Classification of Radiation Therapy Response in the Brain using Neural Networks

Berk Norman and Jack Karisch

April 24, 2016

Contents

1 Problem Statement

1.1 Project Purpose

Stereotactic radiosurgery (SRS) is a form of radiation therapy that involves highly focused radiation beams targeting tumors in the brain. During treatment a patient will lie on a table, which slides into a machine that delivers radiation. A robotic arm controlled by a computer will then focus radiation to the area(s) needing treatment. The procedure is completely painless and does not require putting the patient to sleep. SRS presents an important noninvasive cancer treatment that minimizes the impact on the patient's quality of life. In many cases, patients are able to maintain normal daily activities with minimal signs of disease. Furthermore, SRS has demonstrated efficacy comparable to surgery and has achieved local control [1] rates of over 90% [2]. However, a common complication of SRS is radiation necrosis [2] at or near the treatment site. Radiation necrosis (which we shall refer to as necrosis from now on) is a common delayed effect of SRS that occurs in approximately 10-20% of treated tumors [2].

To further complicate things, there is currently a lack of non-invasive techniques that can differentiate necrosis from recurrence. A conventional magnetic resonance image (MRI) will typically demonstrate similar features of both recurrence and necrosis. In particular, imaging shows a necrotic core surrounded by a contrast enhancing region of edema [3]. Moreover, the lesion is typically at or adjacent to the original site of the tumor. Surgery is perhaps the only sure way to distinguish between recurrence and necrosis, but in some cases patients may be too weak to undergo such a procedure or the region of interest is inoperable. The goal of this project is to develop a machine learning classifier that can accurately differentiate necrosis from recurrence.

2 Data Description

Overview of the data description can be viewed in Table 1.

Due to the large number of observations in the data, training data will consist of 12,000 randomly sampled observations while testing will consist of 3,000 different randomly sampled voxels, These numbers were chosen based on past models used for this type of data.

3 Objective

This project is an extension of our CAAM senior design research project, where we use machine learning algorithms such as radial support vector machines (SVMs) and random forests for classification. Initially, these models were

[1] The arrest of cancer growth at the site of origin

[2] Refers to degradation of brain tissue following intracranial or regional radiation

[3] Watery fluid collecting in the cavities or tissues of the body

Table 1: Data Description Parameters

Origin of Data	Data was supplied by Dr. David Fuentes, from M.D. Anderson, in the form of 19 patients, 9 with necrotic tumors and 10 with recurrent tumors. These raw MRI scans were turned into usable data by only extracting the voxels from the specified tumor locations
Data meaning	Each data observation is a 4D vector of tumor voxel intensity values (each dimension corresponding to 1 of 4 scan types: T1, T2, TC, FLAIR). Each observation is labeled to 1 of the 2 tumor classes that the respective voxel came from.
# data	732128 observations, all labeled
# Input dimensions	4 (T1 intensity, T2, intensity, TC intensity, and FLAIR intensity)
# Classes	2
Input features numerical ranges	T1 Intensity: [-2.1061, 5.1670] T2 Intensity: [-1.7788, 5.1670] TC Intensity: [-3.8876, 10.002] FLAIR intensity: [-1.8086, 5.3017
Output classes	Necrosis: delayed effect of SRS that is the result of inflammation in treated area. Reccurent: residual progressing tumor.
# samples per class	Necrosis: 211114 Recurrent: 521014
Encoding of output features	(1, 0) = Necrosis (0, 1) = Recurrent

trained on a random sample of 9000 voxels. However, this resulted in a high false positive rate of recurrence since the initial data (and the subsampled 9000) had many more recurrent voxels than necrotic ones (see Table 2 for our initial results using Random Forests, which was the best classifier for this approach). Our naive solution to this problem was randomly sampling 4500 necrotic voxels and 4500 recurrent voxels for the training data. This resulted in much better results with an overall accuracy of 89.37%, with 12.99% misclassification of necrotic voxels and 9.68% misclassfication of recurrent voxels.

Our objective of this research within the scope of NML 502 is to see if Neural Network techniques can result in comparable or better classification rates than SVMs and random forests without tampering with the composition of the training data. Additionally, we will use principal component analysis (PCA) and self-organizing maps (SOMs) to better understand variation between tumor

	True Values	
Predicted	Necrosis	Recurrent
Necrosis	615	68
Recurrent	247	2070
Error	**28.7%**	**3.2%**

Table 2: Tree Bootstrapping Confusion Matrix

types since our previous models did not give good visualizations of the data that provided interpretable results.

4 Technical Approach

4.1 PCA

Before we began model building, we conducted PCA on the data to get its 4 principal components. 5000 observations from these principal components were randomly sampled and the first two principal components were plotted against each other (which explained 75.3% of variation within the data). Results from this can be viewed in Figure 1.

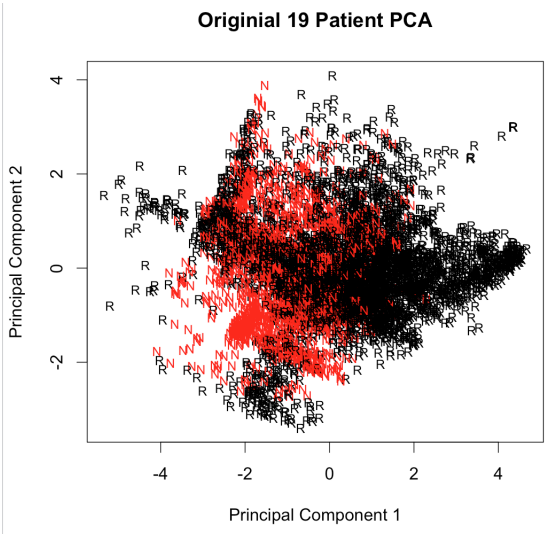


Figure 1: PCA for a random sample of 5000 voxels from the 19 patients' tumor voxels. Red "N"s are necrosis and black "R"s are recurrent.

The PCA graphic does show some separation between the necrotic and recurrent voxels which is a promising feature for differentiating between two classes for modeling. Another interesting feature of this PCA is that there appears to

almost be two clusters of necrotic voxels. This trend shows up again when using SOMs and will be addressed later.

4.2 Backpropagation

Table 3: Parameters of Backpropagation

Learning parameters	
Number of hidden PEs	4, 8, 10, 13, 16
Epoch size	1, 7000, 14000
Initial weights	drawn from U[-0.1, 0.1]
Learning rate (α) for batch	0.00001, 0.00005, 0.0001
Learning rate (α) for on-line	0.005, 0.0075, 0.01, 0.03
Maximum epochs (batch)	3000
Maximum epochs (on-line)	3000000
Stopping criteria	Error (Err_{stop}) ≤ 0.1
Error measure (Err_{stop})	Misclassification rate of network
Momentum	0.2, 0.25, 0.3, 0.4, 0.5, 0.6
Threshold function	tanh, slope parameter = 1
Input / output data, representation, scaling	
Data samples dimensions (N_{tr})	2 parameters by 21000 observations
Scaling	Input data scaled to [0,1]

The backpropagation network used had a single hidden layer. The different ranges of parameters attempted can be found in Table 3. The entire dataset was sampled randomly to give first 9000, then 15000 and 21000 observations; this means that about 70% of each sample was of the recurrent type, because of the distribution of the types in the overall dataset. The smaller sample sizes tended to generalize very poorly. The best results were obtained using 21000 total observations, with 14000 of these used for training and the remaining 7000 used for testing. We first conducted batch learning with the batch size set to 14000, with the maximum number of epochs set to 3000. We then moved on to a batch size of 7000 (half of the training data), and finally tried on-line learning. The error and stopping criterion were the misclassification rate of the network. The stopping criteria and learn rate were scaled depending on the batch size (both smaller when batch size was larger).

4.3 SOM

Table 4: Parameters of Kohonen SOM

Learning parameters	
Initial weights	drawn from U[-0.1, 0.1]
Learning rate (α)	0.1
Learning rate decay function	$\alpha = 0.99997 \times \alpha$ every learning step
Maximum number of iterations	120000
Epoch size	4000
Grid size	25×25
Latice distance metric	Manhattan "city block" distance
Initial neighborhood radius	6 (this will be explained more below table)
Neighborhood decay function	round(n(1-t/maxit)), where t is current iteration NOT learning step and maxit is the maximum number of iterations (this will be explained more below table)
Stopping criteria	error (Err_{stop}) $\leq$ 0.0001 OR learn count (t) $>$ $480,000$
Error measure (Err_{stop})	Root mean squared difference (RMSD) $= \sqrt{\frac{1}{\|W\|}\sum_{k=1}^{\|W\|}(W_{old} - W_{new})^2}$, where W_{old} is the previous iterations weight matrix and W_{new} is the current iterations weight matrix
Input / output data, representation, scaling	
Data samples dimensions (N_{tr})	2 parameters by 20000 observations
Testing samples dimensions (N_{ts})	2 parameters by 3000 observations
Scaling	Input data scaled to [0, 1]

The set up parameters for the constructed SOM can be viewed in Table 4. Smaller grid sizes were first used in this SOM, however since the training data was so large, a larger SOM of 25×25 seemed more appropriate in order to capture more of the variation within the data (this will be explored more in the results section). The stopping criteria error measure chosen for learning is essentially looking at the stabilization of the weights. Once the weights have stopped moving around and updating, their difference between iterations will become minimal and therefore the algorithm will stop. The signatures of each prototype's weight vector were also examined to confirm that the SOM did in

fact converge and has not resulted in a twisted mapping.

Initially, this SOM was going to be used for SOM Hybrid Supervised Learning. However, this gave poor prediction rates and had convergence issues, so, instead each neuron of the SOM was assigned a class based on what class majority of the training data was mapped to it (dead neurons were also assigned). Then, testing data was identified by the class of the neuron it had the minimum distance to. If a testing data point was mapped to a dead neuron in the trained SOM, the majority class of the surrounding non-dead neurons (determined by a Manhattan distance of radius 1) was assigned. In theory, this method is similar to hybrid SOM learning in the winner take all case (see Figure 2).

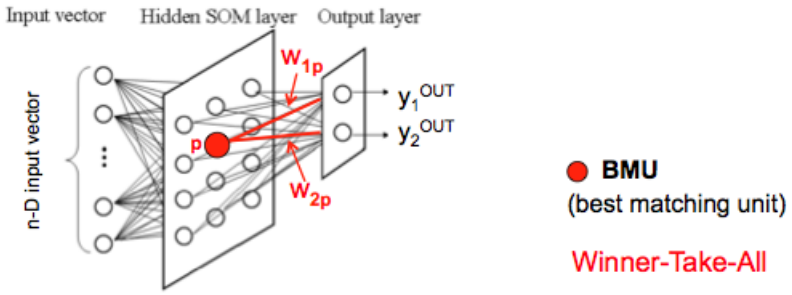


Figure 2: Winner take all with SOM as hidden layer visualization [1].

For this winner take all method, $y_1^{out} = w_{1p} \times 1$ and $y_2^{out} = w_{2p} \times 1$, where p is the location of the winning neuron. Since the outputs for this network only take on values of 0 and 1 for the two output neurons, theoretically if the weights from the SOM hidden layer to the output layer (W) learn correctly, W will just be a series of zeros and ones mapping to the respective outputs of 1 from the SOM to either 1 or 0. This method would work the same as just directly assigning each SOM prototype to a class based on the majority of training data that was mapped to it.

5 Results

5.1 Backpropogation

The optimal configuration for batch and on-line learning can be seen in Table 5. Both techniques were slow to converge due to the large sample used. However, as can be seen in Table 6, on-line learning reached a much lower error in less time than batch. On-line configurations also tended to generalize better. A typical poorly-generalized batch network plot of error over time can be seen in Figure 3. Adding too many hidden PEs, in addition to slowing down training considerably, introduced large swings in error over time, as did momentum values that went significantly over 0.5. Figure 4 shows error by learn step for

Table 5: Optimal Batch and On-line Configurations

Parameter	**Best batch configuration**	**Best on-line configuration**
Number of hidden PEs	10	10
Epoch size	14000	1
Learning rate	0.0001	0.01
Maximum number of epochs	3000	3000000
Stopping criterion	$(Err_{stop}) \leq 0.1$	$(Err_{stop}) \leq 0.1$
Momentum	0.3	0.3

Table 6: Best Results (Average of Three Runs)

	Training error	**Testing error**	**Runtime (seconds)**
Batch	22.5%	22.6%	1340
On-line	14.8%	14.9%	1141

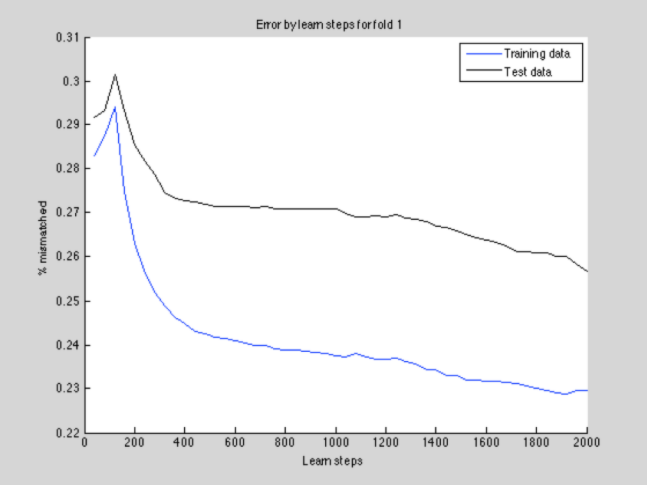


Figure 3: Error by learn step of poorly generalized batch configuration.

	True Values	
Predicted	Necrosis	Recurrent
Necrosis	1305	244
Recurrent	783	4668
Error	**37.5%**	**5.0%**

Table 7: Testing data confusion matrix of best configuration.

on-line learning with 16 hidden PEs and a momentum value of 0.5. The best

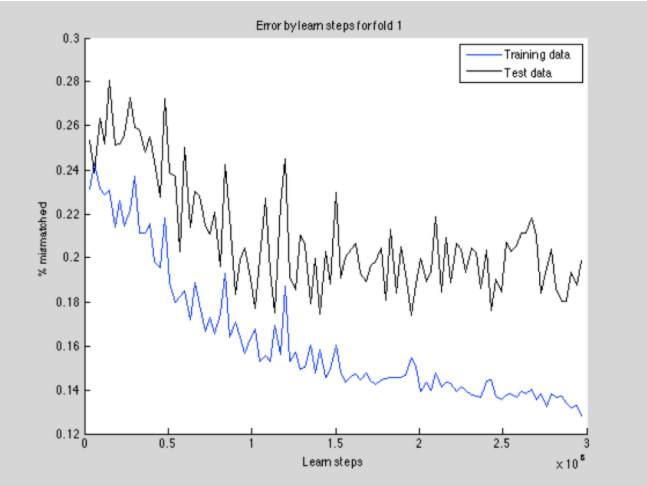


Figure 4: Error by learn step of network with 16 hidden PEs and momentum value of 0.5.

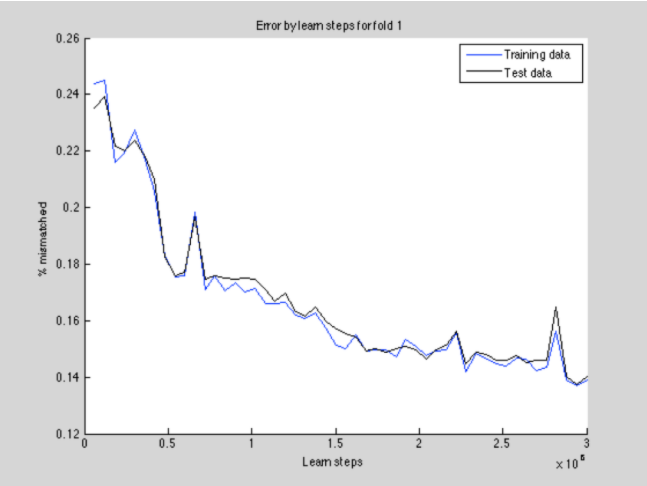
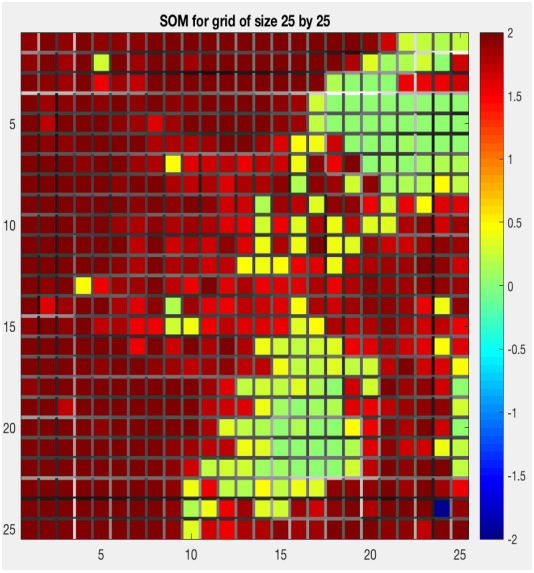


Figure 5: Error by learn step for best overall configuration.

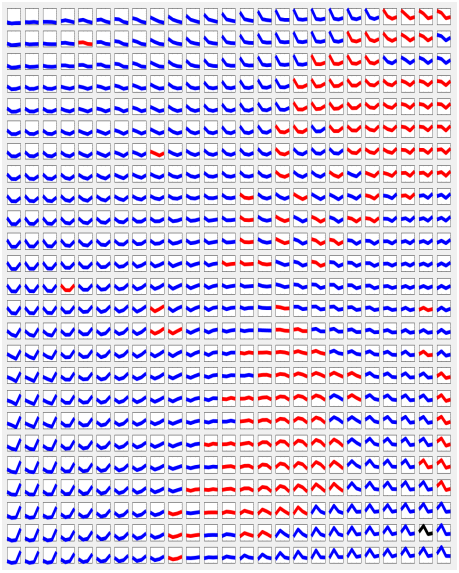
overall configuration yielded the plot in Figure 5 for error over time and the confusion matrix found in Table 7, for the testing data. Although the total error of the best configuration was 14.9%, the vast majority of that error came from the necrosis category, meaning that this method has a very high rate of false positives, at 37.5%. The false positive rate for the best batch configuration was even higher, at 43.2%.

5.2 SOM

The SOM described above took 420,000 learning steps to converge. In order to visualize the learned SOM, a modified U-matrix was constructed with the data mapping density overlayed on the modified U-matrix, which can be viewed in Figure 6.a. For the modified U-matrix, the white borders represent "high walls", indicating a larger distance between the prototype weights which create a type of separation between groups of prototypes. Dark borders indicate a small distance between prototype weights. For the data mapping density, red indicates recurrent data being the majority that was mapped to that particular neuron. Dark red (value=2) means almost 100% of the data mapped to that neuron was recurrent while light red (1.5) means just over 50% of the data mapped to that neuron was recurrent. For necrosis, light green (0) means almost 100% of the data was necrotic while yellow (0.5) means just over 50% of the data mapped to that neuron was necrosis. Finally, a blue neuron indicates a dead-weight that none of the training data was mapped to. The signatures of the SOM prototypes were also examined by plotting the weight vectors in each prototype in order confirm that the map is not twisted and also examine patterns within separate groups of the same class (Figure 6.b).



(a) Density map with modified U-matrix



(b) Neuron data count bar chart.

Figure 6: Gaussian Kohonen SOM results

In the density map modified U-matrix, there are distinct clusters of recurrent-necrosis prototypes which is promising for using a hybrid SOM for supervised

classifying. However an interesting feature of this graph is there are two distinct clusters of necrosis prototypes (also present in PCA). This feature will be addressed in the Conclusions/Next Steps section. Looking at Figure 6.b, these two clusters also have distinct signatures in the weight vector prototype mapping. This figure also shows that the SOM did converge correctly since the weight vectors are not randomly assorted and follow patterns in accordance with their respective class mappings. The fact that there is a few different weight vector groups for the recurrent tumor voxels is not so concerning since recurrent tumors are heterogeneous in nature; usually containing some necrosis within them.

When this method was applied to the training and testing data, their missclassificaiton errors were 13.28% and 14.90%, which were comparable to the previously used machine learning algorithms. However, examining the confusion matrices in Table 8, a high false positive rate is still present for missclassifying necrosis as recurrent, a problem we had hoped to avoid when using truly randomly sampled training data.

Table 8: Confusion matrices for training and test data of SOM classifying

(a) Training data confusion matrix

	True Values	
Predicted	Necrosis	Recurrent
Necrosis	3973	1769
Recurrent	888	13370
Error	**30.81%**	**6.23%**

(b) Testing data confusion matrix

	True Values	
Predicted	Necrosis	Recurrent
Necrosis	557	286
Recurrent	161	1996
Error	**33.93%**	**7.46%**

6 Conclusions/Next Steps

Between backpropagation and SOM classification, the SOM technique performed slightly better in terms of classification accuracy and runtime. However, both methods had very high rates of false positives which greatly weakens their usefulness. A false positive leads to unnecessary surgery, which these classification methods are trying to minimize. They also slightly underperformed the other machine learning techniques (SVM and random forest, which gave classification accuracies of 88.53% and 89.4% respectively). However, the results of these tests do indicate that this data can be classified to a fair level of accuracy, with a low false negative rate, despite the 70:30 distribution of the data.

Possible ways to improve accuracy could include taking larger samples and sampling the data evenly between necrotic and recurrent types, although there are tradeoffs to both of these solutions, namely increased runtime and an unrealistic training distribution, respectively. Other techniques that could improve results include growing or pruning the weight matrix of the ANN and adding one or more additional hidden layers. Finally, another technique to be examined for this project and this data in general is how/why there may be variation within necrosis tumors. In two different methods (PCA and SOM), it was observed

that necrosis voxels separated into two clusters. This could be an issue with the dataset or it could be an overlooked anatomical feature. Either way, it could be useful to include this into modeling by maybe having three classification groups: recurrence, necrosis 1, and necrosis 2. This could potentially improve classification accuracy.

References

[1] Dr. Erzèbet Merènyi. L11.11 SOM Hybrid Classification. 2015.

[2] Paul Brown, David Fuentes, Jeffrey Weinberg. Recurrence or Necrosis? 2015.