Report : Group 67

Objective : Design custom differential equations that can be integrated into neural network architectures to enable dynamic adaptation based on the input data.

Requirements : While exploring different existing optimisers we stumbled across something called Gradient Descent with Momentum. Let me explain it to you !

Gradient descent is an optimization algorithm that follows the negative gradient of an objective function in order to locate the minimum of the function.
A problem with gradient descent is that it can bounce around the search space on optimization problems that have large amounts of curvature or noisy gradients, and it can get stuck in flat spots in the search space that have no gradient.
Momentum is an extension to the gradient descent optimization algorithm that allows the search to build inertia in a direction in the search space and overcome the oscillations of noisy gradients and coast across flat spots of the search space.

$$V_t = \gamma V_{t-1} + \eta \Delta_\theta J$$
$$\theta_t = \theta_{t-1} - \alpha V_t$$

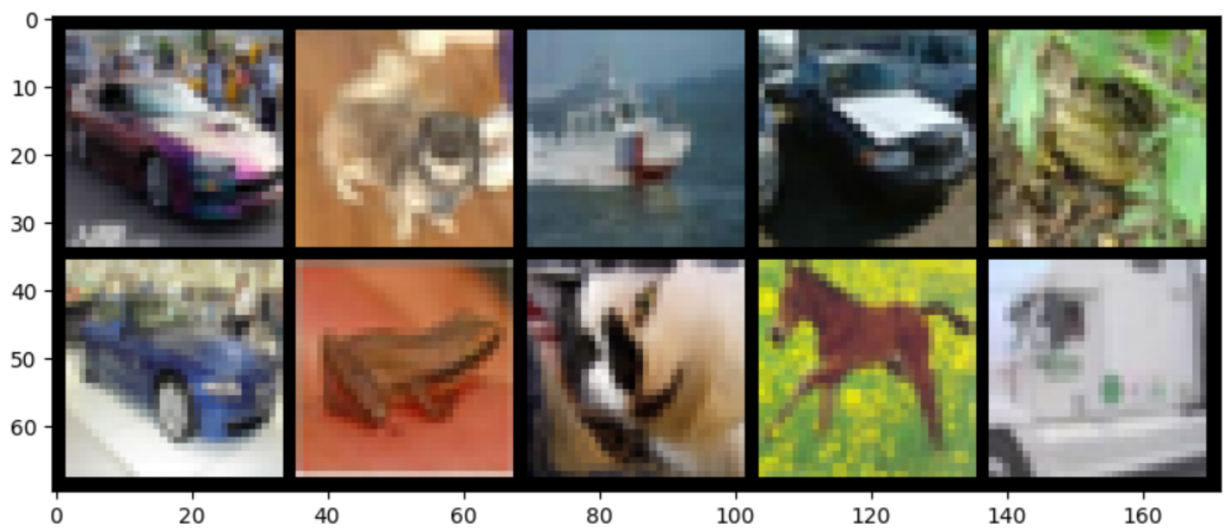
Modification : In the context of gradient descent with momentum, the momentum term is designed to accelerate the optimization process by accumulating past gradients to smooth out oscillations and hasten convergence. However, in scenarios where the loss function is descending a slope, high momentum values can lead to overshooting the minimum.
To address this concern, a strategy can be employed to dynamically adjust the momentum based on the behavior of the gradient. Specifically, when computing the gradient at a given point $x+\Delta x$, if the observed gradient is decreasing or approaching zero, it is indicative of proximity to a minimum. Consequently, a mechanism can be implemented to decrease the momentum, thereby mitigating the risk of overshooting the minimum.
This adaptive adjustment aligns with the goal of maintaining stability during the optimization process, allowing for efficient convergence without compromising the

precision of reaching the minimum. Such a dynamic momentum adaptation strategy enhances the robustness of the optimization algorithm, particularly in scenarios where the landscape of the loss function exhibits varying degrees of curvature.

The new modified equation is :

$$V_t = \gamma V_{t-1} + \eta \Delta_\theta J(\theta - \gamma V_{t-1})$$
$$\theta_t = \theta_{t-1} - \alpha V_t$$

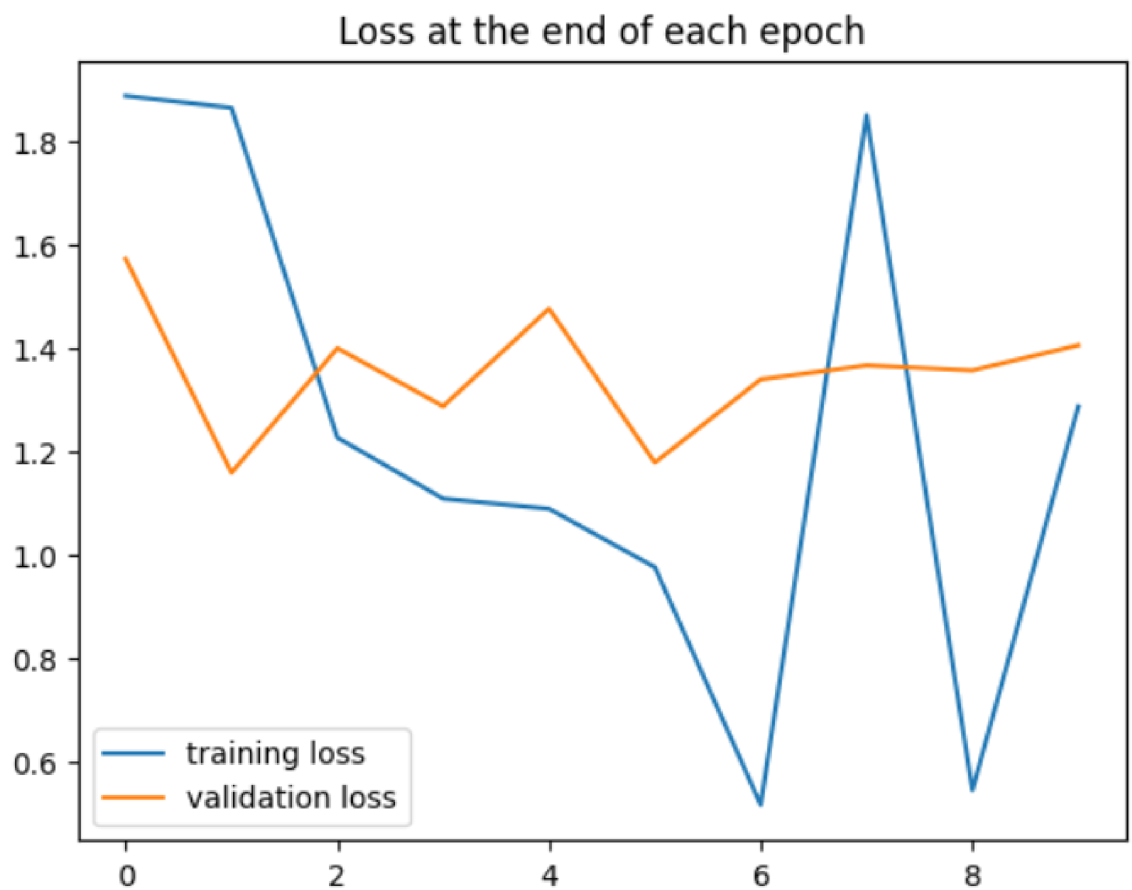
__**Dataset used :**__ We are transitioning from applying CNNs exclusively to grayscale images to extending our focus to images with three color channels, such as RGB.

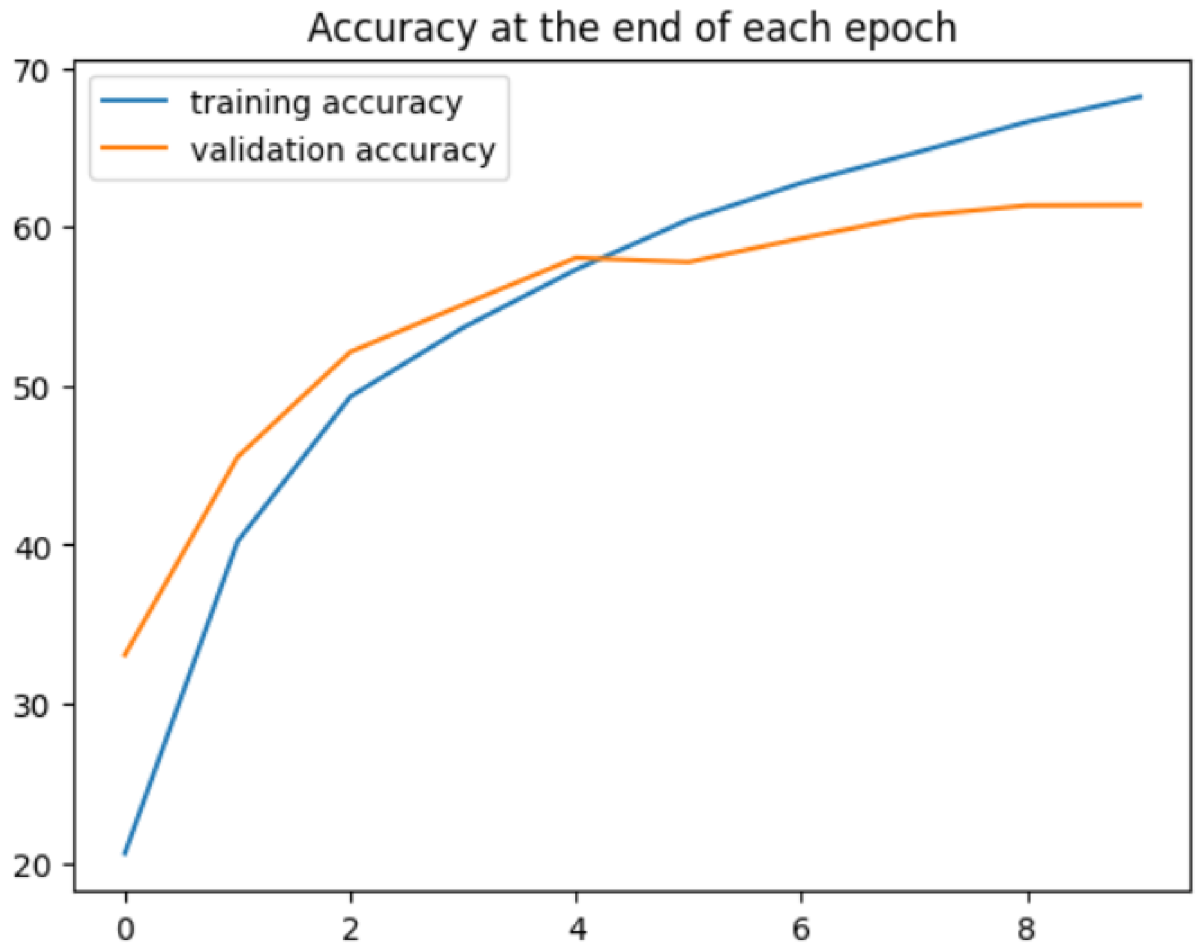


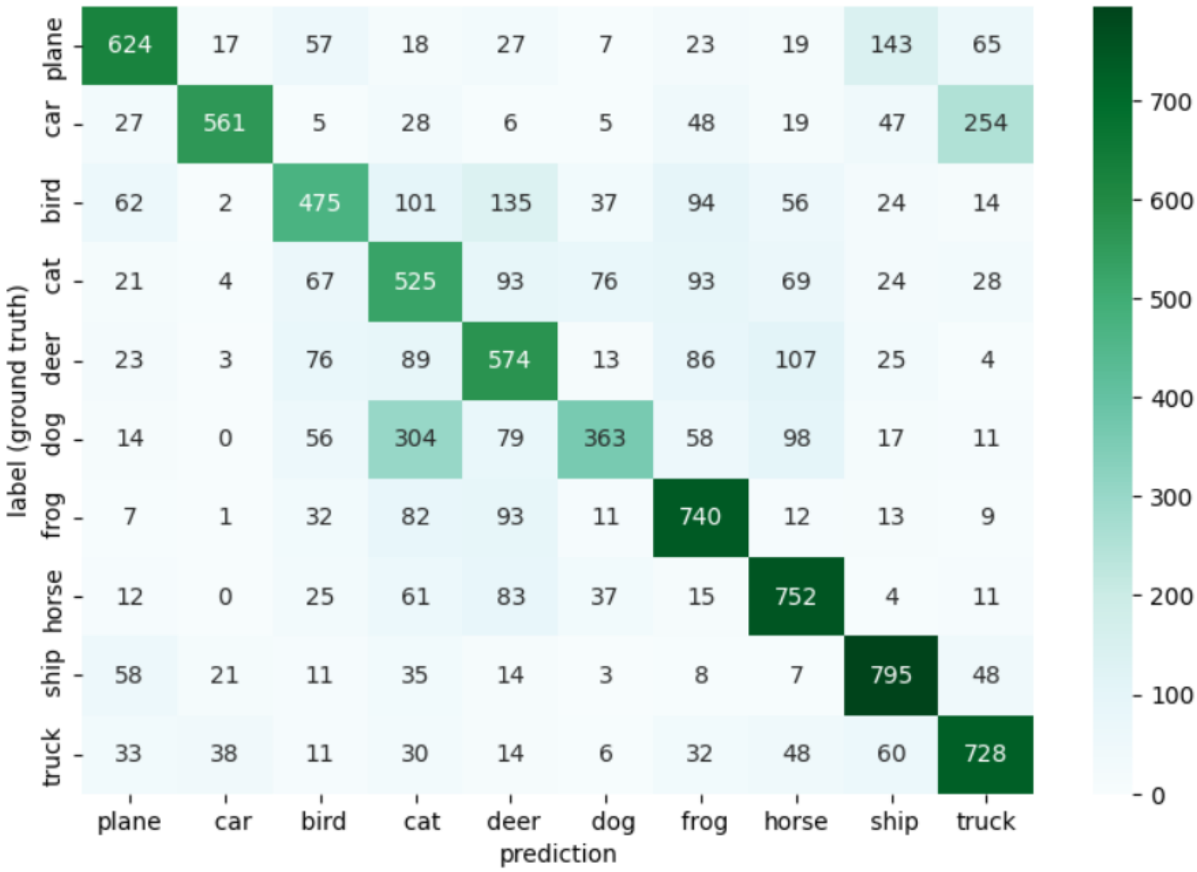
We are currently working with this equation on a classification ML model with a RGB dataset.

Analysis with each optimiser :

1. Analysis of the equation presented by us `Test accuracy: 61.370%`



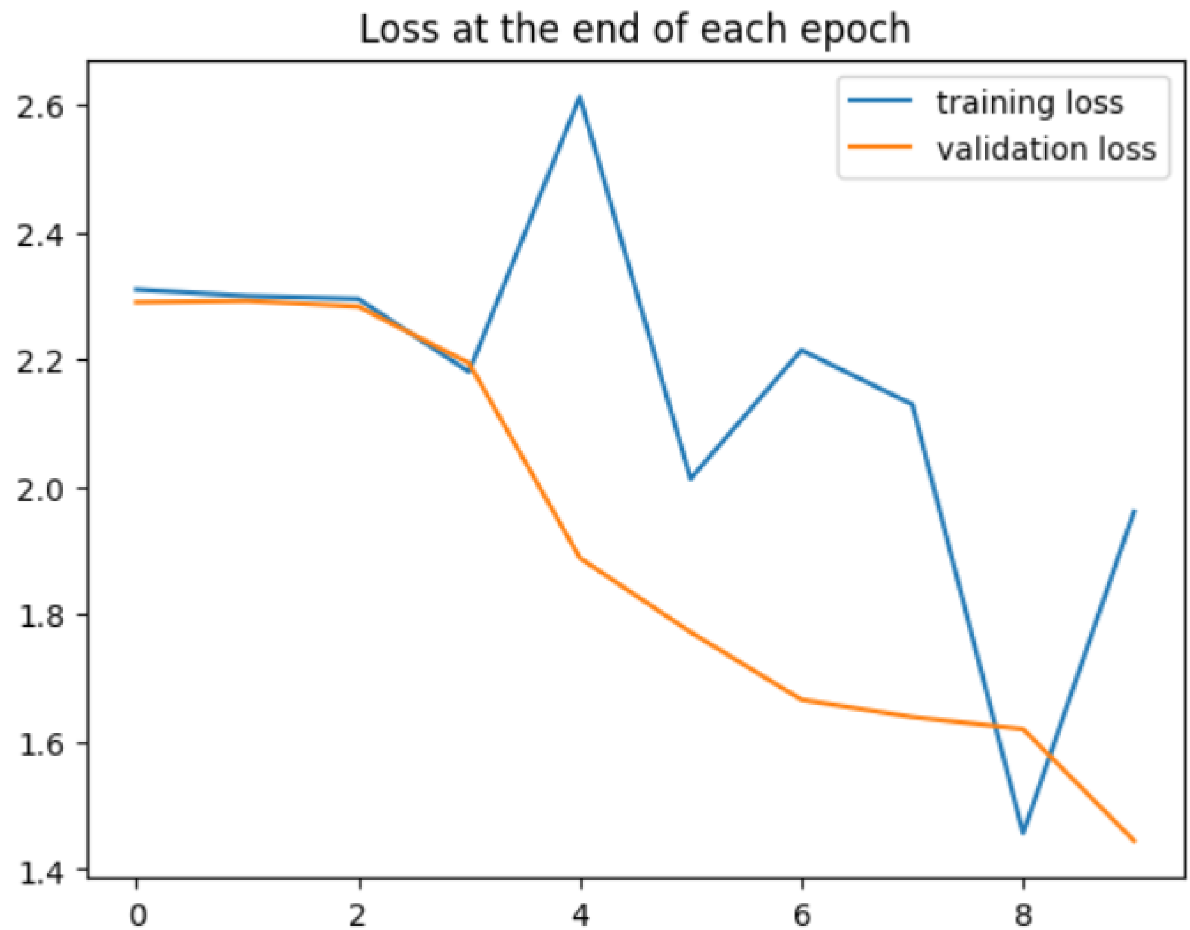

Accuracy at the end of each epoch
training accuracy
validation accuracy
70
60
50
40
30
20
0
2
4
6
8

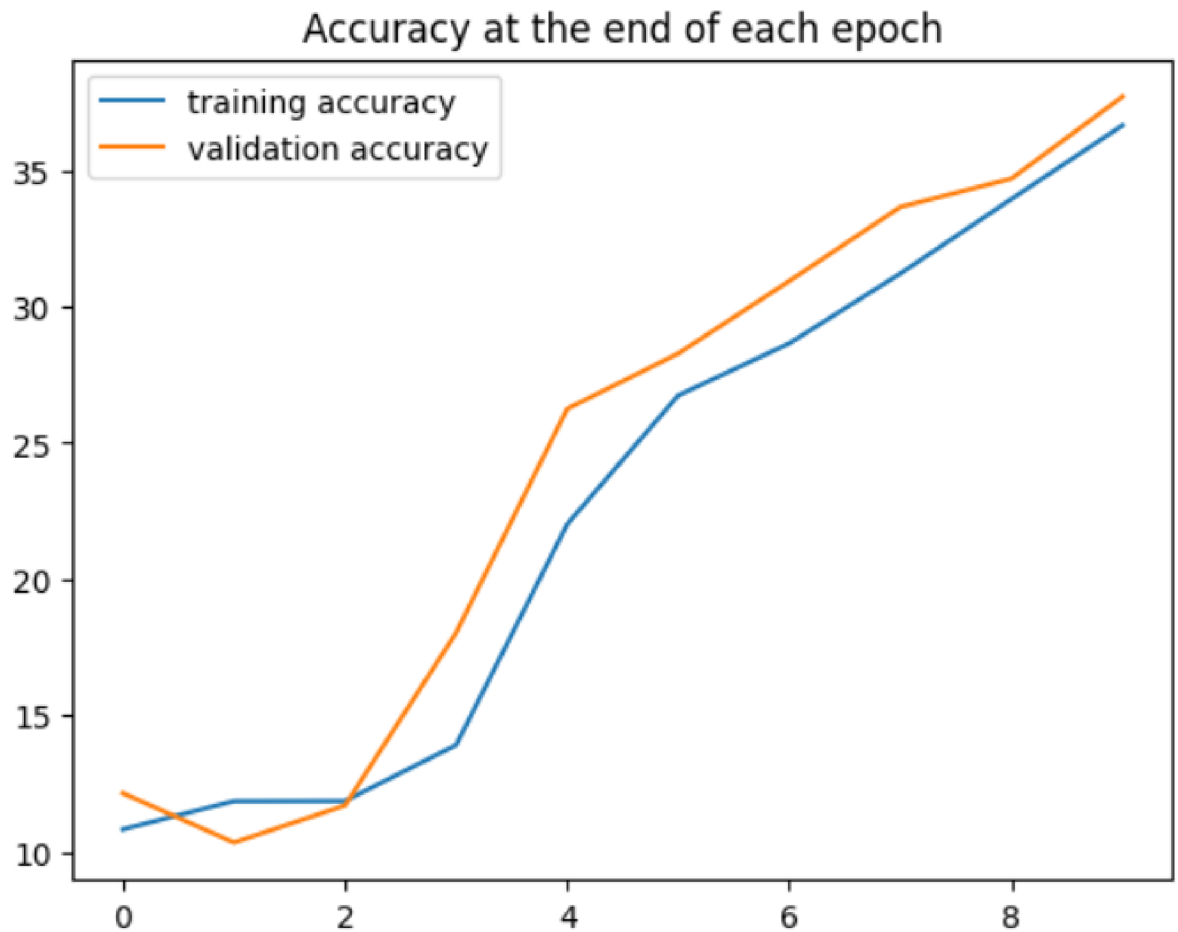


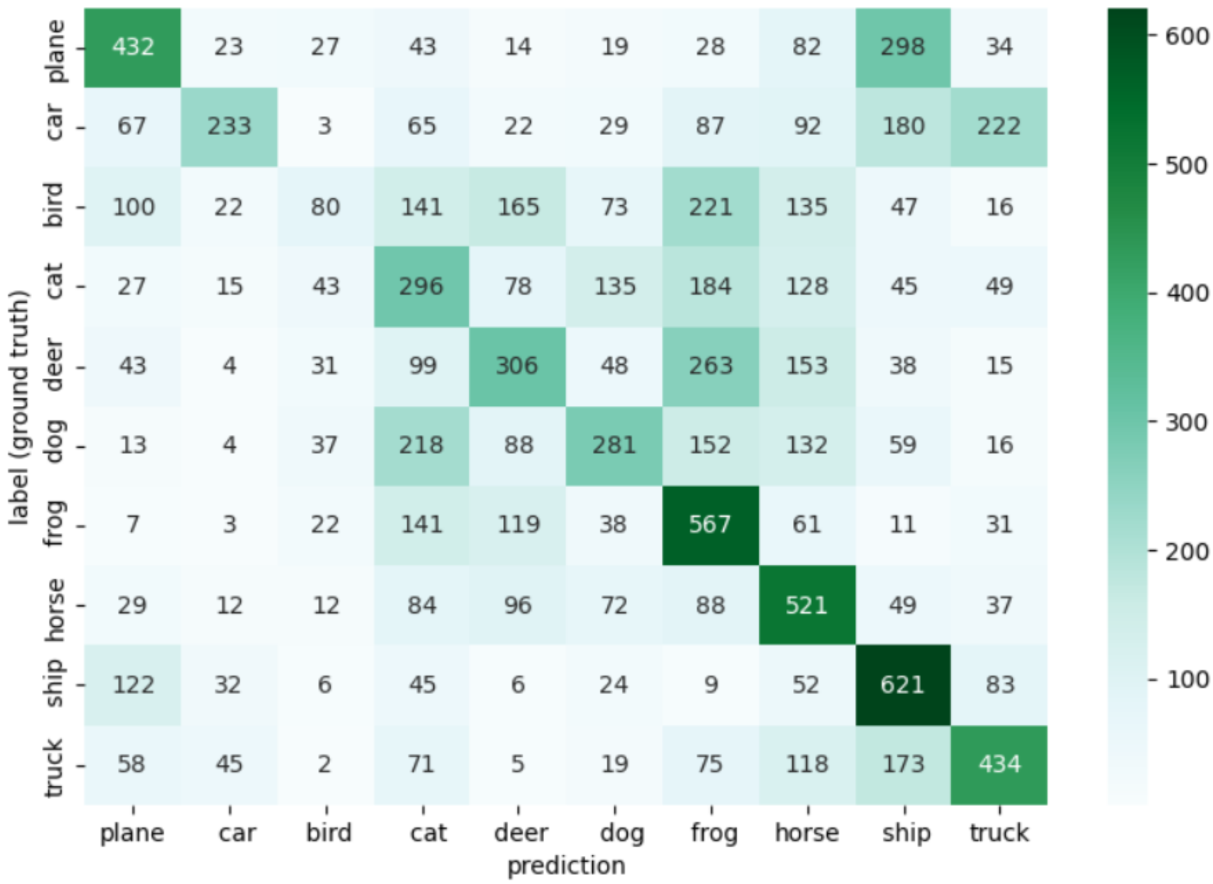
label (ground truth)
plane
car
bird
cat
deer
dog
frog
horse
ship
truck
624 17 57 18 27 7 23 19 143 65
27 561 5 28 6 5 48 19 47 254
62 2 475 101 135 37 94 56 24 14
21 4 67 525 93 76 93 69 24 28
23 3 76 89 574 13 86 107 25 4
14 0 56 304 79 363 58 98 17 11
7 1 32 82 93 11 740 12 13 9
12 0 25 61 83 37 15 752 4 11
58 21 11 35 14 3 8 7 795 48
33 38 11 30 14 6 32 48 60 728
plane
car
bird
cat
deer
dog
frog
horse
ship
truck
prediction
700
600
500
400
300
200
100
0

2. Analysis of Stochastic Gradient

`Test accuracy: 37.710%`

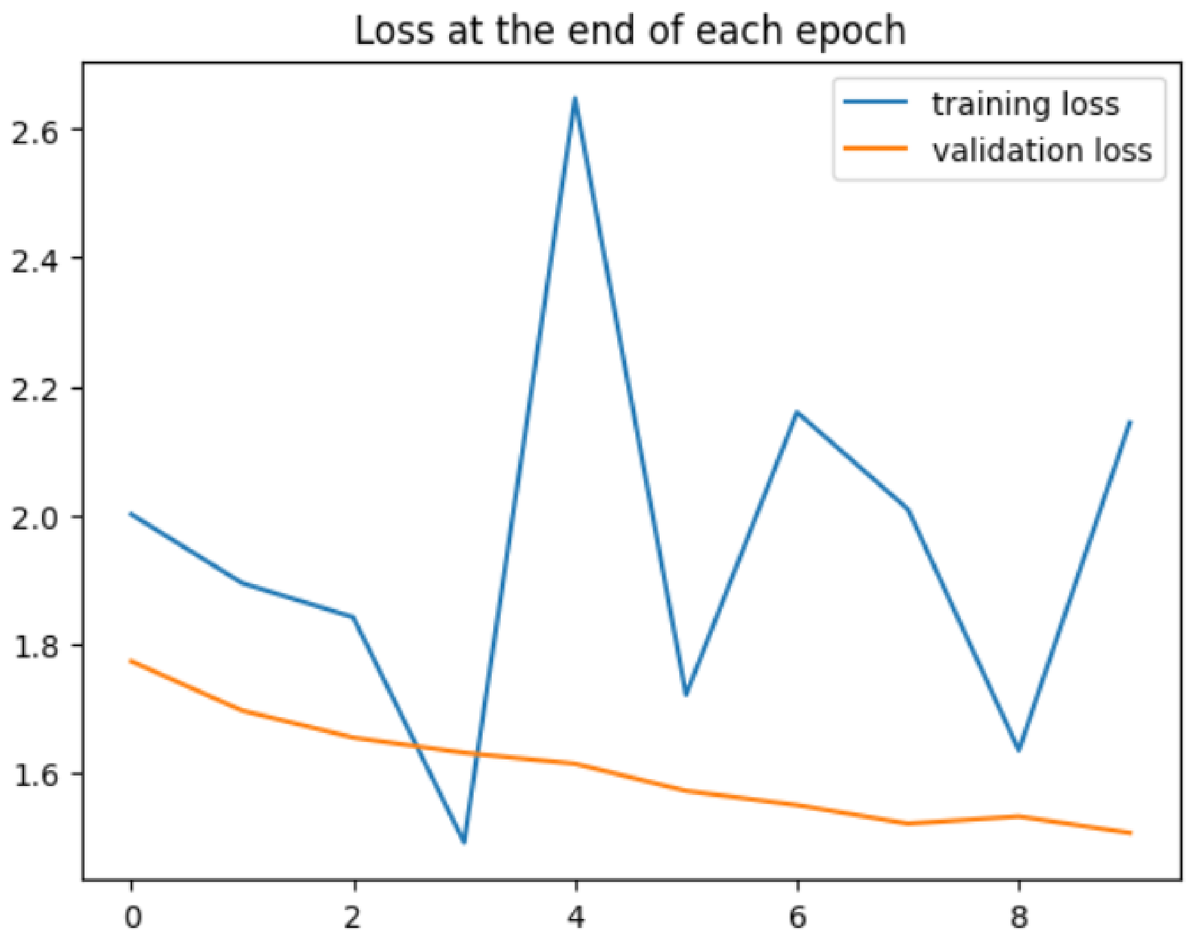


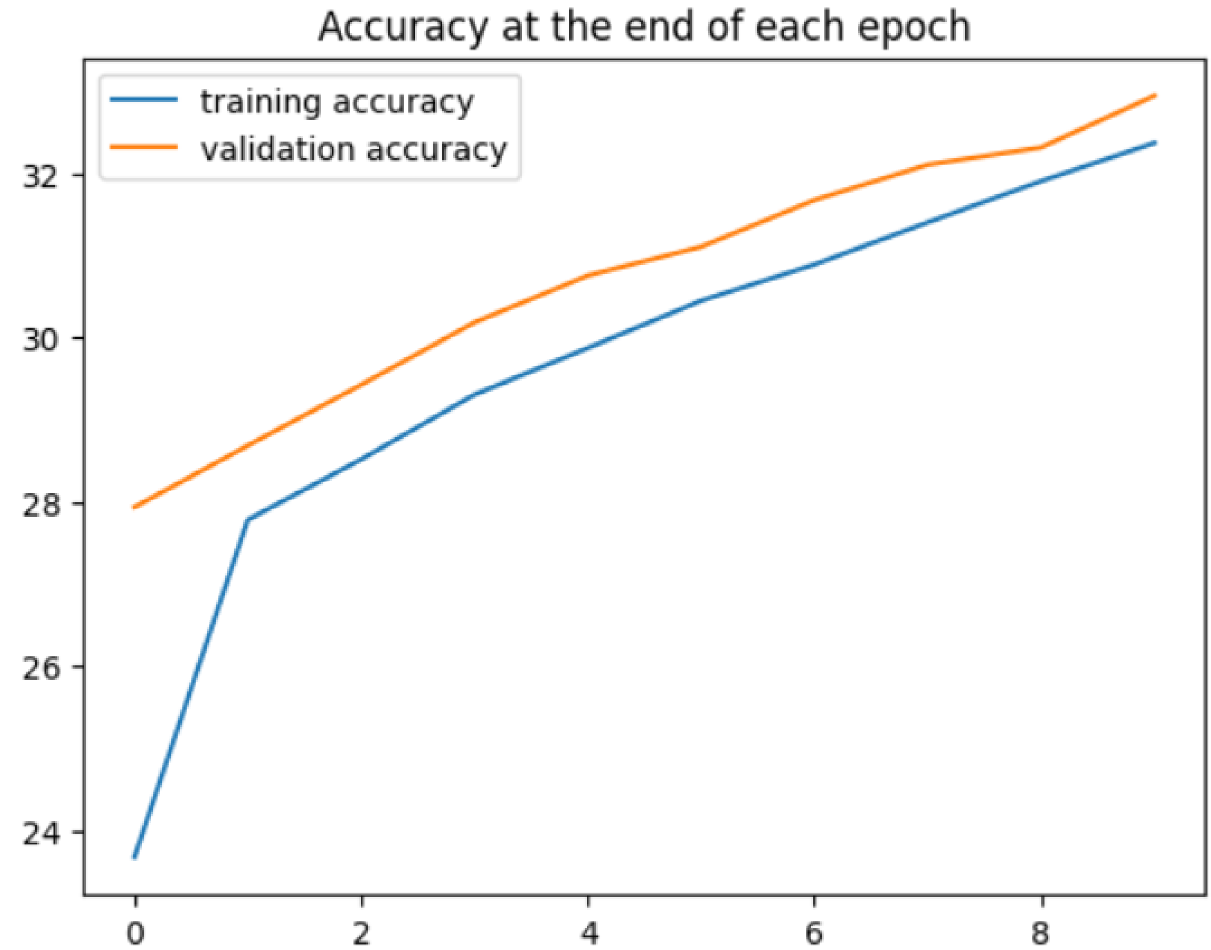

Accuracy at the end of each epoch
training accuracy
validation accuracy
35
30
25
20
15
10
0
2
4
6
8

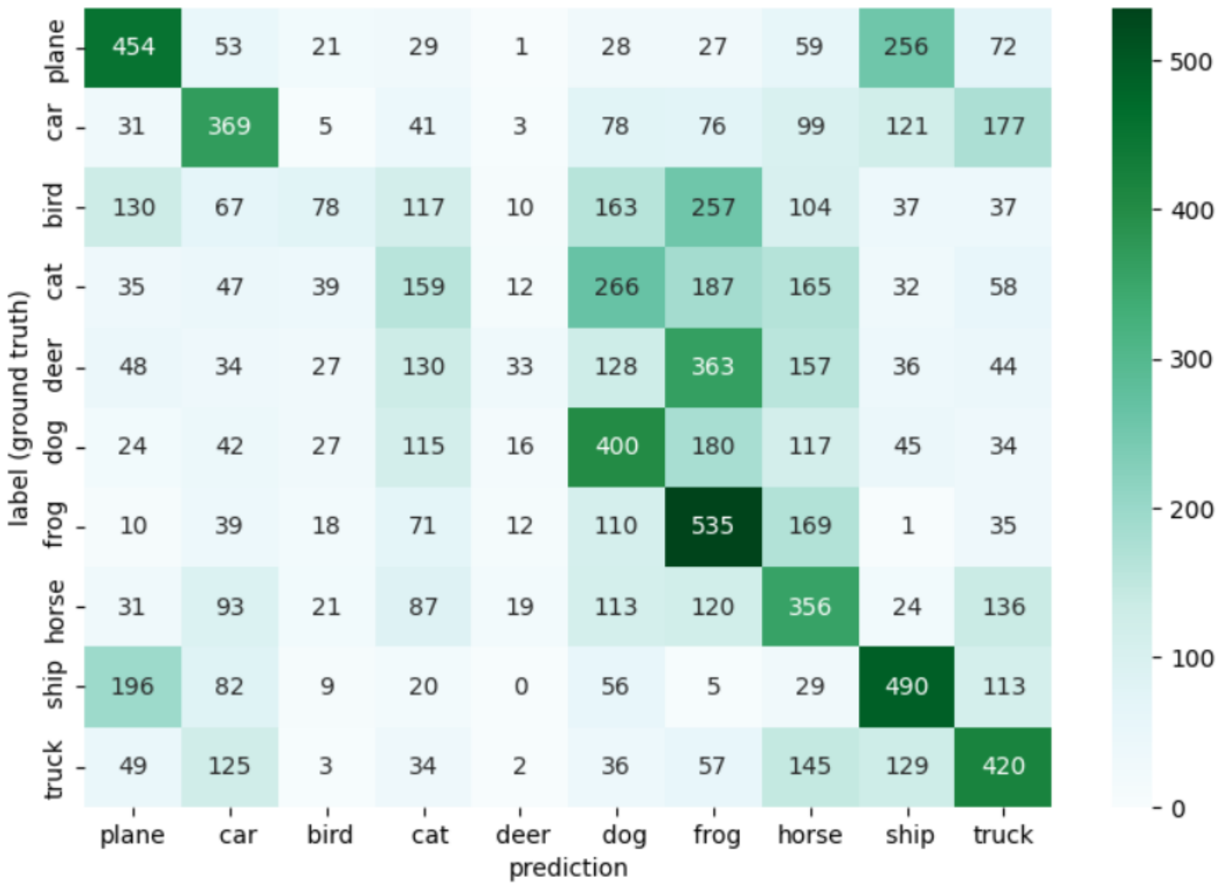


label (ground truth)
plane
car
bird
cat
deer
dog
frog
horse
ship
truck
prediction
600
500
400
300
200
100

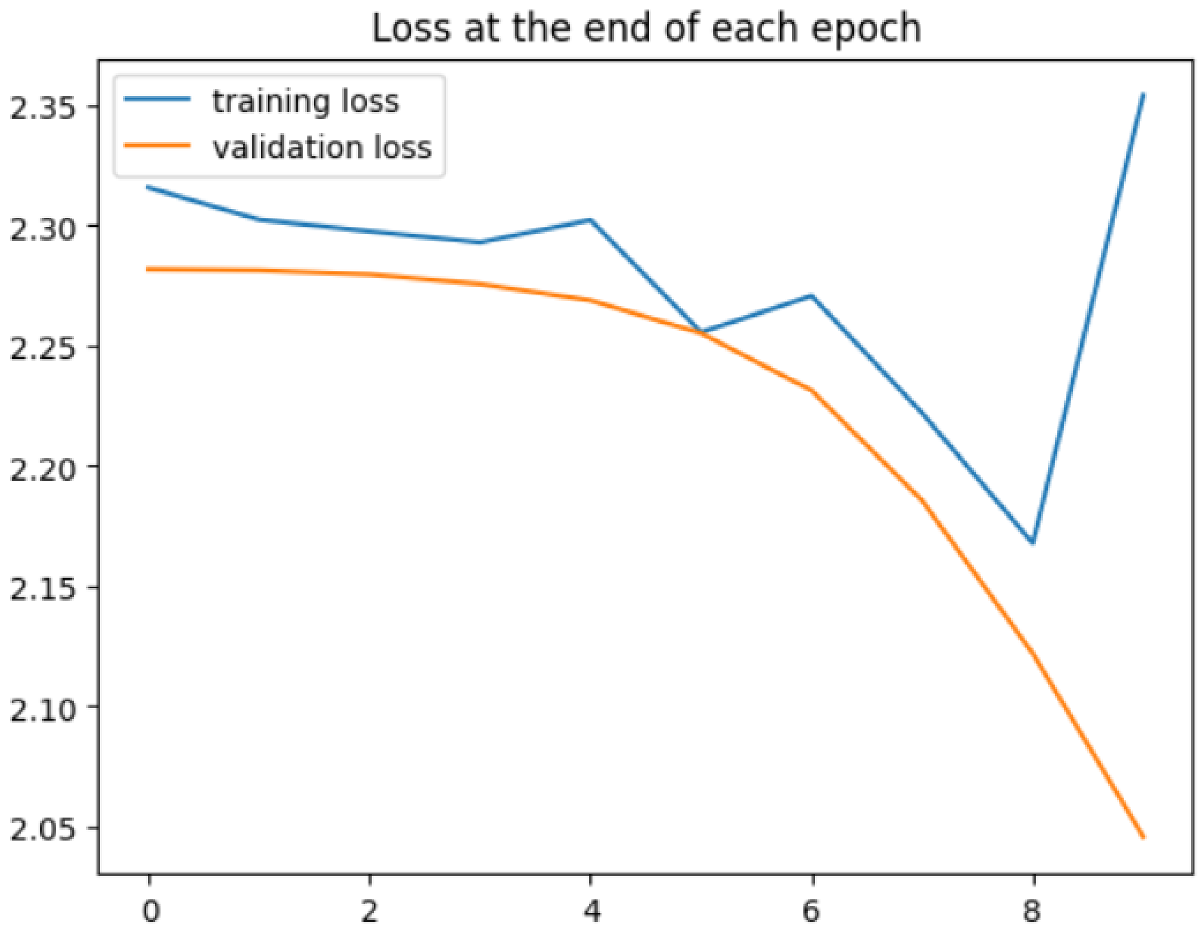
3. Analysis of Adagrad `Test accuracy: 32.940%`

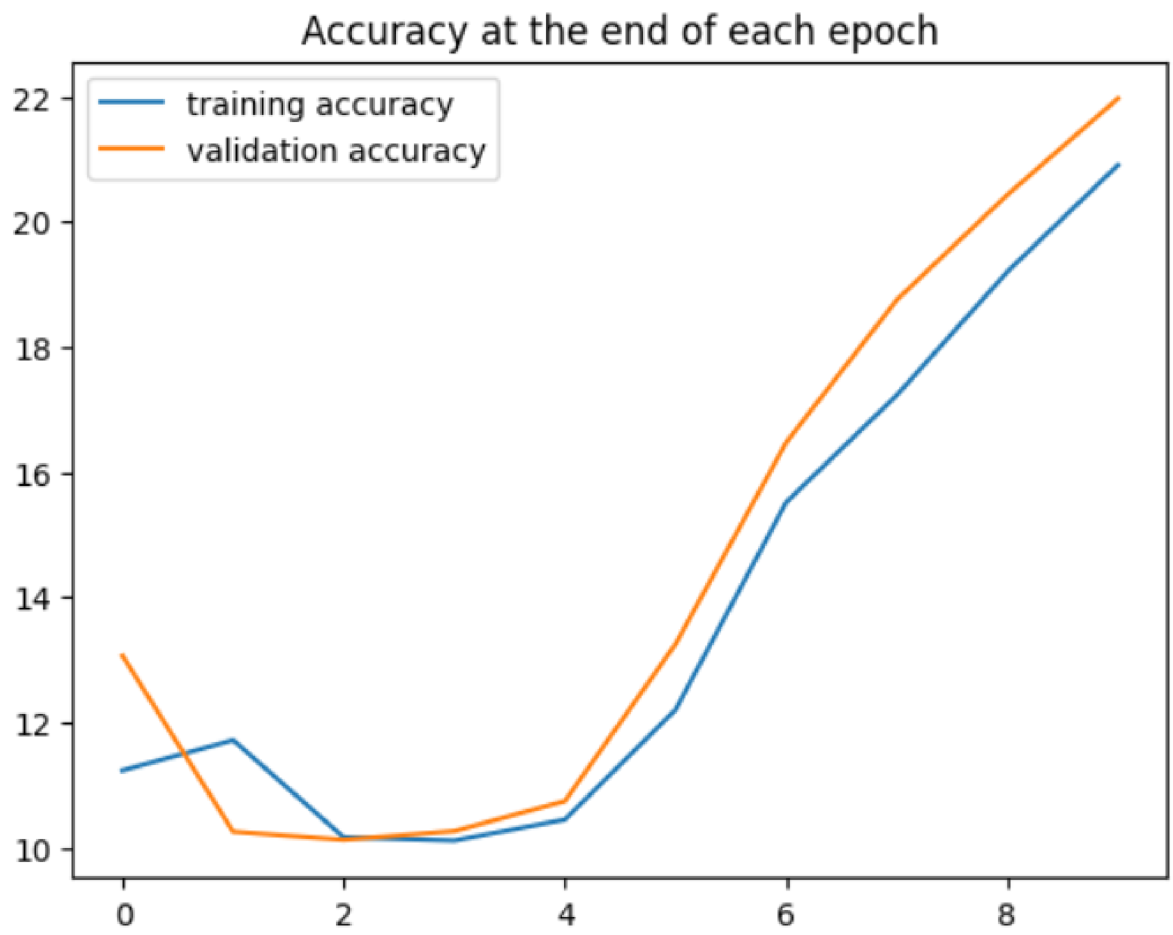



Accuracy at the end of each epoch
training accuracy
validation accuracy
32
30
28
26
24
0
2
4
6
8

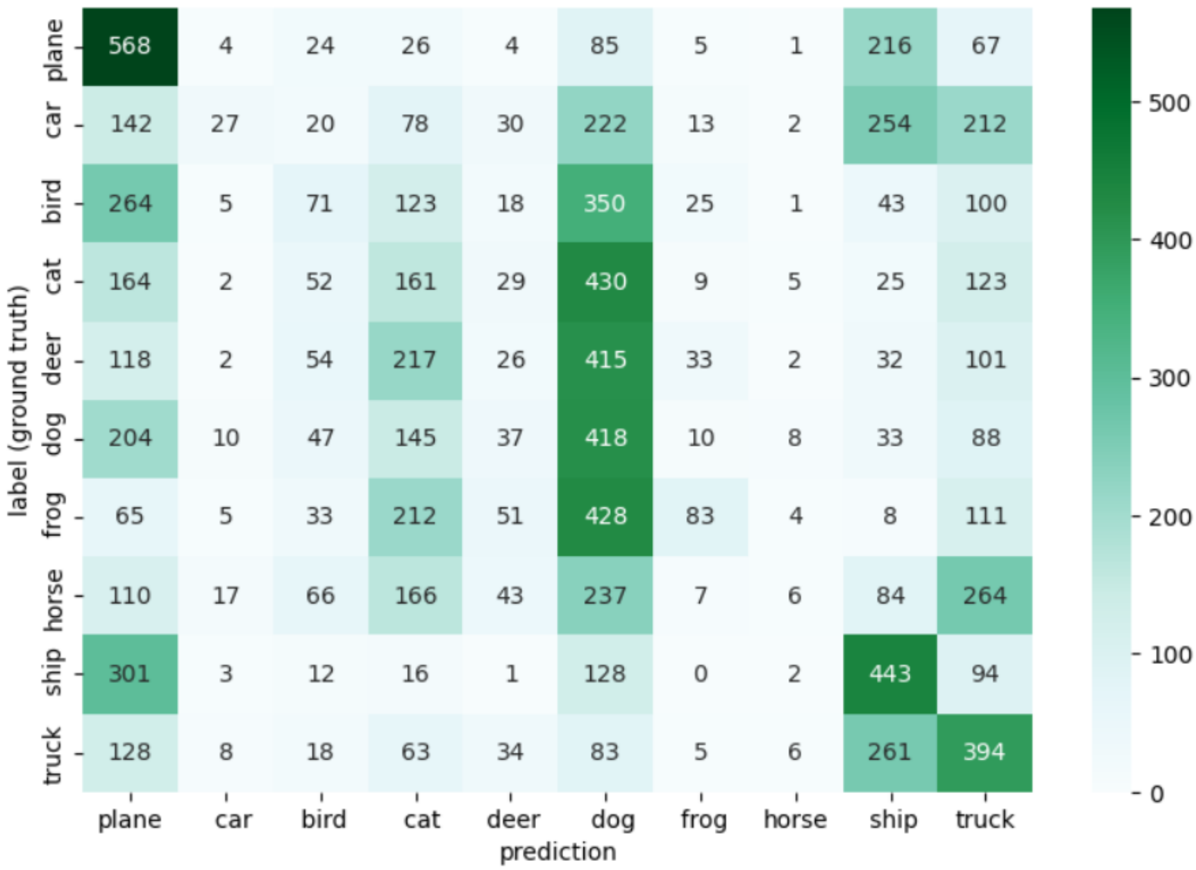

label (ground truth)
prediction
plane
car
bird
cat
deer
dog
frog
horse
ship
truck
454 53 21 29 1 28 27 59 256 72
31 369 5 41 3 78 76 99 121 177
130 67 78 117 10 163 257 104 37 37
35 47 39 159 12 266 187 165 32 58
48 34 27 130 33 128 363 157 36 44
24 42 27 115 16 400 180 117 45 34
10 39 18 71 12 110 535 169 1 35
31 93 21 87 19 113 120 356 24 136
196 82 9 20 0 56 5 29 490 113
49 125 3 34 2 36 57 145 129 420
500
400
300
200
100
0

4. Analysis of Ada Delta **`Test accuracy: 21.970%`**





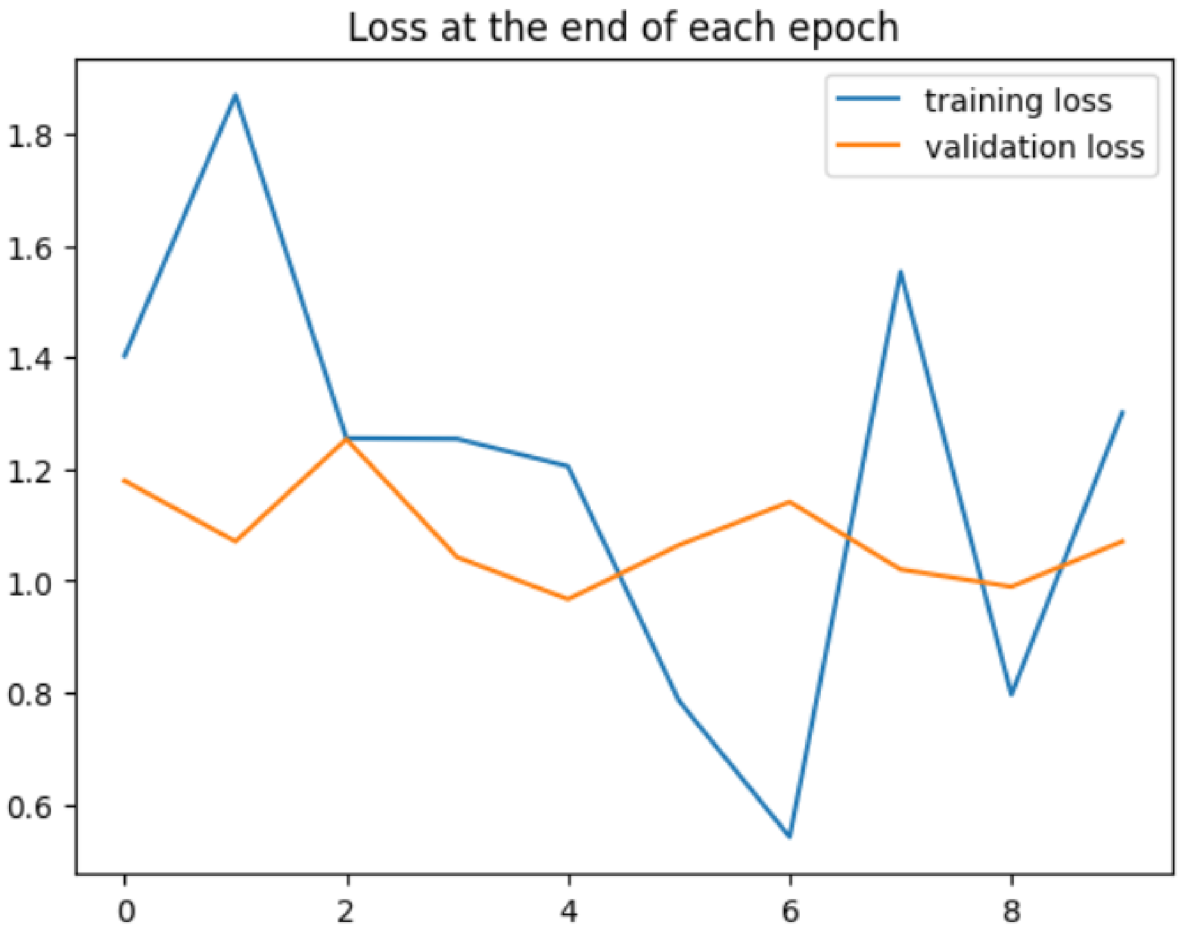
Accuracy at the end of each epoch
training accuracy
validation accuracy
22
20
18
16
14
12
10
0
2
4
6
8

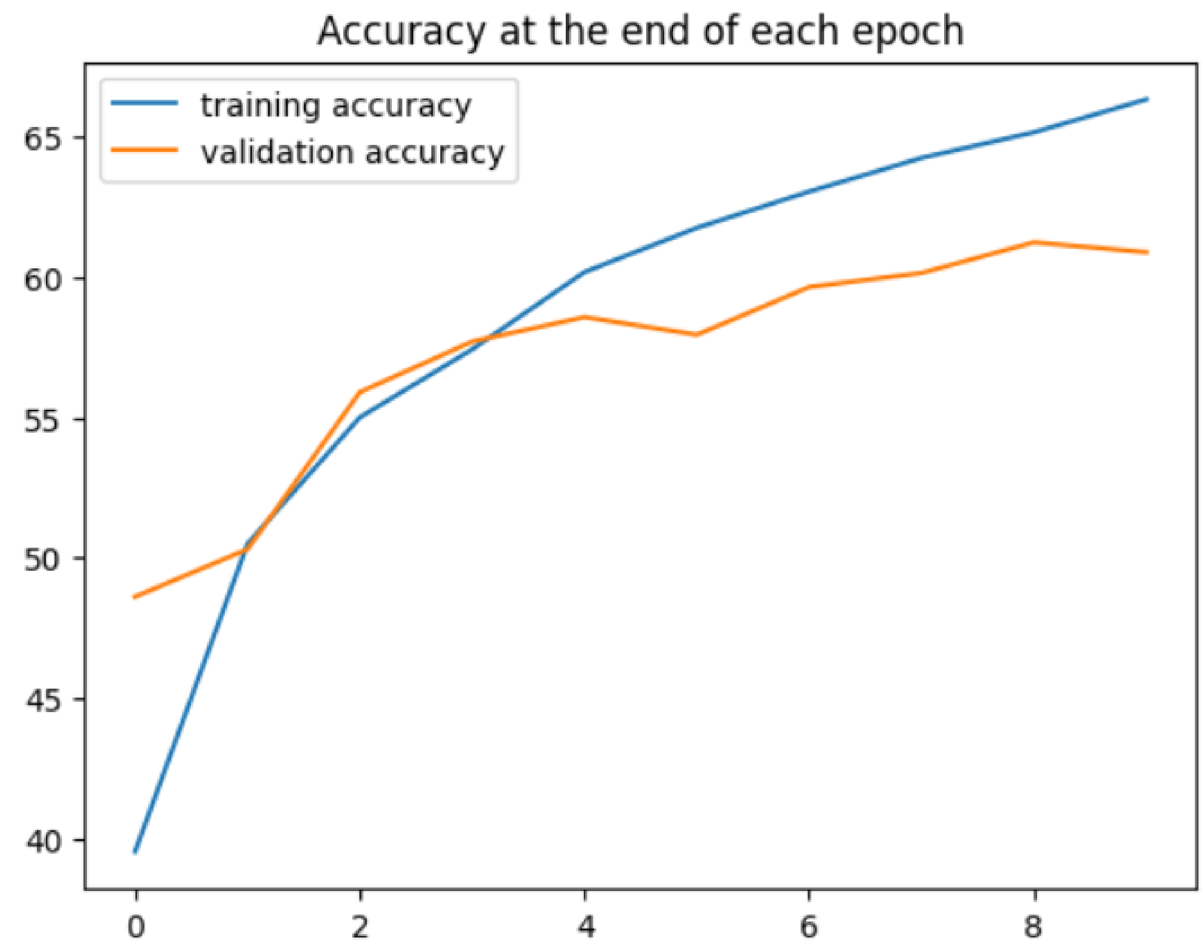


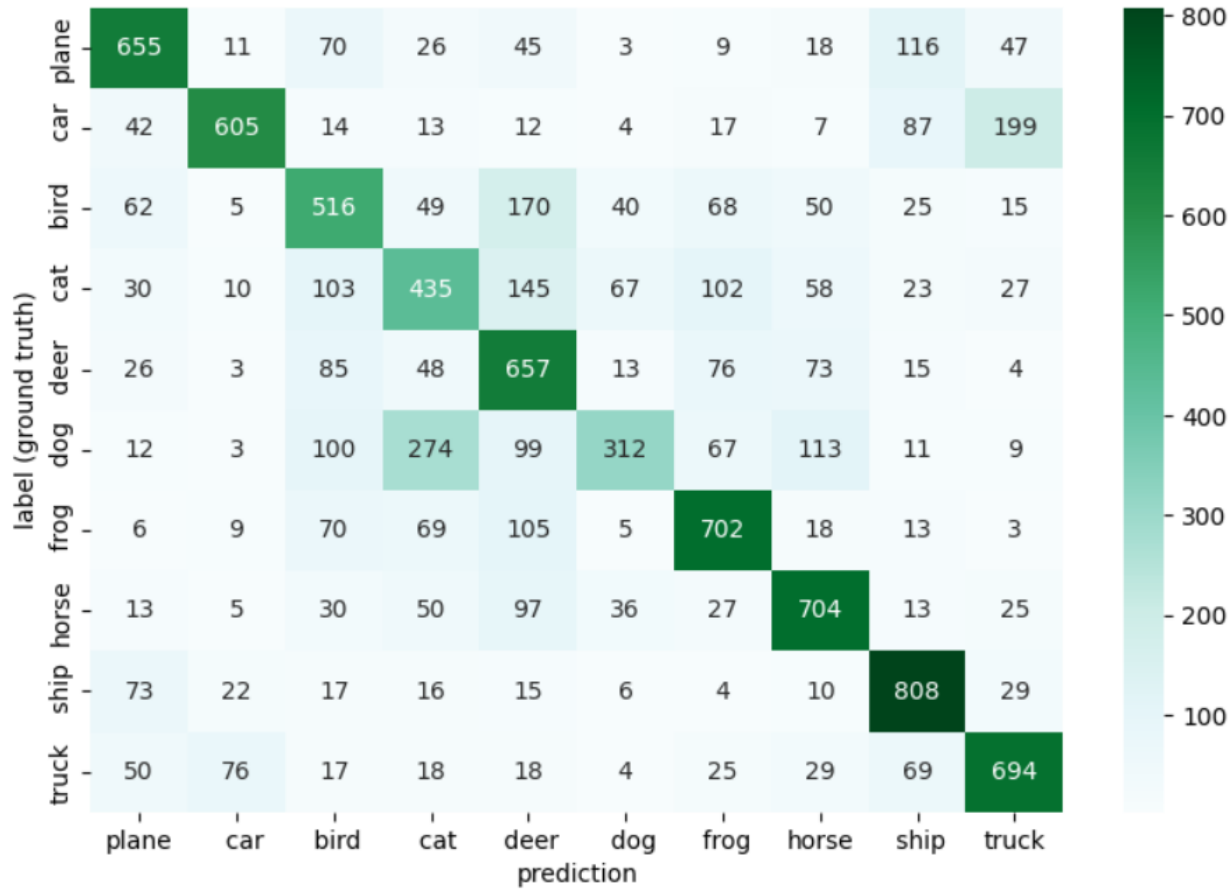
label (ground truth)
prediction
plane
car
bird
cat
deer
dog
frog
horse
ship
truck
568 4 24 26 4 85 5 1 216 67
142 27 20 78 30 222 13 2 254 212
264 5 71 123 18 350 25 1 43 100
164 2 52 161 29 430 9 5 25 123
118 2 54 217 26 415 33 2 32 101
204 10 47 145 37 418 10 8 33 88
65 5 33 212 51 428 83 4 8 111
110 17 66 166 43 237 7 6 84 264
301 3 12 16 1 128 0 2 443 94
128 8 18 63 34 83 5 6 261 394
500
400
300
200
100
0

5. Analysis of Adam

`Test accuracy: 60.880%`




Accuracy at the end of each epoch
training accuracy
validation accuracy
65
60
55
50
45
40
0
2
4
6
8



Conclusion : We are calling this optimiser Gradual Gradient Descent Momentum (GGDM) Method and its comparison with other optimiser is as follows :

AdaDelta **<** Adagrad **<** Stochastic Gradient **<** Adam **<** GGDM

Test Accuracy : 21.97 32.94 37.71 60.88 61.37

Submitted By : Aastha Kandwal, 22124003
Archit Dabral, 22124011