Identification of Dry Patches of Land using Satellite, Topographical and Weather Data, Genetic Feature Selection, Bidirectional Networks

[1]Yasar Adeel Ansari
[1] Research School of Computer Science, Australian National University

Abstract. The problem of land degradation has progressively become more severe in Australia. Apart from making land unsuitable for agriculture, it leads to increased contamination of rivers and gullies. Which is detrimental to the eco system of these water bodies. One of the main contributors to land degradation is the erosion of soil. Although there are many types of erosion, this paper aims to tackle the problem of water erosion. Water erosion is the process of removal of topsoil due to movement of water. Water from rivers, rains, slopes etc. together lead to water erosion. Water erosion is especially severe in dry regions with high slopes. Identification of these regions is necessary to devise preventive strategies against water erosion. A variant of an existing technique using Bi-directional propagation of weights was used for classification. Bi-directional Neural Networks are mainly used in learning weights that can not only predict output patterns given input patterns. But also generate likely input patterns given output patterns. This leads to better generalisation. In order to reduce the dimensionality of the input space, genetic algorithms were used to select the set of features that favour bidirectional propagation of weights.

Keywords: Bidirectional Networks, Genetic Algorithms, Feature Selection.

1 Introduction

The data used to conduct the experiments is a subset of the one used by Milne, Gedeon and Skidmore [1]. The dataset corresponds to the Nullica State Forest region in the south cost of New South Wales, as mentioned by Milne, Gedeon and Skidmore (199. Each sample in the dataset corresponds to a pixel of size 30m x 30m. The data collected for each pixel includes geographical data like slope, aspect and elevation. Weather data such as temperature and rainfall. Landsat TM bands data from Band 1 to 7. Detailed explanation of these features can be found in Section 2.

The Bidirectional Neural Network methodology followed in this paper is heavily inspired from Nejad and Gedeon [2]. As shown by the experimental results in Nejad and Gedeon (1995, Section 4). Bidirectional Neural Networks can be used to train weights that can not only predict output patterns given input patterns. But the weights can also generate input patterns using output patterns. This inverse functionality of the weight matrices leads to better generalisation of the input space (Nejad and Gedeon, 1995, Section 5).

This paper proposes and makes use of two bidirectional neural network models. They will be referred to as simple-BDNN and robust-BDNN here onwards in this paper. The simple-BDNN was used in tandem with genetic algorithms to find features that would favour bidirectional propagation of weights. i.e., the performance of simple-BDNN acted as the fitness function that would be used to select suitable features. The suitable features were then used to train the robust-BDNN model.

The analysis of the ROC curve on class probabilities given by robust-BDNN shows that the trained robust-BDNN can differentiate between dry pixels with the ones that are not dry with a surprisingly high accuracy. This paper shows that bidirectional neural networks could be relied upon for similar tasks.

2 Data

There are 190 samples, each corresponding to a pixel of size 30m x 30m. For each sample, continuous and categorical features were available.

2.1 Preprocessing

The strategy used to encode them as inputs to a neural network was as follows.

i. **Altitude.** Elevation of each pixel. Normalised to be between 0 to 1.

ii. **Slope.** Slope of the pixel. Normalised to be between 0 to 1.

iii. **Temperature.** Temperature at the geographical location of the pixel. Normalised be between 0 to 1.

iv. **Rainfall.** Average rainfall at each pixel. Normalised to be between 0 to 1.

v. **Aspect.** Represents the downslope direction of the maximum rate of change in altitude of a pixel with respect to its neighbouring pixels. It essentially tells in which direction water would likely flow in after falling perpendicularly on a pixel. The original encoding of this feature was categorical. Represented by a single unit that took one of 8 values. This did not reflect the angular nature of aspect data. To feed the true essence of the aspect value to a neural network, it was represented using 4 units. Where each unit corresponded to one of the 4 parts of the compass. Refer Table 1.

Table 1. E1, E2, E3, E4 refer to each unit in the new encoding

Old Encoding	**Compass**	**E1**	**E2**	**E3**	**E4**
0	Flat	0	0	0	0
10	North	1	0.5	0	0.5
20	Northeast	1	1	0	0
30	East	0.5	1	0.5	0
40	Southeast	0	1	1	0
50	South	0	0.5	1	0.5
60	Southwest	0	0	1	1
70	West	0.5	0	0.5	1
80	Northwest	1	0	0	1

vi. **Topography.** This was initially a relative categorical value. Was normalised whilst preserving its relative categorical nature. Refer Table 2

Table 2 Note that both values 48, 64 are considered to be mid slopes

Topography	**Gully**	**Lower slope**	**Mid Slope**		**Upper Slope**	**Ridge**
Old Encoding	16	32	48	64	88	96
New Encoding	0.0	0.25	0.5	0.5	0.75	1.0

vii. **Landsat 7 Bands.** Refers to data collected from the Landsat 7 Satellite's sensors. All bands were normalised.

Table 3. Table information from USGS Website [3]

Band	**What does it sense?**
TM1	Distinguishing soil from vegetation and deciduous from coniferous vegetation
TM2	Emphasizes peak vegetation, which is useful for assessing plant vigor
TM3	Discriminates Vegetation Slopes
TM4	Emphasizes biomass content and shorelines
TM5	Discriminates moisture content of soil and vegetation
TM6	Thermal mapping and estimated soil moisture
TM7	Hydrothermally altered rocks associated with mineral deposits

After pre-processing the 13 features, the dataset consists of 20 columns. As explained in later sections, most features would be highly correlated. And some would not be useful for classification.

2.2. Train and Test Split

The dataset contained 190 samples. 20 percent was used for testing. Training and testing sets were created using stratified sampling [4]. Refer

Table 4. Train and Test sets distribution

Class	Train	Test
Dry	77	19
Not Dry	75	19

3. The simple-BDNN and feature selection.

3.1 simpleBDNN

As discussed earlier, the network is capable of predicting output patterns given input and vice versa.

3.1.1. Predicting Output Patterns

For a sample 'n', with an input pattern 'x', the output pattern is predicted using 2 fully connected linear layers. With the same activations as shown in figure1.

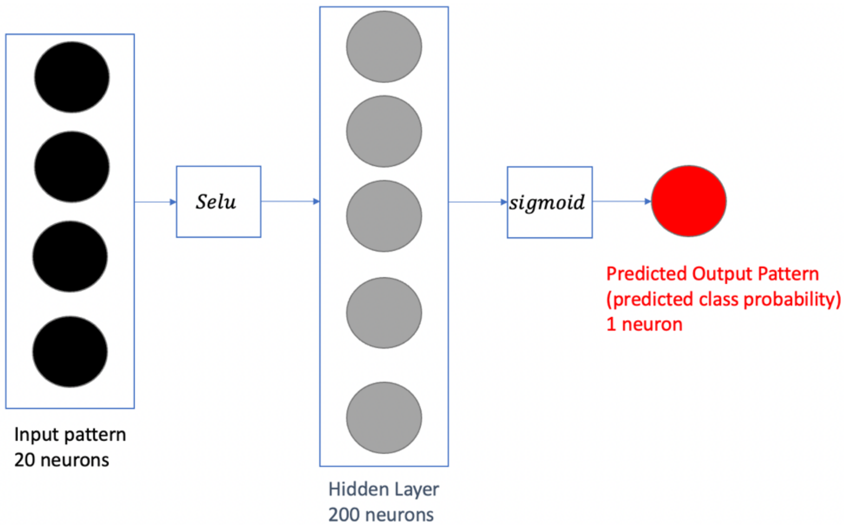

Figure 1 simple-BDNN predicting output pattern

3.1.2. Predicting Input Patterns

For the same sample 'n' mentioned in the previous section, with an output pattern 'y', the input pattern is predicted by reversing the linear layers. Essentially, the order of weights is reversed, and the weights are transposed as shown in figure 2. This way the same weights are used to predict the outputs from inputs and vice versa.

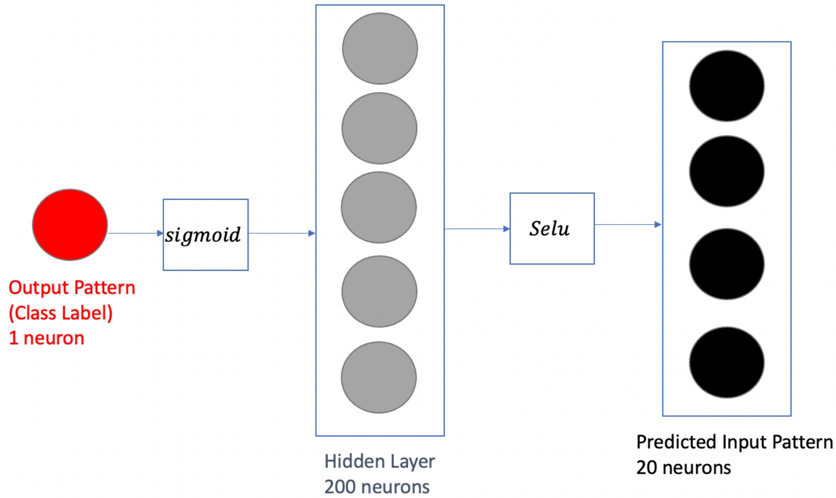

Figure 2. simple-BDNN predicting input pattern

3.1.3 Updating Weights

For each same sample 'n', let the error in the prediction of the output pattern be $\mathbf{E_F}$ and the error in the prediction of the input pattern be $\mathbf{E_B}$. Then we define total error i.e. $\mathbf{E_{Total}} = E_F + E_B$. The total error is used to update the weights with the intention of making them capable of predicting input patterns from output patterns and vice versa.

Note: The loss in E_F was calculated using binary cross entropy loss function. The loss in E_B was calculated using mean squared error.

3.2. Feature Selection

The simple-BDNN model was used for calculating fitness of each subset of features in the feature selection process. It is important to note that the simple-BDNN model is flawed, because it is not trained using data that has a one-to-one mapping between input and output patterns. This would lead to poor learning on bigger input spaces. Classification performance of simple-BDNN was chosen as a measure of fitness in the expectation of choosing subset of features that would not incur a heavy loss when predicting the output patterns from input patterns. And also reduce the input space, choosing a smaller subset of features.

Including all the TM bands (table 3), there are 14 features for each sample encoded using a 20-dimensional vector. There are 3 motivations to using simple-BDNN with genetic algorithms for feature selection

1. Reduce the feature size to make it easy for simple-BDNN and robust-BDNN to predict the input patterns given output patterns and vice versa.
2. Reduce the number of highly correlated features in a feature set.
3. Remove features that are not useful in prediction.

3.2.1. Fitness

The fitness of a feature set is calculated using the following methodology.

1. Divide training data into 5-fold cross validation sets using stratified sampling.
2. For each cross-validation set, train the simple-BDNN on the subset of features and calculate the area under ROC curve for predictions of output patterns in the validation set.
3. The maximum area under the ROC curve among the 5 would be the fitness for each feature set.

The Algorithm for feature selection is as described by Figure3.

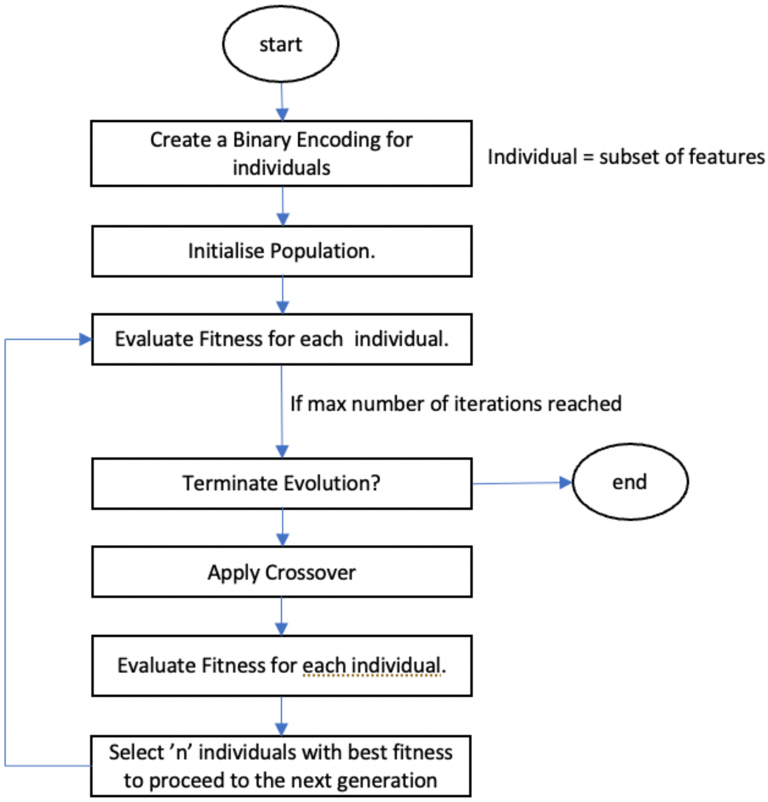

Figure 3 feature selection algorithm

3.3. Performance of selected features on validation and test sets.

Upon termination, the subset if selected features **'best-F'** was as follows.

1. Topography
2. Geology Descriptor
3. Temperature
4. TM3
5. TM6
6. TM7

Their fitness, or the fitness of simple-BDNN when trained using these features on 5-fold cross validation sets was 0.933. i.e., the maximum AUC score achieved among the scores for the 5-fold cross validation sets. The model corresponding to the highest AUC score as chosen.

The chosen model was chosen to predict the dry or not dry class of each pixel in the test set. The area under the ROC curve for test set was 0.83.

4. robust-BDNN

The simple-BDNN whilst predicting input patterns from output patterns, faces a near impossible task. The output pattern can be '1' or '0' indicating whether a pixel is dry or not. Inferring from this number, an 'n' dimensional input pattern is not reasonable. Furthermore, every input pattern is mapped to either '1' or '0' as its output pattern. It is therefore, important to make the mapping between input patterns and output patterns one-to-one.

4.1. Make the dataset invertible

To make the dataset invertible, the paper proposes adding a feature from input patterns to the output patterns. The feature chosen for this task should roughly fulfil the below conditions

1. Not be part of the selected features i.e. **best-F**.
2. Should have high variance. Such that addition of this feature leads to more unique input pattern to output pattern mappings.

The feature selected for this purpose was 'altitude'. But the data was still not invertible after this addition. Random noise was added to the 'altitude' feature. Thus, the data became invertible, and the new problem statement was

- Given a 13-dimensional input pattern space consisting of features in **best-F.** And a 2-dimensional output pattern space corresponding to the class label and altitude. Create a bidirectional network that predicts the input patterns given output patterns and vice versa.

4.2. robust-BDNN Architecture

The robust-BDNN model architecture is as follows

4.2.1. Predict Class Label and Altitude from 'best-F'

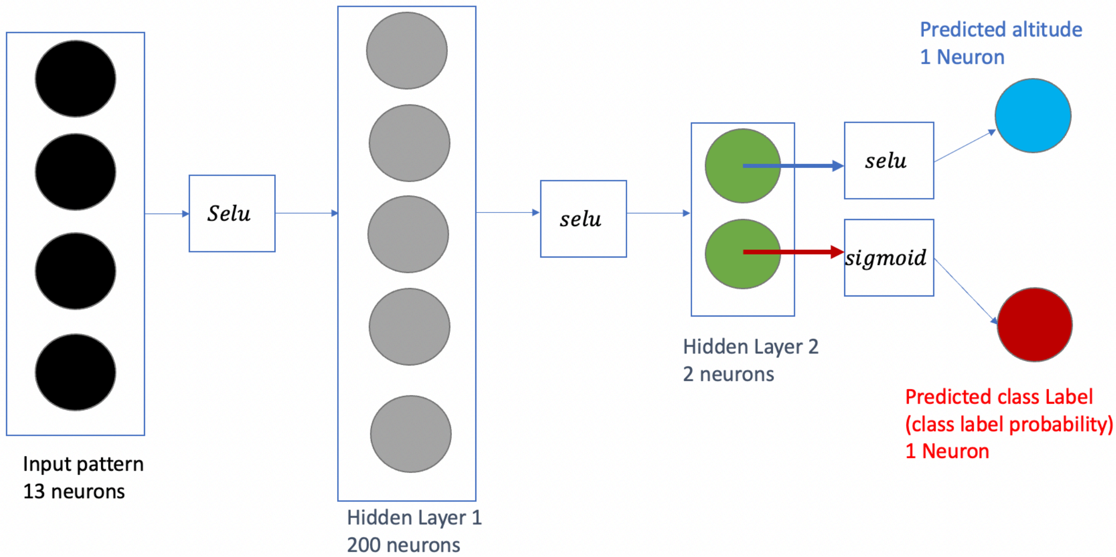

Figure 4. robust-BDNN output prediction

The total error in predicting the output patterns $\mathbf{E_F}$ is sum of

1. Error in predicted class label probability calculated using binary cross entropy loss.
2. Error in predicting the altitude calculated using mean squared loss.

4.2.2. Predict 'best-F' from class Label

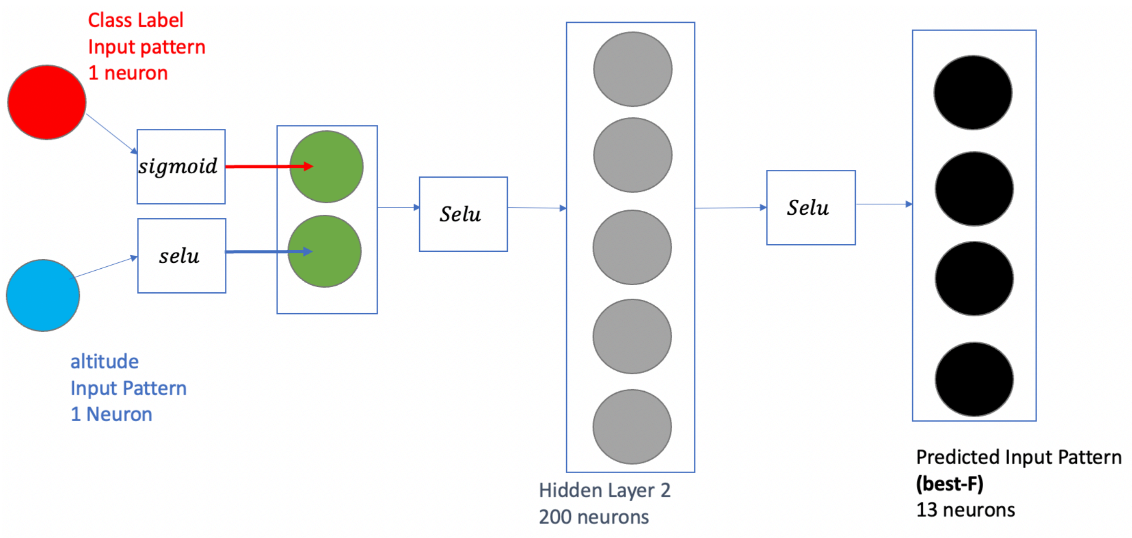

Figure 5. robust-BDNN backward output prediction

The error in predicting the input patterns $\mathbf{E_B}$ is calculated using mean squared error against the original input patterns.

4.2.3. Updating Weights

The total error for each sample be $\mathbf{E} = (E_B + E_F)$. Backpropagation is performed with the intention of minimising this total error.

4.3. Results

The robust-BDNN was trained on the training set using the stratified k-fold cross validation technique. 5-fold cross validation set were chosen. The model was tuned to give maximum validation accuracy in all the 5 validation models.

The model corresponding to the cross-validation set whose predictions of class label that gave the highest area under the ROC curve was chosen. The AUC for the best model on its respective validation set was 0.9377

The model gave impressing results of 0.93 AUC for the test set as well.

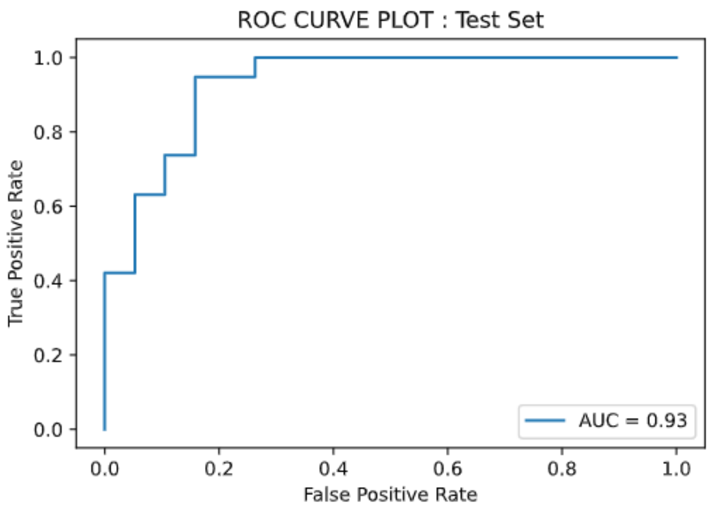

Figure 6. Classifier ROC Curve on Test Set

A balance between false positives and false negatives is also achieved using a threshold of 0.51.

Class	Dry	Not Dry
Predicted Dry	16	3
Predicted Not Dry	3	16

Figure 7 confusion matrix on test set with a classifier threshold of 0.51

Achieving a balance in false positives and false negatives without severely effecting the overall accuracy is crucial when devising strategies to combat erosion. The robust-BDNN model is able to provide accurate information on the dryness of pixels in the area.

5. Conclusion and Future Work

The paper has demonstrated the tremendous ability of bio inspired learning mechanisms like bidirectional neural networks and genetic algorithms in solving real world problems. Future work would include extending the bidirectional neural network methodology to deep neural networks. Genetic Algorithms could also be used in tandem. Further testing on different datasets would also be conducted.

References

1. Milne LK, Gedeon TD, and Skidmore AK. Classifying dry sclerophyll forest from augmented satelite data : Comparing neural network, decision tree & maximum likelihood. In Proc. 6th Australian Conference on Neural Networks, pages 160{163, 1995.
2. Nejad, A. F., and T. D. Gedeon. 'Bidirectional Neural Networks and Class Prototypes', Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 3/(1995), pp. 1322-1327 vol.3.
3. https://www.usgs.gov/faqs/what-are-best-landsat-spectral-bands-use-my-research?qt-news_science_products=0#qt-news_science_products
4. Singh R., Mangat N.S. (1996) Stratified Sampling. In: Elements of Survey Sampling. Kluwer Texts in the Mathematical Sciences (A Graduate-Level Book Series), vol 15. Springer, Dordrecht.
5. G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, CoRR, arXiv:1706.02515, 2017.
6. Kingma, D and Ba, J . (2015) *Adam: A method for Stochastic Optimization*. Available at: https://arxiv.org/pdf/1412.6980.pdf
7. Bradley, A. P. (1997). The Use o the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, 30(6), 1145–1159