CS57800: Statistical Machine Learning

Homework 3

Yu-Jung Chou

chou63@purdue.edu

Due: Nov 13, 2018 on Tuesday

1 Open-Ended Question

1.1 Calculate the gradient of the loss function.

$$
\begin{aligned}
Err(w) &= -\sum_i \Big\{ y^i log\Big[g(w,x^i)\Big] + (1-y^i) log\Big[1-g(w,x^i)\Big] \Big\} \\
\frac{\partial Err(w)}{\partial w} &= -\sum_i \Big[y^i \frac{1}{g(w,x^i)} - (1-y^i)\frac{1}{1-g(w,x^i)} \Big] \frac{\partial}{\partial w} g(w,x^i) \\
&= -\sum_i \Big[y^i \frac{1}{g(w,x^i)} - (1-y^i)\frac{1}{1-g(w,x^i)} \Big] \Big[\frac{1}{1+e^{-wx^i}} \Big] \Big[\frac{e^{-wx^i}}{1+e^{-wx^i}} \Big] x^i \\
&= -\sum_i \Big[y^i \frac{1}{g(w,x^i)} - (1-y^i)\frac{1}{1-g(w,x^i)} \Big] g(w,x^i)(1-g(w,x^i)) x^i \\
&= -\sum_i \Big[y^i (1-g(w,x^i)) - (1-y^i) g(w,x^i) \Big] x^i \\
&= -\sum_i \Big[y^i - g(w,x^i) \Big] x^i
\end{aligned}
$$

1.2 Prove that the logistic loss function is convex.

A function is convex it is twice differentiable and $\nabla^2 f(x) \geq 0$

$$
\begin{aligned}
Err'(w) &= -\sum \Big[y - g(w,x) \Big] x \\
Err''(w) &= -\sum -\frac{\partial g(w,x)}{\partial w} x \\
&= \sum g(w,x) \Big[1 - g(w,x) \Big] x
\end{aligned}
$$

Since $0 \leq g(w,x) \leq 1$ and input $x \geq 0$, thus our logistic loss function $Err''(w) \geq 0$, which is convex.

1.3 What is regularization? Why is it used?

Regularization is a technique used to prevent overfitting and improve the generalizability of a learned model. It adds a penalty on different parameters of the model. Hence, the model will be less likely to fit the noise of the training data and will improve the generalization abilities of the model.

1.4 What will be the gradient of the loss function if you add L2 regularization term $\frac{1}{2}\lambda \parallel w^2 \parallel$ with the logistic loss?

According to the sum rule of derivatives, we can simply add the derivative of the L2 regularization term to the original gradient of the loss function we get from 1.1

$$\frac{\partial}{\partial w}\frac{1}{2}\lambda \parallel w^2 \parallel = \frac{\partial}{\partial w}\frac{1}{2}\lambda \parallel w \parallel^2 = \lambda \parallel w \parallel$$

The gradient of the loss function with L2 regularization:

$$-\Sigma[y - g(w, x)]x + \lambda \parallel w \parallel$$

1.5 In GD, you need to stop training when the model converges. How will you know that your model has converged? State the stopping criteria and apply the criteria throughout your implementation. Comment on the convergence in Stochastic Gradient Descent.

The model will converged when the gradient is equal to zero since our function is convex. In a convex function, the improvement will decrease as we are getting closer to the minimum of the convex function, thus a stopping criteria is set for us to stop at a point that the improvement is close to zero and hard to observe.

For the convergence in Stochastic Gradient Descent, it converges faster than Batch Gradient Descent. One reason is that Stochastic Gradient Descent updates after each training dataset, which is faster. Second, perhaps a local minimum is found rather than the global minimum.

1.6 What will be the effect of bias term in GD/SGD learning? Justify your answer. If you think it will be useful, use that in the implementation in the same manner you used it in case of perceptron.

The bias term in GD/SGD learning is the shift of the activation function in order to add flexibility to our model to fit the data better. It is set as an always on input and is adjusted by the corresponding weight.

2 Batch Gradient Descent with Logistic Function

2.1 Algorithm

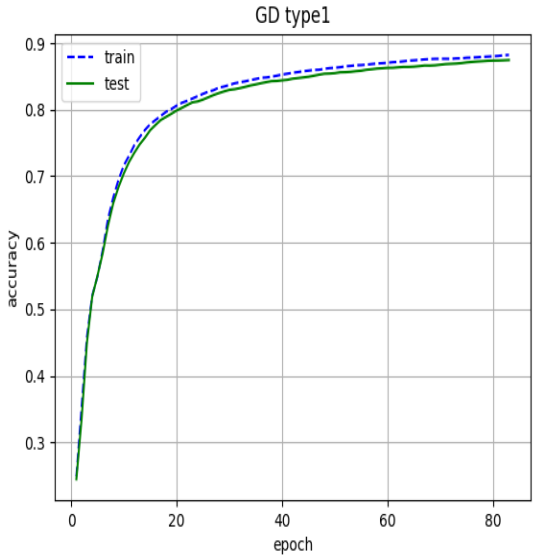
Algorithm 1: Batch Gradient Descent with Logistic Function $(D, Epoch, \alpha)$

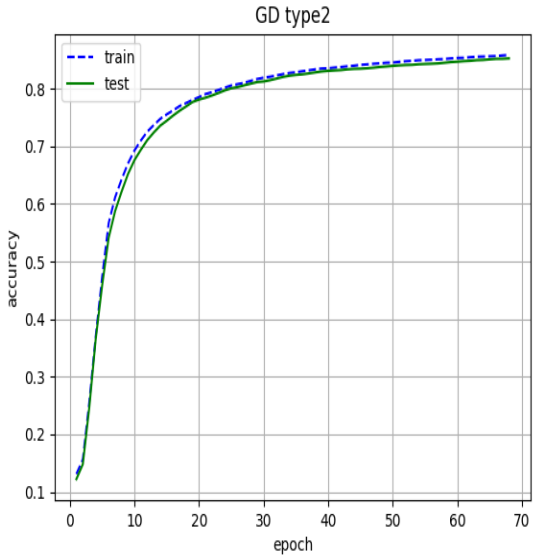
$w \leftarrow [Rand_0, Rand_1, ..., Rand_n]$;
for $i = 0$ .. $Epoch$ **do**
  $\Delta w = [0, 0, ..., 0]$;
  for $(x, y) \in D$ **do**
    $z = w^T x$;
    $h = \frac{1}{1+e^{-z}}$;
    $GD = (y - h) * x$;
    $\Delta w \leftarrow \Delta w + \alpha * GD$
  $w \leftarrow \Delta w + w$
return w

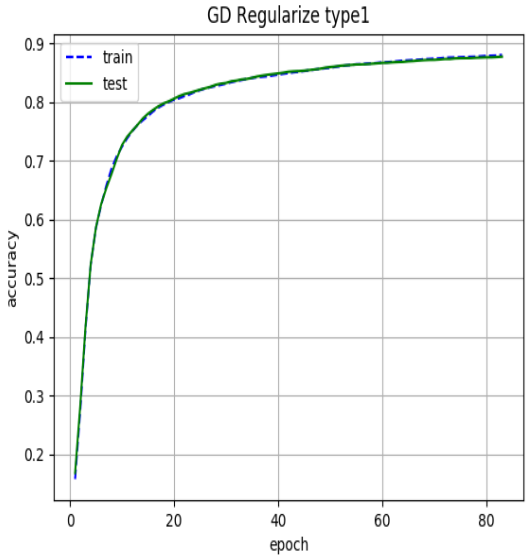
Algorithm 2: BGD Classification (X)

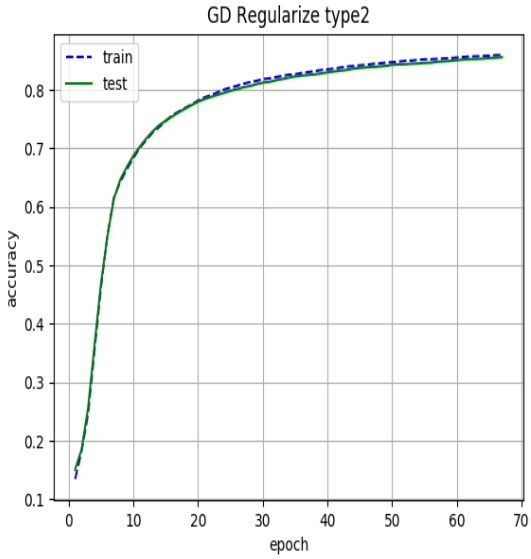
$z = w^T x$;
$h = \frac{1}{1+e^{-z}}$;
return $argmax(h)$

2.2 Result of Implementation









2.3 Analysis

Comparing type2 and type1, type2 has a slightly lower accuracy but converges faster with less epochs.

Comparing with and without regularization, with regularization will help prevent overfitting, but our model is not overfitting, thus the regularization won't make a big difference in our case.

3 Stochastic Gradient Descent with Logistic Function

3.1 Algorithm

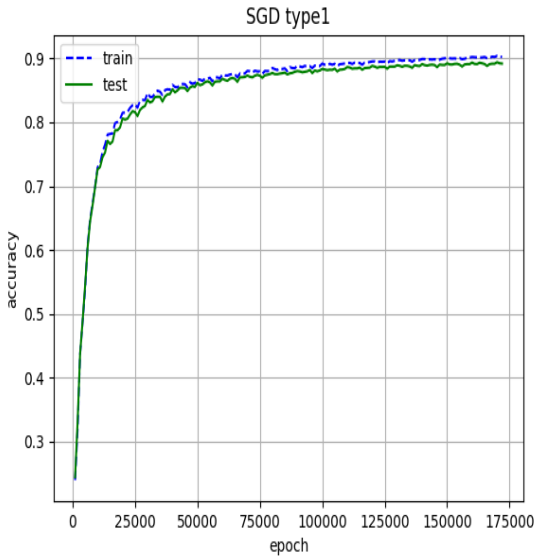
Algorithm 3: Stochastic Gradient Descent with Logistic Function $(D, Epoch, \alpha)$

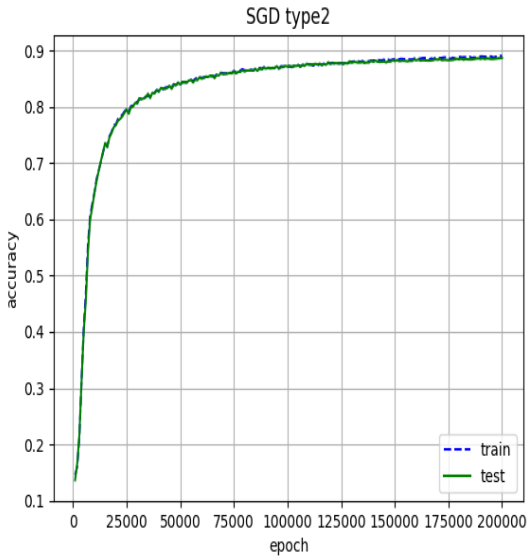
$w \leftarrow [Rand_0, Rand_1, ..., Rand_n]$;
for $i = 0$ .. $Epoch$ **do**
  for $(x, y) \in D$ **do**
    $z = w^T x$;
    $h = \frac{1}{1+e^{-z}}$;
    $GD = (y - h) * x$;
    $w \leftarrow w + \alpha * GD$
return w

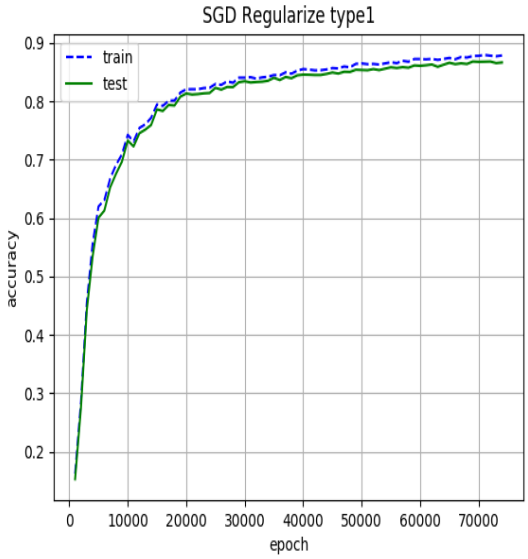
Algorithm 4: SGD Classification (X)

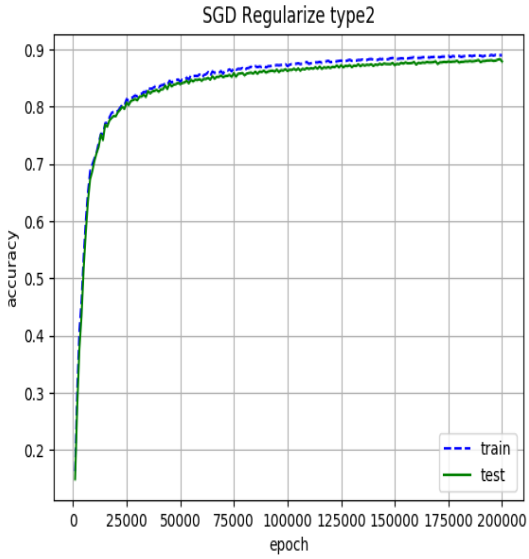
$z = w^T x$;
$h = \frac{1}{1+e^{-z}}$;
return $argmax(h)$

3.2 Result of Implementation









3.3 Analysis

Comparing to Batch Gradient Descent, Stochastic Gradient Descent converges faster.

Comparing type1 and type2, type1 converges faster with less epochs.

Regularization is used to prevent overfitting, but our model is not overfitting, thus the regularization won't make a big difference in our case.