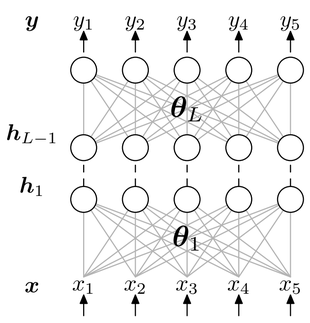
Learning on High-dimensional Neuromanifolds with Relative Natural Gradients

Ke Sun† Frank Nielsen†⋆

†École polytechnique ⋆Sony CSL

Introduction



The **Fisher Information Metric** (FIM) of a **multilayer perceptron** (MLP)

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

is

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

where $l_i(\Theta) = \ln p(y \mid x_i, \Theta)$ denotes the conditional log-likelihood function wrt x_i.

This $g(\Theta)$ serves as a **Riemannian metric** on a **neural manifold** $\mathcal{M} = \{\Theta\}$ (the parameter space of MLP) giving us an intrinsic geometric difference measure

$$\langle \delta\Theta, \delta\Theta \rangle_{g(\Theta)} = \delta\Theta^\top g(\Theta) \delta\Theta,$$

where $\delta\Theta$ denotes a tiny movement on $\mathcal{M}$.

At each time t, **Natural Gradient Descent** minimizes wrt $\delta\Theta$

$$\underbrace{L(\Theta_t + \delta\Theta)}_{\text{loss function}} + \underbrace{\frac{1}{2\gamma} \langle \delta\Theta, \delta\Theta \rangle_{g(\Theta_t)}}_{\text{cost of the movement}} \approx L(\Theta_t) + \delta\Theta^\top \bigtriangledown L(\Theta_t) + \frac{1}{2\gamma} \delta\Theta^\top g(\Theta_t) \delta\Theta,$$

where L is the loss, $\gamma > 0$ is a learning rate, giving a learning step

$$\delta\Theta_t = -\gamma g^{-1}(\Theta_t) \bigtriangledown L(\Theta_t).$$

Advantages: invariant to the choice of the coordinate system; free from plateaus; achieve Fisher efficiency in online learning; $\cdots$
Dis-advantages: expensive, sometimes infeasible, to compute $g^{-1}(\Theta_t)$

Key Insights

- Separate the learning system into subsystems (e.g. layers in MLP)
- Each subsystem has a **local information geometry** characterizing the intrinsic measurements of the corresponding subsystem
- The local information geometry is **dynamic** wrt some hidden variables h. Variations of h will cause variations of the geometric structure
- Learning is a process where a set of collaborative learners move on their sub-manifolds, and the geometries of these sub-manifolds are evolving with the system.

Results

A **hyperbolic tangent neuron** with input x, weights w, stochastic output $y \in \{-1, 1\}$ so that $p(y = 1) = (1 + \tanh(w^\top x))/2$ has the RFIM

$$g^y(w \mid x) = \left(1 - \tanh^2(w^\top x)\right) x x^\top,$$

where the bias term is omitted for simplicity.
A **PReLu neuron** with continuous output y so that

$$p(y \mid w, x) = G(y \mid \mathtt{relu}(w^\top x), \sigma^2), \quad \mathtt{relu}(t) = \begin{cases} t & \text{if } t \geq 0 \\ \iota t & \text{if } t < 0. \end{cases} \quad (0 \leq \iota < 1)$$

has the RFIM

$$g^y(w \mid x) = \frac{1}{\sigma^2} \left[\iota + (1 - \iota) \mathtt{sigm}\left(\frac{1 - \iota}{\omega} w^\top x \right) \right]^2 x x^\top,$$

where $\mathtt{sigm}(t) = 1/(1 + \exp(-t))$, and $\omega > 0$ is a hyper-parameter.
The RFIMs of other types of neurons, one linear layer, one non-linear layer, a soft-max layer, two continuous layers, all have **simple closed form solution**.

Definition (RFIM)

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

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

or simply $g^h(\theta)$, corresponding to the estimation of θ based on observations of h given θ_f.

The geometry of θ is **relative** to the choice of the reference θ_f and response h.

A Problem

FIM becomes more and more difficult to compute as the learner's structure becomes large (deep) and dynamic.

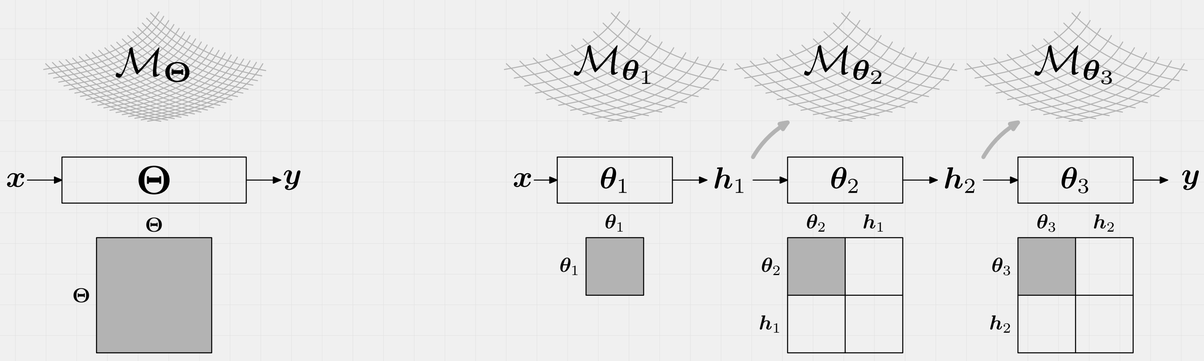
Traditional Solutions

Approximate FIM of the big learning machine.

Our Simple Solution

Compute FIMs of sub-systems without approximation!

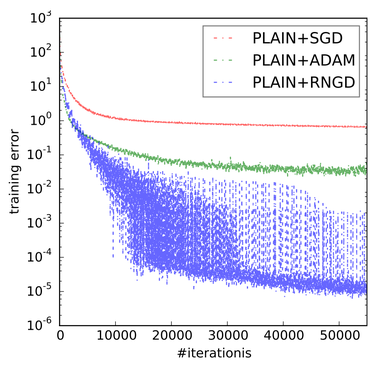
$$p(y \mid \Theta, x) = \sum_{h_1} \sum_{h_2} p(h_1 \mid \theta_1, x) \; p(h_2 \mid \theta_2, h_1) \; p(y \mid \theta_3, h_2)$$



The FIM of each subsystem θ_i is computed by integrating out the variations of the subsystem output h_i instead of integrating out the final output y.

Empirical Evidence

- We are telling, separately, each subsystem to be efficient
- Does this guarantee efficient learning of the whole system?



(Note that y-axis is log scale)

- Plain MLP on MNIST with `relu` neurons
- Shape of MLP: 784-64-64-64-10
- Relative natural gradient descent (RNGD) is **based on closed form RFIMs**
- Objective: sharper learning curve (disregarding overfitting as one may use early stopping)
- Results:
  1. RNGD learns faster in terms of #iterations, however each iteration is still very expensive
  2. RFIM is non-smoothly updated every 100 iterations, causing jitters in the learning curve

RFIM vs Diagonal FIM

Making the diagonals blocks of FIM to be identity have no control on the system variation in one learning step, due to the off-diagonal blocks.
Making the RFIMs to be identity can control the variations of each sub-system thus the whole system in one learning step.

RFIM vs FIM

RFIM suffers less from singularity, because it considers the internal stochasticity of the MLP. The FIM of MLP only considers the final output to be stochastic and is likely to be singular especially with small sample size.

For more details see `http://arxiv.org/abs/1606.06069`

Contact: sunk.edu@gmail.com nielsen@lix.polytechnique.fr