Acoustics 2019
Sound Decisions: Moving forward with Acoustics

Exploring the classification of acoustic transients with machine learning

Nick Padfield

Defence Science and Technology Group, HMAS Stirling, Western Australia

ABSTRACT
Machine learning techniques are so numerous that it can be difficult for a machine learning novice to put them to use in their specific discipline unless they have third-party work to study. There is not a lot of previous work in the discipline of underwater acoustics, but the potential for benefits in classification and detection are clear. This paper describes how desktop tools have been put to work on a transient classification task using a few machine learning techniques. The paper shows how the most complex techniques may seem be the most intuitively appropriate, but can be overkill for this application. A much simpler detector-classifier is demonstrated and advice for fellow researchers in the discipline is provided.

1 INTRODUCTION

Machine learning (ML) has a lot to offer in the domain of underwater sonar. Detection and classification of signals is highly developed using classical techniques and is useful for environmental, commercial and defence applications. ML could be used in this area where data rates are very high, such as large numbers of sensors and high sample rates, or to apply novel techniques. Signal prediction is another ML technique that might be used in sonar, either to anticipate the behaviour of the sound emitter or to model it. Again, this technique would have a wide range of possible applications.

There is not a lot of literature about applying machine learning techniques to underwater sonar time-series data, but there are a great many papers about audio processing particularly in the speech processing domain. There are clearly lessons to be learned from that work for sonar processing. Purwin (Purwin et al., 2019), for example, describes the use of deep learning to audio applications while remaining agnostic to the particular domain.

For this paper, an exploration of basic machine learning techniques as applied to a simple sonar project is described in terms of how that technique might be used in the broader sonar domain and what the author learned from it. It is not a comprehensive state-of-the-art summary (see Purwin for that), it is more a report on the first steps taken by an ML novice and an encouragement to other novices to get started in applying ML in their particular domain.

2 APPROACH

A simple project was chosen as the starting point of this exploration. A project was presented last year at AAS (Du et al., 2018) that used a single hydrophone recording of snapping shrimp in shallow water. That paper describes a regression method using extracted features of the time-series data to successfully detect shrimp snaps with 94% accuracy. In this work, it was decided that the same data could be used for a classification exercise, where three classes were defined; background noise, shrimp snap and dolphin click (Figure 1). The shrimp and dolphin transients were similar enough to provide a good test of the ML techniques that were available.

For this project, only basic readily available tools would be used. This meant using Matlab, with its ML toolbox, and Python, with various ML packages, and a basic desktop PC with no GPU. The learning for this project was about the scope of the ML techniques rather than finding the optimal solution in terms of hardware and software.

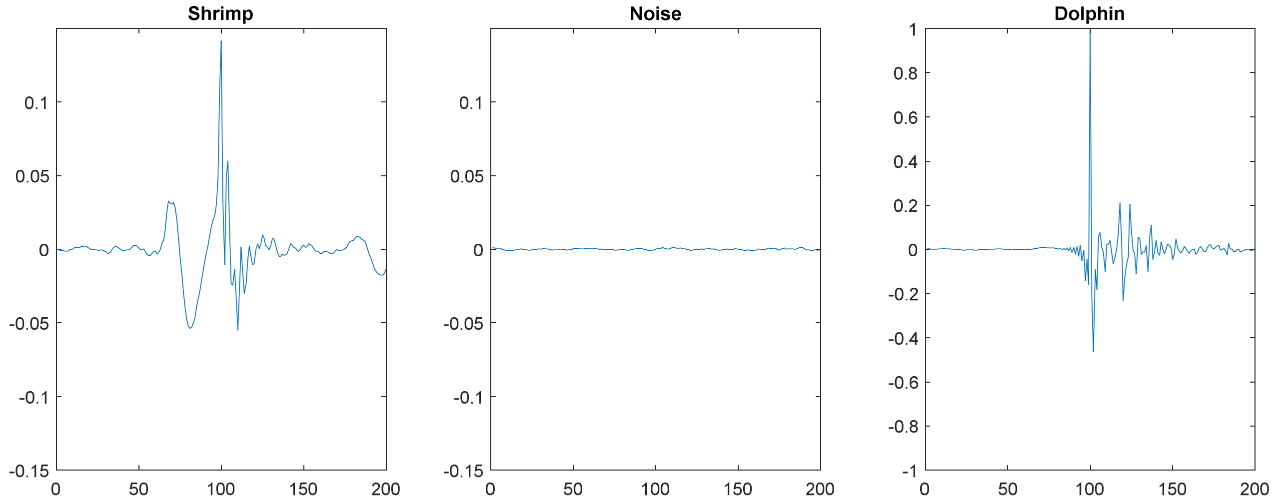

Figure 1: Time series examples of three classes

3 RESULTS

The hydrophone recordings contained thousands of shrimp snaps and hundreds of dolphin clicks. A few hundred of each of the biological transients were identified by hand and low-level noise regions were identified automatically. All these time segments were labelled ready for the classification network. Various ML techniques were investigated as follows.

3.1 Neural Networks

The initial idea was to utilise probably the best known of ML technologies; the convolutional neural network (CNN) which is used extensively for image recognition, particularly using the massive number of images available on the internet. A short period of time data can be converted to an image using the spectrogram, which provides a good summary of the data in temporal and frequency terms (Figure 2).

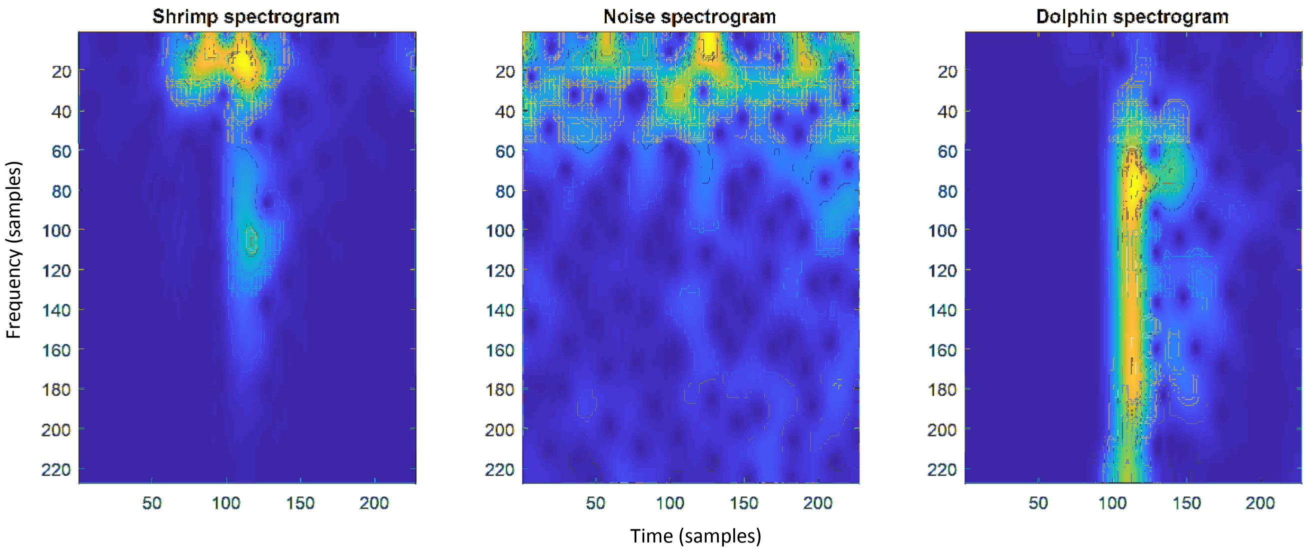

Figure 2: Spectrograms of the three examples

The literature suggests that tens of thousands of examples are required to successfully train a CNN network. This should be expected when we consider the complexity of photographs, but it was not expected that such a large number was required here. The few hundred examples were used in a ML technique called data augmentation, whereby the examples are re-used with some Gaussian noise added and the start position of the example has small jitter applied. This resulted in around 15000 examples in total, 80% of which would be used to train the network and 20% to test it.

The first network used was the CNN AlexNet, which is one of the most successful image recognition networks. Implementations of this network are available for both Matlab and Python and so were easy to use. This network took 10 minutes to train and achieved 98% accuracy on the test data. This was a great result in that it did not take too long and the accuracy was excellent. Creating a simple detector-classifier looked to be easily achievable using ML.

The next step was to simplify the AlexNet to see what layers in this network were necessary for this application. A few of the convolution layers were removed and training was now down to 2 minutes with 99.7% accuracy. It appeared that the complicated structure of AlexNet was not required for this application. Next, all the convolution layers were removed to make the most dramatic difference. The training now took 4 seconds and also achieved 99.7% accuracy (see Table 1 for summary). The simple structure of the spectrograms did not require the convolutional processing of the CNN at all, unlike photographs that contain real-world geometric shapes and repeatable structural relationships as well as much more complexity.

Table 1: Neural network performance summary

Network	Time to Train	Accuracy
AlexNet	10 minutes	98.0%
Reduced AlexNet	2 minutes	99.7%
Deep Neural Network	4 seconds	99.7%

3.2 Linear Classifier

After it was shown that complicated neural networks were not necessary to achieve excellent accuracy of classification for this data, other ML techniques were tried to see if they could offer any new insights. A linear classifier is one that takes multiple-dimension data and tries to find clusters that it can use to define hyperplanes in that data that can be used to separate classes. Figure 3 shows how a two-dimension set of data containing three classes can be easily separated using the two black lines shown.

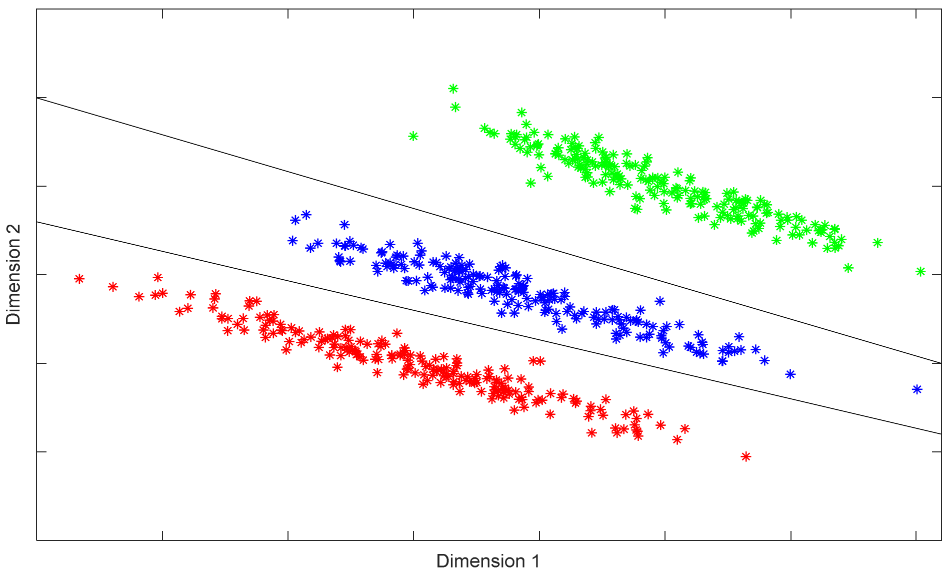

Figure 3: Using two linear classifiers to model three classes

The input to a linear classifier is a vector of data, i.e. Nx1 dimensions. The hydrophone data has more than the two dimensions in Figure 3, but how the data should be input is not straightforward. The obvious option would be to put the time series data in as each example. Unfortunately, the linear classifier did not converge in this case. The next test was with a vectorised version of the spectrogram of each example. This classifier trained in 1.2 seconds and yielded 99.8% accuracy on the test data.

It is extraordinary that the linear classifier has found such decisive separating hyperplanes in this data, but another machine learning technique can be used to confirm that this is indeed the case. Dimensionality reduction takes high-dimension data (our vectorised spectrogram) and reduces it to a lower number of dimensions while retaining distance relationship of the points. Figure 4 shows a portion of the hydrophone data reduced to two dimensions and we see discrete clusters of our three classes. The occasional red dot appearing in the other clusters represent shrimp that look like dolphin or noise, or more likely signals that were misclassified by the human in the preparation stage. In future, a simple dimensionality reduction test would be highly advisable before trying more complicated techniques.

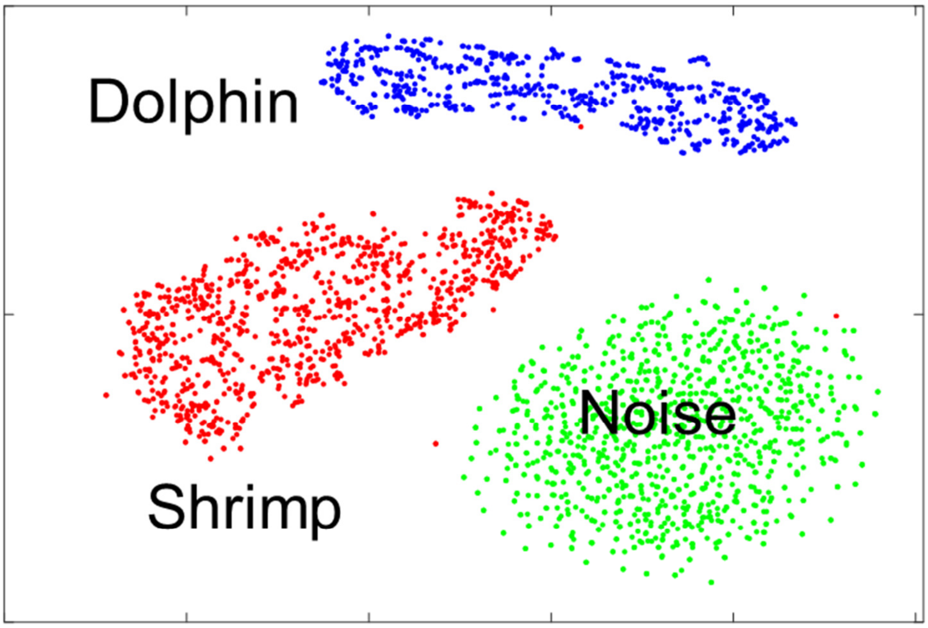

Figure 4: Using two linear classifiers to model three classes

3.3 Detector-Classifier

The linear classifier network can be used as a detector by using it to predict the class of the spectrogram of a short time series in a moving window through an extended period of the hydrophone data. The output of the classifier contains a score for each of the possible classes (called the negative loss) that can be understood as being a likelihood of each class. Figure 5 shows the negative loss for the three classes for a period of the hydrophone data. The time series includes a clear shrimp click at sample 3000 and an unidentified noisy signal at sample 2300. The loss curves on the bottom plot shows that the classifier correctly identifies the clear shrimp click and also suggests that there is a surface replica of that click at sample 3100. The model also suggests that the noisy period at 2300 is a shrimp complete with a surface bounce.

We have a created a detector-classifier that was trained in 1.2 seconds and is picking out signals that the human analyst finds difficult.

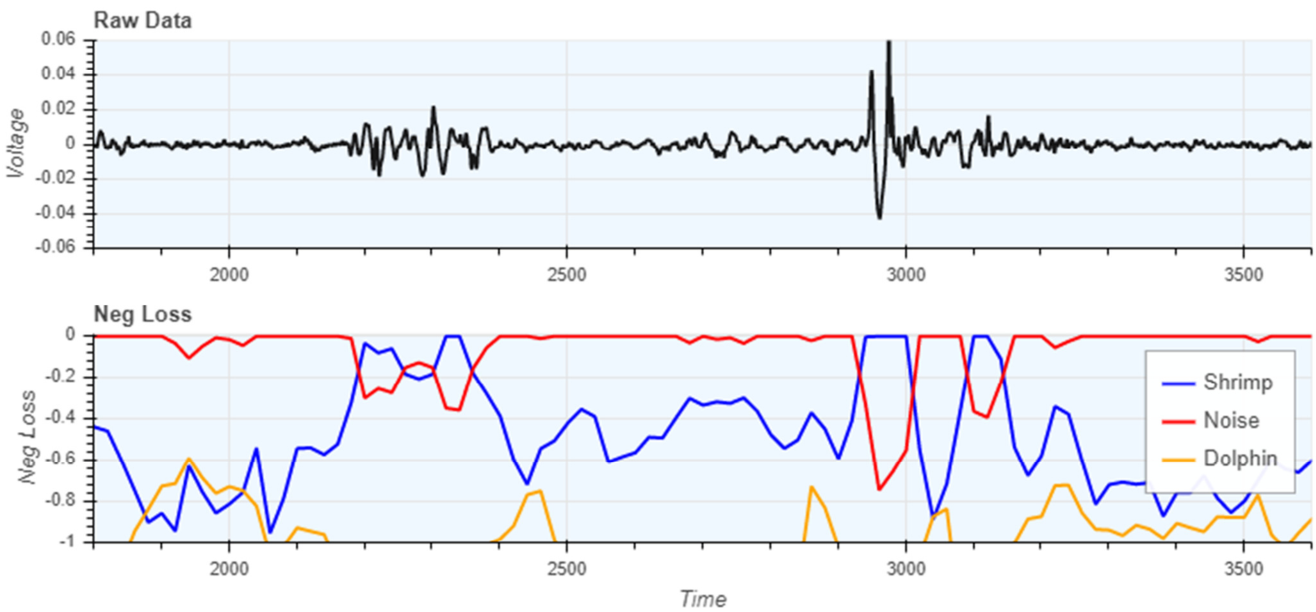

Figure 5: Using a linear classifier model as a detector-classifier

4 CONCLUSIONS

The techniques of machine learning have something to offer all scientific disciplines, but the ML-novice researcher should be careful where they start. Intuition based on expert domain knowledge led this researcher to start with convolutional neural networks when a very simple linear classifier was more than adequate.

A good starting point for any type of input data is the dimensionality reduction test to see if the data is easily linearly separable. If it is, then the classification solution may be very simple and very fast.

In addition, starting with the simple techniques means the researcher can make use of the simple tools to get started straight away. A multiple-GPU system with expensive software toolboxes may not be necessary to get good results.

A simple detector-classifier was demonstrated based on a linear classifier ML network using data recorded from a single hydrophone. Work needs to be done to test whether this system is quantifiably better than classical detection techniques. If successful, the next steps involve moving to multiple sensor data – sonar arrays – to find whether these insights can be carried over into higher data rate systems.

ACKNOWLEDGEMENTS

Many thanks to David Matthews for the collection of the data used here and for the long conversations on machine learning based on his work done at UWA.

REFERENCES

Purwins, Hendrik et al. 2019 “Deep Learning for Audio Signal Processing.” *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019): 206–219. arXiv:1905.00078

Xuhao Du, Andrew Youssef, Yue Wang, Nicholas Padfield, David Matthews 2018 “Detection of Snapping Shrimp Using Machine Learning” *Australian Acoustical Society Conference Proceedings*