Relative Fisher Information and Natural Gradient for Learning Large Modular Models

Ke Sun [1] Frank Nielsen [2,3]

[1]King Abdullah University of Science & Technology (KAUST)

[2]École Polytechnique

[3]Sony CSL

ICML 2017

Fisher Information Metric (FIM)

Consider a statistical model $p(\boldsymbol{x} \mid \boldsymbol{\Theta})$ of order D. The FIM (Hotelling29,Rao45) $\mathcal{I}(\boldsymbol{\Theta}) = (\mathcal{I}_{ij})$ is defined by a $D \times D$ positive semi-definite matrix

$$\mathcal{I}_{ij} = E_p \left[\frac{\partial l}{\partial \Theta_i} \frac{\partial l}{\partial \Theta_j} \right], \tag{1}$$

where $l(\boldsymbol{\Theta}) = \log p(\boldsymbol{x} \mid \boldsymbol{\Theta})$ denotes the log-likelihood.

Equivalent Expressions

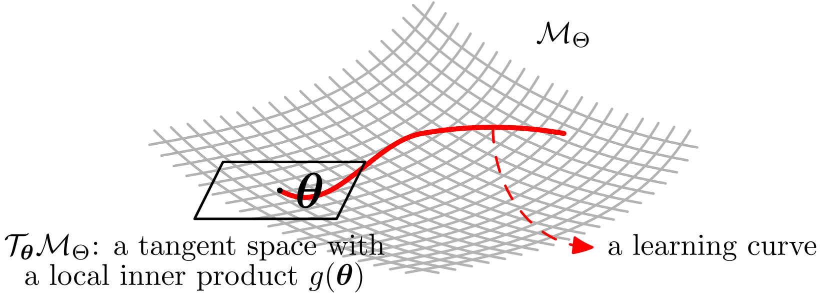
$$
\begin{aligned}
\mathcal{I}_{ij} &= E_p\left[\frac{\partial l}{\partial \Theta_i}\frac{\partial l}{\partial \Theta_j}\right] \\
&= -E_p\left[\frac{\partial^2 l}{\partial \Theta_i \partial \Theta_j}\right] \\
&= 4\int \frac{\partial\sqrt{p(\boldsymbol{x}\mid\boldsymbol{\Theta})}}{\partial \Theta_i}\frac{\partial\sqrt{p(\boldsymbol{x}\mid\boldsymbol{\Theta})}}{\partial \Theta_j}\,d\boldsymbol{x}.
\end{aligned}
$$

Observed FIM (Efron & Hinkley, 1978) With respect to $X_n = \{\boldsymbol{x}_k\}_{k=1}^{n}$,

$$
\hat{\mathcal{I}} = -\nabla^2 l(\boldsymbol{\Theta}\mid X_n) = -\sum_{i=1}^{n}\frac{\partial^2 \log p(\boldsymbol{x}_i\mid\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}\partial\boldsymbol{\Theta}^{\intercal}}.
$$

FIM and Statistical Learning

- Any parametric learning is inside a corresponding parameter manifold $\mathcal{M}_\Theta$



- FIM gives an invariant Riemannian metric $g(\boldsymbol{\Theta}) = \mathcal{I}(\boldsymbol{\Theta})$ for any loss function based on standard f-divergence (KL, cross-entropy, ...)

S. Amari. Information Geometry and Its Applications. 2016.

Invariance

The FIM is *not* invariant and depends on the parameterization:

$$g_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \boldsymbol{J}^\intercal g_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})\boldsymbol{J}$$

where $\boldsymbol{J}$ is the Jacobian matrix $J_{ij} = \frac{\partial \Lambda_i}{\partial \Theta_j}$.

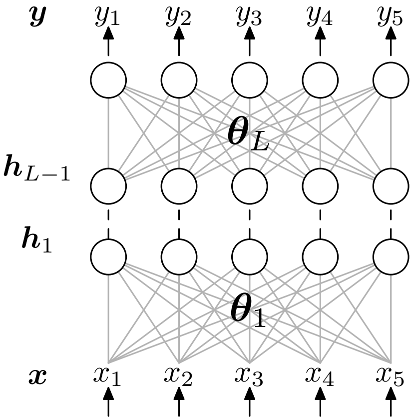
However its measurements such as $\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{g(\boldsymbol{\Theta})}$ is invariant:

$$\begin{aligned}
\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{g(\boldsymbol{\Theta})} &= \delta\boldsymbol{\Theta}^\intercal g(\boldsymbol{\Theta})\delta\boldsymbol{\Theta} \\
&= \delta\boldsymbol{\Theta}^\intercal \boldsymbol{J}^\intercal g_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})\boldsymbol{J}\delta\boldsymbol{\Theta} \\
&= \delta\boldsymbol{\Lambda}^\intercal g_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda})\delta\boldsymbol{\Lambda} \\
&= \langle \delta\boldsymbol{\Lambda}, \delta\boldsymbol{\Lambda} \rangle_{g(\boldsymbol{\Lambda})}.
\end{aligned}$$

Regardless of the choice of the coordinate system, it is essentially the same metric!

Statistical Formulation of a Multilayer Perceptron (MLP)

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \Theta) = \sum_{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{L-1}} p(\boldsymbol{y} \mid \boldsymbol{h}_{L-1}, \boldsymbol{\theta}_L) \cdots p(\boldsymbol{h}_2 \mid \boldsymbol{h}_1, \boldsymbol{\theta}_2) p(\boldsymbol{h}_1 \mid \boldsymbol{x}, \boldsymbol{\theta}_1),$$



The FIM of a MLP

The FIM of a MLP has the following expression

$$g(\boldsymbol{\Theta}) = E_{\boldsymbol{x}\sim\hat{p}(X_n),\,\boldsymbol{y}\sim p(\boldsymbol{y}\,|\,\boldsymbol{x},\boldsymbol{\Theta})}\left[\frac{\partial l}{\partial \boldsymbol{\Theta}}\frac{\partial l}{\partial \boldsymbol{\Theta}^\intercal}\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n} E_{p(\boldsymbol{y}\,|\,\boldsymbol{x}_i,\boldsymbol{\Theta})}\left[\frac{\partial l_i}{\partial \boldsymbol{\Theta}}\frac{\partial l_i}{\partial \boldsymbol{\Theta}^\intercal}\right]$$

where

- $\hat{p}(X_n)$ is the empirical distribution of the samples $X_n = \{\boldsymbol{x}_i\}_{i=1}^n$
- $l_i(\boldsymbol{\Theta}) = \log p(\boldsymbol{y}\,|\,\boldsymbol{x}_i,\,\boldsymbol{\Theta})$ is the conditional log-likelihood

Meaning of the FIM of a MLP

Consider a learning step on $\mathcal{M}_{\boldsymbol{\Theta}}$ from $\boldsymbol{\Theta}$ to $\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}$. The step size

$$\begin{aligned}
\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{g(\boldsymbol{\Theta})} &= \delta\boldsymbol{\Theta}^\intercal g(\boldsymbol{\Theta}) \delta\boldsymbol{\Theta} \\
&= \delta\boldsymbol{\Theta}^\intercal \left\{ \frac{1}{n} \sum_{i=1}^{n} E_{p(\boldsymbol{y} \,|\, \boldsymbol{x}_i, \boldsymbol{\Theta})} \left[\frac{\partial l_i}{\partial \boldsymbol{\Theta}} \frac{\partial l_i}{\partial \boldsymbol{\Theta}^\intercal} \right] \right\} \delta\boldsymbol{\Theta} \\
&= \frac{1}{n} \sum_{i=1}^{n} E_{p(\boldsymbol{y} \,|\, \boldsymbol{x}_i, \boldsymbol{\Theta})} \left[\delta\boldsymbol{\Theta}^\intercal \frac{\partial l_i}{\partial \boldsymbol{\Theta}} \right]^2
\end{aligned}$$

measures how much $\delta\boldsymbol{\Theta}$ is statistically along $\frac{\partial l}{\partial \boldsymbol{\Theta}}$.

Will $\delta\Theta$ make a significant change to the mapping $\boldsymbol{x} \to \boldsymbol{y}$ or not?

Natural Gradient: Seeking a Short Path

Consider $\min_{\mathbf{\Theta}\in\mathcal{M}_{\mathbf{\Theta}}} L(\mathbf{\Theta})$. At $\mathbf{\Theta}_t \in \mathcal{M}_{\mathbf{\Theta}}$, the target is to minimize wrt $\delta\mathbf{\Theta}$

$$\underbrace{L(\mathbf{\Theta}_t + \delta\mathbf{\Theta})}_{\text{Loss function}} + \frac{1}{2\gamma}\underbrace{\langle\delta\mathbf{\Theta}, \delta\mathbf{\Theta}\rangle_{g(\mathbf{\Theta}_t)}}_{\text{Squared step size}} \quad (\gamma\text{: learning rate})$$

$$\approx L(\mathbf{\Theta}_t) + \delta\mathbf{\Theta}^\intercal \bigtriangledown L(\mathbf{\Theta}_t) + \frac{1}{2\gamma}\delta\mathbf{\Theta}^\intercal g(\mathbf{\Theta}_t)\delta\mathbf{\Theta},$$

giving a learning step

$$\delta\mathbf{\Theta}_t = -\gamma\underbrace{g^{-1}(\mathbf{\Theta}_t)\bigtriangledown L(\mathbf{\Theta}_t)}_{\text{natural gradient}}$$

- Equivalence with mirror descent (Raskutti & Mukherjee 2013)

Natural Gradient: Intrinsics

$$\delta\mathbf{\Theta}_t = -\gamma g^{-1}(\mathbf{\Theta}_t) \bigtriangledown L(\mathbf{\Theta}_t)$$

This Riemannnian metric is a property of the parameter space that is independent of the loss function $L(\mathbf{\Theta})$.

The good performance of natural gradient relies on that $L(\mathbf{\Theta})$ is similarly curved as $\log p(\boldsymbol{x} \mid \mathbf{\Theta})$ ($\boldsymbol{x} \sim p(\boldsymbol{x} \mid \mathbf{\Theta})$).

Natural gradient is not universally good for any loss functions.

Natural Gradient: Pros and Cons

Pros

- Invariant (intrinsic) gradient
- Not trapped in plateaus
- Achieve Fisher efficiency in online learning

Cons

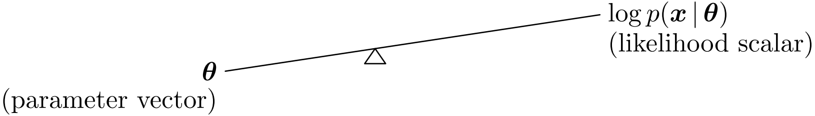
- Too expensive to compute (no closed-form FIM; need matrix inversion)

Relative FIM — Informal Ideas

- Decompose the learning system into subsystems
- The subsystems are interfaced with each other through hidden variables $\boldsymbol{h}_i$
- Some subsystems are interfaced with the I/O environment through $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$
- Compute the subsystem FIM by **integrating out its interface variables** $\boldsymbol{h}_i$, so that the intrinsics of this subsystem can be discussed regardless of the remaining parts

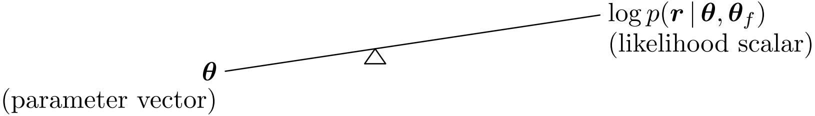
From FIM to Relative FIM (RFIM)

FIM



How sensitive is $\boldsymbol{x}$ wrt tiny movements of $\boldsymbol{\theta}$ on $\mathcal{M}_{\boldsymbol{\theta}}$?

RFIM



Given $\boldsymbol{\theta}_f$, how sensitive is $\boldsymbol{r}$ wrt tiny movements of $\boldsymbol{\theta}$?

Relative FIM — Definition

Given $\boldsymbol{\theta}_f$ (the **reference**), the Relative Fisher Information Metric (RFIM) of $\boldsymbol{\theta}$ wrt $\boldsymbol{h}$ (the **response**) is

$$g^{\boldsymbol{h}}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_f) = E_{p(\boldsymbol{h} \mid \boldsymbol{\theta}, \boldsymbol{\theta}_f)} \left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\boldsymbol{h} \mid \boldsymbol{\theta},\, \boldsymbol{\theta}_f) \frac{\partial}{\partial \boldsymbol{\theta}^\intercal} \ln p(\boldsymbol{h} \mid \boldsymbol{\theta},\, \boldsymbol{\theta}_f) \right],$$

or simply $g^{\boldsymbol{h}}(\boldsymbol{\theta})$.

Meaning: given $\boldsymbol{\theta}_f$, how variations of $\boldsymbol{\theta}$ will affect the response $\boldsymbol{h}$.

Different Subsystems – Simple Examples

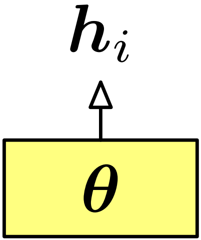


Figure: Generator

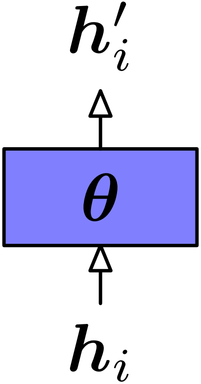
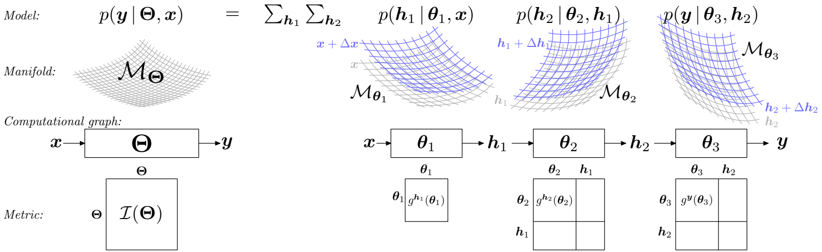


Figure: Discriminator or Regressor

A Dynamic Geometry



- As the interface hidden variables $\boldsymbol{h}_i$ are changing, the subsystem geometry is not absolute but is **relative** to its reference variables provided by adjacent subsystems

RFIM of One `tanh` Neuron

Consider a neuron with input $\boldsymbol{x}$, weights $\boldsymbol{w}$, a hyperbolic tangent activation function, and a stochastic output $y \in \{-1, 1\}$, given by

$$p(y=1) = \frac{1 + \mathtt{tanh}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}{2}, \quad \mathtt{tanh}(t) = \frac{\exp(t) - \exp(-t)}{\exp(t) + \exp(-t)}.$$

$\tilde{\boldsymbol{x}} = (\boldsymbol{x}^\intercal, 1)^\intercal$ denotes the augmented vector of $\boldsymbol{x}$

$$g^y(\boldsymbol{w} \mid \boldsymbol{x}) = \nu_{\mathtt{tanh}}(\boldsymbol{w}, \boldsymbol{x}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal, \quad \nu_{\mathtt{tanh}}(\boldsymbol{w}, \boldsymbol{x}) = \mathrm{sech}^2(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}).$$

RFIM of Parametric Rectified Linear Unit

$$p(y \,|\, \boldsymbol{w}, \boldsymbol{x}) = G(y \,|\, \texttt{relu}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}), \sigma^2), \quad (G \text{ is for Gaussian})$$

$$\texttt{relu}(t) = \begin{cases} t & \text{if } t \geq 0 \\ \iota t & \text{if } t < 0. \end{cases} \quad (0 \leq \iota < 1)$$

By certain assumptions,

$$g^y(\boldsymbol{w} \,|\, \boldsymbol{x}) = \nu_{\texttt{relu}}(\boldsymbol{w}, \boldsymbol{x}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal,$$

$$\nu_{\texttt{relu}}(\boldsymbol{w}, \boldsymbol{x}) = \frac{1}{\sigma^2} \left[\iota + (1 - \iota) \underbrace{\texttt{sigm}}_{\textit{sigmoid}} \left(\frac{1 - \iota}{\omega} \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \right) \right]^2 .$$

Set $\sigma = 1$, $\iota = 0$, it simplifies to

$$\nu_{\texttt{relu}}(\boldsymbol{w}, \boldsymbol{x}) = \texttt{sigm}^2 \left(\frac{1}{\omega} \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \right) .$$

Generic Expression of One-neuron RFIMs

Denote $f \in \{\texttt{tanh}, \texttt{sigm}, \texttt{relu}, \texttt{elu}\}$ to be an element-wise nonlinear activation function. The RFIM is

$$g^y(\boldsymbol{w} \mid \boldsymbol{x}) = \nu_f(\boldsymbol{w}, \boldsymbol{x})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal,$$

where $\nu_f(\boldsymbol{w}, \boldsymbol{x})$ is a positive coefficient with large values in the *linear region*, or the effective learning zone of the neuron.

RFIM of a Linear Layer

$\boldsymbol{x}$: input; $\boldsymbol{W}$: connection weights; $\boldsymbol{y}$: stochastic output following

$$p(\boldsymbol{y} \mid \boldsymbol{W}, \boldsymbol{x}) = G(\boldsymbol{y} \mid \boldsymbol{W}^\intercal \tilde{\boldsymbol{x}}, \sigma^2 \boldsymbol{I}).$$

We vectorize $\boldsymbol{W}$ by stacking its columns $\{\boldsymbol{w}_i\}$. Then

$$g^{\boldsymbol{y}}(\boldsymbol{W} \mid \boldsymbol{x}) = \frac{1}{\sigma^2} \begin{bmatrix} \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & & \\ & \ddots & \\ & & \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \end{bmatrix}.$$

RFIM of a Non-linear Layer

A *nonlinear* layer applies an element-wise activation on $\boldsymbol{W}^\intercal\tilde{\boldsymbol{x}}$. We have

$$g^{\boldsymbol{y}}\left(\boldsymbol{W} \mid \boldsymbol{x}\right) = \begin{bmatrix} \nu_f(\boldsymbol{w}_1, \boldsymbol{x})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & & \\ & \ddots & \\ & & \nu_f(\boldsymbol{w}_m, \boldsymbol{x})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \end{bmatrix},$$

where $\nu_f(\boldsymbol{w}_i, \boldsymbol{x})$ depends on the activation function f.

The RFIMs of single neuron models, a linear layer, a non-linear layer, a soft-max layer, two consecutive layers all have **simple closed form solutions**[1].

[1] See the paper.

List of RFIMs

Subsystem	the RFIM $g^{y}(\boldsymbol{w})$
A tanh neuron	$\mathtt{sech}^2(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A sigm neuron	$\mathtt{sigm}(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\left[1-\mathtt{sigm}(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\right]\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A relu neuron	$\left[\iota+(1-\iota)\mathtt{sigm}\left(\frac{1-\iota}{\omega}\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}\right)\right]^2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A elu neuron	$\begin{cases}\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}\geq 0\\ (\alpha\exp(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}))^2\,\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}<0\end{cases}$
A linear layer	$\mathtt{diag}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal,\cdots,\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right]$
A non-linear layer	$\mathtt{diag}\left[\nu_f(\boldsymbol{w}_1,\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal,\cdots,\nu_f(\boldsymbol{w}_m,\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right]$
A soft-max layer	$\begin{bmatrix}(\eta_1-\eta_1^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & -\eta_1\eta_2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_1\eta_m\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\\ -\eta_2\eta_1\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & (\eta_2-\eta_2^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_2\eta_m\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\\ \vdots & \vdots & \ddots & \vdots\\ -\eta_m\eta_1\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & -\eta_m\eta_2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & (\eta_m-\eta_m^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\end{bmatrix}.$
Two layers	see the paper.

Relative Natural Gradient Descent (RNGD)

For each subsystem,

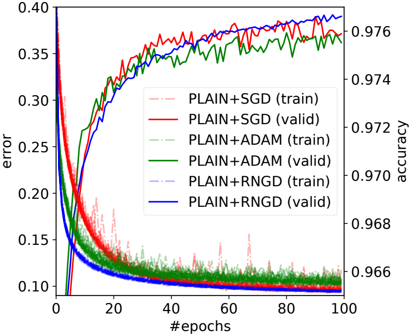
$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \gamma \cdot \underbrace{\left(\bar{g}^{\boldsymbol{h}}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_f)\right)^{-1}}_{\text{inverse RFIM}} \cdot \left.\frac{\partial L}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$$

where

$$\bar{g}^{\boldsymbol{h}}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_f) = \frac{1}{n}\sum_{i=1}^{n} g^{\boldsymbol{h}}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_f^i).$$

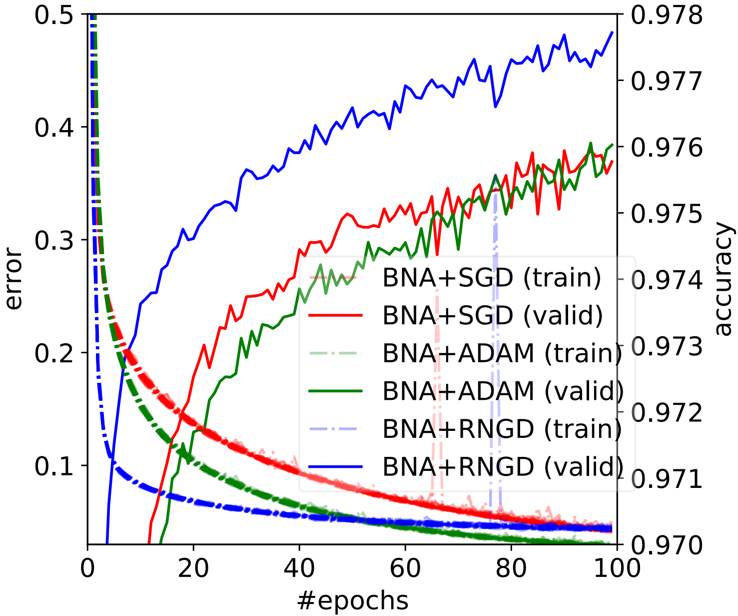
By definition, RFIM is a function of the reference variables.
$\bar{g}^{\boldsymbol{h}}(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_f)$ is its expectation wrt an empirical distribution of $\boldsymbol{\theta}_f$.

Proof-of-concept

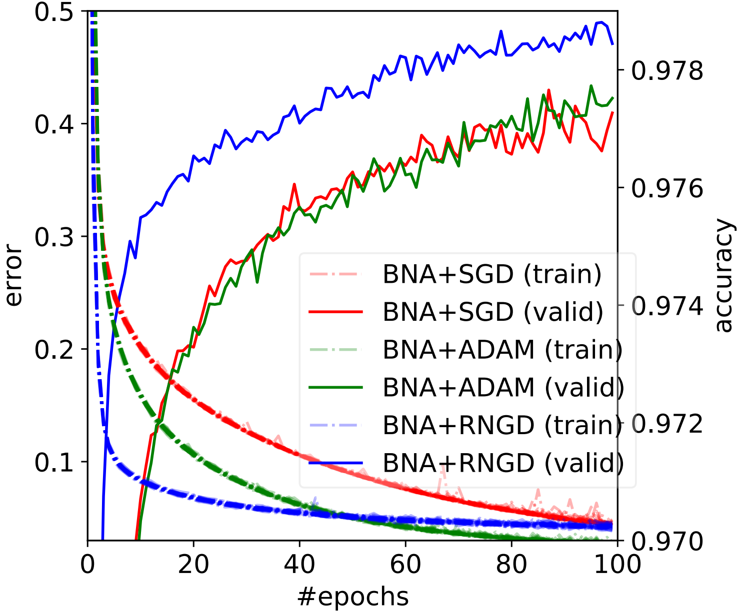


- MLP with shape 784-80-80-80-10
- `relu` activation
- Mini batch size 50
- Recompute the inverse RFIM every 100 mini batchs
- L_2 regularization

BNA: batch normalization (BN) after activation



Change the MLP shape to 784-100-100-100-10



Novel Viewpoint

Learning is a process where a set of collaborative learners move on their sub-manifolds, and the geometries of these sub-manifolds are also evolving with the system.

- Well-suited to parallel computation and distributed learning

Conclusion

- FIM is just a special case of RFIM, where the subsystem is the whole system
- By looking at smaller subsystems, RFIM can have simpler closed-form expressions
- RNGD can be implemented without approximation
- This has the potential to improve learning of large neural networks

codes, updates:

https://www.lix.polytechnique.fr/~nielsen/RFIM/

Thank you!