CS57800: Statistical Machine Learning
Homework 3

Ruoyu Wu
wu1377@purdue.edu

November 15, 2018

1 Open-ended Questions

1.1 Gradient of the loss function

1.1.1 Base Formula

$$z = w^T x \tag{1}$$

$$g(z) = \frac{1}{(1 + e^{-z})} \tag{2}$$

$$Err(w) = -\sum_i y^i log(g(w, x_i)) + (1 - y^i) log(1 - g(w, x_i)) \tag{3}$$

1.1.2 Derivative of Logistic Function

$$\begin{aligned}
\frac{d}{dx} g(z) &= \frac{d}{dx}\left(\frac{1}{1 + e^{-z}}\right) && (4)\\
&= \frac{-\frac{d}{dx}(1 + e^{-x})}{(1 + e^{-x})^2} && (5)\\
&= \frac{e^{-x}}{(1 + e^{-x})^2} && (6)\\
&= \left(\frac{1}{1 + e^{-x}}\right)\left(\frac{e^(-x)}{1 + e^{-x}}\right) && (7)\\
&= g(z)(1 - g(z)) && (8)
\end{aligned}$$

1.1.3 Derivative(Gradient) of Logistic Loss Function

$$
\begin{aligned}
\frac{\partial Err(w)}{\partial w_j} &= -\sum_i [y^i \frac{\partial}{\partial w_j} log(g(w,x_i)) + (1-y^i)\frac{\partial}{\partial w_j} log(1-g(w,x_i))] &(9)\\
&= -\sum_i [y^i \frac{\frac{\partial}{\partial w_j}(g(w,x_i))}{g(w,x_i)} + (1-y^i)\frac{\frac{\partial}{\partial w_j}(1-g(w,x_i))}{1-g(w,x_i)}] &(10)\\
&= -\sum_i [y^i \frac{\frac{\partial}{\partial w_j} g(w^Tx_i)}{g(w,x_i)} + (1-y^i)\frac{\frac{\partial}{\partial w_j}(1-g(w^Tx_i))}{1-g(w,x_i)}] &(11)\\
&= -\sum_i [y^i \frac{g(w^Tx_i)(1-g(w^Tx_i))\frac{\partial}{\partial w_j}(w^Tx_i)}{g(w,x_i)} + (1-y^i)\frac{g(w^Tx_i)(1-g(w^Tx_i))\frac{\partial}{\partial w_j}(w^Tx_i)}{1-g(w,x_i)}] &(12)\\
&= -\sum_i [y^i(1-g(w^Tx_i))x_{ij} + (1-y^i)g(w^Tx_i)x_{ij}] &(13)\\
&= -\sum_i [g(w^T,x_i) - y^i]x_{ij} &(14)
\end{aligned}
$$

1.2 Prove that the logistic loss function is convex

1.2.1 Prove that $-log(g(w,x_i))$ is convex

Theorem: Function convex $\iff$ its epigraph(i.e. the set of points lying above the graph of function) is convex.

$$
\begin{aligned}
A &= \{(w,t)| - log(g(w,x_i)) <= t\} &(15)\\
&= \{(w,t)|log(1+e^{-W^Tx_i} <= t\} &(16)\\
&= \{(w,t)|1+e^{-W^Tx_i} <= e^t\} &(17)\\
&= \{(w,t)|e^{-t}+e^{-W^Tx_i-t} <= 1\} &(18)
\end{aligned}
$$

Since $f(w,t) = e^{-t} + e^{-W^Tx_i-t}$ is convex, A is a sublevel set of $f(w,t)$, then A is a convex set. Since A is the epigraph of $-log(g(w,x_i))$, $-log(g(w,x_i))$ is convex.

1.2.2 Prove that $-log(1-g(w,x_i))$ is convex

Similarly, we can easily prove that $log(1-g(w,x_i))$ is convex function of w.

1.2.3 Prove that $Err(w)$ is convex

The positive linear combination of convex function is still a convex function. And we have proven that for each x_i, $log(g(w,x_i))$ and $log(1-g(w,x_i))$ are convex functions for w. Therefore, $Err(w)$ is a convex function for w.

1.3 What is regularization and why is it used

Regularization is a process(term) when learning a model. It is normally used to prevent overfitting.

1.4 The gradient with L2 regularization

$$Err(w) = -\sum_i [y^i log(g(w, x_i)) + (1 - y^i) log(1 - g(w, x_i))] + \frac{1}{2}\lambda ||w^2|| \tag{19}$$

$$\frac{\partial Err(w)}{\partial w_j} = -\sum_i [g(w^T, x_i) - y^i] x_{ij} + \lambda w_j \tag{20}$$

1.5 Stopping criteria for GD and SGD

For GD, the algorithm will stop when the accuracy of learned model on the validation set has not improved for 5 consecutive steps, or the max_iteration is reached. For SGD, the algorithm will stop when the accuracy of learned model on the validation set has not improved for 5 consecutive steps, or the max_iteration is reached.

1.6 Effect of bias term in GD/SGD

Bias allows us to shift the activation function to the left or right. If we have: $z = w^T x + b$, then we have $g(z) = \frac{1}{(1+e^{w^T x + b})}$. I think bias term is useful, especially in the case of one-layer neural network.

2 Batch Gradient Descent with Logistic Function

2.1

Algorithm 1 BatchGradientDescentLogistic

```
procedure GBDTraining(DATA, W, α, λ)
    if It is the first step then
        size_data ← sizeof(D)
        size_feature ← sizeof(W) + 1
        w_j ← 0, for all j = 1...size_feature.
        gradient_j ← 0, for j = 1...size_feature
    end if
    for (x, y) in D do
        gradient_j ← gradient_j + [g(w^T, x) − y]x_j, for all j = 1...size_feature
    end for
    gradient_j ← gradient_j / size_data + λw_j, for all j = 1...size_feature
    W_j ← W_j − α gradient_j, for all j = 1...size_feature
    return W
end procedure
procedure Train()
    devide DATA into D(for training) and V(for validation)
    while True do
        if Accuracy on V has not improved for consecutive 5 steps then
            break
        end if
        for Each clf in classifiers do
            clf.train(D)
        end for
    end while
end procedure
```

2.2 Graph

Blue lines indicate training curve and green lines indicate testing curve.

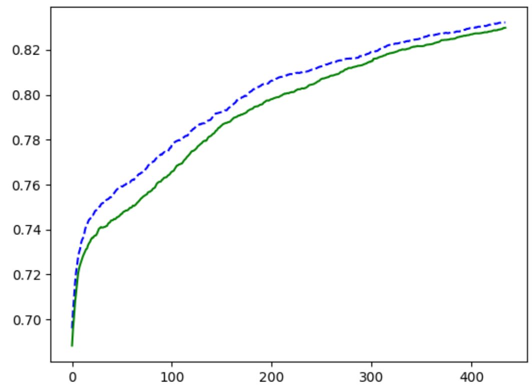

Figure 1: With regularization, type 1

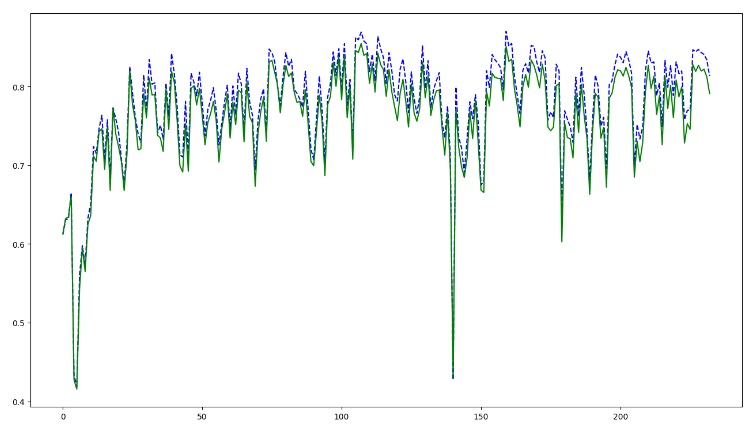

Figure 2: With regularization, type 2

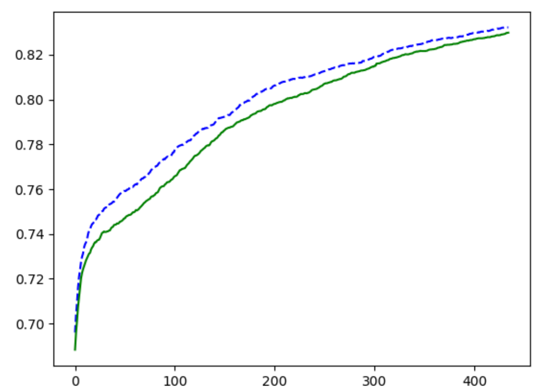

Figure 3: Without regularization, type 1

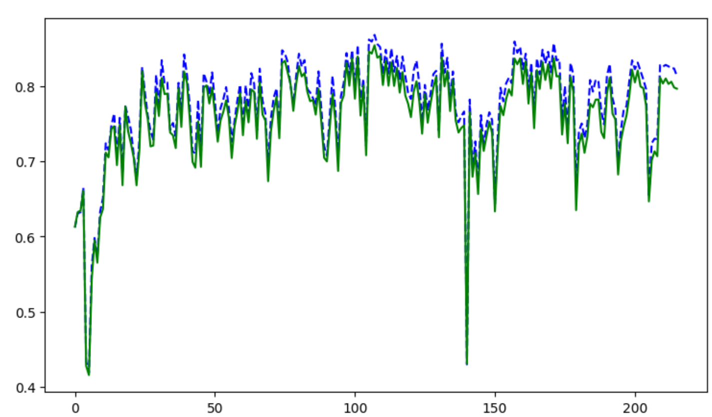

Figure 4: Without regularization, type 2

2.3 Analyze and compare

In terms of convergence speed, regularization term does not have huge impact. Type-2 converges much faster than type-1.

In terms of accuracy, regularization term does not have huge impact. Type-2.

It is because I set the factor of regularization term very small, so it does not make big difference. And type-2 can better represent the image because it filters some noises out and has much smaller feature space, so it achieves better converge speed and accuracy.

3 Stochastic Gradient Descent with Logistic Function

3.1

Algorithm 2 StochasticGradientDescentLogistic

```
procedure TRAINING(D, W, α, λ, SIZE_BATCH, MAX_ITERS)
    if It is the first step then
        size_data ← sizeof(D)
        size_feature ← sizeof(W) + 1
        w_j ← 0, for all j = 1...size_feature.
        gradient_j ← 0, for j = 1...size_feature
    end if
    S ← sample uniformly size_batch of data from D
    for (x,y) in S do
        gradient_j ← gradient_j + [g(w^T, x) − y]x_j, for all j = 1...size_feature
    end for
    gradient_j ← gradient_j / size_batch + λw_j, for all j = 1...size_feature
    W_j ← W_j − αgradient_j, for all j = 1...size_feature
    return W
end procedure
procedure TRAIN()
    devide DATA into D(for training) and V(for validation)
    while True do
        if Accuracy on V has not improved for consecutive 5 steps then
            break
        end if
        for Each clf in classifiers do
            clf.train(D)
        end for
    end while
```

3.2 Graph

Blue lines indicate training curve and green lines indicate testing curve.

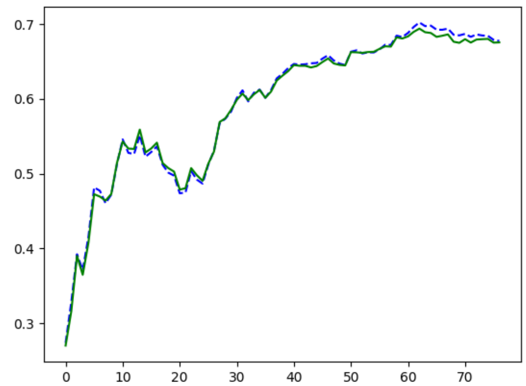

Figure 5: With regularization, type 1

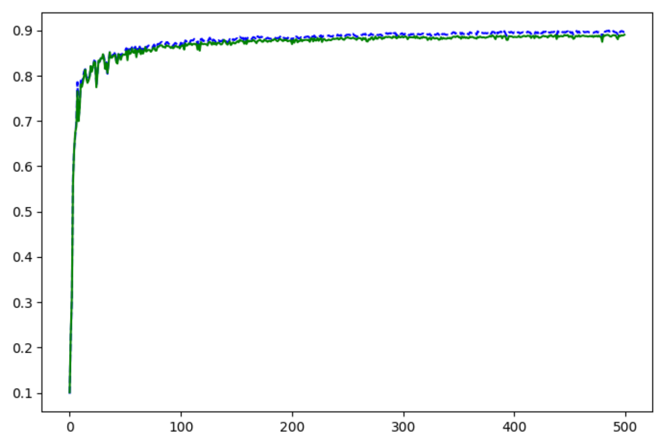

Figure 6: With regularization, type 2

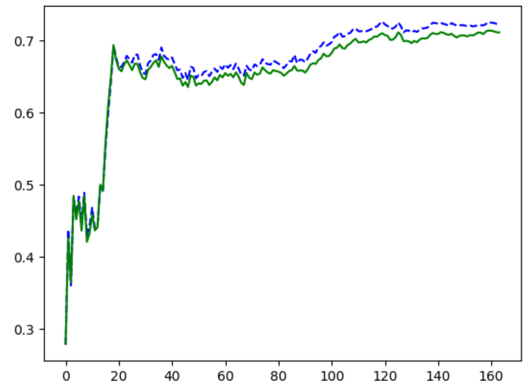

Figure 7: Without regularization, type 1

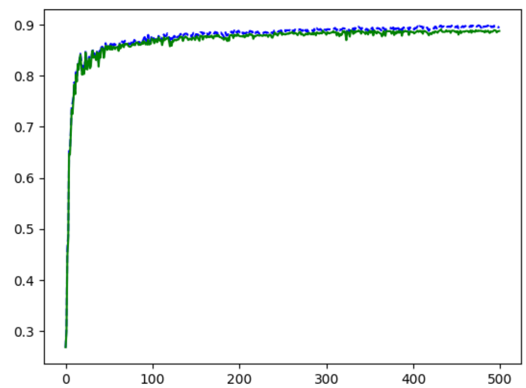

Figure 8: Without regularization, type 2

3.3 Analyze and compare

In terms of convergence speed, It is clear that my convergence condition is not good for the type-2 – they actually converge very fast(around 100 steps), and my convergence condition stop them at max_iter. Generally speaking, type-1 and type-2 converges roughly at the same speed. And with regularization term converges faster than withgout the term.

In terms of the accuracy, type-2 achieves better accuracy than type-1. Regularization term seems has no huge effect on accuracy in this setting.

It is because I set the factor of regularization term very small, so it does not make big difference. And type-2 can better represent the image because it filters some noises out and has much smaller feature space, so it achieves better converge speed and accuracy.