



ADAHESSIAN and PyHessian Tutorial

Zhewei Yao, Amir Gholami, Kurt Keutzer, Michael Mahoney

Rise Camp 2020

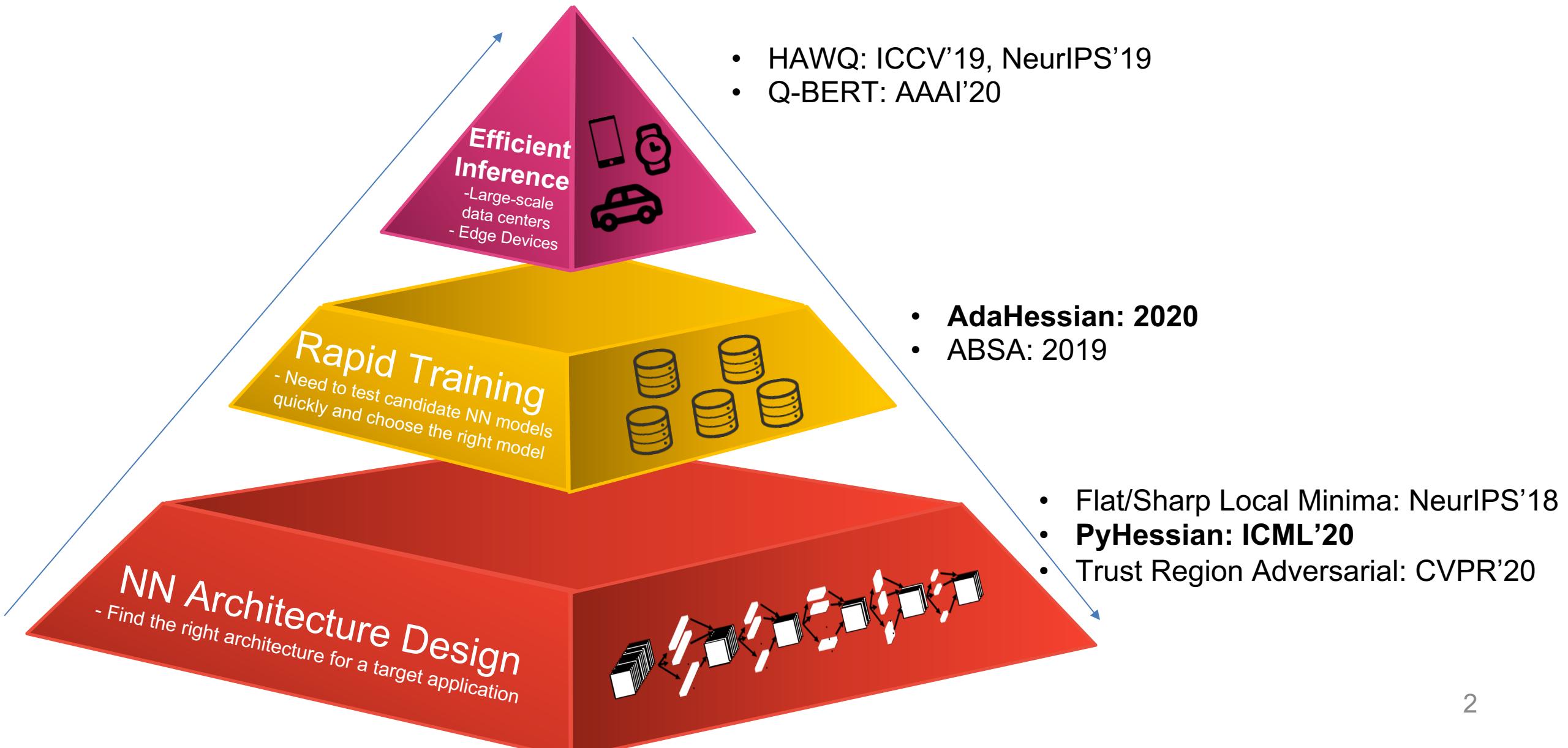


Berkeley
UNIVERSITY OF CALIFORNIA

riselab
UC Berkeley



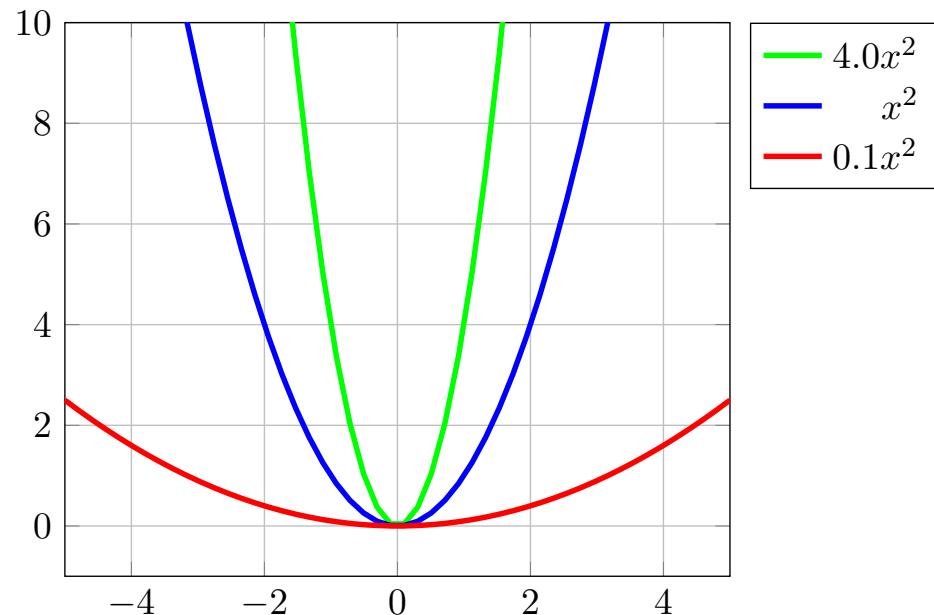
Second Order Method for DNNs



Quantifying “Sharpness”

Second-order information

- Refers to information about a function gained by computing its second derivative
- Reveals information about the function's curvature.

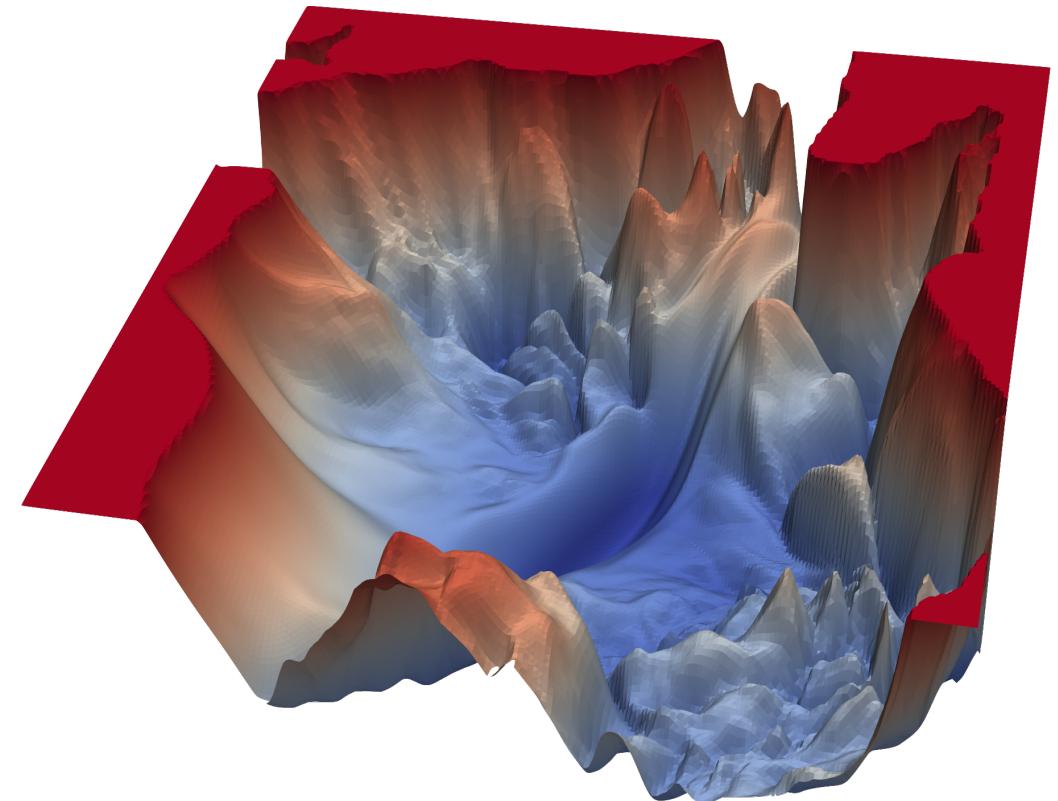
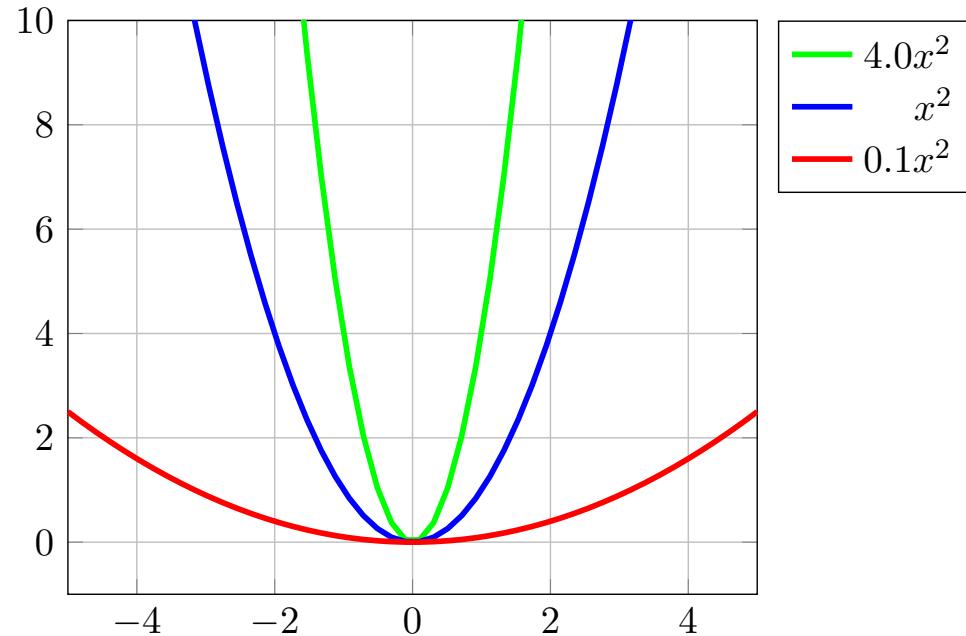


- At the origin, the first derivative of $y = 4x^2$, $y = x^2$, $y = 0.1x^2$ is all the same: 0
- The **second derivative** give more information: 8, 2, and 0.2 respectively

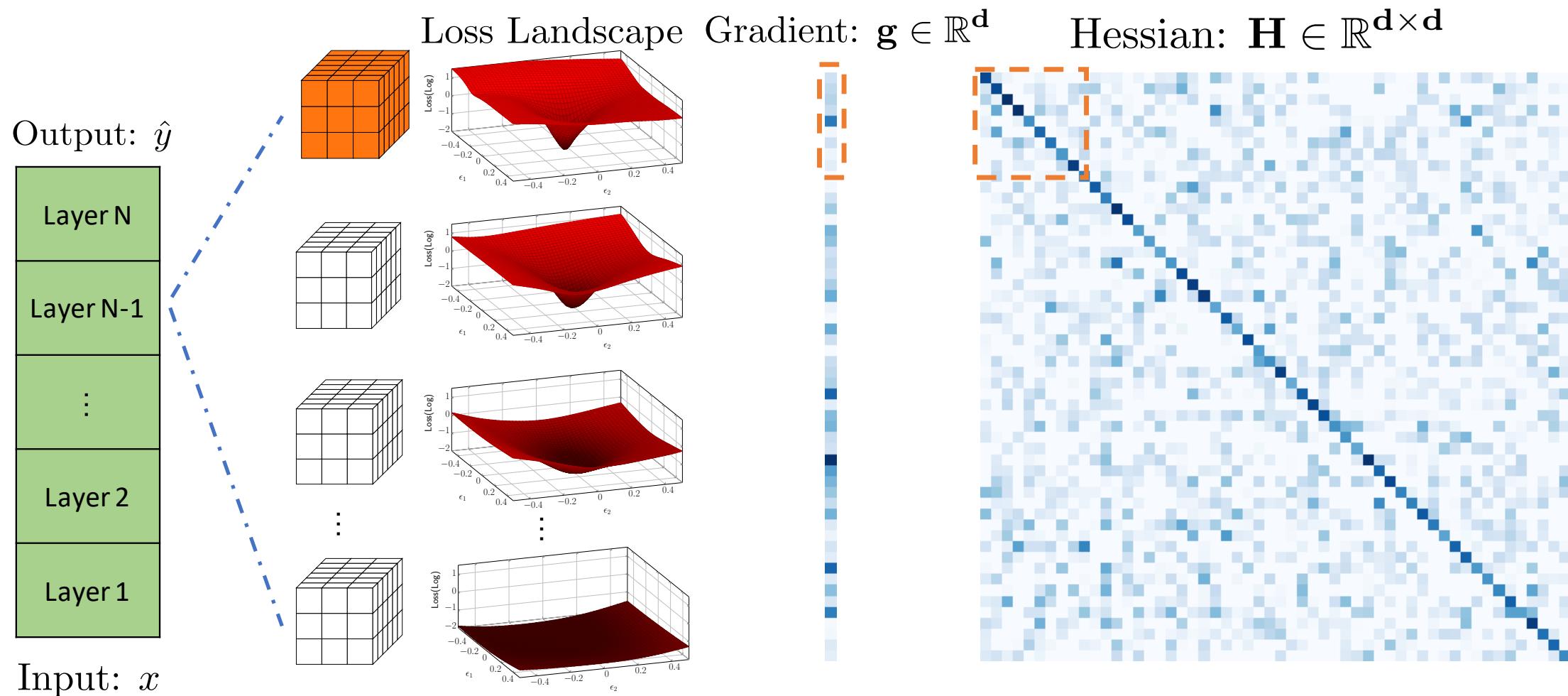
What is second order information?

Second-order information

- Refers to information about a function gained by computing its second derivative
- Reveals information about the function's curvature.



Opening the Black Box with Second Derivative



Using Hessian Diagonal

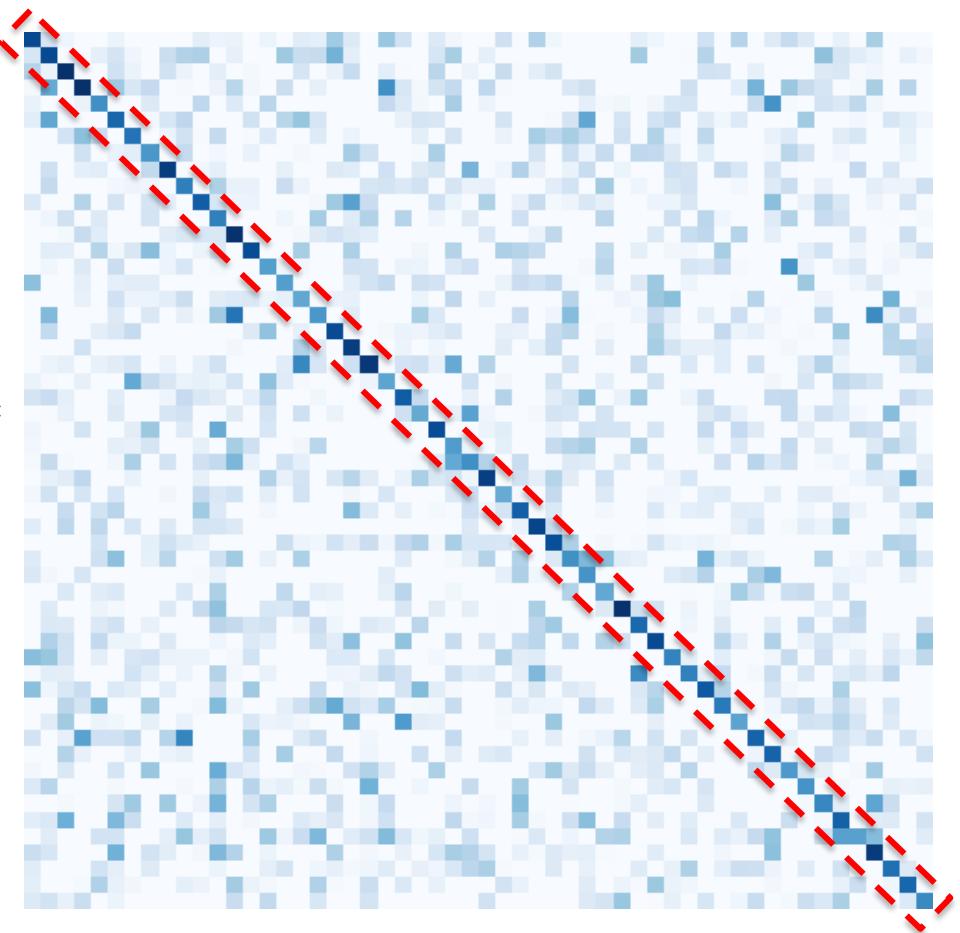
Forming the Hessian is computationally
infeasible:

For ResNet50 with 24M params Hessian is
a matrix of size **24Mx24M**

But what if we just approximate the
Hessian?

Idea: Use Hessian diagonal

$$g = \text{Diag}(H) =$$



Pearlmutter BA. Fast exact multiplication by the Hessian. Neural computation. 1994.

Costas Bekas, Effrosyni Kokiopoulou, and Yousef Saad. An estimator for the diagonal of a matrix. Applied numerical mathematics, 57(11-12):1214– 1229, 2007

Z. Yao*, A. Gholami*, Q. Lei, K. Keutzer, M. Mahoney, Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NeurIPS'18, 2018.

Z. Yao*, A. Gholami*, K. Keutzer, M. Mahoney, PyHessian: Neural Networks Through the Lens of the Hessian, Spotlight at ICML'20 workshop on Beyond First-Order Optimization Methods in Machine Learning Workshop, 2020.

Code: <https://github.com/amirgholami/PyHessian>

Hutchinson's Algorithm

Algorithm 4: Hutchinson Method for Trace Computation

Input: Parameter: θ .

Compute the gradient of θ by backpropagation, *i.e.*, compute $g_\theta = \frac{dL}{d\theta}$.

for $i = 1, 2, \dots$ **do** // Hutchinson Steps

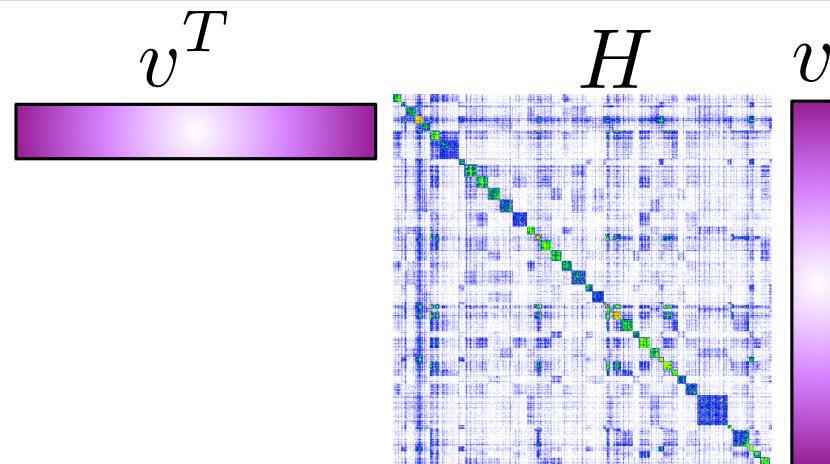
 Draw a random vector v from Rademacher distribution (same dimension as θ).

 Compute $gv = g_\theta^T v$

 Compute Hv by backpropagation, $Hv = \frac{d(gv)}{d\theta}$

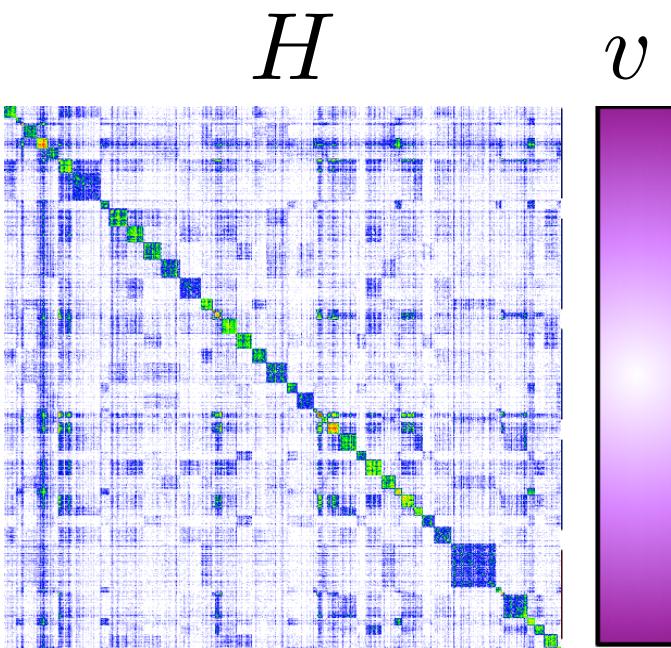
 Compute and record $v^T Hv$

Return the average of all computed $v^T Hv$.



How to Compute Sharpness?

- How to compute Hessian matrix vector multiply without forming Hessian?



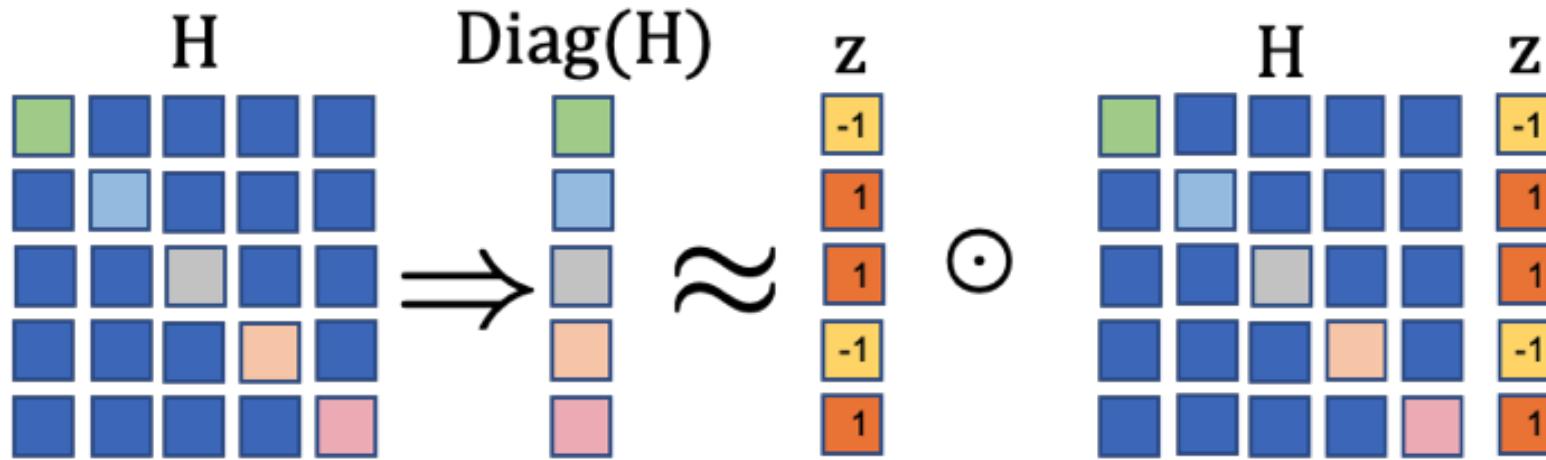
$$g = \frac{dL}{d\theta}$$

$$Hv = \frac{dg^T v}{d\theta} = \frac{dg^T}{d\theta} v + g^T \frac{dv}{d\theta} = \frac{dg^T}{d\theta} v$$

Hessian-vector Product

For a lot of applications, the explicit form of Hessian is not needed. The only requirement is the Hessian-vector product:

$$\frac{\partial g^T v}{\partial w} = \frac{\partial g^T}{\partial w} v + g^T \frac{\partial v}{\partial w} = \frac{\partial g^T}{\partial w} v = Hv.$$



$$\begin{aligned}\text{Diag}(H) &= \mathbb{E}[z \odot (Hz)] \\ \text{s.t. } z &\sim \text{Rademacher}(0.5)\end{aligned}$$

PyHessian Library

The screenshot shows the GitHub repository page for PyHessian. At the top, it displays the repository name "amirgholami / PyHessian" with metrics: 9 stars, 176 forks, and 23 open issues. Below this is a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, and a dropdown menu. The "Code" tab is selected, indicated by a red underline. On the left, there's a sidebar with a "master" dropdown, "Go to file", "Add file", and a "Code" dropdown. The main content area shows a list of commits from "amirgholami". The commits are:

- Update License to MIT (18 days ago)
- checkpoints init commit (10 months ago)
- misc added new illustration for Hessian (2 months ago)
- models init commit (10 months ago)
- pyhessian Update hessian.py (7 months ago)
- .gitignore added publication list (2 months ago)
- Hessian_Tutorial.i... added comments for the tutorial (last month)
- LICENSE Update License to MIT (18 days ago)
- README.md Update README.md (last month)
- density_plot.py init commit (10 months ago)
- example_pyhessi... Change syntax of raise exception in L11... (8 months ago)

On the right side, there are sections for "About", "Readme", "MIT License", "Releases" (with a link to "Create a new release"), and "Packages" (with a link to "Publish your first package").

PyHessian enables:

- Top-k eigenvalues
- Hessian Diagonal
- Hessian Trace
- Estimated Spectral Distribution

PyHessian: <https://github.com/amirgholami/PyHessian>

Motivation

Choosing the right hyper-parameters for optimizing a NN training has become a **dark-art!**

Problems with existing first-order solutions:

- Brute force hyper-parameters tuning
- Even the choice of the optimizer is a hyper-parameter!

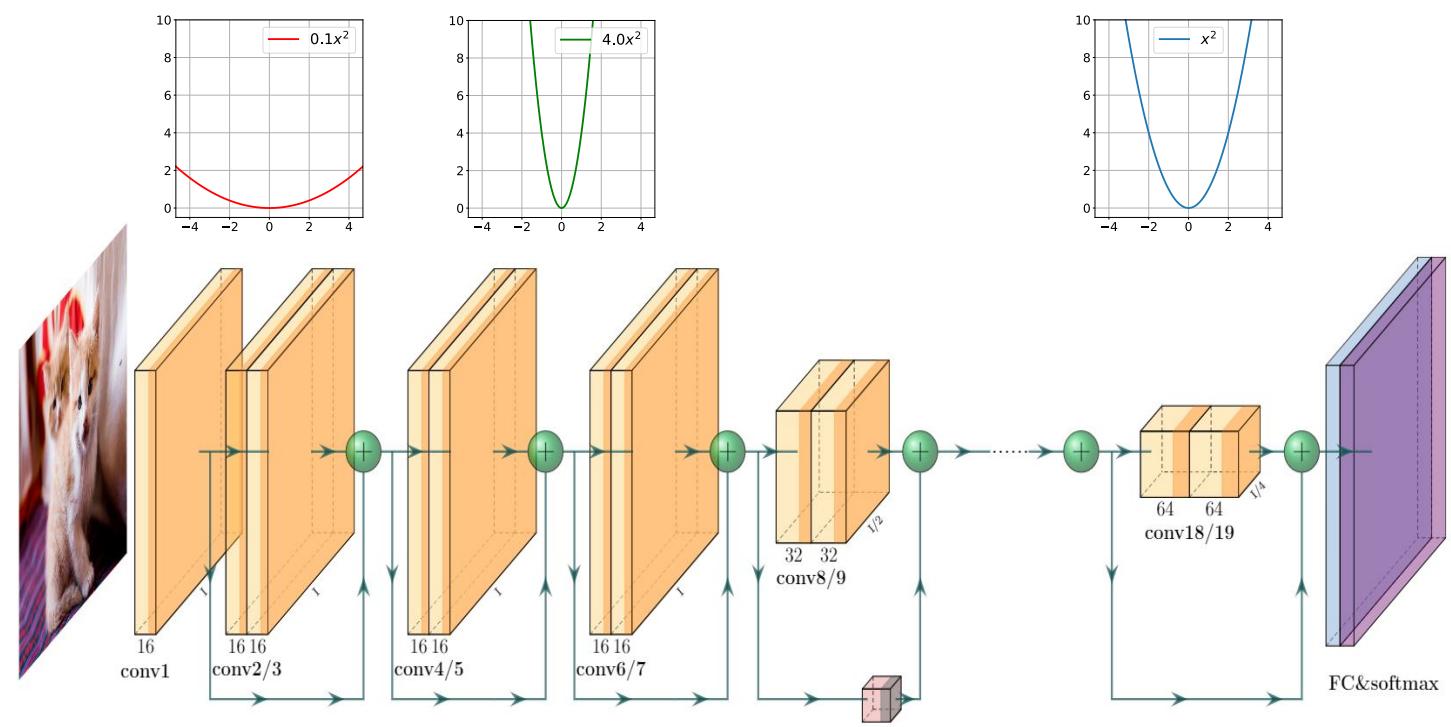
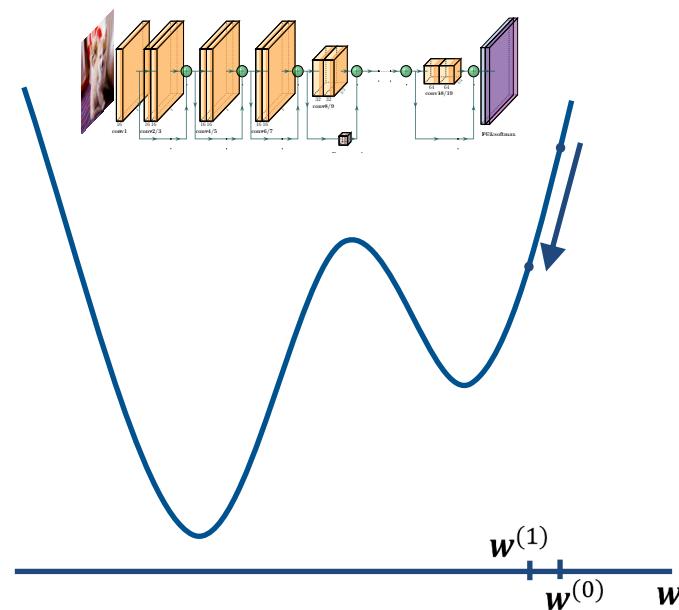


Task	CV	NLP	Recommendation System
Optimizer Choice	SGD	AdamW	Adagrad

SGD Based Training

$$\min_w E(w) = \frac{1}{N} \sum_{i=1}^N cost(w, x_i)$$

$$w^1 = w^0 - \frac{\lambda}{B} \sum_{i=1}^B \frac{\partial E_i(w^0)}{\partial w}$$

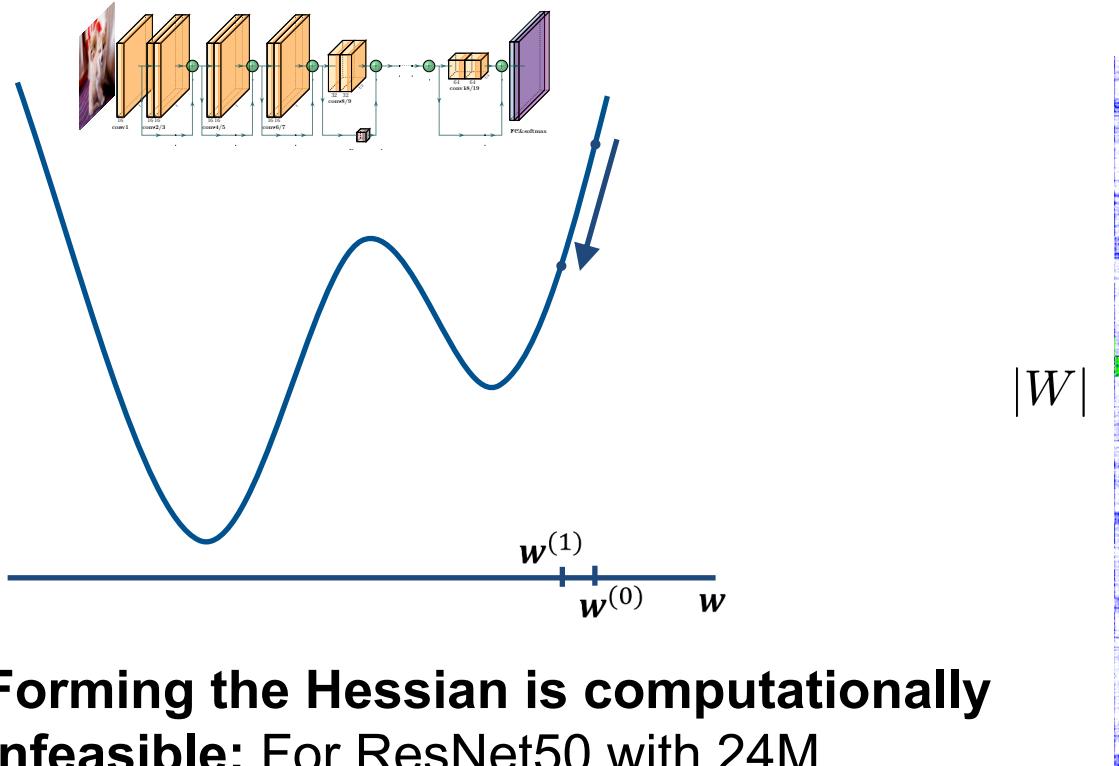


Hessian for DNNs

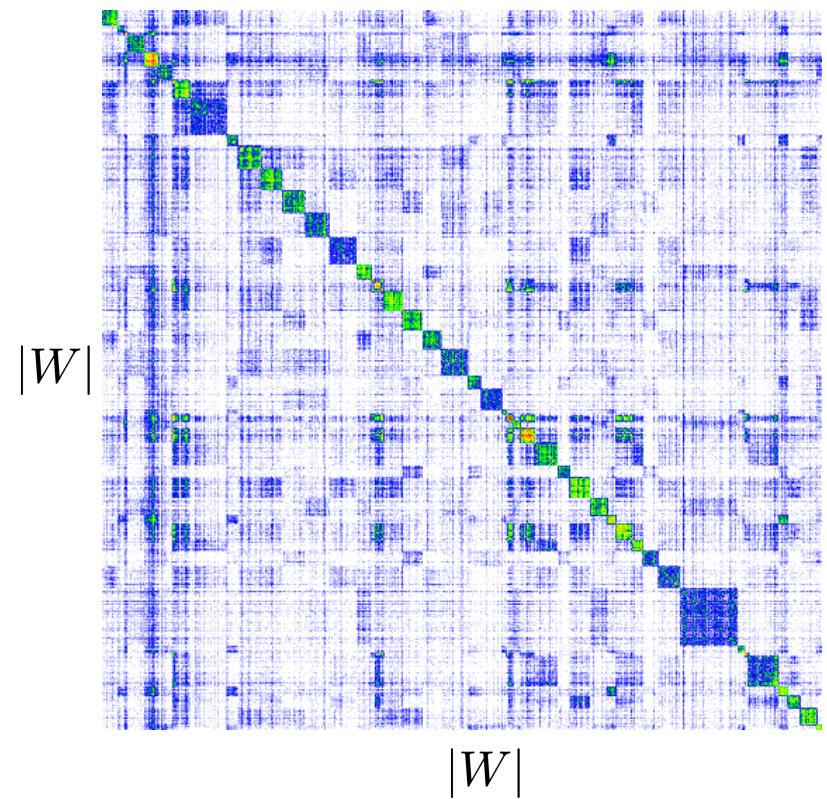
Loss: $\min_w E = \sum_{i=1}^N l(f(x_i; w), y_i)$

Gradient: $\frac{\partial E}{\partial w} \in \mathcal{R}^{|W|}$

Hessian: $\frac{\partial^2 E}{\partial w^2} \in \mathcal{R}^{|W| \times |W|}$



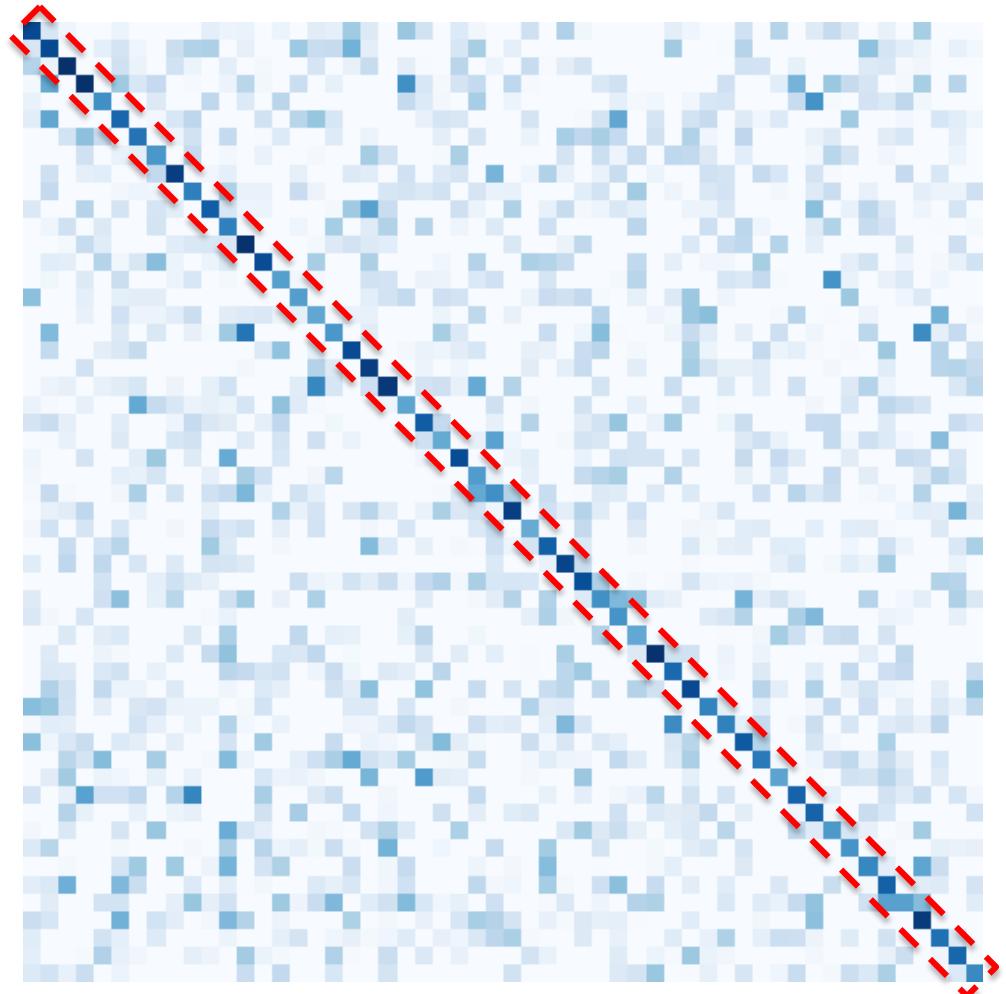
Forming the Hessian is computationally infeasible: For ResNet50 with 24M parameters, the Hessian is a matrix of size 24Mx24M (more than 2PB storage).



AdaHessian

Table 1: Summary of the first and second moments used in different optimization algorithms for updating model parameters ($w_{t+1} = w_t - \eta m_t / v_t$). Here β_1 and β_2 are first and second moment hyperparameters.

Optimizer	m_t	v_t
SGD [36]	$\beta_1 m_{t-1} + (1 - \beta_1) \mathbf{g}_t$	1
Adagrad [16]	\mathbf{g}_t	$\sqrt{\sum_{i=1}^t \mathbf{g}_i \mathbf{g}_i}$
Adam [21]	$\frac{(1 - \beta_1) \sum_{i=1}^t \beta_1^{t-i} \mathbf{g}_i}{1 - \beta_1^t}$	$\sqrt{\frac{(1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \mathbf{g}_i \mathbf{g}_i}{1 - \beta_2^t}}$
RMSProp [40]	\mathbf{g}_t	$\sqrt{\beta_2 v_{t-1}^2 + (1 - \beta_2) \mathbf{g}_t \mathbf{g}_t}$
ADAHESSIAN	$\frac{(1 - \beta_1) \sum_{i=1}^t \beta_1^{t-i} \mathbf{g}_i}{1 - \beta_1^t}$	$\left(\sqrt{\frac{(1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \mathbf{D}_i^{(s)} \mathbf{D}_i^{(s)}}{1 - \beta_2^t}} \right)^k$



Hessian: $\frac{\partial^2 E}{\partial w^2} \in \mathcal{R}^{|W| \times |W|}$ 14

Results on Machine Translation/Language Modeling

Only learning rate and space averaging block size are tuned for ADAHESSIAN

Higher BLEU score is better

Model	IWSLT14	WMT14
	small	base
SGD	$28.57 \pm .15$	26.04
AdamW [24]	$35.66 \pm .11$	28.19
ADAHESSIAN	$35.79 \pm .06$	28.52

Only learning rate and space averaging block size are tuned for ADAHESSIAN

Lower perplexity is better

Model	PTB	Wikitext-103
	Three-Layer	Six-Layer
SGD	59.9 ± 3.0	78.5
AdamW [24]	54.2 ± 1.6	20.9
ADAHESSIAN	51.5 ± 1.2	19.9

AdaHessian Library

amirgholami / adahessian

Unwatch 7 Star 77 Fork

Code Issues 3 Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

amirgholami Merge pull request #6 from nestordemeure/patch-1 ... 246eb77 10 days ago 29 commits

image_classification reorganize the structure 2 months ago

imgs added block size averaging illustration 2 months ago

instruction added block size averaging illustration 2 months ago

transformer Update README.md last month

.gitignore added block size averaging illustration 2 months ago

LICENSE Update License to MIT last month

README.md added link to JAX implementation 11 days ago

README.md

About

ADAHESSIAN: An Adaptive Second Order Optimizer for Machine Learning

second-order-optimization hessian
hessian-free

Readme

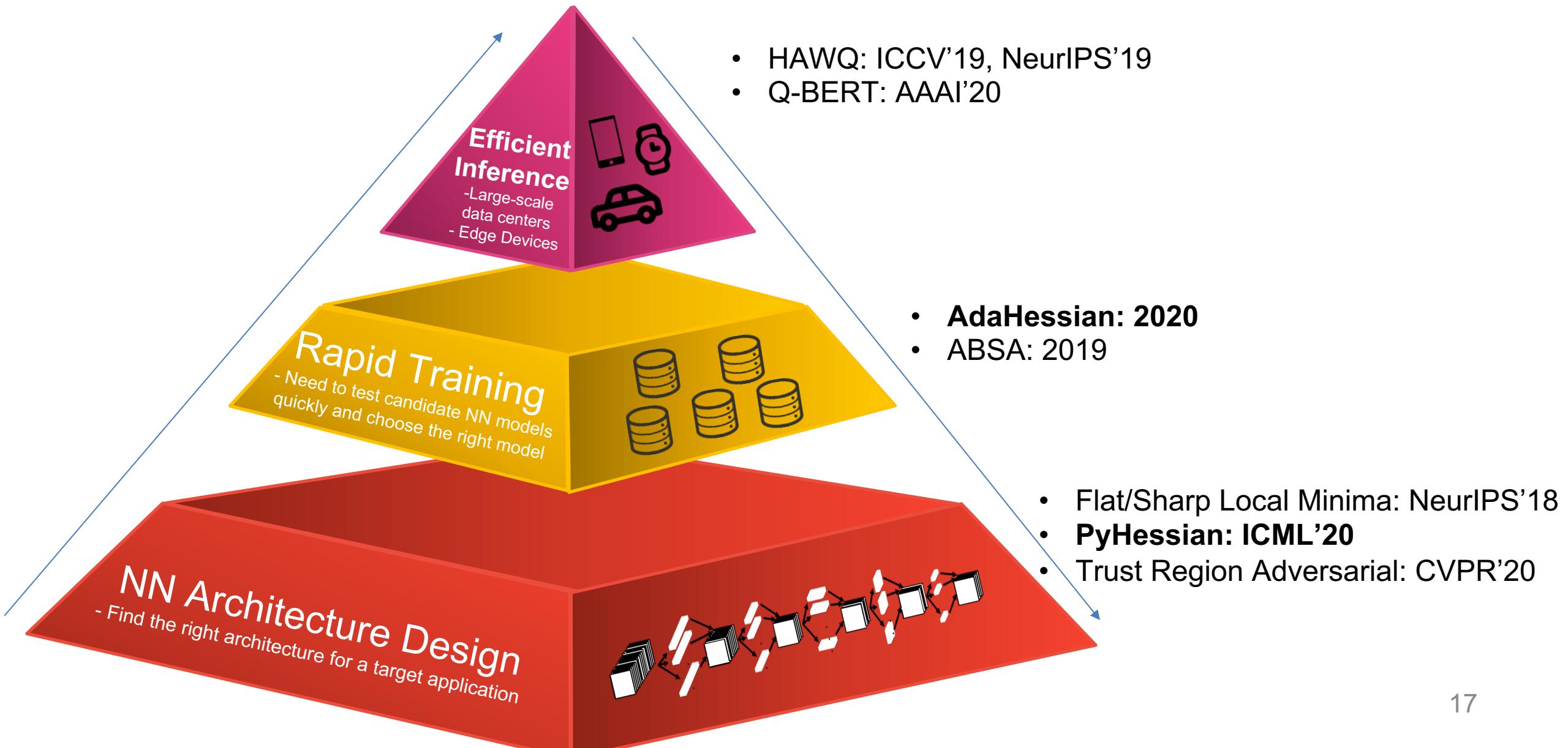
MIT License

Releases

No releases published Create a new release

AdaHessian: <https://github.com/amirgholami/adahessian>

Second Order Method for DNNs



Thank You!

Please contact us if you have any questions:

{zheweiy, amirgh} @ berkeley.edu

Hessian tutorial: <https://github.com/amirgholami/PyHessian/tree/master/pyhessian>

AdaHessian tutorial: https://github.com/yaozhewei/analyze_ada_hessian



Berkeley
UNIVERSITY OF CALIFORNIA

