hybrid model

A THESIS

Presented to the Department of Computer Science and Computer Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

Option in Computer Science

Committee Members:

By Ruben Rosales

May 2020

Abstract

Contents

Illustrations

Figures

Tables

Chapter 1

Introduction

[explain what black box and white box is] In this thesis, we study interpretable machine learning as applied to complex-valued time series. Scientists have studied the use of several machine learning methods such as Convolutional Neural Networks, Recurrent Neural Networks, and Support Vector Machines for time series classification.

[add more reasons] [add how most time series are classified] These methods, however, fall short of allowing users to visualize patterns within their dataset and ().

To address this issue, we propose an interpretable hybrid model that can be extended to any time series dataset. Our architecture is composed of two models, an ensemble method of classifiers that functions as a black box method, and a white-box model that encodes time series as images, attemps to classify them through a CNN, and outputs all images correctly classified in both black-box and white-box method.

In our black-box model, we analyze deep learning models such as CNN's which have become increasingly popular and widely used in the last decade but their complex nature makes it difficult for users to understand what is going on. So, as black-box models get more popular the need for an interpretable accessory? is needed so we propose a hybrid model consisting of a black box for classification and an interpretable white-box model which highlights characteristics of a time series for users to understand.

The need for interpretable models in management problems

The growth of mobile devices and demand for wireless data has created a need for igh quality spectrum sensing and adaptation to improve spectral allocation and interference mitigation is an important route by which we may achieve this. however have been constrained to relatively specialized solutions which lack the generality needed to deal with a complex and growing number emitter types, interference types and propagation environments. This is a significant challenge in the community as expert systems designed to perform well on specialized tasks often lack flexibility and can be expensive and tedious to develop analytically.

Contributions

Data

Our dataset contains two different types of signals, raw and discretized. The raw signal consists of time and amplitude with 31,000 data points while discrete data can vary from 22 to 44. In industry, they do not have access to raw data because it is difficult to track that information so we focus on discrete data for this thesis.

The raw signal has the same time intervals across the entire dataset but it may have different start and end times.

[What is raw signal and discrete, compare them,]

Chapter 2

Preprocessing

We compared this method to other features extracted from Fourier-transform, fast Fourier transform, sliding window method, and polar coordinates taken directly from the signals amplitude/phase value.

Using any of those methods reduced our models accuracy as well as increased training time. For fourier transform, we attempted to perform ft over the entire signal and use amplitude and time as a feature accuracy dropped below 90

We normalize all data using L2 norm and format our data by each subsequent wavelet decomposition. For instance, if we do N levels of decomposition the Nth item would be at the end of the array.

What is it?

[what is feature extraction]

Data pre-processing is concerned with the analysis and manipulation of the collected spectrum data with the aim to arrive at potentially good wireless data representations. The raw samples organized into data vectors rk in the previous block are pipelined as input for signal processing (SP) tools that analyze, process and transform the data to arrive at simple data representations such as frequency, amplitude, phase and spectrum, or more complex features xk such as e.g. cyclostationary features. In addition, feature learning such as deep learning may be utilized to automatically extract more low-level and high-

level features. In many ML applications the choice of features is just as important, if not more important than the choice of the ML algorithm.

Transformations

Transformation 1 : STFT/ FT [what is it]

[fft formula] The Fourier Transform takes a time-based pattern, measures every possible cycle, and returns the overall ”cycle recipe” (the amplitude, offset, and rotation speed for every cycle that was found).

We will begin the analysis of the Fourier Transform on this page. This section gives an introduction to the Fourier Transform and then we take a look at the fundamental properties of the Fourier Transform.

The Fourier Transform is a magical mathematical tool. The Fourier Transform decomposes any function into a sum of sinusoidal basis functions. Each of these basis functions is a complex exponential of a different frequency. The Fourier Transform therefore gives us a unique way of viewing any function - as the sum of simple sinusoids.

The Fourier Series showed us how to rewrite any periodic function into a sum of sinusoids. The Fourier Transform is the extension of this idea to non-periodic functions.

While the the Fourier Transform is a beautiful mathematical tool, its widespread popularity is due to its practical application in virtually every field of science and engineering. It’s hard to understand why the Fourier Transform is so important. But I can assure you it enables the solution to difficult problems be made simpler (and also makes previously unsolved problems solvable). In addition, the Fourier Transform gives us a new method of viewing the world, which is fantastic for giving a more intuitive feel for our universe.

[stft] The Short-time Fourier transform (STFT), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.[1] In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment. One then usually plots the changing spectra as a function of time, known as a spectrogram or waterfall plot.

[why it performed poorly] The problem with STFT is the fact whose roots go back to what is known as the Heisenberg Uncertainty Principle. This principle originally applied to the momentum and location of moving particles, can be applied to time-frequency information of a signal. Simply, this principle states that one cannot know the exact time-frequency representation of a signal, i.e., one cannot know what spectral components exist at what instances of times. What one can know are the time intervals in which certain bands of frequencies exist, which is a resolution problem.

The problem with the STFT has something to do with the width of the window function that is used. To be technically correct, this width of the window function is known as the support of the window. If the window function is narrow, then it is known as compactly supported. This terminology is more often used in the wavelet world, as we will see later.

The problem is that when you are studying a real signal it would be useful to know what the ‘instantaneous frequency’ of the signal is. The instantaneous frequency is the exact frequency of a signal at an exact moment in time. For instance if I was listening to a music track I would like to be able to say ‘at 1 min 59.0423 seconds into the music

track the sound is 1563.2 Hz'. Unfortunately the Fourier Transform cannot do this because there exists a minimum amount of uncertainty between the frequency and time domains, like Heisenberg has a minimum amount of uncertainty between the speed and position of a particle (the area of the box). You can know the moment in time you want to find the frequency for (like the blue box), but because there is a minimum uncertainty, the box of has to stretch out across frequency, meaning that you are unsure of the frequency at that moment in time.

The best you can do with a Fourier Transform is to sample a range of time (for instance, the time signal between 1 min 58 sec and 1 min 59 sec in a song) and find a range of frequencies that were played over that amount of time, as represented by the black box. An example of this can be seen by looking at the third picture in the post again. There is a signal over a range of time (e.g. someone saying 'hello') and the frequency graph is the range of frequencies recorded over that time.

Transformation 2: Abs/angle The representation of a complex number as a sum of a real and imaginary number, z = x + iy, is called its Cartesian representation. We attempt to use the real value of the absolute cartesian representation so we have two features, angle and absolute value,

[purpose and motivation? Behind making it ang/abs]

[definition plus method]

To overcome this resolution problem a Wavelet Transform is used to deconstruct the signal into a load of wavelets being added together. Wavelets are useful because they are limited in time and frequency. Instead of a wavelet lasting forever and having no limit in time, it dies quickly, like the example of different wavelets below shows:

These waves are limited in time, whereas sin() and cos() are not because they continue forever. When a signal is deconstructed into wavelets rather than sin() and cos() it is called a Wavelet Transform. The graph that can be analysed after the transform is in the wavelet domain, rather than the frequency domain. This time limiting quality about wavelets is useful to Engineers as it provides more resolution in the time domain. Instead of modelling with an infinite wave, it is possible to model with a finite wave which you can 'slide' along to time domain. The ability to slide the signal is what gives Engineers a more accurate representation of the signal and therefore a better resolution in time.

Transformation 2: Sliding window We propose a method similar to STFT but in an effort to reduce the size of the data we take the mean at each window which would turn our size into size N where N is equal to the number of windows we use.

Transformation 3: Wavelets

Db4 wavelet http://wavelets.pybytes.com/wavelet/db4/

Model 1: Model 2: Model 3:

[Shape of data comparison]

In the preprocessing stage we tried numerous combinations of features but found the best performing features to come from wavelet decomposition.

Wavelets

We implement the discrete wavelet transform as a filter-bank which means it's composed of cascading high-pass and low-pass filters. This gives us the advantage of splitting a signal into several frequency sub-bands. Wavelet decompositions give us the advantage of gaining features that take into account frequency and time domains.

[explain why they performed better]

[alternatives]

Models

[introduce models definitions]

This section gives an overview of the theoretical background of concepts discussed in this work.

Machine learning algorithms can simply be seen as learning a target function (f) that maps input data (X) to an output (Y). There are several techniques to make this work but we focus on nonparametric algorithms, namely Convolutional Neural Networks (CNN) and Long Short Term Memory networks (LSTM). Nonparametric algorithms are algorithms that attempt to make minimal assumptions about the form of the function so they can learn any form from data provided to it. An example would be a neural network because it has no prior knowledge about what it's classifying and attempts to generalize any new data points that it has not seen before.

[cnn]

A Convolutional Neural Network is a type of deep neural network that can be applied to different domains such as computer vision or time series analysis. It consists of an input and output layer as well as hidden layers. A convolution is an operation between a vector of weights w against an input x. It consists of taking the dot product between m and x were in steps of a filter size defined by n.

Convolutional neural networks perform feature learning via non-linear transformations implemented as a series of nested layers. The input data is a multidimensional data array, called tensor, that is presented at the visible layer. This is typically a grid-like topo-

logical structure, e.g. time-series data, which can be seen as a 1D grid taking samples at regular time intervals, pixels in images with a 2D layout, a 3D structure of videos, etc. Then a series of hidden layers extract several abstract features. Those layers are "hidden" because their values are not given. Instead, the deep learning model must determine which data representations are useful for explaining the relationships in the observed data. Each layer consists of several kernels that perform a convolution over the input; therefore, they are also referred to as convolutional layers. Kernels are feature detectors, that convolve over the input and produce a transformed version of the data at the output. Those are banks of finite impulse response filters as seen in signal processing, just learned on a hierarchy of layers. The filters are usually multidimensional arrays of parameters that are learnt by the learning algorithm [24] through a training process called backpropagation.

[lstm] Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition[2], speech recognition[3][4] and anomaly detection in network traffic or IDS's (intrusion detection systems).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important

events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

[multimodal] Due to the superior performance and computationally tractable representation capability (in vector spaces) in multiple domains such as visual, audio, and text, deep neural networks have gained tremendous popularity in multimodal learning tasks [38, 39, 44]. Typically, domain-specific neural networks are used on different modalities to generate their representations and the individual representations are merged or aggregated. Finally, the prediction is made on top of aggregated representation usually with another a neural network to capture the interactions between modalities and learn complex function mapping between input and output. Addition (or average) and concatenation are two common aggregation methods, i.e., The network structure is illustrated in Figure 1(b) . The arrows are function mappings or computing operations. The dotted boxes are representations of single and combined modality features. We call them additive combinations because their critical step is to add modality hidden vectors (although often in a nonlinear way).

[explain architectures]

CNN

Our CNN has an architecture consisting of 3 convolutional layers each followed by a batch normalization and dropout layer then finally connected to two dense layers (Figure

X). We found no improvements in performance when we attempted to make it deeper by adding additional convolutional layers as well as skip connections.

LSTM

We wanted to keep our LSTM network as small as possible, for X purposes, so we went with two LSTM layers of 128 hidden states and a dropout rate of 4/10 followed by a dense layer of 128 connections. We attempted different state sizes but found 128 to be the smallest number with the best consistent performance.

We propose a multimodal learning architecture for the purpose of time series classification which is illustrated in Figure (X). The multimodal architecture allows our data to vary in size and given that all four levels of decomposition have varying length due to downsampling, we propose each model to learn the representation of each distinct feature. For this architecture we propose two models, a 1D CNN and an LSTM, in which we employ two CNN's and 1 LSTM. They are all concatenated after their respective dense layer which performs the final We tried numerous variations and saw no impact on performance.

black box

Our model is constructed of four different architectures, 2 LSTM and 2 CNN's, that are concatenated at the dense layer. The four features we use are amplitude and phase of each level of decomposition for HV and VV. We chose four because we wanted to take into consideration both HV / VV in the event that one had more descriptive features than the other. Wavelet decomposition produces a complex value but that is unusable in a standard

neural network due to loss of information unless we move to a complex LSTM but as stated in there isn't a significant increase in accuracy.

white box

Our white box is composed of a model that takes in raw data as an input and converts each signal into an image through the MTF method. We then pass those images through a CNN and those that are correctly classified are shown to the user.

In order to classify those images we created a CNN for it specifically and avoided using our existing CNN in order to make it deeper and increase performance when images aren't that classifiable. We explored several architectures, such as, ResNet, Inception-v4, and AlexNet but we decided to create an architecture similar to AlexNet due to the variation and size of datasets we used we couldn't create a model deeper or more complex than it otherwise performance would drop because we didn't have enough data to properly train it. We were able to achieve 60% accuracy on our dataset.

Notation

Layout

Chapter 3

Problem Statement

Chapter 4

Related Work

Chapter 5

Markov Transition Fields

Chapter 6

Main Results

Chapter 7

Conclusion