1 Discussion of Architectures

1.1 Dataset

The dataset contains 9 clinical features and a class variable for 64 patients with breast cancer and 52 healthy control patients, summing to 116 complete records (1160 individual data points). These features consist of age, BMI, 7 blood variables and a binary variable which denotes breast cancer or healthy control class; these are shown in the table below.

Feature Name	Units	Description	Mean	Standard Deviation
Age	years	Age of patient	57	16.1
BMI	kg/m2	Body Mass Index, value derived from the height and mass of a patient	27.58	5.02
Glucose	mg/dL	Concentration of blood glucose	98	22.5
Insulin	µU/mL	Concentration of blood insulin	10.01	10.07
HOMA		Homeostatic Model Assessment - Insulin Resistance. Interaction term: (Glucose*Insulin / 405) [1]	2.70	3.64
Leptin	ng/mL	Concentration of blood leptin	26.62	19.18
Adiponectin	µg/mL	Concentration of blood adiponectin	10.18	6.84
Resistin	ng/mL	Concentration of blood resistin	14.73	12.39
MCP-1	pg/dL	Concentration of blood Monocyte Chemoattractant Protein 1	534.65	345.91
Classification	n/a	Binary class, 1 denoting healthy control and 2 denoting patients with breast cancer	n/a	n/a

The data is assumed to be accurate, collected in a consistent way, within a controlled environment.

1.2 Problem

The aim is to construct an artificial neural network that can classify patients as having or not having breast cancer (that is to say, diagnose), using anthropometric data and blood test data. The problem is to engineer this network to be as accurate as possible and then determine whether it is reliably predictive.

1.3 Architectures

Given that this problem is a classification problem, a pattern recognition network is the solution. Pattern recognition networks attempt to predict discrete classes from a set of predictors (categorical and/or continuous). Pattern recognition networks can come in a variety of architectures.

An architecture that would not be appropriate for this problem would be a single perceptron neural network; the data is too complex and not linearly separable, therefore it would produce very inaccurate results. The opposite can be said for a Convolutional Neural Network (CNN) architecture; the data is not complex enough to justify using one. CNNs are commonly used for complex problems such as image classification [2,3]; a pattern recognition problem from 9 predictors for 2 classes can be accurately achieved using much simpler architectures that are not as computationally expensive as CNNs [4]; that is, using a CNN gains nothing more for more effort. CNNs also require a lot of training data [4], which this dataset cannot provide.

A Multilayer Perceptron (MLP) architecture may be suited to this problem. An MLP is a feedforward neural network which has at least three layers; an input layer, a hidden layer, and an output layer [5,6]. The number of hidden layers can be increased to provide a more accurate model, but overfitting may occur with an excess of layers [7]. All layers of an MLP are linear for pattern recognition problems [8]. MLP units compute the inner product of the inputs and weights [8].

Depending how the data is positioned within the higher dimensions of all 9 predictors, a Radial Basis Function (RBF) may produce better results than an MLP; this would be the case if each class' data points are more clustered about a centre rather than being dependant on some linear or curved decision boundary. RBFs have one input layer, one hidden layer and one output layer - of which there is no variation. Hidden nodes in an RBF have a different operation to the output nodes; their hidden layer is non-linear whereas the output layer *is* linear [8]. The hidden layer units compute the distance between the inputs and the centre of the unit [8].

A Self-Organising Map (SOM) is a neural network that produces a low-dimensional representation of its inputs - a *map* [6]. SOMs can solve classification problems but have problems when the number of data records is low or extraneous [6]; because the dataset only contains 116 records in total for 2 classes, a SOM architecture will not be used for this problem.

Considering the above, MLP and RBF architectures will be taken forward, tested and their results compared to one another to attempt to further select the architecture most suited to this classification problem.

2 Creation and Application of Neural Networks

2.1 Data Investigation and Preprocessing

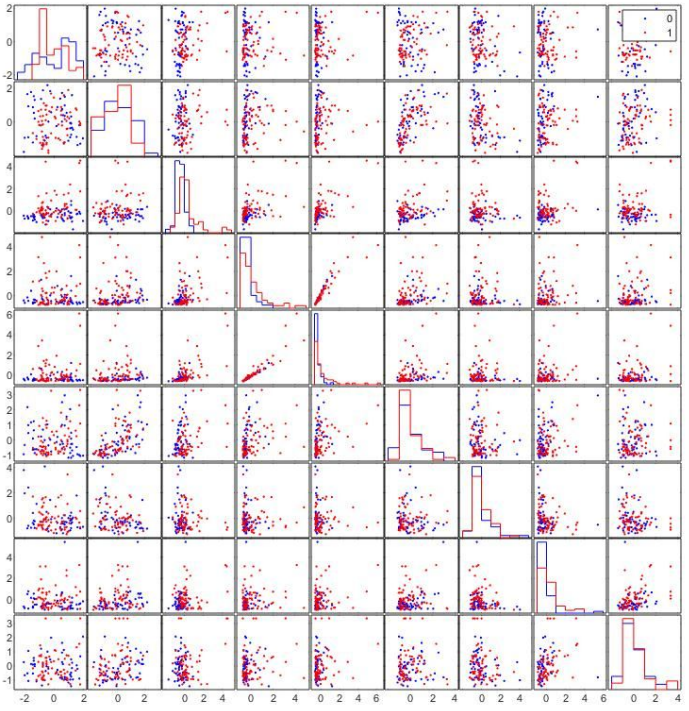
The dataset was loaded into a MATLAB environment and transposed to have features along rows and patient records along columns - because this orientation is the default for many of MATLAB's neural network functions and tools. The data was split into inputs (age, BMI, and the 7 blood variables) and targets (the binary classifier for presence of breast cancer/non-breast cancer). The binary values of the target variable were given as 1 and 2; this was transformed to have values of 0 and 1 as this is the form that the tools need the target data to be in.

The two target classes contain 64 and 52 records. The imbalance is not significant enough to result in poor learning of the data, therefore it does not need to be artificially balanced by duplicating or creating new records.

There are no missing values in the data.

To analyse the data more clearly, the inputs were transformed to have a mean of 0 and a standard deviation of 1 using the *mapstd* function in MATLAB. This allowed identification of potential outliers; there were 23 data points which were more than 3 standard deviations away from the mean. The 23 data points were evenly spread across the blood variables but 4 patient records (numbers 72, 79, 88 and 89) had multiple of the outliers - patient records 79 and 88 had 4

of these outliers each. This may suggest patients 79 and 88 had an erroneous blood sample, or suggest the presence of another illness - this should be clarified with the dataset creators or medical experts. The outliers should not be removed from the model yet as currently there is no evidence stating that these are in fact erroneous.



The input variables were analysed graphically using the *gplotmatrix* function, which plots histograms of each variable and plots each variable against each other. Data points were coloured by class, red representing the presence of breast cancer. This analysis allows the identification of strongly correlated variables. Two variables with perfect correlation together have no more effect (in terms of accuracy) on the neural network than if just one was used, therefore one can be removed; in addition, one of two variables with *strong* correlation also have little effect. The advantage of removing redundant variables is that the neural network has less to compute, reducing the number of dimensions of the input data, therefore becoming more computationally efficient. As shown by the *gplotmatrix* output on the right, variables 4 and 5 have strong correlation (where top and left plots start at 1) - these are Insulin and HOMA; this can be expected as HOMA is an interaction term partly comprised of Insulin. Removing Insulin produced very similar results and has therefore been removed.

The data was separated into training, validation and test datasets with proportions 70%, 15% and 15% respectively from the provided dataset.

2.2 Training Algorithm

The training algorithm used was a *scaled conjugate gradient backpropagation* algorithm (SCG). SCG is the default training algorithm for pattern recognition networks in the MATLAB neural network toolbox. The time complexity was considered using MATLAB's official documentation [9] which gives time comparisons between different training algorithms run on a number of different datasets. SCG performs comparatively well on *all* the benchmark datasets, performing better than the mean each time [9]. Therefore SCG is an appropriate training algorithm in terms of time complexity in relation to the dimensions of the problem dataset.

The SCG was used with its default parameters [10]. The functional parameters (those not applicable to display) and their values are listed below:

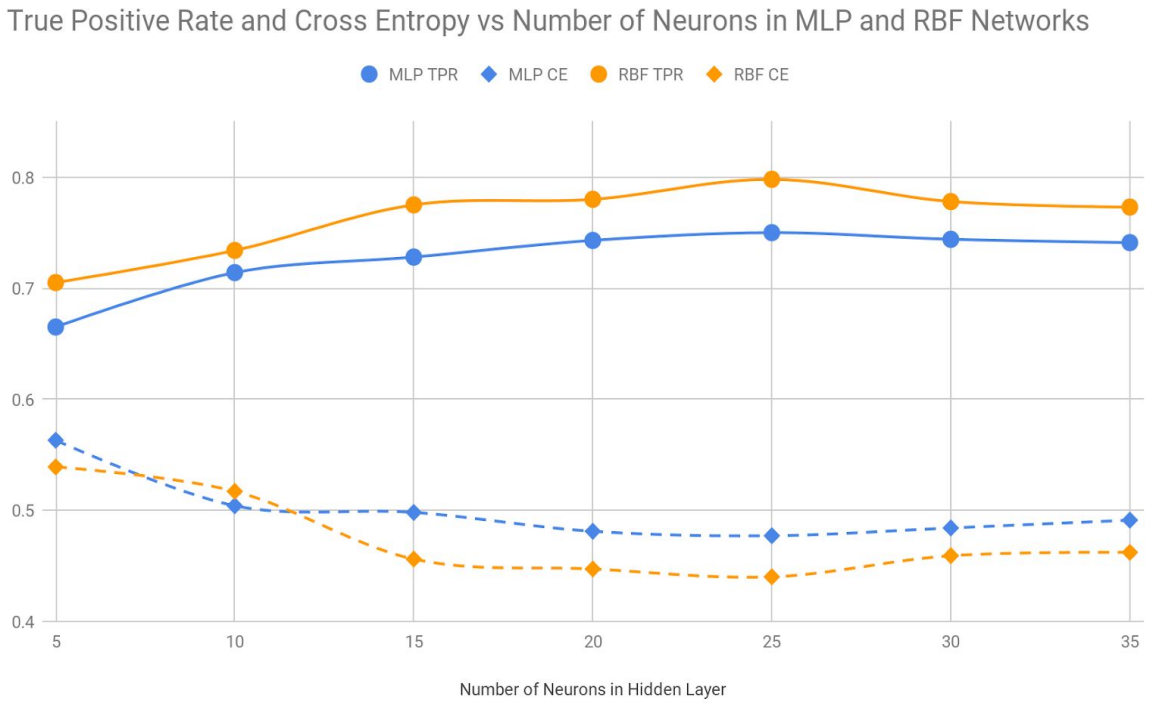
- Maximum number of epochs to train: 1000
- Performance goal: 0
- Maximum time to train: inf (i.e. algorithm is not bound by time)
- Minimum performance gradient: 1e-6
- Maximum validation failures: 6
- Sigma, change in weight for second derivative approximation: 5e-5
- Lambda, parameter for regulating the indefiniteness of the Hessian: 5e-7

2.3 Final Architecture

To decide which architecture to use for the final model, a number of different structures were tested for MLP and RBF networks - structure being the number of layers and neurons. They were evaluated on the basis of their true positive rate (TPR) and cross entropy error (CE) on the test dataset (see 3.1 for details).

Firstly MLPs were tested with 1 hidden layer and 5, 10, 15, 20, 25, 30 and 35 neurons. Then MLPs with 2 hidden layers were tested with 5, 10, 15, 20, 25, 30 neurons in each layer (summing to 36 distinct 2-layer tests). The RBF network was also tested with 5, 10, 15, 20, 25, 30 and 35 neurons.

Below is a graph of the TPR and CE against the number of neurons for both the RBF and single-hidden-layer MLP networks. The higher the TPR, the more accurate the model (in general); the lower the CE, the more accurate the model (in general). It shows that the RBFs outperforms most of the MLPs according to these metrics.



Increasing the number of layers for the MLP networks marginally increased the TPR and reduced the CE, but not enough to improve on the values of the RBF implementations. The best outcome for a multi-hidden-layered MLP was a TPR of 76.0% and a CE of 0.467; this had two layers with 25 and 30 neurons respectively.

Therefore the final architecture will be that of an RBF. The RBF architecture consistently outperformed an equivalent MLP, including those with different numbers of layers and neurons.

The RBF (and single-hidden-layer MLP) performed best with 25 neurons. Networks with less neurons were unable to fit the data as well. Networks with more than 25 neurons produced marginally worse results, this is likely to be due to overfitting of the training data.

3 Results and Evaluation

3.1 Comparisons

The final architecture was decided after running several different networks on the test dataset. The network which produced the highest TPR and lowest CE was taken forward. These metrics indicate the accuracy of the model.

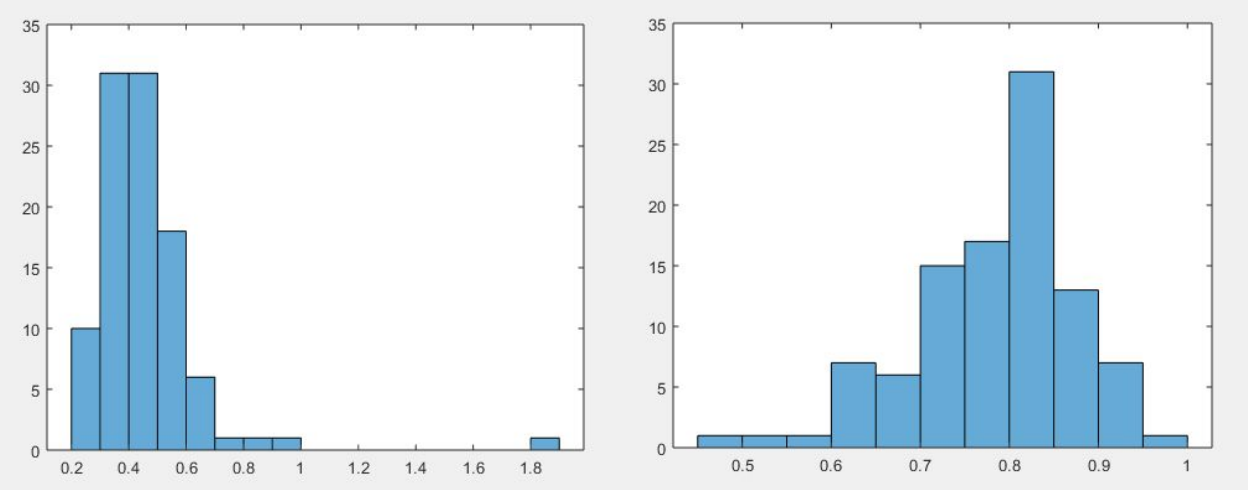
The TPR gives the rate at which actual positives are correctly classified by the model; i.e. the rate at which patients with breast cancer are diagnosed as such. Therefore the higher this rate, the higher the proportion of correct diagnoses. The true negative rate (TNR) should also be considered, ensuring patients without the illness will not be classified as such and needlessly treated. The TNR was analysed in the final architecture.

The CE value is calculated from a cross entropy loss function. CE differs to Mean Squared Error (MSE) by only calculating error from the predicted class; MSE calculations include error values from all classes in the model [11]. For a pattern recognition problem such as this, CE is more appropriate as an evaluation metric because of that reason - the concern is not with classes that were not predicted.

Testing each network repeatedly produced a range of results for the CE and TPR. Artificial neural networks will always have a random element to them due to the random initialisation of their weights; this can be solved by assigning a fixed random number generator seed, or testing each network repeatedly and taking average values. This problem can be particularly apparent when there is data sparsity, due to the small number of test runs; cross validation testing can also help in this regard. The networks were each tested 100 times; the TPR and CE values discussed are the means from these 100 iterations. These means allowed a more reliable evaluation and comparison of the different network architectures.

3.2 Synopsis

There was a large amount of variation in the results, as shown by the two histograms below: the left histogram showing the frequencies of CE values, the right histogram showing the frequencies of TPR values, each out of the 100 training runs. Each graph somewhat resembles a normal distribution however, giving a clear mean for both.



The final network produced the confusion matrix below. It shows that the network correctly diagnoses more than it correctly classifies patients as not having the illness, but both rates are quite high.

	True	False
Positive	86.4%	13.6%
Negative	80.0%	20.0%

The CE value of the final network was 0.459.

3.3 Conclusions

The results from the artificial neural network demonstrate that there is potential for blood sample measurements to be predictive of the presence of breast cancer. The TPR and TNR are significantly high and suggest *some* predictive power of the input parameters; however, in the context of medical diagnoses, a false negative rate of 20.0% is worrying - this means that a fifth of the time, the presence of breast cancer will be missed. The CE value of 0.459 is moderate, this also suggests the network has *some* predictive capability but is not particularly strong.

The problem should be investigated further. A larger dataset may enable more conclusive and reliable results to be produced; this would allow the existing network to have more data to train and test on, but also allow the possibility for more complex artificial neural networks to be used - such as a CNN, which requires a very large training dataset.

4 Further Application

4.1 Mammograms

Artificial neural networks (ANNs) are often useful for medical diagnoses [3,12], like this breast cancer classification. This concept can be taken further by classifying samples into one of three or more classes, becoming a multiclass classification problem. Mammograms provide X-ray images of a breast, allowing the identification of lesions thus possible diagnoses of health problems. Assuming access to a large number of mammograms with appropriate class labels, a multiclass ANN could class mammograms into stages of breast cancer - which describes the size and spread of the cancer, giving severity on a scale of 0-4 [13]. The main use of mammograms is to screen for breast cancer, but have been shown to provide insight into illnesses separate from breast cancer, such as Cardiovascular Disease [3]. Other possible developments could work to classify mammogram measurements into different types of illness.

Given that mammograms are x-ray *images,* a CNN would be a suitable ANN architecture to use for this classification problem. CNNs are suited to work with large complex datasets [4].

The input data would need to be preprocessed. A method for entering images into the CNN is to convert them from an M by N matrix into an MN by 1 vector, to enable analysis of each pixel as an independent feature [14]. Any input data should be normalised to have each pixel intensity value between 0 and 1. The output data can be represented categorically or as a one-hot encoding. The advantage of a one-hot encoding is that each class has its own neuron to with its own separate classification probability.

If not provided, the mammogram dataset will need to be manually separated into three independent datasets; training, validation and test. CNNs need lots of data to train on [4], so the training data should hold plenty of records.

Image convolution requires a set of *hyperparameters* [4]; number of kernels to convolve, the dimensions of each kernel, the stride of each kernel, padding of the input image to avoid a decrease in image size, pooling size - in addition to the standard neural network parameters such as number of layers and number of neurons.

To evaluate a CNN, with a one-hot output encoding, considering the cross-entropy loss is more important than a mean squared error loss [4]; the lower the cross-entropy loss, the higher the model's confidence in the correct class - without influence from probability of other classes [4]. A CNN with a categorical output can be evaluated with the mean squared error loss. Any pattern recognition network can be evaluated by analysing a confusion matrix, and calculating the true positive rate of the model on the test data. Given the severity of breast cancer, the network must be as accurate as possible, with several tests on a large test dataset. Cross validation will allow the complete dataset to be used (iteratively) as test data.

Provided the network is evaluated as accurate, by considering the above values, any positive outputs should be investigated further. Negative outputs with a probabilistic value close to the decision boundary between positive and negative could also be investigated further, to ensure false negative results are caught and acted upon.

5 References

[1] D. R. Matthews et al., "Homeostasis model assessment: insulin resistance and β-cell function from fasting plasma glucose and insulin concentration in man," *Diabetologia, vol. 28*, pp. 412-419, July 1985. [Online]. Available: https://link.springer.com/content/pdf/10.1007/BF00280883.pdf. [Accessed: 19 March 2019].

[2] D. Cireşan et al., "Multi-column Deep Neural Networks for Image Classification," *Technical Report No. IDSIA-04-12*, February 2012. [Online]. Available: https://arxiv.org/pdf/1202.2/745.pdf. [Accessed: 19 March 2019].

[3] J. Wang et al., "Detecting Cardiovascular Disease from Mammograms with Deep Learning," *IEEE Transaction on Medical Imaging*, vol. 36, May 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5522710/. [Accessed 19 March 2019].

[4] S. O'Keefe, "Introduction to Neural Networks, Convolutional Networks," University of York. [Online]. Available: https://vle.york.ac.uk/bbcswebdav/pid-2993960-dt-content-rid-7635834_2/xid-7635834_2. [Accessed: 19 March 2019].

[5] D. Gil and M. Johnsson, "Supervised SOM Based Architecture versus Multilayer Perceptron and RBF Networks," April 2010. [Online]. Available: http://www.ep.liu.se/ecp/048/005/ecp1048005.pdf. [Accessed: 19 March 2019].

[6] S. B. Wankhede, "Analytical Study of Neural Network Techniques: SOM, MLP and Classifier-A Survey," *IOSR Journal of Computer Engineering*, vol. 16, ver. 7, pp. 86-92, June 2014. [Online]. Available: http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue3/Version-7/N016378692.pdf. [Accessed: 19 March 2019].

[7] S. O'Keefe, "Introduction to Neural Networks, Solving more complex problems," University of York. [Online]. Available:

https://vle.york.ac.uk/bbcswebdav/pid-2993906-dt-content-rid-7635801_2/xid-7635801_2. [Accessed: 19 March 2019].

[8] S. O'Keefe, "Introduction to Neural Networks, Radial Basis Function Networks," University of York. [Online]. Available: https://vle.york.ac.uk/bbcswebdav/pid-2993933-dt-content-rid-7635819_2/xid-7635819_2. [Accessed: 19 March 2019].

[9] MathWorks. *Choose a Multilayer Neural Network Training Function*, uk.mathworks.com. [Online]. Available: https://uk.mathworks.com/help/deeplearning/ug/choose-a-multilayer-neural-network-training-function.html [Accessed: 19 March 2019].

[10] MathWorks. *trainscg*, uk.mathworks.com. [Online]. Available: https://uk.mathworks.com/help/deeplearning/ref/trainscg.html [Accessed: 19 March 2019].

[11] J. McCaffrey (2014, Apr. 22). *Neural Network Cross Entropy Error*, Visual Studio Magazine [Online]. Available: https://visualstudiomagazine.com/articles/2014/04/01/neural-network-cross-entropy-error.aspx. [Accessed: 19 March 2019].

[12] F. Amato et al., "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine*, vol. 11, pp. 47-58, December 2013. [Online]. Available: https://www.researchgate.net/profile/Eladia_Pena-Mendez/publication/250310836_Artificial_neural_networks_in_medical_diagnosis/links/5698d76608aea2d743771eef/Artificial-neural-networks-in-medical-diagnosis.pdf. [Accessed: 19 March 2019].

[13] NHS. *Diagnosis, Breast cancer in women*, www.nhs.uk. [Online]. Available: https://www.nhs.uk/conditions/breast-cancer/diagnosis/. [Accessed: 19 March 2019].

[14] S. O'Keefe, "Introduction to Neural Networks, Practical 8, Convolutional Neural Networks," University of York. [Online]. Available: https://vle.york.ac.uk/bbcswebdav/pid-2993963-dt-content-rid-7635839_2/xid-7635839_2. [Accessed: 19 March 2019].