Artificial Neural Network – Homework 1
Font Classification Using Multiple Layer Perceptron

Ammar Abdurrauf - 170421930

Introduction

Datasets of 7 alphabets (A, B, C, D, E, J, K) in three different fonts were given to train a model using multiple layer perceptron method. Before training the model, the dataset was given some noises by randomly changing some pixels to a random integer among -1, 0, and 1 representing black, grey, and white respectively. Various optimization algorithms were used to train the model. The learning rate and several activation functions were variated to observe and analyze the behavior of the trained model.

Methods and Dataset

Dataset

The datasets contain matrices of 7 alphabets with three different fonts resulting a total of 21 images. The matrices define a 9 by 7 pixels consisting of -1 and 1, where -1 represents black and 1 represents white. The datasets were imported to the Google Colab as matrices. Figure 1 shows the dataset used in this work.

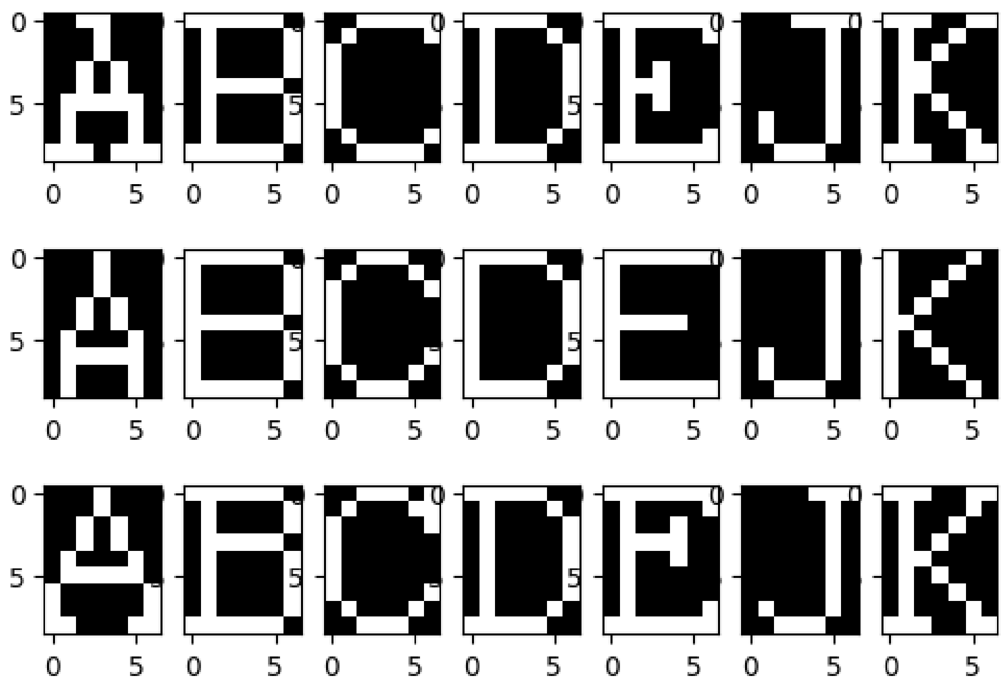


Figure 1. Datasets consist of different alphabets and fonts

Adding Noise to The Dataset

Two different approaches to add noises to the dataset were done. Firstly, 7 randomly selected pixels in a font were changed to a value among -1, 0, and 1 representing black, grey, and white respectively. The second approach also selects 7 random pixels in a font and changes those pixels by reversing the value inside that pixel, this is done by multiplying the value within that pixel with -1. Finally, to obtain more datasets, 100 iterations for generating new fonts with noises were conducted for each font. Two new datasets each containing 2121 different alphabets were obtained (4242 in total). Datasets that added noise using the first and second approaches will be labeled as dataset #1 and dataset #2 respectively for convenience. The samples of font with noises can be seen in Figure 2 and 3.

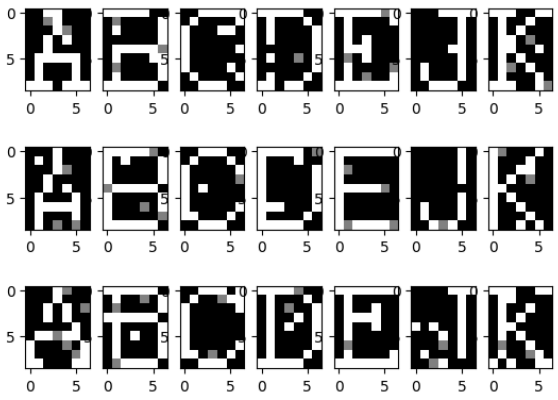


Figure 2. Font with noise #1

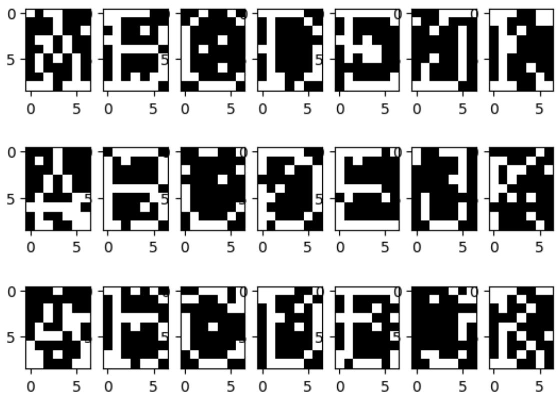


Figure 3. Font with noise #2

Variating The Optimizer Algorithm and Parameters

The optimizer algorithms used to train the model were Adam (Adaptive Moment Estimation), RMSprop (Root Mean Square propagation), and SGD (Stochastic Gradient Descent) optimizers. The dense of the layers was kept to 9 by 7 input layers (63 in total), 10 layers with an activation function (will be variated in several trainings), and 7 output layers with softmax activation function. The variation of the optimizer, learning rate, and activation function used for training the model is shown in Table 1.

Training No	Optimizer	Learning Rate	Dataset #1	Dataset #2
			Activation Function on 2nd layer	
1	Adam	0.001	Relu	Tanh
2		0.01	Sigmoid	Relu
3		0.2	Tanh	Sigmoid
4	RMSprop	0.001	Tanh	Sigmoid
5		0.01	Relu	Tanh
6		0.2	Sigmoid	Relu
7	SGD	0.001	Sigmoid	Relu
8		0.01	Tanh	Sigmoid
9		0.2	Relu	Tanh

Table 1. Variations of optimizer and parameters for the training

Results and Discussion

The model training was conducted with 11 epoch and validation split of 0.2 for every datasets. The training resulted 18 models and history graphs during the training, such as accuracy, validation accuracy, loss, validation loss, and confusion matrix. The model's precision, recall, accuracy, and F1 scores are listed in Table 2.

Training No	Dataset #1 (%)				Dataset #2 (%)			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
1	99.84	99.84	99.84	99.84	99.68	99.68	99.68	99.69
2	100	100	100	100	99.53	99.52	99.52	99.53
3	95.19	94.30	94.29	94.34	95.12	93.97	93.67	94.03
4	100	100	100	100	99.68	99.68	99.68	99.69
5	100	100	100	100	99.68	99.68	99.68	99.69
6	99.84	99.84	99.84	99.84	46.25	56.83	48.37	57.23
7	99.53	99.53	99.53	99.53	98.59	98.59	98.58	98.58
8	100	100	100	100	99.53	99.52	99.52	99.53
9	78.37	85.40	80.68	85.53	99.22	99.21	99.21	99.21

Table 2. Model's precision, recall, F1, and accuracy scores

Dataset #1

Model training using Dataset #1 was conducted with several optimizer algorithms and variation of learning rate and activation function. As shown in Table 2, the model that used the RMSprop optimizer algorithm during training achieved the best results among the others. It reached 100% accuracy and F1 score when using a combination of the tanh and relu activation functions, along with learning rates of 0.001 and 0.01. Model training that uses learning rate of 0.01 shows a perfect output in precision, recall, F1 score, and accuracy. On the other hand, models trained with a learning rate of 0.2 seem to result in lower F1 scores compared to other learning rates. The model trained using the SGD optimizer, relu as the activation function, and a learning rate of 0.2 shows the lowest value among the results.

Dataset #2

In the Dataset #2, the fonts were added noises by reversing selected random pixel. This results more hardly recognizable fonts even by human. Therefore, as shown in Table 2. the results of training did not generate any perfect model. The highest accuracy and F1 score were achieved by the model that was trained using the Adam and RMSprop optimizer algorithms, with learning rates of 0.001 and 0.01, and using the tanh and sigmoid activation functions. On the contrary, the lowest precision, recall, F1 score, and accuracy values were resulted from the training model that used RMSprop along with relu activation function and 0.2 learning rate. However, the majority of the results demonstrate good F1 scores and accuracy, successfully recognizing most of the test dataset.

The complete graphs depicting the training history of all models, including accuracy and validation accuracy, loss and validation loss, as well as the confusion matrix, are presented in figures below.

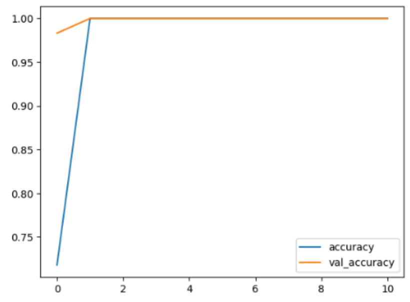
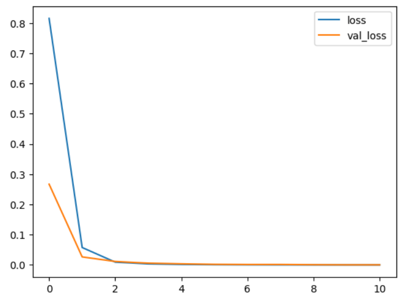
Discussion

As presented in Table 2., it can be seen that the model trained using learning rate of 0.01 reached the highest accuracy among other learning rates for both datasets, regardless of the types of optimizer algorithm and activation function used in that training. It can be concluded that a learning rate of 0.01 is optimal for training a model for font classification, even in the presence of noises. In Dataset #2, the result of training model using SGD as the optimizer function and tanh as the activation function along with 0.2 learning rate resulted in nearly perfect accuracy and F1 score. This contradicts the result of model trained using the RMSprop optimizer and relu activation function which produced worse accuracy and F1 score even when it used the same learning rate.

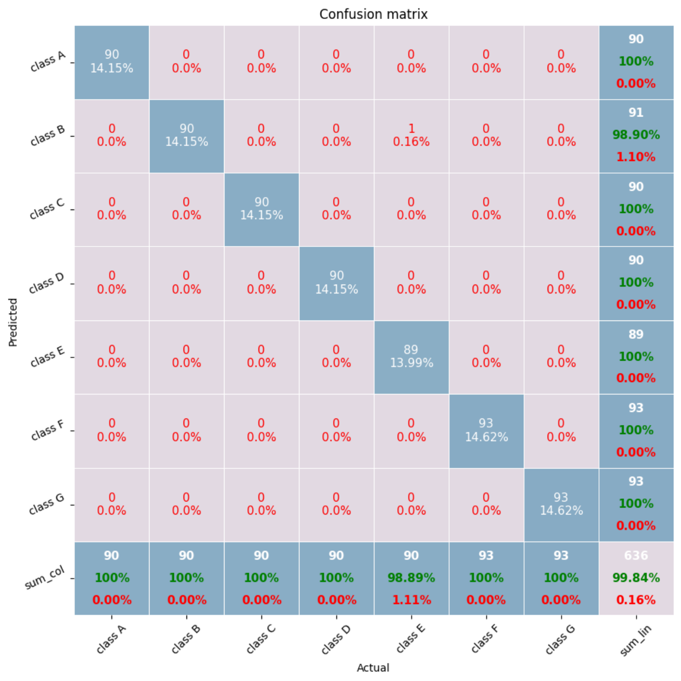
Adam Optimizer

Training Results Using Adam Optimizer - Dataset #1

Learning rate: 0.001
Activation Function on 2nd layer: Relu

Accuracy & Val Accuracy	Loss & Val Loss
	

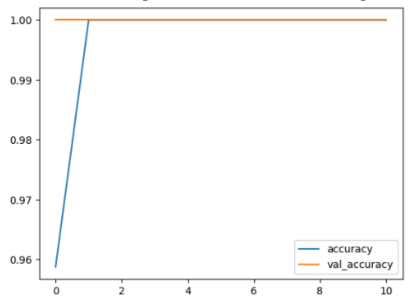
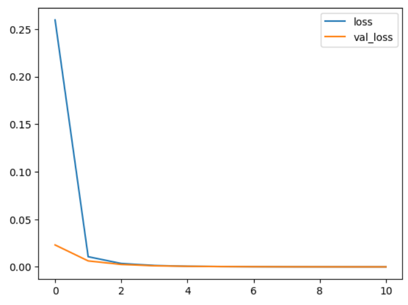
Confusion Matrix



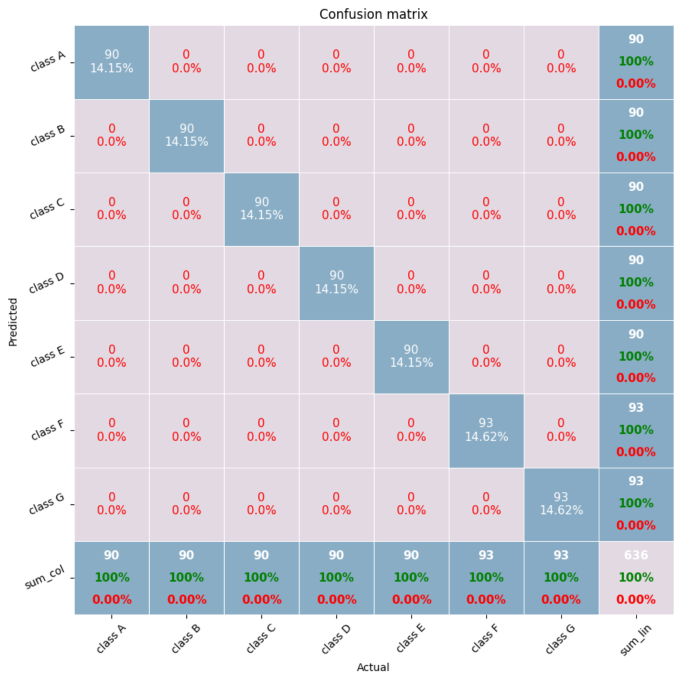
	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	91 98.90% 1.10%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	89 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 98.89% 1.11%	93 100% 0.00%	93 100% 0.00%	636 99.84% 0.16%

Accuracy:
99.84%

Learning rate: 0.01
Activation Function on 2nd layer: Sigmoid

Accuracy & Val Accuracy	Loss & Val Loss
	

Confusion Matrix


Confusion matrix

Predicted	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	90 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 100% 0.00%

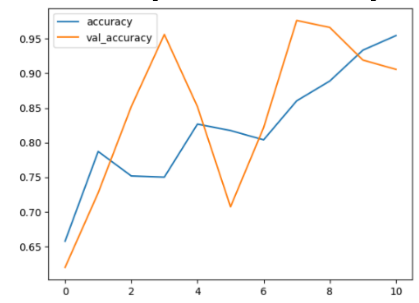
Actual

Accuracy: 100%

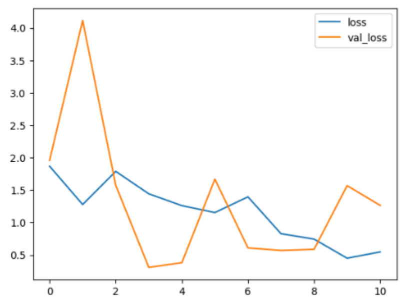
Learning rate: 0.2

Activation Function on 2nd layer: Tanh

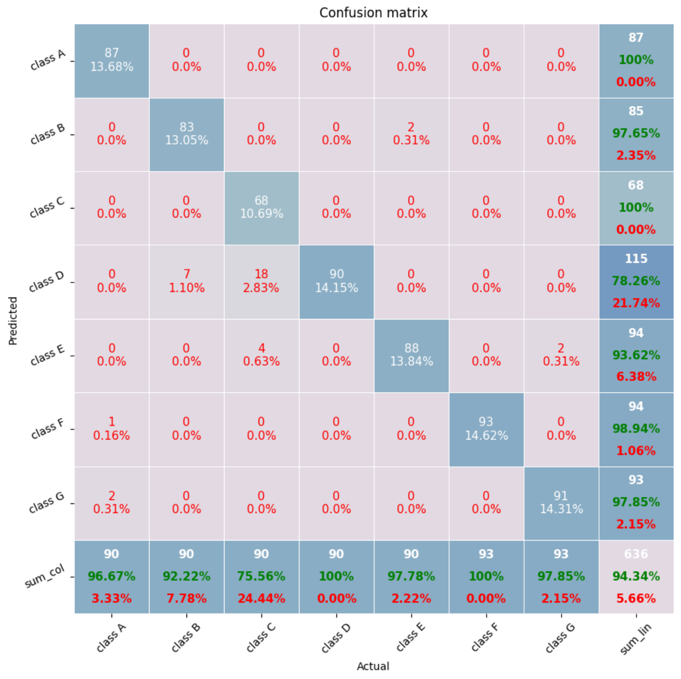
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix


Confusion matrix

Predicted	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	87 13.68%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87 100% 0.00%
class B	0 0.0%	83 13.05%	0 0.0%	0 0.0%	2 0.31%	0 0.0%	0 0.0%	85 97.65% 2.35%
class C	0 0.0%	0 0.0%	68 10.69%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	68 100% 0.00%
class D	0 0.0%	7 1.10%	18 2.83%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	115 78.26% 21.74%
class E	0 0.0%	0 0.0%	4 0.63%	0 0.0%	88 13.84%	0 0.0%	2 0.31%	94 93.62% 6.38%
class F	1 0.16%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	94 98.94% 1.06%
class G	2 0.31%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	91 14.31%	93 97.85% 2.15%
sum_col	90 96.67% 3.33%	90 92.22% 7.78%	90 75.56% 24.44%	90 100% 0.00%	90 97.78% 2.22%	93 100% 0.00%	93 97.85% 2.15%	636 94.34% 5.66%

Actual

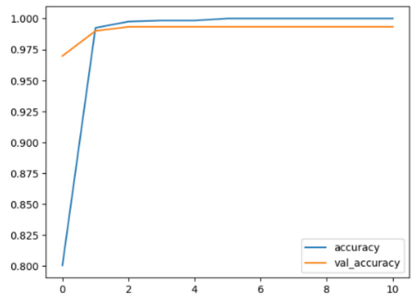
Accuracy: 94.34%

Figure 4.. Training results using Adam optimizer - Dataset #1

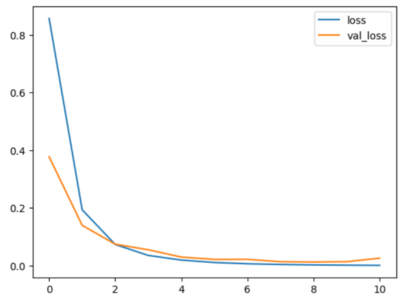
Training Results Using Adam Optimizer - Dataset #2

Learning rate: 0.001
Activation Function on 2nd layer: Tanh

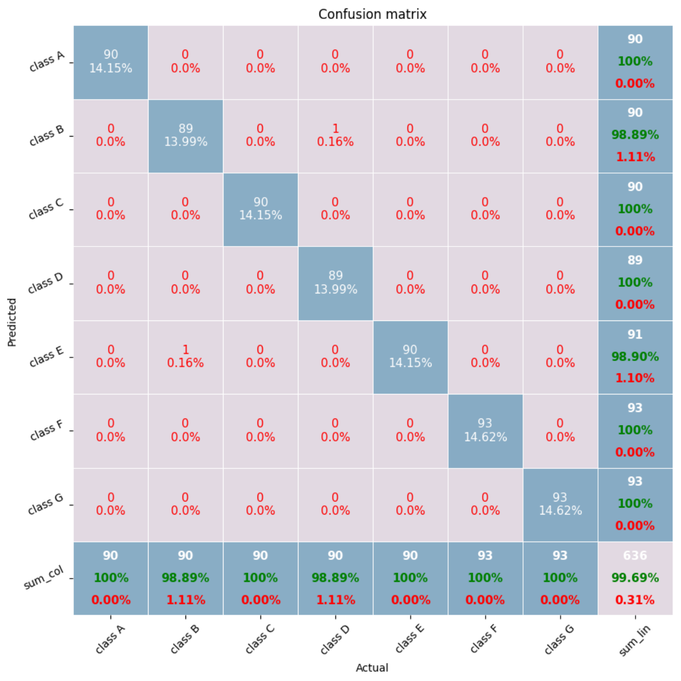
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



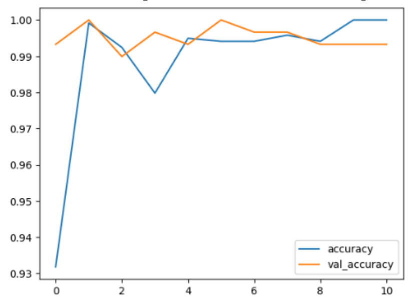
Confusion matrix
Predicted
Actual

	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	89 13.99%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	0 0.0%	90 98.89% 1.11%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	89 100% 0.00%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	91 98.90% 1.10%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 99.69% 0.31%

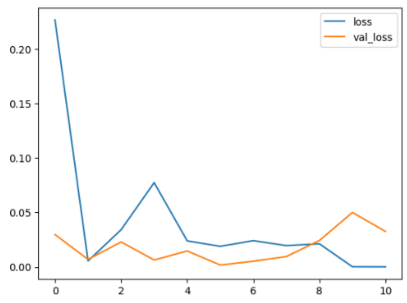
Accuracy:
99.69%

Learning rate: 0.01
Activation Function on 2nd layer: Relu

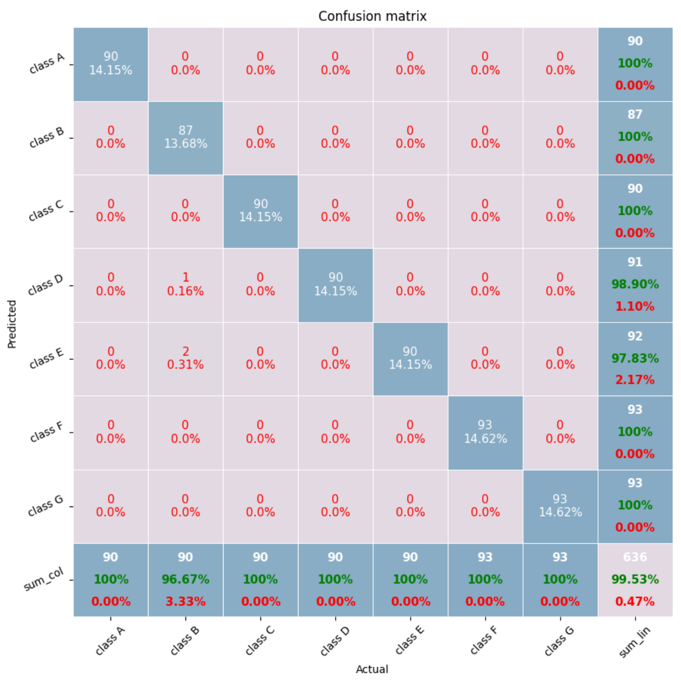
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix

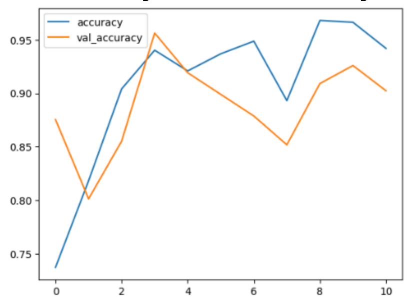

Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	87 13.68%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	1 0.16%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	91 98.90% 1.10%
class E	0 0.0%	2 0.31%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	92 97.83% 2.17%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 96.67% 3.33%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 99.53% 0.47%

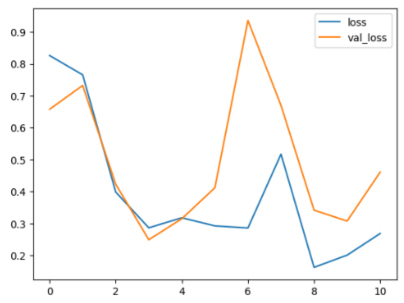
Accuracy:
99.53%

Learning rate: 0.2
Activation Function on 2nd layer: Sigmoid

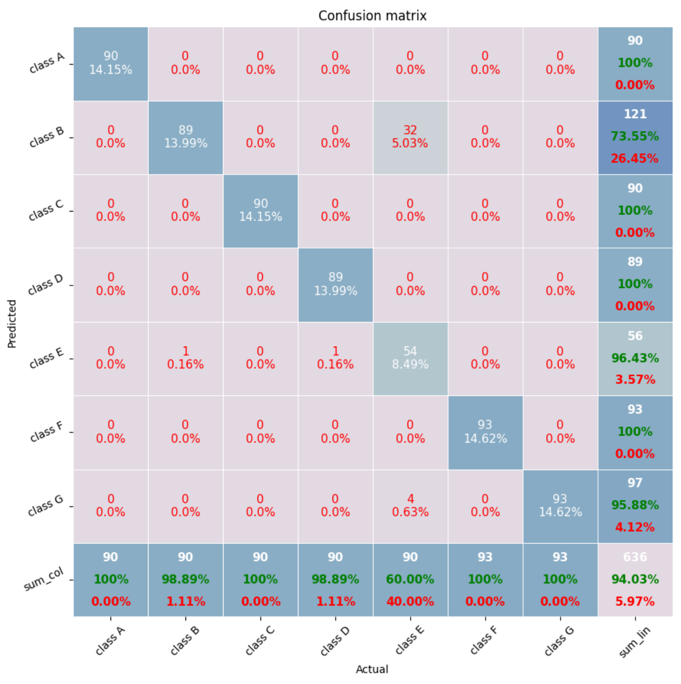
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix


Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	89 13.99%	0 0.0%	0 0.0%	32 5.03%	0 0.0%	0 0.0%	121 73.55% 26.45%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	89 100% 0.00%
class E	0 0.0%	1 0.16%	0 0.0%	1 0.16%	54 8.49%	0 0.0%	0 0.0%	56 96.43% 3.57%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.63%	0 0.0%	93 14.62%	97 95.88% 4.12%
sum_col	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	90 98.89% 1.11%	90 60.00% 40.00%	93 100% 0.00%	93 100% 0.00%	636 94.03% 5.97%

Accuracy:
94.03%

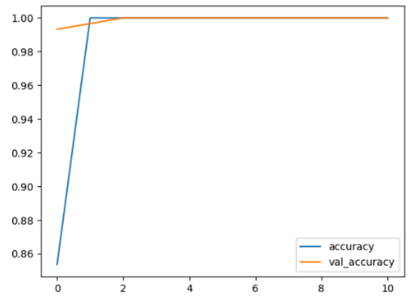
Figure 5. Training results using Adam optimizer - Dataset #2

RMSprop Optimizer

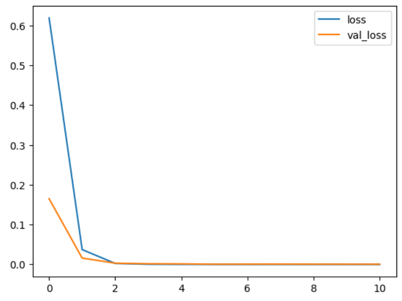
Training Results Using RMSprop Optimizer - Dataset #1

Learning rate: 0.001
Activation Function on 2nd layer: Tanh

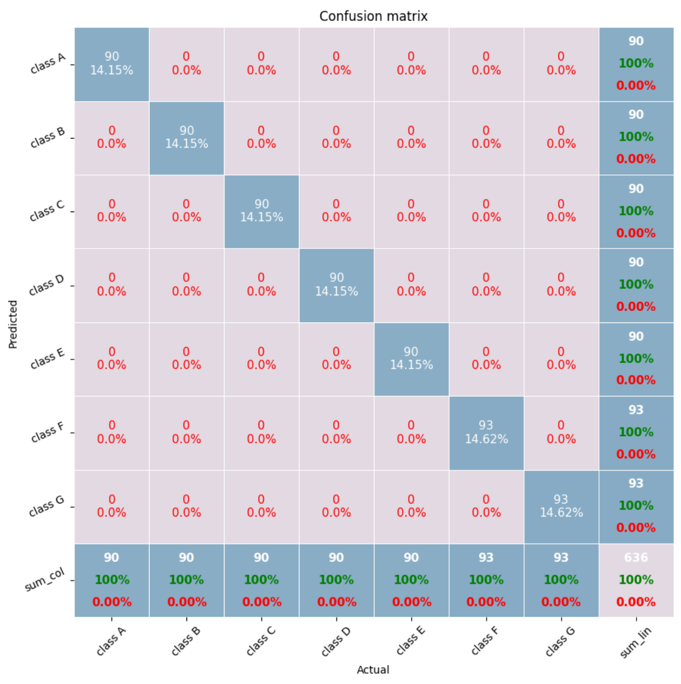
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



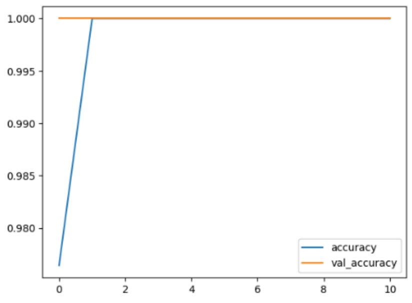
Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	90 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 100% 0.00%

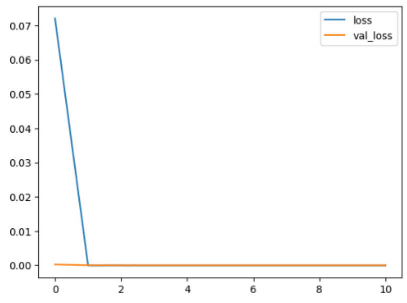
Accuracy:
100%

Learning rate: 0.01
Activation Function on 2nd layer: Relu

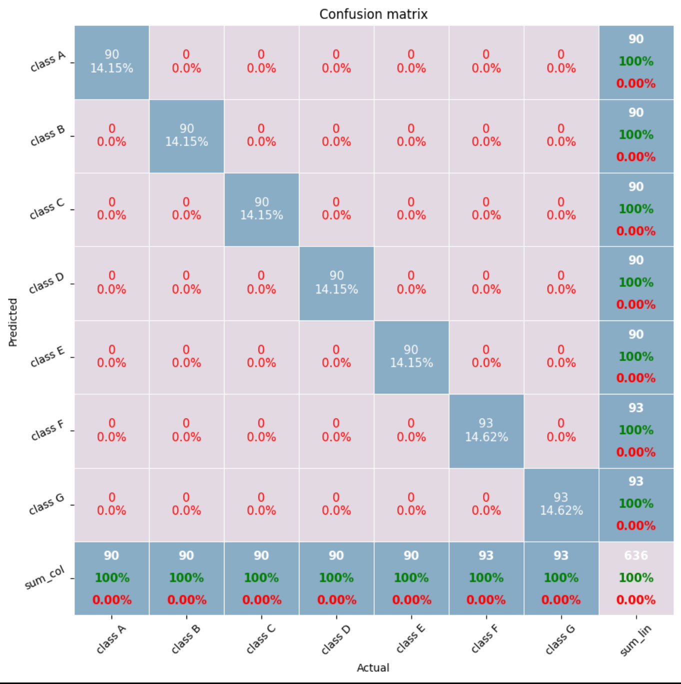
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix


Confusion matrix

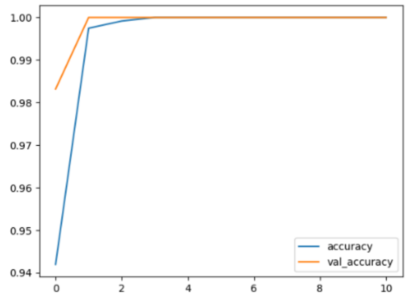
Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	90 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 100% 0.00%

Test Accuracy: 100 %

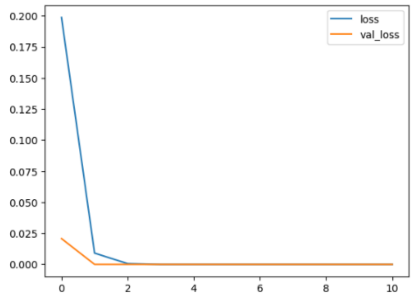
Learning rate: 0.2

Activation Function on 2nd layer: Sigmoid

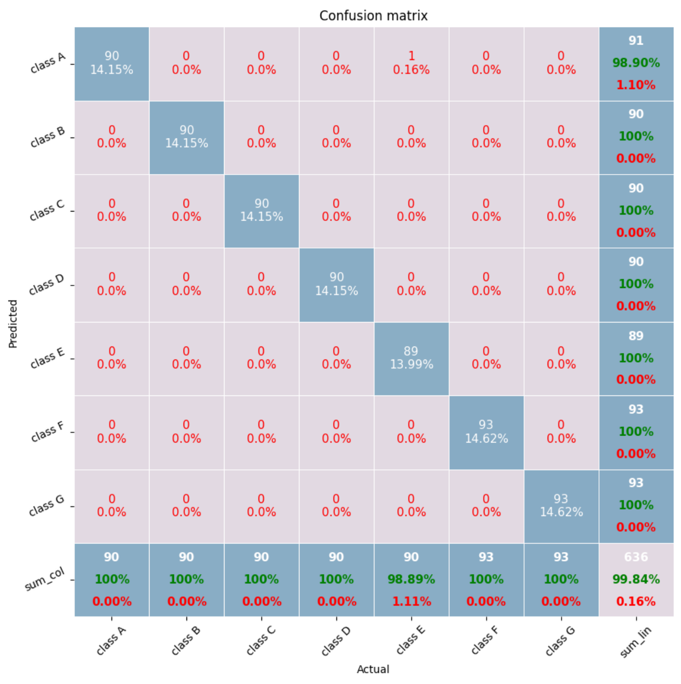
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix


Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	91 98.90% 1.10%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	89 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 98.89% 1.11%	93 100% 0.00%	93 100% 0.00%	636 99.84% 0.16%

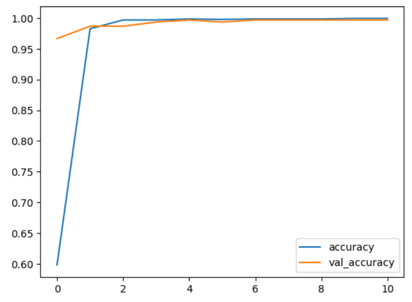
Accuracy: 99.84%

Figure 6. Training results using RMSprop optimizer - Dataset #1

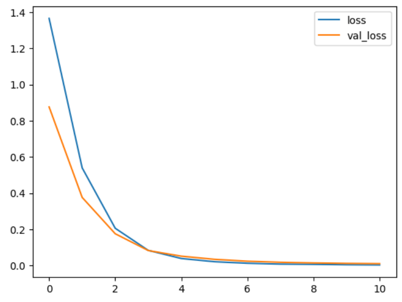
Training Results Using RMSprop Optimizer - Dataset #2

Learning rate: 0.001
Activation Function on 2nd layer: Sigmoid

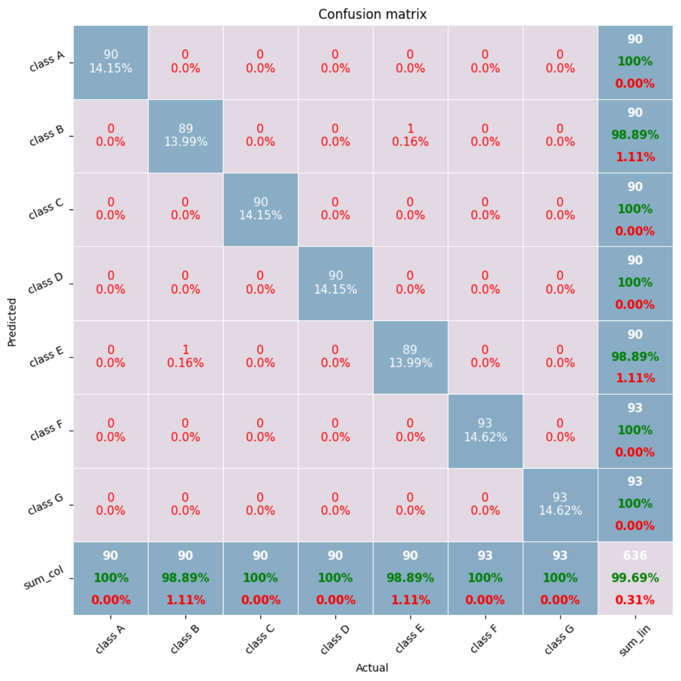
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix

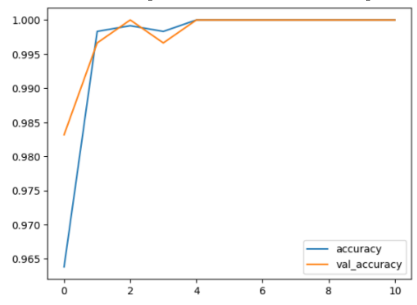

Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	89 13.99%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	90 98.89% 1.11%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	90 98.89% 1.11%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	90 100% 0.00%	90 98.89% 1.11%	93 100% 0.00%	93 100% 0.00%	636 99.69% 0.31%

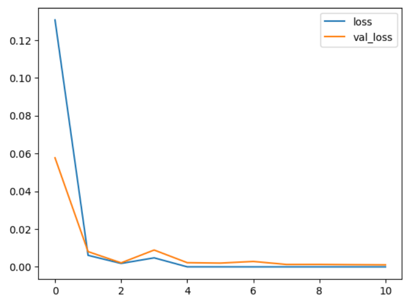
Test Accuracy:
99.69%

Learning rate: 0.01
Activation Function on 2nd layer: Tanh

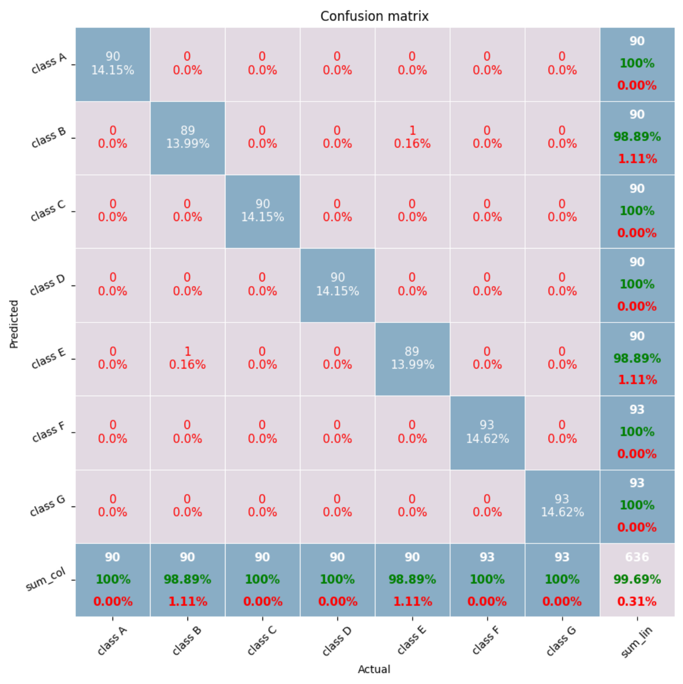
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix

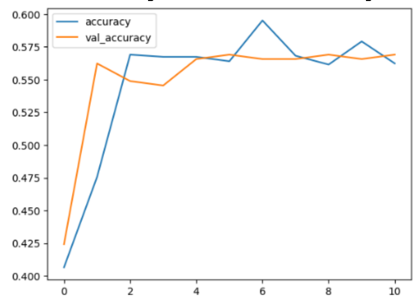


	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	89 13.99%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	90 98.89% 1.11%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	90 98.89% 1.11%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	90 100% 0.00%	90 98.89% 1.11%	93 100% 0.00%	93 100% 0.00%	636 99.69% 0.31%

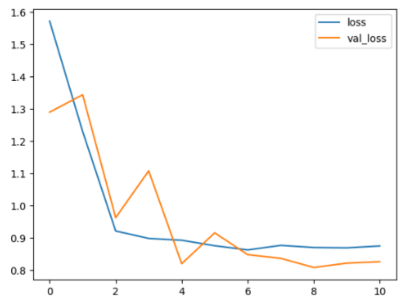
Test Accuracy:
99.69%

Learning rate: 0.2
Activation Function on 2nd layer: Relu

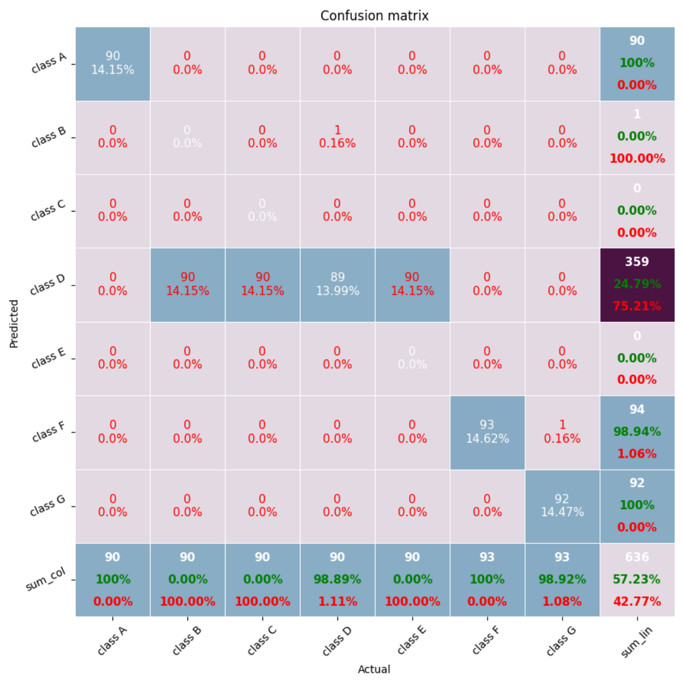
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	0 0.0%	1 0.00% 100.00%
class C	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.00% 0.00%
class D	0 0.0%	90 14.15%	90 14.15%	89 13.99%	90 14.15%	0 0.0%	0 0.0%	359 24.79% 75.21%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.00% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	1 0.16%	94 98.94% 1.06%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92 14.47%	92 100% 0.00%
sum_col	90 100% 0.00%	90 0.00% 100.00%	90 0.00% 100.00%	90 98.89% 1.11%	90 0.00% 100.00%	93 100% 0.00%	93 98.92% 1.08%	636 57.23% 42.77%

Test Accuracy:
57.23%

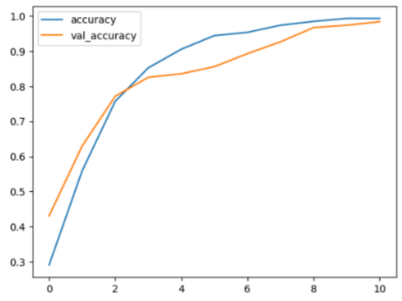
Figure 7. Training results using RMSprop optimizer - Dataset #2

SGD Optimizer

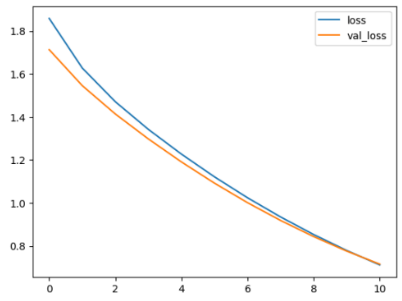
Training Results Using SGD Optimizer - Dataset #1

Learning rate: 0.001
Activation Function on 2nd layer: Sigmoid

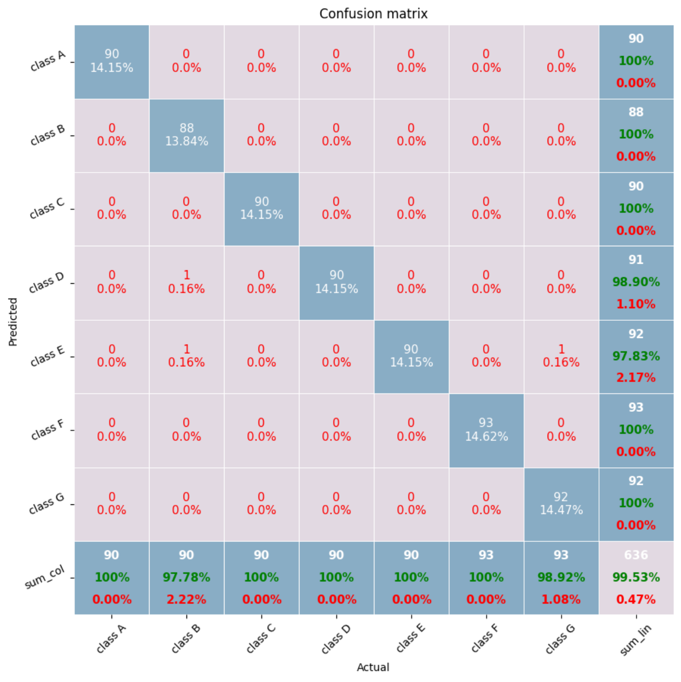
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



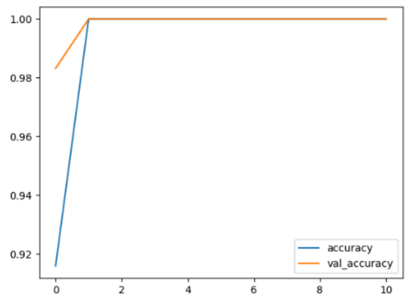
Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	88 13.84%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	88 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	1 0.16%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	91 98.90% 1.10%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	1 0.16%	92 97.83% 2.17%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92 14.47%	92 100% 0.00%
sum_col	90 100% 0.00%	90 97.78% 2.22%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 98.92% 1.08%	636 99.53% 0.47%

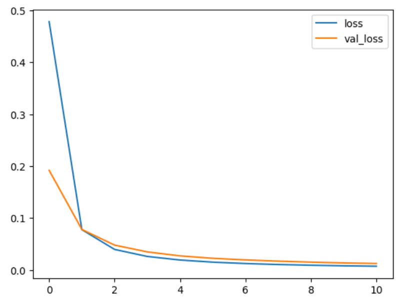
Accuracy:
99.53%

Learning rate: 0.01
Activation Function on 2nd layer: Tanh

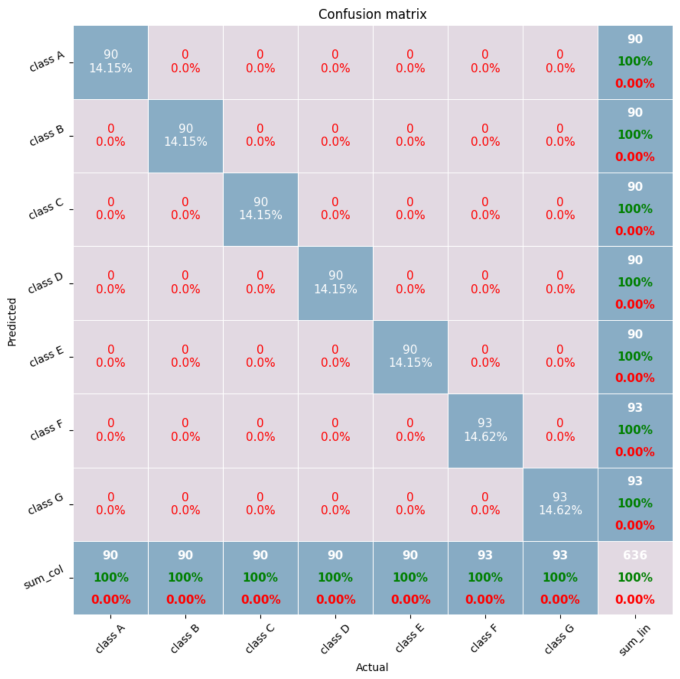
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



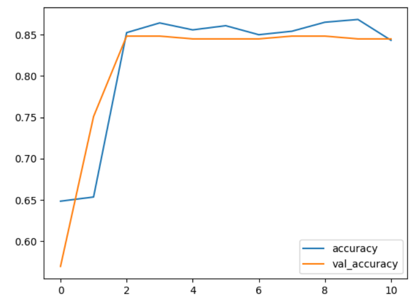
Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	90 100% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 100% 0.00%

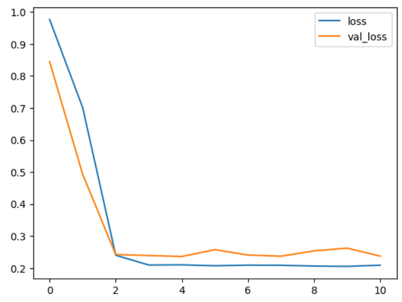
Accuracy:
100%

Learning rate: 0.2
Activation Function on 2nd layer: Relu

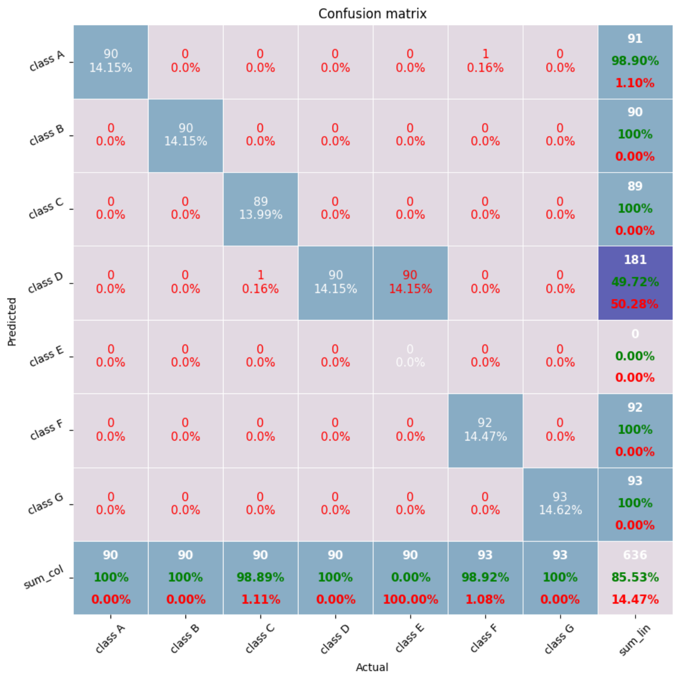
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	91 98.90% 1.10%
class B	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class C	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	89 100% 0.00%
class D	0 0.0%	0 0.0%	1 0.16%	90 14.15%	90 14.15%	0 0.0%	0 0.0%	181 49.72% 50.28%
class E	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.00% 0.00%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92 14.47%	0 0.0%	92 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	90 0.00% 100.00%	93 98.92% 1.08%	93 100% 0.00%	636 85.53% 14.47%

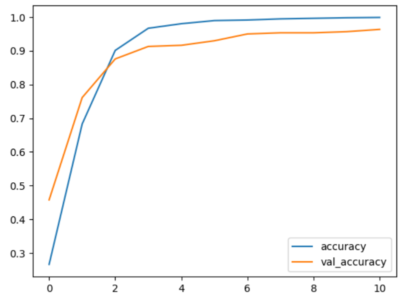
Accuracy:
85.53%

Figure 8. Training results using SGD optimizer - Dataset #1

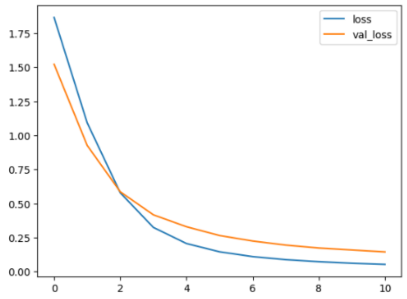
Training Results Using SGD Optimizer - Dataset #2

Learning rate: 0.001
Activation Function on 2nd layer: Relu

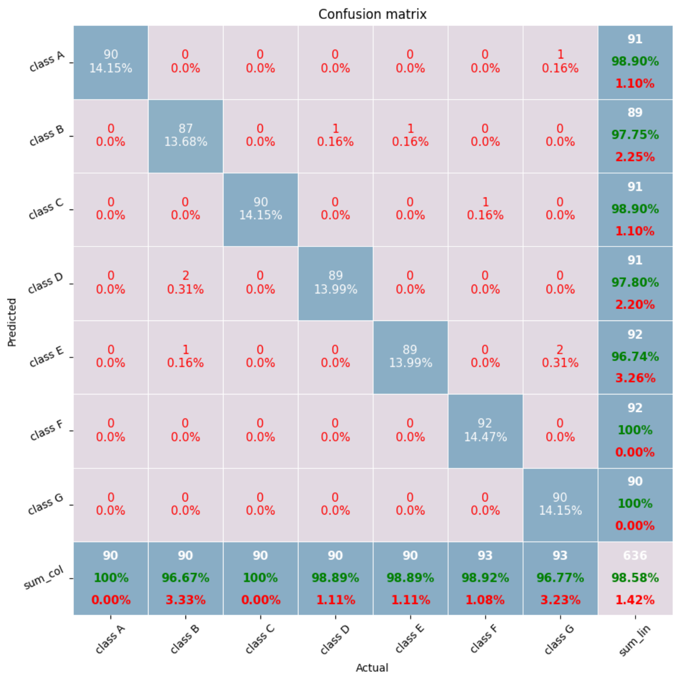
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix



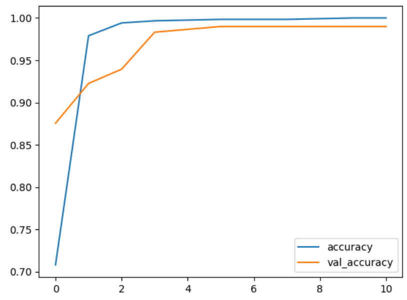
Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.16%	91 98.90% 1.10%
class B	0 0.0%	87 13.68%	0 0.0%	1 0.16%	1 0.16%	0 0.0%	0 0.0%	89 97.75% 2.25%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	91 98.90% 1.10%
class D	0 0.0%	2 0.31%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	91 97.80% 2.20%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	2 0.31%	92 96.74% 3.26%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92 14.47%	0 0.0%	92 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 14.15%	90 100% 0.00%
sum_col	90 100% 0.00%	90 96.67% 3.33%	90 100% 0.00%	90 98.89% 1.11%	90 98.89% 1.11%	93 98.92% 1.08%	93 96.77% 3.23%	636 98.58% 1.42%

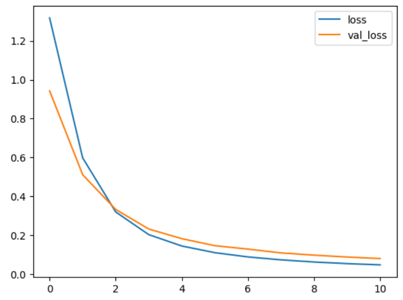
Test Accuracy:
98.58%

Learning rate: 0.01
Activation Function on 2nd layer: Sigmoid

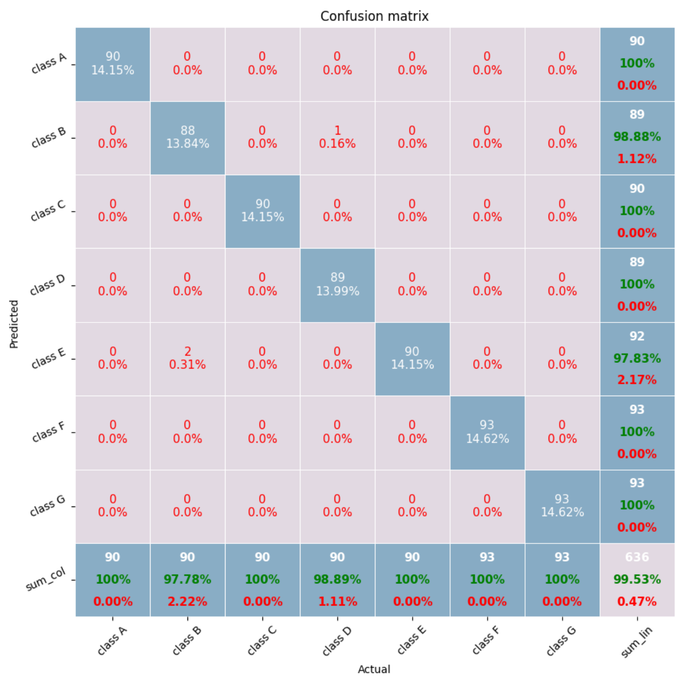
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix

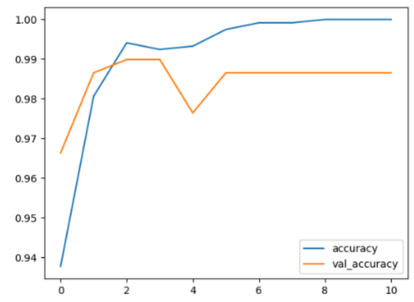

Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class B	0 0.0%	88 13.84%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	0 0.0%	89 98.88% 1.12%
class C	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	89 100% 0.00%
class E	0 0.0%	2 0.31%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	92 97.83% 2.17%
class F	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	93 100% 0.00%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	93 14.62%	93 100% 0.00%
sum_col	90 100% 0.00%	90 97.78% 2.22%	90 100% 0.00%	90 98.89% 1.11%	90 100% 0.00%	93 100% 0.00%	93 100% 0.00%	636 99.53% 0.47%

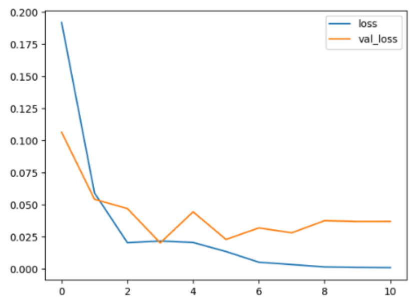
Test Accuracy:
99.53%

Learning rate: 0.2
Activation Function on 2nd layer: Tanh

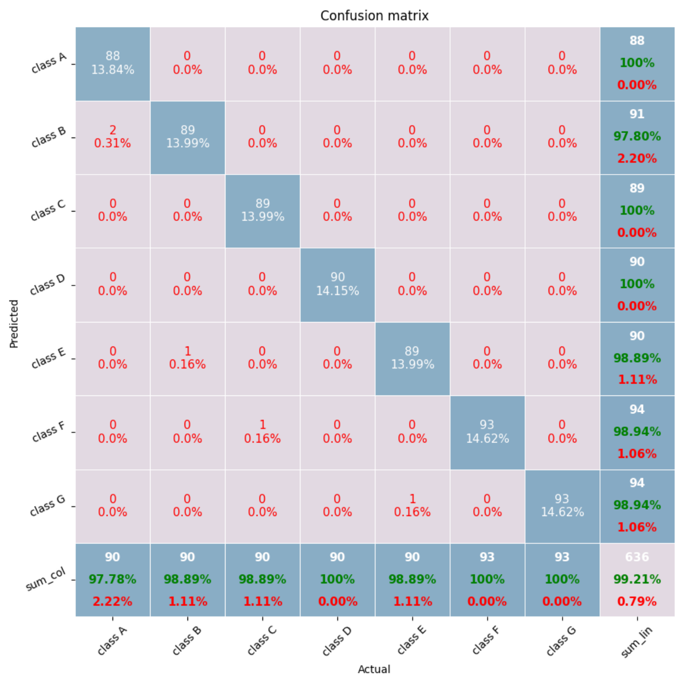
Accuracy & Val Accuracy



Loss & Val Loss



Confusion Matrix


Confusion matrix

Predicted \ Actual	class A	class B	class C	class D	class E	class F	class G	sum_lin
class A	88 13.84%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	88 100% 0.00%
class B	2 0.31%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	91 97.80% 2.20%
class C	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	89 100% 0.00%
class D	0 0.0%	0 0.0%	0 0.0%	90 14.15%	0 0.0%	0 0.0%	0 0.0%	90 100% 0.00%
class E	0 0.0%	1 0.16%	0 0.0%	0 0.0%	89 13.99%	0 0.0%	0 0.0%	90 98.89% 1.11%
class F	0 0.0%	0 0.0%	1 0.16%	0 0.0%	0 0.0%	93 14.62%	0 0.0%	94 98.94% 1.06%
class G	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.16%	0 0.0%	93 14.62%	94 98.94% 1.06%
sum_col	90 97.78% 2.22%	90 98.89% 1.11%	90 98.89% 1.11%	90 100% 0.00%	90 98.89% 1.11%	93 100% 0.00%	93 100% 0.00%	636 99.21% 0.79%

Test Accuracy:
99.21%

Figure 9. Training results using SGD optimizer - Dataset #2

Project Link

https://colab.research.google.com/drive/1tqy1Zfy956cdJHTCIrYkpc2y3rOCarAI