Abstract

The sun spot time series is a random series generated with a complicated nonlinear system. Time delay neural networks (TDNN) are typical nonlinear systems that could be used to perform time series prediction. In this project, we applied the TDNN trained with both MSE and MEE criteria to predict the time series. Both criteria showed the effectiveness of the nonlinear systems. Furthermore, the TDNN system trained with MEE criteria performs even better that the conventional MSE criteria in one-step prediction. By this result, we showed the advantage of utilizing the information theory to minimizing the error entropy over the MSE criteria.

1. Introduction

The goal of this project is to implement a Time Delay Neural Network (TDNN) trained with both the MSE and the MEE criteria to perform 1-step prediction of the sun spot time series data. Because of the nonlinearity of the system, the TDNN would be suitable to predict this dataset since it is nonlinear in nature. We choose the topology of the TDNN to be 5-X-1, which stands for 5 neurons in the input layer, X neurons in the hidden layer, and 1 neuron in the output layer. The neuron in the output layer is simply a linear combiner without the nonlinear part. The number of neurons in the hidden layer is considered as one free parameter which is to be optimized in our experimental simulations.

The TDNN will be trained with both the MSE and the MEE criteria in this project. The conventional MSE criteria is defined as minimizing the mean squared error. In practice, it is often achieved adaptively using the simple gradient descent or LMS algorithm. Further, the cost function $J_w(e) = E[(z - f(x, w))^2]$ taking the quadratic form would ensure the existence of a global minimum. In fact, in cases where the Gaussian assumption holds true, the MSE criteria is able to provide the optimal solution. However, when this assumption is relaxed, since the quadratic cost function tends to put substantial amount of emphasis on the outliers, this criteria is no longer optimal. In this project, since the neurons in the hidden layer are incorporated with a nonlinear element, the Gaussian assumption is generally not true. Therefore, we can expect that the MEE criteria would outperform the MSE criteria since the former attempts to minimize the error entropy. While we are actually trying to implement the MEE, the distribution of the error would become more and more peaky. If an additional constraint forcing zero-mean on the error samples exists, ideally the MEE criteria would yield a delta function centered at the origin indicating the error free condition. Although attempting to minimize the error entropy directly modifies the shape of the error density and is computational complex, some fast recursive algorithms such as MEE-RIP would still make it especially suitable for training nonlinear systems.

2. Designing Principles

In this section, we are going to discuss the basic concerns and methods involved in the designing and training of our TDNN predictor. The selection of the free parameters is discussed in detail in this section.

2.1. Topology Selection

The sun spot time series is a complicated stochastic series generated by an underlying nonlinear chaotic
system. In light of predicting the time series well, we choose the TDNN topology to be 5-X-1 which contains three layers. The input layer is chosen to contain 5 input taps in which the previous 5 points of the dataset is fed to the taps with the topmost tap storing the most recent data sample. The reason why we choose 5 input taps is because that all linear systems trained in previous homework assignments perform best in a 5-tap FIR filter configuration. Then the signals are fed forward to the first hidden layer containing X number of neurons. Each neuron contains a linear combiner and a nonlinear activation function. The data samples fed into the neurons in the hidden layer are first multiplied by their corresponding synaptic weights. Then the output potentials of the linear combiners are further modified by the activation function. Then the output layer contains only one single neuron with only the linear combiner part. This configuration gives only one output sample at a time which represents the predicted value for the sample of the next point.

Now the question comes to selecting the best value of X to yield the minimal predicting errors. From the appearance of the dataset, we noticed that the sun spot pattern includes a relative slow cycle plus a random fluctuation superimposed on it. This distinct characteristic probably indicates that the current data sample depends on quite a few past samples but by no means this memory length gets very large. Hence, it is suggested that we let X range from 3 to 5 and train the TDNNs on the same training set, respectively. Then the normalized MSE is calculated and compared. We only optimize the topology of the TDNN trained with the MEE criteria. For the TDNN trained with the MEE criteria, the same X is used for fair comparison in the following sections. The delta rule and back-propagation are adopted while training the TDNN with the MEE criteria. This rule is stated as follows

1. Evaluate local gradients for the output layer for \( s, t = 1, \ldots, N \) and \( j = 1, \ldots, m_l \)

\[
\delta^1_j(s, t) = -V'(e_j(s) - e_j(t)) \cdot \phi'(V^1_j(s)) \\
\delta^0_j(t, s) = -V'(e_j(t) - e_j(s)) \cdot \phi'(V^0_j(t))
\]

2. For layer index \( l \) decreasing evaluate the local gradients

\[
\delta^l_j(s, t) = \phi'(V^l_j(a)) \sum_{k=1}^{m_{l+1}} \delta^{l+1}_k(s, t) w_{kj}^{l+1} \\
\delta^l_j(t, s) = \phi'(V^l_j(t)) \sum_{k=1}^{m_{l+1}} \delta^{l+1}_k(t, s) w_{kj}^{l+1}
\]

3. Evaluate the weight updates for each layer \( l \)

\[
\Delta w_{ji}^l = -\eta(\delta^l_j(s, t) y_{ji}^{l-1}(s) + \delta^l_j(t, s) y_{ji}^{l-1}(t))
\]

We trained the TDNNs on a set of training data of length 1000 which is normalized to the range of \([0, 1]\). Then we use the final weights of the TDNNs to predict the same dataset. While training different TDNNs, we utilized the MEE-RIP algorithm of \( L = 50, \lambda = 0.05 \) to speed up the training process. The kernel size and the step size are chosen and fixed to be \( \sigma = 0.5 \) and \( \eta = 0.01 \) in this simulation. We use the hyperbolic tangent \( \phi(v) = a \tanh(b v) \) as the activation function with \( a = 1, b = 3 \). After 1000 epochs of training, all TDNNs with different number of neurons in the hidden layer converge and are tracking the data correctly. The last 500 data samples are used to conduct a cross validation and the normalized mean square prediction errors are calculated for each TDNN. The mean square prediction error for each TDNN is plotted in Figure 1. From this figure, we can therefore select \( X \) to be 5 and this number will be fixed during all subsequent experiments.

![MSE vs X](image)

**Figure 1.** The MSE plotted against no. of PEs in the hidden layer

### 2.2. Optimizing the Kernel Size and Step-size

In this experiment, we runned the TDNN with different kernel sizes and the stepsizes to try to select a fast adaptation speed but still converges. The experiment is based on the same training dataset with 1000
samples. The initial guess of the kernel size is calculated by the Silverman’s rule which is defined as

$$
\sigma_{opt} = \sigma_X \left\{ 4N^{-1}(2d + 1)^{-1} \right\}^{1/d}
$$

where we first use the signal standard deviation to substitute $\sigma_X$ which means that the predicted signal is zero. The actual error signal standard deviation would be smaller than this. The estimated kernel size by this rule is $\sigma = 0.042$. However, in practice, this kernel size is further adjusted since such a small kernel size tends to make the performance surface spiky and hard to handle in the process of weight adaptation. The probable flat spots and local minima that might encounter during the process of adaptation makes the select of the stepsize very hard if only the gradient descent method is used. Thus we start with the kernel size $\sigma = 0.4$ to ensure the smoothness of the performance surface and then choose around this value. Finally we choose the combination of kernel size and the stepsize to be $\sigma = 0.5$ and $\eta = 0.02$ to compromise between the adaptation speed and the fine tuning around the minimum. The adaptation process and the normalized MSE for this specific parameter setting are shown in Figure 2. It can be seen that roughly after 400 epochs of training, the adaptation process can reach an MSE smaller than 0.4.

3. Simulation Results

In this section we provide the prediction results for the sun spot time series for the 5-5-1 TDNN trained with both the MEE and MSE criteria. Figure 3 shows a block of data as well as the predicted values using the TDNN trained with MEE. The predicted series tracks the characteristics of the original dataset pretty well. The error histogram for the whole test set is plotted in Figure 4. As we can see that the prediction error for TDNN trained with MEE criteria does not differ substantially from the TDNN trained with MSE criteria. Both TDNNs produced a relatively big error with the mean square error for MEE and MSE being 0.1400 and 0.1540, respectively. This is probably due to the selection of the model topology. For the MEE criteria, using the conventional gradient descent rule in training imposes a dilemma between the kernel size and the stepsize. For larger kernel sizes, the stepsize can be larger but for small kernel sizes the stepsize should be kept small enough to prevent divergence. However, larger kernel size tends to emphasize the outliers and thus deteriorating the performance. In this experiment, when the kernel size is small, there is a good chance that the weight adaptation process does not converge for increased stepsize. Thus, we finally decided to choose a relative large kernel size $\sigma = 0.5$ to ensure convergence. But we have to keep in mind that the performance could be suboptimal.

Next we did the autonomous prediction for the two TDNNs and the optimal Wiener filter trained in Homework 1. In this simulation, we try to generate the sun spot time series with the trained models. Five initial samples from the original dataset are fed to the models as the initial condition. Then instead of continuing to feed the test dataset to the input, the output of the model is fed back as the latest input. In this experiment, the ability of the algorithms to characterize the property of the dataset is demonstrated by the length of samples that the given systems can predict correctly. We chose 50 different starting point as initial conditions
to the systems and the average length that the normalized error is less than 0.8 is calculated for each model. The results are shown in Table 1.

This results seems very unreasonable and I still don’t find the answer to this. One possible reason is that for MSE and MEE training, I chose different starting point in the test series as the MEE-RIP needs a memory length for the error samples. Choosing a different starting point is purely for the sake of the easiness in programming. However, on the other hand since all algorithms converges after the test dataset ends, the TDNN trained with MEE criteria should perform better than any other algorithms since it gets the smallest mean square prediction error. Anyway, this problem should be further investigated in the future.

4. Conclusions

In this project, we studied and compared the performances of a time delay neural network train both with the MSE and the MEE criteria in a nonlinear time series prediction task. Both the MSE and the MEE criteria produced very satisfying prediction results with the MEE criteria slightly superior to the MSE criteria. This result demonstrates the advantage of using

5. References