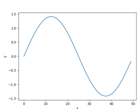
Sin Cos Waveform Detection using LSTMs

Sachin Goyal, EE UG, IIT Bombay

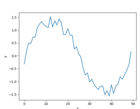
I. INTRODUCTION

This documents deals with making a Waveform detector using LSTMs. To be precise, we are posed with a challenge wherein a signal is being fed to us and we have to detect or raise an alarm when some particular types of waveform (Fig. 1 and Fig. 2) are detected. The incoming data maybe noisy. Now since the waves we want to detect have a time series pattern and are a time sequence, we use LSTMs for this detection task. The incoming signal is sampled using a sliding window. The signal captured by the window is then passed through our LSTM model which gives 3 types of outputs:

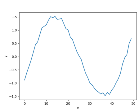
1) Output 0: Means no particular waveform detected
2) Output 1: Means waveforms of category 1 detected
3) Output 2: Means waveforms of category 2 detected



(a) No Noise and Distortion

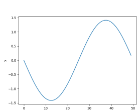


(b) Noisy Version

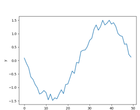


(c) Noisy and Distorted

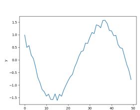
Fig. 1: Waves Classified in Category 1



(a) No Noise and Distortion

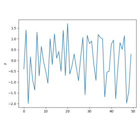


(b) Noisy Version

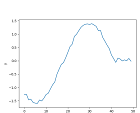


(c) Noisy and Distorted

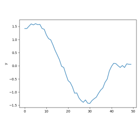
Fig. 2: Waves Classified in Category 2



(a)



(b)



(c)

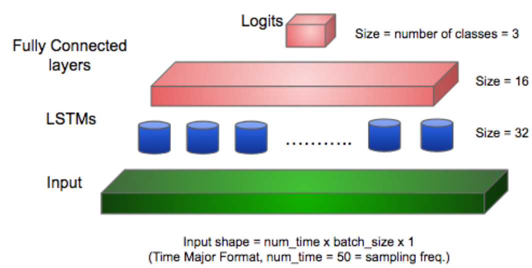
Fig. 3: No Alarm Raised on such samples, Category 0

II. TRAINING LSTM

I created 20k samples of such **individual wavelets** from the 3 categories varying the amount of noise, magnitude of noise and distortion magnitude. Some small phase shifts (upto 20 degrees) were also applied to waveforms of category 1 and 2 to make the classifier robust to small shifts or errors in the input sequence.

III. MACHINE LEARNING ARCHITECTURE

I used a 1 layer LSTM with 32 hidden units and time step set equal to sampling frequency which is 50. The last output from each LSTM cell of each hidden unit is chosen and then passed through a network of 2 fully connected layers gradually decreasing the output vector dimension from 32 to 16 and finally to 3. Fig. 4 gives a pictorial description of our Neural Network. Table 1 lists other hyper-paramter values



(a)

Fig. 4: Neural Network Architecture Used

Hyper-Parameters	value
Batch Size	256
LSTM Time Steps	50
Maximum Gradient Norm	10
Learning Rate	10^{-4}
Epochs	20

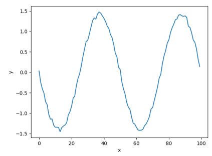
TABLE I: List of all network Hyper-Parameter values

IV. RESULTS

The accuracy of LSTM was first of all tested on discrete randomly generated wavelets (but of form similar to category 0,1,2). The accuracy achieved was around **92.75%**.

For the real scenario where we will have input samples from consecutive sliding windows, I have shown some of the results I simulated. Each window is of length 50 samples and is shifted by 5 for the next sampling. The accuracy was quite satisfying for the sample waves I used. Fig. 5 and Fig. 6 show some sample waves I used as input.

Waveform in fig. 5 is of length 100, hence 11 sliding window samples each of length 50 with shift of 5 samples were acquired. The output sequence was 2,2,0,0,1,1,0,0,0,2,2. If one observes closely, the network is successfully able to

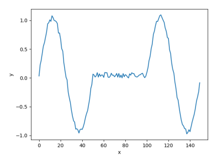


(a)

Fig. 5: Test Waveform 1. Output: 2,2,0,0,1,1,0,0,0,2,2

identify the Sin wave (Category 1) in the middle of the input sequence.

In the output of test wave 2 (Fig. 6), we can see that the network labels the last 3 samples as 1 which may be debated for its correctness.



(a)

Fig. 6: Test Waveform 2. Output: 1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1

V. Discussion

I presently have used just the wave value as a single feature input to the LSTM. We could also try using 2 feature inputs like value combined with derivative for more complex tasks.