Report for OMDS project

Part1

Develop a neural network from scratch

Omid Ghorbani

Student number: 2100649

1. the final setting for selected non-linearity, L, N, and λ; how and why did you choose them;

- for numerical stability RELU activation function has been used. It performs better than TANH.
- Base on the normal term of lambda and N and L some different architecture defined and then let the cross validation to choose the best between them.

2. which optimization routine you used for solving the minimization problem, the setting of its parameters (max number of iterations, etc.) and the returned message in output (successful optimization or others, number of iterations, starting/final value of the objective function) if any;

The code uses **gradient-based optimization** to minimize the cost function during the training of a neural network model. (Gradient descent algorithm)

iters = 1500 for training the final model after selecting the best hyperparameters.

2.1 Cost Function Improvement

Starting value of objective function: 788

Final value of objective function: 60.23

3.The initial and final values of the regularized error on the training set; the value of the error on the validation set (avg of errors on the k-folds); the final value of the test error.

Error for training set:

1: Initial value is: 96.16 %

2- Final value is: 23.55%

Error for validation set

1- Initial value is: 97.18 %

2- Final value is: 23.81 %

Final value of Test error: 26.91 %

Test cost: 79.40

4. the initial and final MAPE (average of MAPE values obtained with k-fold) on the training set, the final MAPE on validation and test set

number of iteration for K-fold was 800 with k = 5

Average Training MAPE: 27.83%

Average Validation MAPE: 27.84%

Initial MAPE for train: 96.32 %

Final MAPE for train: 28.63 %

Initial MAPE for Validation: 96.35 %

Final MAPE for Validation: 28.72 %

Table1

Performance				
Final train loss	Final test loss	Final train MAPE	Final Test MAPE	Optimization Time
60.23	**23.86 %**	**23.55 %**	**23.81%**	**30m**

Table2

Setting								
Hidden layer 1		Hidden layer 2		Hidden layer 3		Hidden layer 4		Other
No Neurons	Non linearity	No Neurons	Non linearity	No Neurons	Non linearity	No Neurons	Non linearity	Regularization term
32	RELU	80	RELU	60	RELU	20	RELU	L2

Part2/Solve a Dual SVM For Gender Classification

Only for Question 2: the final setting for the hyperparameter C (upper bound of the constraints of the dual problem) and of the hyperparameter of the kernel chosen; how you have chosen them and if you could identify values that highlight over/underfitting;

hyperparameter tuning using grid search method used to evaluate the best parameter for solving the problem and it turn out C = 1 and Gaussian kernel work the best. Since the C value of 1 is moderate, I believe the model is neither overfit nor underfit.

Only for Question 2: which optimization routine you use for solving the quadratic minimization problem and the setting of its parameters, if any (write DEFAULT if you have not changed them);

cvxopt library used to solve the qp problem. cvxopt uses numerical methods (like the interior-point method) to find the optimal values of the Lagrange multipliers that maximize the dual objective of the SVM.

Machine learning performances: write the value of the accuracy on the training and test set; (for question 2, write also the value of the validation accuracy).

Best Cross-Validation Accuracy q2: 0.918

Training Set Accuracy Q2: 0.92

Test Set Accuracy Q2: 0.92

Training Accuracy q3: 91.00%

Test Accuracy q3: 90.33%

Optimization performance: report the initial and final value of the objective function of the dual problem and the number of iterations, either as it is returned by the optimization routine used or evaluated by yourselves.

Q2 initial: 0

Q2 final: 126.15

Q3 initial: 0

Q3 final: 326

	Hyperparameters			**Ml PERFORMANCE**		**Optimization performance**	
QUESTIONS	**Kernel**	C	**P(or γ**	**Train Accuracy**	**Test Accuracy**	**Number of Iterations**	**CPU Time**
Q2	Gaussian	**1**	**0.5**	**92**	**92**	**-**	**81 Seconds**

Q3	**Gaussian**	**10**	**0.5**	**91**	**90.33**	**15400**	**6.03 seconds**
Q4	**Gaussian**	**1**	**0.5**	**94.76**	**89.56**	**-**	**8 seconds**