DD2437-ANN Lab2 Presentation

Group 16, Feb 11

Ming Jiang | mingj@kth.se

Wang ZiYuan | wangz@kth.se

Table of Contents

01

Function Approximation

Function Approximation(data without noise)

n_nodes in [2, 10] *

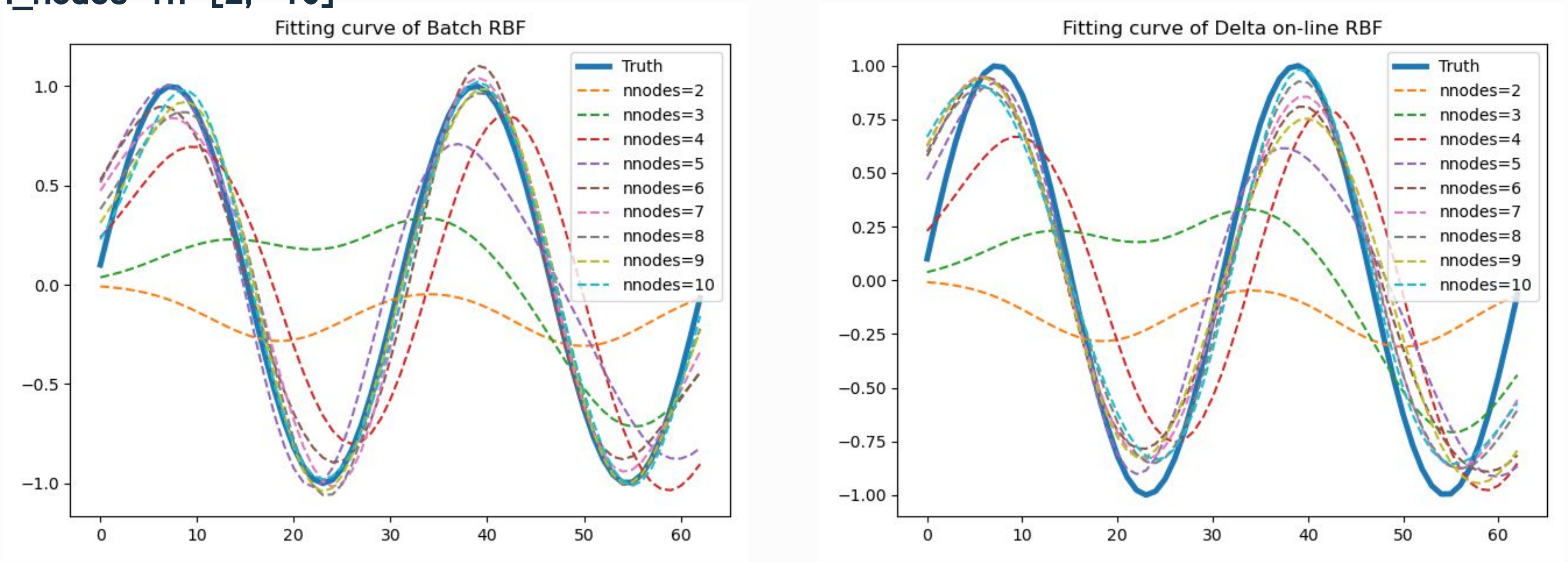


Fig. Fitting Curve with Different Hidden Node Numbers(width = 2 *π / n_nodes})

- The curve tends to fit better with the increase of hidden nodes number.

MAE Thresholds

Table. MAE Thresholds and Hidden Nodes Number

Threshold	Absolute Residual Error	Hidden Nodes
0.1	0.0824	9
0.01	0.0078	17
0.001	0.0008	34

- When the hidden nodes beyond 34, the absolute residual error is less than 0.001, which means that the nodes number is enough to fit with the dataset.

*sin(2x) function, batch mode.

Output Transformation for 0 Error

run in batch mode,

n_nodes=10

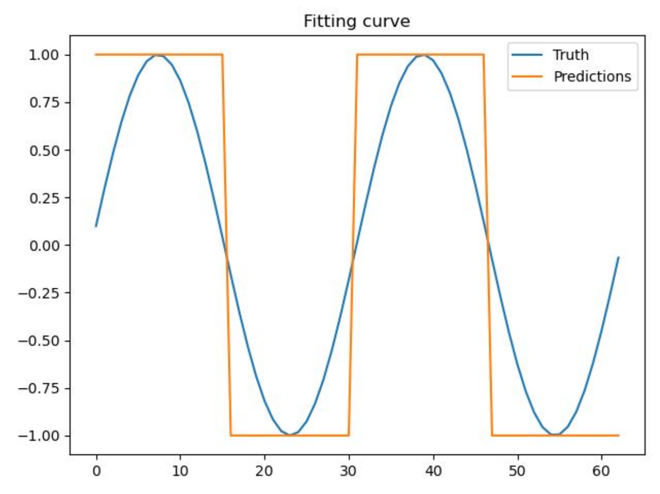


Fig. *square(2x) Function Approximation*

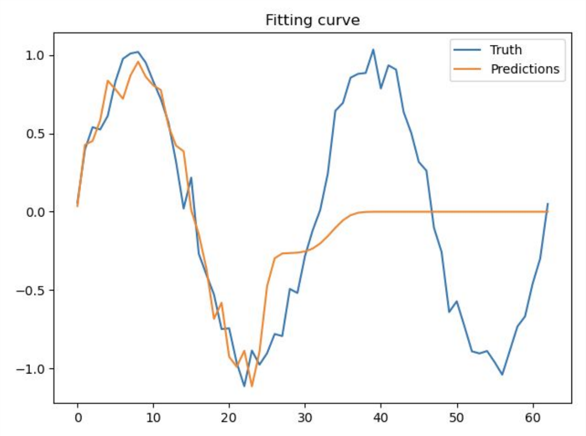
- For the square(2x) function, we can set the output threshold to reduce the residual error to 0. And this kind of output from RBF is actually a **classification** problem.

02

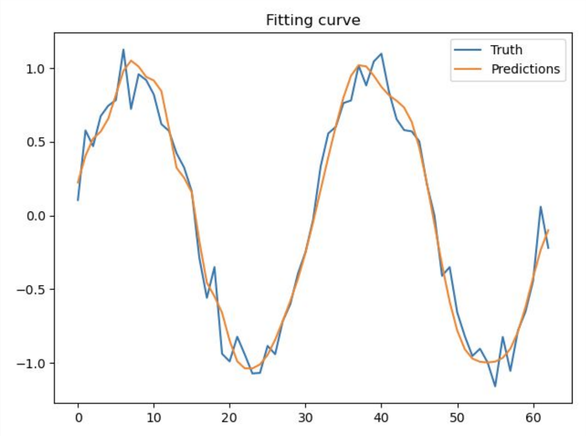
Competitive Learning

Competitive Learning

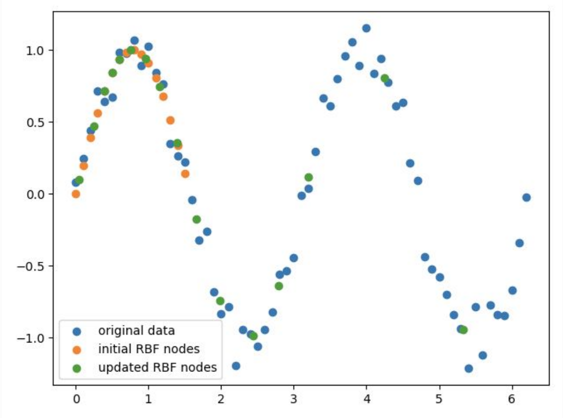
- Initialize the RBF nodes randomly, then error is 0.0979, while using competitive learning the error is reduced to 0.0910*. Competitive learning helps in correcting the RBF nodes.
- Initialization of nodes is very important for RBF network.
- Competitive learning helps RBF nodes distributed more evenly.



iteration = 100, eta = 0.01



iteration = 10000, eta = 0.1



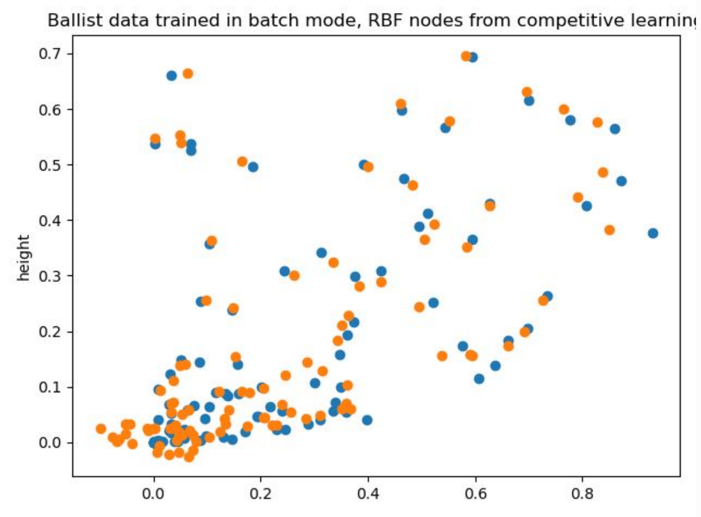
RBF nodes update, iteration = 10000, eta = 0.1

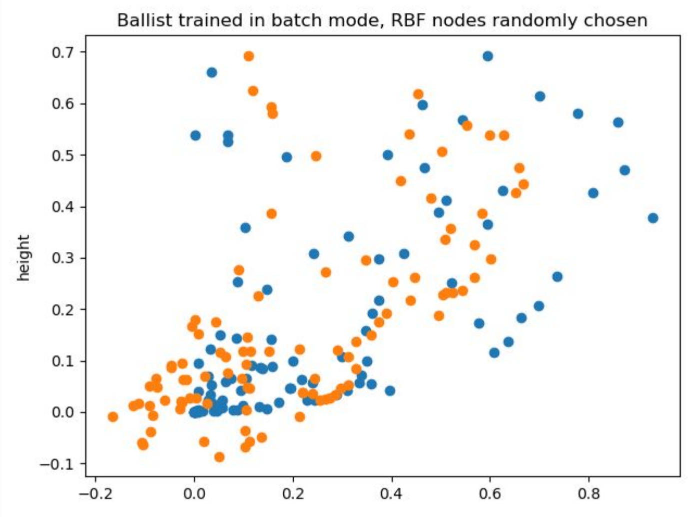
Fig. Effect of competitive learning on RBF nodes

*batch mode, sin(2x) function, sigma = 1.0, n_nodes = 16

Competitive Learning - 2D Data

- The performance of RBF network relies much on the initialization of RBF nodes. If we randomly choose them, then the final results will vary with the selection.





error = 0.0334 for the *distance* label, 0.0232 for the *height* label

error = 0.0628 for the *distance* label, 0.0378 for the *height* label

Fig. Predictions on the 2D Ballist Data

*sigma = 1.0, n_nodes = 12

03

RBF Regression: Noisy Data

Data Comparison

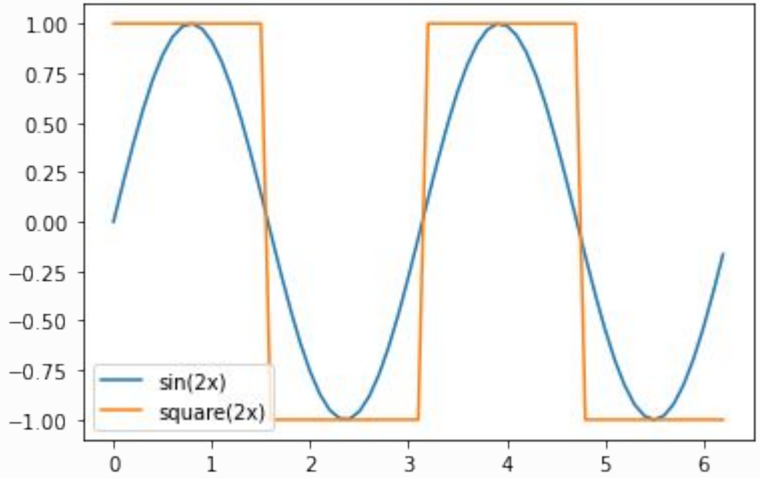


Fig. Original Clean Data

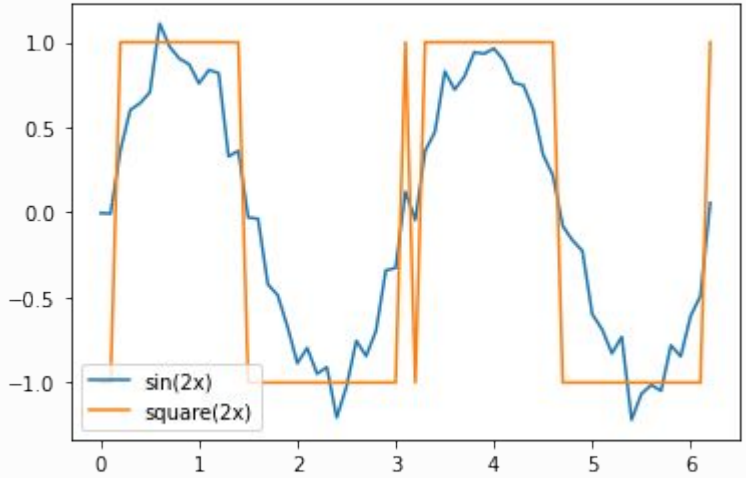


Fig. Noisy Data

- Due to the shape and the value of given data, we decide that MAE captures the accuracy much better than MSE.

*sigma = 0.1

Importance of Units Parameter

- Overall, the best set up is Units=20, Sigma=0.2
- Sigma and the amount of Units defined the localizing capacity of given network
- We evenly distributed nodes, the best set could offer 0.0952 error rate, instead of random, 0.9524

Table 2: Grid Search for $sin(2x)$ and $square(2x)$ Approximation

Functions:		$sin(2x)$				$square(2x)$			
		Clean Data		Noisy Data		Clean Data		Noisy Data	
Hidden Nodes	Sigma	Batch	Online	Batch	Online	Batch	Online	Batch	Online
8	0.2	0.1019	0.1053	0.1456	0.1445	0.0	0.0	0.0317	0.0317
8	0.6	0.0862	0.116	0.1237	0.1255	0.0635	0.0952	0.0635	0.0952
8	1.0	0.0722	0.1951	0.1234	0.1757	0.0317	0.0952	0.0317	0.0635
16	0.2	0.0239	0.0898	0.1028	0.1383	0.0952	0.0635	0.0635	0.0317
16	0.6	0.0064	0.1026	0.1013	0.1153	0.0635	0.0317	0.0317	0.0317
16	1.0	0.0032	0.15	0.0979	0.2068	0.031	0.127	0.0635	0.0317
20	0.2	0.019	0.0625	0.1046	0.1364	0.0635	0.0635	0.0317	0.0
20	0.6	0.0038	0.1173	0.1002	0.1206	0.0952	0.0	0.0317	0.0317
20	1.0	0.0029	0.1242	0.0994	0.145	0.0952	0.0317	0.0	0.0317

*epoch = 1000, eta=0.01

Importance of Eta

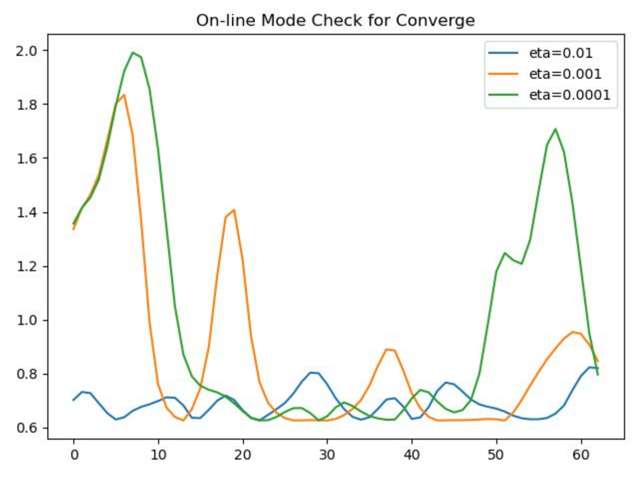


Fig. Online Training

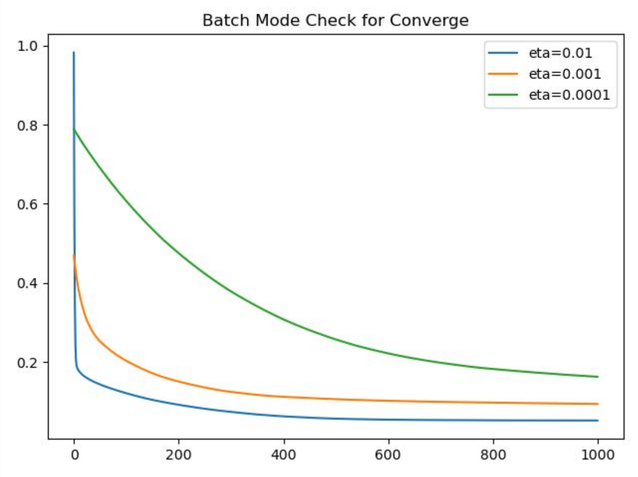


Fig. Batch Training

- With the trend of eta increase, we can see that the speed of converge as well as the robustness of given model is better, we could assume that the model is not trapped in local minimum.
- The error, against the trend of eta, increase from 0.0914 to 0.1799.

*epochs = 1000, n_nodes=20, sigma=0.2

Comparison with Perceptron

- We can see that regardless of the parameter, in this case, RBF network outperform the Perceptron.
- With the method of projecting the input space into higher dimensions, it allowed the model to capture fluctuations vastly better.

Table 3: Compassion between RBF and Perceptron

Function	Hidden Nodes	RBF MSE	Perceptron MSE
$sin(2x)$	8	0.0509	0.4428
$sin(2x)$	12	0.0338	0.4362
$sin(2x)$	20	0.0277	0.4389
$square(2x)$	8	0.381	1.3968
$square(2x)$	12	0.3175	1.3333
$square(2x)$	20	0.127	1.3333

*epochs = 1000, learning rate=0.001

04

SOM Network

4.1 The Clustering of Given Animals

Table 4: Re-ordered Animal List

Animal	**spider**	housefly	moskito	butterfly	dragonfly
Animal	grasshoppe	beetle	**pelican**	duck	penguin
Animal	ostrich	**frog**	**seaturtle**	crocodile	**horse**
Animal	rabbit	elephant	bat	rat	skunk
Animal	hyena	bear	dog	walrus	lion
Animal	cat	ape			

- We can see that the ordering is given by: the first cluster is Insect, followed by Birds, merching into Amphibians, Reptiles and Mammals.
- Initialization do have an impact on the final ordering, not the clusters

*radius=50, eta=0.2, epochs=20

4.2 Tour Trajectory

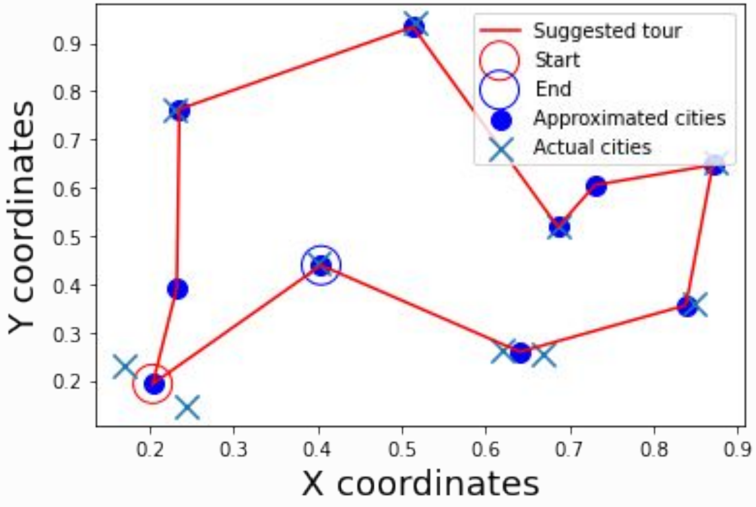


Fig. *Good Initialization*

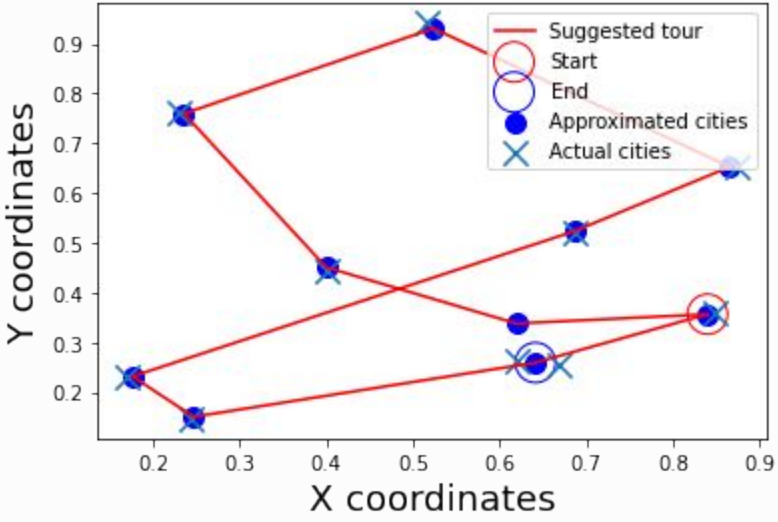


Fig. *Bad Initialization*

- We can see the impact of initialization based on different setting.
- With the close input nodes, we can see the limitation of given model.

---

*epochs = 300, learning_rate = 0.01

4.3 Vote Clustering

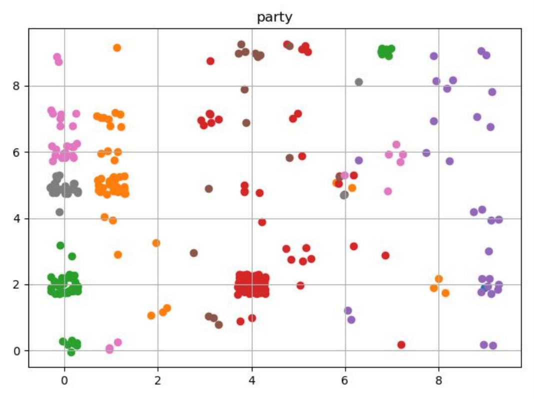


Fig. Clustering on Party

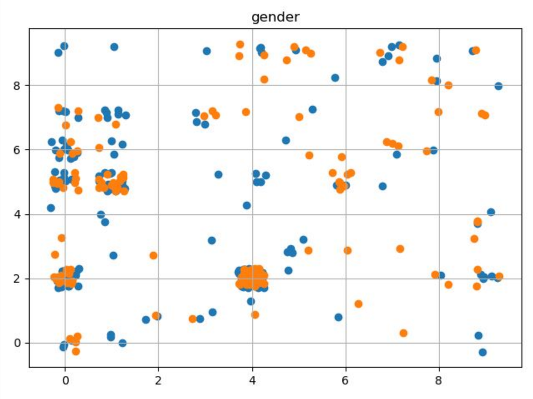


Fig. Clustering on Gender

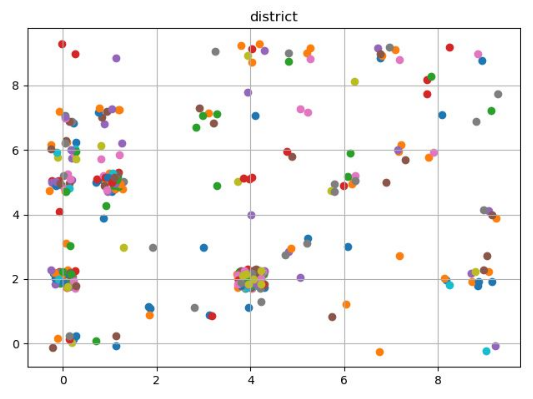


Fig. Clustering on District

- We can only observe certain pattern in the clustering on Party
- The conclusion do fit the reality.

*epochs = 20

Q & A

Thank you!