Principles of Riemannian Geometry in Neural Networks

Michael Hauser, Asok Ray
Pennsylvania State University

Presented by Chenyang Tao

Nov 16, 2018
Brief Summary

Goal

- This study deals with neural networks in the sense of geometric transformations acting on the coordinate representation of the underlying data manifold which the data is sampled from.

Contributions

- $C^k$ neural net and ResNet
- Understanding Riemannian metric tensor learned by NN
- Alternative Lie Group theory for the metric tensor
- Backpropagation as Lie Group actions
Pros:
- R3: The paper presents a very interesting framework for working with neural network representations.

Cons (that were taken out of context):
- R1: There need more discussions and comparisons in the derivations of the theory and in the experiments.
- R3: The paper is not always completely clear; experiments are a bit disappointing; does not make practical application easy.

Why it got in?
- R2: Although my understanding of the technical arguments are limited, I believe that it could be of great interest to the NIPS community.

(Heavy notations. So if you got confused, don’t feel bad.)
Getting Prepared

Notations

- $x^{(l)}$, $\phi^{(l)}$: $l$-th layer, coordinate transformation, etc.
- $A^a_b$: tensor index (just treat it as $A_{ab}$ for matrices)

Some concepts/assumptions

- $M$ is a topological manifold locally homeomorphic to $\mathbb{R}^{\dim M}$
- The homeomorphism $x : U \to x(U) \subseteq \mathbb{R}^{\dim M}$ is called a coordinate system on $U \subseteq M$
- A feedforward network is considered as coordinate transformations: $\phi^{(l)} : x^{(l)}(M) \to (\phi^{(l)} \circ x^{(l)})(M)$
  - Some activations functions like ReLU are not bijection therefore not a proper coordinate transformation
### Einstein notation

- Just put back the summation
  - Einstein $\Rightarrow \alpha_i z_i = \sum_i \alpha_i z_i \leftarrow$ Regular

### Metric tensor $g(x)$

- Think of it as a localized Mahalanobis distance
  - $ds = \sqrt{\sum_{ij} g_{ij}(x) \, dx_i \, dx_j}$
Differentiable Neural Networks

ResNet as discretized dynamical systems

- $x^{(l+1)} = x^{(l)} + f_l(x^{(l)}) \Delta l$
- $\delta x^{(l)} := x^{(l+1)} - x^{(l)} = f_l(x^{(l)}) \Delta l$
- $k^{th}$ order differentiable smoothness
  $\delta^2 x^{(l)} := x^{(l+1)} - 2x^{(l)} + x^{(l-1)} = f_l(x^{(l)}) \Delta^2$
- Autonomous $\frac{dx}{dl} := f(x(l))$, non-autonomous $\frac{dx}{dl} := f(x(l), l)$

$C^k$ neural networks

- $C^0$ (MLP): $x^{(l)} = f_l(x^{l-1})$ ; $C^1$ (ResNet): $x^{(l)} = x^{(l-1)} + f_l(x^{(l)}) \Delta l$
- $C^2$ (Momentum ResNet): $x^{(l+1)} = x^{(l)} + f_l(x^{(l)}) \Delta l^2 + \delta x^{(l-1)}$
  - $\delta x^{(l-1)} = x^{(l)} - x^{(l-1)}$
A Riemannian manifold \((M, g)\) is a real smooth manifold \(M\) with an inner product, defined by the psd metric tensor \(g\), varying smoothly on the tangent space of \(M\).

The metric in the output coordinates is assumed to be Euclidean: \(g(x^{(L)})_{a_L b_L} := \eta_{a_L b_L}\)

Metric tensor transforms:
\[
g(x^{(l)})_{a_l b_l} = \left(\frac{\partial x^{(l+1)}}{\partial x^{(l)}}\right)^{a_l+1} \cdot \left(\frac{\partial x^{(l+1)}}{\partial x^{(l)}}\right)^{b_l+1} \cdot g(x^{(l+1)})_{a_{l+1} b_{l+1}}
\]

\[
g(x^{(l)})_{a_l b_l} = \prod_{l'=l}^{L-1} \left[\left(\frac{\partial x^{(l'+1)}}{\partial x^{(l')}}\right)^{a_{l'+1}} \cdot \left(\frac{\partial x^{(l'+1)}}{\partial x^{(l')}}\right)^{b_{l'+1}} \cdot \eta_{a_L b_L}\right]
\]
A concrete example with ResNet

- If the network is taken to be the residual net

\[
\left( \frac{\partial x^{(l+1)}}{\partial x^{(l)}} \right)^{a_{l+1}} \cdot a_l = \delta^{a_{l+1}} \cdot + \left( \frac{\partial f_l(x^{(l)})}{\partial x^{(l)}} \right)^{a_{l+1}} \cdot \Delta l
\]

- 

\[
P^{a_l} \cdot a_l := \prod_{l'=l}^{L-1} \left[ \delta^{a_{l'}+1} \cdot + \left( \frac{\partial f_{l'}(z^{(l'+1)})}{\partial z^{l'+1}} \right)^{a_{l'+1}} \cdot e_{l'+1} \cdot \left( \frac{\partial z^{(l'+1)}}{\partial x^{(l')}} \right)^{e_{l'+1}} \cdot a_{l'} \right]^{a_{l+1}}
\]

where 

\[
z^{(l+1)} := W^{(l)} x^{(l)} + b^{(l)}
\]

\[
ds^2 = \eta_{ab} P^{a_l} \cdot P^{b_l} \ d x^{a_l} d x^{b_l}
\]
Metric Tensor Visualization

layer 0

layer 1

layer 2

layer 3

layer 4

layer 5
As $L \to \infty$, the infinite product converges in the limit.

For varying layerswise dimension, manifold can be submersed and immersed into lower and higher dimensional spaces so long as the rank of the pushforward Jacobian is constant for every $p \in M$.

The work is to show the coordinate representation of the metric tensor can be backpropagated through to the input so that distance can be measured in the input coordinates.
My remarks

- Using the metric induced, we can define the geodesics $d_{(M,g_0)}(x, y)$ to define similarities in the input space;
- This differs from measuring similarity in the *feature space*, e.g., $d_{(M,g_0)}(x, y) \neq \|x^{(L)} - y^{(L)}\|$.
- Yet it is not computationally feasible to compute the geodesic distance with the induced metric.
Lie Group Actions on the Metric Fibre Bundle

Definitions (Principal fiber bundle)

- A bundle \((E, \pi, M)\) is called a principle \(G\)-bundle if
  1. \(E\) is equipped with a right \(G\)-action \(< : E \times G \to E\)
  2. The right \(G\)-action \(<\) is free (no fixed point)
  3. \((E, \pi, M)\) is (bundle) isomorphic to \((E, \rho, E/G)\) where the surjective projection map \(\rho : E \to E/G\) is defined by \(\rho(\epsilon) := [\epsilon]\) as the equivalent class of points of \(\epsilon\)

Definitions (Associated fibre bundle)

- Given a \(G\)-principle bundle and a smooth manifold \(F\) on which exists a left \(G\)-action \(\triangleright : G \times F \to F\), the associated fibre bundle \((P_F, \pi_F, M)\) is defined as follows:
  1. Let \(\sim_G\) be the relation on \(P \times F\) defined as follows:
     \[(p, f) \sim_G (p', f') \iff \exists h \in G : p' = p < h \quad \text{and} \quad f' = h^{-1} \triangleright f\]
     and thus \(P_F := (P \times F) / \sim_G\)
  2. Define \(\pi_F := P_F \to M\) by \(\pi_F([[(p, f)]])) := \pi(p)\)
Neural network actions on the manifold $M$ are a (layerwise) sequence of left $G$-actions on the associated (metric space) fibre bundle. Let $d = \dim M$

- The structure group $G$ is the general linear group of dimension $d$ over $\mathbb{R}$: $G = GL(d, \mathbb{R}) := \{\phi : \mathbb{R}^d \to \mathbb{R}^d | \det \phi \neq 0\}$
- The principle bundle $P$ is taken to be the frame bundle: $P = LM := \bigcup_{p \in \mathcal{M}} L_p M$
- The right $G$-action $\lhd : LM \times GL(d, \mathbb{R}) \to LM$ is defined by:
  $e \lhd h = (e_1, \ldots, e_d) \lhd h := (h_{a_1}^{a_1} e_{a_1}, \ldots, h_{a_d}^{a_d} e_{a_d})$
- The fibre $F$ in the associated bundle is the metric tensor space $F = (\mathbb{R}^d)^* \times (\mathbb{R}^d)^*$, where $*$ denotes the cospace, and the left $G$-action $\triangleright : GL(d, \mathbb{R}) \times F \to F$ is defined as the inverse of the left (right) $G$-action $(h^{-1} \triangleright g)_{a_ib_l} := (g \triangleright h)_{a_ib_l} = g_{a_{l+1}b_{l+1}} h_{a_l}^{a_{l+1}} h_{b_l}^{b_{l+1}}$

\[
\left(h_{0}^{-1} \triangleright h_{1}^{-1} \triangleright \cdots \triangleright h_{L}^{-1} g \right)_{a_0b_0} = (h_{0}^{-1} \bullet h_{1}^{-1} \bullet \cdots \bullet h_{L}^{-1}) \triangleright g_{a_Lb_L} = \\
\prod_{l'=L-1}^{0} \left(h_{a_l+1}^{a_l'} \cdot h_{b_l+1}^{b_l'} \cdot g_{a_Lb_L} \right)
\]
Error backpropagation as a sequence of right Lie Group actions

- Standard backpropagation for error $E$ wrt layer weight $W^{(l-1)}$

\[
\frac{\partial E}{\partial W^{(l-1)}} = \left( \frac{\partial E}{\partial x^{(L)}} \right)_{a_L} \prod_{l' = L-1}^{l} \left( \frac{\partial x^{(l'+1)}}{\partial x^{l'}} \right)^{a_{l'+1}} \cdot \left( \frac{\partial x^{(l)}}{\partial W^{(l-1)}} \right)^{a_l}
\]  

- As such, error backpropagation is a sequence of right $G$-actions $\prod_{l' = L-1}^{l} \left( \frac{\partial x^{(l'+1)}}{\partial x^{l'}} \right)^{a_{l'+1}} \cdot \left( \frac{\partial x^{(l)}}{\partial x^{L}} \right)^{a_L}$ on the output frame bundle.
Only experiments on spiral disentanglement is considered

- $C^k$ networks
- Effect of batch-size
- Effect of network depth
$C^k$ Neural Networks

(a) A $C^0$ network with sharply changing layer-wise particle trajectories.

(b) A $C^1$ network with smooth layer-wise particle trajectories.

(c) A $C^2$ network also exhibits smooth layer-wise particle trajectories.

(d) A combination $C^0$ and $C^1$ network, where the identity connection is left out in layer 6.
(a) A batch size of 300 for untangling data. As early as layer 4 the input connected sets have been disconnected and the data are untangled in an unintuitive way. This means a more complex coordinate representation of the data manifold was learned.

(b) A batch size of 1000 for untangling data. Because the large batch size can well-sample the data manifold, the spiral sets stay connected and are untangled in an intuitive way. This means a simple coordinate representation of the data manifold was learned.
Effect of Network Depth

(a) A 10 layer $C^1$ network struggles to separate the spirals and has 1% error rate.

(b) A 20 layer $C^1$ network is able to separate the spirals and has 0% error rate.

(c) A 40 layer $C^1$ network is able to separate the spirals and has 0% error rate.