L-Università ta' Malta
Faculty of Information &
Communication Technology

Advanced Computer Vision Individual Assignment

Matthias Bartolo

B.Sc. It (Hons) Artificial Intelligence (Third Year)

Study Unit Code:	**ARI3129**
Study Unit:	**Advanced Computer Vision for Artificial Intelligence**
Lecturer:	**Dr Dylan Seychell**
Date:	**17 October 2023**

Contents

1 Neural Networks & TensorFlow Playground

Note that the prompts utilised for the ChatGPT section, can be accessed through the following link:
https://chat.openai.com/share/6ba74944-49dc-419c-a8c1-e623ad7cbfdd

1.1 Task 1: Understanding Activation Functions

1.1.1 Question

1. Create a neural network with just one hidden layer from [1].
2. Experiment with various activation functions like ReLU, Sigmoid, and Tanh.
3. **Objective:**
    - Note the number of epochs for the model to converge with each activation function.
    - Observe if some activation functions fail to model certain patterns in the data.

1.1.2 Answer

The created neural network with one hidden layer utilised for this experiment and which can be seen in Figure 1.1 has the following properties:

1. Hidden Layer consisting of four Neurons
2. Learning Rate set to 0.03
3. Problem Type set to Classification
4. Regularisation set to None
5. Dataset set to Circle

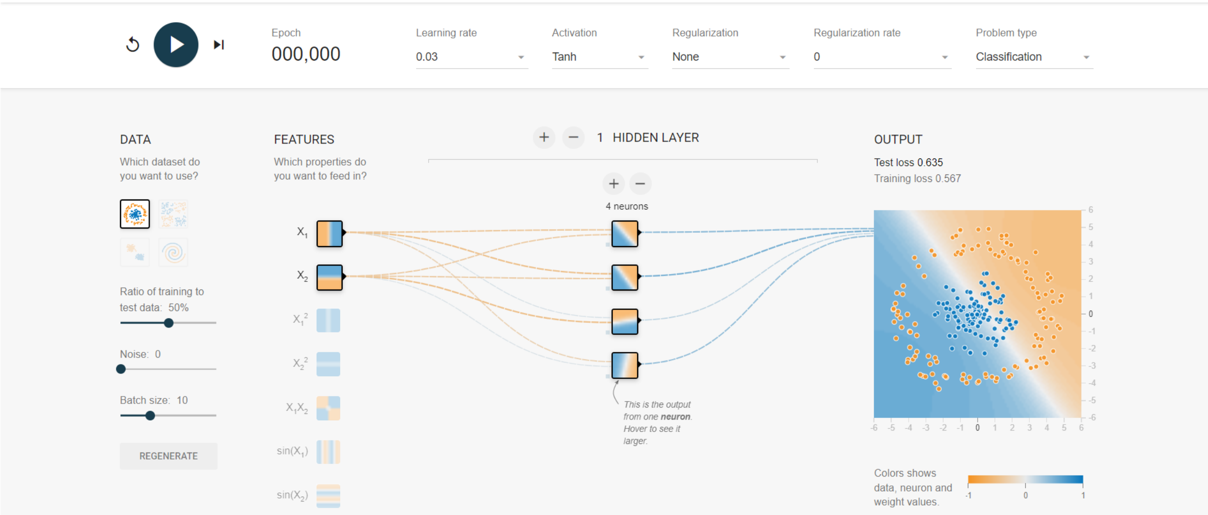

Figure 1.1 Showing the created neural network with one hidden layer.

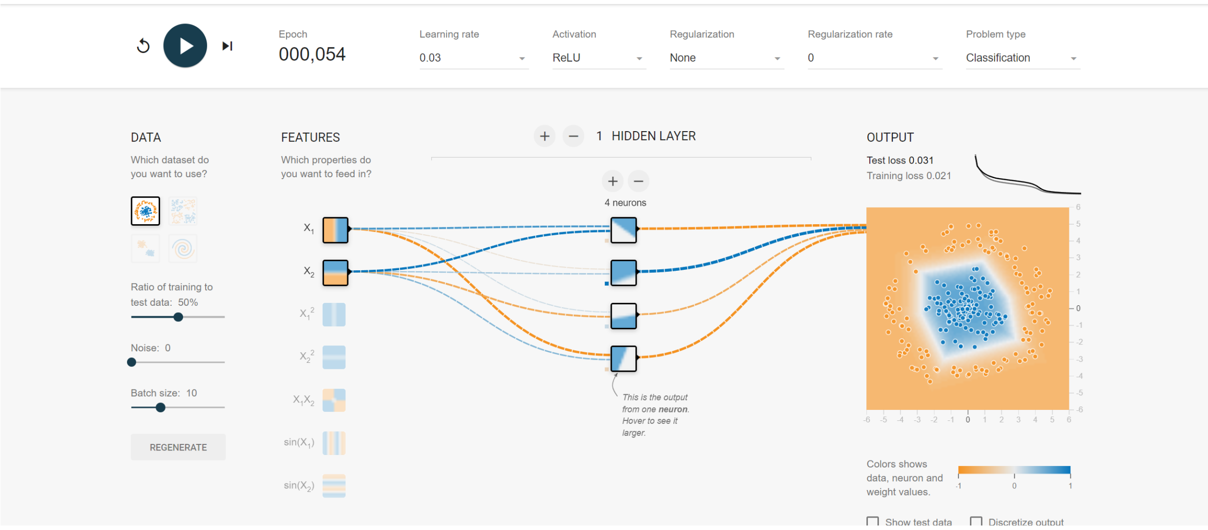

Figure 1.2 Showing the created neural network which learned to fit the Circle dataset.

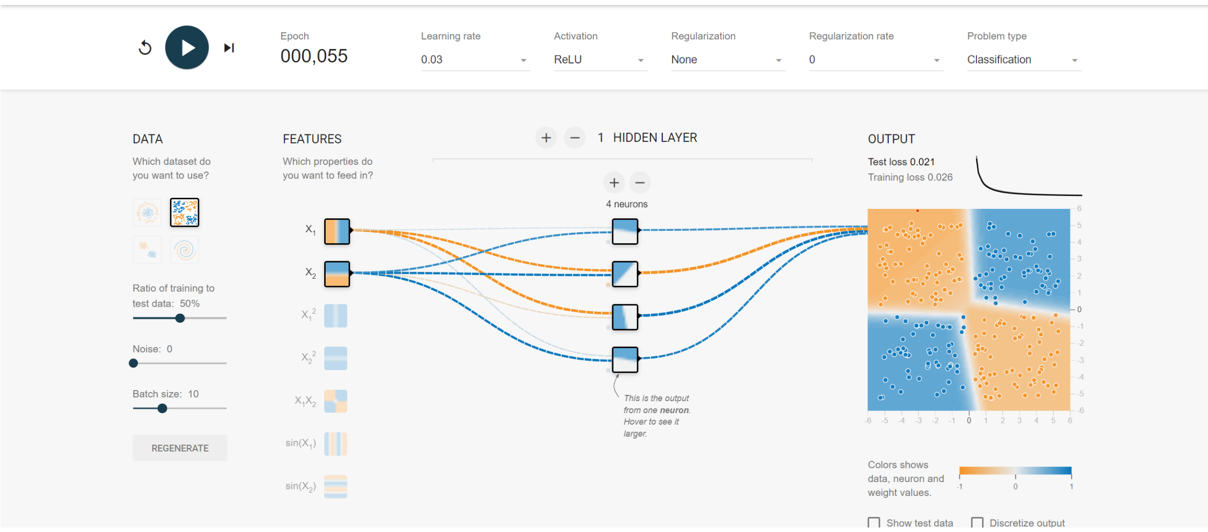

Figure 1.3 Showing the created neural network which learned to fit the Exclusive or dataset.

Activation Function	Epochs converged	Converged
ReLU	54	Yes
Tanh	100	Yes
Sigmoid	200	Yes
Linear	∞	No

Table 1.1 Results obtained for changing the Activation Function in the neural network shown in Figure 1.1, using the Circle dataset.

Activation Function	Epochs converged	Converged
ReLU	55	Yes
Tanh	90	Yes
Sigmoid	250	Yes
Linear	∞	No

Table 1.2 Results obtained for changing the Activation Function in the neural network shown in Figure 1.1, using the Exclusive or dataset.

Activation Function	Epochs converged	Converged
ReLU	∞	No
Tanh	∞	No
Sigmoid	∞	No
Linear	∞	No

Table 1.3 Results obtained for changing the Activation Function in the neural network shown in Figure 1.1, using the Spiral dataset.

Activation Function	Epochs converged	Converged
ReLU	10	Yes
Tanh	15	Yes
Sigmoid	20	Yes
Linear	15	Yes

Table 1.4 Results obtained for changing the Activation Function in the neural network shown in Figure 1.1, using the Gaussian dataset.

From the above results in Tables: 1.1, 1.2, 1.4, and 1.3, it was noted that generally ReLU is the best performing function which enables the neural network to converge the fastest on the given datasets. Additionally, it was also noted that Tanh was the second best performing function, followed by Sigmoid, then Linear. It was generally noted, that the best three activation functions always converged, except for the instance of the Spiral dataset, where neither of the models could fit the dataset, due to the fact that only a single hidden layer is being used. Moreover, it was also noted that the Linear activation function did not converge for the Circle and Exclusive or datasets, as the model could not fit the data. However, this was not the case for the Gaussian dataset, as all of the activation functions managed to enable the created neural network to converge.

1.1.3 Refining and Validating Answer with ChatGPT

Prompting ChatGPT with the following:
"How could the above paragraph be improved, without drastically increasing the word limit, both from a research and academic English perspective?"

Results Obtained:
"The results presented in Tables 1.1, 1.2, 1.4, and 1.3 indicate that the ReLU activation function consistently outperformed others in terms of facilitating faster convergence in the neural network on the provided datasets. Following closely, the Tanh activation function emerged as the second-best performer, with Sigmoid and Linear being subsequent choices. It is noteworthy that, in most cases, the top three activation functions demonstrated effective convergence. However, a noteworthy exception was observed with the Spiral dataset, where none of the models could adequately capture the dataset's complexity due to the utilization of only a single hidden layer.

Furthermore, it is important to mention that the Linear activation function failed to converge when applied to the Circle and Exclusive OR datasets, indicating its limitations in modeling these particular datasets. However, this was not the case with the Gaussian dataset, where all activation functions successfully enabled the neural network to converge.

In summary, these findings suggest that the choice of activation function plays a crucial role in the convergence and performance of neural networks across various datasets. The results provide valuable insights into the effectiveness of different activation functions in different scenarios, shedding light on their suitability for specific tasks."

1.2 Task 2: Exploring Learning Rates

1.2.1 Question

1. Use a pre-set neural network architecture for this task.
2. Experiment with learning rates.
3. Objective:
    - Record the number of epochs it takes to converge for each learning rate.
    - Note any oscillations or overshooting in the loss graph for high learning rates.

1.2.2 Answer

The pre-set neural network architecture was set to have the following properties:

1. The default number of hidden layers was set to two
2. Activation Function set to ReLU
3. Problem Type set to Classification
4. Regularisation set to None
5. Dataset set to Circle

Learning Rate	Epochs converged	Converged	Oscillating
10	∞	No	Yes
3	∞	No	Yes
1	26	Yes	No
0.3	19	Yes	No
0.1	20	Yes	No
0.03	50	Yes	No
0.01	100	Yes	No
0.003	300	Yes	No
0.001	700	Yes	No
0.0001	∞	No	No
0.00001	∞	No	No

Table 1.5 Results obtained for changing the Learning Rate in pre-set neural network architecture, using the Circle dataset and ReLU activation function.

From the results obtained in Table 1.5, it was noted that learning rates between 0.1 and 1 present the optimal learning rates given the specified scenario and Circle dataset. Additionally, it was also noted that high learning rates, which were larger than 1, tended to overshoot and display oscillations in the output graph. Moreover, it was also noted that a small learning rate such as 0.0001 and high learning rates such as 10 did not enable the neural network to converge.

1.2.3 Refining and Validating Answer with ChatGPT

Prompting ChatGPT with the following:
"How could the above paragraph be improved without drastically increasing the word limit while also managing to include other details that were omitted? Additionally, try to also improve it from an academic English and AI researcher perspective."

Results Obtained:
"Analysis of the results presented in Table 1.5 reveals crucial insights into the selection of optimal learning rates within the specified context, particularly when dealing with the Circle dataset. Notably, learning rates ranging from 0.1 to 1 emerged as the most effective choices, exhibiting favorable convergence characteristics.

Furthermore, it is essential to highlight that higher learning rates, exceeding a value of 1, exhibited a tendency to overshoot during training, resulting in oscillations within the output graph. This observation underscores the importance of avoiding excessively high learning rates, as they can impede the stability of the learning process and hinder convergence.

Conversely, excessively low learning rates, exemplified by the value of 0.0001, and extremely high learning rates, such as 10, were found to be suboptimal. These learning rates failed to facilitate the desired convergence of the neural network. This insight is instrumental in guiding the selection of learning rates for training neural networks, as it underscores the necessity of striking a balance between rates that are neither too small nor excessively large.

In summary, the findings from this analysis underscore the critical role that the choice of learning rate plays in the training process of neural networks. The identified optimal range of learning rates, coupled with the cautionary notes on extreme values, offers valuable guidance to AI researchers and practitioners seeking to achieve efficient and stable training of neural networks in similar scenarios."

1.3 Task 3: Playing with Layers and Neurons

1.3.1 Question

1. Start with a neural network that has one hidden layer.
2. Incrementally add more layers and neurons to the hidden layers.
3. Objective:
   - Compare the decision boundary's complexity when using few layers and neurons versus more layers and neurons.
   - Note the number of epochs it takes for the model to converge as you increase the complexity

1.3.2 Answer

The pre-set neural network architecture was set to have the following properties:

1. The learning rate was set to 0.01
2. Activation Function set to ReLU
3. Problem Type set to Classification
4. Regularisation set to None
5. Dataset set to Circle

Hidden Layers	Number of Neurons	Epochs converged	Converged
1	1	∞	No
1	3	100	Yes
1	5	58	Yes
2	3	70	Yes
2	5	61	Yes
3	3	100	Yes
3	5	60	Yes

Table 1.6 Results obtained for changing the number of hidden layers in pre-set neural network architecture, using the Circle dataset and ReLU activation function.

From the results obtained in Table 1.6, it was noted that with less hidden layers and neurons, the decision boundary is simple and may struggle to converge when trying to learn complex data patterns. On the other hand, a large amount of hidden layers and neurons in the neural network, would enable the model's capacity to learn complex data patterns, resulting in a decision boundary that is more complex and can better suit the training data. Additionally, it was also noted that with more layers and neurons, the number of epochs required to converge is less than when compared to the same neural network with less neurons and layers. Furthermore, when only one neuron was utilised, the decision boundary was only a straight line; therefore, more neurons and hidden layers are needed to make decision borders more complicated.

1.3.3 Refining and Validating Answer with ChatGPT

Prompting ChatGPT with the following:
"How could the above paragraph be improved without drastically increasing the word limit while also managing to include other details that were omitted? Additionally, try to also improve it from an academic English and AI researcher perspective. Finally, compare these observations with the current literature, and discuss the validity of these claims."

Results Obtained:
"The findings extracted from the results presented in Table 1.6 yield significant insights into the relationship between the architecture of neural networks and their ability to capture complex data patterns. Notably, when neural networks possess fewer hidden layers and neurons, they tend to manifest simpler decision boundaries, which may struggle to converge effectively when tasked with learning intricate data patterns.

In contrast, a substantial increase in the number of hidden layers and neurons within the neural network enhances its capacity to comprehend intricate data structures. Consequently, this augmentation results in more intricate decision boundaries that align closely with the nuances present in the training data. It is important to emphasize that this observation underscores the role of network architecture in accommodating the complexity of the target data.

Another noteworthy observation pertains to the relationship between network depth and training convergence. Networks with greater depth and neuron count exhibit a tendency to converge in fewer epochs compared to their shallower counterparts. This aspect is critical in optimizing training efficiency and expediting the convergence process.

Moreover, the distinct behavior when utilizing only a single neuron merits attention. In such cases, the decision boundary is limited to a linear configuration—a mere straight line. To establish more nuanced decision boundaries, a more substantial allocation of neurons and hidden layers becomes imperative.

Comparing these findings with existing literature, it is evident that they align with the broader consensus in the field of deep learning. Researchers have consistently recognized that the architectural complexity of neural networks significantly influences their capacity to capture intricate data patterns. Moreover, the relationship between network depth and faster convergence has been a subject of investigation in various studies, affirming the validity of the observations made in this research.

In summary, the insights gleaned from this analysis underscore the critical role of neural network architecture in adapting to the complexity of data patterns. These observations resonate with established research, affirming their credibility and relevance within the domain of AI research and practice."

1.4 Task 4: Optimisation Algorithms

1.4.1 Question

1. For this task, you are to use the ConvnetJS library [2]. Use a consistent neural network architecture for this task.
2. Change the optimisation algorithm from SGD to others like AdaGrad or Momentum.
3. Objective:
   - Record the number of epochs for the model to converge using different optimisation algorithms and any other observations.
   - Note if specific algorithms are more sensitive to initial conditions or if they plateau more easily.

1.4.2 Answer

The neural network which was used for this experiment, consisted of two fully connected hidden layers, each having the ReLU activation functions, and the softmax activation function for output. Additionally, to facilitate comparisons in this section, the following settings were set to be constant during execution:

1. Batch Size set to 8
2. L2 Decay set to 0.001
3. Learning Rate set to 0.01
4. Activation Function set to ReLU

The following optimization methods were evaluated: SGD, SGD with Momentum (0.91), Adagrad, Adagrad with Momentum (0.91), Windowgrad, Windowgrad with Momentum (0.91), Adadelta, Adadelta with Momentum (0.91), Nesterov, and Nesterov with Momentum (0.91). This is illustrated in code listing 1.1.

```
// lets use an example fully-connected 2-layer ReLU net
var layer_defs = [];
layer_defs.push({type:'input', out_sx:24, out_sy:24, out_depth:1});
layer_defs.push({type:'fc', num_neurons:20, activation:'relu'});
layer_defs.push({type:'fc', num_neurons:20, activation:'relu'});
layer_defs.push({type:'softmax', num_classes:10});

// below fill out the trainer specs you wish to evaluate, and give them
    names for legend
var LR = 0.01; // learning rate
var BS = 8; // batch size
var L2 = 0.001; // L2 weight decay

nets = [];
trainer_defs = [];
trainer_defs.push({learning_rate:LR, method: 'sgd', momentum: 0.0,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'sgd', momentum: 0.91,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'adagrad', momentum: 0.0,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'adagrad', momentum: 0.91,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'windowgrad', momentum: 0.0,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'windowgrad', momentum: 0.91,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'adadelta', momentum: 0.0,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'adadelta', momentum: 0.91,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'nesterov', momentum: 0.0,
    batch_size:BS, l2_decay:L2});
trainer_defs.push({learning_rate:LR, method: 'nesterov', momentum: 0.91,
    batch_size:BS, l2_decay:L2});

// names for all trainers above
legend = ['sgd', 'sgd+momentum', 'adagrad', 'adagrad+momentum','windowgrad'
    ,'windowgrad+momentum', 'adadelta', 'adadelta+momentum', 'nesterov', '
    nesterov+momentum'];
```

Listing 1.1 ConvNetJS code listing

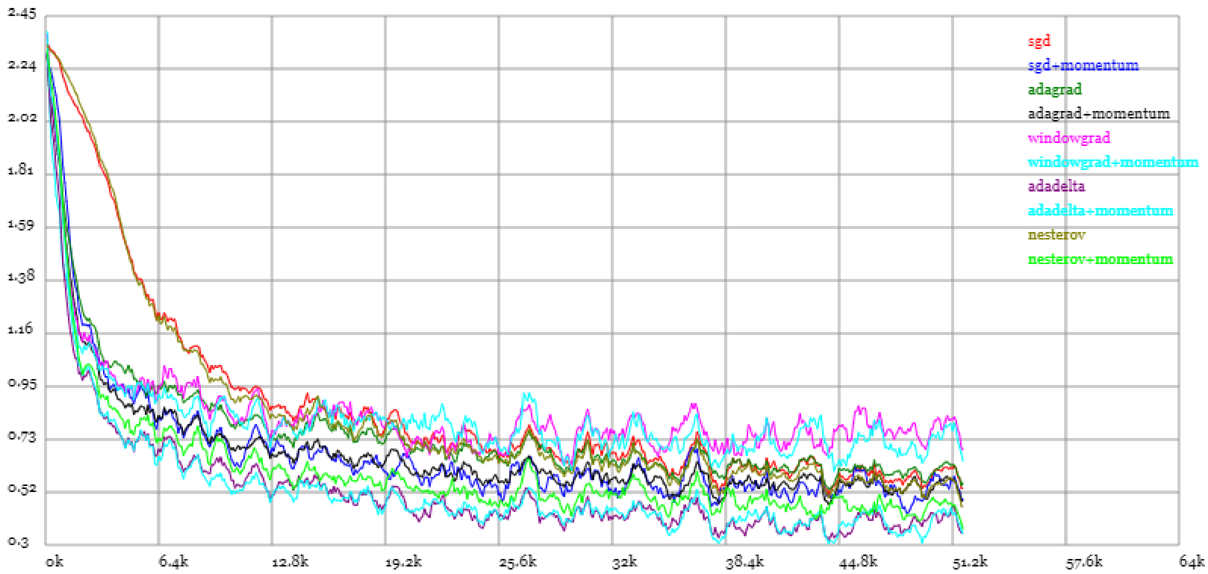

Figure 1.4 Loss vs. Number of examples seen

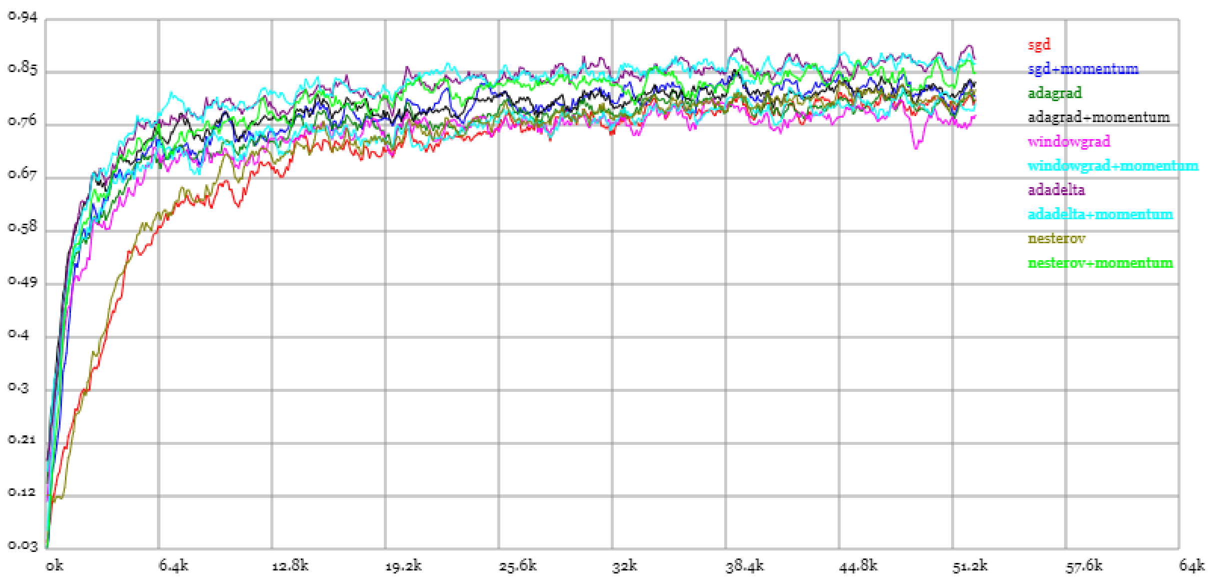

Figure 1.5 Testing Accuracy vs. Number of examples seen

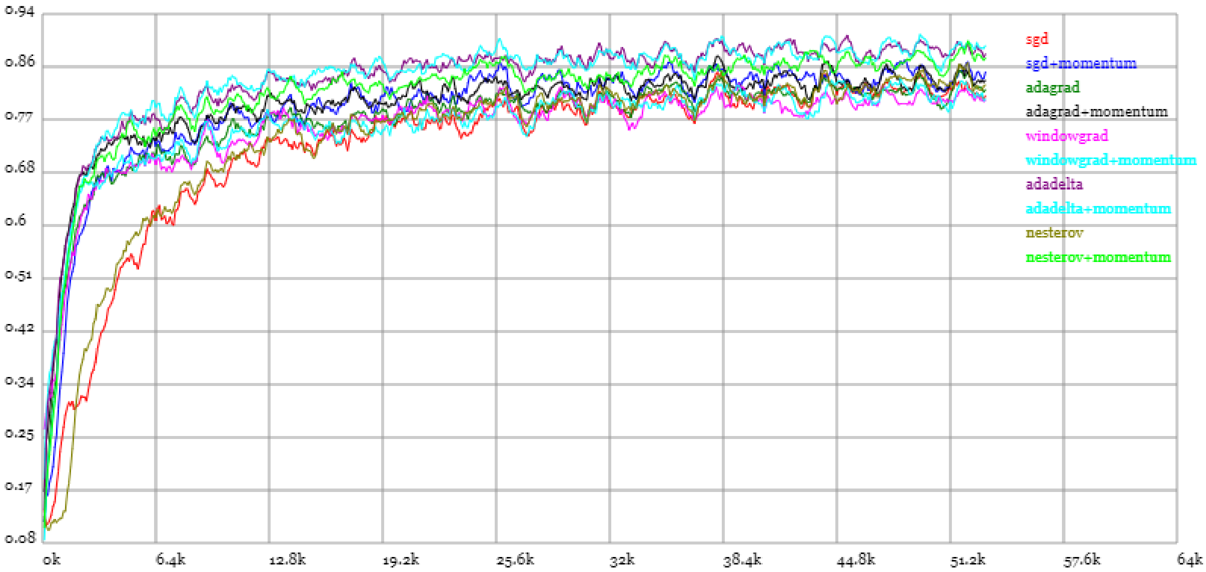

Figure 1.6 Training Accuracy vs. Number of examples seen

From the output graphs displayed in Figures 1.4, 1.5 and 1.6 it was noted that the best performing optimisation function was **Adadelta**. This was concluded as both the Adadelta and Adadelta with momentum functions had the best training accuracy, testing accuracy and least loss, when compared to the other functions. Additionally, it was also noted that given momentum, the optimisation functions converge much quicker than those without, when compared to their counterparts such as **SGD**.

1.4.3 Refining and Validating Answer with ChatGPT

Prompting ChatGPT with the following:
"How could the above paragraph be improved without drastically increasing the word limit while also managing to include other details that were omitted? Additionally, try to also improve it from an academic English and AI researcher perspective. Finally, compare these observations with the current literature, and discuss the validity of these claims."

Results Obtained:
"Analysis of the output graphs, as depicted in Figures 1.4, 1.5, and 1.6, revealed that the optimization function that consistently demonstrated superior performance was Adadelta. This assertion is supported by the fact that both Adadelta and Adadelta with momentum consistently exhibited the highest training accuracy, testing accuracy, and the least loss when compared to the alternative optimization functions.

Furthermore, it is crucial to highlight the impact of incorporating momentum into the optimization process. When momentum was introduced, the optimization functions exhibited significantly faster convergence rates compared to their non-momentum counterparts, notably surpassing traditional methods like Stochastic Gradient Descent (SGD). This observation underscores the efficacy of momentum in enhancing the optimization process, allowing neural networks to reach convergence more efficiently.

In terms of alignment with existing literature, the superiority of Adadelta and the benefits of momentum-enhanced optimization are well-documented. Numerous studies in the field of deep learning have corroborated the advantages of Adadelta in terms of robustness and adaptability to varying learning rates. Additionally, the role of momentum in accelerating convergence and mitigating optimization challenges has been widely acknowledged in prior research, affirming the validity of the findings in this study.

To summarize, the empirical findings presented in this analysis affirm the superiority of Adadelta as an optimization function, particularly when used in conjunction with momentum. These results align with established literature and contribute to the ongoing discourse on optimization strategies in the realm of artificial intelligence research and practice."

1.5 Conclusion

In summary, this study has provided a comprehensive understanding of how a neural network could be fine-tuned. Modifying parameters such as learning rates and activation functions might significantly impact the model's performance, as achieving a balance between model complexity and efficiency is crucial. Selecting the right tool is important when it comes to factors such as activation functions or the optimisation functions.

References

[1] Google Research. “Tensorflow playground.” (2016), [Online]. Available: https://playground.tensorflow.org/.

[2] A. Karpathy. “Convnetjs: Demonstrations.” (2022), [Online]. Available: https://cs.stanford.edu/people/karpathy/convnetjs/demo/trainers.html.

Plagiarism Form

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I / ~~We~~* , the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / ~~our~~* work, except where acknowledged and referenced.

I / ~~We~~* understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N.B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

StudentName	Signature
StudentName	Signature
StudentName	Signature
Matthias Bartolo	
StudentName	Signature

ARI3129	Advanced Computer Vision Individual Assignment
CourseCode	Title of work submitted

17 October 2023
Date