Learning Invariant Riemannian Geometric Representations Using Deep Nets

Suhas Lohit and Pavan Turaga
Arizona State University

Invariants in Computer Vision and Non-Euclidean Constraints

- Non-Euclidean constraints arise often in computer vision because of invariance requirements (e.g. illumination, deformation etc.) and the task at hand (e.g. saliency detection)
- These non-Euclidean constraints mean that conventional vector space machine learning does not apply directly.
- Several invariance constraints are expressible in the language of Riemannian geometry.

Measuring invariances in deep-networks

"Measuring Invariances in Deep Networks", Ian J. Goodfellow, Quoc V. Le, Andrew M. Saxe, Honglak Lee, Andrew Y. Ng, NIPS 2009.

"A surprising finding in our experiments with visual data is that stacked autoencoders yield only modest improvements in invariance as depth increases."

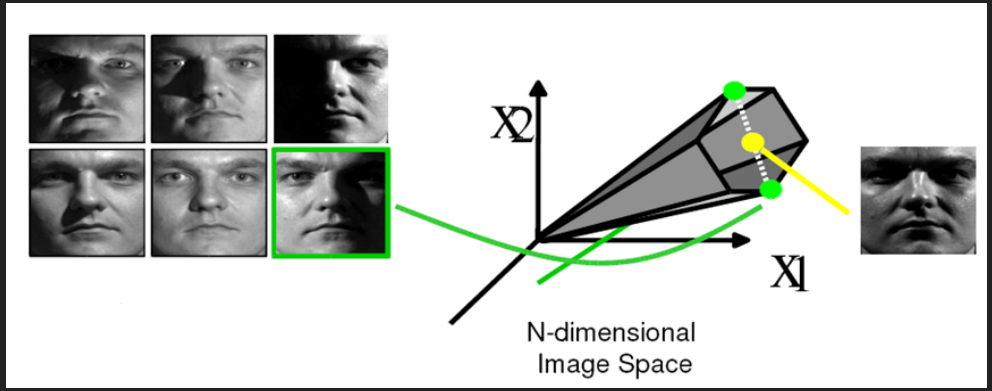
"Another interesting finding is that by incorporating sparsity, networks can become more invariant."

"….. aproaches to achieving invariance such as max-pooling and weight- sharing…. not obvious how to extend these explicit strategies to become invariant to more intricate transformations like large-angle out-of-plane rotations and complex illumination changes…."

Some examples from past work

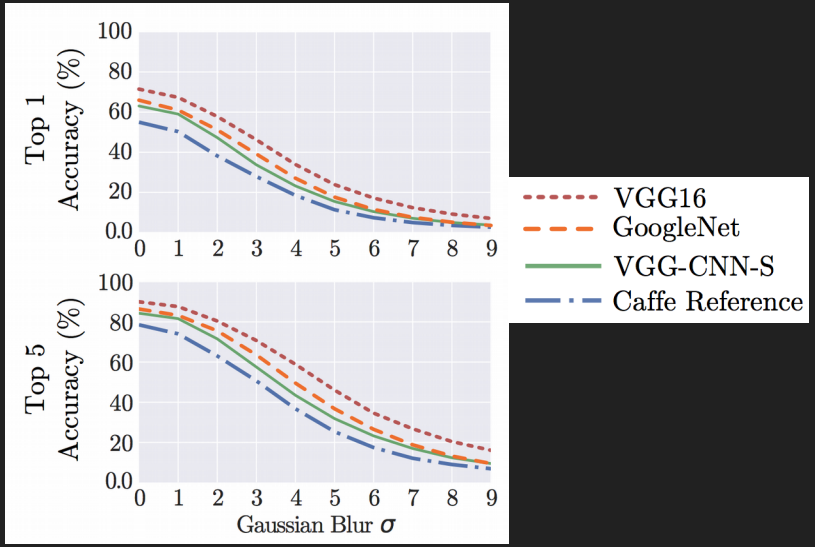
Illumination invariance

- Current approach: data augmentation by sampling from PCA space of RGB pixels in training set.
  - A.Krizhevsky, I. Sutskever and G.E.Hinton, “Imagenet classification with deep convolutional neural networks”, In Advances in neural information processing systems 2012 (pp. 1097-1105).
- Several known results relating to illumination modeling in natural images. How do we leverage these?



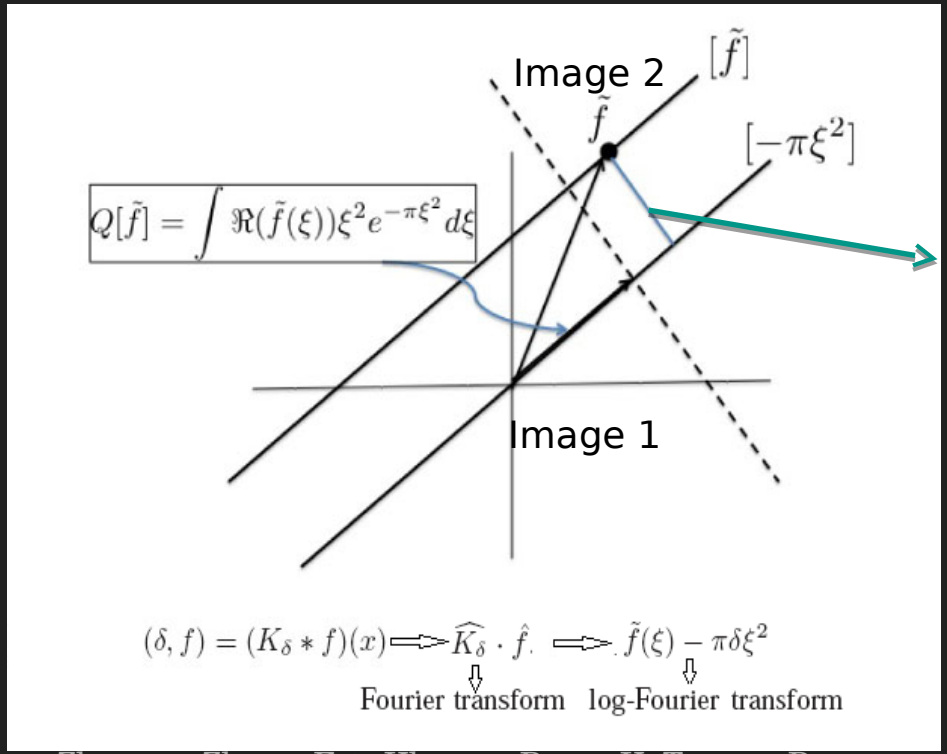
What Is the Set of Images of an Object under All Possible Illumination Conditions? P. N. Belhumeur, and D. J. Kriegman, IJCV 1998.

Blur invariance



"Understanding How Image Quality Affects Deep Neural Networks", Samuel Dodge, Lina Karam, Proceedings of the Conference on the Quality of Multimedia Experience (QoMEX), June 6-8, 2016

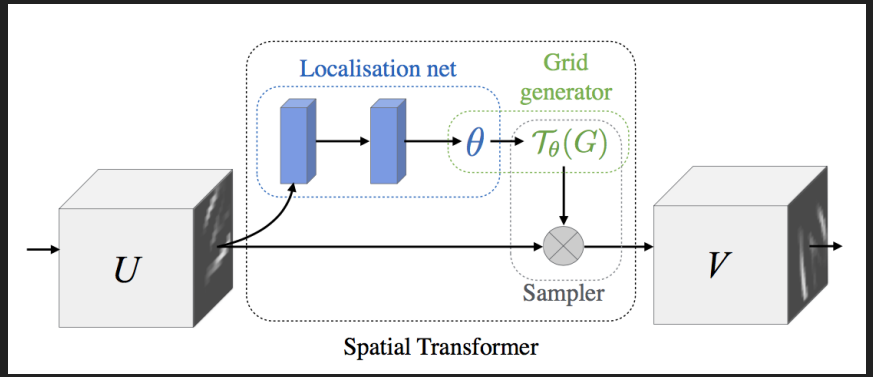
Under Gaussian blur, orbits of images are straight lines in log-Fourier space



Zhengwu Zhang, Eric Klassen, Pavan K. Turaga, Rama Chellappa, Anuj Srivastava, "Blurring-invariant Riemannian metrics for comparing signals and images". ICCV 2011: 1770-1775

Mis-alignment invariance

- New solutions incl. Spatial Transformer Networks
- 2D geometric normalization is learned end-to-end with no augmentation, no template.
- Converges to whatever alignment results in increased recognition accuracy.



Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu:Spatial Transformer Networks. NIPS 2015: 2017-2025

Video-related examples

- Shift/initial condition invariance in linear dynamical modeling
- Topological-invariance in non-linear dynamical modeling
- More on these in my next talk

How to train deep neural networks that respect Riemannian constraints?

Related Work

- Deep learning for non-Euclidean constraints is gaining interest, mainly enforcing geometry at the input:
  - Graphs (e.g. Bruna et al. 2014)
  - 3D shapes as Riemannian manifolds (Masci et al. 2015)
  - Points on Grassmann manifold (Huang et al. 2016a), Lie groups (Huang et al. 2016b) and SPD matrices (Huang et al. 2017)
- Some recent work on predicting SE(3) elements which form a Lie group by mapping to the Lie algebra in order to satisfy constraints (Byravan and Fox 2016, Clark et al. 2017)

Two examples we consider

- Face to illumination subspace
  - Estimate illumination—invariant representation from a single face image, while explicitly enforcing Grassmannian geometry of space of subspaces.

- Activations to probability-density functions
  - How to use geometry of density functions, while mapping activation layer to pdfs, and while training the deep net. We use square-root representation of pdfs, thereby pdf space becomes a sphere.

Differences from conventional framework

1. The loss function

- Since we regress to manifold-valued data, the loss function which is usually a simple L-2 function, should use the geometry of the output space. In other words, loss should be based on the geodesic distance.
- However, it may always not be possible to derive the geodesic distance function in closed form or it may not be able to implement it easily as a differentiable layer in a neural network to put it in the framework of standard backpropagation.

Differences from conventional framework

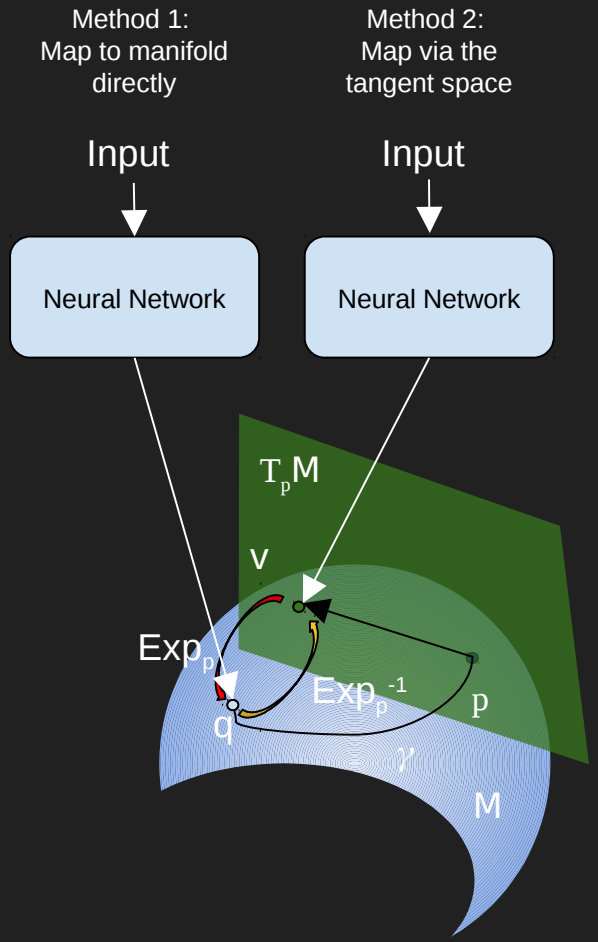
2. Satisfying manifold constraints exactly

Unless constraint satisfaction can be implemented as a differentiable layer it is not immediately clear how to satisfy the constraints exactly so that the outputs of the network will lie on the manifold.

Our solutions

1. Map to manifold directly

For “simple” manifolds, we propose directly mapping to the manifold. By “simple”, we mean a manifold where constraints can be satisfied with differentiable layers and have a closed form geodesic distance function that is differentiable. E.g. n-1 sphere

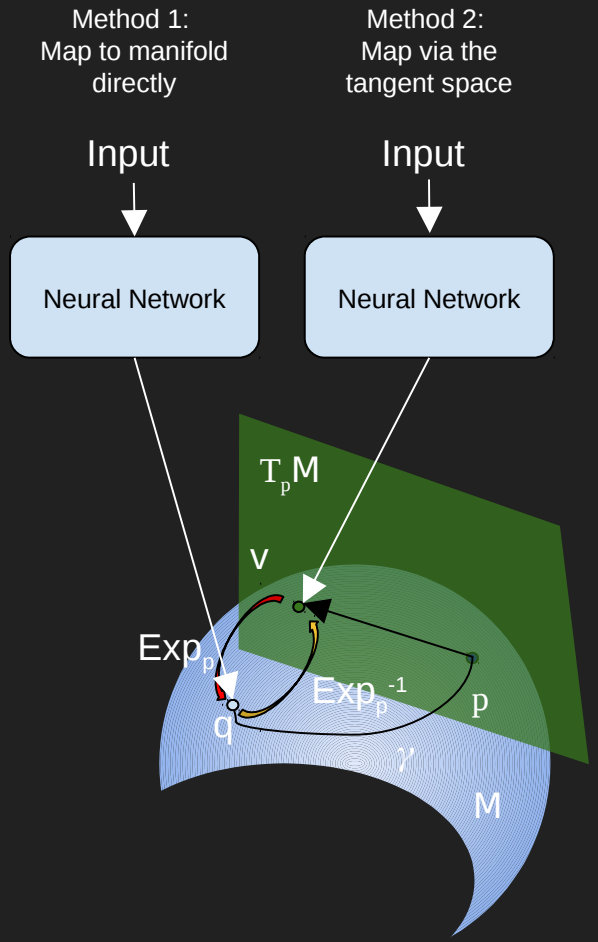


Our solutions

2. Map via tangent space

For other manifolds, we propose first learning to map to a tangent space. For manifolds like Stiefel and Grassmann, it is easier to map to the tangent space while satisfying constraints exactly. The Euclidean loss function is more meaningful here. We assume that all data points are close to pole of the tangent space.

At test time, we first map to the tangent space using the network and then employ the exponential map



1.Subspace Prediction Regression on the Grassmann manifold

$$\mathcal{G}_{n,d} = \{[U]\}, [U] = \{UQ | U \in \mathbb{R}^{n \times d}, U^T U = I \text{ and } Q \text{ is orthogonal}\}$$

Illumination Subspace Prediction

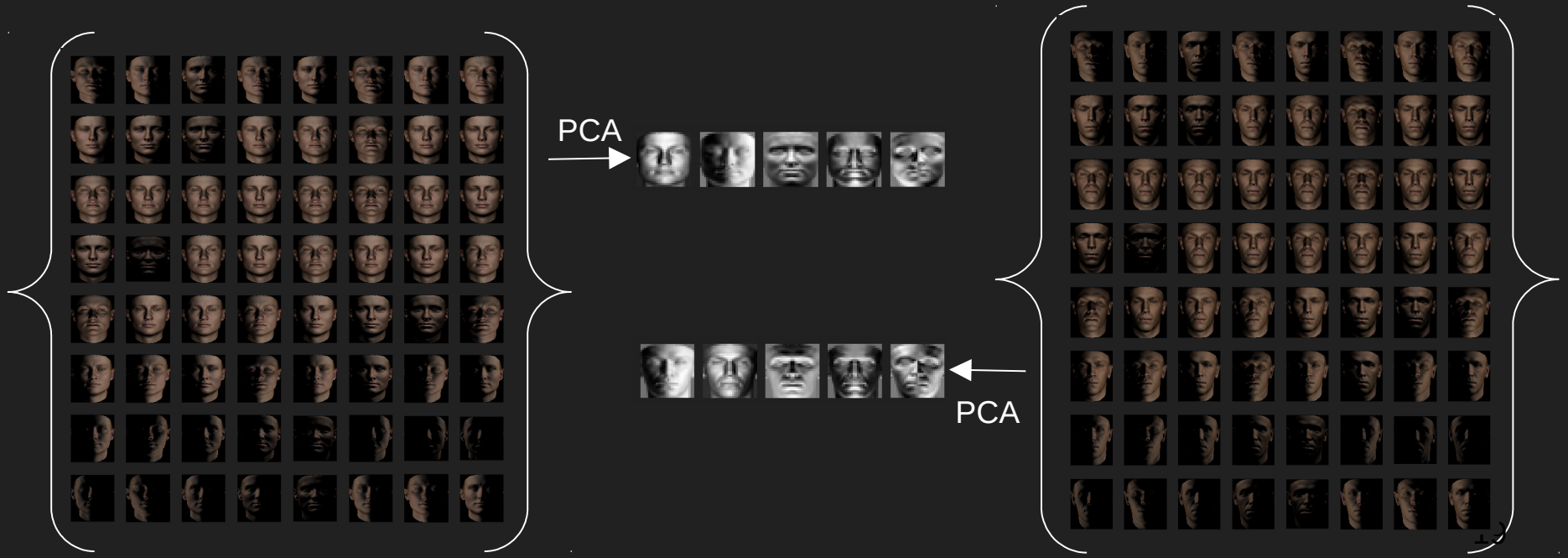
- Illumination subspaces for faces: For a given human face, the set of images obtained by varying the illumination directions, lies close to a low-dimensional subspace. And the eigenfaces follow specific patterns (Hallinan 1994)
- Top eigenfaces, obtained using the PCA of the image set, serve as a basis for the illumination subspace
- Subspaces are points on the Grassmann manifold
- We design an ill-posed inverse problem for illustration of regression on the Grassmannian: Given a face image of an unseen subject under unknown illumination, predict the illumination subspace.

Dataset for Face→Illumination Subspace

- Synthetic dataset using 3D face models (Paysan et al.) of 250 subjects under 64 illumination conditions. Illumination subspace is calculated using PCA

Dataset for Face→Illumination Subspace

- Synthetic dataset using 3D face models (Paysan et al.) of 250 subjects under 64 illumination conditions. Illumination subspace is calculated using PCA



Illumination Subspace Prediction - Mapping to the Grassmann Manifold?

- If we regress to eigenvectors U, it is a mapping to the Stiefel manifold
- We argue that the better representation is Grassmann because it is invariant to the sign flips of the eigenfaces and their permutations
- **Baseline**: Directly regress the eigenfaces U with Euclidean distance as the loss function; fails because of conflicting information in the groundtruth data in the training set (sign flips and permutations)
- Ou preferred representation for regression is UU^T which has size n^2

Illumination Subspace Prediction - Mapping to the Grassmann Tangent Space

- We choose a pole based on the training set (e.g. Frechet mean)
- Calculate the tangent vectors corresponding to groundtruth outputs using the logarithm map of the Grassmann manifold (Srivastava et al. 2004)
- This is invariant to sign flips and permutations by design
- The tangent vectors at the identity matrix have nice structure:

$$X = \begin{bmatrix} \mathbf{0}_d & A \\ -A^T & \mathbf{0}_{n-d} \end{bmatrix}, A \in \mathbb{R}^{d \times (n-d)}$$

- Completely deter learn a mapping using a neural network to regress to the A matrix given a face image

Illumination Subspace Prediction - Mapping to the Grassmann Tangent Space

- Loss function is the Euclidean distance on the tangent space
- At test time, we initially get the Â, hence the tangent vector. We can compute the desired point on Grassmann manifold using the Exponential map (Srivastava et al. 2004)
- In both cases, the neural network has nearly the same architecture (3 conv -> 2 fc layers)

Results

Performance metric: Mean geodesic distance (D_G) between the predicted subspace and the groundtruth subspace over the test set (lower is better)

$$D_G(U_1, U_2) = \sqrt{\sum_{i=1}^{d} \theta_i^2}, \quad \theta_i' s \text{ are the arccosine of the top singular values of } U_1^T U_2$$

Subspace Dim (d)	Baseline	GrassmannNet-TS	
		Pole = P1	Pole = P2
3	0.6613	0.3991	**0.3953**
4	1.0997	0.5489	**0.5913**
5	1.4558	0.8694	**0.6174**

P1: Su
P2: Fr

Results

Input	Ground-truth PCs	Output of baseline n/w	Output of GrassmannNet-TS
		 $D_G = 1.6694$	 $D_G = 0.7006$
		 $D_G = 1.2998$	 $D_G = 0.7238$
		 $D_G = 0.7797$	 $D_G = 0.5966$
		 $D_G = 1.5355$	 $D_G = 0.6170$
		 $D_G = 1.6760$	 $D_G = 0.4420$
		 $D_G = 1.6703$	 $D_G = 0.4939$

2. Image Classification
Regression on the unit hypersphere

$$\mathrm{S}^{C-1} = \{(x_1, \ldots, x_C) \in \mathbb{R}^C | \sum_{i=1}^{C} x_i^2 = 1\}$$

Image Classification - Mapping to the Hypersphere

- Many-to-one equivalence relation between final activation values and underlying pdf. Usually, projection implemented by soft-max.
- We can map probability mass functions to the positive orthant of the unit hypersphere using the square-root parametrization, which has advantages (Srivastava et al. 2007)
- Classification $\mathbf{p}_S(i) = \sqrt{\mathbf{p}_{pd}(i)}, \quad i = 1, \ldots, C$! is number of classes)

Image Classification - Mapping to the Hypersphere

- Unit norm constraint is easily satisfied with a differentiable normalization layer

$$\mathbf{p}_S = \frac{\mathbf{p}}{||\mathbf{p}||_2}, \quad \mathbf{p} \in \mathbb{R}^C, \mathbf{p}_S \in \mathrm{S}^{C-1}$$

- Loss function is based on the geodesic distance, which is available in closed-form and differentiable

$$L_{geo} = 1 - \cos(\theta), \quad \theta = D_G(\mathbf{g}_S, \frac{\hat{\mathbf{x}}}{||\hat{\mathbf{x}}||})$$

$$D_G(\mathbf{x}, \mathbf{y}) = \arccos(\mathbf{x}^T \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathrm{S}^{C-1}$$

Image Classification - Mapping to the Tangent Space of Hypersphere

- Pole of tangent space is chosen as the point on sphere, $\mathbf{u}_S$, corresponding to the uniform probability distribution
- Tangent vectors have constraints which can be satisfied using a projection layer onto the tangent space

$$T_{\mathbf{u}_S}\mathbb{S}^{C-1} = \{\xi \in \mathbb{R}^C | \mathbf{u}_S^T \xi = 0\}$$

- In the case of hypersphere, projection onto tangent space is closed form and differentiable and is implemented as a layer in the network

$$P_{\mathbf{u}_S}(\mathbf{v}) = (\mathbf{I}_n - \mathbf{u}_S \mathbf{u}_S^T)\mathbf{v}, \quad \mathbf{v} \in \mathbb{R}^C, P_{\mathbf{u}_S}(\mathbf{v}) \in T_{\mathbf{u}_S}\mathcal{S}^{C-1}$$

Image Classification - Mapping to the Tangent Space of Hypersphere

- Euclidean distance on the tangent space is used as the loss function

$$L_{tan} = ||\mathbf{g}_T - P_{\mathbf{u}_S}(\mathbf{v})||_2$$

- At test time, after the feedforward pass through the network, exponential map is used to determine the point on the hypersphere and hence the pdf (Srivastava et al. 2007, Boumal et al. 2014)

$$\exp_{\mathbf{u}_S}(\xi) = \cos(||\xi||)\mathbf{u}_S + \sin(||\xi||)\frac{\xi}{||\xi||}$$

Datasets

MNIST handwritten digit recognition

- 10 classes (0-9)
- 50000 train + 10000 test
- 28 x 28 Grayscale Images
- LeNet-5 Architecture
  - (2 conv → 2 fc → constraint satisfaction layer)

Constraint satisfaction layer depends on method:

- PDF: softmax
- Hypersphere: unit normalization
- Tangent space of hypersphere: projection layer

Datasets

CIFAR-10 object recognition

- 10 classes
- 50000 train + 10000 test
- 32 x 32 RGB Images
- CNN architecture:
  - 2 conv → 2 fc → constraint satisfaction layer

Constraint satisfaction layer depends on method:

- PDF: softmax
- Hypersphere: unit normalization
- Tangent space of hypersphere: projection layer

Results on MNIST and CIFAR-10

Mean accuracy (%), standard deviation (%), p-value ($\alpha = 0.05$) compared to baseline (softmax, cross entropy) on test set, averaged over 10 runs

Output & Loss function	MNIST	CIFAR-10
Probability Mass Function & Cross entropy	99.224 ± 0.0306	78.685 ± 0.3493
Hypersphere & Euclidean	99.263 ± 0.0479 (p=0.0437)	79.738 ± 0.4009, (p<0.0001)
Hypersphere & Geodesic	99.293 ± 0.0343 (p=0.0002)	**80.024 ± 0.5131, (p<0.0001)**
Tangent space of hypersphere & Euclidean	**99.332 ± 0.0600 (p<0.0001)**	76.047 ± 1.6225, (p<0.0001)

Conclusion and Future Work

- We have showed that deep learning architectures can be extended to nonlinear target domains, exploiting the knowledge of data geometry
- Through applications on the Grassmannian and the hypersphere, we have demonstrated that understanding geometric properties can lead to making informed choices about the loss function and exactly satisfying output constraints in a deep learning setting
- Can we extend it to problem domains where the data are spread wider from the centroid? And non-differentiable manifolds?
- Big ticket item for geometry community: provide guaranteed invariance.

References

- P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, 2009.
- N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre, et al. Manopt, a matlab toolbox for optimization on manifolds. Journal of Machine Learning Research, 15(1):1455–1459, 2014.
- J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. International Conference on Learning Representations, 2014.
- A. Byravan and D. Fox. Se3-nets: Learning rigid body motion using deep neural networks. IEEE International Conference on Robotics and Automation, 2016.
- R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. AAAI Conference on Artificial Intelligence, 2017.
- P. W. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994.
- Z. Huang, J. Wu, and L. Van Gool. Building deep networks on Grassmann manifolds. arXiv preprint arXiv:1611.05742, 2016.
- Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on Lie groups for skeleton-based action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- Z. Huang and L. Van Gool. A Riemannian network for SPD matrix learning. AAAI Conference on Artificial Intelligence, 2017.

References

- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on Riemannian manifolds. In Proceedings of the IEEE International Conference on Computer Vision workshops, pages 37–45, 2015.
- P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In Sixth IEEE International Conference on Advanced video and signal based surveillance, pages 296–301. IEEE, 2009
- A. Srivastava and E. Klassen. Bayesian and geometric subspace tracking. Advances in Applied Probability, 36(01):43– 56, 2004.
- A. Srivastava, I. Jermyn, and S. Joshi. Riemannian analysis of probability density functions with applications in vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007.
- A. Srivastava, P. Turaga, Riemannian Computing in Computer Vision, Springer 2015

Thank you