Early Stopping Criteria to Counteract Overfitting in Genetic Programming

Clíodhna Tuite, Alexandros Agapitos, Michael O'Neill, and Anthony Brabazon
Financial Mathematics and Computation Cluster
Natural Computing Research and Applications Group
Complex and Adaptive Systems Laboratory
University College Dublin, Ireland
cliodhna.tuite@gmail.com, alexandros.agapitos@ucd.ie, m.oneill@ucd.ie,
anthony.brabazon@ucd.ie

ABSTRACT

Early stopping typically stops training the first time validation fitness disimproves. This may not be the best strategy given that validation fitness can subsequently increase or decrease. We examine the effects of stopping subsequent to the first disimprovement in validation fitness, on symbolic regression problems. Stopping points are determined using criteria which measure generalisation loss and training progress. Results suggest that these criteria can improve the generalistion ability of symbolic regression functions evolved using Grammar-based GP.

Categories and Subject Descriptors: I.2.8 [Problem Solving, Control Methods, and Search]: Heuristic methods

General Terms: Measurement, Experimentation

Keywords: grammatical evolution, symbolic regression, overfitting

1. OVERFITTING AND EARLY STOPPING

Overfitting is a problem which can arise in machine learning and optimisation techniques such as Genetic Programming (GP)[7, 1]. A model is described as overfitting the data if, while having a good fit on the training data, there exists an alternative model which fits the data as a whole better, despite having a worse fit on the training data [4].

Early stopping is a method used to counteract overfitting [2], whereby training is stopped when overfitting begins to take place [3]. It has been critisized in [8], which examines the use of early stopping when training a neural network. According to Prechelt, in most cases the validation set error doesn't monotonically improve during the early stage of training, before monotonically disimproving after overfitting takes place. He states that real validation error curves almost always have more than one local minimum. The question then becomes, when should early stopping take place? Previous work [9] indicates that early stopping should not necessarily take place the first time validation set error disimproves during a symbolic regression run using Grammar-based GP. With the aim of developing techniques to counteract overfitting in Grammar-based GP, the classes of stopping criteria in [8] were implemented here on symbolic regression problems.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

1.1 Classes of Stopping Criteria

In [8], Prechelt implements three classes of stopping criteria.

Generalisation loss (in %) at epoch t, is given by:

$$GL(t) = 100 \times \left(\frac{E_{va}}{E_{opt}} - 1 \right)$$

where E_{va} is the validation error at the current epoch, and E_{opt} is the minimum validation error observed up until the current epoch. The first class of criteria uses the threshold value of α:

$$GL_{\alpha} : \text{stop at } 1^{\text{st}} \text{ epoch } t \text{ with } GL(t) > \alpha$$

If training is progressing well, generalisation losses are assumed to have a higher chance of being 'repaired'. Training progress, over k generations is given (in per thousand) by:

$$P_k(t) = 1000 \times \left(\frac{\sum_{t'=t-k+1}^{t} E_{tr}\left(t'\right)}{k \times \min_{t'=t-k+1}^{t} E_{tr}\left(t'\right)} - 1 \right)$$

The second class of stopping criteria in [8] is defined for a strip of length k epochs as:

$$PQ_{\alpha} : \text{stop at } 1^{\text{st}} \text{ end}-\text{of}-\text{strip epoch } t \text{ with } \frac{GL(t)}{P_k(t)} > \alpha$$

The third class of criteria identified recorded the sign of changes in generalisation error, and stopped when this error increased in a predefined number of successive strips [8]:

$$UP_s : \text{stop at epoch } t \text{ iff } UP_{s-1} \text{ stopped at epoch } t-k$$
$$\text{and } \mathrm{E}_{va} > E_{va}\left(t-k\right)$$
$$\mathrm{UP}_1 : \text{stop at } 1^{\text{st}} \text{ end}-\text{of}-\text{strip epoch } t$$
$$\text{with } \mathrm{E}_{va}\left(t\right) > E_{va}\left(t-k\right)$$

where s is the number of successive strips.

All stopping criteria decide to stop at some time t during training, and the result of training is then the set of weights, or the evolved model (in the case of GP) that exhibited the lowest validation error prior to training being stopped.

2. INVESTIGATIONS

Grammatical Evolution in Java [6, 5] was used to fit models to 2 symbolic regression problems, with target functions:

$$Y = 0.3X \times \sin \frac{\pi x}{5}$$

Training dataset range: [-1, 1]. Validation and test dataset ranges: [-2, 2].

$$Y = 6x^3 + x^2 - 10x - 2$$

Training dataset range: [0, 2]. Validation and test dataset ranges: [-1, 3].

Each run was allowed to complete, and therefore possibly overfit the training data, in order to identify the *optimal* stopping point.

Table 1: Number of Times Result Produced at Generation of Global Minimum Validation Error, and Global Minimum Test Set Error. Total of 30 runs.

Symbolic Regression 1

Crit	Result of Run Global Min Val	Result of Run Global Min Test
GL_2	16	14
GL_5	16	14
GL_{10}	17	15
GL_{20}	18	16
GL_{30}	18	16
PQ_1	24	22
$PQ_{2.5}$	26	24
PQ_5	28	26
$PQ_{7.5}$	28	26
UP_2	29	27
UP_3	30	28
UP_4	30	28
UP_5	30	28

Symbolic Regression 2

Crit	Result of Run Global Min Val	Result of Run Global Min Test
GL_2	8	8
GL_5	8	8
GL_{10}	8	8
GL_{20}	8	9
GL_{30}	8	9
PQ_1	15	11
$PQ_{2.5}$	17	11
PQ_5	19	11
$PQ_{7.5}$	21	12
UP_2	22	14
UP_3	30	17
UP_4	30	17
UP_5	30	17

3. RESULTS

Each criterion dictates that training should stop at a stopping generation. The generation prior to the stopping generation at which the validation set error was at a minimum is the generation of the result. The model that has been evolved at this generation is the result of the run. The test set is independent of both training and validation sets, and is used to evaluate the generalisation ability of the result of the run after training has stopped. Table 1 summarises the results of applying each stopping criterion to both symbolic regressions over 30 runs. If the threshold value of the criterion was never breached, the stopping point was taken to be the last generation of the run. In applying the stopping criteria, we are hoping to stop training at the optimal time. This means stopping as soon as possible after the lowest validation set error of the run is observed. [8] defines a *good* criterion, as among those that find the lowest validation set error for the entire run. For the first regression, Table 1 shows that the UP class of criteria are extremely good at outputting the result of the run at the generation at which both validation and test set errors were at the global minimum. The GL criteria output the result of the run at the global minimum validation and test error for about half of the runs, and the PQ criteria are somewhere in between. For the second regression, the generation of the result is less likely to correspond to the global minimum test error, than to the global minimum validation error. The UP criteria are still the best at finding a *good* solution.

Prechelt [8] judges the *efficiency* of a criterion based on how long training continues after the final solution has been seen. We found some evidence that the 'less efficient' criteria (the UP criteria) trade off longer training for greater accuracy in finding the optimal validation error.

4. CONCLUSIONS

This paper explored the use of early stopping criteria with Grammar-based GP of symbolic regression functions. UP criteria are good at finding the globally best validation fitness. They can however take longer to stop training than the alternatives - if a low training time is important, then the GL criteria may be a better choice.

5. REFERENCES

[1] L. Becker and M. Seshadri. Comprehensibility and overfitting avoidance in genetic programming for technical trading rules. *Worcester Polytechnic Institute, Computer Science Technical Report*, 2003.

[2] Z. Cataltepe, Y. Abu-Mostafa, and M. Magdon-Ismail. No free lunch for early stopping. *Neural Computation*, 11(4):995–1009, 1999.

[3] R. Gencay and M. Qi. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *Neural Networks, IEEE Transactions on*, 12(4):726–734, 2002.

[4] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[5] M. O'Neill, E. Hemberg, C. Gilligan, E. Bartley, J. McDermott, and A. Brabazon. GEVA: grammatical evolution in Java. *ACM SIGEVOlution*, 3(2):17–22, 2008.

[6] M. O'Neill and C. Ryan. *Grammatical Evolution: Evolutionary automatic programming in an arbitrary language*. Springer Netherlands, 2003.

[7] G. Paris, D. Robilliard, and C. Fonlupt. Exploring overfitting in genetic programming. In *Artificial Evolution*, pages 267–277. Springer, 2004.

[8] L. Prechelt. Early stopping-but when? *Neural Networks: Tricks of the trade*, pages 553–553, 1998.

[9] C. Tuite, A. Agapitos, M. O'Neill, and A. Brabazon. A Preliminary Investigation of Overfitting in Evolutionary Driven Model Induction: Implications for Financial Modelling. *Applications of Evolutionary Computation (Proceedings Forthcoming)*, 2011.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/SRC/FM1389.