Metric Perspective of Stochastic Optimizers

Asahi Ushio

E-mail: ushio@ykw.elec.keio.ac.jp
Dept. Electronics and Electrical Engineering, Keio University, Japan

Internal Research Seminar
July 19, 2017

Note: the author has moved to Cogent Labs as a researcher from April 2018.

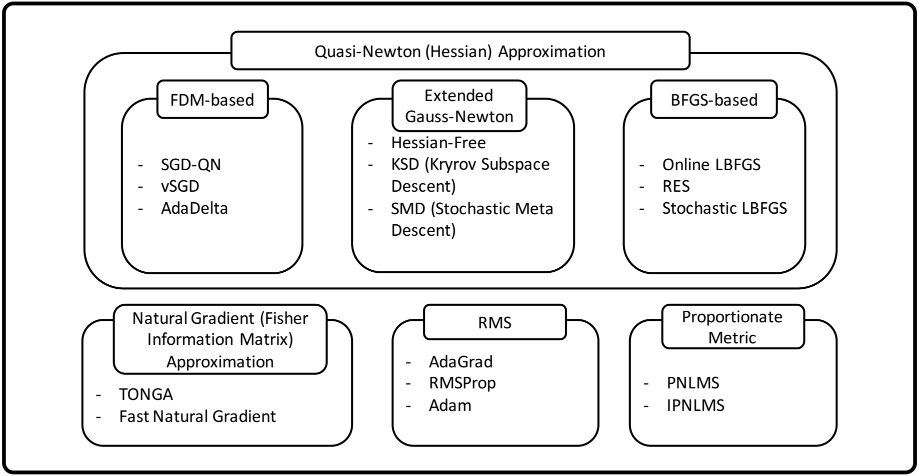
Overview

Outline

Explanation of stochastic optimizers in machine learning, especially, from the perspective of each metric.

- Mostly, stochastic optimizers can be divided into three types of metric.
  1. Quasi-Newton Method Type
     1. Finite Difference Method (FDM): SGD-QN [1], AdaDelta [2], VSGD [3, 4]
     2. Extended Gauss-Newton: KSD [5] , SMD [6], HF [7]
     3. LBFGS: stochastic LBFGS [8, 9], RES [10],
  2. Natural Gradient Type: Natural Gradient [11], TONGA [12, 13]
  3. Root Mean Square (RMS) Type: AdaGrad [14], RMSprop [15], Adam [16]

Overview of Stochastic Algorithm



Problem Setting

- **Model:** For an input $\boldsymbol{x}_t \in \mathbb{R}^n$, the output $\hat{y}_t \in \mathbb{R}$ is derived by

$$\text{Activation} : \hat{y}_t = M(z_t) \tag{1}$$

$$\text{Output} : z_t = N_{\boldsymbol{w}}(\boldsymbol{x}_t) \in \mathbb{R}. \tag{2}$$

- **Loss:** With instantaneous loss function $l_t(\boldsymbol{w})$ of parameter $\boldsymbol{w}$,

$$L(\boldsymbol{w}) = \mathbb{E}\left[l_t(\boldsymbol{w})\right]_t . \tag{3}$$

Problem Setting	$M(z)$	$N_{\boldsymbol{w}}(\boldsymbol{x})$	$l_t(\hat{y}_t)$	y_t
Regression	z	$\boldsymbol{w}^\top \boldsymbol{x}$	$\frac{\|\|\hat{y}_t - y_t\|\|^2}{2}$	$y_t \in \mathbb{R}$
Classification	$\frac{1}{1+e^{-z}}$	$\boldsymbol{w}^\top \boldsymbol{x}$	$-\left[y_t \log(\hat{y}_t) + (1-y_t)\log(1-\hat{y}_t)\right]$	$y_t \in \{0,1\}$
Multi-Classification	$\frac{e^{z_i}}{\sum_{i=1}^k e^{z_i}}$	$\boldsymbol{w}_i^\top \boldsymbol{x}$	$-\sum_{i=1}^k y_{t,i} \log(\hat{y}_i)$	$y_{t,i} \in \{0,1\}$

Brief Illustration of Model

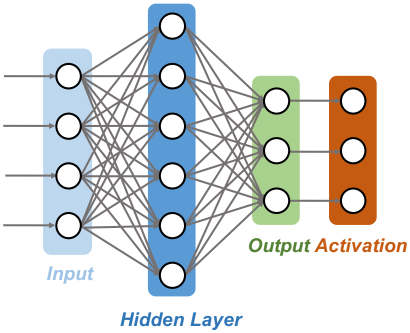


Figure: Neural Network

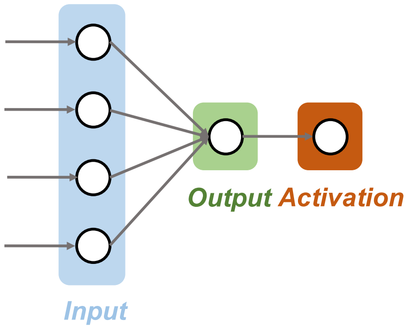


Figure: Linear Model

Stochastic Optimization

- **Purpose:** Find $\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} L(\boldsymbol{w})$ by stochastic approximation.

SGD (Stochastic Gradient Descent) and Varinats

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_t \boldsymbol{g}_t, \quad \eta_t \in \mathbb{R} \text{ s.t. } \lim_{t\to\infty} \eta_t = 0 \text{ and } \lim_{t\to\infty} \sum_{i=1}^{t} \eta_i = \infty \tag{4}$$

$$\text{Vanilla SGD}: \quad \boldsymbol{g}_t = \frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1})$$

$$\text{Momentum}: \quad \boldsymbol{g}_t = \gamma \boldsymbol{g}_{t-1} + (1-\gamma)\frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1}), \quad \gamma \in \mathbb{R}$$

$$\text{NAG}: \quad \boldsymbol{g}_t = \gamma \boldsymbol{g}_{t-1} + (1-\gamma)\frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1} - \gamma \boldsymbol{g}_{t-1})$$

$$\text{Minibatch}: \quad \boldsymbol{g}_t = \frac{1}{I}\sum_{i=0}^{I-1} \frac{d}{d\boldsymbol{w}} l_{t-i}(\boldsymbol{w}_{t-1}), \quad I \in \mathbb{R}.$$

- Suppose $l_t(\boldsymbol{w})$ is differentiable.

Quasi-Newton Method

- **Newton Method** employs Hessian matrix:

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - B_t \boldsymbol{g}_t \tag{5}$$
$$B_t = H_t^{-1}, \quad H_t = \frac{d^2 L(\boldsymbol{w}_{t-1})}{d\boldsymbol{w}^2}.$$

- **Quasi-Newton** employs Hessian approximation $\hat{H}_t$ instead of H_t.
  1. FDM (Finite Difference Method): SGD-QN [1], AdaDelta [2], VSGD [3, 4]
  2. Extended Gauss-Newton Approximation
  3. LBFGS
- Diagonal approximation is often used in stochastic optimization:

$$\hat{H}_t = \mathrm{diag}\left(h_{1,t} \dots h_{n,t}\right) \tag{6}$$

SGD-QN

- **SGD-QN** [1] employs instantaneous estimator of Hessian:

$$\frac{1}{h_{i,t}} := \frac{\eta}{t}\mathbb{E}\left[\frac{1}{h_{i,\tau}^{FDM}}\right]_{\tau=1}^{t} \tag{7}$$

$$\mathbb{E}\left[\frac{1}{h_{i,\tau}^{FDM}}\right]_{\tau=1}^{t} \approx \frac{1}{\bar{h}_{i,t}} := \alpha\frac{1}{h_{i,t}^{FDM}} + (1-\alpha)\frac{1}{\bar{h}_{i,t-1}} \tag{8}$$

$$h_{i,t}^{FDM} := \frac{g_{i,t} - g_{i,t-1}}{w_{i,t-1} - w_{i,t-2}} \tag{9}$$

- The FDM approximation (9) is called secant condition in quasi-newton method context.

AdaDelta

- SGD updates (5) can be reformulated

$$B_t = -(\boldsymbol{w}_t - \boldsymbol{w}_{t-1})\boldsymbol{g}_t^{-\mathsf{T}}. \tag{10}$$

- **AdaDelta** [2] approximates Hessian by (10) :

$$h_{i,t} := \mathbb{E}\left[-\frac{g_{i,t}}{w_{i,t} - w_{i,t-1}}\right]_{\tau=1}^{t} \approx \frac{\mathrm{RMS}\left[g_{i,t}\right]}{\mathrm{RMS}\left[w_{i,t-1} - w_{i,t-2}\right]}. \tag{11}$$

  Here $w_{i,t} - w_{i,t-1}$ is not known at t, so approximated by $w_{i,t-1} - w_{i,t-2}$.

- With numerical stability parameter ϵ: (sensitive parameter in practice)

$$\mathrm{RMS}\left[g_t\right] := \sqrt{\mathbb{E}\left[g_\tau^2\right]_{\tau=1}^{t} + \epsilon} \tag{12}$$

$$\mathbb{E}\left[g_\tau^2\right]_{\tau=1}^{t} \approx \bar{g}_t := \alpha g_t^2 + (1-\alpha)\bar{g}_{t-1} \tag{13}$$

VSGD: Quadratic Loss

Quadratic Approximation of Loss

- Taylor expansion gives

$$l_t(\boldsymbol{w}) \approx l_t(\boldsymbol{a}) + \frac{dl_t(\boldsymbol{a})}{d\boldsymbol{w}}^{\mathsf{T}}(\boldsymbol{w}-\boldsymbol{a}) + \frac{1}{2}(\boldsymbol{w}-\boldsymbol{a})^{\mathsf{T}}\frac{dl_t^2(\boldsymbol{a})}{d\boldsymbol{w}^2}(\boldsymbol{w}-\boldsymbol{a}), \quad \forall \boldsymbol{a} \in \mathbb{R}^n$$

- For $\hat{\boldsymbol{w}}_t = \underset{\boldsymbol{w}\in\mathbb{R}^n}{\arg\,\min}\, l_t(\boldsymbol{w})$,

$$l_t(\hat{\boldsymbol{w}}_t) = 0, \quad \frac{dl_t(\hat{\boldsymbol{w}}_t)}{d\boldsymbol{w}} = \boldsymbol{0}. \tag{14}$$

Then $l_t(\boldsymbol{w})$ can be locally approximated by the quadratic function

$$l_t(\boldsymbol{w}) \approx \frac{1}{2}(\boldsymbol{w}-\hat{\boldsymbol{w}}_t)^{\mathsf{T}}\frac{d^2 l_t(\hat{\boldsymbol{w}}_t)}{d\boldsymbol{w}^2}(\boldsymbol{w}-\hat{\boldsymbol{w}}_t). \tag{15}$$

VSGD: Noisy Quadratic Loss

Noisy Quadratic Approximation of Loss with Diagonal Hessian Approximation

- Suppose $\hat{\boldsymbol{w}}_t \sim \mathcal{N}(\hat{\boldsymbol{w}}, \mathrm{diag}\left(\sigma_1^2, \ldots, \sigma_n^2\right))$:

$$
\begin{aligned}
l_t^q(\boldsymbol{w}) :&= \frac{1}{2}(\boldsymbol{w} - \hat{\boldsymbol{w}}_t)^\mathsf{T} \hat{H}_t (\boldsymbol{w} - \hat{\boldsymbol{w}}_t) \\
&= \frac{1}{2} \sum_{i=1}^{n} h_{i,t}(w_i - \hat{w}_{i,t})^2 \qquad (16)
\end{aligned}
$$

- SGD for l_t^q with element-wise learning rate $\eta_{i,t}$:

$$
\begin{aligned}
w_{i,t} &= w_{i,t-1} - \eta_{i,t} \frac{d l_t^q(\boldsymbol{w}_{t-1})}{d w_i} \\
&= w_{i,t-1} - \eta_{i,t} h_{i,t}(w_{i,t-1} - \hat{w}_{i,t}) \\
&= w_{i,t-1} - \eta_{i,t} h_{i,t}(w_{i,t-1} - \hat{w}_i + u_i), \quad u_{i,t} \sim \mathcal{N}(0, \sigma_i^2). \qquad (17)
\end{aligned}
$$

VSGD: Adaptive Learning Rate

Greedy Optimal Learning Rate

- **VSGD (variance SGD)** [3, 4] choose the learning rate, which minimize the conditional expectation of loss function:

$$\begin{aligned} \eta_{i,t} :&= \arg\min_{\eta} \left\{ \mathbb{E}\left[l_t^q(w_{i,t}) | w_{i,t-1} \right]_t \right\} \\ &= \arg\min_{\eta} \left\{ \mathbb{E}\left[\left(w_{i,t-1} - \frac{1}{\eta} h_{i,t}(w_{i,t-1} - \hat{w}_i + u_i) - \hat{w}_{i,t} \right)^2 \right]_t \right\} \\ &= \frac{1}{h_{i,t}} \frac{(w_{i,t-1} - \hat{w}_i)^2}{(w_{i,t-1} - \hat{w}_i)^2 + \sigma_i^2} \end{aligned} \tag{18}$$

VSGD: Variance Approximation

- In practice, $\hat{\boldsymbol{w}}$ is unknown so approximated by

$$
\begin{aligned}
(w_{i,t-1} - \hat{w}_i)^2 &= \left(\mathbb{E}\left[w_{i,\tau} - \hat{w}_{i,\tau}\right]_{\tau=1}^{t-1}\right)^2 \\
&= \left(\mathbb{E}\left[g_{i,\tau}\right]_{\tau=1}^{t-1}\right)^2
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
(w_{i,t-1} - \hat{w}_i)^2 + \sigma_i^2 &= \mathbb{E}\left[(w_{i,\tau} - \hat{w}_{i,\tau})^2\right]_{\tau=1}^{t-1} \\
&= \mathbb{E}\left[g_{i,\tau}^2\right]_{\tau=1}^{t-1}
\end{aligned} \tag{20}
$$

where

$$
\mathbb{E}\left[g_{i,\tau}\right]_{\tau=1}^{t} \approx \bar{g}_{i,t} := \alpha_{i,t} g_{i,t} + (1 - \alpha_{i,t}) \bar{g}_{i,t-1} \tag{21}
$$

$$
\mathbb{E}\left[g_{i,\tau}^2\right]_{\tau=1}^{t} \approx \bar{v}_{i,t} := \alpha_{i,t} g_{i,t}^2 + (1 - \alpha_{i,t}) \bar{v}_{i,t-1}. \tag{22}
$$

VSGD: Hessian Approximation

- Hessian is approximated by $h_{i,t} := \mathbb{E}\left[h_{i,\tau}\right]_{\tau=1}^{t}$.
- Two approximation of $\mathbb{E}\left[h_{i,\tau}\right]_{\tau=1}^{t}$ based on FDM:

$$\text{(Scheme 1)}\ \bar{h}_{i,t} := \alpha_{i,t}\hat{h}_{i,t} + (1-\alpha_{i,t})\bar{h}_{i,t} \tag{23}$$

$$\text{(Scheme 2)}\ \bar{h}_{i,t} := \frac{\mathbb{E}\left[\left(\hat{h}_{i,t}\right)^2\right]}{\mathbb{E}\left[\hat{h}_{i,t}\right]} \tag{24}$$

$$\mathbb{E}\left[\left(\hat{h}_{i,t}\right)^2\right] \approx v_{i,t} := \alpha_{i,t}\left(\hat{h}_{i,t}\right)^2 + (1-\alpha_{i,t})v_{i,t}$$

$$\mathbb{E}\left[\hat{h}_{i,t}\right] \approx m_{i,t} := \alpha_{i,t}\hat{h}_{i,t} + (1-\alpha_{i,t})m_{i,t}$$

where

$$\hat{h}_{i,t} := \frac{g_{i,t} - dl_t(w_{i,t} + \bar{g}_{i,t})/d\boldsymbol{w}}{\bar{g}_{i,t}}. \tag{25}$$

VSGD: Weight Decay

Weight Sequence: Weight is update by following heuristic rule

$$\alpha_{i,t} := \left(1 - \frac{\bar{g}_{i,t}^2}{\bar{v}_{i,t}}\right) \alpha_{i,t-1} + 1. \tag{26}$$

RMS: AdaGrad, RMSprop, Adam

- **AdaGrad** [14], **Adam** [16], **RMSProp** [15] can be summarized as

$$B_t = \eta_t R_t \tag{27}$$

$$R_t := \operatorname{diag}\left(1/r_{1,t} \dots 1/r_{n,t}\right) \tag{28}$$

$$r_{i,t} := \mathrm{RMS}\left[g_{i,t}\right] = \sqrt{\mathbb{E}\left[g_{i,t}^2\right]} \tag{29}$$

- Approximation of expectation and learning rate is different:

$$(\text{AdaGrad})\ \mathbb{E}\left[g_{i,t}^2\right] \approx \frac{1}{t}\sum_{\tau=1}^{t} g_{i,\tau}^2, \quad \eta_t = \eta/\sqrt{t} \tag{30}$$

$$(\text{RMSProp})\ \mathbb{E}\left[g_{i,t}^2\right] \approx \bar{v}_{i,t}, \quad \eta_t = \eta \tag{31}$$

$$(\text{Adam})\ \mathbb{E}\left[g_{i,t}^2\right] \approx \hat{v}_{i,t} := \frac{\bar{v}_{i,t}}{1-\alpha^t}, \quad \eta_t = \eta\frac{1}{1-\gamma^t} \tag{32}$$

where $\bar{v}_{i,t} := \alpha g_{i,t}^2 + (1-\alpha)\bar{v}_{i,t-1}$.

Adam

Moment Bias

- For the momentum sequence:

$$\boldsymbol{g}_t := \gamma \boldsymbol{g}_{t-1} + (1-\gamma)\frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1}) = (1-\gamma)\sum_{i=1}^{t} \gamma^{t-i} \frac{d}{d\boldsymbol{w}} l_i(\boldsymbol{w}_{i-1}),$$

the expectation of $\boldsymbol{g}_t$ includes **moment bias** $(1-\gamma^t)$ such as

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{g}_t\right]_t &= \mathbb{E}\left[(1-\gamma)\sum_{i=1}^{t} \gamma^{t-i} \frac{d}{d\boldsymbol{w}} l_i(\boldsymbol{w}_{i-1})\right]_t \\ &= \mathbb{E}\left[\frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1})\right]_t (1-\gamma)\sum_{i=1}^{t}\gamma^{t-i} = \mathbb{E}\left[\frac{d}{d\boldsymbol{w}} l_t(\boldsymbol{w}_{t-1})\right]_t (1-\gamma^t).\end{aligned}$$

- Adam's learning rate is aimed to reduce the moment bias.

Extended Gauss-Newton

Relation to the Gauss-Newton

- Gauss-Newton is an approximation of Hessian, which is limited to the squared loss function.
- **Extended Gauss-Newton** is an extension of Gauss-Newton by [6].
  1. Applicable to any loss function.

Multilayer Perceptron Model

- The loss function (3) can be seen as

$$L(\hat{y}) = \mathbb{E}\left[l_t(\hat{y}_t)\right]_t \tag{33}$$

where

$$\text{Activation} : \hat{y}_t = M(z_t) \tag{1}$$

$$\text{Output} : z_t = N_{\boldsymbol{w}}(\boldsymbol{x}_t). \tag{2}$$

Extended Gauss-Newton: Hessian Derivation

- Hessian of (33) is

$$
\begin{aligned}
H &= \frac{d^2 L(\hat{y})}{d\boldsymbol{w}^2} \\
&= \frac{d}{d\boldsymbol{w}} \left\{ \frac{dz}{d\boldsymbol{w}} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) \right\} \\
&= \frac{dz}{d\boldsymbol{w}} \frac{d}{d\boldsymbol{w}} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right)^{\mathsf{T}} + \frac{d^2 z}{d\boldsymbol{w}^2} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) \\
&= \frac{dz}{d\boldsymbol{w}} \frac{dz}{d\boldsymbol{w}}^{\mathsf{T}} \frac{d}{dz} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) + \frac{d^2 z}{d\boldsymbol{w}^2} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) \\
&= J_N J_N^{\mathsf{T}} \frac{d}{dz} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) + \frac{d^2 z}{d\boldsymbol{w}^2} \left(\frac{d\hat{y}}{dz} \frac{dL(\hat{y})}{d\hat{y}} \right) \qquad (34)
\end{aligned}
$$

where $J_N = \frac{dz}{d\boldsymbol{w}}$ is Jacobian.

Extended Gauss-Newton: General Case and Regression

Extended Gauss-Newton

- Ignore the 2nd order derivation in (34),

$$H_{GN} = J_N J_N^\mathsf{T} \frac{d}{dz}\left(\frac{d\hat{y}}{dz}\frac{dL(\hat{y})}{d\hat{y}}\right). \tag{35}$$

- **Regression:** Since $\frac{d\hat{y}}{dz} = 1$,

$$\frac{d}{dz}\left(\frac{d\hat{y}}{dz}\frac{dL(\hat{y})}{d\hat{y}}\right) = \mathbb{E}_t\left[\frac{d}{dz}\frac{d}{d\hat{y}}l_t(\hat{y})\right] = \mathbb{E}_t\left[\frac{d}{dz}(\hat{y} - y_t)\right] = 1. \tag{36}$$

- Extended Gauss-Newton approximation is

$$H_{GN} = J_N J_N^\mathsf{T} \tag{37}$$

Extended Gauss-Newton: Classification

$$\frac{d}{dz}\left(\frac{d\hat{y}}{dz}\frac{dL(\hat{y})}{d\hat{y}}\right) = \mathbb{E}_t\left[\frac{d}{dz}\left(\frac{d\hat{y}}{dz}\frac{d}{d\hat{y}}l_t(\hat{y})\right)\right] = -\mathbb{E}_t\left[\frac{d}{dz}\left((\hat{y}-\hat{y}^2)\left(\frac{y_t}{\hat{y}} - \frac{1-y_t}{1-\hat{y}}\right)\right)\right]$$
$$= \mathbb{E}_t\left[\frac{d}{dz}(\hat{y} - y_t)\right] = \hat{y} - \hat{y}^2 \tag{38}$$

where

$$\frac{d\hat{y}}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} - \frac{1}{(1+e^{-z})^2} = \hat{y} - \hat{y}^2$$
$$\frac{d}{d\hat{y}}l_t(\hat{y}) = -\left(\frac{y_t}{\hat{y}} - \frac{1-y_t}{1-\hat{y}}\right)$$

- Extended Gauss-Newton approximation is

$$H_{GN} = (\hat{y} - \hat{y}^2)J_N J_N^{\mathsf{T}} \tag{39}$$

Extended Gauss-Newton: Multi-class Classification

$$\begin{aligned}
\frac{d}{dz_i}\left(\frac{d\hat{y}_i}{dz_i}\frac{d}{d\hat{y}_i}l_t(\hat{y})\right) &= \frac{d}{dz_i}\left[\left\{\frac{e^{z_i}}{\sum_{i=1}^{k}e^{z_i}} - \left(\frac{e^{z_i}}{\sum_{i=1}^{k}e^{z_i}}\right)^2\right\}\left(-\frac{\sum_{i=1}^{k}e^{z_i}}{e^{z_i}}\right)\right] \\
&= \frac{d}{dz_i}\left[\frac{e^{z_i}}{\sum_{i=1}^{k}e^{z_i}} - 1\right] \\
&= \frac{e^{z_i}}{\sum_{i=1}^{k}e^{z_i}} - \left(\frac{e^{z_i}}{\sum_{i=1}^{k}e^{z_i}}\right)^2 = \hat{y}_i - \hat{y}_i^2
\end{aligned} \tag{40}$$

- Extended Gauss-Newton approximation is

$$H_{GN,i} = (\hat{y}_i - \hat{y}_i^2)J_{N,i}J_{N,i}^{\mathsf{T}} \tag{41}$$

Natural Gradient

Fisher Information Matrix

- Suppose the observation y is sampled via

$$y \sim p(\hat{y}). \tag{42}$$

- Then **fisher information matrix** becomes

$$F = \frac{d \log p(y|\hat{y})}{d\boldsymbol{w}} \frac{d \log p(y|\hat{y})}{d\boldsymbol{w}}^{\mathsf{T}}. \tag{43}$$

Natural Gradient

$$\begin{aligned} \boldsymbol{w}_t &= \boldsymbol{w}_{t-1} - B_t \boldsymbol{g}_t \\ B_t &= F^{-1} \end{aligned} \tag{44}$$

Natural Gradient: Regression

- **Regression:** Suppose gaussian distribution,

$$p(y|\hat{y},\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\hat{y}-y)^2}{2\sigma^2}}. \tag{45}$$

- Fisher information matrix:

$$F = \frac{(\hat{y}-y)^2}{\sigma^4} J_N J_N^{\mathsf{T}} \tag{46}$$

where

$$\begin{aligned} \frac{d \log p(y|\hat{y},\sigma)}{d\boldsymbol{w}} &= \frac{d}{d\boldsymbol{w}} \left\{ -\frac{(\hat{y}-y)^2}{2\sigma^2} \right\} \\ &= -\frac{\hat{y}-y}{\sigma^2} \frac{d\hat{y}}{d\boldsymbol{w}} \\ &= -\frac{\hat{y}-y}{\sigma^2} \frac{dz}{d\boldsymbol{w}} \frac{d\hat{y}}{dz} = -\frac{\hat{y}-y}{\sigma^2} J_N \end{aligned} \tag{47}$$

Natural Gradient: Classification

- **Classification:** Suppose binomial distribution,

$$p(y|\hat{y}) = \hat{y}^{y}(1-\hat{y})^{1-y}. \tag{48}$$

- Fisher information matrix:

$$F = (y-\hat{y})^2 J_N J_N^{\mathsf{T}} \tag{49}$$

where

$$\begin{aligned}
\frac{d \log p(y|\hat{y},\sigma)}{d\boldsymbol{w}} &= \frac{dz}{d\boldsymbol{w}} \frac{d\hat{y}}{dz} \frac{d}{d\hat{y}} \left\{ y \log \hat{y} + (1-y) \log (1-\hat{y}) \right\} \\
&= J_N \frac{d\hat{y}}{dz} \left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}} \right) \\
&= J_N (1-\hat{y})\hat{y} \left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}} \right) = J_N (y-\hat{y})
\end{aligned} \tag{50}$$

Natural Gradient: Multi-class Classification

- **Multi-class Classification:** Suppose multinomial distribution

$$p(y_1, .., y_k | \hat{y}_1, ..., \hat{y}_k) = \prod_{i=1}^{k} \hat{y}_i^{y_i} \tag{51}$$

- Fisher information matrix:

$$F_i = \{(1 - \hat{y}_i) y_i\}^2 J_{N,i} J_{N,i}^{\mathsf{T}} \tag{52}$$

where

$$\frac{d \log \{p(y_1, .., y_k | \hat{y}_1, ..., \hat{y}_k)\}}{d\boldsymbol{w}_i} = \frac{dz_i}{d\boldsymbol{w}_i} \frac{d\hat{y}_i}{dz_i} \frac{d \sum_{i=1}^{k} y_i \log(\hat{y}_i)}{d\hat{y}_i} \tag{53}$$

$$= J_{N,i} \left\{ (\hat{y}_i - \hat{y}_i^2) \frac{y_i}{\hat{y}_i} \right\} = J_{N,i} (1 - \hat{y}_i) y_i \tag{54}$$

Experiment: Settings

Data

- **Regression:** Synthetic data, 1000 features.
- **Classification:** MNIST (hand written digits), 764 features, $1 \sim 9$ labels.

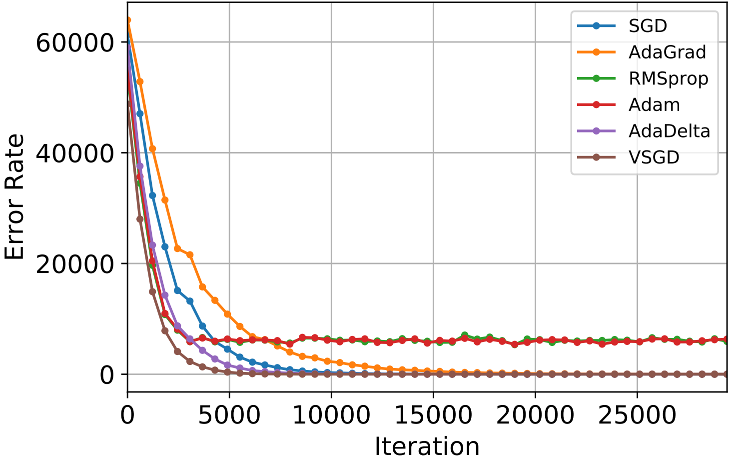
Hyperparameters

- **Grid Search:** Employ the best performed hyperparameters.
  1. Learning rate.
  2. Weight of RMS.

Evaluation

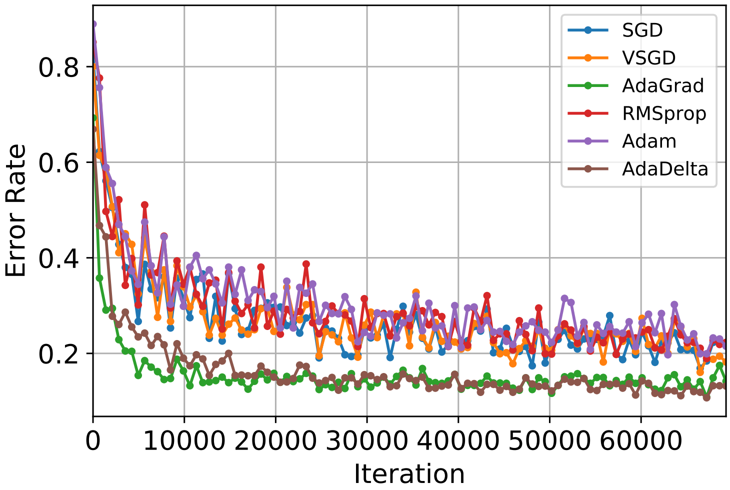
- **Regression:** Mean square error.
- **Classification:** Misclassification rate.

Experiment: Regression



- **VSGD is the best even though tuning free.**

Experiment: Classification



- **AdaDelta is the best even though tuning free.**

Conclusion & Future Work

Conclusion

- **Summarize stochastic optimizers from the view of its metric.**
  - Quasi-Newton type, RMS type and Natural Gradient type.
- **Conduct brief experiments (classification and regression).**
  - See the efficacies of tuning free algorithm.

Antoine Bordes, Léon Bottou, and Patrick Gallinari,
"Sgd-qn: Careful quasi-newton stochastic gradient descent,"
Journal of Machine Learning Research, vol. 10, no. Jul, pp. 1737–1754, 2009.

Matthew D Zeiler,
"Adadelta: an adaptive learning rate method,"
arXiv preprint arXiv:1212.5701, 2012.

Tom Schaul, Sixin Zhang, and Yann LeCun,
"No more pesky learning rates,"
in *International Conference on Machine Learning*, 2013, pp. 343–351.

Tom Schaul and Yann LeCun,
"Adaptive learning rates and parallelization for stochastic, sparse, non-smooth gradients,"
in *International Conference on Learning Representations*, Scottsdale, AZ, 2013.

Oriol Vinyals and Daniel Povey,
"Krylov subspace descent for deep learning,"
in *Artificial Intelligence and Statistics*, 2012, pp. 1261–1268.

- Nicol N Schraudolph,
  "Fast curvature matrix-vector products for second-order gradient descent,"
  Neural computation, vol. 14, no. 7, pp. 1723–1738, 2002.

- James Martens,
  "Deep learning via hessian-free optimization,"
  in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 735–742.

- Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer,
  "A stochastic quasi-newton method for large-scale optimization,"
  SIAM Journal on Optimization, vol. 26, no. 2, pp. 1008–1031, 2016.

- Nicol N Schraudolph, Jin Yu, and Simon Günter,
  "A stochastic quasi-newton method for online convex optimization,"
  in *Artificial Intelligence and Statistics*, 2007, pp. 436–443.

- Aryan Mokhtari and Alejandro Ribeiro,
  "Res: Regularized stochastic bfgs algorithm,"
  IEEE Transactions on Signal Processing, vol. 62, no. 23, pp. 6089–6104, 2014.

Shun-Ichi Amari,
"Natural gradient works efficiently in learning,"
Neural computation, vol. 10, no. 2, pp. 251–276, 1998.

Nicolas L Roux, Pierre-Antoine Manzagol, and Yoshua Bengio,
"Topmoumoute online natural gradient algorithm,"
in *Advances in neural information processing systems*, 2008, pp. 849–856.

Nicolas L Roux and Andrew W Fitzgibbon,
"A fast natural newton method,"
in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 623–630.

John Duchi, Elad Hazan, and Yoram Singer,
"Adaptive subgradient methods for online learning and stochastic optimization,"
Journal of Machine Learning Research, vol. 12, no. Jul, pp. 2121–2159, 2011.

Tijmen Tieleman and Geoffrey Hinton,
"Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,"
COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26–31, 2012.

Diederik Kingma and Jimmy Ba,
"Adam: A method for stochastic optimization,"
in *International Conference on Learning and Representations*, 2015, pp. 1–13.