NYCU Introduction to Machine Learning, Homework 4

[112550198], [簡嫚萱]

Part. 1, Kaggle (70% [50% comes from the competition]):

(20%) Introduction of your idea, methods, and key to success

Create a 3-5 page slide (e.g., MS PowerPoint, Google Slides, etc.) with no title or thank-you page to introduce your work.

Things you should cover - Include and not limit to

- How do you process the data? Have you done any special processing that significantly boosted performance?
- What is your model architecture? Do you make any changes/modifications to the model? Does it improve performance?
- How do you train the model? Do you use any special techniques (e.g., ensembles or other methods) to improve performance?
- Other details you want to mention that improve the performance
- Paste the snapshot of your Kaggle public leaderboard as an appendix (Not count to the page limit)

Hint:

1. Make your slide presentation clear and informative, and TAs will evaluate its completeness and content.
2. Read some literature to see how they report their method and results.

The slide is submitted with other files.

Part. 2, Questions (30%):

1. (10%) Based on the "SVMs vs. Logistic regression" lecture slide, explain which kinds of training data points mainly determine the classifier learned by SVM and which types of points influence Logistic Regression, and briefly justify your answer by referring to the shapes of hinge loss and logistic loss.

   SVM: The classifier learned by SVM is mainly determined by the **support vectors**, namely, the training data points that lie or inside the margin.
   The hinge loss is $E(y,\ f(x))\ = max(0,\ 1\ -\ yf(x))$. That is, for points with $yf(x)\ \geq\ 1$ (correctly classified and outside the margin), the loss is equal to zero. Therefore, these points don't contribute to the function, and only the points on or inside the margin (the support vectors) affect the learned classifier.
   Logistic regression: The classifier learned by Logistic Regression is influenced by **all training data points**, including those that are far from the decision boundary.
   The logistic loss is $E(y,\ f(x))\ =\ log(1\ +\ exp(-\ yf(x)))$ is strictly positive to all points, although it decreases smoothly as $yf(x)$ increases. As a result, every training data point contributes to the loss, so all data points affect the learned model.

2. (15%) For an SVM with a Linear Kernel, determine whether to use the Primal or Dual Form for the datasets below. Justify your choice based on: (a) Optimization variables & computational complexity, (b) Memory requirements (specifically the size of the Gram Matrix), (c) Prediction cost.
   - Dataset A: N=100, M=20,000
   - Dataset B: N=1,000,000, M=20

Dataset A:

(a) Optimization variables & computational complexity
In the primal form, the number of optimization variables equals the feature dimension $M = 20000$, while in the dual form it equals the number of training samples $N = 100$. Since $N \ll M$, the dual optimization problem is much smaller and computationally more efficient.

(b) Memory requirements (Gram matrix)
The dual formulation requires storing the Gram matrix of size $N \times N = 100 \times 100$, which is very small and easily fits in memory.

(c) Prediction cost
Prediction in the dual form depends on the number of support vectors. Since the number of support vectors is at most $N = 100$ and typically much smaller, the prediction cost is acceptable.

As a result, for Dataset A, the **dual form** is preferred because it involves fewer optimization variables and a small Gram matrix.

Dataset B:

(a) Optimization variables & computational complexity
In the primal form, the number of optimization variables equals the feature dimension $M = 20$, whereas the dual form would require optimizing $N = 10^6$ variables.
Thus, the primal optimization problem is significantly smaller and more efficient.

(b) Memory requirements (Gram matrix)
The dual formulation requires storing a Gram matrix of size $N \times N = 10^{12}$ entries, which is infeasible in practice due to extreme memory requirements.

(c) Prediction cost
Prediction in the primal form only requires computing an inner product $w^T x$ with complexity $O(M)$. Since $M = 20$, prediction is very fast and scalable.

Therefore, for dataset B, the **primal form** is preferred because it avoids the infeasible Gram matrix and enables efficient training and prediction.

3. (5%) To train a neural network, what do we need to optimize it? (How do we know the network is good or not?) Also, what algorithm can we use to optimize the neural network? (the most basic one).

What we have to optimize is the **loss function**, which measures the difference between the predictions generated by the network and the ground truth. A smaller loss value indicates that the predictions from the network are closer to the true targets, meaning the network performs better.
The most basic algorithm for optimizing a neural network is Gradient Descent. Gradients of the loss with respect to the network parameters, such as weights and biases, are computed using backpropagation. The parameters are then updated in the direction that minimizes the loss.
The update rule is $w^{new} = w^{old} - \eta \nabla E$, where ∇E is the gradient with respect to w and η is learning rate.