COL774 Assignment 2

Rishit Jakharia, 2022CS11621

September 22, 2024

1 Part C: Multi-Class Classification

1.1 Key Components

1.1.1 Layer Definitions

- **Linear Layer:** Implements fully connected layers using He initialization for weights. It includes methods for forward propagation and backpropagation to compute gradients.
- **Activation Functions:**
  - **Sigmoid:** Applies the sigmoid function and calculates gradients for backpropagation.

$$f(x) = \frac{1}{1 + e^{-x}}$$

  - **Softmax:** Computes softmax for multi-class probabilities and its gradient.

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

1.1.2 Sequential Model

The `Sequential` class manages a sequence of layers, facilitating forward and backward passes while handling parameter management.

1.1.3 Optimizers

The `Optimizers` class implements various optimization algorithms (SGD, RMSProp, Adam), managing learning rates, momentum, and other hyperparameters.

Table 1: Optimiser vs loss at best setting (after preliminary testing)

Optimizer	Loss at best setting
SGD	2.34
Momentum	2.14
RMSProp	2.04
Adam	1.68

1.1.4 Training Process

Defines training parameters, including epochs, batch size, and learning rate. The `train` function manages the training loop, handling data loading, forward propagation, loss calculation, backpropagation, and optimizer steps.

1.2 Training Parameters

After some initial testing, we set the optimizer to adam, the following results are show using Adam optimizer For a *epochs = 25*, we found the following lowest score (early stopping was used here to prevent higher learning rates to increase cost)

Table 2: Batch Size, Learning rate vs loss

Batch Size ↓ Learning Rate →	O(1e-5)	O(1e-4)	O(1e-3)	O(1e-2)	O(1e-1)
1	2.34	2.30	2.02	2.10	2.22
32	2.33	2.29	2.12	2.13	2.25
64	2.33	2.29	2.10	2.13	2.30
128	2.33	2.28	1.97	2.11	2.22
256	2.34	2.26	1.98	2.01	2.12
512	2.32	2.25	2.10	2.15	2.32
1024	2.34	2.26	2.11	2.17	2.33

Now cosidering a narrower grid, aroung 128-256 and O(1e-3)

Table 3: Batch Size, Learning rate vs loss

Batch Size ↓ Learning Rate →	1e-3	3e-3	5e-3	7e-3	9e-3
128	1.97	1.90	1.91	1.94	1.95
200	1.97	1.89	1.92	1.93	1.93
256	1.98	1.94	1.90	1.90	1.93

- **Epochs:** 40
- **Batch Size:** 200
- **Learning Rate:** 2e-3
- **Optimizer:** Adam

2 Part D: Best Overall Model

3 Model Architecture

The model architecture is structured as follows:

Table 4: Model Architecture Summary

Layer Type	Number of Units	Activation Function
Input Layer	625	-
Fully Connected Layer	512	ReLU
Batch Normalization	-	-
Fully Connected Layer	256	ReLU
Batch Normalization	-	-
Fully Connected Layer	128	ReLU
Batch Normalization	-	-
Fully Connected Layer	64	ReLU
Batch Normalization	-	-
Fully Connected Layer	32	ReLU
Batch Normalization	-	-
Output Layer	8	Softmax

3.1 Rationale Behind Architecture Choices

The choice of architecture was influenced by several factors:

- **Layer Depth and Width**: Deeper networks can capture more complex patterns in data. The width of each layer was selected based on a balance between model complexity and computational feasibility.
- **Activation Functions**: ReLU was chosen due to its efficiency in training deep networks and its ability to mitigate the vanishing gradient problem.
- **Batch Normalization**: This was added after each fully connected layer to stabilize learning and accelerate convergence by normalizing layer inputs.

4 Hyperparameter Selection

The hyperparameters were tuned based on part c, as the base, and only learning rate was largely affected after the tuning

4.1 Learning Rate

The learning rate was set to 7×10^{-3}. This value was determined through grid search, assessing the model's convergence speed and loss reduction on validation data. Too high a learning rate led to divergence, while too low resulted in prolonged training times.

4.2 Advancements Logged, in Tables

4.2.1 Architecture

Table 5: Architecture, vs loss at basic setting

(Batch Size, Learning Rate)	1 Hidden Layers	4 Hidden Layers	5 Hidden Layers
(256, 1e-3)	2.31	1.98	1.97
(200, 1e-3)	2.25	1.97	1.96
(128, 1e-3)	2.24	1.97	1.96
(64, 1e-3)	2.22	2.10	1.98
(32, 1e-3)	2.25	2.12	1.98

4.2.2 Activation Functions

Below Calculations are done with the 5 Hidden Layer Network

Table 6: Activation Function, vs loss at basic setting

(Batch Size, Learning Rate)	Sigmoid	ReLU	LeakyReLU
(256, 1e-3)	1.97	1.63	2.11
(200, 1e-3)	1.96	1.60	2.10
(128, 1e-3)	1.96	1.67	2.21
(64, 1e-3)	1.98	1.68	2.32
(32, 1e-3)	1.98	1.68	2.31

4.2.3 Dropout and BatchNorm Layers

Below Calculations are done with the 5 Hidden Layer Network with ReLU Activation Function.
Note: Calculations with Batch Norm Layer were done after "failure" of dropout layer.

Table 7: Dropout and BatchNorm Layers

(Batch Size, Learning Rate)	Dropout	BatchNorm
(256, 1e-3)	2.27	0.012
(200, 1e-3)	2.36	0.009
(128, 1e-3)	2.36	0.010
(64, 1e-3)	2.38	0.011
(32, 1e-3)	2.38	0.012

Hence, in conclusion the final architecture was selected.

4.3 Learning Curve with chosen parameters

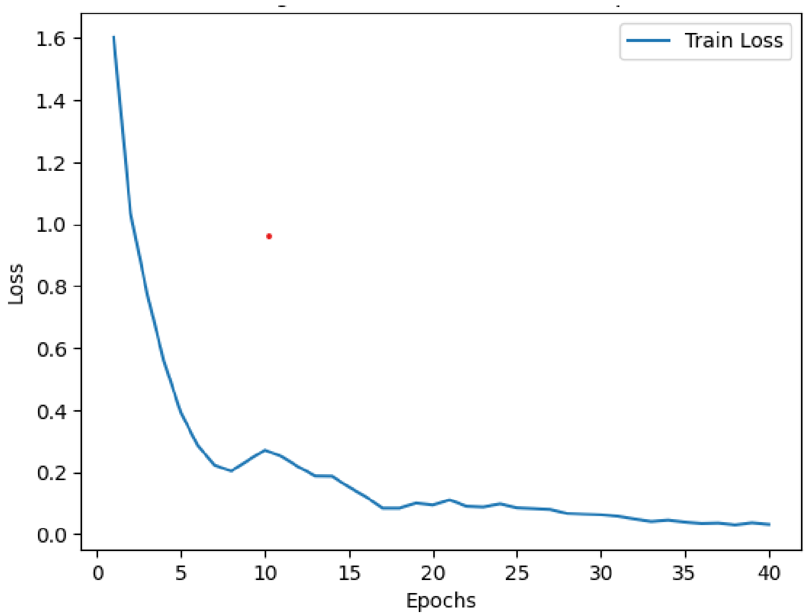

Figure 1: Learning Curve Showing Training Loss Over Epochs