CSE343: Machine Learning

Assignment-3

Shubham Sharma (2021099)

November 11, 2024

1 Section A (Theoretical)

Solution (a)

Given

$$\begin{aligned} \alpha &= 0.01 \\ \text{Inputs: } &[1, 2, 3] \\ \text{Targets: } &[3, 4, 5] \end{aligned}$$

Assuming initial parameters:

$$\begin{aligned} w_1 = 0.5, \quad b_1 = 0.1 \\ w_2 = -0.3, \quad b_2 = 0.2 \end{aligned}$$

Forward Pass

Hidden Layer Activation: $h_i = \text{ReLU}(w_1 \cdot x_i + b_1) = max(0, (w_1 \cdot x_i + b_1))$
Output Layer Activation: $y_i = w_2 \cdot h_i + b_2$

For $x = 1$:

$$\begin{aligned} h_1 &= \text{ReLU}(0.5 \cdot 1 + 0.1) = max(0, 0.6) = 0.6 \\ y_1 &= -0.3 \cdot 0.6 + 0.2 = 0.02 \end{aligned}$$

For $x = 2$:

$$\begin{aligned} h_2 &= \text{ReLU}(0.5 \cdot 2 + 0.1) = max(0, 1.1) = 1.1 \\ y_2 &= -0.3 \cdot 1.1 + 0.2 = -0.13 \end{aligned}$$

For $x = 3$:

$$\begin{aligned} h_3 &= \text{ReLU}(0.5 \cdot 3 + 0.1) = max(0, 1.6) = 1.6 \\ y_3 &= -0.3 \cdot 1.6 + 0.2 = -0.28 \end{aligned}$$

Loss Calculation (Mean Squared Error)

The MSE loss is calculated as:

$$\text{Loss} = \frac{1}{3} \sum_{i=1}^{3} (y_i - t_i)^2$$

where $t = [3, 4, 5]$.

$$\text{Loss} = \frac{1}{3}(8.9404 + 17.0149 + 27.8784) = \frac{53.8337}{3} \approx 17.9446$$

Backpropagation: Gradient Computation

Output Layer Gradients (w_2 and b_2)

For each output y_i, we calculate:

$$\frac{\partial \text{Loss}}{\partial y_i} = \frac{2}{3}(y_i - t_i)$$

Then, for w_2:

$$\frac{\partial \text{Loss}}{\partial w_2} = \frac{1}{3}\sum_{i=1}^{3} \frac{\partial \text{Loss}}{\partial y_i} \cdot h_i$$

And for b_2:

$$\frac{\partial \text{Loss}}{\partial b_2} = \frac{1}{3}\sum_{i=1}^{3} \frac{\partial \text{Loss}}{\partial y_i}$$

For $x = 1, y_1 = 0.02, t_1 = 3$:

$$\frac{\partial \text{Loss}}{\partial y_1} = \frac{2}{3}(0.02 - 3) = -1.9867$$

For $x = 2, y_2 = -0.13, t_2 = 4$:

$$\frac{\partial \text{Loss}}{\partial y_2} = \frac{2}{3}(-0.13 - 4) = -2.7533$$

For $x = 3, y_3 = -0.28, t_3 = 5$:

$$\frac{\partial \text{Loss}}{\partial y_3} = \frac{2}{3}(-0.28 - 5) = -3.52$$

Gradient for w_2:

$$\frac{\partial \text{Loss}}{\partial w_2} = \frac{1}{3}((-1.9867 \cdot 0.6) + (-2.7533 \cdot 1.1) + (-3.52 \cdot 1.6)) = -3.2842$$

Gradient for b_2:

$$\frac{\partial \text{Loss}}{\partial b_2} = \frac{1}{3}(-1.9867 - 2.7533 - 3.52) = -2.7533$$

Hidden Layer Gradients (w_1 and b_1)

Gradient of Loss w.r.t h_i:

$$\frac{\partial \text{Loss}}{\partial h_i} = \frac{\partial \text{Loss}}{\partial y_i} \cdot w_2$$

Gradient of Loss w.r.t w_1 and b_1:

$$\frac{\partial \text{Loss}}{\partial w_1} = \frac{1}{3}\sum_{i=1}^{3} \frac{\partial \text{Loss}}{\partial h_i} \cdot x_i \cdot \text{ReLU}'(z_i)$$

$$\frac{\partial \text{Loss}}{\partial b_1} = \frac{1}{3}\sum_{i=1}^{3} \frac{\partial \text{Loss}}{\partial h_i} \cdot \text{ReLU}'(z_i)$$

where $\text{ReLU}'(z_i)$ is the derivative of ReLU at $z_i = w_1 \cdot x_i + b_1$.

For $x = 1, \text{ReLU}'(0.6) = 1$:

$$\frac{\partial \text{Loss}}{\partial h_1} = -1.9867 \cdot (-0.3) = 0.59601$$

For $x = 2$, $\text{ReLU}'(1.1) = 1$:

$$\frac{\partial \text{Loss}}{\partial h_2} = -2.7533 \cdot (-0.3) = 0.826$$

For $x = 3$, $\text{ReLU}'(1.6) = 1$:

$$\frac{\partial \text{Loss}}{\partial h_3} = -3.52 \cdot (-0.3) = 1.056$$

Gradient for w_1:

$$\frac{\partial \text{Loss}}{\partial w_1} = \frac{1}{3}(0.59601 \cdot 1 + 0.826 \cdot 2 + 1.056 \cdot 3)$$

$$= \frac{1}{3}(0.59601 + 1.652 + 3.168) = \frac{5.41601}{3} \approx 1.8053$$

Gradient for b_1:

$$\frac{\partial \text{Loss}}{\partial b_1} = \frac{1}{3}(0.59601 + 0.826 + 1.056) = \frac{2.47801}{3} \approx 0.826$$

Parameter Update

Using a given learning rate $\alpha = 0.01$:

Update w_2:

$$w_2 = w_2 - \alpha \cdot \frac{\partial \text{Loss}}{\partial w_2} = -0.3 - 0.01 \cdot (-3.2842) = -0.3 + 0.032842 = \textbf{-0.2672}$$

Update b_2:

$$b_2 = b_2 - \alpha \cdot \frac{\partial \text{Loss}}{\partial b_2} = 0.2 - 0.01 \cdot (-2.7533) = 0.2 + 0.0275 = \textbf{0.2275}$$

Update w_1:

$$w_1 = w_1 - \alpha \cdot \frac{\partial \text{Loss}}{\partial w_1} = 0.5 - 0.01 \cdot 1.8053 = 0.5 - 0.01805 = \textbf{0.4819}$$

Update b_1:

$$b_1 = b_1 - \alpha \cdot \frac{\partial \text{Loss}}{\partial b_1} = 0.1 - 0.01 \cdot 0.826 = 0.1 - 0.00826 = \textbf{0.0917}$$

Updated Parameters

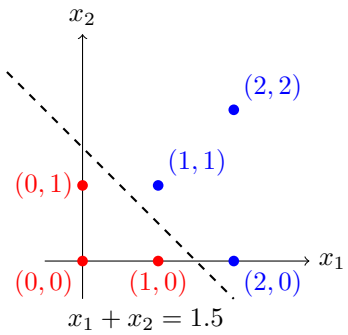
$$w_1 = 0.4819$$
$$b_1 = 0.0917$$
$$w_2 = -0.2672$$
$$b_2 = 0.2275$$

Solution (b)

Class	x_1	x_2	Label
+	0	0	+
+	1	0	+
+	0	1	+
-	1	1	-
-	2	2	-
-	2	0	-

Table 1: Given Datapoints

a) Linear Separability of Points



Class	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_1 + \mathbf{x}_2$
+	0	0	0
+	1	0	1
+	0	1	1
−	1	1	2
−	2	2	4
−	2	0	2

From the table:

- For the positive class, $x_1 + x_2$ values are 0 and 1.
- For the negative class, $x_1 + x_2$ values are 2 and 4.

We choose the line $x_1 + x_2 = 1.5$ as the separating line:

- All positive points satisfy $x_1 + x_2 \leq 1.5$.
- All negative points satisfy $x_1 + x_2 > 1.5$.

Thus, the points are **linearly separable** by the line $x_1 + x_2 = 1.5$. Also, from the graph, it is clearly visible that data points are **linearly separable**.

b) Finding Weight Vector and Support Vector

From the Graph, I observed the below support vectors:

- Positive support vectors: $(0, 1)$ and $(1, 0)$
- Negative support vectors: $(1, 1)$ and $(2, 0)$

Using support vectors:

$$w_1 \cdot 0 + w_2 \cdot 1 + b = 1, \tag{1}$$
$$w_1 \cdot 1 + w_2 \cdot 0 + b = 1, \tag{2}$$
$$w_1 \cdot 1 + w_2 \cdot 1 + b = -1, \tag{3}$$
$$w_1 \cdot 2 + w_2 \cdot 0 + b = -1. \tag{4}$$

Solving these equations step-by-step:

1. From the equation (1), we have $w_2 = 1 - b$.
2. Substitute $w_2 = 1 - b$ into the equation (3):

$$\begin{aligned} w_1 + (1 - b) + b &= -1 \\ w_1 + 1 &= -1 \\ w_1 &= -2. \end{aligned}$$

3. Substitute $w_1 = -2$ into the equation (2):

$$\begin{aligned} -2 + b &= 1 \\ b &= 3. \end{aligned}$$

4. Substitute $b = 3$ into the equation (1) to find w_2:

$$\begin{aligned} w_2 &= 1 - 3 \\ w_2 &= -2. \end{aligned}$$

So,

$$\mathbf{w_1 = -2}, \quad \mathbf{w_2 = -2}, \quad \mathbf{b = 3}$$

The equation of the maximum margin hyperplane is:

$$-2x_1 - 2x_2 + 3 = 0,$$

or equivalently:

$$x_1 + x_2 = 1.5.$$

Solution (c)

(a) Calculate the margin of the classifier.

Margin γ for an SVM is:

$$\gamma = \frac{2}{\sqrt{w_1^2 + w_2^2}}$$

Substitute $w_1 = -2$ and $w_2 = 0$:

$$\sqrt{w_1^2 + w_2^2} = \sqrt{(-2)^2 + 0^2} = \sqrt{4} = 2$$

Thus, the margin is:

$$\gamma = \frac{2}{2} = \mathbf{1}$$

(b) Identify the support vectors.

Support vectors are the points that lie on the margin boundaries, where $\mathbf{y\ (w \cdot x + b) = 1}$.

Sample No.	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{y}$	$\mathbf{w \cdot x + b}$	$\mathbf{y(w \cdot x + b)}$
1	1	2	+1	$(-2 \cdot 1) + (0 \cdot 2) + 5 = 3$	$1 \cdot 3 = 3$
2	2	3	+1	$(-2 \cdot 2) + (0 \cdot 3) + 5 = 1$	$1 \cdot 1 = \mathbf{1}$
3	3	3	−1	$(-2 \cdot 3) + (0 \cdot 3) + 5 = -1$	$-1 \cdot -1 = \mathbf{1}$
4	4	1	−1	$(-2 \cdot 4) + (0 \cdot 1) + 5 = -3$	$-1 \cdot -3 = 3$

From the table, we observe that **Samples 2 and 3** satisfy $y(w \cdot x + b) = 1$, so they are the support vectors.

(c) Predict the class of a new point

Using the decision function:

$$f(x) = w \cdot x + b = -2 \cdot x_1 + 0 \cdot x_2 + 5$$

For $(x_1, x_2) = (1, 3)$:

$$f(x) = -2 \cdot 1 + 0 \cdot 3 + 5 = -2 + 5 = 3$$

Since $f(x) > 0$, the classifier predicts $\mathbf{y = +1}$ for this new point.

2 Section C (Algorithm implementation using packages)

Part 1: Normalization and Visualization of Sample Images

a) Visualization of Sample Images

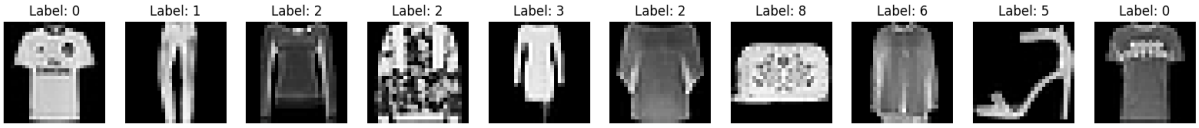


Figure 1: First 10 Sample Images from Test Dataset

All the data points in the dataset are first normalized by dividing them by 255.0 then the first 10 images from the test dataset are displayed after reshaping the features to 28x28.

Part 2: Training model and Calculating Training and Validation Loss

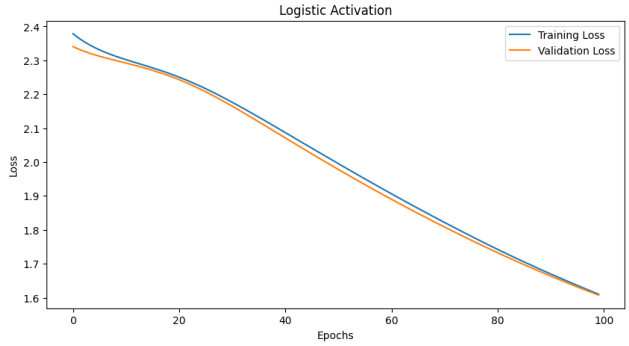


Figure 2: Losses vs Epochs for **Logistic** Activation

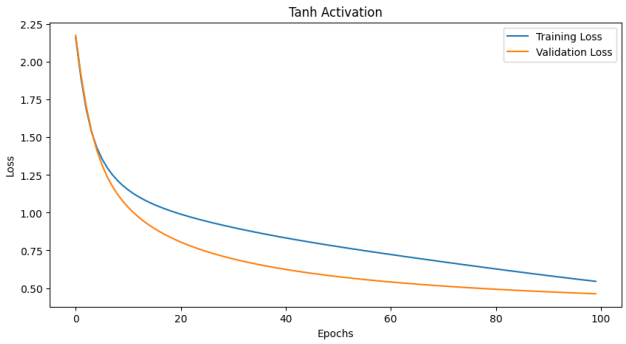


Figure 3: Losses vs Epochs for **Tanh** Activation

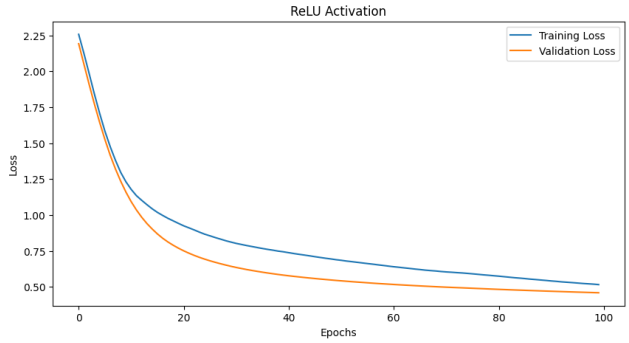


Figure 4: Losses vs Epochs for **ReLU** Activation

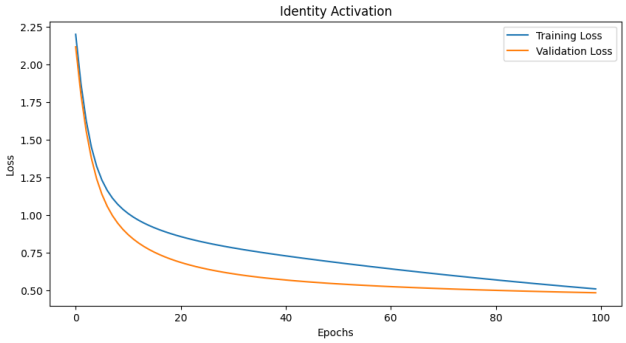


Figure 5: Losses vs Epochs for **Identity** Activation

Activation Function	**Test Accuracy**
Logistic	0.533
Tanh	0.8375
ReLU	0.835
Identity	0.831

Table 2: Different Activation Functions and their Accuracies

Best performing Activation function is **tanh** and some major observations are below:-

1. Logistic activation has poor performance (53.3%) due to vanishing gradients, which hinders the learning of complex patterns.
2. Other activation functions reduce vanishing gradients—Tanh scales to (-1,1), ReLU introduces sparsity, and Identity allows linear propagation, leading to better performance around 83%.
3. Random Variability in weight initialization causes slight differences in convergence paths, affecting accuracy.
4. I observed on each run Tanh, ReLU, or Identity to perform best depending on the random factors such as random weights, convergence path and different local minima etc.

Part 3: Grid Search CV using Best Activation Function

Hyperparameter	**Values (Grid Search)**	**Best Value**
solver	{'adam', 'sgd'}	'adam'
learning_rate_init	{0.1, 0.0001, 2e-3, 2e-4, 2e-5}	2e-3
batch_size	{64, 128, 256}	128

Table 3: MLP Classifier Hyperparameter Grid and Best Parameters from Grid Search

Performed the Grid Search CV with 3 folds. I performed Gird Search on three parameters, such as solver, learning_rate and batch_size, which resulted in a total of **30 combinations** and **90 fits**. In all of the combinations, the activation function we used is the best activation function that we found in the previous part.

Part 4: Training MLP Regressor Model for Image Regeneration Task

a) Layer Sizes

Layer Size	**[128, 64, 32, 64, 128]**

Chosen layer sizes $\mathbf{c = 128}$, $\mathbf{b = 64}$, and $\mathbf{a = 32}$ are ideal for regeneration tasks as they form symmetric structures and compress and reconstruct images efficiently.

b) Plotting Curves

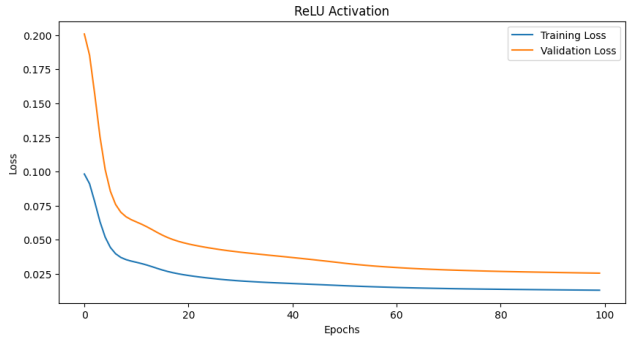


Figure 6: Losses vs Epochs for **ReLU** Activation

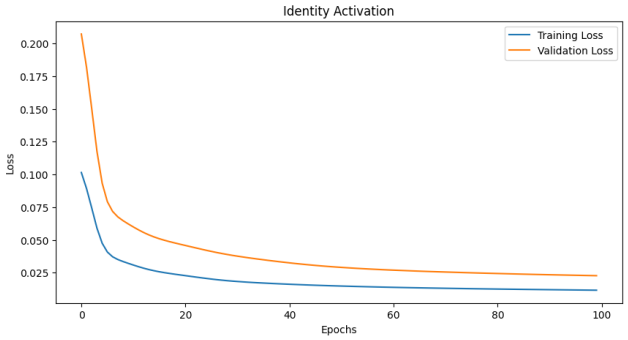


Figure 7: Losses vs Epochs for **identity** Activation

As clearly visible from both graphs, the loss continuously decreases with each epoch, and we conclude that our models are learning correctly.

c) Training Models

Activation Function	**RMSE Score**
ReLU	0.145426
Identity	0.154098

Table 4: Different Activation Functions and their RMSE Score

d) Visualization of Regenerated Images

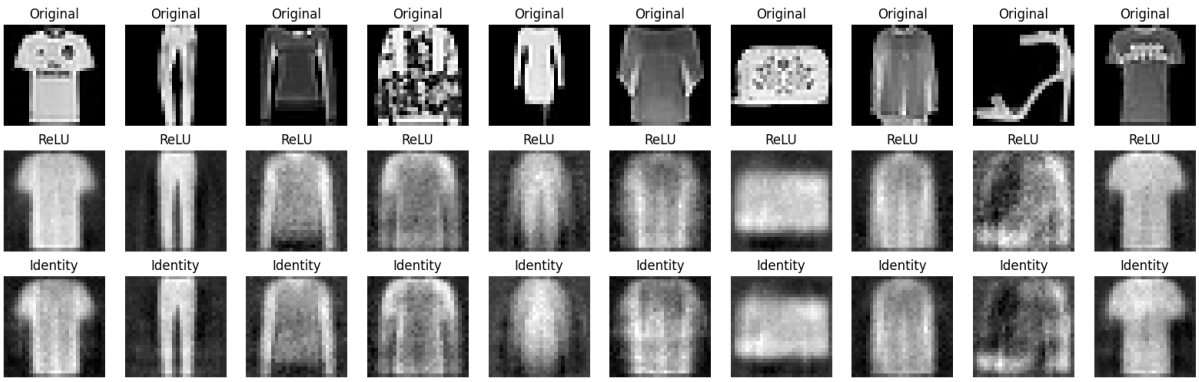


Figure 8: Comparision of First 10 Sample Images from Test Dataset

Observations:-

1. I observed that the model which uses the ReLU activation function produces slightly clearer and sharper regenerated images compared to the Identity activation function because ReLU is a non-linear activation function and better captures complex patterns.

2. Images generated using the Identity activation function are slightly blurred compared to ReLU, which directly shows that the Identity activation function is not good at capturing complex details.

3. In Images generated by the ReLU activation function, the object outlines are slightly sharper than the identity, which makes it visually easy to recognize the object in the image.

Part 5: Extracting Features and Training Classifiers on Extracted Features

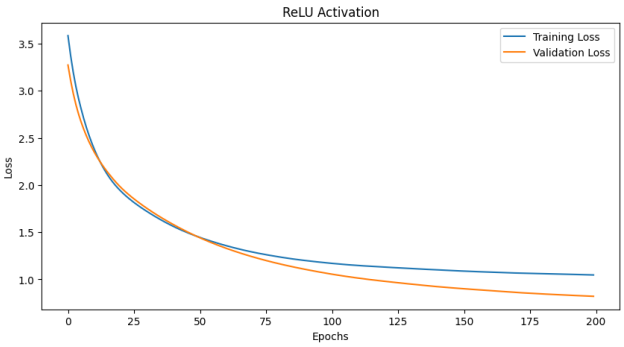


Figure 9: Losses vs Epochs for **ReLU** Activation

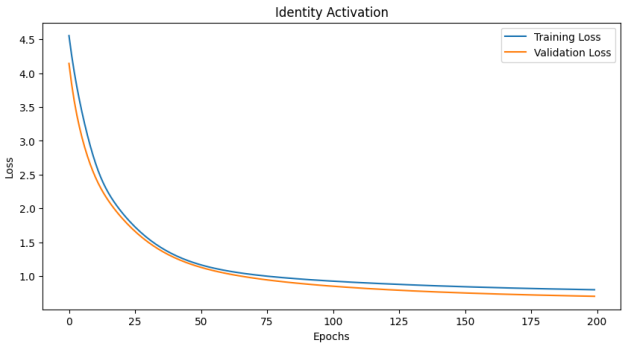


Figure 10: Losses vs Epochs for **identity** Activation

Activation Function	**Test Accuracy**
ReLU	0.692
Identity	0.741

Table 5: Different Activation Functions and their Accuracies

Contrast of Part 5 with Part 2 is below:-

1. In Part 2, the classifier model with ReLU and Identity activation gives 83.5% and 83.1% accuracy, respectively, while in Part 5, the classifier model with ReLU and Identity activation gives 69.2% and 74.1% accuracy, respectively.

Possible Reason:-

1. Extracted feature vectors keep the important pattern and make them ideal for quickly training smaller classifiers.

2. Feature vectors reduce data complexity but keep essential details, helping smaller classifiers in achieving decent accuracy.

3. Features from deeper layers capture detailed patterns and boost image classification accuracy.