STAT 535 Homework 4
Out November 2, 2021
Due November 9, 2021
©Marina Meilă
mmp@stat.washington.edu

Problem 1 – Logit loss and backpropagation - NOT GRADED
The *logit loss*

$$\hat{L}_{\text{logit}}(w) \;=\; \ln(1+e^{-yw^Tx}),\; x, w \in \mathbb{R}^d,\; y = \pm 1 \tag{1}$$

is the negative log-likelihood of observation (x, y) under the logistic regression model $P(y = 1|x, w) = \phi(w^T x)$ where ϕ is the logistic function.

a. Show that the partial derivatives $\frac{\partial \hat{L}_{\text{logit}}}{\partial w_i}$, $\frac{\partial \hat{L}_{\text{logit}}}{\partial x_i}$ for $\hat{L}_{\text{logit}}$ in (1) can be rewritten as

$$\frac{\partial \hat{L}_{\text{logit}}}{\partial w_i} \;=\; -(1 - P(y|x, w))yx_i \tag{2}$$

$$\frac{\partial \hat{L}_{\text{logit}}}{\partial x_i} \;=\; -(1 - P(y|x, w))yw_i. \tag{3}$$

The elegant formulas above hold for a larger class of statistical models, called Generalized Linear Models, as shown in Lecture II

Problem 2 – Logit loss Hessian
Assume that you have a data set $\mathcal{D} = \{(x^i, y^i), i = 1 : n\}, x^i, w \in \mathbb{R}^d$.

a. Calculate the expression of $\nabla^2 \hat{L}_{\text{logit}}$ for a single data point x. Simplify your result using $\phi(yw^T x)$ and its derivatives conveniently.

[b. – Not graded] Show that the gradient of $\hat{L}_{\text{logit}}(w; \mathcal{D})$ is a linear combination of the x^i vectors.

c. Show that if $n < d$ the Hessian of $\hat{L}_{\text{logit}}(w; \mathcal{D})$ has at least one 0 eigenvalue, and conclude that $\hat{L}_{\text{logit}}(w; \mathcal{D})$ is not strongly convex in this case.

d. – Optional, extra credit If $||x^i|| \leq R$, find a constant M sufficiently large so that $\nabla^2 \hat{L}_{\text{logit}}(w; \mathcal{D}) \prec MI_n$. Hence, $\hat{L}_{\text{logit}}$ is smooth.

Problem 3 – Ridge regression
In this problem you will perform ridge regression on the function $f^*(x) = 0.1x^2 + x + 1$ on $[0, 1]$. In the file `hw4_rr.dat` you will find a set of n (x^i, y^i) values with $y^i = f^*(x^i)$.

a Let $f(x) = \beta_0 + \beta_1 x$ be the predictor of y; β_0, β_1 will be estimated by Ridge Regression with regularization parameter λ. Denote $\beta_{0,1}(\lambda)$ the result of this estimation. Let the data matrix be the row vector $X = [x^1 \ldots x^n]$, and define the column vector $y = [y^1 \ldots y^n]^T$

Write the expressions of $\beta_0(\lambda)$, $\beta_1(\lambda)$ as functions of X, y, λ.

b Now choose a set of λ values including 0 and n. Calculate $\beta_{0,1}(\lambda)$, $\hat{L}_{LS}(\lambda)$ and $J(\lambda)$. Plot on the same graph $\beta_{0,1}(\lambda)$ vs λ.

c Plot on the same graph $\hat{L}_{LS}(\lambda)$ and $J(\lambda)$ vs λ. Comment on what you observe in the graphs of b, c.

Problem 4 – Descent algorithms for training a neural network
This problem asks you to train a neural network to classify the data sets given on the Assignments web page. The inputs are 2-dimensional, outputs are ± 1, one data point/line. *Submit the code for this problem.*

Objective to minimize is $\hat{L}_{\text{logit}}(\beta, W) = -\frac{1}{n}\text{log-likelihood}(\mathcal{D}|\beta, W)$ and $\beta \in \mathbb{R}^{m+1}, W \in \mathbb{R}^{(d+1)\times m}$ are the neural net parameters.

Algorithms: steepest descent with fixed step size. You need to implement the algorithm yourself. [Optional, for extra credit: implement Newton, or run Newton, LBFGS quasi Newton from library code.]

Dataset $\mathcal{D}$ given `hw4-nn-train-100.dat`

a. Plot the data set in $\mathbb{R}^2$, representing each class with a different color or symbol.

b. Based on the plot in **a.**, is it possible to get $\hat{L}_{01} = 0$ for $m = 2$? Explain.

c. Choose a number $m \geq 3$ hidden units and train the neural network on the $\mathcal{D}$. Obtain the best empirical $\hat{L}_{01}$ you can. *Note that larger m values may be easier to train.*

Explain how you chose the initial points. It's ok to plot the data and look at it or even to make a sketch of the solution you want to find. (If you implement more than one algorithm, start them all from the same initial point.)

The training algorithm will converge to a local optimum. It's OK to look at this local optimum and try other initial points if the found optimum is bad. (Don't forget to use the same initial point for all algos in the results you present in the homework.) It's also recommended to challenge the algorithm by giving it random/uninformative initial points. *Do not start all the parameters at 0 [Why?].*

Chose the stopping criterion $1 - \frac{\hat{L}^{k+1}}{\hat{L}^k} \leq tol$ with $tol = 10^{-4}$. If this tolerance cannot be reached in a reasonable number of steps, set a higher tol and report that value.

d. Describe briefly the implementation details of your algorithms. Size of the fixed step, number of iterations (and if it converged or not) and final value of loss functions $\hat{L}_{\text{logit}}$ and $\hat{L}_{01}$. Record also the time each algorithm takes and report it.

[Optional: If you used other algorithms report on those too. If you used line search, report if you bracketed the min or not in line search, what line search method you used (*you can use code from other sources to bracket the mininmum, and you can implement another line search method than Armijo.)*]

e. Estimate the value of L_{logit}, L_{01} by averaging them on the test set `hw4-nn-test.dat` for the

final classifier obtained. Optionally, compute these values at each iteration and plot them in the graphs for **f.**.

f. Plot the values of $\hat{L}_{\text{logit}}$, $\hat{L}_{01}$ and the respective costs on the test set vs the iteration number k. Make two separate plots for the two costs. If you have computed the test set costs at each iteration, plot these too on the respective graphs.

g. Plot the final decision region superimposed on the data.

[**h. Optional but encouraged**] Plot (some of) the β parameters vs k; on a separate plot, show trajectories of β parameters coming from different initializations.

Please make clear, well-scaled, well labeled graphs.