Stop Wasting my FLOPs
Improving the Efficiency of Deep Learning Models

Angelos Katharopoulos

Dec 14th, 2021

Funded by FNSNF

Motivation

Deep neural networks achieve state-of-the-art results in almost all fields of ML.

Motivation

Deep neural networks achieve state-of-the-art results in almost all fields of ML.

However, deep networks are also

- sample inefficient
- overparametrized
- wasting computation

Stop Wasting my FLOPs

- Focus training on "important" samples in the training set
  (Katharopoulos and Fleuret, ICML 2018)

Stop Wasting my FLOPs

- Focus training on “important” samples in the training set
  (Katharopoulos and Fleuret, ICML 2018)
- Focus computation on “important” parts of the samples
  (Katharopoulos and Fleuret, ICML 2019)

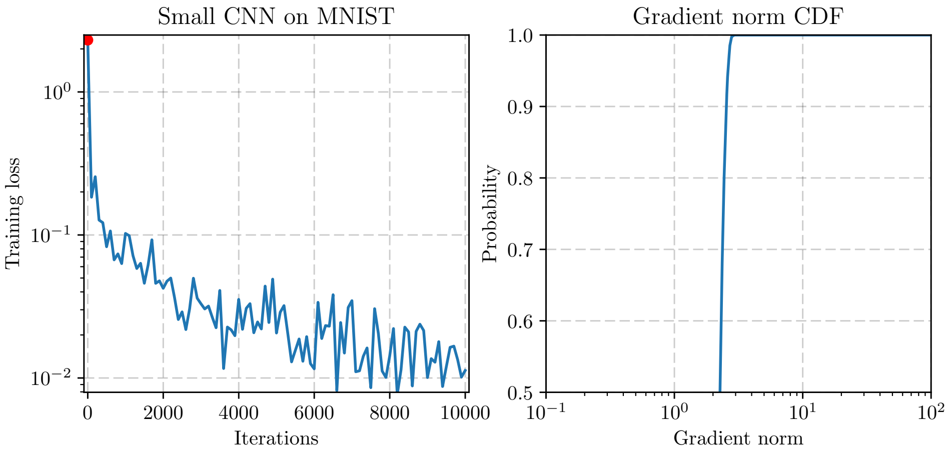
Stop Wasting my FLOPs

- Focus training on "important" samples in the training set
  (Katharopoulos and Fleuret, ICML 2018)
- Focus computation on "important" parts of the samples
  (Katharopoulos and Fleuret, ICML 2019)
- Reduce the computational complexity of self-attention to linear from quadratic
  (Katharopoulos, Vyas, Pappas, and Fleuret, ICML 2020)
- Approximate self-attention using clustering
  (Vyas, Katharopoulos, and Fleuret, NeurIPS 2020)

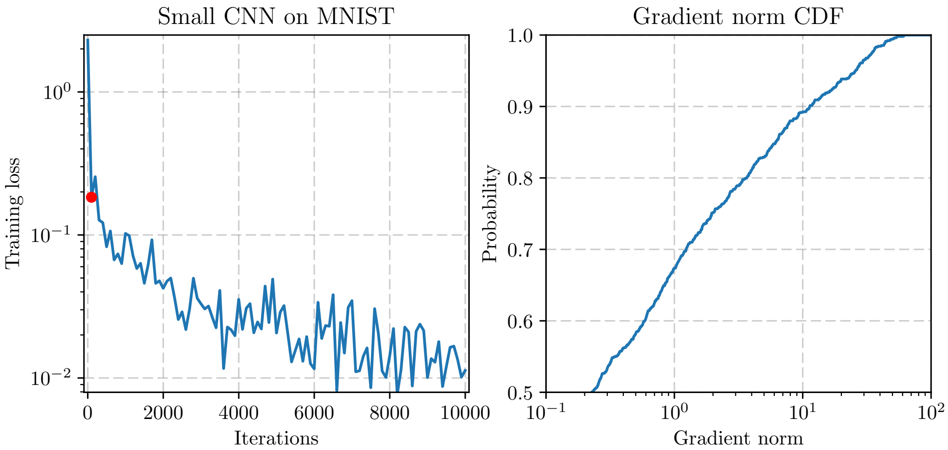
Not All Samples Are Created Equal
Deep Learning with Importance Sampling

(Katharopoulos and Fleuret, ICML 2018)

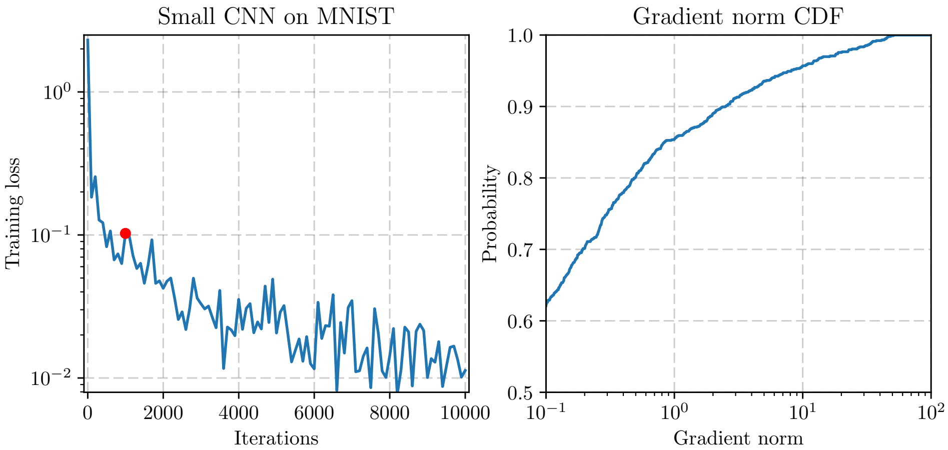
Evolution of gradient norms during training



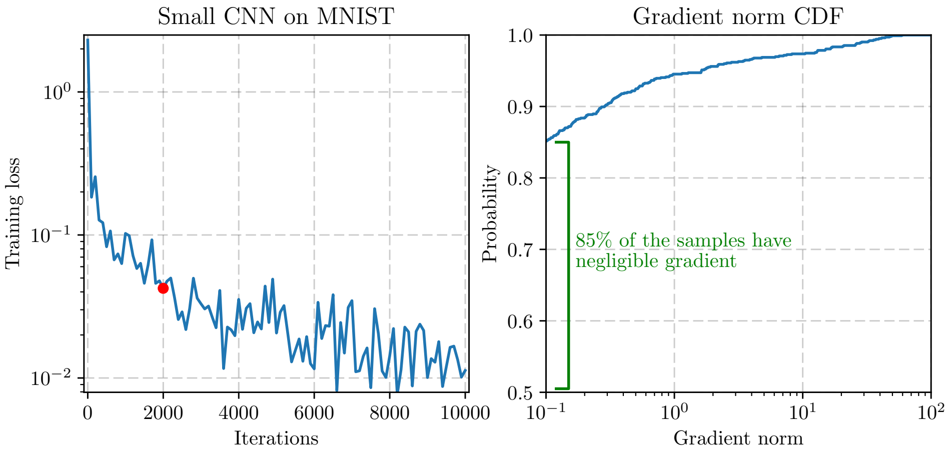
Evolution of gradient norms during training



Evolution of gradient norms during training



Evolution of gradient norms during training



Related work

- Sample points proportionally to the gradient norm (Needell et al., 2014; Zhao and Zhang, 2015; Alain et al., 2015)
- SVRG type methods (Johnson and Zhang, 2013; Defazio et al., 2014; Lei et al., 2017)
- Sample using the loss
  - Hard/Semi-hard sample mining (Schroff et al., 2015; Simo-Serra et al., 2015)
  - Online Batch Selection (Loshchilov and Hutter, 2015)
  - Prioritized Experience Replay (Schaul et al., 2015)

Related work

- ~~Sample points proportionally to the gradient norm~~ (Needell et al., 2014; Zhao and Zhang, 2015; Alain et al., 2015)
- ~~SVRG type methods~~ (Johnson and Zhang, 2013; Defazio et al., 2014; Lei et al., 2017)
- Sample using the loss
  - Hard/Semi-hard sample mining (Schroff et al., 2015; Simo-Serra et al., 2015)
  - Online Batch Selection (Loshchilov and Hutter, 2015)
  - Prioritized Experience Replay (Schaul et al., 2015)

Contributions

- Derive a fast to compute importance distribution
- Variance cannot always be reduced so start importance sampling when it is useful

Contributions

- Derive a fast to compute importance distribution
- Variance cannot always be reduced so start importance sampling when it is useful

- Package everything in an embarassingly simple to use library BONUS

Deriving the sampling distribution (1)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

$$P^* = \arg\min_P \mathrm{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \arg\min \mathbb{E}_P\left[w_i^2 \left\|G_i\right\|_2^2\right]$$

To simplify, we minimize an upper bound

$$\left\|G_i\right\|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P\left[w_i^2 \left\|G_i\right\|_2^2\right] \leq \min_P \mathbb{E}_P\left[w_i^2 \hat{G}_i^{\,2}\right]$$

Deriving the sampling distribution (1)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

$$P^* = \arg\min_P \operatorname{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \arg\min \mathbb{E}_P\left[w_i^2 \left\| G_i \right\|_2^2\right]$$

To simplify, we minimize an upper bound

$$\left\| G_i \right\|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P\left[w_i^2 \left\| G_i \right\|_2^2\right] \leq \min_P \mathbb{E}_P\left[w_i^2 \hat{G}_i^{\;2}\right]$$

Deriving the sampling distribution (1)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

$$P^* = \arg\min_P \operatorname{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \arg\min \mathbb{E}_P\left[{w_i}^2 \left\| G_i \right\|_2^2\right]$$

To simplify, we minimize an upper bound

$$\left\| G_i \right\|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P\left[{w_i}^2 \left\| G_i \right\|_2^2\right] \leq \min_P \mathbb{E}_P\left[{w_i}^2 \hat{G}_i^{\,2}\right]$$

Deriving the sampling distribution (1)

Similar to Zhao and Zhang (2015) we want to minimize the variance of the gradients.

$$P^* = \arg\min_P \operatorname{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \arg\min \mathbb{E}_P\left[w_i^2 \left\|G_i\right\|_2^2\right]$$

To simplify, we minimize an upper bound

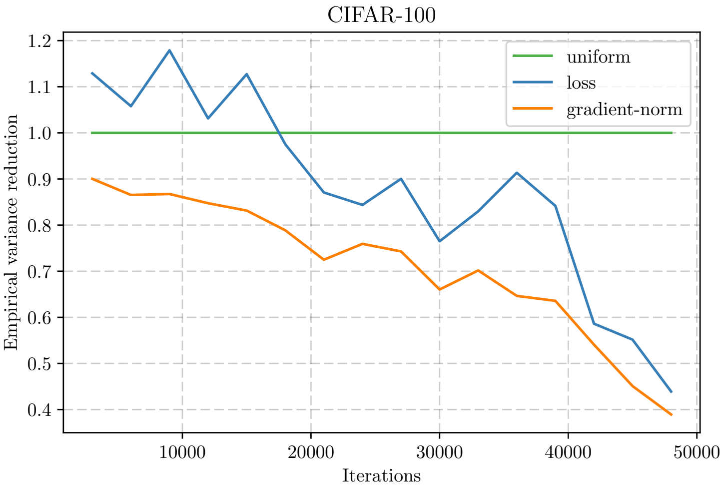
$$\left\|G_i\right\|_2 \leq \hat{G}_i \iff \min_P \mathbb{E}_P\left[w_i^2 \left\|G_i\right\|_2^2\right] \leq \min_P \mathbb{E}_P\left[w_i^2 \hat{G}_i^{\,2}\right]$$

Deriving the sampling distribution (2)

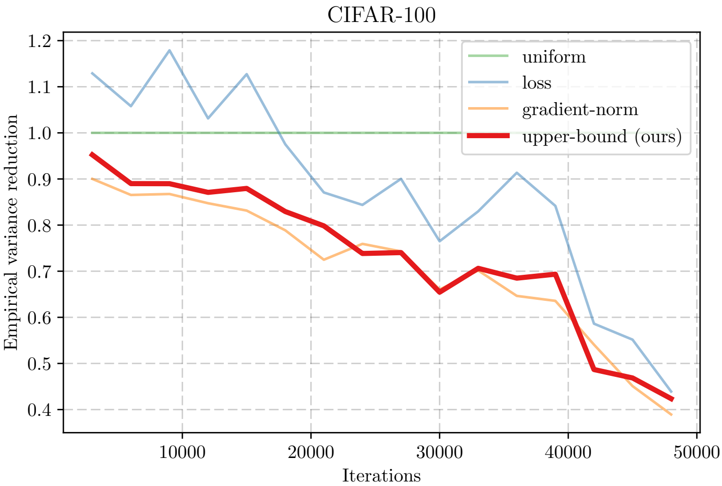
We show that we can upper bound the gradient norm of the parameters using the norm of the gradient with respect to the pre-activation outputs of the last layer.

We conjecture that batch normalization and weight initialization make it tight.

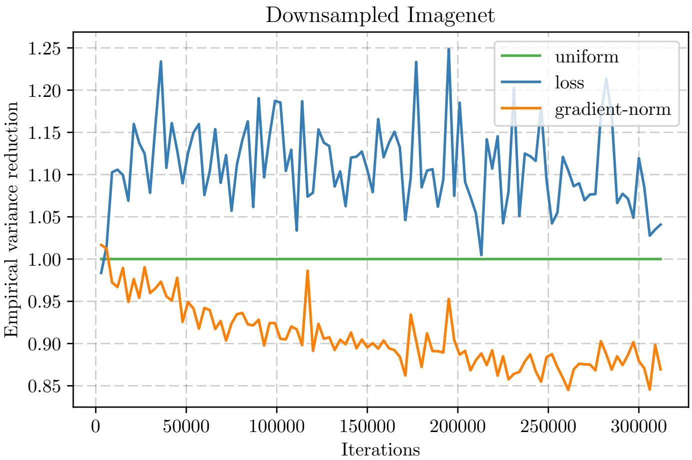
Variance reduction achieved with our upper-bound



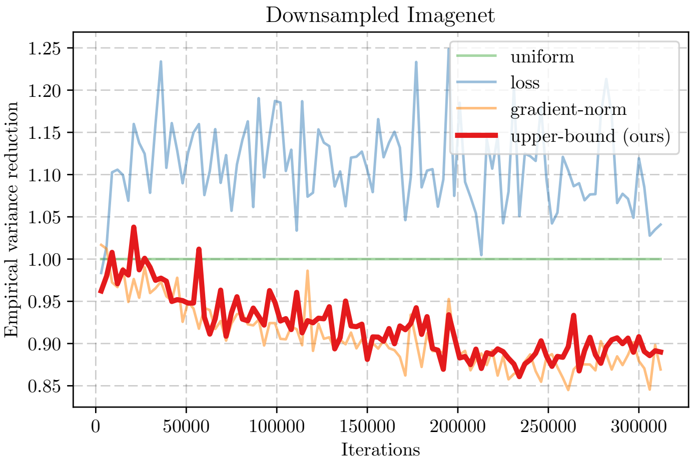
Variance reduction achieved with our upper-bound



Variance reduction achieved with our upper-bound



Variance reduction achieved with our upper-bound



Is the upper-bound enough to speed up training?

Not really, because

- a forward pass on the whole dataset is still prohibitive
- the importance distribution can be arbitrarily close to uniform

Two key ideas

- Sample a **large batch** (B) randomly and resample a **small batch** (b) with importance
- Start importance sampling when the variance will be reduced

When do we start importance sampling?

We start importance sampling when the variance reduction is large enough

$$\mathrm{Tr}\left(\mathbb{V}_u[G_i]\right) - \mathrm{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \frac{1}{B}\sum_{i=1}^{B}\|G_i\|_2^2 \sum_{i=1}^{B}(p_i - u)^2 \propto \underbrace{\sum_{i=1}^{B}(p_i - u)^2}_{\substack{\text{distance of importance}\\ \text{distribution to uniform}}}$$

When do we start importance sampling?

We start importance sampling when the variance reduction is large enough

$$\mathrm{Tr}\left(\mathbb{V}_u[G_i]\right) - \mathrm{Tr}\left(\mathbb{V}_P[w_i G_i]\right) = \frac{1}{B}\sum_{i=1}^{B}\|G_i\|_2^2 \sum_{i=1}^{B}(p_i - u)^2 \propto \underbrace{\sum_{i=1}^{B}(p_i - u)^2}_{\text{distance of importance distribution to uniform}}$$

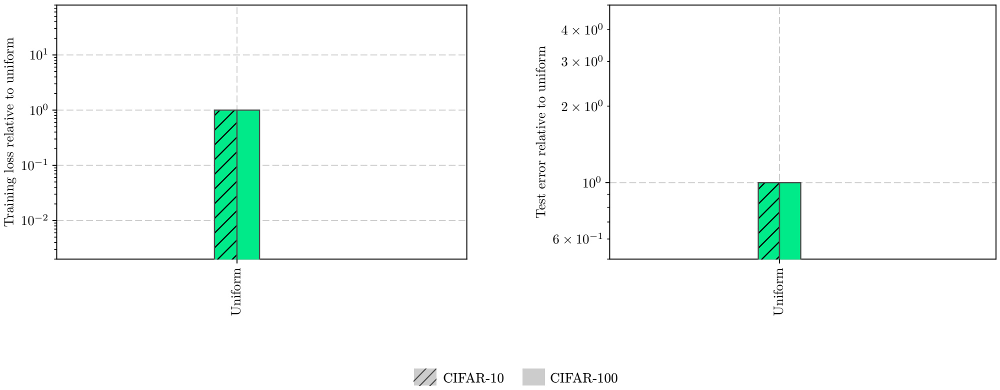
We show that the **equivalent batch increment** $\tau \geq \left(1 - \frac{\sum_i (p_i - u)^2}{\sum_i p_i^2}\right)^{-1}$ which allows us to perform importance sampling when

$$\underbrace{B t_{\text{forward}} + b\left(t_{\text{forward}} + t_{\text{backward}}\right)}_{\text{Time for } \textbf{importance sampling iteration}} \leq \underbrace{\tau\left(t_{\text{forward}} + t_{\text{backward}}\right) b}_{\text{Time for equivalent } \textbf{uniform sampling iteration}}$$

Experimental setup

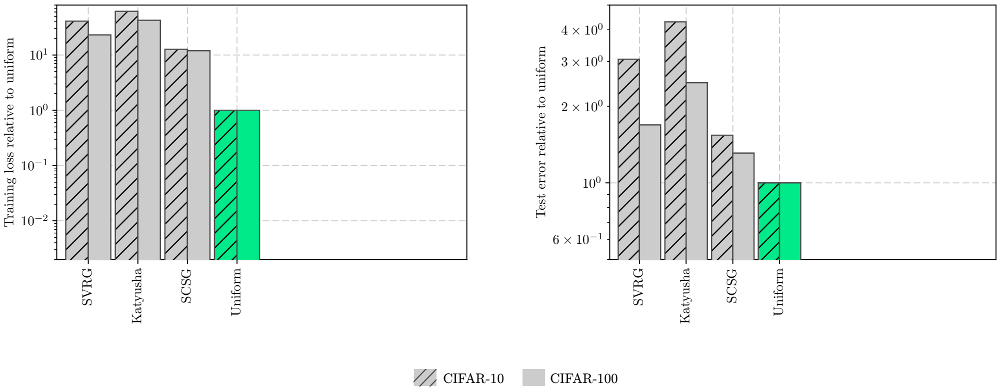
- We fix a time budget for all methods and compare the achieved training loss and test error
- We evaluate on image classification on CIFAR-10/100

Importance sampling for image classification



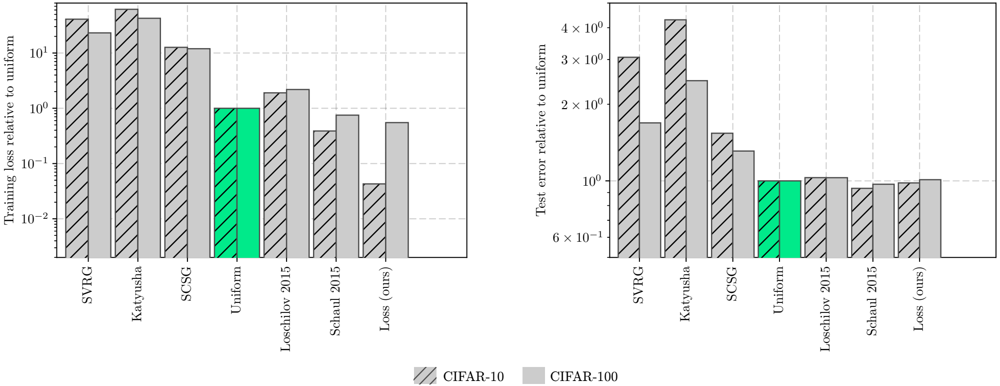
Importance sampling for image classification

- SVRG methods do not work for Deep Learning



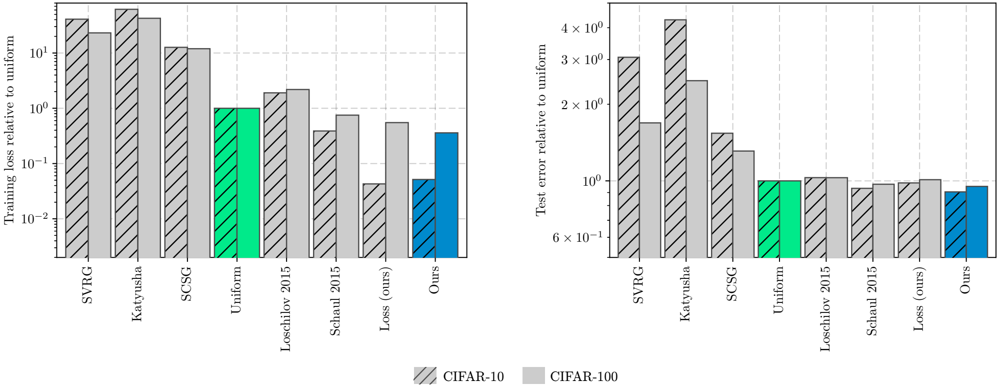
Importance sampling for image classification

- SVRG methods do not work for Deep Learning
- Our loss-based sampling outperfoms existing loss based methods



Importance sampling for image classification

- SVRG methods do not work for Deep Learning
- Our loss-based sampling outperfoms existing loss based methods
- Improvement from **3**× **to 10**× compared to training loss with uniform sampling

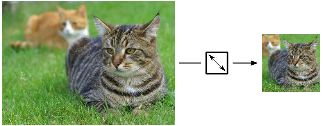


Processing Megapixel Images with Deep Attention-Sampling Models

(Katharopoulos and Fleuret, ICML 2019)

How do DNNs process large images?

Cropping and downsampling to a manageable resolution (e.g. 224×224)



Dividing the image into patches and processing them separately

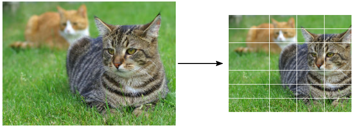


*image taken from the Imagenet dataset

Our contributions

- **Sample from a soft attention** to only process a **fraction of the image** in high resolution.
- Derive **gradients through the sampling** for all parameters which allows to train our models end-to-end.
- Disentangle the computational and memory requirements from the input resolution.

Soft Attention

Given an input x we define a neural network $\Psi(x)$ that uses attention

$$\Psi(x) = g\left(\sum_{i=1}^{K} a(x)_i f(x)_i\right) = g\left(\mathbb{E}_{I \sim a(x)}[f(x)_I]\right),$$

where $f(x) \in \mathbb{R}^{K \times D}$ are the features and $a(x) \in \mathbb{R}_+^K$ is the attention distribution.

Attention Sampling

We approximate $\Psi(x)$ by Monte Carlo

$$\Psi(x) \approx g\left(\frac{1}{N}\sum_{q \in Q} f(x)_q\right) \text{ where } Q = \{q_i \sim a(x) \mid i \in \{1, 2, \ldots, N\}\}.$$

We show that

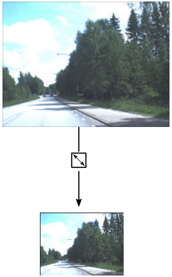
- Sampling from the attention is optimal to approximate $\Psi(x)$ if $\|f(x)_i\| = \|f(x)_j\| \; \forall \, i, j$
- We can compute the gradients both for the parameters $a(\cdot)$ and $f(\cdot)$

Processing Megapixel Images with Deep Attention-Sampling Models

Image x

Processing Megapixel Images with Deep Attention-Sampling Models

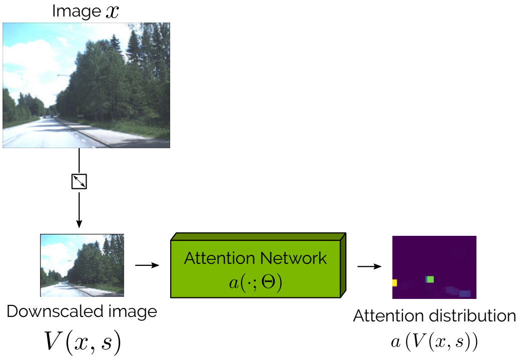
Image x



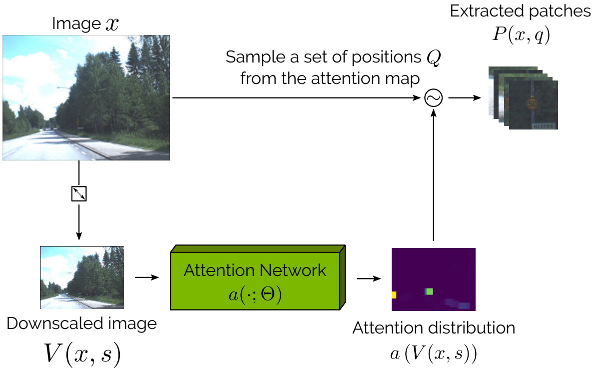
Downscaled image

$V(x, s)$

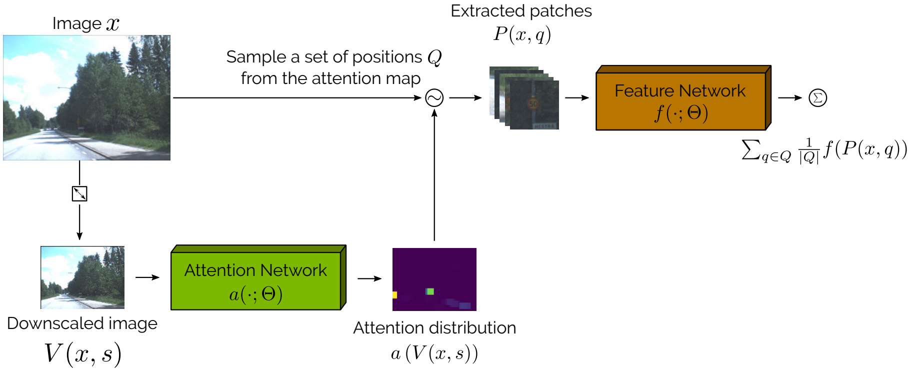
Processing Megapixel Images with Deep Attention-Sampling Models



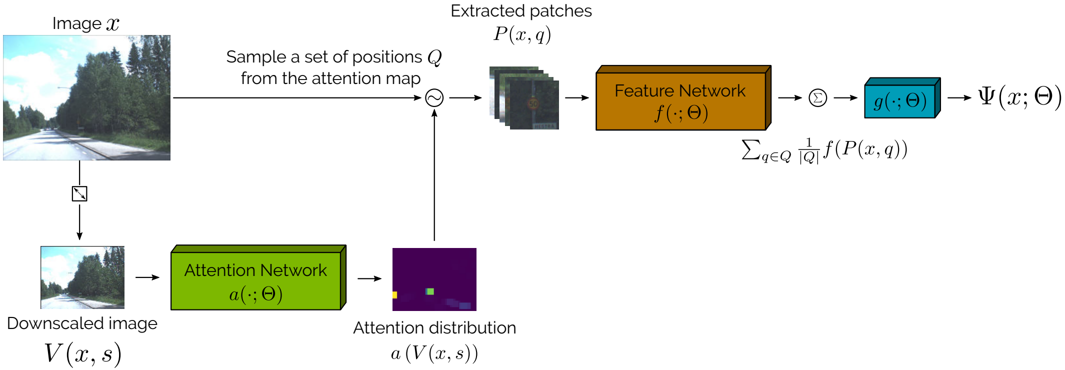
Processing Megapixel Images with Deep Attention-Sampling Models



Processing Megapixel Images with Deep Attention-Sampling Models



Processing Megapixel Images with Deep Attention-Sampling Models



Experiments

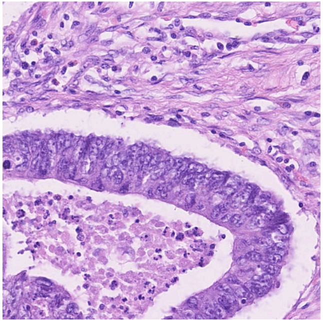
Baselines

- Attention-Based Deep Multiple Instance Learning (Ilse et al., 2018)
- Shallow ResNets at various input scales

Datasets

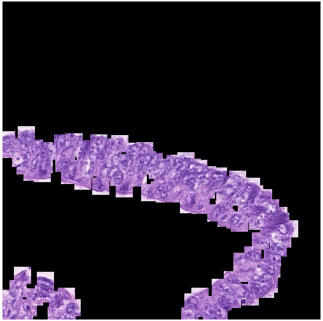
- Histopathology dataset for detecting images that contain epithelial cells (Sirinukunwattana et al., 2016)
- Speed limit sign detection (Larsson and Felsberg, 2011)

Qualitative evaluation of the attention distribution (1)



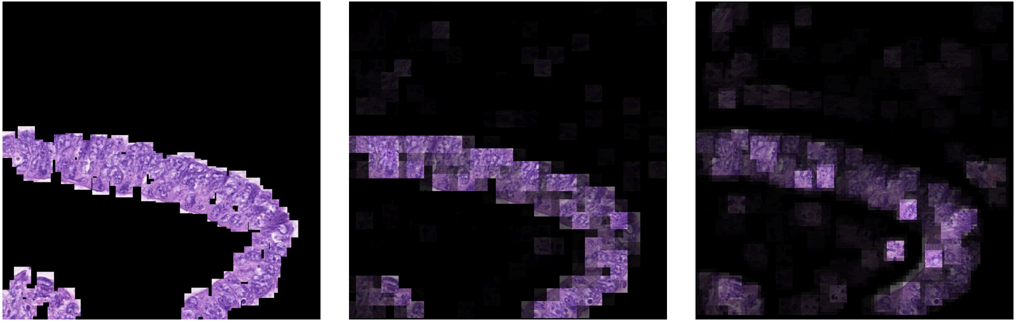
Full Image

Qualitative evaluation of the attention distribution (1)



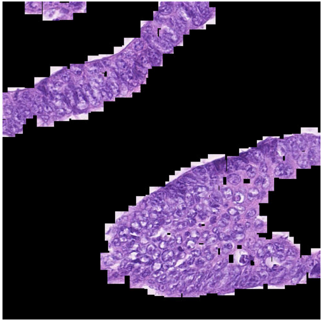
Epithelial Cells

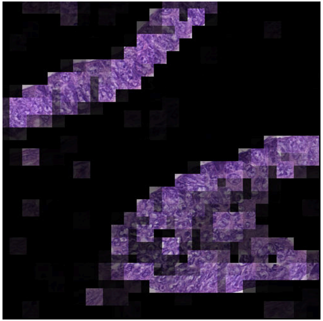
Qualitative evaluation of the attention distribution (1)

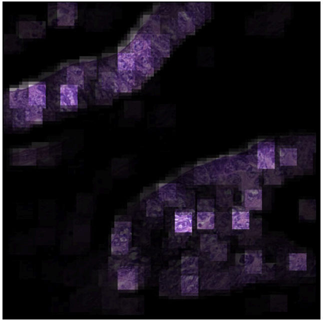


Epithelial Cells   Ilse et al. (2018)   Attention Sampling

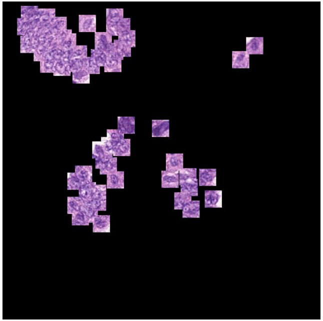
Qualitative evaluation of the attention distribution (2)


Epithelial Cells

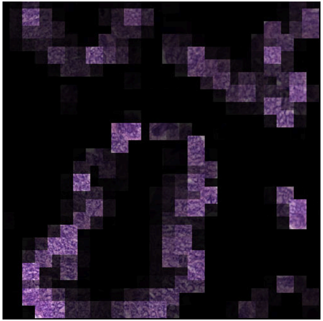

Ilse et al. (2018)


Attention Sampling

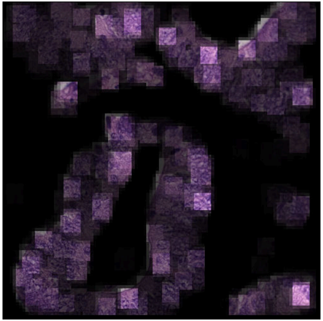
Qualitative evaluation of the attention distribution (3)



Epithelial Cells



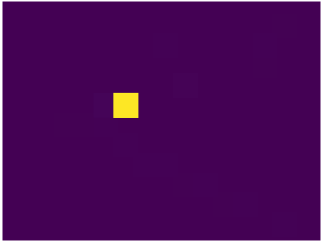
Ilse et al. (2018)



Attention Sampling

Qualitative evaluation of the attention distribution (4)

Ground Truth

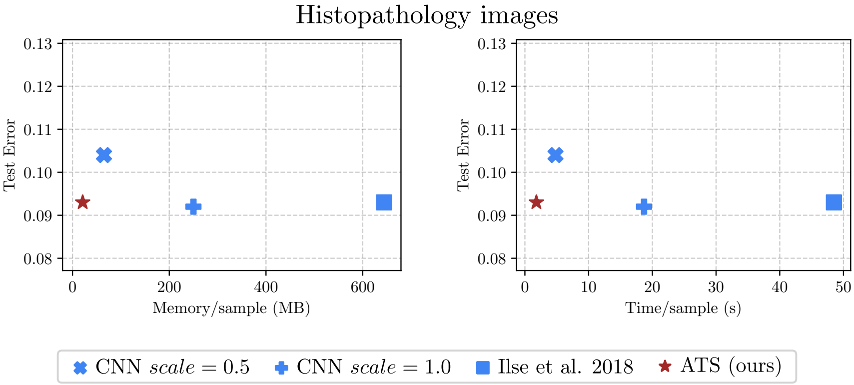


Ilse et al. (2018)

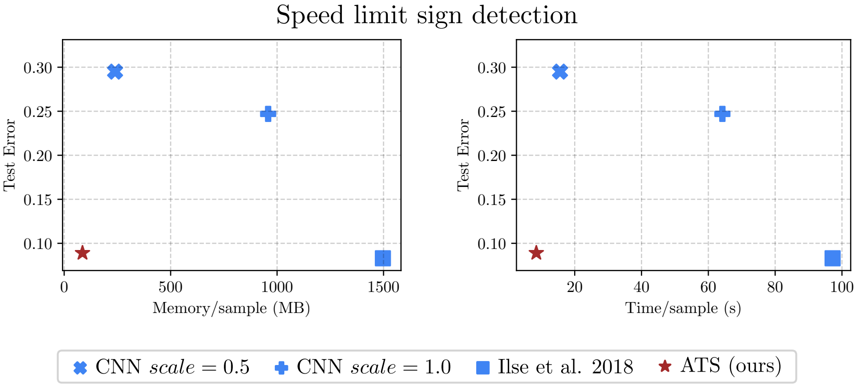
Attention Sampling

Extracted patch

Quantitative evaluation of attention sampling (1)



Quantitative evaluation of attention sampling (2)



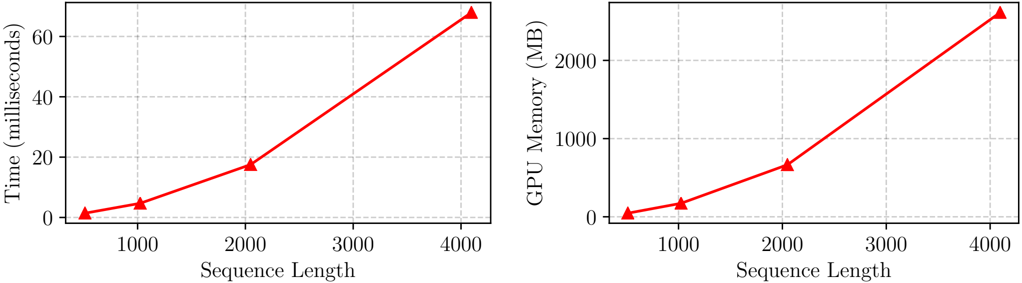
Scaling transformers to large sequences

Using kernels and clustering

(Katharopoulos, Vyas, Pappas, and Fleuret, ICML 2020)
(Vyas, Katharopoulos, and Fleuret, NeurIPS 2020)

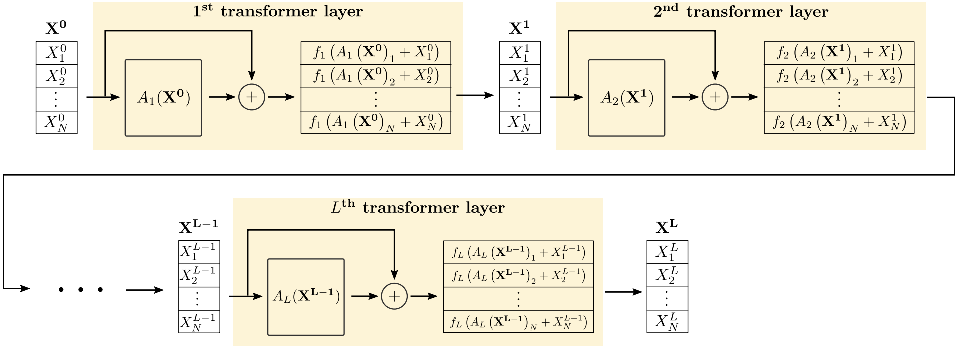
Transformers are hard to scale

Self-attention **computation and memory scales** as $\mathcal{O}\left(N^2\right)$ with respect to the **sequence length**.

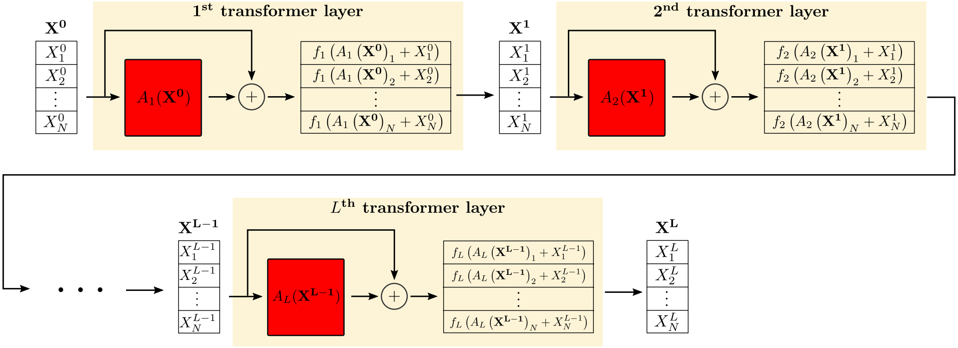


A single self-attention layer in an NVIDIA GTX 1080 Ti

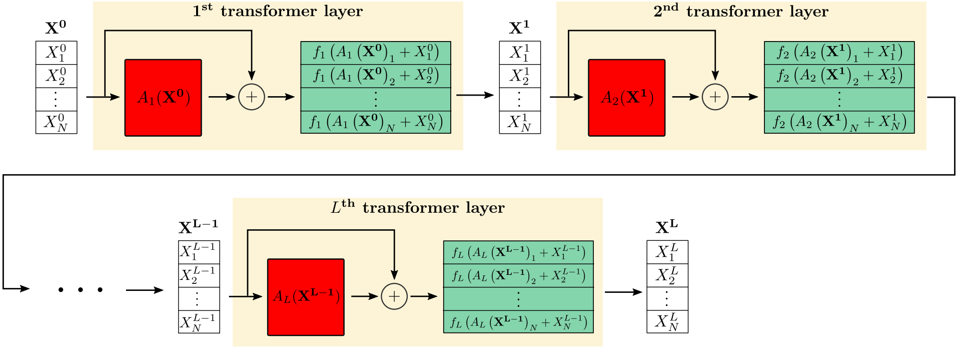
Definition of a transformer



Definition of a transformer



Definition of a transformer



Self-Attention

The commonly used attention mechanism is the scaled dot product attention

$$
\begin{aligned}
Q &= XW_Q \\
K &= XW_K \\
V &= XW_V \\
A_l(X) = V' &= \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V
\end{aligned}
$$

Self-Attention

The commonly used attention mechanism is the scaled dot product attention

$$
\begin{aligned}
Q &= XW_Q \\
K &= XW_K \\
V &= XW_V \\
A_l(X) = V' &= \text{softmax}\left(\boxed{\frac{QK^T}{\sqrt{D}}}\right)V
\end{aligned}
$$

Quadratic complexity

Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention

- A transformer model with **linear complexity** both for memory and computation **during training**
- A transformer model with **linear computational complexity and constant memory** for **autoregressive inference**
- Unravel the **relation between transformers and RNNs**

Linear Attention

What if we write the self-attention using an **arbitrary similarity score?**

$$V'_i = \frac{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right)}$$

Linear Attention

What if this similarity is a kernel, namely $\mathrm{sim}(a, b) = \phi(a)^T \phi(b)$?

$$V_i' = \frac{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j)\, V_j}{\sum_{j=1}^{N} \mathrm{sim}(Q_i, K_j)}$$

Kernelization

$$= \frac{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)\, V_j}{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)}$$

Linear Attention

Matrix products are associative which makes the attention computation $\mathcal{O}(N)$ with respect to the sequence length.

$$
\begin{aligned}
V_i' &= \frac{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right)} \\
&\text{Kernelization} \\
&= \frac{\sum_{j=1}^{N} \phi\left(Q_i\right)^T \phi\left(K_j\right) V_j}{\sum_{j=1}^{N} \phi\left(Q_i\right)^T \phi\left(K_j\right)} \\
&\text{Associativity property} \\
&= \frac{\phi\left(Q_i\right)^T \sum_{j=1}^{N} \phi\left(K_j\right) V_j^T}{\phi\left(Q_i\right)^T \sum_{j=1}^{N} \phi\left(K_j\right)}
\end{aligned}
$$

Causal Masking

Causal masking is used to efficiently train autoregressive transformers.

Causal Masking

Causal masking is used to efficiently train autoregressive transformers.

Non-autoregressive

$$V_i' = \frac{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{N} \operatorname{sim}\left(Q_i, K_j\right)}$$

Autoregressive

$$V_i' = \frac{\sum_{j=1}^{i} \operatorname{sim}\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{i} \operatorname{sim}\left(Q_i, K_j\right)}$$

Causal Masking

Causal masking is used to efficiently train autoregressive transformers.

Non-autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \sum_{j=1}^N \phi\left(K_j\right) V_j^T}{\phi\left(Q_i\right)^T \sum_{j=1}^N \phi\left(K_j\right)}$$

Autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \sum_{j=1}^i \phi\left(K_j\right) V_j^T}{\phi\left(Q_i\right)^T \sum_{j=1}^i \phi\left(K_j\right)}$$

Causal Masking

Causal masking is used to efficiently train autoregressive transformers.

Non-autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \overbrace{\sum_{j=1}^{N} \phi\left(K_j\right) V_j^T}^{S}}{\phi\left(Q_i\right)^T \underbrace{\sum_{j=1}^{N} \phi\left(K_j\right)}_{Z}}$$

Autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \overbrace{\sum_{j=1}^{i} \phi\left(K_j\right) V_j^T}^{S_i}}{\phi\left(Q_i\right)^T \underbrace{\sum_{j=1}^{i} \phi\left(K_j\right)}_{Z_i}}$$

Causal Masking

Causal masking is used to efficiently train autoregressive transformers.

Non-autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \overbrace{\sum_{j=1}^{N} \phi\left(K_j\right) V_j^T}^{S}}{\phi\left(Q_i\right)^T \underbrace{\sum_{j=1}^{N} \phi\left(K_j\right)}_{Z}}$$

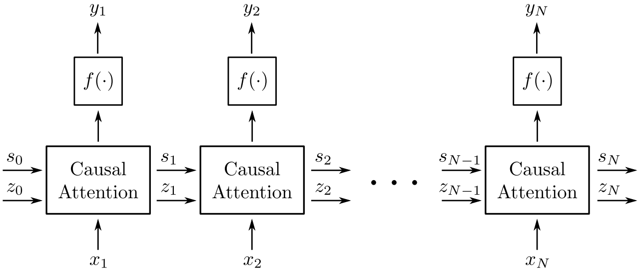
Autoregressive

$$V_i' = \frac{\phi\left(Q_i\right)^T \overbrace{\sum_{j=1}^{i} \phi\left(K_j\right) V_j^T}^{S_i}}{\phi\left(Q_i\right)^T \underbrace{\sum_{j=1}^{i} \phi\left(K_j\right)}_{Z_i}}$$

Naive computation of S_i and Z_i results in quadratic complexity.

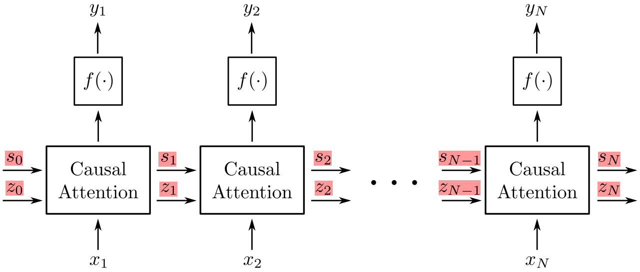
Transformers are RNNs

Autoregressive transformers can be written as a function that **receives an input** x_i, **modifies the internal state** $\{s_{i-1}, z_{i-1}\}$ and **predicts an output** y_i.



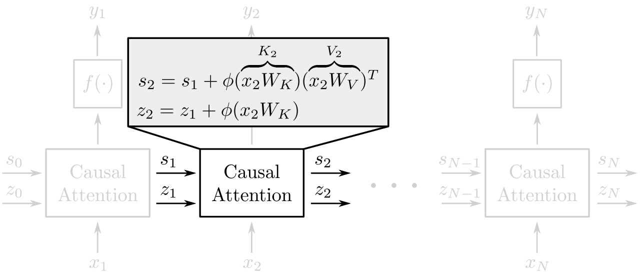
Transformers are RNNs

Autoregressive transformers can be written as a function that **receives an input** x_i, **modifies the internal state** $\{s_{i-1}, z_{i-1}\}$ and **predicts an output** y_i.



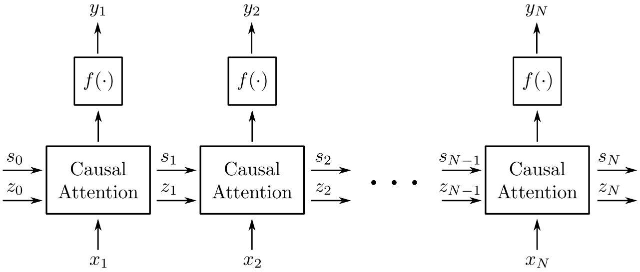
Transformers are RNNs

Autoregressive transformers can be written as a function that **receives an input** x_i, **modifies the internal state** $\{s_{i-1}, z_{i-1}\}$ and **predicts an output** y_i.



Transformers are RNNs

Autoregressive transformers can be written as a function that **receives an input** x_i, **modifies the internal state** $\{s_{i-1}, z_{i-1}\}$ and **predicts an output** y_i.



Autoregressive inference with **linear complexity and constant memory**.

Practical implications

- Our **theoretical analysis holds for all transformers** even when using infinite dimensional feature maps
- We need a simple **finite dimensional feature map** to speed up computation
- We **derive the gradients as cumulative sums** which allows for a significant speed-up

Experimental setup

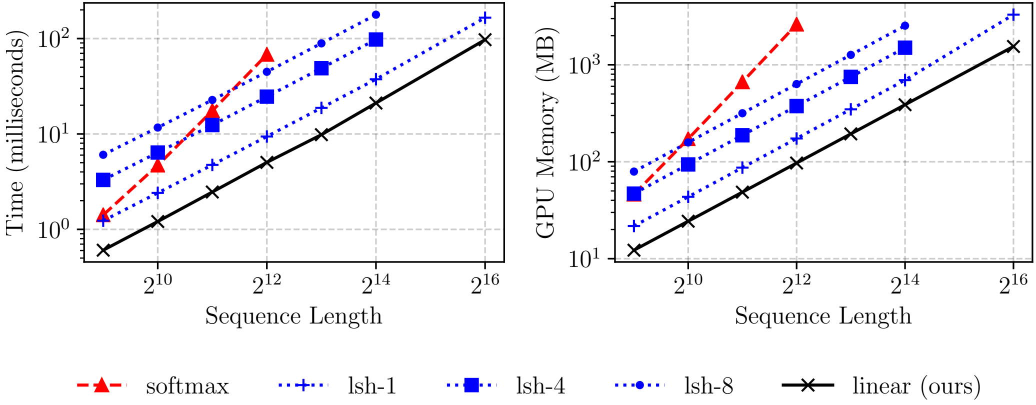
Baselines

- Softmax transformer (Vaswani et al., 2017)
- LSH attention from Reformer (Kitaev et al., 2020)

Experiments

- Artificial benchmark for computational and memory requirements
- Autoregressive image generation on MNIST and CIFAR-10

Benchmark


Time (milliseconds)
10^0
10^1
10^2
2^{10}
2^{12}
2^{14}
2^{16}
Sequence Length
GPU Memory (MB)
10^1
10^2
10^3
2^{10}
2^{12}
2^{14}
2^{16}
Sequence Length
softmax
lsh-1
lsh-4
lsh-8
linear (ours)

Autoregressive image generation

Unconditional samples after 250 epochs on MNIST

Ours (0.644 bpd)

Softmax (0.621 bpd)

LSH-1 (0.745 bpd)

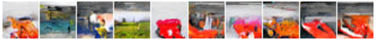
LSH-4 (0.676 bpd)

Unconditional samples after 1 GPU week on CIFAR-10

Ours (3.40 bpd)

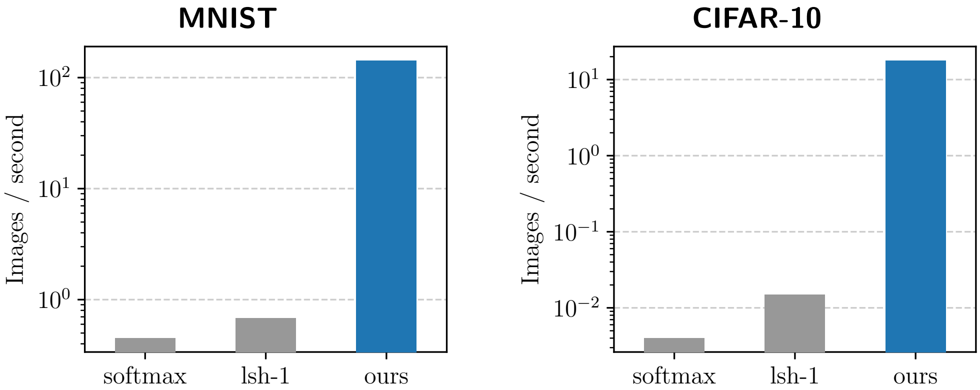
Softmax (3.47 bpd)

LSH-1 (3.39 bpd)

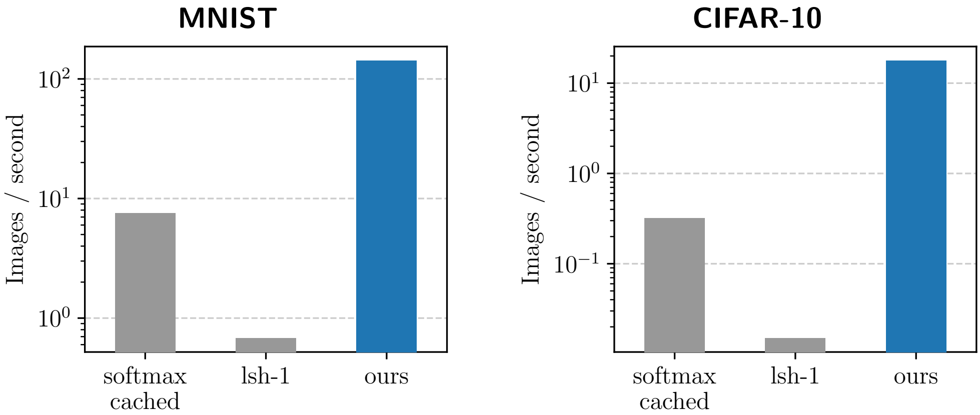


LSH-4 (3.51 bpd)

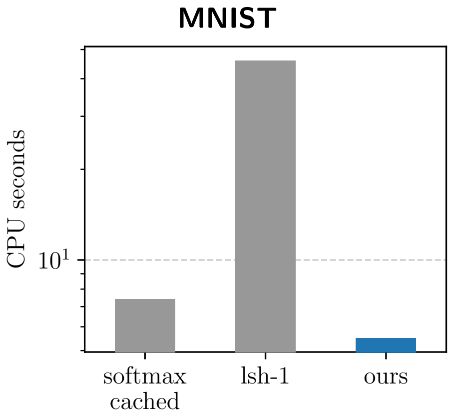
Autoregressive image generation throughput

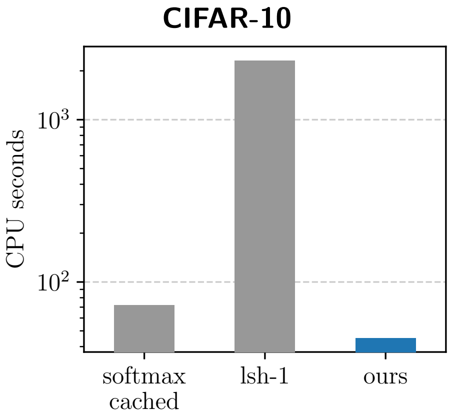


Autoregressive image generation throughput



Autoregressive image generation latency





Summary

- **Kernel feature maps** and **associativity of matrix products** yield an attention with linear complexity.
- Computing the key value matrix as a **cumulative sum** extends our efficient attention computation to the autoregressive case
- Using the RNN formulation to perform autoregressive inference requires **constant memory** and is **many times faster**

Summary

- **Kernel feature maps** and **associativity of matrix products** yield an attention with linear complexity.
- Computing the key value matrix as a **cumulative sum** extends our efficient attention computation to the autoregressive case
- Using the RNN formulation to perform autoregressive inference requires **constant memory** and is **many times faster**

Linear transformers are **backwards incompatible** to softmax transformers.

Fast Transformers with Clustered Attention

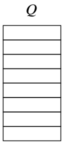
- A fast **approximation of self-attention** by clustering the queries
- Linear computational and memory complexity for a fixed number of clusters
- Approximation of pretrained transformers **without finetuning and without loss in performance**

Softmax approximation

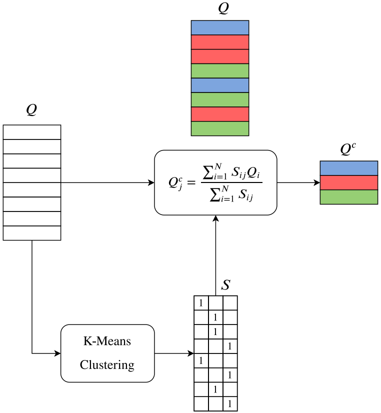
Given Q_i and Q_j such that $\|Q_i - Q_j\|_2 \leq \epsilon$ then

$$\left\| \operatorname{softmax}\left(Q_i K^T\right) - \operatorname{softmax}\left(Q_j K^T\right) \right\|_2 \leq \epsilon \|K\|_2$$

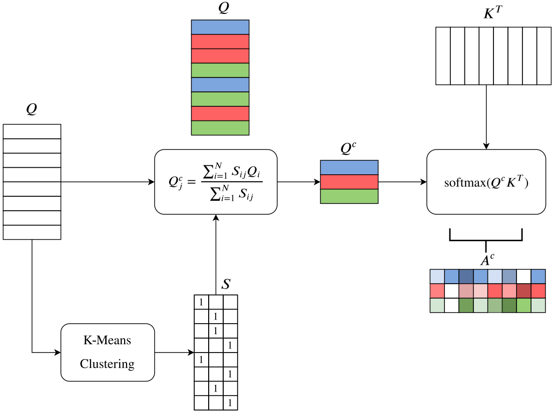
Clustered attention



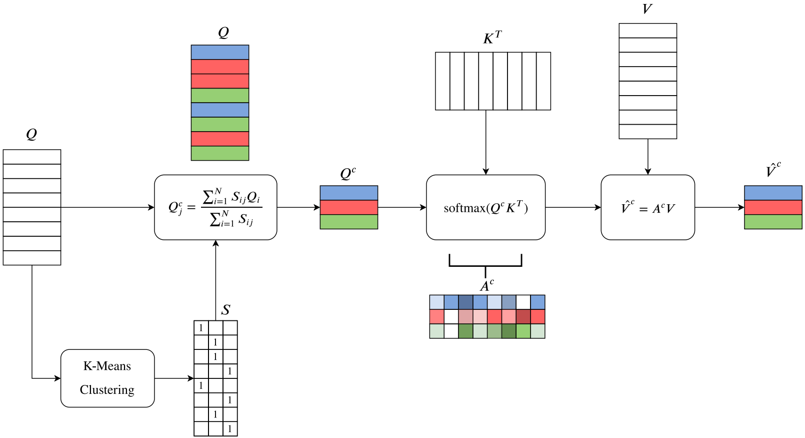
Clustered attention



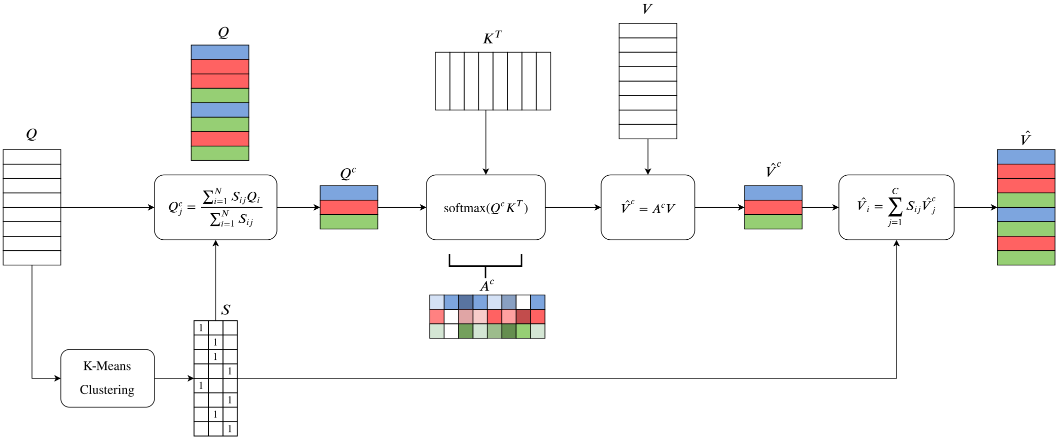
Clustered attention



Clustered attention



Clustered attention



Improved Clustered Attention

For a single **query** Q_i and its **corresponding** cluster **centroid** Q_j^c, standard attention is approximated as:

$$A_i = \text{softmax}\left(Q_i K^T\right) \approx \text{softmax}\left(Q_j^c K^T\right) = A_i^c$$

Improved Clustered Attention

For a single **query** Q_i and its **corresponding** cluster **centroid** Q_j^c, standard attention is approximated as:

$$A_i = \text{softmax}\left(Q_i K^T\right) \approx \text{softmax}\left(Q_j^c K^T\right) = A_i^c$$

Using even **a few exact dot products** improves this approximation.

Improved Clustered Attention

Given a set of key indices $T = \{k_1, k_2, \dots\}$

$$A^t_{ik} = \begin{cases} w \frac{\exp Q_i K_k^T}{\sum_{r \in T} \exp Q_i K_r^T} & k \in T \\ A^c_{ik} & k \notin T \end{cases}$$

Finally, we show that $|A - A^c|_1 \geq |A - A^t|_1$ which improves our previous approximation.

Experimental setup

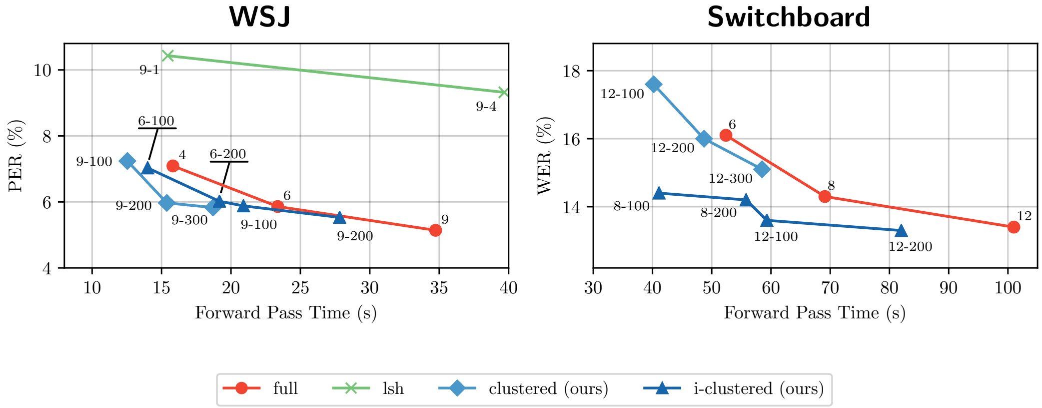
Baselines

- Softmax transformer (Vaswani et al., 2017)
- LSH attention from Reformer (Kitaev et al., 2020)
- FAVOR random Fourier features from Performer (Choromanski et al., 2020)

Experiments

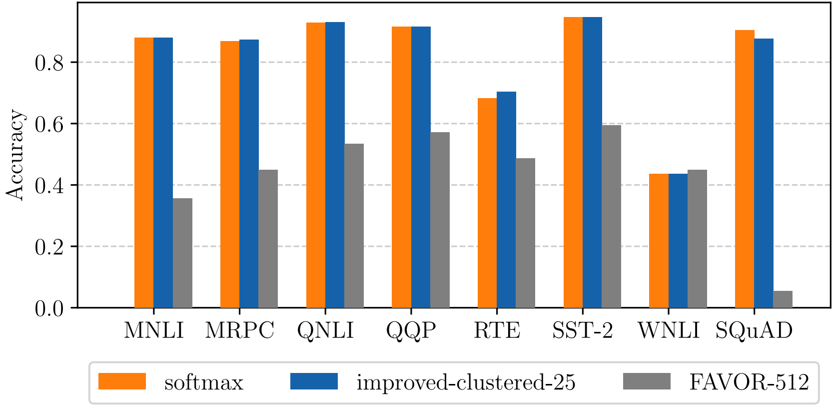
- Automatic speech recognition on WSJ and Switchboard
- Approximation of pretrained RoBERTa on GLUE and SQuAD
- Approximation of pretrained Wav2Vec

Automatic Speech Recognition



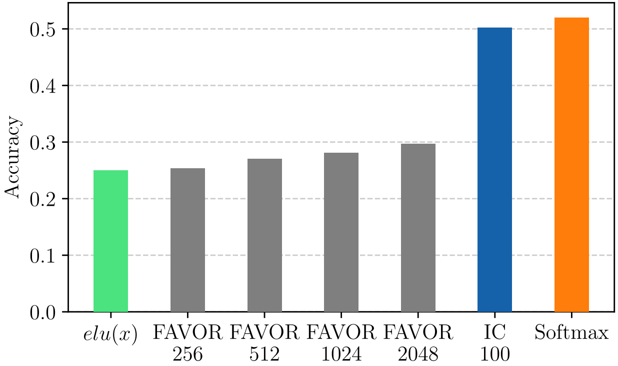
RoBERTa approximation

RoBERTa approximation on GLUE and SQuAD benchmarks with **25 clusters**.



Wav2Vec approximation

Wav2Vec approximation on LibriSpeech.



Conclusions & Future research directions

Conclusions

Goal: Remove unnecessary computation from Deep Networks

Conclusions

Goal: Remove unnecessary computation from Deep Networks

- **Avoid computing zero gradients** with importance sampling to select informative data points

Conclusions

Goal: Remove unnecessary computation from Deep Networks

- **Avoid computing zero gradients** with importance sampling to select informative data points
- **Avoid computing features** for parts of the input **that do not contribute** to the prediction using attention sampling

Conclusions

Goal: Remove unnecessary computation from Deep Networks

- **Avoid computing zero gradients** with importance sampling to select informative data points
- **Avoid computing features** for parts of the input **that do not contribute** to the prediction using attention sampling
- **Avoid computing** all elements of **the attention matrix** using
  1. kernelized linear attention that never computes an attention matrix
  2. clustering to group the computations

Future research directions

- Increasing the representation capacity of linear attention models

No one model works best for all possible situations.
– No Free Lunch Theorem

Future research directions

- Increasing the representation capacity of linear attention models
- Learnable and GPU friendly sparsity

> *A research idea wins because it is suited to the available software and hardware and not because the idea is superior.*
>
> *– Sarah Hooker, The Hardware Lottery*

Future research directions

- Increasing the representation capacity of linear attention models
- Learnable and GPU friendly sparsity
- Efficient transformers enable new applications to computer vision and multi-modal training

Thank you for your time!

References I

A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018. URL http://idiap.ch/~katharas/pdfs/is-icml.pdf.

A. Katharopoulos and F. Fleuret. Processing megapixel images with deep attention-sampling models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. URL http://idiap.ch/~katharas/pdfs/ats-icml.pdf.

A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/pdf/2006.16236.pdf.

A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. *arXiv preprint arXiv:2007.04825*, 2020.

References II

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2014.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1–9, 2015.

Guillaume Alain, Alex Lamb, Chinnadhurai Sankar, Aaron Courville, and Yoshua Bengio. Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

References III

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2345–2355, 2017.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 118–126. IEEE, 2015.

References IV

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Maximilian Ilse, Jakub M Tomczak, and Max Welling. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

Korsuk Sirinukunwattana, Shan e Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Trans. Med. Imaging*, 35(5):1196–1206, 2016.

Fredrik Larsson and Michael Felsberg. Using fourier descriptors and spatial models for traffic sign recognition. In *Scandinavian conference on image analysis*, pages 238–249. Springer, 2011.

References V

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.