University College London Educated Warm-Start team

Imraj Singh, Alexander Denker

Department of Computer Science
University College London

November 15, 2024

PET Rapid Image reconstruction Challenge

Goal of the challenge

Obtain an estimate to

$$\mathbf{x}_{\text{ref}} = \underset{\mathbf{x}\in\mathbb{R}_{\geq 0}}{\operatorname{argmax}}\{\Phi^{\mathbf{y}}(\mathbf{x}) := \mathcal{L}_{\mathbf{y}}(\mathbf{x}) - \beta R(\mathbf{x})\},$$

as **fast** as possible (measured in terms of computation time).

$\Rightarrow$ Combine deep learning as a warm start for an optimisation algorithm

How can Deep Learning help us?

Learning-to-Optimise[1], e.g.:

- Learn the full update: $\mathbf{x}_{k+1} = \mathsf{NN}_\theta(\mathbf{x}_k, \tilde{\nabla}_k)$
- Learn the preconditioner: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathsf{NN}_\theta(\mathbf{x}_k)\tilde{\nabla}_k$

$\Rightarrow$ Convergence to $\mathbf{x}_{\text{ref}}$ only under strong constraints ...

1 Chen et al. *Learning to Optimize: A Primer and A Benchmark*, JMLR 2022.

How can Deep Learning help us?

Learning-to-Optimise[1], e.g.:

- Learn the full update: $\mathbf{x}_{k+1} = \mathsf{NN}_\theta(\mathbf{x}_k, \tilde{\nabla}_k)$
- Learn the preconditioner: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathsf{NN}_\theta(\mathbf{x}_k)\tilde{\nabla}_k$

$\Rightarrow$ Convergence to $\mathbf{x}_{\text{ref}}$ only under strong constraints ...
However, for us we were not able to achieve a speed up: large 3D volumes, unstable training, few training samples, too much variety between volumes

[1] Chen et al. *Learning to Optimize: A Primer and A Benchmark*, JMLR 2022.

How can Deep Learning help us?

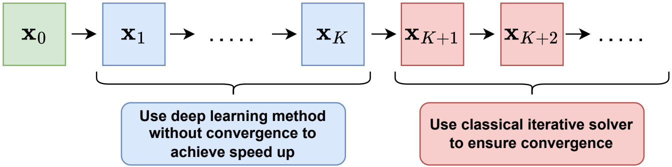
Learning-to-Optimise[1], e.g.:

- Learn the full update: $\mathbf{x}_{k+1} = \mathrm{NN}_\theta(\mathbf{x}_k, \tilde{\nabla}_k)$
- Learn the preconditioner: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathrm{NN}_\theta(\mathbf{x}_k)\tilde{\nabla}_k$

$\Rightarrow$ Convergence to $\mathbf{x}_{\text{ref}}$ only under strong constraints ...

However, for us we were not able to achieve a speed up: large 3D volumes, unstable training, few training samples, too much variety between volumes

$\Rightarrow$ Can combine deep learning with optimisation algorithm:



1 Chen et al. *Learning to Optimize: A Primer and A Benchmark*, JMLR 2022.

Educated Warm Start

Our approach:

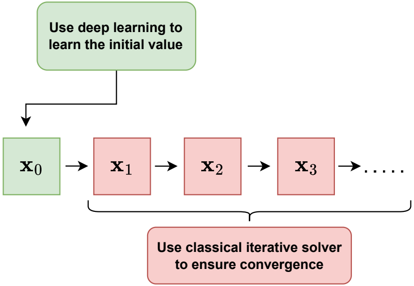
- Use neural network to provide a suitable warm start image $\mathbf{x}_0$
- Supervised train using mean-squared-error with $\{\mathbf{x}_{\text{OSEM},i}, \mathbf{x}_{\text{ref},i}\}_{i=1}^{N_s}$:

$$\min_{\theta} \sum_{i=1}^{N} \|\text{NN}_\theta(\mathbf{x}_{\text{OSEM},i}) - \mathbf{x}_{\text{ref},i}\|_2^2$$

- Make network 1-homogenous (no bias + ReLU activation), i.e.,

$$\text{NN}_\theta(\lambda \mathbf{x}) = \lambda \text{NN}_\theta(\mathbf{x}), \quad \lambda > 0$$

$\Rightarrow$ Easier generalisation to different intensities



Optimisation Algorithm

First-Order Optimisation

Given an initialisation $\mathbf{x}_0$, we iterate

$$\mathbf{x}_{k+1} = \mathcal{P}_{\geq 0}[\mathbf{x}_k + \alpha_k \mathbf{D}_k \tilde{\nabla}_k] \quad i = 0, 1, 2, \ldots$$

with $\mathcal{P}_{\geq 0}$ a non-negativity projection.

Choices:

1. Initialisation $\mathbf{x}_0$
2. Step size rule α_k
3. Preconditioner $\mathbf{D}_k$
4. Gradient approximation $\tilde{\nabla}_k \Rightarrow$ based on subsets of $\mathbf{y}$, i.e. $\nabla \Phi_k^{\mathbf{y}}(\mathbf{x}_k)$

$+$ extensions: acceleration or momentum terms

Algorithm 1 - SAGA with Momentum

- Step size $\alpha_k > 0$: Distance over weighted Gradient (DOwG):

$$\text{Distance estimator:} \quad \tilde{r}_k \leftarrow \max\left(\|\mathbf{x}_k - \mathbf{x}_0\|, \tilde{r}_{k-1}\right)$$

$$\text{Weighted gradient sum:} \quad v_k \leftarrow v_{k-1} + \tilde{r}_k^2 \|\tilde{\nabla}_k\|^2$$

$$\text{Step-size:} \quad \alpha_k \leftarrow \frac{\tilde{r}_k^2}{\sqrt{v_k}}$$

- Expectation Maximisation (EM) Preconditioner $\mathbf{D}_k = (\mathbf{x}_k + \varepsilon) \oslash \mathbf{A}^T \mathbf{1}$
- SAGA Gradient estimator $\tilde{\nabla}_k = \nabla \Phi^{\mathbf{y}}(\tilde{\mathbf{x}}) + \nabla \Phi_k^{\mathbf{y}}(\mathbf{x}_k) - \nabla \Phi_k^{\mathbf{y}}(\tilde{\mathbf{x}})$
- Katyusha-like momentum

$$\tilde{\nabla}_k = \nabla \Phi^{\mathbf{y}}(\tilde{\mathbf{x}}) + \nabla \Phi_k^{\mathbf{y}}(\mathbf{x}_k) - \nabla \Phi_k^{\mathbf{y}}(\tilde{\mathbf{x}})$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_k \tilde{\nabla}_k$$

$$\mathbf{x}_{k+1} = 0.5\mathbf{z}_{k+1} + 0.5\tilde{\mathbf{x}}$$

Subsets ordered in a staggered fashion, accessed in Herman-Meyer order, $\tilde{\mathbf{x}}$ is the last prediction of the previous epoch.

Algorithm 2 - SGD

- Step size $\alpha_k > 0$: Distance over weighted Gradient (DOwG)[2]
- Adaptive Preconditioner $\mathbf{D}_k$:

$$\mathbf{D}_k = \begin{cases} (\mathbf{x}_k + \varepsilon) \oslash \mathbf{A}^T \mathbf{1} & \text{if Poisson likelihood dominates} \\ \mathbf{1} \oslash (\kappa^2 + \mathrm{diag}[\nabla^2 R(\mathbf{x}_k)]) & \text{if RDP strong} \end{cases}$$

$\kappa^2 = \mathbf{A}^\top \mathrm{diag}\left[\frac{\mathbf{y}}{(\mathbf{A}\mathbf{x}_{\mathsf{OSEM}}+\mathbf{b})^2}\right] \mathbf{A1}$ - Approximate row-sum of likelihood Hessian

- SGD Gradient estimator $\tilde{\nabla}_k = \nabla \Phi_k^{\mathbf{y}}(\mathbf{x}_k)$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{D}_k \nabla \Phi_k^{\mathbf{y}}(\mathbf{x}_k)$$

Subsets ordered in a staggered fashion, accessed in Herman-Meyer order.

2 Khaled et al. *DOwG unleashed: An efficient universal parameter-free gradient descent method*, NIPS 2018.

Algorithm 3 - Full GD

- Step size $\alpha_k > 0$: Barzilai-Borwein long step size
- Adaptive Preconditioner $\mathbf{D}_k$:

$$\mathbf{D}_k = \begin{cases} (\mathbf{x}_k + \varepsilon) \oslash \mathbf{A}^T \mathbf{1} & \text{if Poisson likelihood dominates} \\ \mathbf{1} \oslash (\kappa^2 + \mathrm{diag}[\nabla^2 R(\mathbf{x}_k)]) & \text{if RDP strong} \end{cases}$$

$\kappa^2 = \mathbf{A}^\top \mathrm{diag}\left[\frac{\mathbf{y}}{(\mathbf{A}\mathbf{x}_{\mathsf{OSEM}} + \mathbf{b})^2}\right] \mathbf{A}\mathbf{1}$ - Approximate row-sum of likelihood Hessian

- Full GD Gradient estimator $\tilde{\nabla}_k = \nabla \Phi^{\mathbf{y}}(\mathbf{x}_k)$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{D}_k \nabla \Phi^{\mathbf{y}}(\mathbf{x}_k)$$

This is our "stable" method as we observed greatly varying performance of our subset-based algorithms between datasets.

Other ideas tested

Many different ideas were tested and it was found that ideas that worked on some datasets, would fail on others.

Deep learning ideas:

- input to network
- network architecture (unrolling)
- training loss

Optimisation ideas:

- Adaptive Preconditioner: Adam, Adadelta, Adamax
- Row-sum RDP hessian rather than diagonal
- EM-preconditioner + RDP Hessian approximation
- Asymptotically convergent implementation via gradient accumulation
- Momentum terms
- Lots of heuristic choices for step-sizes, initial step-size etc

Thank you for listening!

We'd also like to thank the organisers of the challenge, and Željko Kereta for his valuable discussions.

Considering the expected progress

For SGD the expected progress is[3]:

$$\mathbb{E}[\Phi_i^{\mathbf{y}}(\mathbf{x}_{k+1})] \leq \Phi^{\mathbf{y}}(\mathbf{x}_k) - \underbrace{\left(\alpha_k - \frac{L\alpha_k^2}{2}\right)||\nabla\Phi^{\mathbf{y}}(\mathbf{x}_k)||^2}_{\text{descent term}} + \underbrace{\frac{L\alpha_k^2\sigma_k^2}{2}}_{\text{variance term}}$$

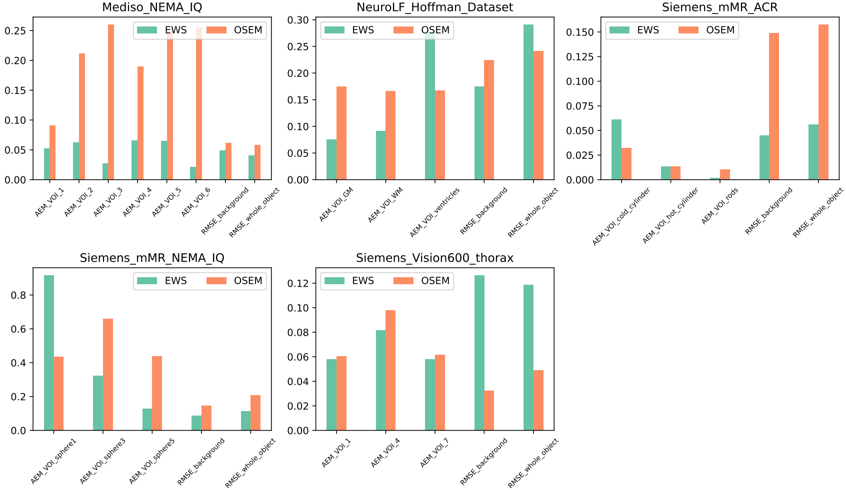
L and $\sigma_k = \mathbb{E}[\nabla\Phi_k^{\mathbf{y}}(\mathbf{x}_k) - \nabla\Phi^{\mathbf{y}}(\mathbf{x}_k)]$ vary significantly between datasets. Also α_k needs to trade-off descent and variance terms

This is a heuristic/hyperparameter tuning nightmare between datasets...

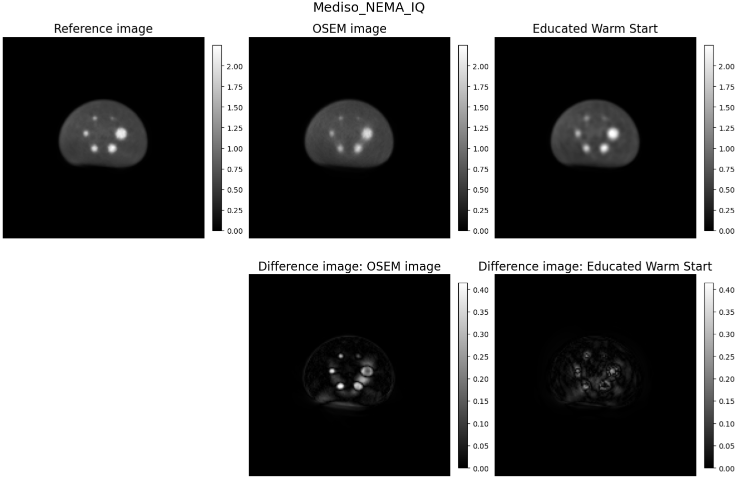
KISS principle: Keep It Simple Stupid!

3 Bottou et al. *Optimization Methods for Large-Scale Machine Learning*, SIAM Review 2018.

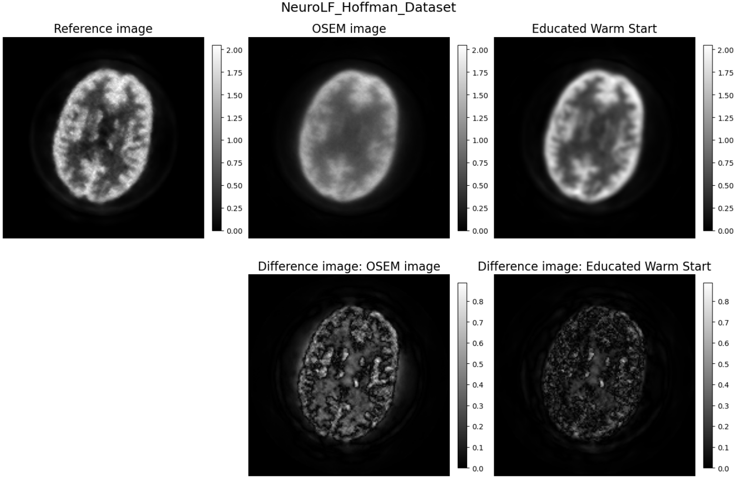
Educated Warm Start - Results



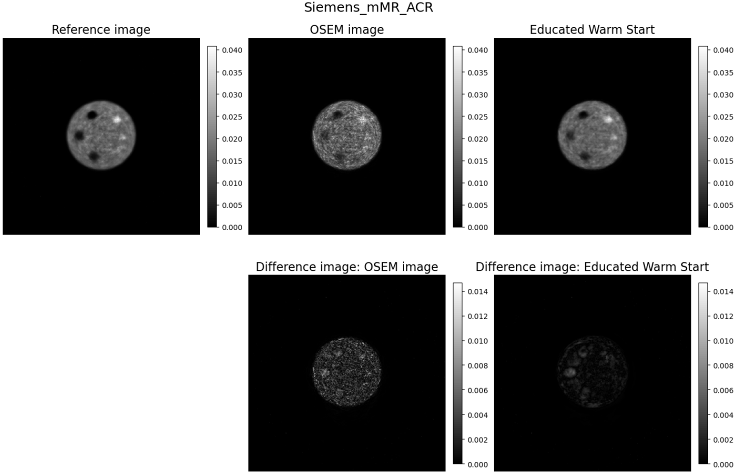
Educated Warm Start - Results



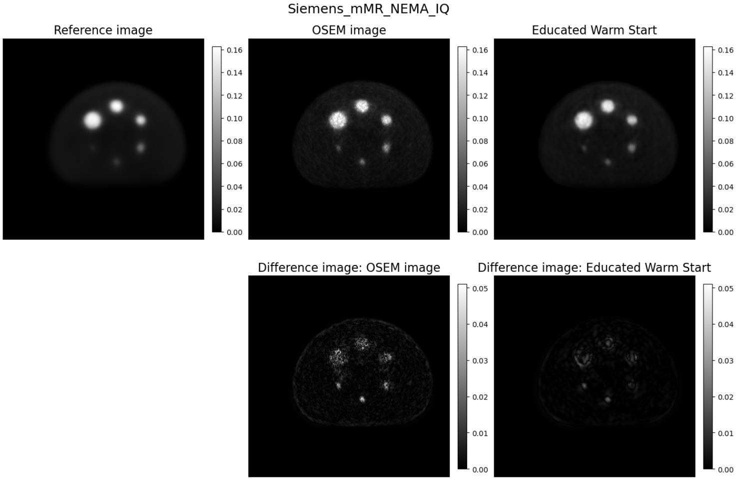
Educated Warm Start - Results



Educated Warm Start - Results



Educated Warm Start - Results



Educated Warm Start - Results