HLCV EXERCISE 2 REPORT

AKSHAY JOSHI
2581346
s8akjosh@stud.uni-saarland.de

ANKIT AGRAWAL
2581532
s8anagra@stud.uni-saarland.de

SUSHMITA NAIR
2581308
s8sunair@stud.uni-saarland.de

Question 1: Implementing the Feedforward Model

Results:

Your scores:

```
[[0.3644621  0.22911264 0.40642526]
 [0.47590629 0.17217039 0.35192332]
 [0.43035767 0.26164229 0.30800004]
 [0.41583127 0.2983228  0.28584593]
 [0.36328815 0.32279939 0.31391246]]
```

Correct scores:

```
[[0.3644621  0.22911264 0.40642526]
 [0.47590629 0.17217039 0.35192332]
 [0.43035767 0.26164229 0.30800004]
 [0.41583127 0.2983228  0.28584593]
 [0.36328815 0.32279939 0.31391246]]
```

Difference between your scores and correct scores:
2.9173411658645065e-08

Difference between your loss and correct loss:
1.794120407794253e-13

Question 2: Backpropagation

Results:

W2 Max Relative Error: 3.440708e-09
b2 Max Relative Error: 3.865070e-11
W1 Max Relative Error: 3.561318e-09
b1 Max Relative Error: 1.555470e-09

Computation of the derivatives mentioned in the question is as follows: (please refer next page)

(2)

* Calculating the Derivatives of network params / Performing Backpropogation :-

→ Given :-

- Input : x (image)
- Output : $f_\theta(x) = a^{(3)}$ (Probabilities of x belonging to K)

Activations :-

- Hidden Layer : ReLU $[f(x) = \max(0, z)]$
- Output Layer : Softmax $[\sigma(x)_i = e^{x_i} / \sum_{j=1}^{K} e^{x_i}]$

Model :-

- $a^{(1)} = x$ # Input
- $z^{(2)} = w^{(1)} \cdot a^{(1)} + b^{(1)}$ # multiply input by weight + bias
- $a^{(2)} = \phi(z^{(2)})$ # Pass $z^{(2)}$ through ReLU
- $z^{(3)} = w^{(2)} \cdot a^{(2)} + b^{(2)}$
- $a^{(3)} = \psi(z^{(3)})$ # output obtained using Softmax.

Parameters :-

$\theta = (w^{(1)}, b^{(1)}, w^{(2)}, b^{(2)})$

- Cross Entropy loss function over Full Batch

$$J(\theta, \{x_i, y_i\}_{i=1}^{N}) = \frac{1}{N} \sum_{i=1}^{N} -\log \left[\frac{\exp^{z_{y_i}^{(3)}}}{\sum_j \exp^{z_j^{(3)}}} \right]$$

(a) Deriving the above eqn w.r.t z

WKT, $\frac{\partial J}{\partial z^{(3)}} = \frac{\partial J}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}}$ — (1)

Let,

$$\frac{\partial J}{\partial a^{(3)}} = \frac{\partial}{\partial a^{(3)}} \cdot \left[1/N \cdot \sum_{i=1}^{N} -\log\left(a^{(3)}\right) \right]$$

Then we would get:

$$= \frac{\partial}{\partial a^{(3)}} \left[1/N \cdot \sum_{i=1}^{N} \underbrace{-\log\left(a^{(3)}\right)}_{\text{(using log to transform)}} \cdot \frac{da^{(3)}}{dz^{(3)}} \right.$$

$$= \frac{-1}{N} \cdot \sum_{i=1}^{N} \frac{1}{\left(a^{(3)}\right)} \cdot (1) \cdot \underbrace{\frac{da^{(3)}}{dz^{(3)}}}_{\text{(can write this as)}}$$

$$= \left(\frac{-1}{N} \cdot \sum_{i=1}^{N} \frac{1}{\left(a^{(3)}\right)} \right) \cdot \left(\frac{\partial}{\partial z^{(3)}} \underbrace{\psi\left(z^{(3)}\right)}_{\text{softmax}} \right)$$

- We already know the derivative of softmax function [Reference: Wikipedia/math.stackexchange]

  - For $i = j$

$$\psi\left(z^{(3)}\right) = s_i (1 - s_j)$$

  - For $i \neq j$

$$\psi\left(z^{(3)}\right) = -s_j s_i$$

i.e $$\psi\left(z^{(3)}\right) = \begin{cases} s_i (1 - s_j) & , i = j \\ -s_j s_i & , i \neq j \end{cases}$$

Substituting the values of $\psi\left(z^{(3)}\right)$ in (1)

- when $i = j$

$$= \frac{-1}{N} \cdot \sum_{i=1}^{N} \frac{1}{\psi\left(z_{y_i}^{(3)}\right)} \cdot \psi\left(z_{y_i}^{(3)}\right) \left(1 - \psi\left(z_{y_i}^{(3)}\right)\right)$$

(multiplying by '-')

$$= \frac{1}{N} \left(\psi\left(z_{y_i}^{(3)}\right) - 1 \right)$$

• When $i \neq j$

$$= \frac{-1}{N} \cdot \sum_{i=1}^{N} \frac{1}{\psi(z_{yi}^{(3)})} \cdot -\psi(z_{yj}^{(3)}) \cdot \psi(z_{yi}^{(3)})$$

$$= \frac{1}{N} \cdot \psi(z_{yj}^{(3)})$$

∴ The generalized form:

$$\therefore \boxed{\frac{\partial J}{\partial z} = \frac{1}{N} (\psi(z^{(3)}) - \Delta)}$$

Δ is either 1 or 0

$$\Delta_{ij} = \begin{cases} 1, & y_i = j \\ 0, & y_i \neq j \end{cases}$$

(b) Deriving the Loss function w.r.t $w^{(2)}$

Given,

$$\frac{dJ}{dw^{(2)}} = \frac{dJ}{dz^{(3)}} \cdot \frac{dz^{(3)}}{dw^{(2)}}$$

During backprop this can be solved as (using ①)

$$\frac{dJ}{dw^{(2)}} = \underbrace{\frac{dJ}{da^{(3)}} \cdot \frac{da^{(3)}}{dz^{(3)}}}_{\text{use previous result}} \cdot \frac{dz^{(3)}}{dw^{(2)}}$$

$$= \frac{1}{N} (\psi(z^{(3)}) - \Delta) \cdot \frac{d}{dw^{(2)}} (a^{(2)} \cdot w^{(2)} + b^{(2)})$$

$$= \frac{1}{N} (\psi(z^{(3)}) - \Delta) \cdot (a^{(2)}(1))^{T} \quad \text{—②}$$

Now, considering the influence of L2 regularization/ weight penalty on loss:

Given:

$$\tilde{J}(\theta) = \frac{1}{N}\sum_{i=1}^{N} -\log\left[\frac{\exp^{z_{y_i}^{(3)}}}{\sum_j \exp^{z_j^{(3)}}}\right] + \lambda\left(\|w^{(1)}\|_2^2 + \|w^{(2)}\|_2^2\right)$$

Deriving the above w.r.t $w^{(2)}$

$$\frac{\partial \tilde{J}}{\partial w^{(2)}} = \frac{\partial}{\partial w^{(2)}}\frac{1}{N}\sum_{i=1}^{N} -\log\left[\frac{\exp^{z_{y_i}^{(3)}}}{\sum_j \exp^{z_j^{(3)}}}\right] + \frac{\partial}{\partial w^{(2)}}\lambda\left(\|w^{(1)}\|_2^2 + \|w^{(2)}\|_2^2\right)$$

[using ②]

$$= \frac{1}{N}\left(\psi(z^{(3)}) - \Delta\right)a^{(2)^T} + 2\lambda W^{(2)}$$

③ Finally, derivatives of loss w.r.t all the model parameters θ are:

- $\frac{\partial J}{\partial b^{(2)}} = \frac{\partial J}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial b^{(2)}}$

[using ①]

$$= \frac{1}{N}\cdot\left(\psi(z^{(3)}) - \Delta\right)\cdot\frac{\partial}{\partial b^{(2)}}\left(a^{(2)}w^{(2)} + b^{(2)}\right)$$

$$= \frac{1}{N}\cdot\left(\psi(z^{(3)}) - \Delta\right)\cdot$$

- $\frac{\partial J}{\partial w^{(1)}} = \frac{\partial J}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial w^{(1)}}$

$$= \frac{1}{N}\left(\psi(z^{(3)}) - \Delta\right)\frac{\partial}{\partial a^{(2)}}\left(a^{(2)}w^{(2)} + b^{(2)}\right)\cdot$$

$$\frac{\partial}{\partial z^{(2)}}\phi(z^{(2)})\cdot\frac{\partial}{\partial w^{(1)}}\cdot\left(a^{(1)}\cdot w^{(1)} + b^{(1)}\right) + \frac{\partial}{\partial w^{(2)}}\cdot$$

$$\left(\lambda\left(\|w^{(1)}\|_2^2 + \|w^{(2)}\|_2^2\right)\right)$$

$$= \frac{1}{N}(\psi(z^{(3)}) - \Delta) \cdot (w^{(2)})(1) \cdot \phi(z^{(2)}) \cdot a^{(1)}(i) + 2\lambda w^{(1)}$$

- Finally,

$$\frac{\partial J}{\partial b^{(1)}} = \frac{\partial J}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial b^{(1)}}$$

(using all the previously computed results)

$$= \frac{1}{N}(\psi(z^{(3)}) - \Delta) \cdot \frac{\partial}{\partial a^{(2)}}(a^{(2)} w^{(2)} + b^{(2)}) \cdot \frac{\partial}{\partial z^{(2)}} \phi(z^{(2)}) \cdot 1$$

$$= \frac{1}{N}(\psi(z^{(3)}) - \Delta) \cdot (w^{(2)}) \cdot \phi(z^{(2)})$$

Question 3

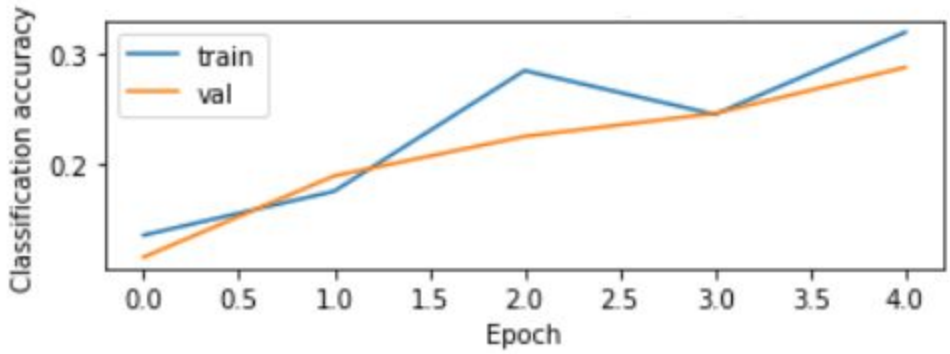
Experimental Setup :

3a) Model with default values for hyperparameters:

Train data shape: (49000, 3072)
Train labels shape: (49000,)
Validation data shape: (1000, 3072)
Validation labels shape: (1000,)
Test data shape: (1000, 3072)
Test labels shape: (1000,)

Hidden layer size: 50
Number of iterators: 1000
Batch size = 200
Learning Rate: 0.0001
Regularization strength: 0.25
Learning Rate Decay: 0.95

Validation accuracy: 0.287



3b)
Experiment Setup:

Train data shape: (49000, 3072)
Train labels shape: (49000,)
Validation data shape: (1000, 3072)
Validation labels shape: (1000,)
Test data shape: (1000, 3072)
Test labels shape: (1000,)

We have 5 hyperparameters(Hidden layer size, number of iterators, batch size, learning rate, regularization strength) for which we want to find the best hyperparameters combination which results in the highest accuracy for validation data.

We have implemented grid search for the following values of hyperparameters.
And looking at the plot we see for which values we are overfitting the model. And In the end we select the hyperparameters which result in the best validation accuracy for our model.

```
Hidden layer size:  [50, 60, 70]
Number of iterators:  [1000, 2000, 3000]
Batch size:   [400, 500, 600]
Regularization strength:  [0.25, 0.30, 0.35]
Learning Rate:  0.001
Learning Rate Decay:  0.95
```

Resulting in a total combination of 81 combinations.

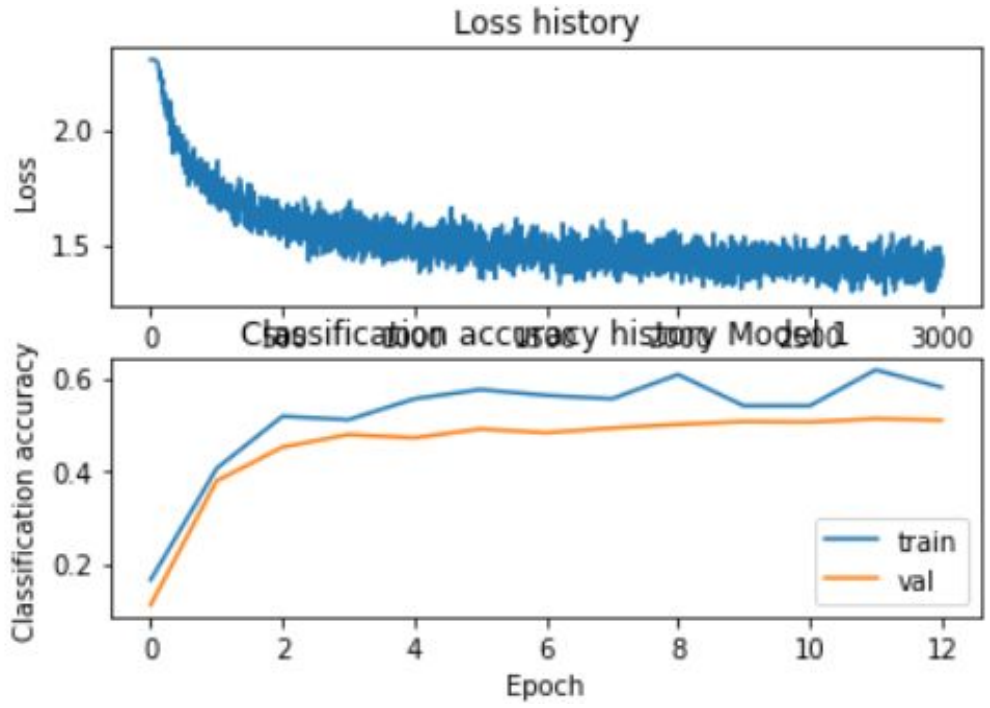
Result:

The best model had a **validation accuracy of 53.4%**
This was found for the following combinations of hyperparameters:
Hidden layer size: 70
Number of iterators: 2000
Batch size: 600
Learning Rate: 0.001
Regularization strength: 0.25
Learning Rate Decay: 0.95
New Validation accuracy: 0.534
Test accuracy: 0.528

Analysis:
Below are examples of some models along with their validation accuracy plot.
(complete log for all 81 iterations can be found in log_p3b.txt file)

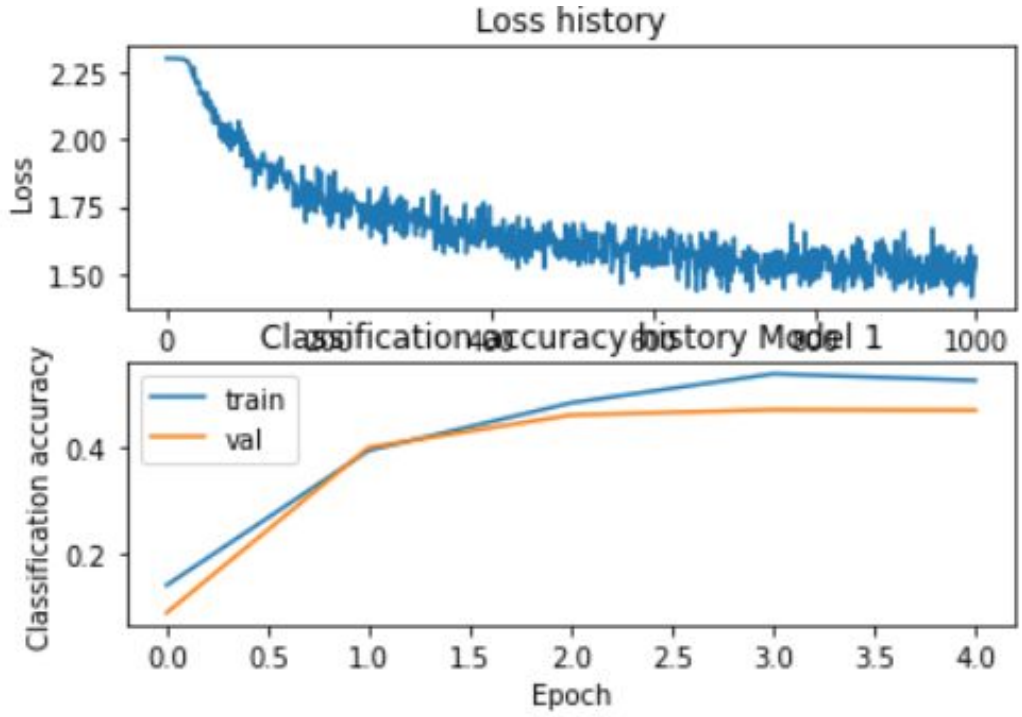
Model 1:

```
Hidden layer size:  60
Number of iterators:  3000
Batch size:  400
Learning Rate:  0.001
Regularization strength:  0.35
Learning Rate Decay:  0.95
New Validation accuracy:  0.518
Test accuracy:  0.519
```



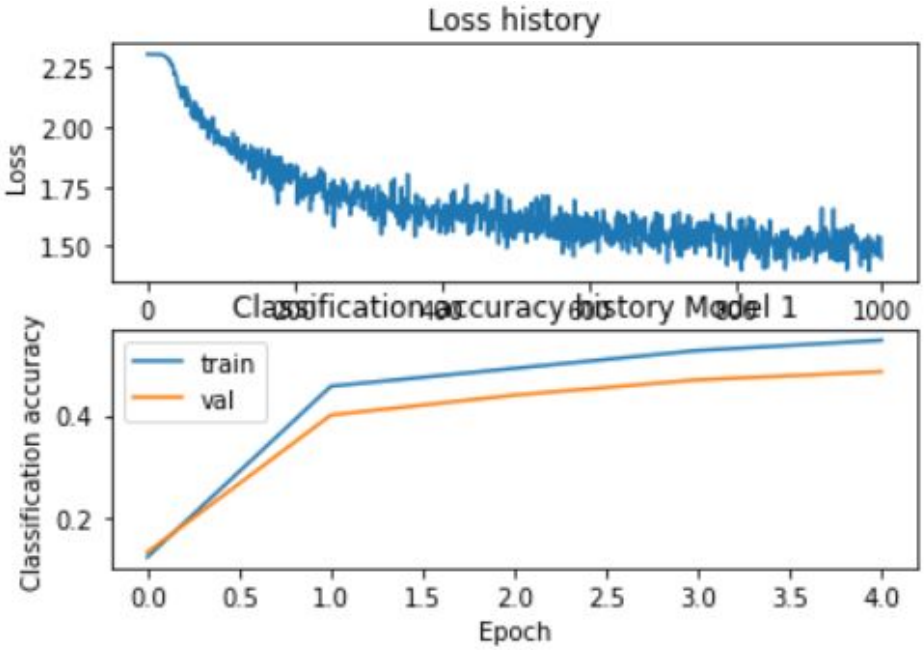
Model 2:

Hidden layer size: 60
Number of iterators: 1000
Batch size: 400
Learning Rate: 0.001
Regularization strength: 0.35
Learning Rate Decay: 0.95
New Validation accuracy: 0.478
Test accuracy: 0.499



Model 3:

Hidden layer size: 50
Number of iterators: 1000
Batch size: 400
Learning Rate: 0.001
Regularization strength: 0.3
Learning Rate Decay: 0.95
New Validation accuracy: 0.477
Test accuracy: 0.466



(complete log for all 81 iterations can be found in log_p3b.txt file)

Question 4

Experimental Setup :

All 5 models in this experiment are MLPs with ReLU as activations between each layer. All other parameters and hyper-parameters remain the same.

1) Model 1: 2-Layer MLP with hidden layer of size 50 neurons.
2) Model 2: 2-Layer MLP with hidden layer of size 60 neurons
3) Model 3: 3-Layer MLP with each hidden layer sized 50 and 60 neutrons respectively.
4) Model 4: 4-Layer MLP with each hidden layer sized 50, 60 and 50 neutrons respectively.
5) Model 5: 5-Layer MLP with each hidden layer sized 50, 60, 60 and 50 respectively.

Results :

Test and validation accuracies have been reported from the last epoch.

Model	Loss	Train Accuracy	Validation Accuracy
Model 1	1.2319	51.1%	50%
Model 2	**1.2972**	**49.65%**	**52.7%**
Model 3	1.2411	48.35%	52.7%
Model 4	2.3024	9.55%	7.9%
Model 5	2.3027	9.25%	7.8%

Observations :

Among Models 1, 2 and 3, validation set accuracy is highest for Model 2 and Model 3 and they have comparable train accuracies as well.
We can see that deeper models like Model 4 and Model 5 don't fit the data at all since they both have train accuracies below the baseline of 10% (since there are 10 classes). This could be due to high model bais.
Therefore, the optimal model is **Model 2**, the simplest model with good performance.
Evaluation with test data on Model 2 gives 50% accuracy.