UNIVERSITY of OTAGO
Te Whare Wānanga o Otāgo
NEW ZEALAND

BUSINESS SCHOOL
Te Kura Pakihi

ISSN 1178-2293 (Online)

University of Otago
Economics Discussion Papers
No. 1205

November 2012

Artificial Neural Networks and Aggregate Consumption Patterns in New Zealand

Dan Farhat
University of Otago

November 2012

Address for correspondence:

Dan Farhat
Department of Economics
University of Otago
PO Box 56
Dunedin
NEW ZEALAND
Email: dan.farhat@otago.ac.nz
Telephone: 64 3 479 8645

Artificial Neural Networks and Aggregate Consumption Patterns in New Zealand

Daniel Farhat
University of Otago – Department of Economics
PO Box 54, Dunedin, New Zealand 9054
dan.farhat@otago.ac.nz
(Tel) +64 03 479 8645

[DRAFT: September 2012]

Abstract

This study uses artificial neural networks (ANNs) to reproduce aggregate per-capita consumption patterns for the New Zealand economy. Results suggest that non-linear ANNs can outperform a linear econometric model at out-of-sample forecasting. The best ANN at matching in-sample data, however, is rarely the best predictor. To improve the accuracy of ANNs using only in-sample information, methods for combining heterogeneous ANN forecasts are explored. The frequency that an individual ANN is a top performer during in-sample training plays a beneficial role in consistently producing accurate out-of-sample patterns. Possible avenues for incorporating ANN structures into social simulation models of consumption are discussed.

Keywords: Artificial neural networks, forecasting, aggregate consumption, social simulation

JEL codes: C45, E17, E27

1. Introduction

This study uses *artificial neural networks* (ANNs) to mimic the patterns of a particular macroeconomic variable in the New Zealand economy: changes in the growth rate of per-capita consumption. ANNs are mathematical algorithms that transform input data into outputs. They function in a fashion similar to neurons in the brain, which enhance or inhibit incoming impulses before transferring them on to other neurons. The highly flexible, massively parallel and potentially non-linear structure of ANNs, along with their ability to actively learn from their input information, makes them ideal for identifying complex relationships in data. As such, they have been shown to be extremely useful for solving pattern recognition, optimization and forecasting problems.

A confluence of social science, computer science and biology which has been occurring since the 1990s inspires this project. In standard macroeconomic models, the household sector is often comprised of homogeneous, forward-looking, perfectly rational agents. These agents are embedded into a grand model of the economy and participate in deriving a state of general equilibrium. Frameworks of this sort are favoured because they produce unique, tractable equilibria which are straightforward to interpret; however, they also have several shortcomings. For ordinary people, it is difficult to think like *homo oeconomicus*. In reality, diverse populations of imperfect individuals follow rules-of-thumb and rely on social systems to make decisions. Further, the outcomes from these models are artificially-coordinated (i.e. we can derive equilibria but cannot explain how they are attained or maintained). With modern computing, complex systems similar to those seen in nature can be crafted. Based on the one-to-one interactions of heterogeneous individuals, these systems produce (and, therefore, *explain*[1]) aggregate phenomena.[2] Reasonable rules guiding the choices and interactions of agents are required to create well-performing artificial social systems. ANNs, themselves inspired by biological systems, offer a potential means for establishing these rules.

In the 1990s and 2000s, research demonstrated that ANNs can perform well in reproducing patterns seen in several types of financial and economic data.[3] It was also shown, however, that they do not outperform traditional methods consistently. As a result, researchers tend to evaluate the usefulness of ANNs on a case-by-case basis. One of the few applications of ANNs to aggregate consumption, done by Church and Curram (1996) for the United Kingdom, compares the forecasting potential of a carefully chosen ANN (which can be formulated independent of mainstream economic theory) to structured empirical models. Their central finding is that although

1 See Epstein (1999) for the relationship between explanation and systemic generation.

2 For examples of this work, see Tesfatsion (2002, 2005), Tesfatsion & Judd (2006), Gatti, et al. (2005), Mirowski (2007), Gaffeo et al. (2008) and Raberto et al. (2008).

3 Zhang et al. (1998), Vellido et al. (1999), and Kourentzes and Crone (2010) review a variety of neural network applications and describe several of the main advantages and shortcomings of this approach.

ANNs are more flexible in the types of variables which can be included and do not require a large number of data points to produce useful forecasts, they do not outperform standard econometric approaches significantly. This study shows that ANNs are capable of producing aggregate consumption patterns for New Zealand more accurately than a 'standard' linear econometric model, a result which both supports and contrasts the findings of Church and Curram (1996). The usefulness of using ANNs (in general) to establish decision rules related to consumption patterns in systemic models is highlighted.

In addition to evaluating their ability to reproduce New Zealand consumption data, this project also attempts to improve the overall accuracy of ANN systems by adding diversity. Several studies in the forecasting literature note that imperfect, heterogeneous forecasts can be combined to produce a 'social forecast' with enhanced accuracy. In nature, as well, evidence suggests that ecosystem diversity aids in system stability[4], which implies that the aggregated predictions of more heterogeneous communities may be more resilient to environmental shocks. Provided the method for combining forecasts is wisely chosen, the estimated patterns from a large variety of ANNs may be able to capture aggregate consumption patterns in New Zealand with more precision. This paper evaluates multiple forecast combination methods and shows that some methods are capable of constructing improved out-of-sample patterns. In doing so, we can interpret the model in this project as a primitive artificial society in which the knowledge of individual agents is combined. The results from this exercise inform future models in which heterogeneous individuals interact when making consumption decisions.

The remainder of this paper is as follows. First, a brief description of the algorithm used to construct and train the ANNs in this study is provided. In this algorithm, heterogeneous agents with private and incomplete information are asked to make forecasts about changes in aggregate per-capita consumption in New Zealand. These forecasts are derived by training an ANN for each individual to understand the underlying relationship between that individual's private input data and changes in consumption. Agents are allowed to differ not only by the amount of information they are given, but also by the topologies of their ANN. The individual forecasts are then aggregated into a social forecast. The precision of the individual forecasts and the social forecast is assessed using out-of-sample data. The article ends by motivating future work from the results.

2. Method

Beltratti et al. (1996), Jain and Mao (1996), Warner and Misra (1996), Cooper (1999), Gonzalez (2000), and Detienne et al. (2003) provide easy-to-follow introductory guides on ANN algorithms. While these algorithms can become very complex depending on the problem they are attempting to

---

4 See McCann (2000) or Ives & Carpenter (2007) for an overview of the ecological relationship between diversity and stability.

solve, they are best understood by looking at the simplest type of ANN (known as a *perceptron*). In this basic model (Figure 1) K different pieces of input data (X_k, k = 1...K) are fed into the network to produce an output ( ). To generate , the input data is passed through "hidden layers" where different mathematical transformations are made. The raw data is first weighted (by ω_k) then summed along with a constant term (or *bias* - ω_0). The result is then passed through a function (called an *activation function*, A). The activation function is often either linear or *sigmoid* (s-shaped) in nature. The result from this step is then weighted (γ_1) and summed with a second bias term (γ_0) to generate . The perceptron shown in Figure 1 can be thought of as a single neuron which enhances or inhibits input data so as to produce reasonably accurate outputs (just as biological neurons enhance or inhibit incoming impulses before passing them on to other neurons). To do this, appropriate weights (ω's and γ's) must be found (in other words, the model must be *trained*).

The model shown in Figure 2 is used for this study. It works in the same manner as the simple perceptron except that there are *H*+1 neurons whose combined efforts produce the output, . One neuron has a linear activation function and simply passes the input data directly to the output (this is shown in the "lower level" [dashed lines] of the diagram). The other *H* neurons use non-linear transformations of the input data (denoted as A_h for simplicity – shown in the "upper level" [solid lines] of the diagram). For any non-linear neuron, *h*, we can represent the data transformation process algebraically as:

$$N_h = A(n_h)$$

where ω_{kh} is the weight that neuron h places on input k, ω_{0h} is the bias term, and N_h is the neuron's product used later to construct the output. The input variables in Figure 2 are *fully-connected* to the non-linear neurons (all input variables are used by each non-linear neuron), however it is possible to easily generate ANNs with *semi-connected* inputs by forcing select ω-weights to zero. This provides greater scope for incorporating diversity into the analysis.

When the inputs to (and the output from) each neuron can be either positive or negative, the hyperbolic tangent sigmoid function (tanh) is useful to use:

As a result, N_h will be bounded by $\pm$ 1.

[Figure 1 here]

[Figure 2 here]

An additional bias term along with weighted results from both the linear and non-linear transformations are used to calculate the final output:

where γ_0 is the bias, γ_h is the weight assigned to the non-linear neuron, N_h, and δ_k is the weight assigned to the input, k (produced from the linear neuron).

The ANN model described in Figure 2 proves to be quite convenient. Many econometric analyses rely on linear models, but ANNs allow for non-linear relationships between inputs and outputs which can enhance the model's fit. Kuan and White (1994) note that the presence of the linear "lower level" in the ANN described above makes the model directly relatable to many standard linear econometric analyses. The ANN is, in effect, a linear model of the form augmented by a non-linear function, .

We can create larger ANNs by adding additional hidden layers (where the output, , is passed on as an input to other neurons) if we wish. As noted by Hornik et al. (1989) and Hornik (1991), ANNs can approximate any functional relationship between inputs and outputs to an arbitrary precision provided that there are a sufficient number of hidden layers in the model (hence, they are known as 'universal approximators'). Cybenko (1989), Hornik et al. (1990), and Barron (1993) note that ANNs with a single layer may also be universal approximators provided that the activation functions used in the model satisfy certain properties (namely, smoothness, which is common in sigmoid functions) and the number of neurons (H) is large enough. As a result, the simple ANN model shown in Figure 2 is sufficient for capturing the relationship between the inputs and outputs analysed below.[5]

For time series data, it is useful to keep track of the time periods from which the data is drawn. Denoting T as the total number of available periods, we feed input data associated with any period $t \in T$, $\{X_{1t}; X_{2t}; ...; X_{Kt}\}$, into the ANN to produce an output for the corresponding time period, . The input and output data can be organized as time-ordered matrices:

and the time-indexed representation of the generalised ANN can be written as:

$$N_{ht} = A(n_{ht})$$

'Training' the ANN to produce acceptable outputs from input data involves determining appropriate values for the weights (ω's, γ's, δ's) in the model. To do this, a training algorithm similar to that described by Aminian et al. (2006) is employed. First, all available data is divided into three sets: a *training set*, a *validation set*, and a *forecasting set*. The training set and the validation set together form the in-sample data used to derive the weights. The forecast set is used to test the performance of the ANN. In the case of time-series data, these sets need not be time-ordered (i.e. we can randomly select time periods into each set). In this study, the forecast set is first pre-determined (for convenience, it is set to the last 16 periods of the available data). The remaining

[5] Although it is common to use ANNs with a single hidden layer for forecasting, as is done in this project, models with a large number of non-linear neurons (H) may take excessive amounts of computation time to fit to data. As a result, some researchers expand the number of hidden layers. (Zhang et al., 1998).

periods are randomly divided between the training and validation sets with approximately 70% of the in-sample data allocated to the training set and the remaining 30% to the validation set.[6]

Denoting G as the number of periods in the training set and V as the number of periods in the validation set, the neural network is trained using *back-propagation* (BP), a common procedure to determining the weights. In BP, initial values for the weights are first guessed.[7] The following iterative algorithm is then followed:

1. Using provided weights and the input data from the training set, the ANN produces output estimates.
2. The estimation error is calculated for each period in the training set (-), and a measure of fitness proportional to the mean squared error (MSE) is computed:

3. The weights are then updated with the goal of reducing estimation errors (thus improving the model's fit to the training set data).[8] Using the results from the training set, the weights are adjusted by:

The parameter μ is scale parameter that controls the rate at which the ANN learns. This parameter must be chosen with care as a μ too large can result in excessive imprecision while a μ too small

6 In this study, either a prescribed ANN structure is simulated multiple times, or multiple ANN structures are simulated simultaneously. In any single simulation, the training set and validation set always differ from those any other simulation due to the random allocation of the data. The forecast set, however, remains the same across simulations so that their performance can be compared.

7 Random starting weights are chosen. Choosing appropriate initial weights is important as it is possible for the BP algorithm to converge to local optima which produce poorly-performing ANNs. To reduce these occurrences, 500 starting points are randomly selected for each ANN and the BP algorithm is performed. The starting point that produces the best fit to the validation set is used. Note that this procedure does not identify a starting point associated with a global optimum.

8 See Beltratti et al. (1996) or Warner and Misra (1996) for a more detailed description of the weight updating process for the basic BP algorithm.

can take a great deal of computing time or result in convergence to a poor-fitting local optimum. In this study, $\mu = 0.01$.

With the tanh activation function, it can be shown:

4. After the weights are adjusted, the algorithm repeats at step 1.

   The training set MSE will fall with each iteration. If we allow the BP algorithm to operate until the MSE of the training set is minimized, the ANN will 'over-fit' this data. In other words, the ANN will memorize the training patterns and fail to learn the underlying relationship between inputs and outputs. As a result, the ANN will perform poorly at producing out-of-sample patterns. To avoid this, we compute how well the ANN fits the validation set during step 2 of each iteration:

We then allow the weights to be updated until the MSE of the validation set is minimized.[9] While we may not get an ANN that can fit all the in-sample data perfectly, out-of-sample forecasting is improved.

Once an ANN is trained using the BP algorithm above, the model is used to produce output estimates for the forecast set. Denoting F as the number of periods in the forecast set, the associated errors and MSE are then computed to evaluate the model's fit:

9 To induce efficiency in running the program, the BP algorithm operates until reductions in the MSE of the validation are sufficiently small (less than 1E-6 in this study).

To combine the forecasts of individual ANNs (each trained using the method above) into an aggregate forecast, we assign each ANN a weight and compute the weighted sum of their outputs. Denoting J as the total number of ANNs that we wish to combine and θ_j as a weight assigned to an individual simulation, this is:

where denotes the social forecast for period, f, in the forecast set. There is a large literature on the optimal combination of forecasts to draw upon when choosing θ_j (see Clemen (1989) and de Menezes et al. (2000) for a review). The objective of many of these methods is to create a combination of forecasts which is more accurate than any of its components. Four approaches in particular will be used in the analysis below:

1. *The simple average*. This approach weights all ANN forecasts equally: $\theta_j = 1/J$.

2. *Best overall performers*. This scheme assumes that ANNs that are good at overall in-sample forecasting will also be good at out-of-sample forecasting. The MSE for the entire training phase (training set + validation set) is constructed for each ANN:

   $; \tau = 1...(G+V)$

   Two weighting regimes are considered: one in which the ANN with the lowest MSE_j^{TP} receives all of the weight and one in which the top 25% of ANNs (i.e. the 0.25×J ANNs with the lowest MSE_j^{TP}s) are equally weighted ($\theta_j = 1/(0.25\times J)$ if in the top 25%, else $\theta_j = 0$). Note this scheme can result in ANNs with poor in-sample performance having no input into the combined forecast, even though these ANNs may have valuable information to contribute.

3. *Error-based weighting*. This approach was first developed for combining two forecasting methods by Bates and Granger (1969), and later described for multiple methods by Newbold and Granger (1974). ANNs with lower in-sample MSEs are given a relatively higher weight in constructing the aggregate:

   Note that the w most 'recent' periods in the non-forecasting set (the training and validation sets combined) are used to create the weights. If we wish to consider all of the available non-forecasting set data, we can set w = G + V – 1. A low w indicates that success in matching the most current data is important to producing an accurate aggregate forecast. Unlike scheme 2 above, this scheme allows all methods to receive some weight. However, the weights are biased towards good in-sample performers.

4. *Period-by-period outperformance*. De Menezes et al. (2000) note that assigning weights to each model based on how *frequently* it outperforms other models can produce combined forecasts with increased predictive accuracy. Unlike the combination methods above, how well a model performs during individual periods is scrutinized as opposed to overall performance. Often, a Bayesian approach is adopted to derive these weights from prior distributions (see, for example, Bunn (1975)). In a less rigorous approach with a similar flavour, the method used here establishes weights by scoring each ANN a point for each period in the non-forecast set that it is a "top performer". Formally:
   a. Select a parameter σ, $0 < \sigma < 1$, which represents the cut-off value for top performer status (e.g. if $\sigma = 10\%$, a top performer is among the 10% of the population with the lowest absolute error).
   b. For each ANN, compute how often the ANN was a top performer in the non-forecast set as defined by σ:

where:

and P_i is the ANN's total average point accumulation.

c. Normalise the weights to sum to 1:

As a starting case, σ = 25%. Note that ANNs that are most *often* successful relative to the other available ANNs will receive higher weight regardless of their accuracy; the *size* of the estimation error is not taken into account in this combination scheme.

While there are several other[10] methods for forming combined forecasts described in the forecast literature, the four methods mentioned here are common and simple to implement for a large number of forecasts.

The model described above is programmed into MATLAB. All simulations produce J = 15,000 ANNs. While a large J is desirable, increasing J beyond 15,000 lengthens computation time considerably. In addition to combining the individual forecasts using the methods described in the previous section, the program searches for the ANN which best matches out-of-sample data (an 'optimal' ANN).

3. Simulations and Results

A. Data

Data used in this study includes the growth rate of final private consumption expenditures per worker (c_t), the growth rate of GDP per worker (y_t), the point change in the unemployment rate (u_t), the point change in the money market interest rate (r_t), the CPI inflation rate (p_t) and the point change in the nominal effective exchange rate (q_t). All data pertains to the New Zealand economy, 1992q1 – 2011q3 (see Figure 3). Data for interest rates and exchange rates are sourced from the International Monetary Fund (2011) while all other data are sourced from the OECD (2011).

10 One particular alternative approach involves using OLS to derive the optimal weights. This method, initially suggested by Granger and Ramanathan (1984), has been shown by Hashem (1997) to improve the accuracy of combined ANN forecasts specifically. In the algorithm described above, however, it is difficult to produce reasonable OLS estimates since many of the ANNs generate correlated outputs and there is no guarantee that two ANNs in the sample will not produce identical patterns (i.e. be linearly dependent). Although there are procedures for correcting this potential multicolinearlity, as shown by Hashem (1997), they can be complicated to implement when the number of models is large (as it is in this study).

Consumption growth is the variable of interest. GDP serves as a proxy for income, and the growth rate of GDP serves as a measure of overall economic performance. Changes in the unemployment rate capture labor market dynamics. Changes in the interest rate capture the trade-off between spending and saving. The inflation rate measures fluctuations in the cost of living. Changes in the exchange rate account for incentives to purchase goods from abroad. These variables, commonly seen in both theoretical and empirical business cycle studies, are specific to the household sector (the largest sector in the economy) and represent the set of information that an average member of the population can readily access. A subset of these variables also appears in the study by Church and Curram (1996).

[Figure 3 here]

For the ANN models in the following experiments, the input variables in any period, t, include c_{t-1}, y_t, u_t, r_t, p_t and q_t. The output variable is c_t. As noted above, the final 16 periods of the sample (2007q4 – 2011q3) are reserved to be the forecast set. Because this period includes the most recent severe economic downturn, it offers a stringent testing ground for any forecasting model. The remaining 62 periods are randomly sampled into either the training set or the validation set for each individual ANN forecast (G + V = 62).

B. Benchmark

Two key properties of ANNs are that they are flexible in structure and non-linear in nature. It will be useful to evaluate the performance of ANNs by comparing their results to that of a rigid, linear econometric model. To do this, the following linear model will be employed:

where , k = 0…6, are model coefficients (estimated via OLS using all data not in the forecast set) and ε_t is an error term (assumed to be i.i.d.). Over the forecast set, this model produces a mean squared error of 0.456. The mean squared errors of the aggregated ANN models in the following experiments will be compared to this benchmark to evaluate their performance.

We can think of the linear model as a restricted version of the ANN model described above (with all ω's and γ's forced to zero) which over-fits the in-sample data. Comparing the simulated ANNs to the linear benchmark describes how incorporating non-linearity, learning, and heterogeneity (in both data and structure) succeed in producing out-of-sample patterns. There is no reason to expect this particular linear benchmark model to provide the best possible econometric forecasts given its naivety. In practice, we can choose any linear model that we wish. Provided we structure the inputs to the neural network accordingly, the linear framework will be relatable to the ANN framework. Exploring how ANNs perform compared to more complex linear models is left for future work.

C. Homogeneous linear network structure

First, the linear benchmark model and a linear ANN model are compared. To do this, ANNs are simulated with no non-linear neurons activated (i.e. H = 0, only the 'bottom level' in Figure 2 is active). Individual ANNs do not differ structurally; all heterogeneity in the model occurs from the random allocation of training and validation sets. Results measuring the out-of-sample fit of each aggregation method relative to the benchmark model are reported in Table 1.

[Table 1 here]

The simulation produced an ANN that can generate an MSE over the forecast set data which is 69% lower than that produced by the linear benchmark model. This outcome suggests ANNs can generate substantial gains; a result contrasting that of Church and Curram (1996). In practice, we may not be able to find this 'optimal' performer as out-of-sample data is often not available at the time we need to produce forecasts.

Turning to the aggregation methods described above (which rely solely on in-sample data), the results in Table 1 suggest that the simple average of all ANNs, the average of the top 25% in-sample performers, and the error-based weighting method perform as well as if not mildly better than the linear benchmark; a result supporting the weak benefits of ANNS found in Church and Curram (1996). The outperformance weighting method, however, produces substantially more accurate forecasts than the linear benchmark. Taken together, these outcomes illustrate that a diverse population of imperfect individuals can generate a reasonable picture of reality. Further, ample improvements in pattern reproduction are achievable when the *frequency* of success is accounted for in combining forecasts rather than overall precision.

With a linear ANN structure, looking at the best in-sample performer (an *expert*) and using their ANN alone for out-of-sample forecasting produces the lowest MSE compared to the other aggregation methods. Note that this expert ANN is *not* the best out-of-sample forecaster. Further, the performance of the expert exceeds that of the outperformance weighting by 5.26% of the linear benchmark MSE – a small margin. Nonetheless, without access to out-of-sample data, the forecasts of the experts are most reliable when agents employ linear neural networks.

D. Homogeneous non-linear network structure

Next, non-linearity is accommodated. Table 2 reports the performance of ANNs with fully-connected inputs and $H > 0$ non-linear nodes. It is assumed that every ANN is homogeneous in structure. As in the previous experiment, individual ANNs differ only by the randomly selected training and validation sets used for training.

[Table 2 here]

The results in Table 2 show that adding non-linearity can increase the accuracy of *individual* ANNs at generating out-of-sample patterns. For example, with $H = 1$ non-linear neurons, the 'optimal' out-of-sample performer produces an MSE that is 81% lower than the linear benchmark; an improvement upon the 'optimal' linear ANN in the previous section (which was 69% more accurate than the benchmark). This outcome further substantiates the benefits of ANN algorithms over traditional methods: non-linearity can produce more precise consumption patterns.

As above, the simple average of all ANNs, the average of the top 25% in-sample performers, and the error-based method are mildly more accurate than the linear benchmark. Further, the outperformance aggregation method generates substantial improvements for a low number of non-linear neurons. While this aggregation method performed better when there were no non-linear neurons, the reduction in accuracy is rather small (7.02% of the linear model's MSE).

Unlike the previous experiment, however, the best in-sample performer no longer reasonably produces out-of-sample patterns. It is likely this occurs because the non-linear ANN structure of the 'expert' allowed them to excel at learning patterns in their own training set which differ substantially from those in the forecast set. These results suggest it is better to rely on the combined information of many than the expertise of a single forecasting method when non-linearity is present.

We can induce additional diversity by making inputs to non-linear neurons semi-connected (done by randomly forcing selected ω-weights to zero[11]). In these simulations, the number of non-linear neurons is identical for each ANN and ANNs continue to differ by their randomly selected training and validation sets. Simulation results are reported in Table 3.

[Table 3 here]

The results from this experiment make the same implications as those in the experiment above when inputs were fully connected: (1) the simulation produces 'optimal' out-of-sample forecasters that significantly improve upon the linear model; (2) the simple average of all ANNs, the average of the top 25% in-sample performers, and the error-based method outperform the linear benchmark mildly; (3) the best in-sample performer no longer produces accurate forecasts consistently; and (4) the outperformance aggregation method continues to perform well. These outcomes indicate that the ANN framework is robust to diversity of input information.

E. Heterogeneous non-linear network structure

To further evaluate the power of heterogeneity in producing reliable forecasts, added diversity can be imposed on the internal structure of individual neural networks. For this exercise, the number of non-linear neurons for each ANN is assigned randomly (done by forcing randomly selected γ-weights to zero). Both fully-connected inputs and semi-connected inputs (in which case ANNs also differ according to the inputs fed into each non-linear neuron) are considered. To allow for differing degrees of non-linearity, both a low number and a high number of potential non-linear neurons is simulated (maximum $H = 5$ and maximum $H = 10$ respectively). Results are reported in Table 4.

[Table 4 here]

As in previous experiments, the ANN framework with structural heterogeneity can produce 'optimal' out-of-sample forecasters which are more precise than the linear benchmark. Further, the simple average of all ANNs, the average of the top 25% in-sample performers, and the error-based

[11] Note that it is possible for some non-linear neurons to be completely severed from input data, thus rendering the neuron inactive.

method continue to perform as well as if not mildly better than the linear benchmark. The 'expert' at in-sample forecasting predicts out-of-sample patterns quite poorly, however, and the outperformance aggregation method only produces substantial gains when inputs are semi-connected and the number of potential non-linear neurons is low. Combined with the results in the previous sections, these outcomes indicate that the construction and combination of ANN forecasts generates improved accuracy in the presence of heterogeneity in access to information (either the distribution of training/validation set or availability of input data), but not diversity of non-linear network structure.

F. Using out-of-sample results to select aggregation method parameters

The experiments in the previous sections used arbitrarily chosen parameter values for w (in the error-based aggregation method) and σ (in the outperformance aggregation method). With computational simulation, we can use out-of-sample data to identify alternative values for these parameters which reduce forecast set MSEs.[12] Table 5 (a – c) reports simulation results for the ANN structures described in parts (C) and (D) above using alternative values of σ and w. For the error-based weighting method, weights are computed for different values of w (w = 5, 10, 15, 20, 30, 40, 50) and the MSE results of the subsequent social forecast are reported. For the best period-performers, an optimising algorithm determines the value of *σ* which produces a social forecast generating the lowest MSE over the forecast set (=σ**)*.

For the error-based weighting method, the best choice of w is low compared to the number of periods available (i.e. we should focus on how well each ANN performs during the most recent periods when assigning its weight in the social forecast). At w = 15, aggregate forecasts produced from the error-based weighting method can generate MSEs 37% to 40% lower than the linear benchmark; a striking improvement upon the mild success of the error-based weighting method in the previous experiments (w = 62). This outcome does not depend on the linearity of the ANN nor on input connectedness. Note that w = 10 and w = 20 produce MSEs of a similar accuracy (less than 1% less accurate than w = 15 in many cases). Non-linear ANNs with varying degrees of complexity (H = 1:5) also produce similar MSEs. These results imply the value of w matters somewhat in achieving accurate out-of-sample estimates from the error-based aggregation method, but there is a range of choices for w and H which can perform equally well.

12 In practice, out-of-sample data is usually not available. Hence, we cannot establish 'optimal' values for w and σ beforehand. The following exercise identifies what values of these parameters we *should have* chosen in the previous experiments to generate the most accurate forecasts.

In the outperformance weighting method, the forecast sample MSEs produced are also quite low (32% to 40% lower than the benchmark model). However, to achieve this fit σ must be relatively large (the top 36% to 80% performers need to score points). In general, σ^* increases with the number of non-linear neurons. Recall that the best aggregation method in the previous section, which is the outperformance method with $\sigma = 25\%$, improved upon the linear benchmark by 39% for linear ANNs, 32% for non-linear ANNs with fully-connected inputs and 35.5% for non-linear ANNs with semi-connected inputs. The 'optimised' outperformance measure ($\sigma^* = 36\%$, 52% and 48% respectively) generates corresponding further improvements of only 1.1%, 7.2% and 3.3% for these models. These extra improvements are quite mild, indicating that our uninformed choice of σ in the previous experiments yielded fairly reasonable estimates.

[Table 5 here]

G. Discussion and applications

Under many circumstances, the artificial neural networks considered in this study can predict aggregate consumption patterns better than a standard linear econometric model. In many cases, the inclusion of diversity and the aggregation of private knowledge into social knowledge aids in producing accurate out-of-sample patterns. Although the linear benchmark is naively chosen, this result supplements those of Church and Curram (1996). We can interpret the ANN meta-algorithm described above as creating a community of individuals whose exclusive understanding of the environment is combined. From this perspective, the elements needed for a more interactive agent-based framework to function well in generating macroeconomic phenomena seen in real data are implied.

If out-of-sample data is available, an 'optimal' ANN can be identified which improves upon the linear benchmark model by as much as 81%. This dramatic improvement is largest when ANNs feature fully-connected inputs and a single non-linear neuron, but is also strong for all other ANN specifications. In an advanced framework, agents that produce patterns poorly can update their personal ANN by colluding with this individual. Alternatively, the optimal individual can be made to play a more significant role in learning via reproductive processes.[13] Interactivity with or by the

[13] This refers to the use of genetic algorithms (GA) in learning about optimal decisions rules. See Arifovic (1994) for an example of this type of learning procedure.

optimal agent saves energy within the system since obtaining and re-processing information can be foregone, yet patterns that more closely resemble real data can be generated.

When out-of-sample data cannot be used to identify an 'optimal' ANN, we can combine the heterogeneous forecasts to produce fairly accurate patterns using wisely chosen weighted aggregation. When ANNs are linear, identifying an 'expert' at in-sample forecasting and assigning all weight to their method generates the most accurate out-of-sample patterns. Large improvements over the linear benchmark achieved, but these gains are not as great as those achieved by the 'optimal agent'. Nonetheless, this 'expert' agent can be relied upon in a more interactive framework to take up the role the optimal agent would have played (as described above).

Note that combining forecasts based on period-by-period top performance also produces significant improvements over the benchmark model. In fact, when ANNs accommodate non-linearity, the period-by-period top performance weighting method is best (and the predictions of 'experts' is often the worst). These results are robust to changes in the connectedness of inputs (but not to diversity in the number of non-linear neurons). An interactive agent embedded in large-scale model may therefore improve their own predictions by evaluating the relative frequency of their peer's successes. Doing this is more sophisticated than equally weighting their peer's understanding of the environment (as was done for the simple-average weighting regime) and requires less detailed information about the size of their peer's errors (which is needed for the error-based weighting regime).

Further gains can be made by wisely selecting parameters in the error-based weighting method and the outperformance method. Although optimal choices for σ in the period-by-period top performance method generate improvements over the uninformed choice (σ = 25%), the value of this parameter is fairly large and the gains are rather small. With a fairly low w, the error-based weighting method can be made to produce accurate out-of-sample patterns from a variety of network structures. This can limit the amount of detailed information than an agent would need to collect about the mistakes made by their peers to increase the accuracy of their forecasts in an advanced interactive model.

4. Concluding Remarks

In this study, ANNs are shown to produce realistic patterns of aggregate per-capita consumption in the New Zealand economy. In simulation, it is possible to find ANNs which substantially outperform a linear econometric model at out-of-sample forecasting (by 69% for linear ANNs, by 46% to 81% for fully-connected non-linear ANNs, and by 63% to 72% for semi-connected non-linear ANNs). Finding these 'optimal' performers requires knowledge about out-of-sample performance which, in many cases, is unavailable. Often, the best ANN at in-sample pattern production is not the best out-of-sample forecaster.

Because ANNs accommodate diversity and can be trained in large numbers, we can combine their imperfect, heterogeneous forecasts into a social forecast. This task utilises in-sample data only and can generate reasonably accurate forecasts. Using an aggregation method in which an ANN that is more frequently a top performer receives more weight in the construction of the social forecast is shown to consistently improve upon the linear benchmark (by 43% for linear ANNs, by 20% to 32% for fully-connected non-linear ANNs with simple internal structures [low H], and by 20% to 36% for semi-connected non-linear ANNs with simple internal structures). Simpler aggregation methods (equal weighting of all ANNs or equal weighting of top overall in-sample performers) mildly outperform the benchmark model as to more complex methods involving the size of in-sample errors.

Several avenues for future work appear. The linear benchmark model specified in this study is a useful starting point, but is simply chosen. Further comparisons to more complicated benchmark models can be performed and is left for future work. Taken at face value, however, ANNs seem to produce aggregate consumption patterns fairly accurately; an outcome which substantiates their use in more interactive, agent-based computational models. These models, known for their ability to feature realism and complexity, can generate aggregate consumption patterns from the bottom-up based on how information is shared between heterogeneous peers. The analysis above alludes to methods for successfully incorporating ANNs into this type of model. The next step in this line of research is to generate populations of consumers who think less like economists and more like ordinary people.

References

Arifovic, J. (1994). Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control*, 18(1), 3-28.

Aminian, F., Suarez, E. D., Aminian, M. & Walz, D. T. (2006). Forecasting economic data with neural networks. *Computational Economics*, 28(1), 71-88.

Barron, A. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930-945.

Bates, J. M. & Granger, C. W. J. (1969). The combination of forecasts. *Operational Research Quarterly*, 20(4), 451-468.

Beltratti, A., Margarita, S., & Terna, P. (1996). *Neural Networks for Economics and Financial Modelling*. London: International Thomson Computer Press.

Bunn, D. W. (1975). A Bayesian approach to the linear combination of forecasts. *Operational Research Quarterly*, 26(2), 325-329.

Church, K. B. & Curram, S. P. (1996). Forecasting consumers' expenditure: A comparison between econometric and neural network models. *International Journal of Forecasting*, 12(2), 255-267.

Clemen, R. T. (1989). Combining forecasts: A review and notated bibliography. *International Journal of Forecasting*, 5(4), 559- 583.

Cooper, J. C. B. (1999). Artificial neural networks versus multivariate statistics: An application from economics. *Journal of Applied Statistics*, 26(8), 909-921.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303-314.

De Menezes, L. M., Bunn, D. W. & Taylor, J. W. (2000). Review of guidelines for the use of combined forecasts. *European Journal of Operational Research*, 120(1), 190-204.

Detienne, K. B., Detienne, D. H. & Joshi, S. A. (2003), Neural networks as statistical tools for business researchers. *Organizational Research Methods*, 6(2), 236-265.

Epstein, J. M. (1999), Agent-based computational models and generative social science. *Complexity*, 4(5), 41-60.

Gaffeo, E., Delli Gatti, D., Desiderio, S., & Gallegati, M. (2008). Adaptive microfoundations for emergent macroeconomics. *Universita Degli Studi Di Trento Working Paper No. 2/2008.*

Gatti, D., Di Guilmi, C., Gaffeo, E., Giulioni, G., Gallegati, M., & Palestrini, A. (2005). A new approach to business fluctuations: heterogeneous interacting agents, scaling laws and financial fragility. *Journal of Economic Behavior and Organization*, 56(4), 489-512.

Gonzalez, S. (2000). Neural networks for macroeconomic forecasting: A complementary approach to linear regression models. *Department of Finance Canada Working Paper 2000-07.*

Granger, C. W. J. & Ramanathan, R. (1984). Improved methods of combining forecasts. *Journal of Forecasting*, 3(2), 197-204.

Hashem, S. (1997). Optimal linear combinations of neural networks. *Neural Networks*, 10(4), 599-614.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks,* 4(2), 251-257.

Hornik, K., Stinchcombe, M. & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Hornik, K., Stinchcombe, M. & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5), 551-560.

International Monetary Fund [IMF] (2011). e-Library Data. Available at http://elibrary-data.imf.org/.

Ives, A. R. & Carpenter, S. R. (2007). Stability and diversity of ecosystems. *Science*, 317(58), 58-62.

Jain, A. K. & Mao, J. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3), 31-44.

Kourentzes, N. & Crone, S. (2010). Advances in forecasting with artificial neural networks. *Lancaster University Management School Working Paper 2010/023.*

Kuan, C.-M. & White, H. (1994). Artificial neural networks: An econometric perspective. *Econometric Reviews*, 13(1), 1-91.

McCann, K. S. (2000). The diversity-stability debate. *Nature*, 405, 228-233.

Mirowski, P. (2007). Markets come to bits: Evolution, computation and markomata in economic science. *Journal of Economic Behavior and Organization*, 63(2), 209-242.

Newbold, P. & Granger, C. W. J. (1974). Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society, Series A (General)*, 137(2), 131-165.

Organisation for Economic Co-operation and Development [OECD] (2011). OECD.StatExtracts. Available at http://stats.oecd.org.

Raberto, M., Teglio, A. & Cincotti, S. (2008). Integrating real and financial markets in an agent-based economic model: An application to monetary policy design. *Computational Economics*, 32(1-2), 147-162.

Tesfatsion, L. (2002). Agent-based computational economics: Growing economies from the bottom up. *Artificial Life*, 8(1), 55-82.
Tesfatsion, L. (2005). Agent-based computational modeling and macroeconomics. *ISU Economic Report 05023: July 2005.*
Tesfatsion, L. & Judd, K. (eds) (2006). *Handbook of Computational Economics, V.2: Agent-based Computational Economics*. Amsterdam: Elsevier.
Vellido, A., Lisboa, P. J. G. & Vaughan, J. (1999). Neural networks in business: A survey of applications (1992 – 1998). *Expert Systems with Applications*, 17(1), 51-70.
Warner, B. & Misra, M. (1996). Understanding neural networks as statistical tools. *The American Statistician*, 50(4), 284-293.
Zhang, G., Patuwo, B. E. & Hu, M. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), 35-62.

Tables and Figures

Figure 1– A perceptron.

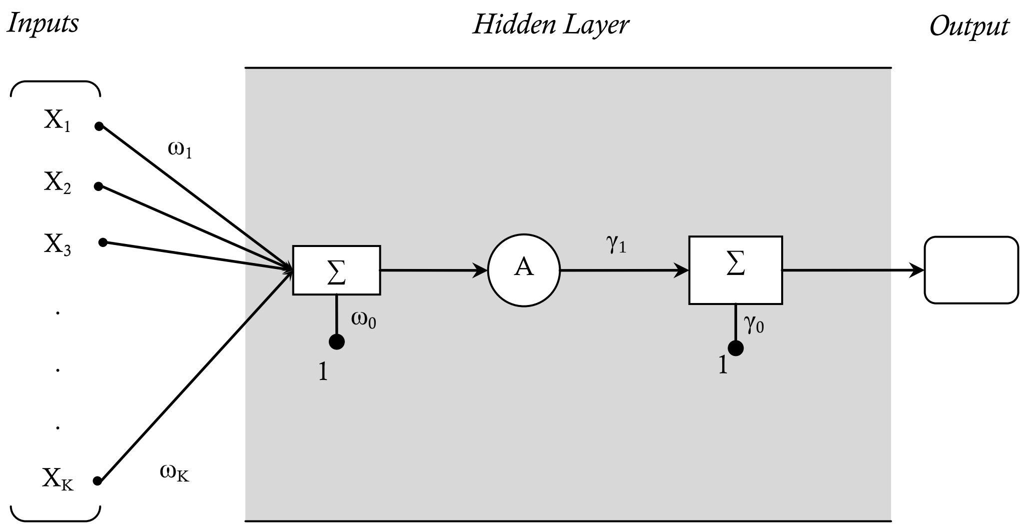


Figure 2 – A simple ANN with 1 linear and H non-linear neurons.

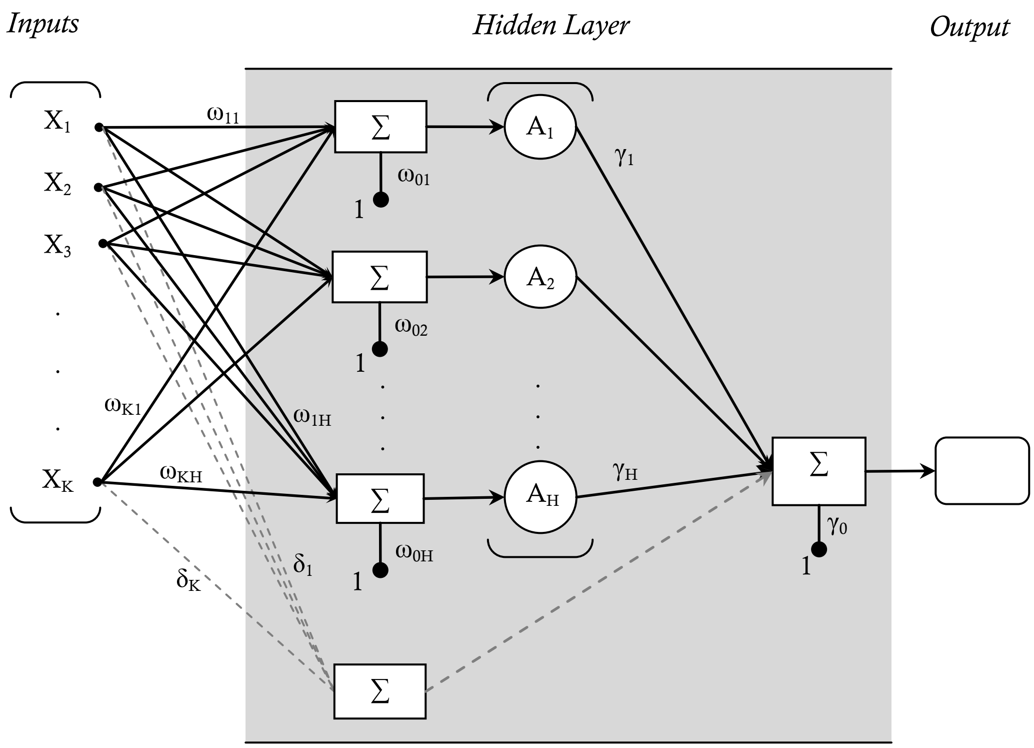
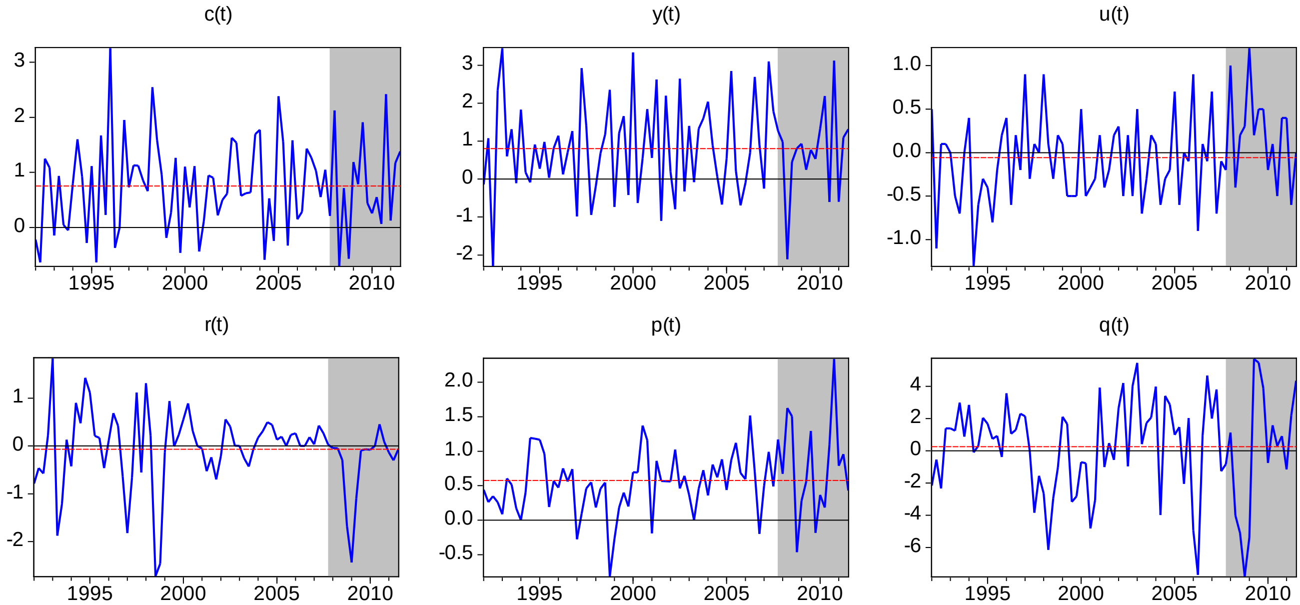


Figure 3 – Empirical regularities of the New Zealand economy (1992q1-2011q3).



Source: IMF (2011), OECD (2011).
Notes: Forecast set (shaded). Sample average (---).

Table 1 – Mean squared forecast error of ANNs with no non-linear neurons (measured as % deviation from linear benchmark).

Aggregation Method	Forecast Set MSE
Simple Average	-7.98
Best In-sample Performer	**-44.30**
Top 25% In-sample Performers	-12.94
Error-based Weighting (w=62)	-11.62
Period-by-period Outperformance (σ = 25%)	-39.04
Best Out-of-sample Performer	*-68.86*

Notes: Best performing aggregation method is bolded.

Table 2 – Mean squared forecast error of homogeneous non-linear ANNs with fully-connected inputs (measured as % deviation from linear benchmark).

	Forecast Set MSE					
Aggregation Method	H = 1	H = 2	H = 3	H = 4	H = 5	H = 10
Simple Average	-10.53	-10.53	1.54	-6.80	-2.63	-3.95
Best In-sample Performer	-0.66	-3.29	140.13	45.39	56.80	70.61
Top 25% In-sample Performers	-10.53	-9.21	-8.99	-8.99	-7.24	1.75
Error-based Weighting (w=62)	-10.96	-11.84	-7.24	-8.77	-8.33	-3.07
Period-by-period Outperformance (σ = 25%)	**-32.02**	-23.90	-20.39	1.97	-21.27	203.95
Best Out-of-sample Performer	*-80.70*	*-68.20*	*-72.37*	*-67.98*	*-68.20*	*-46.49*

Notes: Best performing aggregation method is bolded.

Table 3 – Mean squared forecast error of homogeneous non-linear ANNs with semi-connected inputs (measured as % deviation from linear benchmark).

	Forecast Set MSE					
Aggregation Method	H = 1	H = 2	H = 3	H = 4	H = 5	H = 10
Simple Average	-0.66	-5.48	-5.48	-11.62	-6.80	0.44
Best In-sample Performer	-24.34	8.99	275.44	142.76	19.52	298.25
Top 25% In-sample Performers	-7.24	-9.87	-9.87	-14.69	-12.50	-6.36
Error-based Weighting (w=62)	-6.80	-9.65	-9.43	-12.72	-11.62	-3.73
Period-by-period Outperformance (σ = 25%)	**-35.53**	-32.68	-29.82	-21.93	-19.96	18.64
Best Out-of-sample Performer	*-67.32*	*-69.52*	*-71.71*	*-67.54*	*-63.82*	*-66.45*

Notes: Best performing aggregation method is bolded.

Table 4 – Mean squared forecast error of heterogeneous non-linear ANNs (measured as % deviation from linear benchmark).

	Forecast Set MSE			
	Fully-connected		Semi-connected	
Aggregation Method	Max H = 5	Max H = 10	Max H = 5	Max H = 10
Simple Average	-8.55	-8.99	-2.63	0.88
Best In-sample Performer	160.53	92.32	56.80	223.90
Top 25% In-sample Performers	-4.39	-4.17	-7.24	-2.41
Error-based Weighting (w=62)	-6.36	-5.26	-8.33	-3.29
Period-by-period Outperformance (σ = 25%)	27.63	211.40	**-21.27**	18.64
Best Out-of-sample Performer	*-75.23*	*-66.67*	*-68.20*	*-69.08*

Notes: Best-performing aggregation method is bolded.

Table 5 – Optimised mean squared forecast error of fully- and semi-connected ANNs (measured as % deviation from linear benchmark) for error-based and period-performance aggregation regimes.

(a) Linear network structure

	Forecast Set MSE
Error-based Weighting	
w = 5	-28.51
w = 10	-38.82
w = 15	-39.25
w = 20	-38.16
w = 30	-32.02
w = 40	-13.16
w = 50	-18.42
Period-by-period Outperformance (σ = σ*)	**-40.13**
σ*	36%

Notes: Best performing aggregation method is bolded. Neural networks that improve upon the benchmark model by more than 35% are highlighted.

(b) Non-linear network structure with fully-connected inputs

	Forecast Set MSE					
	H = 1	H = 2	H = 3	H = 4	H = 5	H = 10
Error-based Weighting						
w = 5	-28.29	-28.73	-27.19	-26.97	-26.32	-20.61
w = 10	-39.04	-39.04	-37.50	-37.28	-36.84	-33.77
w = 15	-39.69	**-39.91**	-38.60	-38.38	-37.50	-34.65
w = 20	-38.60	-39.04	-37.72	-37.72	-36.62	-34.43
w = 30	-32.68	-33.77	-32.24	-32.89	-30.70	-29.82
w = 40	-12.94	-14.04	-10.09	-11.40	-10.75	-7.02
w = 50	-18.86	-20.18	-16.89	-18.64	-16.67	-14.47
Period-by-period Outperformance (σ = σ*)	-39.25	-38.60	-37.28	-36.84	-35.53	-31.36
σ*	52%	58%	56%	65%	56%	80%

Notes: Best performing aggregation method is bolded. Neural networks that improve upon the benchmark model by more than 35% are highlighted.

(c) Non-linear network structure with semi-connected inputs

	Forecast Set MSE					
	H = 1	H = 2	H = 3	H = 4	H = 5	H = 10
Error-based Weighting						
w = 5	-27.19	-28.95	-28.07	-28.95	-28.95	-25.00
w = 10	-37.72	-38.60	-38.82	-39.04	-38.16	-36.40
w = 15	-38.38	-39.25	-39.47	**-39.91**	-39.04	-37.28
w = 20	-37.06	-38.16	-38.16	-39.04	-38.16	-36.40
w = 30	-30.26	-32.02	-32.02	-33.99	-33.33	-30.70
w = 40	-8.99	-11.62	-11.62	-14.47	-13.82	-7.02
w = 50	-14.91	-17.76	-17.76	-20.61	-20.18	-14.69
Period-by-period Outperformance (σ = σ*)	-38.38	-38.82	-38.38	-38.60	-37.72	-35.96
σ*	42%	48%	52%	59%	60%	65%

Notes: Best performing aggregation method is bolded. Neural networks that improve upon the benchmark model by more than 35% are highlighted.