Classify Anger Veracity from Pupillary Responses: Exploring Different Methods on Time Series Prediction using Casper and LSTM

Cai Yang
Research School of Computer Science, Australian National University
u6625166@anu.edu.au

Abstract. This paper studies the problem of classifying anger veracity based on participants' pupillary responses. A constructively learning algorithm called Casper with Resilient Back Propagation and its variants are applied to the statistical summary of the data. Meanwhile, A Long Short-Term Memory based model is applied directly to the original time series data. Genetic algorithm is used when selecting values of hyperparameters of the two models. The experiments have shown both Casper and LSTM-based model are capable of predicting anger veracity with accuracy over 70%, which is higher than the verbal responses from participants. In the end, we show some worthwhile future work.

Keywords: Neural Network. Casper. RPROP. SARPROP. ReSARPROP. LSTM. Time Series. Genetic Algorithm.

1 Introduction

Interactions with face expressions have become rather compelling in recent years, A face showing emotion reflects the internal mental state of the displayer of the emotion and is arguably also an attempt to influence the internal mental state of the observer of the displayed emotion [7]. For example, a smile is a complex and multi-purpose facial display that conveys not only the meaning of happiness [1], but can also be identified as frustration, depression, empathetic, surprise, polite disagreement, pain and even more [2]. Similarly, anger could also be acted or real. Detecting the veracity of such emotions have become intense and popular with the help of modern deep learning techniques.

Emotions are likely to be reflected by some physiological signals, which usually vary with time. Hence the dataset recording such signals are likely to be time-series format. There are usually three typical ways of dealing with time series data: distanced-based, statistical-based methods and Deep-learning based methods. The key idea of distance-based methods is to construct effective distance metrics or kernel functions to measure similarity, which can be used in Support Vector Machine and K Nearest Neighbors. Statistical-based methods require us to manually extract statistical features from time series data such as mean, standard deviation and skewness. Deep-learning based methods take time series data directly as input and produce results based on the task goal.

In this report, experiments are designed using both statistical-based methods and Deep-learning based methods to show which one could produce better results on anger veracity classification in terms of pupillary responses. Moreover, a vanilla neural network and Casper are used to train on statistical summary while an LSTM based model is use train on time series data.

One major difficulty with training models is the choice for hyperparameters. Deterministic grid search on a predefined sequence of values may not yield satisfying results since there is no clear guideline which values should be included in the search. Meanwhile, the time complexity grows fast with the number of hyperparameters which makes the computation time-consuming. In this case, genetic algorithm was used in hyperparameter tuning, which can produce promising results.

This paper is organized as follows. Section 2 shows some related work on emotions classification and time series predictions. Details on models and genetic algorithms used in the experiments are presented in section 3. The data preprocessing method is covered in section 4. The details of experiments are presented in section 5. Results and discussions from experiments are in section 6. Conclusions and future work are proposed in the section 7. It is also worth mentioning that the purpose of the paper is not to beat any state-of-art model but rather to explore and compare different techniques that can be used to perform classification on time series data.

2 Related Work

2.1 Emotion Classification

Recent research and experiments have been performed a lot on emotion classification tasks to detect real and fake emotions from participants. [5, 6] conducted experiments to classify whether observers' smiles are real or fake. The former was based on observers' GSR (Galvanic Skin Response) while the latter was based on more physiological signals such as PR (Pupillary Response), BVP (Blood Volume Pulse), and GSR (Galvanic Skin Response).

Experiments from [7] collected participants' verbal responses and their pupillary responses in viewing two types of anger stimuli and use them to classify anger veracity. Their results have showed significant improvement from the accuracy of human verbal responses (60%) to the accuracy of ensembles of machine classifiers trained on pupillary responses (95%). The experiments performed in this paper continue with the dataset in [7].

2.2 Time Series Data Prediction

Time series is a sequence of data points in a given time interval. It has been commonly used in many fields of studies such as weather forecasting and psychological signals. Time series can be both univariate, where the value of a single variable is collected over time, or multivariate, where values from multiple variables are collected [14]. The dataset used in this paper falls into the former category.

Many methods have been developed over past years to deal with time series data, including distance-based and deep learning models and other techniques. Dynamic Time Warping (DTW) with K Nearest Neighbour has been used in many researches recently, which indicates it is one of the best distance-based methods [14]. Multi-Channel Deep Convolutional Neural Network (MC-DCNN), proposed in 2014, is a deep learning model that first extracts latent features from and then perform classification using an MLP [17]. There are also researches that adapted AlexNet and ResNet on solving time series predictions recently. A detailed analysis on different deep learning models can be found in [16].

3 Methodology

2.1 Casper

Cascade network algorithm employing Progressive RPROP, Casper in short, was originally proposed by [4] to overcome the weaknesses of Cascor such as incapability of generalization on regression and classification tasks and excessively large architecture produced due to weights freezing.

Casper constructs cascade networks in a similar manner to Cascor, which adds a single hidden neuron once a time during training phase. According to [4], the criterion for installing a new hidden neuron is that either the training loss has decreased at least 1% during a $15+P\times N$ time period, where P is a hyperparameter and N is the current number of hidden neurons or the number of training iterations has reached a predefined threshold. It has been shown Casper was able to reduce the number of neurons needed to train a neural network and improve the generalization on different types of tasks.

RPROP has been shown to be one of the faster convergent back propagation algorithms and the adaption of it does not reply on the magnitude of the gradient but only the sign, which allows the step size to be adapted without having the size of the gradient interfere with the adaptation process [8]. In details, three different learning rates are adopted based on the position of weights. The idea is to adopt higher learning rate for weights associated with edges connected to the new neuron, a moderate learning rate for those connected from new neuron to the output neuron and a lower learning rate for the rest.

A figure of Casper architecture is displayed below in Fig 1.

2.2 Long Short-Term Memory

Long short-term memory (LSTM for short) is an important achievement in the area of deep learning. It has received a lot attention due to the broad applications of LSTM-based networks such as machine translation, music generation and language modeling.

Standard recurrent neural networks (RNN) suffer from the issue of vanishing or exploding gradient and are hard to train on tasks that require long-term dependencies. LSTM is a special type of RNN that deals with such problems by maintaining different types of gates: input gate, forget gate and output gate. Meanwhile, nonlinear units used in RNN have been replaced by memory blocks in LSTM.

Input gates control what portion of the new input goes into the current hidden unit based on previous content of memory cell, hidden unit and the current input. Similarly, output gates control the content coming out of the memory cell and into the current hidden unit. Forget fates control when the portion of information will be forgotten. Such architectures allow a better control over the information flow passing through the memory block and hence LSTM can obtain better results on long-term dependency tasks.

A figure of LSTM layer is displayed below in Fig 2.

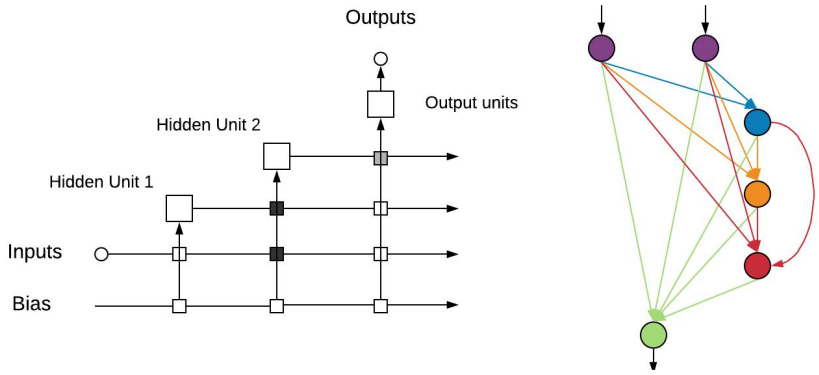

Fig 1. An illustration of Casper. Hidden unit 2 is the new-installed neuron. The boxes with dark color represent weights with learning rate L1. The boxes with shadow color represents learning rate L2 and the rest boxes stand for learning rates L3. L1 > L2 > L3. For the right one, top 2 represent input units while bottom one is output unit. The rest are hidden units.

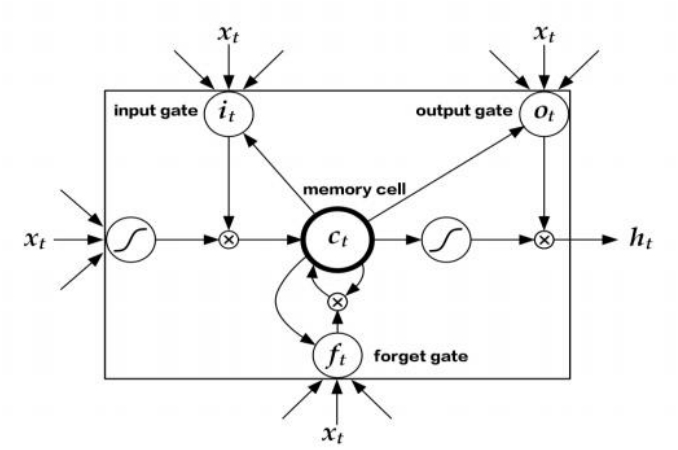

Fig 2. An illustration of LSTM structure. x_t represents the input. h_t is the hidden output. c_t is the memory block. i_t , f_t and o_t are input gates, forget gates and output gates respectively.

2.3 Genetic Algorithm

Genetic Algorithm is a meta-heuristic that has been commonly used in areas such as optimization and operations research. It borrows the idea from biological behaviors such as selection, crossover and mutation. It is able to produce some high-quality solutions. In genetic algorithm, bit string representation is adopted, which make it easy and fast to produce results. The general process for genetic algorithm is displayed below in Fig 3.

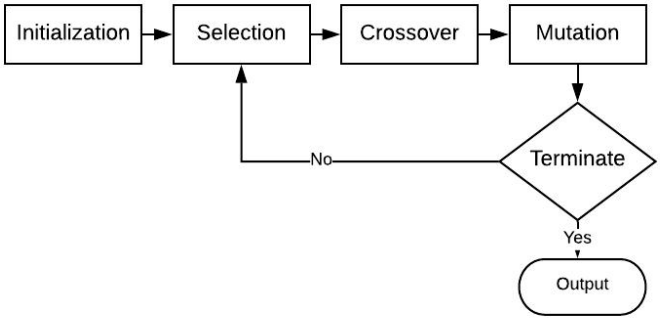

Fig 3. A process showing how genetic algorithm works in general. The evaluation of fitness values has been incorporated into selection part.

The algorithm starts with an randomly initialized population and then evaluate the fitness of each chromosome, where fitness is defined as average testing accuracy in a 5-fold cross validation in this task. Probability of a chromosome is selected is proportional to its corresponding fitness value. Crossover operators are performed on the parents selected to generate offspring, during which mutation might happen with small probability to increase the diversity of the population. Termination conditions are usually some predefined threshold on fitness value or number of generations. The algorithm terminates once the condition is satisfied.

3 Data Preprocessing

The statistical data was obtained by sticking to the methods used in [6]. The missing data caused by accidental eye blinks was reconstructed using a cubic spline interpolation and smoothed using a 10-point Hann moving window average filter respectively. Six statistical features are extracted, including mean, standard deviation, means of absolute values of the first and second order differences of the process signals and its two major principal components. These features are easy to compute and cover the typical characteristics of the signals. All the features are normalized into range 0 to 1 to avoid dominant data.

The times series data was first processed using the same method above. All the processed signals are collected together as the final dataset. There are 10 records of empty data for some participants on several videos and they were decided to be dropped since the number of empty data is small compared to the overall number of data. Class balance was still maintained after dropping them.

4 Experiments

All training phases are performed on training and testing dataset through a 5-fold cross validation to make sure the results are more reliable. Meanwhile, all the experiments were performed using Python under Pytorch with version *1.5.1* and scikit-learn with version *0.22.2.post1*.

4.1 Vanilla Neural Network

A one hidden layer vanilla neural network is created in this experiment as a baseline model. It has 28 hidden units with Logistic Sigmoid activation function. The neural network is trained 400 epochs with *Adam* optimizer along with learning rate of 0.001. All the hyperparameters used here are chosen through cross validation instead of genetic algorithms since this model serves as a baseline model.

A plot of cross entropy loss during training phase of one fold is displayed below in Fig 4.

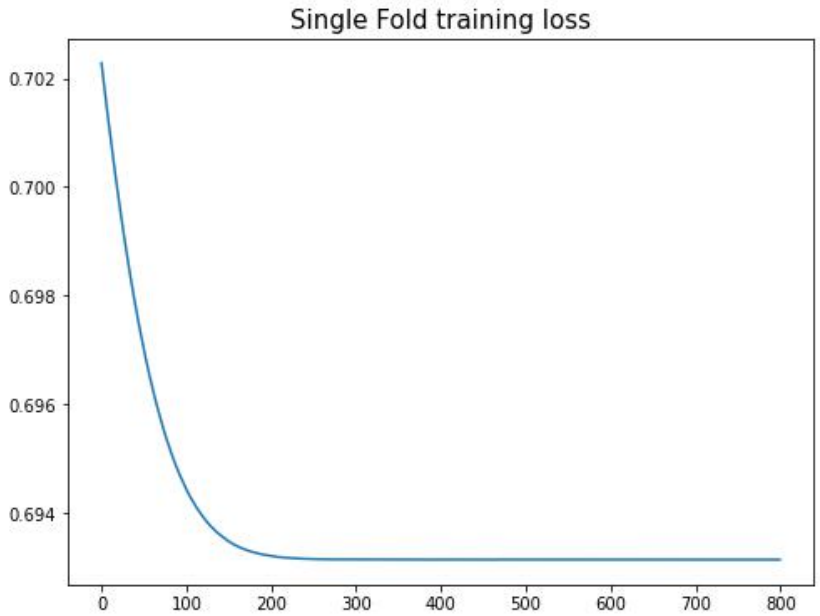

Fig 4. A plot of training loss of neural network. The X-axis is number of iterations while the Y-axis is cross entropy loss.

4.2 Casper

Hyperbolic tangent function is used as activation function between two hidden neurons and logistic sigmoid function is used before output neurons which has been tested to show best performance in general.

The maximum number of hidden units, the value of P and learning rates are selected through genetic algorithm since in general these hyperparamters will affect performance of Casper more than others. The rest hyperparameters such as multiplicative increase and decrease factors (etas), minimal and maximal allowed step sizes (deltas) are set to their default values: (0.5, 1.2) and (10^{-6} , 50) respectively. The best fitness value obtained in each generation is displayed below, where 20 generations in total are run.

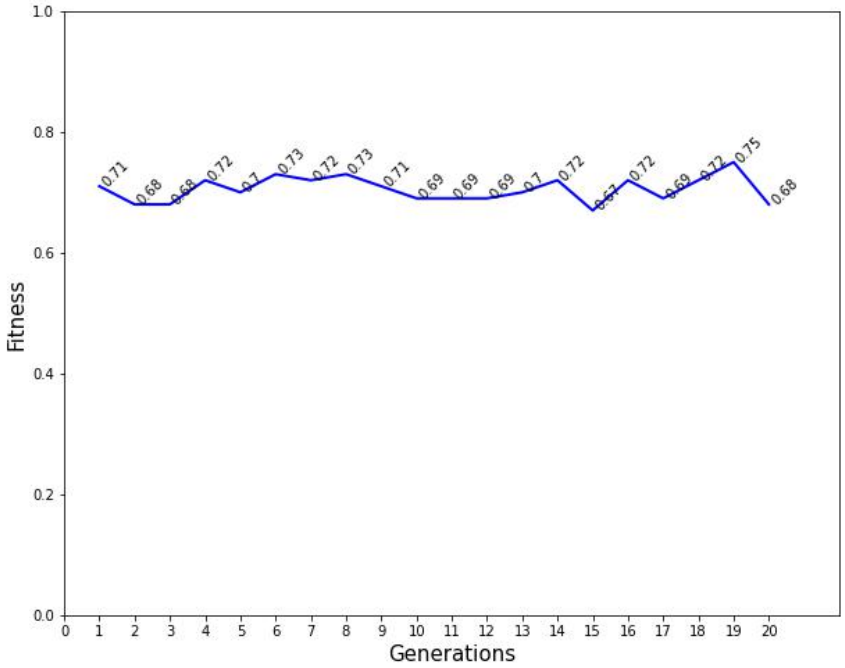

Fig 5. A plot of best fitness value obtained through a 20-generation genetic algorithm with Casper. The horizontal axis is the number of generation while the vertical axis represents the fitness value (average testing accuracy).

The best fitness value obtained in a 20-generation training is 0.75, which corresponds to setting the maximum number of hidden neurons to 15, P to 2, learning rates to 0.07, 0.0675 and 0.004 respectively.

However, even if the model is trained using the best hyperparameters obtained from genetic algorithm, one of the major issues remaining is the difficulty to get out of local minimum. If we plot the training loss against iterations of one fold, which is displayed below in Fig 6, we can see the model is stuck in plateau for long time and unable to get out.

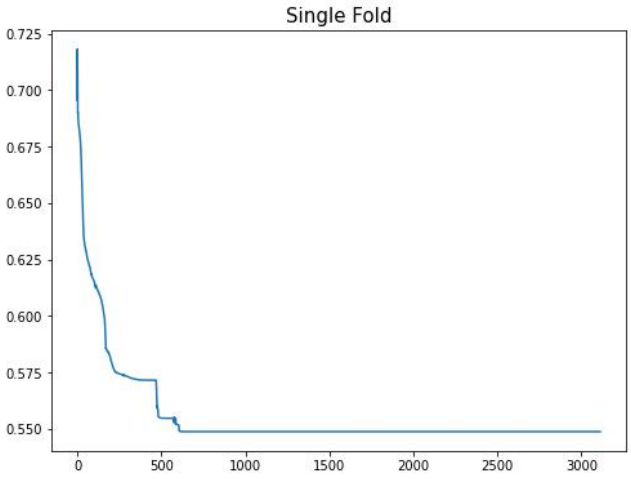

Fig 6. Training loss using Casper with RPROP. Horizontal axis is the number of iterations and vertical axis is the cross entropy loss.

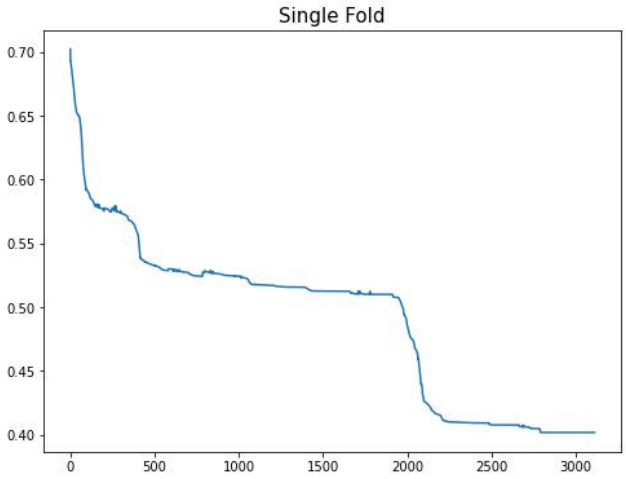

Fig 7. Training loss using Casper with SARPROP. Horizontal axis is the number of iterations and vertical axis is the cross entropy loss.

4.2.1 Simulated Annealing RPROP

To overcome the problem that model is stuck in local minimum in last section, Simulated Annealing RPROP (SARPROP) [8] is further used.

The idea of SARPROP is based on simulated annealing, which is a meta-heuristic widely used in areas such as operations research. An extra hyperparameter, T, is introduced which serves a role similar to temperature. A new value SA, which is defined as $SA = 2^{-T \bullet Epoch}$, serves the purpose of adding noise to weights update and adding penalty to weights decay during gradient calculation. A detailed algorithm could be found in [8].

All the hyperparameter values are set to be the same as Casper and the value of T is set to be 0.05. The training loss is plotted in Fig 7. As we can see, the model is able to get out of local minimum with the help of parameter T. However, it should be noted that there is still no guarantee that SARPROP will converge to a good local minimum, only that the likelihood of such is increased. It may well be the case that the effect of noise and weight decay will push SARPROP away from a good solution [8].

4.3 LSTM

The LSTM-based model consists of a single layer LSTM and a three-layer feed forward block. The LSTM layer takes the pupillary responses directly as its input, whose output will be fed into the linear block. Logistic Sigmoid function is used as the activation function in the end since the model is trained to solve an classification task.

One major issue when dealing with the pupillary responses is that, the length of time series data will change over videos since those videos used an stimuli are of different lengths. In other words, the model should be able to deal with variable length time series data. However, variable length time series prediction is still an open research topic and there is no rule of thumb on how to process it to make the results best. In this experiment, all the shorter time series data are padded with 0 to match the length of the longest sequence. The value of 0 serves a purpose of masking the current timestep and will be skipped during training.

Same as Casper, hyperparamters such as the number of hidden dimensions, epochs for training and learning rates are also selected through genetic algorithm. The best fitness values obtained during 20 generations is displayed below. The highest fitness, 0.71, corresponds to 51 hidden dimensions, 638 training epochs and a learning rate of 0.0647.

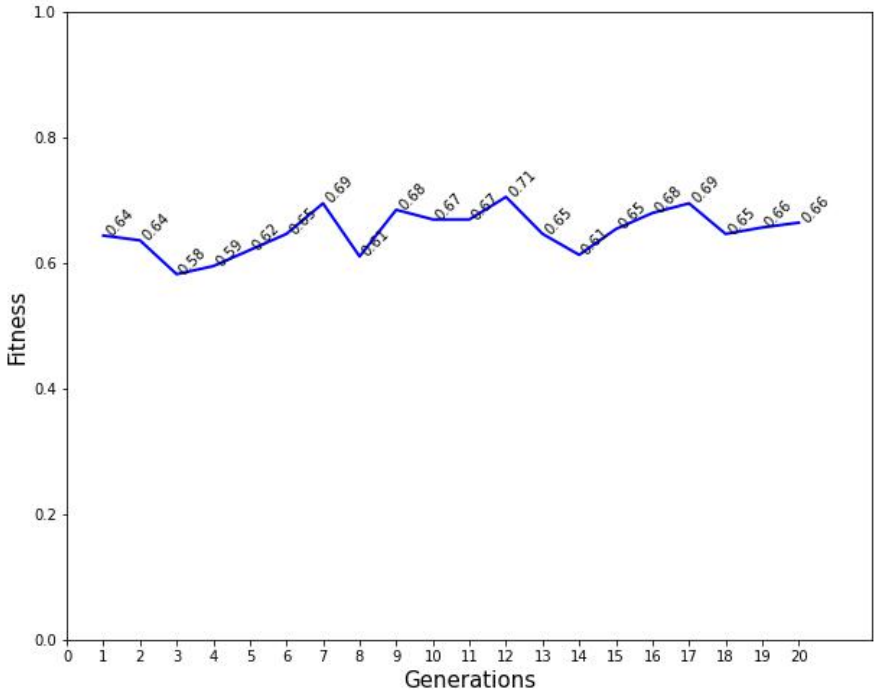

Fig 8. A plot of best fitness value obtained through a 20-generation genetic algorithm with LSTM-based model. The horizontal axis is the number of generation while the vertical axis represents the fitness value (average testing accuracy).

5 Results and Discussions

In this section, results obtained from previous models are displayed and compared below in Table 1 along with the results obtained from original paper [7].

	RPROP	SARPROP	NN	LSTM	Verbal	Original
Accuracy	72.25%	81.25%	57.50%	71.00%	60.00%	95.00%
Testing Loss	0.5094	0.4828	0.6677	0.5086	-	-

Table 1. The table collects all the results from experiments on previous models. The accuracy and testing cross entropy loss were averaged in the end. The results from [7] are in the two rightmost columns, where Verbal stands for verbal responses from participants and Original represents the methods that authors used in [7].

The experiments have shown that both Casper and LSTM can achieve good results on classifying the anger veracity. Moreover, a simulated annealing based training approach can help Casper jump out of local minimum and produce a better result. Although the accuracy obtained is non-competitive to [7], they are already better than the verbal responses from participants. Meanwhile, the results have shown that deep learning models have the potential to outperform humans in terms of emotions veracity classifications.

6 Conclusion and Future work

This paper explored different methods and models on classifying anger veracity based on time series data. Casper models with different optimizers were trained on statistical summary of the data while an LSTM-based model was directly trained on time series data. The values of key hyperparamters are selected through genetic algorithm. The comparison between them has been made and discussed. The corresponding issues of each model are also mentioned in its corresponding section. In the end, we can draw the conclusion that both Casper and LSTM-based models are able to outperform human verbal responses on this classification task.

Future work can be continued in many directions. Further Experiments on psychological data based emotion predictions could be carried on to make a thorough comparison between different models and human responses. For example, a similar experiment can be performed to detect the sadness veracity of participants.

As mentioned in section 2.2, numerous deep learning models have been proposed and improved on solving time series prediction tasks. Deep learning approaches can be divided into tow major groups: generative and discriminative [16]. Generative models are usually set up as unsupervised tasks with purpose of learning the best mapping from time series to a latent space, which can help to represent the time series data before training. One typical example of such model is Auto-Encoder. Discriminative models usually take raw data directly as its input and output corresponding probabilities, which are trained on a end-to-end basis. Moreover, it is also worthwhile designing models that are able to deal with variable length time series data. Such models have the potential to generalize well on different types of tasks since they are able to capture the dynamics brought by time series data.

Further research on applying other variants of Casper models could also be worthwhile. For example, Layered_Casper [12], which adds a single hidden neuron in a layer up to a limit rather than stack hidden units together, has been proved to be able to reduce the number of parameters needed to solve a task compared to Casper.

References

[1] Ekman, P., Davidson, R.J., and Friesen, W.V., "The Duchenne smile: Emotional expression and brain physiology: II", J Personality and Social Psychology (58), 1990, pp. 342—353.

[2] Hoque, M., Morency, L.P., and Picard, R.W., "Are You Friendly or Just Polite?Analysis of Smiles in Spontaneous Face-to-Face Interactions", Affective Computing and Intelligent Interaction, LNCS (6974), 2011, pp. 135—144.

[3] Fahlman, S.E., and Lebiere, C., "The cascade-correlation learning architecture", In Advances in Neural Information Processing II, Touretzky, Ed. San Mateo, CA: Morgan Kauffman, 1990, pp. 524-532.

[4] Treadgold, N.K. and Gedeon, T.D., "A Cascade Network Employing Progressive RPROP", Int. Work Conf. on Artificial and Natural Neural Networks, 1997, pp. 733-742.

[5] Hossain, M.Z. and Gedeon, T.D., "Observer's Galvanic Skin Response for Discriminating Real from Fake Smiles", Australian Conference on Information Systems, Wollongong, 2016.

[6] Hossain, M.Z. and Gedeon, T.D., "Classifying Posed and Real Smiles from Observers' Peripheral Physiology", 11th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth '17), EAI Conference Series, Barcelona, Spain, 2017.

[7] Chen, L., Gedeon, T.D., Hossain, M. Z., and Caldwell, S. (2017, November). "Are you really angry?: detecting emotion veracity as a proposed tool for interaction", In Proceedings of the 29th Australian Conference on Computer-Human Interaction (pp. 412-416). ACM.

[8] Treadgold, N.K. and Gedeon, T.D., "Simulated Annealing and Weight Decay in Adaptive Learning: The SARPROP Algorithm", IEEE Transactions on Neural Networks, Vol. 9, No. 4, 1998.

[9] Riedmiller M. and Braun H., "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in Proc. ICNN 93, San Francisco, CA, 1993, pp. 586–591.

[10] Treadgold, N.K. and Gedeon, T.D., "Extending and benchmarking the CasPer algorithm", Australian Joint Conference on Artificial Intelligence, 1997, pp. 398-406.

[11] Treadgold, N.K. and Gedeon, T.D., "Exploring architecture variations in constructive cascade networks", IEEE World Congress on Computational Intelligence, IEEE International Joint Conference on Neural Networks Proceedings, vol. 1,pp. 343-348, May, 1998.

[12] Shen, T. and Zhu, D., "Layered_CasPer: Layered Cascade Artificial Neural Networks", IEEE World Congress on Computational Intelligence, 2012.

[13] Elsworth, S. and Guttel, S., "Time Series Forescasting Using LSTM Networks: A Symbolic Approach", arXiv preprint arXiv: 2003.05672v1, 2020.

[14] Karim, F., Majumdar, S., Darabi, H. and Harford, S., "Multivariate LSTM-FCNs for Time Series Classification", arXiv preprint arXiv: 1801.04503v2, 2019.

[15] Lei, Y. and Wu, Z., "Time Series Classification Based on Statistical Features", EURASIP Jounal on Wireless Communications and Networking, 2020, Article Number: 46.

[16] Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P.A., "Deep Learning for Time Series Classification: A Review", Data Mining and Knowledge Discovery (33), 2019, pp.917-963.

[17] Zheng Y., Liu Q., Chen E., Ge Y., Zhao J.L., "Time Series Classification Using Multi-channels Deep Convolutional Neural Networks", In: International Conference on Web-Age Information Management, Springer, 2014, pp. 298–310.