Assignment 1 Report

Tanat Tangun

August 2022

1 Flood Dataset

Problem

We want to predict water level at 7 hours ahead given station 1 and station 2 data at present and 3 hours before.

	A	B	C	D	E	F	G	H	I
1	s1_t3	s1_t2	s1_t1	s1_t0	s2_t3	s2_t2	s2_t1	s2_t0	t7
2	95	95	95	95	148	149	150	150	153
3	95	95	95	95	149	150	150	150	153
4	95	95	95	95	150	150	150	150	153
5	95	95	95	95	150	150	150	150	153
6	95	95	95	95	150	150	150	152	153
7	95	95	95	95	150	150	152	152	153
8	95	95	95	96	150	152	152	153	153
9	95	95	96	97	152	152	153	153	153
10	95	96	97	98	152	153	153	153	153
11	96	97	98	100	153	153	153	153	154
12	97	98	100	100	153	153	153	153	155
13	98	100	100	100	153	153	153	153	156
14	100	100	100	101	153	153	153	153	156

Figure 1: Examples of given data where s1_t3 is a water level from 3 hours before at station 1, t7 is water level at 7 hours ahead, and so on.

Parameters Setting

- All nodes use *sigmoid* as an activation function except output node that use *linear* function.
- Weights is a random number between $[-1, 1]$
- Each layer's bias is 1
- Use MSE (Mean Squared Error) as a loss function.

Training

Use 10% cross-validation, and preprocess our data by using training set *mean* and *std* to standardize (For each data point x we calculate new $x' = \frac{x-mean}{std}$) both training set and validation set before training with SGD (Stochastic Gradient Descent) algorithm. Then, we train each cross-validation set for 1000 epochs.

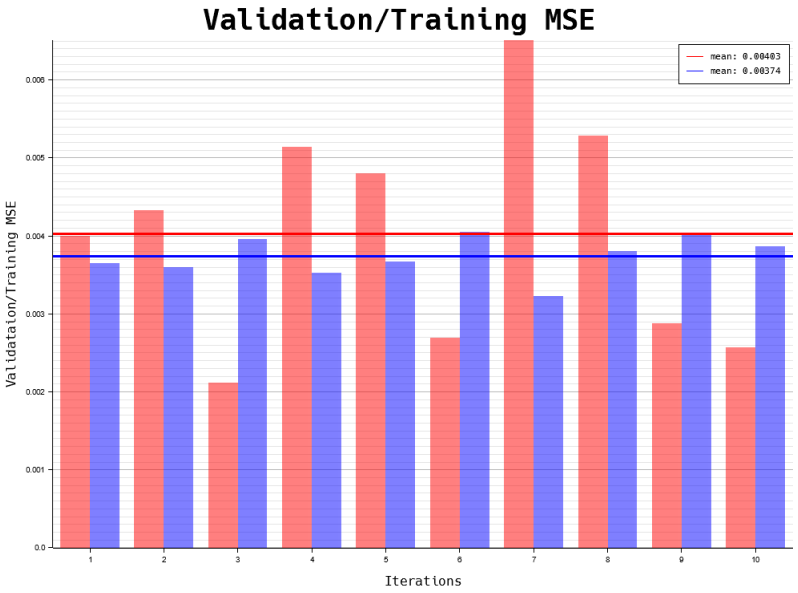
We will create one base network that should perform good enough and create a variation base on that network, that is train with no *momentum*, train with smaller *learning rate*, and add more layers or hidden nodes to see that if we introduce those variations, will the network perform better, converge faster or no improvement at all?

Training Result

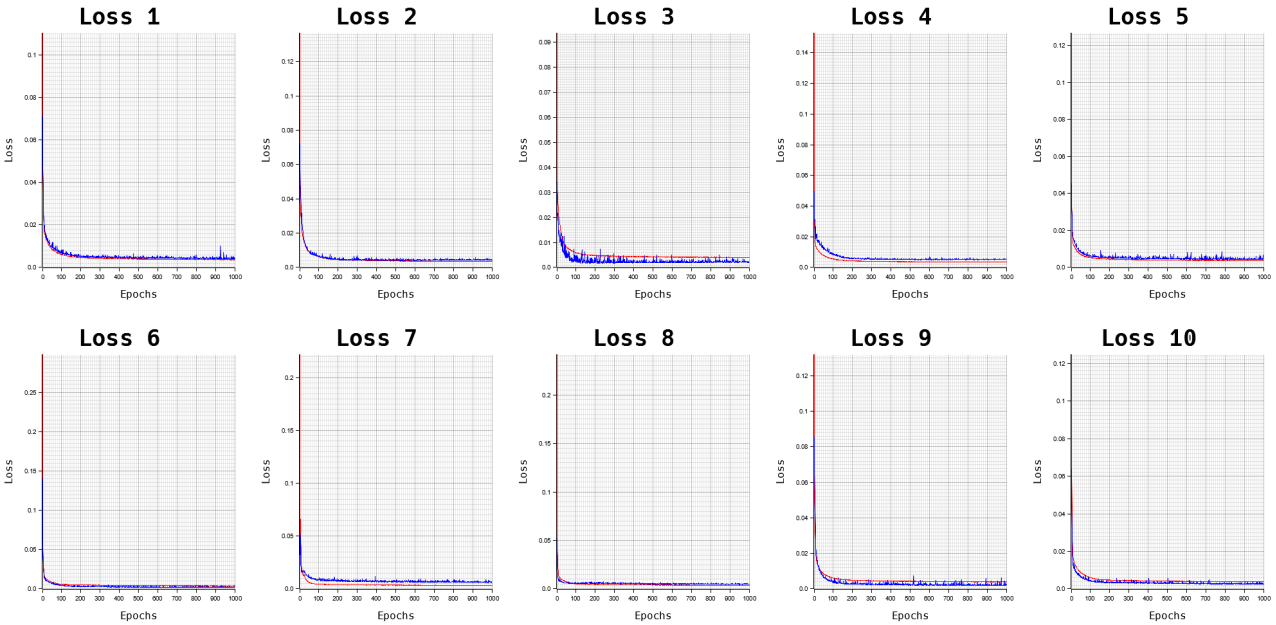
Flood-8-4-1

Our base network that contains only 8 input nodes, 1 hidden layer with 4 nodes, and 1 output node train with $lr = 0.01$ and $momentum = 0.01$.

Below is the graphs we get from training this network. The training process take about 60 seconds.



(a) Each iteration training (blue) and validation (red) MSE at last epoch



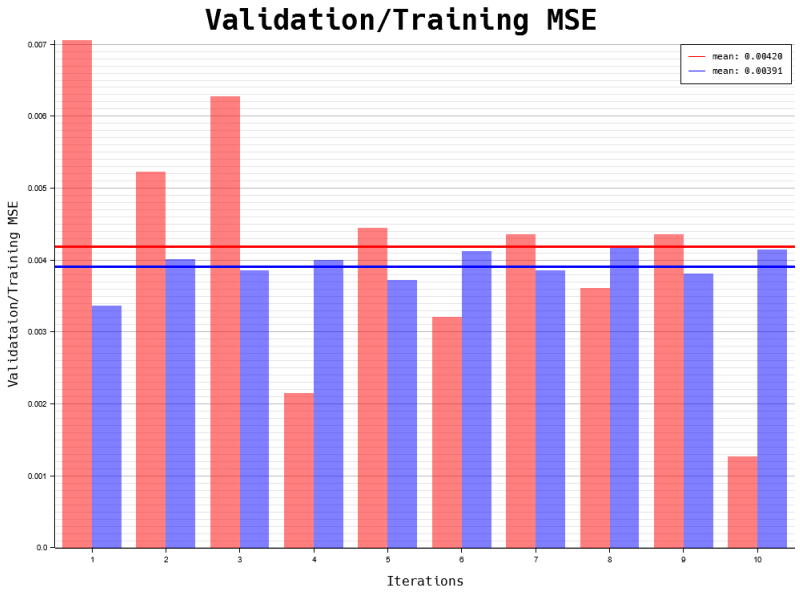
(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

Figure 2: Training result of Flood-8-4-1.

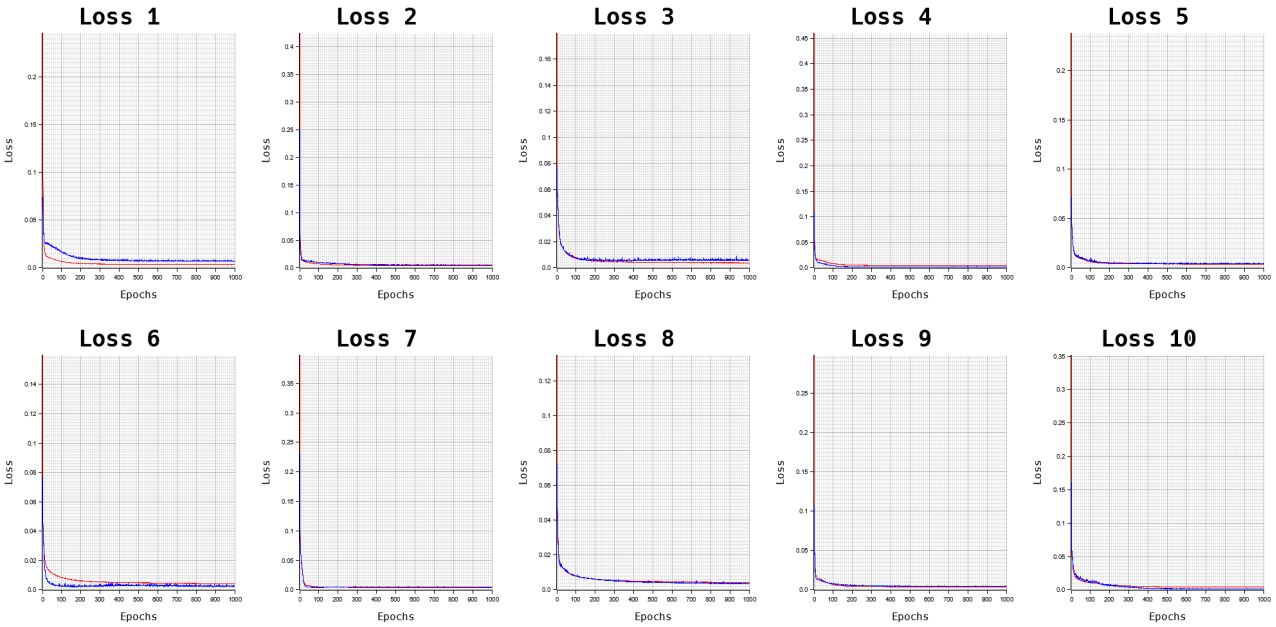
Flood-8-4-1 with no momentum

Same base network but train with $lr = 0.01$ and $momentum = 0$.

Below is the graphs we get from training this network. The training process take about 60 seconds.



(a) Each iteration training (blue) and validation (red) MSE at last epoch



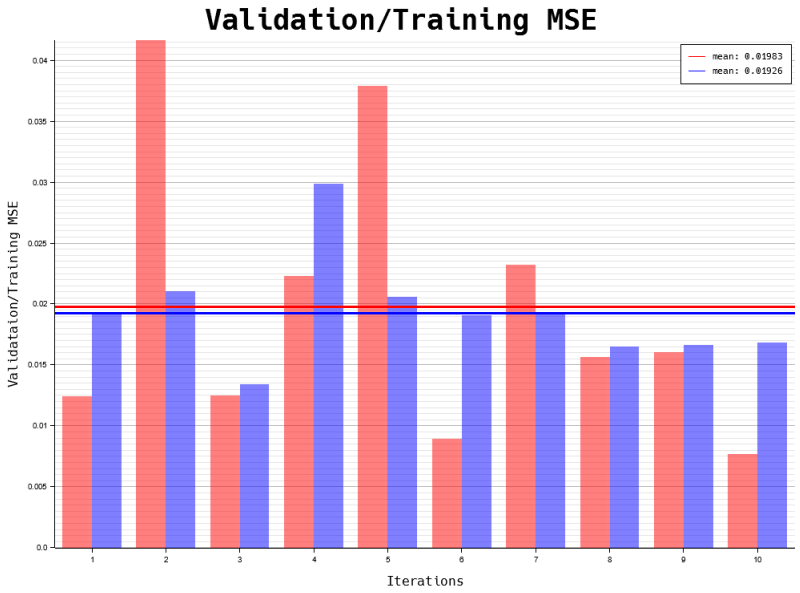
(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

Figure 3: Training result of Flood-8-4-1.

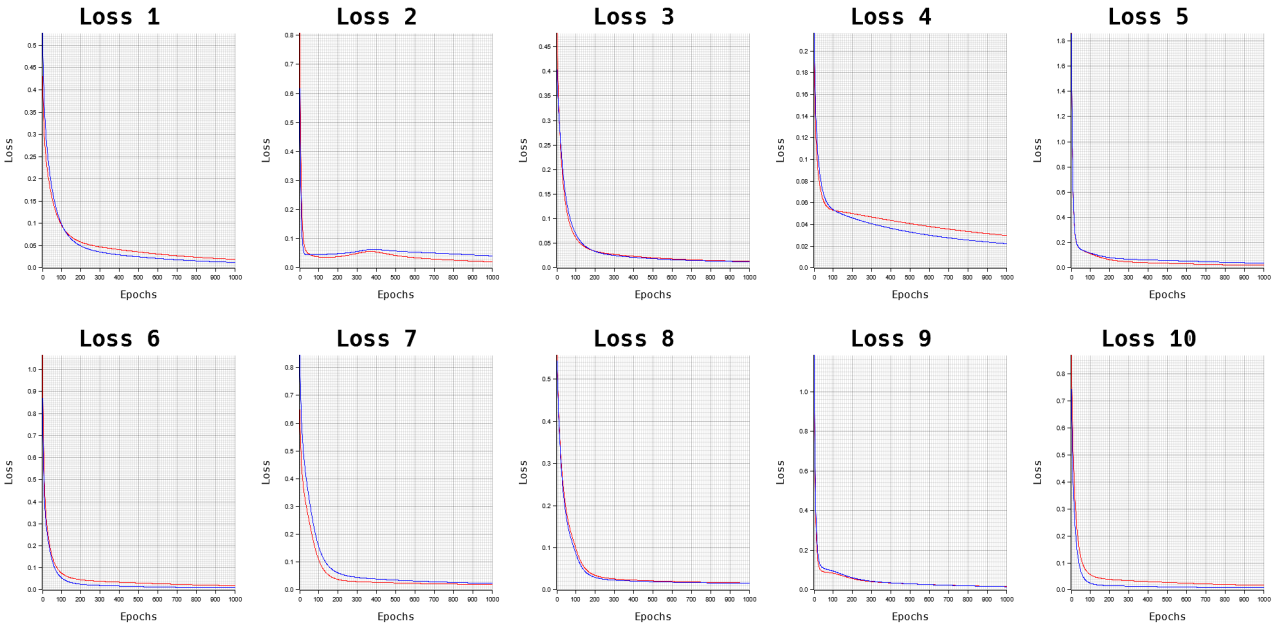
Flood-8-4-1 with small learning rate

Same base network but train with $lr = 0.0001$ and $momentum = 0.01$.

Below is the graphs we get from training this network. The training process take about 59 seconds.



(a) Each iteration training (blue) and validation (red) MSE at last epoch



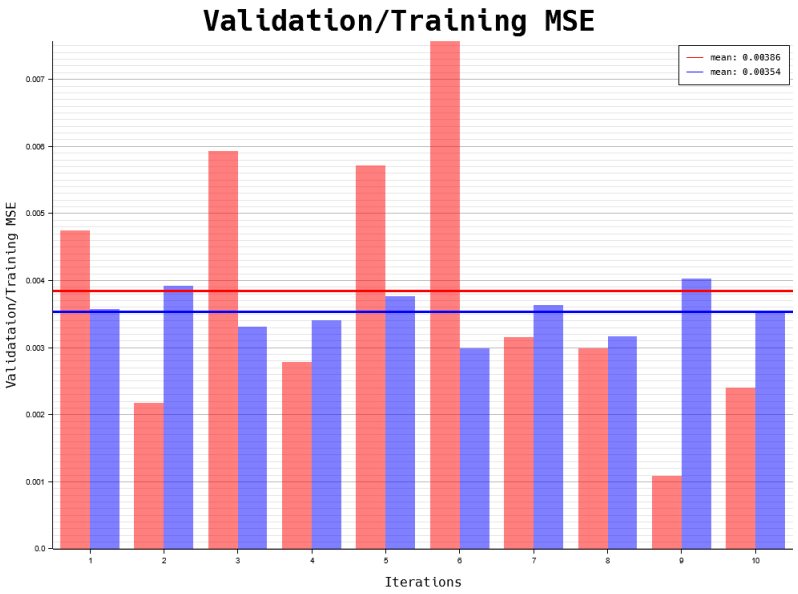
(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

Figure 4: Training result of Flood-8-4-1.

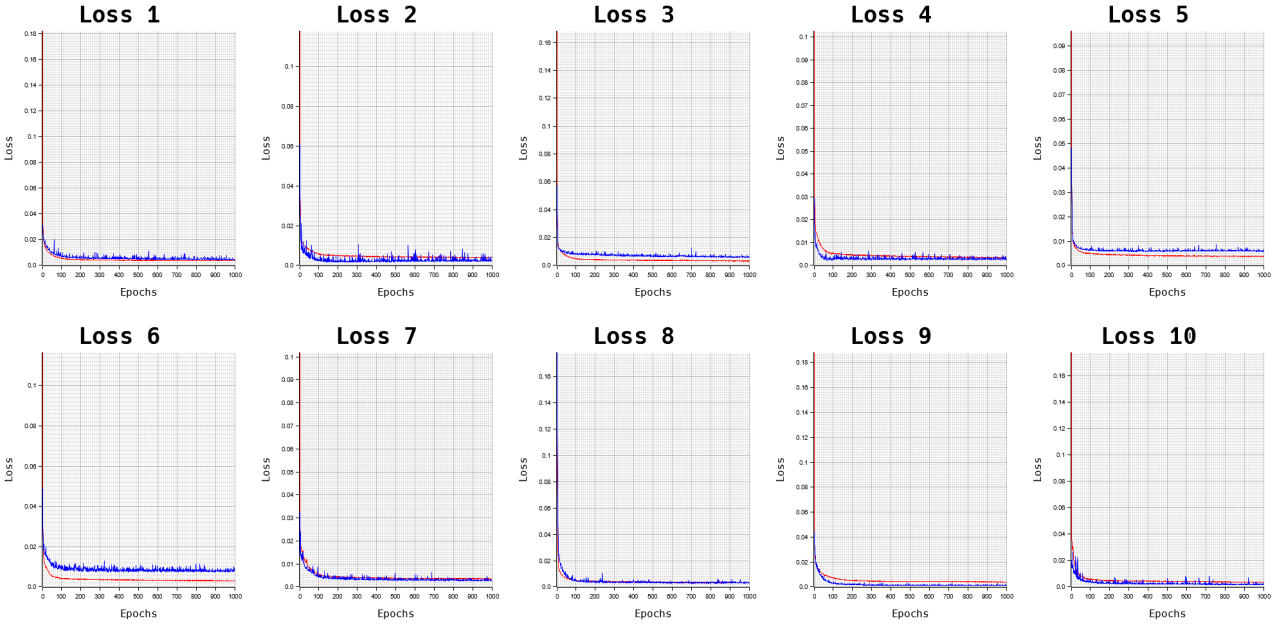
Flood-8-8-1

Bigger network that contains 8 input nodes, 1 hidden layer with 8 nodes, and 1 output node train with $lr = 0.01$ and $momentum = 0.01$.

Below is the graphs we get from training this network. The training process take about 105 seconds.



(a) Each iteration training (blue) and validation (red) MSE at last epoch



(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

Figure 5: Training result of Flood-8-4-1.

Analysis

From table 1, we can see that network with same size will use around the same amount of training time, but with bigger network the training time get longer. We can also see that Flood-8-4-1 with small learning rate is not performing well (has the biggest validation set mean MSE) compare with other network, the network seem to be stuck at this gradient level (see fig. 4 the network is slowly converging but too slow). Lastly, we can also see that the best performing network Flood-8-8-1's validation set mean MSE is not much better than Flood-8-4-1 and Flood-8-4-1 with no momentum, the biggest different is just 0.34×10^{-3}.

Model	Training Time (seconds)	Validation Set Mean MSE (10^{-3})
Flood-8-4-1	60	4.03
Flood-8-4-1 with no momentum	60	4.20
Flood-8-4-1 with small learniing rate	59	19.83
Flood-8-8-1	105	3.86

Table 1: Training time and validation mean MSE (blue line on (a) fig. 2 - fig. 5) of each Flood network.

2 CrossPat Dataset

Problem

We want to predict the class (1 of possible 2 classes) that belong to our inputs (or features).

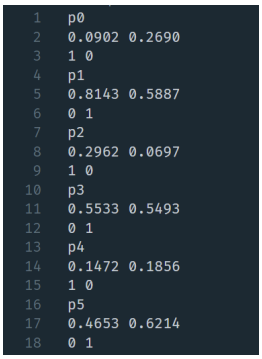


Figure 6: Examples of given data where px is an object and first line after it is its features, second line is its class

Parameters Setting

- All nodes use *sigmoid* as an activation function.
- Weights is a random number between $[-1, 1]$
- Each layer's bias is 1
- Use MSE (Mean Squared Error) as a loss function.

Training

Use 10% cross-validation with no data preprocessing, and train with SGD (Stochastic Gradient Descent) algorithm. Then, we train each cross-validation set for 5000 epochs.

We will create one base network that should perform good enough and create a variation base on that network, that is train with no *momentum*, train with smaller *learning rate*, and add more layers or hidden nodes to see that if we introduce those variations, will the network perform better, converge faster or no improvement at all?

Training Result

Accuracy is calculate using this equation $\frac{TP+TN}{TP+TN+FN+FP}$ where TP, TN, FN, FP comes from confustion matrix.

Cross-2-4-1

Our base network with 2 input nodes, 1 hidden layer with 4 nodes, and 1 output node train with $lr = 0.01$ and $momentum = 0.01$. We use only 1 output node because this is a binary classification task so we can just map a pair $(1, 0) \rightarrow 1$ and $(0, 1) \rightarrow 0$

Cross-2-4-1 with no momentum

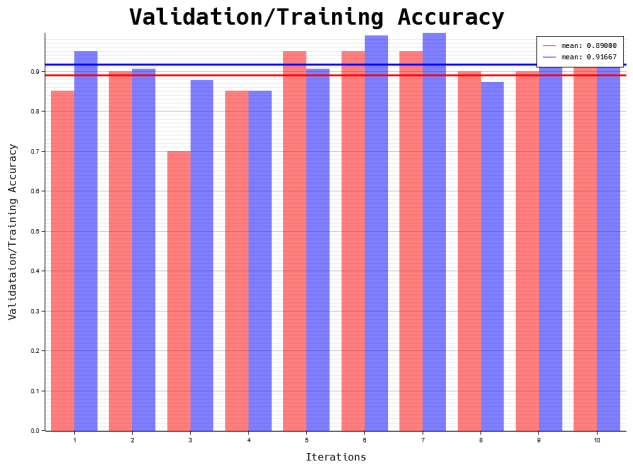
Same base network train with $lr = 0.01$ and $momentum = 0.0$.

Cross-2-4-1 with smaller learning rate

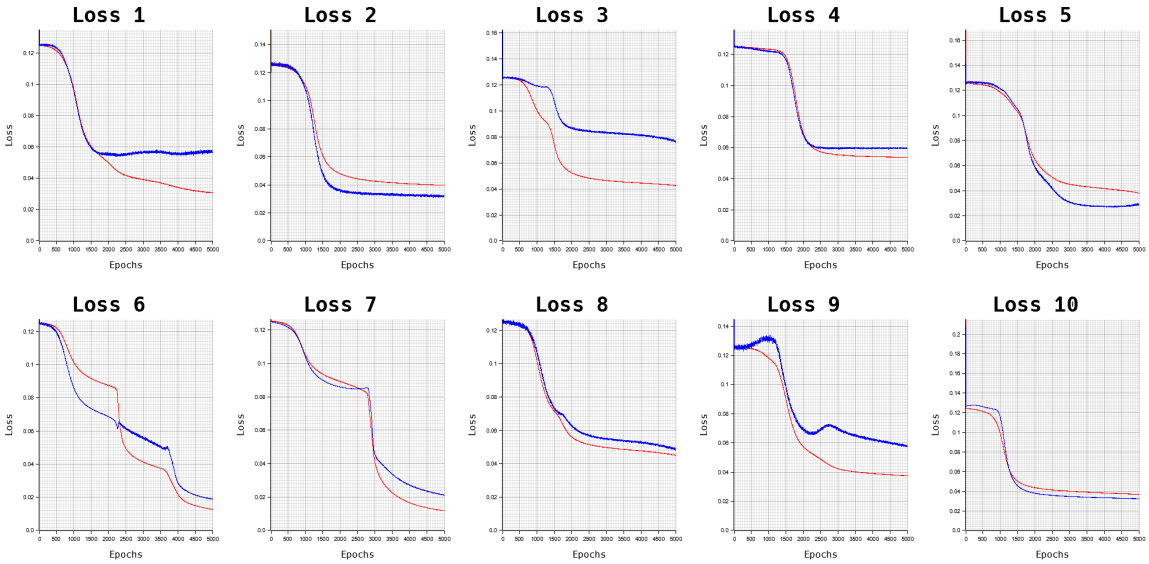
Same base network train with $lr = 0.0001$ and $momentum = 0.01$.

Cross-2-8-1

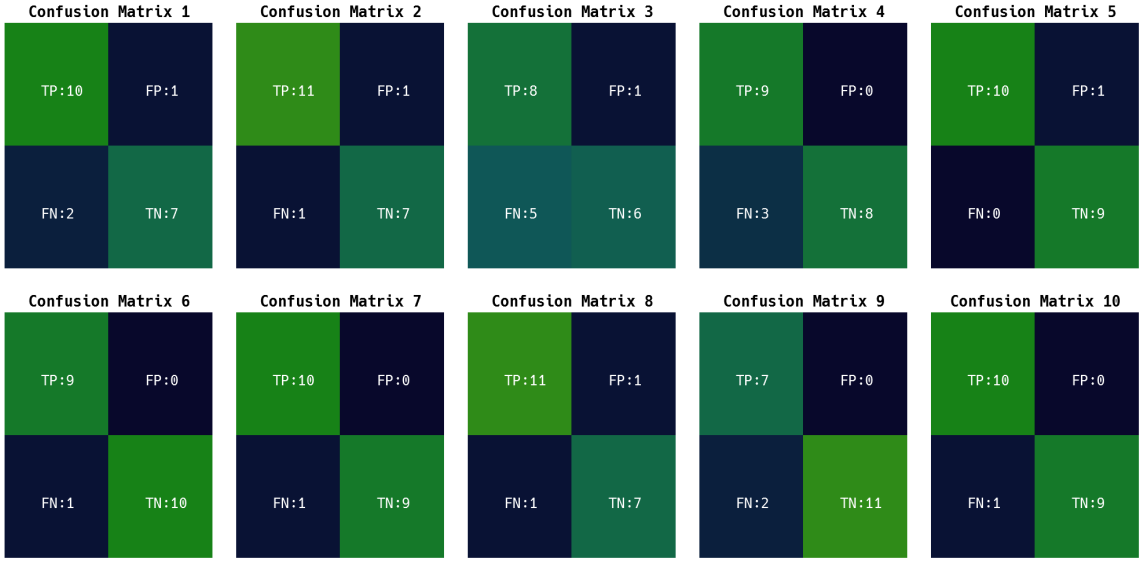
Bigger network that contains 2 input nodes, 1 hidden layer with 4 nodes, and 1 output node train with $lr = 0.01$ and $momentum = 0.01$.



(a) Each iteration training (blue) and validation (red) set accuracy at last epoch

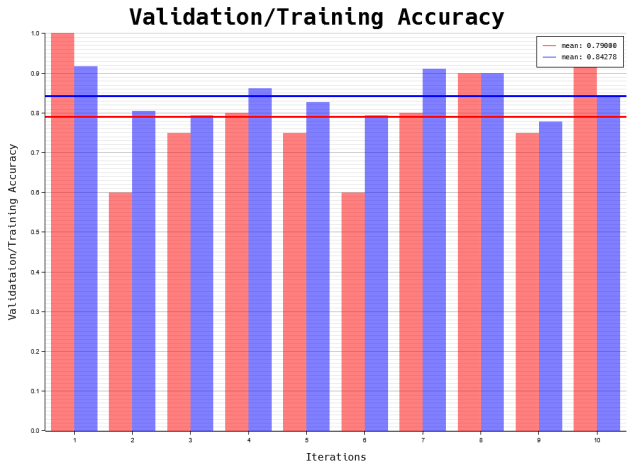


(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

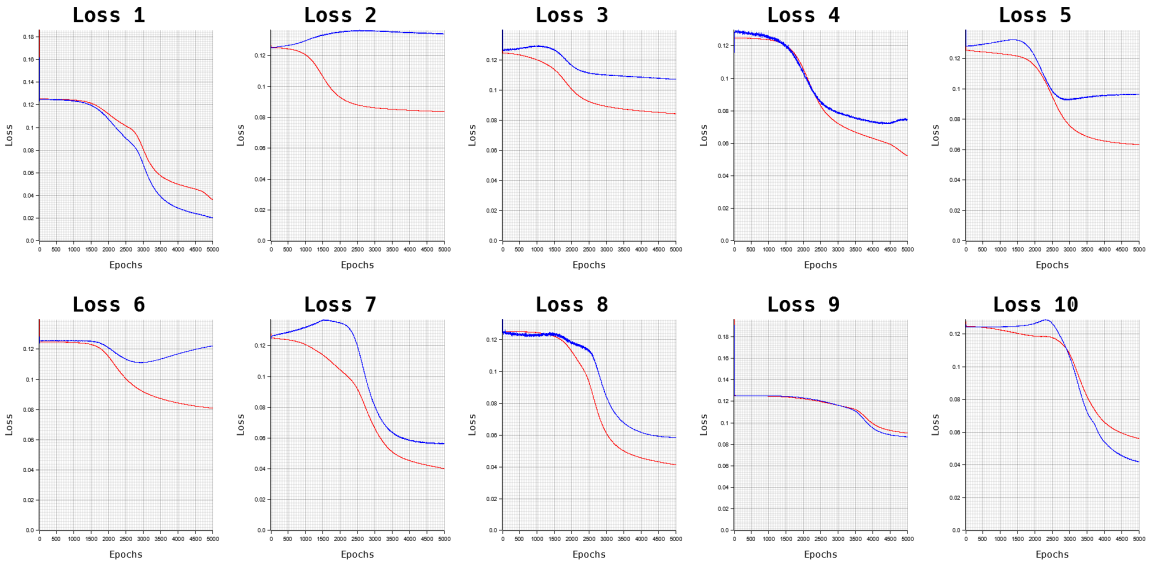


(c) Each iterations validation set confusion matrix where y-axis is an actual class $1, 0$ top to bottom and x-axis is predicted class $1, 0$ left to right.

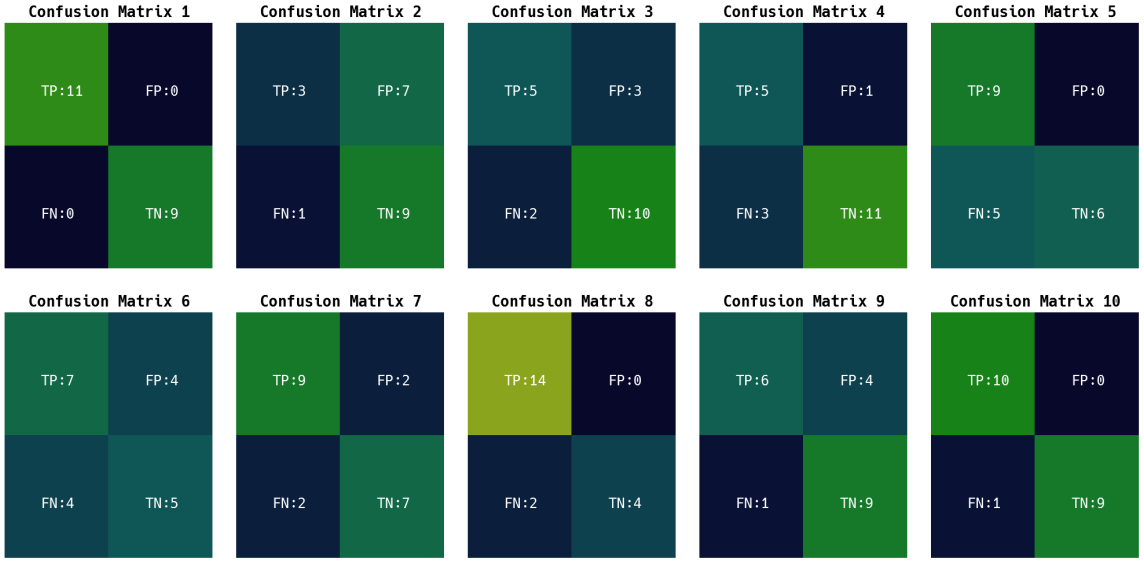
Figure 7: Training result of Cross-2-4-1.



(a) Each iteration training (blue) and validation (red) set accuracy at last epoch

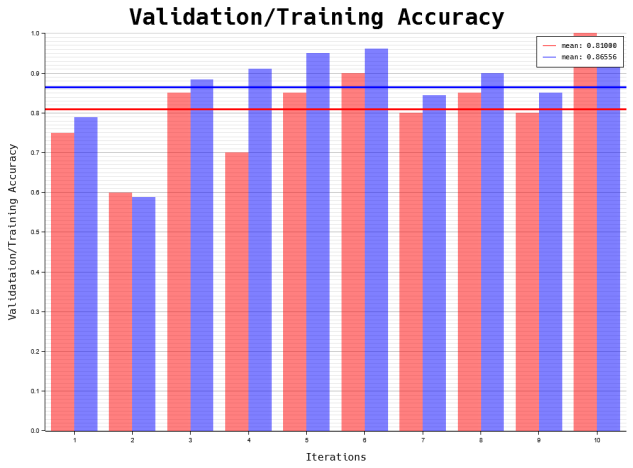


(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

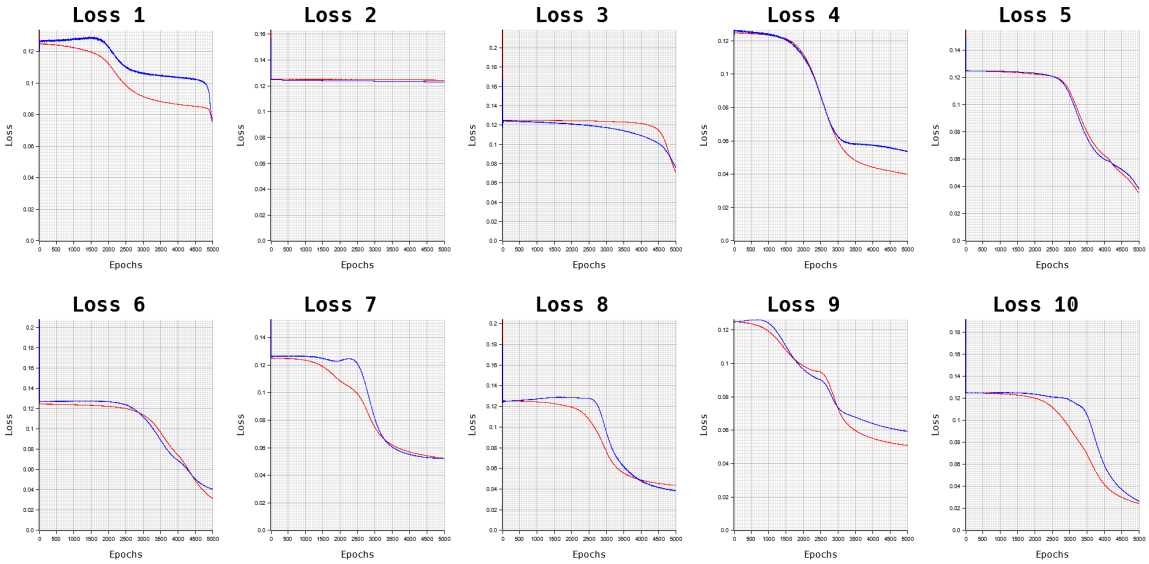


(c) Each iterations validation set confusion matrix where y-axis is an actual class $1, 0$ top to bottom and x-axis is predicted class $1, 0$ left to right.

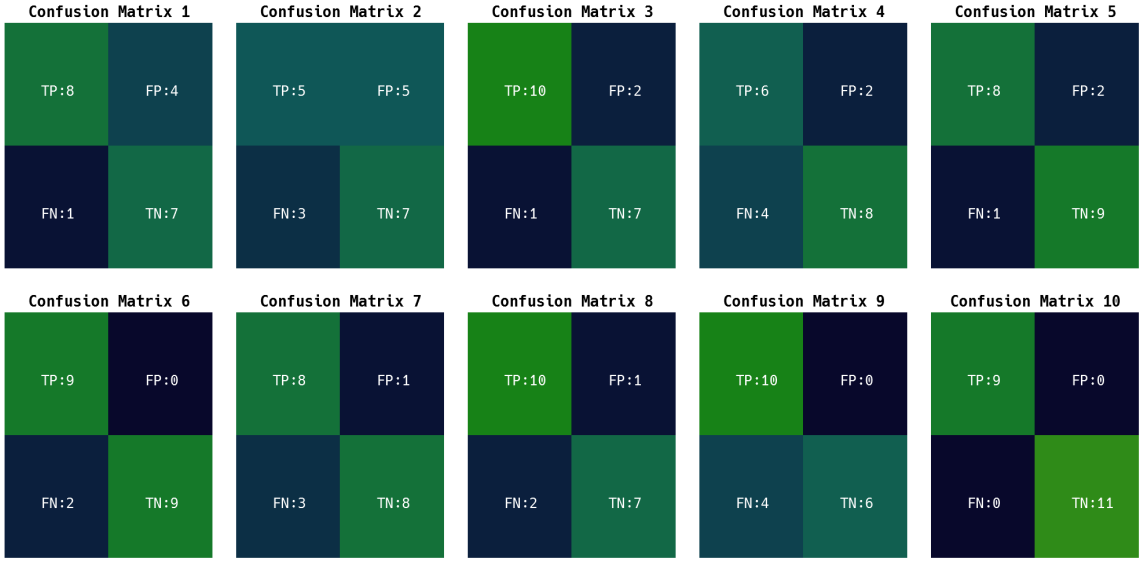
Figure 8: Training result of Cross-2-4-1.



(a) Each iteration training (blue) and validation (red) set accuracy at last epoch

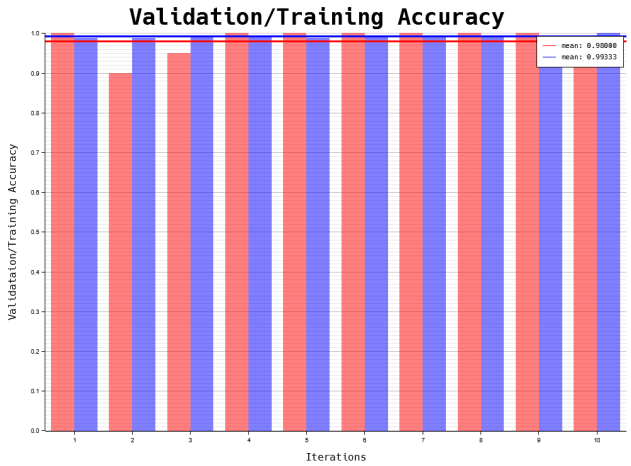


(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.

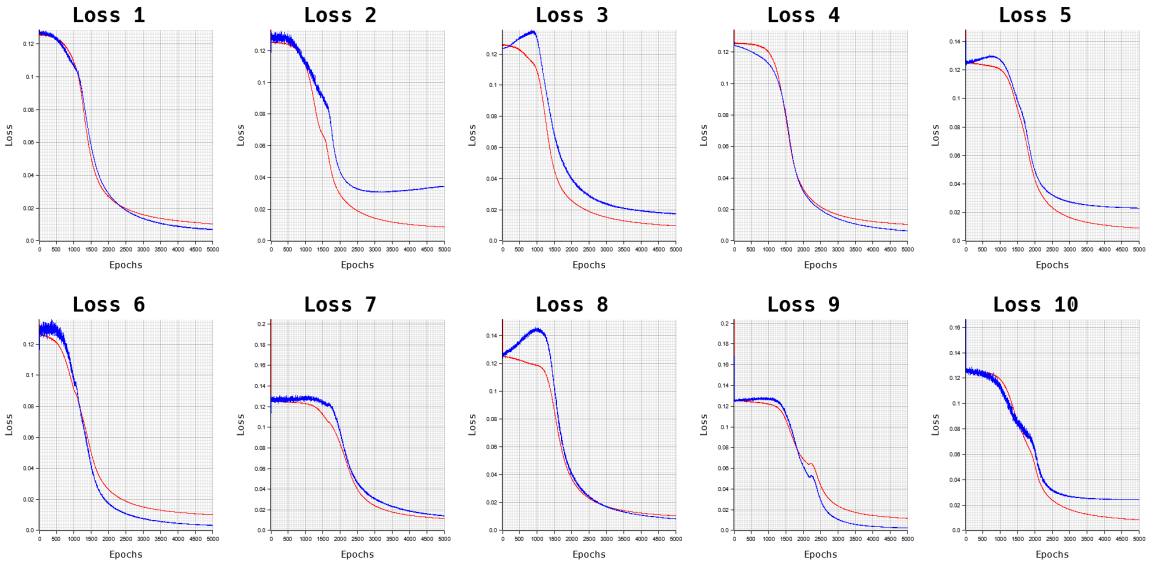


(c) Each iterations validation set confusion matrix where y-axis is an actual class $1, 0$ top to bottom and x-axis is predicted class $1, 0$ left to right.

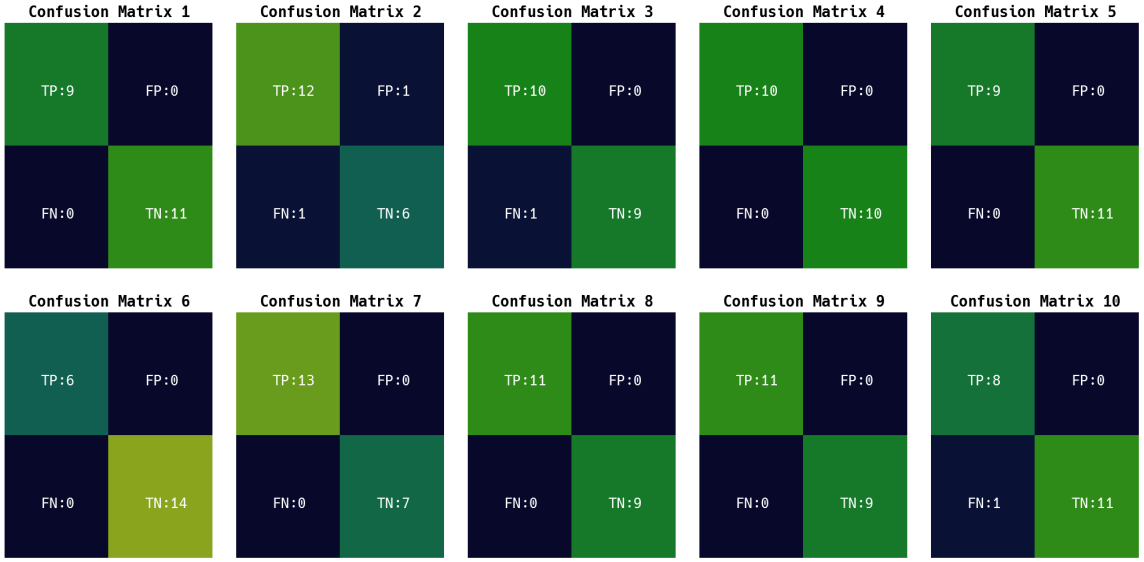
Figure 9: Training result of Cross-2-4-1.



(a) Each iteration training (blue) and validation (red) set accuracy at last epoch



(b) Each iteration training MSE (blue) and validation MSE (red) at each epoch.



(c) Each iterations validation set confusion matrix where y-axis is an actual class $1, 0$ top to bottom and x-axis is predicted class $1, 0$ left to right.

Figure 10: Training result of Cross-2-4-1.

Analysis

From table 2, we can see that

	Cross-2-4-1	Cross-2-4-1 with no momentum	Cross-2-4-1 with small learning rate	Cross-2-8-1
Training Time (seconds)	**121**	123	122	218
Validation Mean Accuracy	0.915	0.810	0.795	**0.970**

Table 2: Training time and validation mean accuracy (blue line on (a) fig. 7 - fig. 10) of each Flood network