CS 230 Project Milestone: Estimating the bandwidth of bandlimited signals using neural networks

Project Category: Supervised Learning, modelling, signal processing

Jonathan Tuck

February 23, 2018

Contents

1 Introduction **3**

2 Data **3**

3 Approach **4**

4 Preliminary results **4**

1 Introduction

In the field of signal processing, filtering is among the most ubiquitous of topics. Typically, when one wants to filter a signal, one must know what type of filtering should be done (*i.e.*, low-pass filtering, high-pass filtering, *etc.*,) which also implicitly requires the prior knowledge of the bandwidth of the signal. Unfortunately, unknown signals' bandwidths are typically only found by computing their Fourier transform and observing a steep dropoff in magnitude for a given frequency.

Currently, the method of computing a Fourier transform for a discrete signal can be computed in $O(n \log n)$ time [Bra78, Osg17].

The goal of this project is to estimate the bandwidth of a bandlimited signal given only its time domain representation. This neural network should be able to determine the bandwidth of any signal that is input, without having to compute the Fourier transform. Ultimately, the results of this project lead to a multi-purpose filter for use in signal processing topics.

2 Data

Training data. An advantage of this project is that the data is readily abundant and training data can be synthetically generated efficiently. Since all signals are simply sums of sinusoids at different frequencies and amplitudes, it is appropriate to synthetically generate data by simply adding randomly generated sinusoids together, some with various types of noise (*e.g.*, white noise, pink noise, *etc.*) For each example, we shall randomly choose the number of sinusoids to be added, and their frequencies. The amplitudes will all be normalized to 1; this is a reasonable normalization to make, as amplitude (to scale) has no effect on the bandwidth of a signal.

In order to keep the scope of this project aimed at bandlimited signals, we shall *a priori* determine a maximum bandwidth such that no signal shall have more than 5% of their spectrum outside that bandwidth (this allows us to consider noisy signals, as noise typically is considered to have infinite bandwidth.)

As data on computers is inherently discrete, it makes sense to consider discrete-time time series data. Specifically, the time series data is discretized into 1000 elements which correspond to $t \in [0, 1]$. That is, the input signal is $x(t) \in \mathbf{R}^{1000}$.

Test data. The test data shall be generated in the same way that the training data is generated. In this way, the training data and the test data shall be drawn from the same distributions, so as to minimize variance.

Data splits. In this project, we split our m examples into training, development, and test sets. We have chosen that 90% of the data shall go into the training set, 5% shall go into the development set, and the remaining 5% shall go into the test set.

3 Approach

The problem of determining the bandwidth of a signal is a supervised learning problem, and so neural networks for supervised learning problems will be used. We do not yet know the appropriate deep learning architecture for this project, but as the act of convolution is deeply related to Fourier transforms [Bra78, Osg17], we shall attempt to use architectures such as convolutional neural networks [LBBH98, RSA15].

Neural network architecture. As of now, the neural network architecture is four layers of fully connected layers, with 100, 50, 25, and 1 nodes, respectively. All of the activation functions are rectified linear units (ReLUs), except for the third layer's activation function, which is a sigmoid.

Evaluation metrics. The evaluation of this project shall be based on how close the neural network estimates for a signal's bandwidth are to the actual signal's bandwidth. Specifically, cost function is is

$$\mathcal{J} = (1/m)\|B - \hat{B}\|_2^2,$$

where m is the number of samples in the set, $B \in \mathbf{R}^m$ is the vector of actual signal bandwidths for the m examples, and $\hat{B} \in \mathbf{R}^m$ is the vector of signal bandwidth estimates for the m examples. Sometimes, in signal processing literature, this particular loss function is referred to as the *mean squared error* of $\hat{B}$ [GD10].

4 Preliminary results

Although the project is far from completion, we have successfully established a working baseline model for the project. Right now, for a possible maximum bandwidth of 20 Hz, 3000 examples, a learning rate of 0.0001, and a minibatch size of 50, our neural network architecture achieves an MSE of approximately 0.002354 on the training data and an MSE of approximately 0.2231 on the test data after 2500 epochs. Figure 1 is a plot of the cost versus epoch on the training data for this particular problem instance and initialization. The code for this project can be viewed at `https://github.com/jonathantuck/CS230-project`.

Work to be completed by the end of the project. The success of the current iteration of the neural network is a promising start, but should be improved upon. Most importantly, our current results indicate that there may be a variance problem with our current iteration of the model. The majority of the time spent until the end of the project will be to experiment with the number of layers, the number of nodes per layer, the optimizer selection, the type of regularization used (if needed), and the minibatch size. In addition, other types of neural network architectures will be considered, such as convolutional neural networks, if they produce better results.

Lastly, a significant portion of time will be spent on communicating my findings via both the poster presentation and in the final project report.

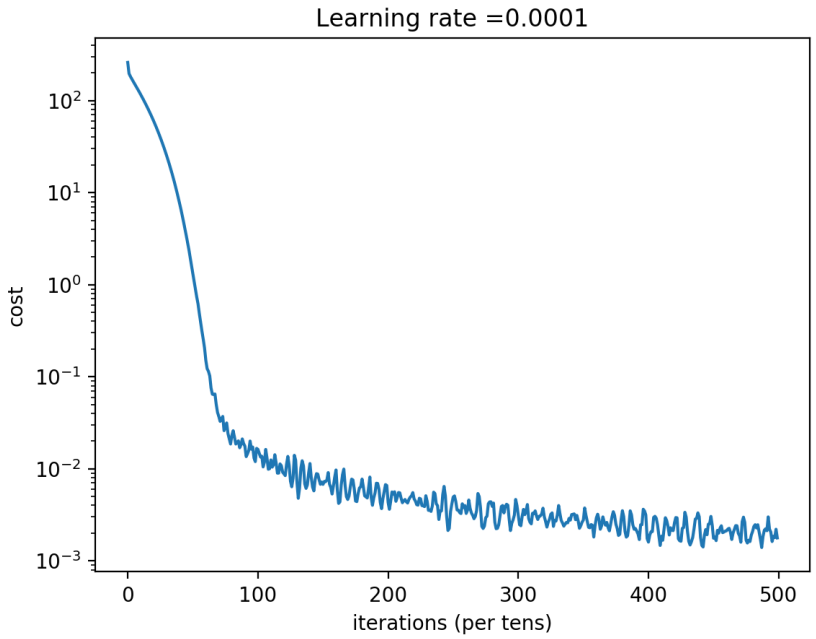

Figure 1: Cost versus epoch on the training data for problem instance with a possible maximum bandwidth of 20 Hz, 3000 examples, a learning rate of 0.0001, and a minibatch size of 50.

References

[Bra78] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, Tokyo, second edition, 1978.

[GD10] R. Gray and L. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[LBBH98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[Osg17] B. Osgood. *Lectures on the Fourier Transform and its Applications*. McGraw Hill, first edition, 2017.

[RSA15] O. Rippel, J. Snoek, and R. P. Adams. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems 28*, pages 2449–2457. MIT Press, 2015.