Multi-task Modular Backpropagation for Dynamic Time Series Prediction

Dr. Rohitash Chandra

1. Centre for Translational Data Science
http://sydney.edu.au/data-science
2. School of Geosciences

International Joint Conference on Neural Networks
Rio, Brazil

July 9, 2018

Overview

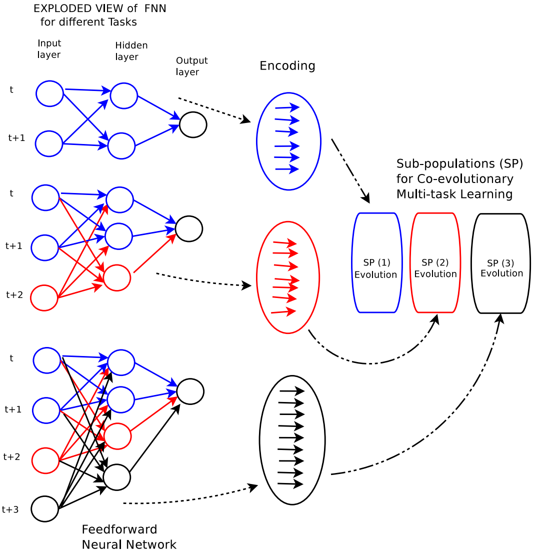
Background

- In certain types of problems, such as emerging storms or cyclones, robust prediction is needed even when partial information is available. Dynamic time series prediction refers to "on the fly" prediction given partial information.
- Due to knowledge representation as modules, the system should be able to make decisions with some degree of uncertainty even if some of the modules are damaged or missing.
- Such motivations come from biological neural systems, i.e due to modular knowledge representation, one is able to see even if one eye is damaged.

Background and Motivation

- Multi-task learning employs shared representation knowledge for learning multiple instances from the same problem.
- In the case of time series, multi-task learning can consider different a set of embedding dimensions as tasks that have shared knowledge representation.
- Recently, a neu- roevolution approach called co-evolutionary multi-task learning has been proposed to provide robust prediction for dynamic time series.
- In this paper, we adapt the method with multi-task modu- lar backpropagation that features gradient descent and transfer learning.

CMTL


EXPLODED VIEW of FNN
for different Tasks
Input
layer
Hidden
layer
Output
layer
Encoding
t
t+1
t
t+1
t+2
t
t+1
t+2
t+3
Sub-populations (SP)
for Co-evolutionary
Multi-task Learning
SP (1)
Evolution
SP (2)
Evolution
SP (3)
Evolution
Feedforward
Neural Network

Time series

The set of input features is defined as the timespan S_n which is essentially a subset of the embedding dimension (D) from space space reconstruction via Taken's theorem. The timespan is composed of overlapping data-points defined as follows.

$$S_1 = [y_t] \tag{1}$$

$$S_2 = [y_t, y_{t-1}] \tag{2}$$

$$S_n = [y_t, y_{t-1}, ..., y_{t-n}] \tag{3}$$

where $D == S_n$

Multi-task learning

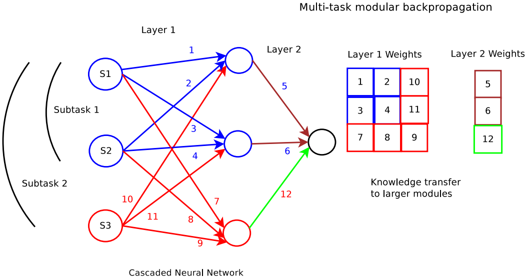


Figure: This cascaded modular neural network which can also be viewed and implemented as an ensemble of neural networks. The colours associated with the synapses in the network are linked to their encoding that are given as different modules. Subtask 1 employs a network topology with 2 hidden neurons while the rest of the modules add extra input and hidden neurons.

Method

In the case of dynamic time series, consider a set of unique values for embedding dimension given by vector

$$\Omega_m = [D_1, D_2, ...D_M] \tag{4}$$

where M is the number of cases of ψ considered, given $M \leq D$. Hence, the problem of dynamic time series prediction can be given by Ψ_m, where $m = 1, 2, ..., M$.

$$\Psi_m = x(t), x(t-1), ..., x(t-\Omega_m) \tag{5}$$

Method

The subtask θ_m is constructed in a cascaded network architecture as follows.

$$\begin{aligned}
\Phi_1 &= [\omega_1, \upsilon_1]; \quad \theta_1 = (\Phi_1) \\
\Phi_2 &= [\omega_2, \upsilon_2]; \quad \theta_2 = [\theta_1, \Phi_2] \\
&\vdots \\
\Phi_M &= [\omega_M, \upsilon_M]; \quad \theta_M = [\theta_{M-1}, \Phi_M]
\end{aligned} \tag{6}$$

The subtasks considered for optimisation via modular back-propagation is therefore $\boldsymbol{\Phi} = (\Phi_1, \ldots, \Phi_M)$.

Method

$$
\begin{aligned}
y_1 &= f(\theta_1, \Psi_1) \\
y_2 &= f(\theta_2, \Psi_2) \\
&\vdots \\
y_M &= f(\theta_M, \Psi_M)
\end{aligned}
\tag{7}
$$

Given T samples of data, the loss L for sample t can be calculated by root mean squared error.

$$
L_t = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (\hat{y} - y_m)^2} \tag{8}
$$

where $\hat{y}$ is the observed time series and y_m is the prediction given by subtask m.

Algorithm

Data: Reconstructed time series data
Result: Loss E given by Equation 9

Initialisation: Create n cascaded modules that correspond to timespan S_n from reconstructed time series. $\Phi_n = network(i_n, h_n, o_n)$, where i is number of input neurons, and h is number of hidden neurons. o_n represents number of output neurons which is fixed as 1.

while *not termination* **do**
    for *each cascaded-module* Φ_n **do**
        for *depth* d **do**
            i. Forward-propagate(Φ_n, S_n)
            ii. Back-propagate(Φ_n, S_n)
        end
        transfer(Φ_n, Φ_{n+1})
        if *restore-knowledge* **then**
            restore(Φ_n, Φ_{n-1})
        else
        end
    end
end

Algorithm 1: Multi-task modular backpropagation

Design of Experiments

- In the benchmark chaotic time series problems, the Mackey-Glass, Lorenz, Henon and Rossler are the four simulated time series problems. The experiments use the chaotic time series with length of 1000 generated by the respective chaotic attractor. The first 500 samples are used for training and the remaining for testing.
- In all cases, the phase space of the original time series is reconstructed with the embedding dimensions for 3 datasets for the respective subtasks with a set of unique values in the embedding dimension with increasing order $\Omega = 3, 5, 7$ and time lag $T = 2$.

Results

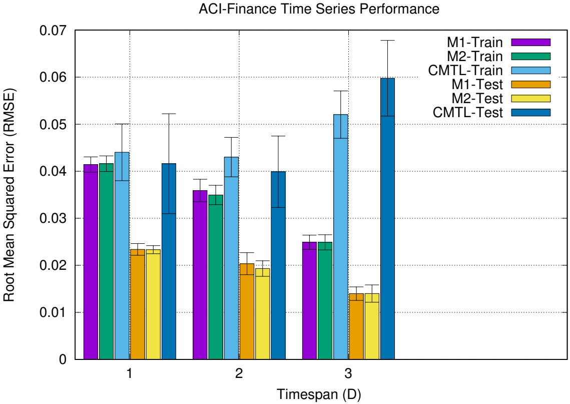


Figure: Performance given by EA, CNE, CMTL for ACI-Finance time series

Results

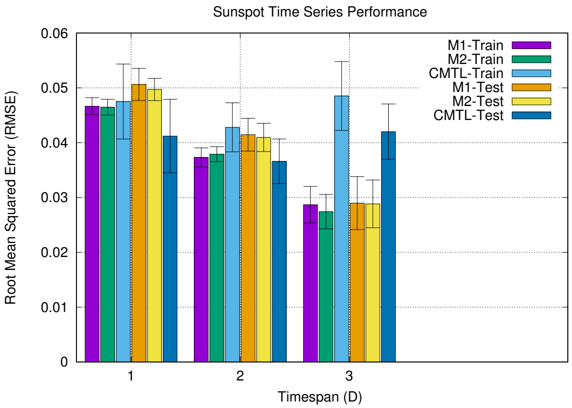


Figure: Performance given by EA, CNE, CMTL for Sunspot time series

Results

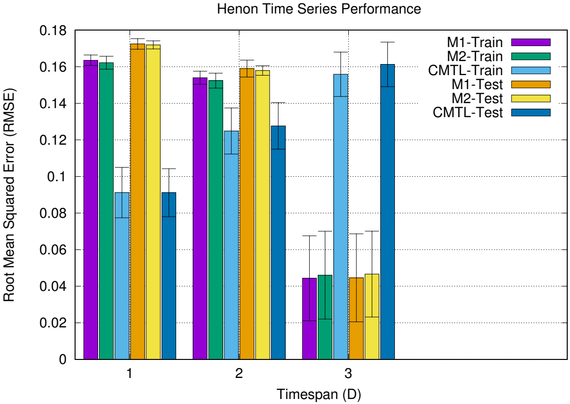


Figure: Performance given by EA, CNE, CMTL for Henon time series

Results

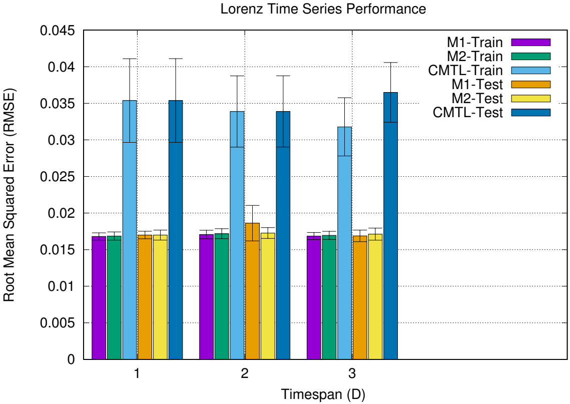


Figure: Performance given by EA, CNE, CMTL for Lorenz time series

Discussion

- Transfer and restore knowledge strategy presented ensured that knowledge in previous subtask is not refined with the rest of the knowledge in the current subtask.
- This is implemented through a restore mechanism where although all the knowledge in the current subtask is refined, those from the previous subtask is restored via memory units.
- It is important to study how the knowledge gained from previous cascaded ensemble contribute to the performance of the architecture. This could give further insight to how knowledge is preserved in biological neural learning when new tasks are learned as done for the case of pattern classification in previous work

Conclusions and Future Work

- The training algorithm employed dynamic optimisation strategy where the knowledge from the foundational subtask was utilized in the subsequent subtasks through transfer learning.
- In general, both instances of the proposed algorithm has scaled better as the size of the timespan increases when compared to CMTL.
- There is scope to develop hybrid methods that could provide a synergy of CMTL with the proposed gradient based approach. Moreover, Bayesian methods could be developed for uncertainty quantification for dynamic time series problems.

Thank You

More information: https://rohitash-chandra.github.io