Relative Fisher Information and Natural Gradient for Learning Large Modular Models

Ke Sun[‡] Frank Nielsen[†⋆]

[‡] King Abdullah University of Science & Technology (KAUST) [†]École polytechnique [⋆]Sony CSL

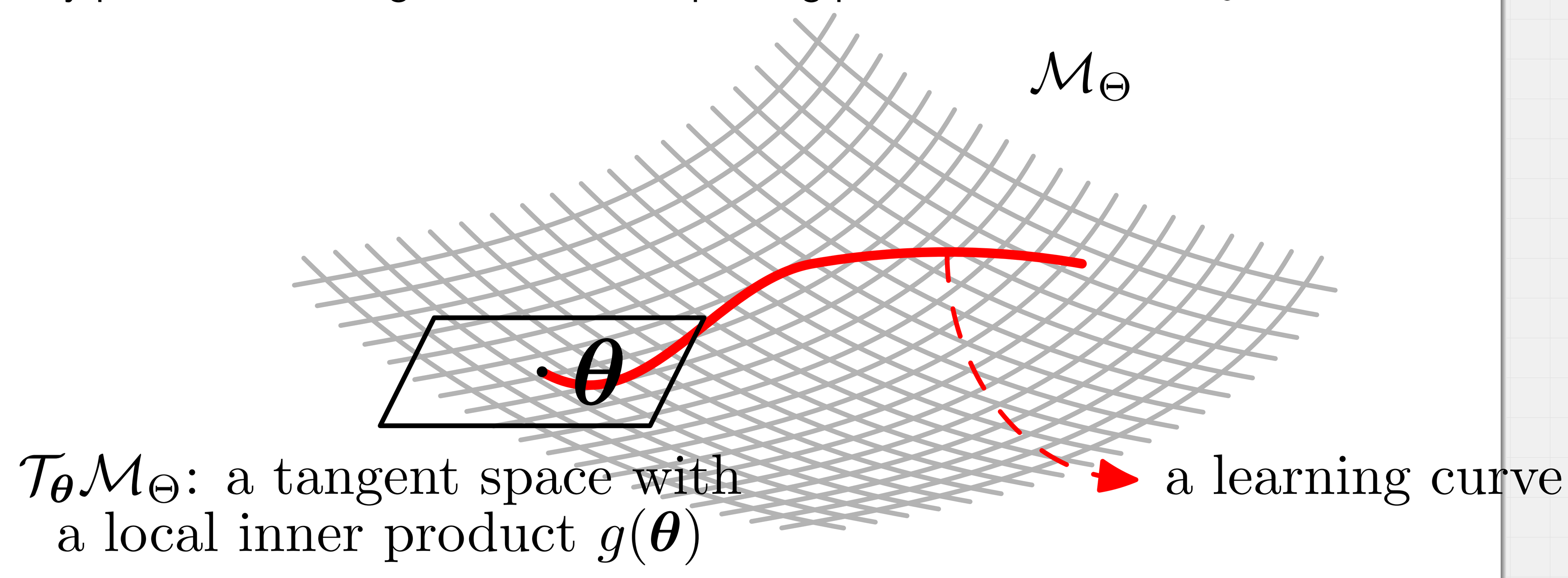
Fisher Information Metric (FIM)

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

$$\mathcal{I}_{ij} = E_p\left[\frac{\partial l}{\partial \Theta_i}\frac{\partial l}{\partial \Theta_j}\right] = -E_p\left[\frac{\partial^2 l}{\partial \Theta_i \partial \Theta_j}\right] = 4\int \frac{\partial\sqrt{p(x \mid \Theta)}}{\partial \Theta_i}\frac{\partial\sqrt{p(x \mid \Theta)}}{\partial \Theta_j}\mathrm{d}x, \qquad (1)$$

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

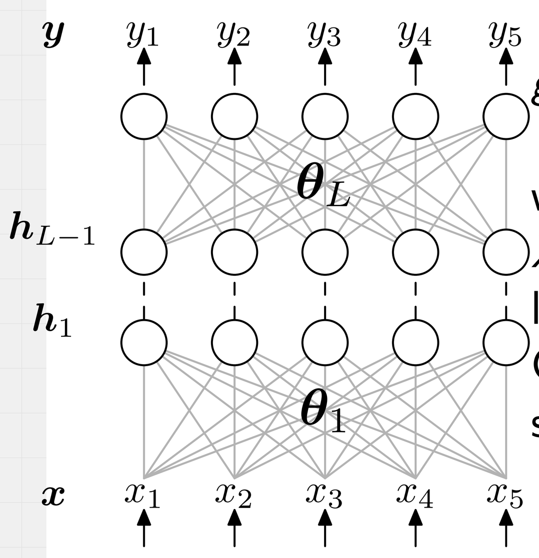
- Any parametric learning is inside a corresponding parameter manifold $\mathcal{M}_\Theta$



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

FIM of a Multilayer Perceptron (MLP)

$$p(y \mid x, \Theta) = \sum_{h_1,\cdots,h_{L-1}} p(y \mid h_{L-1}, \theta_L)\cdots p(h_2 \mid h_1, \theta_2)p(h_1 \mid x, \theta_1),$$



The FIM of a MLP has the following expression

$$g(\Theta) = E_{x\sim\hat{p}(X_n),\, y\sim p(y \mid x, \Theta)}\left[\frac{\partial l}{\partial \Theta}\frac{\partial l}{\partial \Theta^\intercal}\right] = \frac{1}{n}\sum_{i=1}^{n} E_{p(y \mid x_i, \Theta)}\left[\frac{\partial l_i}{\partial \Theta}\frac{\partial l_i}{\partial \Theta^\intercal}\right],$$

where $\hat{p}(X_n)$ is the empirical distribution of the samples $X_n = \{x_i\}_{i=1}^n$, and $l_i(\Theta) = \log p(y \mid x_i, \Theta)$ is the conditional log-likelihood.

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

$$\langle \delta\Theta, \delta\Theta \rangle_{g(\Theta)} = \delta\Theta^\intercal g(\Theta)\delta\Theta = \frac{1}{n}\sum_{i=1}^{n} E_{p(y \mid x_i, \Theta)}\left[\delta\Theta^\intercal \frac{\partial l_i}{\partial \Theta}\right]^2$$

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

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

Natural Gradient

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

$$\underbrace{L(\Theta_t + \delta\Theta)}_{\text{Loss function}} + \frac{1}{2\gamma}\underbrace{\langle \delta\Theta, \delta\Theta \rangle_{g(\Theta_t)}}_{\text{Squared step size}} \approx L(\Theta_t) + \delta\Theta^\intercal \bigtriangledown L(\Theta_t) + \frac{1}{2\gamma}\delta\Theta^\intercal g(\Theta_t)\delta\Theta,$$

giving a learning step

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

- Equivalence with mirror descent **(Raskutti & Mukherjee 2013)**

Pros

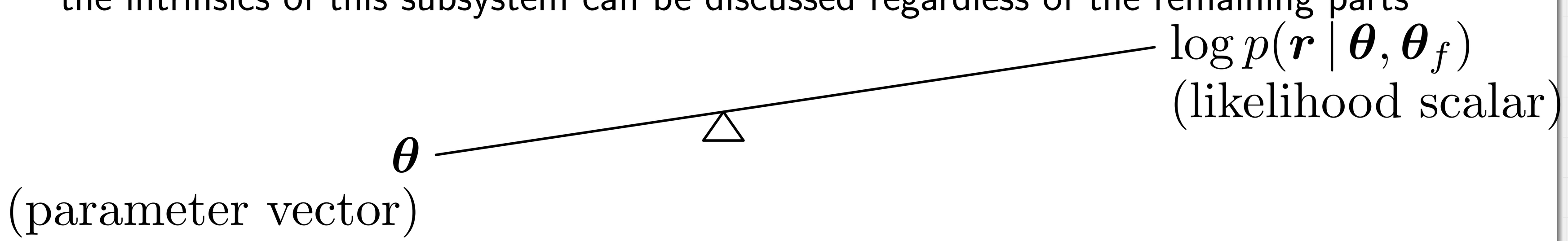
- Invariant (intrinsic) gradient
- Not trapped in plateaus
- Achieve Fisher efficiency in online learning

Cons

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

Relative FIM (RFIM) — Informal Ideas

- Decompose the learning system into subsystems
- The subsystems are interfaced with each other through hidden variables h_i
- Some subsystems are interfaced with the I/O environment through x_i and y_i
- Compute the subsystem FIM by **integrating out its interface variables** h_i, so that the intrinsics of this subsystem can be discussed regardless of the remaining parts



Given θ_f, how sensitive is r wrt tiny movements of θ?

RFIM – Formal Definition

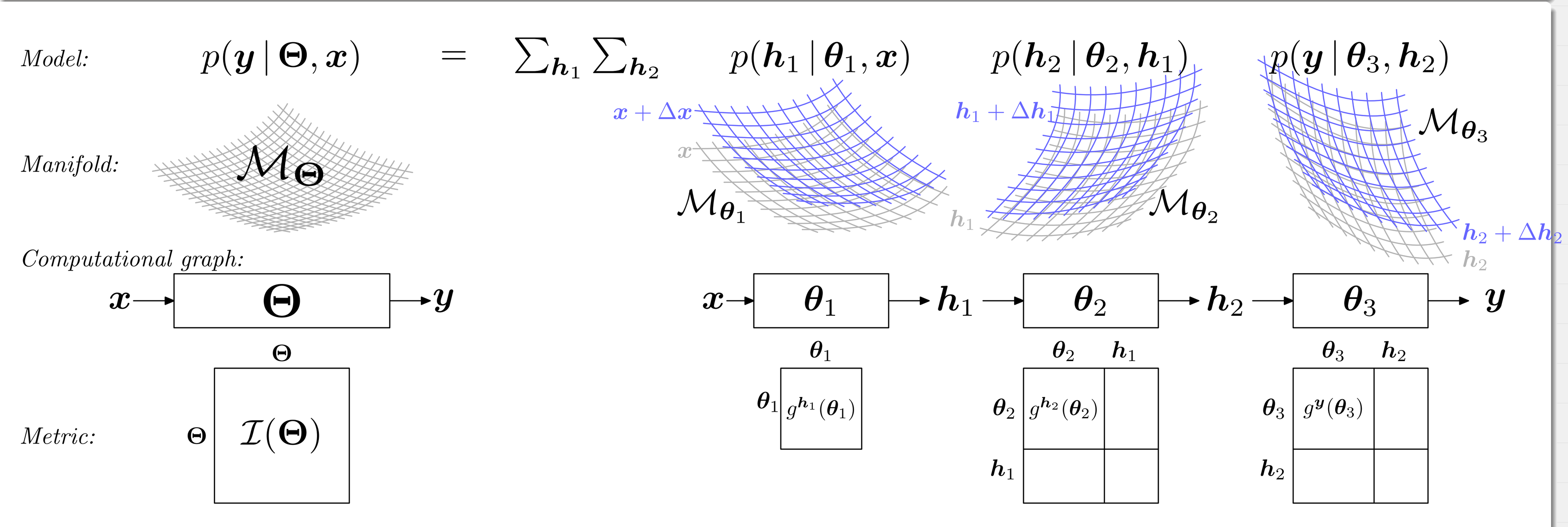
Given θ_f (the **reference**), the Relative Fisher Information Metric (RFIM) of θ wrt h (the **response**) is

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

or simply $g^h(\theta)$.

- RFIM includes FIM as a special case.
- RFIM is dynamic wrt the reference θ_f

A Dynamic Geometry



- As the interface hidden variables 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 x, weights w, a hyperbolic tangent activation function, and a stochastic output $y \in \{-1, 1\}$, given by

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

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

$$g^y(w \mid x) = \nu_{\mathtt{tanh}}(w, x)\tilde{x}\tilde{x}^\intercal, \quad \nu_{\mathtt{tanh}}(w, x) = \mathtt{sech}^2(w^\intercal \tilde{x}) = 1 - \mathtt{tanh}^2(w^\intercal \tilde{x}).$$

Meaning: The RFIM has a large magnitude on the "learning zone" of the neuron.

A List of RFIMs

Subsystem	the RFIM $g^y(w)$
A `tanh` neuron	$\mathtt{sech}^2(w^\intercal\tilde{x})\tilde{x}\tilde{x}^\intercal$
A `sigm` neuron	$\mathtt{sigm}(w^\intercal\tilde{x})\left[1 - \mathtt{sigm}(w^\intercal\tilde{x})\right]\tilde{x}\tilde{x}^\intercal$
A `relu` neuron	$\left[\iota + (1-\iota)\mathtt{sigm}\left(\frac{1-\iota}{\omega}w^\intercal\tilde{x}\right)\right]^2\tilde{x}\tilde{x}^\intercal$
A `elu` neuron	$\begin{cases}\tilde{x}\tilde{x}^\intercal & \text{if } w^\intercal\tilde{x} \ge 0\\ \left(\alpha\exp(w^\intercal\tilde{x})\right)^2\tilde{x}\tilde{x}^\intercal & \text{if } w^\intercal\tilde{x} < 0\end{cases}$
A linear layer	$\mathtt{diag}\left[\tilde{x}\tilde{x}^\intercal, \cdots, \tilde{x}\tilde{x}^\intercal\right]$
A non-linear layer	$\mathtt{diag}\left[\nu_f(w_1, \tilde{x})\tilde{x}\tilde{x}^\intercal, \cdots, \nu_f(w_m, \tilde{x})\tilde{x}\tilde{x}^\intercal\right]$
A soft-max layer	$\begin{bmatrix}(\eta_1 - \eta_1^2)\tilde{x}\tilde{x}^\intercal & -\eta_1\eta_2\tilde{x}\tilde{x}^\intercal & \cdots & -\eta_1\eta_m\tilde{x}\tilde{x}^\intercal\\ -\eta_2\eta_1\tilde{x}\tilde{x}^\intercal & (\eta_2 - \eta_2^2)\tilde{x}\tilde{x}^\intercal & \cdots & -\eta_2\eta_m\tilde{x}\tilde{x}^\intercal\\ \vdots & \vdots & \ddots & \vdots\\ -\eta_m\eta_1\tilde{x}\tilde{x}^\intercal & -\eta_m\eta_2\tilde{x}\tilde{x}^\intercal & \cdots & (\eta_m - \eta_m^2)\tilde{x}\tilde{x}^\intercal\end{bmatrix}$.
Two layers	see the paper.

Relative Natural Gradient Descent (RNGD)

For each subsystem,

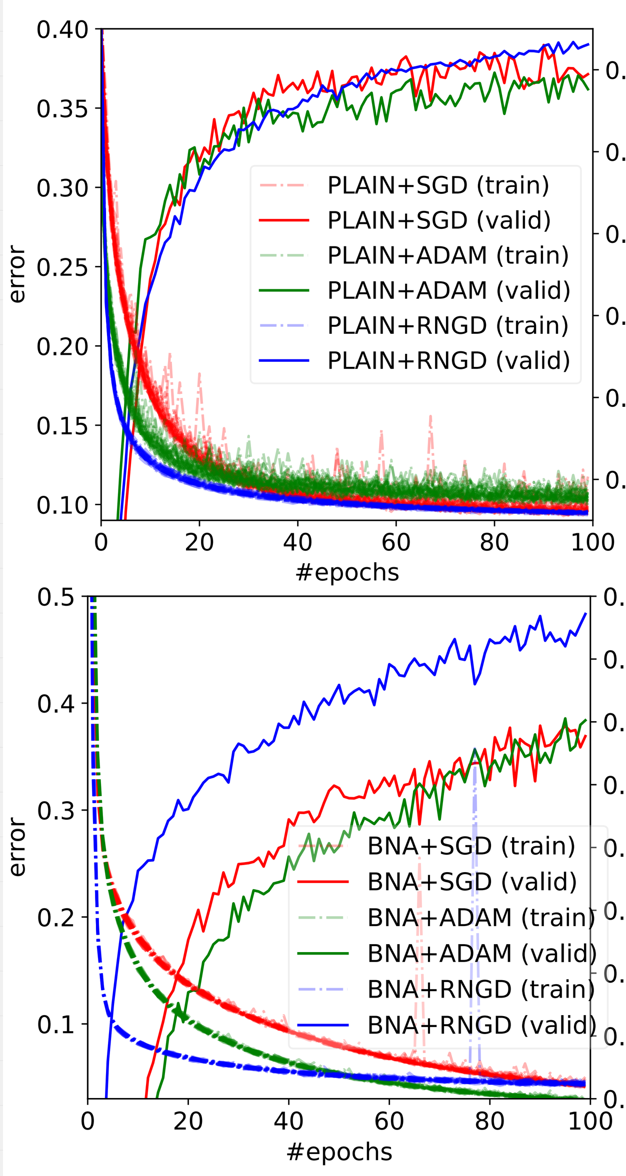
$$\theta_{t+1} \leftarrow \theta_t - \gamma \cdot \underbrace{\left(\bar{g}^h(\theta_t \mid \theta_f)\right)^{-1}}_{\text{inverse RFIM}} \cdot \left.\frac{\partial L}{\partial \theta}\right|_{\theta=\theta_t}$$

where

$$\bar{g}^h(\theta_t \mid \theta_f) = \frac{1}{n}\sum_{i=1}^{n} g^h(\theta_t \mid \theta_f^i).$$

By definition, RFIM is a function of the reference variables. $\bar{g}^h(\theta_t \mid \theta_f)$ is its expectation wrt an empirical distribution of θ_f.

A Proof-of-concept



- MLP with shape 784-80-80-80-10
- `relu` activation
- Mini-batch size 50
- No dropout
- L_2 regularization
- Maintain an exponential moving average of the RFIM
- Recompute the inverse RFIM every 100 mini-batchs
- PLAIN: a plain MLP
- BNA: a MLP with batch normalization

Observations

- RNGD achieved sharper learning curve in terms of # iterations
- The computation cost of each epoch is several times more expensive
- RNGD can give better local optima

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
- Unlike NGD, RNGD can be implemented without approximation
- RFIM provides an accurate terminology to support feature whitening, natural neural networks, etc.

Contact: sunk@ieee.org nielsen@lix.polytechnique.fr Codes: https://www.lix.polytechnique.fr/~nielsen/RFIM/