DATA STRUCTURE BASED THEORY FOR DEEP LEARNING

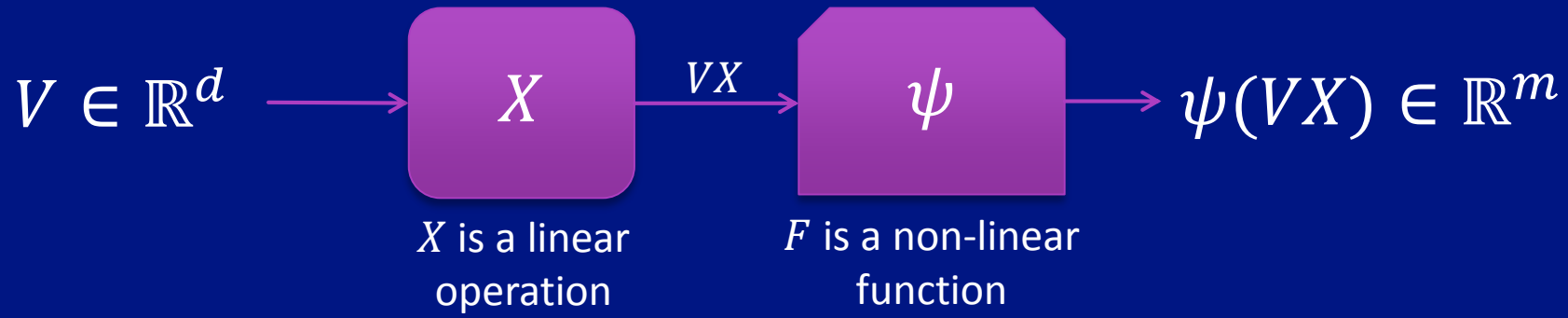
RAJA GIRYES
TEL AVIV UNIVERSITY

GUILLERMO SAPIRO
DUKE UNIVERSITY

Mathematics of Deep Learning
Computer Vision and Pattern Recognition (CVPR)
June 26, 2016

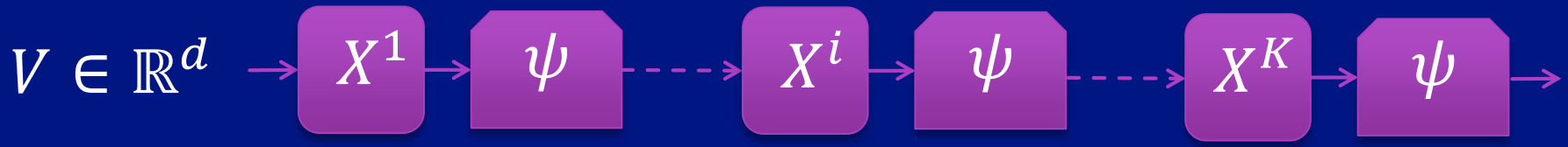
DEEP NEURAL NETWORKS (DNN)

- One layer of a neural net



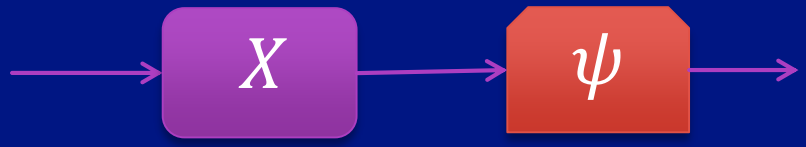
- Concatenation of the layers creates the whole net

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



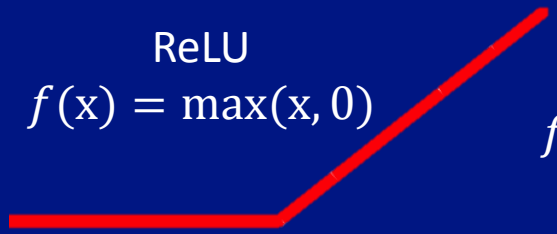
THE NON-LINEAR PART

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

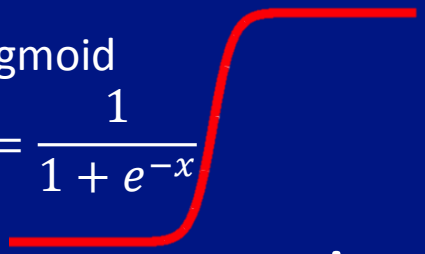


- f is the (point-wise) activation function

ReLU
$f(x) = \max(x, 0)$

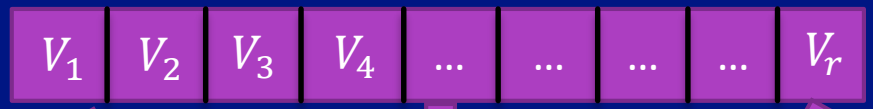


Sigmoid
$f(x) = \frac{1}{1 + e^{-x}}$



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

- g is a pooling or an aggregation operator.



Max pooling
$\max_i V_i$

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

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

WHY DNN WORK?

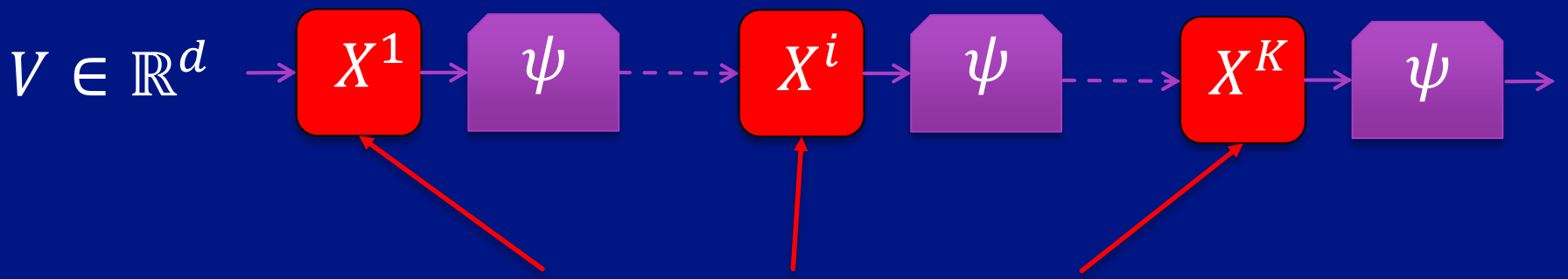

What is so special with the DNN structure?
What is the role of the depth of DNN?
What is the capability of DNN?
How many training samples do we need?
What is the role of pooling?
What is the role of the activation function?
What happens to the data throughout the layers?

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 et al. 2014], invariance to more complex deformations [Bruna & Mallat, 2013] and better modeling of correlations of the input [Cohen et al. 2016]
- Number of training samples scales as the number of parameters [Shalev-Shwartz & Ben-David 2014] or the norm of the weights in the DNN [Neyshabur et al. 2015]
- 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]
- Neural networks are capable of approximating the solution of optimi-zation problems such as ℓ_1-minimization [Bruna et al. 2016], [Giryes et al. 2016]

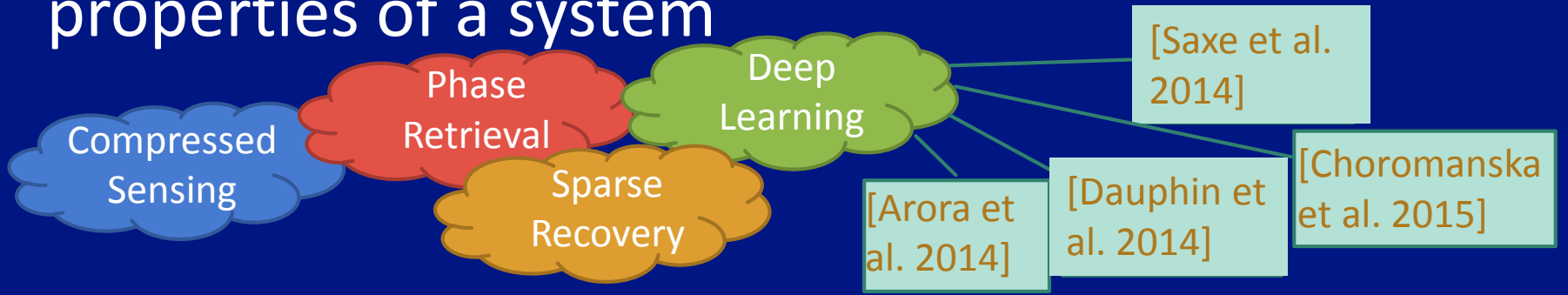

DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Important goal of training: Classify the boundary points between the different classes in the data.
Take Home Message
Random Gaussian weights are good for classifying the average points in the data.
Generalization error depends on the DNN input margin
Deep learning can be viewed as a metric learning.

ASSUMPTIONS – GAUSSIAN WEIGHTS

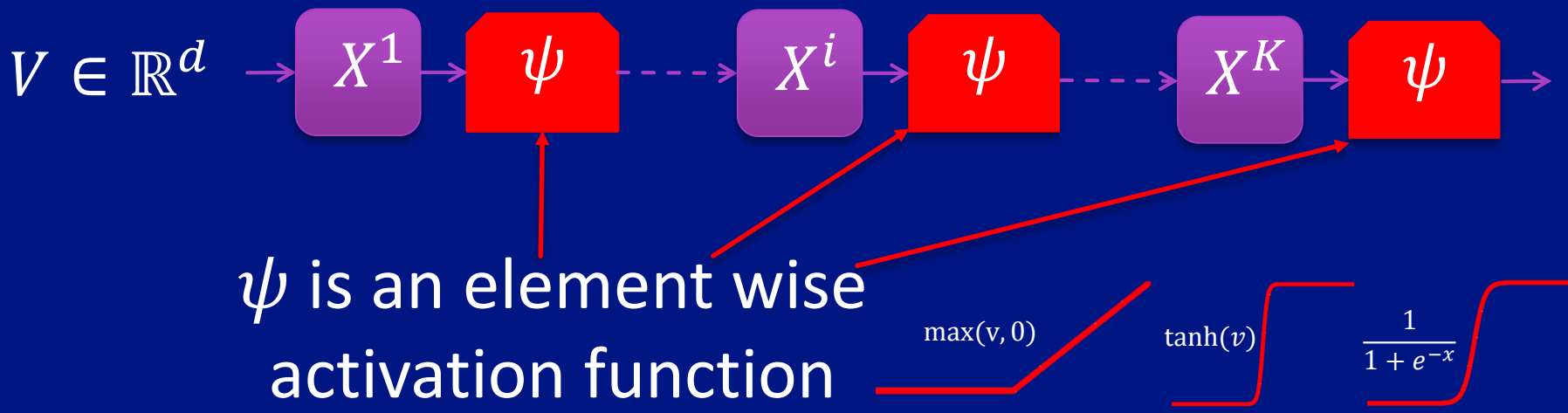


$X^i, \ldots, X^i, \ldots, X^K$ are random Gaussian matrices

- Infusion of random weights reveals internal properties of a system



ASSUMPTIONS – NO POOLING

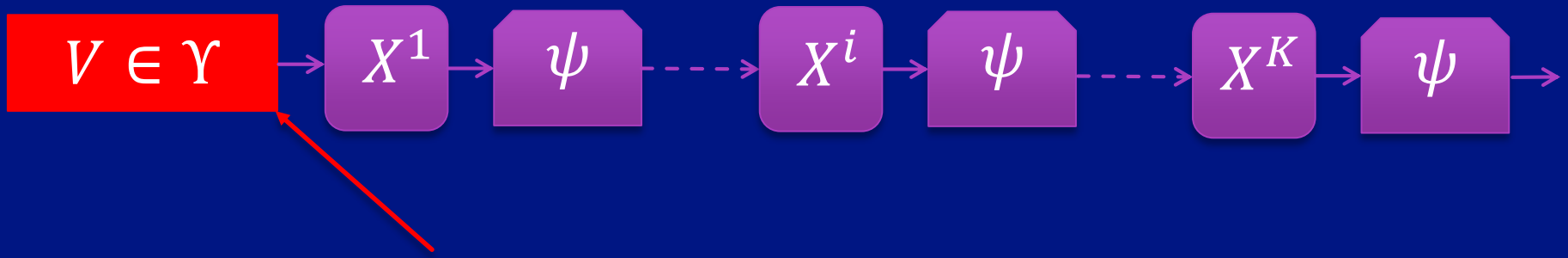


- Pooling provides invariance [Boureau et. al. 2010, Bruna et. al. 2013].

➢ 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



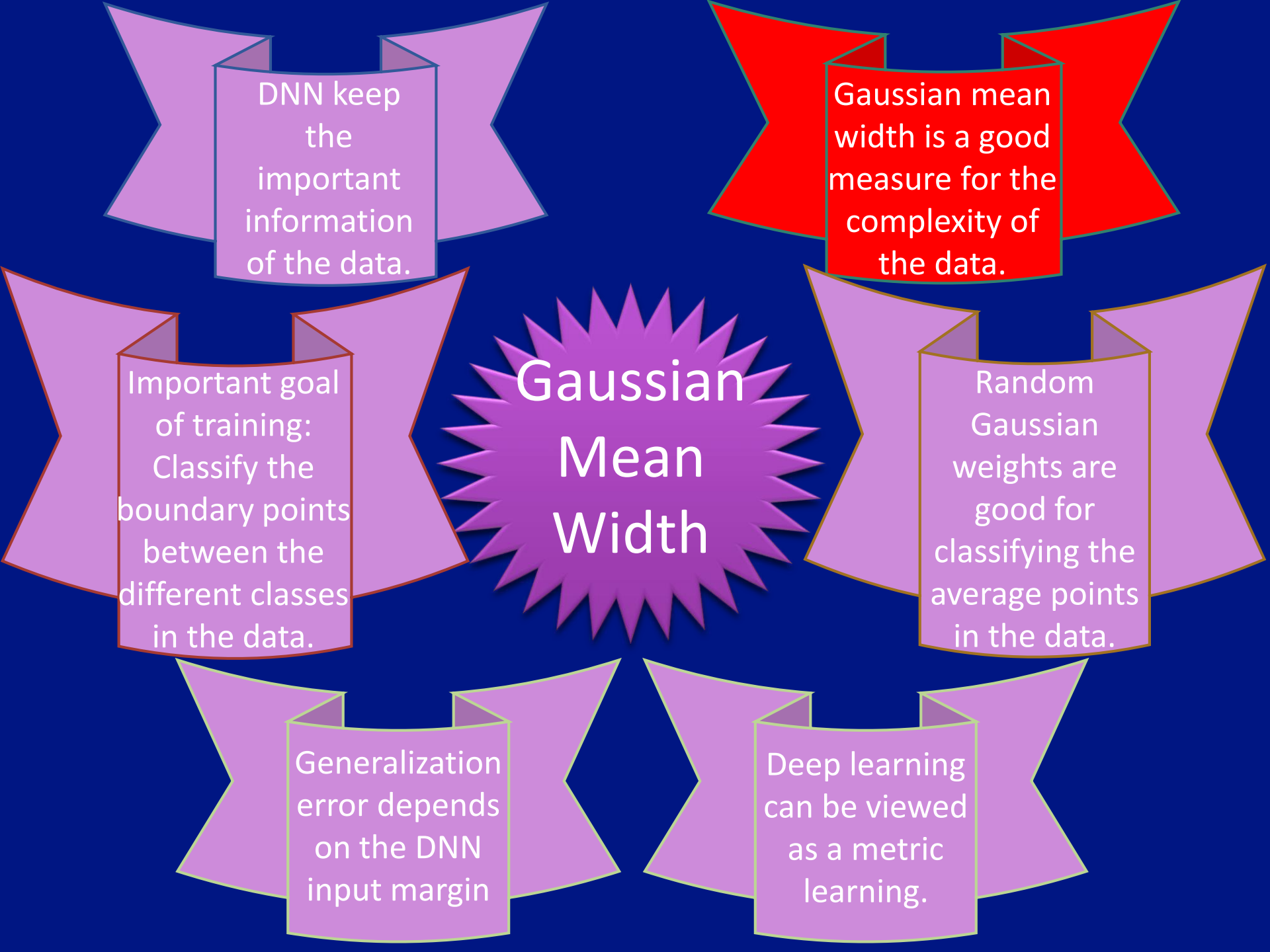
Υ is a low dimensional set

Gaussian Mixture Models (GMM)

Low Rank Matrices

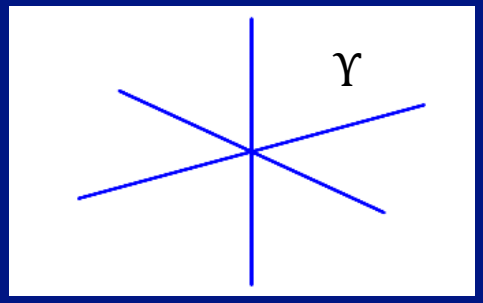
Signals with Sparse Representations

Low Dimensional Manifolds

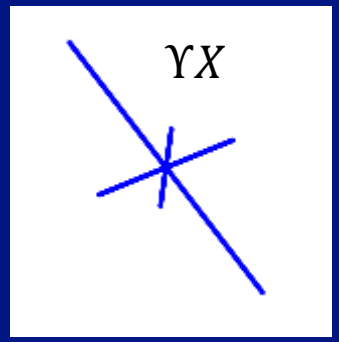

DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Important goal of training: Classify the boundary points between the different classes in the data.
Gaussian Mean Width
Random Gaussian weights are good for classifying the average points in the data.
Generalization error depends on the DNN input margin
Deep learning can be viewed as a metric learning.

WHAT HAPPENS TO SPARSE DATA IN DNN?

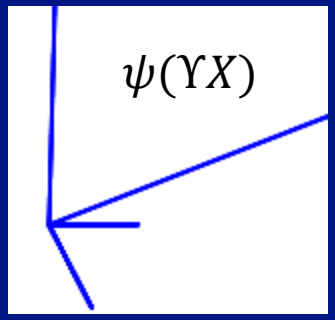
- Let Υ be sparsely represented data
  - Example: $\Upsilon = \{V \in \mathbb{R}^3 : \|V\|_0 \leq 1\}$



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



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

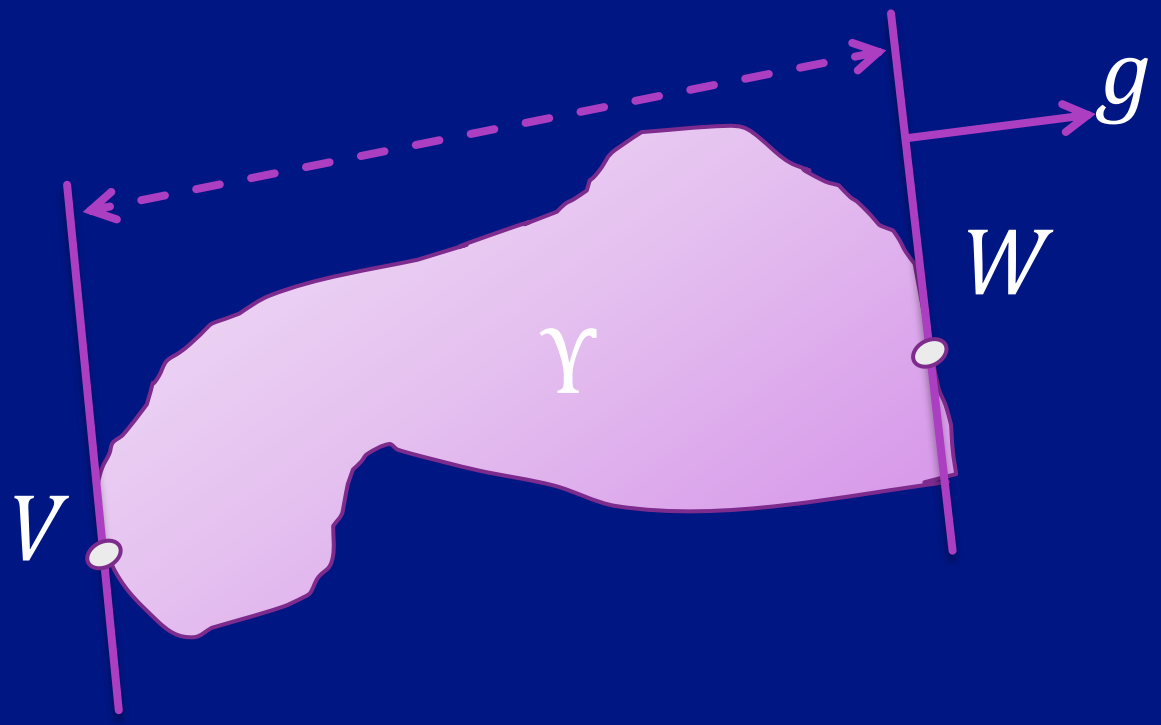


GAUSSIAN MEAN WIDTH

- 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:

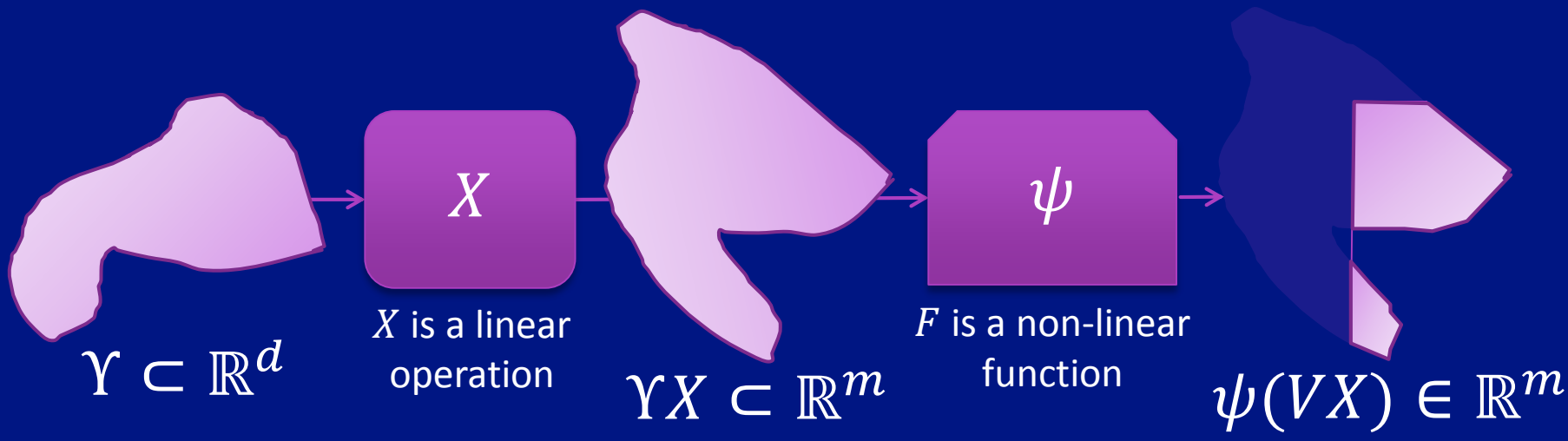
If $\Upsilon \subset \mathbb{B}^d$ is a Gaussian Mixture Model with k Gaussians then

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

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

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

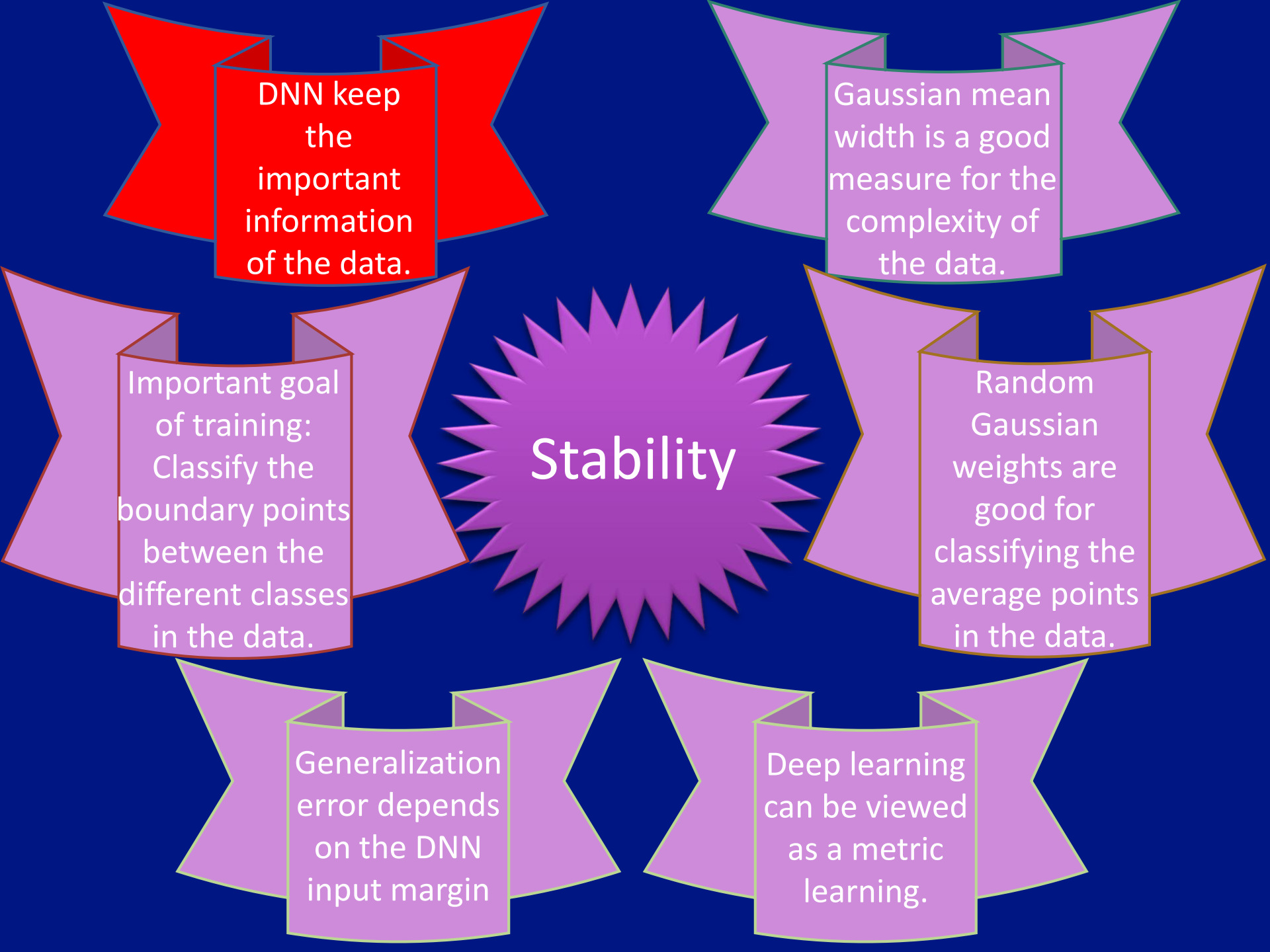
GAUSSIAN MEAN WIDTH IN DNN



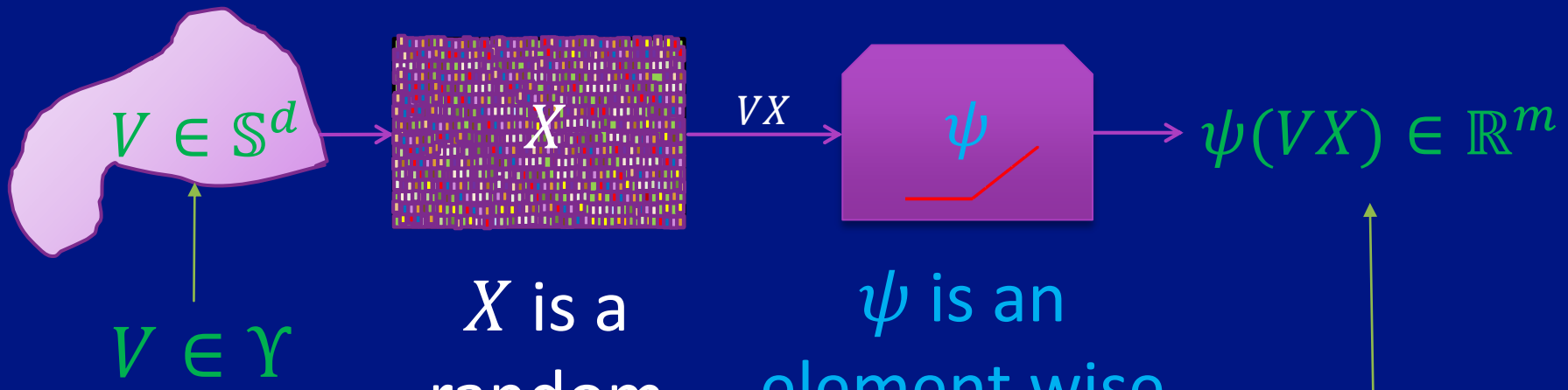
Theorem 1: small $\frac{\omega^2(\Upsilon)}{m}$ imply $\omega^2(\Upsilon) \approx \omega^2(\psi(VX))$

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

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

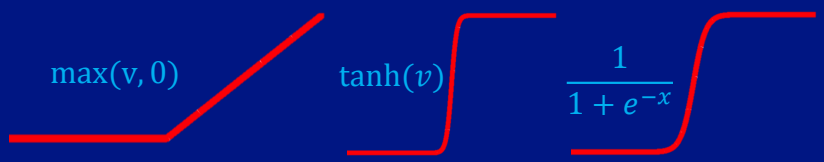

DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Important goal of training: Classify the boundary points between the different classes in the data.
Stability
Random Gaussian weights are good for classifying the average points in the data.
Generalization error depends on the DNN input margin
Deep learning can be viewed as a metric learning.

ASSUMPTIONS



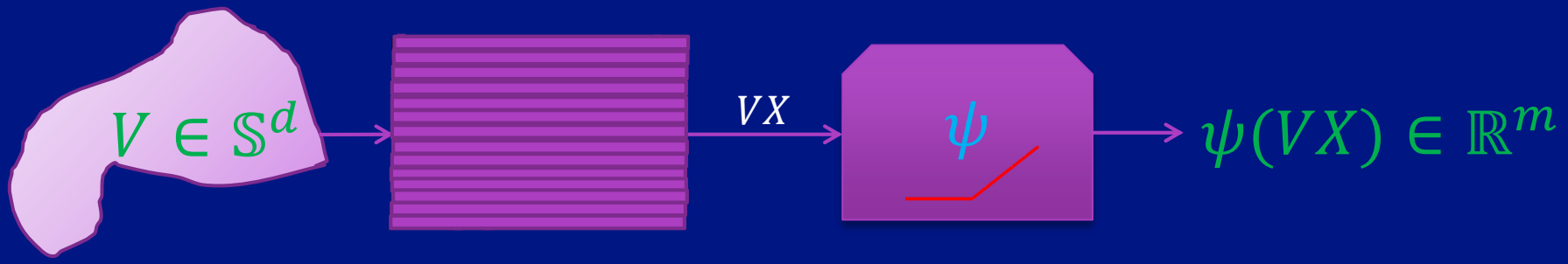
X is a random Gaussian matrix

ψ is an element wise activation function

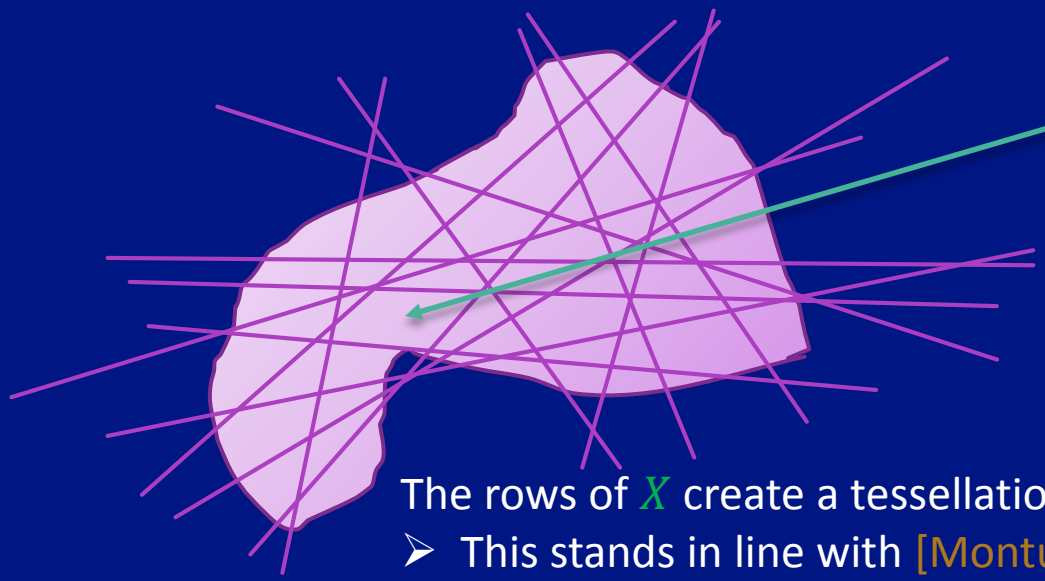


$$m = O\left(\delta^{-6}\omega^2(\Upsilon)\right)$$

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].



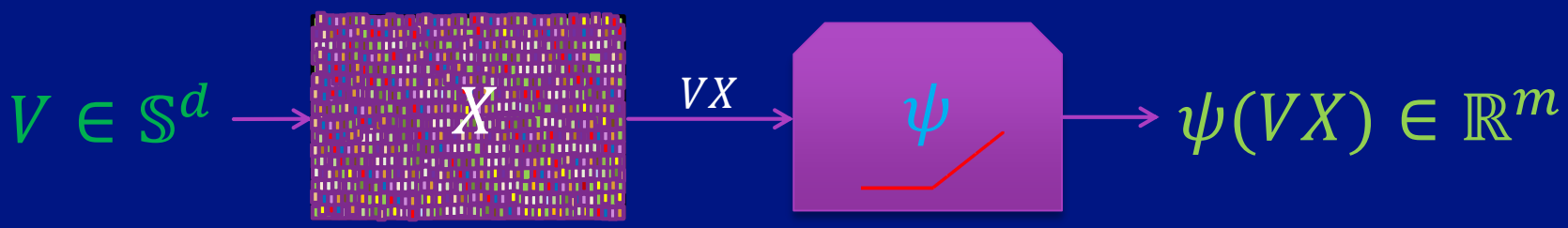
- If two points belong to the same tile then their distance $< \delta$

➡ 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

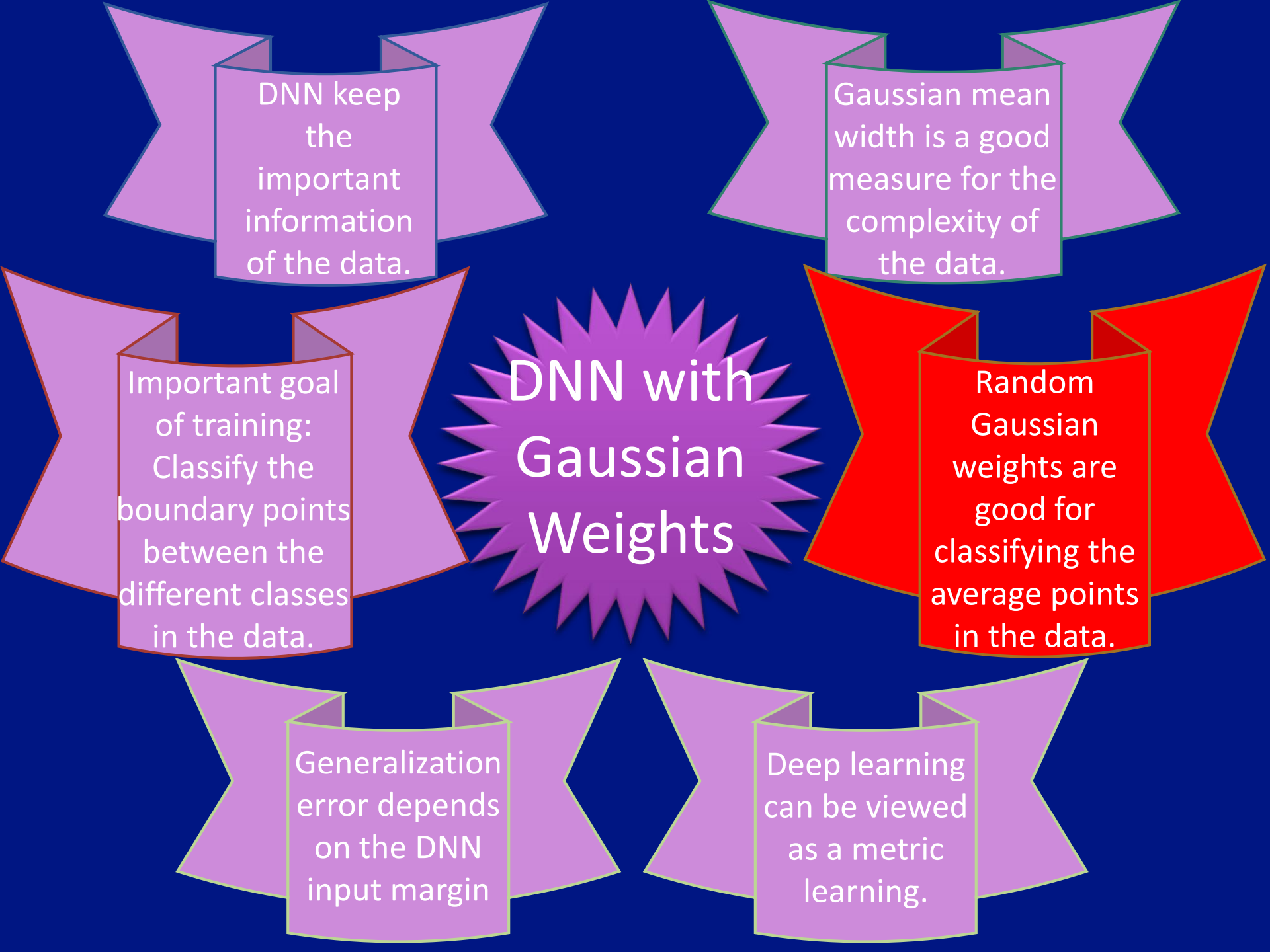


Theorem 3: There exists an algorithm $\mathcal{A}$ such that

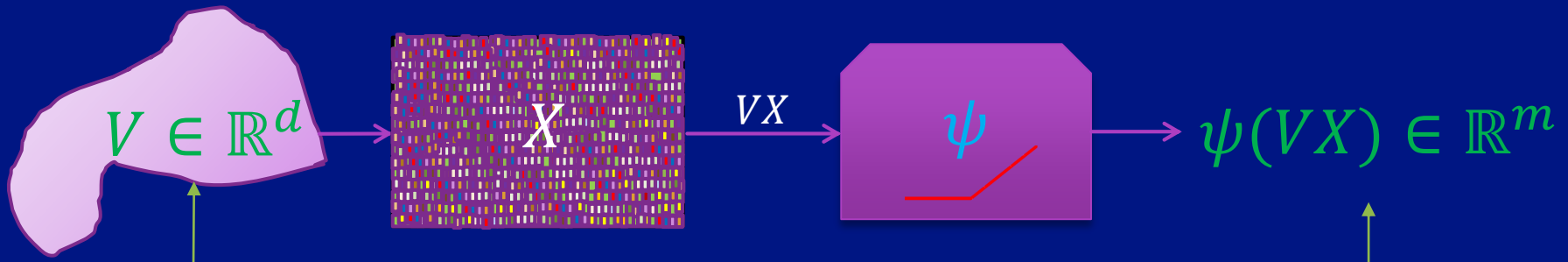
$$\|V - \mathcal{A}(\psi(VX))\| < O\left(\frac{\omega(\Upsilon)}{\sqrt{m}}\right) = O(\delta^3)$$

[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 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.
DNN with Gaussian Weights
Random Gaussian weights are good for classifying the average points in the data.
Generalization error depends on the DNN input margin
Deep learning can be viewed as a metric learning.

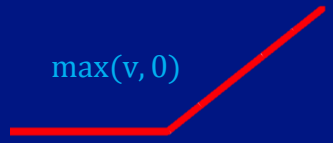
ASSUMPTIONS



$V \in \Upsilon$

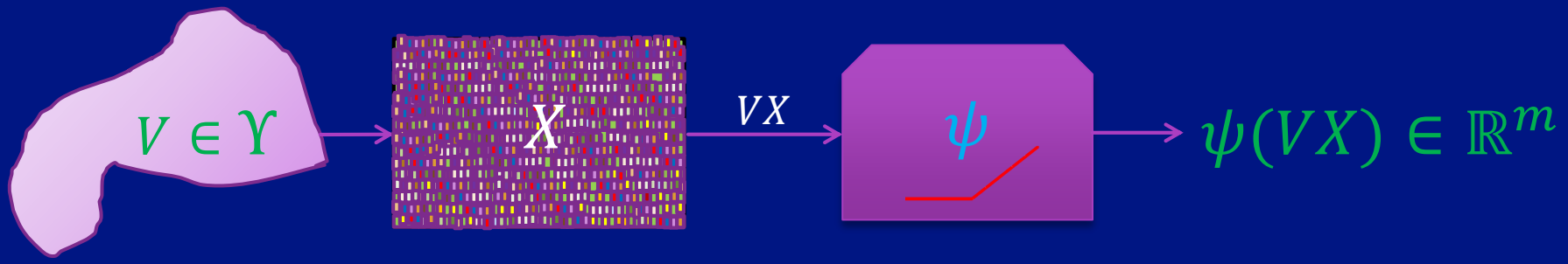
X is a random Gaussian matrix

ψ is the ReLU



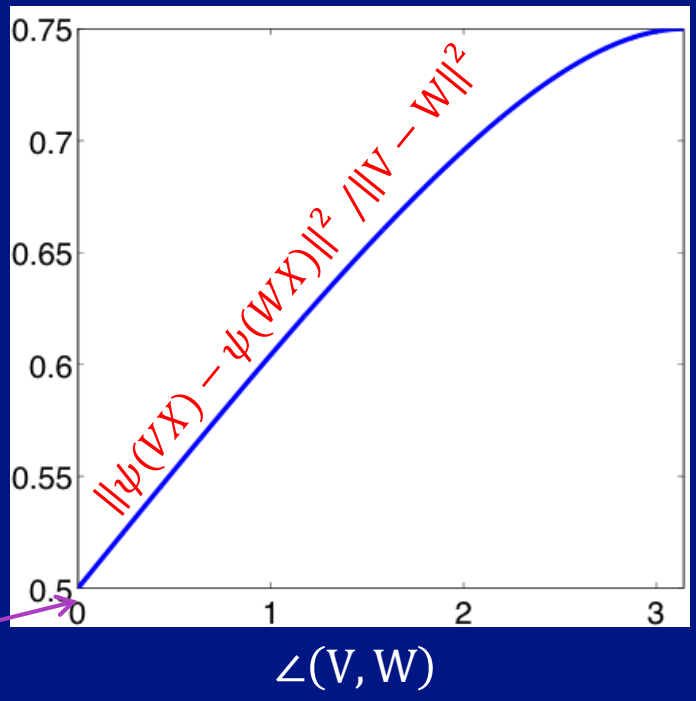
$$m = O\left(\delta^{-4}\omega^4(\Upsilon)\right)$$

DISTANCE DISTORTION



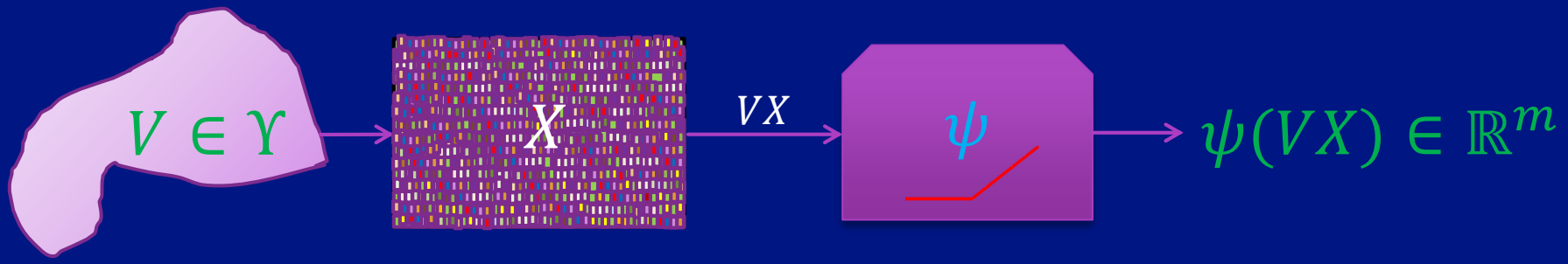
Theorem 4: for $V, W \in \Upsilon$

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



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

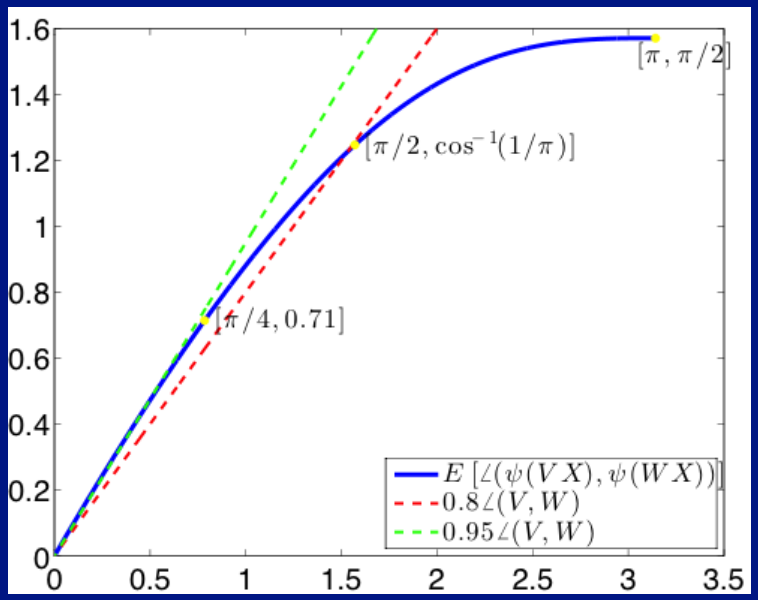
ANGLE DISTORTION



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

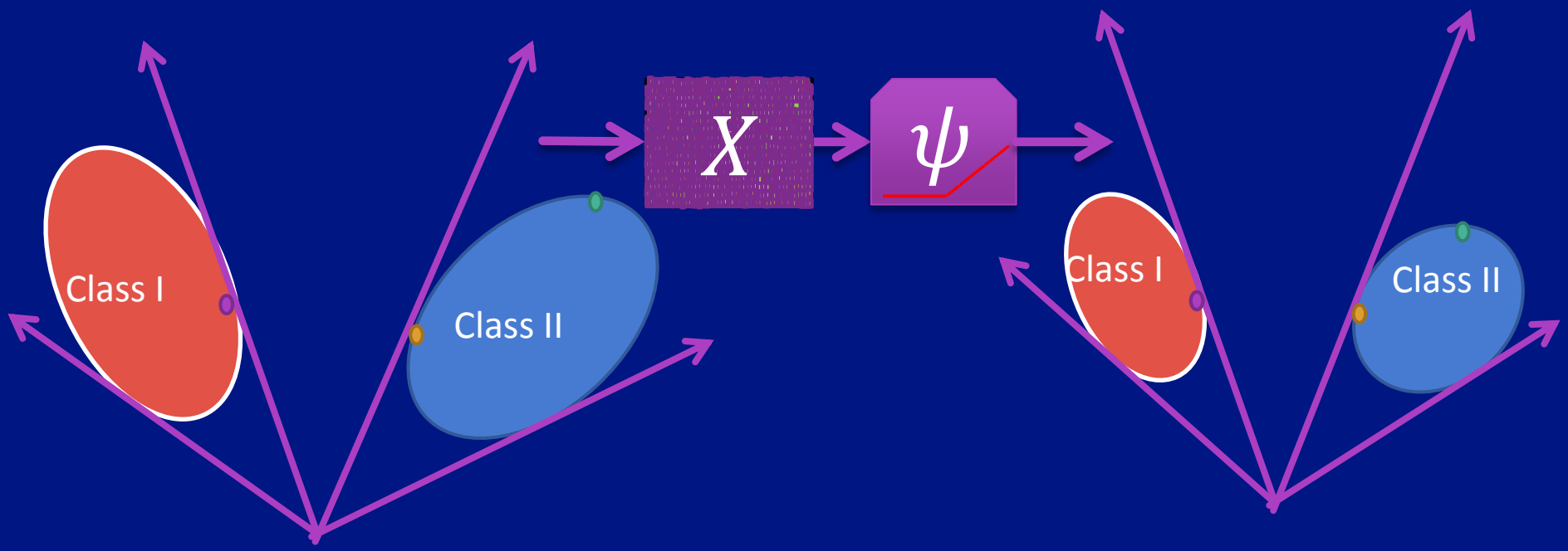
$$\left|\cos\angle(\psi(VX),\psi(WX)) - \cos\angle(V,W)\right| \leq \frac{1}{\pi}\left(\sin\angle(V,W)\right)^{\frac{1}{2}}$$

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



$\angle(V,W)$

DISTANCE AND ANGLES DISTORTION



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

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 theorems remains also in the presence of pooling and convolutions.

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.

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.

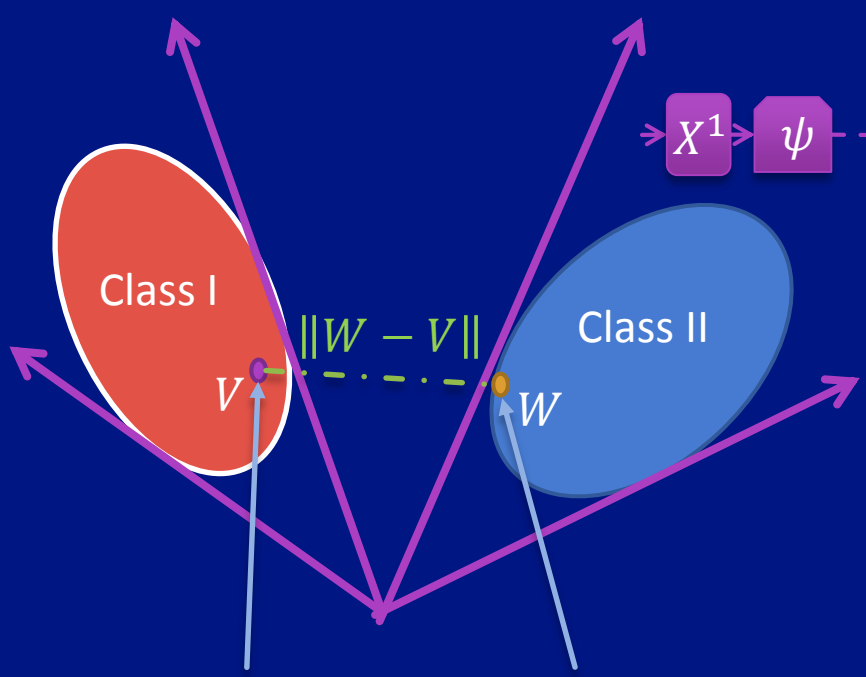
Generalization error depends on the DNN input margin

Deep learning can be viewed as a metric learning.

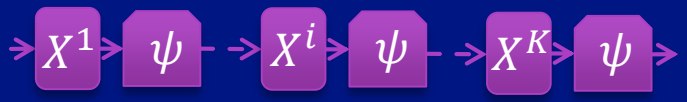
ROLE OF TRAINING

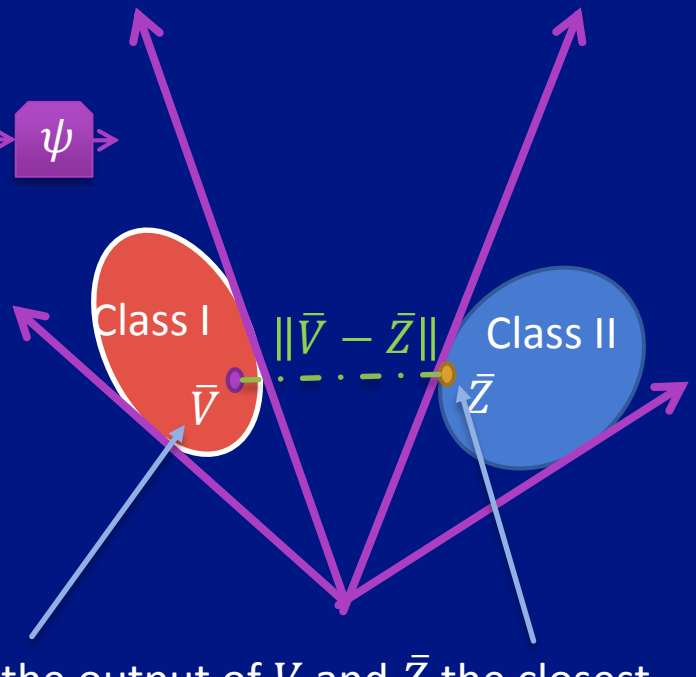
- Having a theory for Gaussian weights we test the behavior of DNN after training.
- We looked at the MNIST, CIFAR-10 and ImageNet datasets.
- We will present here only the ImageNet results.
- We use a state-of-the-art pre-trained network for ImageNet [Simonyan & Zisserman, 2014].
- We compute inter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

INTER BOUNDARY POINTS DISTANCE RATIO



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

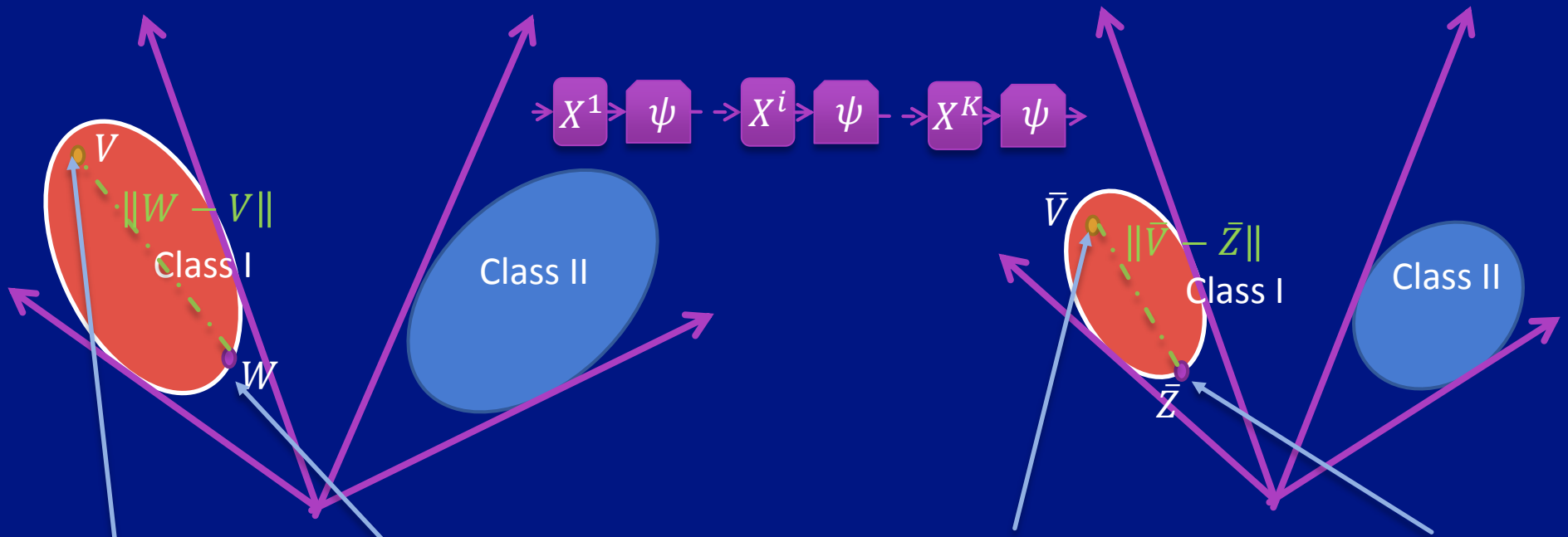




$\bar{V}$ is the output of V and $\bar{Z}$ the closest point to $\bar{V}$ at the output from a different class.

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

INTRA BOUNDARY POINTS DISTANCE RATIO



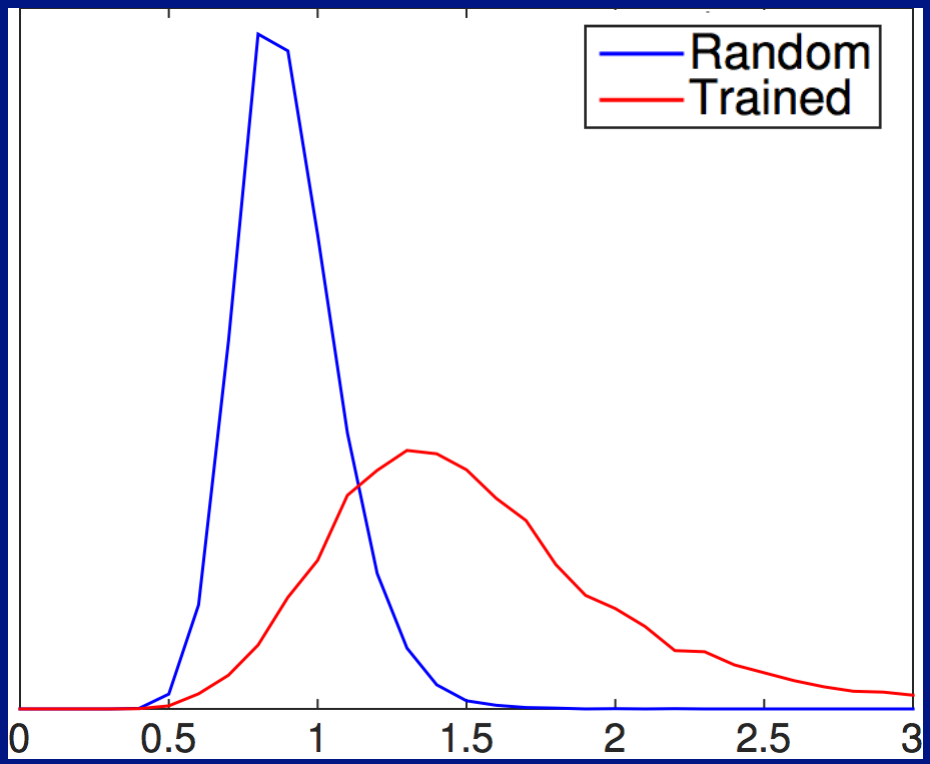
Let V be a point and W its farthest point from the same class.

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

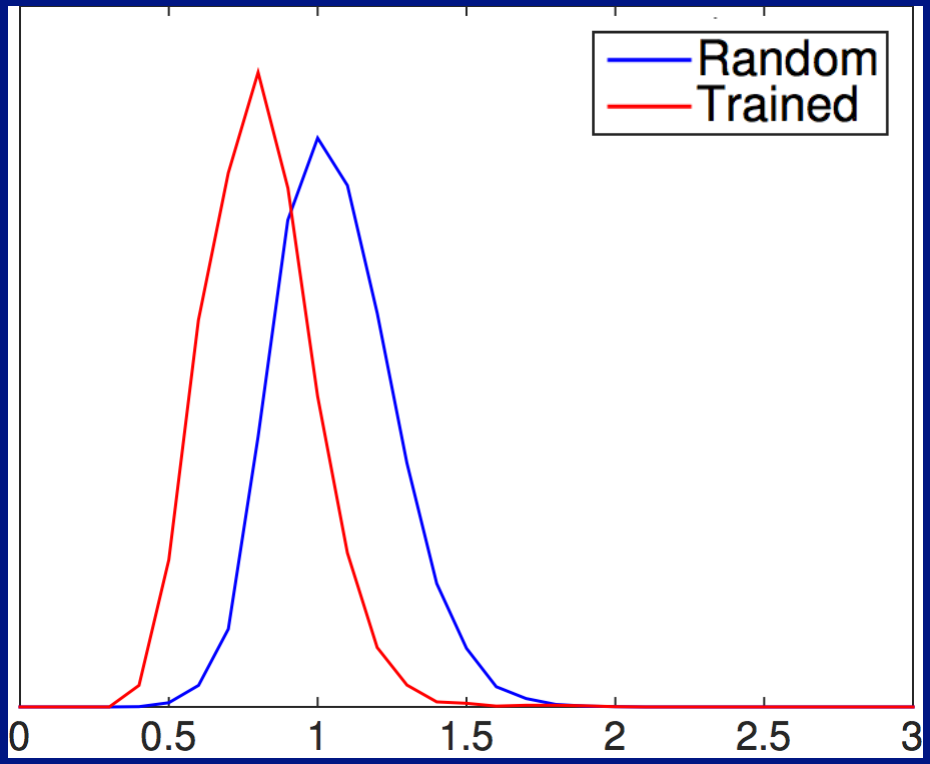
BOUNDARY DISTANCE RATIO

Inter-class



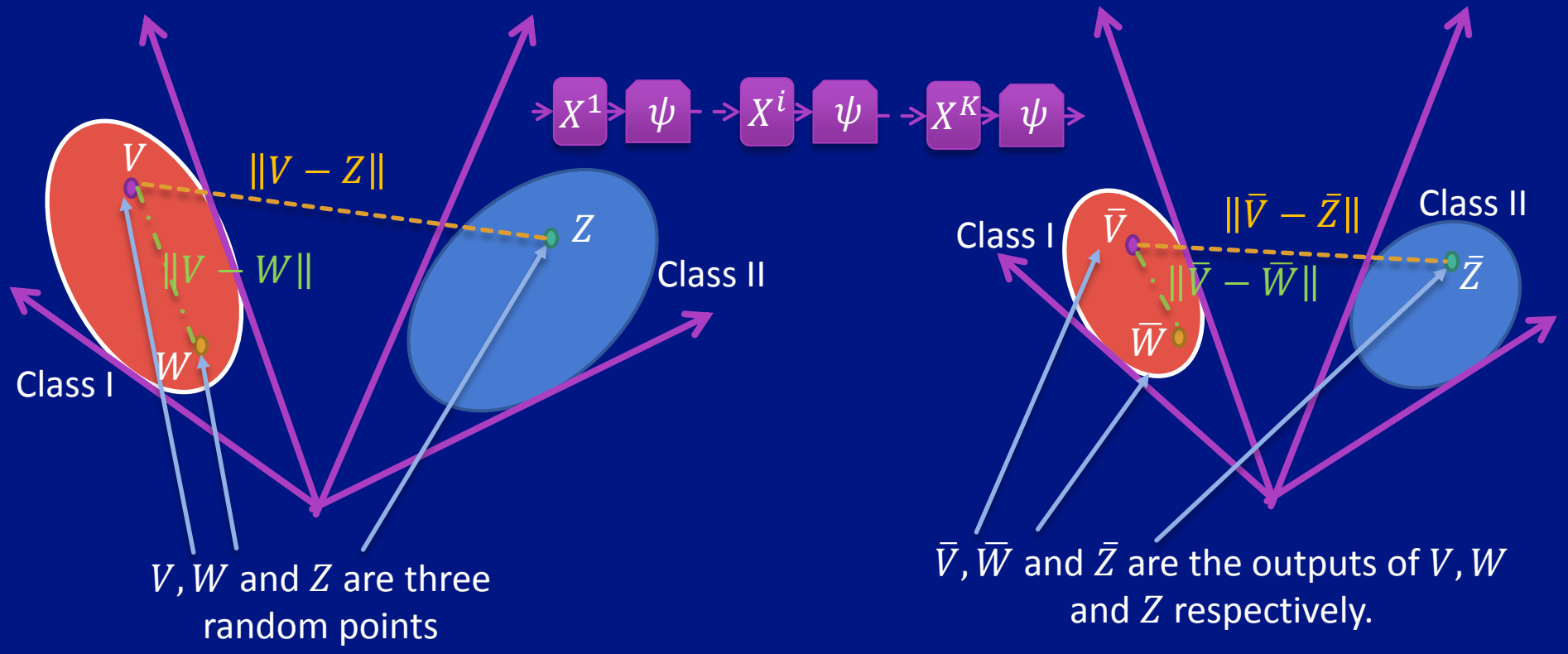
$$\frac{\|\bar{V} - \bar{Z}\|}{\|W - V\|}$$

Intra-class



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

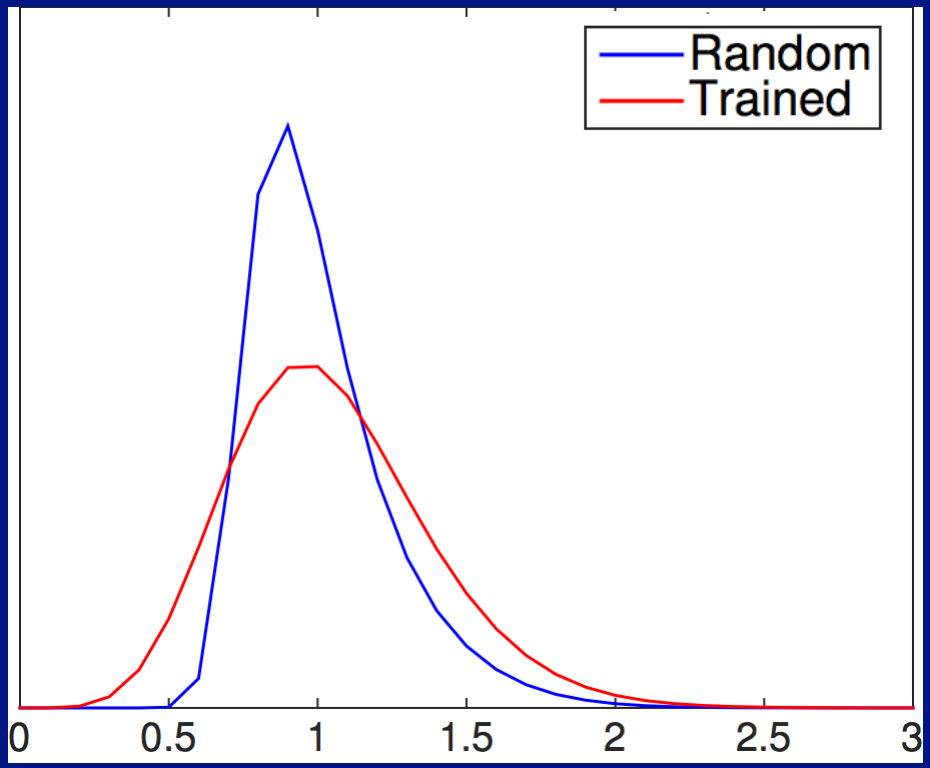
AVERAGE POINTS DISTANCE RATIO



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

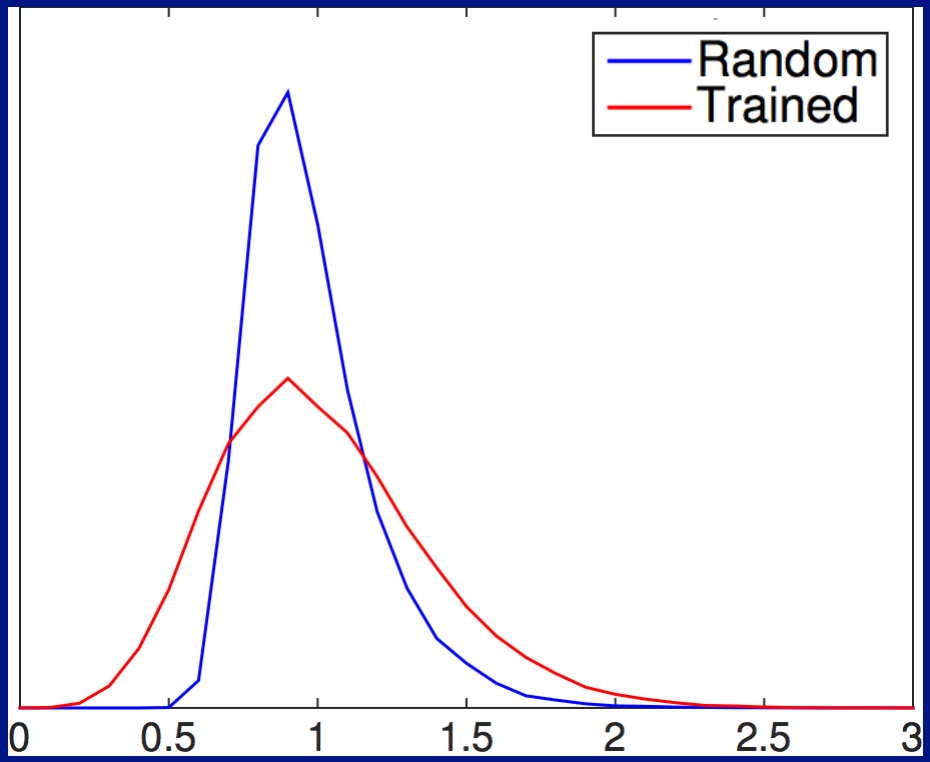
AVERAGE DISTANCE RATIO

Inter-class



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

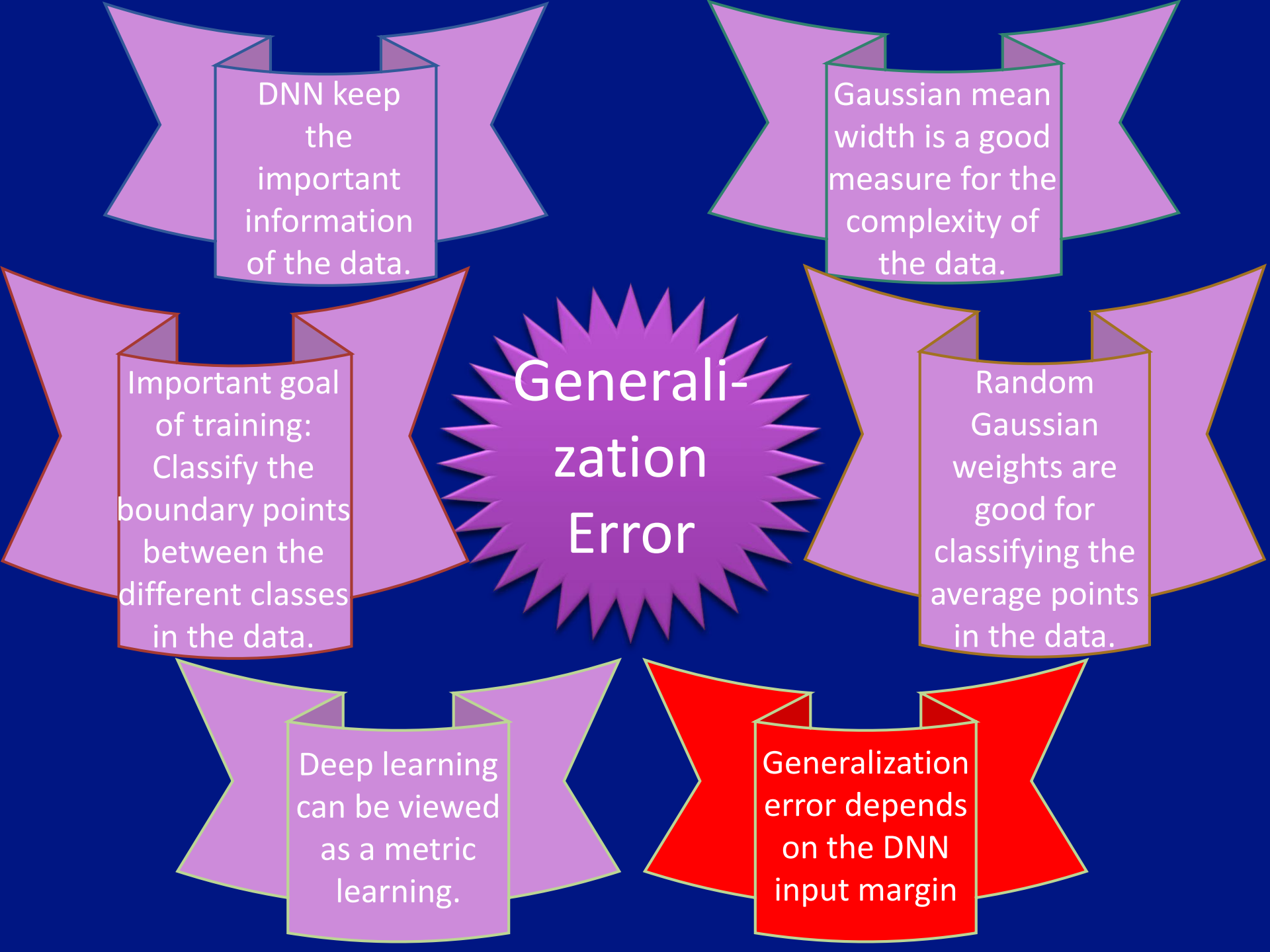
Intra-class



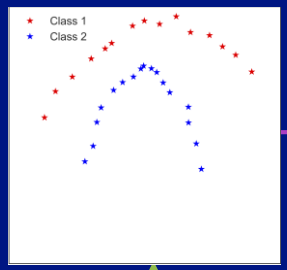
$$\frac{\|\bar{V}-\bar{W}\|}{\|V-W\|}$$

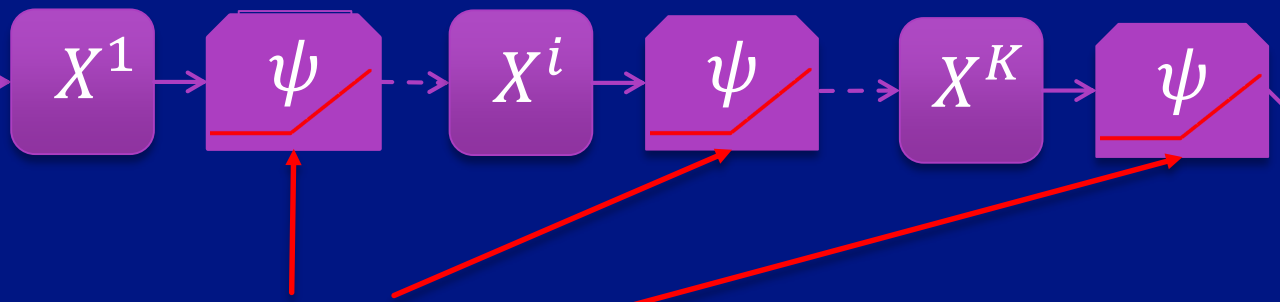
ROLE OF TRAINING

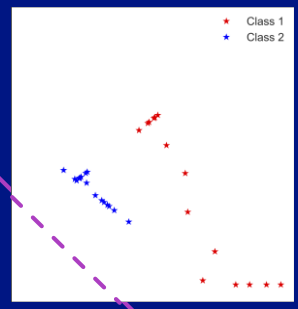
- On average distances are preserved in the trained and random networks.
- The difference is with respect to the boundary points.
- The inter distances become larger.
- The intra distances shrink.


DNN 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.
Generali-zation Error
Random Gaussian weights are good for classifying the average points in the data.
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

ASSUMPTIONS



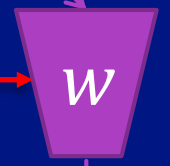


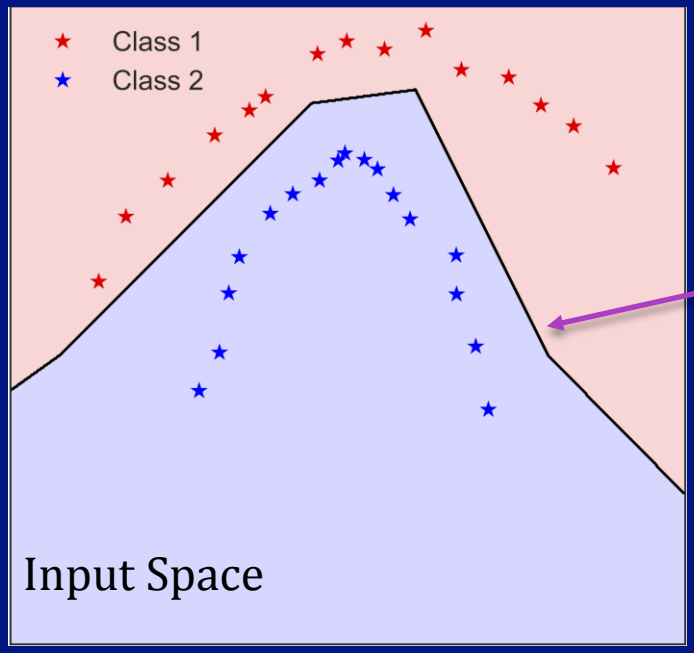


Two Classes $\in \Upsilon$

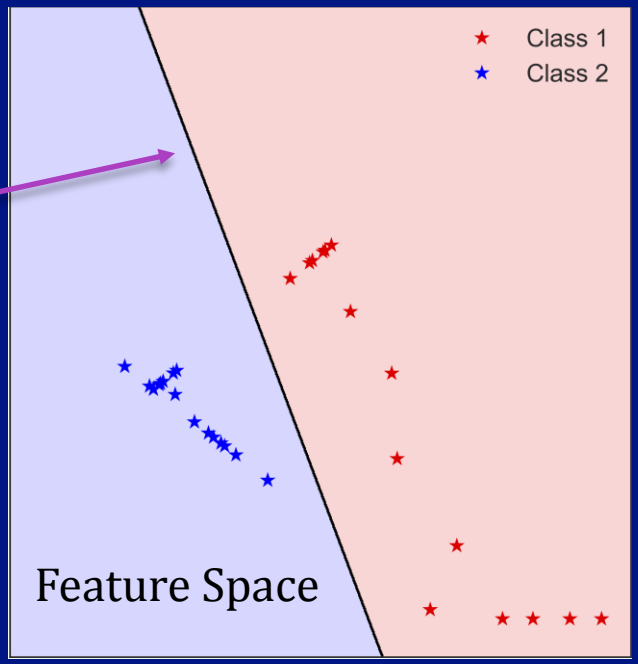
ψ is the ReLU

Linear classifier





$w^T \Phi(X^1, X^2, \ldots, X^K) = 0$



GENERALIZATION ERROR (GE)

- In training, we reduce the classification error of the training data ℓ_{training} as the number of training examples L increases.
- However, we are interested to reduce the error of the (unknown) testing data ℓ_{test} as L increases.
- The difference between the two is the generalization error

$$\text{GE} = \ell_{\text{training}} - \ell_{\text{test}}$$

⇒ It is important to understand the GE of DNN

GE BOUNDS

- Using the VC dimension it can be shown that

$$\mathrm{GE} \leq O\left(\sqrt{\mathrm{DNN\ params} \cdot \frac{\log(L)}{L}}\right)$$

[Shalev-Shwartz and Ben-David, 2014].

- The GE was bounded also by the DNN weights

$$\mathrm{GE} \leq \frac{1}{\sqrt{L}} 2^K \|w\|_2 \prod_i \left\|X^i\right\|_{2,2}$$

[Neyshabur et al., 2015].

GE BOUNDS

- Using the VC dimension it can be shown that

$$\mathrm{GE} \leq O\left(\sqrt{\text{DNN params} \cdot \frac{\log(L)}{L}}\right)$$

[Shalev-Shwartz and Ben-David, 2014].

- The GE was bounded also by the DNN weights

$$\mathrm{GE} \leq \frac{1}{\sqrt{L}} 2^{K} \|w\|_2 \prod_i \left\|X^i\right\|_{2,2}$$

[Neyshabur et al., 2015].

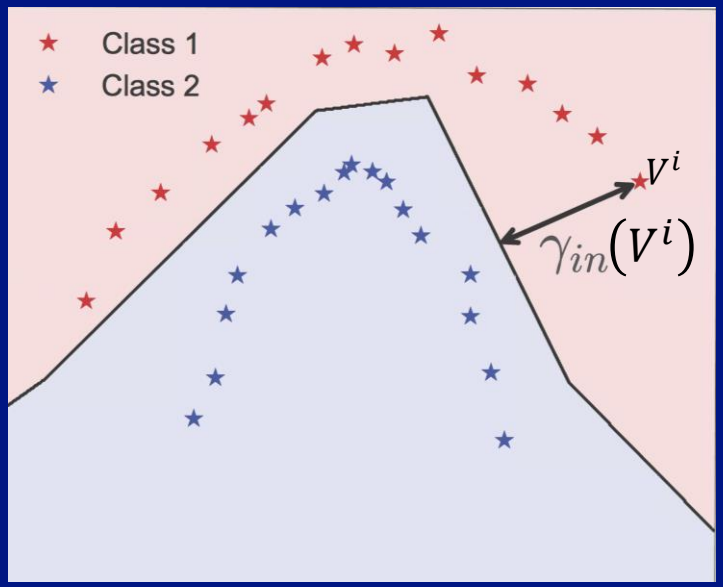
- Note that in both cases the GE grows with the depth

DNN INPUT MARGIN

- Theorem 6: If for every input margin $\gamma_{in}(V^i) > \gamma$

  then $GE \leq \sqrt{N_{\gamma/2}(\Upsilon)/\sqrt{m}}$ [Sokolic, Giryes, Sapiro, Rodrigues, 2016]

- $N_{\gamma/2}(\Upsilon)$ is the covering number of the data Υ.
- $N_{\gamma/2}(\Upsilon)$ gets smaller as γ gets smaller
- Bound is independent of depth
- Our theory relies on the robustness framework [Xu & Mannor, 2015].



INPUT MARGIN BOUND

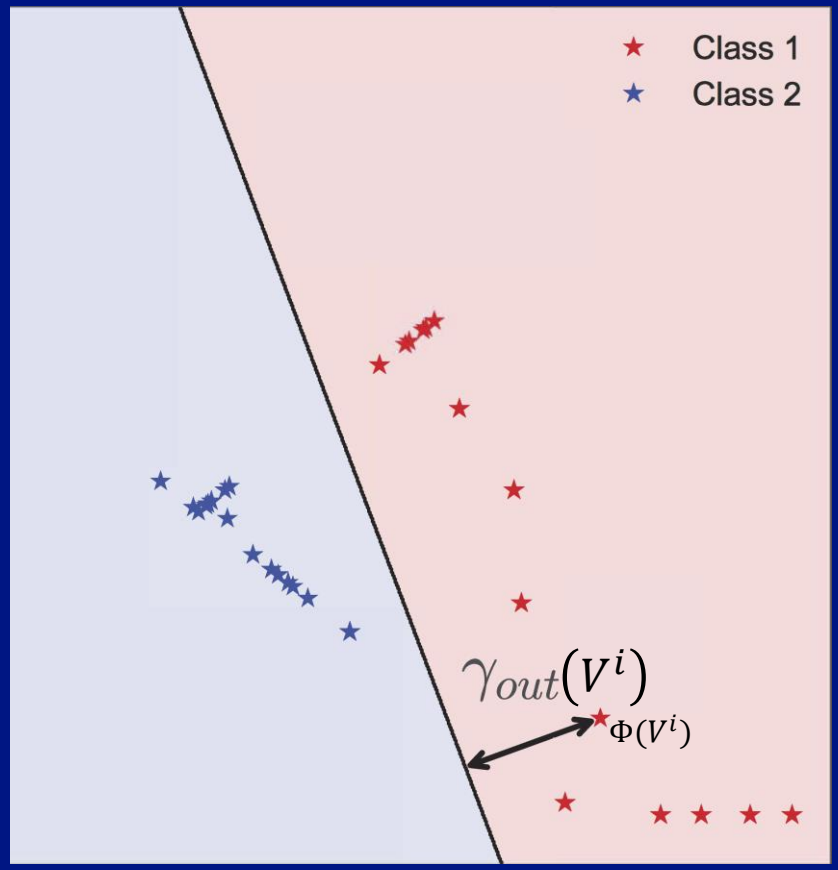
- Maximizing the input margin directly is hard
- Our strategy: relate the input margin to the output margin $\gamma_{out}(V^i)$ and other DNN properties
- Theorem 7:

$$\gamma_{in}(V^i) \geqslant \frac{\gamma_{out}(V^i)}{\sup_{V \in \mathcal{Y}} \left\| \frac{V}{\|V\|_2} J(V) \right\|_2}$$

$$\geqslant \frac{\gamma_{out}(V^i)}{\prod_{1 \leqslant i \leqslant K} \|X^i\|_2}$$

$$\geqslant \frac{\gamma_{out}(V^i)}{\prod_{1 \leqslant i \leqslant K} \|X^i\|_F}$$

[Sokolic, Giryes, Sapiro, Rodrigues, 2016]

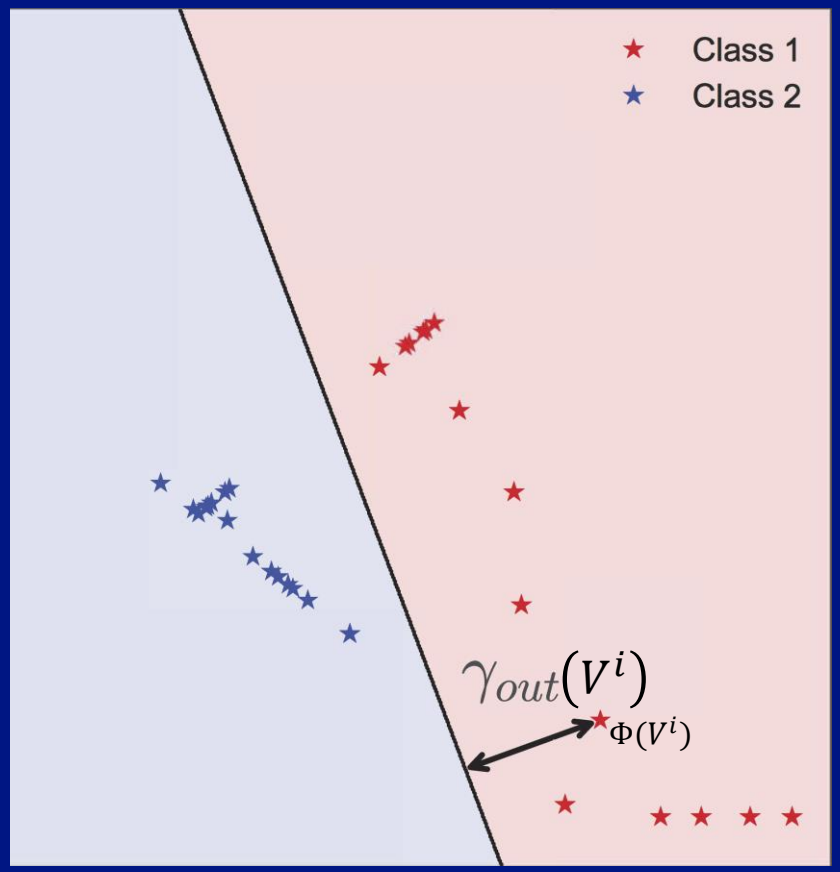


OUTPUT MARGIN

- Theorem 7: $\gamma_{in}(V^i) \geq \frac{\gamma_{out}(V^i)}{\sup_{V \in \mathcal{Y}} \left\| \frac{V}{\|V\|_2} J(V) \right\|_2} \geq$

$$\frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_2} \geq \frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_F}$$

- Output margin is easier to maximize – SVM problem
- Maximized by many cost functions, e.g., hinge loss.

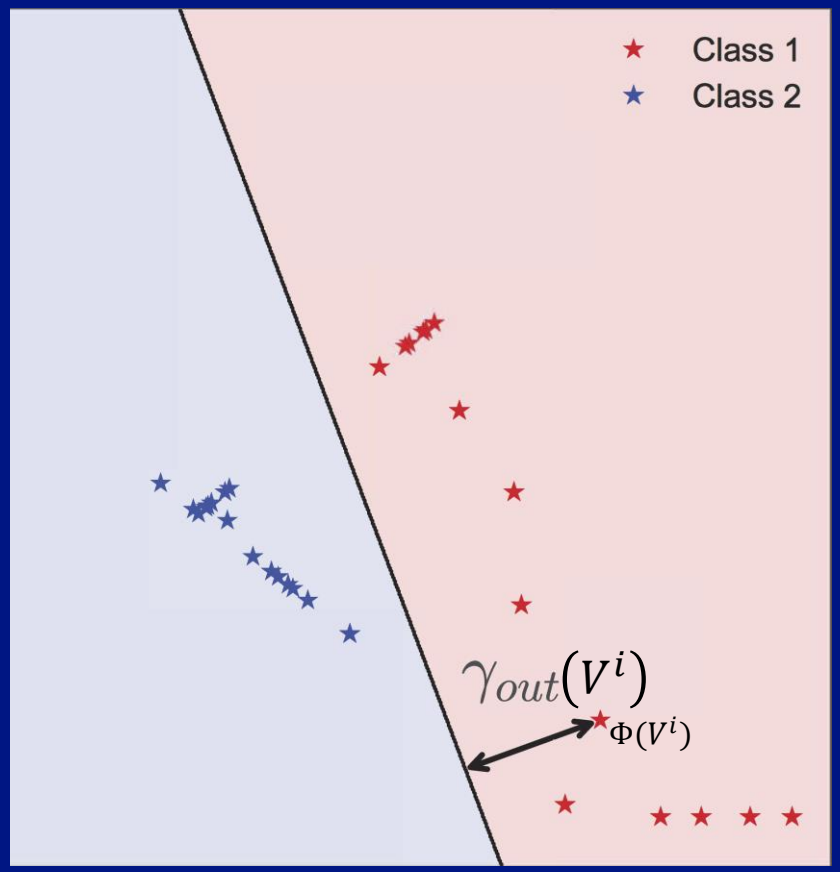


GE AND WEIGHT DECAY

- Theorem 7: $\gamma_{in}(V^i) \geq \frac{\gamma_{out}(V^i)}{\sup_{V \in \mathcal{V}} \left\| \frac{V}{\|V\|_2} J(V) \right\|_2} \geq$

$$\frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_2} \geq \frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_F}$$

- Bounding the weights increases the input margin
- Weight decay regularization decreases the GE
- Related to regularization used by [Haeffele & Vidal, 2015]

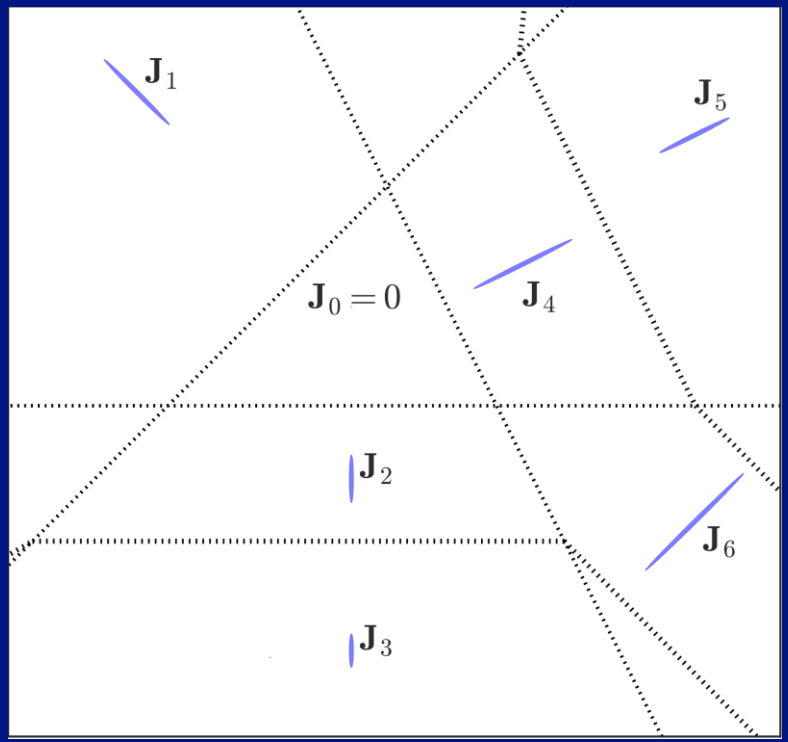


JACOBIAN BASED REGULARIZATION

- Theorem 7: $\gamma_{in}(V^i) \geq \frac{\gamma_{out}(V^i)}{\sup_{V \in \Upsilon} \left\| \frac{V}{\|V\|_2} J(V) \right\|_2} \geq$

$$\frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_2} \geq \frac{\gamma_{out}(V^i)}{\prod_{1 \leq i \leq K} \|X^i\|_F}$$

- $J(V)$ is the Jacobian of the DNN at point V.
- $J(\cdot)$ is piecewise constant.
- Using the Jacobian of the DNN leads to a better bound.
- ⇨ New regularization technique.



RESULTS

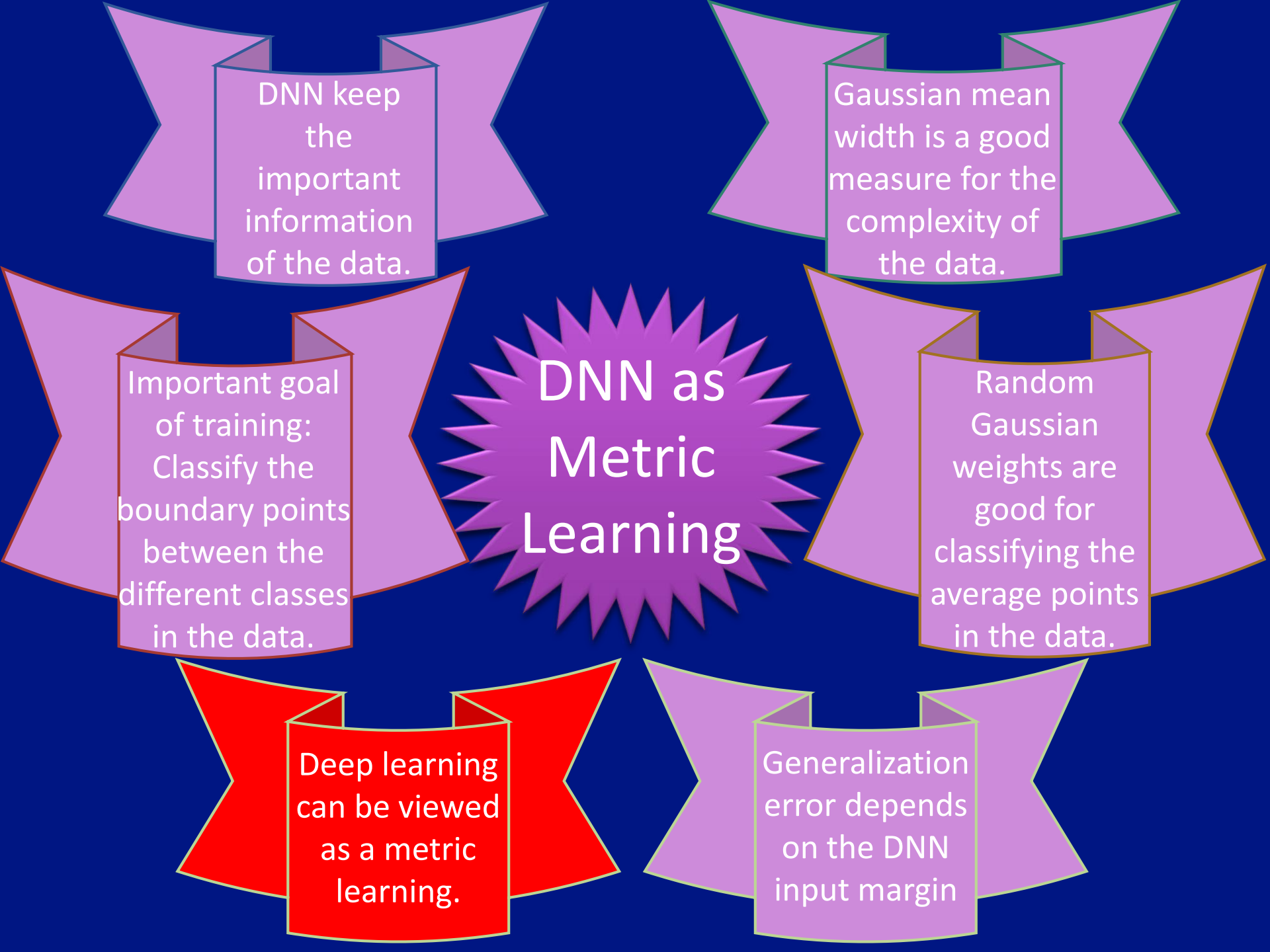
- Better performance with less training samples

MNIST Dataset

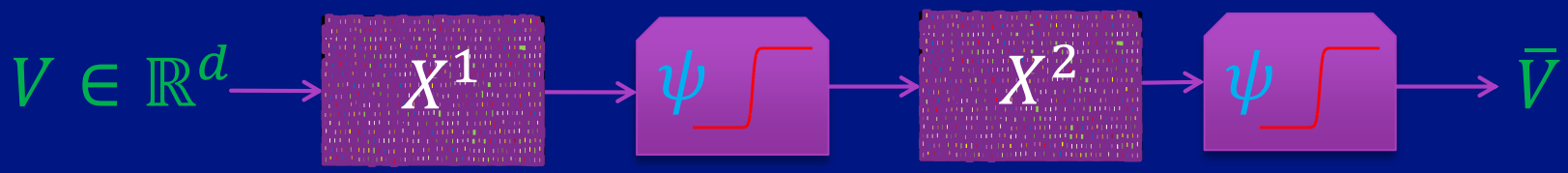
		256 samples			512 samples			1024 samples		
loss	# layers	no reg.	WD	LM	no reg.	WD	LM	no reg.	WD	LM
hinge	2	88.37	89.88	**93.83**	93.99	94.62	95.49	95.79	96.57	97.45
hinge	3	87.22	89.31	93.22	93.41	93.97	**95.76**	95.46	96.45	**97.60**
CCE	2	88.45	88.45	92.77	92.29	93.14	95.25	95.38	95.79	96.89
CCE	3	89.05	89.05	93.10	91.81	93.02	95.32	95.11	95.86	97.14

- CCE: the categorical cross entropy.
- WD: weight decay regularization.
- LM: Jacobian based regularization for large margin.
- Note that hinge loss generalizes better than CCE and that LM is better than WD as predicted by our theory.

[Sokolic, Giryes, Sapiro, Rodrigues, 2016]


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.
DNN as Metric Learning
Random Gaussian weights are good for classifying the average points in the data.
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

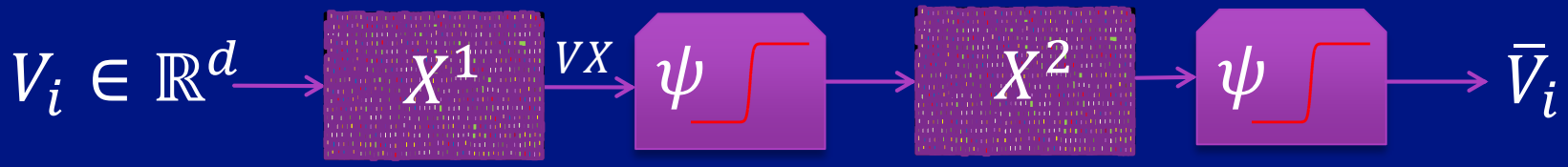
ASSUMPTIONS



X is fully connected and trained

ψ is the hyperbolic tan

METRIC LEARNING BASED TRAINING



- Cosine Objective:

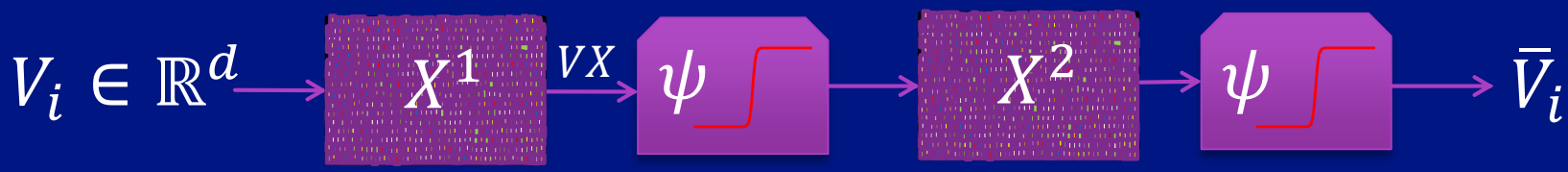
$$\min_{X^1,X^2} \sum_{i,j \in Training\ Set} \left(\frac{\bar{V}_i{}^T \bar{V}_j}{\|\bar{V}_i\| \|\bar{V}_j\|} - \vartheta_{i,j} \right)^2$$

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



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

- Euclidean Objective:

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

Classification term

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

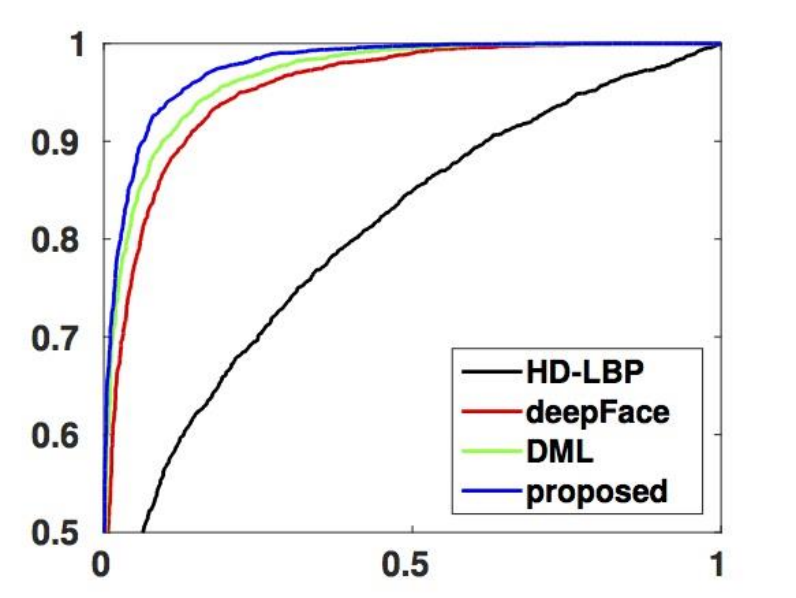
RESULTS

- 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]


DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Take Home Message
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

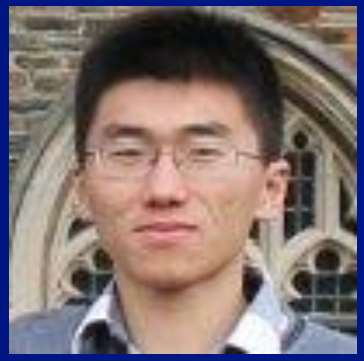
Alex Bronstein
Technion

Guillermo Sapiro
Duke University

Robert Calderbank
Duke University

Miguel Rodrigues
University College London

Qiang Qiu
Duke University



Jiaji Huang
Duke University

Jure Sokolic
University College London

Grants

NSF **NSSEFF**

ONR **ARO**

NGA **ERC**

QUESTIONS?

WEB.ENG.TAU.AC.IL/~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*

J. SOKOLIC, R. GIRYES, G. SAPIRO, M. R. D. RODRIGUES, MARGIN PRESERVATION OF DEEP NEURAL NETWORKS