Examining the Internal Complexity of a Neural Network Trained with Divisia Component Data

Vincent A. Schmidt[1] Jane M. Binner[2]

[1]United States Air Force Research Laboratory Dayton, Ohio USA
[2]Aston University Birmingham, UK

Abstract

This paper studies the ability of standard feedforward neural network to support the generation of a reasonable rule set for Divisia data. This is done by examining the internal weight structure of the network when trained with Divisia component data, based on requirements levied by the Aggregate Feedforward Neural Network (AFFNN). Complexity measures are examined to assist with the selection of Divisia component encoding characteristics and training methods. Encodings and architectures that increase complexity of the internal network weight structure tend to increase the potential maximum number of generated rules.[1]

Keywords: Divisia, Inflation, Neural Network, Complexity, Data Mining, Rule Generation

1 Introduction

In recent years the relationship between "money" and the macroeconomy has assumed prominence in the academic literature and in Central Banks circles. Although some Central Bankers have stated that they have formally abandoned the notion of using monetary aggregates as indicators of the impact of their policies on the economy, research into the link between some kind of monetary aggregate and the price level is still prevalent. Attention is increasingly turning to the method of aggregation employed in the construction of monetary indices. The most sophisticated index number used thus far relies upon the formulation devised by Divisia [1]. The construction has it roots firmly based in microeconomic aggregation theory and statistical index number theory.

Our hypothesis is that measures of money constructed using the Divisia index number formulation are superior indicators of monetary conditions compared to their simple sum counterparts. Our hypothesis is reinforced by a growing body of evidence from empirical studies around the world which demonstrate that weighted index number measures may be able to overcome the drawbacks of the simple sum, provided the underlying economic weak separability and linear homogeneity assumptions are satisfied. Ultimately, such evidence could reinstate monetary targeting as an acceptable method of macroeconomic control, including price regulation.

As stated, the goal is to identify relationships capable of accurately predicting inflation as a function of Divisia component data, assuming such relationships exist and are well-defined. Econometricians have devised a variety of models and techniques to meet this challenge, and research in this area continues to evolve.

One approach we have examined is using a model that generates rules extracted from a neural network trained with Divisia component data. This model, the Aggregate Feedforward Neural Network (AFFNN) [2], was tested on Divisia component data following the footsteps of an earlier proof-of-concept experiment successfully performed by Gazely and Binner [3]. This study's simple neural model demonstrated the promise of a connectionist approach for learning Divisia-Inflation relationships, but was not devised to express the results in human-understandable terminology. This is the reason behind continuing the research using the AFFNN as a platform. The AFFNN model was originally selected because of its existing rule extraction capability, and has already demonstrated the ability to successfully train and extract rules based on Divisia component data [4].

The AFFNN is a novel general-purpose feedforward connectionist model originally developed for data mining [5]. The final part of the AFFNN system includes an automated decompositional rule extraction capability, where rules are expressed in human-readable and machine-executable MATLAB code. This allows the rules to be inspected and validated by subject-matter experts, and provides the capability for the rules to be used independently of the trained AFFNN model in classification and prediction applications.

[1]USAF cleared for Public Release by ASC Public Affairs. Document Number AFRL-WS 05-0545.

This series of joint UK-US studies has taken a methodical, steady approach to the challenge of discovering and describing Divisia-Inflation relationships. The original research showed the feasibility of using the AFFNN to learn these relationships, as well as its ability to produce sets of rules describing the learned relationships [6]. However, those studies were necessarily limited to experiments and tests with a specific encoding of the network's inputs. High-level analysis of the resulting rulesets showed that the decompositional algorithm produced a very large number of complex rules. We believe that the ruleset can be simplified and reduced if the complexity of the internal weight structure can be reduced. The purpose of this stage of the research is to examine factors necessary to define and reduce the complexity of the trained network and consider the impact on potential rulesets. To do this, we must develop an understanding of the source and nature of the complexity, the topic of this paper.

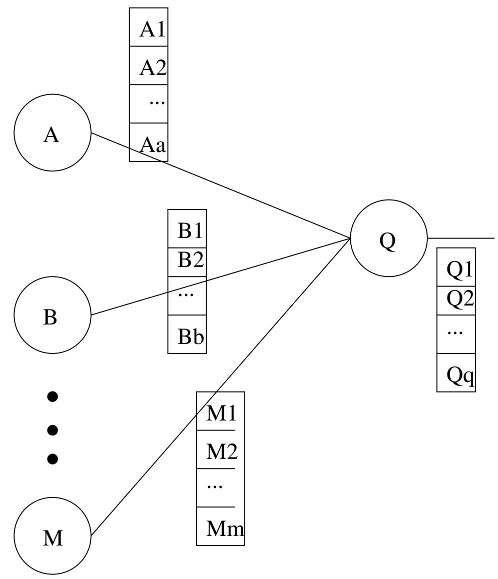

Figure 1: Arbitrary Network Elements

2 Defining Complexity

The tremendous interest in connectionist models across domains has resulted in a sizeable collection of generic and specific optimizations for such models. In most instances, the literature is interested in optimal training, fast learning, and low error rates. This is often accomplished through careful selection of network architectures, features and related encoding, and careful network pruning to reduce overall network complexity and achieve these goals. While we certainly acknowledge the importance and practicality of these efforts, we are not interested (at this stage in our research) in reducing network complexity in that sense, through making changes to the architecture using techniques such as network pruning. Instead, we'd like to examine the impact of holding the trained architecture constant while reviewing and manipulating internal network weights and the interpretations of those weights.

Our goal is best demonstrated by example. Consider Figure 1, a representation of a portion of a trained neural network, where the nodes A, B, ..., M represent all nodes producing inputs to node Q. With the decompositional rule extractor used by the AFFNN, all outputs from each node are independently clustered for each link, and the results discretized. These discretized values are used as inputs to the next node. In the figure, node A yields clustered values A1..Aa, node B yields B1..Bb, and so on. The outputs of the output node (Q) are similarly clustered. The decompositional algorithm generates rules for each discrete element of the output cluster (Q1..Qq) based on combinations of the cluster values (nodes A..M) feeding node Q.

Clearly, increased numbers of input nodes, along with increased cluster sizes for each input node, results in combinatorial explosion. We want to determine if there is a reasonable method by which we can decrease the number of elements in the input clusters (the outputs of nodes A..M), and simultaneously keep the network architecture constant[2]. We refer to the clusters of the inputs nodes as the internal network state, the complexity of which we aim to reduce. This is in contrast with the internal architecture, which describes network links and connectivity and must remain constant.

Searching the literature did not yield any comments characterizing neural network complexity in terms of learned network weights or clusters of weights. Most connectionist complexity measures seem to be based on network architecture. Several sources cite research about the information content of the inputs themselves; such complexity is frequently measured using information-theoretic techniques, entropy, energy-based functions, etc., and some references are dedicated to the analysis of complex systems, but these do not meet our immediate needs. Our supposedly unique requirement and interest in the internal network state drives this stage of the research: can we identify, measure, and potentially manage the internal complexity of the model?

[2]Maintaining a constant network architecture is a requirement levied by the AFFNN approach.

3 Experimental Setup

The AFFNN is a tightly coupled system when fully engaged, but since we're only interested in inspecting a single output feature (Inflation), the full capability of the AFFNN is not strictly necessary here. For now we continue to experiment with the full AFFNN because of the existing rule extraction capability. After we rewrite and generalize the rule extractor we expect to be able to concentrate on testing limited neural networks containing a single hidden layer and one encoded output feature (Inflation), folding what we've learned back into the full AFFNN in later work.

Remaining within the requirement that the network architecture is static once defined, some elements impacting internal complexity can be identified and measured. In logical order from network definition through analysis:

1. Feature complexity – A significant contributor to the complexity of the system depends on the set of input features selected. The Divisia component data is a predetermined set of features. Information-theoretic and statistical methods can be used to generate measurements of complexity for this data. This study will not include the selection of additional features or the explicit omission of Divisia components.

2. Feature encoding – Poorly chosen, complex, or ill-defined encodings can have a negative impact on network training and require larger feature space, resulting in a need for larger connectionist models. Deterministic selection of optimal encodings is an open issue, and will certainly not be resolved here. However, heuristics and experimentation are helpful in weeding out bad choices and converging toward "better" ones. Some of this work has already been performed in our earlier studies, cited in the references.

3. Architecture definition – Traditional measures of numbers of hidden layers, links, connectivity, activation functions, and similar items come into play. For the AFFNN, these must remain static once they are defined, so no modifications in these parameters are possible. (Similarly, network training is frequently an open issue, especially for new problems.) Traditional measurements of network size, training convergence, and testing error can be made.

4. Network interpretation – The AFFNN's decompositional rule extractor uses the trained neural network as its input, generating rules based on discretized activation function outputs and weight clusters. This process often requires more effort than network training, and is heavily impacted by the internal network complexity. Results can be measured in terms of rule complexity, quantity, and accuracy.

Table 1: Average MSE / Typical N across 3 trials

Architecture	1-of-M	Thermometer
32–6–6	0.0228 / 250000	0.0104 / 100000
32–8–6	0.0196 / 475000	0.0118 / 680000
32–2–6	0.0530 / 50	0.0336 / 30

Given this collection of elements impacting (or impacted by) internal network complexity, it might be useful to examine the number of clusters feeding the output node as the network is trained, reporting network (training) accuracy as training proceeds.

The experiment varies the input encoding and basic architecture across a preset number of training iterations. Training is halted, inspected, and continued until the iteration maximum is reached. The measurements obtained should be useful in guiding the design of an AFFNN or similar model for use with Divisia component data, producing extracted rules that are reduced in complexity and quantity. Specific selection of the study criteria and results are described in the following section.

4 Results and Discussion

Current (1977 Q1 – 2004 Q1) seasonally adjusted Divisia component data is automatically discretized and 1-of-M encoded, resulting in 32 binary-valued inputs for each data case. Inflation outputs were automatically clustered into six segments, encoded as 1-of-M for half of the experiment, and thermometer-encoded for the other half. (These encoding methods are commonly used for discrete neural network applications.) Training was performed using the standard RProp algorithm provided in MATLAB's Neural Network Toolbox, using 85 randomly selected cases of data (of 106 total), with MSE and cluster information collected at iterations 50, 500, 1000, and 2500. Training was terminated at 2500 iterations in every trial. Table 1 shows the average training MSE at the end of 2500 iterations of training for three trials in each condition listed in the table (left side of the "/"). The architecture column indicates the fully-connected network architecture of input, hidden, and output nodes. The hidden layer used logsig (sigmoid) activation functions, and the output layer used unbounded linear activation functions.

The algorithm used to cluster the internal network activation values is the same algorithm used to discretize network inputs and outputs, and is based on the standard deviation of differences in a sorted set of values [2]. We want to count the number of elements in each cluster of discretized activation outputs (cluster size) for each node in the hidden layer. (These are the clusters labeled A1..Aa – M1..Mm in Figure 1.) The product of these values determines the maximum number of rules the extraction algorithm could potentially generate, referred to as N. Table 1 (to the right of the "/") indicates approximated typical values for N resulting from this experiment.

In general, as the training progressed and the network MSE moved toward convergence, we found that N was reduced. This conforms to our expectation that the network becomes more focused as convergence occurs, reducing the number of features in each cluster. In a few cases N reached a relative plateau even though the MSE improved marginally during training. This is probably caused by tradeoffs in the network's weight structure during training.

Although not part of the original plan, we also used this opportunity to inspect the complexity of the output nodes. We observed that the networks using thermometer-encoded Inflation generated substantially smaller number of cluster features than those using 1-of-M encoding for Inflation. Specifically, this results in the potential for a maximum of thousands of solutions for the thermometer-encoded target, but a maximum of tens of thousands of solutions when using 1-of-M encoding. Additional investigation will be necessary to determine how the individual rule (solution) complexity is impacted in these cases.

5 Summary

Complexity can be tedious to define and measure, especially since system complexity includes the complexity of the inputs as well as that of the system itself. In the case of neural networks, even different encodings of the same inputs can impact system complexity. Similarly, the mechanics of the neural network allows the same architecture to exhibit different levels of complexity based on initial conditions, training methods, and final training state.

In this brief study, curiosity and necessity drove us to consider the question of identifying, measuring, and managing inner state complexity as a stepping stone to improving rule quality and quantity. The nature of the AFFNN constrains the operations we can perform when addressing these issues, limiting our investigation to characteristics that allow us to maintain a constant network architecture during and after training. This allows us to experiment with input encoding and weight interpretation (including extraction algorithms), but not with pruning methods, which modify network architecture.

The initial data results look promising, indicating that we may be able to control inner network complexity with a careful balance of encoding, architecture, and interpretation, since complexity does seem to decrease for this dataset as network performance improves.

The next step is to fold the knowledge obtained from this study back into the AFFNN model and evaluate if, indeed, the AFFNN approach is the best value for continuing research in this area, or if reimplementing the AFFNN's decompositional rule extractor for simple feedforward neural networks provides a more useful and practical mechanism. Other possibilities include making changes to the clustering algorithm used in the AFFNN's decompositional rule extractor, as well as modifying the extraction algorithm itself. Since rule extraction is still a hot topic, we expect to be introduced to interesting alternatives as we continue to look at new methods in related research.

References

[1] F. Divisia. L'indice mone'taire et al the'orie de la monnaie. *Rev. d'Economie. Political.*, 39:980–1008, 1925.

[2] Vincent A. Schmidt. *An Aggregate Connectionist Approach for Discovering Association Rules.* PhD thesis, Wright State University, Dayton, OH, May 2002.

[3] Alicia M. Gazely and Jane M. Binner. Optimal weights for Divisia aggregation using a neural network approach. In *5th Biennial Conference on Alternative Perspectives on Finance*, Dundee, 2000.

[4] Vincent A. Schmidt and Jane M. Binner. Searching for Divisia / inflation relationships with the aggregate feedforward neural network. *Advances in Econometrics*, 19:225–241, 2004.

[5] Vincent A. Schmidt and C. L. Philip Chen. Using the aggregate feedforward neural network for rule extraction. *International Journal on Fuzzy Systems*, 4(3):795–807, 2002.

[6] Vincent A. Schmidt and Jane M. Binner. Determining Divisia rules using the aggregate feedforward neural network. In *Proceedings of the 2003 International Conference on Artificial Intelligence*, pages 68–74, Las Vegas, Nevada, 2003.