Determining Divisia Rules Using the Aggregate Feedforward Neural Network

Vincent A. Schmidt
Wright State University
Dayton, Ohio USA

Jane M. Binner
Nottingham Trent University
Nottingham, UK

Abstract

This paper introduces a mechanism for generating human–readable and machine–executable rules that characterize the money–price relationship, defined as the relationship between the rate of growth of the money supply and inflation. Divisia component data is used to train an Aggregate Feedforward Neural Network (AFFNN), a general-purpose connectionist architecture originally developed to assist with data mining activities. The rules extracted from the trained AFFNN meaningfully and accurately describe inflation in terms of the Divisia component dataset.

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

1 Introduction

If macroeconomists ever agree on anything, it is that a relationship exists between the rate of growth of the money supply and inflation. The conventional way of measuring the amount of money circulating in an economy is to simply sum the various constituent liquid liabilities of commercial and savings banks. However the belief is widely established that this method of arriving at broad money aggregates is seriously flawed and based on untenable assumptions, as shown in Belongia [1]. From a micro-demand perspective it is hard to justify adding together component assets having differing yields that vary over time, especially since the accepted view allows only perfect substitutes to be combined as one "commodity."

Barnett pioneered the use of the Divisia monetary aggregate as an alternative to simple sum aggregation [2,3]. By drawing on statistical index number theory and consumer demand and aggregation theory, he advocated the use of the Divisia chain-linked index numbers as a means of constructing a sophisticated weighted index number measure of money. A set of weights are required in the formation of the aggregate to measure the flow of monetary services provided by the stock of monetary assets. The potential advantage of Divisia monetary aggregates is that the weights can vary over time in response to shifts in the yield curve and to financial innovation, which alter the opportunity costs of holding monetary assets. (See Fisher, Hudson, and Pradhan [4] and Mullineux [5] for detailed discussions on the construction of Divisia monetary aggregates and associated problems.) Proponents of weighted index number aggregation contend that Divisia M4 endogenizes at least some of the major innovations that clearly distorted simple sum M4 in the 1980s, especially the payment of competitive interest rates on checking deposits.

Clearly, the foundations of the construction of monetary aggregates are well rooted in monetary aggregation theory and require extremely strong assumptions. (Barnett and Serletis give a detailed treatment of the theory of monetary aggregation [6].) However, the underlying philosophy of the current research is that all assumptions can be weakened and the Divisia formulation can still be improved. Recent research has focused on accounting for the riskiness of the asset in the index construction; see Barnett et al [7,8] for such efforts in the USA and Drake et al [9] and Binner and Elger [10] for approaches in the UK.

Can a new empirically weighted measure of money be constructed that more closely captures the monetary services flow provided by the component assets? The answer to this question may lie in the use of the neural network, one of many tools from the realm of Artificial Intelligence. Connectionist models, as neural networks are also called, tend to be robust with respect to noisy data, often able to learn and generalize with a high degree of accuracy. Fast desktop computers and the availability of inexpensive and high-quality neural network design software grants the casual researcher easy access to these tools without requiring a steep learning curve or advanced knowledge of connectionist principles. This is especially helpful, since simple networks often provide great insight into the behavior and relationships of the data being examined. The use of neural networks to learn the relationship between Divisia asset components and inflation was first attempted very successfully for UK Divisia M4 by Gazely and Binner [11]. They demonstrated that a properly generated neural network is expressive enough to learn relationships between Divisia components and inflation to the extent that even a reasonably simple architecture can outperform traditional Divisia measurements under many circumstances, and performs comparably at its worst.

Unfortunately, the trained neural network still guards its secrets carefully; it can categorize and classify previously unseen data and predict appropriate outcomes, but casual examination of network weights is not sufficient to comprehend its operation. Efforts to describe the "black box" neural model in understandable and identifiable terms are numerous and ongoing [12,13]. Such efforts often require specialized neural models and are most appropriate in limited domains.

The Aggregate Feedforward Neural Network (AFFNN) is a general-purpose connectionist model that does not unnecessarily restrict the domain or training capability of the modeler [14]. The goal of the AFFNN is to take collection of K inputs and discover the relationships among them, non-autoassociatively. The resulting network lends itself to rule extraction, such that the learned relationships can be expressed in terms the original inputs by a series of readable and executable *if-then* rules [15].

The following sections of this paper briefly describe the AFFNN and its utility in learning specific relationships using the monetary component asset data of UK Divisia M4[1]. The AFFNN architecture and training are presented along with a sample generated rule and a comparison to a typical traditional feedforward neural model where rules are not generated.

2 AFFNN Using Divisia Components

The AFFNN is a supervised fully-connected feedforward connectionist model, typically with a single hidden layer, where the input

[1] Component data is available on the Internet at http://www.bankofengland.co.uk/mfsd/index.htm (Bank of England Statistical Abstracts, Part 2, Section A, tables 12.1 and 12.2).

and target vectors are identical and contain K encoded attributes. The AFFNN allows the user to select almost any supervised training and performance functions, but also contains an exclusive set of additional functions enforcing the network's ability to learn attributes A_i, $1 \leq i \leq K$, as a function of the remaining attributes for every instance of i simultaneously within the same network. More formally, if notation A_{i^*} represents the set of attributes such that all attributes except for A_i are included in A_{i^*}, and $O_{\hat{i}}$ is the set of attributes including only A_i, then the network endeavors to learn the functions:

$$O_{\hat{i}} = f(A_{i^*}), \forall 1 \leq i \leq K \qquad (1)$$

simultaneously for all values of i when such functions exist, given an adequate number of hidden nodes and sufficient network training. There is no restriction on the supervised training algorithm or network performance function used, since there are unique training and performance function wedges supporting the AFFNN, functions not seen by the user, interacting with the functions the user has selected.

The task at hand is the application of the AFFNN to Divisia component data, generally used to compute Divisia indices. This data is automatically clustered and thermometer-encoded for use in the neural network. An AFFNN using quarterly Divisia components, expressed as a percentage increase from values exactly four quarters earlier, includes attributes:

- Notes and Coin (NC) encoded into 7 levels,
- Non-Interest Bearing Bank Deposits (NIBD) encoded into 14 levels,
- Interest Bearing Bank Sight Deposits (IBSD) encoded into 4 levels,
- Interest Bearing Bank Time Deposits (IBTD) encoded into 7 levels, and
- Building Society Deposits (BSD) encoded into 7 levels.

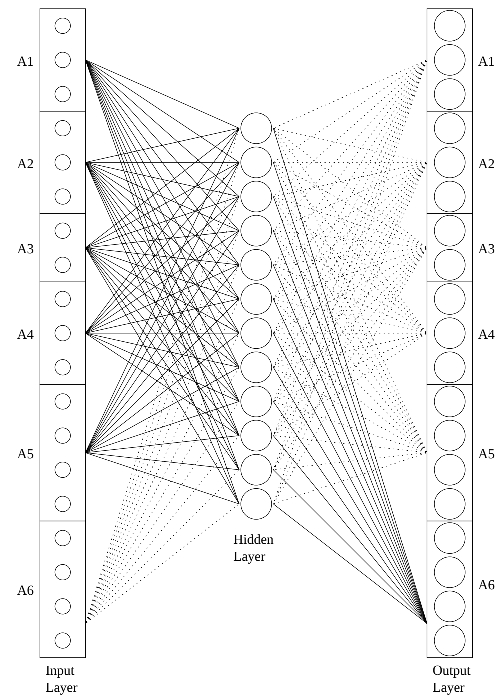

Figure 1: Divisia AFFNN Block Diagram

These five attributes, along with inflation growth rate (also expressed as a percentage of increase and encoded into 4 levels), comprise six component attributes for AFFNN training. Figure 1 shows a block diagram of an AFFNN using these components. To conserve space, this figure depicts a 19-12-19 AFFNN, although the actual network used in this research effort is a 43-15-43 model (43 sigmoid nodes in the input and hidden layers, and 15 linear nodes in the output layer). Rules are generated using the weights and values propagating along the solid lines in the figure, with the dashed lines being ignored. In this example, attribute A6 is the only attribute receiving weights at the output nodes, but does not contribute as an input.

The raw Divisia component data is available as quarterly figures from Q1 of 1977 through Q1 of 2001, yielding 96 exemplar vectors. Computing percentage of change for each successive corresponding quarter (the same quarter in the previous year) reduces the available data to 94 exemplars. The AFFNN was trained with 80% (75 vectors) of the exemplars selected at random, and the remaining 20% (19 vectors) were set aside for validation and testing. Training was performed for 4000 epochs (943 seconds) using a custom Matlab model[2]. Training mean-squared error (MSE) was 0.0079, with 93.3% of the training data and 84.2% of the test data properly classified for the inflation component of the AFFNN output.

3 Rule Generation

As mentioned in the previous section, an automated clustering algorithm determined that inflation data (percentage change in growth rate of price level) should be reclassified into four distinct levels, specifically:

L_1 = (-0.0033, 0.0124]

L_2 = (0.0124, 0.0558]

L_3 = (0.0558, 0.1116]

L_4 = (0.1116,∞)

As anticipated, the AFFNN learned the general function[3]:

$$f(NC, NIBD, IBSD, IBTD, BSD) \Rightarrow L_x, L_x \in \{L_1, L_2, L_3, L_4\} \tag{2}$$

Rule generation techniques applied to the trained network allow a series of relationships to be extracted from this network, written in terms of the original attributes used for network training. The rule extraction and generation procedures are described in Schmidt and Chen [16].

[2]Linux-based dual 300 MHz Pentium II PC with 128 Mb RAM

[3]Traditional feedforward neural networks would also be capable of learning such relationships; the chief advantage of the AFFNN is that it learns the other five sets of relationships, for NC, NIBD, IBSD, IBTD, and BSD, simultaneously within the same network. The functions for the Inflation attribute, represented by L_x in Equation 2, are the only ones of immediate interest in this study.

```
if {
     (nc <= -0.004833) )
       & ((nibd <= -0.091788)
         | (nibd > -0.010042
            & nibd <= 0.049695)
         | (nibd > 0.082643))
       & (ibsd <= 0.148376) )
       & (ibtd > 0.008503) )
       & (bsd > 0.082032)
} then return true;
```

Figure 2: Example Expression for L_3

The rule extraction algorithm is a typical decompositional technique that creates rules by examining and combining the activation values within the trained AFFNN. Rules can be requested for each data class of a specified attribute. For example, requesting the rules for L_3 yields 408 distinct expressions described in terms of the original attribute values. As an example, one of the generated expressions is shown in Figure 2. This expression describes one set of conditions under which L_3 is true. If the values of NC, NIBD, IBSD, IBTD, and BSD satisfy any of the 408 generated expressions, then L_3 ("Inflation % change is in the range (0.0558,0.1116]") is satisfied.

Such rules are easy to interpret and are also simpler to validate by econometricians. Clearly it is easier for econometricians to comprehend and comment on a rule such as the one shown in Figure 2 than to grapple with internal neural network weights, biases, and activation functions directly.

4 Discussion of Results

One advantage of rule generation is that the generated rules are actually executable code and can process data independently of the original AFFNN. Running the training and testing data through the rules generates results very similar to the results of the original network. (The rule extraction algorithm relaxes a few constraints that may allow the generated rules to process borderline cases differently than the AFFNN would compute them, so absolutely identical AFFNN and rule results are not expected in all cases.)

With respect to the encoding of the Divisia component data as exercised, generated rules tended to use all five attributes, indicating the relative importance of all attributes. Expression simplification, one step of the rule generation process, commonly folded two or more levels of the NIBD component data together (as in Figure 2), showing the single rule would be true for two or more levels of the attribute. This also indicates that specific levels of the NIBD data were often not key contributors, but that general levels of NIBD did play a role in determining system outcomes. This was also true for the BSD attribute. In contrast, the system rarely folded IBSD or IBTD together, indicating that the inflation growth rate may be closely tied to specific levels of these two attributes. These results, which might initially seem to conflict with earlier sensitivity analysis findings by Gazely and Binner [11], are not alarming, since the AFFNN observations are based on the importance of particular levels of attributes in certain equations, rather than weight sensitivities in the overall network.

To form a basis for comparison, an individual supervised feedforward neural network was created and trained using the same data encoding as the AFFNN. There were 39 (43-4) encoded inputs, 6 sigmoid nodes in the hidden layer, and 4 linear nodes at the output layer. This network was trained for 4000 epochs (94 seconds, MSE=0.004789, although MSE was stable near 1000 epochs) with the identical 80% training dataset, and tested with the identical 20% testing dataset. This system yielded 97.33% training accuracy and 78.95% testing accuracy, results very comparable to the results obtained by the AFFNN (93.3% and 84.2%, respectively). Although the decompositional rule extraction technique applied by the AFFNN is general enough to handle general-purpose feedforward neural networks, it is currently only implemented to process the AFFNN structure, so no rules were generated for the individual network. Regardless of this drawback, it is clear that the AFFNN performed nearly as well as the independent network on the training data, and was able to generalize better on the test data. This strengthens arguments in favor of using of the AFFNN for such monetary component asset data.

Given the size of the dataset and the small number of attributes, a large number of rules were extracted from the trained AFFNN. Such an unexpectedly liberal quantity of rules clearly suggests one or more of these conclusions:

1. the data has not been optimally encoded or clustered well enough for learning to take place,

2. the relationships among these data are extremely complex and cannot be adequately captured with the expressive power utilized, or

3. there is not a sufficient mechanism to accurately capture the relationships, and the network is merely learning a somewhat random set of relationships based on the data available.

Given that other techniques such as those used in Gazely and Binner [11] have been able to learn these relationships, the first point is the most likely candidate conclusion, with the second point a distant second and the third point being possible, but highly unlikely. Whatever the actual reason, the results obtained in this proof-of-concept study demonstrate the potential complexity of attempting to describe relationships in even small datasets.

5 Summary

The Aggregate Feedforward Neural Network is helpful in its ability to both generalize and describe the resulting relationships. Applying a simple decompositional technique to the AFFNN yields a collection of descriptive *if-then* rules which may prove insightful in identifying or explaining the relationships between various monetary assets and the corresponding growth rate of inflation. The network construction is straight forward and the AFFNN trains reasonably quickly. Extracted Divisia component rules are expressed in terms of the component names and are simple to read and validate by econometricians.

The nature of the rules generated for Divisia component asset data underscores the importance of selecting appropriate heuristics to allow relationships to be discovered and properly expressed. Inappropriate encoding or excessively fine granularity results in a lack of ability to adequately learn relationships, and may also contribute to excessive quantities of rules being extracted from the network. Selecting the proper values and combinations of attributes is a significant step. This study settled on one obvious encoding mechanism (percentage variation among values, with automated clustering and thermometer encoding), but other encoding mechanisms may be more appropriate. Closer examination of the rules generated for each output level may also give insight into such areas.

Additional research will investigate other encoding mechanisms in an effort to simplify both the number and complexity of generated rules. It would also be beneficial to examine alternative connectionist approaches for both learning and expressing the percentage changes in the Divisia component relationships as they apply to rates of inflation growth. The full expressive power of the AFFNN comes into play only when there are multiple relationships among attributes. Lack of such relationships cripples the AFFNN's ability to generalize effectively. It is not yet clear if that is the case with Divisia data.

The process described here clearly identifies the Aggregate Feedforward Neural Network as a useful tool for examining relationships among the data components typically used to compute Divisia indices. Future research will evaluate the performance of the new UK Divisia, constructed using the rule generation mechanism described here, in a simple inflation forecasting experiment. Additional studies and comparisons with contemporary and traditional techniques will assist in identifying the role of connectionist models in this revolutionary work.

References

[1] Michael T. Belongia. Measurement matters: recent results from monetary economics reexamined. *Journal of Political Economy*, 104(5):1065–1083, 1996.

[2] William A. Barnett. The user cost of money. *Economic Letters*, 1(2):145–149, 1978. Reprinted in W.A. Barnett and A. Serletis (Eds.) The Theory of Monetary Aggregation, North-Holland, Amsterdam, Chapter 1, 2000, 6–10.

[3] William A. Barnett. Economic monetary aggregates; an application of index number and aggregation theory. *Journal of Econometrics*, 14(1):11–48, 1980. Reprinted in W.A. Barnett and A. Serletis (Eds.) The Theory of Monetary Aggregation, North-Holland, Amsterdam, Chapter 2, 2000, 11–48.

[4] Paul Fisher, Suzanne Hudson, and Mahmood Pradhan. Divisia indices for money: an appraisal of theory and practice. Technical Report 9, Bank of England Working Paper Series, 1993.

[5] Andy W. Mullineux, editor. *Financial Innovation, Banking and Monetary Aggregates*, chapter 1, pages 1–12. Edward Elgar, Cheltenham, UK, 1996.

[6] William A. Barnett and Apostolos Serletis, editors. *The Theory of Monetary*

Aggregation. North-Holland, Amsterdam, 2000.

[7] William A. Barnett. Exact aggregation under risk. In William A. Barnett, Maurice Salles, Herve' Moulin, and Norman Schofield, editors, *Social Choice, Welfare and Ethics*, pages 353–374, Cambridge, 1995. Proceedings of the Eighth International Symposium in Economic Theory and Econometrics, Cambridge University Press. Reprinted in Barnett, W.A. and Serletis, A. (Eds.) (2000), The Theory of Monetary Aggregation, North-Holland, Amsterdam, Chapter 10, pp. 195–216.

[8] William A. Barnett, Yi Liu, and Mark Jensen. The CAPM risk adjustment for exact aggregation over financial assets. *Macroeconomic Dynamics*, 1:485–512, 1997. Reprinted in Barnett, WA. and Serletis, A. (Eds.) (2000), The Theory of Monetary Aggregation, North-Holland, Amsterdam, Chapter 12, pp. 245–73.

[9] Leigh M. Drake, Adrian Fleissig, and Andy W. Mullineux. Are 'risky' assets substitutes for 'monetary' assets? evidence from an aim demand system. *Economic Inquiry*, 37:510–526, 1999.

[10] Jane M. Binner and Thomas Elger. The uk personal sector demand for risky money. *Economic Inquiry*, 2002. Under review.

[11] 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.

[12] Geoffrey G. Towell and Jude W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101, 1993.

[13] Hongjun Lu, Rudy Setiono, and Huan Liu. NeuroRule: A connectionist approach to data mining. In *Proceedings of the 21st VLDB Conference*, Zurich, Swizerland, 1995.

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

[15] Vincent A. Schmidt and C. L. Philip Chen. Extracting rules from the aggregate feedforward neural network. In *Proceedings of the 2002 International Conference on Artificial Intelligence*, 2002.

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