ON THE STABILITY OF DEEP NETWORKS

AND THEIR RELATIONSHIP TO COMPRESSED SENSING AND METRIC LEARNING

RAJA GIRYES AND GUILLERMO SAPIRO

DUKE UNIVERSITY

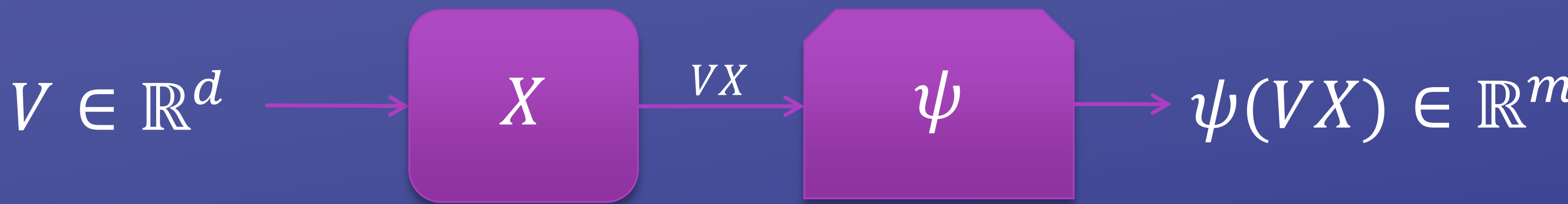
Mathematics of Deep Learning

International Conference on Computer Vision (ICCV)

December 12, 2015

DEEP NEURAL NETWORKS (DNN)

- One layer of a neural net

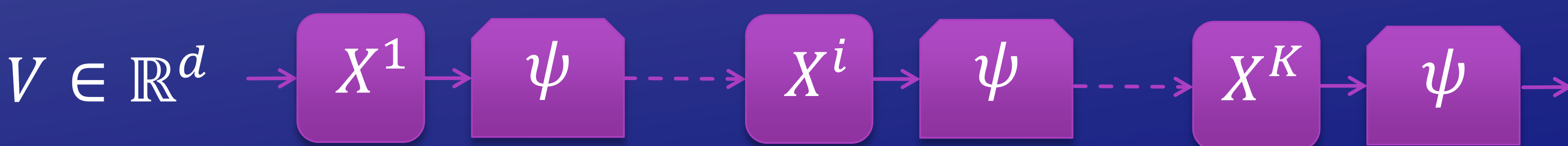


X is a linear operation

F is a non-linear function

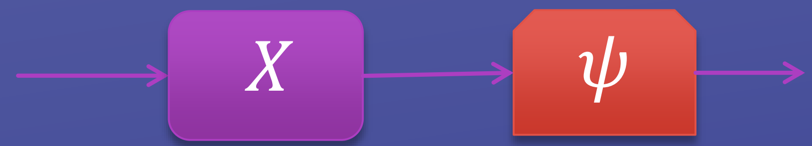
- Concatenation of the layers creates the whole net

$$\Phi(X^1, X^2, \dots, X^K) = \psi(\psi(\psi(VX^1)X^2) \dots X^K)$$



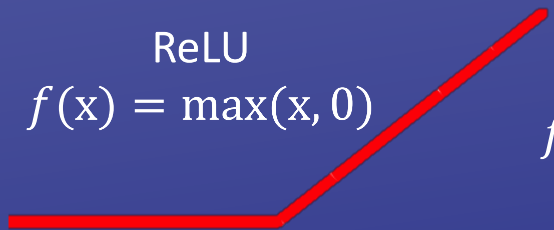
THE NON-LINEAR PART

- Usually $\psi = g \circ f$.

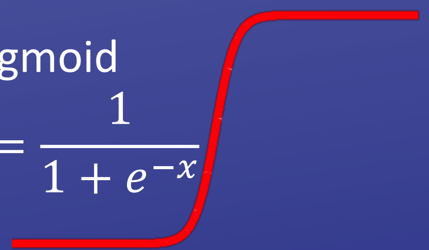


- f is the (point-wise) activation function

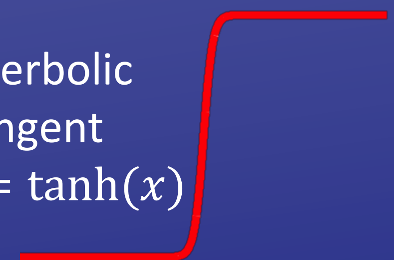
ReLU
$f(\mathrm{x}) = \max(\mathrm{x}, 0)$



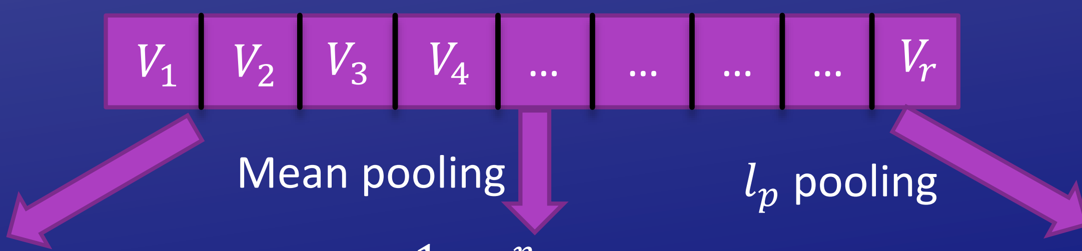
Sigmoid
$f(x) = \dfrac{1}{1 + e^{-x}}$



Hyperbolic tangent
$f(x) = \tanh(x)$



- g is a pooling or an aggregation operator.



Max pooling
$\max_i V_i$

Mean pooling
$\dfrac{1}{n}\sum_{i=1}^{n} V_i$

l_p pooling
$\sqrt[p]{\sum_{i=1}^{n} V_i^p}$

WHY DNN WORK?

What is so special with the DNN structure?

Why a local training of each layer is a good choice?

What is the role of the depth of DNN?

How many training samples do we need?

What is the role of the activation function?

What is the role of pooling?

What happens to the data throughout the layers?

SAMPLE OF RELATED EXISTING THEORY

- Universal approximation for any measurable Borel functions [Hornik et. al., 1989, Cybenko 1989].
- Depth of a network provides an exponential complexity compared to the number parameters [Montúfar. al. 2014]
- Pooling stage provides shift invariance [Bruna et. al. 2013]
- Relation of pooling and phase retrieval [Bruna et. al. 2014].
- Deeper networks have more local minima that are close to the global one and less saddle points [Saxe et. al. 2014], [Dauphin et. al. 2014] [Choromanska et. al. 2015] [Haeffele & Vidal, 2015]

Take Home Message

DNN keep the important information of the data.

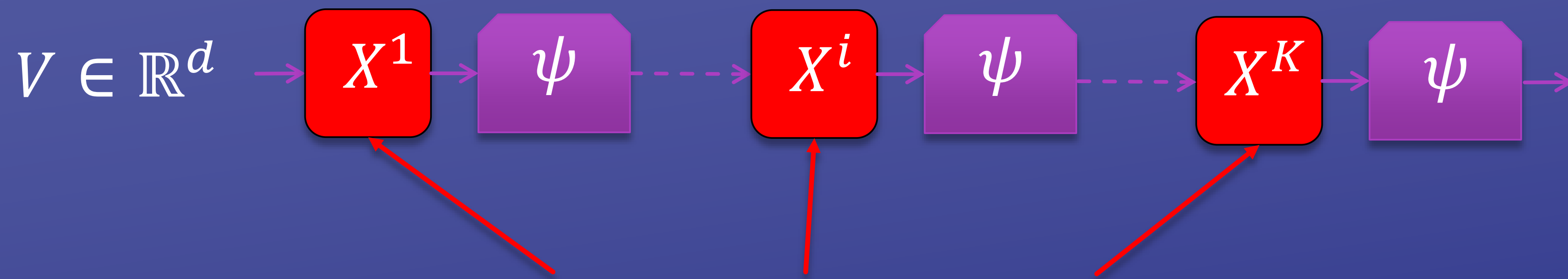
Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

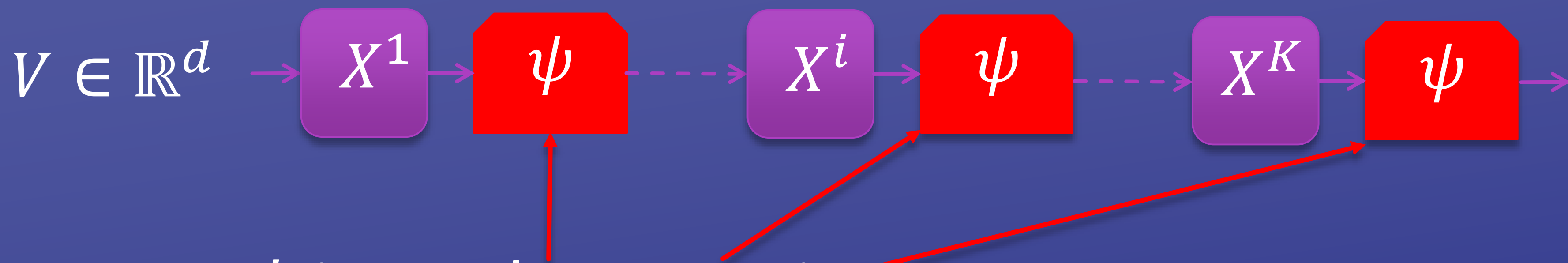
ASSUMPTIONS – GAUSSIAN WEIGHTS



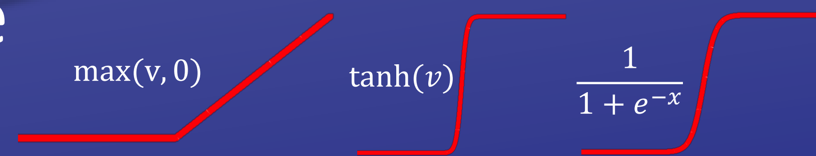
$X^i, \ldots, X^i, \ldots, X^K$ are random Gaussian matrices

- Infusion of random weights reveals internal properties of a system

ASSUMPTIONS – NO POOLING



ψ is an element wise activation function



- Pooling provides invariance [Boureau et. al. 2010, Bruna et. al. 2013].

➢ We assume that all equivalent points in the data were merged together and omit this stage.

➢ Reveals the role of the other components in the DNN.

ASSUMPTIONS – LOW DIMENSIONAL DATA

$$V \in \Upsilon \rightarrow X^1 \rightarrow \psi \dashrightarrow X^i \rightarrow \psi \dashrightarrow X^K \rightarrow \psi \rightarrow$$

Υ is a low dimensional set

- Gaussian Mixture Models (GMM)
- Low Rank Matrices
- Signals with Sparse Representations
- Low Dimensional Manifolds

Gaussian Mean Width

DNN keep the important information of the data.

Gaussian mean width is a good measure for the complexity of the data.

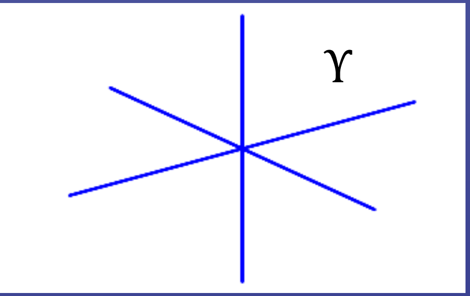
Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

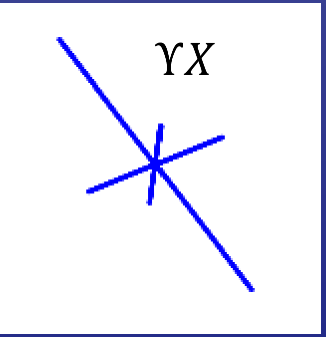
Deep learning can be viewed as a metric learning.

WHAT HAPPENS TO SPARSE DATA IN DNN?

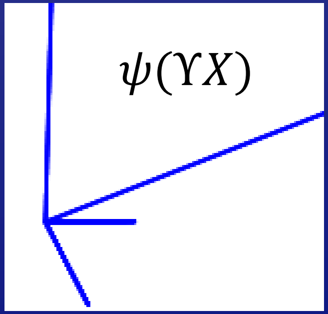
- Let Υ be sparsely represented data
  - Example: $\Upsilon = \{V \in \mathbb{R}^3 : \|V\|_0 \leq 1\}$



- $\Upsilon\mathrm{X}$ is still sparsely represented data
  - Example: $\Upsilon\mathrm{X} = \{V \in \mathbb{R}^3 : \exists W \in \mathbb{R}^3, V = XW, \|W\|_0 \leq 1\}$



- $\psi(\Upsilon\mathrm{X})$ not sparsely represented
- But is still low dimensional

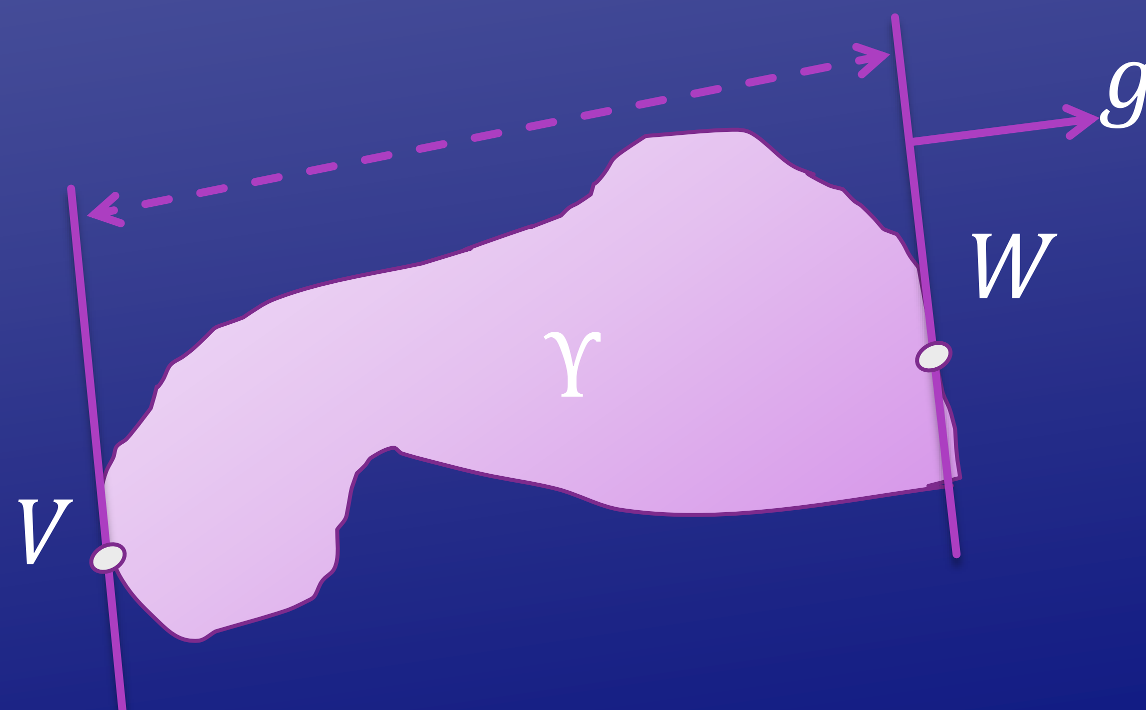


GAUSSIAN MEAN WIDTH

- Gaussian mean width:

$$\omega(\Upsilon) = E \sup_{V,W \in \Upsilon} \langle V - W, g \rangle, \qquad g \sim N(0, I).$$

The width of the set Υ in the direction of g:



MEASURE FOR LOW DIMENSIONALITY

- Gaussian mean width:

$$\omega(\Upsilon) = E \sup_{V,W\in\Upsilon} \langle V - W, g\rangle, \qquad g \sim N(0, I).$$

- $\omega^2(\Upsilon)$ is a measure for the dimensionality of the data.
- Examples:

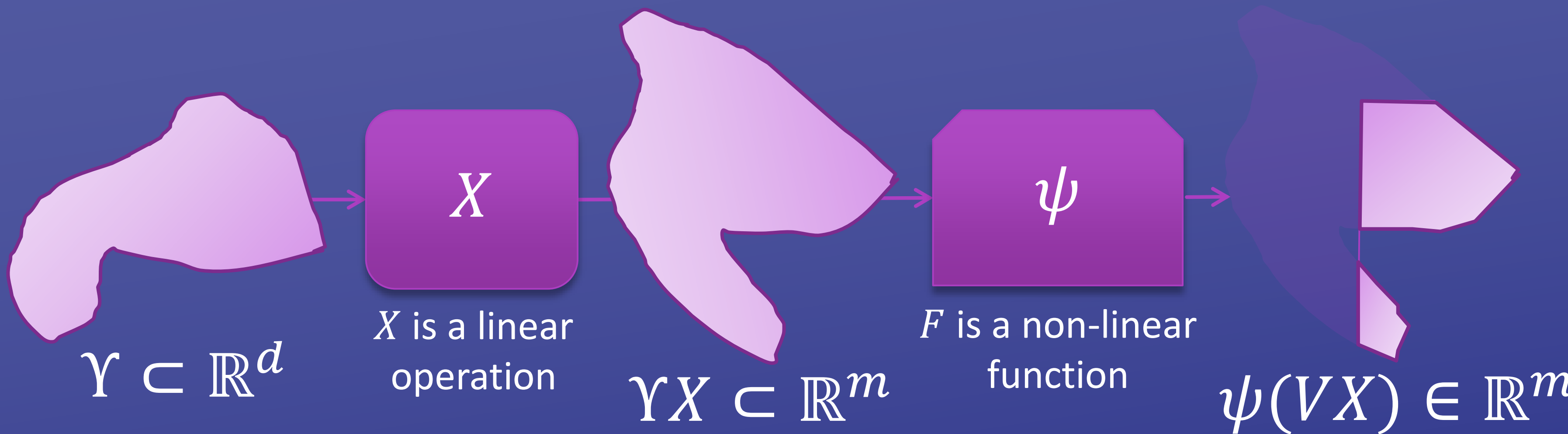
If $\Upsilon \subset \mathbb{B}^d$ is a Gaussian Mixture Model with k Gaussians then

$$\omega^2(\Upsilon) = O(k)$$

If $\Upsilon \subset \mathbb{B}^d$ is a data with k-sparse representations then

$$\omega^2(\Upsilon) = O(k \log d)$$

GAUSSIAN MEAN WIDTH IN DNN



Theorem 1: small $\frac{\omega^2(\Upsilon)}{m}$ imply $\omega^2(\Upsilon) \approx \omega^2(\psi(VX))$

Small $\omega^2(\Upsilon)$ → Small $\omega^2(\psi(VX))$

➡ It is sufficient to provide proofs only for a single layer

Stability

DNN keep the important information of the data.

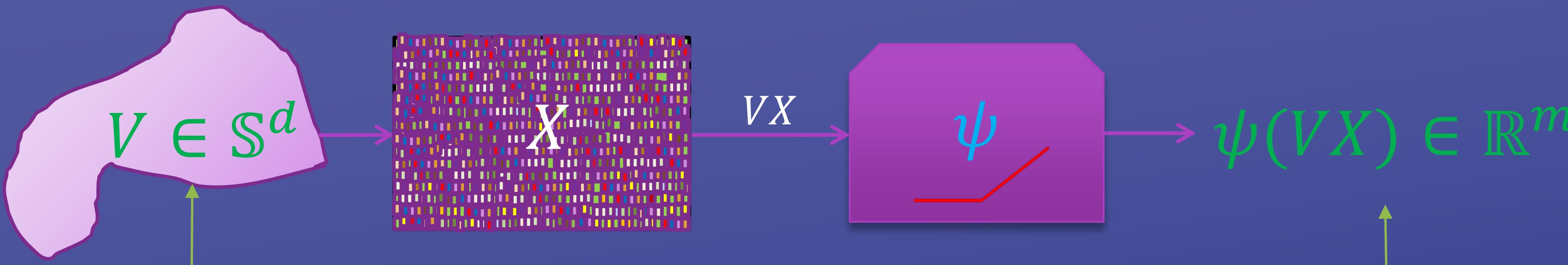
Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

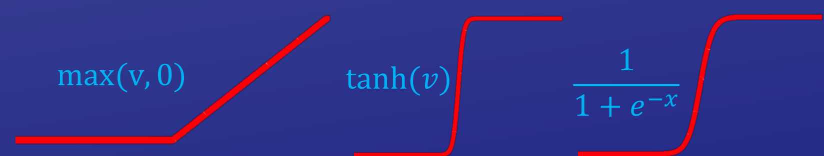
ASSUMPTIONS



$V \in \Upsilon$

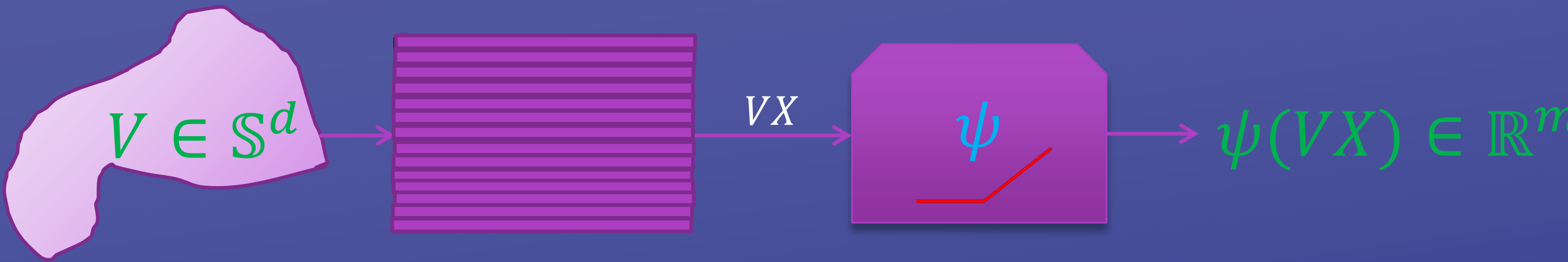
X is a random Gaussian matrix

ψ is an element wise activation function

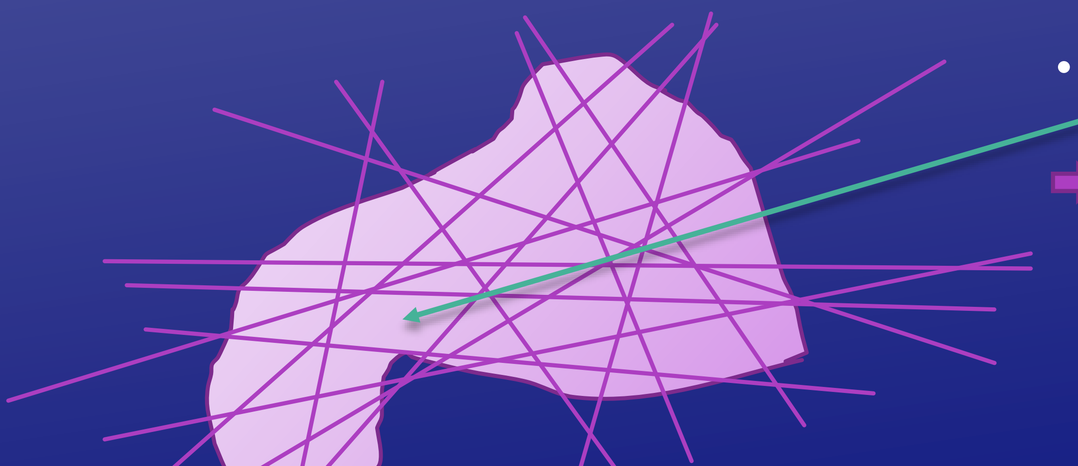


$m = O\left(\delta^{-6}\omega^2(\Upsilon)\right)$

ISOMETRY IN A SINGLE LAYER



Theorem 2: $\psi(\cdot\, X)$ is a δ-isometry in the Gromov-Hausdorff sense between the sphere $\mathbb{S}^{d-1}$ and the Hamming cube [Plan & Vershynin, 2014, Giryes, Sapiro & Bronstein 2015].

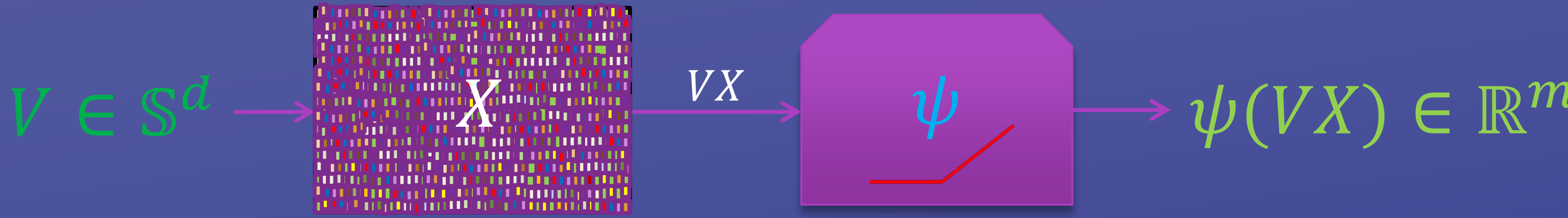


- If two points belong to the same tile then their distance $< \delta$
- Each layer of the network keeps the main information of the data

The rows of X create a tessellation of the space.

- This stands in line with [Montúfar et. al. 2014]

ONE LAYER STABLE EMBEDDING



Theorem 3: There exists an algorithm $\mathcal{A}$ such that

$$\|V - \mathcal{A}(\psi(VX))\| < O\left(\frac{\omega(\Upsilon)}{\sqrt{m}}\right) = O(\delta^3)$$

[Plan & Vershynin, 2013, Giryes, Sapiro & Bronstein 2015].

- After K layers we have an error $O(K\delta^3)$
- Stands in line with [Mahendran and Vedaldi, 2015].
- DNN keep the important information of the data

DNN with Gaussian Weights

DNN keep the important information of the data.

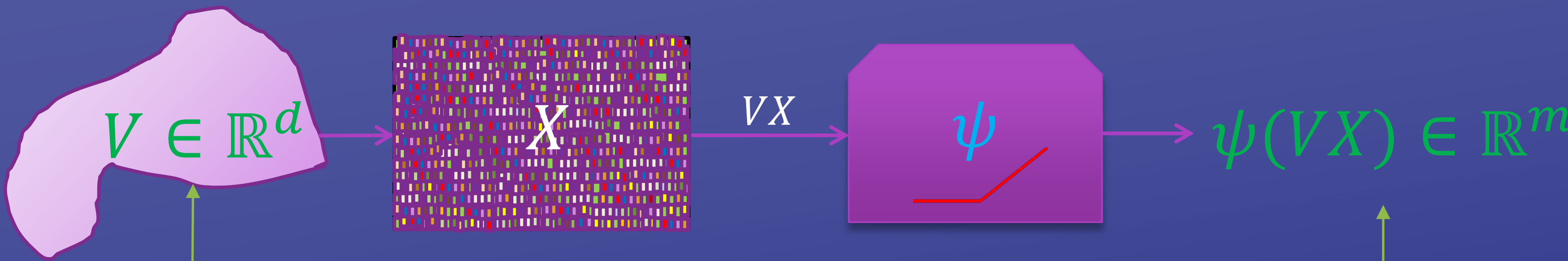
Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

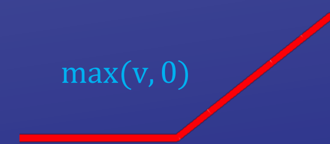
ASSUMPTIONS



$V \in \Upsilon$

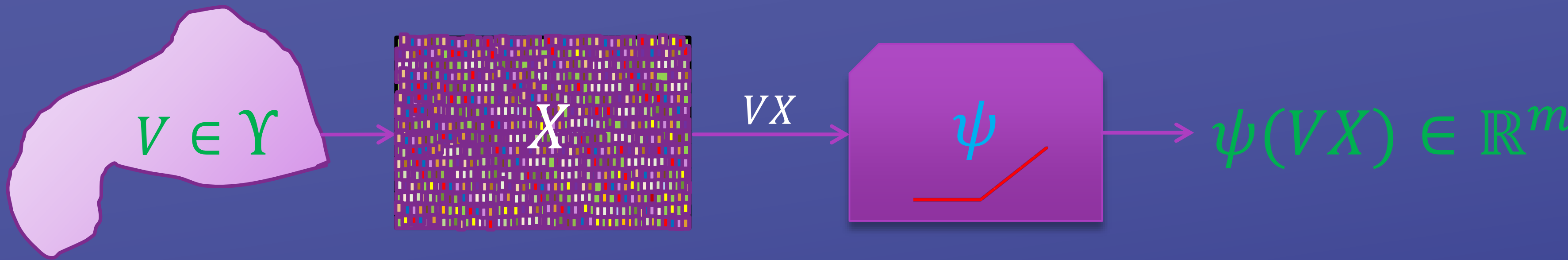
X is a random Gaussian matrix

ψ is the ReLU



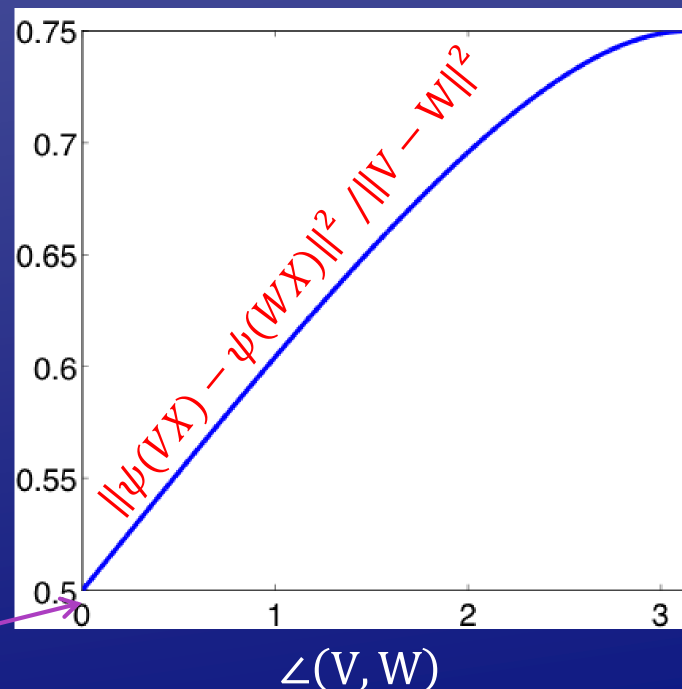
$$m = O\left(\delta^{-4}\omega^4(\Upsilon)\right)$$

DISTANCE DISTORTION



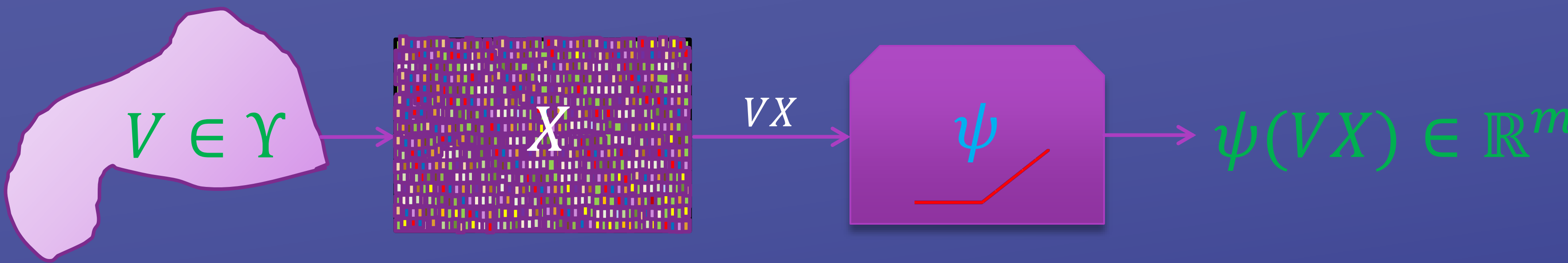
Theorem 4: for $V, W \in \Upsilon$

$$\left| \|\psi(VX) - \psi(WX)\|^2 - \tfrac{1}{2}\|\mathrm{V} - \mathrm{W}\|^2 - \frac{\|\mathrm{V}\|\,\|\mathrm{W}\|}{\pi}(\sin\angle(\mathrm{V},\mathrm{W}) \right|$$



The smaller $\angle(\mathrm{V},\mathrm{W})$ the smaller the distance we get between the points

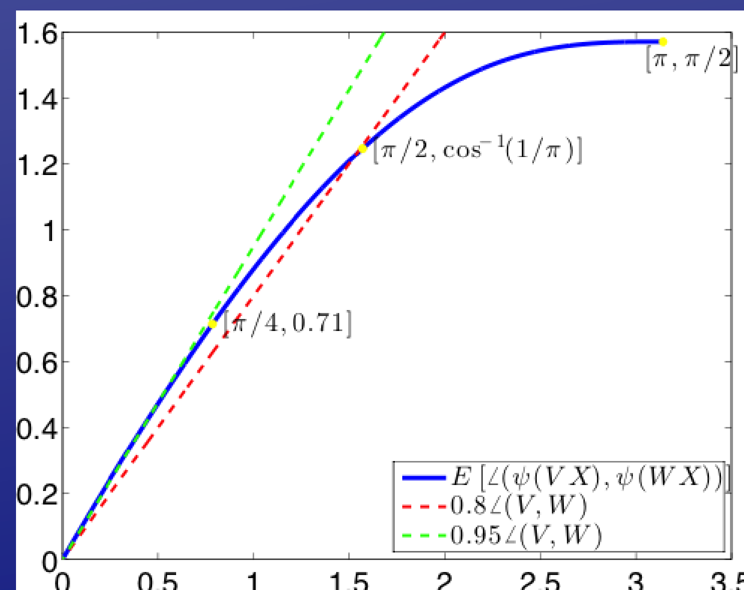
ANGLE DISTORTION



Theorem 5: for $V, W \in \Upsilon$

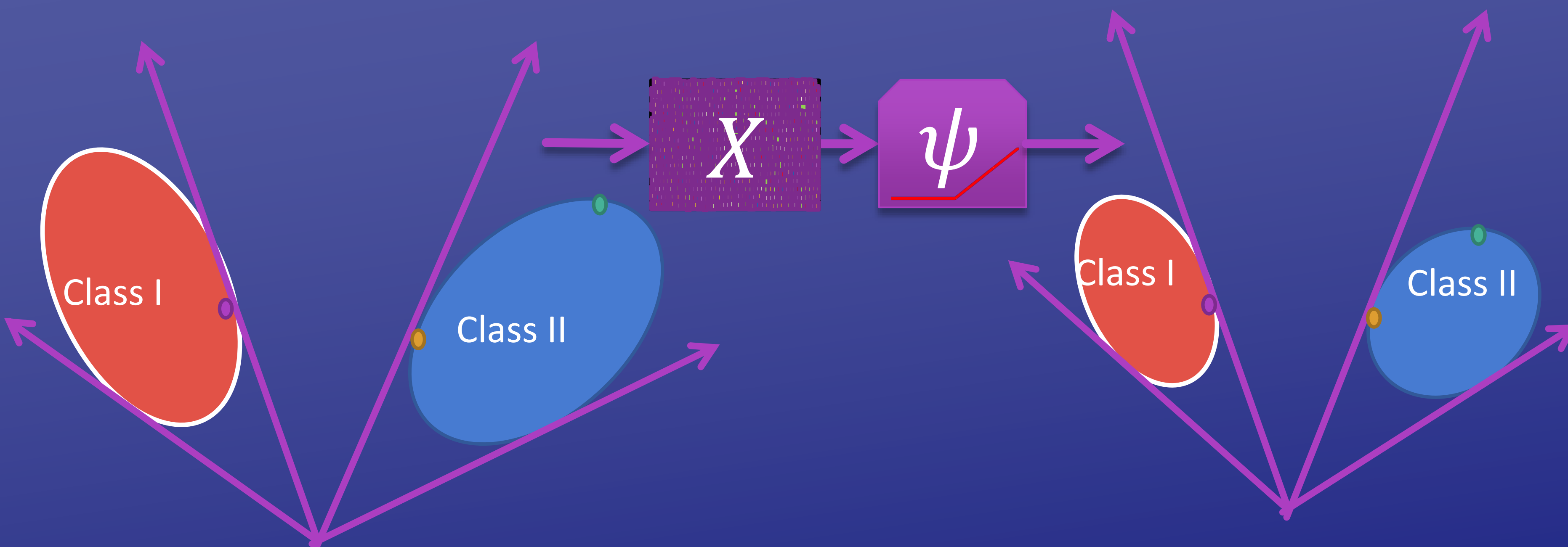
$$\left| \cos\angle(\psi(VX), \psi(WX)) - \cos\angle(V, W) - \frac{1}{\pi}(\sin\angle(V, W) \right|$$

Behavior of $\angle(\psi(VX), \psi(WX))$



$\angle(\mathrm{V}, \mathrm{W})$

DISTANCE AND ANGLES DISTORTION



Points with small angles between them become closer than points with larger angles between them

TRAINING DATA SIZE

- Stability in the network implies that close points in the input are close also at the output

➡ Having a good network for an ε-net of the input set Υ guarantees a good network for all the points in Υ.

➡ Using Sudakov minoration the number of data points is

$$\exp(\omega^2(\Upsilon)/\varepsilon^2)\,.$$

➡ Though this is not a tight bound, it introduces the Gaussian mean width $\omega\,(\Upsilon)$ as a measure for the complexity of the input data and the required number of training samples.

POOLING AND CONVOLUTIONS

- We test empirically this behavior on convolutional neural networks (CNN) with random weights and the MNIST, CIFAR-10 and ImageNet datasets.
- The behavior predicted in the theorem remains also in the presence of pooling and convolutions.

Role of Training

DNN keep the important information of the data.

Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

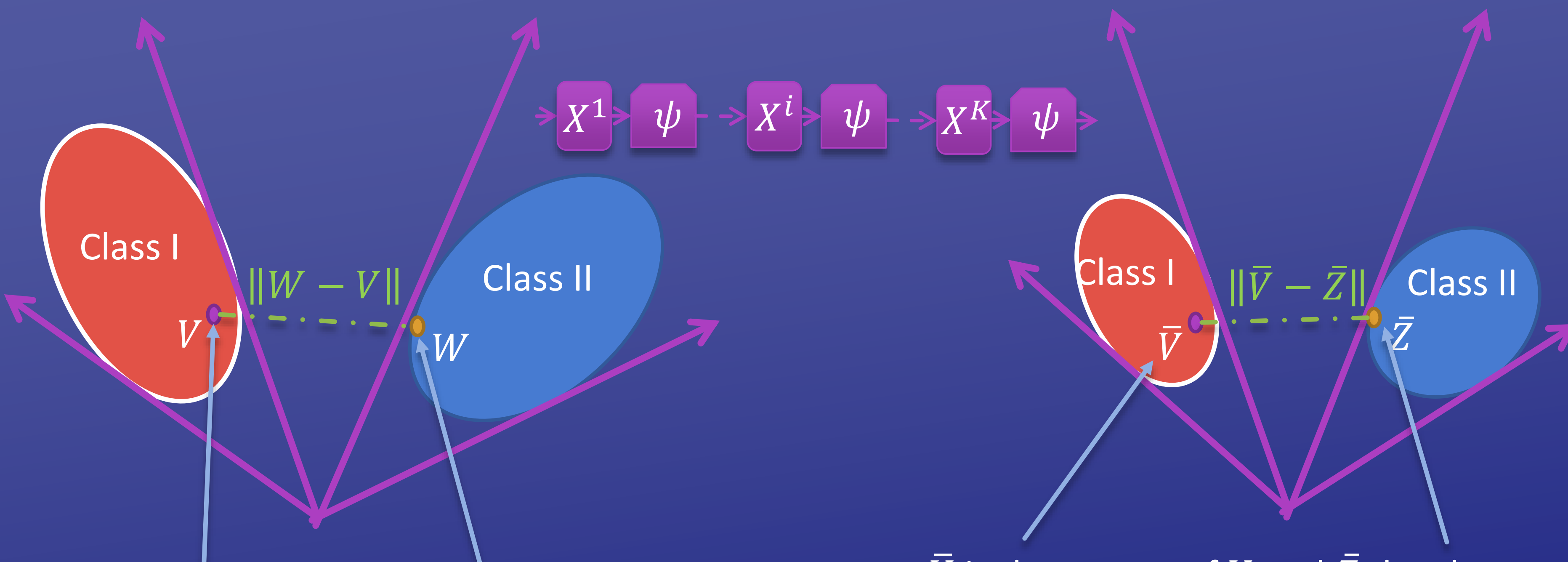
Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

ROLE OF TRAINING

- Having a theory for Gaussian weights we test the behavior of DNN after training.
- We looked at the MNIST, CIFAR-10 and ImageNet datasets.
- We will present here only the ImageNet results.
- We use a state-of-the-art pre-trained network for ImageNet [Simonyan & Zisserman, 2014].
- We compute inter and intra class distances.

INTER BOUNDARY POINTS DISTANCE RATIO

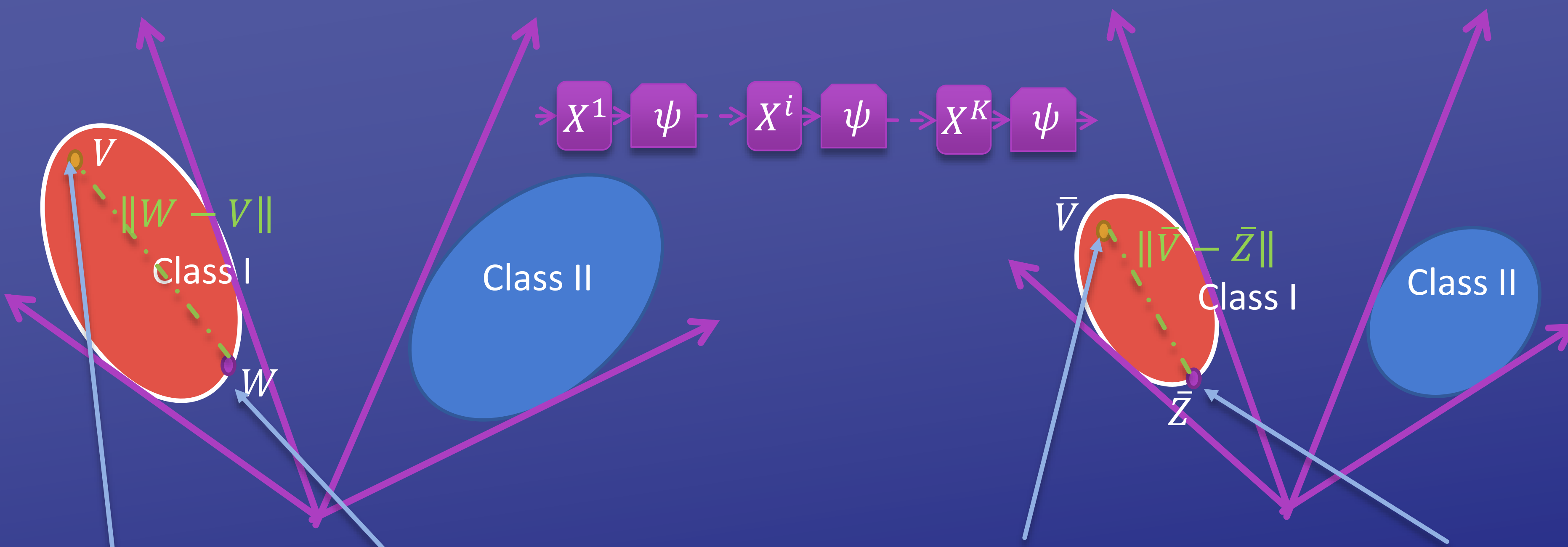


V is a random point and W its closest point from a different class.

$\bar{V}$ is the output of V and $\bar{Z}$ the closest point to $\bar{V}$ at the output from a different class.

Compute the distance ratio: $\frac{\|\bar{V}-\bar{Z}\|}{\|W-V\|}$

INTRA BOUNDARY POINTS DISTANCE RATIO



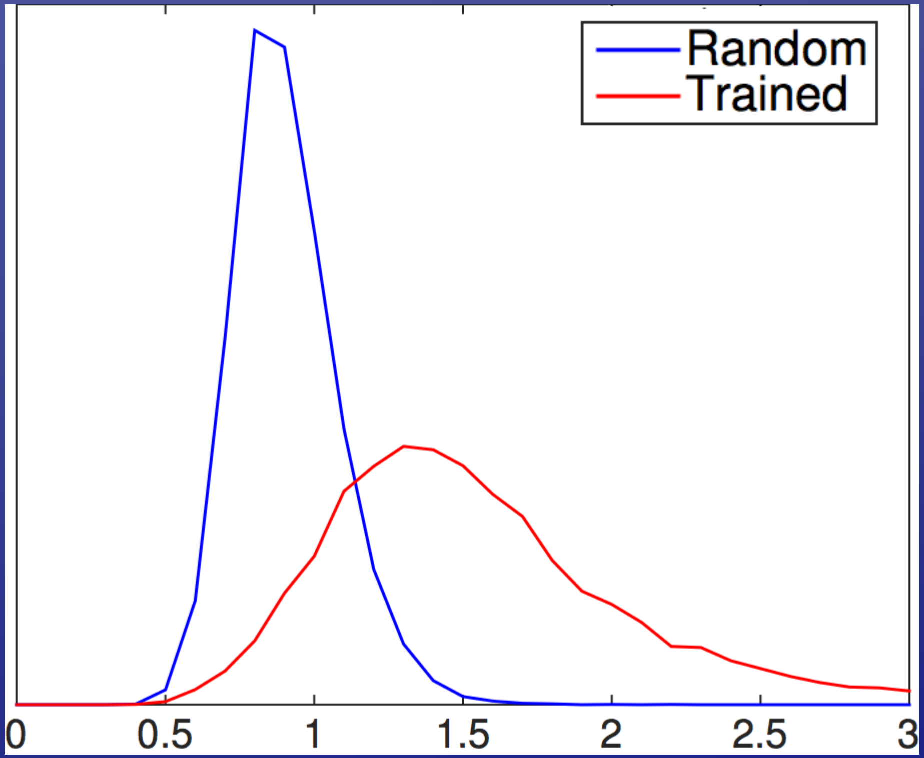
Let V be a point and W its farthest point from the same class.

Let $\bar{V}$ be the output of V and $\bar{Z}$ the farthest point from $\bar{V}$ at the output from the same class

Compute the distance ratio: $\frac{\|\bar{V}-\bar{Z}\|}{\|W-V\|}$

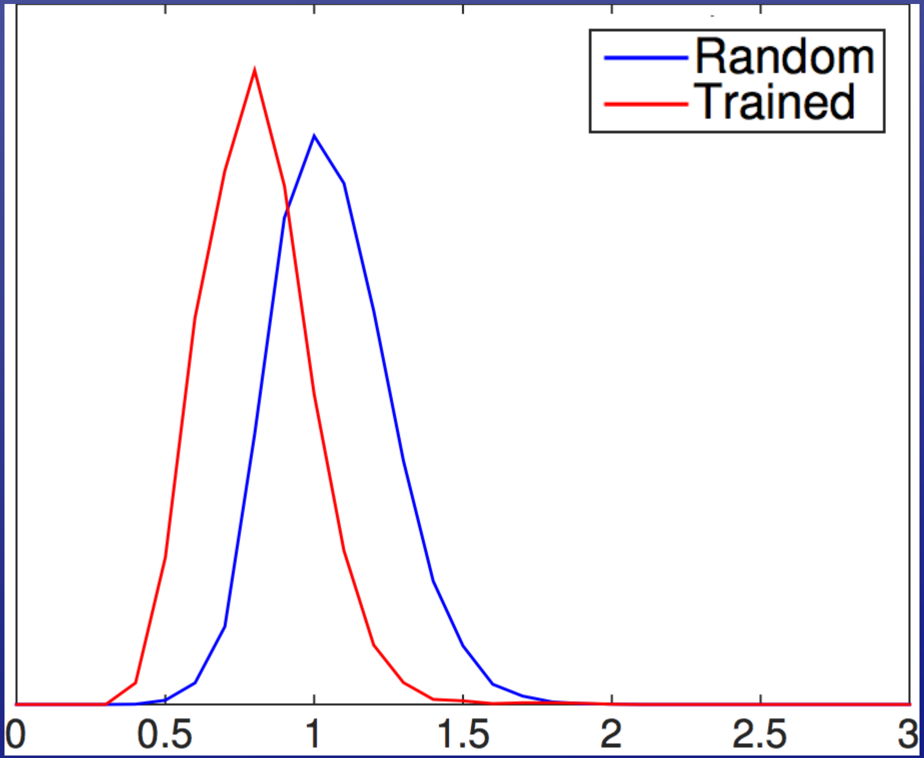
BOUNDARY DISTANCE RATIO

Inter-class



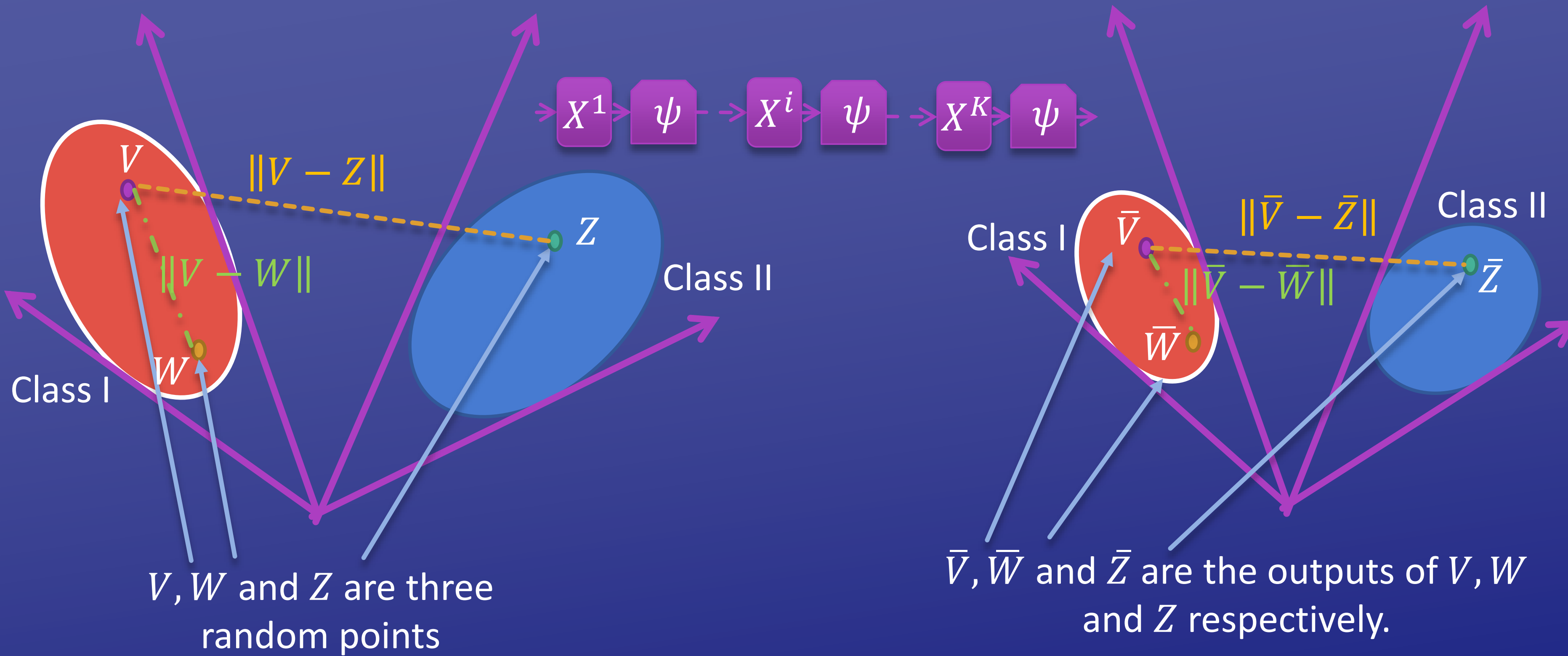
$$\frac{\|\bar{V} - \bar{Z}\|}{\|W - V\|}$$

Intra-class



$$\frac{\|\bar{V} - \bar{Z}\|}{\|W - V\|}$$

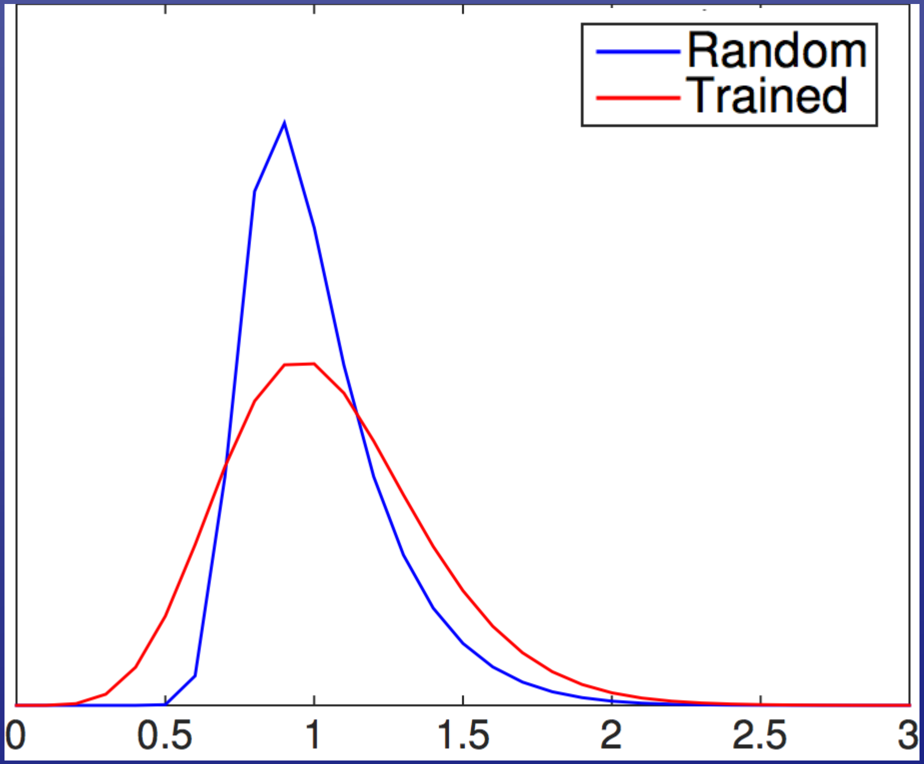
AVERAGE POINTS DISTANCE RATIO



Compute the distance ratios: $\frac{\|\bar{V}-\bar{W}\|}{\|V-W\|}, \frac{\|\bar{V}-\bar{Z}\|}{\|V-Z\|}$

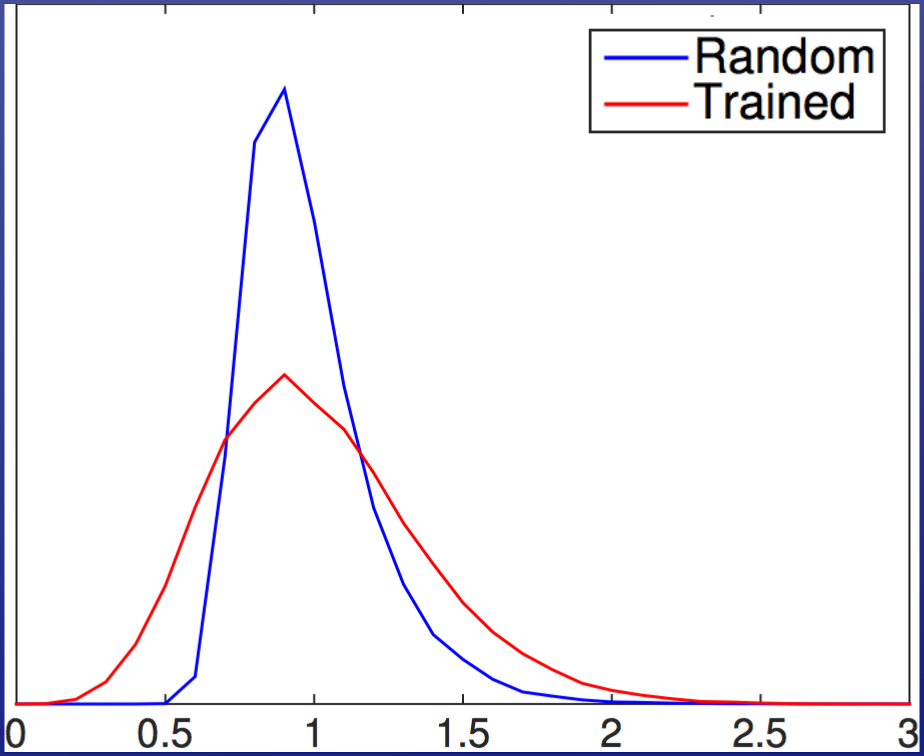
AVERAGE DISTANCE RATIO

Inter-class



$$\frac{\|\bar{V} - \bar{Z}\|}{\|V - Z\|}$$

Intra-class



$$\frac{\|\bar{V} - \bar{W}\|}{\|V - W\|}$$

ROLE OF TRAINING

- On average distances are preserved in the trained and random networks.
- The difference is with respect to the boundary points.
- The inter distances become larger.
- The intra distances shrink.

DNN as Metric Learning

DNN keep the important information of the data.

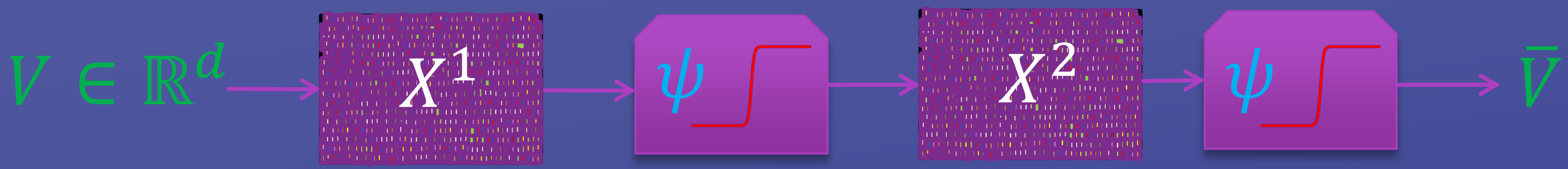
Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

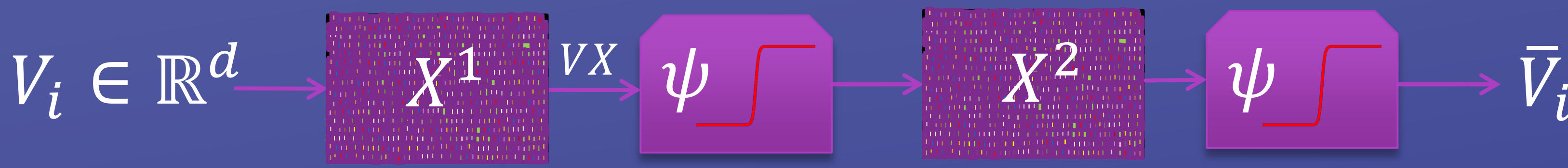
ASSUMPTIONS



X is fully connected and trained

ψ is the hyperbolic tan

METRIC LEARNING BASED TRAINING



- Cosine Objective:

$$\min_{X^1, X^2} \sum_{i,j \in Training\ Set} \left(\frac{\bar{V}_i{}^T \bar{V}_j}{\|\bar{V}_i\| \|\bar{V}_j\|} - \vartheta_{i,j} \right)^2$$

Classification term

Metric preservation term

$$\vartheta_{i,j} = \begin{cases} \lambda + (1 - \lambda) \dfrac{V_i{}^T V_j}{\|V_i\| \|V_j\|} & i,j \in same\ class \\ -1 & i,j \in different\ clas \end{cases}$$

METRIC LEARNING BASED TRAINING

$$V_i \in \mathbb{R}^d \rightarrow X^1 \xrightarrow{VX} \psi \rightarrow X^2 \rightarrow \psi \rightarrow \bar{V}_i$$

$$l_{ij} = \begin{cases} 1 & i,j \in \begin{matrix} same \\ class \end{matrix} \\ -1 & i,j \in \begin{matrix} different \\ class \end{matrix} \end{cases} \qquad l_{ij} = \begin{cases} \begin{matrix} average\ intra \\ class\ distance \end{matrix} & i,j \in \begin{matrix} same \\ class \end{matrix} \\ \begin{matrix} average\ inter \\ class\ distance \end{matrix} & i,j \in \begin{matrix} different \\ class \end{matrix} \end{cases}$$

- Euclidean Objective:

$$\min_{X^1,X^2} \frac{\lambda}{\left|\begin{matrix} Training \\ Set \end{matrix}\right|} \sum_{\begin{matrix} i,j \in Training \\ Set \end{matrix}} \underbrace{\left[l_{ij} \|\bar{V}_i - \bar{V}_j\| - t_{ij} \right]_+}_{\text{Classification term}} + \frac{1-\lambda}{|Neighbours|} \sum_{\begin{matrix} V_i, V_j\ are \\ neighbours \end{matrix}} \underbrace{\left| \|\bar{V}_i - \bar{V}_j\| - \|V_i - V_j\| \right|}_{\text{Metric learning term}}$$

ROBUSTNESS OF THIS NETWORK

- Metric learning objectives impose stability
- Similar to what we have in the random case
- Close points at the input are close at the output
- Using the theory of (T, ϵ)-robustness, the generalization error scales as $\sqrt{\frac{T}{|Training\ set|}}$
- T is the covering number.
- Also here, the number of training samples scales as
$$\exp(\omega^2(\Upsilon)/\varepsilon^2).$$

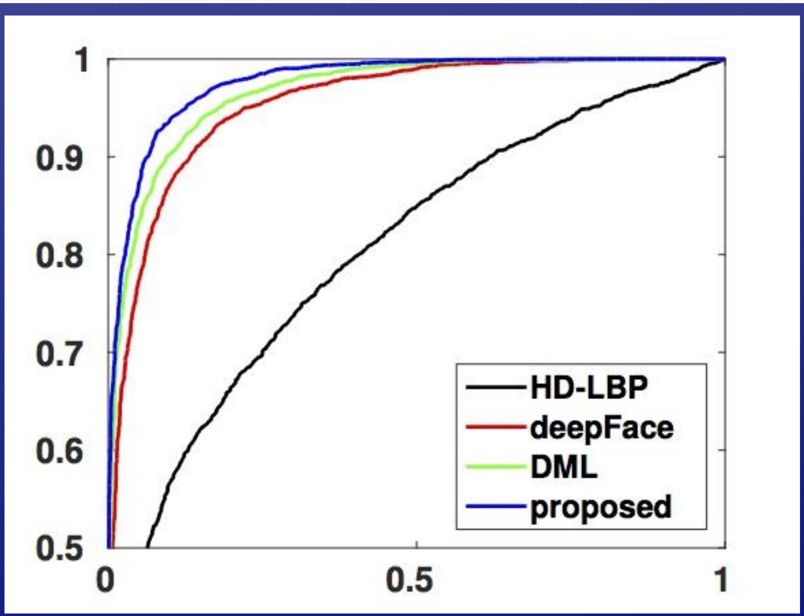
RESULTS

- Better performance with less training samples

MNIST Dataset

#Training/class	30	50	70	100
original pixels	81.91%	86.18%	86.86%	88.49%
LeNet	87.51%	89.89%	91.24%	92.75%
Proposed 1	92.32%	94.45%	95.67%	96.19%
Proposed 2	**94.14%**	**95.20%**	**96.05%**	**96.21%**

Faces in the wild ROC curve



[Huang, Qiu, Sapiro, Calderbank, 2015]

Take Home Message

DNN keep the important information of the data.

Gaussian mean width is a good measure for the complexity of the data.

Important goal of training: Classify the boundary points between the different classes in the data.

Random Gaussian weights are good for classifying the average points in the data.

Deep learning can be viewed as a metric learning.

ACKNOWLEDGEMENTS

Alex Bronstein

Tel Aviv University

Guillermo Sapiro

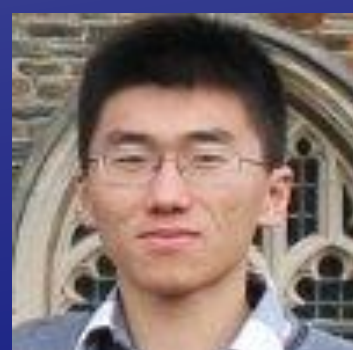
Duke University

Robert Calderbank

Duke University

Qiang Qiu

Duke University



Jiaji Huang

Duke Universiy

Grants

NSF

ONR

NGA

NSSEFF

ARO

ERC

QUESTIONS?

SITES.DUKE.EDU/RAJA

REFERENCES

R. GIRYES, G. SAPIRO, A. M. BRONSTEIN, *DEEP NEURAL NETWORKS WITH RANDOM GAUSSIAN WEIGHTS: A UNIVERSAL CLASSIFICATION STRATEGY?*

J. HUANG, Q. QIU, G. SAPIRO, R. CALDERBANK, *DISCRIMINATIVE GEOMETRY-AWARE DEEP TRANSFORM*

J. HUANG, Q. QIU, G. SAPIRO, R. CALDERBANK, *DISCRIMINATIVE ROBUST TRANSFORMATION LEARNING*