NYCU Introduction to Machine Learning, Homework 4

112550077, 劉逢穎

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 url of my google slides :

https://docs.google.com/presentation/d/1zSBXXxXvrAwBGtLCGgfByegQK0JBRu9rMYQHYNul Ur4/edit?usp=sharing

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.

   Type your answer here.

   SVM:

   Only data points near or on the wrong side of the margin boundary mainly determine the classifier. The hinge loss is zero for correctly classified points with sufficient margin (when $y \cdot f(x) \geq 1$). Points far from the decision boundary contribute nothing to the loss and don't influence the final classifier.

   Logistic Regression:

   All training data points influence the classifier, though with varying degrees. The logistic loss never reaches zero - even correctly classified points far from the boundary still contribute a small positive loss. The logistic loss decays exponentially but continuously as confidence increases, meaning distant points still have gradients that affect the optimization.

Hinge loss: max(0, 1 - y·f(x)) => Creates a "dead zone" where loss = 0
Logistic loss: log(1 + exp(-y·f(x))) => Always positive, asymptotically approaches zero but never reaches it.

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

Type your answer here.
For dataset A

(a) Optimization variables & computational complexity:
Primal: Optimize over M weights → O(M) = O(20,000) variables
Dual: Optimize over N Lagrange multipliers → O(N) = O(100) variables
Dual optimization (SMO/QP) is extremely efficient when N is small

(b) Memory requirements :
Gram Matrix size N × N = 100 × 100 = 10,000 entries
While the Gram matrix is small, we don't need it at all in primal form
Primal stores weight vector M = 20,000 entries
Dual is memory-efficient because N is small.

(c) Prediction cost :
Primal: O(M) = O(20,000) - dot product with weight vector
Dual: O(N·M) = O(100·20,000) = O(2M) - requires summing over support vectors

Conclusion: We should choose Dual Form for dataset A.

For dataset B

(a) Optimization variables & computational complexity
Dual variables: N = 1,000,000 → impossible for QP/SMO.
Primal variables: M = 20 → extremely small and easy to optimize using SGD or liblinear.
The Dual form is computationally infeasible; Primal is ideal.

(b) Memory: Gram Matrix
Gram matrix size = 1,000,000 × 1,000,000 = 10^{12} entries
Storage in double precision = 8 TB (completely impractical).
Dual form is impossible due to Gram matrix size; Primal uses almost no memory.

(c) Prediction cost
Primal prediction:$w^{\top}x$
Cost = O(M) = 20 operations → extremely fast.
Dual prediction: $\sum_{i \in SV} \alpha_i \, y_i K(x_i, x)$

Would require summing over potentially thousands of support vectors → inefficient.

Conclusion: We should choose Primal Form for dataset B.

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).

Type your answer here.

We need to optimize a loss function such as cross-entropy loss that measures how well the network's predictions match the true labels on the training data. The loss tells us how good or bad the network is - lower loss means better performance.
The most basic optimization algorithm is Stochastic Gradient Descent.
First, it compute the gradient of the loss with respect to network parameters using backpropagation, and updating weights in the opposite direction of the gradient: $w \leftarrow w - \eta \cdot \nabla L(w)$. Then, repeat the process until convergence.