**Intelligent Discrete Fourier Transform**

*Author:* Oisín Watkins

*Date:* 15/01/2021

*Abstract*

This project sets out to implement a new way for sensor / signal data to be read and understood by Neural Networks, of varying types, that replaces the common approaches of Recurrent Neural Networks (RNN’s) or their equivalents and Convolutional Neural Networks (CNN’s) with pre-processed data. The resulting algorithm amounts to an Intelligent Discrete Fourier Transform, which simultaneously transforms time-domain data to frequency-domain data and learns what content of the signal is meaningful to the given application. The final implementation was built using the TensorFlow API and is available for download from github.

*Introduction*

At the time of writing this, the common practice for handling signal data using Neural Networks breaks down into one of two choices: Recurrent Neural Networks (RNN’s) and their equivalents, or pre-processing data into images to accommodate Convolutional Neural Networks (CNN’s). Neither approach is optimised to the task. RNN’s (or more recently 1-D CNN’s) handle series data, which in the context of signal processing is the equivalent of *only* working in the time domain. While it is possible, the amount of computation required to overcome the limitations of working in the time domain prohibit the use of RNN’s / 1-D CNN’s in most complex real-time applications. Likewise, pre-processing data into image format before using a conventional CNN is both time consuming and wasteful from a processing standpoint. The need to generate an image before being able to discern meaning from the data all but exclude this setup from being viable in edge-sensor networks or in real-time systems.

What both approaches lack is the ability to transform sensor data from time-domain to frequency-domain. Certainly, the sensor data – to – image approach attempts to overcome this hinderance, but the question remains: why perform unnecessary processing on a signal prior to discerning meaning from the signal. In this regard the RNN approach has the lead, as it does not pre-process the data at all, but being limited to the time-domain does hold this approach at a disadvantage. The goal of this project is to find a way to pull the best parts from both approaches to make one computationally efficient, real-time system capable approach of handling signal data in the frequency domain. The result was a Discrete Fourier Transform which learns what parts of the signal are meaningful for the given application (an Intelligent Discrete Fourier Transform, if you will).

After the initial tests and demonstrations using MATLAB, a DFT layer was written using the TensorFlow API in Python. As a test of efficacy, this new layer was used to build a model which categorised 1s audio clips of one-word voice commands (eg: “yes”, “no”, “up”, “down”, etc.), and the effectiveness of this model was compared to that of a 1-D ConvNet used as a demonstration in numerous online guides for audio processing with Artificial Intelligence. In this paper, we will review the mathematics behind the Intelligent DFT, the resulting implementation in TensorFlow, and the limitations of this implementation. For reference, the entire body of code discussed in this paper is available at: <https://github.com/OisinWatkins/Intelligent_Signal_Reader>

Bear in mind, these initial tests were performed as a proof of concept. They do not represent the last word on how well this approach handles time-domain data compared to the more typical 1-D ConvNet approach. However, they do indicate that this Intelligent DFT approach does stand to provide similar levels of performance to the more typical approach with greatly reduced processing times. These results suggest that more study on this approach is warranted for the potential improvements it offers to real-time signal processing AI applications.

*Overview*

The Fourier Transform is a well-documented variant of the Laplace Transform commonly used in the realm of signal processing. Formally it is defined as:

~

However, implementing this equation on digital systems requires the use of Discrete Time, rather than the native Continuous Time equation shown in figure 1. This Discrete Time Fourier Transform (more commonly referred to as the Discrete Fourier Transform, or DFT) is formally defined as:

~

As written by ~ the DFT can be implemented using Matrix Algebra, as shown in figure 3.

~

It is this matrix implementation of the DFT that forms the basis of this research. Using this approach, one could theoretically incorporate a DFT into a larger Deep Learning network. Given that backpropagation simply adjusts network parameters along the negative of the error gradient, and that the derivative of the error w.r.t. the values of the DFT matrix is linear, there should be few issues with incorporating this equation into larger networks.

Principally this does hold true, however as will be shown in the following sections, there are still some issues with implementation. Foremost among them is the inability of most Deep Learning frameworks to accommodate Imaginary Values in the Error Gradient. At the time of writing this it does not pose a major issue. With careful implementation it is possible to build a fully working Neural Network that uses Imaginary values in its layers. However, this limitation does have an impact on model performance. More on this later.

*Implementation*

For this project the TensorFlow API was used to define a DFT layer.

*Results*

*Discussion*

*Future Research*

*References*