Multi-task Modular Backpropagation for Dynamic Time Series Prediction

Rohitash Chandra [†¶]
[†] Centre for Translational Data Science, The University of Sydney, NSW 2006, Australia
[¶] School of Geosciences, The University of Sydney, NSW 2006, Australia

***Abstract*—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. Recently, a neuroevolution 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 modular backpropagation that features gradient descent and transfer learning. The method is tested on benchmark chaotic time series problems and compared with its counterparts. The results show that the method can alleviate the problems associated with timely convergence of the neuroevolution approach and provides better performance.**

***Index Terms*—Modular backpropagation, multi-task learning, time series prediction, dynamic time series prediction**

I. Introduction

Time series prediction typically requires data-driven knowledge about past behaviour of the phenomenon in order to make accurate predictions [1], [2], [3]. The past behaviour is defined by a set of temporal data points that are reconstructed using sliding windows which is also known as the embedding dimension [4]. The optimal embedding dimension that fully captures at the behaviour of the time series while alleviating the noise depends on the model and the characteristics of the time series. This needs to be predetermined experimentally. Often, there are a set of values that give good or acceptable prediction performance as experimentally shown in some cases [5], however, only a single value can be used.

Recently, dynamic time series prediction has been introduced that defines robust prediction given with minimum information about past behaviour of the time series [6]. It requires a single model to make prediction given reconstructed time series that features a set of unique values for the embedding dimension rather than a single value. Dynamic time series prediction was motivated from the use of recurrent neural networks (RNNs) for cyclone wind-intensity prediction [5]. It was reported that RNNs trained with a predefined embedding dimension can only generalise well for the same embedding dimension. The embedding dimension can have an impact for the case of cyclones, where the given data was recorded at 6 hour intervals and robust prediction was required [7], [5].

Multi-task learning employs shared representation knowledge for learning related tasks for improved generalisation and efficient training performance [8], [9]. It has been applied to a number of domains that mainly includes pattern classification and has been gaining interest in deep learning for computer vision applications [10], [11]. Neuroevolution considers the use of evolutionary algorithms for training neural networks. Neuroevolution considers fixed and evolving neural network topologies for various types of problems that range from control, time series prediction, and pattern classification [12], [13]. Recently, a neuroevolution strategy, called co-evolutionary multi-task learning (CMTL) has been proposed as a way to address dynamic time series prediction [6]. A variation of CMTL with predictive recurrence network has also been used for multi-step time series prediction [14]. Essentially, CMTL uses a coevolutionary algorithm for training cascaded neural network architectures. The cascade is a set of unique values in the embedding dimension used for reconstruction of the original time series [6] or the set of prediction horizon that require predictive recurrence in the network architecture [14]. Although, CMTL has given very promising performance for the different applications, there are challenges in timely convergence when considering large datasets due to the evolutionary operators. Therefore, back-propagation methods with help of gradient descent can provide alternative methods for training. It is important to ensure that the network modularity [15], [16] is retained during learning, whereby the knowledge from basic network module can be reused with addition of cascades without disruptions to knowledge in other modules. Recently, modular backpropagation based on multi-task learning was proposed for pattern recognition which provided decision given partial information [17]. This motivates the synergy of multi-task and modular learning for dynamic time series prediction.

In this paper, we apply multi-task modular backpropagation for the application of dynamic time series prediction. This provides an alternate approach for training cascaded network architectures for dynamic time series that have typically been done through CMTL. The method is tested on benchmark chaotic time series problems and compared with its counterparts.

The rest of the paper is organised as follows. Section 2 presents multi-task modular back-propagation for dynamic time series prediction. Section 3 presents the results with a discussion. Section 4 presents the conclusions and directions for future research.

II. Background and related work

A. Multi-task learning

Multi-task learning features shared knowledge representation amongst related tasks [8] that have been applied for different types problems that include supervised and unsupervised learning [18], [19], [20], [21]. A major challenge of multi-task learning has been 'negative transfer' which is a concept from behavioural psychology that refers to the interference of the previous knowledge with new learning [22]. The major approach to address it has been through the grouping of tasks where knowledge transfer is performed only within each group [23], [24]. Bakker *et al.* presented a Bayesian approach in which some of the model parameters were shared and others loosely connected through a joint prior distribution and the degree of similarity between the tasks are inferred via the posterior distribution [24]. Zhang and Yeung presented a convex formulation for multi-task metric learning by modelling the task relationships in the form of a task covariance matrix [23]. Furthermore, Zhong *et al.* presented flexible multi-task learning framework to identify latent grouping structures in order to restrict negative knowledge transfer [25].

Multi-task learning has also inspired optimisation methods such as evolutionary algorithms where transfer optimisation exploits solutions from related problems [26]. This motivated the development of evolutionary multi-tasking for training feedforward neural networks for modular knowledge representation [27]. The different subtasks were implemented as different topologies given by the number of hidden neurons with the goal to store knowledge in modules so that perturbation of certain modules do not affect the decision making process as a whole.

In the case of time series, Xu *et al.* applied multi-task regression for forecasting problem [28] which provided a synergy of ensemble forecasting [29] with multi-task regression to estimate the optimal weights for combining the ensembles. As noted earlier, multi-task learning has been also used for multi-step-ahead and dynamic time series prediction [14], [6].

B. Modular learning

Modular neural networks were introduced for visual recognition tasks which produced promising generalisation capability [15]. Modular neural network architectures have been beneficial for hardware implementations [30]. Modular networks feature simultaneous optimization of structure that can be implemented and reconfigured in digital hardware such as field programmable gate arrays [31]. Clune *et al.* presented an architecture where the performance and connection costs were optimised through neuroevolution that improved fully connected neural networks [16]. They have also demonstrated to learn new tasks without forgetting old ones [32] which also have the capability to learn new tasks faster from knowledge of previous tasks. Recently, Watanabe *et al.* presented presented a modular representation of layered neural networks decomposes a trained neural network into multiple small independent networks that can be used for discovering knowledge [33]. Cascade correlation is a modular neural network architecture that automatically develops a network architecture while training [34]. The method has been applied to the to spiral problem that has been not possible using conventional feedforward networks They have also been used for time series prediction [35].

III. Multi-task modular backpropagation for dynamic time series

A. Preliminaries

In order to use neural networks for time series prediction, the original time series is typically reconstructed as a preprocessing step. This is typically done using Taken's theorem [4] which states that the properties of the dynamical system through which the time series is generated can be preserved through phase space reconstruction. The procedure is also known as *embedding* which essentially defines how the time series is broken down into uniformly sized windows. The size of the window is defined by the the embedding dimension (D) which is taken after every time-lag (T). Hence, given an observed time series $x(t)$, an embedded phase space $Y(t) = [(x(t), x(t-T), ..., x(t-(D-1)T)]$ can be generated, where, T is the time delay, D is the embedding dimension (window), $t = (D-1)T, DT, ..., N-1$, and N is the length of the original time series. Note that there are no overlaps between the windows when $T \geq D$. A feedforward neural network with one hidden layer, typically used for time series prediction is defined below.

$$E(y_s) = g(\delta_o + \sum_{j=1}^{J} v_j g\left(\delta_h + \sum_{d=1}^{D} w_{dj} y_{t-d}\right)) \quad (1)$$

where $E(y_s)$ is the one step ahead prediction, δ_o and δ_h are the bias weights for the output o and hidden h layer, respectively. V_j is the weight which maps the hidden layer j to the output layer. w_{dj} is the weight which maps input time series window y_{t-d} to the hidden layer j and g is the activation function. We used a sigmoid unit as the activation function for the hidden and output layer units.

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

$$\Omega_m = [D_1, D_2, ...D_M] \quad (2)$$

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 $m = 1, 2, ..., M$.

$$\Psi_m = x(t), x(t-1), ..., x(t-\Omega_m) \quad (3)$$

B. Multi-task modular backpropagation

We present a cascaded neural network architecture that is trained through multi-task modular back-propagation algorithm for addressing dynamic time series prediction. In other words, the neural network should be able to provide a prediction given any of the input features defined by the set

of values in the embedding dimension which is reconstructed the time series. Hence, the set of input features defines the module of knowledge in a cascaded network architecture that addresses the the dynamic nature of the problem.

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.

$$S_1 = [y_t] \tag{4}$$
$$S_2 = [y_t, y_{t-1}] \tag{5}$$
$$S_n = [y_t, y_{t-1}, ..., y_{t-n}] \tag{6}$$

where $D == S_n$

The optimisation problem is learning the weights of a cascaded neural network architecture. The base problem is the network module that is defined by lowest number of input features and hidden neurons. The weights in the base network are part of larger cascaded network modules that consist of more hidden neurons and input features as shown in Figure 1. This can be viewed as modules of knowledge that are combined for larger subtasks that use knowledge from smaller subtasks as building blocks. The cascaded network architecture can also be viewed as an ensemble of neural networks that feature distinct topologies in terms of number input and hidden neurons as shown in Figure 1. Since we consider one-step-ahead time series problem, one neuron in output layer is used for all the respective modules.

We note that a task of prediction given different sets of input features is not similar to conventional multi-task learning where the output of the network defines the task. In our case, the task is defined by the input of the network, whereas the output of the network is fixed. Hence, we refer to the notion of tasks as subtasks since it is defined by the input of the network. A subtask refers to the problem of learning the given modules that are constructed given timespan S_n in a cascaded manner. Therefore, the input-hidden layer ω_m weights and the hidden-output layer υ_m weights are combined for the respective module m. The base module is given as $\Phi_1 = [\omega_1, \upsilon_1]$. The *subtask* defines the problem of training the respective weights of a module Φ_m given input Ψ_m and a training algorithm. Note that Figure 1 explicitly shows the region (weights) of the network module that are considered for ω_2 and ω_1, respectively. The subtask θ_m is constructed in a cascaded network architecture as follows.

$$\begin{aligned}
&\Phi_1 = [\omega_1, \upsilon_1]; \;\; \theta_1 = (\Phi_1) \\
&\Phi_2 = [\omega_2, \upsilon_2]; \;\; \theta_2 = [\theta_1, \Phi_2] \\
&\quad \vdots \\
&\Phi_M = [\omega_M, \upsilon_M]; \;\; \theta_M = [\theta_{M-1}, \Phi_M]
\end{aligned} \tag{7}$$

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

$$\begin{aligned}
y_1 &= f(\theta_1, \Psi_1) \\
y_2 &= f(\theta_2, \Psi_2) \\
&\vdots \\
y_M &= f(\theta_M, \Psi_M)
\end{aligned} \tag{8}$$

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

$$L_t = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (\hat{y} - y_m)^2} \tag{9}$$

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

The training is done through modular backpropagation that is given in Algorithm 1. The algorithm begins by creating ensembles of cascaded modules where the number of input neurons are defined by timespan S_n from the reconstructed time series. The number of hidden neurons vary in increasing order. For each cascaded module, gradient descent weight update is used for a fixed number of epochs (cycle) until the termination condition has been reached. The approach is called multi-task modular back-propagation since it has been motivated by multi-task learning. Moreover, the gradient descent weight update for the different cascaded modules refers to modular backpropagation. The termination condition can be a minimum loss or maximum number of epochs.

After each cycle of weight updates, the knowledge is transferred from one cascade module into the other with increasing order or size of the cascade (See Line 8 of Algorithm 1). It is an important question whether to preserve knowledge from previous cascaded module or to refine the entire knowledge altogether. Therefore, Line 9 - 11 of Algorithm 1 provides the option that ensures knowledge from previous subtask is retained and only the knowledge in the new cascade module is refined. This will be experimentally evaluated - which will give us further insight as to what practice leads to better performance for the different types of problems considered which is given in the next section. The implementation of Algorithm 1 in Python is given online [1]

IV. Experiments and Results

This section presents an experimental study that evaluates two instances of the proposed algorithm for dynamic time series prediction.

A. Benchmark Chaotic Time Series Problems

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.

[1] https://github.com/rohitash-chandra/multitask-modular-backpropagation

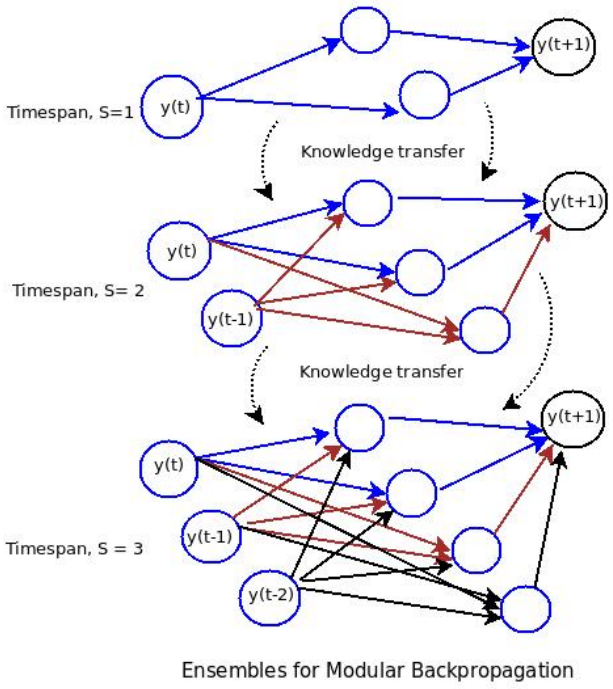

Fig. 1: Note that the knowledge module m_1 for Timespan 1 employs a network topology with 2 hidden neurons. The rest of the tasks add extra input and hidden neurons.

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

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 $T = 2$. The simulated and real-world time series were scaled in the range [0,1]. Further details of each of the time series problem is given in our previous work [6].

B. Results

The results in Figure 2 - Figure 8 show the performance of the proposed method that implement two scenarios from Algorithm 1. Method 1 (M1) is transfer knowledge only while Method 2 (M2) is transfer and restore knowledge. The results further compares the performance with co-evolutionary multi-task learning (CMTL) which was proposed for dynamic time series prediction [6]. The performance of the training and testing dataset given by respective methods is given for the different subtasks that resemble the embedding dimension (Timespan (D)).

The results show that both instances of the proposed algorithm show better results than CMTL for Mackey-Glass and Lorenz problems. The same trend is shown for the Rossler time series. Note that these are simulated time series problems that do not contain noise. The Henon problem shows a distinctly different trend where the performance of the proposed algorithm is much poorer when compared to CMTL for the shorter timespan ($D = 3$ and $D = 5$) The performance is better than CMTL, only for $D = 7$.

The proposed algorithm achieves slightly better performance when compared with CMTL for ($D = 3$ and $D = 5$) in the Sunspot problem. The performance is much better for $D = 7$. There is a similar trend in the Lazer problem. The ACI-Finance problem provides distinctly different performance where the proposed algorithm gives poorer performance when compares to CMTL for all the cases.

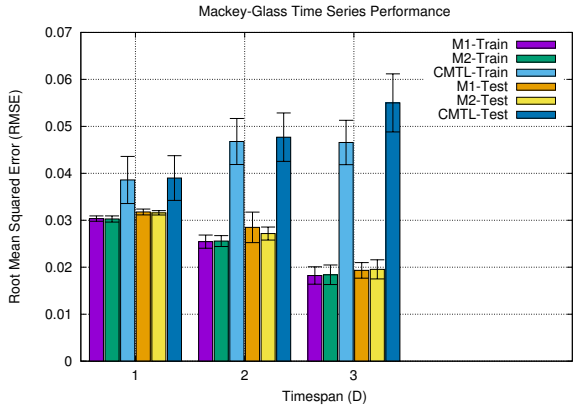

Fig. 2: Performance given by EA, CNE, CMTL for Mackey-Glass time series

The results in 9 and 10 show the performance of the algorithm with 7 subtasks that are composed with shortest timespan.

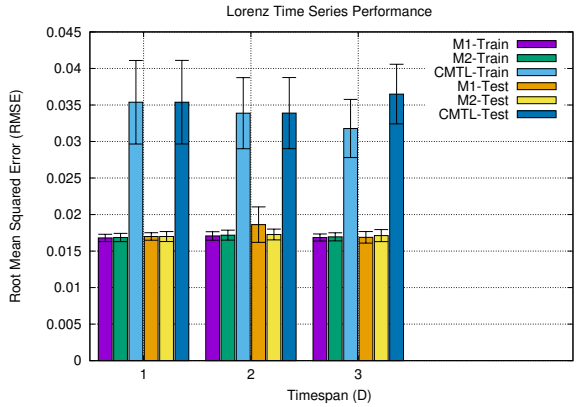

Fig. 3: Performance given by EA, CNE, CMTL for Lorenz time series

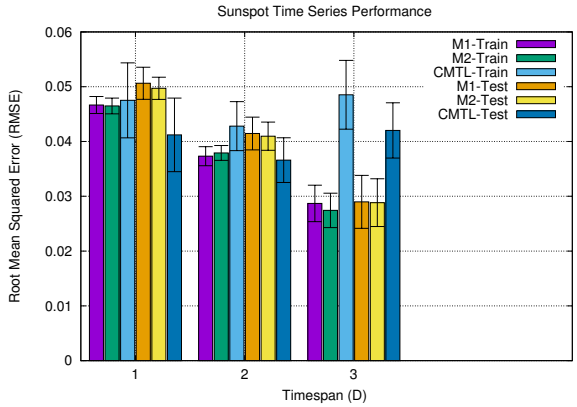

Fig. 6: Performance given by EA, CNE, CMTL for Sunspot time series

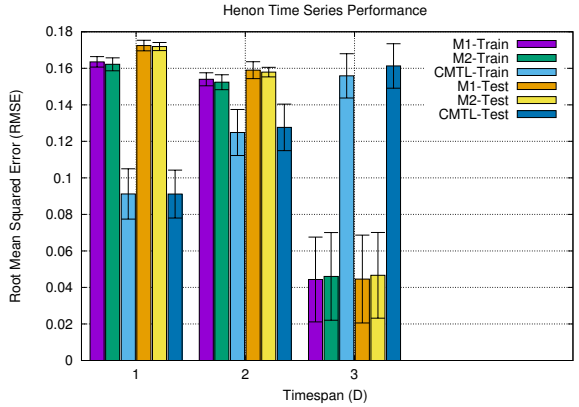

Fig. 4: Performance given by EA, CNE, CMTL for Henon time series

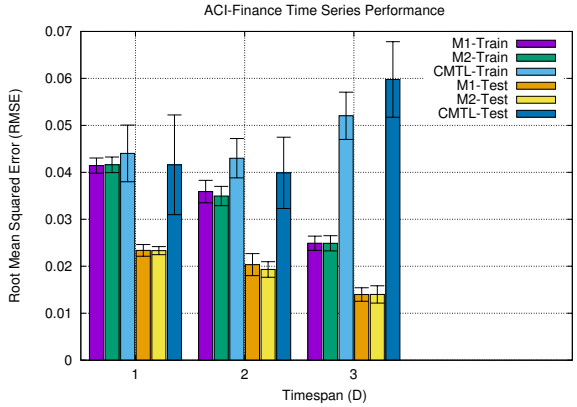

Fig. 7: Performance given by EA, CNE, CMTL for ACI-Finance time series

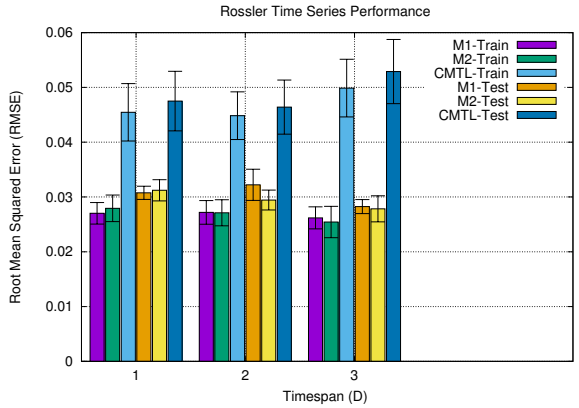

Fig. 5: Performance given by EA, CNE, CMTL for Rossler time series

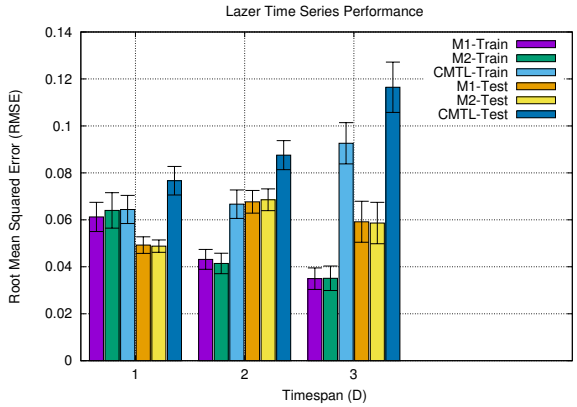

Fig. 8: Performance given by EA, CNE, CMTL for Lazer time series

V. 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. The results in general show that there is not much difference in performance between the two different instances of the algorithms (M1 vs M2). This suggests that the implementation to transfer and

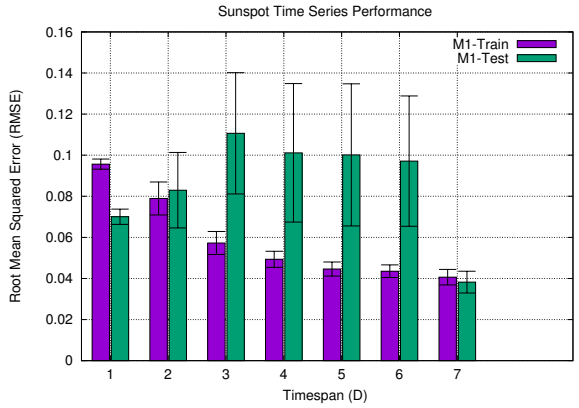

Fig. 9: Performance given by EA, CNE, CMTL for Sunspot time series

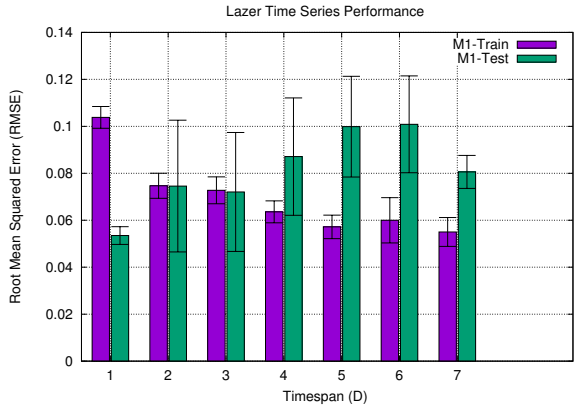

Fig. 10: Performance given by EA, CNE, CMTL for Lazer time series

restore knowledge has little effect when compared to transfer knowledge only. This could be due to the fact that there is not a substantial difference of knowledge in between the subtasks. Given substantial difference, there would be greater difference in performance when the knowledge transferred is further refined. 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 [17].

In general, both instances of the proposed algorithm has scaled better as the size of the timespan increases when compared to CMTL. It is also observed that the generalization performance deteriorates after timespan greater than two for the Sunspot problem shown in Figure 9. This tread is similar for the Lazer problem for the timespan greater than three as shown in Figure 10.

VI. Conclusions and Future Work

We presented an cascaded network architecture motivated by multi-task learning for dynamic time series prediction. We used an ensemble approach for implementation of modular backpropagation algorithm that decomposed the dynamic time series problem as subtasks that have interdependencies due to shared knowledge representation via multi-task learning. The training algorithm employed dynamic optimisation strategy where the knowledge from the foundational subtask was utilized in the subsequent subtasks through transfer learning.

The results show that the performance of the proposed method has been improved for most of the problems, in particular, the synthetic time series problems that do not contain noise. In case of the real-world problems, we observe that the proposed method improves the performance of CMTL in some of the cases only. This reflects the properties of evolutionary learning in comparison with gradient based learning for real-world problems that featured noise.

The challenge in the future is to improve the performance for such problems so that the advantage of gradient based learning can be fully utilised. Therefore, 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.

References

[1] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica: Journal of the Econometric Society*, pp. 357–384, 1989.

[2] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks," *Neural networks*, vol. 5, no. 6, pp. 961–970, 1992.

[3] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[4] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.

[5] R. Deo and R. Chandra, "Identification of minimal timespan problem for recurrent neural networks with application to cyclone wind-intensity prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, July 2016, p. In Press.

[6] R. Chandra, Y. Ong, and C. Goh, "Co-evolutionary multi-task learning for dynamic time series prediction," *CoRR*, vol. abs/1703.01887, 2017. [Online]. Available: http://arxiv.org/abs/1703.01887

[7] (2015) JTWC tropical cyclone best track data site. [Online]. Available: http://www.usno.navy.mil/NOOC/nmfc-ph/RSS/jtwc/

[8] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[9] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 615–637, 2005.

[10] H. Zheng, X. Geng, D. Tao, and Z. Jin, "A multi-task model for simultaneous face identification and facial expression recognition," *Neurocomputing*, vol. 171, pp. 515 – 523, 2016.

[11] T. Zeng and S. Ji, "Deep convolutional neural networks for multi-instance multi-task learning," in *Data Mining (ICDM), 2015 IEEE International Conference on*, Nov 2015, pp. 579–588.

[12] P. Angeline, G. Saunders, and J. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *Neural Networks, IEEE Transactions on*, vol. 5, no. 1, pp. 54 –65, jan 1994.

[13] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep 1999.

[14] R. Chandra, Y. Ong, and C. Goh, "Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction," *Neurocomputing*, vol. 243, pp. 21–34, 2017. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.02.065

[15] B. L. Happel and J. M. Murre, "Design and evolution of modular neural network architectures," *Neural Networks*, vol. 7, no. 67, pp. 985 – 1004, 1994, models of Neurodynamics and Behavior. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608005801558

[16] J. Clune, J.-B. Mouret, and H. Lipson, "The evolutionary origins of modularity," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 280, no. 1755, 2013.

[17] R. Chandra, "Multi-task modular backpropagation for feature-based pattern classification," in *Neural Information Processing - 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part VI*, 2017, pp. 558–566.

[18] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Dec. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1046920.1194905

[19] L. Jacob, J. philippe Vert, and F. R. Bach, "Clustered multi-task learning: A convex formulation," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 745–752. [Online]. Available: http://papers.nips.cc/paper/3499-clustered-multi-task-learning-a-convex-formulation.pdf

[20] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning shared structures from multiple tasks," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 137–144. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553392

[21] J. Chen, J. Liu, and J. Ye, "Learning incoherent sparse and low-rank patterns from multiple tasks," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 4, pp. 22:1–22:31, Feb. 2012. [Online]. Available: http://doi.acm.org.ezlibproxy1.ntu.edu.sg/10.1145/2086737.2086742

[22] O. Griffiths, A. M. Johnson, and C. J. Mitchell, "Negative transfer in human associative learning," *Psychological science*, vol. 22, no. 9, pp. 1198–1204, 2011.

[23] Y. Zhang and D.-Y. Yeung, "Transfer metric learning by learning task relationships," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 1199–1208. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835954

[24] B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1162/153244304322765658

[25] S. Zhong, J. Pu, Y.-G. Jiang, R. Feng, and X. Xue, "Flexible multi-task learning with latent task grouping," *Neurocomputing*, vol. 189, pp. 179 – 188, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216000035

[26] A. Gupta, Y. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2016.

[27] R. Chandra, A. Gupta, Y.-S. Ong, and C.-K. Goh, "Evolutionary multi-task learning for modular knowledge representation in neural networks," *Neural Processing Letters*, Oct 2017. [Online]. Available: https://doi.org/10.1007/s11063-017-9718-z

[28] J. Xu, P. N. Tan, and L. Luo, "Orion: Online regularized multi-task regression and its application to ensemble forecasting," in *2014 IEEE International Conference on Data Mining*, Dec 2014, pp. 1061–1066.

[29] M. Leutbecher and T. Palmer, "Ensemble forecasting," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3515 – 3539, 2008, predicting weather, climate and extreme events. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999107000812

[30] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing*, vol. 74, no. 13, pp. 239 – 255, 2010, artificial Brains.

[31] S.-W. Moon and S.-G. Kong, "Block-based neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 307–317, Mar 2001.

[32] K. O. Ellefsen, J.-B. Mouret, and J. Clune, "Neural modularity helps organisms evolve to learn new skills without forgetting old skills," *PLoS Comput Biol*, vol. 11, no. 4, pp. 1–24, 04 2015.

[33] C. Watanabe, K. Hiramatsu, and K. Kashino, "Modular representation of layered neural networks," *Neural Networks*, vol. 97, no. Supplement C, pp. 62 – 73, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608017302319

[34] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 524–532. [Online]. Available: http://papers.nips.cc/paper/207-the-cascade-correlation-learning-architecture.pdf

[35] R. S. Crowder, "Predicting the mackey-glass time series with cascade-correlation learning," in *Proc. 1990 Connectionist Models Summer School*. Carnegie Mellon Univ., Pittsburgh, PA, 1990, pp. 117–123.