ML Project Slides

Michele Di Niccola, Valerio Russo
Risi Scotti
MD in Digital Humanities, MD in Artificial Intelligence
m.diniccola@studenti.unipi.it, v.russo22@studenti.unipi.it
28/01/2025

Project B

Introduction and objectives

This project is focused around developing an Artificial Neural Network, RBF Network and a Ridge Regression model, each one able to solve both classification and regression problems. The aim of the project was to pursue a rigorous method to select the best models and validate their results. To find the best configuration, we adopted a random search with 5-fold cross validation. Then the models were tested on an internal test set, providing an unbiased estimate of the performance of the models. The correctness of the implementation was tested on the well-known “Monk’s problem”. Subsequently, a selected neural network was deployed and used for predicting the outputs of the ML-CUP24-TS dataset. The results of the model selection phase and the experiments are described in the following sections.

Method

Libraries: Keras, sklearn on Python

Implemented models:

Neural Network
- ReLu, Sigmoid(for classification)
- Adam
- Minibatch
- L2 regularization
- Early stopping

RBF Network
- RBF, Sigmoid (for classification)
- Adam
- Minibatch
- Early stopping

Ridge Regression
- Adam
- Minibatch
- L2 regularization
- Early stopping

Monk Results

The inputs were transformed with a one-hot-encoding on all three Monks datasets.

The NN's topology used for MONKs consisted of 1 hidden layer with 4 neurons, with ReLu for the hidden layer and Sigmoid for the output layer, while the Ridge was a simple linear model with L2 penalty. RBF Network results are in the appendix.

NN HPs

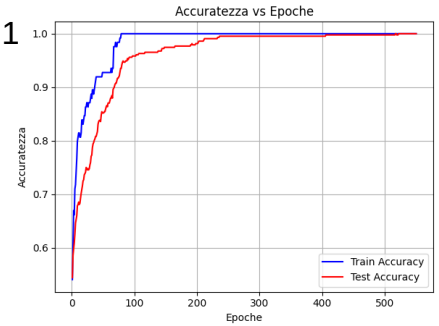
Task	lambda	dropout	eta	batch_size	accuracy/val	mse/val
Monk-1	1.10E-05	0.0	0.0066	16	100%/100%	1.25E-05/0.0009
Monk-2	0.0028	0.1	0.0068	16	97.6%/97.2%	0.042/0.042
Monk-3	3,26E-06	0.0	0.0042	16	95.1%/94.6%	0.027/0.036

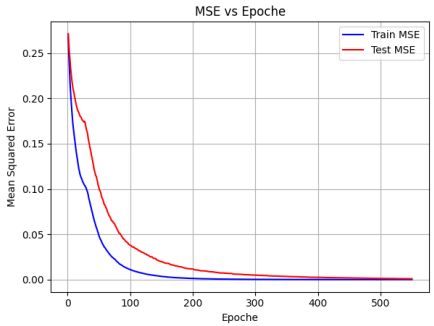
Ridge HPs

Task	lambda	eta	batch_size	accuracy/val accuracy	mse/val mse
Monk-1	0.00044	0.01	96	78.2%/72..4%	0.157/0.185
Monk-2	0.31	0.001	48	62.1%/66.8%	0.227/0.221
Monk-3	0.042	0.029	128	93.4%/97.2%	0.080/0.065

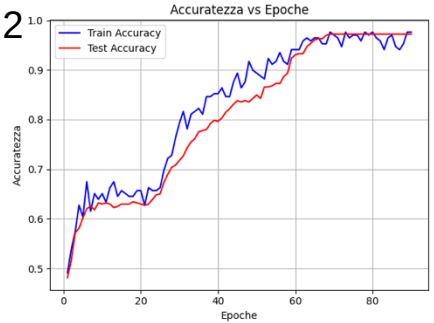
Monk Results

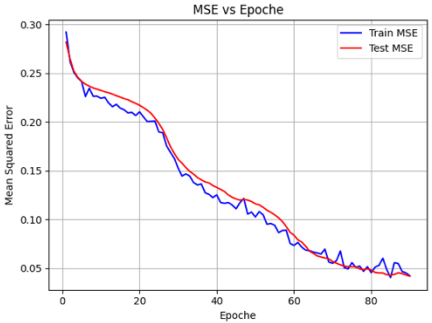
1



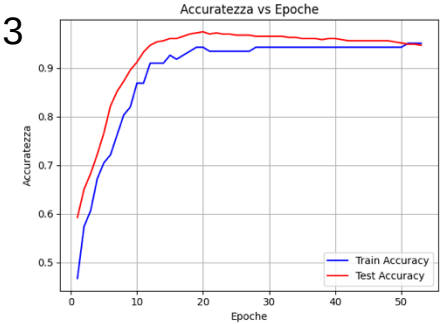


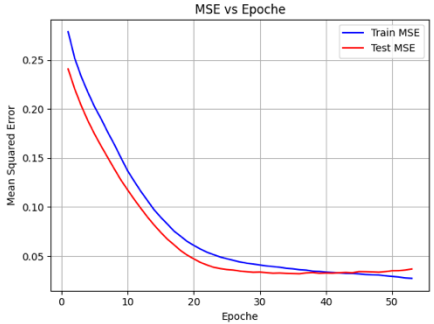
2





3

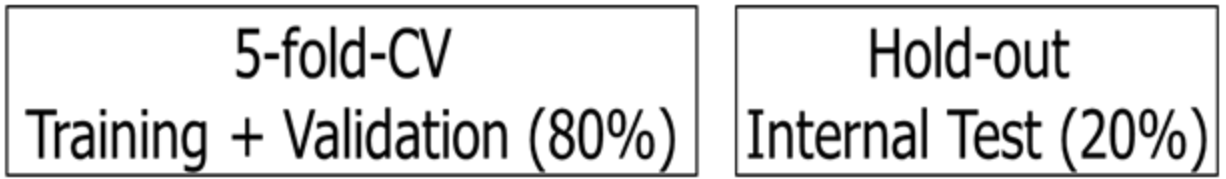




Monk 1, 2, 3 graphs with NN(1 hidden layer with 4 neurons, with ReLu for the hidden layer and Sigmoid for the output layer)

CUP Validation schema: data splitting

The ML-CUP24-TR dataset was split into development set(80%) and internal test set(20%).



CUP Validation schema: model selection

NN HP Space

Hp	Range	Description
units	32, 64, 96, 128	number of units per layer
num_layers	1, 2, 3	number of hidden layers
lambda	1e-6 a 1e-2 (log)	L2 regularization coefficient(lambda)
dropout	0.0, 0.1, 0.2, 0.3	Dropout percentage
eta	1e-4 a 1e-2 (log)	Learning rate (eta)

Ridge HP Space

Hp	Range	Description
lambda	1e-6 a 1.0 (log)	L2 regularization coefficient(lambda)
eta	1e-6 a 1e-1 (log)	Learning rate (eta)

The adopted procedure consisted of a 5-fold cross validation with random search (50 hp configurations per fold). Subsequently, the best model was selected and trained on the whole development set and then tested on the internal test set. Lastly, the model was retrained on the whole ML-CUP24-TR dataset before being used for predicting the outputs of the blind test.

RBF Hp Space

Hp	Range	Description
n_centers	[100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, 170, 175, 180, 185, 190, 195, 200]	Number of RBF centers (used in the RBF layer)
gamma	[0.1, 0.15, 0.2, 0.25, 0.3]	Gamma value for the RBF activation function
eta	[0.01, 0.05, 0.1, 0.2, 0.3, 0.4]	Learning rate for the Adam optimizer

CUP Results

We first tried using Grid Search, but due to time constraints we settled on using random search (Keras Tuner). It took us up to 12 hours to run with a M1 and a M2 Macbook Air(with NN). With random search and 250 hyperparameter sets took us only 25-30 minutes per run(for all models).

Hyperparameters and performance of each fold's best model(NN) in the final CV in term of validation MEE

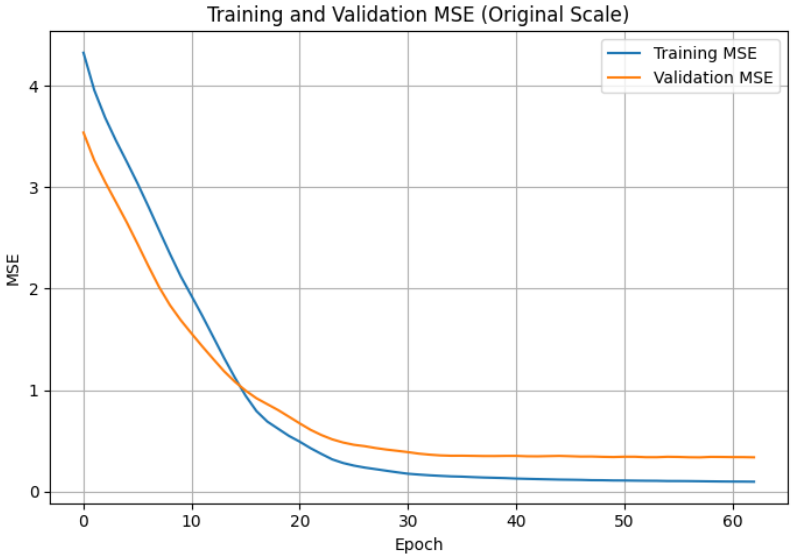
units	num_layers	lambda	dropout	eta	batch_size	fold	mean_euclidean_error	val_mean_euclidean_error
128	2	4.0E-05	0.1	0.0003340373742720006	16	1	0.5823448209471724	0.654441903092893
32	1	1.2E-05	0.0	0.005486272562976355	128	2	0.4876851786935979	0.7673934876174269
64	3	1.7E-06	0.1	0.0004826038221672518	16	3	0.5497460574487568	0.8026834356043445
96	1	1.9E-06	0.1	0.0010731280858402223	32	4	0.5228849806981878	0.5822320423174928
32	3	5.0E-06	0.0	0.0022232368862489644	112	5	0.5176003019192953	0.5473737418140548

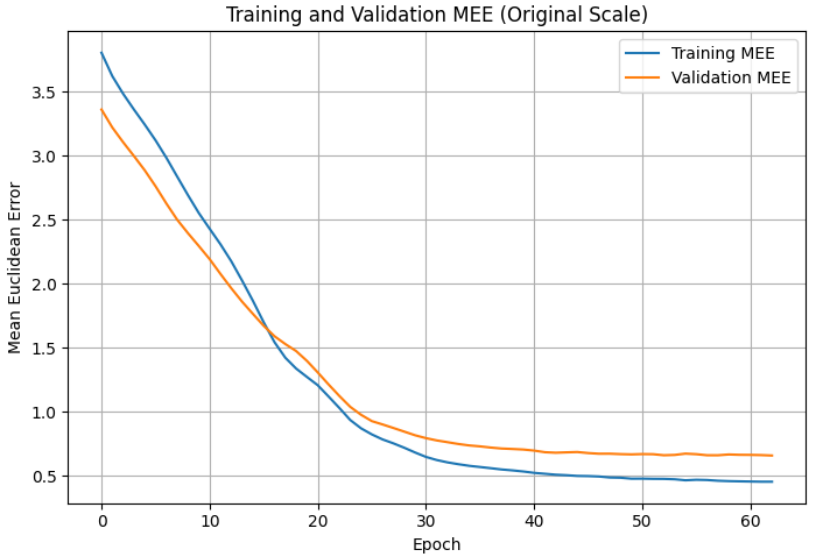
The final model(NN), built with the best hyperparameters(best in terms of MEE), retrained on all the training set and tested on the internal test set

units	num_layers	lambda	dropout	eta	batch_size	fold	mean_euclidean_error	test_mean_euclidean_error
32	3	5.0E-06	0.0	0.0022232368862489644	112	5	0.45305779036346755	0.6576384612834375

CUP Results

The Final model's learning curve(NN used for predictions on the CUP TS)

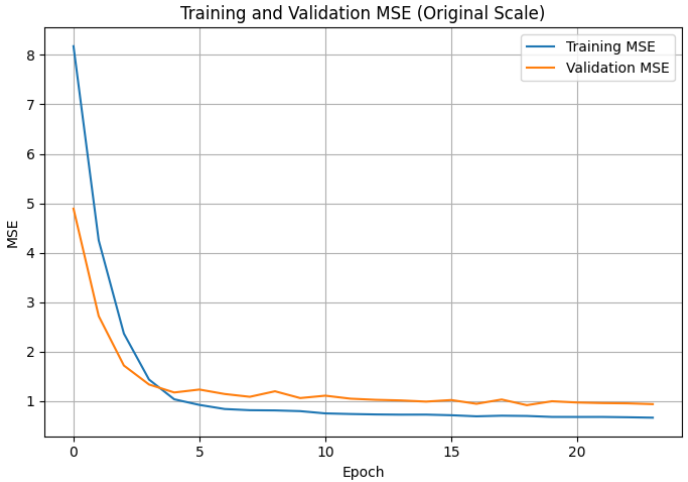


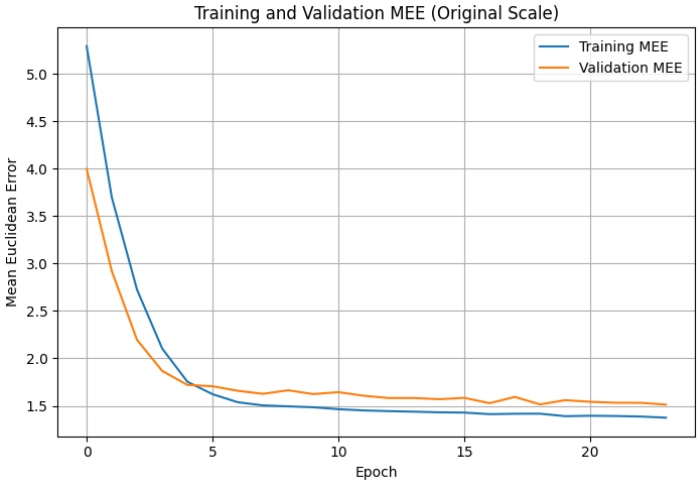


CUP Results

Hyperparameters and learning curve of the final Ridge Regression model

lambda	eta	batch_size	mse	val_mse	mean_euclidean_error	test_mean_euclidean_error
4,68E-05	0.016107733753153157	16	0.6627398304113608	0.9369705357283487	1,37476226522372	1.5126772751701543

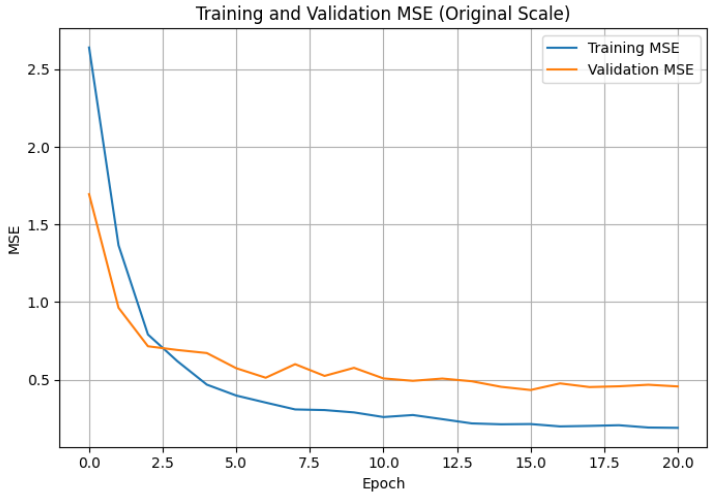


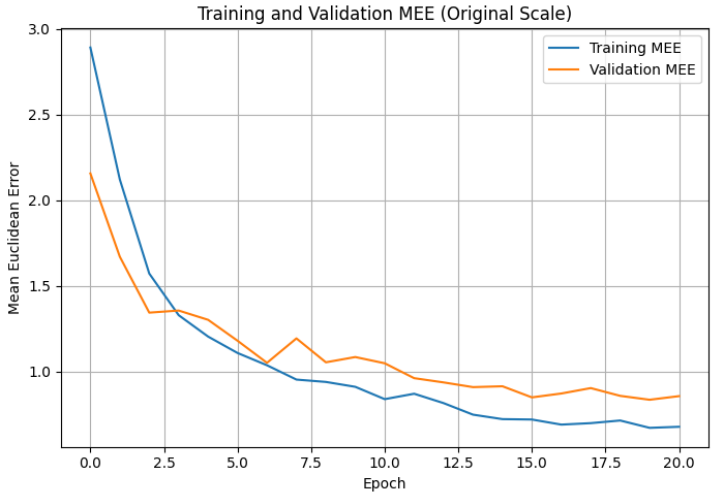


CUP Results

Hyperparameters and learning curve of the final RBF Network model

n_centers	gamma	eta	batch_size	mse	val_mse	mean_euclidean_error	test_mean_euclidean_error
125	0.22	0.28	48	0.18880158858599283	0.45574237736130377	0.6787818508542773	0.8575121332574434





Discussion

In our experiments, we found that:

- Ridge Regression performs consistently worse than NN, but the trend inverts when comparing the results on Monk-3 dataset.
- Batch size can change drastically a model’s performance. We had an issue with Keras Tuner and the batch size was fixed, after solving it, our models’s performances improved.
- Random search can be a powerful tool to search a broad and continuous hyperparameter space.
- One-hot encoding made a great difference for Monk tests.

Conclusions

We found this experience to be very interesting and useful for developing a better understanding of the theoretical aspects studied during the course.

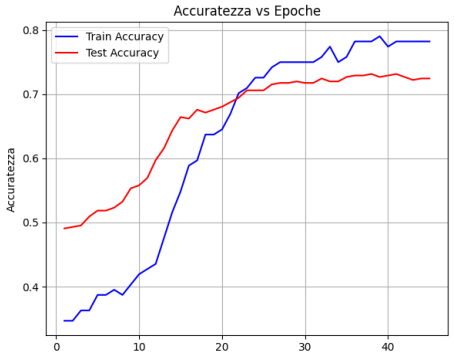
We acknowledge that this was a key experience, that thought us how important it is to work in a group, essential to prepare us to face the future challenges of work environment.

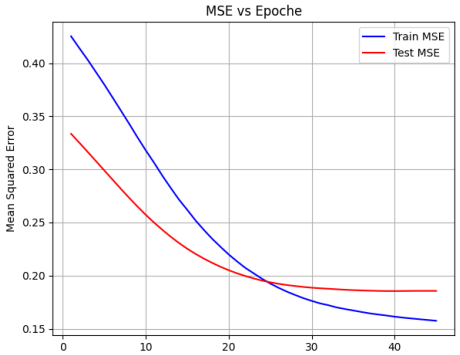
The predictions were saved on the file Risi_Scotti_ML-CUP24-TS.csv.

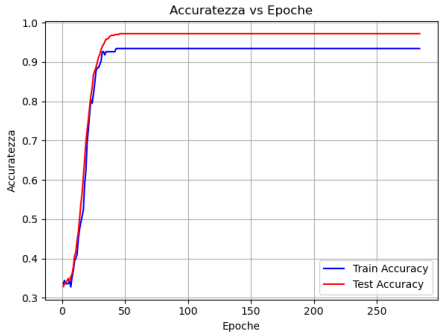
Our nickname is Risi Scotti.

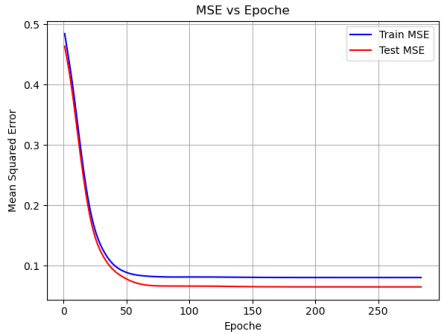
Appendix

Monk 1 Ridge

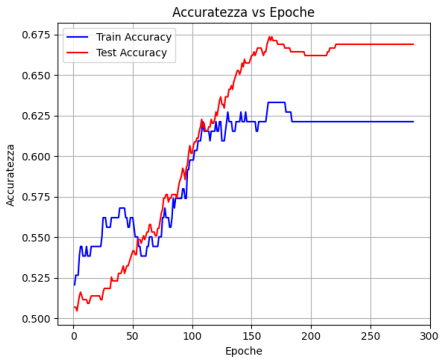


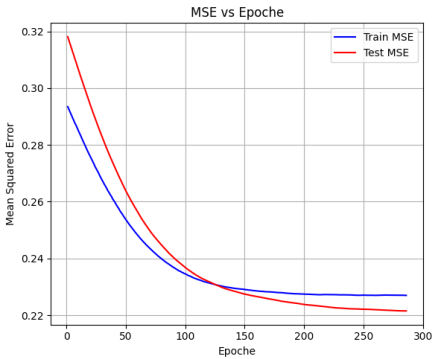






Monk 2 Ridge





Monk 3 Ridge

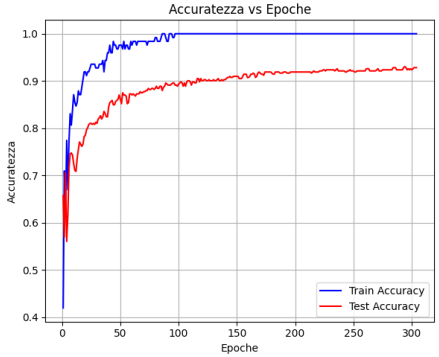
Appendix

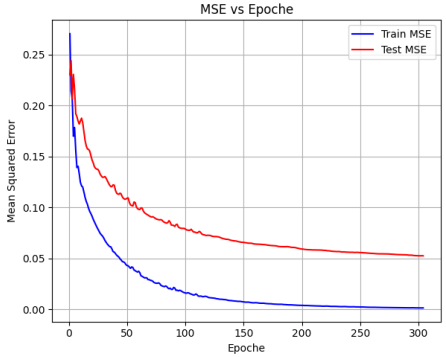
Ridge Regression Hyperparameter Tuning(CUP) - Best model for each fold

lambda	eta	batch_size	fold	mse	val_mse	mean_euclidean_error	val_mean_euclidean_error
0.00024683720448102 04	0.0669915437673078 3	16	1	0.65855465451636 41	0.82232525996774 59	1.3812835445176600	1.37550579946326
3.55E-06	0.0084564342966125 6	16	2	0.76076292074268 4	0.80887691132617 93	1.4591746435557400	1.5208871570526400
0.00048574595827698 64	0.0497479594662474 24	96	3	0.65623665723530 63	0.80573709747239 37	1.33637090492062	1.5138571313484300
0.00069623244017176 76	0.0297612711429131 6	16	4	0.65385605775582 2	0.74527384693898 86	1.340182301904970	1.485118935370190
4.68E-05	0.0161077337531531 57	16	5	0.71450587230778 95	0.61146969496579 47	1.4245221374248300	1.3057131054250200

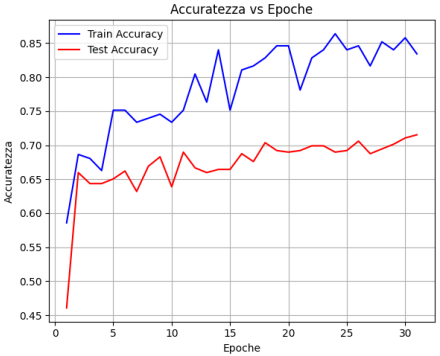
Appendix

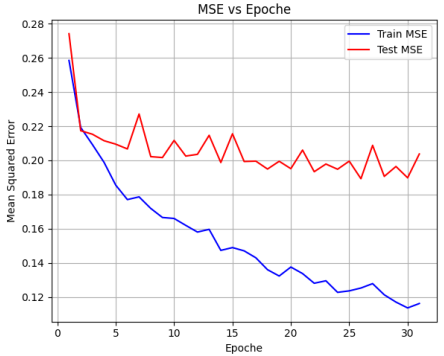
Monk 1 RBF

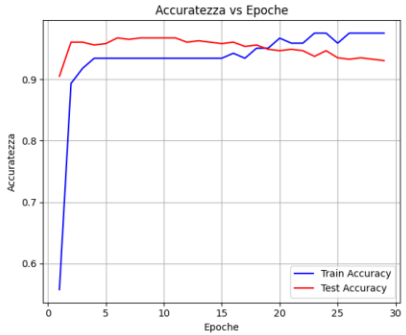


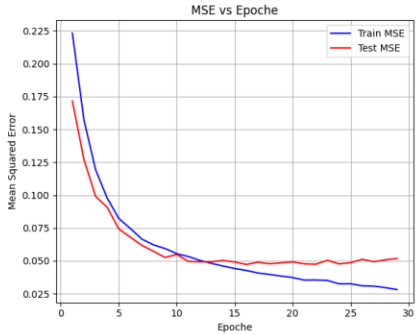


Monk 2 RBF









Monk 3 RBF

Appendix

RBF Monk results

Task	centers	gamma	eta	batch_size	accuracy/val_accuracy	mse/val_mse
Monk-1	145	0.125	0.31	80	100%/92.8%	0.0014/0.052
Monk-2	120	0.150000000000000002	0.34	16	83.4%/71.5%	0.1161/0.2037
Monk-3	175	0.175	0.2	16	97.5%/93%	0.0281/0.051

RBF Network Hyperparameter Tuning(CUP) - Best model for each fold

centers	gamma	eta	batch_size	fold	mse	val_mse	mean_euclidean_error	val_mean_euclidean_error
195	0.28	0.23	32	1	6.177328851585040	5.509577590931970	4.470656594060990	4.162796733295300
160	0.22	0.09999999999999999	32	2	4.394603325158180	5.9287496041694400	3.588052221595290	4.342357352797260
185	0.17	0.16	16	3	4.518683959296190	4.198017274497010	3.7957561386794800	3.671718817215620
125	0.22	0.28	48	4	4.374723044405690	4.168321347721280	3.827341492109410	3.6425679674265700
140	0.27	0.31	48	5	7.102200713668150	7.1806889595744000	4.722077436918210	4.790320696960230