From Two-Class Linear Discriminant Analysis to Interpretable Multilayer Perceptron Design

C.-C. Jay Kuo
University of Southern California

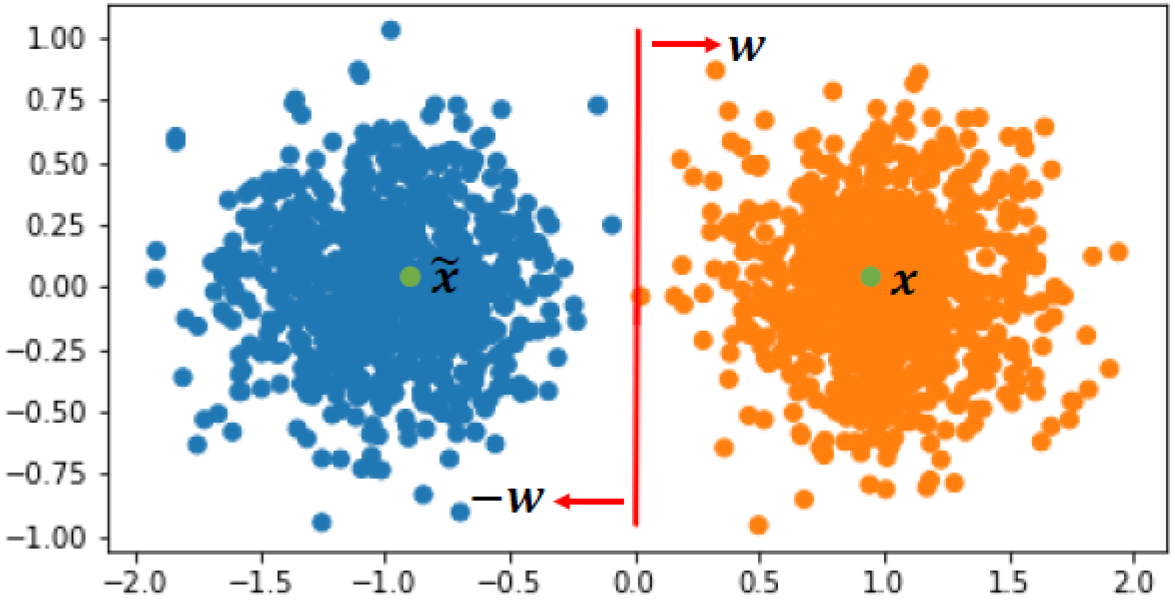
Research Background

- Multilayer Perceptron (MLP)
  - Proposed by Rosenblatt in 1958
  - Intensively studied in late 80’s and early 90’s
    - One important architecture for ANN (artificial neural networks)
  - Two main issues
    - Architecture design is ad hoc
    - Lack of theoretical support
- Main theoretical results
  - Universal approximators
    - by Cybenko (1989) and Hornik, Stinchocombe and White (1989)

Two-Class Linear Discriminant Analysis (LDA)

- Multi-dimensional input space
- Gaussian distributed random vectors
- Two object classes (orange and blue)

$$w^T x + b = 0$$



$w^T x + b < 0$ $\quad$ $w^T x + b > 0$

Class C2 $\quad$ **Class C1**

Homoscedasticity

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$$

Solution: a partitioning hyperplane

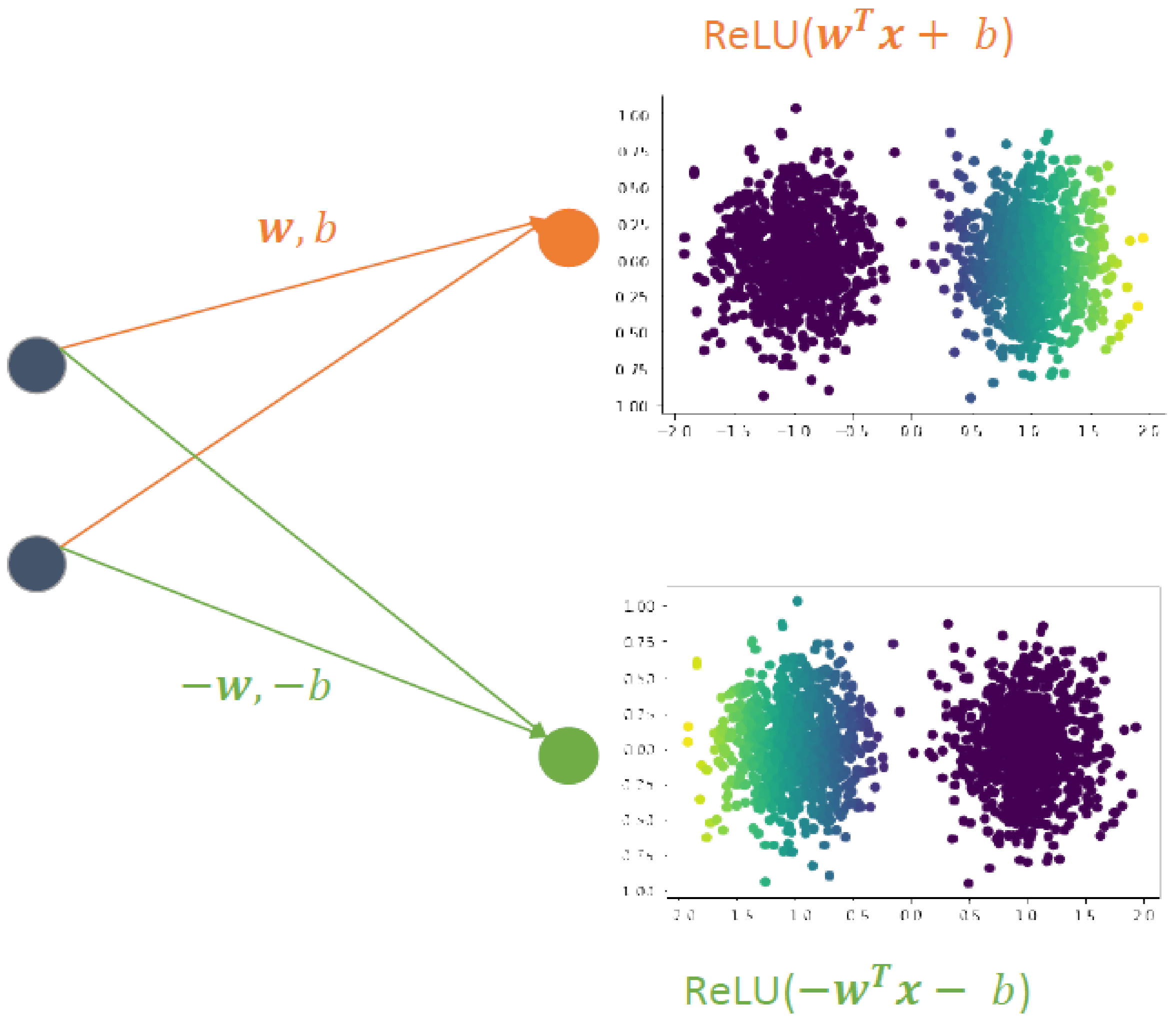
$$\boldsymbol{w}^T \boldsymbol{x} + b = 0,$$

$$\mathbf{w} = (w_1, w_2)^T = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2).$$

$$b = \frac{1}{2}\boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \log \frac{p}{1-p}$$

and p is the probability for x belonging to class C1

Relationship with the 1st Layer of MLP

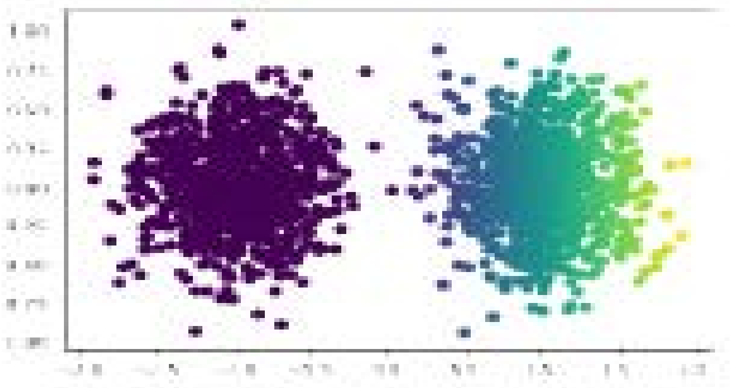


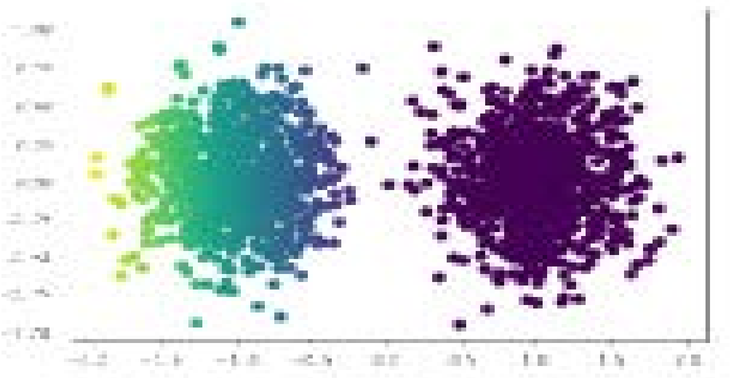
Two Questions:

1) **Why 2 neurons?**
2) **Why ReLU nonlinear activation?**

Answers to 2 Questions (1)

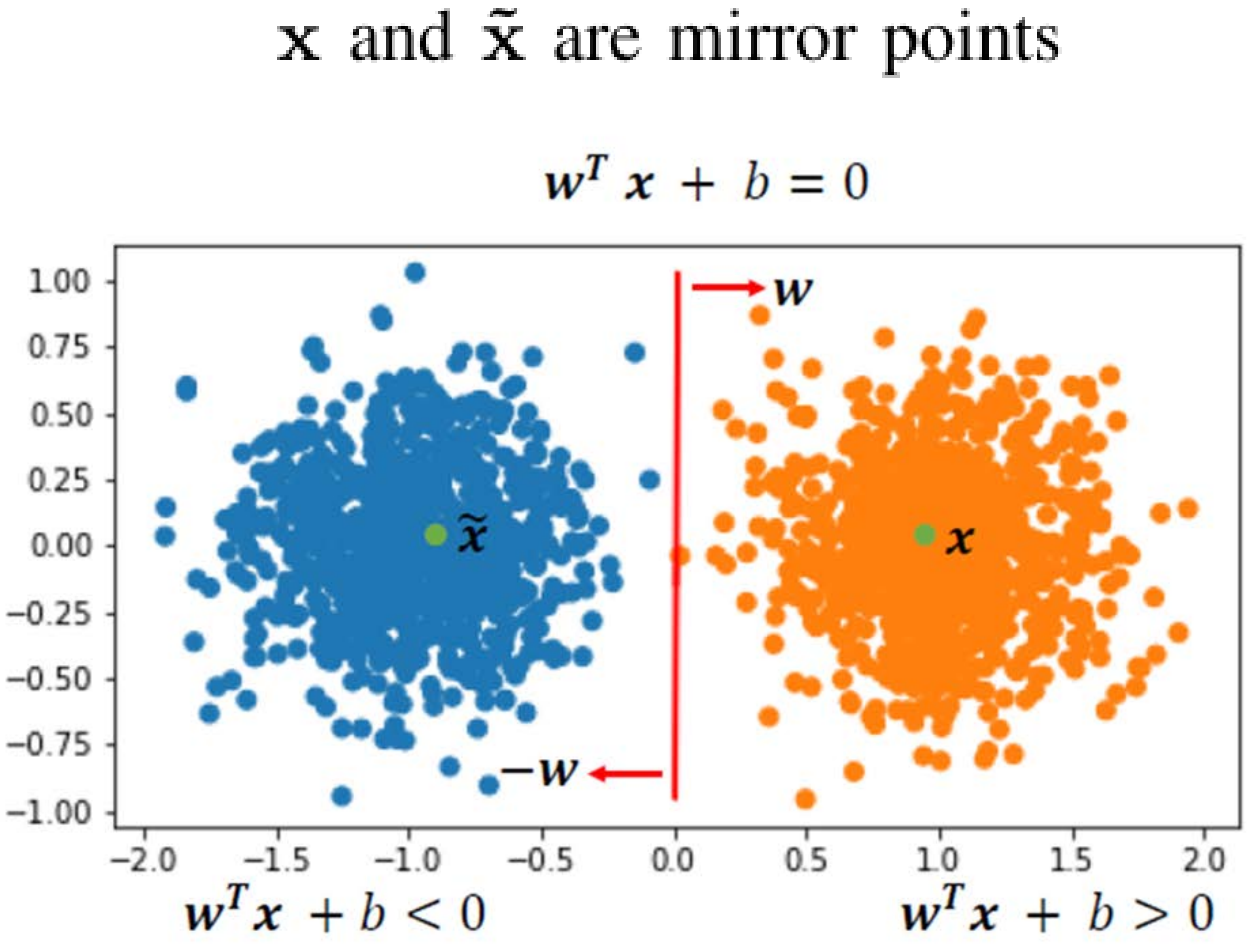
- Two neurons -> preserve responses in both sides of the hyperplanes





Answers to 2 Questions (2)

- ReLU nonlinear activation -> resolve the sign confusion problem



Next Stage Convolution

$$\mathbf{z}_j = \sum_i \tilde{\mathbf{w}}_{ji} \mathbf{y}_i$$

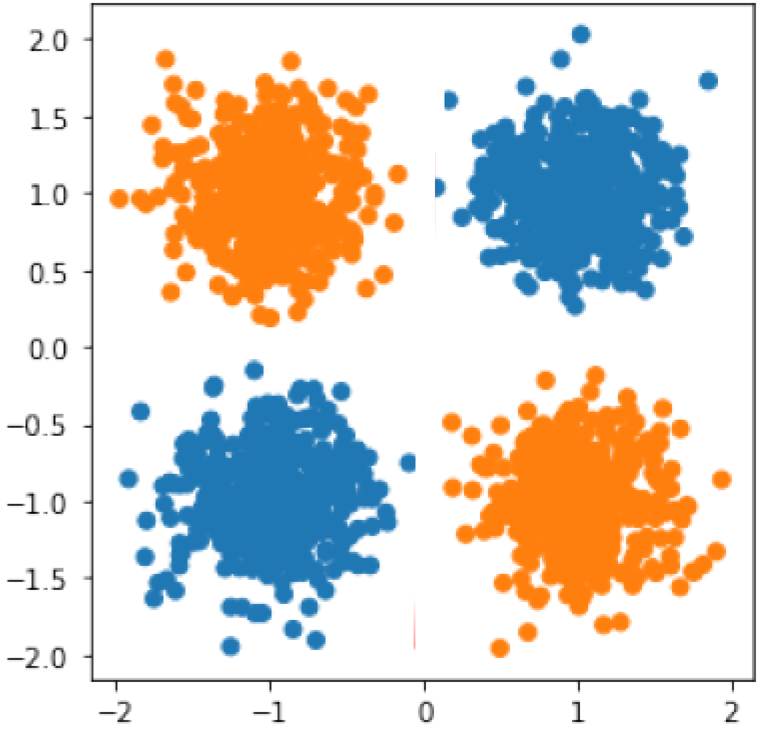
It cannot differentiate if no ReLU

- **Positive response multiplied by positive weights (+,+)**
- **Negative response multiplied by negative weights (-,-)**

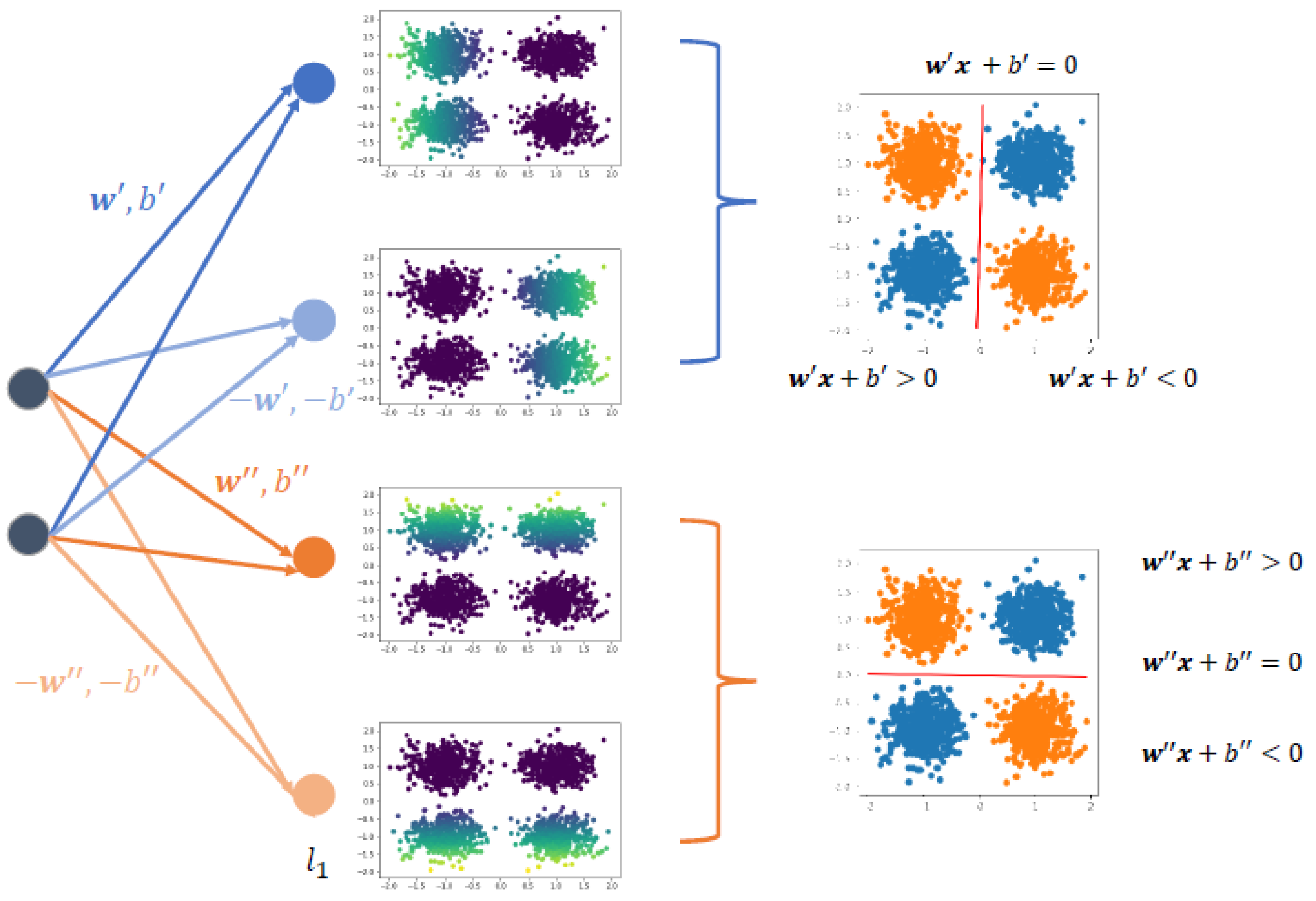
Similarly, it cannot differentiate if no ReLU

- (+,-), (-,+)

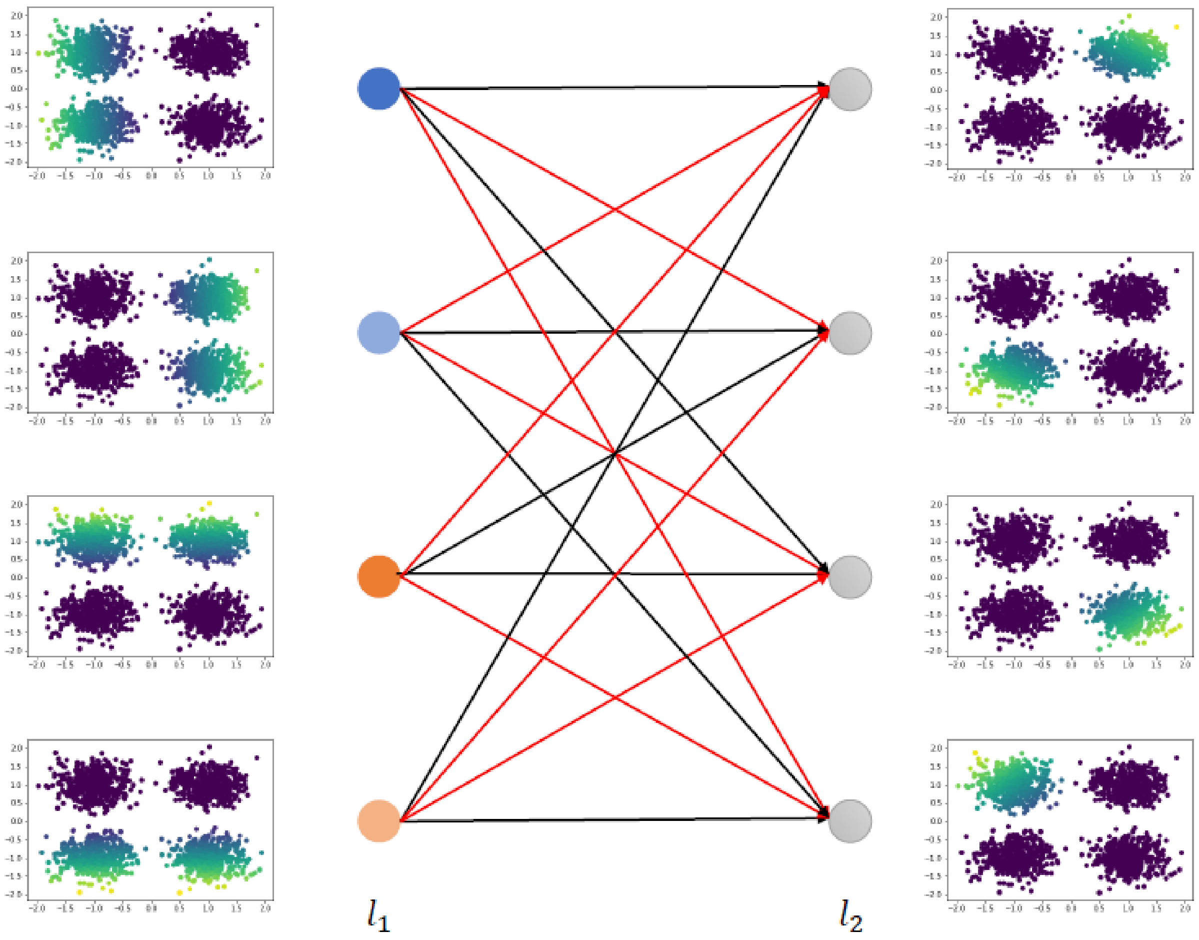
Illustrative Example: 4-Gaussian-Blobs



Stage 1 (from Input Layer to the 1st Layer)

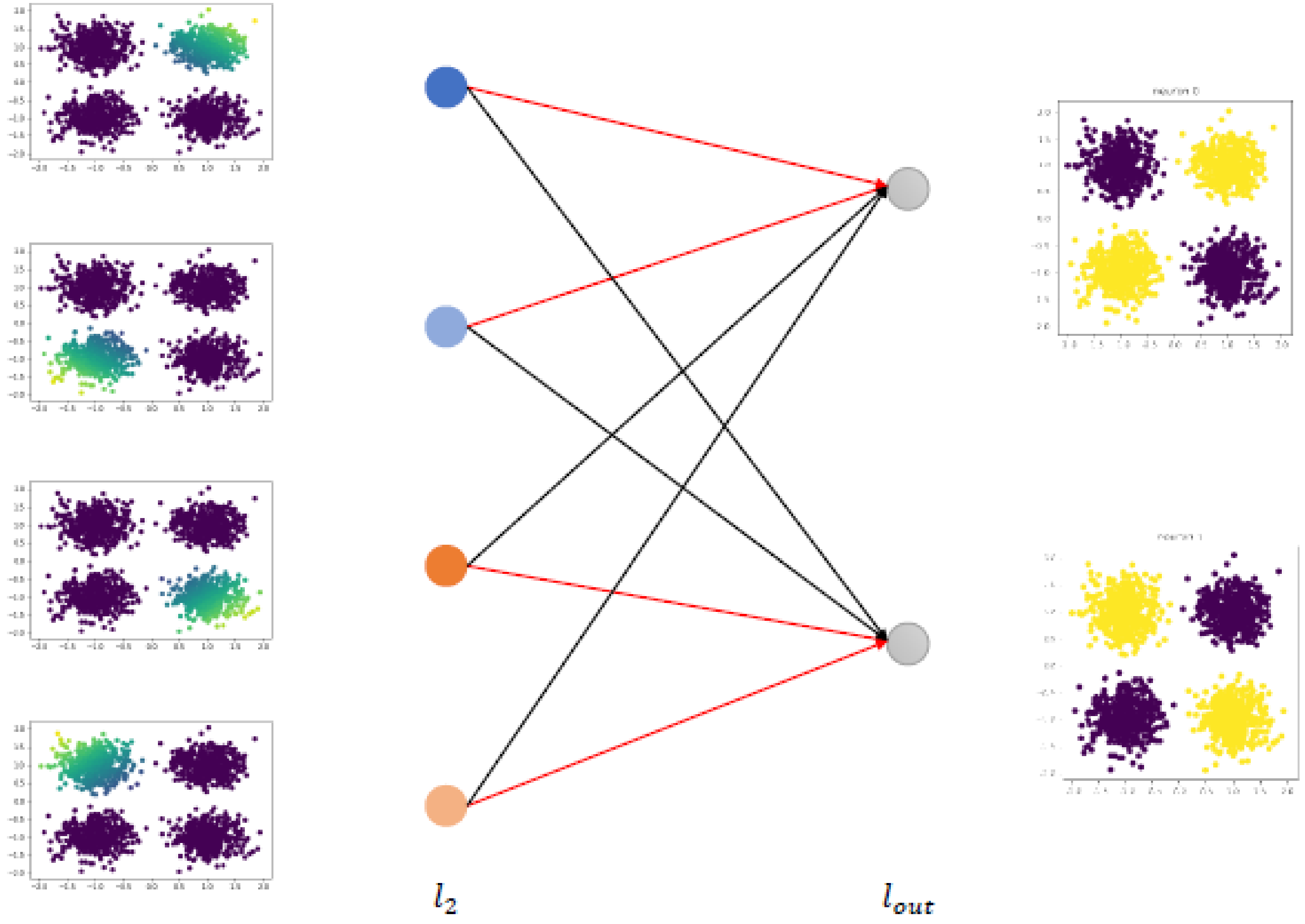


Stage 2 (from the 1st Layer to the 2nd Layer)



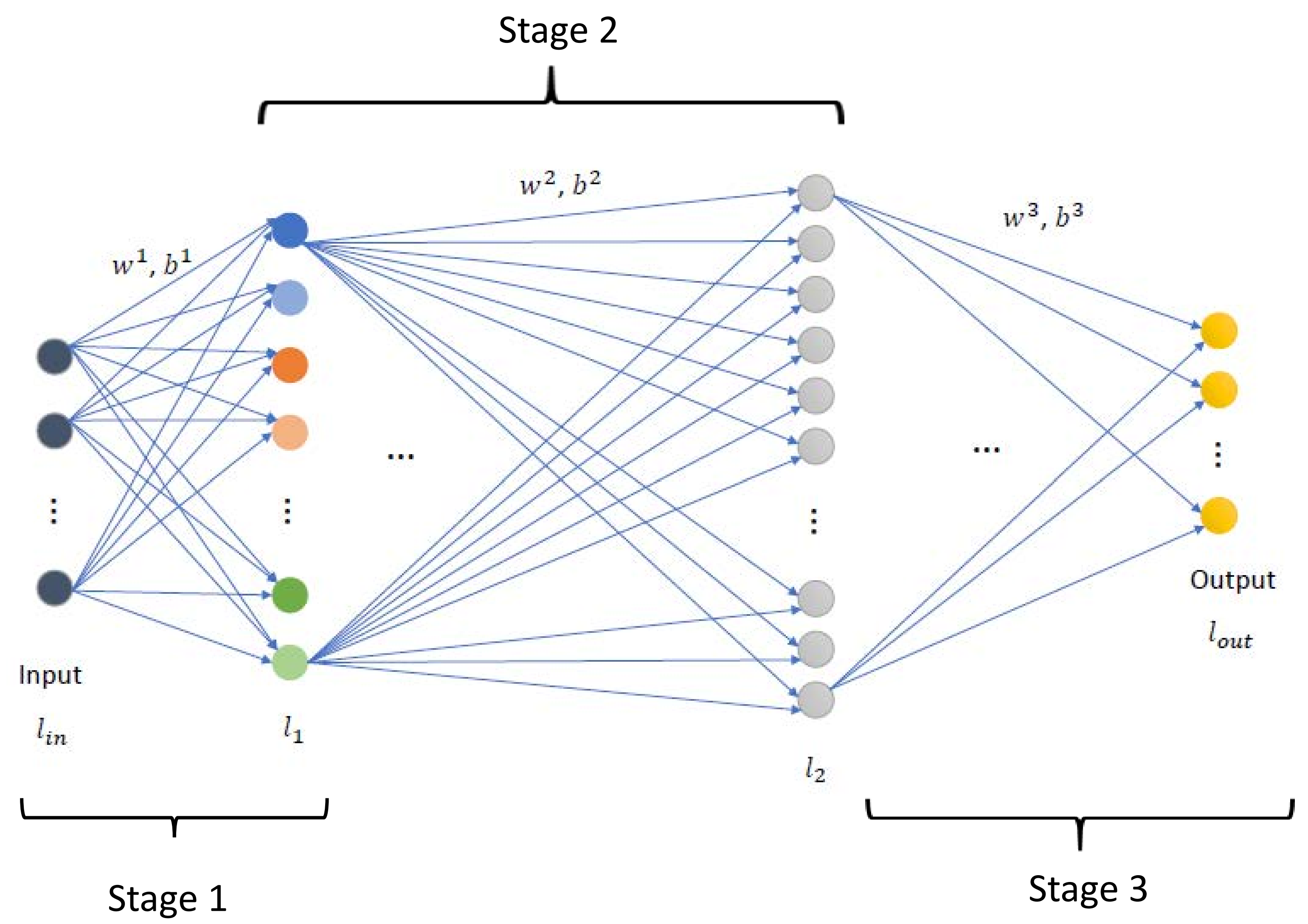
Weight of red link: 1
Weight of black link: -P

Stage 3 (from the 2nd Layer to Output Layer)



Weight of red link: 1
Weight of black link: 0

Proposed MLP Architecture (4 Layers, 3 Stages)



Generalization: Feedforward MLP (FF-MLP)

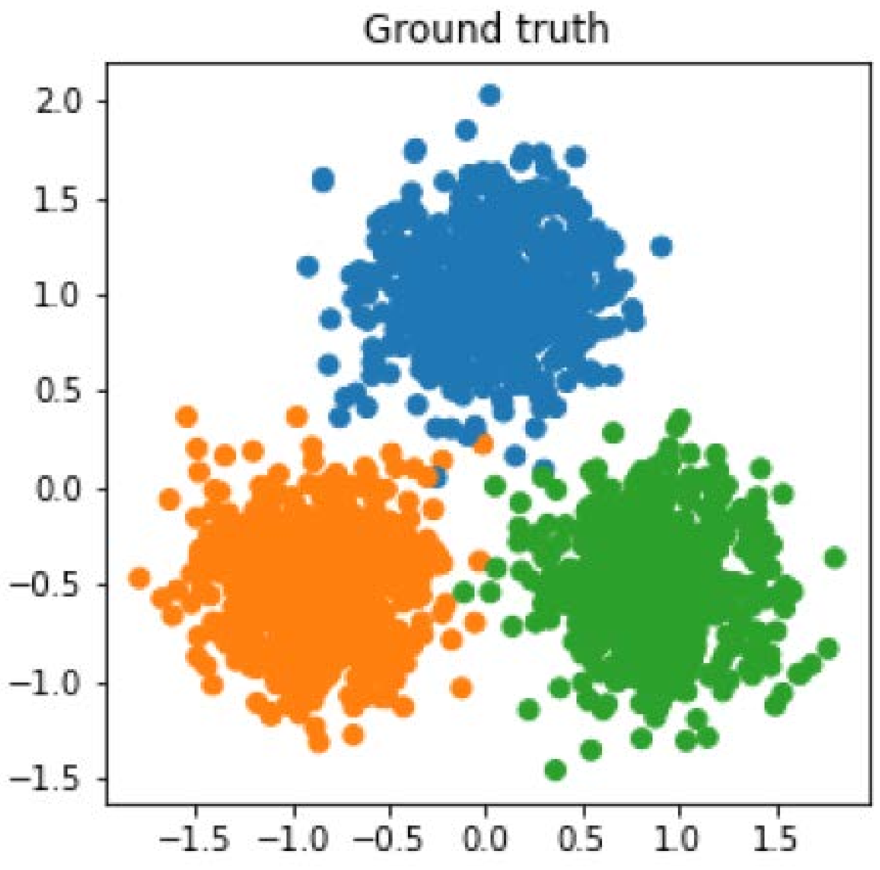
- Determine the network architecture and link weights in one-pass feedforward manner
  - Stage 1: Half-Space Partitioning
  - Stage 2: Subspace Isolation
  - Stage 3: Subspace-Class Connection
- Neuron Numbers

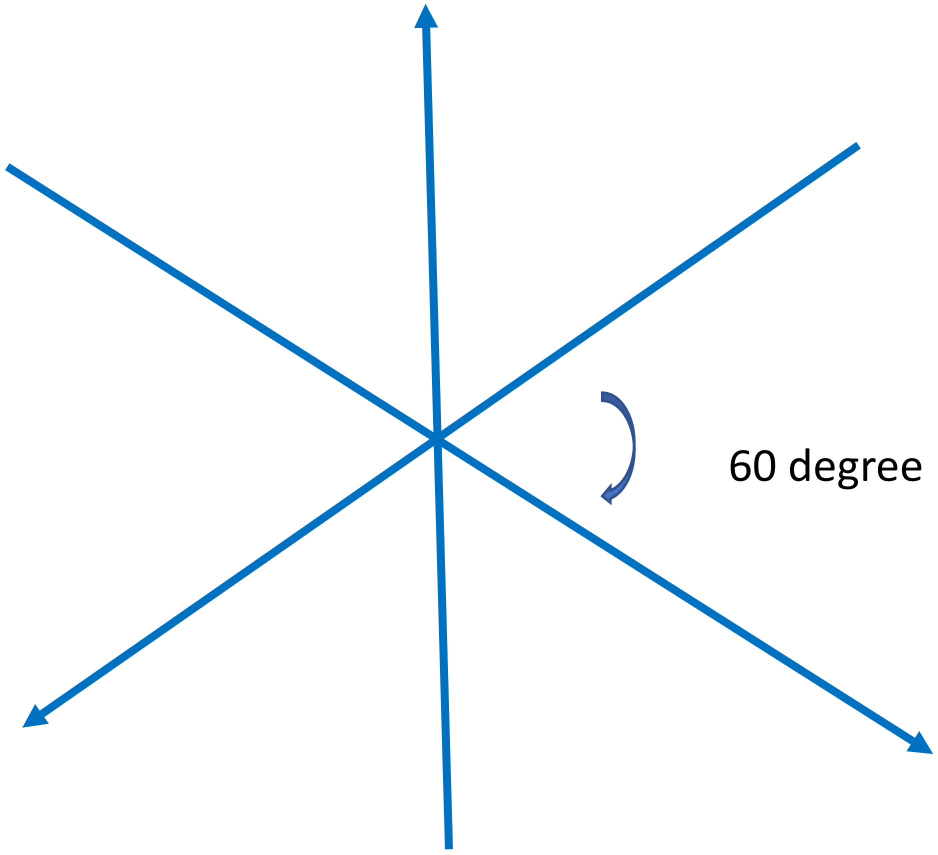
$$D_{in} = N, \quad D_{out} = C. \qquad D_1 \leq 2 \binom{G}{2}, \quad G \leq D_2 \leq 2^{G(G-1)/2}$$

- Link Weights

link weights in Stage 1 are determined by each individual 2-class LDA,
link weights in Stage 2 are either 1 or -P, and
link weights in Stage 3 are either 1 or 0.

Illustrative Example: 3-Gaussian Blobs (1)

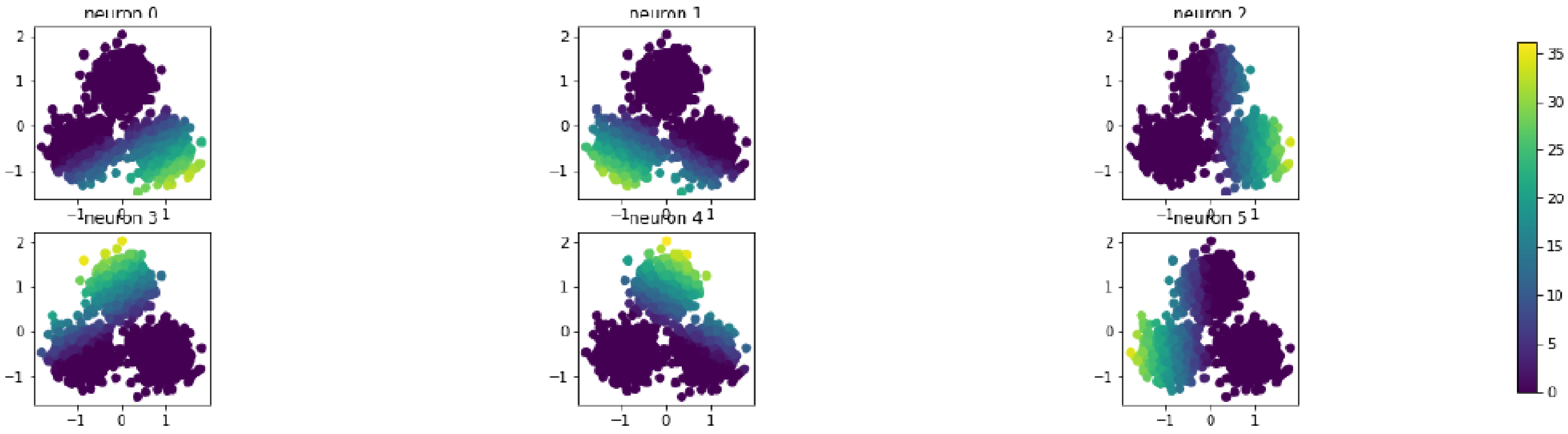




3 partitioning lines

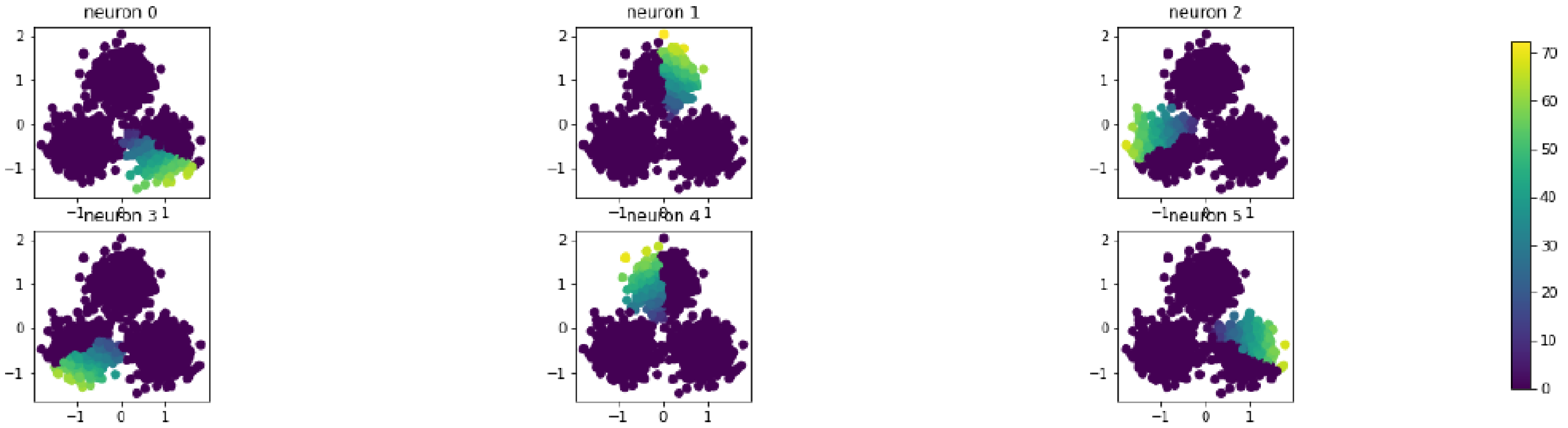
Illustrative Example: 3-Gaussian Blobs (2)

- Responses at the first layer

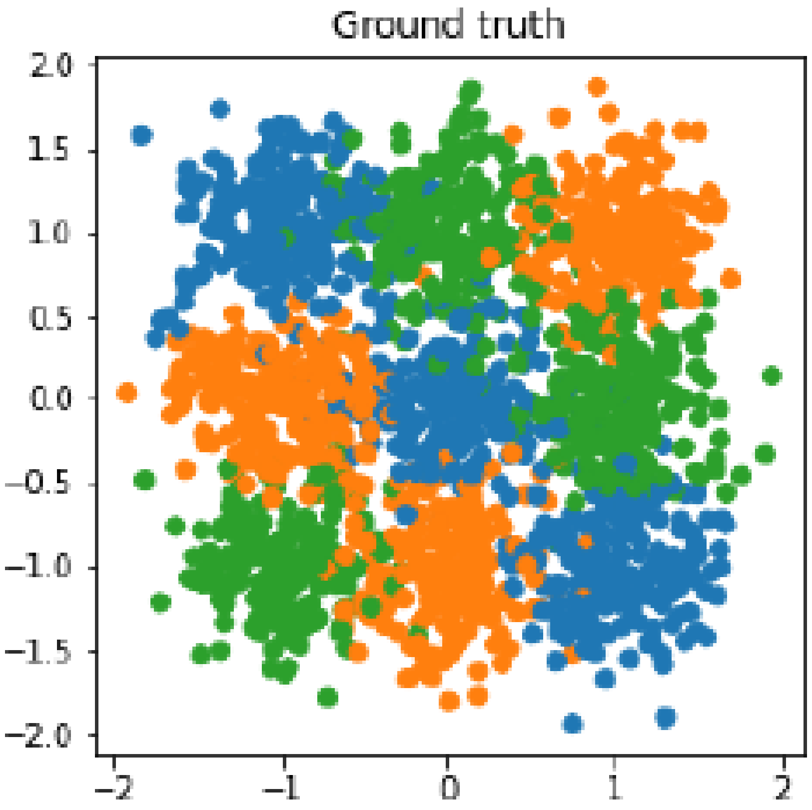


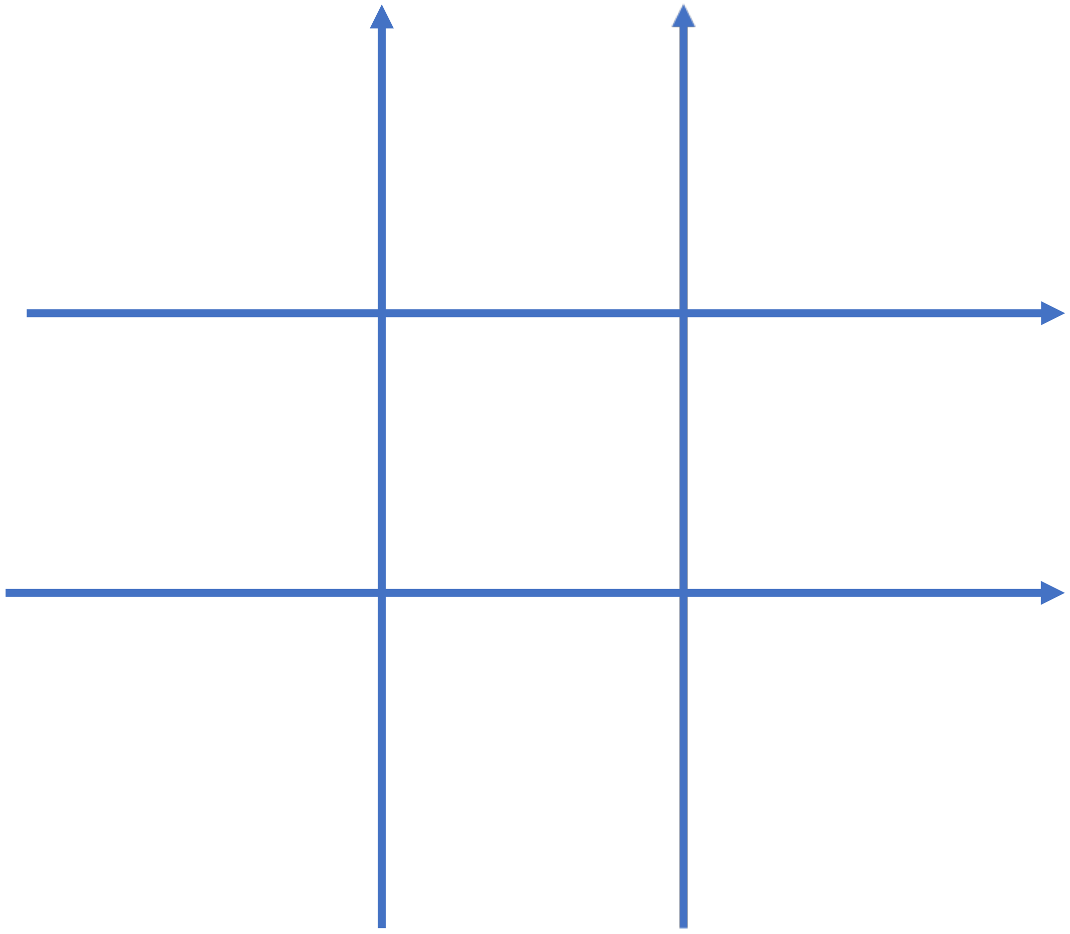
Illustrative Example: 3-Gaussian Blobs (3)

- Responses at the second layer



Illustrative Examples: 9-Gaussian Blobs (1)

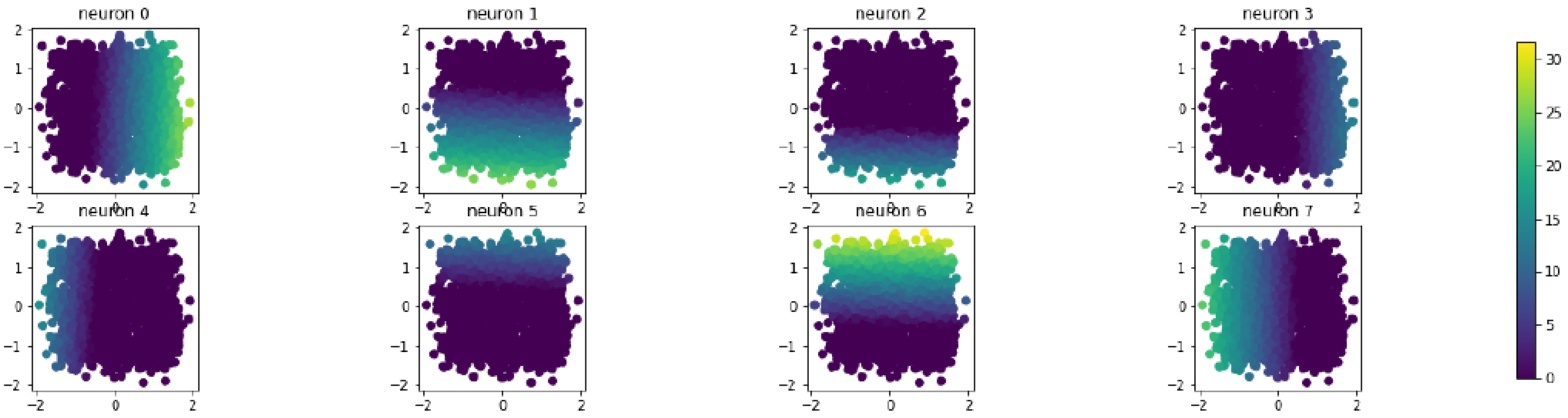




4 partitioning lines

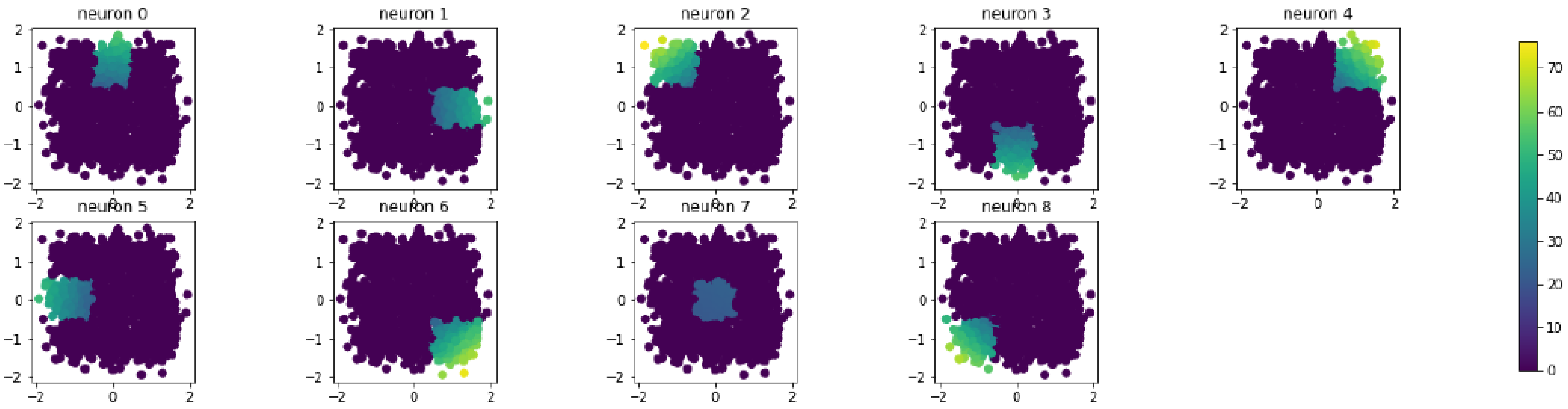
Illustrative Example: 9-Gaussian Blobs (2)

- Responses at the first layer



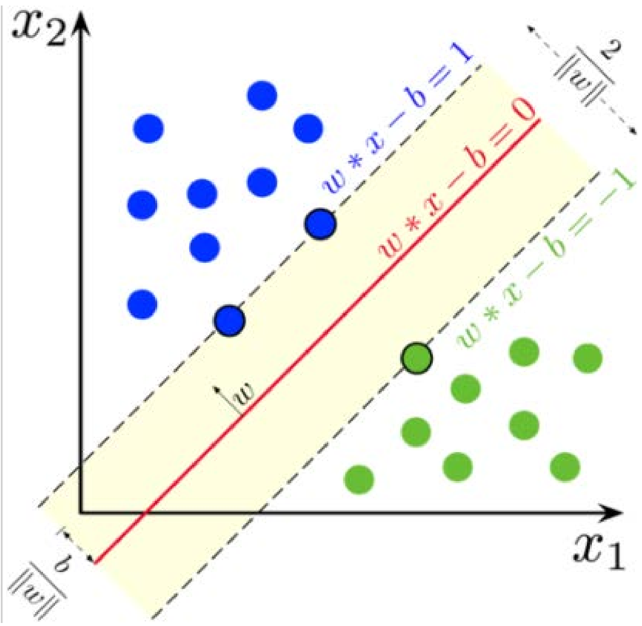
Illustrative Example: 9-Gaussian Blobs (3)

- Responses at the second layer

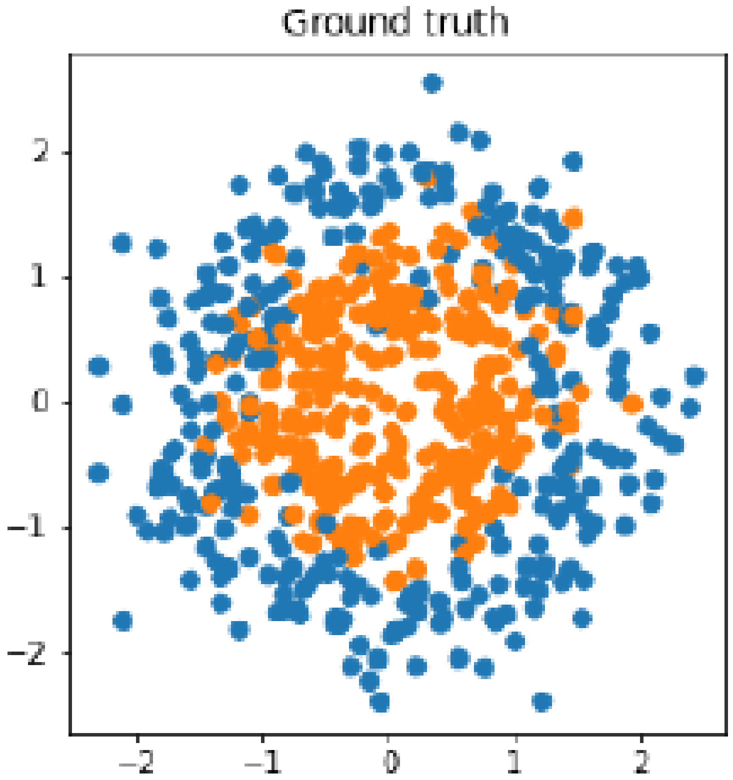


Comparison with SVM

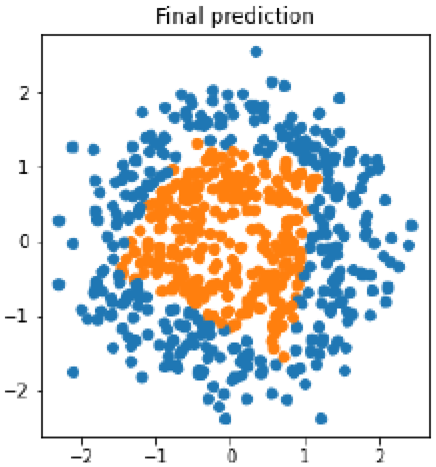
- SVM contains only one-stage (no ReLU is needed)
  - SVM partitions boundaries of classes by supporting vectors
  - It can handle non-convex shapes
- FF-MLP contains three stages (ReLU is needed)
  - FF-MLP partitions one full space into many half subspaces in Stage 1
  - FF-MLP isolate regions in Stage 2
  - FF-MLP connects regions to its class type in Stage 3
- SVM is slow for multi-class classification problem (a generalization of a two-class classifier)



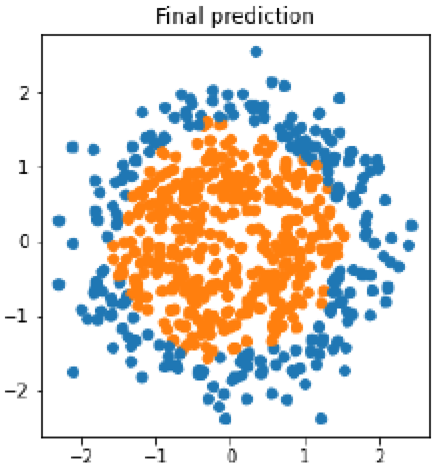
Illustrative Examples: Circle-and-Ring



Classification Results



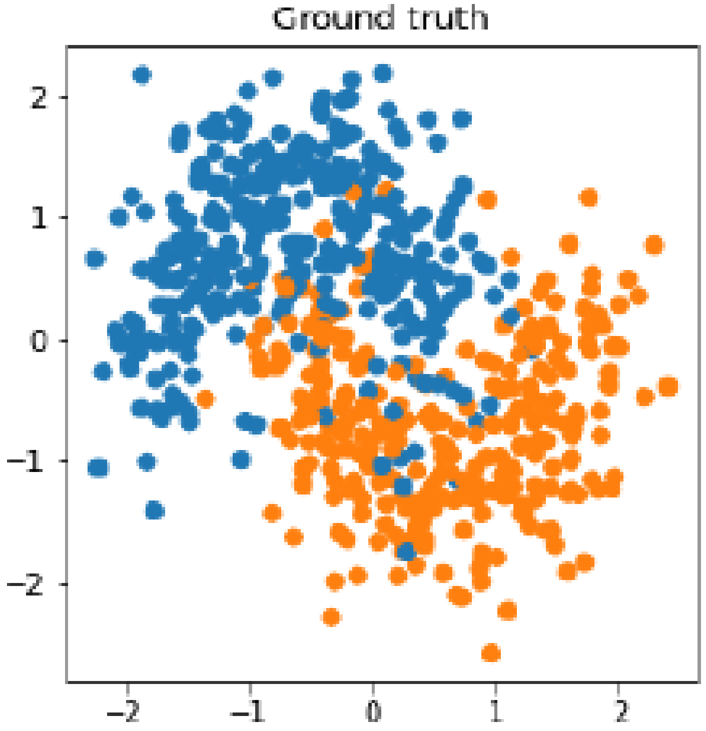
(a) 4 Components

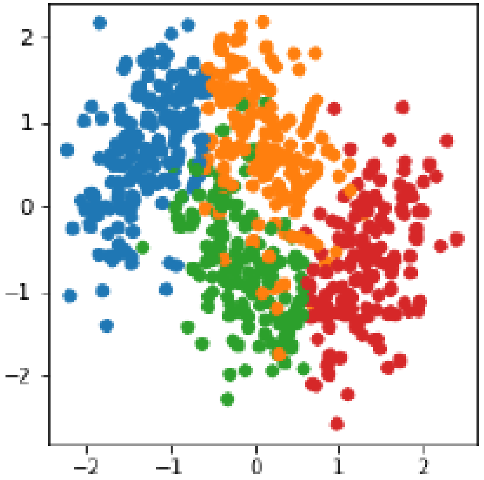


(b) 16 Components

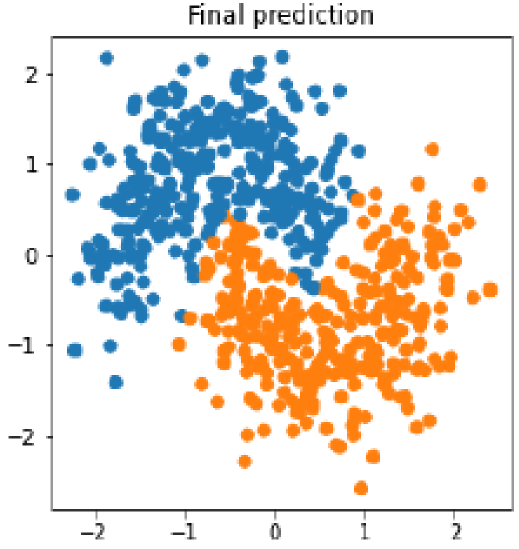
Approximation of the outer ring with 4 Gaussian and 16 Gaussian components

More Illustrative Examples: 2-New-Moons





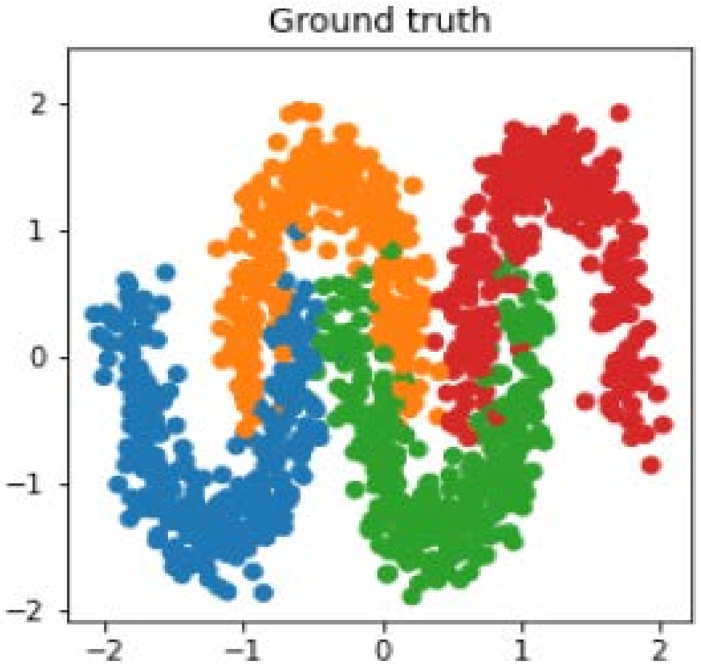
Each moon is approximated by 2 Gaussian components

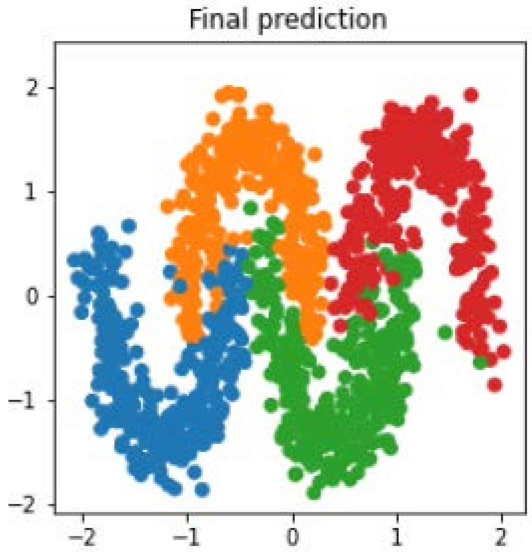


Final Classification result

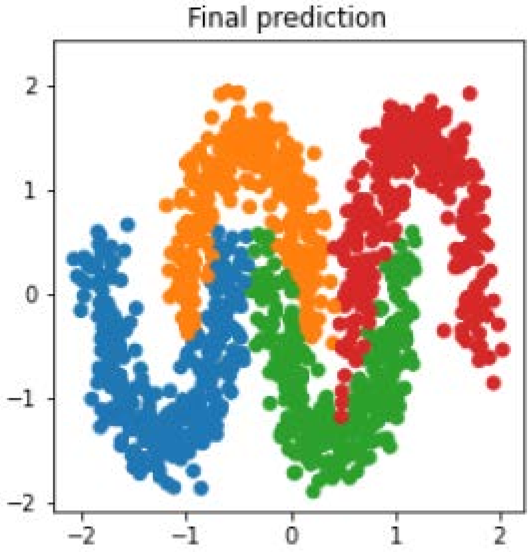
More Illustrative Examples: 4-New-Moons

Classification Results





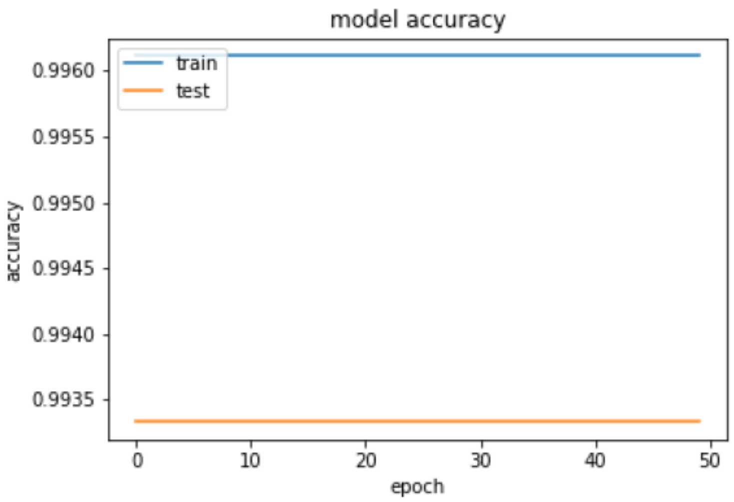
Each Moon Approximated by 2 Gaussian Components



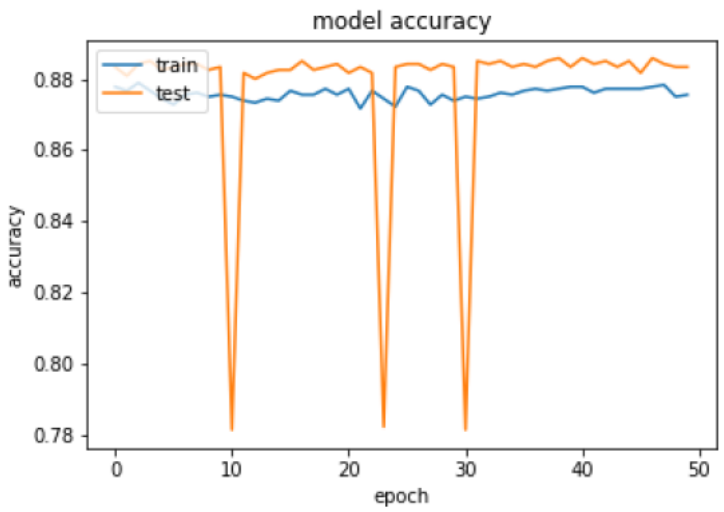
Each Moon Approximated by 3 Gaussian Components

Will BP Help Improve Performance?

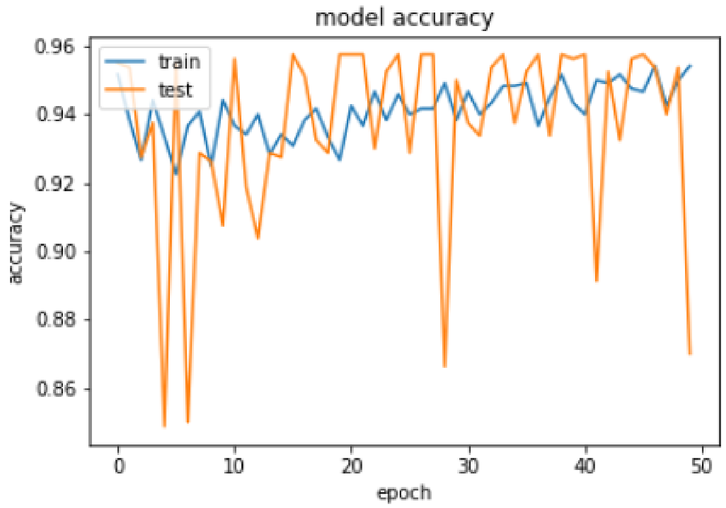
- With FF-MLP as the MLP architecture and initialization, we perform BP



(a) 3-Gaussian-blobs



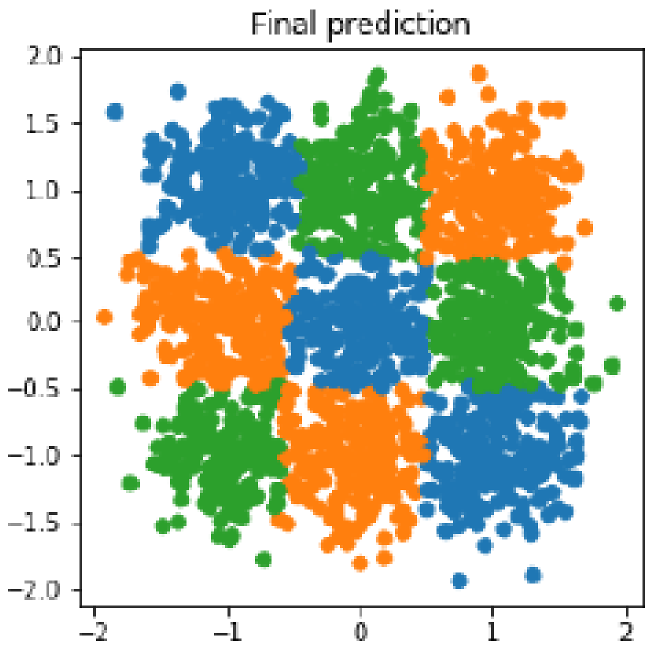
(b) 9-Gaussian-blobs



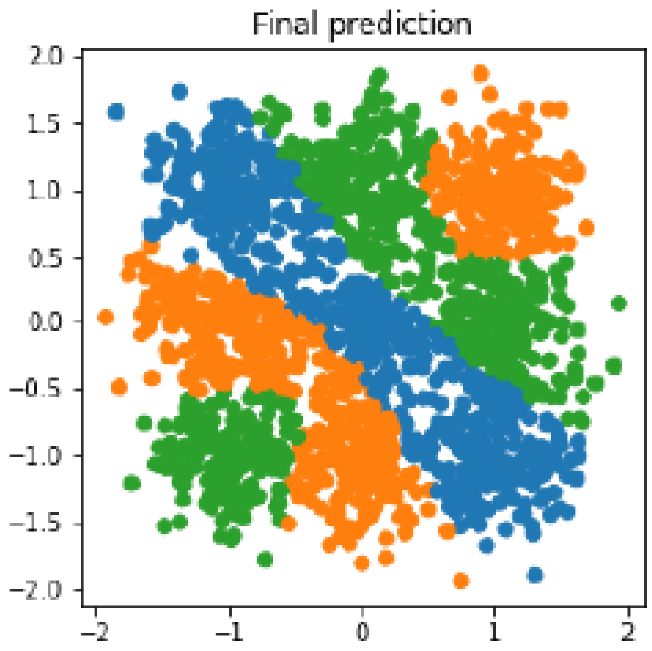
(c) 4-new-moons

Comparison of FF-MLP and Random Initializations

9-Gaussian-Blobs



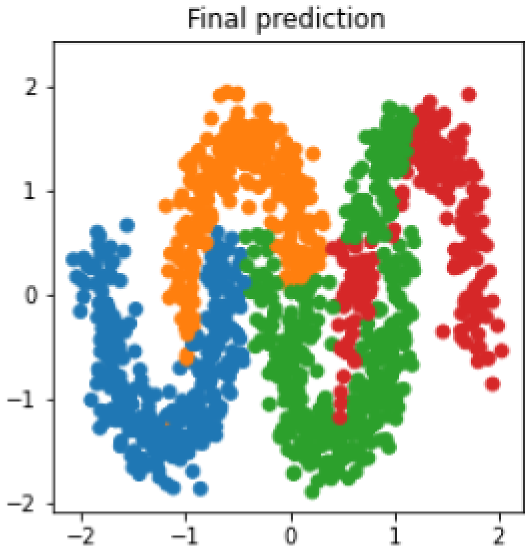
(a) FF-MLP initialization



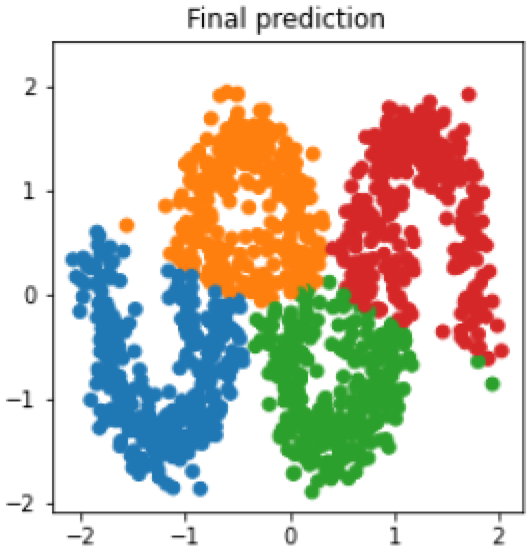
(b) random initialization

Comparison of FF-MLP and Random Initializations

4-New-Moons



(a) FF-MLP initialization



(b) random initialization

Comparison of Classification Accuracy

Dataset	FF-MLP		BP-MLP with FF-MLP init.(50)		BP-MLP with random init. (50)		BP-MLP with random init. (15)	
	train	test	train	test	train	test	train	test
2 Gaussian blobs	**100.00**	**100.00**	**100.00**	**100.00**	99.97 ± 0.03	99.90 ± 0.04	99.91 ± 0.05	99.88 ± 0.10
XOR	**100.00**	**99.83**	**100.00**	**99.83**	99.83 ± 0.16	99.42 ± 0.24	93.20 ± 11.05	92.90 ± 11.06
3-Gaussian-blobs	99.67	99.33	99.67	99.33	**99.68** ± 0.06	**99.38** ± 0.05	99.48 ± 0.30	99.17 ± 0.48
9-Gaussian-blobs (0.1)	**89.11**	**88.58**	70.89	71.08	84.68 ± 0.19	85.75 ± 0.24	78.71 ± 2.46	78.33 ± 3.14
9-Gaussian-blobs (0.3)	**88.11**	**88.83**	88.06	88.58	81.62 ± 6.14	81.35 ± 7.29	61.71 ± 9.40	61.12 ± 8.87
circle-and-ring (4)	88.83	**87.25**	**89.00**	86.50	81.93 ± 7.22	82.80 ± 5.27	70.57 ± 13.42	71.25 ± 11.27
circle-and-ring (16)	83.17	80.50	85.67	**88.00**	**86.20** ± 1.41	85.05 ± 1.85	66.20 ± 9.33	65.30 ± 11.05
2-new-moons	**88.17**	**91.25**	**88.17**	**91.25**	83.97 ± 1.24	87.60 ± 0.52	82.10 ± 1.15	86.60 ± 0.58
4-new-moons (2)	**94.33**	**92.62**	84.75	80.87	86.73 ± 0.11	83.92 ± 0.34	86.00 ± 0.23	83.17 ± 0.44
4-new-moons (3)	**95.75**	**95.38**	87.50	87.00	86.90 ± 0.25	84.00 ± 0.33	85.00 ± 0.98	82.37 ± 0.76

TABLE I
COMPARISON OF TRAINING AND TESTING CLASSIFICATION PERFORMANCE BETWEEN FF-MLP, BP-MLP WITH FF-MLP INITIALIZATION AND BP-MLP WITH RANDOM INITIALIZATION. THE BEST MEAN TRAINING AND TESTING ACCURACY ARE HIGHLIGHTED IN BOLD.

Higher Dimension Datasets

- Iris dataset: 3 classes, 4 dimensions, 150 samples per class
- Wine dataset: 3 classes, 13 dimensions, 59, 71 and 48 samples in each class
- Breast Cancer Wisconsin (BCW) dataset: 2 classes, 30 dimensions, 569 samples in total
- Pima Indians diabetes dataset: 2 classes, 8 dimensions, 768 samples

Classification Performance

Dataset	D_{in}	D_{out}	D_1	D_2	Accuracy					
					FF-MLP		BP-MLP/random init. (50)		BP-MLP/random init. (15)	
					train	test	train	test	train	test
Iris	4	3	4	3	**96.67**	**98.33**	65.33 ± 23.82	64.67 ± 27.09	47.11 ± 27.08	48.33 ± 29.98
Wine	13	3	6	6	**97.17**	**94.44**	85.66 ± 4.08	79.72 ± 9.45	64.34 ± 7.29	61.39 ± 8.53
B.C.W	30	2	2	2	**96.77**	94.30	95.89 ± 0.85	**97.02** ± 0.57	89.79 ± 2.41	91.49 ± 1.19
Pima	8	2	18	88	**91.06**	73.89	80.34 ± 1.74	**75.54** ± 0.73	77.02 ± 2.89	73.76 ± 1.45

TABLE II
TRAINING AND TESTING ACCURACY RESULTS OF FF-MLP AND BP-MLP WITH RANDOM INITILIALZATION FOR FOUR HIGHER-DIMENSIONAL DATASETS. THE BEST MEAN TRAINING AND TESTING ACCURACY ARE HIGHLIGHTED IN BOLD.

Time Complexity

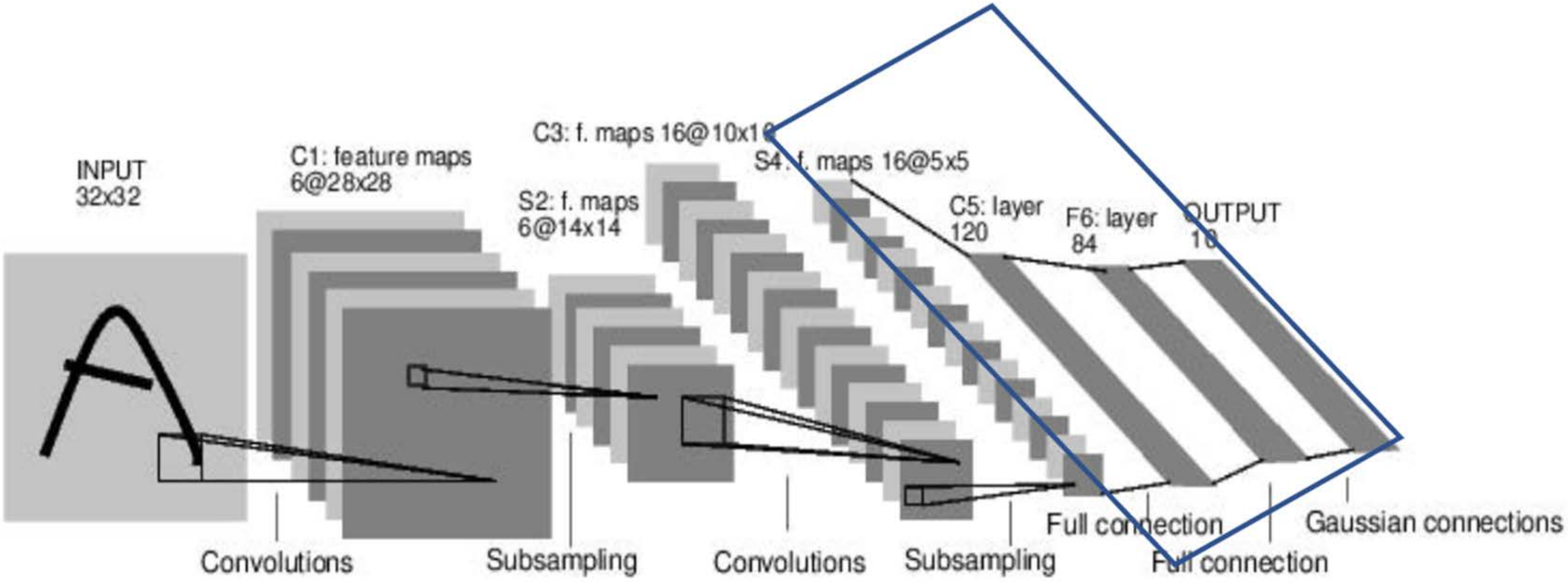
Dataset	GMM	Boundary construction	Region representation	Classes assignment	Total	BP (15)	BP (50)
2 Gaussian blobs	0.00000	0.00385	0.00112	0.00009	**0.00506**	2.77509 ± 0.18903	8.02358 ± 0.07385
XOR	0.00000	0.01756	0.00093	0.00007	**0.01855**	2.88595 ± 0.06279	8.50156 ± 0.14128
3-Gaussian-blobs	0.00000	0.01119	0.00126	0.00008	**0.01253**	2.78903 ± 0.07796	8.26536 ± 0.17778
9-Gaussian-blobs (0.1)	0.00000	0.22982	0.00698	0.00066	**0.23746**	2.77764 ± 0.14215	8.34885 ± 0.28903
9-Gaussian-blobs (0.3)	0.00000	2.11159	0.00156	0.00010	**2.11325**	2.79140 ± 0.06179	8.51242 ± 0.24676
circle-and-ring (4)	0.02012	0.01202	0.00056	0.00006	**0.03277**	1.50861 ± 0.14825	3.79068 ± 0.28088
circle-and-ring (16)	0.04232	0.05182	0.00205	0.00020	**0.09640**	1.43951 ± 0.15573	3.80061 ± 0.13775
2-new-moons	0.01835	0.01111	0.00053	0.00006	**0.03006**	1.44454 ± 0.06723	3.64791 ± 0.08565
4-new-moons (2)	0.04541	0.14471	0.00461	0.00054	**0.19527**	2.03826 ± 0.12244	5.62977 ± 0.05140
4-new-moons (3)	0.03712	11.17161	0.00206	0.00021	11.21100	**1.98338** ± 0.04357	5.71387 ± 0.14150
Iris	0.02112	0.02632	0.00011	0.00002	**0.04757**	0.73724 ± 0.01419	1.60543 ± 0.14658
Wine	0.01238	0.03551	0.00015	0.00003	**0.04807**	0.81173 ± 0.01280	1.72276 ± 0.07268
B.C.W	0.01701	0.03375	0.00026	0.00003	**0.05106**	1.08800 ± 0.05579	2.73232 ± 0.12023
Pima	0.03365	0.16127	0.00074	0.00039	**0.19604**	0.96707 ± 0.03306	2.32731 ± 0.10882

TABLE III

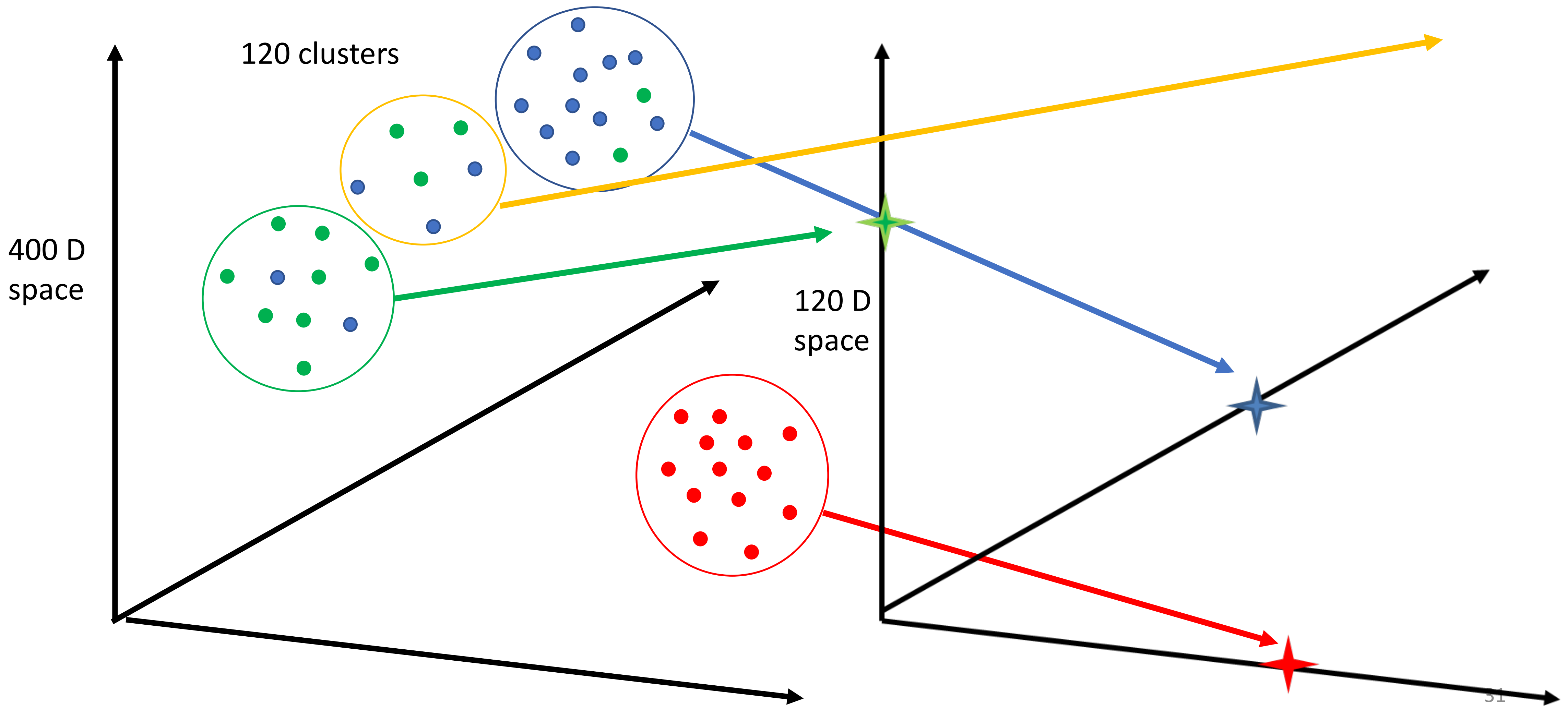
COMPARISON OF COMPUTATION TIME IN SECONDS OF FF-MLP (LEFT) AND BP-MLP (RIGHT) WITH 15 AND 50 EPOCHS. THE MEAN AND STANDARD DEVIATION OF COMPUTATION TIME IN 5 RUNS ARE REPORTED FOR BP-MLP. THE SHORTEST RUNNING TIME IS HIGHLIGHTED IN BOLD.

Relationship with Feedforward CNNs

- Feedforward CNN
  - Convolutional layers: spatial-spectral transform (e.g. Saab transform and Saak transform)
  - Fully connected layers: linear least-squared-regression



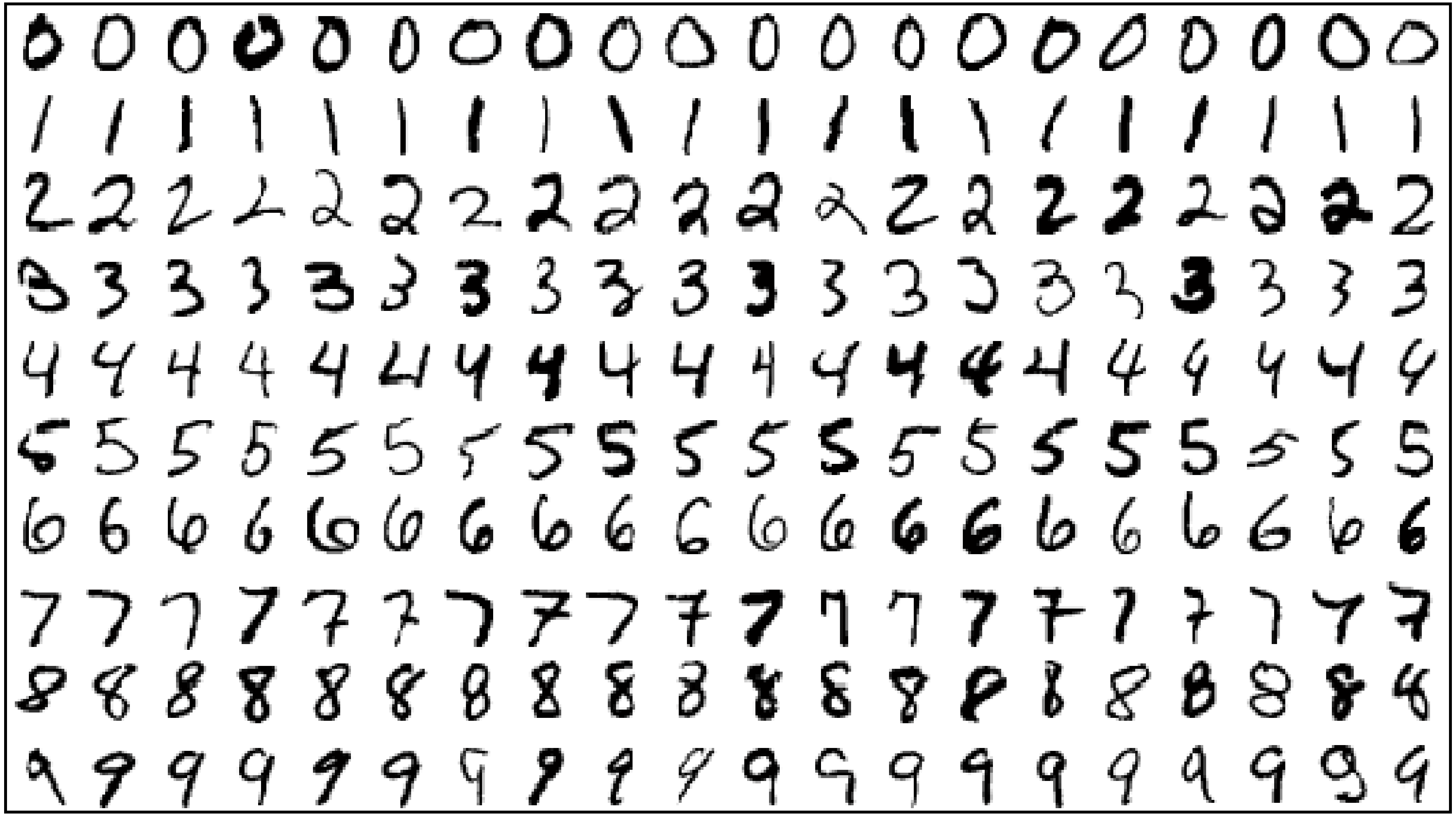
Linear Least-Squared-Regression



Why Pseudo-Labels?

To address intra-class variability

MNIST Dataset



Conclusion

- A new interpretation of MLP
  - Generalization of two-class LDA
  - MLP and SVM are quite close to each other
    - Differences lies in the order of “class separation” or “Gaussian blobs separation”
  - FF-MLP is easy to design with excellent performance
- Do we really need BP-MLP?
  - How to justify end-to-end optimization of neural networks in general?

Reference

- Ruiyuan Lin, Zhiruo Zhou, Suya You, Raghuveer Rao and C.-C. Jay Kuo, “From two-class linear discriminant analysis to interpretable multilayer perceptron design,” arXiv preprint arXiv:2009.04442.