# Relative Fisher Information and Natural Gradient for Learning Large Modular Models

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

## Abstract

Fisher information and natural gradient provided deep insights and powerful tools to artificial neural networks. However related analysis becomes more and more difficult as the learner's structure turns large and complex. This paper makes a preliminary step towards a new direction. We extract a local component from a large neural system, and define its relative Fisher information metric that describes accurately this small component, and is invariant to the other parts of the system. This concept is important because the geometry structure is much simplified and it can be easily applied to guide the learning of neural networks. We provide an analysis on a list of commonly used components, and demonstrate how to use this concept to further improve optimization.

## 1. Fisher Information Metric

The Fisher Information Metric (FIM) $\mathcal{I}(\boldsymbol{\Theta}) = (\mathcal{I}_{ij})$ of a statistical parametric model $p(\boldsymbol{x} \,|\, \boldsymbol{\Theta})$ of order $D$ is defined by a $D \times D$ positive semidefinite (psd) matrix ($\mathcal{I}(\boldsymbol{\Theta}) \succeq 0$) with coefficients $\mathcal{I}_{ij} = E_p \left[ \frac{\partial l}{\partial \Theta_i} \frac{\partial l}{\partial \Theta_j} \right]$, where $l(\boldsymbol{\Theta})$ denotes the log-density function $\log p(\boldsymbol{x} \,|\, \boldsymbol{\Theta})$. Under light regularity conditions, FIM can be rewritten equivalently as

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

As its empirical counterpart, the observed FIM (Efron & Hinkley, 1978) with respect to (wrt) a sample set $X_n = \{\boldsymbol{x}_k\}_{k=1}^n$ is $\hat{\mathcal{I}}(\boldsymbol{\Theta} \,|\, X_n) = -\nabla^2 l(\boldsymbol{\Theta} \,|\, X_n)$, which is often evaluated at the maximum likelihood estimate $\boldsymbol{\Theta} = \hat{\boldsymbol{\Theta}}(X_n)$. By the law of large numbers, $\hat{\mathcal{I}}(\boldsymbol{\Theta})$ converges to the (expected) FIM $\mathcal{I}(\boldsymbol{\Theta})$ as $n \to \infty$.

[1]King Abdullah University of Science and Technology (KAUST), Saudi Arabia [2]École Polytechnique, France [3]Sony Computer Science Laboratories Inc., Japan. Correspondence to: Ke Sun <sunk@ieee.org>, Frank Nielsen <Frank.Nielsen@acm.org>.

*Proceedings of the $34^{th}$ International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. 

The FIM is *not* invariant and depends on the parameterization. We can optionally write $\mathcal{I}(\boldsymbol{\Theta})$ as $\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta})$ to emphasize the coordinate system. By definition, $\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \boldsymbol{J}^\intercal \mathcal{I}_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda}) \boldsymbol{J}$ where $\boldsymbol{J} = (J_{ij})$, $J_{ij} = \frac{\partial \Lambda_i}{\partial \Theta_j}$ is the Jacobian matrix. For example, the FIM of regular natural exponential families (NEFs) $l(\boldsymbol{\Theta}) = \boldsymbol{\Theta}^\intercal \boldsymbol{t}(\boldsymbol{x}) - F(\boldsymbol{\Theta})$ (log-linear models with sufficient statistics $\boldsymbol{t}(\boldsymbol{x})$) is $\mathcal{I}(\boldsymbol{\Theta}) = \nabla^2 F(\boldsymbol{\Theta}) \succ 0$, the Hessian of the log-normalizer function $F(\boldsymbol{\Theta})$. Although exponential families can approximate arbitrarily *any* smooth density (Cobb et al., 1983), the log-normalizer function may not be available in closed-form nor computationally tractable (Montanari, 2015).

The FIM is an important concept for statistical machine learning. It gives a Riemannian metric (Hotelling, 1929; Rao, 1945) of the learning parameter space which is unique (Čencov, 1982; Dowty, 2017). Hence *any* learning is in a space that is intrinsically curved based on the FIM, regardless of the choice of the coordinate system. It also gives a bound (Fréchet, 1943; Cramér, 1946; Nielsen, 2013) of learning efficiency saying that the variance of *any* unbiased learning of $\boldsymbol{\Theta}$ is at least $\mathcal{I}^{-1}(\boldsymbol{\Theta})/n$, where $n$ is the i.i.d. sample size. The FIM is applied to neural network optimization (Amari, 1997), metric learning (Lebanon, 2005), reinforcement learning (Thomas, 2014) and manifold learning (Sun & Marchand-Maillet, 2014).

However computing the FIM is expensive. Besides the fact that learning machines have often singularities (Watanabe, 2009) ($|\mathcal{I}(\boldsymbol{\Theta})| = 0$, not full rank) characterized by plateaux in gradient learning, computing/estimating the FIM of a large neuron system (e.g. one with millions of parameters, Szegedy, Christian et al. 2015) is very challenging due to the finiteness of data, and the huge number $\frac{D(D+1)}{2}$ of matrix coefficients to evaluate. Furthermore, gradient descent techniques require inverting this large matrix and tuning the learning rate.

To tackle this problem, past works mainly focus on how to approximate the FIM with a block diagonal form (Kurita, 1994; Le Roux et al., 2008; Martens, 2010; Pascanu & Bengio, 2014; Martens & Grosse, 2015) or quasi-diagonal form (Ollivier, 2013; Marceau-Caron & Ollivier, 2016). This global approach faces increasing approximation error and increasing computational cost as the system scales up

and as complex and dynamic structures (Looks et al., 2017) emerge.

This work aims at a different local approach. The idea is to accurately describe the information geometry (IG) in a subsystem of the large learning system, which is invariant to the scaling up and structural change of the global system, so that the local machinery, including optimization, can be discussed regardless of the other parts.

For this purpose, a novel concept, the Relative Fisher Information Metric (RFIM), is defined. Unlike the traditional geometric view of a high-dimensional parameter manifold, RFIMs defines *multiple projected low-dimensional geometries of subsystems*. This geometry is correlated to the parameters beyond the subsystem and is therefore considered *dynamic*. It can be used to characterize the efficiency of a local learning process. Taking this stance has potential in deep learning because a deep neural network can be decomposed into many local components such as neurons or layers. The RFIM is well suited to the compositional block structures of neural networks. The RFIM can be used for out-of-core learning.

The paper is organized as follows. Sec. 2 reviews natural gradient within the context of Multi-Layer Perceptrons (MLPs). Sec. 3 formally defines the RFIM, and gives a table of RFIMs of several commonly used subsystems. Sec. 4 discusses the advantages of using the RFIM as compared to the FIM. Sec. 5 gives an algorithmic framework and proof-of-concept experiments on neural network optimization. Sec. 6 presents related works on parameter diagonalization. Sec. 7 concludes this work and further hints at perspectives.

## 2. Natural Gradient: Review and Insights

Consider a MLP $\boldsymbol{x} \xrightarrow{\boldsymbol{\theta}_1} \boldsymbol{h}_1 \cdots \boldsymbol{h}_{L-1} \xrightarrow{\boldsymbol{\theta}_L} \boldsymbol{y}$, whose statistical model is the following conditional distribution

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

The often intractable sum over $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{L-1}$ can be get rid off by deteriorating $p(\boldsymbol{h}_1 \,|\, \boldsymbol{x}, \boldsymbol{\theta}_1)$, $\cdots$, $p(\boldsymbol{h}_{L-1} \,|\, \boldsymbol{h}_{L-2}, \boldsymbol{\theta}_{L-1})$ to Dirac's deltas $\delta$, and letting merely the last layer $p(\boldsymbol{y} \,|\, \boldsymbol{h}_{L-1}, \boldsymbol{\theta}_L)$ be stochastic. Other models such as restricted Boltzmann machines (Nair & Hinton, 2010; Montavon & Müller, 2012), deep belief networks (Hinton et al., 2006), dropout (Wager et al., 2013), and variational autoencoders (Kingma & Welling, 2014) do consider the $\boldsymbol{h}_i$'s to be stochastic.

The tensor metric of the *neuromanifold* (Amari, 1995) $\mathcal{M}$, consisting of all MLPs with the same architecture but different parameter values, is locally defined by the FIM. Because a MLP corresponds to a conditional distribution, its FIM is a function of the input $\boldsymbol{x}$. By taking an empirical average over the input samples $\{\boldsymbol{x}_k\}_{k=1}^n$, the FIM of a MLP can be expressed as $\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \frac{1}{n}\sum_{k=1}^n E_{p(\boldsymbol{y} \,|\, \boldsymbol{x}_k, \boldsymbol{\Theta})}\left[\frac{\partial l_k}{\partial \boldsymbol{\Theta}} \frac{\partial l_k}{\partial \boldsymbol{\Theta}^\intercal}\right]$, where $l_k(\boldsymbol{\Theta}) = \log p(\boldsymbol{y} \,|\, \boldsymbol{x}_k, \boldsymbol{\Theta})$ denotes the conditional log-likelihood function wrt $\boldsymbol{x}_k$.

To understand the meaning of the Riemannian metric $\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta})$, it measures the intrinsic difference between two nearby neural networks around $\boldsymbol{\Theta} \in \mathcal{M}$. A learning step can be regarded as a tiny displacement $\delta\boldsymbol{\Theta}$ on $\mathcal{M}$. According to the FIM, the infinitesimal square distance

$$\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta})} = \frac{1}{n}\sum_{k=1}^n E_{p(\boldsymbol{y} \,|\, \boldsymbol{x}_k, \boldsymbol{\Theta})}\left[\left(\delta\boldsymbol{\Theta}^\intercal \frac{\partial l_k}{\partial \boldsymbol{\Theta}}\right)^2\right] \tag{1}$$

measures how much $\delta\boldsymbol{\Theta}$ (with a radius constraint) is statistically along $\frac{\partial l}{\partial \boldsymbol{\Theta}}$, or equivalently how much $\delta\boldsymbol{\Theta}$ affects intrinsically the conditional distribution $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\Theta})$.

Consider the negative log-likelihood function $L(\boldsymbol{\Theta}) = -\sum_{k=1}^n \log p(\boldsymbol{y}_k \,|\, \boldsymbol{x}_k, \boldsymbol{\Theta})$ wrt the observed pairs $\{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^n$, we try to minimize the loss while maintaining a small learning step size $\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta})}$ on $\mathcal{M}$. At $\boldsymbol{\Theta}_t \in \mathcal{M}$, the target is to minimize wrt $\delta\boldsymbol{\Theta}$ the Lagrange function

$$\begin{aligned} &L(\boldsymbol{\Theta}_t + \delta\boldsymbol{\Theta}) + \frac{1}{2\gamma}\langle \delta\boldsymbol{\Theta}, \delta\boldsymbol{\Theta} \rangle_{\mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_t)} \\ &\approx L(\boldsymbol{\Theta}_t) + \delta\boldsymbol{\Theta}^\intercal \bigtriangledown_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}_t) + \frac{1}{2\gamma}\delta\boldsymbol{\Theta}^\intercal \mathcal{I}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_t)\delta\boldsymbol{\Theta}, \end{aligned}$$

where $\gamma > 0$ is a learning rate. The optimal solution of the above quadratic optimization gives a learning step

$$\delta\boldsymbol{\Theta}_t = -\gamma \mathcal{I}_{\boldsymbol{\Theta}}^{-1}(\boldsymbol{\Theta}_t) \bigtriangledown_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}_t).$$

In this update procedure, $\tilde{\nabla}_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}) = \mathcal{I}_{\boldsymbol{\Theta}}^{-1}(\boldsymbol{\Theta}) \bigtriangledown_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta})$ replaces the role of the usual gradient $\nabla_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta})$ and is called the *natural gradient* (Amari, 1997).

Although the FIM depends on the chosen parameterization, the natural gradient is *invariant* to reparameterization. Let $\boldsymbol{\Lambda}$ be another coordinate system and $\boldsymbol{J}$ be the Jacobian matrix of the mapping $\boldsymbol{\Theta} \to \boldsymbol{\Lambda}$. Then we have

$$\begin{aligned} \mathcal{I}_{\boldsymbol{\Theta}}^{-1}(\boldsymbol{\Theta}) \bigtriangledown_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}) &= \left(\boldsymbol{J}^\intercal \mathcal{I}_{\boldsymbol{\Lambda}}(\boldsymbol{\Lambda}) \boldsymbol{J}\right)^{-1} \boldsymbol{J}^\intercal \bigtriangledown_{\boldsymbol{\Lambda}} L(\boldsymbol{\Lambda}) \\ &= \boldsymbol{J}^{-1} \mathcal{I}_{\boldsymbol{\Lambda}}^{-1}(\boldsymbol{\Lambda}) \bigtriangledown_{\boldsymbol{\Lambda}} L(\boldsymbol{\Lambda}), \end{aligned}$$

showing that $\tilde{\nabla}_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta})$ and $\tilde{\nabla}_{\boldsymbol{\Lambda}} L(\boldsymbol{\Lambda})$ are the same dynamic up to coordinate transformation. As the learning rate $\gamma$ is not infinitesimal in practice, natural gradient descent actually depends on the coordinate system (see e.g. Martens 2014). Other intriguing properties of natural gradient optimization lie in being free from getting trapped in plateaux of the error surface, and attaining Fisher efficiency in online learning (see Sec. 4 Amari 1998).

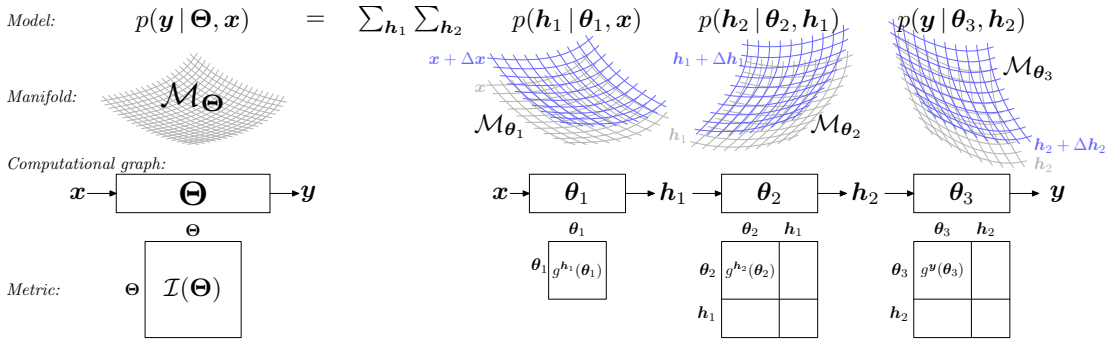



*Figure 1.* (left) The traditional global geometry of a MLP; (right) information geometry of subsystems. The gray and blue meshes show that the subsystem geometry is *dynamic* when the reference variable makes a tiny move. The square under the (sub-)system means the (R-)FIM is computed by (i) computing the FIM in the traditional way wrt all free parameters that affect the system output; (ii) choosing a sub-block that contains only the internal parameters of the (sub-)system and regarding the remaining variables as the reference.

For the sake of simplicity, we do not discuss singular FIMs with a subset of parameters having zero metric. This set of parameters forms an analytic variety (Watanabe, 2009), and technically the MLP as a statistical model is said to be non-regular (and the parameter $\boldsymbol{\Theta}$ is not identifiable). The natural gradient has been extended (Thomas, 2014) to cope with singular FIMs having positive semi-definite matrices by taking the Moore-Penrose pseudo-inverse (that coincides with the inverse matrix for full rank matrices).

In the family of $2^{\text{nd}}$-order optimization methods, a fuzzy line can be drawn from the natural gradient and alternative methods such as the Hessian-free optimization (Martens, 2010). By definition, the FIM is a property of the parameter space which is independent or weakly dependent on the input samples. For example, the FIM of a MLP is independent of $\{\boldsymbol{y}_i\}$. In contrast, the Hessian (or related concepts such as the Gauss-Newton matrix, Martens 2014) is a property of the learning cost function wrt the input samples.

Bonnabel (Bonnabel, 2013) proposed to use the Riemannian exponential map to define a gradient descent step, thus ensuring to stay on the manifold for any chosen learning rate. Convergence is proven for Hadamard manifolds (of negative curvatures). However, it is not mathematically tractable to express the exponential map of hierarchical model manifolds like the neuromanifold.

## 3. RFIM: Definition and Expressions

In general, for large parametric systems, it is impossible to diagonalize or decorrelate all the parameters, so that we split instead all random variables into three parts $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}$ and $\boldsymbol{h}$. We examine their intuitive meanings before giving the formal definition. The *reference*, $\boldsymbol{\theta}_f$, consists of the majority of the random variables that are considered fixed (therefore allowing us to simplify the analysis). This is in analogy to the notion of a reference frame in physics. $\boldsymbol{\theta}$ is the subsystem parameters, resembling the long-term memory adapting slowly to the observations (e.g. neural network weights). The *response* $\boldsymbol{h}$ is a random variable that reacts to the variations of $\boldsymbol{\theta}$. Usually, $\boldsymbol{h}$ is the output of the subsystem that is connected to neighbour subsystems (e.g. hidden layer outputs). Formally, a subsystem which factorizes the learning machine is characterized by the conditional distribution $p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)$, where $\boldsymbol{\theta}$ can be estimated based on $\boldsymbol{h}$ and $\boldsymbol{\theta}_f$. We make the following definition.

**Definition 1** (RFIM). *Given $\boldsymbol{\theta}_f$, the RFIM* [1] *of $\boldsymbol{\theta}$ wrt $\boldsymbol{h}$ is*

$$g^{\boldsymbol{h}}(\boldsymbol{\theta}\,|\,\boldsymbol{\theta}_f) \stackrel{\text{def}}{=} E_{p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\log p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)\frac{\partial}{\partial\boldsymbol{\theta}^{\intercal}}\log p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)\right],$$

*or simply $g^{\boldsymbol{h}}(\boldsymbol{\theta})$, corresponding to the estimation of $\boldsymbol{\theta}$ based on observations of $\boldsymbol{h}$ given $\boldsymbol{\theta}_f$.*

For example, consider a MLP. If we choose $\boldsymbol{\theta}_f$ to be the input features $\boldsymbol{x}$, choose $\boldsymbol{h}$ to be the final output $\boldsymbol{y}$, and choose $\boldsymbol{\theta}$ to be all the network weights $\boldsymbol{\Theta}$, then the RFIM becomes the FIM: $\mathcal{I}(\boldsymbol{\Theta}) = g^{\boldsymbol{y}}(\boldsymbol{\Theta}\,|\,\boldsymbol{x})$.

More generally, we can choose the response $\boldsymbol{h}$ to be other than the observables to compute the Fisher information of subsystems, especially dynamically during the learning of the global machine. To see the meaning of the RFIM, similar to eq. (1), the infinitesimal square distance $\langle\delta\boldsymbol{\theta},\delta\boldsymbol{\theta}\rangle_{g^{\boldsymbol{h}}(\boldsymbol{\theta})} = E_{p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)}\left[\left(\delta\boldsymbol{\theta}^{\intercal}\frac{\partial}{\partial\boldsymbol{\theta}}\log p(\boldsymbol{h}\,|\,\boldsymbol{\theta},\boldsymbol{\theta}_f)\right)^2\right]$ measures how much $\delta\boldsymbol{\theta}$ impacts intrinsically the stochastic mapping $\boldsymbol{\theta}\to\boldsymbol{h}$ which features the subsystem. We have the following proposition following definition 1.

**Proposition 2** (Relative Geometry Consistency). *If $\boldsymbol{\theta}_1$ consists of a subset of $\boldsymbol{\theta}_2$ so that $\boldsymbol{\theta}_2 = (\boldsymbol{\theta}_1,\tilde{\boldsymbol{\theta}}_1)$, then $\forall\tilde{\boldsymbol{\theta}}_1$, $\mathcal{M}_{\boldsymbol{\theta}_1}$ with the metric $g^{\boldsymbol{h}}(\boldsymbol{\theta}_1\,|\,\tilde{\boldsymbol{\theta}}_1)$ has exactly the same Rie-*

[1] We use the same term "relative FIM" (Zegers, 2015) with a different definition.

*mannian metric with the sub-manifold* $\{\boldsymbol{\theta}_2 \in \mathcal{M}_{\boldsymbol{\theta}_2} : \tilde{\boldsymbol{\theta}}_1 \textit{ is fixed}\}$ *induced by the ambient metric* $g^{\boldsymbol{h}}(\boldsymbol{\theta}_2)$.

When the response $\boldsymbol{h}$ is chosen, then different splits of $(\boldsymbol{\theta}, \boldsymbol{\theta}_f)$ are consistent with the same ambient geometry.

Figure 1 shows the traditional global geometry of a learning system, where the curvature is defined by the learner's parameter sensitivity to the external environment ($\boldsymbol{x}$ and $\boldsymbol{y}$), as compared to the information geometry of subsystems, where the curvature is defined by the parameter sensitivity wrt hidden interface variables $\boldsymbol{h}$. The two-colored meshes show that the geometry structure is dynamic and varies with the reference variable $\boldsymbol{\theta}_f$.

One should not confuse the RFIM with the diagonal blocks of the FIM (Kurita, 1994). Both their meanings and expressions are different. The RFIM is computed by integrating out the hidden response variables $\boldsymbol{h}$. The FIM is always computed by integrating out the observables $\boldsymbol{x}$ and $\boldsymbol{y}$. Hence the RFIM is a more general concept and includes the FIM as a special case. This highlights a main difference with the backpropagated metric (Ollivier, 2013), which essentially considers parameter sensitivity wrt the final output. Despite the fact that the FIMs of small parametric structures such as single neurons was studied (Amari, 1997), we are not looking at a small single-component system but a component embedded in a large system, targeting at improving the large system.

In the following we provide a short table of commonly used RFIMs for future reference (the RFIMs listed are mostly straightforward from definition 1, with detailed derivations given in the supplementary material). This is meaningful since the RFIM is a new concept. We also want to demonstrate these simple closed form expressions without any approximations.

### 3.1. RFIMs of One Neuron

We start from the RFIM of single neuron models. Consider a stochastic neuron with input $\boldsymbol{x}$ and weights $\boldsymbol{w}$. After a nonlinear activation function $f$, the output $y$ is randomized surrounding the mean $f(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})$ with a variance. Throughout this paper $\tilde{\boldsymbol{x}} = (\boldsymbol{x}^\intercal, 1)^\intercal$ denotes the augmented vector of $\boldsymbol{x}$ (homogeneous coordinates) so that $\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}$ contains a bias term, and a general linear transformation can be written simply as $\boldsymbol{A}\tilde{\boldsymbol{x}}$.

Using $\boldsymbol{x}$ as the reference, the RFIM of $\boldsymbol{w}$ with respect to $y$ has a common form $g^y(\boldsymbol{w} \,|\, \boldsymbol{x}) = \nu_f(\boldsymbol{w}, \boldsymbol{x}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal$, 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. This agrees with early studies on single neuron FIMs (Amari, 1997; Kurita, 1994).

If $f(t) = \texttt{tanh}(t)$ is the hyperbolic tangent function, then $\nu_f(\boldsymbol{w}, \boldsymbol{x}) = \texttt{sech}^2(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})$, where $\texttt{sech}(t) = \frac{2}{\exp(t)+\exp(-t)}$ is the hyperbolic secant function. Similarly, if $f(t) = \texttt{sigm}(t)$ is the sigmoid function, then $\nu_f(\boldsymbol{w}, \boldsymbol{x}) = \texttt{sigm}\left(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}\right)\left[1 - \texttt{sigm}\left(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}\right)\right]$.

If $f$ is defined by Parametric Rectified Linear Unit (PReLU) (He et al., 2015), which includes Rectified Linear Unit (ReLU) (Nair & Hinton, 2010) as a special case, so that $f(t) = t\ (t \geq 0)$, $f(t) = \iota t\ (t < 0)$, $0 \leq \iota < 1$, then under certain approximations (see supplementary material)

$$\nu_f(\boldsymbol{w}, \boldsymbol{x}) = \left[\iota + (1-\iota)\texttt{sigm}\left(\frac{1-\iota}{\omega}\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}\right)\right]^2,$$

where $\omega > 0$ is a hyper-parameter (e.g. $\omega = 1$).

For the exponential linear unit (ELU) (Clevert et al., 2015), $f(t) = t\ (t \geq 0)$, $f(t) = \alpha\left(\exp(t) - 1\right)\ (t < 0)$, where $\alpha > 0$ is a hyper-parameter. We get

$$\nu_f(\boldsymbol{w}, \boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \geq 0 \\ \alpha^2 \exp\left(2\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}\right) & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} < 0. \end{cases}$$

### 3.2. RFIM of One Layer

Let $D$ denote the dimensionality of the corresponding variable. A linear layer with input $\boldsymbol{x}$, connection weights $\boldsymbol{W} = \left[\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{D_y}\right]$, and stochastic output $\boldsymbol{y}$ can be represented by $\boldsymbol{y} \sim G(\boldsymbol{W}^\intercal \tilde{\boldsymbol{x}}, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix, and $\sigma$ is the scale of the observation noise, and $G(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We vectorize $\boldsymbol{W}$ by stacking its columns $\{\boldsymbol{w}_i\}$. Then $g^{\boldsymbol{y}}(\boldsymbol{W} \,|\, \boldsymbol{x})$ is a tensor of size $(D_x+1)D_y \times (D_x+1)D_y$, given by $g^{\boldsymbol{y}}(\boldsymbol{W} \,|\, \boldsymbol{x}) = \texttt{diag}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal, \cdots, \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right]$, where $\texttt{diag}(\cdot)$ means the (block) diagonal matrix constructed by the given matrix entries.

A *nonlinear* layer increments a linear layer by adding an element-wise activation function applied on $\boldsymbol{W}^\intercal \tilde{\boldsymbol{x}}$, and then randomized wrt the choice of the neuron. By definition 1, its RFIM is given by

$$g^{\boldsymbol{y}}(\boldsymbol{W} \,|\, \boldsymbol{x}) = \texttt{diag}\left[\nu_f(\boldsymbol{w}_1, \boldsymbol{x})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal, \cdots, \nu_f(\boldsymbol{w}_m, \boldsymbol{x})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right], \quad (2)$$

where $\nu_f(\boldsymbol{w}_i, \boldsymbol{x})$ is given in Subsec. 3.1.

A *softmax* layer, which often appears as the last layer of a MLP, is given by $y \in \{1, \ldots, m\}$, where $p(y) = \eta_y = \frac{\exp(\boldsymbol{w}_y \tilde{\boldsymbol{x}})}{\sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})}$. Its RFIM is a dense matrix given by

$$g^{\boldsymbol{y}}(\boldsymbol{W}) = \begin{bmatrix} (\eta_1 - \eta_1^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_1\eta_m\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \\ -\eta_2\eta_1\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_2\eta_m\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \\ \vdots & \ddots & \vdots \\ -\eta_m\eta_1\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \cdots & (\eta_m - \eta_m^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \end{bmatrix}.$$

Notice that its $i$'th diagonal block $(\eta_i - \eta_i^2)\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$ resembles the RFIM of a single $\texttt{sigm}$ neuron.

### 3.3. RFIM of Two Layers

By eq. (2), the one-layer RFIM is a product metric (Jost, 2011) and does not consider the *inter-neuron correlations*, which must be obtained by looking at a larger subsystem. Consider a two-layer model with stochastic output $\boldsymbol{y}$ around the mean vector $f(\boldsymbol{C}^\intercal \tilde{\boldsymbol{h}})$, where $\boldsymbol{h} = f\left(\boldsymbol{W}^\intercal \tilde{\boldsymbol{x}}\right)$. For simplicity, we ignore inter-layer correlations between the first layer and the second layer and focus on the inter-neuron correlations within the first layer. To do this, both $\boldsymbol{x}$ and $\boldsymbol{C}$ are considered as references to compute the RFIM of $\boldsymbol{W}$. By definition 1, $g^{\boldsymbol{y}}(\boldsymbol{W} \,|\, \boldsymbol{x}, \boldsymbol{C}) = [\boldsymbol{G}_{ij}]_{D_h \times D_h}$ and each block has the form

$$\boldsymbol{G}_{ij} = \sum_{l=1}^{D_y} c_{il} c_{jl} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) \nu_f(\boldsymbol{w}_i, \boldsymbol{x}) \nu_f(\boldsymbol{w}_j, \boldsymbol{x}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal.$$

Now that we have the one-layer and two-layer RFIMs, we can either split a given feed-forward neural network into one-layer subsystems or into two-layer subsystems. A trade-off is that using a larger subsystem entails greater analytical and computational difficulty, although it could more accurately model the global system dynamics. In the extreme case, the FIM is obtained if the whole system is considered as one single subsystem.

## 4. RFIM: Key Advantages

This section discusses the theoretical advantages of the RFIM over the FIM. Consider wlog a MLP with Bernoulli outputs $\boldsymbol{y} \in \{0, 1\}^m$, whose mean $\boldsymbol{\mu}$ is a deterministic function depending on the input $\boldsymbol{x}$ and the network parameters $\boldsymbol{\Theta}$. By Sec. 2, the FIM of the MLP can be computed as (see supplementary for proof)

$$\mathcal{I}(\boldsymbol{\Theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{\mu_j(\boldsymbol{x}_i)(1 - \mu_j(\boldsymbol{x}_i))} \frac{\partial \mu_j(\boldsymbol{x}_i)}{\partial \boldsymbol{\Theta}} \frac{\partial \mu_j(\boldsymbol{x}_i)}{\partial \boldsymbol{\Theta}^\intercal}. \tag{3}$$

Therefore $\texttt{rank}(\mathcal{I}(\boldsymbol{\Theta})) \leq nm$. The rank of a diagonal block of $\mathcal{I}(\boldsymbol{\Theta})$ corresponding to one layer is even smaller. In a deep neural network (e.g. Szegedy, Christian et al. 2015), if the sample size $n < \dim(\boldsymbol{\Theta})/m$, then $\mathcal{I}(\boldsymbol{\Theta})$ is doomed to be singular. All methods trying to approximate the FIM suffer from this problem and therefore rely on proper regularizations. If the network is decomposed into layers, the RFIM of each subsystem (layer) is given by eq. (2). Each sample can contribute maximally 1 to the rank of the neuron-RFIM and can contribute maximally $D_y$ to the rank of the layer-RFIM. It only requires $\max_i\{\dim(\boldsymbol{w}_i)\}$ (the maximum layer width) observations to have a full rank RFIM, where $\boldsymbol{w}_i$ is the weight vector of the $i$'th neuron. The RFIM is expected to have a much higher rank than the FIM. Higher rank means less singularity and more information is captured. Models that can be distinguished by the RFIM may be identical in the sense of the FIM. Essentially, the RFIM integrates the *internal randomness* (Bengio, 2013) of the neural system by considering the output of each layer as a random variable. In theory, the FIM should also consider stochastic neurons. However it requires marginalizing the joint distribution of $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{y}$. This makes the already infeasible computation even more challenging.

The RFIM is not an approximation of the FIM but is an *accurate* metric, defining the geometry of $\boldsymbol{\theta}$ wrt to its direct response $\boldsymbol{h}$ in the system, or adjacent nodes in a graphical model. By the example in fig. 1, $g^{\boldsymbol{y}}(\boldsymbol{\theta}_L)$ of the last layer is exactly the corresponding block in $\mathcal{I}(\boldsymbol{\Theta})$: they both characterize how $\boldsymbol{\theta}_L$ affects the mapping $\boldsymbol{h}_{L-1} \to \boldsymbol{y}$. They start to diverge from the second to last layer. To compute the geometry of $\boldsymbol{\theta}_{L-1}$, the RFIM looks at how $\boldsymbol{\theta}_{L-1}$ affects the local mapping $\boldsymbol{h}_{L-2} \to \boldsymbol{h}_{L-1}$, which can be measured reliably regardless of the rest of the system (think of a "debugging" process to separate and measure a single component). In contrast, the FIM examines how $\boldsymbol{\theta}_{L-1}$ affects the non-local mapping $\boldsymbol{h}_{L-2} \to \boldsymbol{y}$. This is a difficult task because it must consider the correlation between different layers. As an approximation, the block diagonalized version of the FIM ignores such correlations and therefore faces the loss of accuracy.

The RFIM makes it possible to maintain global system stability so that the intrinsic variations of different subsystems are balanced during learning. Consider a set of interconnected subsystems with internal parameters $\{\boldsymbol{\theta}_l\}$ and the corresponding response variables $\{\boldsymbol{h}_l\}$. The RFIM $g^{\boldsymbol{h}_l}(\boldsymbol{\theta}_l)$ measures how much the likelihood surface of $\boldsymbol{h}_l$ is curved wrt a small learning step $\delta\boldsymbol{\theta}_l$. By constraining the squared Riemannian distance $\delta\boldsymbol{\theta}_l^\intercal g^{\boldsymbol{h}_l}(\boldsymbol{\theta}_l)\delta\boldsymbol{\theta}_l$ having similar scales, different subsystems will present similar variations during learning. Within one subsystem, the learning along sensitive parameter directions is penalized. Among different subsystems, the learning of sensitive subsystems is penalized. Globally, the inter-subsystem stochastic connections have similar variance, maintaining a stable reference system and achieving efficient learning. This is similar to the idea of batch normalization (BN) (Ioffe & Szegedy, 2015) but has a deeper theoretical foundation.

Formally, we have the following theorem.

**Theorem 3.** *Consider a learning system represented by a joint distribution $p(\boldsymbol{x}, \boldsymbol{h})$ of $\boldsymbol{x}$ (observables) and $\boldsymbol{h}$ (hidden variables which connect subsystems). The joint FIM $\mathcal{J}(\boldsymbol{\Theta}) = E_p\left(\frac{\log p(\boldsymbol{x},\boldsymbol{h} \,|\, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \frac{\log p(\boldsymbol{x},\boldsymbol{h} \,|\, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}^\intercal}\right)$ has a block diagonal form. Each block is $E_p(g^{\boldsymbol{h}}(\boldsymbol{\theta}))$, where $\boldsymbol{\theta}$ is the parameters within a subsystem and $\boldsymbol{h}$ is its response variables to neighour subsystems.*

The global correspondence of the local RFIM is the joint

FIM. By theorem 3, the square distance $d\boldsymbol{\Theta}^\intercal \mathcal{J}(\boldsymbol{\Theta}) d\boldsymbol{\Theta} = E_p(\sum_l d\boldsymbol{\theta}_l^\intercal g^{\boldsymbol{h}_l}(\boldsymbol{\theta}_l) d\boldsymbol{\theta}_l)$ measures the system variance, including both the observables $\boldsymbol{x}$ and the hidden variables $\boldsymbol{h}$. An intrinsic trade-off between the RFIM and the FIM is learning system stability versus efficiency. Normalizing the FIM is more efficient because it helps to achieve Fisher efficiency (Amari, 1998). Normalizing the RFIM is more stable since the hidden variations are bounded, which only guarantees subsystem Fisher efficiency characterized by the Cramér-Rao lower bound of local parameters.

## 5. Relative Natural Gradient Descent

The traditional *non-parametric* way of applying natural gradient requires re-calculating the FIM and solving a large linear system in each learning step. Besides the huge computational cost, it has a large approximation error. For example during online learning, a mini-batch of samples cannot faithfully reflect the "true" geometry, which has to integrate the risk of sample variations. That is, the FIM of a mini-batch is likely to be singular or poorly conditioned.

A recent series of efforts (Montavon & Müller, 2012; Raiko et al., 2012; Desjardins et al., 2015) are gearing towards a parametric approach to applying natural gradient, which memorizes and learns a geometry. For example, natural neural networks (Desjardins et al., 2015) augment each layer with a redundant linear layer, and let these linear layers parametrize the geometry of the neural manifold.

By dividing the learning system into subsystems, the RFIM potentially gives a systematical implementation of parametric natural gradient descent. The memory complexity of storing the Riemannian metric has been reduced from $O(D^2)$ to $O(\sum_i D_i^2)$, where $D_i = \dim(\boldsymbol{w}_i)$ is the size of the $i$'th neuron. Consider there are $M$ neurons in total, then the memory cost is reduced by a factor of $M$. The computational complexity has been reduced from $O(D^\varrho)$ ($\varrho \approx 2.373$, Williams 2012) to $O(\sum_i D_i^\varrho)$. Optimization based on RFIM is called Relative Natural Gradient Descent (RNGD).

The good performance of batch normalization (Ioffe & Szegedy, 2015) provides an empirical support for the RFIM. Basically, BN uses an *inter-sample* normalization layer to transform the layer input $\boldsymbol{x}$ to $\boldsymbol{z}$ with zero mean and unit variance and thus reduces "internal covariate shift". In a typical case, above this normalization layer is a linear layer given by $\boldsymbol{y} = \boldsymbol{W}^\intercal \tilde{\boldsymbol{z}}$. If each dimension of $\boldsymbol{z}$ is normalized, then the diagonal blocks of the linear layer RFIM $g^{\boldsymbol{y}}(\boldsymbol{W}) = \mathtt{diag}[\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}^\intercal, \cdots, \tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}^\intercal]$ become a covariance matrix with identity diagonal entries (after taking an empirical average). This gives the coordinate system $\boldsymbol{W}$ a well conditioned RFIM for efficient learning.

### 5.1. RNGD with a `relu` MLP

This subsection builds a proof-of-concept experiment on MLP optimization. We partition the MLP into layers (one layer consists of a linear layer plus an element-wise nonlinear activation function) as the subsystems. By eq. (2), the RFIM of layer $l$ ($l = 1, \cdots, L$) with input $\boldsymbol{h}_{l-1}$ ($\boldsymbol{h}_0 = \boldsymbol{x}$) and weights $\{\boldsymbol{w}_{l1}, \cdots, \boldsymbol{w}_{lm_l}\}$ is

$$\mathtt{diag}\left[\nu_f(\boldsymbol{w}_{l1}, \boldsymbol{h}_{l-1})\tilde{\boldsymbol{h}}_{l-1}\tilde{\boldsymbol{h}}_{l-1}^\intercal, \cdots, \nu_f(\boldsymbol{w}_{lm_l}, \boldsymbol{h}_{l-l})\tilde{\boldsymbol{h}}_{l-1}\tilde{\boldsymbol{h}}_{l-1}^\intercal\right].$$

The subsystem stability during one learning step $\delta\boldsymbol{w}$ can be measured geometrically by $\sum_{l=1}^{L}\sum_{i=1}^{m_l}\nu_f(\boldsymbol{w}_{li}, \boldsymbol{h}_{l-1})(\delta\boldsymbol{w}_{li}^\intercal\tilde{\boldsymbol{h}}_{l-1})^2$. Using this term as the geometric cost (the Lagrange term) in the trust region approach in Sec. 2, we get the following RNGD method. In a stochastic gradient descent scenario, each neuron $i$ in layer $l$ is updated by

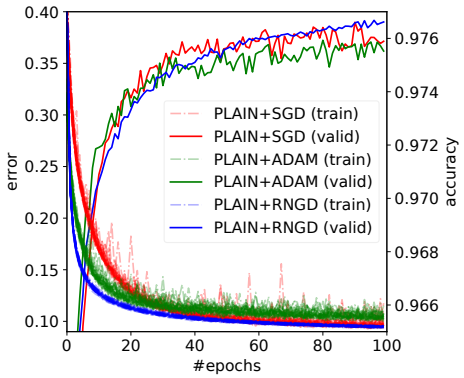
$$\boldsymbol{w}_{li}^{\text{new}} \leftarrow \boldsymbol{w}_{li}^{\text{old}} - \boldsymbol{G}_{li}^{-1}\frac{\partial E}{\partial \boldsymbol{w}_{li}},$$

where $E$ is the cost function and $\boldsymbol{G}_{li}$ is a *learned metric*. The consideration is that a mini-batch of samples do not contain enough information to compute the RFIM, which should be averaged over all training samples. Therefore, for the $i$'th neuron in layer $l$, $\boldsymbol{G}_{li}$ is initialized to identity, and is updated based on

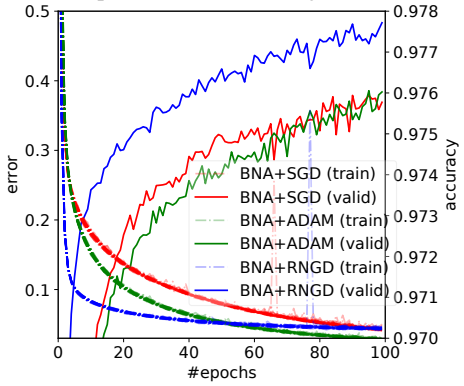
$$\boldsymbol{G}_{li}^{\text{new}} \leftarrow (1-\lambda)\boldsymbol{G}_{li}^{\text{old}} + \lambda\overline{\nu_f(\boldsymbol{w}_{li}, \boldsymbol{h}_{l-1})\tilde{\boldsymbol{h}}_{l-1}\tilde{\boldsymbol{h}}_{l-1}^\intercal + \epsilon\boldsymbol{I}},$$

where $\epsilon > 0$ is a hyper-parameter to avoid singularity caused by small sample size, and the average is taken over all samples in a mini-batch, and $\lambda$ is a learning rate. In theory, $\lambda$ should be gradually reduced to zero to guarantee the convergence of this geometry learning. To avoid solving a linear system in each iteration, every $T$ iterations we recompute and store $\boldsymbol{G}_{li}^{-1}$ based on the most updated $\boldsymbol{G}_{li}$. In the next $T$ iterations, this $\boldsymbol{G}_{li}^{-1}$ will be used as an approximation of the inverse RFIM. For the input layer which scales with the number of input features, and the final softmax layer, we apply instead the RFIM of the corresponding linear layer to improve the computational efficiency.

We compare different optimizers on classifying MNIST digits. The network has shape 784-80-80-80-10, with `relu` activation units, a final soft-max layer, and uses the per-sample average cross-entropy with $L_2$-regularization as the learning cost function. We experiment on two different architectures: one is a plain MLP (PLAIN); the other has a batch normalization layer after each hidden layer (BNA), where a rescaling parameter is applied to ensure enough flexibility of the parametric structure (Ioffe & Szegedy, 2015). For simplicity, the architecture, mini-batch size (50), and $L_2$ regularization strength ($10^{-3}$) are fixed to be the same for all compared methods. The observations are consistent when these configurations vary.




(a) A plain MLP with $L_2$ regularization




(b) A MLP with batch normalization and $L_2$ regularization

*Figure 2.* Learning curves of different optimizers on a MLP with two different architectures (with and without BN). The best learning rate for each method is selected based on the validation accuracy. Using this learning rate, the learning curves wrt 40 different random initializations are shown. The mean validation curve is shown for a clear visualization.

Figure 2 shows the learning curves of different methods. SGD is stochastic gradient descent. ADAM is the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Our RNGD is implemented by modifying TensorFlow's (Abadi, Martín et al., 2015) SGD optimizer. We set empirically $T = 100$, $\lambda = 0.005$ and $\omega = 1$.

RNGD presents a sharper learning curve and better generalization, especially when it is combined with BN. In this case, the final tranining error of RNGD is slightly larger than ADAM because by validation it favors a larger learning rate, which is applied on the neural network weights (based on RNGD) and BN parameters (based on SGD). For the ReLU activation, $\nu_f(\boldsymbol{w}_i, \boldsymbol{x})$ is approximately binary, emphasizing such *informative samples* with $\boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}} > 0$, which are the ones contributing to the learning of $\boldsymbol{w}_i$ with non-zero gradient values. Each output neuron has a different subset of informative samples. RNGD normalizes $\boldsymbol{x}$ differently wrt different output neurons, so that the informative samples for each output neuron are centered and decorrelated.

In the above experiment, RNGD's computational time per each epoch is roughly $4 \sim 10$ times more than SGD and ADAM on a modern graphic card. Therefore in terms of wall clock time RNGD does not show advantages. This can be improved by more efficient implementations with low rank approximation techniques and early stopping. Our RNGD prototype hints at a promising direction to develop scalable $2^{\text{nd}}$-order deep learning optimizers based on the RFIM.

## 6. Related Works on FIM Diagonalization

One may ponder whether we can always find a suitable parameterization that yields a diagonal FIM that is straightforward to invert. This fundamental problem of parameter orthogonalization was first investigated by Jeffreys (1998) for decorrelating the *parameters of interest* from the *nuisance parameters*. Fisher diagonalization yields parameter orthogonalization (Cox & Reid, 1987), and is proved useful when estimating $\hat{\boldsymbol{\Theta}}$ using a maximum likelihood estimator (MLE) that is asymptotically normally distributed, $\hat{\boldsymbol{\Theta}}_n \sim G(\boldsymbol{\Theta}, \mathcal{I}^{-1}(\boldsymbol{\Theta})/n)$, and efficient since the variance of the estimator matches the Cramér-Rao lower bound. Using the chain rule, this amounts to find a suitable parameterization $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{\Theta})$ satisfying

$$\sum_{i,j} E\left[\frac{\partial^2 l}{\partial\Theta_i \partial\Theta_j}\right] \frac{\partial\Theta_i}{\partial\Omega_k}\frac{\partial\Theta_j}{\partial\Omega_l} = 0, \quad \forall k \neq l.$$

Thus in general, we end up with $\binom{D}{2} = \frac{D(D-1)}{2}$ (nonlinear) partial differential equations to satisfy (Huzurbazar, 1950). Therefore, in general there is no solution when $\binom{D}{2} > D$, that is when $D > 3$. When $D = 2$, the single differential equation is usually solvable and tractable, and the solution may not be unique: For example, Huzurbazar (1950) reports two orthogonalization schemes for the location-scale families $\{\frac{1}{\sigma}p_0(\frac{x-\mu}{\sigma})\}$ that include the Gaussian family and the Cauchy family. Sometimes, the structure of the differential equation system yields a solution: For example, Jeffreys (1998) reported a parameter orthogonalization for Pearson's distributions of type I which is of order $D = 4$. Cox and Reid (1987) further investigated this topic with application to conditional inference, and provide examples (including the Weibull distribution).

From the viewpoint of geometry, the FIM induces a Riemannian manifold with metric tensor $g(\boldsymbol{\Theta}) = \mathcal{I}(\boldsymbol{\Theta})$. When the FIM may be degenerate, this yields a pseudo-Riemannian manifold (Thomas, 2014). In differential geometry, orthogonalization amounts to transforming the square length infinitesimal element $g_{ij}\mathrm{d}\Theta_i\Theta_j$ of a Riemannian geometry into an orthogonal system $\omega$ with match-

ing square length infinitesimal element $\Omega_{ii}\mathrm{d}\Omega_i^2$. However, such a global orthogonal metric does not exist (Huzurbazar, 1950) when $D > 3$ for an arbitrary metric tensor, although interesting Riemannian parameterization structures may be derived in Riemannian 4D geometry (Grant & Vickers, 2009).

For NEFs, the FIM can be made *block-diagonal* easily by using the *mixed coordinate system* (Amari, 2016) $(\boldsymbol{\Theta}_{1:k}, \boldsymbol{H}_{k+1:D})$, where $\boldsymbol{H} = E_p[\boldsymbol{t}(\boldsymbol{x})] = \nabla F(\boldsymbol{\Theta})$ is the moment parameter, for any $k \in \{1, ..., D-1\}$, where $\boldsymbol{v}_{b:e}$ denotes the subvector $(v_b, ..., v_e)^\intercal$ of $\boldsymbol{v}$. The geometry of NEFs is a dually flat structure (Amari, 2016) induced by the convex mgf, the potential function. It defines a dual affine coordinate systems $e^i = \partial_i = \frac{\partial}{\partial H_i}$ and $e_j = \partial^j = \frac{\partial}{\partial \Theta^j}$ that are orthogonal: $\langle e^i, e_j \rangle = \delta_j^i$, where $\delta_j^i = 1$ iff $i = j$ and $\delta_j^i = 0$ otherwise. Hence the FIM has two diagonal blocks. Those dual affine coordinate systems are defined up to an affine invertible transformation: $\tilde{\boldsymbol{\Theta}} = \boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{b}$, $\tilde{\boldsymbol{H}} = \boldsymbol{A}^{-1}\boldsymbol{H} + \boldsymbol{c}$. In particular, for any order-2 NEF ($D = 2$), we can *always* obtain two mixed parameterizations $(\Theta_1, H_2)$ or $(H_1, \Theta_2)$.

The RFIM contributes another line of thought in parameter diagonalization. We investigate the Fisher information of *hidden variables*, or internal interfaces in the learning machine. This is novel since the majority of previous works concentrate on the FIM of the observables, or the external interface of the machine. From a causality perspective, we factor out the main cause (parameters within the subsystem) of the response variable with a direct action-reaction relationship, and regard the remaining parameters as a *reference* that can be easily estimated by the empirical distribution. This simplification may lead to broader applications of Fisher information in machine learning.

The particular case of a mixed coordinate system (that is not an affine coordinate system) induces in information geometry (Amari, 2016) a dual pair of orthogonal e- and m-orthogonal foliations. Our splits in RFIMs consider general non-orthogonal foliations that provide the factorization decompositions of the whole manifold into submanifolds, that are the leaves of the foliation (see section 3.7 of Amari & Nagaoka 2000).

## 7. Conclusion and Discussions

We investigate local structures of large learning systems using the new concept of Relative Fisher Information Metric. The key advantage of this approach is that the local learning dynamics can be analyzed in an accurate way without approximation. We present a core list of such local structures in neural networks, and give their corresponding RFIMs. This list of recipes can be used to provide guiding principles to design new optimizers for deep learning.

Our work applies to mirror descent as well since natural gradient is related to mirror descent (Raskutti & Mukherjee, 2015) as follows: In mirror descent to minimize a cost function $E(\boldsymbol{\Theta})$, given a strictly convex distance function $D(\cdot, \cdot)$ in the first argument (playing the role of the proximity function), we express the gradient descent step as:

$$\boldsymbol{\Theta}_{t+1} = \arg\min_{\boldsymbol{\Theta}} \left\{ \boldsymbol{\Theta}^\top \nabla E(\boldsymbol{\Theta}_t) + \frac{1}{\gamma} D(\boldsymbol{\Theta}, \boldsymbol{\Theta}_t) \right\}.$$

When $D(\boldsymbol{\Theta}, \boldsymbol{\Theta}')$ is chosen as a Bregman divergence $B_F(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = F(\boldsymbol{\Theta}) - F(\boldsymbol{\Theta}') - (\boldsymbol{\Theta} - \boldsymbol{\Theta}')^\top \nabla F(\boldsymbol{\Theta}')$ wrt to a convex function $F$, it has been proved that the mirror descent on the $\boldsymbol{\Theta}$-parameterization is equivalent (Raskutti & Mukherjee, 2015) to the natural gradient optimization on the induced Riemannian manifold with metric tensor $(\nabla^2 F(\boldsymbol{\Theta}))$ parameterized by the dual coordinate system $\boldsymbol{H} = \nabla F(\boldsymbol{\Theta})$.

In general, to perform a Riemannian gradient descent for minimizing a real-valued function $f(\boldsymbol{\Theta})$ on the manifold, one needs to choose a proper metric tensor given in matrix form $\boldsymbol{G}(\boldsymbol{\Theta})$. Thomas (2014) constructed a toy example showing that the natural gradient may diverge while the ordinary gradient (for $\boldsymbol{G} = \boldsymbol{I}$) converges. Recently, Thomas et al. (2016) proposed a new kind of descent method based on what they called the Energetic Natural Gradient that generalizes the natural gradient. The energy distance $D_E(p(\Theta_1), p(\Theta_2))^2 = E[2d_{p(\Theta_1)}(X, Y) - d_{p(\Theta_1)}(X, X') - d_{p(\Theta_1)}(Y, Y')]$ where $X, X' \sim p(\Theta_1)$ and $Y, Y' \sim p(\Theta_2)$, where $d_{p(\Theta_1)}(\cdot, \cdot)$ is a distance metric over the support. Using a Taylor's expansion on their energy distance, they get the Energy Information Matrix (in a way similar to recovering the FIM from a Taylor's expansion of any $f$-divergence like the Kullback-Leibler divergence). Their idea is to incorporate prior knowledge on the structure of the support (observation space) to define energy distance. Twisting the geometry of the support (say, Wasserstein's optimal transport) with the geometry of the parametric distributions (Fisher-Rao geodesic distances) is indeed important (Chizat et al., 2015). In information geometry, invariance on the support is provided by a Markov morphism that is a probabilistic mapping of the support to itself (Čencov, 1982). There is no neighbourhood structure on the support in IG. Markov morphism includes deterministic transformation of a random variable by a statistic. It is well-known that $\mathcal{I}_T(\Theta) \preceq \mathcal{I}_X(\Theta)$ with equality iff. $T = T(X)$ is a sufficient statistic of $X$. Thus to get the same invariance for the energy distance (Thomas et al., 2016), one shall further require $d_{p(\Theta)}(T(X), T(Y)) = d_{p(\Theta)}(X, Y)$.

We believe that RFIMs will provide a sound methodology to build further efficient systems for deep learning. The full source codes to reproduce the experimental results are available at `https://www.lix.polytechnique.fr/~nielsen/RFIM`.

## Acknowledgements

The authors would like to thank the anonymous reviewers and Yann Ollivier for the helpful comments. This work was mainly conducted when the first author was a postdoctoral researcher at École Polytechnique.

## References

Abadi, Martín et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Amari, Shun'ichi. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9): 1379–1408, 1995.

Amari, Shun'ichi. Neural learning in structured parameter spaces – natural Riemannian gradient. In *NIPS 9*, pp. 127–133. MIT Press, 1997.

Amari, Shun'ichi. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, 1998.

Amari, Shun'ichi. *Information Geometry and its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer Japan, 2016.

Amari, Shun'ichi and Nagaoka, Hiroshi. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. AMS and OUP, 2000. (Published in Japanese in 1993).

Bengio, Yoshua. Estimating or propagating gradients through stochastic neurons. *CoRR*, abs/1305.2982, 2013.

Bonnabel, Silvère. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Automat. Contr.*, 58(9): 2217–2229, 2013.

Čencov, Nikolaĭ Nikolaevich. *Statistical decision rules and optimal inference*, volume 53 of *Translations of Mathematical Monographs*. American Mathematical Society, 1982. Translation from the Russian (published in 1972) edited by Lev J. Leifman.

Chizat, Lenaic, Schmitzer, Bernhard, Peyré, Gabriel, and Vialard, François-Xavier. An Interpolating Distance between Optimal Transport and Fisher-Rao. *arXiv e-prints*, 2015. 1506.06430 [math.AP].

Clevert, Djork-Arné, Unterthiner, Thomas, and Hochreiter, Sepp. Fast and accurate deep network learning by exponential linear units (ELUs). *CoRR*, abs/1511.07289, 2015.

Cobb, Loren, Koppstein, Peter, and Chen, Neng Hsin. Estimation and moment recursion relations for multimodal distributions of the exponential family. *JASA*, 78(381): 124–130, 1983.

Cox, D. R. and Reid, N. Parameter orthogonality and approximate conditional inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 49(1):1–39, 1987.

Cramér, Harald. *Mathematical Methods of Statistics*, volume 9 of *Princeton Mathematical Series*. Princeton University Press, 1946.

Desjardins, Guillaume, Simonyan, Karen, Pascanu, Razvan, and Kavukcuoglu, Koray. Natural neural networks. In *NIPS 28*, pp. 2071–2079. Curran Associates, Inc., 2015.

Dowty, James G. Chentsov's theorem for exponential families. *arXiv preprints*, 2017. 1701.08895 [math.ST].

Efron, Bradley and Hinkley, David V. Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika*, 65(3): 457–487, 1978.

Fréchet, Maurice. Sur l'extension de certaines evaluations statistiques au cas de petits echantillons. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute*, 11(3/4):182–205, 1943.

Grant, James DE and Vickers, JA. Block diagonalization of four-dimensional metrics. *Classical and Quantum Gravity*, 26(23):235014, 2009.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, 2015.

Hinton, Geoffrey E., Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006.

Hotelling, Harold. Spaces of statistical parameters. *American Mathematical Society Meeting*, 1929. (unpublished. Presented orally by O. Ore during the meeting).

Huzurbazar, Vasant Shankar. Probability distributions and orthogonal parameters. *Mathematical Proceedings of the Cambridge Philosophical Society*, 46(02):281–284, 1950.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML; JMLR: W&CP 37*, pp. 448–456, 2015.

Jeffreys, Harold. *Theory of Probability*. Oxford Classic Texts in the Physical Sciences. OUP, 3rd edition, 1998. First published in 1939.

Jost, Jürgen. *Riemannian Geometry and Geometric Analysis*. Springer, 6th edition, 2011.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. In *ICLR*, 2014. arXiv:1312.6114 [stat.ML].

Kurita, Takio. Iterative weighted least squares algorithms for neural networks classifiers. *New Generation Computing*, 12(4):375–394, 1994.

Le Roux, Nicolas, Manzagol, Pierre-Antoine, and Bengio, Yoshua. Topmoumoute online natural gradient algorithm. In *NIPS 20*, pp. 849–856. Curran Associates, Inc., 2008.

Lebanon, Guy. *Riemannian geometry and statistical machine learning*. PhD thesis, CMU, 2005.

Looks, Moshe, Herreshoff, Marcello, Hutchins, DeLesley, and Norvig, Peter. Deep learning with dynamic computation graphs. In *ICLR*, 2017. arXiv:1702.02181 [cs.NE].

Marceau-Caron, Gaétan and Ollivier, Yann. Practical Riemannian neural networks. *CoRR*, abs/1602.08007, 2016.

Martens, James. Deep learning via Hessian-free optimization. In *ICML*, pp. 735–742, 2010.

Martens, James. New perspectives on the natural gradient method. *CoRR*, abs/1412.1193, 2014.

Martens, James and Grosse, Roger. Optimizing neural networks with Kronecker-factored approximate curvature. In *ICML; JMLR: W&CP 37*, pp. 2408–2417, 2015.

Montanari, Andrea. Computational implications of reducing data to sufficient statistics. *Electron. J. Statist.*, 9(2): 2370–2390, 2015.

Montavon, Grégoire and Müller, Klaus-Robert. Deep Boltzmann machines and the centering trick. In *Neural Networks: Tricks of the Trade*, pp. 621–637. Springer Berlin Heidelberg, 2nd edition, 2012.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pp. 807–814, 2010.

Nielsen, Frank. Cramér-Rao lower bound and information geometry. *CoRR*, abs/1301.3578, 2013.

Ollivier, Yann. Riemannian metrics for neural networks. *CoRR*, abs/1303.0818, 2013.

Pascanu, Razvan and Bengio, Yoshua. Revisiting natural gradient for deep networks. In *ICLR*, 2014. arXiv:1301.3584 [cs.LG].

Raiko, Tapani, Valpola, Harri, and LeCun, Yann. Deep learning made easier by linear transformations in perceptrons. In *AISTATS; JMLR W&CP 22*, pp. 924–932, 2012.

Rao, Calyampudi Radhakrishna. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Cal. Math. Soc.*, 37(3):81–91, 1945.

Raskutti, Garvesh and Mukherjee, Sayan. The information geometry of mirror descent. In *Geometric Science of Information (GSI)*, volume 9389 of *LNCS*, pp. 359–368. Springer, 2015.

Sun, Ke and Marchand-Maillet, Stéphane. An information geometry of statistical manifold learning. In *ICML; JMLR W&CP 32(2)*, pp. 1–9, 2014.

Szegedy, Christian et al. Going deeper with convolutions. In *CVPR*, 2015.

Thomas, Philip. GeNGA: A generalization of natural gradient ascent with positive and negative convergence results. In *ICML; JMLR W&CP 32*, pp. 1575–1583, 2014.

Thomas, Philip, da Silva, B. C., Dann, C., and Brunskill, E. Energetic natural gradient descent. In *ICML*, 2016.

Wager, Stefan, Wang, Sida, and Liang, Percy S. Dropout training as adaptive regularization. In *NIPS 26*, pp. 351–359. Curran Associates, Inc., 2013.

Watanabe, Sumio. *Algebraic Geometry and Statistical Learning Theory*, volume 25 of *Cambridge Monographs on Applied and Computational Mathematics*. CUP, 2009.

Williams, Virginia Vassilevska. Multiplying matrices faster than Coppersmith-Winograd. In *Annual ACM Symposium on Theory of Computing*, STOC'12, pp. 887–898, 2012.

Zegers, Pablo. Fisher information properties. *Entropy*, 17: 4918–4939, 2015.