DATA STRUCTURE BASED THEORY FOR DEEP LEARNING

RAJA GIRYES

TEL AVIV UNIVERSITY

Mathematics of Deep Learning Tutorial

EUSIPCO Conference, Budapest, Hungary

August 29, 2016

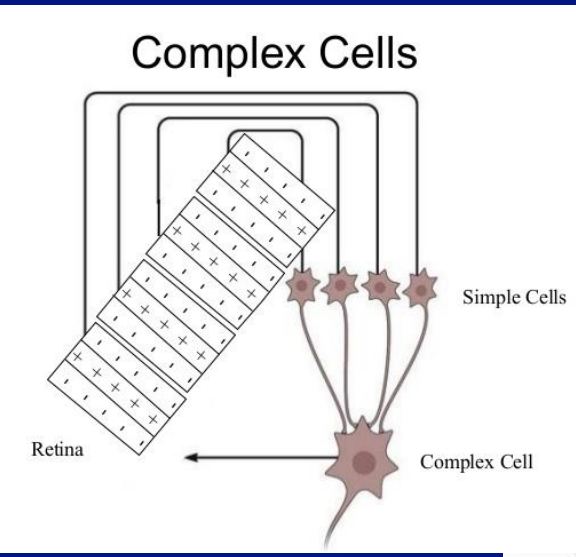
AGENDA

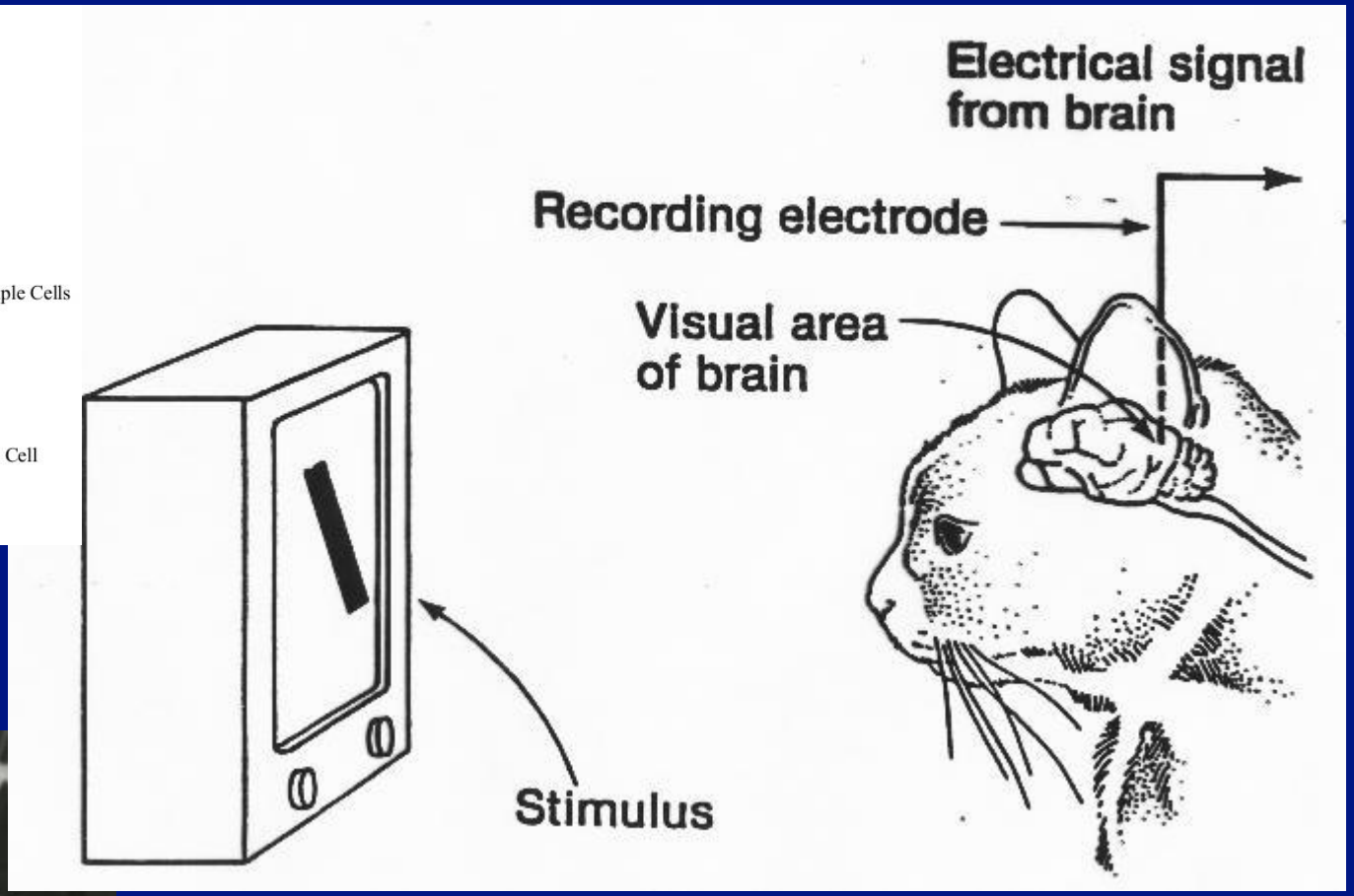
- History of deep learning.
- A sample of existing theory for deep learning.
- Neural networks with random Gaussian weights.
- Generalization error of deep neural networks.
- Deep Learning as metric learning.
- Solving minimization problems with deep learning.

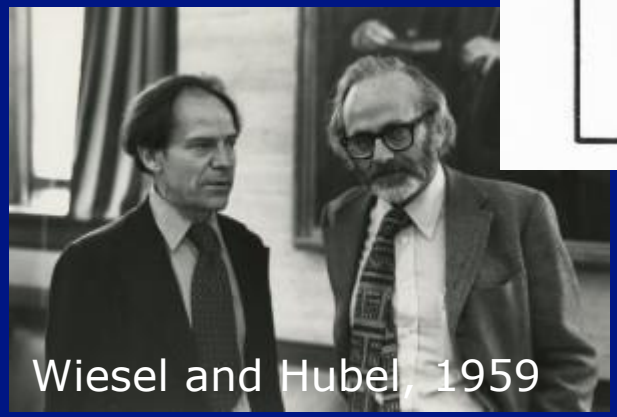
FIRST LEARNING PROGRAM

"field of study that gives computers the ability to learn without being explicitly programmed". [Arthur Samuel, 1959]

RELATION TO VISION

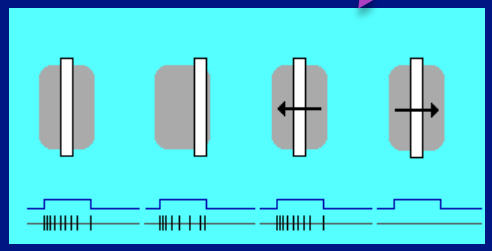


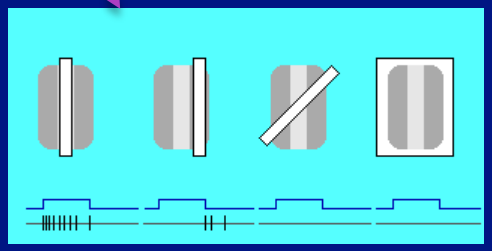


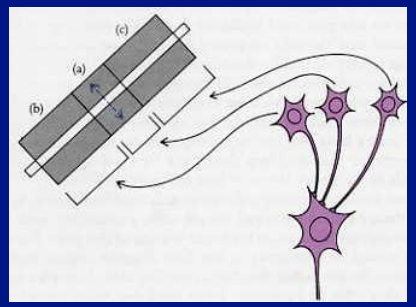

Wiesel and Hubel, 1959

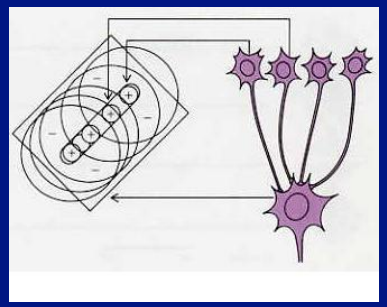
HUMAN VISUAL SYSTEM

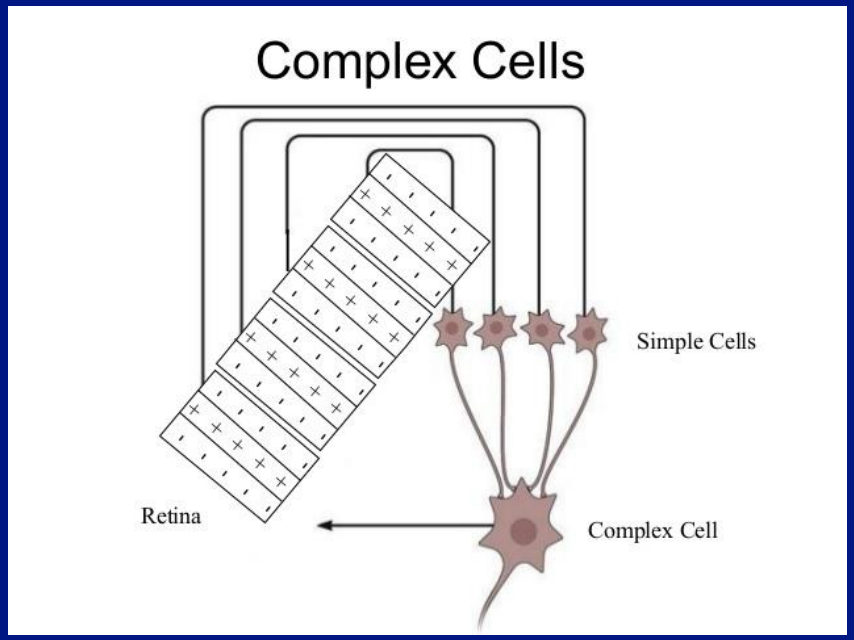
In the visual cortex there are two types of neurons: Simple and complex



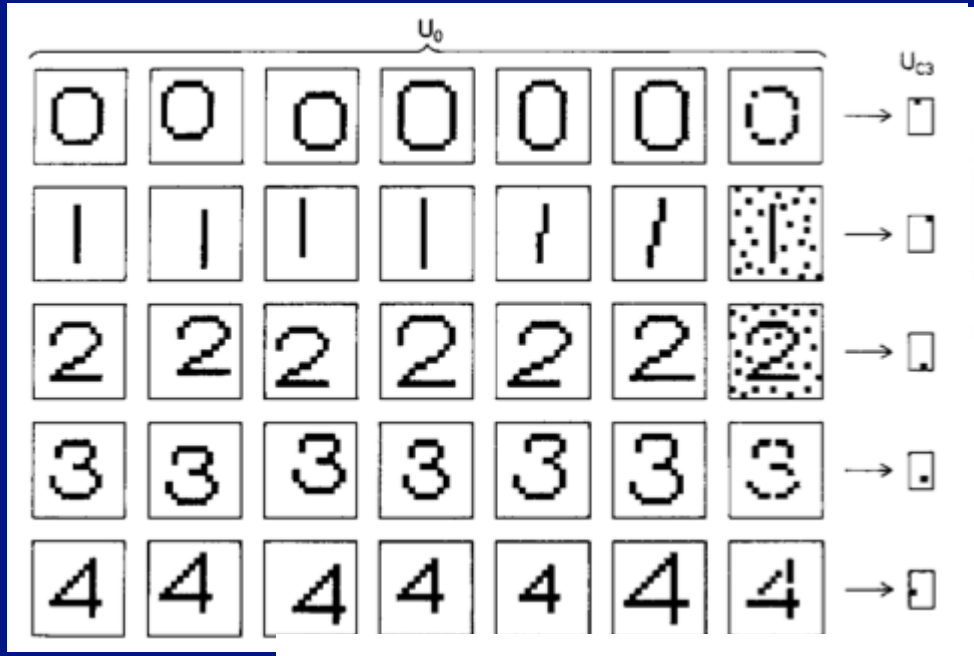


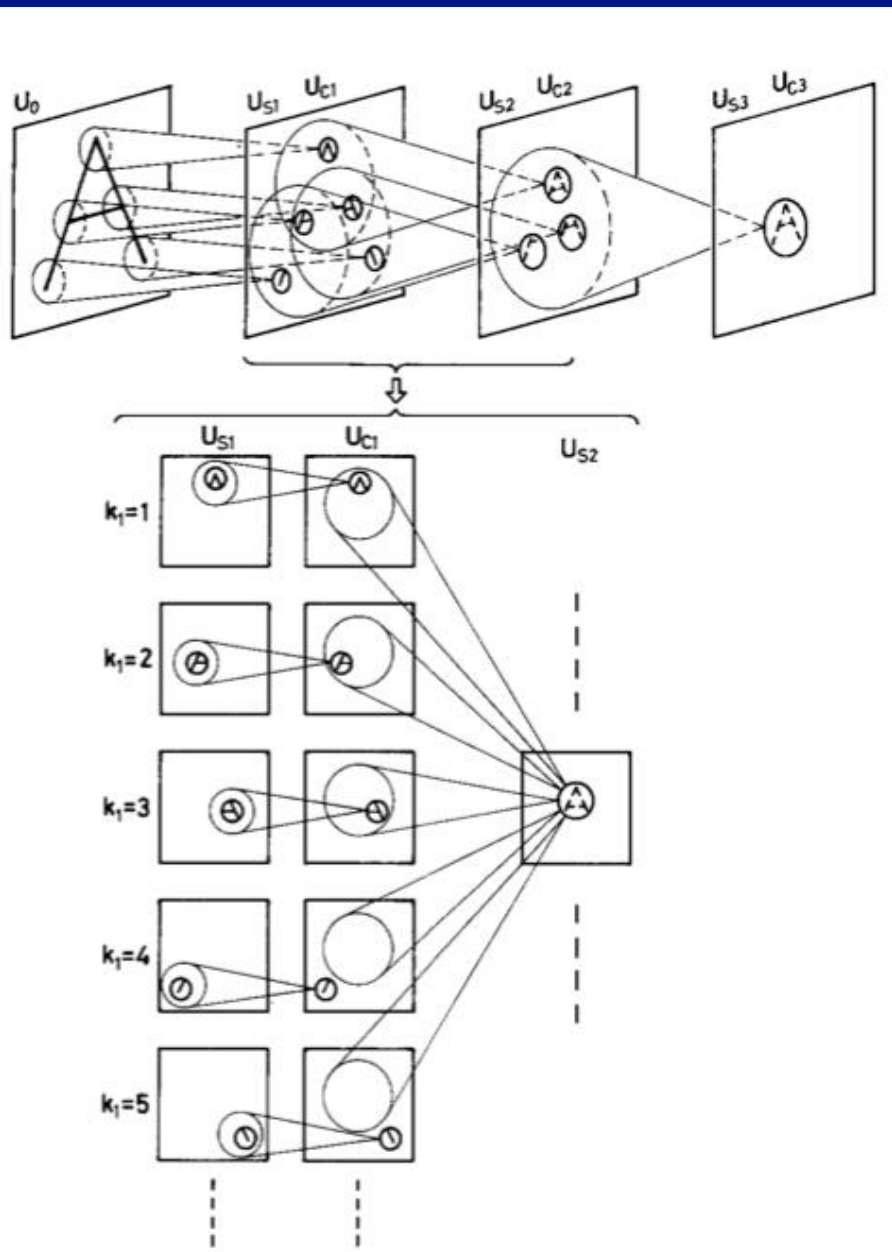


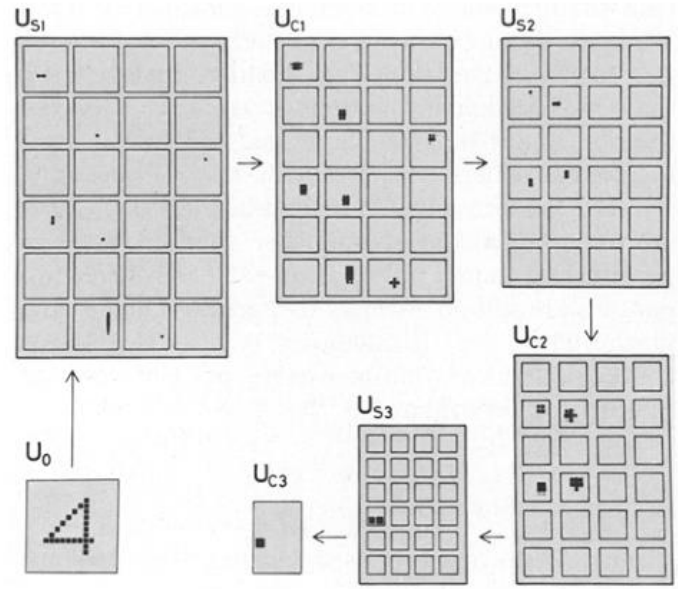




IMITATING THE HUMAN BRAIN

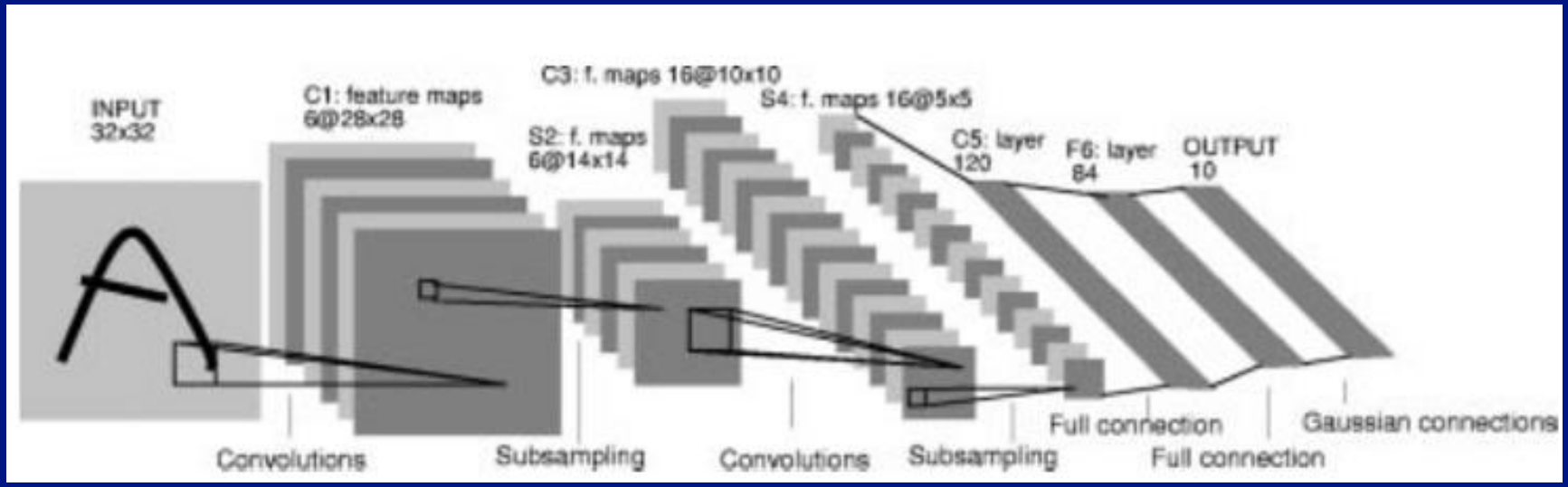






Fukushima 1980

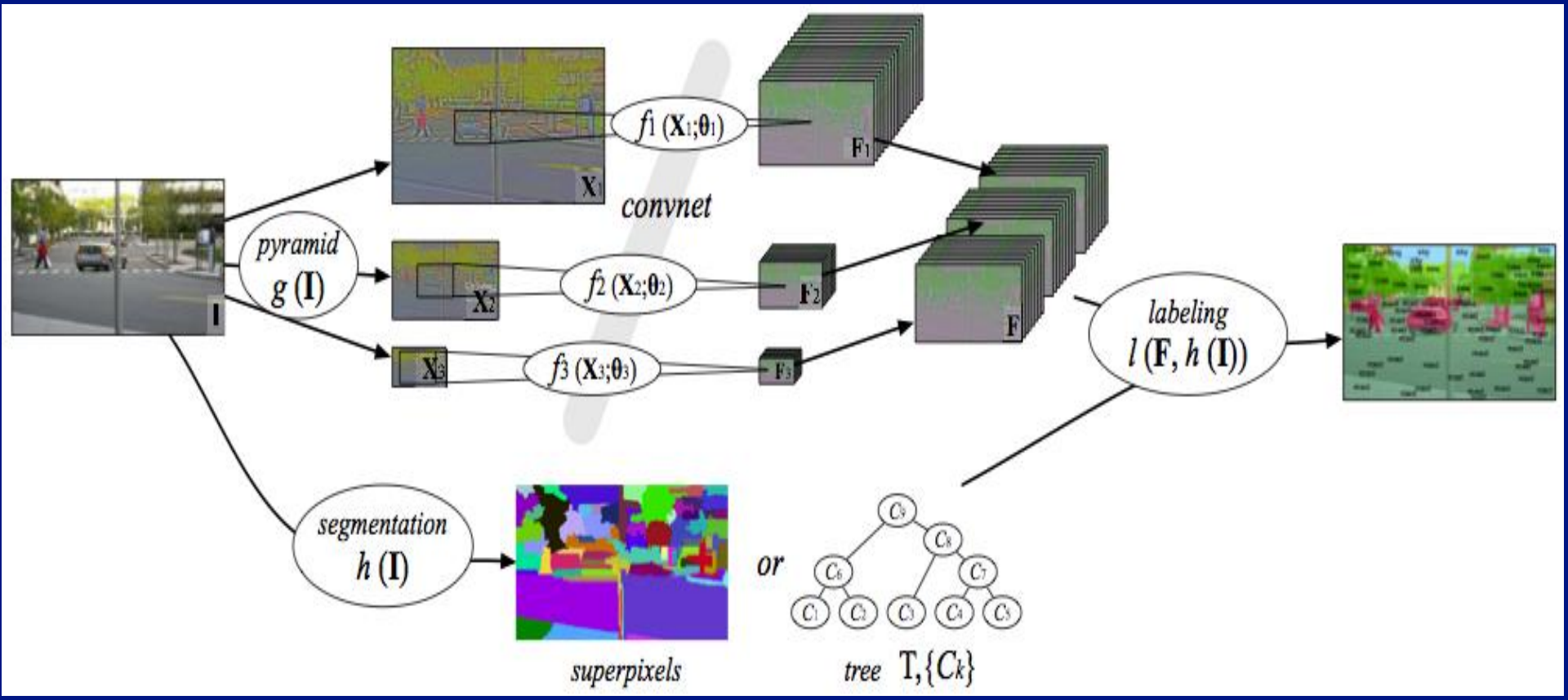
CONVOLUTIONAL NEURAL NETWORKS



- Introduction of convolutional neural networks [LeCun et. al. 1989]
- Training by backpropagation

SCENE PARSING

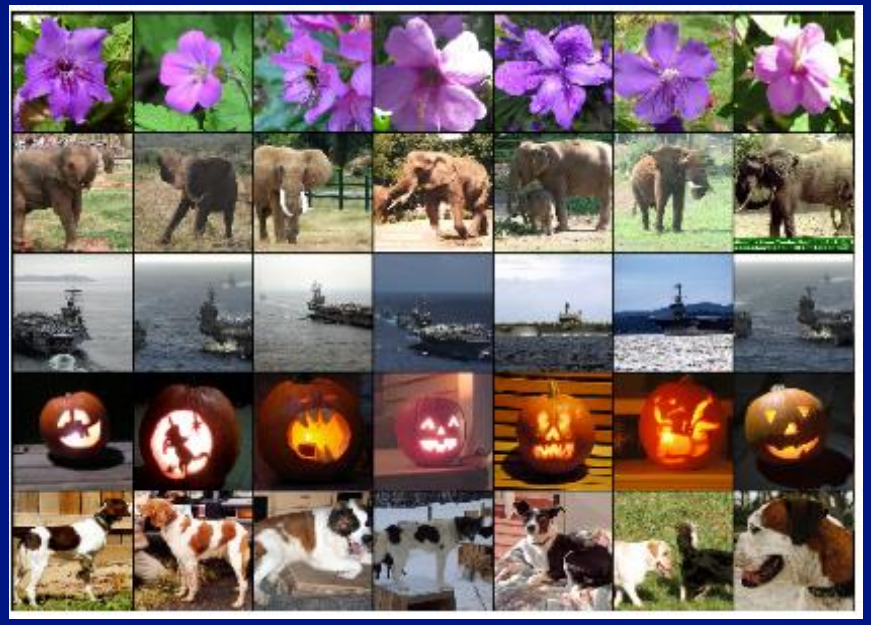
- Deep Learning usage before 2012:



[Farabet et al., 2012, 2013]

2012 IMAGENET DEEP LEARNING BREAKTHROUGH

- Imagenet dataset
- 1.4 Million images
- 1000 categories
- 1.2 Million for training
- 150000 for testing
- 50000 for validation

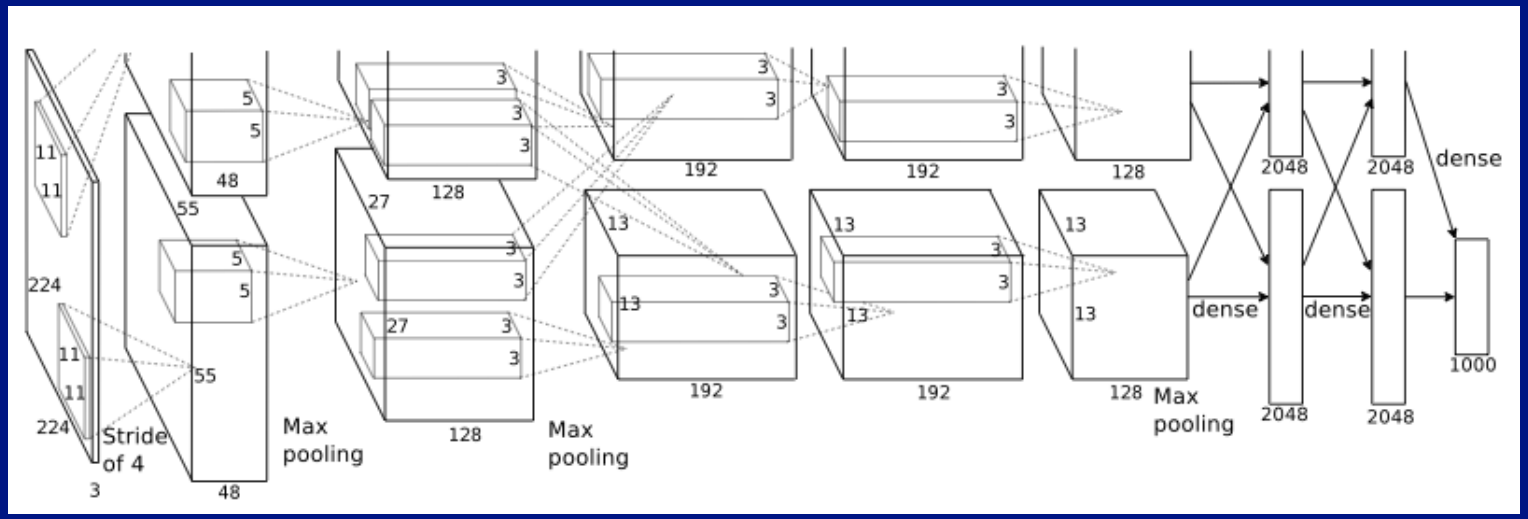


Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

[Krizhevsky, Sutskever & Hinton, 2012]

Today deep learning achieves 3.5% by 152 layers [He, Zhang, Ren & Sun, 2016]

DEEP NETWORK STRUCTURE

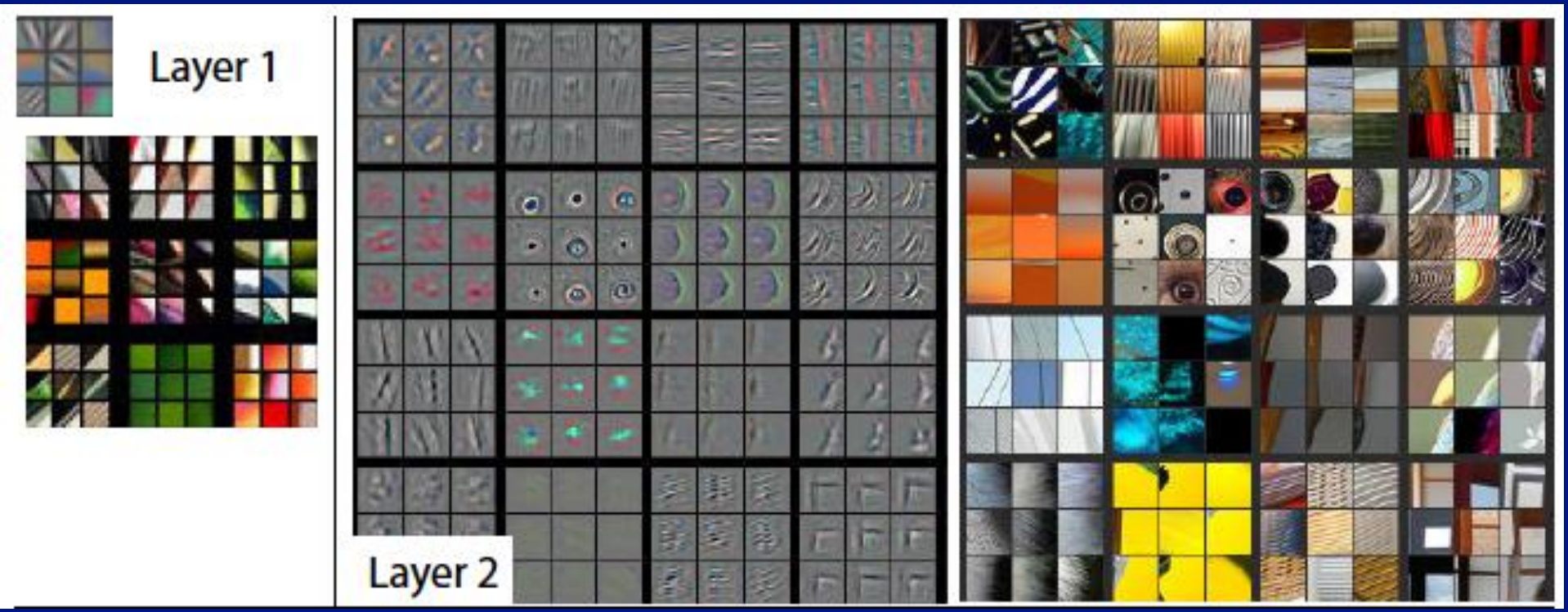


[Krizhevsky, Sutskever & Hinton, 2012]

- What each layer of the network learns?

LAYERS STRUCTURE

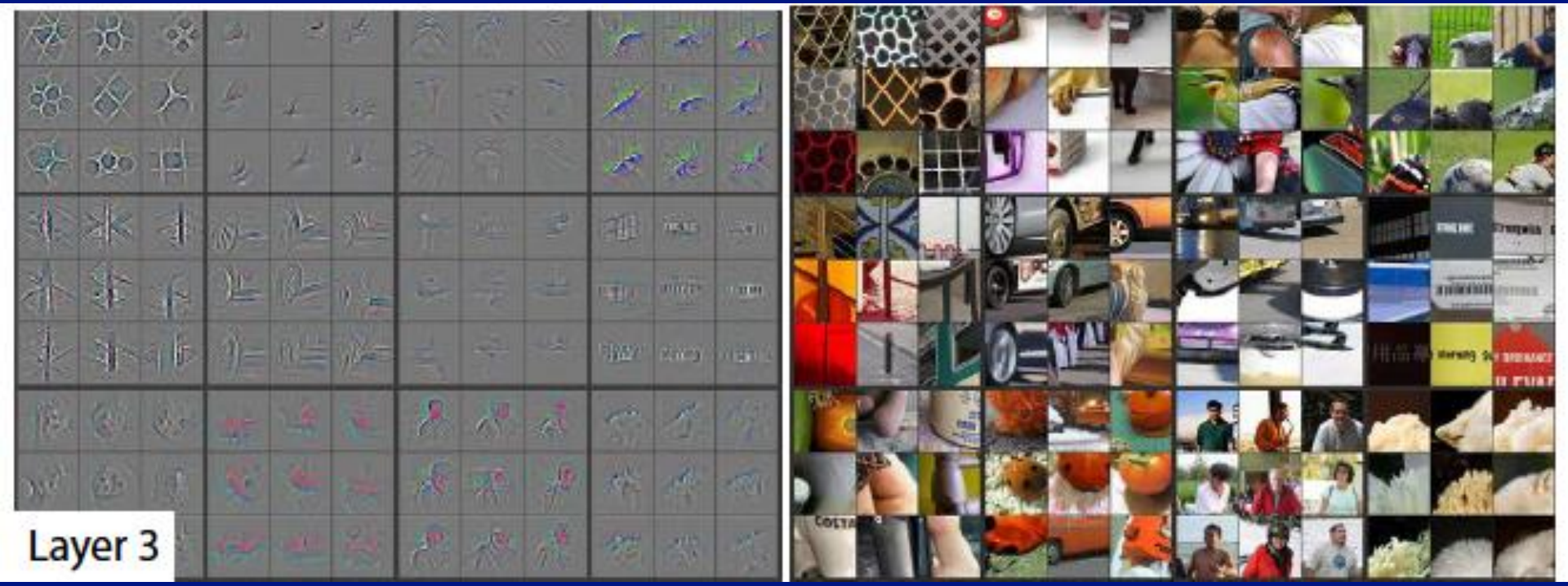
- First layers detect simple patterns that corresponds to simple objects



[Zeiler & Fergus, 2014]

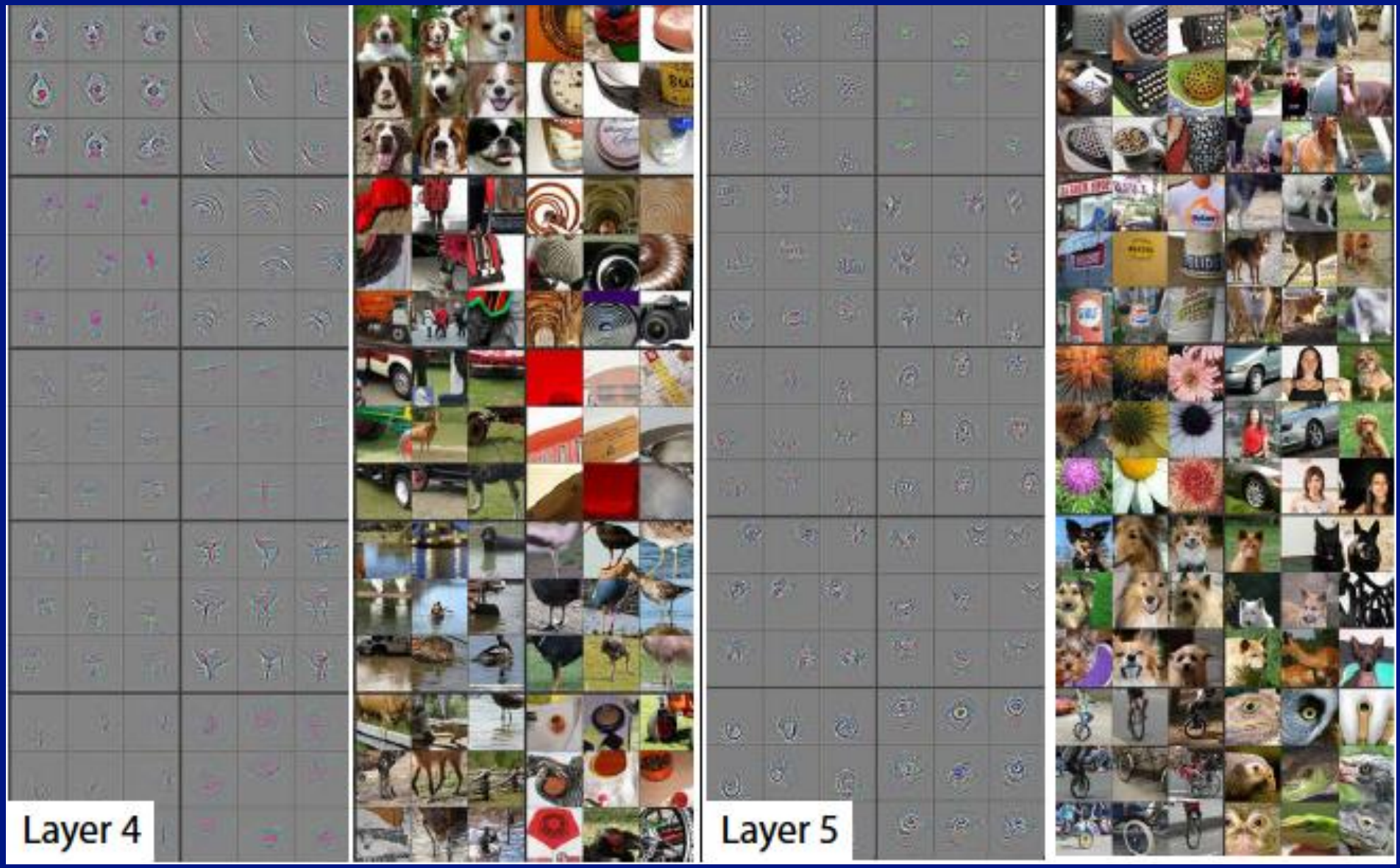
LAYERS STRUCTURE

- Deeper layers detects more complex patterns corresponding to more complex objects.



[Zeiler & Fergus, 2014]

LAYERS STRUCTURE



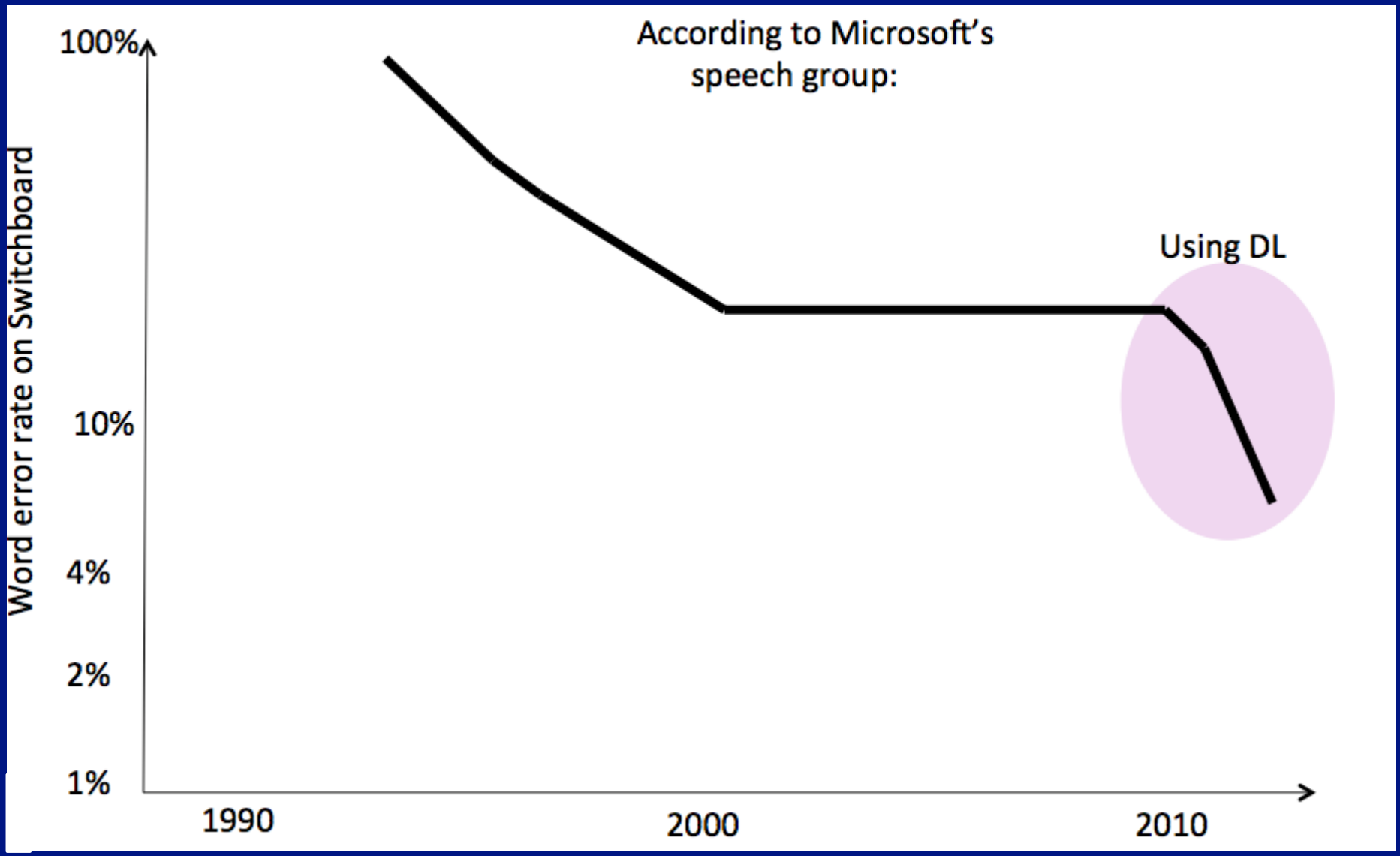
[Zeiler & Fergus, 2014]

WHY THINGS WORK BETTER TODAY?

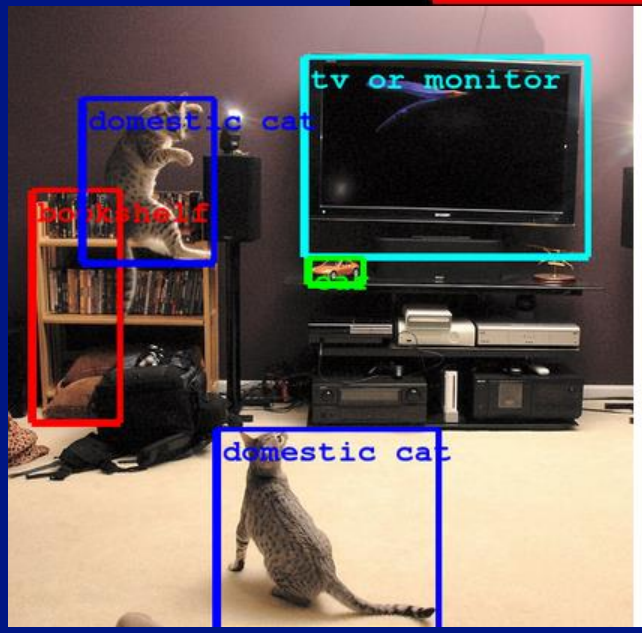
- More data
- Better Hardware (GPU)
- Better learning regularization (dropout)

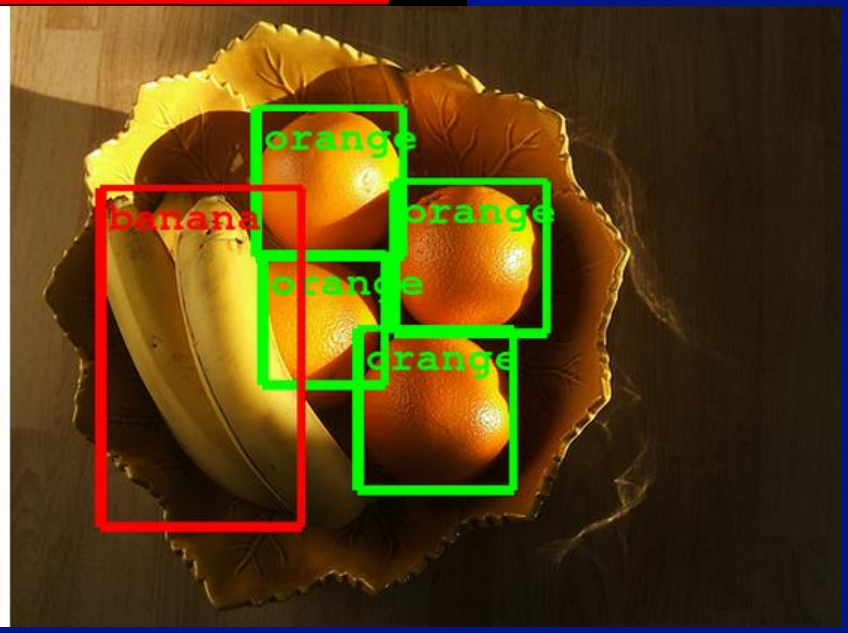
- Deep learning impact and success is not unique only to image classification.

DEEP LEARNING FOR SPEECH RECOGNITION



OBJECT DETECTION





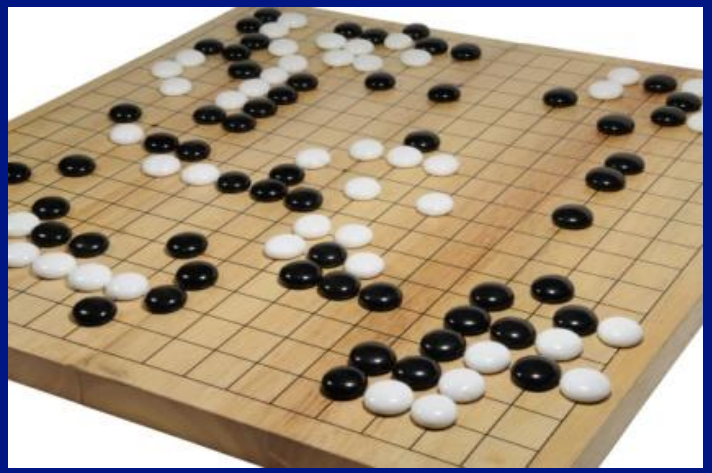
[Szegedy et al., 2015]

GAME PLAYING

[Mnih et al., 2013, 2015]

GO GAME

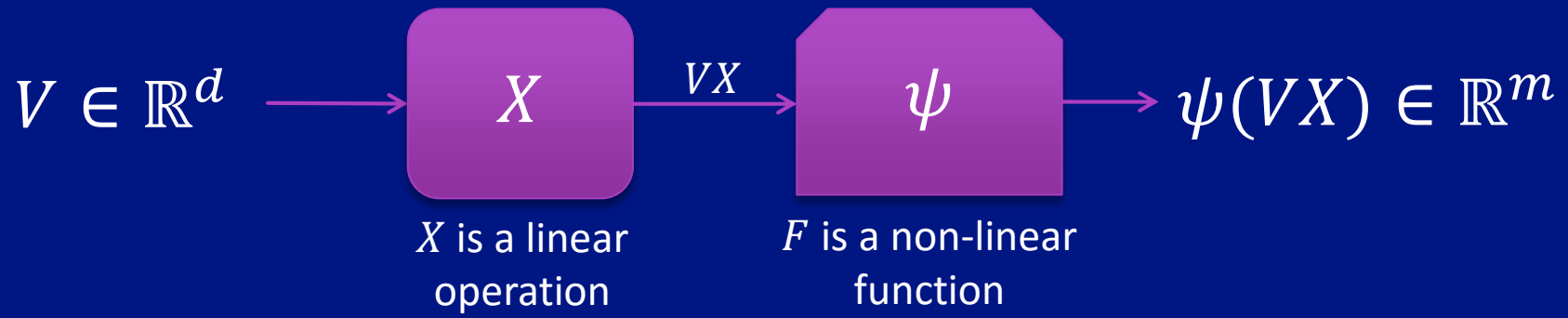
- AlphaGo - First computer program to ever beat a professional player at the game of go



- Program created by Google DeepMind
- Game strategy learned using deep learning [Silver et al., 2016].

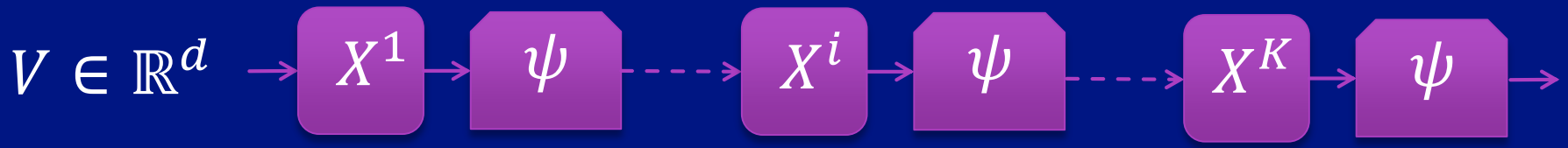
DEEP NEURAL NETWORKS (DNN)

- One layer of a neural net

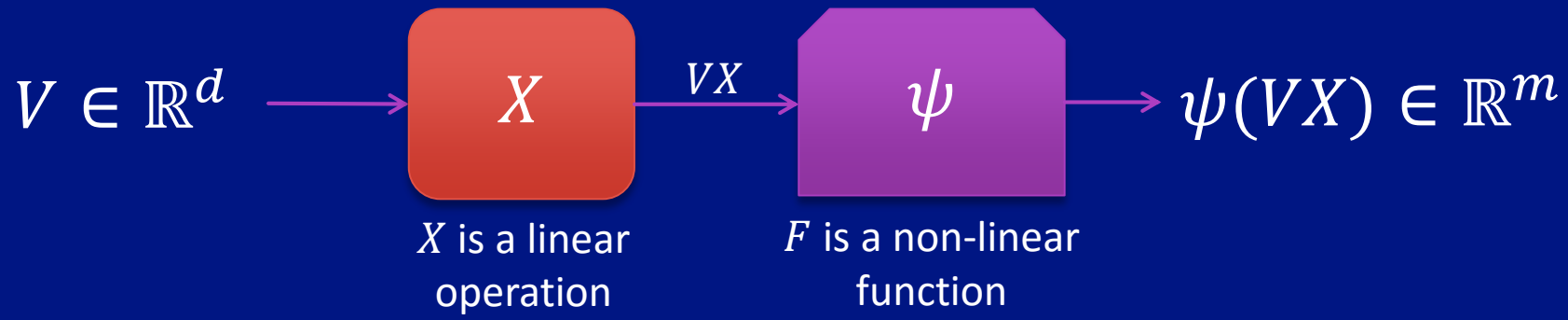


- Concatenation of the layers creates the whole net

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



CONVOLUTIONAL NEURAL NETWORKS (CNN)

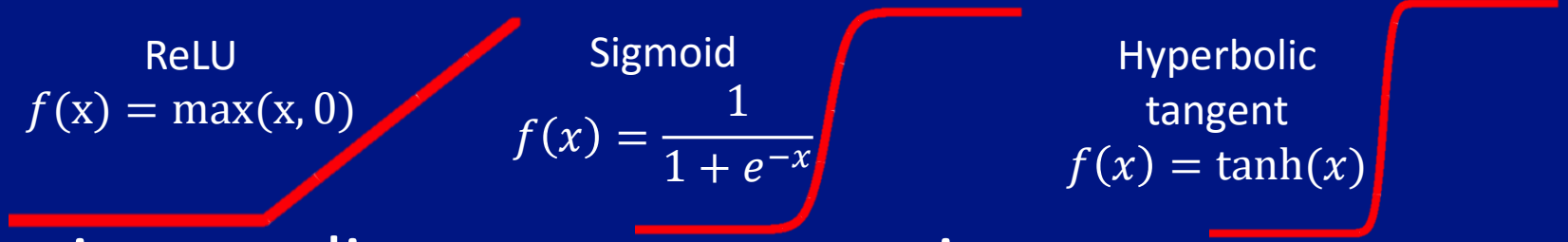


- In many cases, X is selected to be a convolution.
- This operator is shift invariant.
- CNN are commonly used with images as they are typically shift invariant.

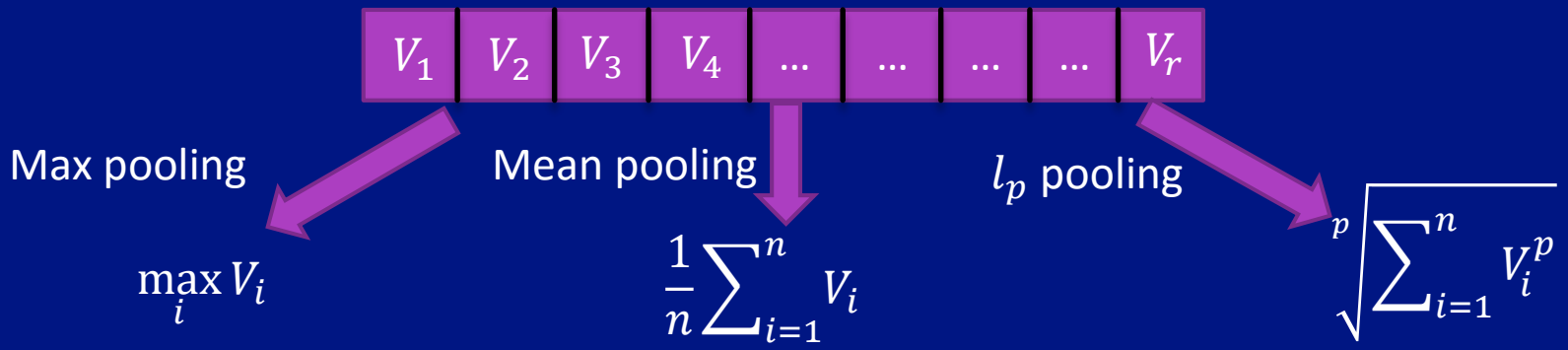
THE NON-LINEAR PART

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

- f is the (point-wise) activation function



- g is a pooling or an aggregation operator.



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?

REPRESENTATION POWER

- Neural nets serve as a universal approximation for any measurable Borel functions [Cybenko 1989, Hornik 1991].
- In particular, let the non-linearity ψ be a bounded, non-constant continuous function, I_m be the m-dimensional hypercube, and $C(I_m)$ be the space of continuous functions on I_m. Then for any $f \in C(I_m)$ and $\epsilon > 0$, there exists $m > 0$, and $X \in \mathbb{R}^{d \times m}$, $B \in \mathbb{R}^m$, $W \in \mathbb{R}^m$ such that the neural network

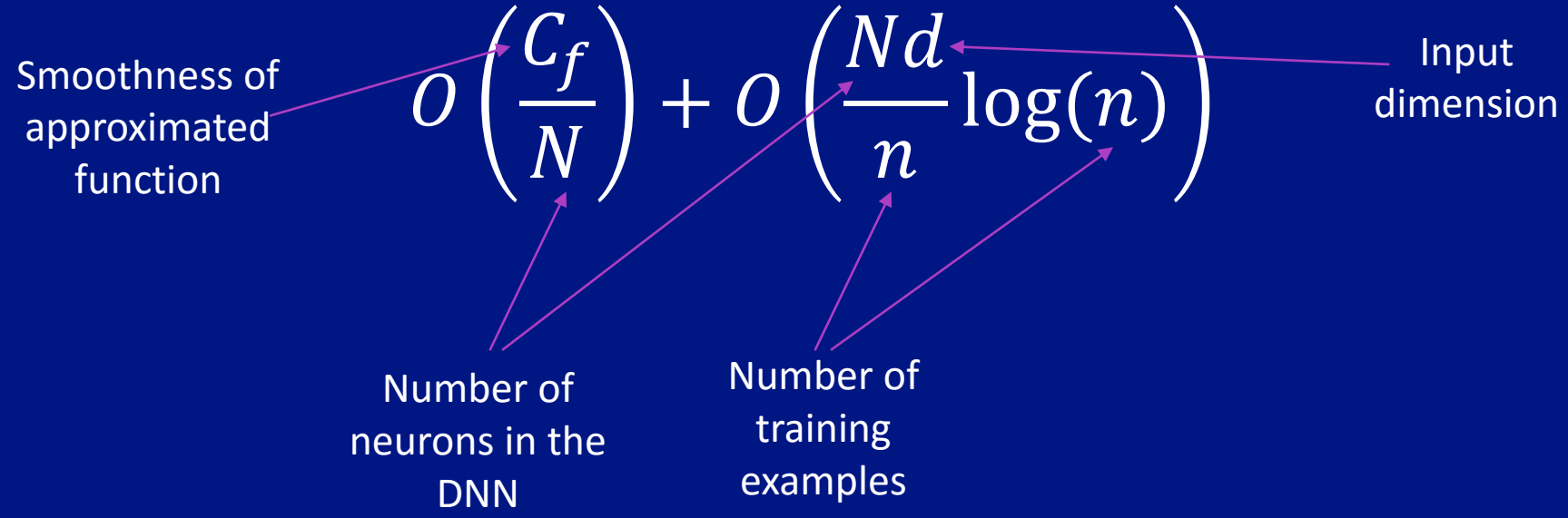
$$F(V) = \psi(VX)W^T$$

approximates f with a precision ϵ:

$$|F(V) - f(V)| < \epsilon, \forall V \in \mathbb{R}^d$$

ESTIMATION ERROR

- The estimation error of a function f by a neural networks scales as [Barron 1992].



DEPTH OF THE NETWORK

- DNN allow representing restricted Boltzmann machines with a number of parameters exponentially greater than the number of the network parameters [Montúfar & Morton, 2014]
- Each DNN layer with ReLU divides the space by a hyper-plane.
- Therefore the depth of the network divides the space into an exponential number of sets compared to the number of parameters [Montúfar, Pascanu, Cho & Bengio, 2014]

DEPTH EFFICIENCY OF CNN

- Function realized by CNN, with ReLU and max-pooling, of polynomial size requires super-polynomial size for being approximated by shallow network [Cohen et al., 2016].
- Standard convolutional network design has learning bias towards statistics of natural images [Cohen et al., 2016].

ROLE OF POOLING

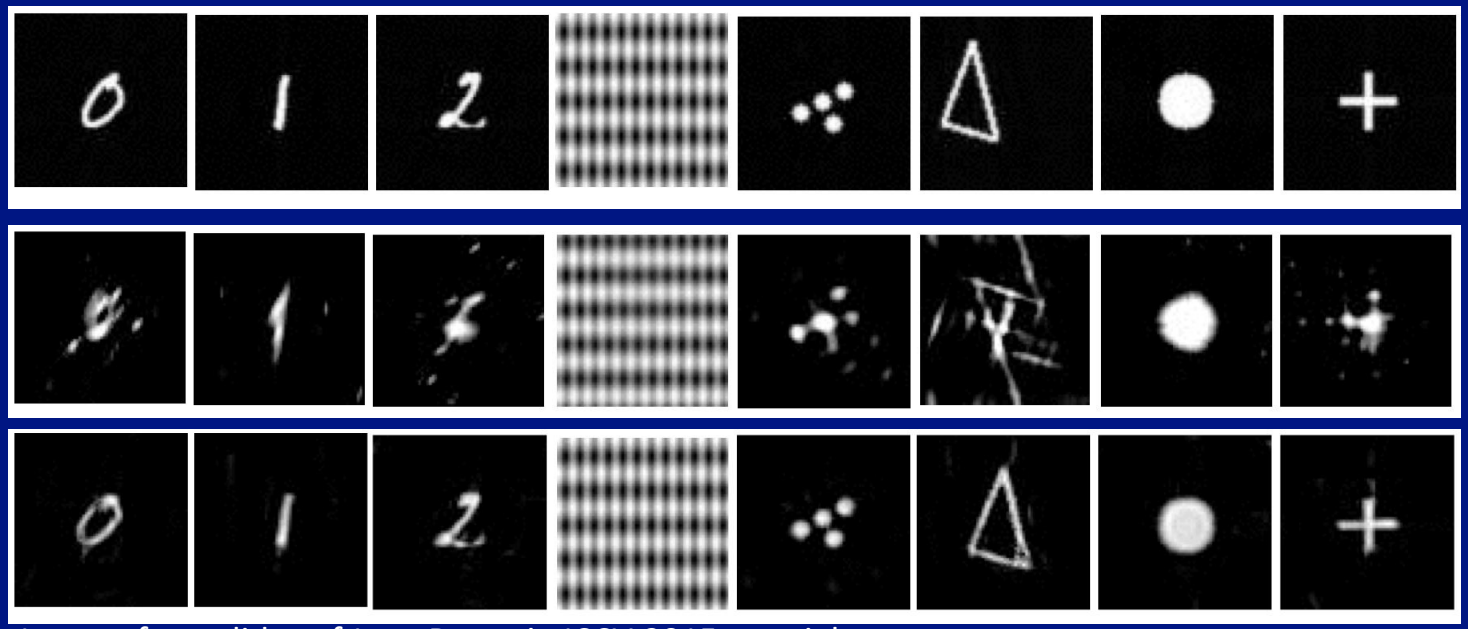
- The pooling stage provides shift invariance [Bruna, LeCun & Szlam, 2013].
- A connection is drawn between the pooling stage and the phase retrieval methods [Bruna, Szlam & LeCun, 2014].
- This allows calculating Lipchitz constants of each DNN layer $\psi(\cdot\, X)$ and empirically recovering the input of a layer from its output.
- However, the Lipchitz constants calculated are very loose and no theoretical guarantees are given for the recovery.

SUFFICIENT STATISTIC AND INVARIANCE

- Given a certain task at hand:
- Minimal sufficient statistic guarantees that we can replace raw data with a representation with smallest complexity and no performance loss.
- Invariance guarantees that the statistic is constant with respect to uninformative transformations of the data.
- CNN are shown to have these properties for many tasks [Soatto & Chiuso, 2016].

SCATTERING TRANSFORMS

- Scattering Transforms - a cascade of wavelet transform convolutions with nonlinear modulus and averaging operators.
- Scattering coefficients are stable encodings of geometry and texture [Bruna & Mallat, 2013]



Original image with d pixels

Recovery from scattering transform with 1 layer

Recovery from scattering transform with 2 layers

Images from slides of Joan Bruna in ICCV 2015 tutorial

SCATTERING TRANSFORMS AND DNN

- More layers create features that can be made invariant to increasingly more complex deformations.
- Layers in a DNN encode complex, class-specific geometry.
- Deeper architectures are able to better capture invariant properties of objects and scenes in images
  [Bruna & Mallat, 2013]

SCATTERING TRANSFORMS AS A METRIC

- Scattering transforms may be used as a metric.
- Inverse problems can be solved by minimizing distance at the scattering transform domain.
- Leads to remarkable results in super-resolution [Bruna, Sprechmann & Lecun, 2016]

SCATTERING SUPER RESOLUTION

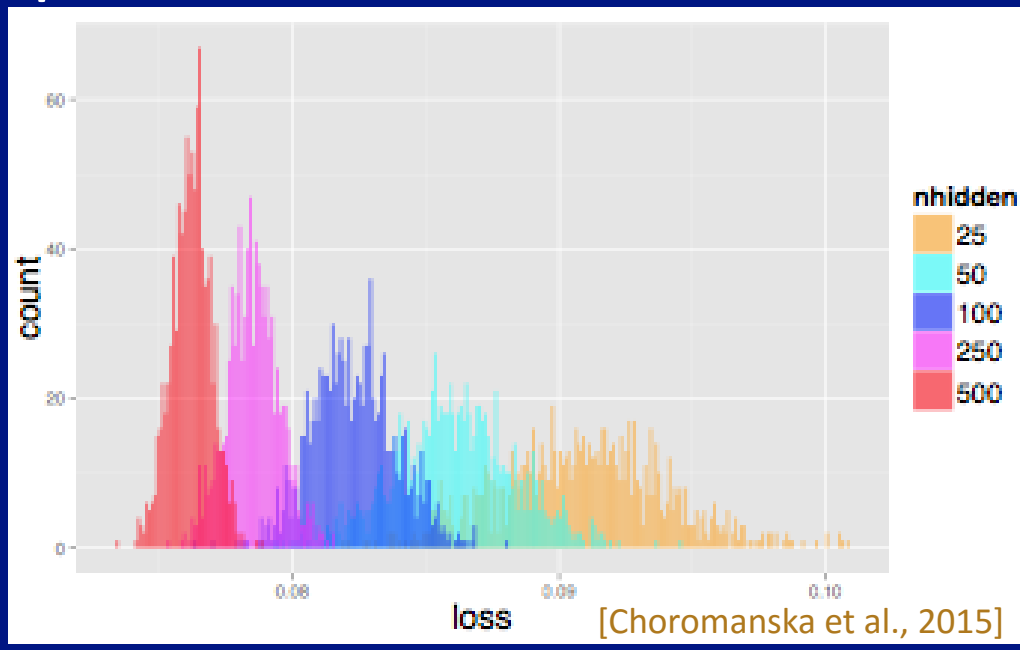
Original | Best Linear Estimate | State-of-the-art | Scattering estimate

[Bruna, Sprechmann & Lecun, 2016]

Images from slides of Joan Bruna in CVPR 2016 tutorial

MINIMIZATION

- The local minima in deep networks are not far from the global minimum.
- saddle points are the main problem of deep Learning optimization.
- Deeper networks have more local minima but less saddle points.

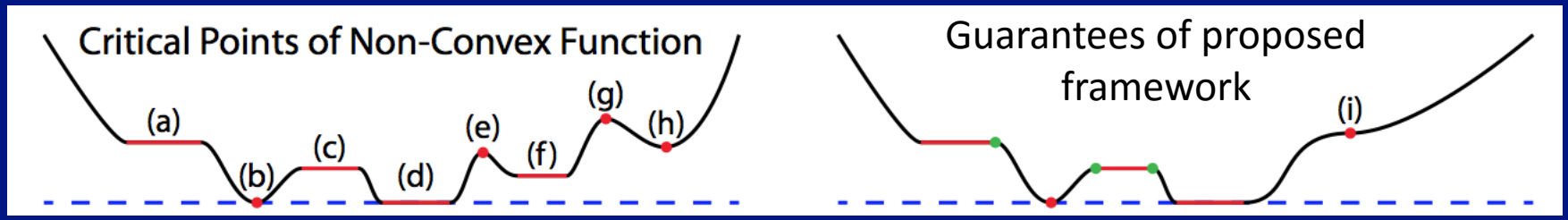


[Choromanska et al., 2015]

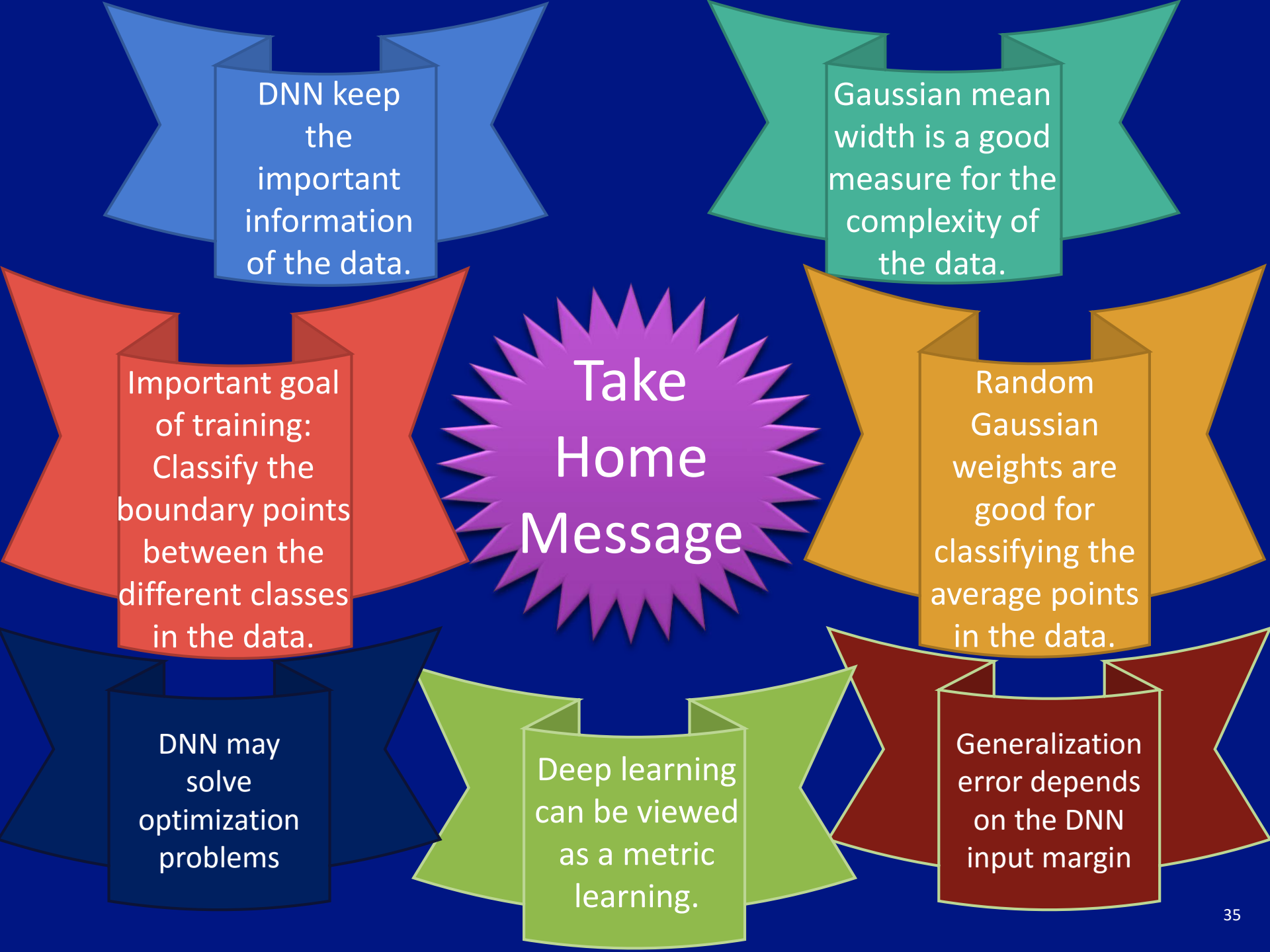
[Saxe, McClelland & Ganguli, 2014], [Dauphin, Pascanu, Gulcehre, Cho, Ganguli & Bengio, 2014] [Choromanska, Henaff, Mathieu, Ben Arous & LeCun, 2015]

GLOBAL OPTIMALITY IN DEEP LEARNING

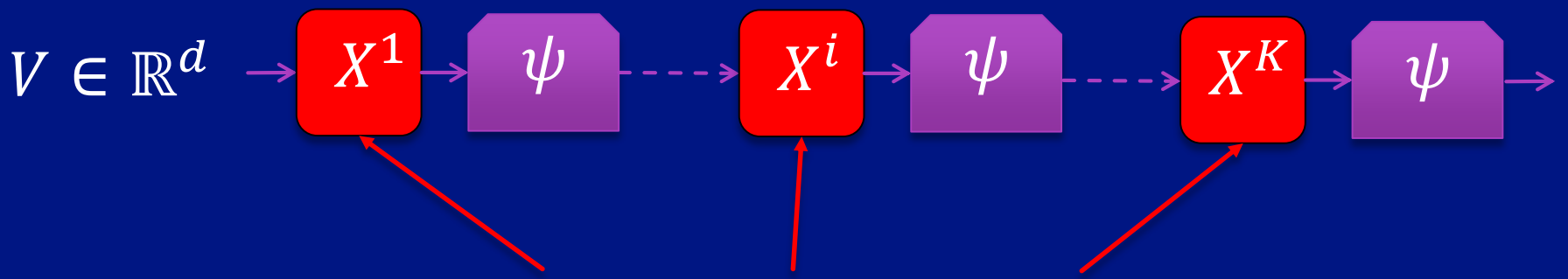
- Deep learning is a positively homogeneous factorization problem, i.e., $\exists p \geq 0$ such that $\forall \alpha \geq 0$ DNN obey $\Phi\left(\alpha X^1, \alpha X^2, \ldots, \alpha X^K\right) = \alpha^p \Phi\left(X^1, X^2, \ldots, X^K\right)$.
- With proper regularization, local minima are global.
- If the network is large enough, global minima can be found by local descent.



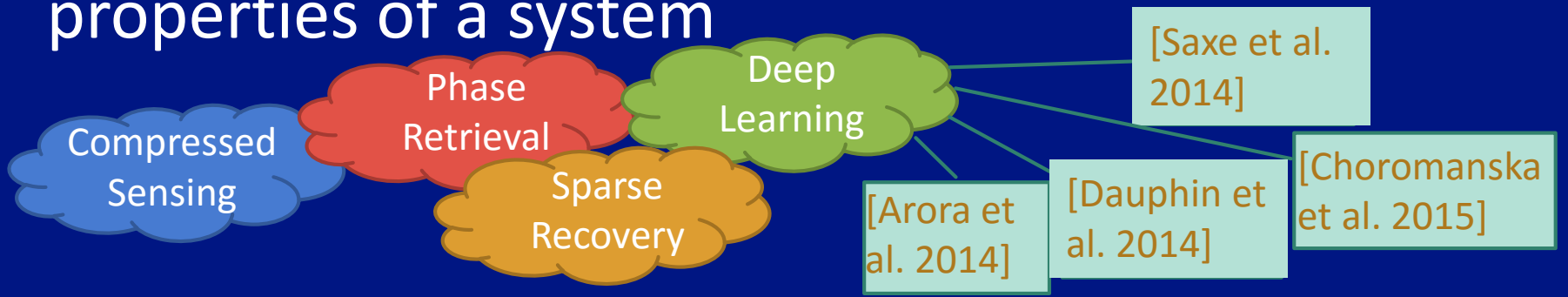
[Haeffele & Vidal, 2015]


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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

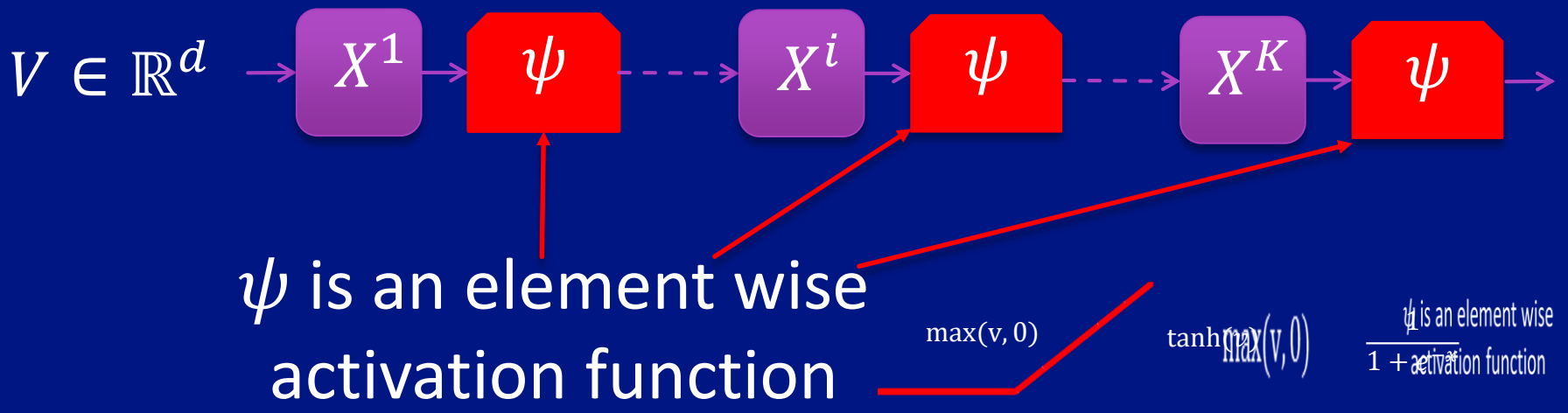
ASSUMPTIONS – GAUSSIAN WEIGHTS



- Infusion of random weights reveals internal properties of a system



ASSUMPTIONS – NO POOLING

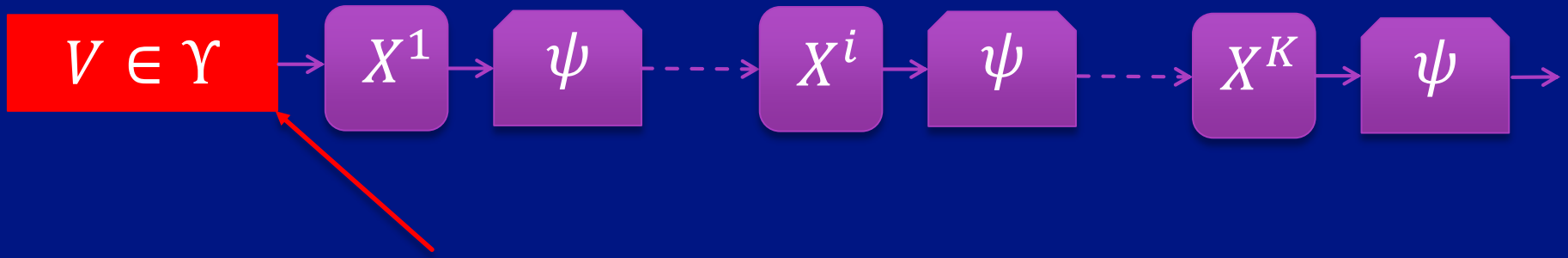


- Pooling provides invariance [Boureau et. al. 2010, Bruna et. al. 2013].

➢ 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



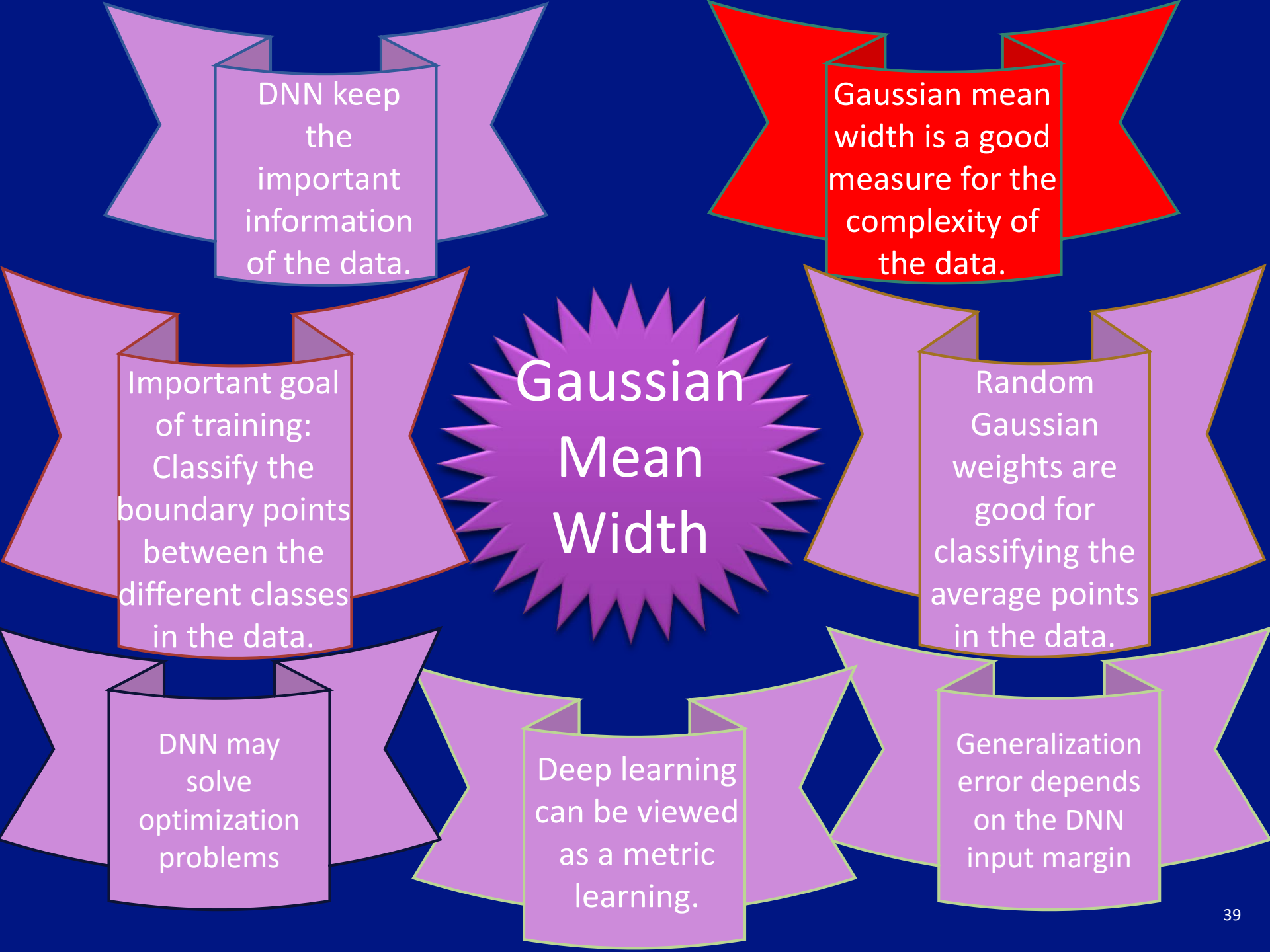
Υ is a low dimensional set

Gaussian Mixture Models (GMM)

Low Rank Matrices

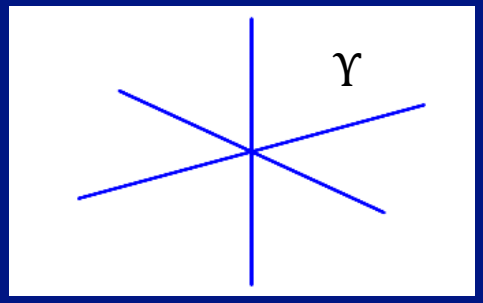
Signals with Sparse Representations

Low Dimensional Manifolds

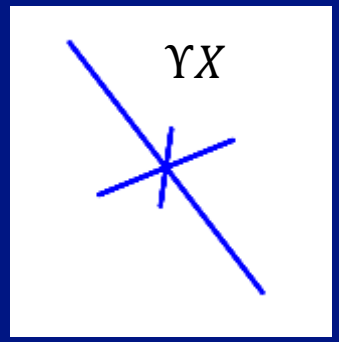

DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Important goal of training: Classify the boundary points between the different classes in the data.
Gaussian Mean Width
Random Gaussian weights are good for classifying the average points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

WHAT HAPPENS TO SPARSE DATA IN DNN?

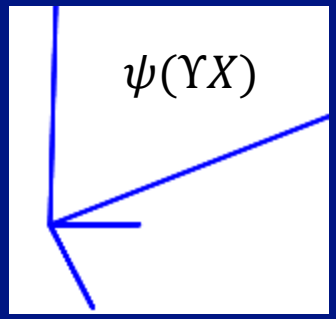
- Let Υ be sparsely represented data
  - Example: $\Upsilon = \{V \in \mathbb{R}^3 : \|V\|_0 \leq 1\}$



- 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

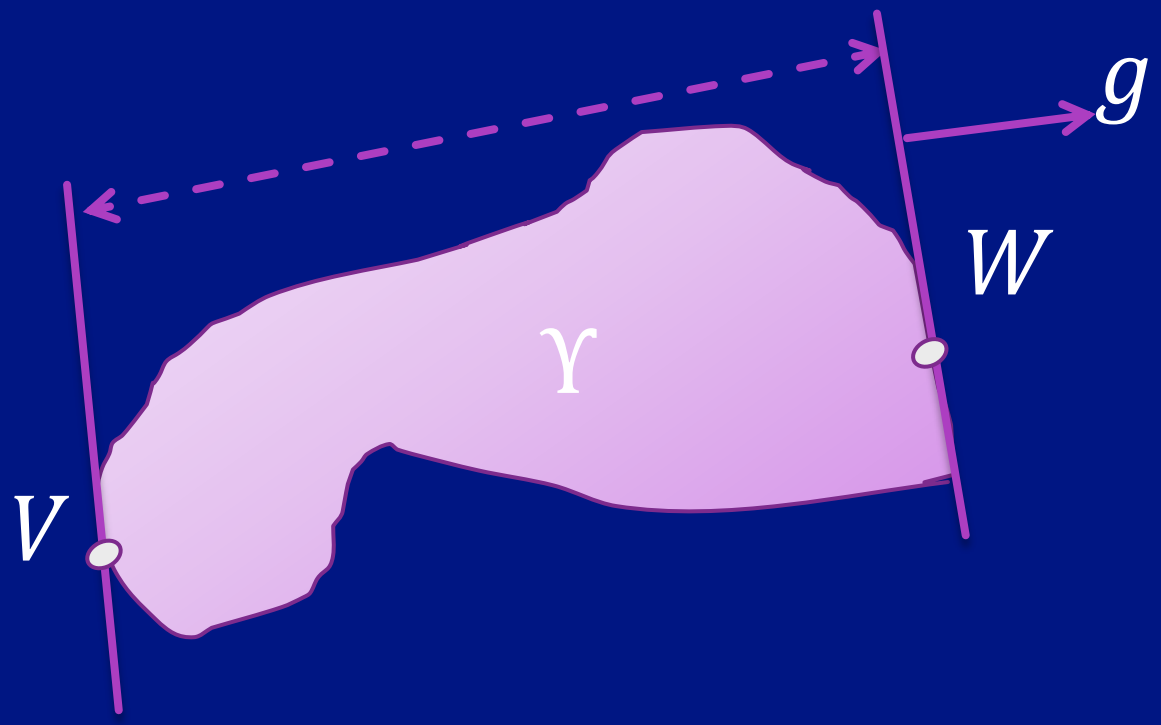


GAUSSIAN MEAN WIDTH

- 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:

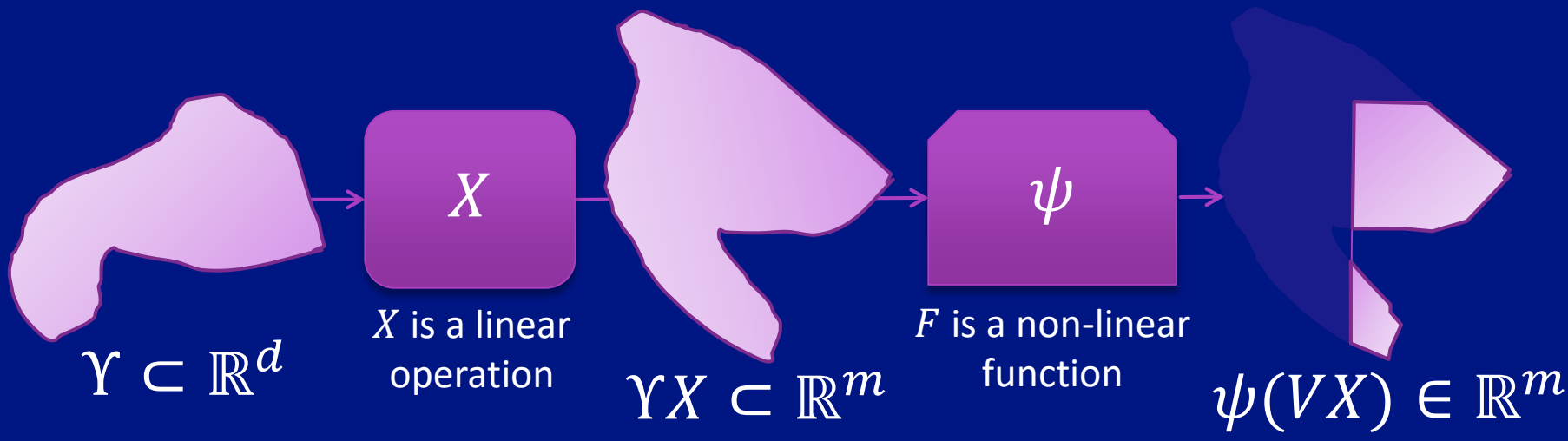
If $\Upsilon \subset \mathbb{B}^d$ is a Gaussian Mixture Model with k Gaussians then

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

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

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

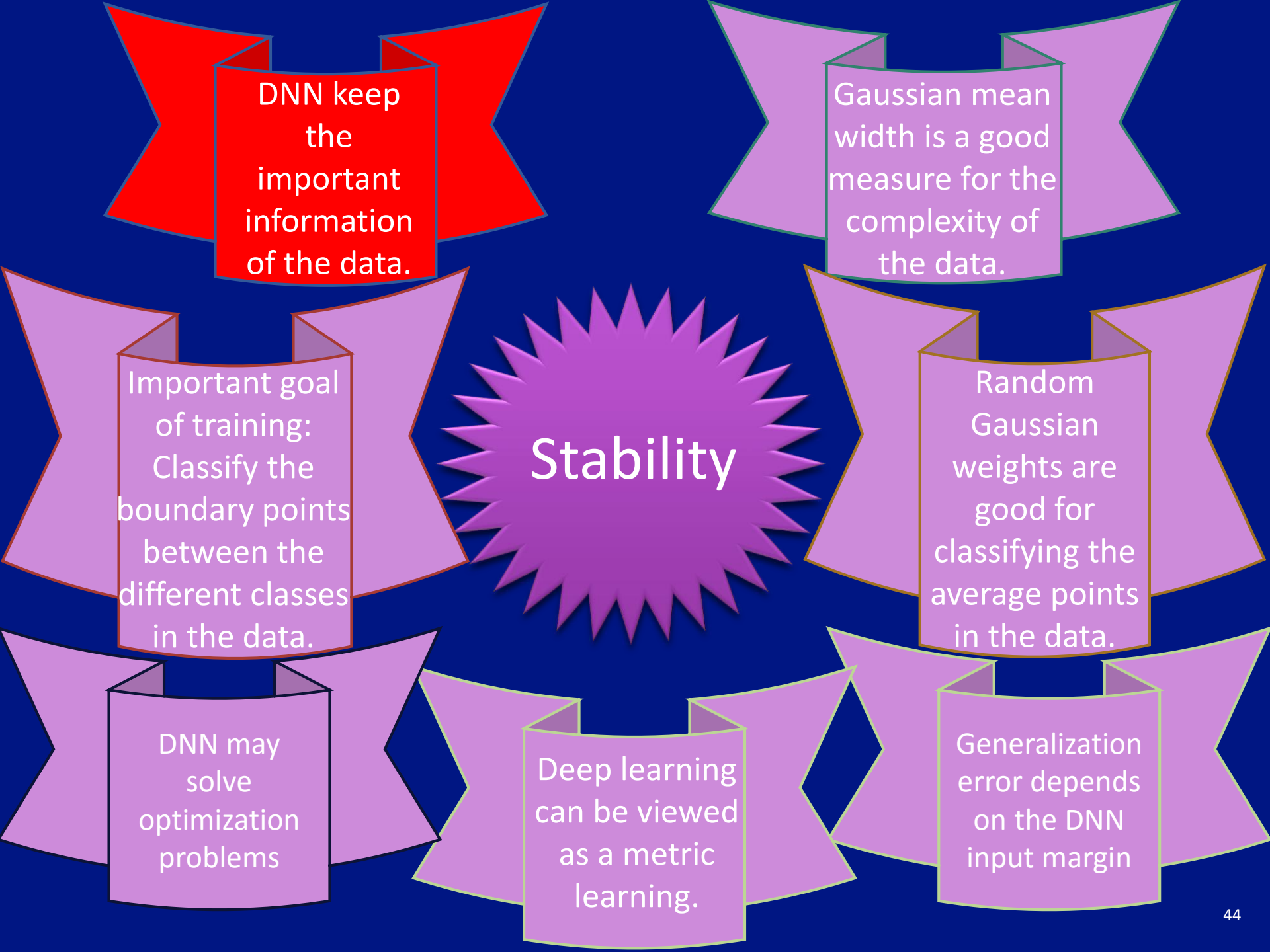
GAUSSIAN MEAN WIDTH IN DNN



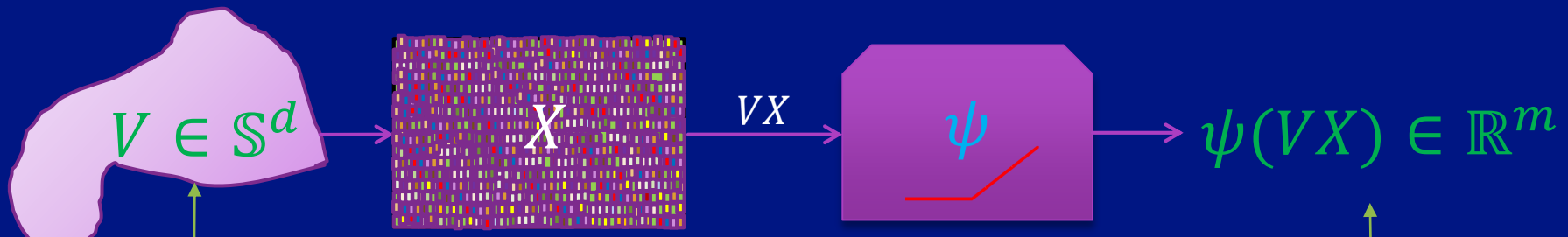
Theorem 1: small $\frac{\omega^2(\Upsilon)}{m}$ imply $\omega^2(\Upsilon) \approx \omega^2(\psi(VX))$

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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

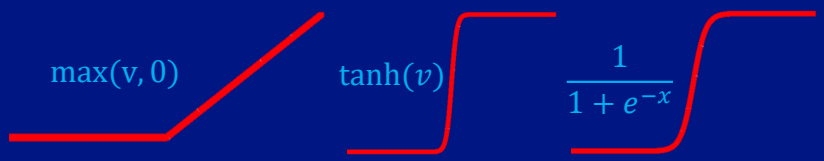
ASSUMPTIONS



$V \in \Upsilon$

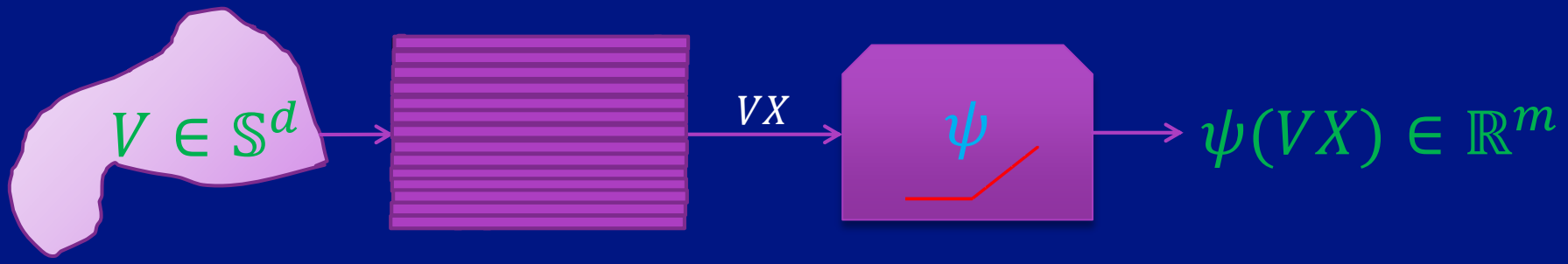
X is a random Gaussian matrix

ψ is an element wise activation function

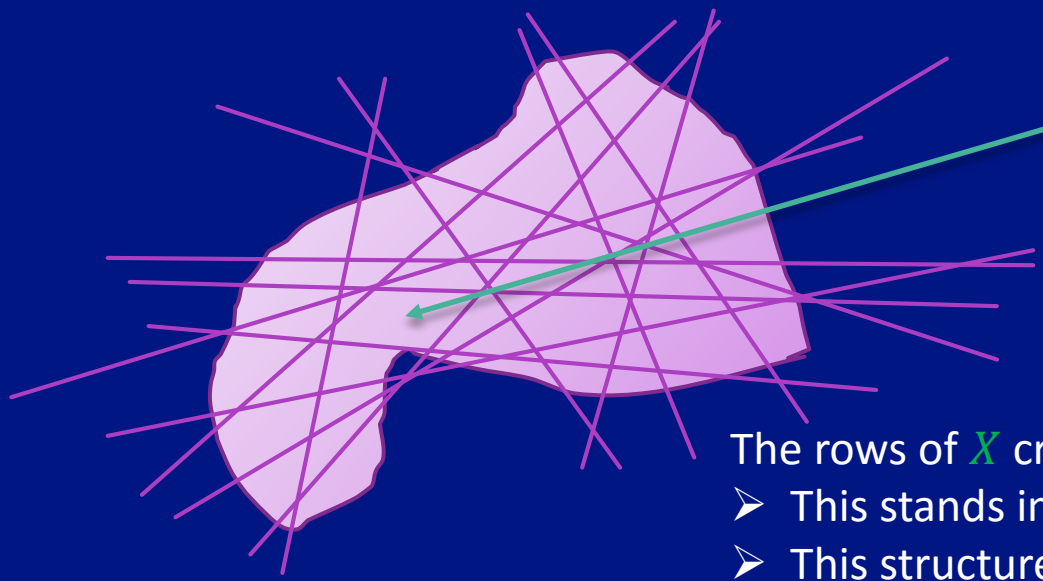


$$m = O\left(\delta^{-6}\omega^2(\Upsilon)\right)$$

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 2016].



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

➡ Each layer of the network keeps the main information of the data

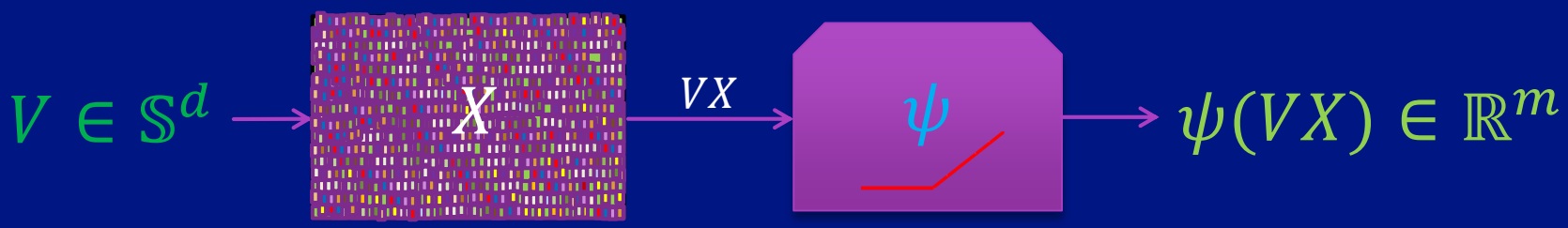
The rows of X create a tessellation of the space.

- ➢ This stands in line with [Montúfar et. al. 2014]
- ➢ This structure can be used for hashing

DNN AND HASHING

- A single layer performs a locally sensitive hashing.
- Deep network with random weights may be designed to do better [Choromanska et al., 2016].
- It is possible to train DNN for hashing, which provides cutting-edge results [Masci et al., 2012], [Lai et al., 2015].

DNN STABLE EMBEDDING



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

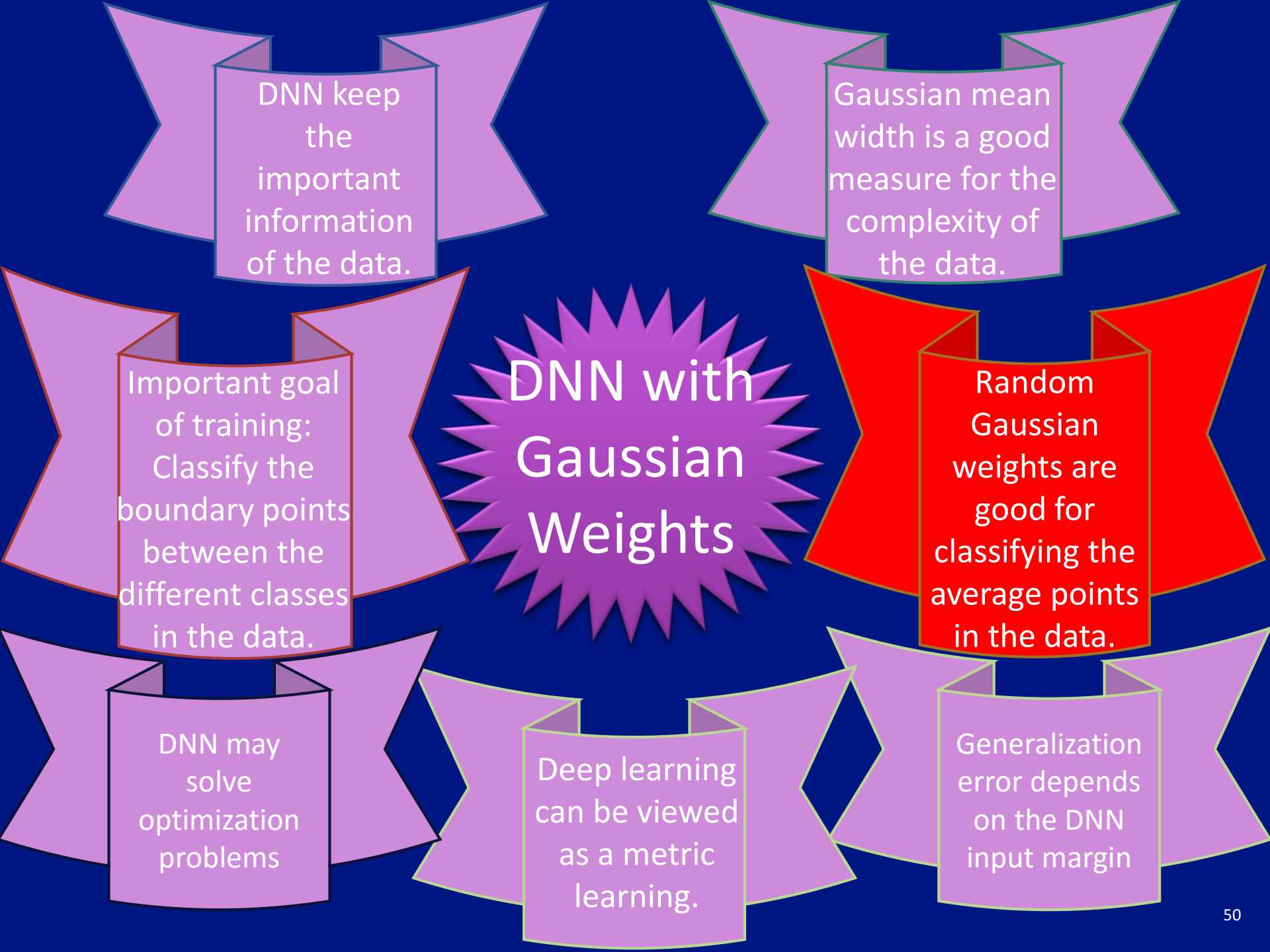
$$\|V - \mathcal{A}(\psi(VX))\| < O\left(\frac{\omega(\Upsilon)}{\sqrt{m}}\right) = O(\delta^3)$$

[Plan & Vershynin, 2013, Giryes, Sapiro & Bronstein 2016].

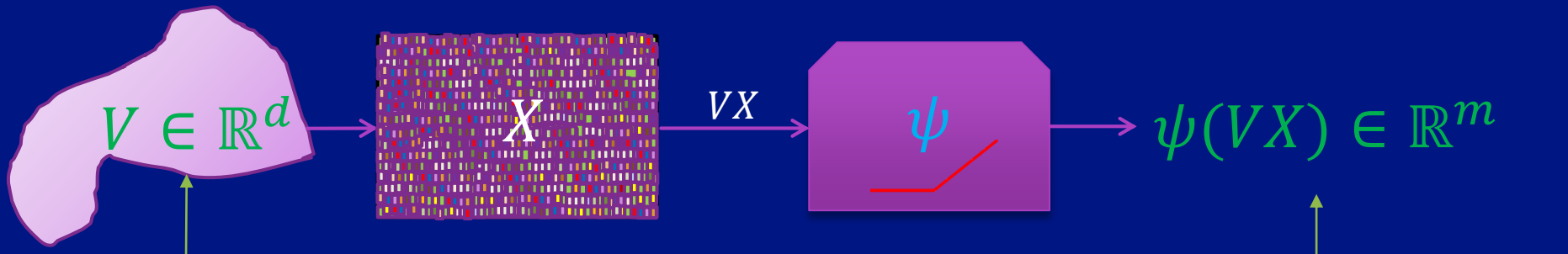
- After K layers we have an error $O(K\delta^3)$
- Stands in line with [Mahendran and Vedaldi, 2015].
- DNN keep the important information of the data

RECOVERY FROM DNN OUTPUT

[Mahendran and Vedaldi, 2015].


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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

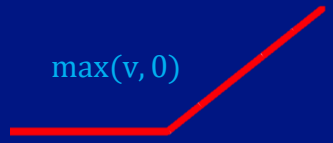
ASSUMPTIONS



$V \in \Upsilon$

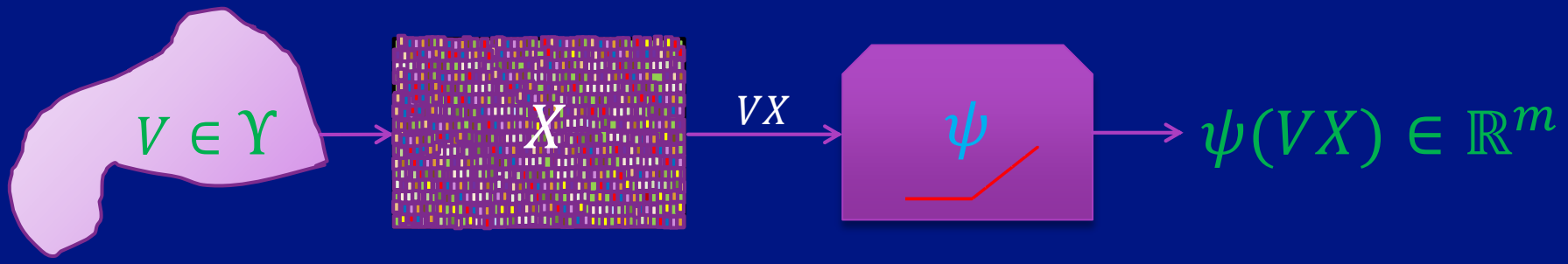
X is a random Gaussian matrix

ψ is the ReLU



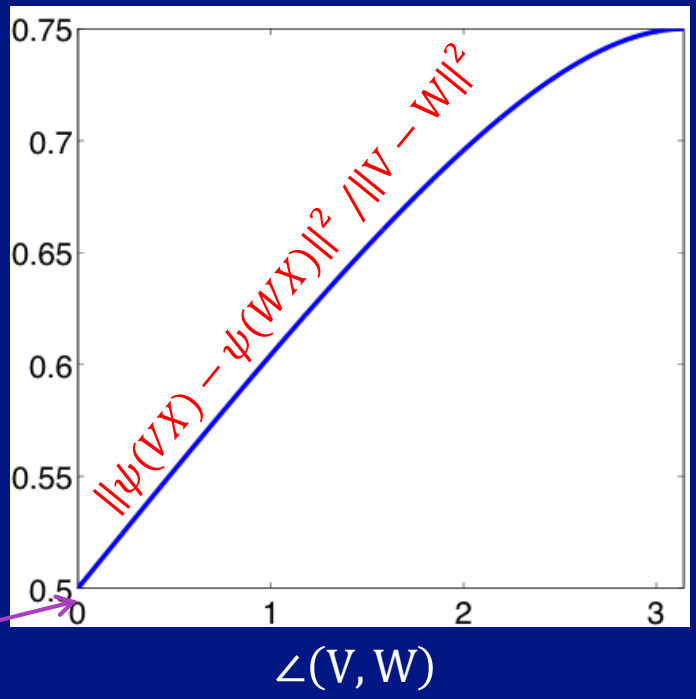
$$m = O\left(\delta^{-4}\omega^4(\Upsilon)\right)$$

DISTANCE DISTORTION



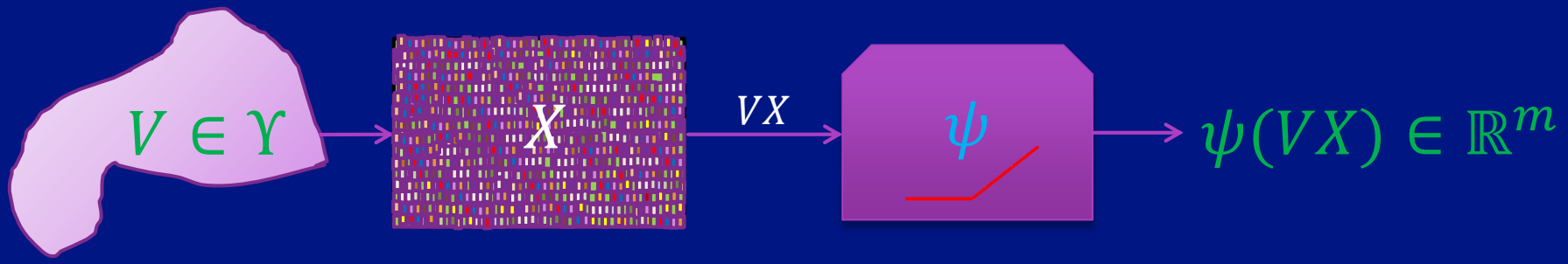
Theorem 4: for $V, W \in \Upsilon$

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



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

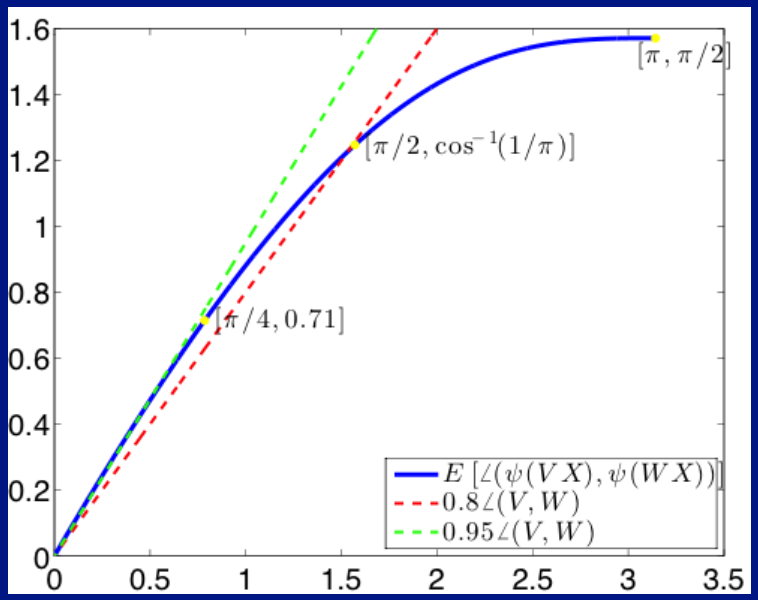
ANGLE DISTORTION



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

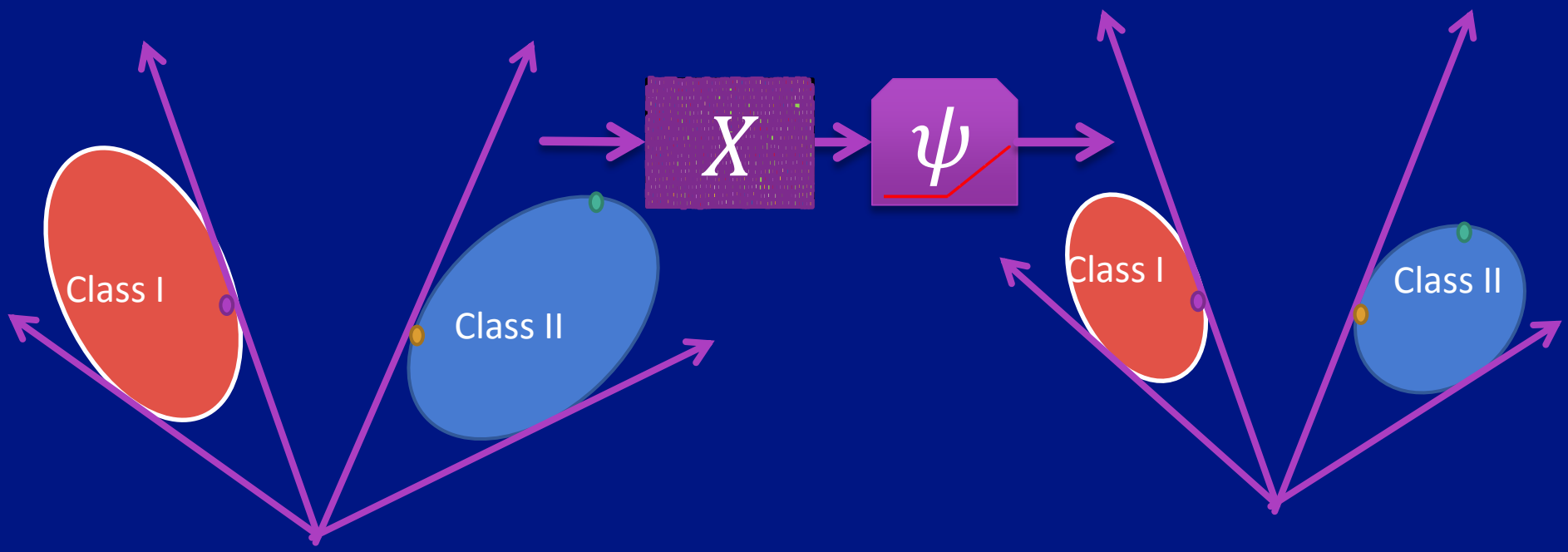
$$\left|\cos\angle(\psi(VX),\psi(WX)) - \cos\angle(V,W)\right| \leq \frac{1}{\pi}\left(\sin\angle(V,W)\right)$$

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

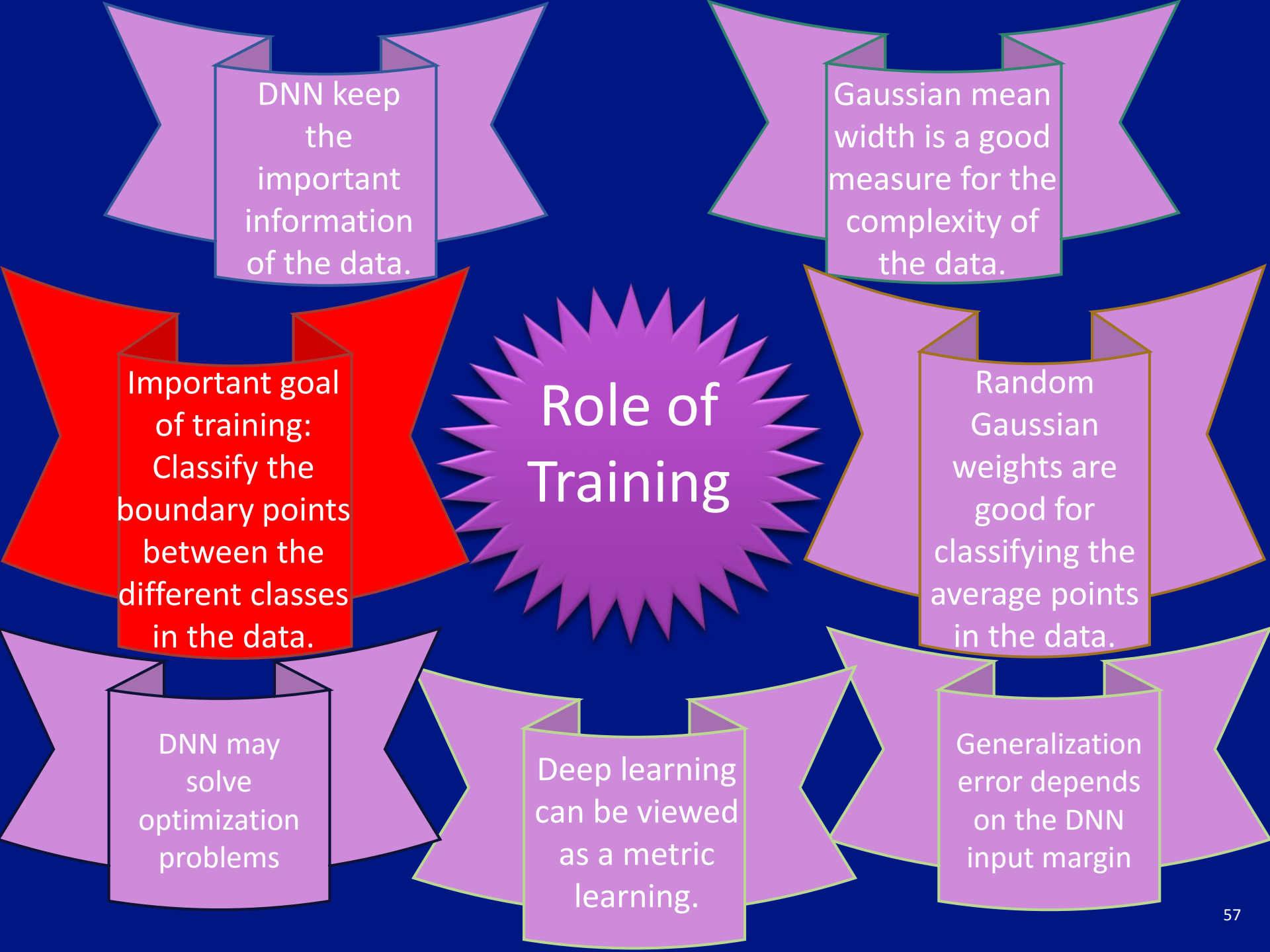
- Stability in the network implies that close points in the input are close also at the output

➡ 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

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

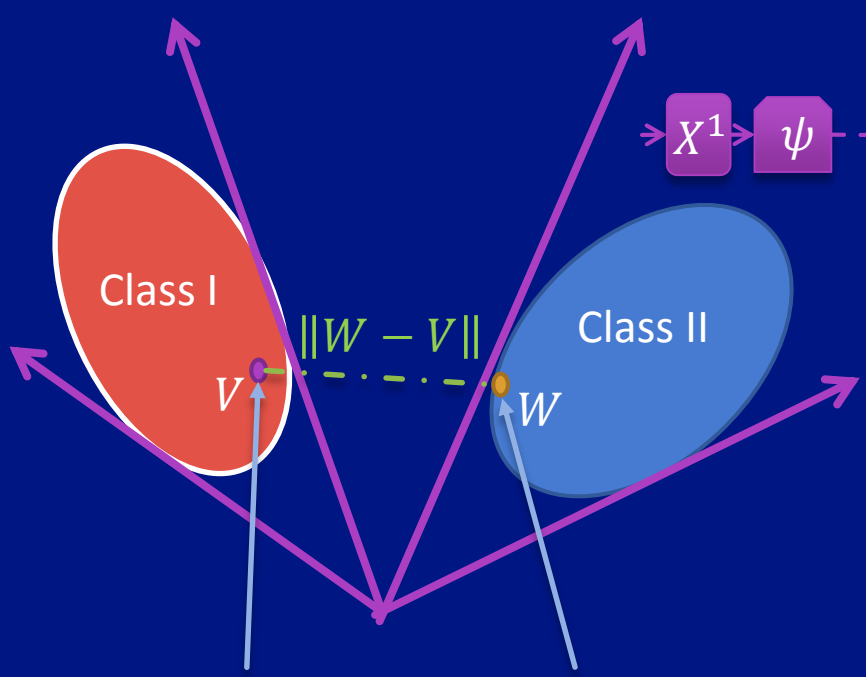
➡ Though this is not a tight bound, it introduces the Gaussian mean width $\omega\,(\Upsilon)$ as a measure for the complexity of the input data and the required number of training samples.


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.
Role of Training
Random Gaussian weights are good for classifying the average points in the data.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

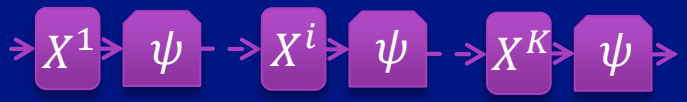
ROLE OF TRAINING

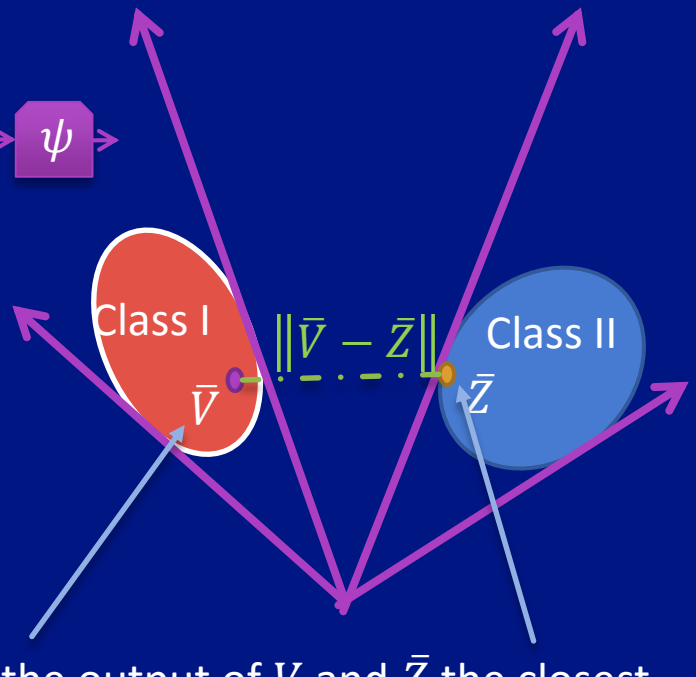
- Having a theory for Gaussian weights we test the behavior of DNN after training.
- We looked at the MNIST, CIFAR-10 and ImageNet datasets.
- We will present here only the ImageNet results.
- We use a state-of-the-art pre-trained network for ImageNet [Simonyan & Zisserman, 2014].
- We compute inter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

INTER BOUNDARY POINTS DISTANCE RATIO



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

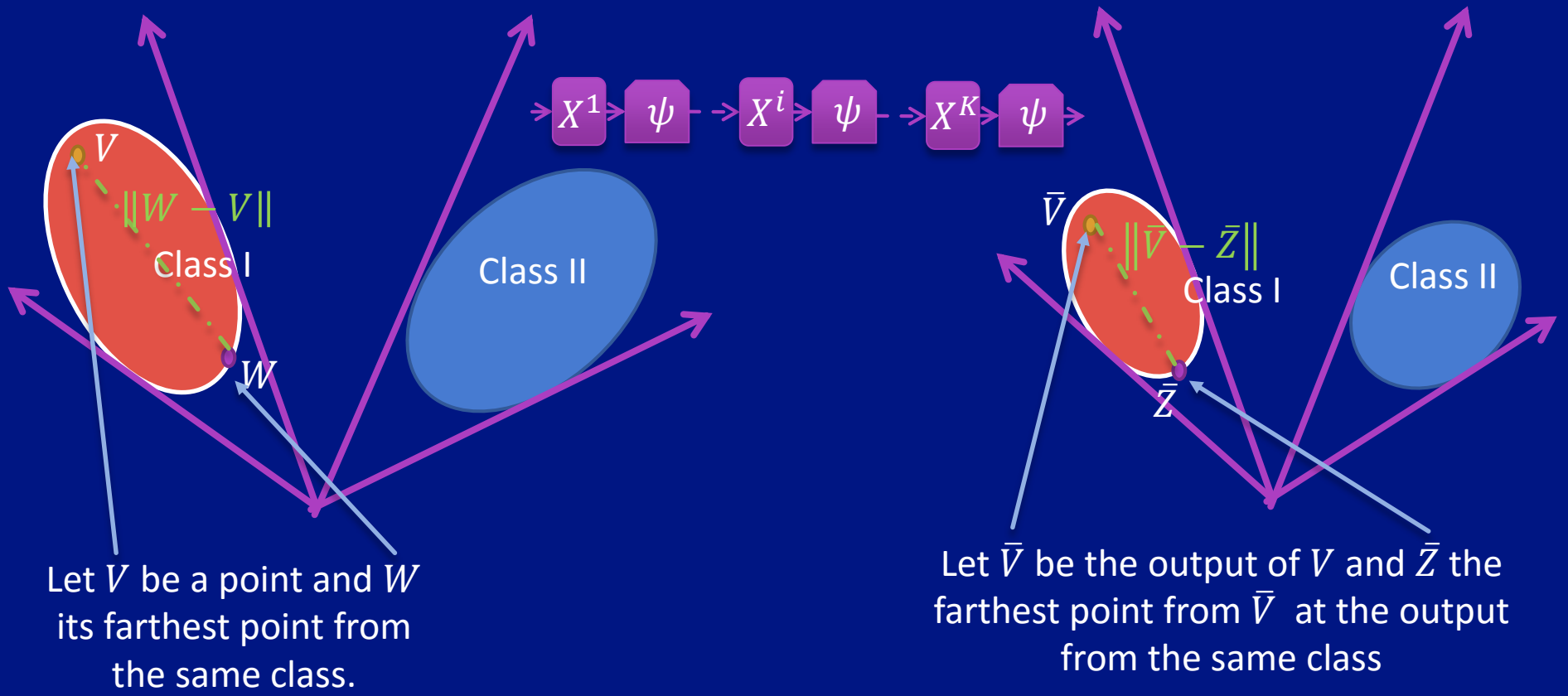




$\bar{V}$ is the output of V and $\bar{Z}$ the closest point to $\bar{V}$ at the output from a different class.

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

INTRA BOUNDARY POINTS DISTANCE RATIO



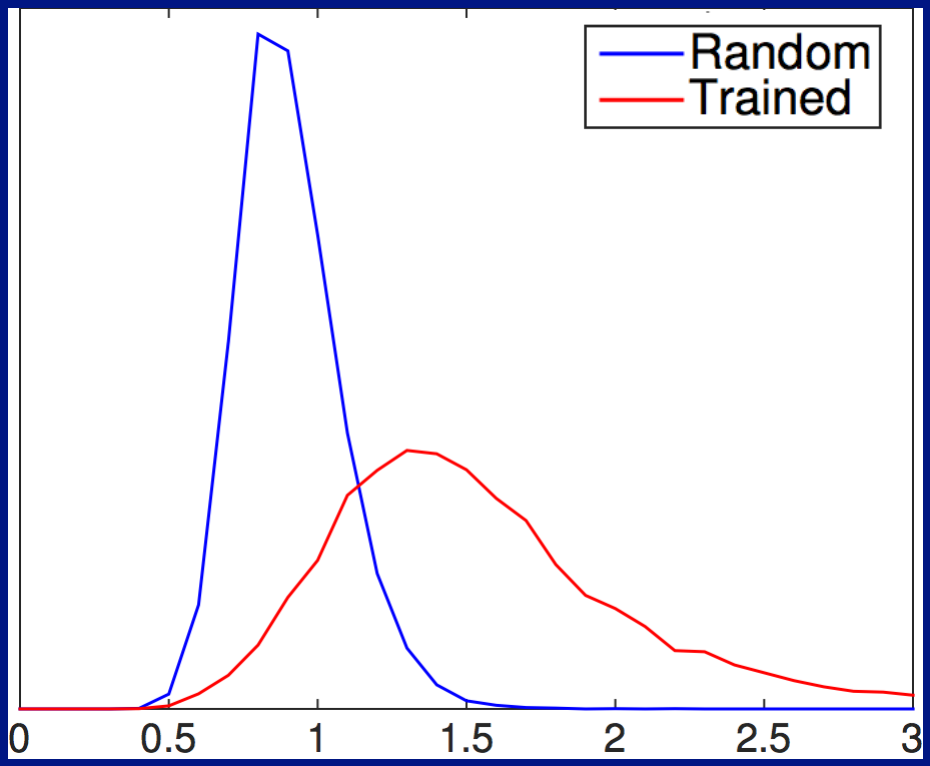
Let V be a point and W its farthest point from the same class.

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

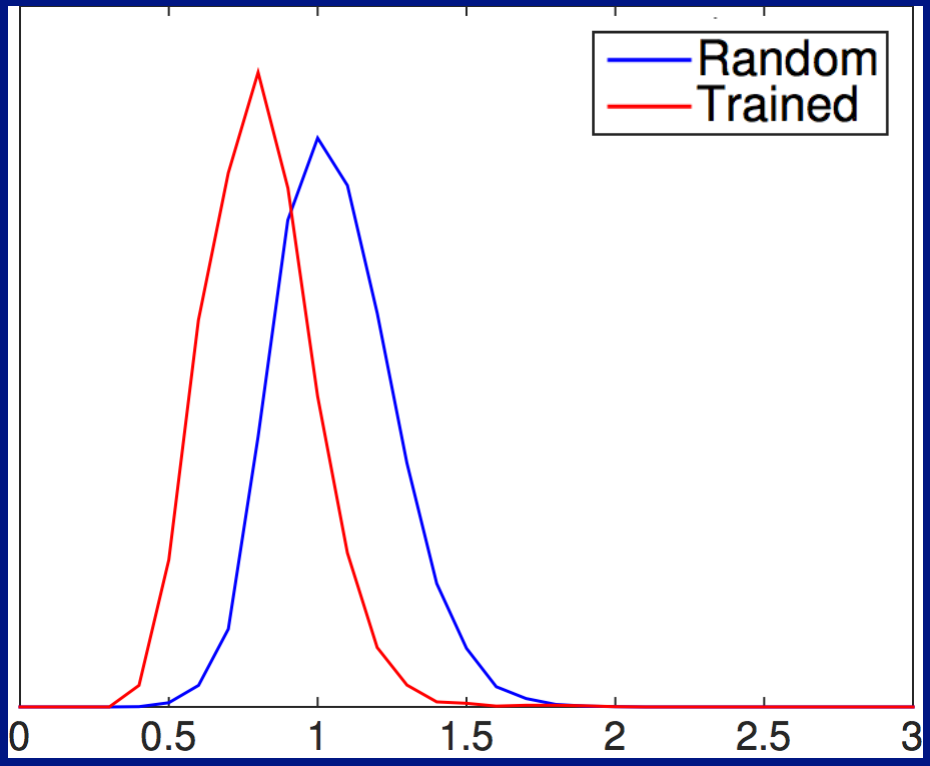
BOUNDARY DISTANCE RATIO

Inter-class



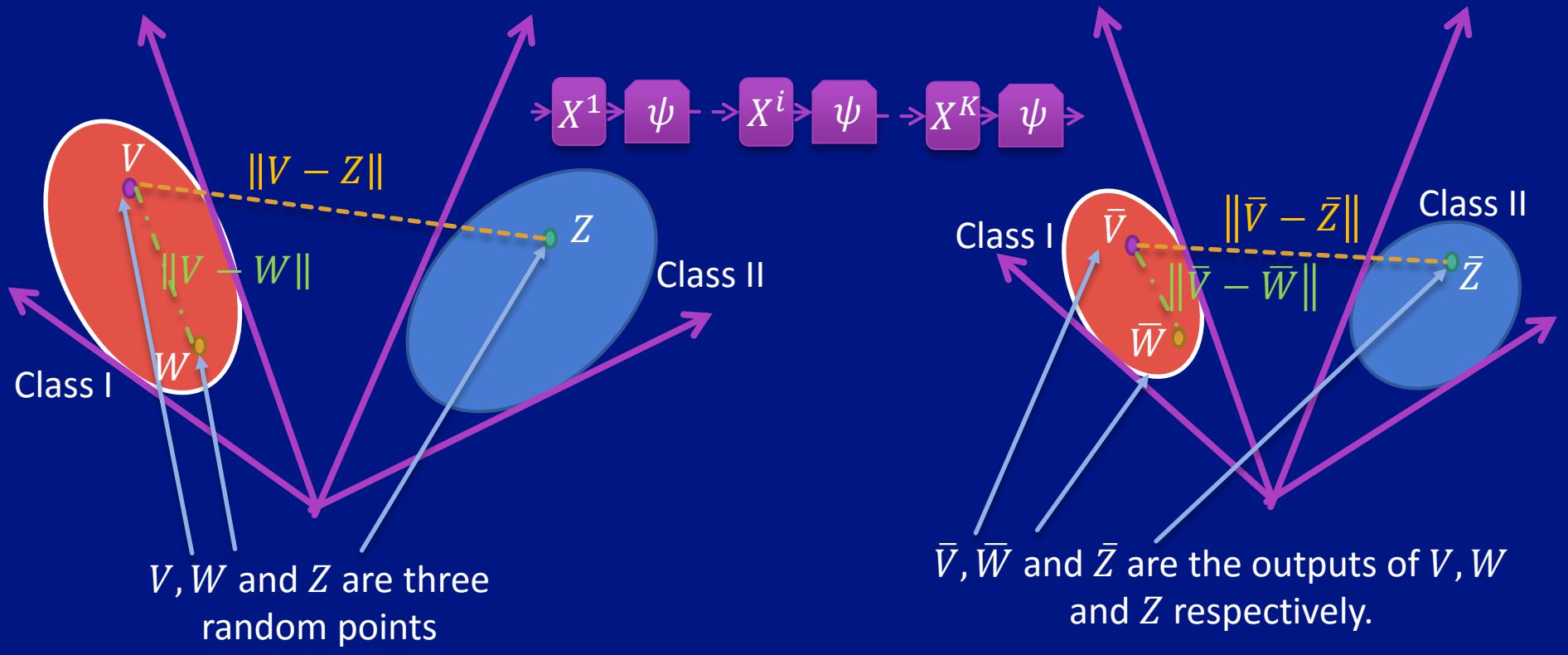
$$\frac{\|\bar{V}-\bar{Z}\|}{\|W-V\|}$$

Intra-class



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

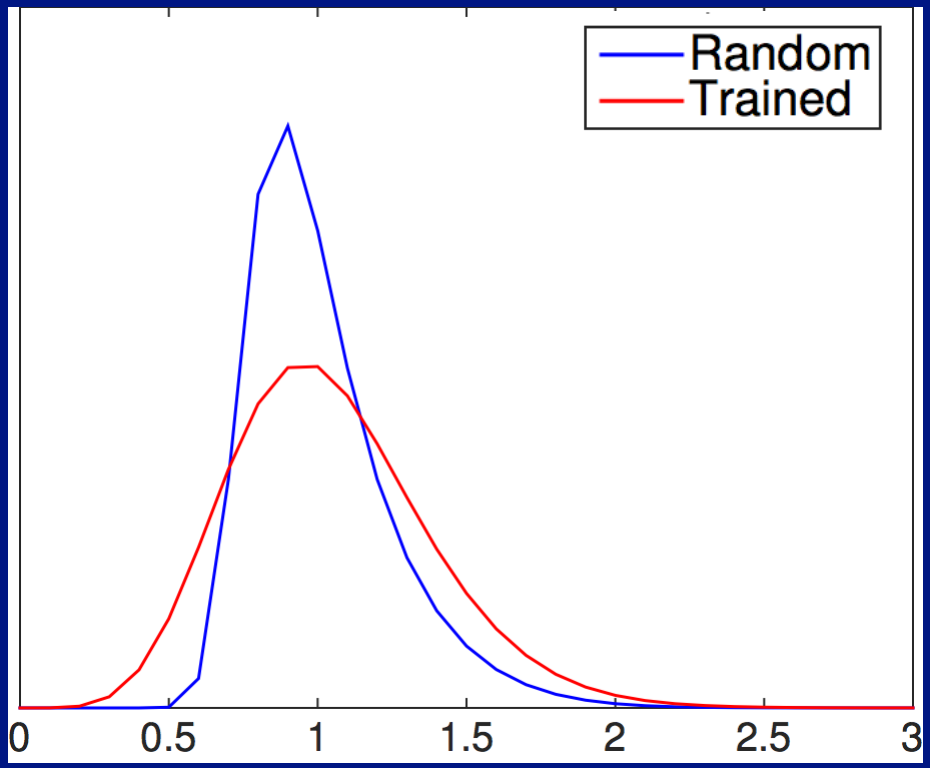
AVERAGE POINTS DISTANCE RATIO



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

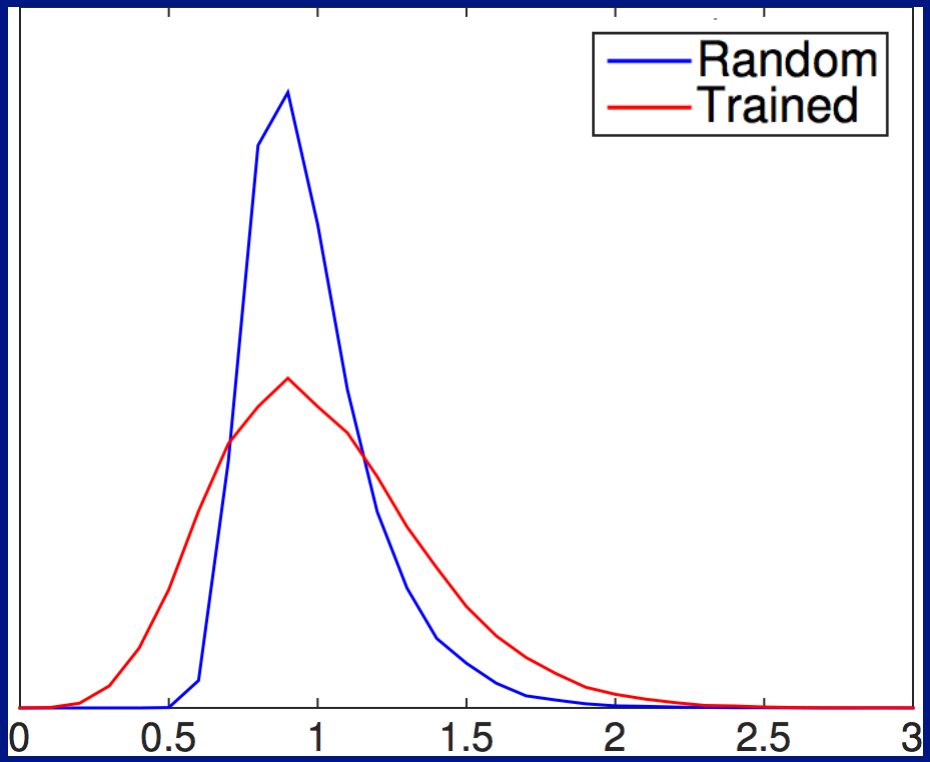
AVERAGE DISTANCE RATIO

Inter-class



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

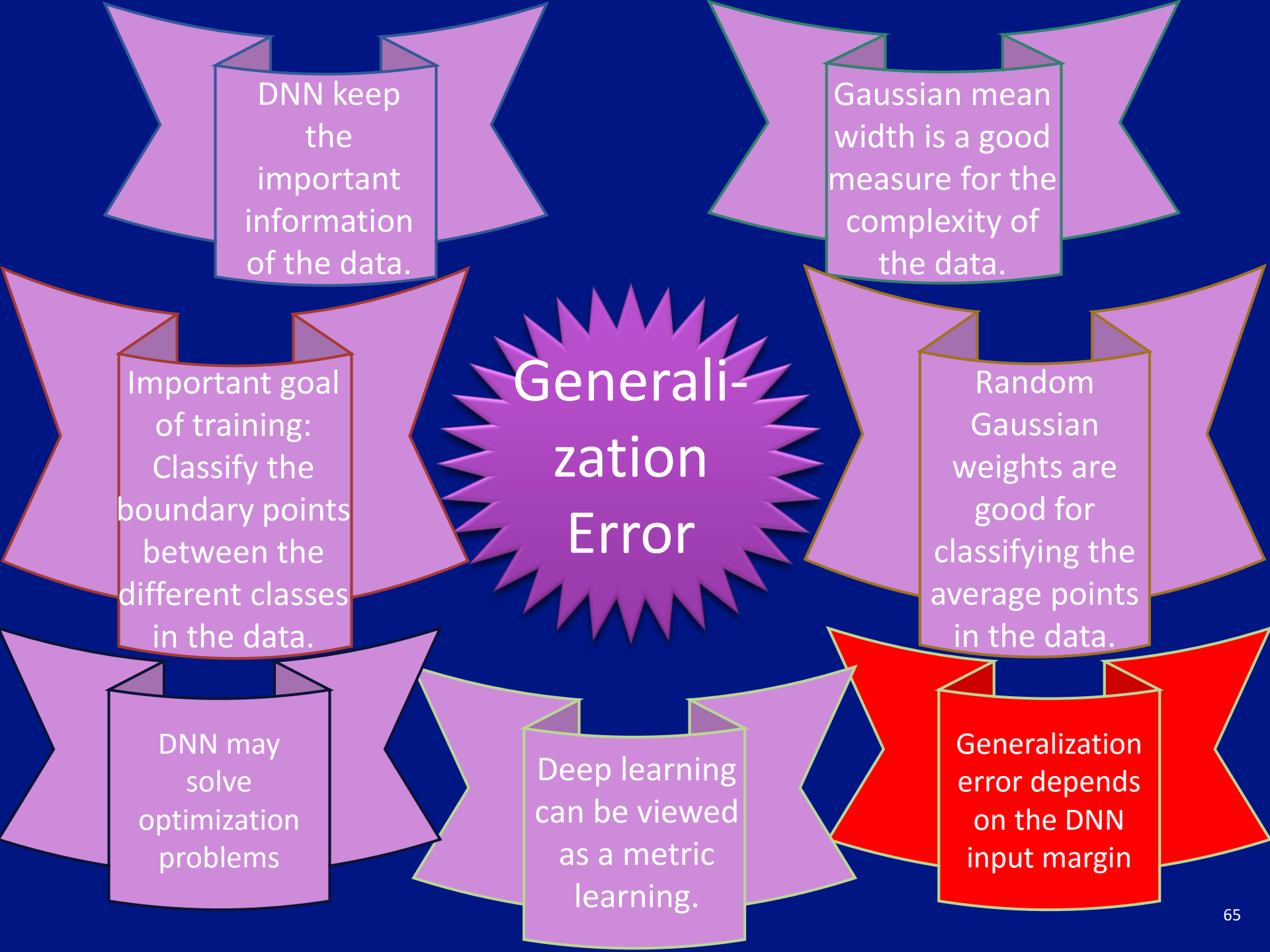
Intra-class



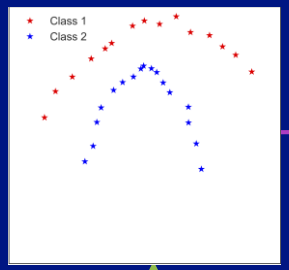
$$\frac{\|\bar{V}-\bar{W}\|}{\|V-W\|}$$

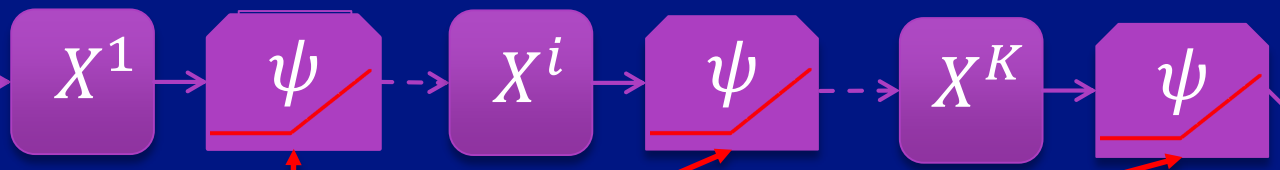
ROLE OF TRAINING

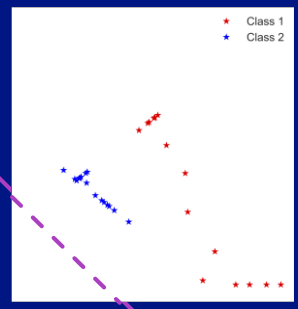
- On average distances are preserved in the trained and random networks.
- The difference is with respect to the boundary points.
- The inter distances become larger.
- The intra distances shrink.


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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

ASSUMPTIONS



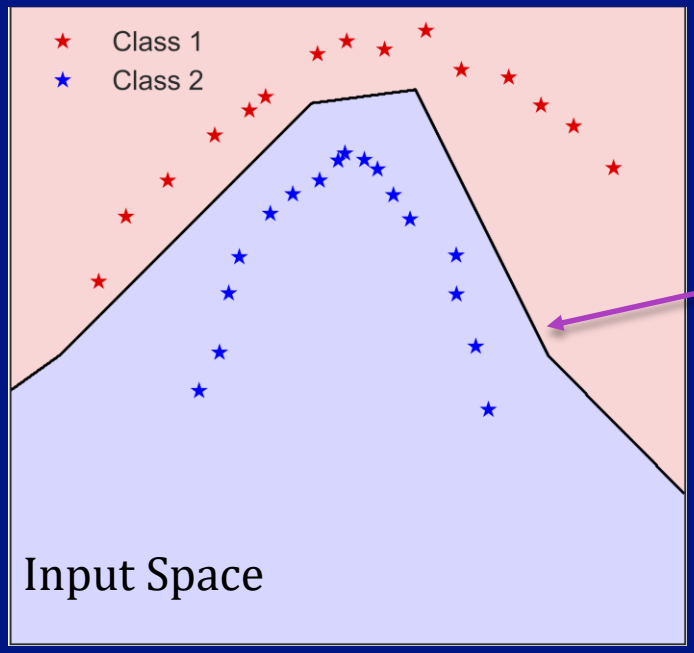




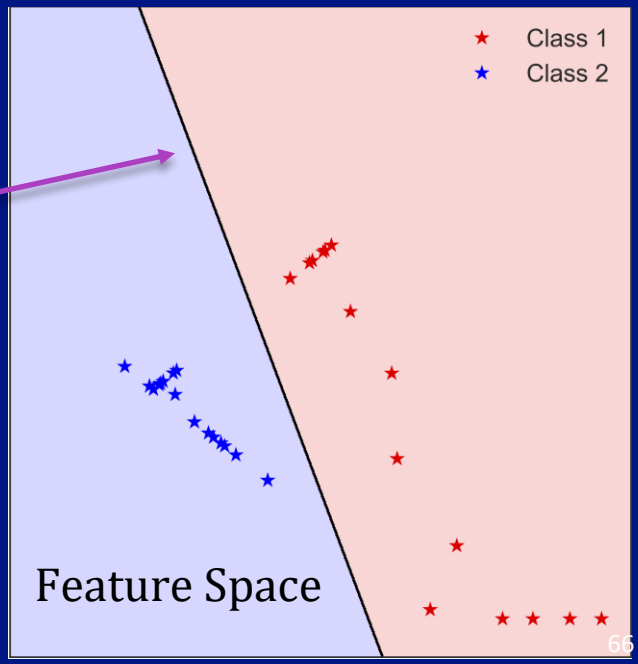
Two Classes $\in \Upsilon$

ψ is the ReLU

Linear classifier w



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



GENERALIZATION ERROR (GE)

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

REGULARIZATION TECHNIQUES

- Weight decay – penalizing DNN weights [Krogh & Hertz, 1992].
- Dropout - randomly drop units (along with their connections) from the neural network during training [Hinton et al., 2012, Srivastava et al., 2014].
- DropConnect – dropout extension [Wan et al., 2013]
- Batch normalization [Ioffe & Szegedy, 2015].
- Stochastic gradient descent (SGD) [Hardt, Recht & Singer, 2016].
- Path-SGD [Neyshabur et al., 2015].

A SAMPLE OF 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].

A SAMPLE OF 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].

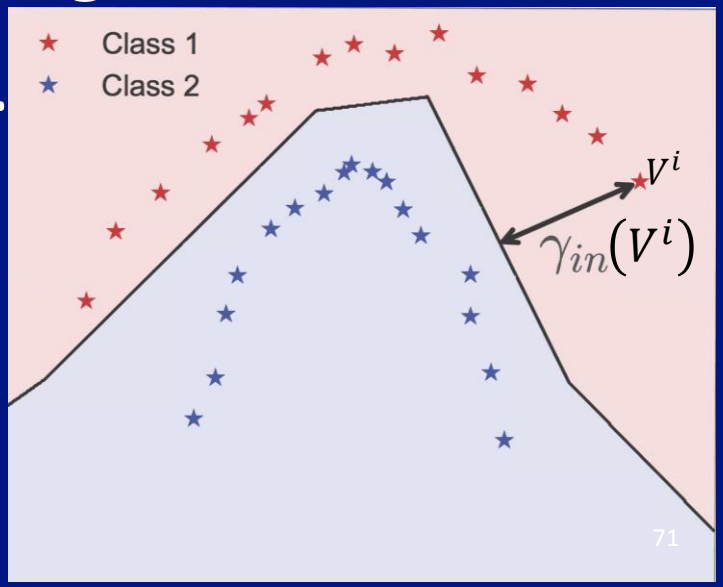
- Note that in both cases the GE grows with the depth

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 larger.
- Bound is independent of depth.
- Our theory relies on the robustness framework [Xu & Mannor, 2015].



INPUT MARGIN BOUND

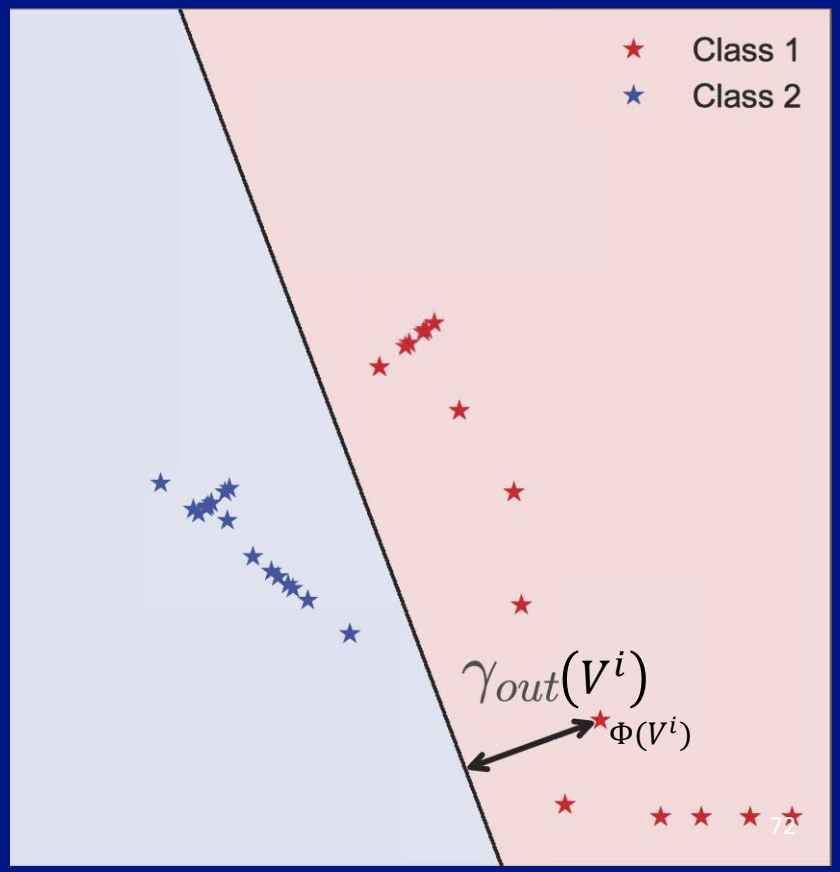
- Maximizing the input margin directly is hard
- Our strategy: relate the input margin to the output margin $\gamma_{out}(V^i)$ and other DNN properties
- Theorem 7:

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

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

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

[Sokolic, Giryes, Sapiro, Rodrigues, 2016]

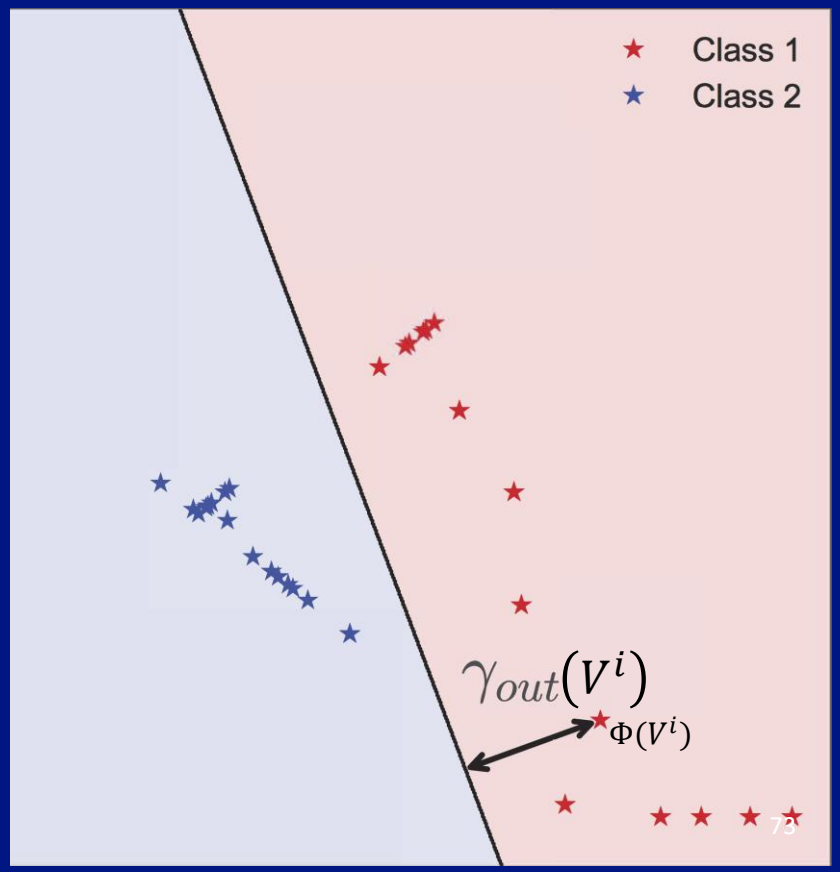


OUTPUT MARGIN

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

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

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

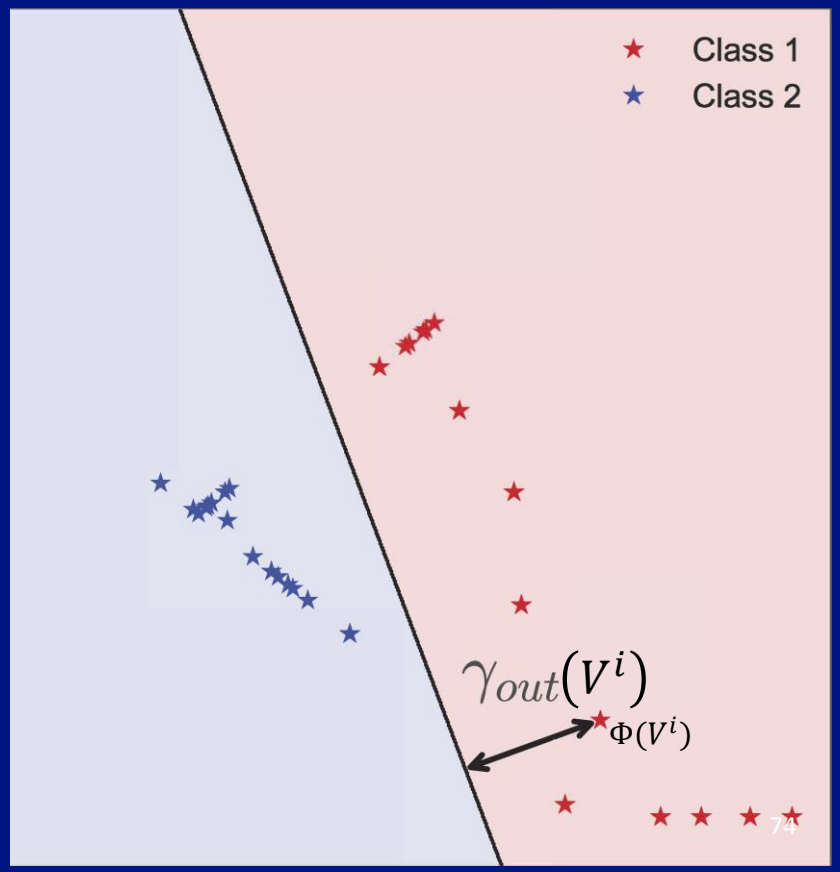


GE AND WEIGHT DECAY

- 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]

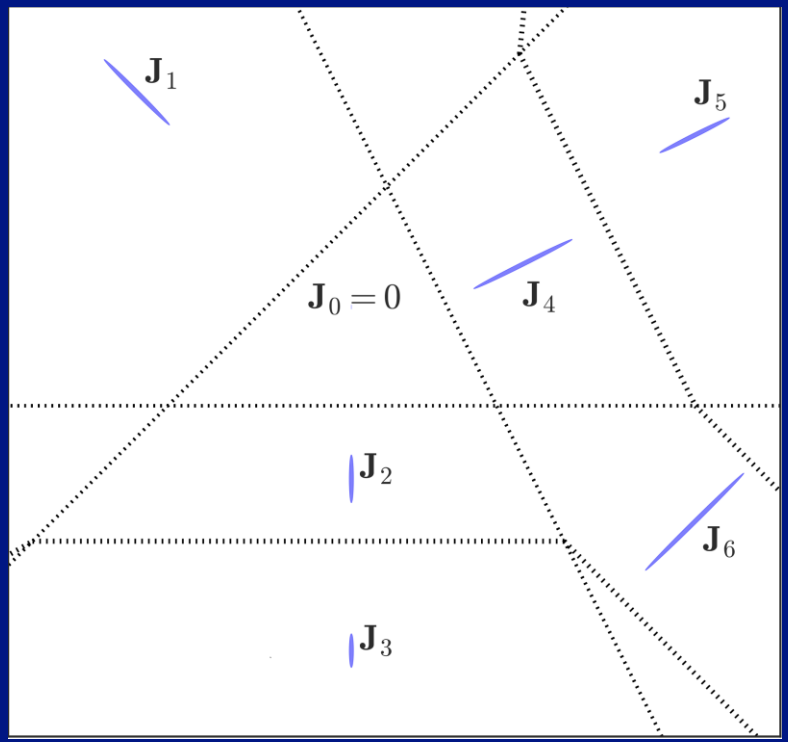


JACOBIAN BASED REGULARIZATION

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

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

- $J(V)$ 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

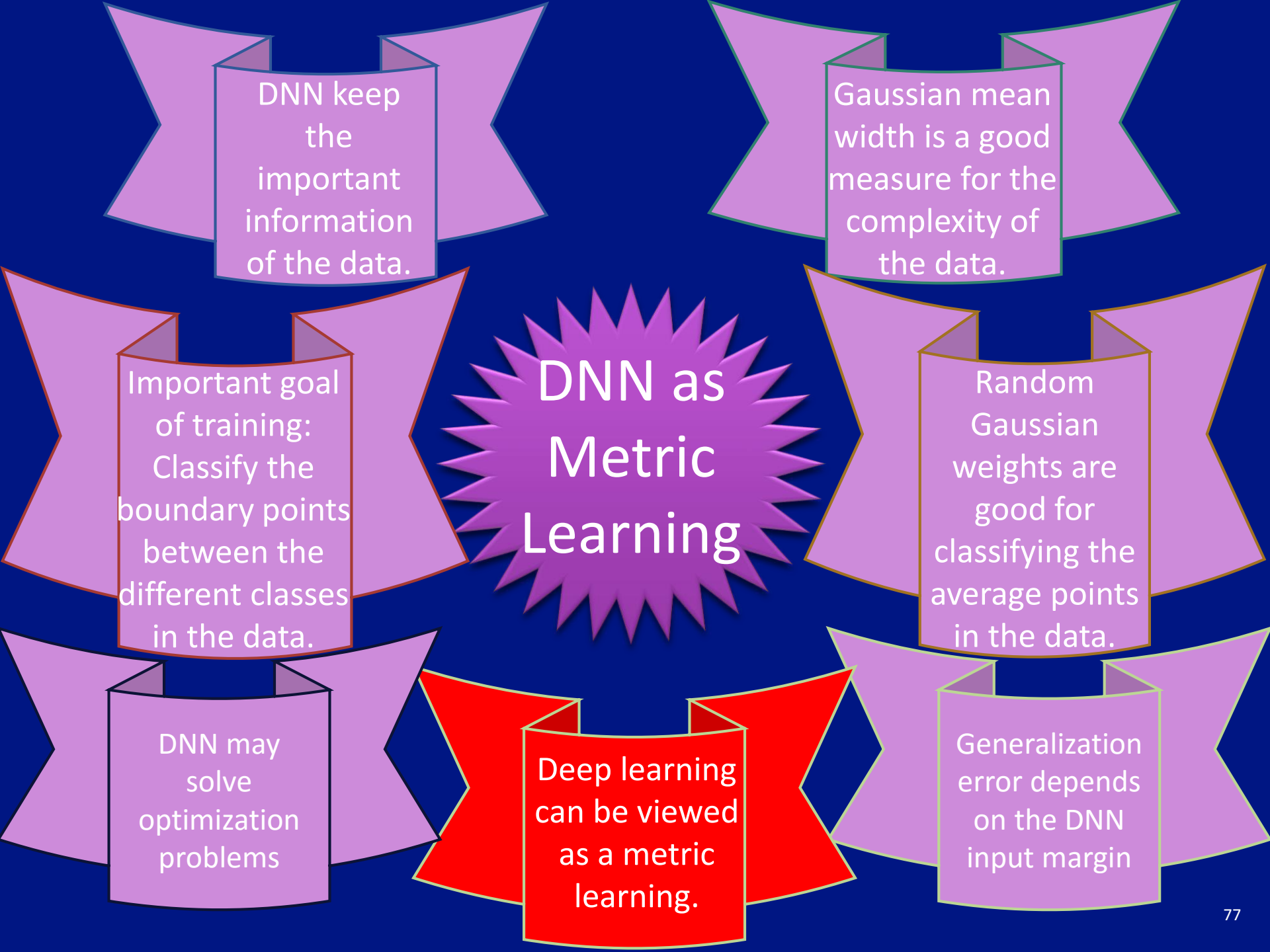
- Better performance with less training samples

MNIST Dataset

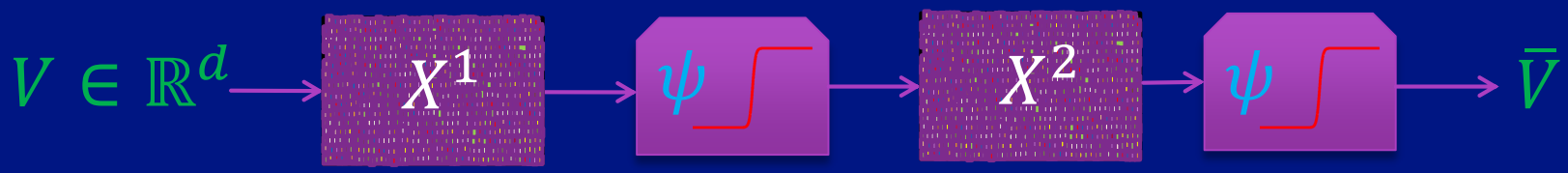
		256 samples			512 samples			1024 samples		
loss	# layers	no reg.	WD	LM	no reg.	WD	LM	no reg.	WD	LM
hinge	2	88.37	89.88	**93.83**	93.99	94.62	95.49	95.79	96.57	97.45
hinge	3	87.22	89.31	93.22	93.41	93.97	**95.76**	95.46	96.45	**97.60**
CCE	2	88.45	88.45	92.77	92.29	93.14	95.25	95.38	95.79	96.89
CCE	3	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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

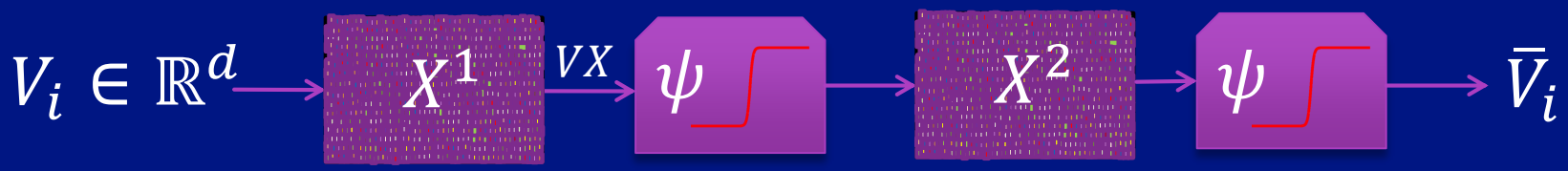
ASSUMPTIONS



X is fully connected and trained

ψ is the hyperbolic tan

METRIC LEARNING BASED TRAINING



- Cosine Objective:

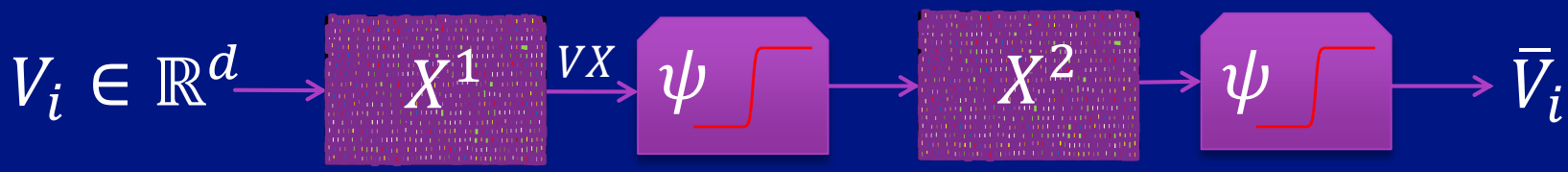
$$\min_{X^1,X^2} \sum_{i,j \in Training\ Set} \left(\frac{\bar{V}_i{}^T \bar{V}_j}{\|\bar{V}_i\| \|\bar{V}_j\|} - \vartheta_{i,j} \right)^2$$

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:

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

Classification term

$$+ \frac{1-\lambda}{|\text{Neighbours}|} \sum_{V_i, V_j \text{ are neighbours}} \left| \|\bar{V}_i - \bar{V}_j\| - \|V_i - V_j\| \right|$$

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

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

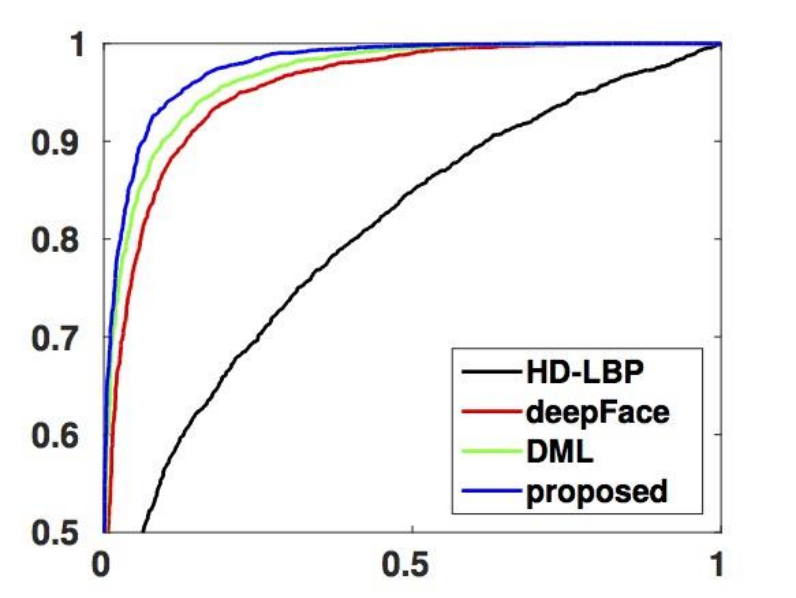
RESULTS

- Better performance with less training samples

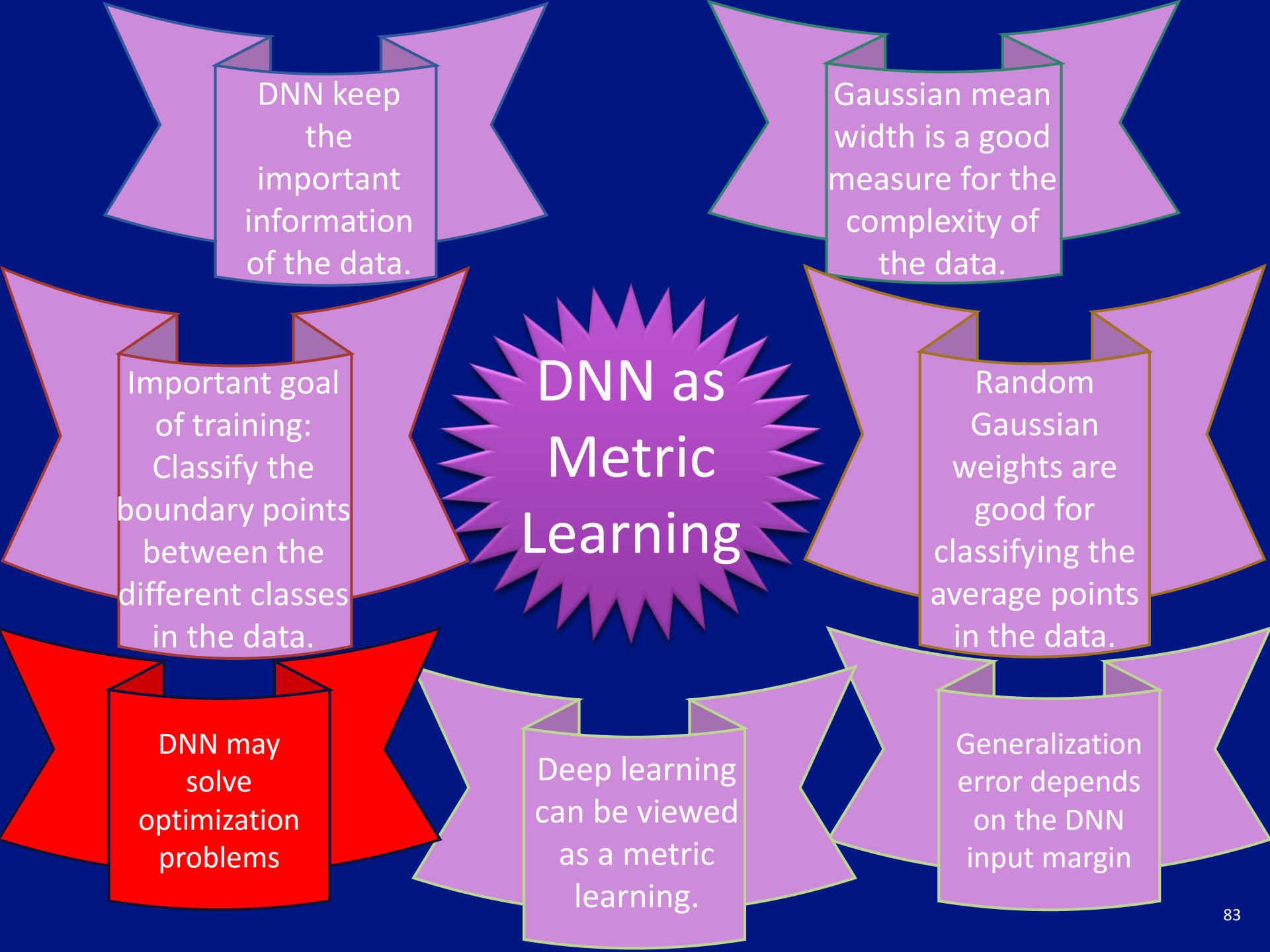
MNIST Dataset

#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%**

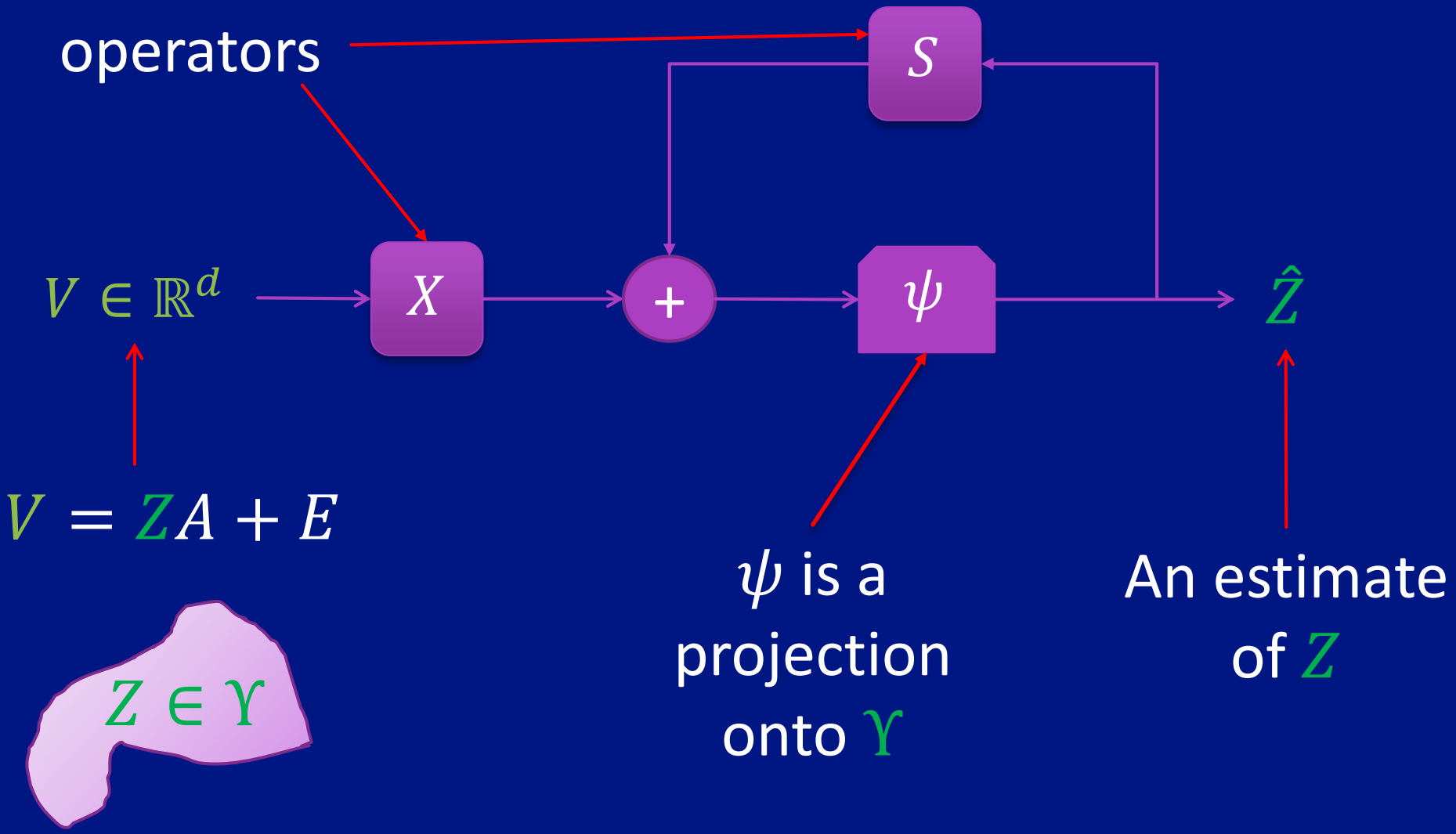
Faces in the wild ROC curve



[Huang, Qiu, Sapiro, Calderbank, 2015]

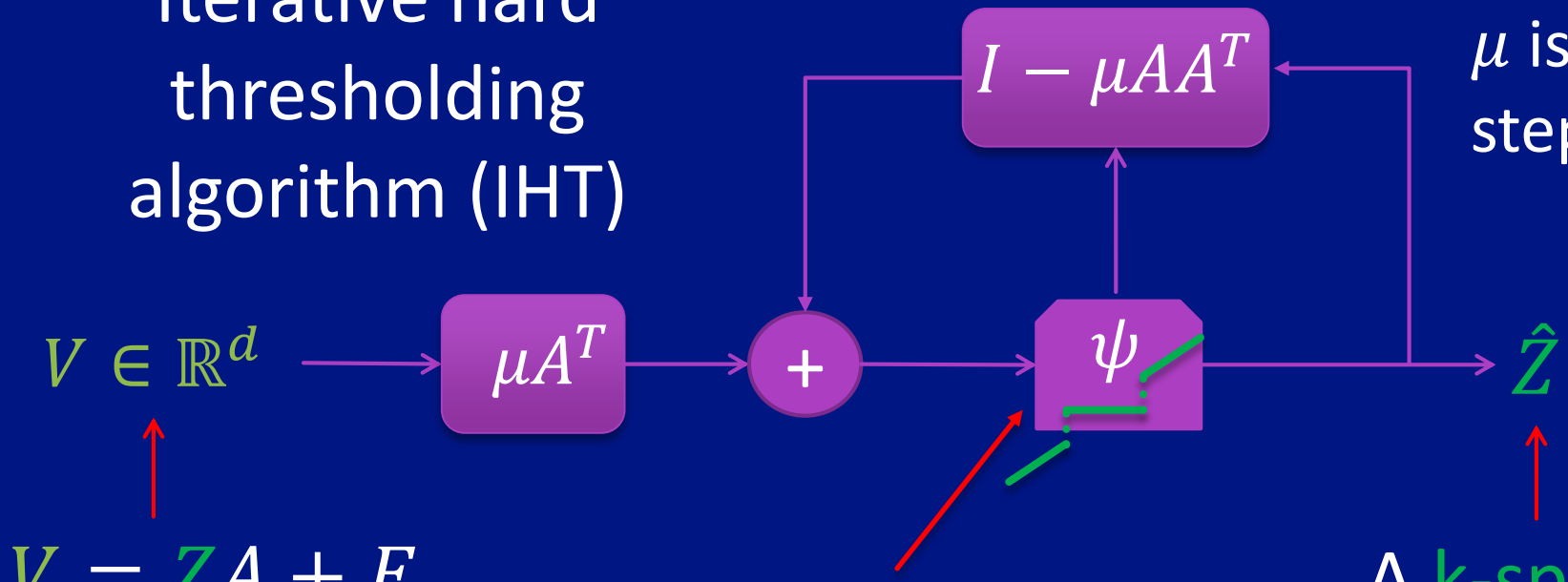

DNN keep the important information of the data.
Gaussian mean width is a good measure for the complexity of the data.
Important goal of training: Classify the boundary points between the different classes in the data.
DNN as Metric Learning
Random Gaussian weights are good for classifying the average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

ASSUMPTIONS



ℓ_0-MINIMIZATION

Iterative hard thresholding algorithm (IHT)



$V = ZA + E$

Z is a k–sparse vecotr

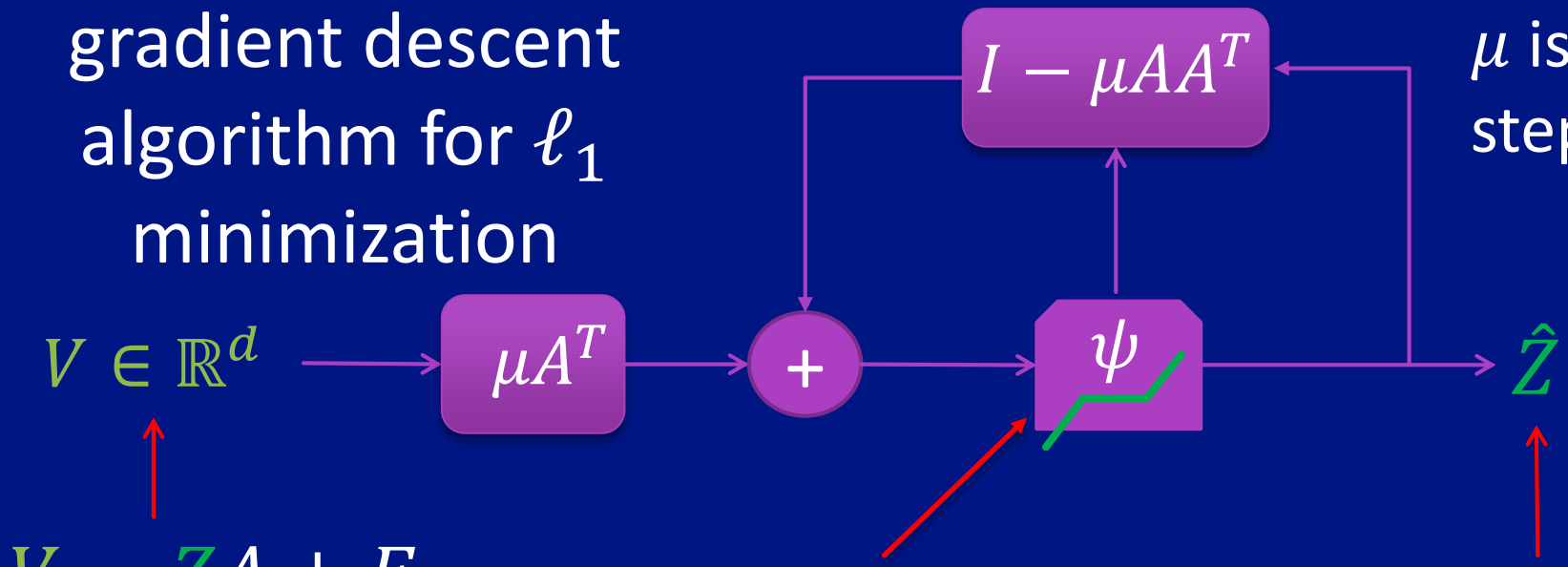
ψ is the hard thresholding operation: keeps the largest k entries

A k-sparse estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t \ \|\tilde{Z}\|_1 \leq k$$

ℓ_1-MINIMIZATION

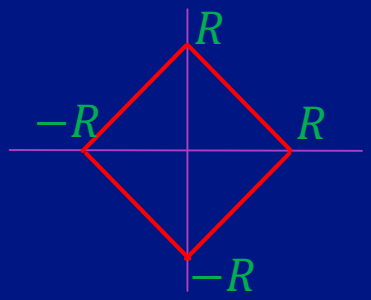
Projected gradient descent algorithm for ℓ_1 minimization



$V = ZA + E$

$\|Z\|_1 \leq R$

ψ projects onto the ℓ_1 ball

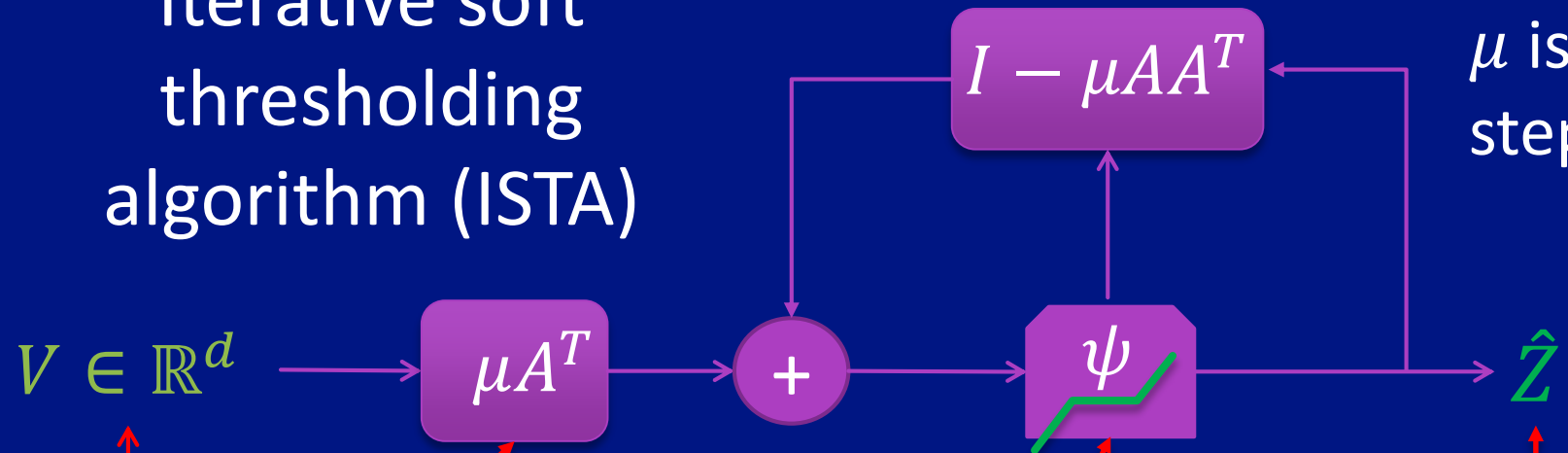


Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t \ \|\tilde{Z}\|_1 \leq R$$

UNCONSTRAINED ℓ_1-MINIMIZATION

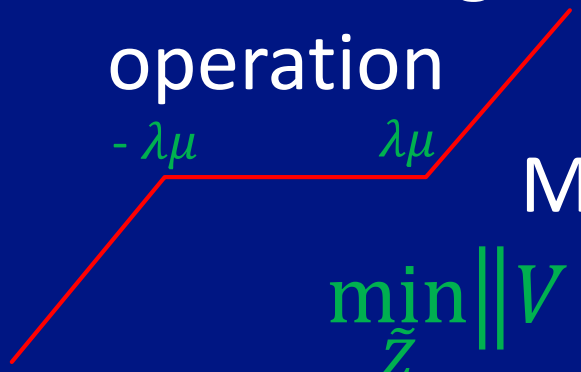
Iterative soft thresholding algorithm (ISTA)



μ is the step size

$\frac{1}{\mu} \geq \|A\|$

Soft thresholding operation



$V = ZA + E$

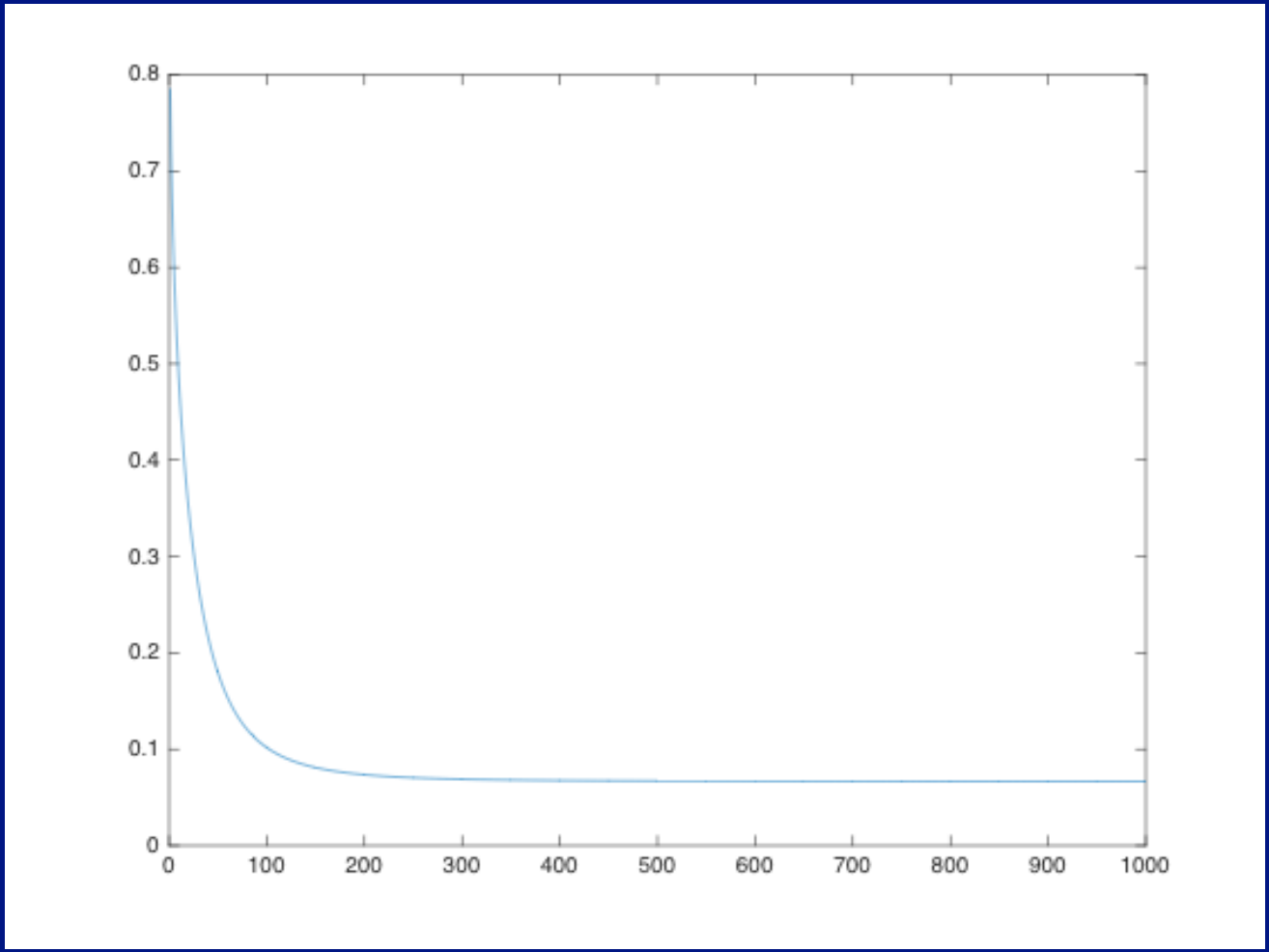
Minimizer of

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| + \lambda\|\tilde{Z}\|_1$$

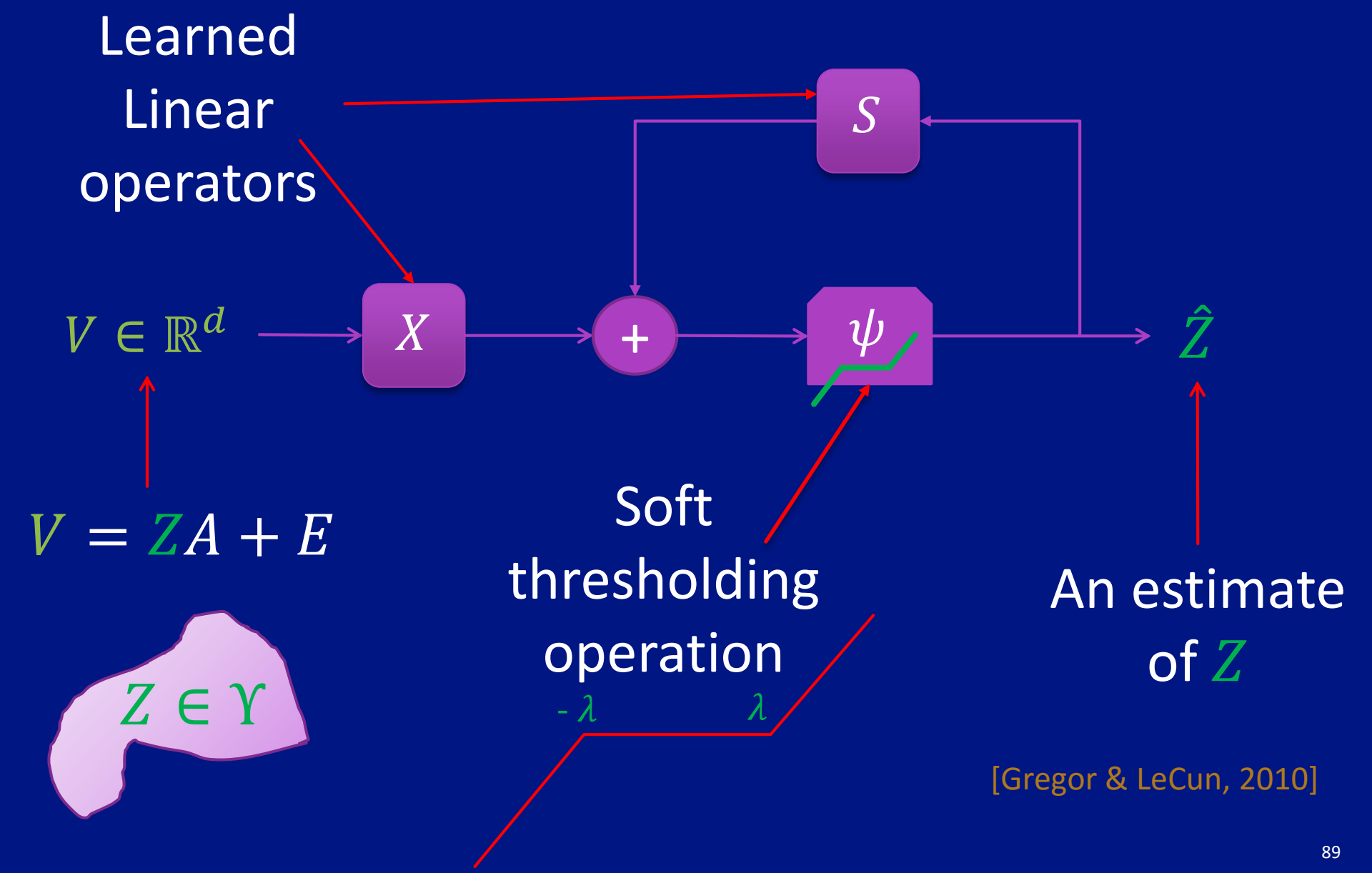
[Daubechies, Defrise, Mol, 2004]

ISTA CONVERGENCE

- Reconstruction error as a function of the number of iterations



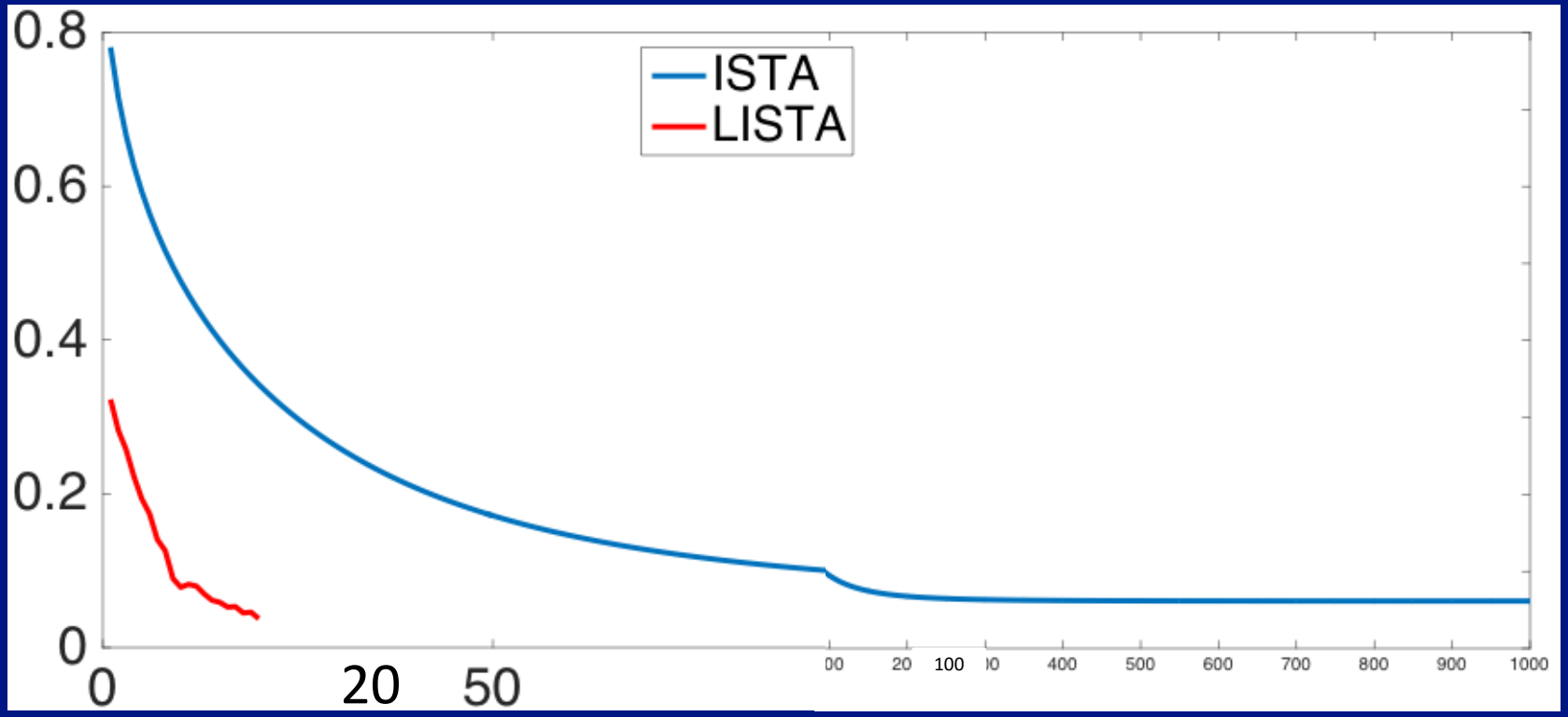
LEARNED ISTA (LISTA)



[Gregor & LeCun, 2010]

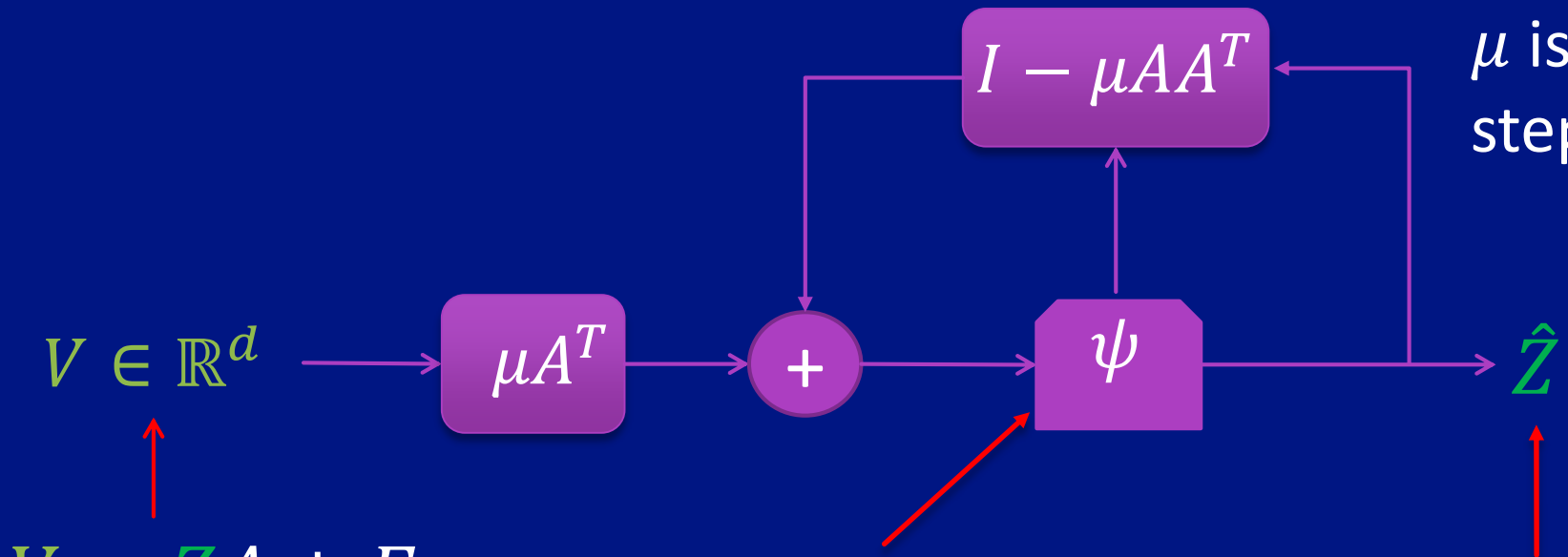
LISTA CONVERGENCE

- Replacing $I - \mu AA^T$ and μA^T in ISTA with the learned X and S improves convergence [Gregor & LeCun, 2010]



- Extensions to other models [Sprechmann, Bronstein & Sapiro, 2015], [Remez, Litani & Bronstein, 2015], [Tompson, Schlachter, Sprechmann & Perlin, 2016].

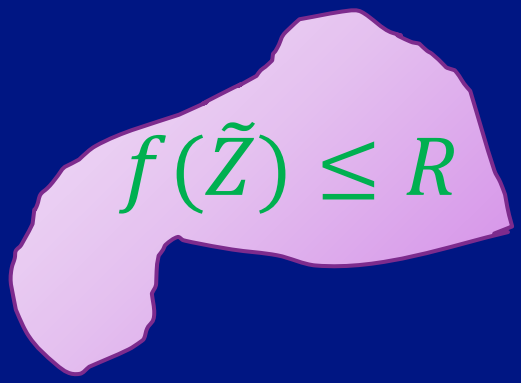
PROJECTED GRADIENT DESCENT (PGD)



$V = ZA + E$

$f(Z) \leq R$

ψ projects onto the set Υ



Estimate of Z.
Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t.\ f(\tilde{Z}) \leq R$$

THEORY FOR PGD

- Theorem 8: Let $Z \in \mathbb{R}^d$, $f: \mathbb{R}^d \to \mathbb{R}$ a proper function, $f(Z) \leq R$, $C_f(x)$ the tangent cone of f at point x, $A \in \mathbb{R}^{d \times m}$ a random Gaussian matrix and $V = ZA + E$. Then the estimate of PGD at iteration t, $\hat{Z}^t$, obeys

$$\left\| \hat{Z}^t - Z \right\| \leq \left(\kappa_f \rho \right)^t \|Z\|,$$

where $\rho = \sup_{U,W \in C_f(x) \cap \mathcal{B}^d} U\left(I - \mu A A^T\right) W^T$

and $\kappa_f = 1$ if f is convex and $\kappa_f = 2$ otherwise. [Oymak, Recht & Soltanolkotabi, 2016].

PGD CONVERGENCE RATE

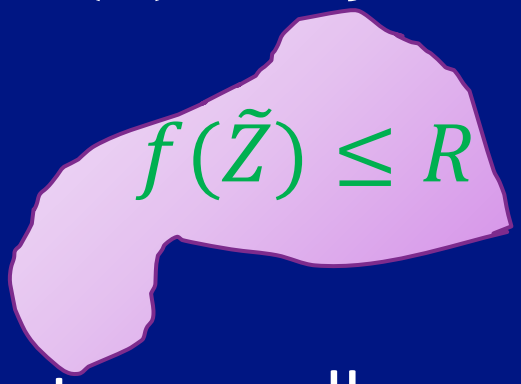
- $\rho = \sup_{U,W} U\left(I - \mu AA^T\right)W^T$ is the convergence rate of PGD.
- Let ω be the Gaussian mean width of $C_f(x) \cap \mathcal{B}^d$.
- If $\mu = \frac{1}{\left(\sqrt{m}+\sqrt{d}\right)^2} \simeq \frac{1}{d}$ then $\rho = 1 - O\left(\frac{\sqrt{m}-\omega}{m+d}\right)$.
- If $\mu = \frac{1}{m}$ then $\rho = O\left(\frac{\omega}{\sqrt{m}}\right)$.

INACCURATE PROJECTION

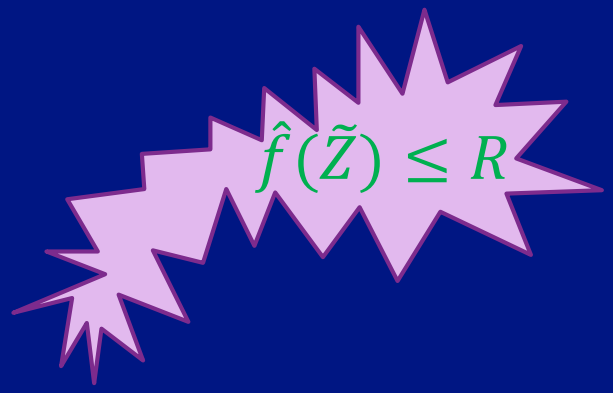
- PGD iterations projects onto $\Upsilon = \{\tilde{Z}: f(\tilde{Z}) \leq R\}$.
- Smaller $\Upsilon \Rightarrow$ Smaller ω.

$\Rightarrow$Faster convergence as

$\rho = 1 - O\left(\frac{\sqrt{m}-\omega}{m+d}\right)$ or $O\left(\frac{\omega}{\sqrt{m}}\right)$

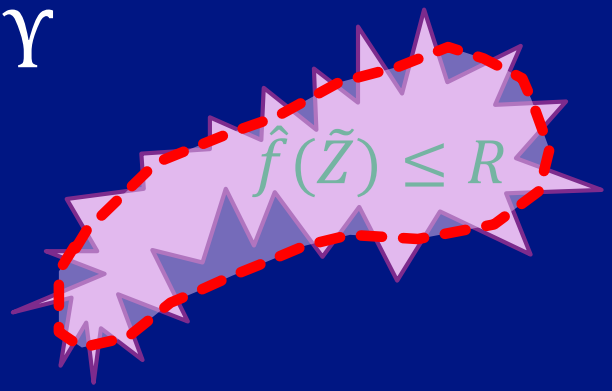


- Let us assume that our signal belongs to a smaller set $\hat{\Upsilon} = \{\tilde{Z}: \hat{f}(\tilde{Z}) \leq R\}$ with $\hat{\omega} \ll \omega$.
- Ideally, we would like to project onto $\hat{\Upsilon}$ instead of Υ.
- This will lead to faster convergence.
- What if such a projection is not feasible?



INACCURATE PROJECTION

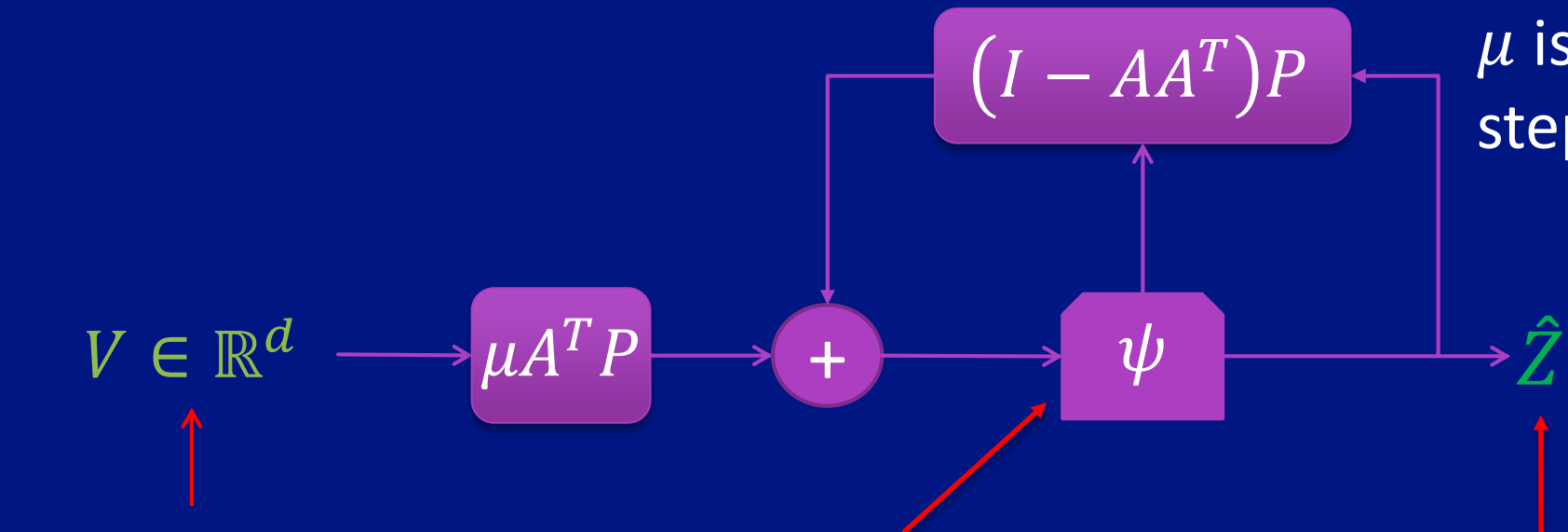
- We will estimate the projection onto $\widehat{\Upsilon}$ by
  - A linear projection P
  - Followed by a projection onto Υ
- Assumptions:
  - $\|P(Z)-Z\| \leq \epsilon$
  - $\left\| \wp_{C_{\hat{f}(Z)}}(U) - \wp_{C_{f(ZP)}}(UP) \right\| \leq \epsilon, \forall U \in \mathbb{R}^d$



Projection of U onto the tangent cone of $\hat{f}$ at point Z

Projection of UP onto the tangent cone of f at point ZP

INACCURATE PGD (IPGD)

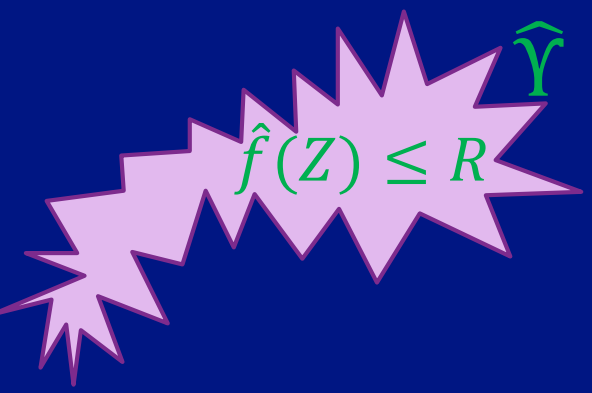


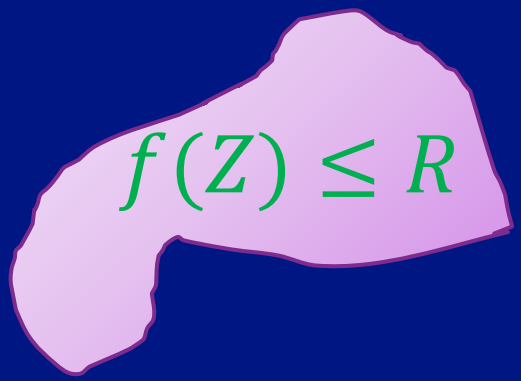
$V = ZA + E$

ψ projects onto the set Υ

Estimate of Z. Aim at solving

$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t.\ \hat{f}(\tilde{Z}) \leq R$$





THEORY FOR IPGD

- Theorem 9: Let $Z \in \mathbb{R}^d$, $f: \mathbb{R}^d \to \mathbb{R}$ a proper function, $f(Z) \leq R$, $\hat{C}_f(x)$ the tangent cone of f at point x, $A \in \mathbb{R}^{d \times m}$ a random Gaussian matrix and $V = ZA + E$. Then the estimate of IPGD at iteration t, $\hat{Z}^t$, obeys

$$\|\hat{Z}^t - Z\| \leq \left(\left(\kappa_f(\hat{\rho} + \epsilon\gamma)\right)^t + \frac{1 - \left(\kappa_f(\hat{\rho} + \epsilon\gamma)\right)^t}{1 - \kappa_f(\hat{\rho} + \epsilon\gamma)} \epsilon \right) \|Z\|,$$

where $\rho = \sup_{U,W \in C_f(x) \cap \mathcal{B}^d} U\left(I - \mu AA^T\right)W^T$

$\gamma = \left\|I - \mu AA^T\right\|$ and κ_f as in Theorem 8.

[Giryes, Eldar, Bronstein & Sapiro, 2016]

CONVERGENCE RATE COMPARISON

- PGD:

$$\left(\kappa_f \rho\right)^t$$

- IPGD:

$$\left(\left(\kappa_f(\hat{\rho}+\epsilon\gamma)\right)^t + \frac{1-\left(\kappa_f(\hat{\rho}+\epsilon\gamma)\right)^t}{1-\kappa_f(\hat{\rho}+\epsilon\gamma)}\epsilon\right)$$

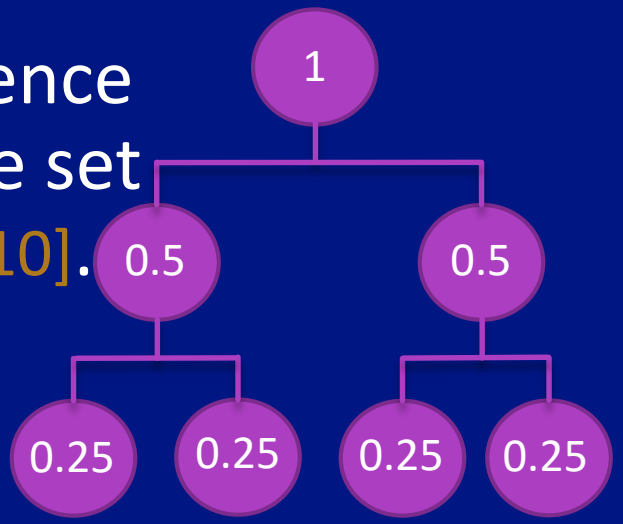
$$\overset{(a)}{\approx} \left(\kappa_f(\hat{\rho})\right)^t + \epsilon \overset{(b)}{\ll} \left(\kappa_f \rho\right)^t$$

(a) assuming that ϵ is negligible compared to $\hat{\rho}$

(b) For small values of t

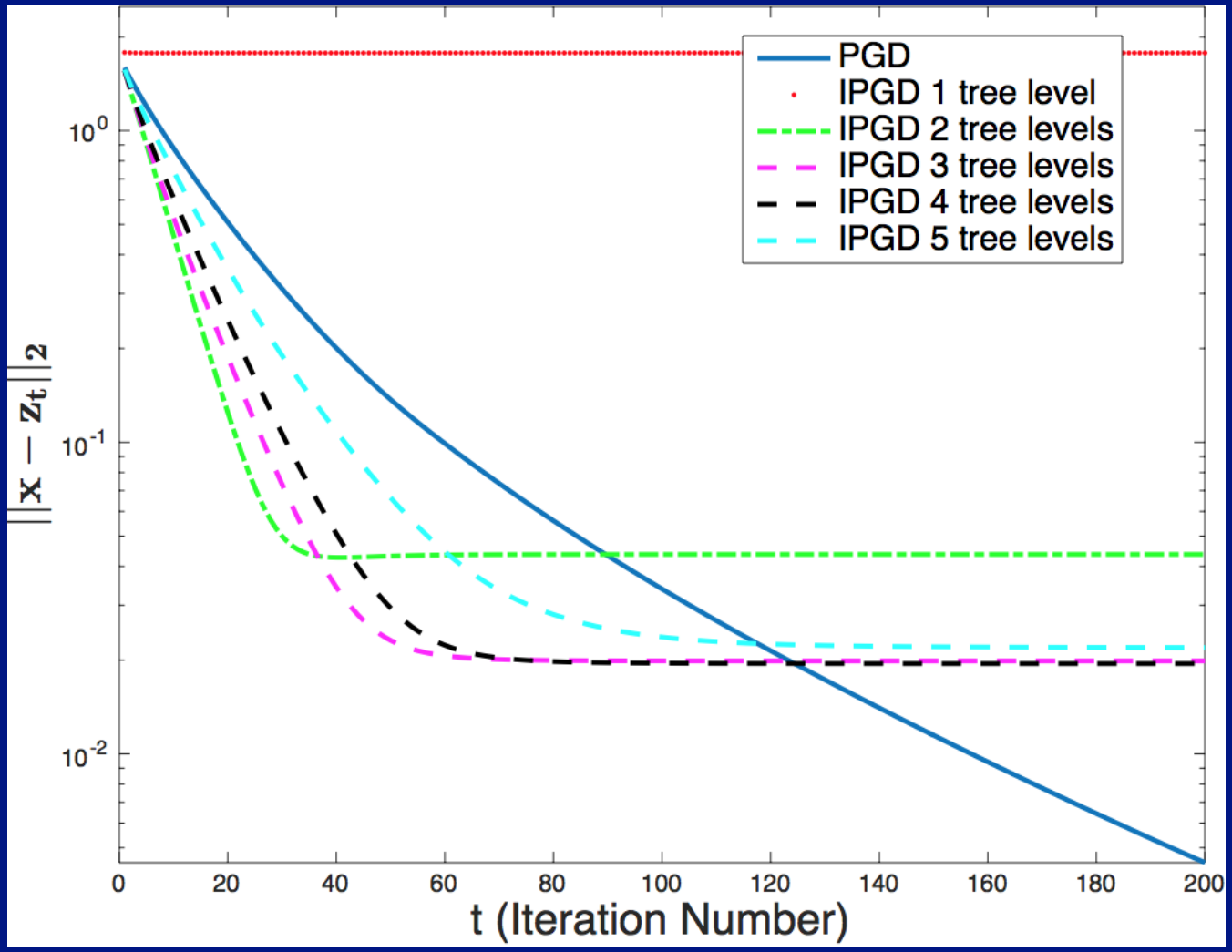
MODEL BASED COMPRESSED SENSING

- $\widehat{\Upsilon}$ is the set of sparse vectors with sparsity patterns that obey a tree structure.
- Projecting onto $\widehat{\Upsilon}$ improves convergence rate compared to projecting onto the set of sparse vectors Υ [Baraniuk et al., 2010].
- The projection onto $\widehat{\Upsilon}$ is more demanding than onto Υ.



- Note that the probability of selecting atoms from lower tree levels is smaller than upper ones.
- P will be a projection onto certain tree levels – zeroing the values at lower levels.

MODEL BASED COMPRESSED SENSING

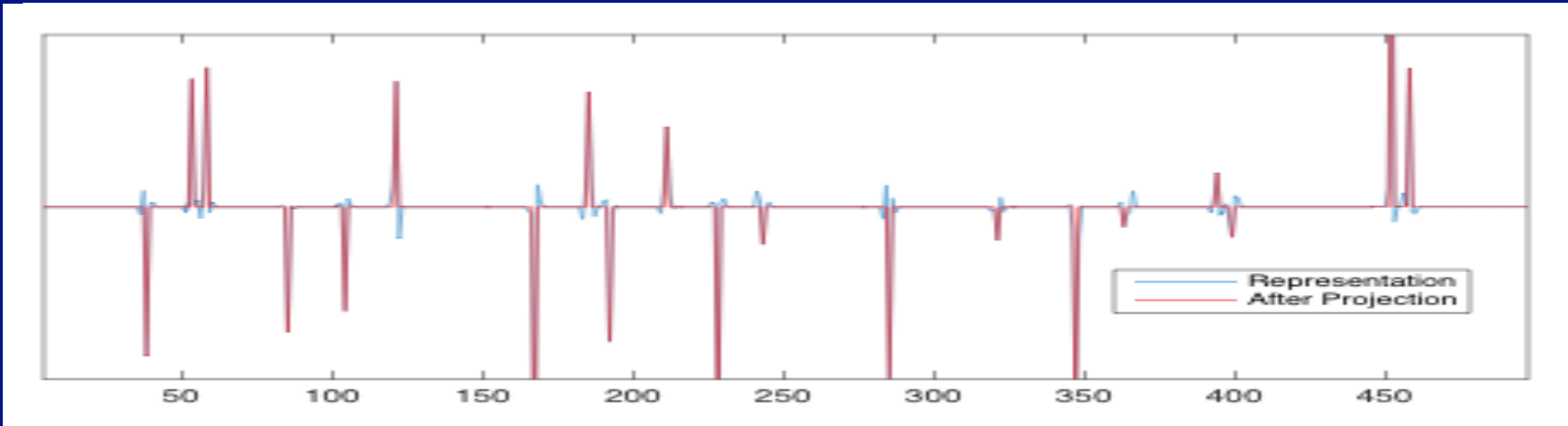


Non-zeros picked entries has zero mean random Gaussian distribution with variance:

- 1 at first two levels
- 0.1^2 at the third level
- 0.01^2 at the rest of the levels

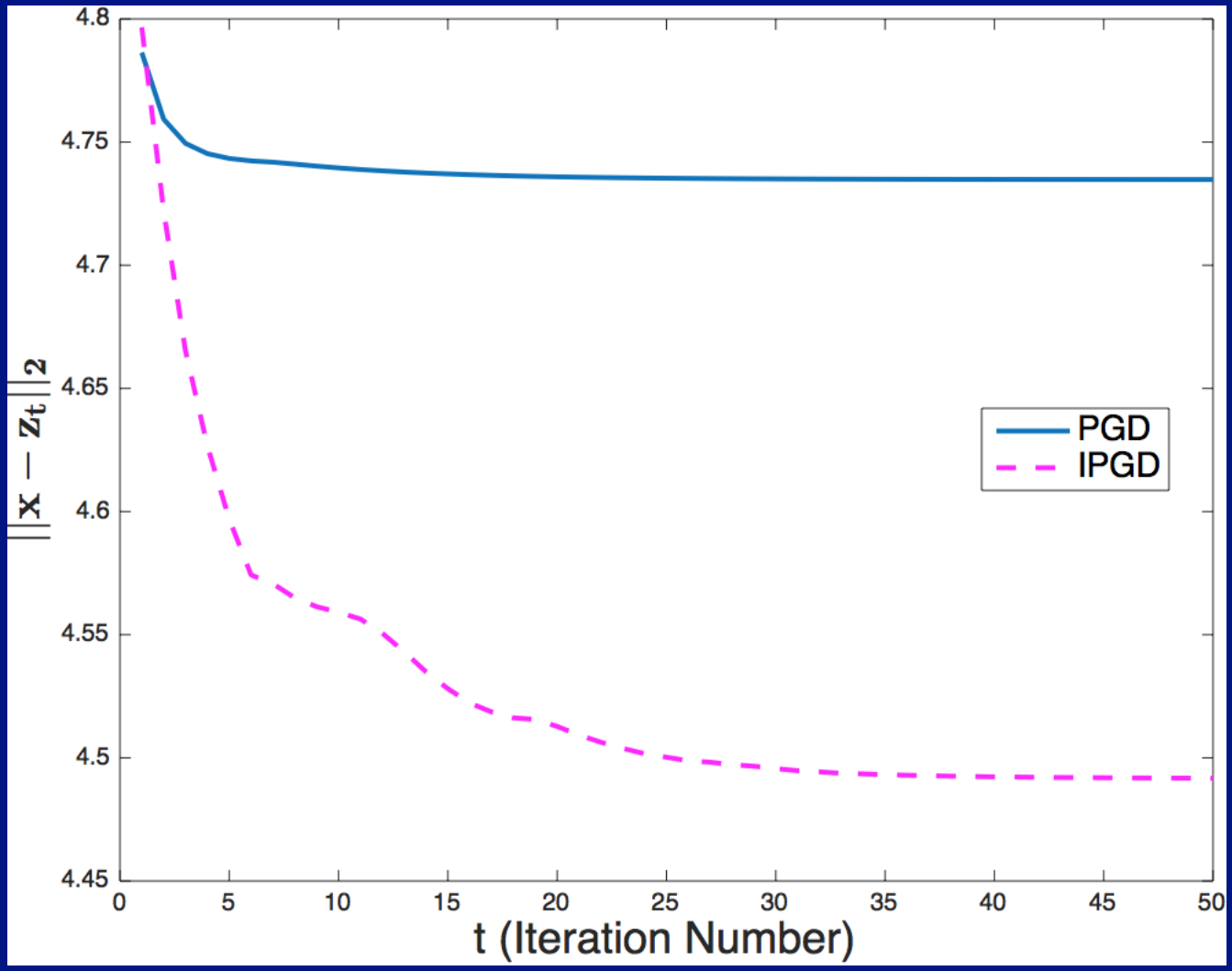
SPECTRAL COMPRESSED SENSING

- $\widehat{\Upsilon}$ is the set of vectors with sparse representation in a 4-times redundant DCT dictionary such that:



- We set P to be a pooling-like operation that keeps in each window of size 5 only the largest value.

SPECTRAL COMPRESSED SENSING

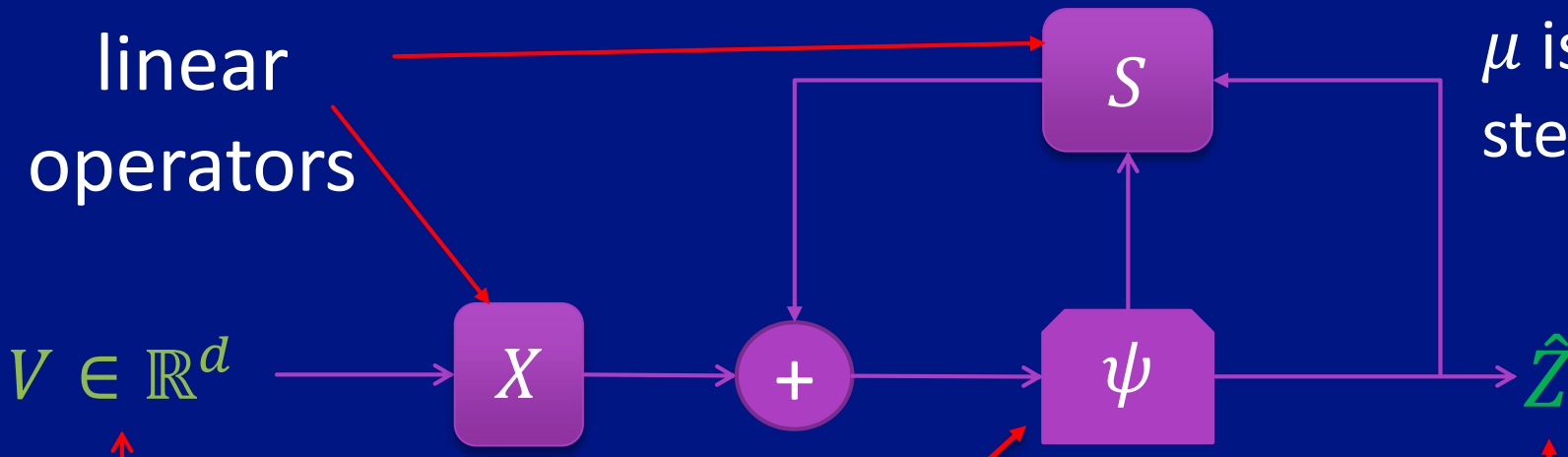


LEARNING THE PROJECTION

- In we have no explicit information about $\widehat{\mathrm{K}}$ it might be desirable to learn the projection.
- Instead of learning P, it is possible to replace $\left(I - AA^T\right)P$ and $\mu A^T P$ with two learned matrices S and X respectively.
- This leads to a very similar scheme to the one of LISTA and provides a theoretical foundation for the success of LISTA.

LEARNED IPGD

Learned linear operators



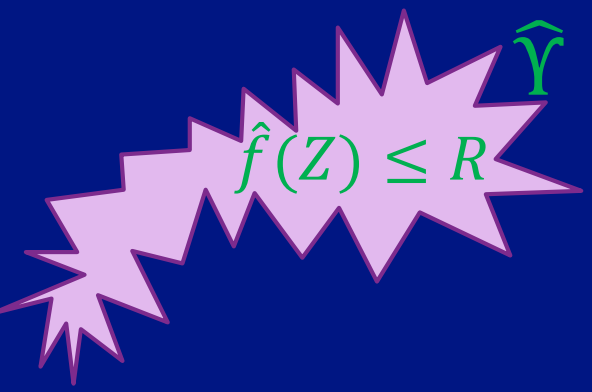
μ is the step size

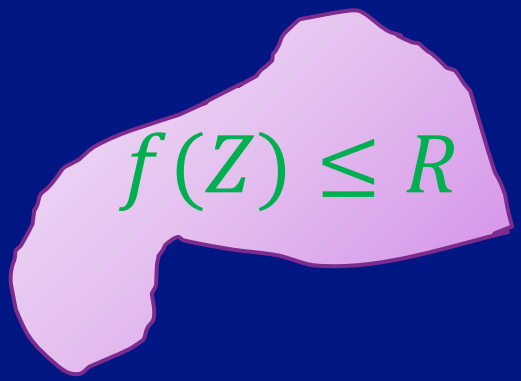
$V = ZA + E$

ψ projects onto the set Υ

Estimate of Z. Aim at solving

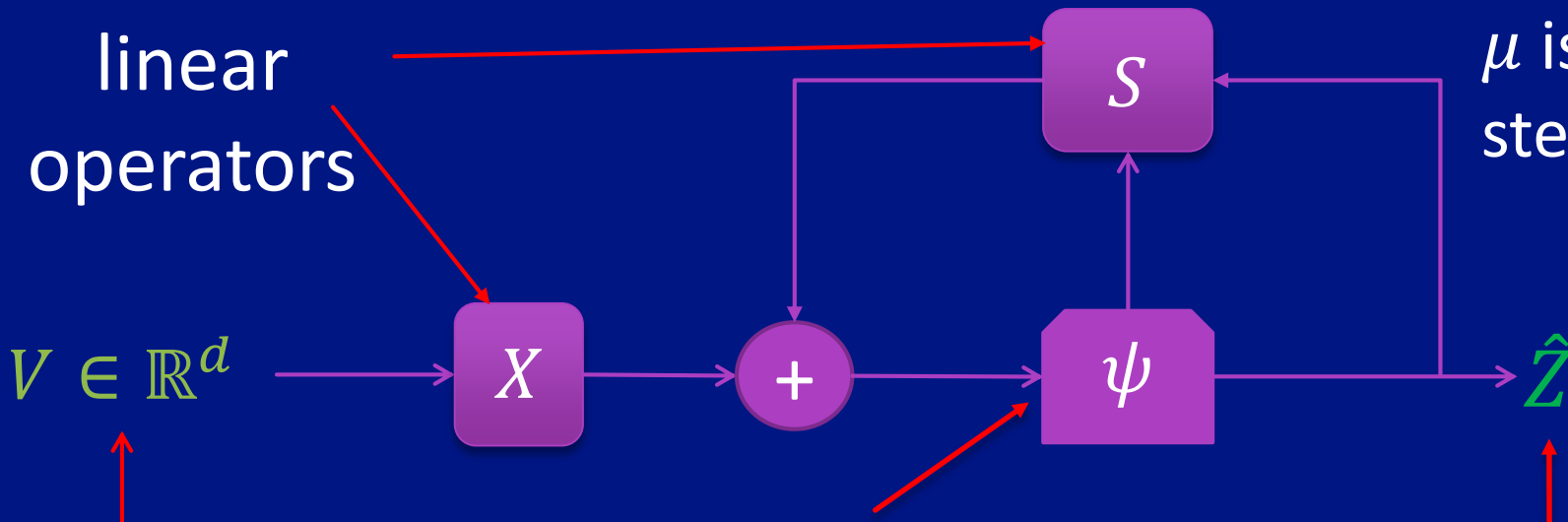
$$\min_{\tilde{Z}} \|V - \tilde{Z}A\| \quad s.t.\ \hat{f}(\tilde{Z}) \leq R$$





LISTA

Learned linear operators



μ is the step size

$V = ZA + E$

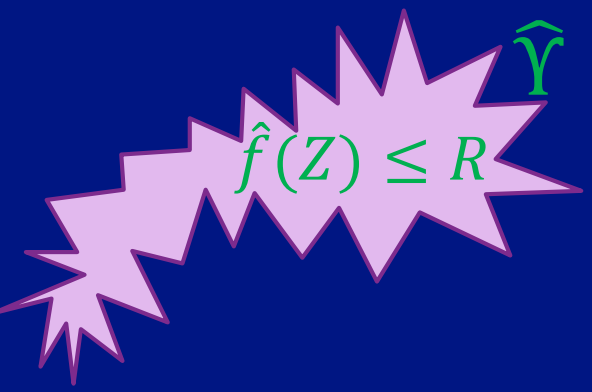
ψ is a proximal mapping.

$\psi(U) = \underset{\tilde{Z}}{\operatorname{argmin}} \|U - \tilde{Z}\| + \lambda f(\tilde{Z})$

Estimate of Z.

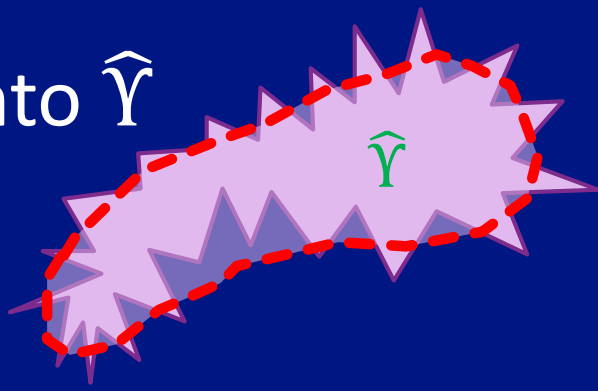
Aim at solving

$\min_{\tilde{Z}} \|V - \tilde{Z}A\| + \lambda f(\tilde{Z})$

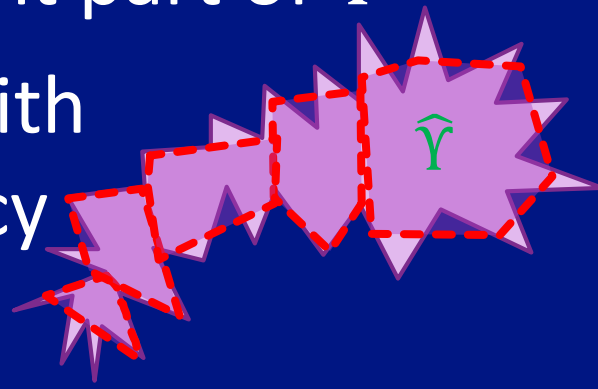


LISTA MIXTURE MODEL

- Approximation of the projection onto $\widehat{\Upsilon}$ with one linear projection may not be accurate enough.

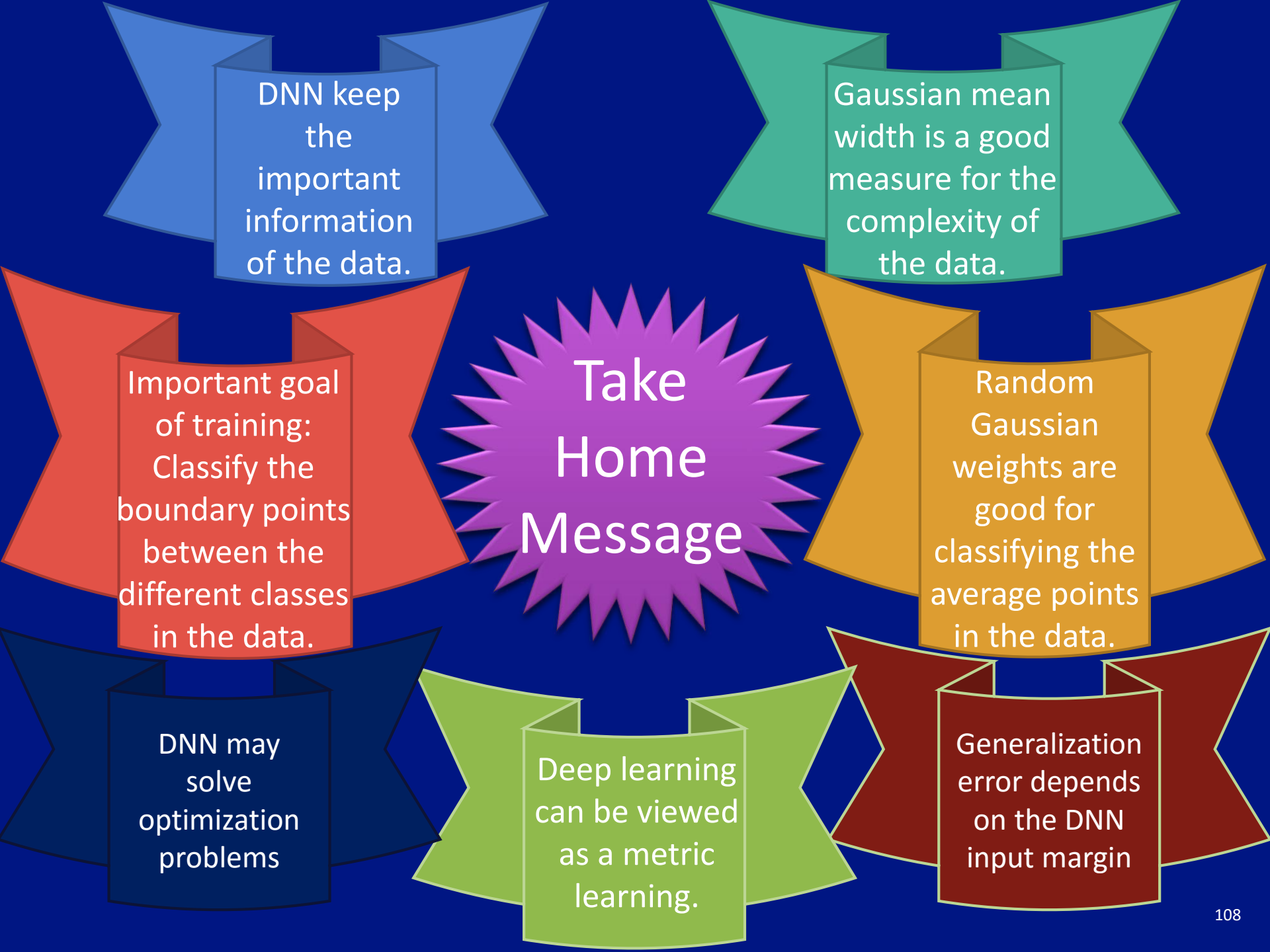


- This requires more LISTA layers/iterations.
- Instead, one may use several LISTA networks, where each approximates a different part of $\widehat{\Upsilon}$
- Training 18 LISTA networks, each with 3 layers, provides the same accuracy like 1 LISTA network with 10 layers.



RELATED WORKS

- In [Bruna et al. 2016] a different route it taken to explain the faster convergence of LISTA. It is shown that a learning may give a gain due to better preconditioning of *A*.


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.
DNN may solve optimization problems
Deep learning can be viewed as a metric learning.
Generalization error depends on the DNN input margin

ACKNOWLEDGEMENTS

Yonina C. Eldar
Technion

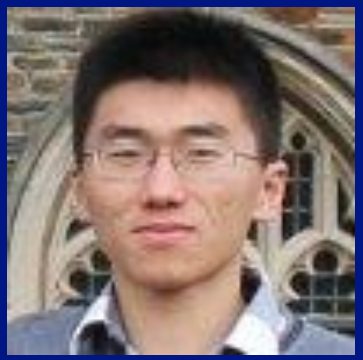
Guillermo Sapiro
Duke University

Robert Calderbank
Duke University

Miguel Rodrigues
University College London

Alex M. Bronstein
Technion

Qiang Qiu
Duke University



Jiaji Huang
Duke University

Jure Sokolic
University College London

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

R. GIRYES, Y. C. ELDAR, A. M. BRONSTEIN, G. SAPIRO, TRADEOFFS BETWEEN CONVERGENCE SPEED AND RECONSTRUCTION ACCURACY IN INVERSE PROBLEMS