ICLR Socials

Optimization in ML and DL - A discussion on theory and practice

Tue 26 Apr & Thu 28 Apr @ 3:00am - 5:00am UTC /
Mon 25 Apr & Wed 27 Apr @ 8:00pm – 10:00pm PST /

Machine Learning and Fitness

Mon 25 Apr @ 8:00pm - 10:00pm PST

Speaker : **Jacob Rafati**
Founder of Workout Vision INC
https://www.linkedin.com/in/jacob-rafati/

Machine Learning and Fitness: Personal training and fitness processes are difficult to optimize manually without using machine learning methods. In this session, Jacob Rafati will talk about the fitness problems and the optimization methods that he is implementing at Workout Vision INC. More Details…

Second-order Optimization in ML/DL

Wed 27 Apr @ 8:00pm - 10:00pm PST

Speaker : **Indra Priyadarsini S**
Ph.D. Candidate, Shizuoka University
https://www.linkedin.com/in/indra-ipd/

Second-order Optimization in ML/DL: Optimization plays an important role in machine learning and deep learning. While first-order gradient-based methods are predominantly used as the first choice in ML and DL, second-order quasi-Newton (QN) methods are not commonly used despite their fast convergences. In this social, we will go through the effectiveness of second order methods in training neural networks and further look into its acceleration using Nesterov's gradient.

ICLR Socials

Optimization in ML and DL

A discussion on theory and practice

Tue 26 Apr & Thu 28 Apr @ 3:00am - 5:00am UTC /
Mon 25 Apr & Wed 27 Apr @ 8:00pm – 10:00pm PST /

Second-order Optimization for Training Neural Networks

S. Indrapriyadarsini

National University Corporation
Shizuoka University

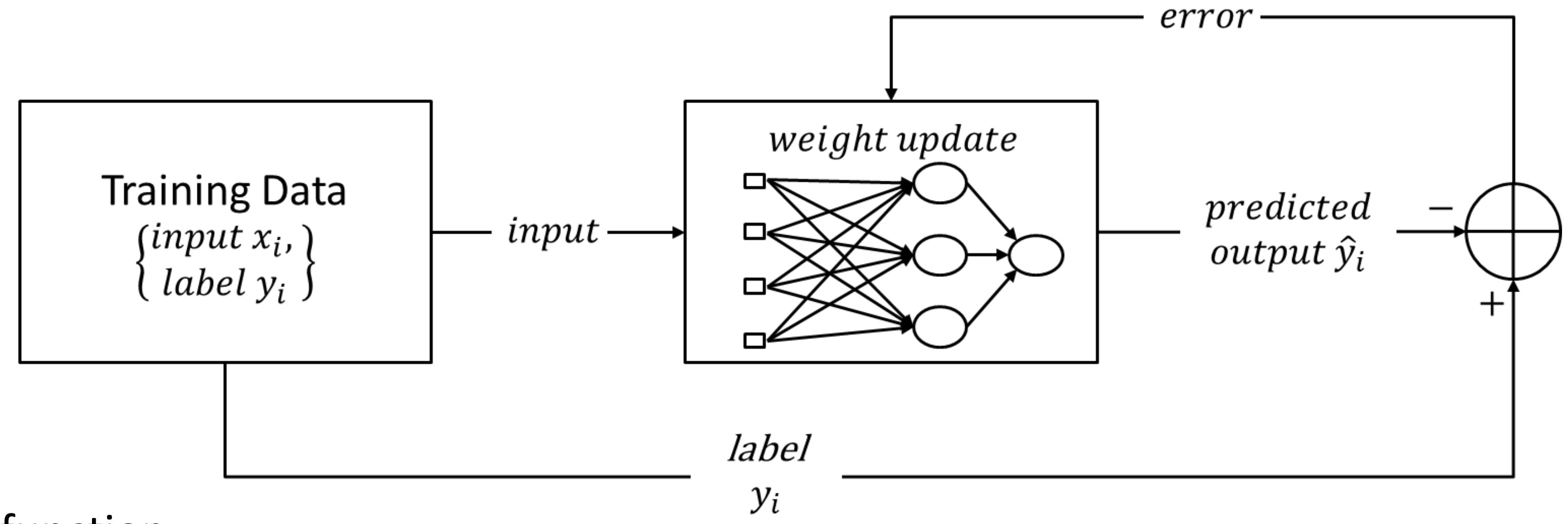
28th April 2022

Outline

- Introduction
- Gradient Based Training
  - First Order Methods
  - Second Order Methods
- Nesterov’s Accelerated quasi-Newton Methods

OPTIMIZATION IN SUPERVISED LEARNING

- Given a dataset (x_i, y_i)
- Neural network : Parameterized model to map function $f_w(x) \rightarrow y$



- Objective function

$$\min_{w \in \mathbb{R}^d} E(w) = \frac{1}{|T_r|} \sum_{i \in T_r} E_i(w) \quad \text{where} \quad E_i(w) = \frac{1}{2} \|y_i - \hat{y}_i\|^2 \quad \text{(Eg. MSE)}$$

GRADIENT BASED ALGORITHMS

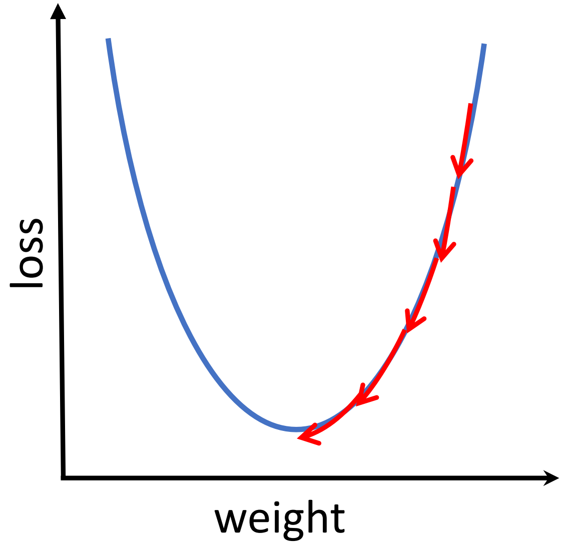
FIRST ORDER METHODS

- Slow convergence in highly non-linear problems
- Simple and low complexity

$$\boldsymbol{w} := \boldsymbol{w} - \alpha \frac{\partial E}{\partial \boldsymbol{w}}$$

Gradient $\nabla E(\boldsymbol{w})$

Classical Momentum
Nesterov’s Accelerated Gradient (NAG)
AdaGrad, RMSProp, Adam



SECOND/APPROXIMATED SECOND ORDER METHODS

- Faster convergence
- Suitable for highly non-linear problems
- High computational cost

$$\boldsymbol{w} := \boldsymbol{w} - \alpha \boldsymbol{H} \nabla E(\boldsymbol{w})$$

Hessian

Newton Method
Quasi-Newton Method (QN)
Nesterov's Accelerated quasi-Newton (NAQ)

FIRST ORDER ALGORITHMS

The weight vector is updated by the update vector $\boldsymbol{v_{k+1}}$ as

$$\boldsymbol{w_{k+1} = w_k + v_{k+1}} \qquad \dots (Eq.\,\mathbf{1})$$

Steepest gradient descent(SGD) with a step size α_k

$$\boldsymbol{v_{k+1} = -\alpha_k \nabla E(w_k)} \qquad \dots (Eq.\,\mathbf{2})$$

Normal Gradient

Classical momentum (CM) method

$$\boldsymbol{v_{k+1} = \mu v_k - \alpha_k \nabla E(w_k)} \qquad \dots (Eq.\,\mathbf{3})$$

Momentum term

Nesterov's Accelerated Gradient (NAG) method

$$\boldsymbol{v_{k+1} = \mu v_k - \alpha_k \nabla E(w_k + \mu v_k)} \qquad \dots (Eq.\,\mathbf{4})$$

Momentum term + Nesterov's Accelerated Gradient (NAG)

QUASI-NEWTON METHOD

The weight is updated with update vector $\boldsymbol{v}_{k+1}$ as:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \boldsymbol{v}_{k+1} \qquad \ldots(Eq.\,\mathbf{5})$$

The weight update of quasi-Newton (QN) method is given as

$$\boldsymbol{v}_{k+1} = -\alpha_k \boldsymbol{H}_k \nabla E(\boldsymbol{w}_k) \qquad \ldots(Eq.\,\mathbf{6})$$

Normal Gradient

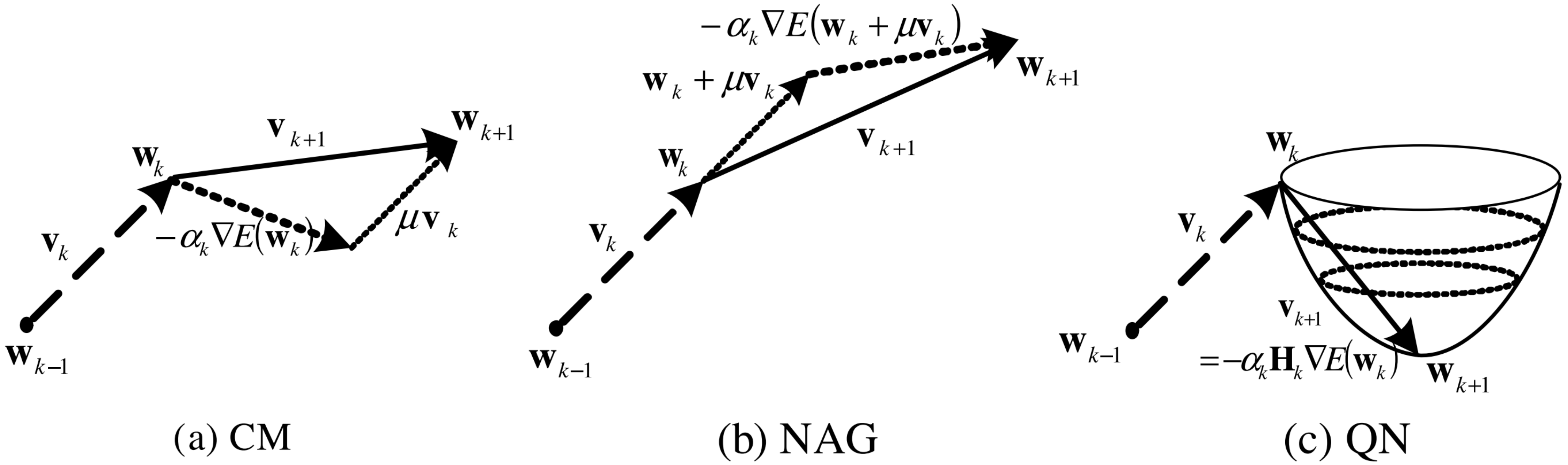
The matrix $\mathbf{H}_k$ is iteratively approximated by BFGS formula

$$\boldsymbol{H}_{k+1} = \left(\boldsymbol{I} - \rho_k \boldsymbol{s}_k {\boldsymbol{y}_k}^T\right) \boldsymbol{H}_k \left(\boldsymbol{I} - \rho_k \boldsymbol{y}_k {\boldsymbol{s}_k}^T\right) + \rho_k \boldsymbol{s}_k {\boldsymbol{s}_k}^T \qquad \ldots(Eq.\,\mathbf{7})$$

$$\rho_k = \frac{1}{{\boldsymbol{y}_k}^T \boldsymbol{s}_k}, \ \boldsymbol{s}_k = \boldsymbol{w}_{k+1} - \boldsymbol{w}_k \text{ and } \boldsymbol{y}_k = \nabla E(\boldsymbol{w}_{k+1}) - \nabla E(\boldsymbol{w}_k) \qquad \ldots(Eq.\,\mathbf{8})$$

Normal Gradients

GEOMETRIC VIEWS



Source: *H. Ninomiya, "A novel quasi-Newton-Optimization for neural network training incorporating Nesterov's accelerated gradient", IEICE NOLTA Journal, Oct. 2017.*

NESTEROV'S ACCELERATED QUASI-NEWTON METHOD (NAQ)

The update vector of NAQ

$$\boldsymbol{w}_{\boldsymbol{k+1}} = \boldsymbol{w}_{\boldsymbol{k}} + \mu \mathbf{v}_k - \alpha_k \boldsymbol{H}_{\boldsymbol{k}} \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}}) \quad \ldots (Eq.\mathbf{9})$$

Momentum term

Nesterov's Accelerated Gradient(NAG)

The matrix $\boldsymbol{H}_{\boldsymbol{k}}$ is iteratively approximated by

$$\boldsymbol{H}_{\boldsymbol{k+1}} = (\boldsymbol{I} - \rho_k \boldsymbol{p}_{\boldsymbol{k}} \boldsymbol{q}_{\boldsymbol{k}}{}^{\boldsymbol{T}}) \boldsymbol{H}_{\boldsymbol{k}} (\boldsymbol{I} - \rho_k \boldsymbol{q}_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}}{}^{\boldsymbol{T}}) + \rho_k \boldsymbol{p}_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}}{}^{\boldsymbol{T}} \quad \ldots (Eq.\mathbf{10})$$

$$\rho_k = \frac{\mathbf{1}}{\boldsymbol{q}_{\boldsymbol{k}}{}^{\boldsymbol{T}} \boldsymbol{p}_{\boldsymbol{k}}}, \; \boldsymbol{p}_{\boldsymbol{k}} = \boldsymbol{w}_{\boldsymbol{k+1}} - (\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}}) \text{ and } \boldsymbol{q}_{\boldsymbol{k}} = \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k+1}}) - \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}})$$

Two gradient computations per iteration

Normal Gradient

Nesterov's Accelerated Gradient(NAG)

H. Ninomiya, "A novel quasi-Newton-Optimization for neural network training incorporating Nesterov's accelerated gradient", IEICE NOLTA Journal, Oct. 2017.

MOMENTUM QUASI-NEWTON METHOD (MOQ)

The update vector of NAQ

$$\boldsymbol{w}_{\boldsymbol{k}+\mathbf{1}} = \boldsymbol{w}_{\boldsymbol{k}} + \mu \mathbf{v}_k - \alpha_k \boldsymbol{H}_{\boldsymbol{k}} \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}}) \quad \ldots (Eq.\mathbf{11})$$

Momentum term

Nesterov's Accelerated Gradient(NAG)

Nesterov's accelerated gradient approximation

$$\boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}}) \approx (1 + \mu_k) \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}}) - \mu_k \boldsymbol{\nabla E}(\mathbf{w}_{\mathbf{k}-\mathbf{1}}) \quad \ldots (Eq.\mathbf{12})$$

and the Hessian matrix $\boldsymbol{H}_{\boldsymbol{k}}$ is updated as

$$\boldsymbol{H}_{\boldsymbol{k}+\mathbf{1}} = (\boldsymbol{I} - \rho_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}} \boldsymbol{q}_{\boldsymbol{k}}{}^{\boldsymbol{T}}) \boldsymbol{H}_{\boldsymbol{k}} (\boldsymbol{I} - \rho_{\boldsymbol{k}} \boldsymbol{q}_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}}{}^{\boldsymbol{T}}) + \rho_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{k}}{}^{\boldsymbol{T}} \quad \ldots (Eq.\mathbf{13})$$

$$\rho_k = \frac{\mathbf{1}}{\boldsymbol{q}_{\boldsymbol{k}}{}^{\boldsymbol{T}} \boldsymbol{p}_{\boldsymbol{k}}}, \; \boldsymbol{p}_{\boldsymbol{k}} = \boldsymbol{w}_{\boldsymbol{k}+\mathbf{1}} - (\boldsymbol{w}_{\boldsymbol{k}} + \mu \boldsymbol{v}_{\boldsymbol{k}}) \text{ and } \boldsymbol{q}_{\boldsymbol{k}} = \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}+\mathbf{1}}) - \{(1 + \mu_k) \boldsymbol{\nabla E}(\boldsymbol{w}_{\boldsymbol{k}}) - \mu_k \boldsymbol{\nabla E}(\mathbf{w}_{\mathbf{k}-\mathbf{1}})\}$$

Shahrzad Mahboubi, S. Indrapriyadarsini, Hiroshi Ninomiya, Hideki Asai, "Momentum acceleration of quasi-Newton Training for Neural Networks", 16th Pacific Rim International Conference on Artificial Intelligence, PRICAI 2019, (pp. 268-281). Springer, Cham.

OBJECTIVES

- Study behavior of first and second order methods in training neural networks
- Investigate and propose Nesterov and momentum accelerated second order methods for training neural networks
- Demonstrate robustness and efficiency of Nesterov and momentum accelerated quasi-Newton methods

QUASI-NEWTON METHOD

- $$E(\mathbf{w}_k + \mathbf{d}) \approx m_k(\mathbf{d}) \approx E(\mathbf{w}_k) + \nabla E(\mathbf{w}_k)^T \mathbf{d} + \frac{1}{2}\, \mathbf{d}^T \nabla^2 E(\mathbf{w}_k) \mathbf{d}. \quad \dots (Eq.\, \mathbf{14})$$

- The minimizer $\mathbf{d}_k$ is given as

$$\mathbf{d}_k = -\nabla^2 E(\mathbf{w}_k)^{-1} \nabla E(\mathbf{w}_k) = -\mathbf{B}_k^{-1} \nabla E(\mathbf{w}_k)\,. \quad \dots (Eq.\, \mathbf{15})$$

- The new iterate $\mathbf{w}_{k+1}$ is given as,

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \mathbf{B}_k^{-1} \nabla E(\mathbf{w}_k), \quad \dots (Eq.\, \mathbf{16})$$

and the quadratic model at the new iterate is given as

$$E(\mathbf{w}_{k+1} + \mathbf{d}) \approx m_{k+1}(\mathbf{d}) \approx E(\mathbf{w}_{k+1}) + \nabla E(\mathbf{w}_{k+1})^T \mathbf{d} + \frac{1}{2}\, \mathbf{d}^T \mathbf{B}_{k+1} \mathbf{d}\,, \quad \dots (Eq.\, \mathbf{17})$$

QUASI-NEWTON METHOD + NESTEROV'S ACCELERATION

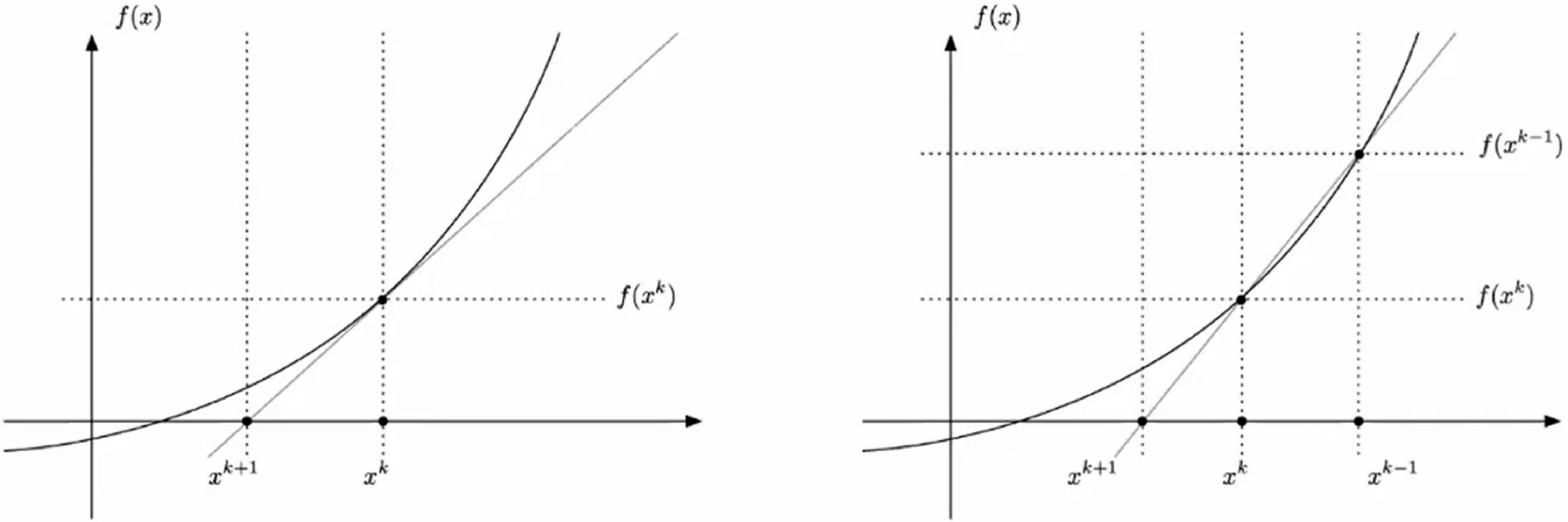
- The Nesterov's acceleration approximates the quadratic model at $\mathbf{w_k} + \mu\mathbf{v_k}$ instead of the iterate at $\mathbf{w_k}$

The new iterate $\mathbf{w}_{k+1}$ is given as,

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_k \mathbf{v}_k - \alpha_k \mathbf{B}_k^{-1} \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k)$$

$$= \mathbf{w}_k + \mu_k \mathbf{v}_k + \alpha_k \mathbf{d}_k.$$

QUASI-NEWTON METHOD : SECANT CONDITION



Newton: $\boldsymbol{B}^k = D\boldsymbol{F}(\boldsymbol{x}^k)$

Direct: $\boldsymbol{B}^k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}) = \boldsymbol{F}(\boldsymbol{x}^k) - \boldsymbol{F}(\boldsymbol{x}^{k-1})$

Dual: $\boldsymbol{x}^k - \boldsymbol{x}^{k-1} = \boldsymbol{H}^k(\boldsymbol{F}(\boldsymbol{x}^k) - \boldsymbol{F}(\boldsymbol{x}^{k-1}))$

QUASI-NEWTON METHOD + NESTEROV'S ACCELERATION

We have

$$E(\boldsymbol{w}_k + d) \approx m_k(\boldsymbol{d})$$
$$E(\boldsymbol{w}_{k+1} + d) \approx m_{k+1}(\boldsymbol{d})$$

Require:

m_{k+1} matches the gradient at the previous **two** iterations, i.e.,

1. $$\nabla m_{k+1}|_{\mathbf{d}=0} = \nabla E(\mathbf{w}_{k+1} + \mathbf{d})|_{\mathbf{d}=0} = \nabla E(\mathbf{w}_{k+1}).$$

2. $$\nabla m_{k+1}|_{\mathbf{d}=-\alpha_k \mathbf{d}_k} = \nabla E(\mathbf{w}_{k+1} + \mathbf{d})|_{\mathbf{d}=-\alpha_k \mathbf{d}_k} = \nabla E(\mathbf{w}_{k+1} - \alpha_k \mathbf{d}_k) = \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k)$$

QUASI-NEWTON METHOD + NESTEROV'S ACCELERATION

Proof: $E(\boldsymbol{w}_{k+1} + d) \approx m_{k+1}(\boldsymbol{d}) = E(\boldsymbol{w}_{k+1}) + \nabla E(\boldsymbol{w}_{k+1})^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d}$

Condition 1: $\nabla m_{k+1}|_{\mathbf{d}=0} = \nabla E(\mathbf{w}_{k+1} + \mathbf{d})|_{\mathbf{d}=0} = \nabla E(\mathbf{w}_{k+1})$.

$$\nabla m_{k+1}(\boldsymbol{d}) = \nabla E(\boldsymbol{w}_{k+1}) + \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d}$$

$$\nabla m_{k+1}(0) = \nabla E(\boldsymbol{w}_{k+1}) + \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d} \ \ |_{\boldsymbol{d}=0} \Rightarrow \boldsymbol{satisfied}$$

Condition 2: $\nabla m_{k+1}|_{\mathbf{d}=-\alpha_k \mathbf{d}_k} = \nabla E(\mathbf{w}_{k+1} + \mathbf{d})|_{\mathbf{d}=-\alpha_k \mathbf{d}_k} = \nabla E(\mathbf{w}_{k+1} - \alpha_k \mathbf{d}_k) = \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k)$

$$\nabla m_{k+1}(-\alpha \boldsymbol{d}_k) = \nabla E(\boldsymbol{w}_{k+1}) - \alpha \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d}_k$$

$$\nabla m_{k+1}(-\alpha \boldsymbol{d}_k) = \nabla E(\boldsymbol{w}_{k+1}) - \alpha \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d}_k = \nabla E(\boldsymbol{w}_{k+1} - \alpha \boldsymbol{d}_k) = \nabla E(\boldsymbol{w}_k + \mu \boldsymbol{v}_k)$$

$$\nabla E(\boldsymbol{w}_{k+1}) - \alpha \nabla^2 E(\boldsymbol{w}_{k+1})\boldsymbol{d}_k = \nabla E(\boldsymbol{w}_k + \mu \boldsymbol{v}_k)$$

$$\nabla E(\boldsymbol{w}_{k+1}) - \nabla E(\boldsymbol{w}_k + \mu \boldsymbol{v}_k) = \boldsymbol{B}_{k+1}(\boldsymbol{w}_{k+1} - (\boldsymbol{w}_k + \mu \boldsymbol{v}_k))$$

$$\boldsymbol{q}_k = \boldsymbol{B}_{k+1}\boldsymbol{p}_k \quad \Rightarrow \boldsymbol{Secant\ Condition}$$

$$(\boldsymbol{p}_k, \boldsymbol{q}_k) \Rightarrow \boldsymbol{Curvature\ Information\ Pair}$$

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

BEALE FUNCTION

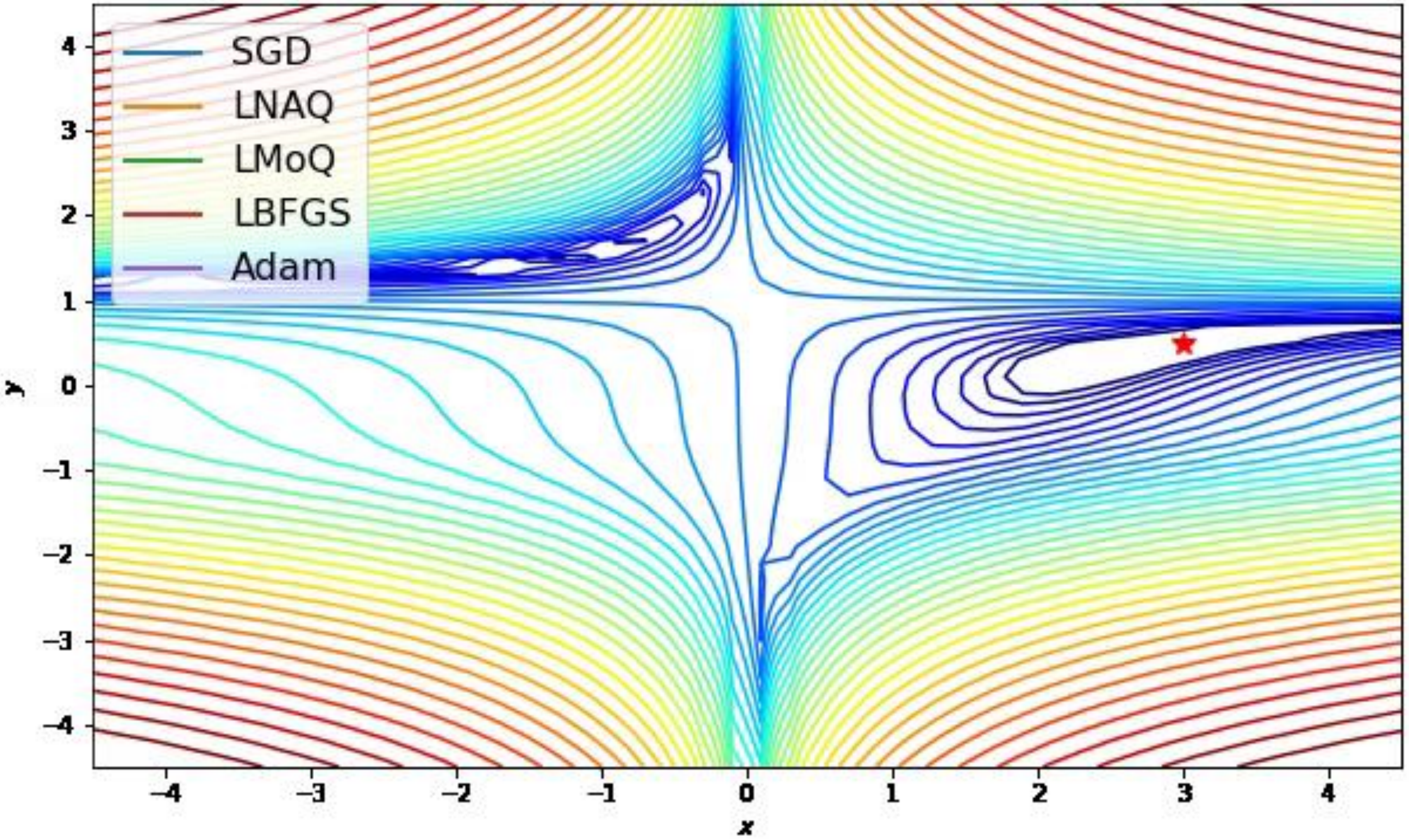
The Beale function is multimodal, with sharp peaks at the corners of the input domain

Unconstrained test function

$$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$$

Global minimum

$f(x^*) = 0$ at $x^* = (3,0.5)$



Global Optimization Test Problems. Retrieved June 2013, from http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.

Modified Nesterov's Accelerated BFGS quasi-Newton – mNAQ

1) Incorporating an additional $\hat{\xi}_k p_k$ term for better convergence

$$p_k = w_{k+1} - (w_k + \mu v_k)$$

$$q_k = \nabla E(w_{k+1}) - \nabla E(w_k + \mu v_k) + \hat{\xi}_k p_k = \varepsilon_k + \hat{\xi}_k p_k$$

⇒ Modified Secant Condition

$$\hat{H}_{k+1} = (I - \rho_k p_k {q_k}^T)\hat{H}_k(I - \rho_k q_k {p_k}^T) + \rho_k p_k {p_k}^T$$

convergence term

2) Eliminating linesearch

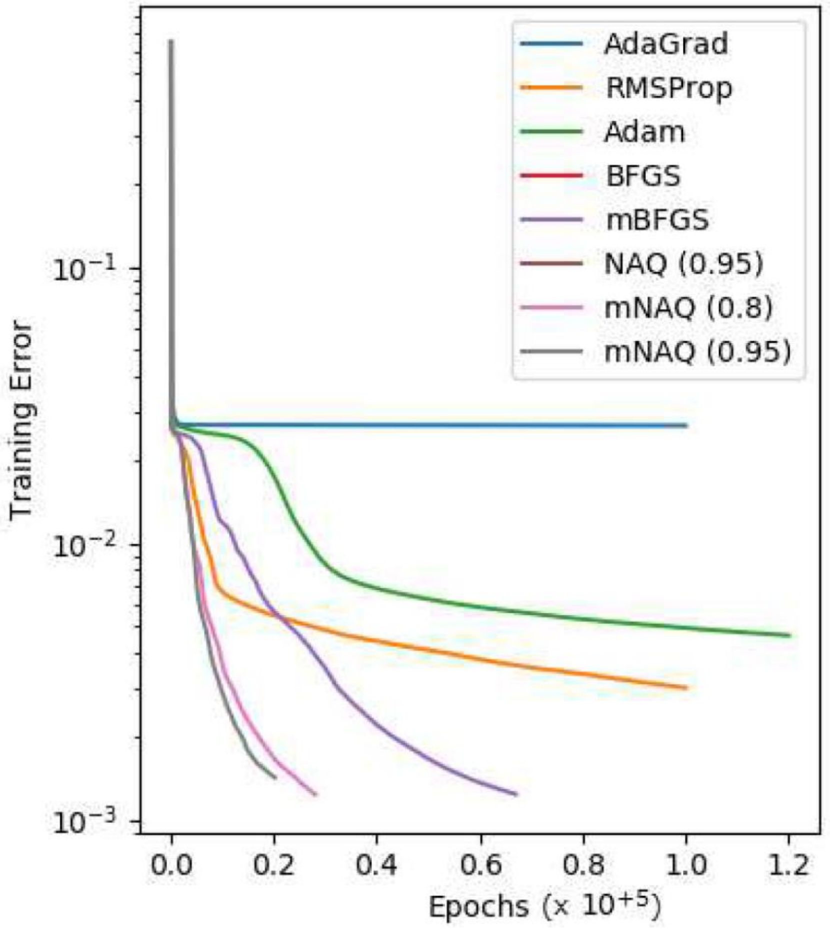
Determine step size α_k using the explicit formula

$$\alpha_k = -\frac{\delta \nabla E(w_k + \mu v_k)^T \hat{g}_k}{\|\hat{g}_k\|^2_{Q_k}} \quad \ldots (Eq.\ 18)$$

Linesearch -> more number of function evaluations -> increased computation time

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "Implementation of a modified Nesterov's Accelerated quasi-Newton method on Tensorflow" In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE (2018) 1147–1154

Function Modeling Example



$$f(a,x,b) = 1 + (x + 2x^2)\sin(-ax^2 + b)$$

- Input nodes = 1
- Hidden neurons = 7
- Output nodes = 1
- Parameters = 22
- Training data : 400
- Test data: 10000

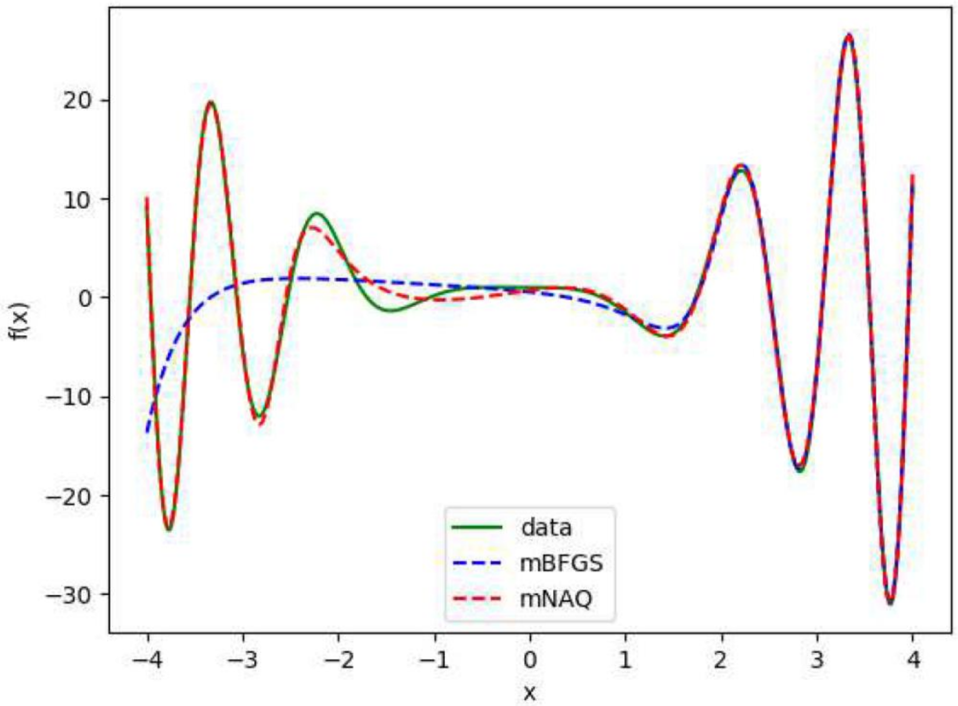
Indrapriyadarsini S., et. al. "Implementation of a modified Nesterov's Accelerated quasi-Newton method on Tensorflow" In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE (2018) 1147–1154

Function Modeling Example

$$f(a,x,b) = 1 + (x + 2x^2)\sin(-ax^2 + b)$$

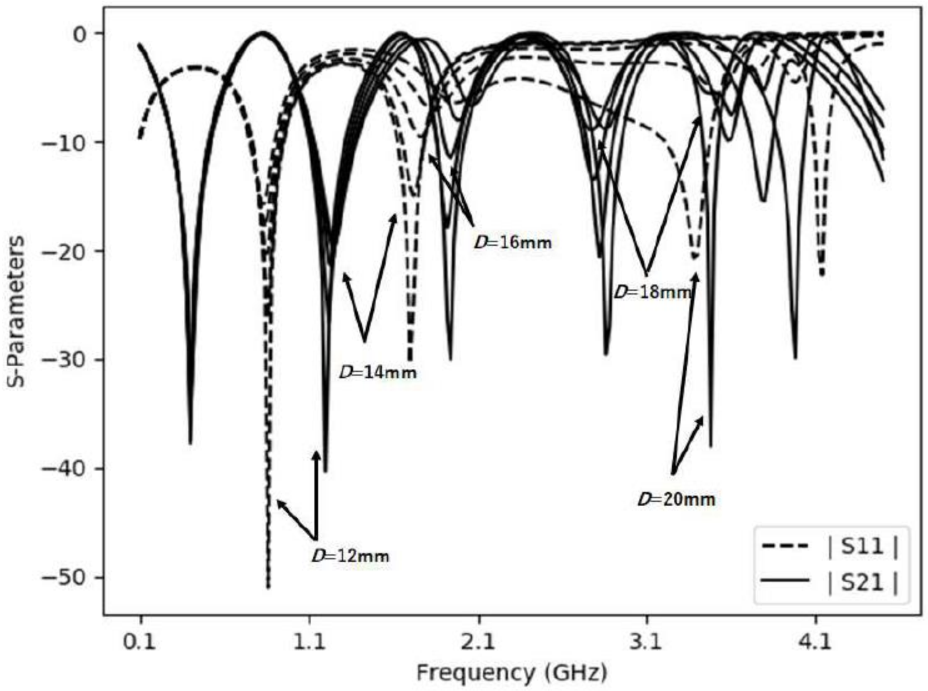
SUMMARY OF SIMULATION RESULTS OF EXAMPLE 1.

Algorithm	μ	$E(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst	Time (s)	Iteration count	$E_{test}(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst
AdaGrad	-	59.8 / 58.6 / 60.2	40	100,000	59.03 / 57.69 / 59.48
RMSprop	-	3.34 / 0.564 / 7.89	41	100,000	3.35 / 0.409 / 8.16
Adam	-	4.15 / 0.324 / 14.3	42	100,000	4.14 / 0.359 / 14.53
BFGS	-	15.14 / 0.650 / 31.80	4.9	3,204	15.14 / 0.650 / 30.66
mBFGS	-	5.24 / 0.194 / 17.8	58	31,370	5.26 / 0.233 / 17.80
mNAQ	0.8	1.94 / 0.307 / 6.33	23	9,006	1.94 / 0.307 / 6.33
	0.85	0.974 / 0.307 / 5.00	19	7,549	0.980 / 0.315 / 5.00
	0.9	1.53 / 0.194 / 13.8	15	5,931	1.53 / 0.194 / 13.80
	0.95	1.30 / 0.195 / 6.31	11	4,461	1.30 / 0.233 / 6.31

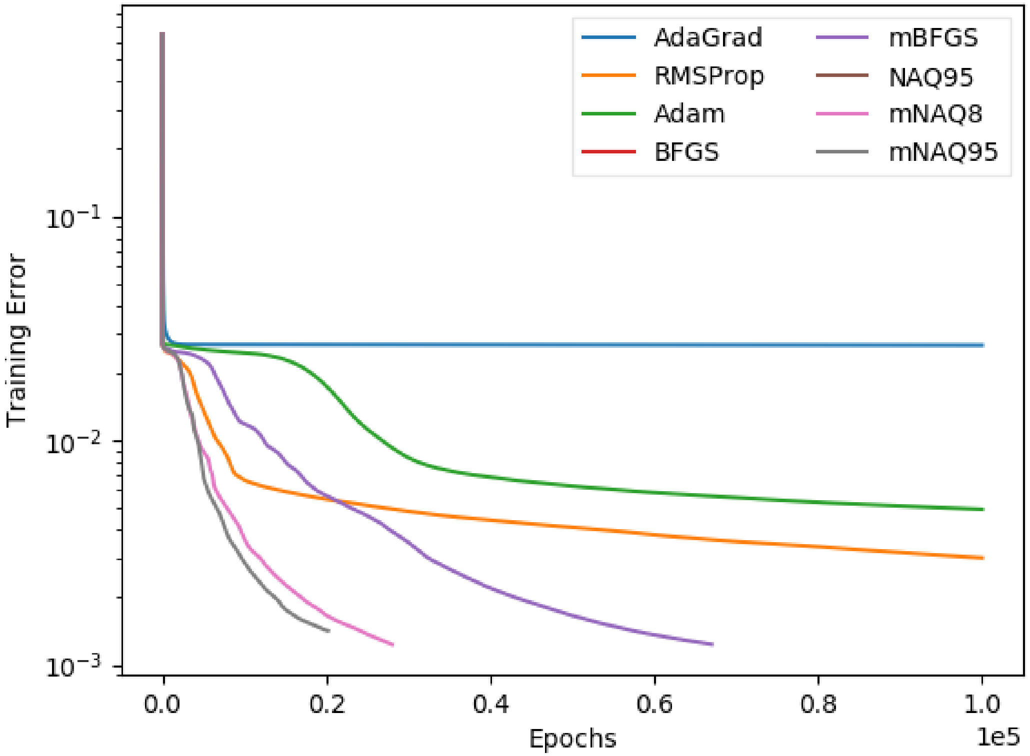


MICROSTRIP LOW PASS FILTER MODELING PROBLEM

Inputs : D=12,14,16,18,20mm
Input frequency f = 0.1 - 4.5GHz
Outputs: S parameters $|s_{11}|$ and $|s_{21}|$



Average training error vs epoch over 15 trials



- Input nodes = 2
- Hidden neurons = 45
- Output nodes = 2
- Parameters = 227
- Training data : 1105
- Test data: 884

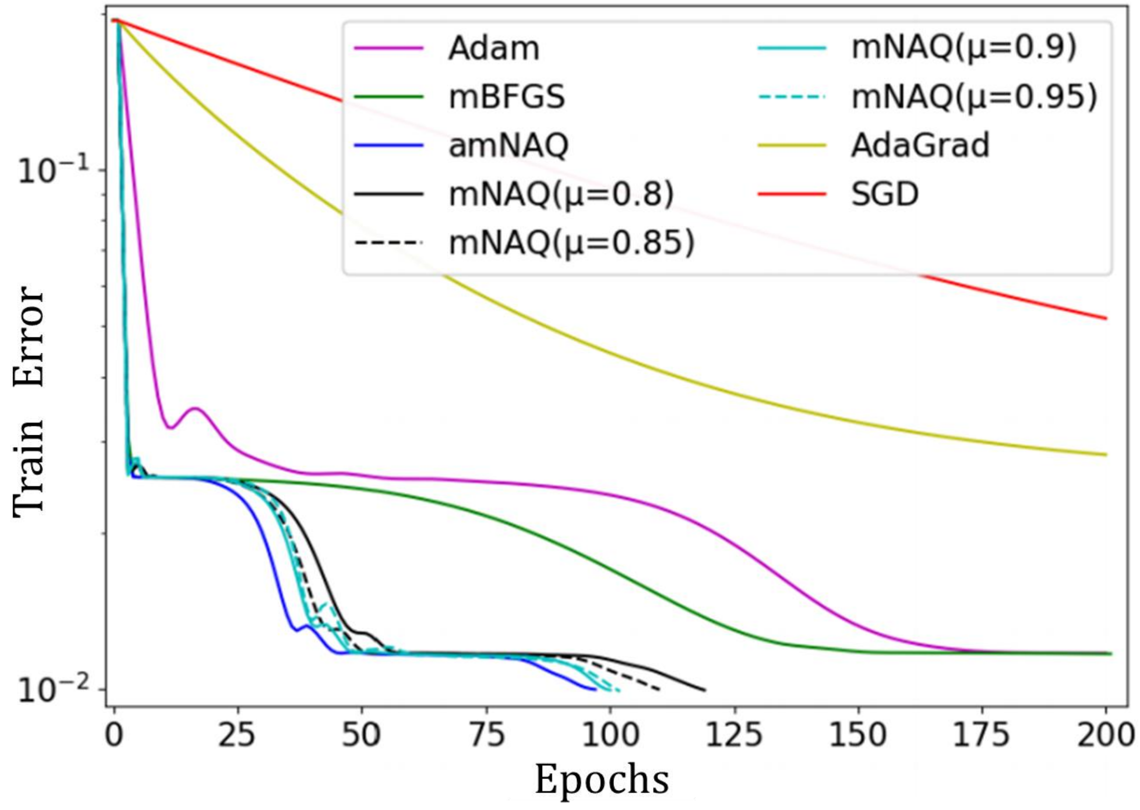
Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "Implementation of a modified Nesterov's Accelerated quasi-Newton method on Tensorflow" In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE (2018) 1147–1154

MICROSTRIP LOW PASS FILTER MODELING PROBLEM

Algorithm	μ	$E(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst	Time (s)	Iteration count	$E_{test}(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst
AdaGrad	-	26.6 / 26.4 / 26.7	112	100,000	22.4 / 22.3 / 22.5
RMSprop	-	2.99 / 2.44 / 4.07	113	100,000	7.00 / 1.88 / 36.0
Adam	-	4.63 / 3.67 / 5.60	137	100,000	37.0 / 3.41 / 212.5
mBFGS	-	1.04 / 0.834 / 1.46	493	81,457	1.01 / 0.529 / 3.52
mNAQ	0.8	0.93 / 0.827 / 1.37	303	38,470	0.744 / 0.534 / 1.07
	0.85	1.02 / 0.756 / 1.62	314	39,678	7.32 / 5.75 / 87.8
	0.9	1.00 / 0.716 / 1.46	242	30,619	0.842 / 0.558 / 1.87
	0.95	1.24 / 0.834 / 1.85	209	26,547	2.08 / 0.600 / 13.7

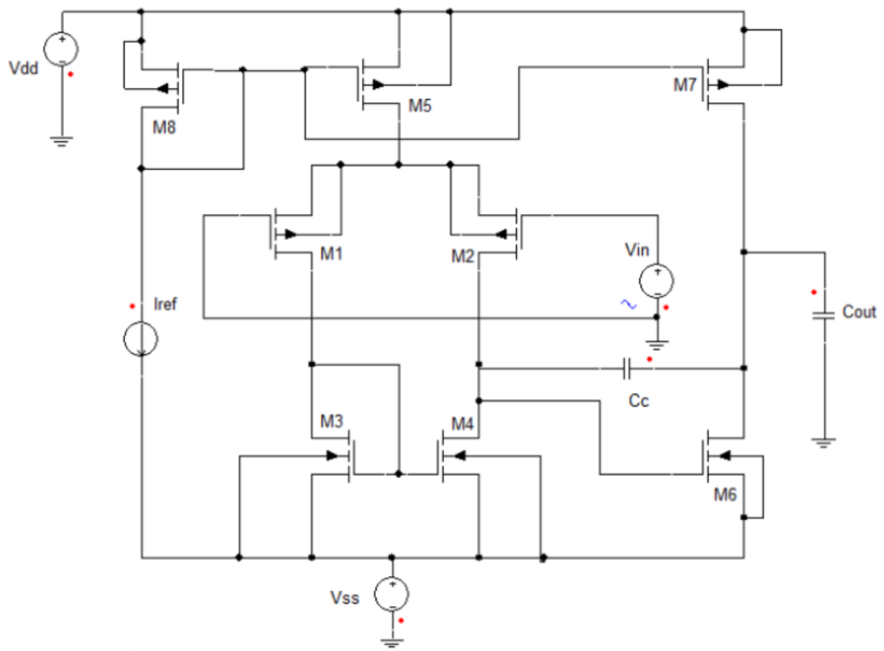
**Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "Implementation of a modified Nesterov's Accelerated quasi-Newton method on Tensorflow" In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE (2018) 1147–1154*

CIRCUIT DESIGN OPTIMIZATION



Design Specification

Parameter	Value
Supply Voltage	$\pm 2.5V$
$\mu_n C_{ox}$	$160\mu A/V^2$
$\mu_p C_{ox}$	$40\mu A/V^2$
Unity GBW	$>$ 1 MHz
Open Loop Gain A_o(dB)	$>$50 dB
Phase Margin	$>$60 deg



Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio and Hideki Asai. "A Neural Network Approach to Analog Circuit Design Optimization using Nesterov's Accelerated Quasi-Newton Method." 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020

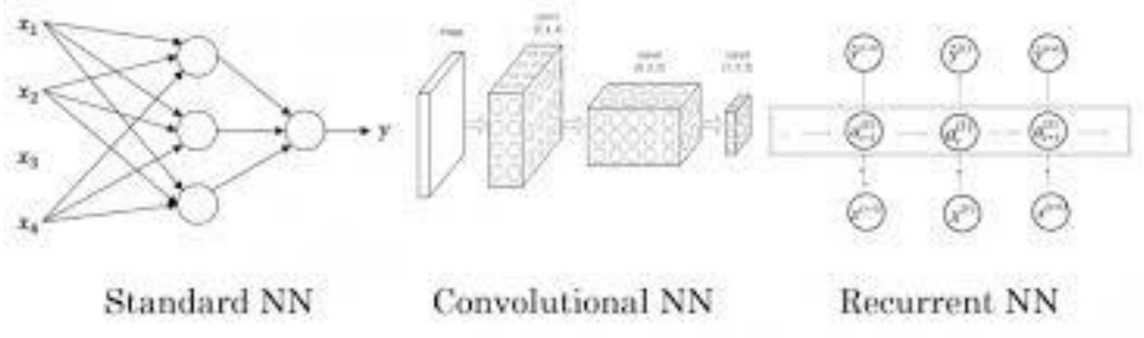
CIRCUIT DESIGN OPTIMIZATION

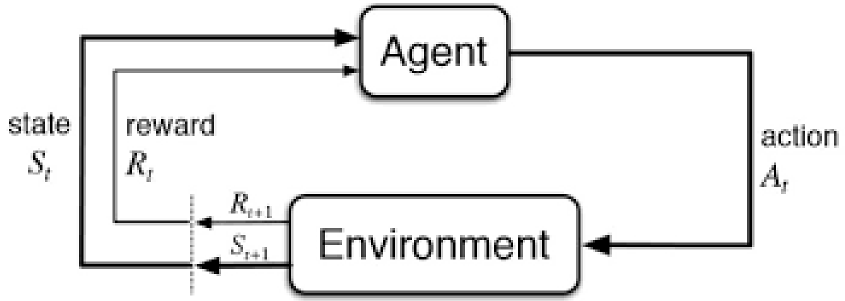
SUMMARY OF THE RESULTS OVER 30 TRIALS

Algorithm	μ_k	$E_{train}(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst	CR (%)	Average epochs	$E_{test}(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst
SGD	-	66.402 / 43.153 / 113.334	-	200	68.428 / 45.852 / 118.620
AdaGrad	-	35.102 / 26.927 / 53.736	-	200	36.784 / 29.450 / 57.535
Adam	-	11.777 / 11.288 / 16.394	-	200	13.576 / 13.103 / 17.860
BFGS	-	11.354 / 11.287 / 11.464	-	200	13.193 / 13.142 / 13.261
mNAQ	0.8	10.010 / 9.892 / 11.194	90	161	11.862 / 11.610 / 13.008
	0.85	10.005 / 9.889 / 11.097	93.3	156	11.859 / 11.616 / 12.907
	0.9	9.966 / 9.880 / 10.478	93.3	156	11.813 / 11.603 / 12.416
	0.95	10.305 / 9.874 / 11.328	63.3	178	12.098 / 11.477 / 13.154
amNAQ	-	9.997 / 9.849 / 11.285	96.7	146	11.799 / 11.546 / 13.105

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio and Hideki Asai. "A Neural Network Approach to Analog Circuit Design Optimization using Nesterov's Accelerated Quasi-Newton Method." 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020

Optimization in Large Scale Problems





Require

- ➢ Fast training
- ➢ Good accuracy
- ➢ Reduce computation cost

STOCHASTIC NESTEROV'S ACCELERATED QUASI-NEWTON

- The update vector of stochastic quasi-Newton (QN) method

$$v_{k+1} = -\alpha_k H_k \nabla E(w_k + \mu v_k, X_k)$$

NAQ computes two gradients per iteration (on same mini-batch)

$$p_k = w_{k+1} - (w_k + \mu v_k)$$

$$q_k = \nabla E(w_{k+1}, X_k) - \nabla E(w_k + \mu v_k, X_k) + \lambda p_k$$

$$\hat{H}_{k+1} = (I - \rho_k p_k {q_k}^T)\hat{H}_k(I - \rho_k q_k {p_k}^T) + \rho_k p_k {p_k}^T$$

Reduced sampling noise

Same computational cost as o(L)BFGS + faster convergence

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "A Stochastic Quasi-Newton Method with Nesterov's Accelerated Gradient", Joint European Conference on Machine Learning and Principles of Knowledge Discovery in Databases, ECML-PKDD, Springer, 2019

STOCHASTIC NESTEROV'S ACCELERATED QUASI-NEWTON

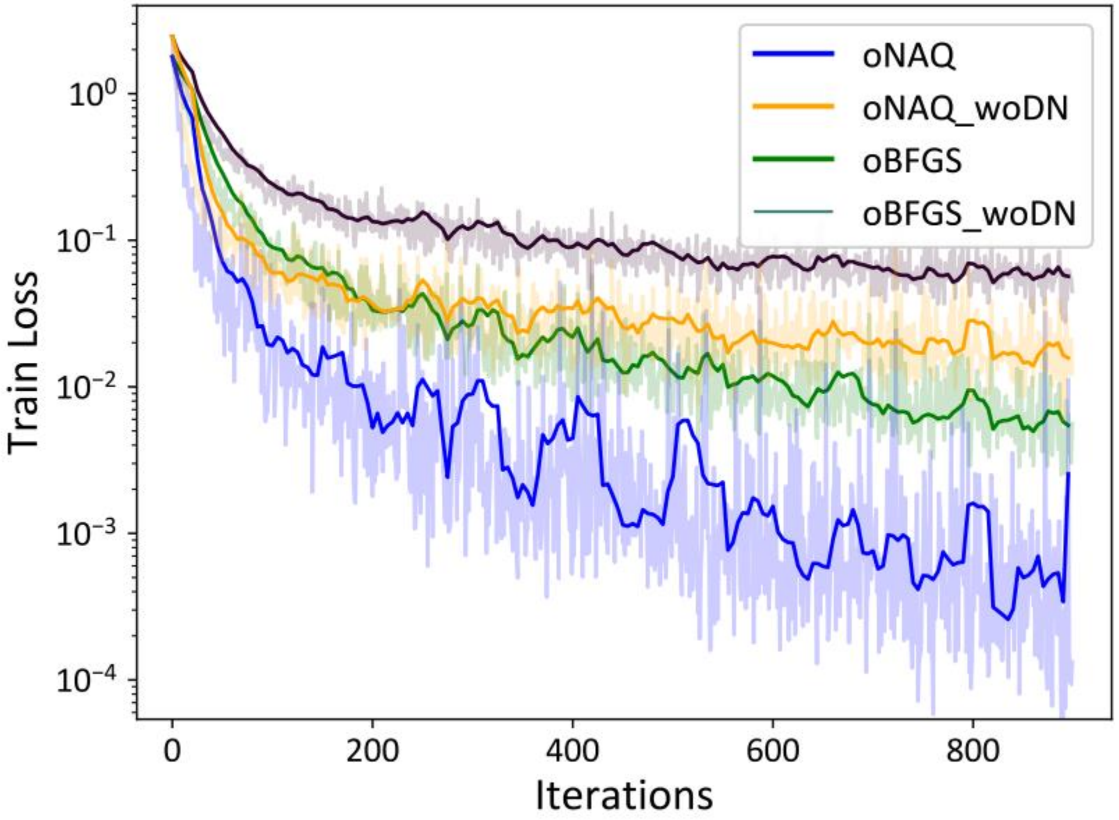
Direction Normalization

Further to improve the stability, direction normalization is introduced.

$$\hat{\boldsymbol{g}}_k \leftarrow -\widehat{\boldsymbol{H}}_k \nabla E(\boldsymbol{w}_k + \mu \boldsymbol{v}_k, \boldsymbol{X}_k)$$

$$\hat{\boldsymbol{g}}_k = \frac{\hat{\boldsymbol{g}}_k}{\|\hat{\boldsymbol{g}}_k\|_2}$$

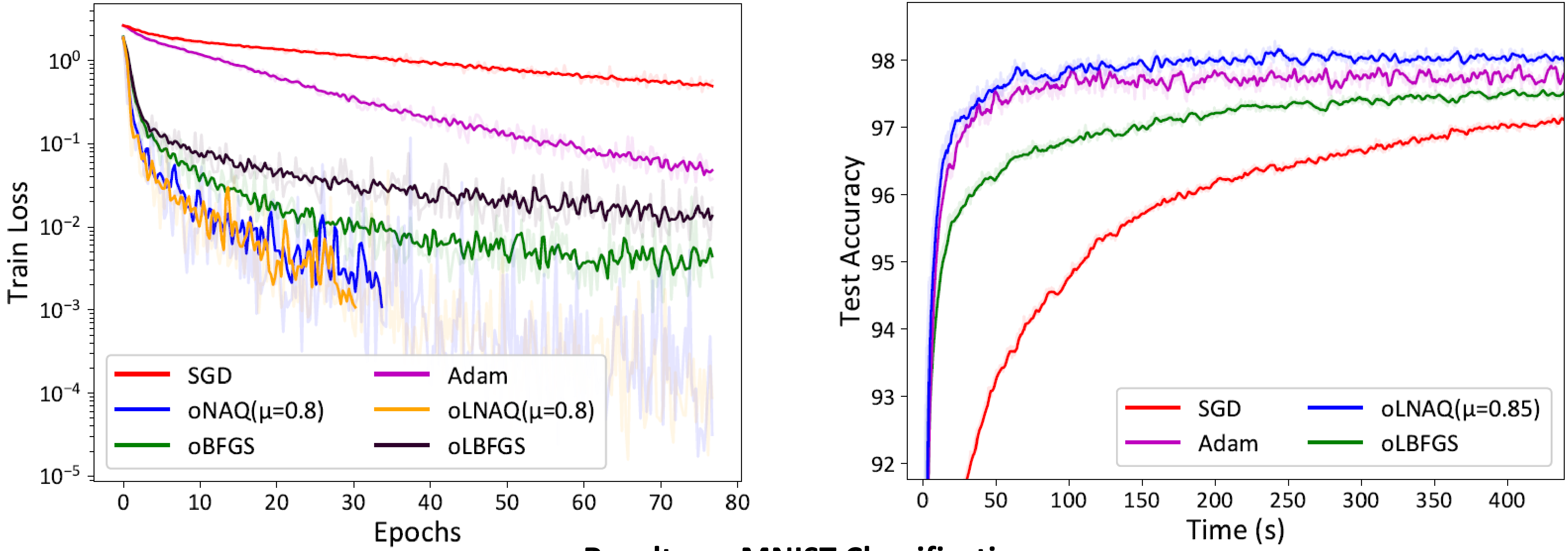
Normalizing the search direction at each iteration ensures that the algorithm does not move too far away from the current objective



Effect of direction normalization

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "A Stochastic Quasi-Newton Method with Nesterov's Accelerated Gradient", Joint European Conference on Machine Learning and Principles of Knowledge Discovery in Databases, ECML-PKDD, Springer, 2019

Stochastic Nesterov's Accelerated quasi-Newton – oNAQ

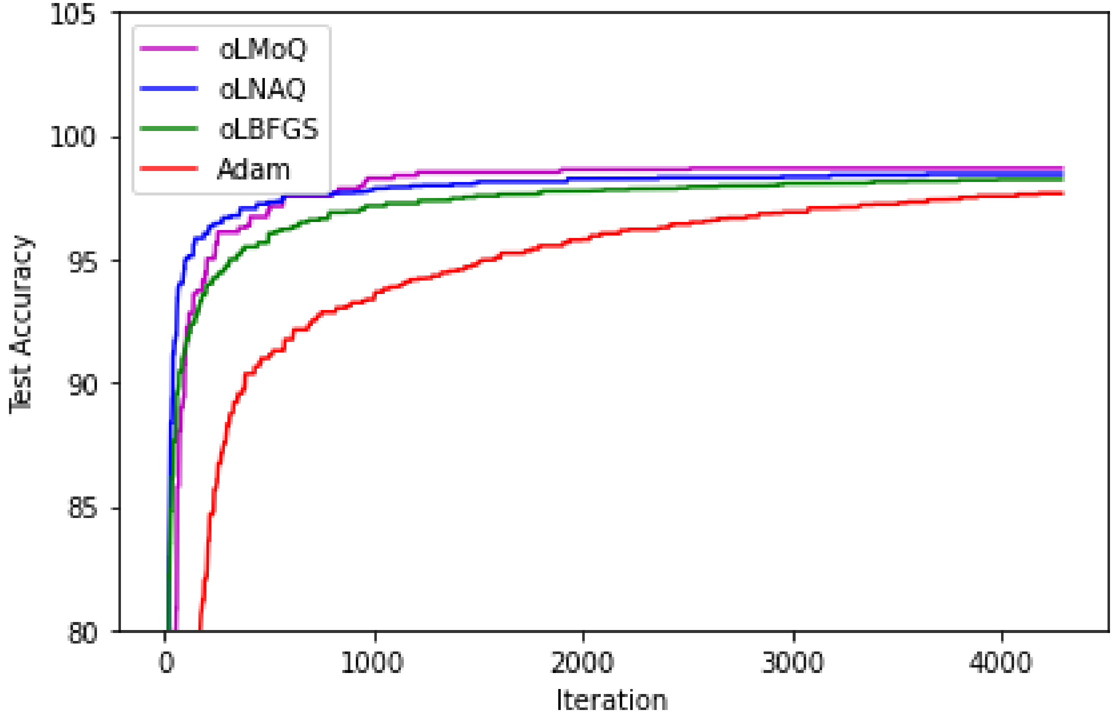


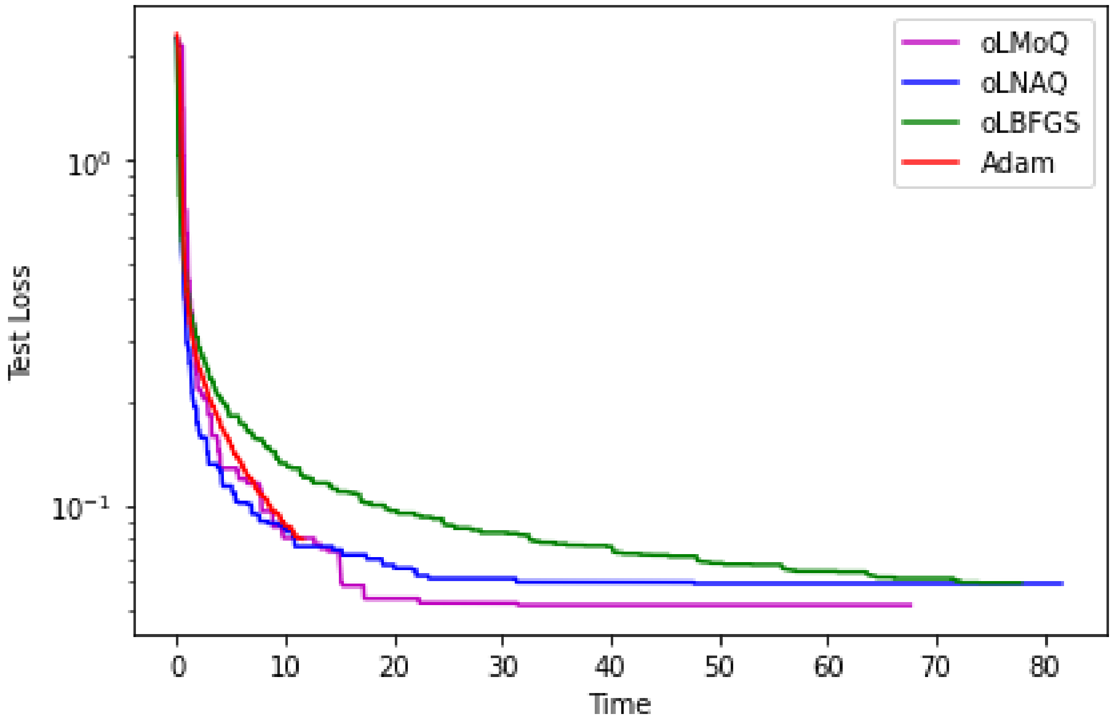
Results on MNIST Classification

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "A Stochastic Quasi-Newton Method with Nesterov's Accelerated Gradient", Joint European Conference on Machine Learning and Principles of Knowledge Discovery in Databases, ECML-PKDD, Springer, 2019

Stochastic Momentum Accelerated quasi-Newton – oMoQ

$$\nabla E(w_k + \mu v_k) \approx (1 + \mu_k)\nabla E(w_k) - \mu_k \nabla E(\mathbf{w_{k-1}})$$





Results on MNIST Classification on LeNet-5

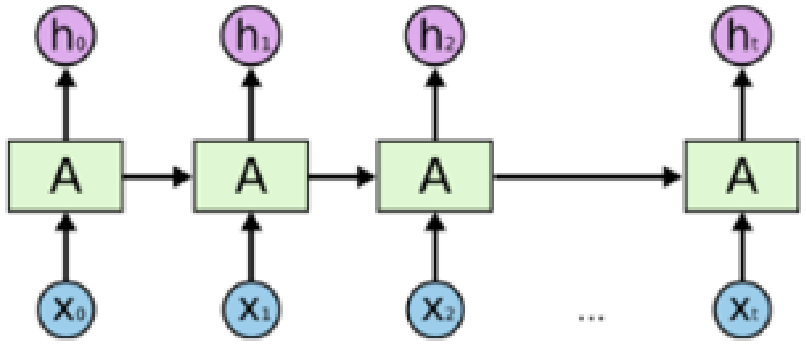
S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "A Stochastic Momentum Accelerated Quasi-Newton Method for Neural Networks (Student Abstract)", Proceedings of the 36th AAAI Conference on Artificial Intelligence, Feb 2022

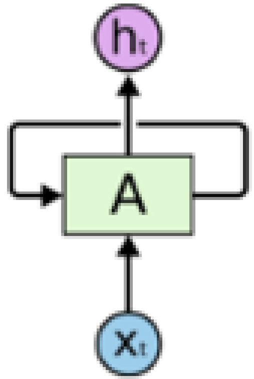
Stochastic Momentum / Nesterov's Accelerated quasi-Newton

Summary of Computational Cost and Storage

	Algorithm	Computational Cost	Storage
full batch	BFGS	$nd + d^2 + \zeta nd$	d^2
	NAQ	$2nd + d^2 + \zeta nd$	d^2
	MoQ	$nd + d^2 + \zeta nd$	$d^2 + d$
	LBFGS	$nd + 4md + 2d + \zeta nd$	$2md$
	LNAQ	$2nd + 4md + 2d + \zeta nd$	$2md$
	LMoQ	$nd + 4md + 2d + \zeta nd$	$(2m+1)d$
online	oBFGS	$2bd + d^2$	d^2
	oNAQ	$2bd + d^2$	d^2
	oMoQ	$bd + d^2$	$d^2 + d$
	oLBFGS	$2bd + 6md$	$2md$
	oLNAQ	$2bd + 6md$	$2md$
	oLMoQ	$bd + 6md$	$(2m+1)d$

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

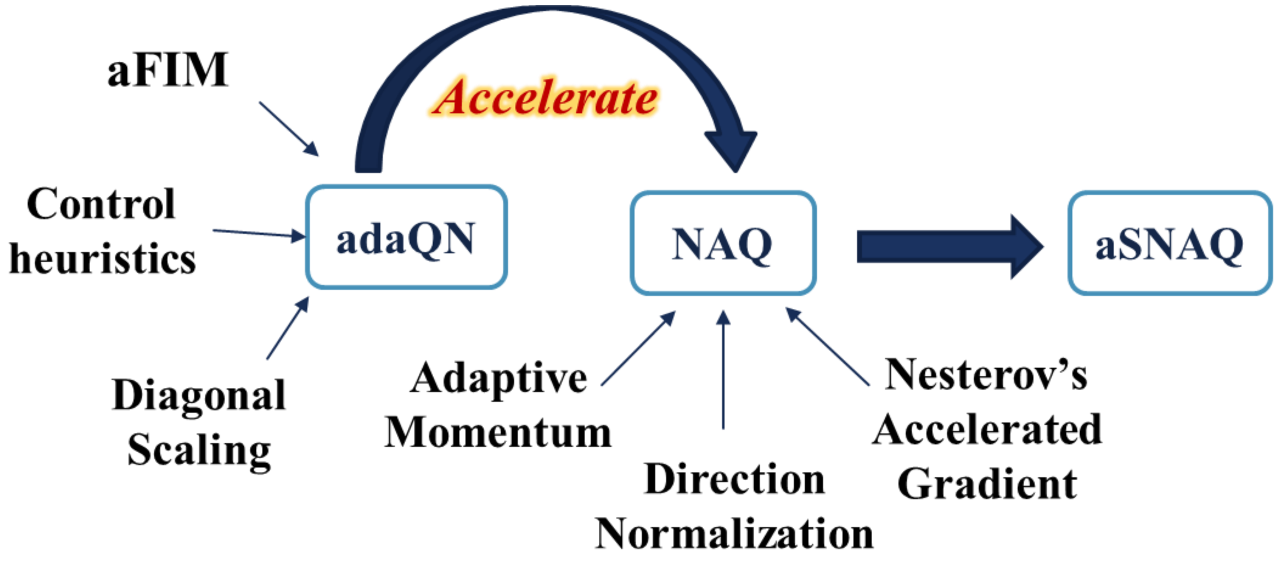




Recurrent Neural Networks

- Backpropagation through time
- Vanishing/exploding gradient
- Difficult training long sequences
- Suitable for dynamic problems

➢ Builds on the algorithmic framework of **SQN** and **adaQN**



Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019 (Best Student Paper Award)

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Nesterov's Accelerated Gradient

Faster convergence by incorporating the Nesterov's accelerated gradient

$$g_k \leftarrow H_k \underline{\nabla E(w_k + \mu v_k)}$$

Nesterov's Accelerated Gradient

Direction Normalization

Direction normalization scales the search direction in each iteration by its l_2 norm

$$g_k = \frac{g_k}{\|g_k\|_2}$$

Initial Hessian scaling

$$[H_k^{(0)}]_{ii} = \frac{1}{\sqrt{\sum_{j=0}^{k} \nabla E(w_j)_i^2} + \varepsilon}$$

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

- **Curvature information matrix**

QN methods generate high-quality steps even with crude curvature information.

Fisher Information matrix (FIM) yields a better estimate of the curvature.

A FIFO memory buffer $\boldsymbol{F}$ of size $\mathrm{m_F}$ accumulates at each iteration the FIM as

$$F_i = \nabla \boldsymbol{E}(\boldsymbol{w}_k) \nabla \boldsymbol{E}(\boldsymbol{w}_k)^T$$

This accumulated FIM is used in the computation of the $\boldsymbol{y}$ vector

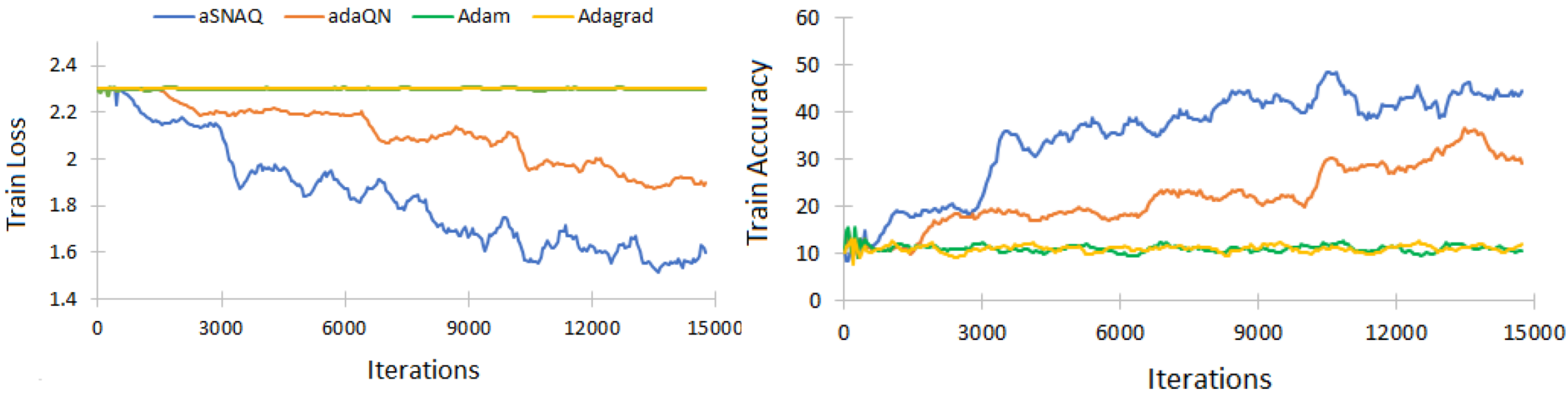
$$\boldsymbol{y} \leftarrow \frac{1}{|F|} \left(\sum_{i=1}^{|F|} F_i \cdot \boldsymbol{s} \right) \quad \text{where} \quad \boldsymbol{s} \leftarrow \boldsymbol{w}_n - \boldsymbol{w}_o$$

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Summary of Computational and Storage Cost.

Algorithm	Computational Cost	Storage
BFGS	$nd + d^2 + \zeta nd$	d^2
NAQ	$2nd + d^2 + \zeta nd$	d^2
adaQN	$bd + (4m_L + m_F + 2)d + (b + 4)d/L$	$(2m_L + m_F)d$
aSNAQ	$2bd + (4m_L + m_F + 3)d + (b + 4)d/L$	$(2m_L + m_F)d$

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

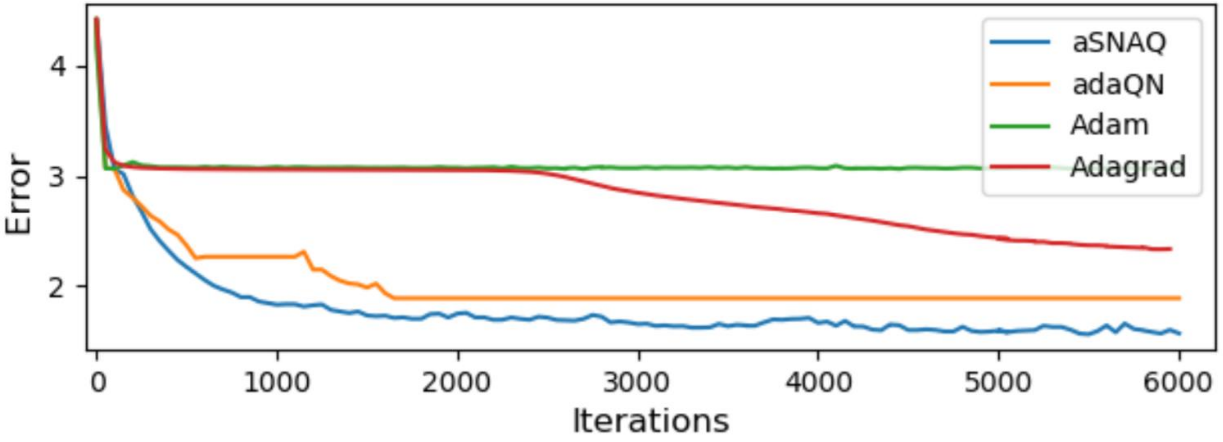


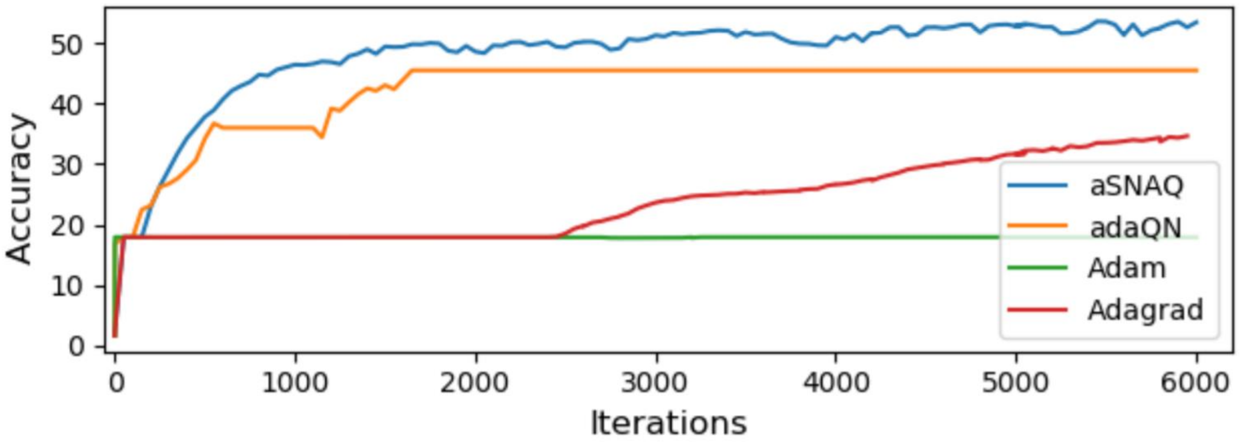
Train loss and train accuracy of MNIST pixel-by-pixel sequence

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019 (Best Student Paper Award)

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Character Level Language modeling (5-layer RNN)

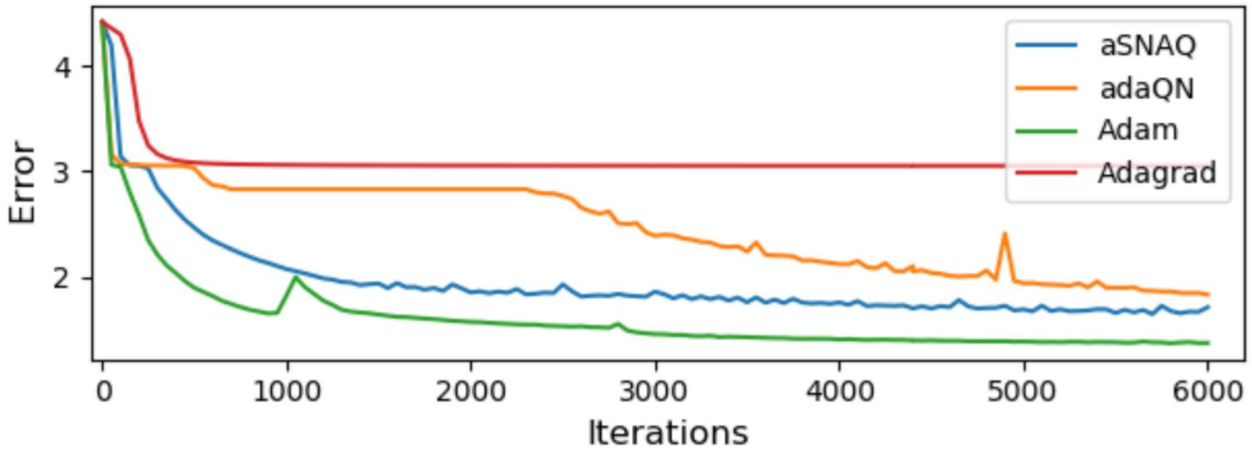


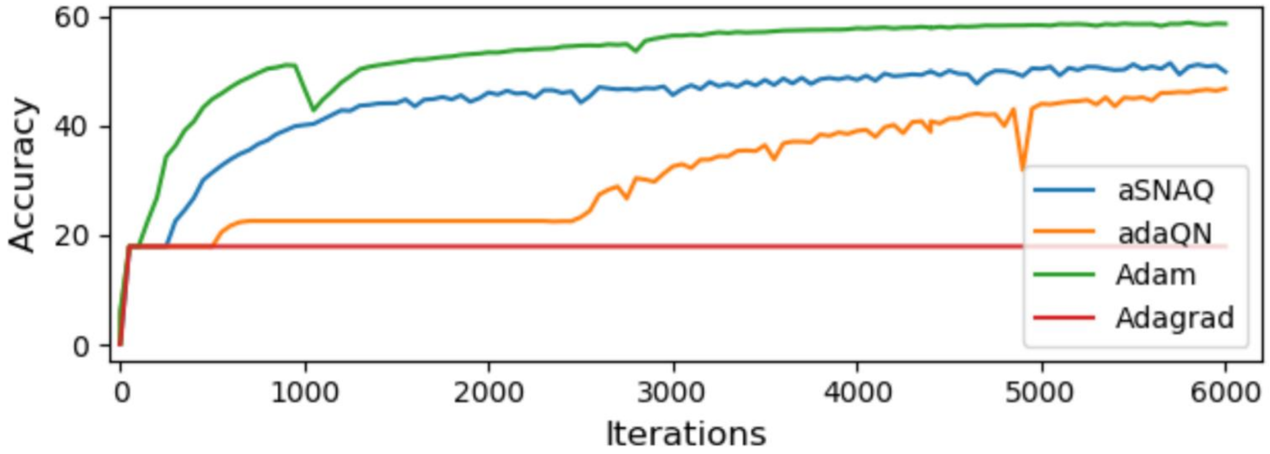


Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019 (Best Student Paper Award) - (Extended paper – NOLTA journal IEICE, Oct 2020)

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Character Level Language modeling (2-layer LSTM)

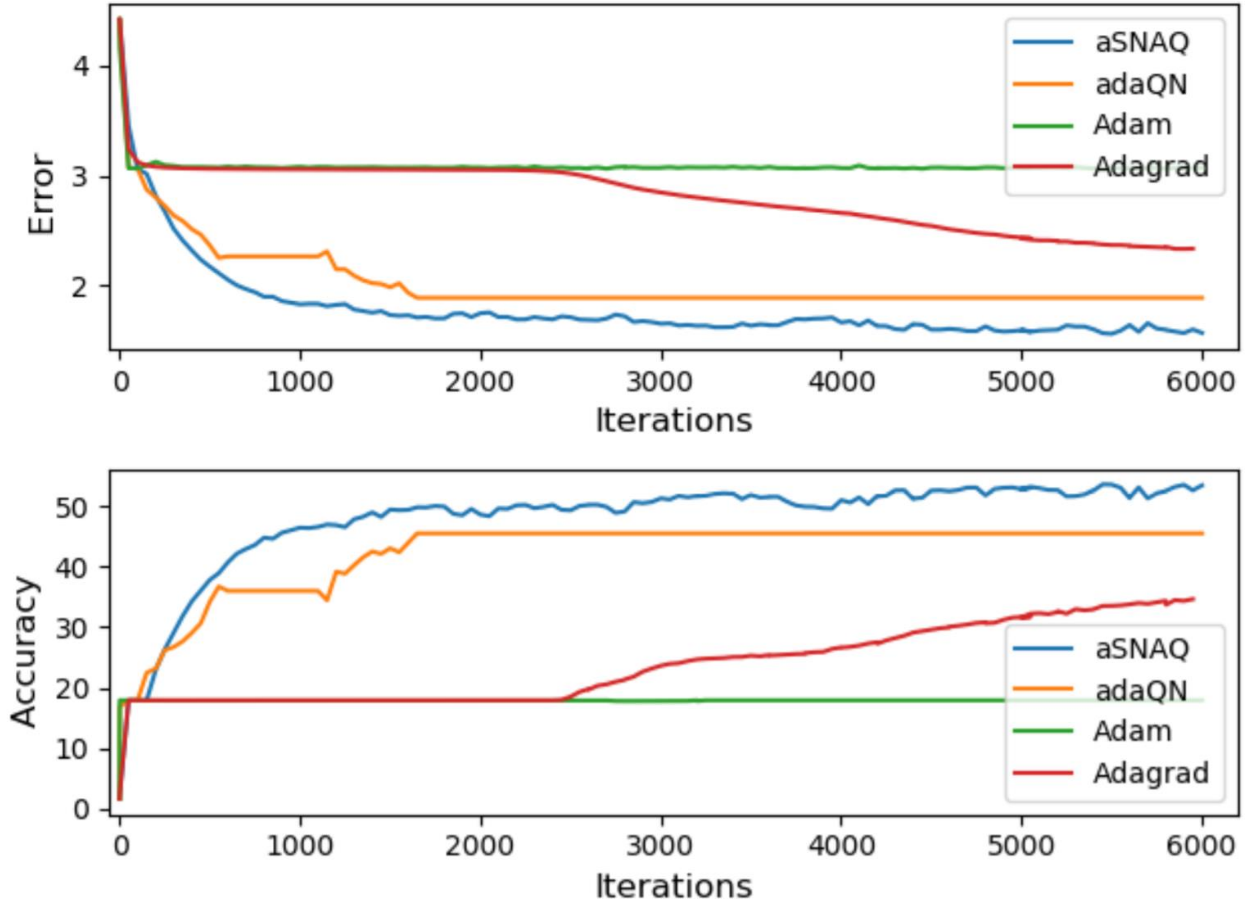




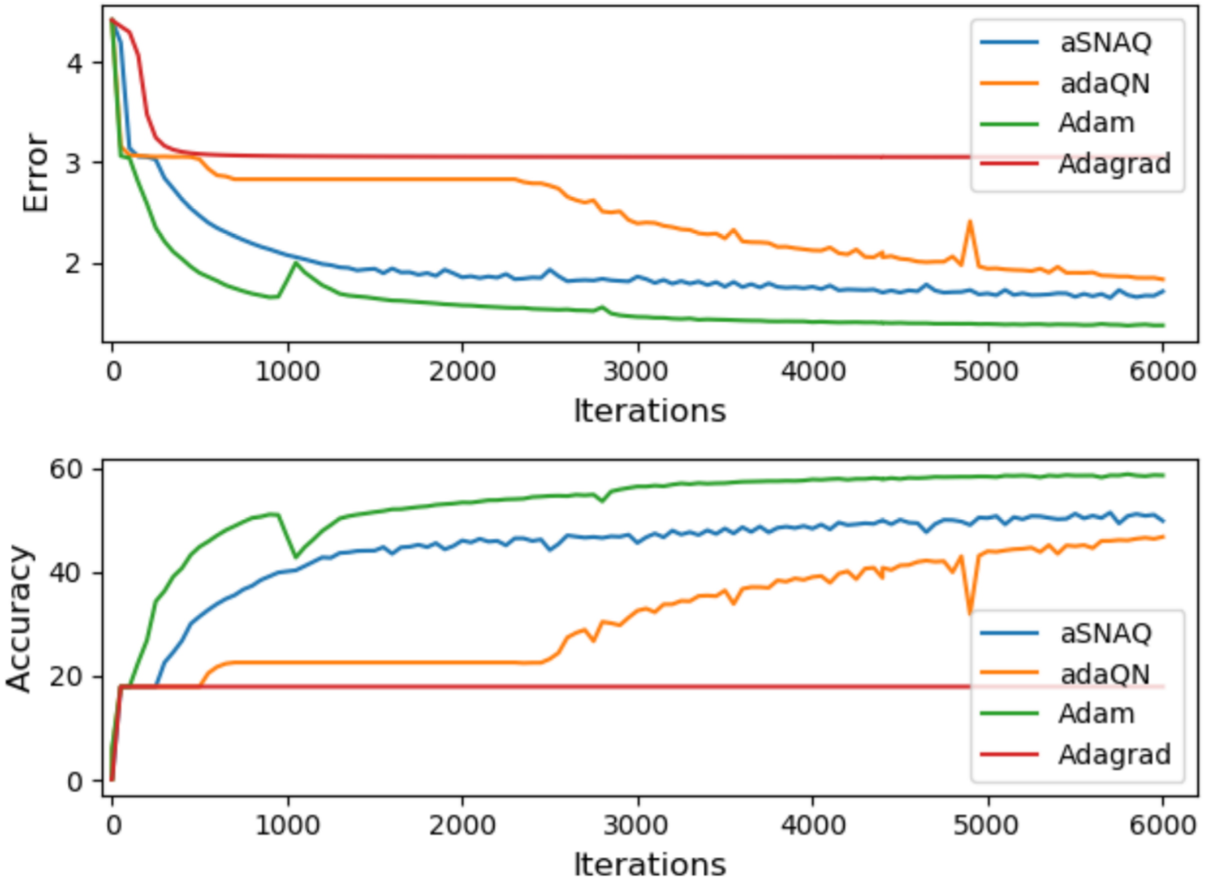
Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019 (Best Student Paper Award) - (Extended paper – NOLTA journal IEICE, Oct 2020)

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Character Level Language modeling (5-layer RNN)



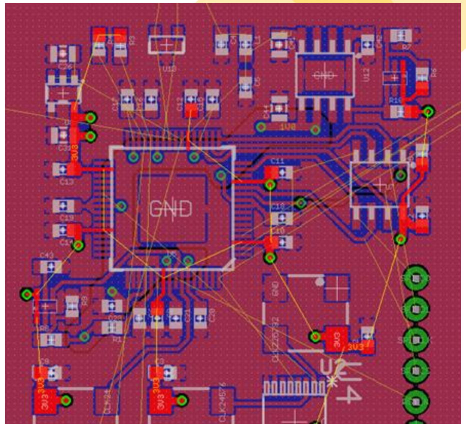
Character Level Language modeling (2-layer LSTM)



Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019 (Best Student Paper Award) - (Extended paper – NOLTA journal IEICE, Oct 2020)

PCB ROUTING USING REINFORCEMENT LEARNING

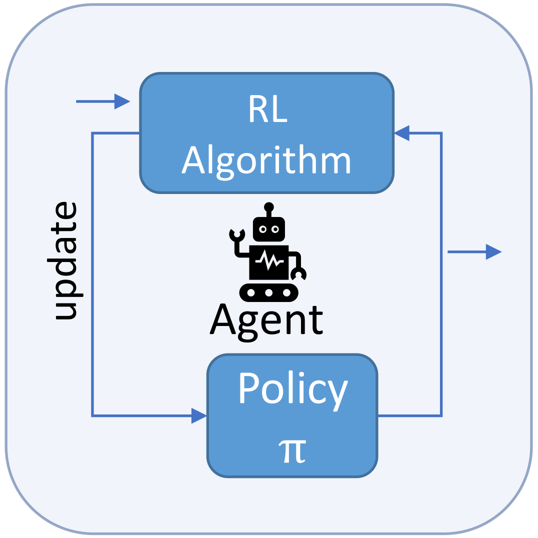
Synthesis and physical design optimizations are the core tasks of the VLSI / ASIC design flow. ***Global routing*** has been a challenging problem in IC physical design.



Objective

Given a netlist with the description of all the components, their connections and position, the goal of the global router is to determine the path of all the connections without violating the constraints and design rules.

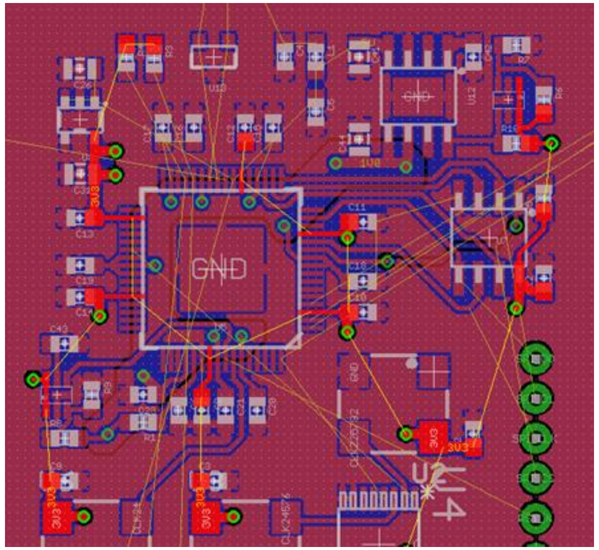
- Route all pins and nets
- Minimize total wirelength (WL)
- Minimize total overflows

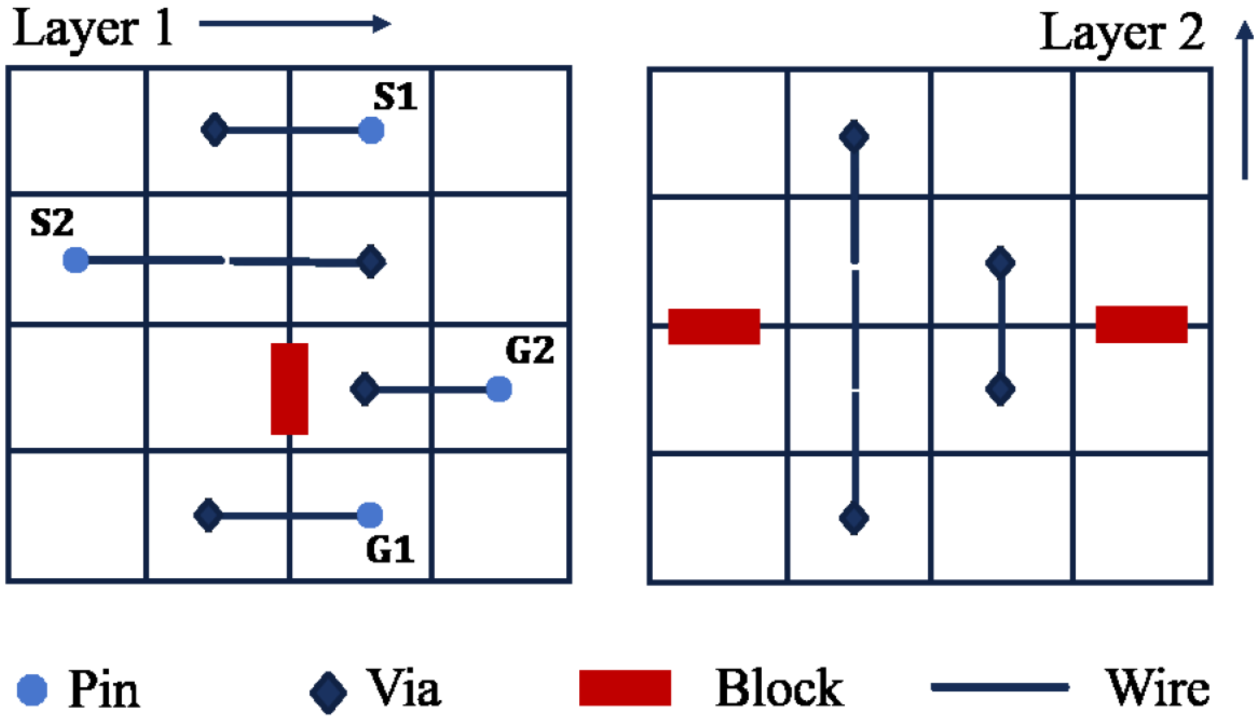


Conventional routing automation tools are usually based on analytical and path search algorithms which are **NP complete**.

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "A Nesterov's Accelerated quasi-Newton method for Global Routing using Deep Reinforcement Learning", International Symposium on Nonlinear Theory and its Applications, NOLTA, IEICE, 2020 (Student Paper Award)

PCB ROUTING USING REINFORCEMENT LEARNING





Objective function:

$$L(w) = E_{(s,a)\sim\zeta}[(Y - Q_w(s,a))^2]$$

where

$$Y = E_{s'\sim\zeta}[R + \gamma Q_{w^-}(s', argmax_{a'} Q_w(s', a'))]$$

Key Takeaways

- In RL the training set is dynamically populated
- DQNs use mean-squared Bellman (non-convex function)
- Second order methods – aSNAQ show better convergence

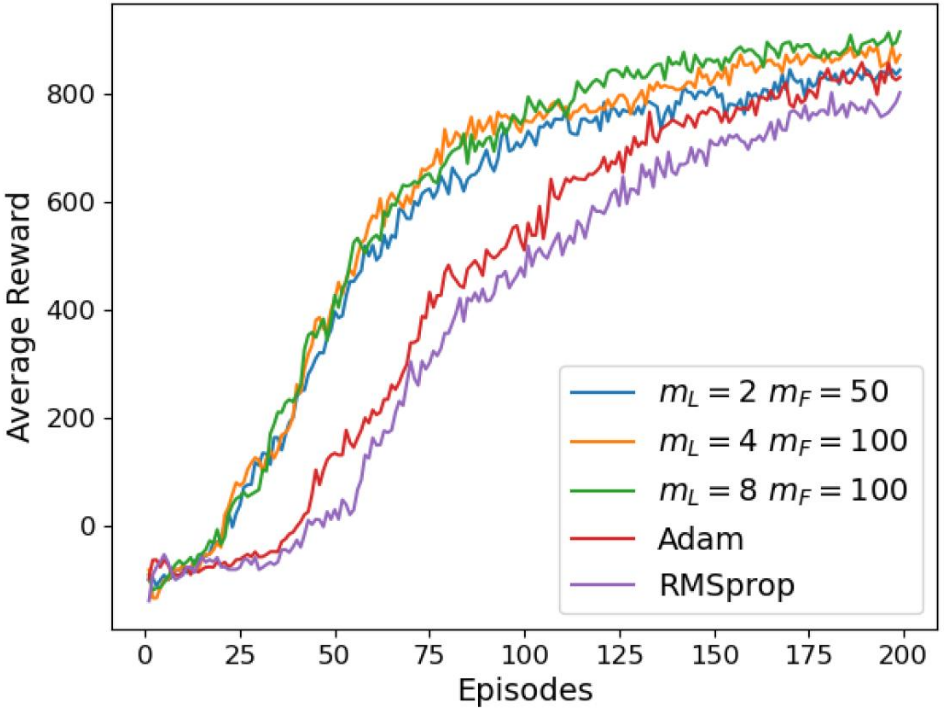


Fig. 4. Average reward over 25 benchmarks with 10 two-pin nets.

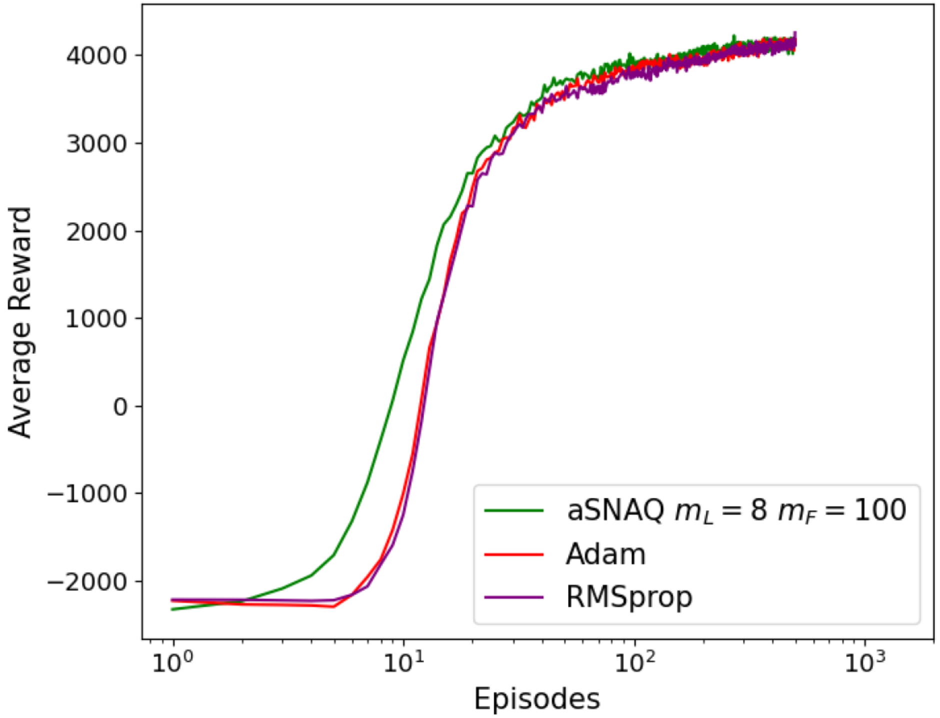


Fig. 5. Average reward over 30 benchmarks with 50 two-pin nets.

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "A Nesterov's Accelerated quasi-Newton method for Global Routing using Deep Reinforcement Learning", International Symposium on Nonlinear Theory and its Applications, NOLTA, IEICE, 2020 (Student Paper Award) - (Extended paper – NOLTA journal IEICE, Jul 2021)

Total 50 Netlists
Max episode $\mathcal{E}$ = 500

— indicates could not be routed within 500 episodes
diff is wirelength reduction compared to A*

Trial	A*			Adam					RMSprop					aSNAQ		
Num	WL	WL	diff	$\mathcal{R}_{best}$	$\mathcal{E}$	Pins	WL	diff	$\mathcal{R}_{best}$	$\mathcal{E}$	Pins	WL	diff	$\mathcal{R}_{best}$	$\mathcal{E}$	Pins
1	390	-	-	4386	465	48	-	-	4363	490	48	**368**	-22	4667	231	50
2	386	-	-	4505	399	49	-	-	4513	483	49	**376**	-10	4610	148	50
3	379	-	-	4234	478	47	-	-	4533	401	49	-	-	4382	344	48
4	369	348	-21	4690	288	50	350	-19	4685	492	50	**345**	-24	4699	75	50
5	366	362	-4	4679	422	50	**361**	-5	4681	430	50	369	+3	4656	458	50
6	352	348	-4	4691	437	50	344	-8	4697	296	50	**335**	-17	4701	157	50
7	430	-	-	4053	485	46	-	-	4322	393	48	-	-	4324	285	48
8	398	-	-	4522	205	49	-	-	4513	455	49	**377**	-21	4663	361	50
9	369	369	0	4669	497	50	**347**	-22	4687	252	50	348	-21	4693	189	50
10	366	359	-7	4674	112	50	375	+9	4660	327	50	**357**	-9	4683	480	50
11	379	380	+1	4660	252	50	380	+1	4658	429	50	-	-	4523	428	49
12	351	**346**	-5	4692	293	50	351	0	4689	340	50	348	-3	4692	93	50
13	395	411	+16	4616	456	50	397	+2	4645	422	50	**394**	-1	4640	193	50
14	340	343	+3	4700	409	50	**338**	-2	4706	381	50	341	+1	4699	49	50
15	375	374	-1	4659	319	50	384	9	4660	313	50	**371**	-4	4668	490	50

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "A Nesterov's Accelerated quasi-Newton method for Global Routing using Deep Reinforcement Learning", International Symposium on Nonlinear Theory and its Applications, NOLTA, IEICE, 2020 (Student Paper Award) - (Extended paper – NOLTA journal IEICE, Jul 2021)

OTHER QUASI-NEWTON METHOD + NESTEROV'S ACCELERATION ?

Method	$B_{k+1} =$	$H_{k+1} = B_{k+1}^{-1} =$
BFGS	$B_k + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k} - \frac{B_k \Delta x_k (B_k \Delta x_k)^{\mathrm{T}}}{\Delta x_k^{\mathrm{T}} B_k \Delta x_k}$	$\left(I - \frac{\Delta x_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}\right) H_k \left(I - \frac{y_k \Delta x_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}\right) + \frac{\Delta x_k \Delta x_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}$
Broyden	$B_k + \frac{y_k - B_k \Delta x_k}{\Delta x_k^{\mathrm{T}} \Delta x_k} \Delta x_k^{\mathrm{T}}$	$H_k + \frac{(\Delta x_k - H_k y_k) \Delta x_k^{\mathrm{T}} H_k}{\Delta x_k^{\mathrm{T}} H_k y_k}$
Broyden family	$(1 - \varphi_k) B_{k+1}^{\mathrm{BFGS}} + \varphi_k B_{k+1}^{\mathrm{DFP}}, \quad \varphi \in [0, 1]$	
DFP	$\left(I - \frac{y_k \Delta x_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}\right) B_k \left(I - \frac{\Delta x_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}\right) + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} \Delta x_k}$	$H_k + \frac{\Delta x_k \Delta x_k^{\mathrm{T}}}{\Delta x_k^{\mathrm{T}} y_k} - \frac{H_k y_k y_k^{\mathrm{T}} H_k}{y_k^{\mathrm{T}} H_k y_k}$
SR1	$B_k + \frac{(y_k - B_k \Delta x_k)(y_k - B_k \Delta x_k)^{\mathrm{T}}}{(y_k - B_k \Delta x_k)^{\mathrm{T}} \Delta x_k}$	$H_k + \frac{(\Delta x_k - H_k y_k)(\Delta x_k - H_k y_k)^{\mathrm{T}}}{(\Delta x_k - H_k y_k)^{\mathrm{T}} y_k}$

Wikipedia

ACCELERATING SR1 WITH NESTEROV'S GRADIENT

- Quasi-Newton + Nesterov's acceleration **satisfies secant condition**
- The Hessian is updated using the Symmetric rank-1 update formula given as

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\mathbf{s}_k - \mathbf{H}_k\mathbf{y}_k)(\mathbf{s}_k - \mathbf{H}_k\mathbf{y}_k)^T}{(\mathbf{s}_k - \mathbf{H}_k\mathbf{y}_k)^T\mathbf{y}_k},$$

where,

$$\mathbf{y}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k + \mu_k\mathbf{v}_k) \text{ and } \mathbf{s}_k = \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu_k\mathbf{v}_k)$$

- Ensure positive semi-definiteness by performing the update only if

$$|\mathbf{s}_k^T(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)| \geq \rho\, ||\mathbf{s}_k||\, ||\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k||$$

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

ACCELERATING SR1 WITH NESTEROV'S GRADIENT : CONVERGENCE

Assumption 1: The sequence of iterates $\mathbf{w}_k$ and $\hat{\mathbf{w}}_k$ remains in the closed and bounded set $\mathbf{\Omega}$ on which the objective function is twice continuously differentiable and has Lipschitz continuous gradient, i.e. there exists a constant $L > 0$ such that

$$||\nabla E(\mathbf{w}_{k+1}) - \nabla E(\hat{\mathbf{w}}_k)|| \leq L||\mathbf{w}_{k+1} - \hat{\mathbf{w}}_k|| \quad \forall\ \mathbf{w}_{k+1},\ \hat{\mathbf{w}}_k \in \mathbb{R}^d$$

If *Assumption 1* holds true, then it implies that the objective function satisfies,

$$E(\boldsymbol{w}_{k+1}) \leq E(\boldsymbol{w}_k + \mu\boldsymbol{v}_k) + \nabla E(\boldsymbol{w}_k + \mu\boldsymbol{v}_k)^T\boldsymbol{d} + \frac{L}{2}\|\boldsymbol{w}_{k+1} - (\boldsymbol{w}_k + \mu\boldsymbol{v}_k)\|_2^2$$

Assumption 2: The Hessian matrix is bounded and well-defined, .i.e, there exists constants ρ and M, such that

$$\rho \leq ||\mathbf{B}_k|| \leq M \quad \forall\ k$$

and for each iteration

$$|\mathbf{s}_k^T(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)| \geq \rho\ ||\mathbf{s}_k||\ ||\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k||$$

Assumption 2 ensures Hessian matrix is symmetric positive semidefinite and bounded

ACCELERATING SR1 WITH NESTEROV'S GRADIENT : CONVERGENCE

Assumption 3: Let $\mathbf{B}_k$ be any $n \times n$ symmetric matrix and $\mathbf{s}_k$ be an optimal solution to the trust region subproblem,

$$\min_{\mathbf{d}} \; m_k(\mathbf{d}) = E(\hat{\mathbf{w}}_k) + \mathbf{d}^T \nabla E(\hat{\mathbf{w}}_k) + \frac{1}{2}\mathbf{d}^T \mathbf{B}_k \mathbf{d},$$

where $\hat{\mathbf{w}}_k + \mathbf{d}$ lies in the trust region. Then for all $k \geq 0$,

$$\left|\nabla E(\hat{\mathbf{w}}_k)^T \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k\right| \geq \frac{1}{2}||\nabla E(\hat{\mathbf{w}}_k)|| \min\left\{\Delta_k, \frac{||\nabla E(\hat{\mathbf{w}}_k)||}{||\mathbf{B}_k||}\right\}$$

Assumption 3 ensures that the subproblem solved by the trust region method is sufficiently optimal at each iteration.

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

ACCELERATING SR1 WITH NESTEROV'S GRADIENT : CONVERGENCE

Lemma : If *Assumptions 1 to 3* holds true, and $\boldsymbol{s}_k$ be an optimal solution to the trust region subproblem, and if the initial Hessian $\boldsymbol{H}_{k+1} = \gamma_k$ is bounded (i.e., $\mathbf{0} \leq \gamma_k \leq \hat{\gamma}_k$) then for all $k \geq 0$, the Hessian update given by the SR1+N algorithm is bounded

$$||\mathbf{B}_{k+1}|| \leq \left(1 + \frac{1}{\rho}\right)^{m_L} \gamma_k + \left[\left(1 + \frac{1}{\rho}\right)^{m_L} - 1\right] M$$

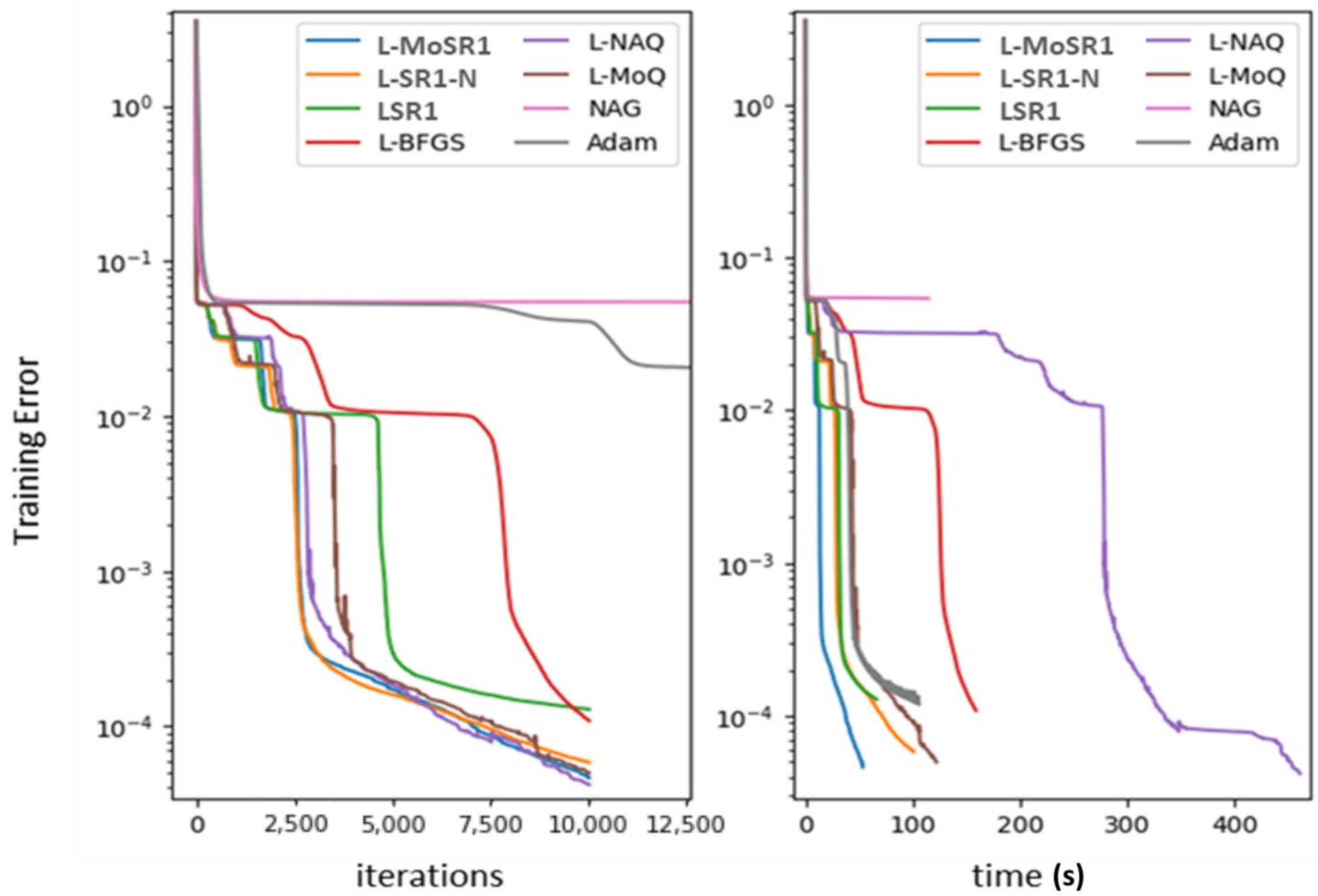
Theorem : Given a level set $\Omega = \{\boldsymbol{w} \in \mathbb{R}^d : E(\boldsymbol{w}) < E(\boldsymbol{w}_0)\}$ that is bounded, let $\{\boldsymbol{w}_k\}$ be the sequence of iterates generated by the SR1+N algorithm. If *Assumptions 1 to 3* holds true, then,

$$\lim_{k \to \infty} ||\nabla E(\mathbf{w}_k)|| = 0.$$

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

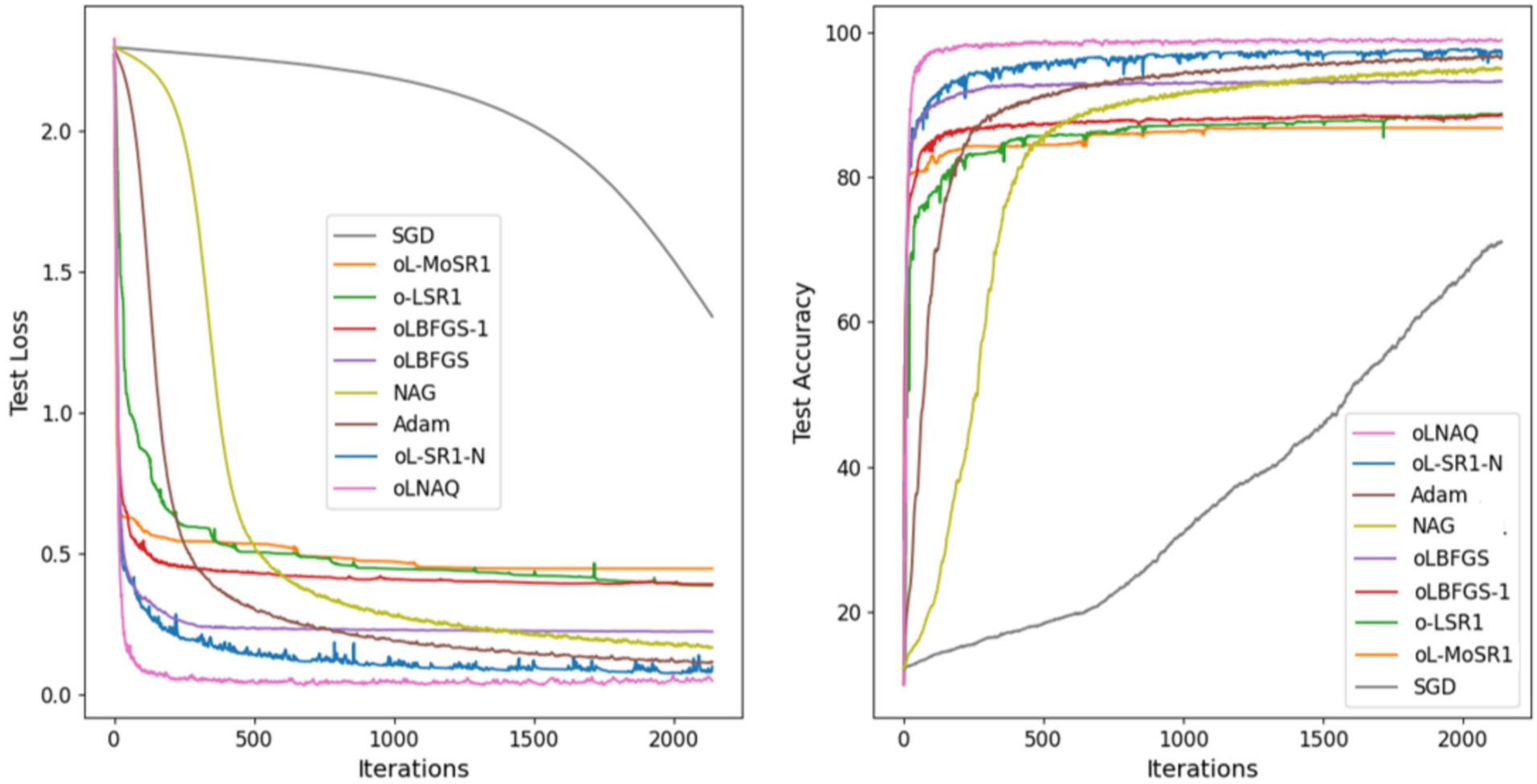
SR1 + NESTEROV'S ACCELERATION (FULL BATCH)

Average results on levy function approximation problem with mL=10 (full batch).



S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

SR1 + NESTEROV'S ACCELERATION (STOCHASTIC)

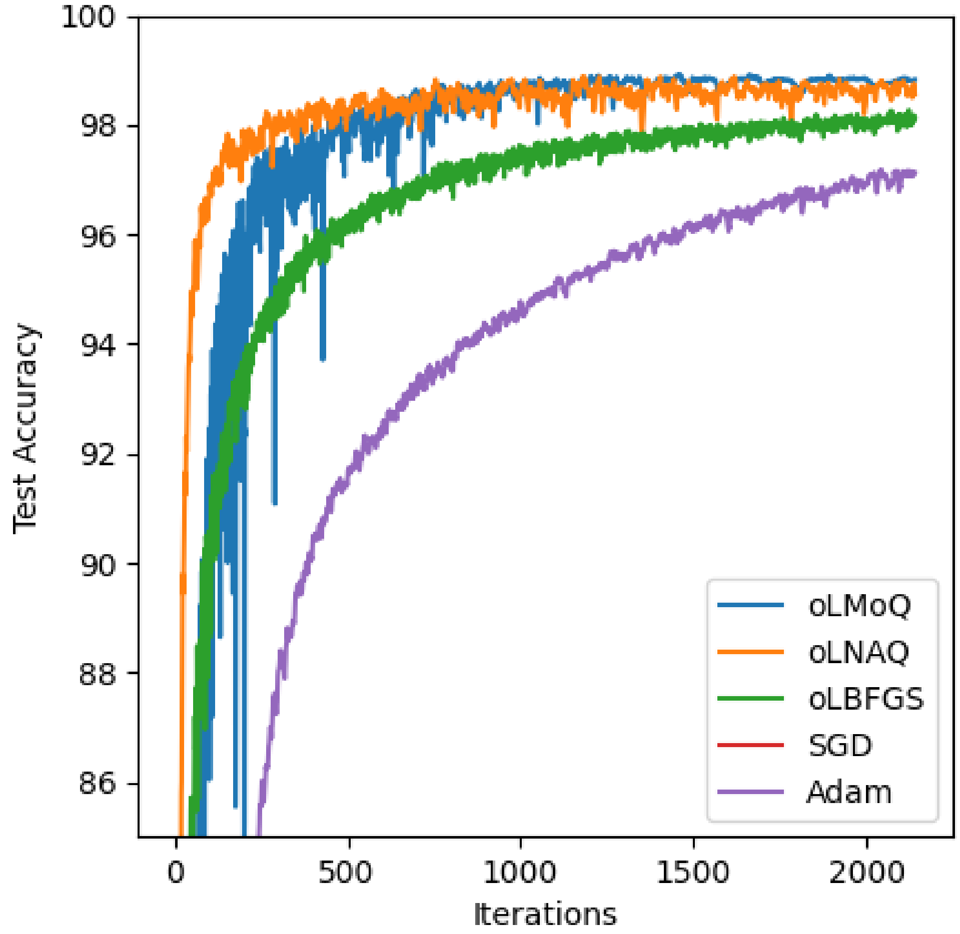


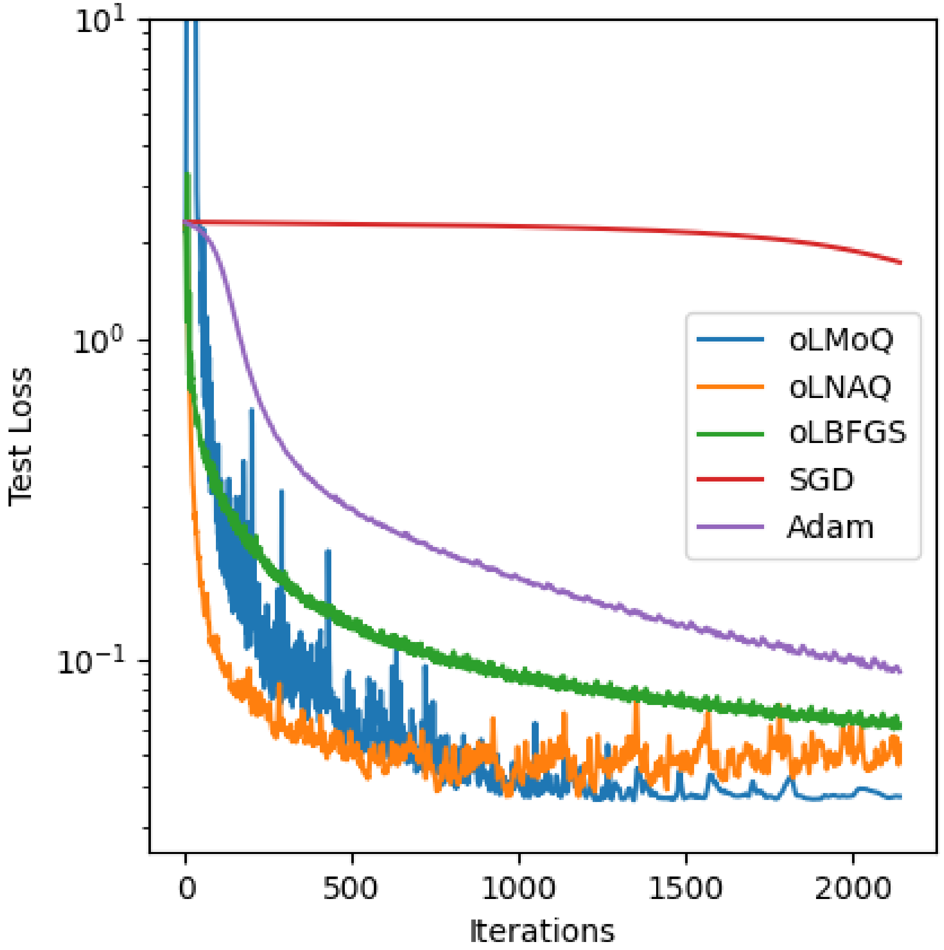
Results of MNIST on LeNet-5 architecture with b=256 and mL=8 .

S. Indrapriyadarsini, Shahrzad Mahboubi, Hiroshi Ninomiya, Takeshi Kamio, Hideki Asai, "Accelerating Symmetric Rank 1 Quasi-Newton Method with Nesterov's Gradient", Algorithms 2022, 15(1), 6;

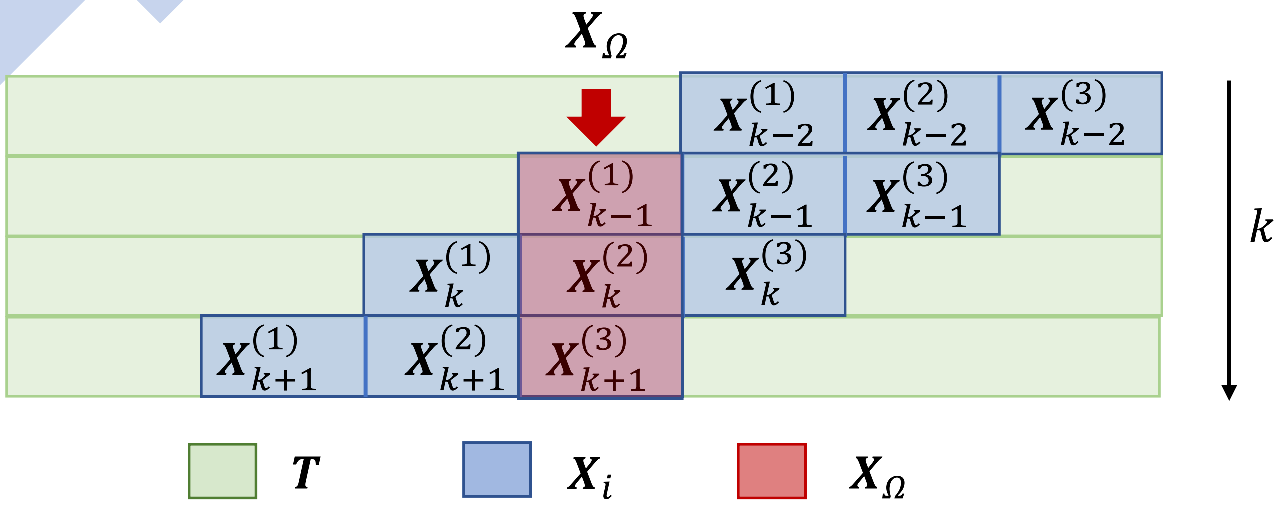
NEXT …

Future Work : oLMoQ stochastic noise reduction





Multi-batch strategy → to MoQ + distributed



$$\nabla E(w_k, X_k) = \frac{1}{n}\sum_{i=1}^{n} \nabla E(w_k, X_i) \quad \text{where } X_i \in T$$

$$= \underbrace{\frac{1}{3n}\sum_{i=1}^{n/3} \nabla E(w_k, X_i)}_{X_k^{(1)}} + \underbrace{\frac{1}{3n}\sum_{i=n/3}^{2n/3} \nabla E(w_k, X_i)}_{X_k^{(2)}} + \underbrace{\frac{1}{3n}\sum_{i=2n/3}^{n} \nabla E(w_k, X_i)}_{X_k^{(3)}}$$

$$y_k = \nabla E\left(w_{k+1}, X_{k+1}^{(3)}\right) - \left\{(1+\mu)\nabla E\left(w_k, X_k^{(2)}\right) - \mu \nabla E\left(w_{k-1}, X_{k-1}^{(1)}\right)\right\}$$

$$= \nabla E(w_{k+1}, X_\Omega) - \{(1+\mu)\nabla E(w_k, X_\Omega) - \mu \nabla E(w_{k-1}, X_\Omega)\}$$

Algorithm 4 oBFGS Method

Require: minibatch X_k, k_{max} and $\lambda \geq 0$,
Initialize: $\mathbf{w}_k \in \mathbb{R}^d$, $\mathbf{H}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$
- $k \leftarrow 1$
- **while** $k < k_{max}$ **do**
  - $\nabla\mathbf{E}_1 \leftarrow \nabla E(\mathbf{w}_k, X_k)$
  - $\mathbf{g}_k \leftarrow -\mathbf{H}_k \nabla E(\mathbf{w}_k, X_k)$
  - $\mathbf{g}_k = \mathbf{g}_k / ||\mathbf{g}_k||_2$
  - Determine α_k using (12)
  - $\mathbf{v}_{k+1} \leftarrow \alpha_k \mathbf{g}_k$
  - $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
  - $\nabla\mathbf{E}_2 \leftarrow \nabla E(\mathbf{w}_{k+1}, X_k)$
  - $\mathbf{s}_k \leftarrow \mathbf{w}_{k+1} - \mathbf{w}_k$
  - $\mathbf{y}_k \leftarrow \nabla\mathbf{E}_2 - \nabla\mathbf{E}_1 + \lambda\mathbf{s}_k$
  - Update $\mathbf{H}_k$ using (4)
  - $k \leftarrow k+1$
- **end while**

Algorithm 5 Proposed oNAQ Method

Require: minibatch X_k, $0 < \mu < 1$ and k_{max}
Initialize: $\mathbf{w}_k \in \mathbb{R}^d$, $\hat{\mathbf{H}}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$
- $k \leftarrow 1$
- **while** $k < k_{max}$ **do**
  - $\nabla\mathbf{E}_1 \leftarrow \nabla E(\mathbf{w}_k + \mu\mathbf{v}_k, X_k)$
  - $\hat{\mathbf{g}}_k \leftarrow -\hat{\mathbf{H}}_k \nabla E(\mathbf{w}_k + \mu\mathbf{v}_k, X_k)$
  - $\hat{\mathbf{g}}_k = \hat{\mathbf{g}}_k / ||\hat{\mathbf{g}}_k||_2$
  - Determine α_k using (17)
  - $\mathbf{v}_{k+1} \leftarrow \mu\mathbf{v}_k + \alpha_k \hat{\mathbf{g}}_k$
  - $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
  - $\nabla\mathbf{E}_2 \leftarrow \nabla E(\mathbf{w}_{k+1}, X_k)$
  - $\mathbf{p}_k \leftarrow \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu\mathbf{v}_k)$
  - $\mathbf{q}_k \leftarrow \nabla\mathbf{E}_2 - \nabla\mathbf{E}_1 + \lambda\mathbf{p}_k$
  - Update $\hat{\mathbf{H}}_k$ using (9)
  - $k \leftarrow k+1$
- **end while**

Algorithm 1: Stochastic MoQ

Require: learning rate schedule, $0 < \mu < 1$ and k_{max}
Ensure: $\mathbf{w}_k \in \mathbb{R}^d$, $\mathbf{H}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$
- Calculate $\nabla\mathbf{E}(\mathbf{w}_k, X_k)$
- **while** $||\nabla\mathbf{E}(\mathbf{w}_k)|| > \epsilon$ and $k < k_{\max}$ **do**
  - Determine learning rate α_k
  - $\nabla\mathbf{E}_1 = (1+\mu)\nabla\mathbf{E}(\mathbf{w}_k, X_k) - \mu\nabla\mathbf{E}(\mathbf{w}_{k-1}, X_{k-1})$
  - $\mathbf{g}_k \leftarrow -\mathbf{H}_k \nabla\mathbf{E}_1$
  - $\mathbf{g}_k = \mathbf{g}_k / ||\mathbf{g}_k||_2$
  - $\mathbf{v}_{k+1} \leftarrow \mu\mathbf{v}_k + \alpha_k\mathbf{g}_k$
  - $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
  - Store $\nabla\mathbf{E}(\mathbf{w}_k, X_k)$
  - Select mini-batch X_{k+1}
  - Calculate $\nabla\mathbf{E}_2 = \nabla\mathbf{E}(\mathbf{w}_{k+1}, X_{k+1})$
  - $\mathbf{s}_k \leftarrow \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu\mathbf{v}_k)$
  - $\mathbf{y}_k \leftarrow \nabla\mathbf{E}_2 - \nabla\mathbf{E}_1 + \lambda\mathbf{s}_k$
  - Update $\mathbf{H}_k$ using (10)
- **end while**

THANK YOU

Contact

Indra Priyadarsini S

Graduate School of Science and Technology
Shizuoka University
indra.ipd@gmail.com

Stochastic Nesterov's Accelerated quasi-Newton – oNAQ

Algorithm 4 oBFGS Method

Require: minibatch X_k, k_{max} and $\lambda \geq 0$,
Initialize: $\mathbf{w}_k \in \mathbb{R}^d$, $\mathbf{H}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$

$k \leftarrow 1$
while $k < k_{max}$ **do**
  $\nabla \mathbf{E}_1 \leftarrow \nabla E(\mathbf{w}_k, X_k)$
  $\mathbf{g}_k \leftarrow -\mathbf{H}_k \nabla E(\mathbf{w}_k, X_k)$
  $\mathbf{g}_k = \mathbf{g}_k / ||\mathbf{g}_k||_2$
  Determine α_k using (12)
  $\mathbf{v}_{k+1} \leftarrow \alpha_k \mathbf{g}_k$
  $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
  $\nabla \mathbf{E}_2 \leftarrow \nabla E(\mathbf{w}_{k+1}, X_k)$
  $\mathbf{s}_k \leftarrow \mathbf{w}_{k+1} - \mathbf{w}_k$
  $\mathbf{y}_k \leftarrow \nabla \mathbf{E}_2 - \nabla \mathbf{E}_1 + \lambda \mathbf{s}_k$
  Update $\mathbf{H}_k$ using (4)
  $k \leftarrow k + 1$
end while

Algorithm 5 Proposed oNAQ Method

Require: minibatch X_k, $0 < \mu < 1$ and k_{max}
Initialize: $\mathbf{w}_k \in \mathbb{R}^d$, $\hat{\mathbf{H}}_k = \epsilon\mathbf{I}$ and $\mathbf{v}_k = 0$

$k \leftarrow 1$
while $k < k_{max}$ **do**
  $\nabla \mathbf{E}_1 \leftarrow \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k, X_k)$
  $\hat{\mathbf{g}}_k \leftarrow -\hat{\mathbf{H}}_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k, X_k)$
  $\hat{\mathbf{g}}_k = \hat{\mathbf{g}}_k / ||\hat{\mathbf{g}}_k||_2$
  Determine α_k using (17)
  $\mathbf{v}_{k+1} \leftarrow \mu \mathbf{v}_k + \alpha_k \hat{\mathbf{g}}_k$
  $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
  $\nabla \mathbf{E}_2 \leftarrow \nabla E(\mathbf{w}_{k+1}, X_k)$
  $\mathbf{p}_k \leftarrow \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu \mathbf{v}_k)$
  $\mathbf{q}_k \leftarrow \nabla \mathbf{E}_2 - \nabla \mathbf{E}_1 + \lambda \mathbf{p}_k$
  Update $\hat{\mathbf{H}}_k$ using (9)
  $k \leftarrow k + 1$
end while

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "A Stochastic Quasi-Newton Method with Nesterov's Accelerated Gradient", Joint European Conference on Machine Learning and Principles of Knowledge Discovery in Databases, ECML-PKDD, Springer, 2019

Adaptive Stochastic Nesterov's Accelerated quasi-Newton – aSNAQ

Algorithm 2 Proposed method - aSNAQ

Require: minibatch X_k, μ_{min}, μ_{max}, k_{max}, aFIM buffer F of size m_F and curvature pair buffer (S, Y) of size m_L , momentum update factor ϕ

Initialize: $\mathbf{w}_k \in \mathbb{R}^d$, $\mu = \mu_{min}$, $\mathbf{v}_k$, $\mathbf{w}_o$, $\mathbf{v}_o$, $\mathbf{w}_s$, $\mathbf{v}_s$, k & $t = 0$

while $k < k_{max}$ **do**
    Calculate $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$
    Determine $\mathbf{g}_k$ using Algorithm 1
    $\mathbf{g}_k = \mathbf{g}_k / \|\mathbf{g}_k\|_2$ ▷ Direction normalization
    $\mathbf{v}_{k+1} \leftarrow \mu \mathbf{v}_k + \alpha_k \mathbf{g}_k$
    $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \mathbf{v}_{k+1}$
    Calculate $\nabla E(\mathbf{w}_{k+1})$ and store in F
    $\mathbf{w}_s = \mathbf{w}_s + \mathbf{w}_k$
    $\mathbf{v}_s = \mathbf{v}_s + \mathbf{v}_k$
    if mod(k , L) = 0 **then**
        Compute average $\mathbf{w}_n = \mathbf{w}_s / L$ and $\mathbf{v}_n = \mathbf{v}_s / L$
        $\mathbf{w}_s = 0$ and $\mathbf{v}_s = 0$
        if $t > 0$ **then**
            if $E(\mathbf{w}_n) > \gamma E(\mathbf{w}_o)$ **then**
                Clear (S, Y) and F buffers
                Reset $\mathbf{w}_k = \mathbf{w}_o$ and $\mathbf{v}_k = \mathbf{v}_o$
                Update $\mu = \max(\mu/\phi, \mu_{min})$
                continue
            end if
            $\mathbf{s} = \mathbf{w}_n - \mathbf{w}_o$
            $\mathbf{y} = \frac{1}{|F|}(\sum_{i=1}^{|F|} F_i \cdot \mathbf{s})$
            Update $\mu = \min(\mu \cdot \phi, \mu_{max})$
            if $\mathbf{s}^T \mathbf{y} > \epsilon \cdot \mathbf{y}^T \mathbf{y}$ **then**
                Store curvature pairs (**s**,**y**) in (S, Y)
            end if
        end if
        Update $\mathbf{w}_o = \mathbf{w}_n$ and $\mathbf{v}_o = \mathbf{v}_n$
        $t \leftarrow t + 1$
    end if
    $k \leftarrow k + 1$
end while

Indrapriyadarsini S., Shahrzad Mahboubi, Hiroshi Ninomiya, and Hideki Asai. "An Adaptive Stochastic Nesterov's Accelerated Quasi-Newton Method for Training RNNs", Nonlinear Theory and its Applications, NOLTA, IEICE, 2019

DERIVATION OF NAQ

- $\mathbf{w}_{k+1} = (\mathbf{w}_k + \mu \mathbf{v}_k) - \nabla^2 E(\mathbf{w}_k + \mu \mathbf{v}_k)^{-1} \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$ ... $(Eq.\mathbf{12})$
- By approximation of the Hessian $\nabla^2 E(\mathbf{w}_k + \mu \mathbf{v}_k)$ using $\widehat{\mathbf{B}}_{k+1}$,
- ***Secant Condition***

$$\mathbf{q}_k = \widehat{\boldsymbol{B}}_{k+1} \mathbf{p}_k \qquad \ldots (Eq.\mathbf{13})$$

$$\mathbf{p}_k = \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu \mathbf{v}_k), \qquad \mathbf{q}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$$

- From secant condition the rank-2 updating formula of this matrix is derived as follows:
- <The update formula of $\widehat{\boldsymbol{B}}_{k+1}$>
  - The matrix $\widehat{\boldsymbol{B}}_{k+1}$ is defined using arbitrary vectors $\mathbf{x}$ and $\mathbf{y}$ and constants a and b as

$$\widehat{\boldsymbol{B}}_{k+1} = \widehat{\boldsymbol{B}}_k + \widehat{\boldsymbol{B}}_k + a\mathbf{x}\mathbf{x}^{\mathrm{T}} + b\mathbf{y}\mathbf{y}^{\mathrm{T}} \qquad \ldots (Eq.\mathbf{14})$$

APPENDIX

- By substituting (14) into the secant condition, arbitrary vectors $\mathbf{x}$ and $\mathbf{y}$ and constants a and b are obtained as

$$\mathbf{x} = \mathbf{q}_k\ ,\ \mathbf{y} = -\widehat{\boldsymbol{B}}_k\mathbf{p}_k \text{ and } a = 1/\mathbf{x}^{\mathrm{T}}\mathbf{p}_k\ ,\ b = 1/\mathbf{y}^{\mathrm{T}}\mathbf{p}_k \quad \ldots(Eq.\,\mathbf{15})$$

- As a result, the rank-2 updating formula for NAQ can be obtained as

$$\widehat{\boldsymbol{B}}_{k+1} = \widehat{\boldsymbol{B}}_k + \mathbf{q}_k\mathbf{q}_k{}^{\mathrm{T}}/\mathbf{q}_k{}^{\mathrm{T}}\mathbf{p}_k - \widehat{\boldsymbol{B}}_k\mathbf{p}_k\mathbf{p}_k{}^{\mathrm{T}}\widehat{\boldsymbol{B}}_k/\mathbf{p}_k{}^{\mathrm{T}}\widehat{\boldsymbol{B}}_k\mathbf{p}_k \quad \ldots(Eq.\,\mathbf{16})$$

- Convergence properties of NAQ -> Proof is omitted here.

a. $\widehat{\boldsymbol{B}}_{k+1}$ of (16) satisfies the secant condition $\mathbf{q}_k = \widehat{\boldsymbol{B}}_{k+1}\mathbf{p}_k$.

b. If $\widehat{\boldsymbol{B}}_k$ is symmetry, $\widehat{\boldsymbol{B}}_{k+1}$ is also symmetry.

c. If $\widehat{\boldsymbol{B}}_k$ is the positive definite matrix, $\widehat{\boldsymbol{B}}_{k+1}$ is also the positive definite.

Modified Nesterov's Accelerated quasi-Newton - mNAQ

1) Incorporating an additional $\hat{\xi}_k p_k$ term for global convergence

$$p_k = w_{k+1} - (w_k + \mu v_k)$$

$$q_k = \nabla E(w_{k+1}) - \nabla E(w_k + \mu v_k) + \hat{\xi}_k p_k = \varepsilon_k + \hat{\xi}_k p_k \qquad \ldots (Eq.10)$$

global convergence term

$$\hat{\xi}_k = \omega \|\nabla E(w_k + \mu v_k)\| + max\left\{-\frac{\varepsilon_k^T p_k}{\|p_k\|^2}, 0\right\}$$

$$\begin{cases} \omega = 2 & if \ \|\nabla E(w_k + \mu v_k)\|^2 > 10^{-2} \\ \omega = 100 & if \ \|\nabla E(w_k + \mu v_k)\|^2 < 10^{-2} \end{cases}$$

$$\hat{H}_{k+1} = (I - \rho_k p_k {q_k}^T)\hat{H}_k(I - \rho_k q_k {p_k}^T) + \rho_k p_k {p_k}^T \qquad \ldots (Eq.11)$$

Modified Nesterov's Accelerated quasi-Newton - mNAQ

2) Eliminating linesearch

Determine step size α_k using the explicit formula

$$\boldsymbol{\alpha_k} = -\frac{\boldsymbol{\delta \nabla E(w_k + \mu v_k)^T \hat{g}_k}}{\boldsymbol{\|\hat{g}_k\|^2}_{\boldsymbol{Q_k}}} \quad \ldots (Eq.\,\mathbf{12})$$

Armijo Linesearch Condition:

$$\boldsymbol{E}(\boldsymbol{w_k} + \mu \boldsymbol{v_k} + \alpha_k \boldsymbol{g_k}) \leq \boldsymbol{E}(\boldsymbol{w_k} + \mu \boldsymbol{v_k}) + \eta \alpha_k \boldsymbol{\nabla E}(\boldsymbol{w_k} + \mu \boldsymbol{v_k})^{\boldsymbol{T}} \boldsymbol{g_k}$$

where $\boldsymbol{g_k} = -\widehat{\boldsymbol{H}}_k \boldsymbol{\nabla E}(\boldsymbol{w_k} + \mu \boldsymbol{v_k})$