Machine Learning CBC Assignment 3 Report

Aadil Khan, Gareth Jones, George Eracleous, Valdas Kriauciukas
Email: {aak08, gdj08, ge108, vk308}@imperial.ac.uk
Group Number: 23
Tutorial Helper: Brais Martinez

Table of Contents

Implementation Details

For this task we had to assess the performance of two types of neural network at an emotion classification problem. The first type of network is a single output network, which after training outputs 1 if the example belongs to the emotion and 0 otherwise. Because we are testing six different emotions it follows that we will built six one-output networks to test our data.

The second network type consists of a single network with six outputs. Each output corresponds to a different emotion class.

In these types of network we had to make sure that only one emotion was selected for an example since our classifiers could assign an example to multiple emotions. To ensure that this is the case we picked the classifier with the maximum output value (most likely).

The results obtained for 10-fold cross validation for the two types are presented in the following sections. After the results are presented we discuss the advantages and disadvantages of these configurations.

(A) Discussion of network parameters

Specific parameters to train each network are needed to improve the rate in which a network performs classification. These include the number of hidden layers in the network, number of neurons in the hidden layers, the learning rate of the system, different activation functions and different training functions.

As the number of hidden layers and the number of neurons in the hidden layers increases the neural network runs the risk of over-fitting, for a binary classification problem two layers was sufficient.

If the learning rate of the system is too high then we may overshoot the optimum learning parameters and therefore degrade the performance of the system. If the learning rate was to slow then we again run the risk of overfitting.

This shows the effect that having incorrect parameter settings can have on our network and so paved the way for us testing out different values to see what works best for these specific classification problems.

We measure the performance during our experiments in searching for the most optimal set of parameters. The measure we have chosen to compare the parameters of the networks is the F1 value. The F1 value has its advantage of being a single number to represent the performance of the network while encapsulating recall and precision.

Our approach to finding the most optimal parameters is through human observation of the results of the experiments. While trying out different network configurations we only altered one parameter while keeping all the others constant to see the effect it had on the results. The experiments could have been done through exhaustive search through the combinations of each set of parameters and recording the F1 measure, however, such a test would result in a large search space and take a long time to compute.

The Parameters

The parameters that were subjected to optimization for increased performance of our trained networks are; the number of hidden layers, the number of neurons for each layer, the learning rate, the activation function and the training function.

The number of hidden layers is largely influenced by that stated by Mitchell in his book, Machine Learning. Mitchell advises that two layers would be an ideal choice to train neural networks, it is most common for an arbitrary task. This was confirmed, through our experiments (using a range of hidden layers reaching 10 layers), that 2 layers provides us with a good performance. It is worthy to note that we are tackling a particular emotion recognition problem which is a task that doesn't demand much complexity. The more layers we have, the more complex paths there are for information to flow through. Standing behind theory and empirical evidence, we have chosen 2 hidden layers.

There is a risk of overfitting if we were to choose a large number of neurons for each hidden layer. The time taken to train a network with a large number of neurons (100, in our case) blows up exponentially. We use a range of values to find a number that would settle with high performance and yet avoids overfitting. We limit our search a number of neurons below 45 (the number of neurons of the input layer). We chose to take a fraction of the number of neurons of the input layer and experimented with 15, 30 and 45. Analysing the F1 measures didn't give us much of a difference between 15 and 30 neurons and so we settled with 15 with the idea that classification is done more efficiently with a smaller network.

The learning rate values we used during our experiments ranged from 0.00002 to 0.003 to 0.02. The pattern we observed was that as the learning rate increases the network's ability to find a classification in a small number of epochs decreases. This is expected as large 'jumps' taken to re-adjust the weights during the actual training can overshoot the required values of those weights. A value of 0.001 was chosen as the optimum as values lower than 0.001 (such as 0.00002) would result in overfitting (as well as prolonging the training of a network).

There are three transfer/activation functions used in pattern recognition; Logsig, Purelin and Tansig. First of all, Purelin can be eliminated as a possible parameter since the experiments clearly showed an immense amount of overfitting of the training data. There is a slight comparison between Logsig and Tansig, though, with a 2 layer neural network, Tansig is advised to be a better choice in performance.

The last of the parameters to optimize is the training function. The controlled experiments involved functions such as Levenberg-Marquardt backpropagation (trainlm), Scaled conjugate gradient backpropagation (trainscg), Bayesian regulation backpropagation (trainbr) and many more (where applicable). We ended up using Bayesian regulation backpropagation, not for the reason of optimal performance, but, for its approach to avoid overfitting among our small data set for emotion classification. More is discussed on Bayesian regulation below.

(B) Individual F1 Measure comparison for 10-fold cross validation

This plot displays the F1 metric per fold for each type of network.

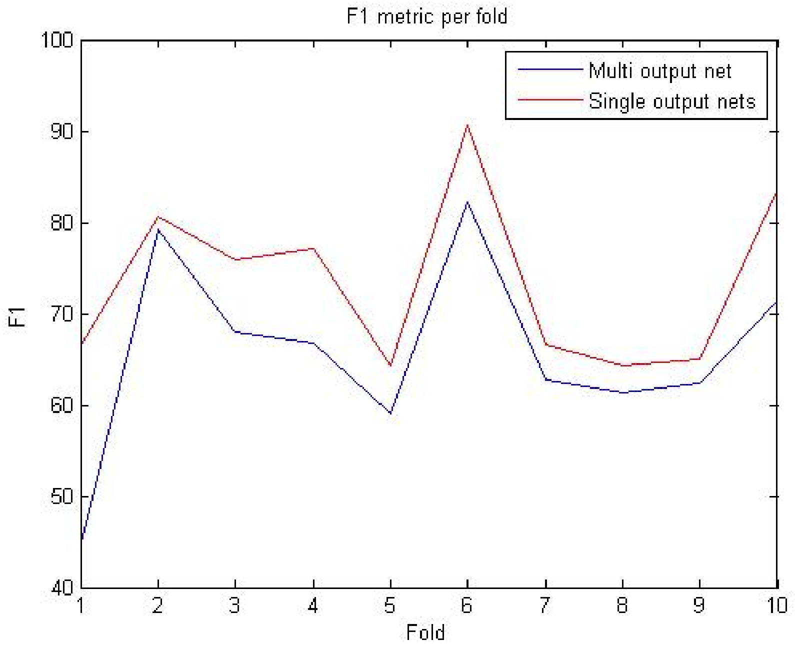
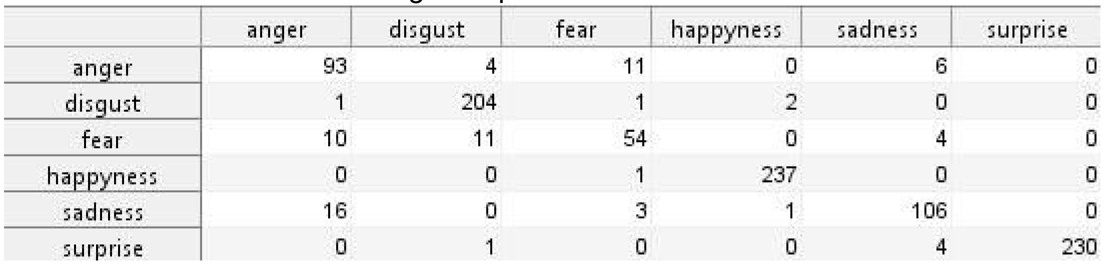


Figure 1: F1 metric for each fold for both types of neural nets. Blue is multiple output single net, red is single output multiple nets.

(C) Confusion Matrices and Average Performance Metrics (10-fold cross validation)

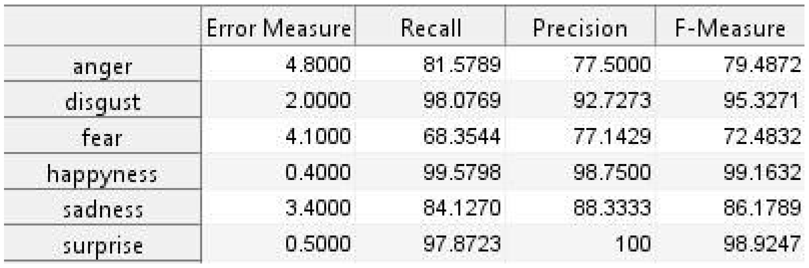
Single Output (6 Networks)

The confusion matrix for six single-output networks.



	anger	disgust	fear	happyness	sadness	surprise
anger	93	4	11	0	6	0
disgust	1	204	1	2	0	0
fear	10	11	54	0	4	0
happyness	0	0	1	237	0	0
sadness	16	0	3	1	106	0
surprise	0	1	0	0	4	230

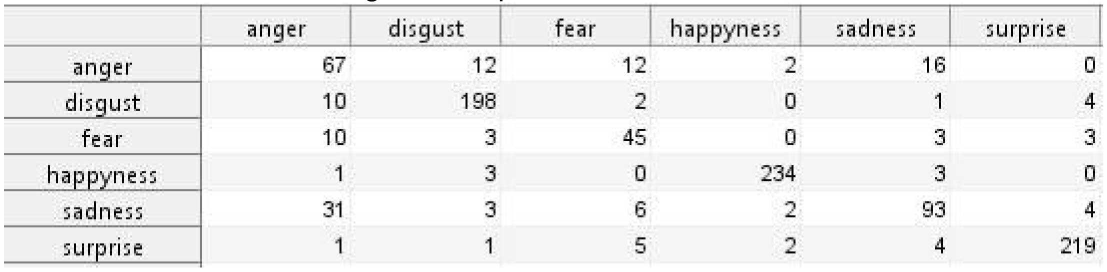
This plot displays the error measure, precision, recall and F1 metric for the six single-output networks per emotion class.



	Error Measure	Recall	Precision	F-Measure
anger	4.8000	81.5789	77.5000	79.4872
disgust	2.0000	98.0769	92.7273	95.3271
fear	4.1000	68.3544	77.1429	72.4832
happyness	0.4000	99.5798	98.7500	99.1632
sadness	3.4000	84.1270	88.3333	86.1789
surprise	0.5000	97.8723	100	98.9247

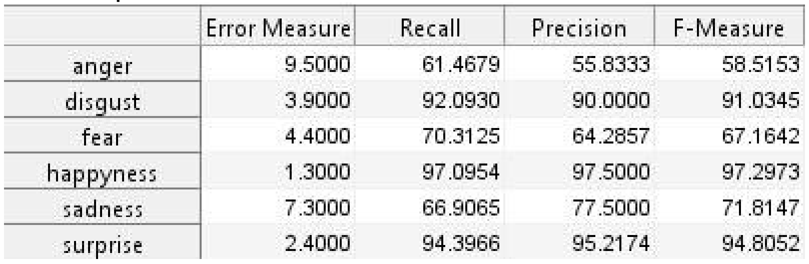
Multiple Output (Single Network)

The confusion matrix for single six-output networks.



	anger	disgust	fear	happyness	sadness	surprise
anger	67	12	12	2	16	0
disgust	10	198	2	0	1	4
fear	10	3	45	0	3	3
happyness	1	3	0	234	3	0
sadness	31	3	6	2	93	4
surprise	1	1	5	2	4	219

This plot displays the error measure, precision, recall and F1 metric for the single six-output network per emotion class.



	Error Measure	Recall	Precision	F-Measure
anger	9.5000	61.4679	55.8333	58.5153
disgust	3.9000	92.0930	90.0000	91.0345
fear	4.4000	70.3125	64.2857	67.1642
happyness	1.3000	97.0954	97.5000	97.2973
sadness	7.3000	66.9065	77.5000	71.8147
surprise	2.4000	94.3966	95.2174	94.8052

(D) Difference between classification approaches

It is clear from the results presented above that the single-output network configuration outperforms the single six-output network approach in terms of the F1 metric. A possible explanation for the results could be that each one of the six different networks specialises on a single emotion during training and therefore becomes much better in recognizing an example of that class. On the other hand, the single multiple-output network has a harder task to do since it needs to classify every single example to a specific class.

A drawback of the six single-output networks is that training and classification takes much longer than the other approach. More specifically, we ran some experiments to measure the time needed to train …. The reason is that during training we need to train six different networks instead of just one. Therefore, if we are concerned about performance in terms of execution time the single six-output configuration might be a better choice but at the same time we sacrifice the classifier's performance in terms of the other metrics.

(E) Overfitting avoidance

Overfitting is one of the problems that can occur during the training of a neural network. This means that the error on the training set is very small, but when new examples are presented to the network the error becomes large. The reason is that the network has memorized the training examples, but it is not able to generalize to new data. In order to avoid overfitting we employed two commonly used methods: Early Stopping (ES) and Bayesian Regularization. During our experiments we experienced overfitting when we had more than 50 neurons per layer and when we had 3 hidden layers. We have used the error plot for the validation set during training to identify the cases when overfitting occured.

Early Stopping

The first approach we used to avoid overfitting was ES. In this technique the data set is divided into three different groups. The first subset is the training set, which is used for computing the gradient and the network weights and biases. The second subset is the validation set and the error of this subset is constantly monitored during training. When the overfitting starts to occur the error on the validation set begins to rise. When this happens the training is halted (after a specified number of iterations), and this gives the final network parameters. The third group is the test set but this is normally not used during training.

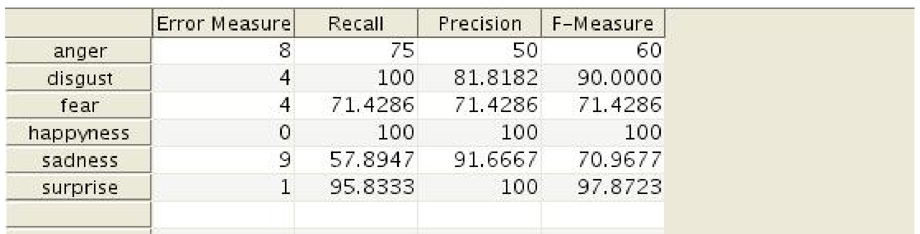
In order to implement ES we have divided our training set into the three subsets discussed above. We experimented with several divisions but we always ensured that the validations set was representative of the training set. We have used random division (dividerand). Also, with ES we had to make sure that the algorithm didn't converge too fast. For this reason we have chosen to use large mu values and we have found trainscg and trainbr to work well with ES.

Bayesian regularization

Regularization is a process which modifies the performance function. By including the sizes of the weights and biases, regularization produces a network that performs well with the training data and has a smoother behavior when presented with new data. Matlab's documentation indicates that BR is very effective when the data set is relatively small or without noise. Also, it is very effective in function approximation networks and less effective in patter recognition tasks. This is obviously a problem in our case since our task is to classify emotions. Finally, another problem with BR is that we have to specify a performance ratio parameter and sometimes this is hard to do.

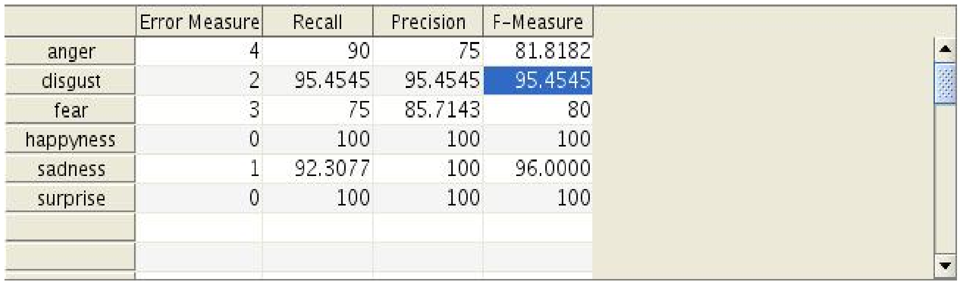
In order, to avoid specifying a performance ratio parameter we have used Matlab's built-in BR function which is *trainbr*. We took extra care to ensure that proper convergence has reached. The sum-squared error and the sum-squared weights should have constant values when the network has converged. When BR is used, the network takes longer to be trained because this method takes longer to converge than ES.

The plot below displays the error measure, precision, recall and F1 metric for the single six-output network per emotion class using BR.



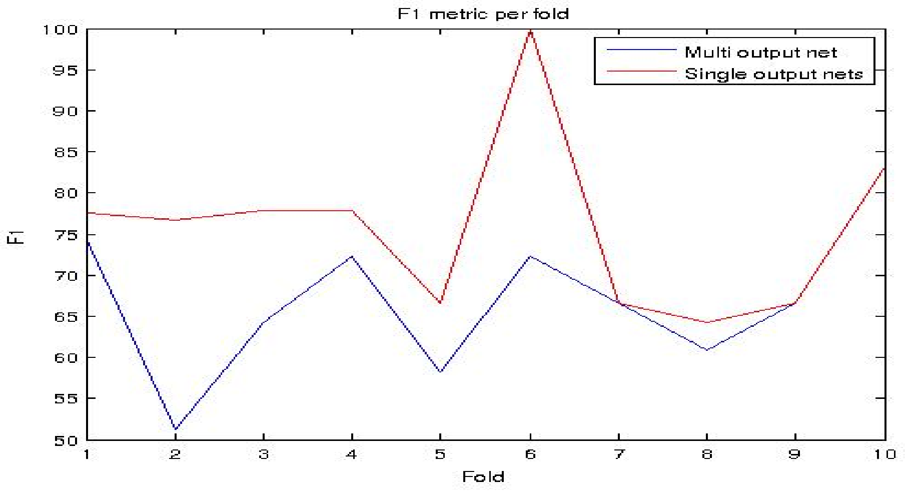
	Error Measure	Recall	Precision	F-Measure
anger	8	75	50	60
disgust	4	100	81.8182	90.0000
fear	4	71.4286	71.4286	71.4286
happyness	0	100	100	100
sadness	9	57.8947	91.6667	70.9677
surprise	1	95.8333	100	97.8723

The plot below displays the error measure, precision, recall and F1 metric for the six single-output networks per emotion class using BR.



	Error Measure	Recall	Precision	F-Measure
anger	4	90	75	81.8182
disgust	2	95.4545	95.4545	95.4545
fear	3	75	85.7143	80
happyness	0	100	100	100
sadness	1	92.3077	100	96.0000
surprise	0	100	100	100

Finally, the plot below demonstrates the F1 metric per fold when BR is used.


F1 metric per fold
Multi output net
Single output nets
F1
Fold