Grado en Ingeniería Informática

Lab Assignment 3: Radial basis functions neural networks

Universidad de Córdoba
Fourth year of "Grado en Ingeniería Informática"
Dpto. de Informática y análisis Numérico
Introduction to computational models
Course 2023-2024

Autor:

Antonio Llamas García i92llgaa@uco.es
31886320V

Docentes:

Pedro A. Gutiérrez Peña pagutierrez@uco.es
Javier Sánchez Monedero jsanchezm@uco.es
César Hervás Martínez chervas@uco.es

Córdoba, 12 de diciembre de 2023

Índice

1. RBF Training Overview

In the RBF training stage, a hybrid supervised/unsupervised approach is employed for training a Radial Basis Function (RBF) neural network. The process involves the following key steps:

1.1. Data preparation

- Input data: training (`train_file`) and test (`test_file`) datasets.
- Output configuration: regression/classification (`classification`) with specified output variables (`outputs`).

1.2. RBF initialization and clustering

- Calculate RBF count based on ratio (`ratio_rbf`) and total training patterns.
- Use K-means clustering to initialize centroids and group patterns.

1.3. Radius adjustment

- Adjust RBF radii to capture clustered pattern variations effectively.

1.4. R matrix calculation

- Generate R matrix from distances between patterns and centroids.

1.5. Coefficient/LR calculation

1.5.1. For regression:

- Invert R matrix and calculate coefficients using linear regression.

1.5.2. For classification:

- Use logistic regression for coefficient calculation.

1.6. Testing set R matrix calculation

- Apply the model to the test set, calculate distances, and generate the test R matrix.

1.7. Prediction and evaluation

- For training/testing:
  - Perform predictions.
  - Evaluate metrics (MSE for regression, CCR for classification).

1.8. Fairness metrics calculation

- If fairness enabled (`fairness`):
  - Identify sensitive features, calculate fairness metrics.

1.9. Model saving

- Save the model to specified file (`model_file`).

2. First Experiment

For all the datasets, consider a number hidden neurons(n1) equal to the (5 %, 15 %, 25 %, 50 %) of the total number of patterns of the dataset. In this stage, for classification problems use L1 regularisation and $\eta = 10^{-5}$.

2.1. Datasets Description

In this section, we provide a brief overview of the datasets considered for this experiment with our program.

2.1.1. Quake Dataset

The Quake dataset consists of 1633 training patterns and 546 testing patterns. It aims to predict the strength of an earthquake measured on the Richter scale. The input variables include the depth of focus, latitude, and longitude.

2.1.2. Sin-Function Dataset

The Sin-function dataset comprises 120 training patterns and 41 testing patterns. It has been generated by adding random noise to the sin function, introducing variability for predictive modeling.

2.1.3. Auto MPG Dataset

The Auto MPG dataset, obtained from the UCI Machine Learning Repository, provides information about various car models, including attributes and fuel efficiency. Binarized discrete attributes and removed model names for a final set of attributes: cylinders, displacement, horsepower, weight, acceleration, modelYear, USA, Europe, Japan, and MPG (target variable).

2.1.4. ProPublica COMPAS Dataset

The ProPublica COMPAS dataset focuses on the performance of the COMPAS algorithm in the U.S. criminal justice system. It analyzes risk scores for risk of recidivism"with binary outcomes indicating rearrest within two years. Attributes include sex, age, age category, juvenile felony/misdemeanor counts, prior counts, race, and charge degree.

2.1.5. noMNIST Dataset

Originally comprising 200,000 training patterns and 10,000 test patterns, the noMNIST dataset has been reduced for computational efficiency. It includes 900 training patterns and 300 test patterns, featuring letters (a to f) written in different typologies or symbols. Images are represented as a squared grid of 28×28 pixels, and classes correspond to written letters (a, b, c, d, e, and f).

2.2. Parameter Description

In this section, we provide a brief overview of the values considered for various parameters in the experiments.

2.2.1. Regression Datasets

1. **Sin-function dataset:**
   - **Parameter:** `ratio_rbf` (`-r`)
   - **Values:** 0.05, 0.15, 0.25, 0.5
2. **Quake dataset:**
   - **Parameter:** `ratio_rbf` (`-r`)
   - **Values:** 0.05, 0.15, 0.25, 0.5
3. **Auto MPG dataset:**

- **Parameter:** `ratio_rbf` (`-r`)
- **Values:** 0.05, 0.15, 0.25, 0.5

2.2.2. Classification Datasets

1. **ProPublica COMPAS dataset:**
   - **Parameters:** `ratio_rbf` (`-r`), `classification` (`-c`), `eta` (`-e`)
   - **Values:** 0.05, 0.15, 0.25, 0.5 (for `ratio_rbf`), 0.00001 (for `eta`)
2. **noMNIST dataset:**
   - **Parameters:** `ratio_rbf` (`-r`), `classification` (`-c`), `eta` (`-e`)
   - **Values:** 0.05, 0.15, 0.25, 0.5 (for `ratio_rbf`), 0.00001 (for `eta`)

For each dataset, the script iterates over different values of the `ratio_rbf` parameter, recording the "Training MSE"for each configuration.

2.3. Experimental Setup

- To assess the program's performance across various datasets and parameter configurations, an experimentation script, named `experiment1.py`, was utilized. This script is located in the `experiments` folder within the compressed file.
- For the "Nomnist"dataset, which involves non-binary classification, a dedicated program was created. This specialized script enables the acquisition of metrics specifically tailored for non-binary classification scenarios.
- The experimentation process systematically adjusted parameters, providing insights into the program's behavior under diverse conditions.
- Scripts and detailed information about the experiments are available in the provided compressed file, facilitating reference and reproducibility.
- This iterative evaluation approach offers a comprehensive understanding of the program's performance, aiding in the refinement and optimization of its behavior for various datasets and use cases.

2.4. Results Obtained

The following table summarizes the results obtained from Experiment 1, where different values of the `ratio_rbf` parameter were explored. The number of hidden neurons considered were 5 %, 15 %, 25 %, and 50 %.

2.4.1. Regression Datasets (MSE)

Dataset	Ratio (%)	Train MSE (Reg)	Test MSE (Reg)
Sin-function	5	0.0138 ± 0.0002	0.0222 ± 0.0002
Sin-function	15	0.0112 ± 0.0001	0.3698 ± 0.0808
Sin-function	25	0.0103 ± 0.0001	3.1956 ± 2.2984
Sin-function	50	0.0104 ± 0.0000	2.0216 ± 0.2969
Quake	5	0.0284 ± 0.0001	0.0284 ± 0.0003
Quake	15	0.0254 ± 0.0001	0.0409 ± 0.0021
Quake	25	0.0222 ± 0.0001	0.9493 ± 0.2966
Quake	50	0.0185 ± 0.0001	157.2650 ± 57.9473
Auto MPG	5	14.9779 ± 0.7403	16.9387 ± 0.9382
Auto MPG	15	7.4294 ± 0.2382	9.2818 ± 0.8491
Auto MPG	25	5.5638 ± 0.1335	12.3930 ± 0.7309
Auto MPG	50	2.6573 ± 0.1210	72.8798 ± 44.7745

Tabla 1: Regression Datasets - Experiment 1 Results (MSE)

2.4.2. Classification Datasets (CCR)

Dataset	Ratio (%)	Train CCR (Class)	Test CCR (Class)
Compas	5	68.07 % ± 0.07 %	66.81 % ± 0.11 %
Compas	15	68.12 % ± 0.03 %	66.76 % ± 0.08 %
Compas	25	68.26 % ± 0.03 %	66.97 % ± 0.05 %
Compas	50	68.22 % ± 0.02 %	67.05 % ± 0.04 %
noMNIST	5	84.13 % ± 0.46 %	88.20 % ± 0.45 %
noMNIST	15	87.47 % ± 0.41 %	90.67 % ± 0.37 %
noMNIST	25	87.96 % ± 0.29 %	90.87 % ± 0.27 %
noMNIST	50	88.31 % ± 0.37 %	91.07 % ± 0.39 %

Tabla 2: Classification Datasets: Correct Classification Rate

2.5. Analysis of Results

The results obtained from the experiments provide valuable insights into the performance of the radial basis function (RBF) network on various datasets. In this subsection, we discuss the key findings and analyze the implications of the results.

2.5.1. Regression Datasets (MSE)

The regression results, as shown in Table 1, reveal the model's performance on different datasets with varying ratios of hidden neurons.

For the Sin-function dataset, we observe that the mean squared error (MSE) decreases as the ratio of hidden neurons increases. This trend is consistent across different values of r, indicating that a higher percentage of hidden neurons contributes to better regression performance.

Similar observations can be made for the Quake and Auto MPG datasets. The decreasing MSE suggests that increasing the number of hidden neurons positively impacts the model's ability to approximate the target function.

However, the results for the Auto MPG dataset at $r = 0{,}5$ raise concerns. The substantially higher MSE on the test set compared to the training set may indicate overfitting, highlighting the need for regularization or further tuning.

2.5.2. Classification Datasets (CCR)

The classification results, presented in Table 2, demonstrate the model's ability to correctly classify instances in the Compas and noMNIST datasets.

Compas Dataset The Correct Classification Rate (CCR) for the Compas dataset remains consistently high across different ratios of hidden neurons. The model achieves an average CCR of over 66 %, indicating its effectiveness in predicting outcomes.

It is essential to address potential algorithmic bias in the Compas dataset, as high classification accuracy alone does not guarantee fairness. Bias assessment should be conducted to ensure that the model's predictions are not disproportionately affecting certain demographic groups. Fairness-aware techniques and metrics should be considered to mitigate biases.

noMNIST Dataset The noMNIST dataset results demonstrate excellent performance, with CCR exceeding 90 % in most cases. The model exhibits robust classification capabilities for character recognition.

2.5.3. Algorithmic Bias Assessment in COMPAS

The high classification accuracy in the Compas dataset raises questions about potential algorithmic bias. Bias assessment is crucial to evaluate whether the model's predictions are fair and unbiased across different demographic groups.

Table 3 presents detailed results for the ProPublica COMPAS dataset, including False Negative (FN), False Positive (FP), True Positive (TP), and True Negative (TN) rates for each value of r.

Future work should focus on employing fairness-aware machine learning techniques and conducting a thorough bias analysis to identify and rectify any disparities in prediction outcomes. This is imperative to ensure that the model's decisions do not contribute to unjust practices or reinforce existing biases in the criminal justice system.

In conclusion, while the RBF network demonstrates promising performance, the discussion emphasizes the importance of a comprehensive analysis, especially in cases where the model's predictions can have significant real-world implications, like this.

- **False Positive (FP) and False Negative (FN) Rates**
  - *False Positives (FP0 and FP1):* These instances occur when the model predicts a positive outcome (e.g., recidivism), but the actual outcome is negative. In the criminal justice context, a false positive can have significant consequences, potentially leading to unnecessary legal actions for an individual predicted to reoffend but who does not. Increasing the threshold generally reduces false positives, which may align with priorities aiming to minimize unwarranted interventions.
  - *False Negatives (FN0 and FN1):* These cases arise when the model predicts a negative outcome, but the actual outcome is positive. In COMPAS, a false negative implies failing to identify an individual at risk of recidivism. As the threshold increases, false negatives tend to rise, posing a concern if the goal is to identify all high-risk individuals accurately.
- **True Positive (TP) and True Negative (TN) Rates**
  - *True Positives (TP0 and TP1):* Instances where the model correctly predicts a positive outcome, and the actual outcome is positive. In the context of COMPAS, correctly identifying individuals at risk of recidivism is crucial. The rates decrease as the threshold increases, indicating a more conservative model.
  - *True Negatives (TN0 and TN1):* These cases involve the model correctly predicting a negative outcome, and the actual outcome is negative. Correctly identifying individuals not at risk is vital in the criminal justice system to avoid unnecessary interventions. The rates generally increase with the threshold.

Ratio (r)	Metric	Training	Test
0.05	FN0	60.37 % ± 0.44 %	51.24 % ± 0.75 %
	FN1	32.46 % ± 0.25 %	37.58 % ± 0.22 %
	FP0	14.02 % ± 0.50 %	21.25 % ± 0.28 %
	FP1	31.09 % ± 0.18 %	28.89 % ± 0.15 %
	TP0	39.63 % ± 0.44 %	48.76 % ± 0.75 %
	TP1	67.54 % ± 0.25 %	62.42 % ± 0.22 %
	TN0	85.98 % ± 0.50 %	78.75 % ± 0.28 %
	TN1	68.91 % ± 0.18 %	71.11 % ± 0.15 %
0.15	FN0	59.26 % ± 0.63 %	50.62 % ± 1.14 %
	FN1	32.23 % ± 0.18 %	37.32 % ± 0.14 %
	FP0	14.64 % ± 0.42 %	21.82 % ± 0.42 %
	FP1	31.18 % ± 0.17 %	29.28 % ± 0.24 %
	TP0	40.74 % ± 0.63 %	49.38 % ± 1.14 %
	TP1	67.77 % ± 0.18 %	62.68 % ± 0.14 %
	TN0	85.36 % ± 0.42 %	78.18 % ± 0.42 %
	TN1	68.82 % ± 0.17 %	70.72 % ± 0.24 %
0.25	FN0	59.23 % ± 0.32 %	50.23 % ± 0.40 %
	FN1	32.19 % ± 0.04 %	37.25 % ± 0.23 %
	FP0	14.04 % ± 0.16 %	21.14 % ± 0.23 %
	FP1	31.21 % ± 0.17 %	29.28 % ± 0.20 %
	TP0	40.77 % ± 0.32 %	49.77 % ± 0.40 %
	TP1	67.81 % ± 0.04 %	62.75 % ± 0.23 %
	TN0	85.96 % ± 0.16 %	78.86 % ± 0.23 %
	TN1	68.79 % ± 0.17 %	70.72 % ± 0.20 %
0.5	FN0	59.08 % ± 0.51 %	49.61 % ± 0.25 %
	FN1	31.61 % ± 0.17 %	36.30 % ± 0.33 %
	FP0	14.76 % ± 0.37 %	21.36 % ± 0.11 %
	FP1	31.54 % ± 0.10 %	29.94 % ± 0.22 %
	TP0	40.92 % ± 0.51 %	50.39 % ± 0.25 %
	TP1	68.39 % ± 0.17 %	63.70 % ± 0.33 %
	TN0	85.24 % ± 0.37 %	78.64 % ± 0.11 %
	TN1	68.46 % ± 0.10 %	70.06 % ± 0.22 %

Tabla 3: Algorithmic Bias Assessment in ProPublica COMPAS Dataset

- **Threshold Variation**
  - *Threshold Adjustment:* Depending on the objectives of the criminal justice system utilizing this model, there's a need to adjust the threshold considering trade-offs between false positives and false negatives. A higher threshold reduces false positives but increases false negatives, and vice versa. The adjustment should align with the system's priorities, emphasizing the importance of minimizing false positives or ensuring comprehensive identification of high-risk individuals. Regular evaluation and adaptation of the threshold contribute to optimizing the model for the specific goals of the criminal justice context.

3. Second Experiment

For the classification problems, once the optimal architecture is determined, experiment with the following values for η: $\eta = 10^{-3}$, $\eta = 10^{-2}$, $\eta = 10^{-1}$, $\eta = 1$, $\eta = 10^{1}$, $\eta = 10^{2}$, $\eta = 10^{3}$, along with the two types of regularization (L2 and L1). What is happening? Compute the difference in number of coefficients for COMPAS and noMNIST dataset when the regularization type is modified (L2 vs L1).

3.1. Datasets Description

In this section, we provide a brief overview of the datasets considered for these experiments with our program.

3.1.1. ProPublica COMPAS Dataset

The ProPublica COMPAS dataset focuses on the performance of the COMPAS algorithm in the U.S. criminal justice system. It analyzes risk scores for risk of recidivism with binary outcomes indicating rearrest within two years. Attributes include sex, age, age category, juvenile felony/misdemeanor counts, prior counts, race, and charge degree.

3.1.2. noMNIST Dataset

Originally comprising 200,000 training patterns and 10,000 test patterns, the noMNIST dataset has been reduced for computational efficiency. It includes 900 training patterns and 300 test patterns, featuring letters (a to f) written in different typologies or symbols. Images are represented as a squared grid of 28×28 pixels, and classes correspond to written letters (a, b, c, d, e, and f).

3.2. Parameter Description

In this section, we provide a brief overview of the values considered for the parameters that are changing in the experiments.

For the ProPublica COMPAS and noMNIST datasets, the experiment aims to identify the optimal architecture with $r = 0{,}5$. The investigation involves varying the learning rate (η) and the type of regularization (L1 or L2).

3.2.1. Best Architecture

The best architecture for both datasets is achieved with $r = 0{,}5$, in wich the CCR obtained is the best..

3.2.2. Parameters Changing

The experiment focuses on two key parameters: the learning rate (η) and the type of regularization (L1 or L2).

3.2.3. Configurations

For each dataset, the following configurations are explored:

1. $\eta = 10^{-3}, L1$
2. $\eta = 10^{-3}, L2$
3. $\eta = 10^{-2}, L1$
4. $\eta = 10^{-2}, L2$
5. $\eta = 10^{-1}, L1$
6. $\eta = 10^{-1}, L2$
7. $\eta = 1, L1$
8. $\eta = 1, L2$

9. $\eta = 10, L1$
10. $\eta = 10, L2$
11. $\eta = 100, L1$
12. $\eta = 100, L2$
13. $\eta = 1000, L1$
14. $\eta = 1000, L2$

3.3. Experimental Setup

- To assess the program's performance in classification datasets under different configurations, we conducted Experiment 2 using the `experiment2.py` script. This script is located in the `experiments` folder within the compressed file, similar to Experiment 1.
- Our focus in Experiment 2 was specifically on classification datasets, as it is in these datasets that L1 and L2 regularizations are applied.
- The dedicated script, `experiment2.py`, systematically explores various parameter settings and reports the number of coefficients used in the regularization process as part of its output.
- For simplicity, we concentrated our study on the datasets relevant to classification scenarios, optimizing the program's behavior through the application of L1 and L2 regularizations.
- Detailed information about the experiment, including scripts and results, can be found in the provided compressed file, ensuring transparency, reference, and reproducibility.
- Through this focused study, we gain valuable insights into the impact of regularization on the program's performance in classification datasets, facilitating further refinement and optimization for diverse use cases.

3.4. Results Obtained

The tables present the number of coefficients and their averages for different configurations of the ProPublica COMPAS and noMNIST datasets. Notably, the regularization parameter (η) and the type of regularization (L1 or L2) influence the number of coefficients. For both datasets, configurations with higher η values tend to result in fewer coefficients, reflecting the impact of regularization strength on the model's complexity.

3.4.1. ProPublica COMPAS Dataset

Configuration	Regularization	Number of Coeffs	Average Coeffs
1	$L1$	[985, 984, 987, 990, 989]	987.00
2	$L2$	[984, 982, 986, 992, 989]	986.60
3	$L1$	[990, 984, 989, 996, 992]	990.20
4	$L2$	[984, 982, 987, 992, 989]	986.80
5	$L1$	[999, 998, 1013, 1005, 1007]	1004.40
6	$L2$	[984, 982, 987, 992, 989]	986.80
7	$L1$	[685, 595, 519, 476, 683]	591.60
8	$L2$	[978, 982, 987, 992, 987]	985.20
9	$L1$	[140, 86, 95, 78, 75]	94.80
10	$L2$	[983, 974, 985, 979, 981]	980.40
11	$L1$	[0, 0, 0, 0, 0]	0.00
12	$L2$	[994, 994, 998, 1000, 995]	996.20
13	$L1$	[0, 0, 0, 0, 0]	0.00
14	$L2$	[1008, 1008, 1008, 1006, 1013]	1008.60

Tabla 4: Results for ProPublica COMPAS Dataset

Configuration	**Type**	**Training CCR (%)**	**Test CCR (%)**	**Avg. Coefficients**
1	L1	68.22 ± 0.03	67.05 ± 0.04	987.00
2	L2	68.21 ± 0.03	67.05 ± 0.04	986.80
3	L1	68.22 ± 0.03	67.05 ± 0.04	990.60
4	L2	68.21 ± 0.02	67.05 ± 0.04	986.60
5	L1	68.26 ± 0.03	67.03 ± 0.06	1003.80
6	L2	68.22 ± 0.02	67.05 ± 0.04	986.20
7	L1	68.17 ± 0.01	66.61 ± 0.04	591.20
8	L2	68.22 ± 0.03	67.03 ± 0.04	984.40
9	L1	67.61 ± 0.03	66.26 ± 0.04	94.60
10	L2	68.24 ± 0.04	66.93 ± 0.04	981.00
11	L1	54.23 ± 0.00	56.32 ± 0.00	0.00
12	L2	67.90 ± 0.04	66.59 ± 0.02	996.20
13	L1	54.23 ± 0.00	56.32 ± 0.00	0.00
14	L2	67.71 ± 0.07	66.12 ± 0.10	1009.00

Tabla 5: Experiment 2 Results(CCR) - ProPublica COMPAS Dataset

3.4.2. noMNIST Dataset

Configuration	Regularization	Number of Coeffs	Average Coeffs
1	$L1$	[887, 901, 912, 873, 881]	890.80
2	$L2$	[885, 903, 912, 872, 880]	890.40
3	$L1$	[890, 901, 915, 877, 891]	894.80
4	$L2$	[886, 902, 911, 871, 881]	890.20
5	$L1$	[903, 919, 930, 896, 915]	912.60
6	$L2$	[886, 900, 908, 867, 881]	888.40
7	$L1$	[452, 416, 430, 428, 446]	434.40
8	$L2$	[861, 878, 877, 844, 862]	864.40
9	$L1$	[74, 84, 84, 85, 83]	82.00
10	$L2$	[847, 861, 863, 847, 857]	855.00
11	$L1$	[0, 0, 0, 0, 0]	0.00
12	$L2$	[956, 984, 968, 971, 963]	968.40
13	$L1$	[0, 0, 0, 0, 0]	0.00
14	$L2$	[1044, 1057, 1040, 1033, 1041]	1043.00

Tabla 6: Results for noMNIST Dataset

3.5. Analysis of Results

The analysis of Experiment 2, focusing on comparing the numbers of coefficients, reveals crucial insights into the behavior of the model under different regularization settings.

3.5.1. ProPublica COMPAS Dataset

Effect of η on Number of Coefficients As η increases from 10^{-3} to 1000, the number of coefficients decreases for both L1 and L2 regularization. This reduction is more pronounced with L1 regularization, indicating a stronger effect on feature sparsity.

Difference in Number of Coefficients (L2 vs L1) The difference in the number of coefficients between L2 and L1 regularization becomes more noticeable for higher η values. L1 regularization leads to more zero-valued coefficients, emphasizing feature selection.

Difference in CCR The COMPAS dataset reveals interesting trends in the evolution of the Correct Classification Rate (CCR) across various configurations. L1 regularization consistently yields high CCR values, showcasing the effectiveness of this regularization technique. The impact of varying η is notable, with CCR generally improving as η increases. However, the trend is not linear, as extremely high η values

Configuration	**Type**	**Training CCR (%)**	**Test CCR (%)**	**Avg. Coefficients**
1	L1	88.31 ± 0.37	91.07 ± 0.39	890.80
2	L2	88.31 ± 0.37	91.07 ± 0.39	890.40
3	L1	88.31 ± 0.37	91.07 ± 0.39	894.80
4	L2	88.31 ± 0.37	91.07 ± 0.39	890.20
5	L1	88.27 ± 0.32	90.87 ± 0.16	912.60
6	L2	88.27 ± 0.38	91.07 ± 0.39	888.40
7	L1	85.09 ± 0.25	88.47 ± 0.27	434.40
8	L2	87.67 ± 0.35	90.87 ± 0.45	864.40
9	L1	71.44 ± 0.21	72.07 ± 0.49	82.00
10	L2	82.87 ± 0.33	86.53 ± 0.45	855.00
11	L1	16.67 ± 0.00	16.67 ± 0.00	0.00
12	L2	75.04 ± 0.18	73.80 ± 0.62	968.40
13	L1	16.67 ± 0.00	16.67 ± 0.00	0.00
14	L2	68.71 ± 0.58	67.33 ± 0.67	1043.00

Tabla 7: Experiment 2 Results(CCR) - noMNIST Dataset

(100 and 1000) lead to a decline in CCR. L2 regularization, on the other hand, exhibits stability across a range of η values, with minimal fluctuations in CCR. The comparison between L1 and L2 regularization highlights the dataset-specific behavior, emphasizing the need for careful consideration of regularization techniques in classification tasks.

3.5.2. noMNIST Dataset

Effect of η on Number of Coefficients Similar to the ProPublica COMPAS dataset, increasing η results in a decrease in the number of coefficients for both L1 and L2 regularization. L1 regularization induces a more substantial reduction in coefficients compared to L2.

Difference in Number of Coefficients (L2 vs L1) The difference in coefficients between L2 and L1 regularization is evident across various η values. L1 tends to drive more coefficients to zero, promoting a simpler and more interpretable model.

Difference in CCR The evolution of Correct Classification Rate (CCR) for the noMNIST dataset follows distinctive patterns across different configurations. For configurations employing L1 regularization, there is a consistent performance improvement as the regularization parameter (η) increases from 10^{-3} to 10^{-1}. However, a sharp decline in CCR is observed when η is set to 1, indicating a sensitivity to the strength of regularization. Interestingly, the drop is more pronounced with L1 regularization compared to L2. The decrease in CCR is substantial for extremely high η values (100 and 1000), reaching an almost negligible level, suggesting that excessive regularization may lead to underfitting and a loss of predictive capacity.

3.6. Recommendations

1. **Best Architecture:** A regularization ratio (r) of 0.5 remains consistent as the best architecture for both datasets.
2. **Impact of η:** Choose an appropriate η to balance overfitting and underfitting. Cross-validation can assist in identifying optimal hyperparameters.
3. **Difference in Regularization Types (L2 vs L1):** L1 regularization tends to produce sparser models, emphasizing feature selection. L2 regularization is less aggressive in eliminating coefficients.
4. **Practical Considerations:** Carefully select regularization parameters based on the trade-off between model complexity and performance.

3.7. Difference in Number of Coefficients (L2 vs L1)

The difference in the number of coefficients between L2 and L1 regularization is substantial, especially as η increases. This difference signifies the contrasting mechanisms of the two regularization types. L1 regularization tends to enforce sparsity by encouraging a subset of features to have zero coefficients. In contrast, L2 regularization penalizes large coefficients without driving them to exactly zero.

For both datasets, the impact of regularization type on the number of coefficients is more pronounced at higher η values. This emphasizes the importance of understanding the interplay between regularization strength and type in shaping the model's architecture and performance. Engineers should carefully consider these factors to tailor models to specific datasets and classification tasks.

4. Third Experiment

Implement the option `-v` so that, in classification problems, the classifier is trained using `sklearn.linear_model.Logis` instead of using the class `sklearn.linear_model.LogisticRegression`. This will automatically calculate (without looking at the test values) the optimal value of η. Set the object so that the values to be tested (`Cs`) are $\eta = 10^{-3}$, $\eta = 10^{-2}$, $\eta = 10^{-1}$, $\eta = 1$, $\eta = 10^{1}$, $\eta = 10^{2}$, and $\eta = 10^{3}$ and that the optimization is done using a stratified k-fold with $k = 3$. Compare, on the two classification datasets, the results you get using this strategy (under the assumption of L_1 and the best ratio of RBFs obtained above). Also, compare the values you choose for η with those obtained by the experimentation of the previous point. What can the differences be due to?

4.1. Datasets Description

In this section, we provide a brief overview of the datasets considered for this experiment with our program.

4.1.1. ProPublica COMPAS Dataset

The ProPublica COMPAS dataset focuses on the performance of the COMPAS algorithm in the U.S. criminal justice system. It analyzes risk scores for risk of recidivism with binary outcomes indicating rearrest within two years. Attributes include sex, age, age category, juvenile felony/misdemeanor counts, prior counts, race, and charge degree.

4.1.2. noMNIST Dataset

Originally comprising 200,000 training patterns and 10,000 test patterns, the noMNIST dataset has been reduced for computational efficiency. It includes 900 training patterns and 300 test patterns, featuring letters (a to f) written in different typologies or symbols. Images are represented as a squared grid of 28×28 pixels, and classes correspond to written letters (a, b, c, d, e, and f).

4.2. Parameters Description

The third experiment involves training a logistic regression model using LogisticRegressionCV on two distinct datasets: ProPublica COMPAS and noMNIST. The primary parameter being varied is the regularization strength represented by the r values. The experiment explores different values of r, specifically 0,05, 0,15, 0,25, and 0,5, to observe their impact on the model's performance. Each configuration specifies the dataset, training set, test set, the varying r value, and the utilization of cross-validation and verbose mode during training. This systematic exploration aims to understand how changes in the regularization strength influence the model's behavior and predictive capabilities for both classification datasets.

4.3. Experimental Setup

- To evaluate the program's performance with logistic regression on various datasets and regularization strengths, Experiment 3 was conducted using the `experiment3.py` script. This script is available in the `experiments` folder within the compressed file, analogous to Experiments 1 and 2.
- Experiment 3 focuses on logistic regression and systematically explores different values of the regularization parameter (r) using the `LogisticRegressionCV` class from the `sklearn.linear_model` module. The script employs a stratified k-fold cross-validation strategy with $k = 3$.
- The primary objective is to observe the impact of varying r values (0.05, 0.15, 0.25, 0.5) on the logistic regression model's performance, particularly in terms of training and testing mean squared error (MSE) and correct classification rate (CCR).
- Additionally, we aim to determine the optimal η parameter by leveraging the `LogisticRegressionCV` class, which automatically calculates the best value among $\eta = 10^{-3}$, $\eta = 10^{-2}$, $\eta = 10^{-1}$, $\eta = 1$, $\eta = 10$, $\eta = 10^{2}$, and $\eta = 10^{3}$ during the optimization process.
- Detailed results and information about Experiment 3, including the script (`experiment3.py`), are provided in the compressed file. This comprehensive documentation ensures transparency, enables reproducibility, and serves as a valuable reference for further analysis.

- Through this systematic exploration, we aim to understand how changes in the regularization strength and η parameter influence the logistic regression model's behavior and performance on the ProPublica COMPAS and noMNIST datasets.

4.4. Results Obtained

This subsection unveils the outcomes derived from Experiment 3, where the radius parameter (r) is systematically varied to assess its impact on logistic regression models. Focusing on the ProPublica COMPAS and noMNIST datasets, the results emphasize the Classification Correct Rate (CCR), providing insights into model performance. Notably, the exploration of different r values unveils the optimal η parameters, denoted as `best_Cs`, shedding light on the most effective regularization strategies for each architectural configuration.

Dataset	Regularization (r)	Training CCR (%)	Test CCR (%)	Best_Cs
ProPublica COMPAS	0.05	68.27 ± 0.11	67.14 ± 0.17	100.0
	0.15	68.27 ± 0.09	67.16 ± 0.17	10.0
	0.25	68.49 ± 0.07	67.58 ± 0.20	10.0
	0.5	68.49 ± 0.10	67.76 ± 0.06	10.0
noMNIST	0.05	85.53 ± 0.76	88.60 ± 0.57	1000.0
	0.15	89.11 ± 0.37	90.87 ± 0.34	1000.0
	0.25	90.40 ± 0.29	91.27 ± 0.49	1000.0
	0.5	91.16 ± 0.30	91.60 ± 0.33	100.0

Tabla 8: Experiment 3 Results: LogisticRegressionCV Parameters and Performance

4.5. Analysis of Results

In this subsection, we delve into a comprehensive analysis of the results obtained from Experiment 3. The primary objective is to evaluate the impact of varying regularization parameters (r) on classification performance and to contrast the findings with those from the preceding experiment. Additionally, we scrutinize the automatic optimization of the learning rate parameter (η) by LogisticRegressionCV and assess its influence on the selection of optimal regularization parameters (`Best_Cs`). This exploration aims to unravel the nuances between manual experimentation and automatic parameter tuning, shedding light on the intricacies of model adaptability to diverse dataset characteristics.

4.5.1. ProPublica COMPAS Dataset

- **Impact of r values on CCR:**
  - As the r value increases, the training and test CCR values remain relatively stable, indicating that changing the radius does not significantly affect the model's predictive performance.
  - The best regularization parameter (`Best_Cs`) consistently converges to 10.0 for different r values.
- **Comparison with Previous Experiment:**
  - The `Best_Cs` values obtained in this experiment using LogisticRegressionCV are notably different from those in the previous experiment, where traditional logistic regression with L1 and L2 regularization was employed.
  - The differences in `Best_Cs` could be attributed to the automatic optimization of η by LogisticRegressionCV, which considers a range of η values without relying on manual experimentation.

4.5.2. noMNIST Dataset

- **Impact of r values on CCR:**
  - Similar to the ProPublica COMPAS dataset, changes in the r values show a consistent pattern with respect to training and test CCR. The variations are relatively minor.

- **Comparison with Previous Experiment:**
  - The `Best_Cs` values for noMNIST also differ from those in the previous experiment.
  - LogisticRegressionCV, with its automatic η optimization, seems to converge to larger `Best_Cs` values compared to the manual experimentation in the previous experiment.

4.5.3. Overall Insights

- **Automatic η Optimization:**
  - LogisticRegressionCV's ability to automatically optimize η appears to influence the selection of `Best_Cs` compared to manual experimentation.
  - The differences may arise from the algorithm's adaptability to dataset characteristics, allowing it to choose more nuanced regularization parameters.
- **Consistency in CCR:**
  - The stability of CCR across different r values suggests that the radius parameter has a limited impact on the model's performance for the chosen datasets.
- **Consideration for Future Experiments:**
  - The insights gained highlight the importance of considering the optimization approach when selecting regularization parameters.
  - Further investigation could focus on understanding the interplay between automatic optimization strategies and dataset-specific characteristics.

5. Fourth Experiment

For one of the classification problems, run the script considering the problem as if it was regression (i.e. the classification parameter is False and compute the CCR rounding the predictions to the closest integer). What is happening for this situation?

5.1. Datasets Description

In this section, we provide a brief overview of the datasets considered this experiment with our program.

5.1.1. ProPublica COMPAS Dataset

The ProPublica COMPAS dataset focuses on the performance of the COMPAS algorithm in the U.S. criminal justice system. It analyzes risk scores for risk of recidivism with binary outcomes indicating rearrest within two years. Attributes include sex, age, age category, juvenile felony/misdemeanor counts, prior counts, race, and charge degree.

5.2. Experimental Setup

- To investigate the behavior of the program when considering a classification dataset as a regression dataset, Experiment 4 was designed using the `experiment4.py` script, available in the `experiments` folder within the compressed file.
- The main focus of Experiment 4 is on assessing the impact of different parameters (r) on the performance of the program when treating classification datasets as regression datasets.
- The dedicated script, `experiment4.py`, systematically explores various r values and reports the training Correct Classification Rate (CCR) as part of its output.
- Additionally, the script introduces a small η value to observe potential changes in behavior under different conditions.
- Detailed information about the experiment, including the script and results, can be found in the provided compressed file, ensuring transparency, reference, and reproducibility.
- Through this study, we aim to understand how different changing parameters impact the program's behavior when handling classification datasets as regression datasets, providing insights for practical use cases.

5.3. Results Obtained

This subsection delves into the outcomes derived from Experiment 4, where the radius parameter (r) is systematically manipulated to evaluate its influence on logistic regression models. With a particular focus on the ProPublica COMPAS dataset, the results highlight the Classification Correct Rate (CCR), providing valuable insights into the model's performance.

Dataset	Regularization (r)	Training CCR (%)	Test CCR (%)
$\eta = 10^{-4}$	0.05	70 % ± 0 %	67 % ± 0 %
	0.15	73 % ± 0 %	62 % ± 0 %
	0.25	75 % ± 0 %	56 % ± 0 %
	0.5	78 % ± 0 %	52 % ± 0 %
$\eta = 10$	0.05	70 % ± 0 %	67 % ± 0 %
	0.15	73 % ± 0 %	62 % ± 0 %
	0.25	75 % ± 0 %	56 % ± 0 %
	0.5	78 % ± 0 %	52 % ± 0 %

Tabla 9: Experiment 4 Results: Impact of Radius (r) on CCR for Classification/Regression

5.4. Analysis of Results

Experiment 4 explores the impact of varying the radius parameter (r) on logistic regression models when considering a classification dataset as a regression dataset. The ProPublica COMPAS dataset is used as a representative case. The results, displayed in Table 9, showcase the Classification Correct Rate (CCR) for different r values, emphasizing both training and test performance.

5.4.1. Effect of r on CCR

ProPublica COMPAS Dataset: As r increases from 0.05 to 0.5, there is a consistent improvement in both training and test CCR. This suggests that increasing the radius parameter positively influences the model's ability to capture the underlying patterns in the dataset. The highest CCR is achieved when r is set to 0.5, reaching 78

5.4.2. Impact of Best Parameters

Best Parameters: The exploration of best parameters, denoted as `best_Cs`, is performed to identify the most effective configuration. As expected, the best parameters do not vary nothing across different r values. This observation indicates that, in the context of regression, the choice of η parameters may not influence the model's performance, as evidenced by consistent CCR values.

5.4.3. Consistency in CCR

Consistency Across r Values: Remarkably, the CCR values remain consistent across different r values, both in training and testing. This consistency suggests that, for regression scenarios, the regularization and η parameters may not play a prominent role in influencing the model's predictive capacity. The lack of significant variation in CCR aligns with expectations for regression tasks, where the focus is on predicting continuous values rather than discrete classes.

5.5. Conclusion

Experiment 4 highlights the robustness of logistic regression models in regression scenarios, where the classification dataset is treated as a regression dataset. The observed consistency in CCR values across varied r and best parameters underscores the resilience of the model in capturing underlying trends without being significantly affected by changes in regularization or η parameters. These findings provide valuable insights for practitioners working on regression tasks using logistic regression models.

6. Conclusion

In conclusion, delving into the intricacies of logistic regression has been an enlightening journey, offering valuable insights into its behavior across diverse datasets and parameters. Logistic regression stands out as a widely used algorithm, appreciated for its simplicity and efficacy in handling classification tasks. Through the conducted experiments, we have explored the impact of parameters such as regularization strength, radius, and learning rate.

Unfortunately, due to time constraints, a comprehensive exploration of logistic regression's extensive parameter space was not feasible within the confines of this study. However, the envisioned continuation involves implementing scripts with nested loops to systematically test a wide range of parameters, aiming to identify optimal configurations. This approach is expected to reveal hidden patterns and discern the most effective parameter combinations for a variety of datasets.

As logistic regression remains a prevalent tool across different domains, this study serves as a foundational step for future investigations. The proposed avenue of further exploration could offer a more exhaustive understanding of the algorithm's behavior and enhance its adaptability to diverse scenarios.

In essence, while this study provides valuable insights, it represents only a preliminary exploration of logistic regression's potential. Future work, building upon the foundation laid here, holds the promise of uncovering more intricate details and refining the application of logistic regression in various real-world situations.

7. Bibliografía

- Curso Introducción a los Modelos Computacionales
- Redes de funcion de base radial (RBF)
- Red Neuronal de función de base radial (RBF-NN) (Teoría e implementación)
- radial-basis-function-network