2025 Fall AI
Assignment 1

Guanheng Chen 2024011829

Q1

Q1.1

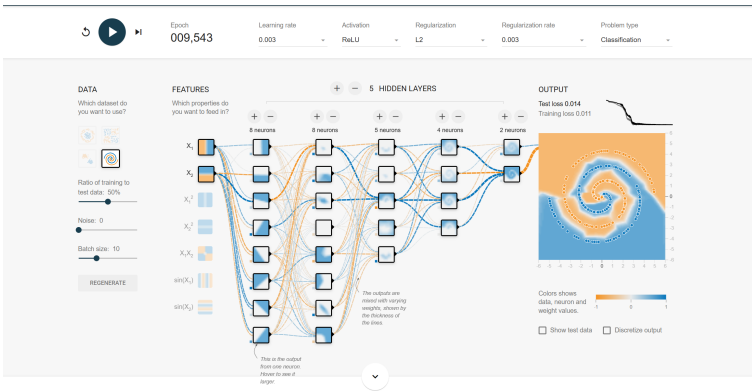

Figure 1: Config for classification

Q1.2

- Learning rate. Learning rate is the key hyperparameter that influence the performance and convergence rate. If Lr is too small, then convergence rate would be slow and trapped in local minimum, impede the good performance. If Lr is too large, we would suffer from severe oscillations and fail to converge. Therefore, we should pick the proper learning rate for our model to reach good convergence rate and performance.
- activation function. According to my test, Relu always achieve the best performance and Relu don't have 'vanishing gradient' problem as well. But Sigmoid and tanh both suffer from 'vanishing gradient problem'.
- the number of hidden layers. From my observation,if the classification task is simple, then small number of hidden layers(like 1or2) performs really well and has fast convergence. If the classification task is complex, then we should have more hidden layers in our models(like task plot in Q1.1), and it'll converge slower.
- Regularization. If their is no regularization, we would encounter overfitting problem which lead to bad performance in testing. Besides, L2 reg performs well and improves our generalization.

Q1.3

Yes, I do. The result is shown in the following figure.

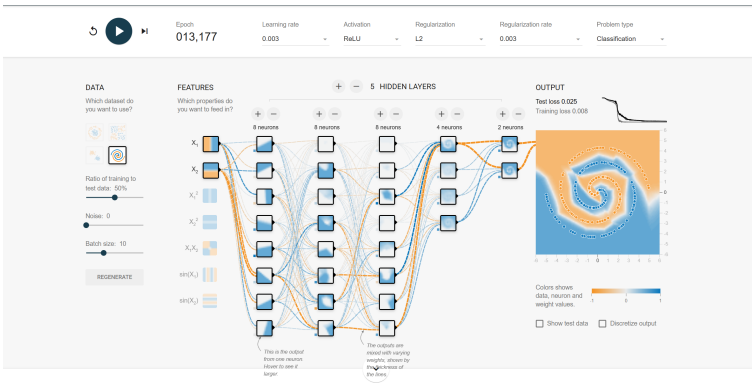

Figure 2: Config for classification

We could see, in the third hidden layer, there is 3 neurons with really low absolute value, so I cancel them and test again. By doing so, I get the result in figure1, a better model which less overfit and has better performance on test set.

Q2

Q2.1

By definition,

$$\theta^* = \arg\min_{\theta} \|y - X\theta\|_2^2.$$

which implies the following statement

$$\frac{\partial}{\partial\theta}\|y - X\theta\|_2^2 = -2X^\top(y - X\theta) = 0$$

equals to

$$X^\top(y - X\theta^*) = 0.$$

rearrange the left and right side, we get

$$\theta^* = (X^\top X)^{-1}X^\top y$$

let

$$r = y = X\theta^*.$$

We have

$$X^\top r = X^\top y - X^\top X\theta^* = 0.$$

Then we are done.

Q2.2

$\Rightarrow$ if there is a intersection in convex hulls, then $\exists z$, s.t

$$z = \sum_i \alpha_i x_i = \sum_i \beta_i y_i.$$

assume that there exist $\mathbf{w}$ and b such that

$$\mathbf{w}^\top \mathbf{x}_i + b > 0, \qquad \mathbf{w}^\top \mathbf{y}_i + b < 0 \quad \forall i.$$

By multipling a scalar to equations,

$$\mathbf{w}^\top (\alpha_i \mathbf{x}_i) + \alpha_i b \geq 0, \qquad \mathbf{w}^\top (\beta_i \mathbf{y}_i) + \beta_i b \leq 0 \quad \forall i.$$

By adding up both sides, we have

$$\mathbf{w}^\top \sum_i (\alpha_i \mathbf{x}_i - \beta_i y_i) > 0.$$

Contradictory!

$\Leftarrow$ if two sets are linearly separable, then for any

$$\alpha_i, \beta_i \geq 0 \qquad \forall i$$

By multipling a scalar to equations,

$$\mathbf{w}^\top (\alpha_i \mathbf{x}_i) + \alpha_i b \geq 0, \qquad \mathbf{w}^\top (\beta_i \mathbf{y}_i) + \beta_i b \leq 0 \quad \forall i.$$

By adding up both sides, we have

$$\mathbf{w}^\top \sum_i (\alpha_i \mathbf{x}_i - \beta_i y_i) \geq 0.$$

$=0$ holds iff $\alpha_i, \beta_i = 0 \forall i$, but it is not going to happen in convex hull, then we are done.

Q3

Q3.1

First, let us calculate the gradient, which is

$$\frac{\partial f(\theta)}{\partial \theta_k} = \frac{1}{N} \sum_{i=1}^{N} [-y^{(i)} x_k^{(i)} + \frac{x_k^{(i)} \exp(\theta^\top x^{(i)})}{1 + \exp(\theta^\top x^{(i)})}] + \lambda \theta_k.$$

We will update θ_k by following rule:

$$\theta_k \leftarrow \theta_k - \eta \left[\frac{1}{N} \sum_{i=1}^{N} [-y^{(i)} x_k^{(i)} + \frac{x_k^{(i)} \exp(\theta^\top x^{(i)})}{1 + \exp(\theta^\top x^{(i)})}] + \lambda \theta_k \right].$$

Q3.2

(1). Without loss of generality, we could suppose training data is separable in the first dimension $x_1^{(i)}$, the dividing boundary is β, then

$$x_1^{(i)} \theta_1 - \beta \theta_1$$

could help us split the training data, and when $\theta_1 \to \infty$, we have our $J(\theta) \to 0$.
(2).

- l_2. After adding L2 term, our gradient += $\lambda\theta_k$, when θ_k becoming bigger, our penalty becomes bigger as well, which impedes the weight scaling Uncontrollably.
- l_1.After adding L1 term, our gradient += $\lambda sign(\theta_k)$, similar to L2 term, we can prevent weight becoming to large by adding L1 term, and we also notice when $\theta_k < 1$ and close to zero, the sign function tend to push θ_k to zero.

Q4

Q4.1

1. By simple calculation, we have $j = -\frac{7}{8}$.
According to lecture, b is more likely to be the output because our filter tend to blur the original image.
2. The output size is (3,2). If we want to maintain the original size, we should do padding: left=right=2, top=bottom=3.
3. Provided that it is a black and white image, so there is a single channel, filter has no depth, so number of parameters is 16.

Q4.2

1. $\frac{\partial J}{\partial \hat{y}_1} = -\frac{2}{y_1 - \hat{y}_1}$.

2. First, note that:

$$\frac{\partial J}{\partial \hat{y}_2} = -\frac{2}{y_2 - \hat{y}_2}.$$

By using chain rule,

Gradient of z_1:

$$\frac{\partial J}{\partial z_1} = -\frac{2\,\hat{y}_1(1-\hat{y}_1)\,3m_{1,1}z_1^2}{y_1 - \hat{y}_1} - \frac{2\,m_{1,2}\cos(z_1)}{y_2 - \hat{y}_2}$$

Gradient of z_2:

$$\frac{\partial J}{\partial z_2} = \frac{2m_{2,2}\sin(z_2)}{y_2 - \hat{y}_2}$$

Gradient of $m_{1,1}$:

$$\frac{\partial J}{\partial m_{1,1}} = -\frac{2\,\hat{y}_1(1-\hat{y}_1)\,z_1^3}{y_1 - \hat{y}_1}$$

Gradient of $m_{1,2}$:

$$\frac{\partial J}{\partial m_{1,2}} = -\frac{2\sin(z_1)}{y_2 - \hat{y}_2}$$

Gradient of c_1:

$$\frac{\partial J}{\partial c_1} = -\frac{2\,\hat{y}_1(1-\hat{y}_1)}{y_1 - \hat{y}_1}$$

Q4.3

Gradient of $w_{2,2}$:

$$\frac{\partial J}{\partial w_{2,2}} = \frac{\partial J}{\partial z_2}x_2^2$$

Gradient of b_2:

$$\frac{\partial J}{\partial b_2} = \frac{\partial J}{\partial z_2}$$

Gradient of $w_{1,1}$:

$$\frac{\partial J}{\partial w_{1,1}} = \begin{cases} 0, & \text{if } w_{1,1}x_1 + w_{2,1}x_2 < 0 \\ \dfrac{\partial J}{\partial z_1} \cdot x_1, & \text{otherwise} \end{cases}$$

Q5

Q5.1

In the beginning stage(A in the figure), the mdoel learns useful features while executing gradient descent, but in second stage, the model tends to overfit, there should be some noise in our data, so our model might try its best to fit the noise which impede the generalization.

Q5.2

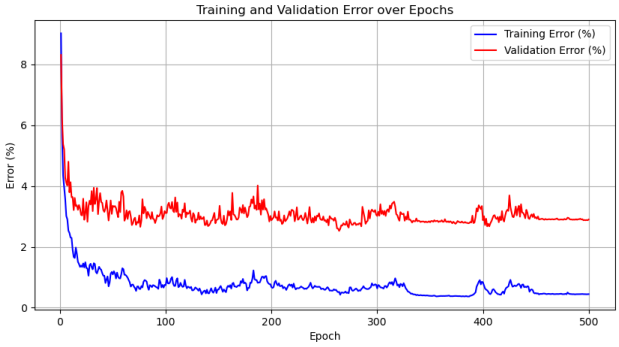

Figure 3: OneLayerNet

Q5.3

With experiments, I found that 0.04 is the best value for b. The training curve is shown in 4. I notice that to b=0.04, the oscillation phenomenon of train_erorr and val_erorr is weaker after 200 epoch, and b=0.04 converge slightly faster.

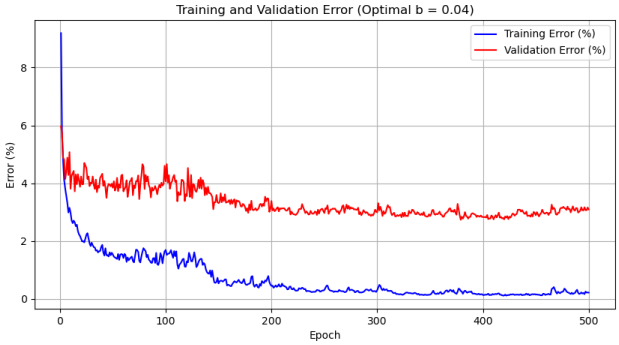

Figure 4: Best_b

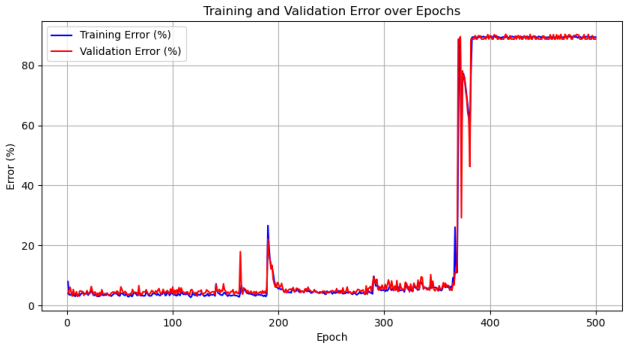

Figure 5: CNN

Q5.4

I think there is no need to chase zero training loss after achieving zero training error. Because in Q5.3, we have already shown that the performance of b=0.0 and b=0.04 is not much difference(b=0.04 even slightly better).

Q5.5

With my observation, CNN's convergence speed is faster than one-hiddenlayer-net. Besides, after 350 epochs, both training error and validation error increased drastically, which implies after reaching the optimal point, the CNN is more sensitive to perturbation.