	Memory (MB)	FLOPs ($\times 10^9$)
Vanilla	0.40	0.41	0.22	9707	148.05
Nyströmformer	0.41	0.39	0.19	2215	0.98
Linformer	0.41	0.34	0.16	2354	1.45
Cosformer	0.16	0.25	0.0	3512	23.14

Table: Metrics derived from the test set.

- Vanilla has the highest Matthews coefficient.
- From the efficient mechanisms, **Nyströmformer** has the best results.

Model comparison

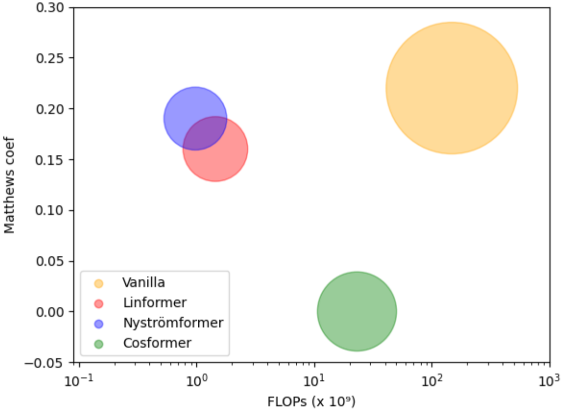


Figure: Performance vs FLOPs plotting. The FLOPs axis is in log scale. The size of the models circles is proportional to their required memory.

Conclusions

- **Nyströmformer:** Achieves the best results. The decision of landmarks may act as sparse attention.
- **Linformer:** Good results but no clear convergence. Its additional projection matrix may help performing visual comprehension tasks
- **Cosformer**: Fails to learn. By design, it may enforce local dependencies, hindering the temporal long term dependencies of videos.

Caution!

We did not perform a training hyperparameter search and our Transformer architecture was limited. No convergence and success of the training guaranteed.

Thanks for your **attention**!

Any questions?