COMP24112 - Machine Learning Coursework Report

Ayush Gupta

7 May 2024

1 Gradient of the Objective Function

To derive our Loss function given below:

$$O = C\sum_{i=1}^{N} \max\left(0, 1 - y_i(w^T x_i + w_0)\right) + \frac{1}{2}w^T w$$

The derivative can be broken down to sum of two derivatives- one for the Hinge Loss Expression -

$$C\sum_{i=1}^{N} \max\left(0, 1 - y_i(w^T x_i + w_0)\right)$$

and another for the regularization term. The derivative of regularization term can be given as following

$$\nabla_w \left(\frac{1}{2}w^T w\right) = w$$

Now, to derivate the Hinge Loss Function, lets examine the term inside the summation:

$$\max(0, 1 - y_i(w^T x_i + w_0))$$

The above expression totally depends on each individual data point, hence there are two cases:
A)When Data Point is Correctly Classifed, in this case the max function just returns a constant 0. Hence the derivative also goes to 0.
B) When Data Point is Missclassifed - $(1 - y_i(w^T x_i + w_0)) > 0$

For the second case the max function returns $(1 - y_i(w^T x_i + w_0))$, which if we augment the x_i with one more attribute with value 1 and have w_0 as augmented into w, so new expression looks like - $(1 - y_i(\tilde{w}^T \tilde{x})) > 0$ Also lets define an indicator function $\text{Ind}(xi, yi)$ which returns 1 if $(1 - y_i(\tilde{w}^T \tilde{x})) > 0$ else 0.

Using this, the derivative for Hinge Loss becomes:

$$\nabla_w \left(HingeLoss\right) = C \sum_{i=1}^{N} (-y_i \tilde{x_i} * Ind(\tilde{x_i}, y_i))$$

Combining both derivatives, we obtain the gradient of the objective function with respect to $\tilde{w}$:

$$C \left(\sum_{i=1}^{N} -y_i \tilde{x_i} * Ind(\tilde{x_i}, y_i) \right) + \tilde{w}$$

2 Interpretation of Classification Accuracies

In Section 2.1, the cost function demonstrated a rapid decrease during training, ultimately stabilizing at a minimum value.

Simultaneously, the accuracy plot depicted noticeable enhancements in the model's predictive capabilities. Notably, achieving a 97% accuracy on unseen test data signifies the model's proficiency in making accurate predictions. Furthermore it highlights that the model avoided overfitting and the regularization parameter was chosen appropriately.

3 Learning Rate Analysis

At the beginning of training, with a relatively low learning rate of approximately 0.001, both the training and test accuracies peaked at around 97%. As the learning rate increased we observed a steady decline in both training and test accuracies. This decline suggests that higher learning rates decreased the model's ability to generalize to unseen data.

3.1 Imapct on Training

Lower learning rates allow for smoother convergence towards the optimal solution but may require more iterations to train the model effectively. Conversely, higher learning rates may lead to faster convergence but risk overshooting the optimal solution and destabilizing training process.

3.2 Effect on Model Performance during Testing

In our case, the decline in test accuracy with higher learning rates underscores the importance of selecting an appropriate learning rate to achieve satisfactory performance on unseen data. Opting for slightly lower learning rates tends to yield better training outcomes, albeit requiring more time for model convergence and training.

4 Interpreting Model Selection Results

ReLU being the chosen activation function indicates the model's effectiveness to accelerate convergence during training and allowing it to capture complex patterns in the data more effectively, because of the non-linearity introduced by ReLU. The selection of (100, 100) indicates a relatively deep network with two hidden layers, each consisting of 100 neurons. Deeper networks have the capacity to learn more intricate representations of the input data.

5 ADAM Vs SGD

Adam with adaptive learning rate shows irregular fluctuations in the cost plot during training, suggesting rapid adjustments in learning rates, which could lead to faster convergence. SGD exhibits smoother progression in the cost plot, albeit potentially slower convergence compared to Adam. Adam initially outperforms SGD in test data accuracy, but both eventually converge to similar levels.

6 Model Development

The model pipeline utilizes an imputer to handle missing values by replacing them with the column mean. StandardScaler is then applied to standardize features, removing the mean and scaling to unit variance. The main model, MLPRegressor, considers hyperparameters such as activation function, solver, regularization parameter (alpha), and number of hidden layers. GridSearchCV conducted an exhaustive search with 5-fold cross-validation over a specified parameter grid to determine the optimal hyperparameter combination.