Computational efficiency of deep nets: pruning and sparsity

THEORIES OF DEEP LEARNING: C6.5,
LECTURE / VIDEO 15
Prof. Jared Tanner
Mathematical Institute
University of Oxford

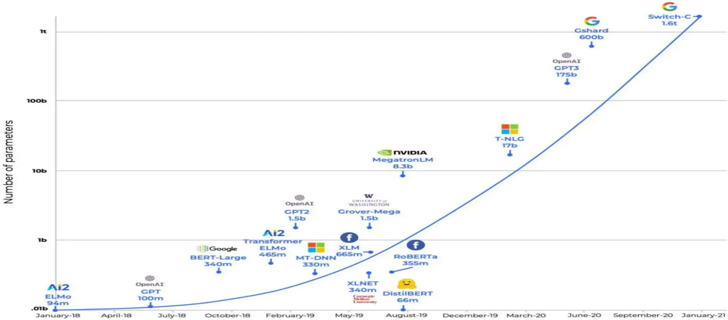
UNIVERSITY OF OXFORD
Mathematical Institute

Oxford Mathematics

The ever increasing size of Machine Learning models

Ever larger deep nets show improved performance

University of Oxford Mathematical Institute



"The results we survey show that today's sparsification methods can lead to a 10-100x reduction in model size, and to corresponding theoretical gains ... all without significant loss of accuracy."
`https://arxiv.org/pdf/2102.00554`

"People are often curious about how much energy a ChatGPT query uses; the average query uses about 0.34 watt-hours, about what an oven would use in a little over one second. (Sam Altman, OpenAI CEO)"
https://blog.samaltman.com/the-gentle-singularity

Review of energy consumption of open source image and video models suggest video generation is about 2000 times more costly than text generation; strongly dependent on text / video length and resolution; details in:
https://arxiv.org/pdf/2509.19222?

Estimate 10-second Sora video consumes 1kWh of electricity; UK house 8-10kWh per day.

The importance of computational efficiency

The energy cost at inference is a primary limitation to deployment

University of Oxford
Mathematical Institute

"On September 30, OpenAI debuted its Sora video creation app for Apple's iOS platform racking up a stunning 1 million downloads in a week despite an invitation-only rollout. By Halloween the app had been downloaded 4 million times, per AppFigures, and was churning out millions of 10-second AI-generated videos per day.

More than $5 billion annualized, or around $15 million per day, according to Forbes estimates and conversations with experts. When Bill Peebles, OpenAI's head of Sora, observed on October 30 that "The economics are currently completely unsustainable," he was right on the money."
https://www.forbes.com/sites/phoebeliu/2025/11/10/openai-spending-ai-generated-sora-videos/

Sparsifying deep networks

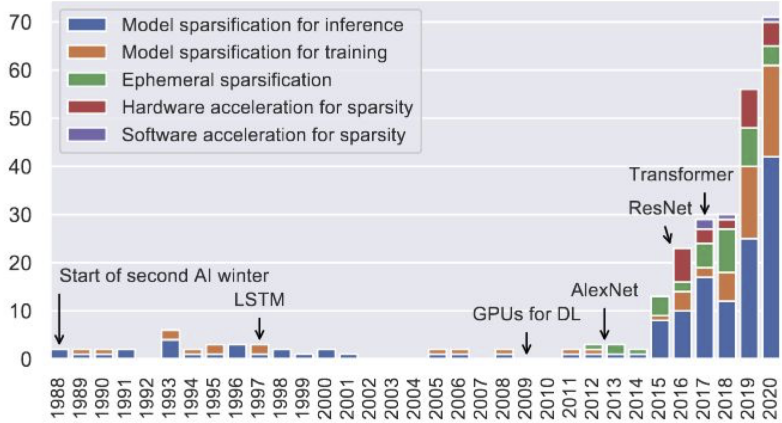
Severe overparameterisation and scale of modern nets

UNIVERSITY OF OXFORD
Mathematical Institute

- Deep nets for some state of the art tasks have vast numbers of trainable parameters:
  - ResNet101, image classification - 45 million parameters
  - GPT-3, text generation - 175 billion parameters
  - T5-XXL, language model - 1.6 trillion parameters
- An approximation theory viewpoint suggest this isn't necessary at inference, see Optimal Approximation with Sparsely Connected Deep Neural Networks, Bolcskei et al. 2019. `https://arxiv.org/abs/1705.01714`
- Practise tell us the number of parameters in MLP and CNNs can be reduced to 5% or fewer parameters without loss of accuracy.

Sparsity throughout deep learning (Hoefler 21')

Sparsity appears in numerous aspects of deep learning

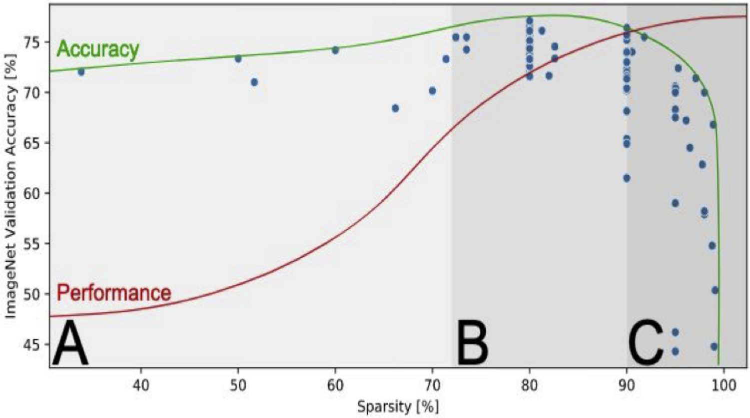


https://arxiv.org/pdf/2102.00554

Sparsifying and the lottery ticket hypothesis

Reducing the number of parameters initially improves accuracy

UNIVERSITY OF OXFORD
Mathematical Institute



https://arxiv.org/pdf/2102.00554.pdf

How to create sparse networks

At initialisation and dynamic pruning

Oxford Mathematical Institute

- Starting with a sparse network: pruning at initialisation:
  - From any network, random or trained, determine a measure of importance for a parameter in the network and set the parameter to zero if below that threshold.
  - Examples include: magnitude of the parameters, gradient of the loss with respect to that parameter, or measure of information flow.
- Dynamic sparsifying the network:
  - Prune as above, but allow some entries to be reintroduced and then pruned again during training.
  - Most successful is to prune based on magnitude and re-introduce based on training gradient magnitude.
- This is an increasingly active area due to the every growing size of networks.

Mathematical Institute

Expand the loss in parameter space and measure loss changes with parameters; identify parameters that don't impact the loss function

$$\mathcal{L}(\theta+\delta\theta)-\mathcal{L}(\theta)\approx(\delta\theta)^T\nabla_\theta\mathcal{L}(\theta)+\frac{1}{2}(\delta\theta)^TH(\theta)(\delta\theta)$$

- Lecun et al. 89' considered convergence, $\nabla_\theta\mathcal{L}(\theta)\approx 0$, and approximate the Hessian $H(\theta)$ as diagonal, then the change in the loww by removing an entry θ_i is $\frac{1}{2}\theta_i^2\nabla_{\theta_i}^2\mathcal{L}$. This is a "saliency score" for each parameter and those with small values can be removed.

  https://proceedings.neurips.cc/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf

- Wang et al. 19' consider a much simply version where the Hessian is the Identity, $H(\theta)=I$, which simplifies to removing the entries with smallest magnitude; Iterative Magnitude Pruning (IMP). https://proceedings.mlr.press/v97/wang19g/wang19g.pdf

- $L0-$regularization of the weights by adding sparsifying weight decay; Louizos et al. 18' https://arxiv.org/pdf/1712.01312
- LAP removes small weights while taking into account magnitude of incoming and outgoing connections; Park et al. 20' https://openreview.net/pdf?id=ryl3ygHYDB
- A review of methods suggests few methods are reliably better than simple magnitude pruning; Gale et al. 19' https://arxiv.org/pdf/1902.09574

Pruning at initialization: example score functions

Randomly initialize, then score the value of weights

University of OXFORD Mathematical Institute

- SNIP by Lee et al. 19'; $\left|\frac{\partial \mathcal{L}(\theta)}{\partial \theta} \cdot \theta\right|$
  https://openreview.net/pdf?id=B1VZqjAcYX
- GraSP by Wang et al. 19' $-\left(H\frac{\partial \mathcal{L}(\theta)}{\partial \theta}\right) \cdot \theta$
  https://openreview.net/pdf?id=SkgsACVKPH
- FORCE by de Jorge et al. 21' $\left|\frac{\partial \mathcal{L}(\tilde{\theta})}{\partial \theta} \cdot \theta\right|$ with $\tilde{\theta}$ after pruning
  https://openreview.net/pdf?id=9GsFOUyUPi
- And many many more....

Dynamic sparse training

Include rules to train, prune, regrow, repeat

- SET by Mocanu et al. 18': magnitude pruning and random regrowing entries
  https://www.nature.com/articles/s41467-018-04316-3.pdf
- DSR by Mostafa and Wang et al. 19': similar to SET but proportion pruned not constant throughout iterates and sparsity across layers can vary
  https://proceedings.mlr.press/v97/mostafa19a/mostafa19a.pdf
- RigL by Evci et al 20': magnitude pruning and regrowing based on gradient magnitude
  https://arxiv.org/pdf/1911.11134
- And many many more....

Dense trained networks can be pruned and re-trained

Succesively increasing sparsity constraints

UNIVERSITY OF OXFORD
Mathematical Institute

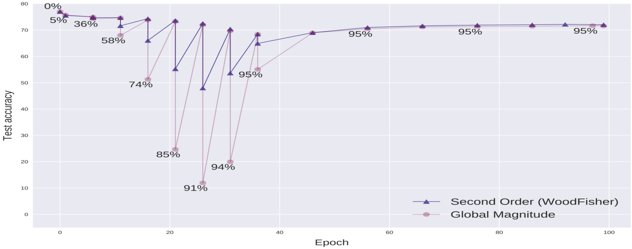


Fig. 9. An illustration of a standard gradual pruning schedule including fine-tuning periods, applied to ResNet-50 on the ImageNet dataset. The graph depicts the evolution of the validation accuracy for two different methods (global magnitude pruning and WoodFisher [Singh and Alistarh 2020]) across time.

https://arxiv.org/pdf/2102.00554

Structure: Entries of dense nets can be pruned individually to effectively architecture adaptation by removing entire filters in CNNs or heads in multi-head attention.

Fraction of active per layer is key (Frankle 21')

Fix a fractional sparsity at initialization

UNIVERSITY OF OXFORD
Mathematical Institute

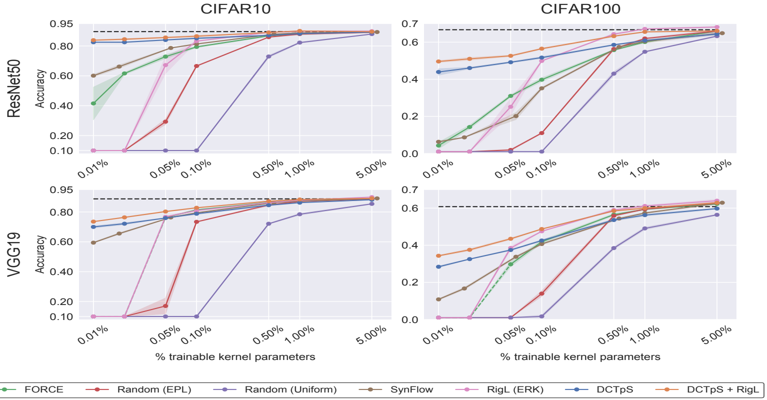
"Pruning neural networks at initialization: why are we missing the mark?" (Frankle et al. 2021) conjecture the key ingredient is the fractional sparsity per layer, not the particular algorithm.
"When using the methods for pruning at initialization, it is possible to reinitialize or layerwise shuffle the unpruned weights without hurting accuracy. This suggests that the useful part of the pruning decisions of SNIP, GraSP, SynFlow, and magnitude at initialization are the layerwise proportions rather than the specific weights or values."
https://arxiv.org/pdf/2009.08576

Extreme sparsity with additive component (Price et al. 21')

Dense for the price of sparse; including a dense additive component

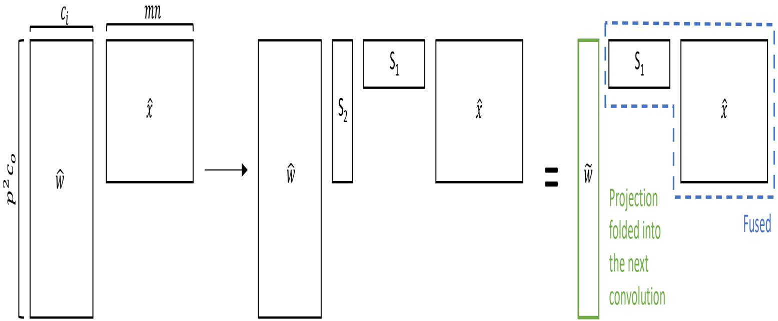
Oxford Mathematical Institute

Rather than sparse weight matrices alone, use a dense fast transform plus a trainable sparse matrix



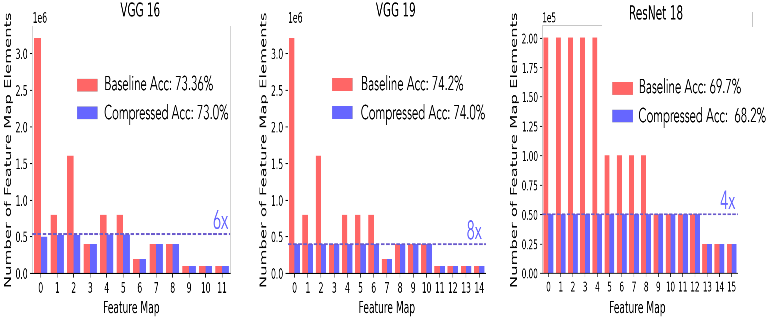
Sketching and low-rank models

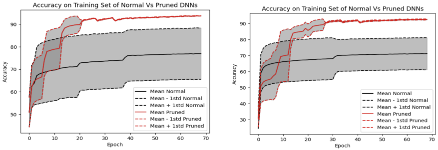
Between the matrix vector multiplication learn sketching matrices S_1 and S_2 then absorb S_2 into the weight matrix and apply S_1 and $\tilde{W}$ sequentially



UNIVERSITY OF OXFORD
Mathematical Institute

The hidden layer outputs have varying sizes. Apply sketching to fit within a maximum memory constraint. Large hidden layers not stored, only their sketches, but retain the original dimensions.





(a) (b)

Figure 4: Comparison of Normal DNN, trained normally, and pruned DNN, trained using the RMT approach on the training set. The sub-figures correspond to the different initial topologies: (a) $[784, 3000, 4000, 3000, 500, 10]$, (b) $[784, 2000, 2000, 2000, 2000, 1000, 500, 10]$.

Spectra of randomly initialized matrices have known distributions. During training check the spectra and project out spectra that is consistent from unlearned random matrices.

Enhancing accuracy in deep learning using random matrix theory by Berlyand et al. 2023.

`https://arxiv.org/abs/2310.03165`

Patches of inputs of a fixed dimension are extracted and arranged into a matrix X, similar to CNNs. They queries, keys, and values are then computed with matrix-products $Q^T = W_Q X^T$, $K^T = W_K X^T$, and $V^T = W_V X^T$ then the re-activation attention layer is

$$H = \text{softmax}\left(QK^T n^{-1/2}\right) V$$

The scaling α is typically $1/2$, but also sometimes 1, and generally Q and V have layer-norm applied to enforce fixed mean and variance. Multi-head attention concatenates many of these and multiply by a weight matrix.

`https://arxiv.org/abs/1706.03762`

`https://arxiv.org/abs/1607.06450`

The product QK^T generates a large matrix whose computation is one of the main bottlnecks for Attention based models. Hashing computes a block sparse approximation as follows: letting $Q, K \in \mathbb{R}^{n\times d}$ draw $W \in \mathbb{R}^{d\times s}$ with $s \ll d$ the hasing number of blocks. Form $QW \in \mathbb{R}n \times p$ where $(QW)_{ij}$ measures the correlation of row i in Q with column j in W. The largest entry in the i^{th} row of QW indicates the column in W most aligned with the i^{th} row in Q; letting λ_j be the index set of rows in Q most aligned with column j of W. Repeat for K to get a similar partitioning $\tilde{\lambda}_j$ of the rows in K. Rather than compute QK^T, form just the s blocks with rows in λ_j and columns in $\tilde{\lambda}_j$ for $j = 1 \ldots s$.

Rather than computing the product

$$H = \text{softmax}\left(QK^T d^{-\alpha}\right) V$$

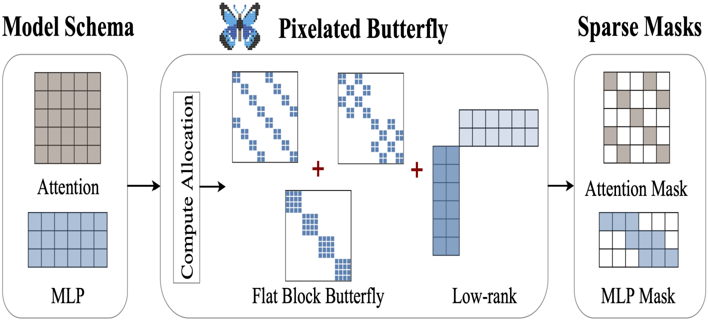
approximate $\left(QK^T d^{-\alpha}\right)$ by hashing, and then sketch its product with V by selecting a subset of columns on A and associated rows in V.

HyperAttention: Long-context attention in near-linear time, Han et al., 2023.

`https://arxiv.org/pdf/2310.05869`

Pruning and low-rank for Attention

Local and global structure by combining sparse plus low-rank



Pixelated Butterfly: Simple and Efficient Sparse Training for Neural Network Models by Dao et al, 2022. `https://arxiv.org/abs/2112.00029`

Low-rank in deep nets is being studied more broadly

LoRA and Galore

LoRA: Low-rank adaptation of large language models by Hu et al. 2021 takes a pre-trained network and includes a low-rank addition for fine-tuning. Specifically, the original weights are held fixed, but for each layer they add a low-rank trainable component for time-tuning / transfer learning.
https://arxiv.org/abs/2106.09685

GaLORE: Memory-efficient LLM training by gradient low-rank projection by Zhao et al. 2024 uses a low-rank gradient update.
https://arxiv.org/abs/2403.03507

"Low-rank" is a growing theme at the leading ML conferences.

Some approximation theory for sparsity

The Lottery Ticket Hypothesis (Frankle 19')

Theory suggesting pruning at initialization is possible

"A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations."

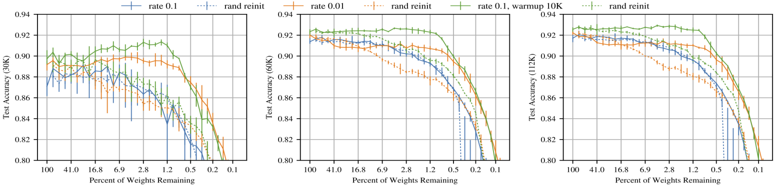


Figure 7: Test accuracy (at 30K, 60K, and 112K iterations) of VGG-19 when iteratively pruned.

`https://arxiv.org/pdf/1803.03635`

Theorem: (Malach et al. 20')

Let F be a network of depth ℓ such that $\|W^F\|_2 \leq 1$, $\|W^F\|_{max} \leq n^{-1/2}$ where n is the width. Fix some $\epsilon, \delta \in (0,1)$. Let G be a network of width poly$(d, n, \ell, \epsilon^{-1}\log(1/\delta)$ and depth 2ℓ with W^G initialized $U[-1,1]$). Then with probability at least $1-\delta$ over the the initialization of G, there exists a sub-network $\tilde{G}$ of G with $\sup_{x\in\chi}|\tilde{G}(x) - F(x)| \leq \epsilon$ and $\tilde{G}$ has at most $\mathcal{O}(dn + n^2\ell)$ neurons.

Note, the sub-network has the same order number of parameters $n^2\ell$, potentially with a worst constant. No direct training!
`https://proceedings.mlr.press/v119/malach20a/malach20a.pdf`

Optimal approximation rates with sparse nets

Bolcskei et al. 2018.

UNIVERSITY OF OXFORD
Mathematical Institute

In the below paper they show that all functions that can be approximated well using "affine-systems" (e.g. Wavelets and similar) can also be well approximated by sparsely connected deep networks.

These results are achieved by considering the class of networks with a fixed budget of parameters and considering the union of possible width and depth networks.

`https://arxiv.org/pdf/1705.01714`

Some theory for initialization and sparsity in the hidden layers

Fully connected and convolutional neural network

Increasingly large networks scaled through n

UNIVERSITY OF OXFORD
Mathematical Institute

We will consider sequences of increasing size fully connected networks

$$h_i^{(\ell)}(x)[n] = \sum_{j=1}^{N_{\ell-1}[n]} W_{ij}^{(\ell)} z_j^{(\ell-1)}(x)[n] + b_i^{(\ell)}, \quad z_j^{(\ell)}(x)[n] = \phi(h_j^{(\ell)}(x)[n]),$$

The objects $h_i^{(\ell)}(x)[n]$ are referred to as pre-activation values and $\phi(\cdot)$ is the nonlinear activation, here applied entrywise.

Typical initializations of such networks have parameters, weight matrices $W^{(\ell)}$ and biases $b^{(\ell)}$, which are drawn i.i.d. such as $\mathcal{N}(0, \sigma_w^2/N^{(\ell)}[n])$ and $\mathcal{N}(0, \sigma_b^2)$.

Intuitively the pre-activation entries are then mean-zero Gaussian.

PSEUDO-IID weight matrices

Examples include: i.i.d. entries, but also unitary, low-rank, and structured sparse matrices

Mathematical Institute

Definition: PSEUDO-IID distribution (Naite Saada et al. 23')

Definition (PSEUDO-IID)

The random matrix $W = (W_{ij}) \in \mathbb{R}^{m \times n}$ is in the PSEUDO-IID distribution with parameter σ^2 if

1. the matrix is row-exchangeable and column-exchangeable,
2. its entries are centered, uncorrelated, with variance $\mathbb{E}(W_{ij}^2) = \frac{\sigma^2}{n}$,
3. $\mathbb{E}\left|\sum_{j=1}^{n} a_j W_{ij}\right|^8 = K \|\mathrm{a}\|_2^8 n^{-4}$ for some constant K,

When weight matrices $W^{(\ell)}$, $1 \leq \ell \leq L+1$, of a neural network are drawn from a PSEUDO-IID distribution, we will say that the network is under the PSEUDO-IID regime.

The PSEUDO-IID distribution generalized i.i.d. to also include suitably drawn low-rank matrices and matrices with *structured sparsity*.

University of Oxford
Mathematical Institute

Thm: GP limit for $\textsc{Pseudo-iid}$ networks (Naite Saada et al. 23')

Fully-connected neural networks with $\textsc{Pseudo-iid}$ weight matrices with parameter σ_W^2 then the sequence of random fields $(i, x) \in [N_\ell] \times \mathcal{X} \mapsto h_i^{(\ell)}(x)[n] \in \mathbb{R}^{N_\ell}$ converges in distribution to a centered Gaussian process $(i, x) \in [N_\ell] \times \mathcal{X} \mapsto h_i^{(\ell)}(x)[*] \in \mathbb{R}^{N_\ell}$, whose covariance function is given by

$$\mathbb{E}\left[h_i^{(\ell)}(x)[*] \cdot h_j^{(\ell)}(x')[*]\right] = \delta_{i,j} K^{(\ell)}(x, x'),$$

where

$$K^{(\ell)}(x, x') = \sigma_b^2 + \sigma_W^2 \mathbb{E}_{(u,v)\sim\mathcal{N}(0, K^{(\ell-1)}(x,x'))}[\phi(u)\phi(v)], \quad \ell \geq 1$$

Similar results for CNNs with increasing number of channels for intermediate layers.

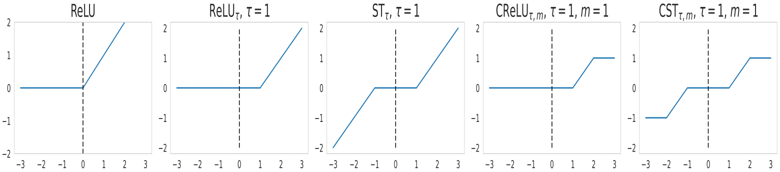
Sparse hidden layers (Price et a. 21')

Two natural options for sparsifying activations: shifted ReLU and Soft Thresholding

Sparse hidden layers reduce network computation: exemplar sparsity inducing activations are the shifted ReLU and soft thresholding

$$\mathrm{ReLU}_\tau(x) = \begin{cases} x - \tau, & \text{if } x > \tau \\ 0, & \text{otherwise,} \end{cases}$$

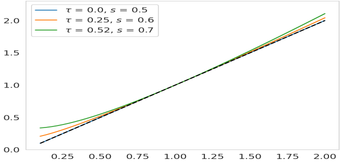
$$\mathrm{ST}_\tau(x) = \begin{cases} x - \mathrm{sign}(x)\tau, & \text{if } |x| > \tau \\ 0, & \text{otherwise,} \end{cases}$$

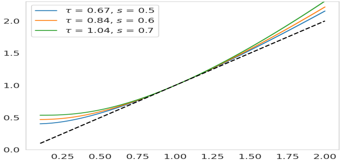


Sparsity inducing activations ReLU_τ and ST_τ are unstable

The EoC $\chi = 1$ condition ensures $V'(q^*) = 1$ and $V''(q^*) > 0$

ReLU_τ and ST_τ with EoC condition $\chi = 1$ causes $V'(q^*) = 1$ and $V''(q^*) > 0$ resulting in unstable GP and failure to train:





Magnitude clipping recovers stable GP, e.g.
$\text{CReLU}_{\tau,m}(x) = max(\text{ReLU}_\tau(x), m)$ and

$$\text{CST}_{\tau,m}(x) = \begin{cases} 0, & \text{if } |x| < \tau \\ x - \text{sign}(x)\tau, & \text{if } \tau < |x| < \tau + m \\ \text{sign}(x)m, & \text{if } |x| > \tau + m. \end{cases}$$

Accuracy vs sparsity for clipped soft thresholding $CST_{\tau,m}$

Fully connected shows full accuracy at 85% sparsity

			DNN on MNIST		CNN on CIFAR10	
s	m	$V'(q^*)$	Acc.	Sparsity	Acc.	Sparsity
0.5	2	0.9	0.92	0.5	0.71	0.5
0.7	1.2	0.7	0.94	0.7	0.68	0.67
0.8	0.72	0.5	0.92	0.80	0.66	0.79
	1.06	0.7	0.95	0.80	0.65	0.78
	1.61	0.9	0.11	0.15	0.31	0.18
0.85	0.67	0.5	0.76	0.85	0.63	0.84
	1	0.7	0.93	0.85	0.63	0.83
	1.5	0.9	0.14	0.11	0.29	0.12

The clipping magnitude, m, controls the stability of the GP by decreasing $V'(q^*)$ but lower values of m limit accuracy

Summary: Sparsity and low-rank structure for efficient DNN

A widely studied topic since 2018, increasingly important with LLMs

- Sparsity and low-rank allow gradual trade-off between accuracy, number of parameters, and robustness
- There are a numerous algorithms and architecture specific variants.
- The focus of pruning and sketching is typically on computationally efficiency, but it also has benefits for robustness; a property far less explored.
- Sparsity inducing activations can be designed to allow training with high fractions of sparsity
- Attention matrices are notably different sparsity structures than need more specialized approximations