Homework 1 for #80245013, "Machine Learning"

Instructor: Prof. Jun Zhu

~~October 13, 2018~~

Requirements:

- We recommend that you typeset your homework using appropriate software such as LaTeX. If you submit your handwritten version, please make sure it is cleanly written up and legible. The TAs will not invest undue effort to decrypt bad handwritings.
- We have programming tasks in each homework. Please submit the source code together with your homework. Please include experiment results using figures or tables in your homework, instead of asking TAs to run your code.
- Please finish your homework independently.

1 Mathematics Basics

1.1 Optimization

Use the Lagrange multiplier method to solve the following problem:

$$\begin{aligned} \min_{x_1,x_2} \quad & x_1^2 + x_2^2 - 1 \qquad (1) \\ s.t. \quad & x_1 + x_2 - 1 = 0 \\ & x_1 - 2x_2 \geq 0 \end{aligned}$$

1.2 Stochastic Process

We toss a fair coin for a number of times and use H(head) and T(tail) to denote the two sides of the coin. Please compute the expected number of tosses we need to observe a first time occurrence of the following consecutive pattern

$$H, \underbrace{T, T, \cdots, T}_{k}.$$

2 SVM

Consider the regression problem with training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ ($\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$). $\epsilon > 0$ denotes a fixed small value. Derive the dual problem of the following primal problem of linear SVM:

$$\begin{aligned}
\min_{\boldsymbol{w}, b, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad &+ \quad C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) \\
s.t. \quad y_i \quad &\leq \quad \boldsymbol{w}^\top \boldsymbol{x}_i + b + \epsilon + \xi_i \ , i = 1, \ldots, N \\
y_i \quad &\geq \quad \boldsymbol{w}^\top \boldsymbol{x}_i + b - \epsilon - \hat{\xi}_i \ , i = 1, \ldots, N \\
\xi_i \quad &\geq \quad 0 \quad \forall\, i = 1, \ldots, N \\
\hat{\xi}_i \quad &\geq \quad 0 \quad \forall\, i = 1, \ldots, N
\end{aligned}$$

3 Deep Neural Networks: Have a Try

To make neural networks work well in practice is not easy in general, since there are too many hyper-parameters to tune such as the choice of the number of hidden layers, the activation function, the learning rate and so on. Besides some general guidelines (some standard techniques which are useful at most cases such as dropout, data augmentation), experience is of great importance.

Though a beginner may often be confused with them, luckily, there are some softwares available on the internet to help you build up a good sense on tuning neural networks.

In this problem, you need to train the neural networks with different choices of hyper-parameters from the following link - A Neural Network Playground (you may need a VPN) - and answer the following questions:

1. Identify the best configuration you find for different problems and datasets. Here you only need to list you configuration for the bottom-right dataset of the classification problem.

2. List your findings that how the learning rate, the activation function, the number of hidden layers and the regularization influence the performance and convergence rate.

4 IRLS for Logistic Regression

For a binary classification problem $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ ($\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$), the probabilistic decision rule according to "logistic regression" is

$$P_{\boldsymbol{w}}(y|\boldsymbol{x}) = \frac{\exp(y\boldsymbol{w}^\top \boldsymbol{x})}{1 + \exp(\boldsymbol{w}^\top \boldsymbol{x})}. \tag{2}$$

And hence the log-likelihood is

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}) &= \log \prod_{i=1}^{N} P_{\boldsymbol{w}}(y_i|\boldsymbol{x}_i) && (3)\\ &= \sum_{i=1}^{N} \left(y_i \boldsymbol{w}^\top \boldsymbol{x}_i - \log(1 + \exp(\boldsymbol{w}^\top \boldsymbol{x}_i)) \right) && (4)\end{aligned}$$

Please implement the IRLS algorithm to estimate the parameters of logistic regression

$$\max_{\boldsymbol{w}} \ \mathcal{L}(\boldsymbol{w}) \tag{5}$$

and the L2-norm regularized logistic regression

$$\max_{\boldsymbol{w}} \ -\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \mathcal{L}(\boldsymbol{w}), \tag{6}$$

where λ is the positive regularization constant.

You may refer to the lecture slides for derivation details but you are more encouraged to derive the iterative update equations yourself.

Please compare the results of the two models on the "UCI a9a" dataset[1]. The suggested performance metrics to investigate are e.g. prediction accuracies (both on training and test data), number of IRLS iterations, L2-norm of $\|\boldsymbol{w}\|_2$, etc. You may need to test a range of λ values with e.g. cross validation for the regularized logistic regression.

[1] `http://ml.cs.tsinghua.edu.cn/~wenbo/data/a9a.zip`