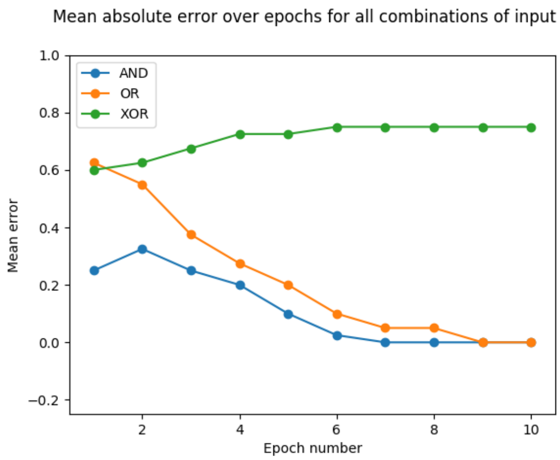
Report Assignment 3

Group 28

Part 2.1: question 1



For each logical operation, the perceptron was run 10 times with 10 epochs. As expected, AND and OR converge to error 0, while XOR operation converges to 0.75 error.

Question 2

10, because we have 10 features.

Question 3

7, since we have 7 target classes.

Question 4

8 neurons per hidden layer and 2 hidden layers.

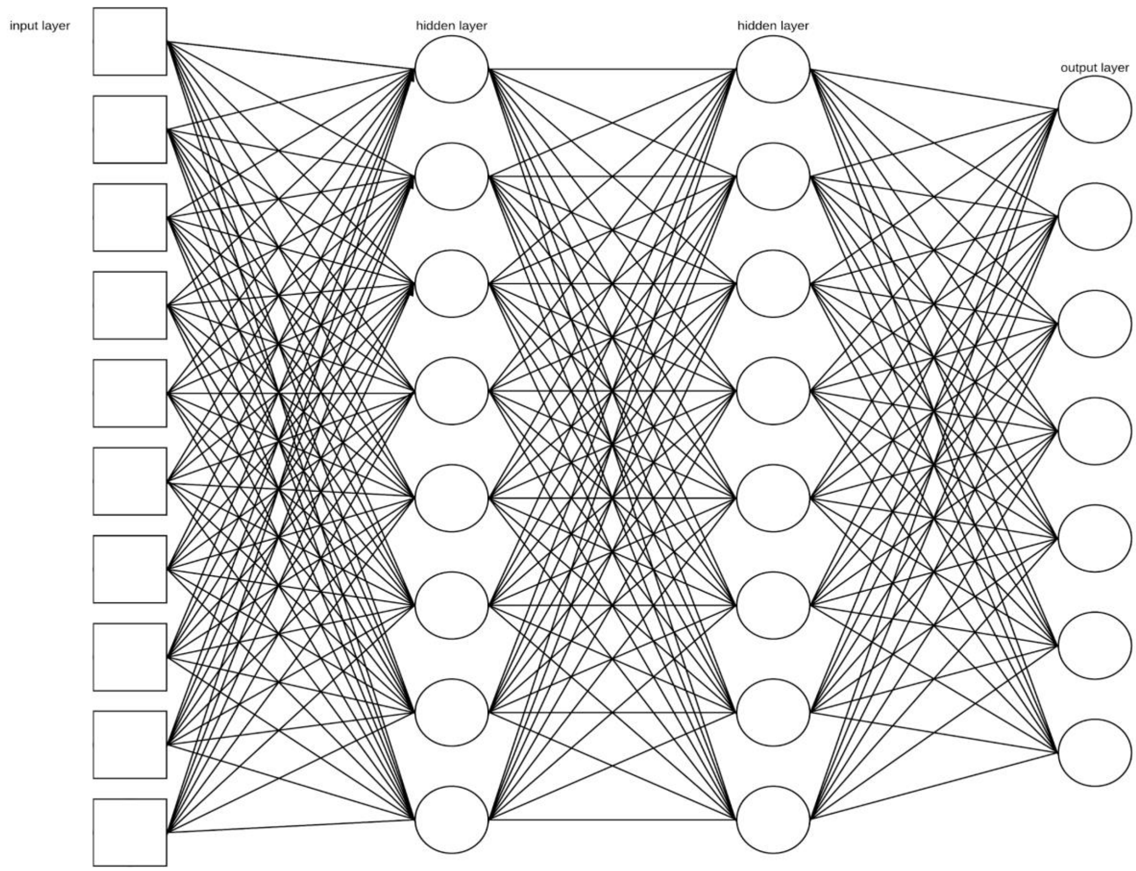
Question 5

The neurons in the back-propagation network use a sigmoid activation function:

$$Y^{sigmoid} = \frac{1}{1 + e^{-X}}$$

This means that each neuron will have an activation between 0 and 1 and in the output layer, we choose the neuron with the highest activation as the predicted target for that sample.

Question 6



Part 2.2: question 7

We divided our set so that the first 6500 elements are in the training set, the next 500 elements are in the validation set and the remaining amount of elements are in the test set. This is under the assumption that the training set should be big enough to contain presumable edge cases, the validation set should be representative measure of overfitting and the test set should be the final measure of how well our data set works.

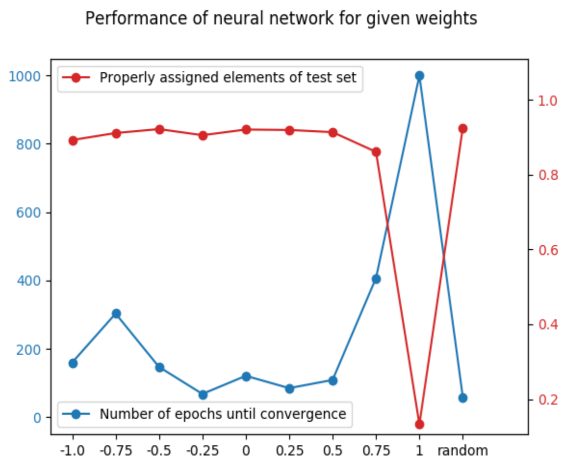
Question 8

We evaluate the performance of our network by the % of elements in the test set that got assigned to the proper class. We do it only once, after we finish the training. The test set is never used for training.
To evaluate performance of the training after every epoch, we use the validation set. It’s elements are also never used for training.

Question 9

We decide to stop training the neural network under the conditions that (1) the validation set has at least 50% elements classified properly, (2) there were at least 10 epochs, (3) the progress of training became stagnant, (4) the network is leaning towards overfitting.
As the measure for condition(3), we compute the mean of the last 10 epochs of the % of properly classified validation sets and check if the last 10 results differ by more than a set value from that mean. As the measure for condition (4), we check that if the first 3 conditions are fulfilled, if the validation set % of properly classified elements is decreasing, it means that the network is overfitting.

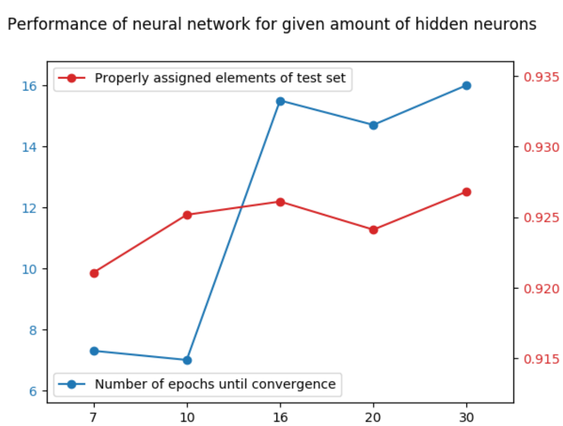
Question 10



The network was initialized 9 times with set weights as described on axis x. The 10th time we initialized the network with random weights as advised by the course book Artificial Intelligence, in the range (-2.4/F_i, 2.4/F_i) where F_i is the number of inputs of the given neuron. All of the trials were run with learning rate 0.05.
While the best performance occurred for randomized weights, it’s visibly the best for weights between -0,5 and 0,5. For numbers smaller than given range the performance isn’t as bad as for weights bigger than the given range.
For the weight 1 the success rate was random (1/7) until the training timed out on 1000 epochs.

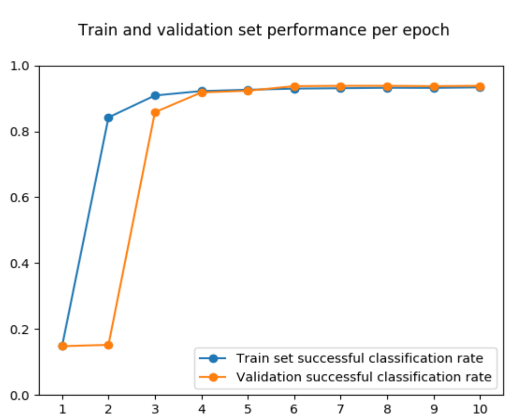
Part 2.3: question 11

For cross-validation, we decided to use elements 0-7000 for train and validation sets, and the rest elements for test set. We decided to use 7-fold cross-validation, so we have 7 validation sets consisting of 0-1000 indinces of elements, 1000-2000 indices of elements, 2000-3000, etc until 7000, and corresponding train set consist of the remaing elements in the first 7000 indexed elements.



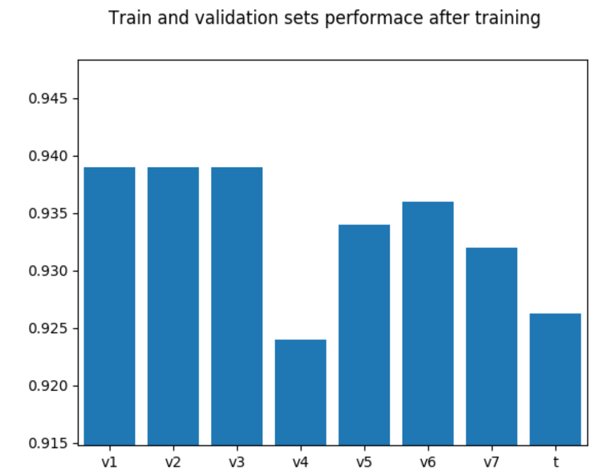
For each number of neurons, the given amounts are averaged from 10 separate trainings of the neural network.
The networks with 7 and 10 hidden neurons had 1 hidden layer, the other networks had 2 hidden layers with the neurons spread evenly between them.
When using only 1 hidden layer, the networks need much less time until convergence. For a network with 1 hidden layer, if there are at least as many neurons as in the input layer, the network converges to a much better classification rate.

Question 12



Since we had 7 train sets and seven corresponding validation sets as per cross-validation requirement, and the question asked only about 1 train/validation set graph, we decided to chose a random train set and corresponding validation set.
The train set consisted of elements between 0-4000 and 5000-7000, while the validation set consisted of elements between 4000 and 5000.
We chose the architecture with two hidden layers with 15 neurons, as the exercise 11 revealed it was the most successful on average.

Question 13



Values v1-v7 represent 7 validation sets at the end of training. T represents the test set at the end of training. While the test set performs slightly worse than the best of validation sets, it’s expectable since because of cross validation they are still included in the train set, 6 out of 7 training sessions. However, the test set doesn’t have the worst performance when compared to all the validation set, so we can conclude that the network isn’t overfitted, as in over 92,5% of the cases it classified the results properly

Question 14

		actual class						
		1	2	3	4	5	6	7
predicted class	class 1	107	1	2	1	1	0	3
	class 2	0	121	0	0	2	2	1
	class 3	3	1	113	1	2	0	1
	class 4	1	0	0	95	2	1	6
	class 5	1	0	4	0	111	5	0
	class 6	0	0	3	2	1	137	3
	class 7	1	1	2	6	1	2	107

Figure 1 Confusion matrix for the neural network test set

		actual class									
		1	2	3	4	5	6	7	false positive	true positive	PPV
predicted class	class 1	107	1	2	1	1	0	3	8	107	0,9304347826
	class 2	0	121	0	0	2	2	1	5	121	0,9603174603
	class 3	3	1	113	1	2	0	1	8	113	0,9338842975
	class 4	1	0	0	95	2	1	6	10	95	0,9047619048
	class 5	1	0	4	0	111	5	0	10	111	0,9173553719
	class 6	0	0	3	2	1	137	3	7	137	0,9513888889
	class 7	1	1	2	6	1	2	107	13	107	0,8916666667
	FN	5	3	14	10	9	10	14			
	TP	107	121	113	95	111	137	107			
	TPR	0,955357140	0,9758064516	0,8897637795	0,9047619048	0,925	0,9319727891	0,8842975207			

False Positive occurs for the values that were classified as class x despite belonging to a different class. False Negative occurs when an element was classified as a different class despite belonging to a given class
PPV or precision is the value of TP/TP+FP. The neural network has the worst precision in classifying class 7 and class 4.
TPR or true positive rate is the value of TP/ TP+FN. The value misclassifies elements from class 3 and class 7 the most.
In general, class 7 is the most troublesome for this neural network.

Part 2.5: question 16

```
Grid search on grid:
-hidden_layer_sizes: [[5, 5], [5, 5, 5], [15, 15], [15], [8, 8]]
-activation: ['logistic', 'identity']
-alpha: [0, 0.01]
-solver: ['adam']
-learning_rate_init: [0.005, 0.01, 0.02]
-max_iter: [200, 1000]
-batch_size: ['auto']
-validation_fraction: [0.058672640000000005]
-early_stopping: [True]
-n_iter_no_change: [10]
-tol: [0.0001]

Fitting 5 folds for each of 120 candidates, totalling 600 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 12 concurrent workers.
[Parallel(n_jobs=-1)]: Done  26 tasks      | elapsed:    7.8s
[Parallel(n_jobs=-1)]: Done 176 tasks      | elapsed:   33.4s
[Parallel(n_jobs=-1)]: Done 426 tasks      | elapsed:  1.0min
[Parallel(n_jobs=-1)]: Done 600 out of 600 | elapsed:  1.2min finished

Best parameters found by grid search:
-activation: identity
-alpha: 0
-batch_size: auto
-early_stopping: True
-hidden_layer_sizes: [15]
-learning_rate_init: 0.01
-max_iter: 1000
-n_iter_no_change: 10
-solver: adam
-tol: 0.0001
```

Figure 2 Grid search

Our current parameters:
hidden layer sizes [15],
learning rate: 0.02,
iteration size: 1000

```
precent of correct predictions: 0.9238875878220141
```

As we can see, the learning rate differs with what we have, but the optimal iterations and hidden layer size are the same.

Question 17

After changing the learning rate to 0.01 the percent of correct predictions has gone down:

```
precent of correct predictions: 0.1451990632318501
```

In conclusion, the parameters from the grid search do not give us better performance.
We assume that the percentage of correct predictions is so off due to not adding our alpha to the grid search.