Perceiver: General Perception with Iterative Attention

by DeepMind

GLAMI AI
Radek Bartyzal
13. 4. 2021

Motivation

Current state:

- NN architectures usually differ based on the input modality = audio, image, video, …
- e.g. CNNs and Visual Transformers rely on image specific locality assumption
  - => convolutions, splitting the image into a grid

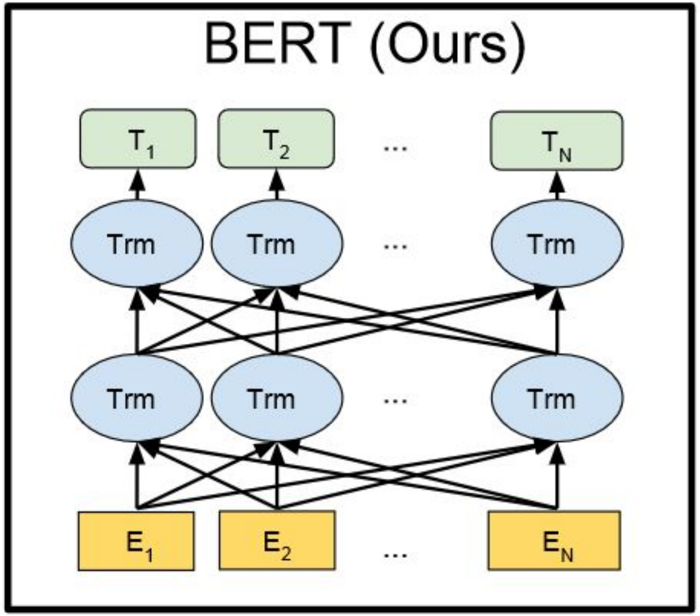
Author’s goal:

- create competitive architecture without relying on these assumptions
  - => attend to the individual pixels in an image
  - => use same architecture for audio, video, 3D point cloud
- enable larger inputs to Transformers

Regular Transformers

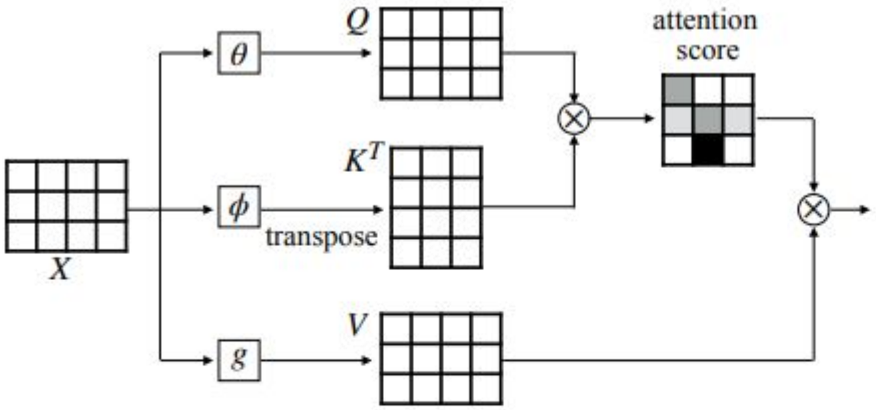
- transform a set of tokens into another set of tokens of same length = 1 transformer layer

- core of the transformer layer = self-attention



Self Attention

- M input tokens = e.g. word embeddings
- from each input token, create Query, Key, Value vectors
- dot product Query * Key vectors => attention matrix of MxM
- dot product Att matrix * Value vectors => new set of token vectors
- => O(M^2) space, O(M^2 * d) time

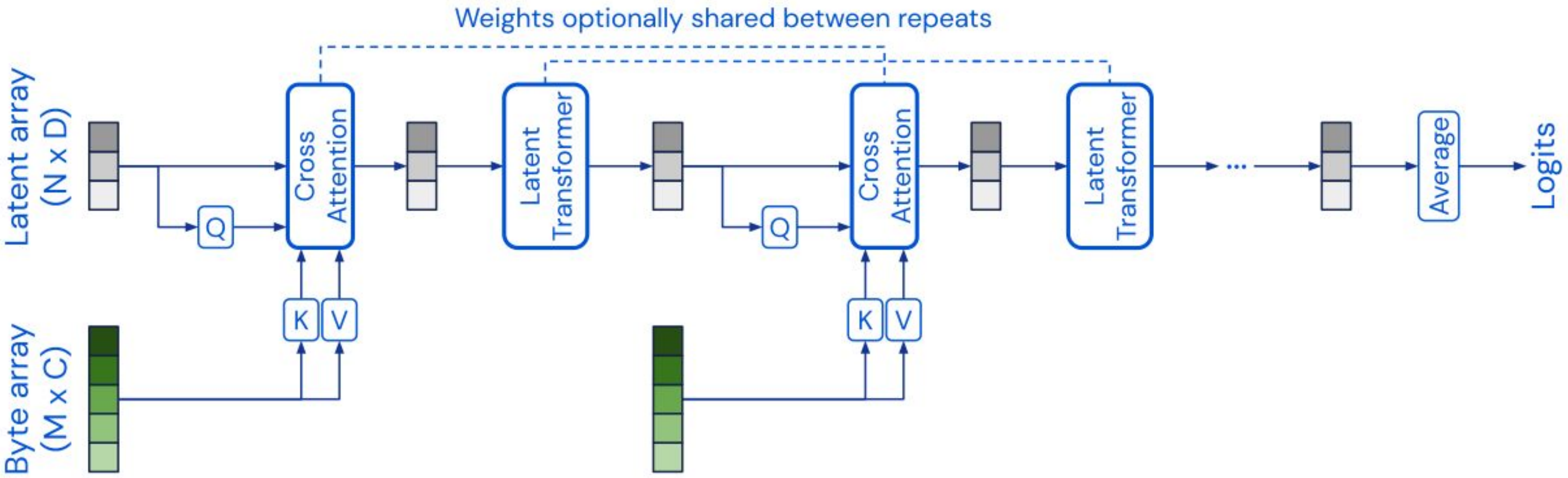


Perceiver architecture

- 224x224 image = 50 000 pixels = M
- => impossible to calculate self-attention with O(50000*50000)
- => use **cross-attention** to inject the input information into the latent representation
- => attention from N latent vectors to M input tokens = O(N*M), N << M

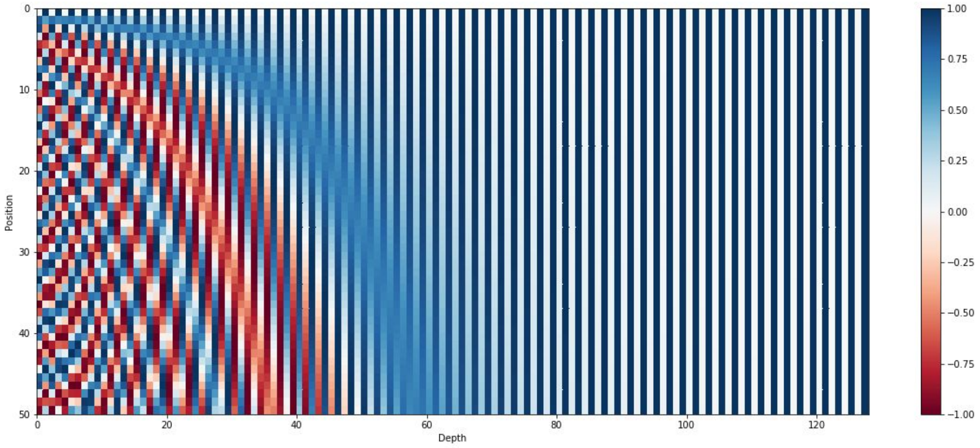
- follow the cross-attention by regular self-attention on the latent space = O(N*N)
- repeat blocks of [cross-attention->regular-latent-attention]

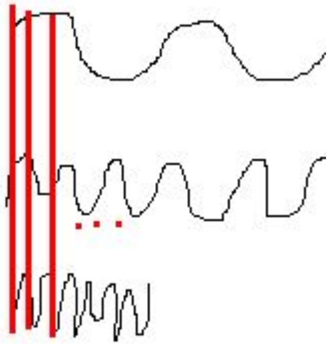
Perceiver architecture



Positional encodings = these are domain dependent

- spatial information is important, Transformer are invariant to it
- => use positional encodings = **Scalable Fourier features**
  - get k frequency bands, k-th band has frequency 2^k
  - sample positional encodings along those bands
  - concat these encodings to the input token vectors
  - 1 row on the figure below = 1 positional encoding





Weight sharing => it’s a RNN

- share weights between latent transformer towers
- = cross-attention compresses the input to N tokens which is then passed to a stack of regular transformer layers

- => it’s RNN unrolled to X steps, getting the same input each step but with a different projection
- = RNN with a cross-attentional input projection, a bottlenecked latent dimensionality, and a latent transformer recurrent core

Architecture details

- latent transformer uses the GPT-2 architecture
- N = number of latent vectors <= 1024
- why RNN? - the bottleneck latent representation might not get all the needed info on the first try => send it in again = like skip connections
- positional encodings are designed to reflect structure in input data
  - = 2D for images, 3D for video, etc
  - the adaptation is simple

Experiments: ImageNet

- 224x224 crops
- positional encodings: using the (x, y) positions on the 224 × 224 input crop
  - 64 bands and a maximum resolution of 224 pixels
- use RandAugment
- 8 cross-attentions with 6 latent transformer layers each => 48 layers in total
- N=1024 latent vectors with 512 channels
- shares weights for all but the first cross-attend and latent transformer modules
  => **only 44M parameters** , ResNet-50 has **23M**

Experiment: Imagenet

- 1st block: input = image => same perf. as ResNet-50
- 2nd block: input = image + fourier (x,y) positional encodings = input to Perceiver

ResNet-50 (He et al., 2016)	76.9
ViT-B-16 (Dosovitskiy et al., 2021)	**77.9**
ResNet-50 (RGB+FF)	73.5
ViT-B-16 (RGB+FF)	76.7
Transformer (64x64)	57.0
Perceiver	76.4

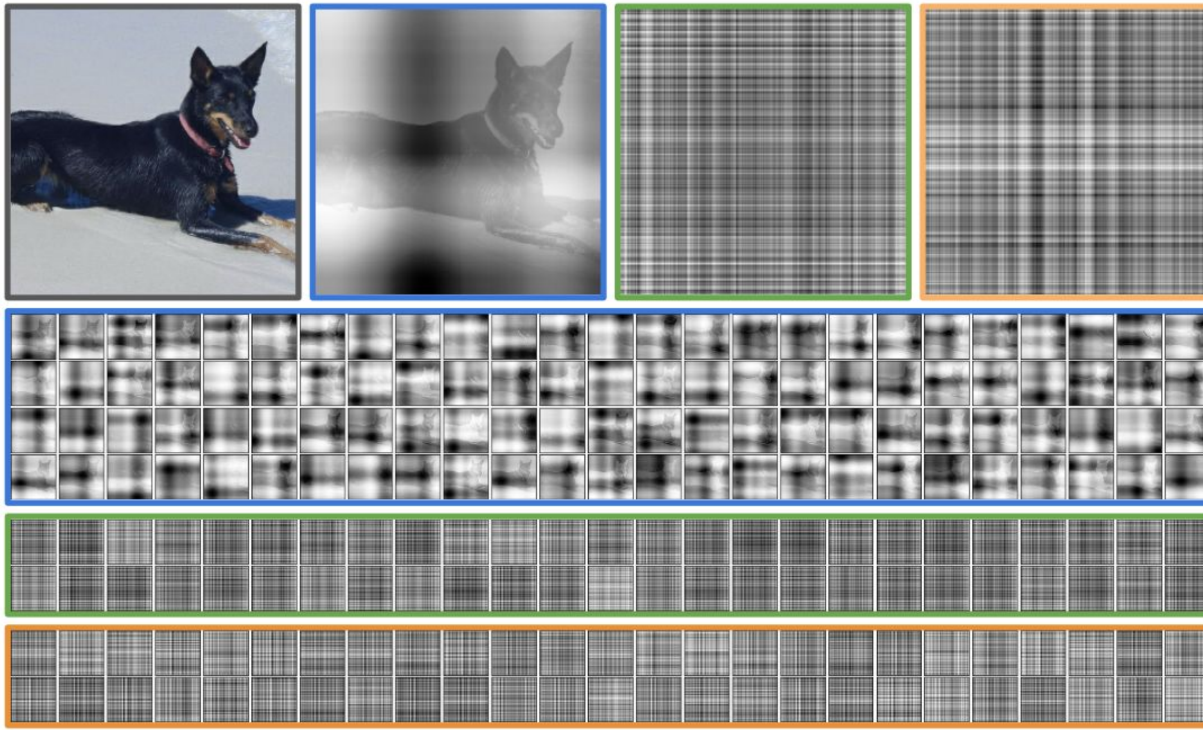
Experiment: Imagenet

- Fixed: permuted with a constant permutation for all images over the dataset
- Random: per-image permutation of pixels
  - BUT the positional encodings still tell you which pixels are neighbors
  - => Perceiver is invariant to this permutation
  - => ResNet with CNN kernels is broken = not that interesting
- All methods receive identical input features (RGB+FF)

	Fixed	Random	Rec. Field
ResNet-50 (RGB+FF)	39.4	14.3	49
ViT-B-16 (RGB+FF)	61.7	16.1	256
Transformer (64x64)	57.0	57.0	4,096
Perceiver	**76.4**	**76.4**	50,176

Experiment: ImageNet

- blue=attention map from 1st layer cross-attention
- green=2nd layer cr. at.
- orange=last layer cr. at.



- 2-8th share weights
  - including the cross-att.
- 1st layer attends to edges
  - the dog is NOT overlayed over the attention map images

Experiments: Audio + Video = AudioSet dataset

- video is passed in in small chunks = not whole clip at once
- audio passed in in 1.28s chunks

Model / Inputs	Audio	Video	A+V
Benchmark (Gemmeke et al., 2017)	31.4	-	-
Attention (Kong et al., 2018)	32.7	-	-
Multi-level Attention (Yu et al., 2018)	36.0	-	-
ResNet-50 (Ford et al., 2019)	38.0	-	-
CNN-14 (Kong et al., 2020)	43.1	-	-
CNN-14 (no balancing & no aug) (Kong et al., 2020)	37.5	-	-
G-blend (Wang et al., 2020b)	32.4	18.8	40.2
Attention AV-fusion (Fayek & Kumar, 2020)	38.4	25.7	46.2
Perceiver	**44.9**	**38.0**	**47.3**

Experiments: 3D Point Clouds = Model-Net40 dataset

-

	Accuracy
PointNet++ (Qi et al., 2017)	**91.9**
ResNet-50 (FF)	66.3
ViT-B-2 (FF)	78.9
ViT-B-4 (FF)	73.4
ViT-B-8 (FF)	65.3
ViT-B-16 (FF)	59.6
Transformer (44x44)	82.1
Perceiver	**85.7**

Sources

- https://arxiv.org/abs/2103.03206
-