UNIVERSITY OF COPENHAGEN

Quantum Tensor Networks for Medical Image Analysis

Beyond the Patterns Lecture Series @ FAU

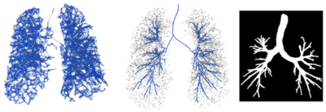
Raghavendra Selvan

Assistant Professor
Dept. of Computer Science (ML Section)
Dept. of Neuroscience (Kiehn Lab)
University of Copenhagen

raghav@di.ku.dk
`https://raghavian.github.io`
@raghavian

May 26, 2021

Core Research: Medical Image Analysis



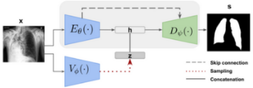
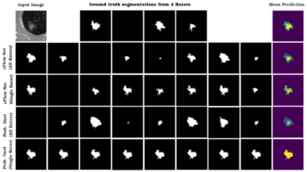


Figure 2. Overview of the proposed model with a variational encoder for data imputation, $V_\phi(\cdot)$ and a U-net type segmentation network with encoder $E_\theta(\cdot)$ and decoder $D_\psi(\cdot)$ (highlighted inside the grey box). The decoder is shared between the data imputation block and the segmentation network.



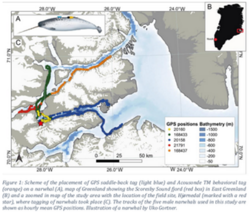
scientific reports

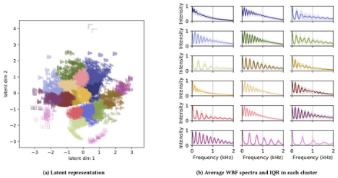
Check for updates

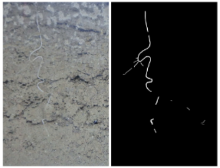
OPEN **Developing and validating COVID-19 adverse outcome risk prediction models from a bi-national European cohort of 5594 patients**

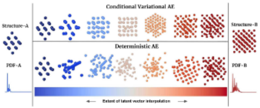
[1] Graph Refinement based Airway Extraction using Mean-Field Networks and Graph Neural Networks (2020), Extraction of Airways from Volumetric Data (2018) - PhD Thesis
[2] Uncertainty quantification in medical image segmentation with Normalizing Flows (2020)
[3] Lung Segmentation from Chest X-rays using Variational Data Imputation (2020)

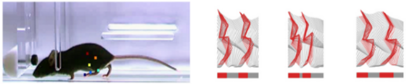
Research Interests: Datascience Collaborations











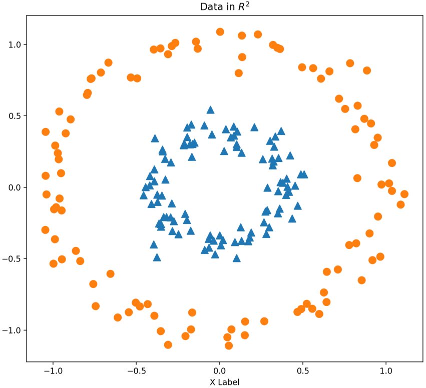
[1] Detection of foraging behavior from accelerometer data using U-Net type convolutional networks (2021)
[2] Dynamic β-VAEs for quantifying biodiversity by clustering optically recorded insect signals (2021)
[3] Segmentation of Roots in Soil with U-Net (2020)
[4] Characterising the atomic structure of mono-metallic nanoparticles from x-ray scattering data using conditional generative models (2020)
[5] Locomotor deficits in ALS mice are paralleled by loss of V1-interneuron-connections onto fast motor neurons (2020)

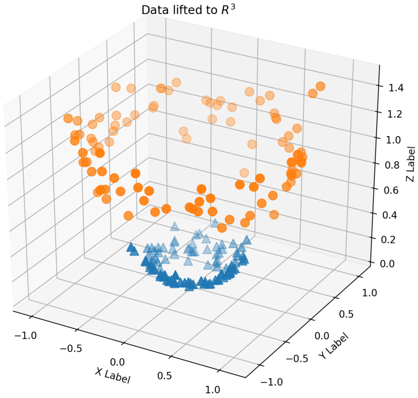
Other interests that help my research

Enjoy 20 ± 5 km trail runs!

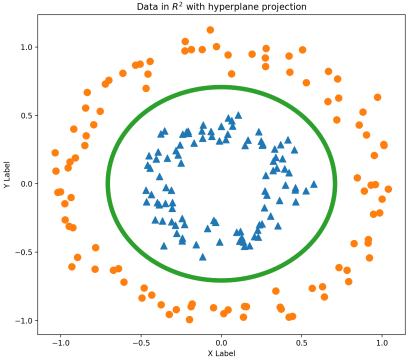
Quantum Tensor Networks for Medical Image Analysis

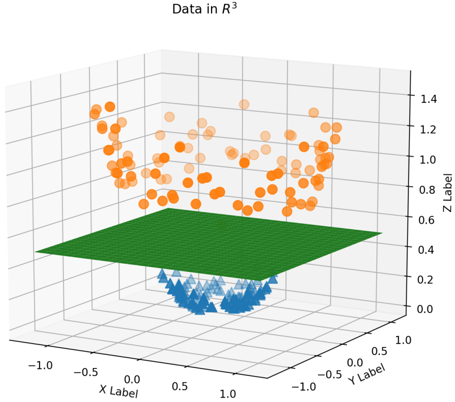
How far can we push linear decision boundaries?





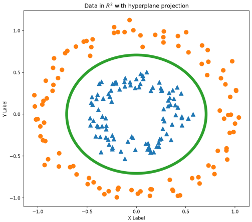
How far can we push linear decision boundaries?

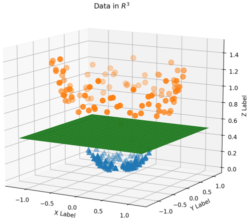




Adapted with permission from Erik Kim. `https://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`

Decision boundaries in low/high dimensions





- Non-linear decisions in lower dimensions
- Neural networks with non-linearities
- Learned features

- Kernel lift to higher dimensions
- Support Vector Machines
- Simple feature crafting helps

Can we bridge these worlds with Tensor Networks?

Tensor networks are:

Learnable approximations of linear models operating in exponentially high dimensional spaces. Formally, tensor networks are factorisations of higher order tensors into networks of lower order tensors.

Tensor networks can efficiently represent high order tensors.

- Studying quantum wave functions [1]
- Compression (data/neural networks) [2]
- Understanding expressive power of neural networks [3]
- Supervised learning [4]

[1] Y-Y Shi et al. Classical simulation of quantum many-body systems with a tree tensor network. Physical Review. 2006

[2] Andrzej Cichocki et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. 2016

[3] Ivan Glasser et al. Expressive power of tensor-network factorizations for probabilistic modeling. NeurIPS 2019

[4] Stoudenmire, E., Schwab, D.J.: Supervised learning with tensor networks. NeurIPS (2016)

This talk is based on our following manuscripts

[1] *Tensor Networks for Medical Image Classification*. R Selvan, EB Dam ; Proceedings of the Third Conference on Medical Imaging with Deep Learning, PMLR 121:721-732, 2020.
[2] *Locally orderless tensor networks for classifying two-and three-dimensional medical images*; R Selvan, S Ørting, EB Dam, 2021, Journal of Machine Learning for Biomedical Imaging (MELBA)
[3] *Multi-layered tensor networks for image classification*. R Selvan, S Ørting, EB Dam; First Workshop on Quantum Tensor Networks in Machine Learning, NeurIPS 2020
[4] *Segmenting two-dimensional structures with strided tensor networks*; R Selvan, EB Dam, J Petersen; 27th international conference on Information Processing in Medical Imaging (IPMI)

One slide introduction to tensor notation

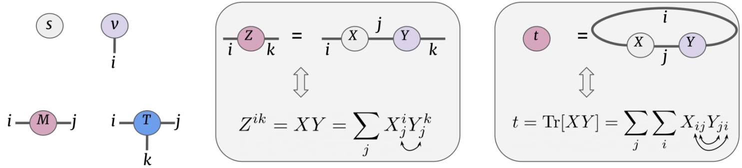


Figure 1: (left) Tensor notation depicting a scalar s, vector v^i, matrix M^{ij} and a general 3-D tensor T^{ijk}. (center) Tensor notation for matrix multiplication or *tensor contraction*, which are used extensively in the matrix product state networks used in this work. We adhere to the convention that the contracted indices are written as subscripts. (right) Tensor notation for trace of product of two matrices.

Penrose, R.: Applications of negative dimensional tensors. Combinatorial mathematics and its applications 1, 221–244 (1971)

Linear model in high dimensions: Feature Maps

Given a vector $\mathbf{x} \in [0,1]^N = [x_1, \dots x_N]$, obtained by flattening 2D/3D image with N pixels,

Consider a d-dimensional pixel-wise local feature map, $\phi^{i_j}(x_j) \in [0,1]^d$ of the form:

$$\phi^{i_j}(x_j) = [\cos(\frac{\pi}{2}x_j), \sin(\frac{\pi}{2}x_j)], \tag{1}$$

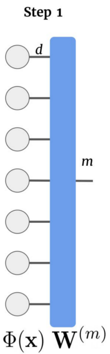
Tensor outer product of the local feature maps yields a joint feature map:

$$\Phi^{i_1,i_2,\dots i_N}(\mathbf{x}) = \phi^{i_1}(x_1) \otimes \phi^{i_2}(x_2) \otimes \cdots \phi^{i_N}(x_N) \in [0,1]^{d^N} \tag{2}$$

Joint feature map

$\Phi(\mathbf{x})$ is an order N tensor or equivalently input image is a vector in d^N-dimensional space

Linear model in high dimensions: Classification Rule



Decision rule for an M-class classification task:

$$f^m(\mathbf{x}) = \left(W^m_{i_1,\dots i_N}\right) \cdot \left(\Phi_{i_1,\dots i_N}(\mathbf{x})\right) \quad (3)$$

where W is an order (N+1) weight tensor and $m = [0, \dots M-1]$

W has $M \cdot d^N$ tunable weights

With a gray scale image of size 16×16 as input and $d = 2$
W has $2 \cdot 2^{1024} \approx 10^{79}$ parameters; about the same as the number of atoms in the observable universe! (10^{80})

Approximate tensor dot product with MPS

- Matrix Product State (MPS) is a type of Tensor Network[5]
- Also known as Tensor Train Networks[6]
- Factorisation of order N tensor into chain of order 3 tensors
- Bond dimension β controls quality of approx.
- Reduces computation complexity from d^N to $N \cdot \beta^3 \cdot d$ (linear in N)

$$W^{m,i_1,i_2,\dots i_N} = \sum_{\alpha_1,\alpha_2,\dots\alpha_N} A^{i_1}_{\alpha_1} A^{i_2}_{\alpha_1\alpha_2} A^{i_3}_{\alpha_2\alpha_3} \dots A^{m,i_j}_{\alpha_j\alpha_{j+1}} \dots A^{i_N}_{\alpha_N} \tag{4}$$

[5] David Perez-Garcia et al. Matrix product state representations. arXiv preprint quant-ph/0608197, 2006.

[6] Ivan V Oseledets. Tensor-train decomposition. 2011

MPS Example

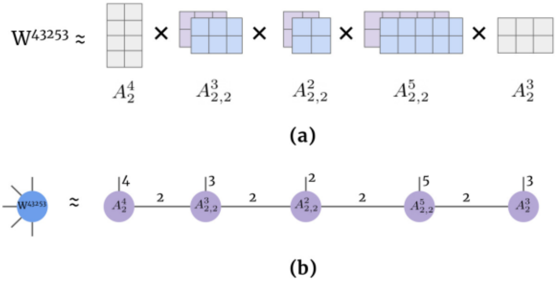


Illustration of MPS factorisation of an order-5 tensor W^{43253} into five tensors of lower order (up to order-3) based on Equation (6). The bond dimension in this factorisation is $\beta = 2$ seen as the subscript indices which are contracted. The tensor W^{43253} has $4x3x2x5x3 = 360$ parameters whereas the MPS approximation requires $8 + 12 + 8 + 20 + 6 = 54$ parameters.

Influence of bond dimension on approximation quality

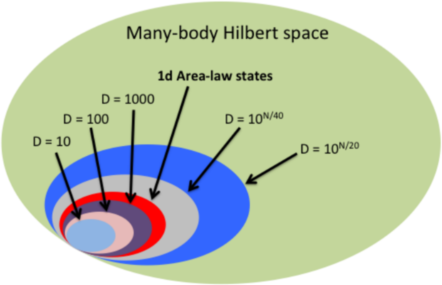
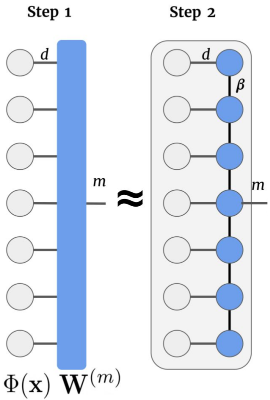


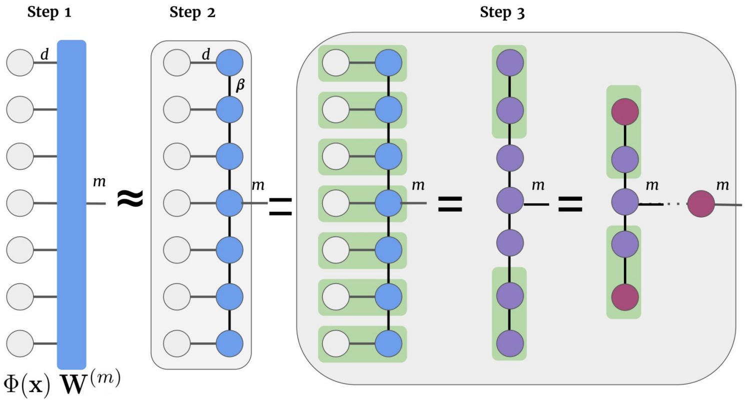
Figure 13: (color online) Onion-like structure of the Hilbert space of a $1d$ quantum many-body system. MPS with finite bond dimension reproduce the properties of the corner of the Hilbert space satisfying the $1d$ area-law for the entanglement entropy. If the bond dimension increases, then the size of the manifold of accessible states also grows. For bond dimensions D sufficiently large (i.e. exponential in the size N of the system), MPS can actually reproduce states beyond the $1d$ area-law and, eventually, cover the whole Hilbert space. Compare this figure to Fig.(4).

Figure from Roman Orus. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States (2014)

Matrix Product State: Tensor Notation



Matrix Product State: Tensor Contraction



Reduces computation complexity from d^N to $N \cdot \beta^3 \cdot d$ (linear in N)

Overview

Tensor Networks for Medical Image Classification

- MPS is defined for 1-d inputs
- 2D images are flattened in existing literature
- No existing work on 3D data
- Loss of spatial structure
- Flattening discards useful information; more so for downstream tasks using medical images

High level idea

- Flatten small regions, assuming local orderlessness.
- Location specific patch-level MPS
- Aggregate at multiple resolutions.

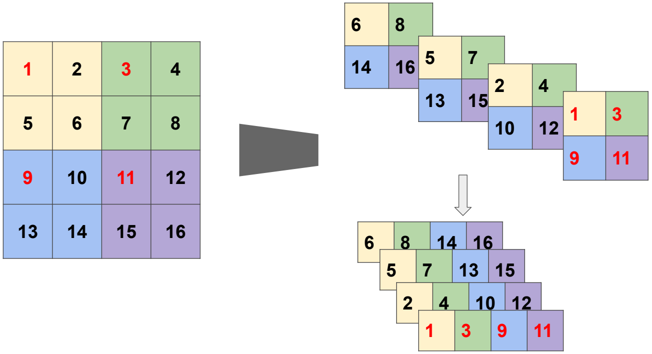
Locally orderless Tensor Network: LoTeNet

Extending Tensor Networks to medical images

1. Partition image into small patches
2. *Squeeze* patches to retain spatial information
3. Perform MPS contraction at patch level
4. Aggregate and perform squeeze + MPS at next resolution
5. Output decision boundary

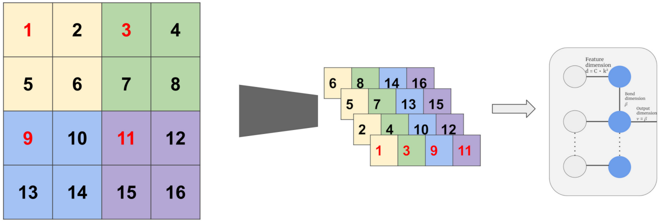
LoTeNet: Partition and Squeeze

LoTeNet: Partition and Squeeze

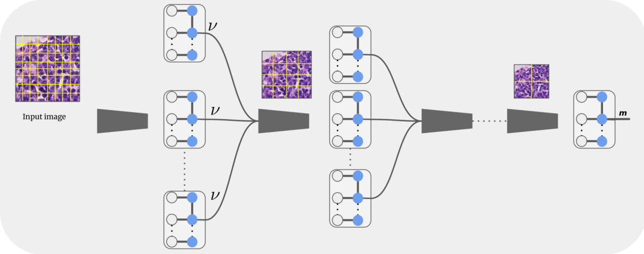


Squeeze operation with stride $k = 2$. A $4 \times 4 \times 1$ image patch is reshaped into $2 \times 2 \times 4$ stack which then yields a vector of size 4 with feature dimension d=4.

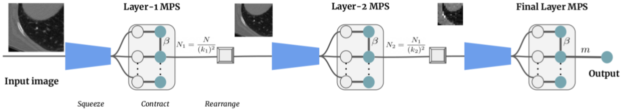
LoTeNet: Patch level MPS



LoTeNet: The Final Model

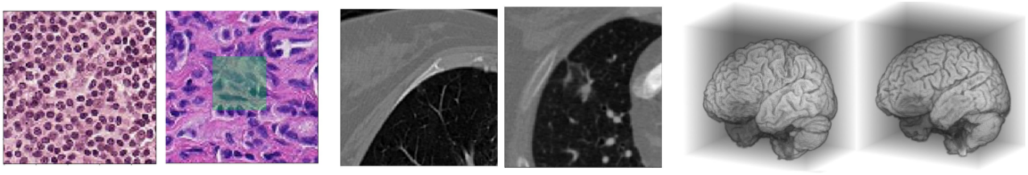


- Model parameters (MPS weights) are optimized using backpropagation.
- Weights of MPS per layer can be shared without performance degradation[7]



[7] *Multi-layered tensor networks for image classification*. R Selvan, S Ørting, EB Dam; First Workshop on Quantum Tensor Networks in Machine Learning, NeurIPS 2020

Model Evaluation: Datasets



a) PCam

b) LIDC

c) OASIS

- The PatchCamelyon (PCam) dataset
- Binary classification
- Positive label indicates $\geq$ One pixel with tumour
- Image patches of size 96×96 px
- $220k$ patches for training-validation $(80:20)$
- $57.5k$ test patches

- 128×128 px image patches
- $15k$ patches. $60:20:20$ splits for training/validation/test
- Annotated by 4 radiologists. Originally a segmentation dataset

- $128 \times 128 \times 128$ px skull stripped volumes
- 155 subjects
- Binary classes: Alzheimer's diseases (AD), Cognitively normal (CN)

Model Evaluation on 2D data: Results

Table: Performance comparison on PCam dataset (left) and LIDC dataset (right). For all models, we specify the GPU memory utilisation in gigabytes. Best AUC across all models is shown in boldface.

PCam Models	GPU(GB)	AUC	LIDC Models	GPU(GB)	AUC
Rotation Eq-CNN	11.0	**0.963**	LoTeNet (ours)	0.7	**0.874**
Densenet	10.5	0.962	Tensor Net-X ($\beta = 10$)	4.5	0.847
LoTeNet (ours)	0.8	0.943	Densenet	10.5	0.829
Tensor Net-X ($\beta = 10$)	5.2	0.908	Tensor Net-X ($\beta = 5$)	1.5	0.823

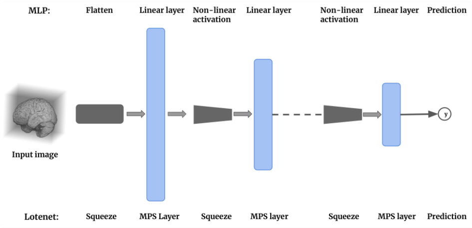
Model Evaluation on 3D data: Results

Table: Performance comparison on OASIS dataset with comparing methods using 3D and 2D inputs. The performance is reported as balanced accuracy (BA) averaged over 5-fold cross validation. Number of parameters, maximum GPU utilization (GPU) and computation time per training epoch (t) for all methods are also reported.

OASIS Models	Input	# Param.	GPU (GB)	t (s)	Average BA
LoTeNet (ours)	3D	52M	2.1	45.3	$\mathbf{0.71 \pm 0.09}$
Subject-level CNN	3D	1M	8.8	8.7	0.67 ± 0.08
CNN Baseline	3D	6.4M	11.5	12.8	0.64 ± 0.05
MLP Baseline	3D	78M	4.5	4.1	0.63 ± 0.03
Densenet	2D	0.2M	10.5	80.1	0.67 ± 0.04
LoTeNet (ours)	2D	0.4M	0.7	81.3	0.65 ± 0.03

Summary

- Lifting data to high dimensional data is useful
- Tensor networks like MPS efficiently approximate high order tensors
- Proposed LoTeNet for 2D/3D medical image classification
- Different paradigm compared to feed-forward NNs or CNNs



- Low GPU memory requirement (no intermediate feature-maps, contractions)

What about segmentation tasks?

Overview

Segmentation poses additional challenges to Tensor Networks

- Dot product based linear models
- Final ouptut is in label space (not image space)
- Segmentation is harder than classification when flattening images

High level idea

- Pose segmentation as a pixel-wise classification task
- Act on small patches as in LoTeNet
- Use the same MPS for all patches (weight-sharing)

Based on work under review. R Selvan, EB Dam, J Petersen; Segmenting two-dimensional structures with strided tensor networks.

Tensor networks for segmentation

Given a 2D image, $X \in \mathbb{R}^{H\times W\times C}$ with $N = H \times W$ pixels, C channels.

Segmenting X into M classes, is given as

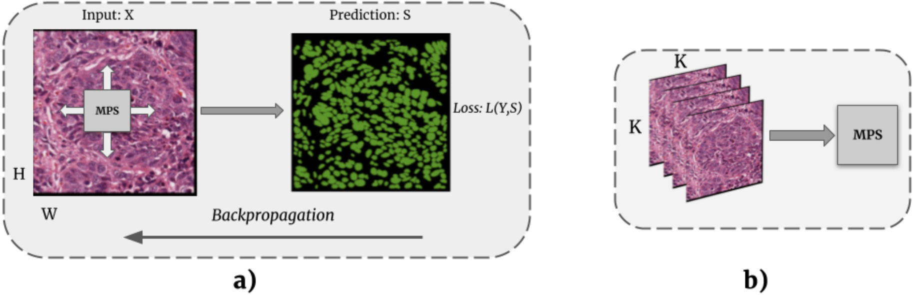
$$f(\cdot\,;\ W) : X \mapsto Y \in \{0,1\}^{H\times W\times M}. \tag{5}$$

Using MPS on $K \times K$ patches the strided tensor network formulation is:

$$f(\mathbf{x}; W_K^m) = \{W_K^m \cdot \Phi(\mathbf{x}_{(i,j)})\} \quad \forall\, i = 1,\ldots,H/K,\ j = 1,\ldots,W/K \tag{6}$$

where superscript index m on the weight tensor is the output index of dimension N. And (i,j) are row i and column j indices of the image grid with patches of size $K \times K$.

Strided Tensor Networks for segmenting 2D structures



Model evaluation: Datasets

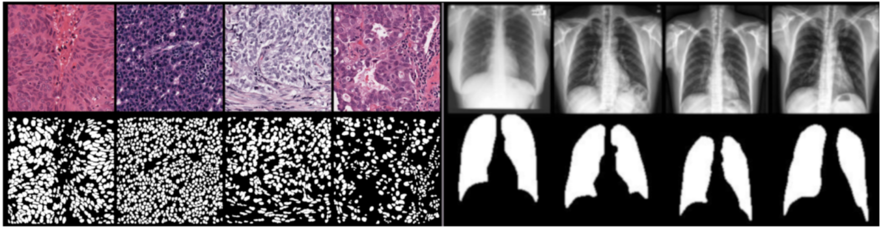


Fig. 4. (First four columns) Sample images from the MO-Nuseg dataset comprising histopathology slides from multiple organs (top row) and the corresponding binary masks (bottom row). (Last four columns) Sample chest X-ray images from the Lung-CXR dataset with corresponding binary masks.

- Multi-organ nuclei segmentation
- Stained tissue microscopy images
- 1000 x 1000 px
- 30 training/14 testing

- Chest X-ray images
- Shenzhen and Montgomery hospital
- 128 x 128 px
- 528 training/ 176 test

Model evaluation: Results

Table: Test set performance comparison for segmenting nuclei from the stained tissue images (MO-NuSeg) and segmenting lungs from chest CT (Lung-CXR). For all models, we report the number of parameters $|\Theta|$, computation time per training epoch, area under the curve of the precision-recall curve (PRAUC) and average Dice accuracy (with standard deviation over the test set). The representation (Repr.) used by each of the methods at input is also mentioned.

Dataset	Models	Repr.	$\lvert\Theta\rvert$	t(s)	PRAUC	Dice
MO-NuSeg	Strided TeNet (ours)	1D	$5.1K$	21.2	0.78	$\mathbf{0.70 \pm 0.10}$
	U-net	2D	$500K$	24.5	**0.81**	$\mathbf{0.70 \pm 0.08}$
	MPS TeNet	1D	$58.9M$	240.1	0.55	0.52 ± 0.09
	CNN	2D	–	510	–	0.69 ± 0.10
Lung-CXR	Strided TeNet (ours)	1D	$2.0M$	6.1	0.97	0.93 ± 0.06
	U-net	2D	$4.2M$	4.5	**0.98**	$\mathbf{0.95 \pm 0.02}$
	MPS TeNet	1D	$8.2M$	35.7	0.67	0.57 ± 0.09
	MLP	1D	$2.1M$	4.1	0.95	0.89 ± 0.05

Model evaluation: Qualitative results

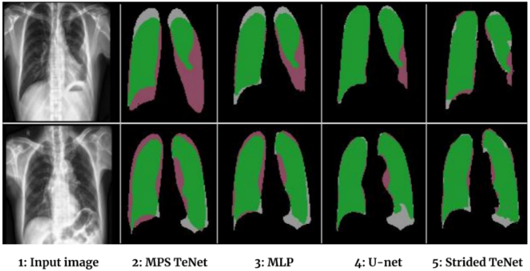


Fig. 5. Two test set CXRs from Lung-CXR dataset along with the predicted segmentations from the different models. All images are upsampled for better visualisation. (Prediction Legend – Green: True Positive, Grey: False Negative, Pink: False Positive)

Summary

- Adapted tensor networks (MPS) for segmentation
- Using weight-shared MPS with patch strides does it
- Comparable performance with U-net
- Can MPS be thought of learning a single filter?

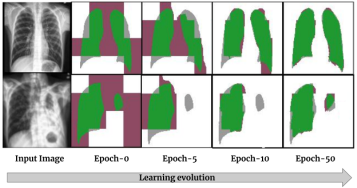


Fig. 6. Progression of learning for the strided tensor network, for the task of segmenting lung regions. Two *validation* set input chest CT images (column 1) and the corresponding predictions at different training epochs are visualized overlaid with the ground truth segmentation. Images are of size 128×128 and the stride is over 32×32 regions. All images are upsampled for better visualisation. (Prediction Legend – Green: True Positive, Grey: False Negative, Pink: False Positive)

Overall Summary

+ Tensor networks for supervised medical image analysis
+ Linear models in high dimensional spaces
+ Single model hyperparameter (β) for LoTeNet
+ Two model hyperparameters for Strident Tensor network (β, K)
+ Inadverent conseqence of reduced GPU utilization
- Tendency to overfit
- No granular control of model complexity
- Not optimized for efficiency, yet

Conclusion

- Lifting data to high dimensional data makes linear models as powerful as NNs
- Synergy between Quantum Physics and ML
- Different paradigm compared to feed-forward NNs or CNNs
- More formal connections to GPs, Kernel methods are ongoing
- Packages on several platforms: TorchMPS (Pytorch), iTensor (Julia), TensorNetwork (TensorFlow) and some on Jax too!
- New and exciting applications are to be expected

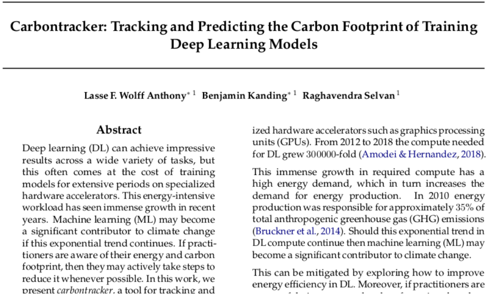
Ivan Glasser et al. From Probabilistic Graphical Models to Generalized Tensor Networks for Supervised Learning, (2020)

Samuel Cavinato et al. Optimizing Radiotherapy Plans for Cancer Treatment with Tensor Networks, (2020)

Erdong Guo et al. Infinitely Wide Tensor Networks as Gaussian Process (2021)

Thanks!

- My collaborators: Erik B Dam, Silas Ørting, Jens Petersen
- Model and data are available here: Classification model: `https://github.com/raghavian/lotenet_pytorch`
  Segmentation model: `https://github.com/raghavian/strided-tenet`
- raghav@di.ku.dk



Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models

Lasse F. Wolff Anthony* [1] Benjamin Kanding* [1] Raghavendra Selvan [1]

Abstract

Deep learning (DL) can achieve impressive results across a wide variety of tasks, but this often comes at the cost of training models for extensive periods on specialized hardware accelerators. This energy-intensive workload has seen immense growth in recent years. Machine learning (ML) may become a significant contributor to climate change if this exponential trend continues. If practitioners are aware of their energy and carbon footprint, then they may actively take steps to reduce it whenever possible. In this work, we present *carbontracker*, a tool for tracking and

ized hardware accelerators such as graphics processing units (GPUs). From 2012 to 2018 the compute needed for DL grew 300000-fold (Amodei & Hernandez, 2018).

This immense growth in required compute has a high energy demand, which in turn increases the demand for energy production. In 2010 energy production was responsible for approximately 35% of total anthropogenic greenhouse gas (GHG) emissions (Bruckner et al., 2014). Should this exponential trend in DL compute continue then machine learning (ML) may become a significant contributor to climate change.

This can be mitigated by exploring how to improve energy efficiency in DL. Moreover, if practitioners are

pip install carbontracker