CONTEXT SENSITIVITY WITH NEURAL NETWORKS IN FINANCIAL DECISION PROCESSES

Charles Wong, Boston University
Massimiliano Versace, Boston University

ABSTRACT

Context modifies the influence of any trading indicator. Ceteris paribus, a buyer would be more cautious buying in a selling market context than in a buying market. In order for automated, adaptive systems like neural networks to better emulate and assist human decision-making, they need to be context sensitive. Most prior research applying neural networks to trading decision support systems neglected to extract contextual cues, rendering the systems blind to market conditions. This paper explores the theoretical development and quantitative evaluation of context sensitivity in a novel fast learning neural network architecture, Echo ARTMAP. The simulated risk and cost adjusted trading results compare very favorably on a 10-year, random stock study against the market random walk, regression, auto-regression, and multiple neural network models typically used in prior studies. By combining human trader techniques with biologically inspired neural network models, Echo ARTMAP may represent a new tool with which to assist in financial decision-making and to explore life-like context sensitivity.

JEL: G11, G17

KEYWORDS: Recurrent neural networks, context sensitivity, financial forecasting, investment decisions

INTRODUCTION

Stock prices refer to the latest mutually decided transaction price and time between a voluntary buyer and seller. If the stock prices over time are increasing, they indicate that the buying interest exceeds the selling interest. This signals a bullish or optimistic market context favorable to investment, all else equal. A successful trader (Schwager, 1994) often considers the underlying market sentiment when making decisions. This sensitivity to context in decision-making is one of the hallmarks of human intelligence (Akman, 2002).

Human subjects often treat similar tasks differently under different contexts (e.g. Carraher, Carraher, & Schliemann, 1985; Bjorklund & Rosenblum, 2002). Working memory allows features to be tracked over time to extract a context (Kane & Engle, 2002; Baddeley & Logie, 1999). Context sensitivity theoretically enables the decision-maker to disambiguate different feature inputs that may be identical at single points in time (Kane & Engle, 2002).

To better model human decision-making with context sensitivity, an automatic decision system must be context sensitive (see Figure 1, left). Tracking the price over time to determine whether the market is uptrending (bullish) or downtrending (bearish) intuitively provides contextual cues (Schwager, 1994). This paper introduces a context sensitive neural network decision system, Echo ARTMAP.

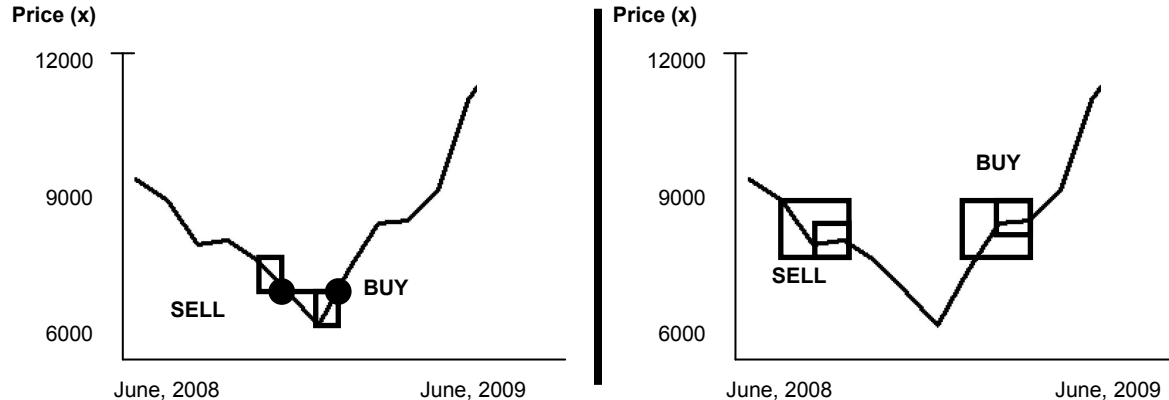Figure 1: Extracting Contextual Patterns from Historical Prices

Figure 1: One-year daily prices for the Dow Jones Industrial Average. (left) Identical point inputs (the horizontal line indicates same price) lead to different classes. The boxes show different contexts that can disambiguate the two inputs. (right) Identical short duration inputs (small boxes) lead to different classes. The longer duration input (large boxes) can disambiguate the two inputs.

The remaining sections of this paper divide as follows: Section II reviews the recent literature applying neural models to financial time series. Section III provides a brief review of the default ARTMAP neural network kernel mechanism as a base for later extension; Section IV demonstrates how to theoretically adapt the ARTMAP to context sensitivity; Section V outlines the data and methodology; Section VI provides the results and discussion; and Section VII contains concluding remarks.

LITERATURE REVIEW

Neural networks are biologically inspired, automated, and adaptive analysis models that can better accommodate non-linear, non-random, and non-stationary financial time series than alternatives (e.g. Lo, 2001; Lo & Repin, 2002; Lo, 2007; Gaganis, Pasiouras, & Doumpos, 2007; Yu, Wang, & Lai, 2008). Much research in the past decade applies them to financial time series with typically strong and compelling empirical results. Our survey of 25 studies published in the past decade (Wong & Versace, 2011a) divides the network models into four categories with breakdowns: (1) slow learning context blind, 56%; (2) fast learning context blind, 28%; (3) slow learning context sensitive, 16%; and (4) fast learning context sensitive, 0%.

Saad, et al. (1998), compare multiple neural models on financial forecasting accuracy. Data included the daily prices for 10 different stocks over one year representing high volatility, consumer, and cyclical industries. Models included a fast learning radial basis function, a slow learning backpropagation network, and a slow context sensitive recurrent backpropagation model. Results showed that all three networks provided similar performance.

Versace, et al. (2004), apply genetic algorithms to determine the network, structure, and input features for financial forecasting. Data included 300 daily prices from the Dow Jones Industrial Average. Features included a series of technical indicators. Models included both fast learning and slow context sensitive networks. Results showed that using a genetic algorithm to choose and design the network and features could generate significant accuracy in trading decisions.

Sun, et al. (2005), use a fast learning neural model for time series. Data included two years of daily S&P 500 and Shanghai Stock Exchange indices. Their model updates the fast learning radial basis function with a novel Fisher's optimal partition algorithm for determining basis function centers and sizes with

dynamic adjustment. Results show that the updated fast learning model provides significant improvements.

Zhang, et al. (2005), explore slow learning backpropagation networks for financial time series analysis. Data included seven years of daily Shanghai Composite Index data. Enhancements to the slow learning model apply normalization and kernel smoothing to reduce noise. Results show that the slow learning models consistently outperformed the buy-and-hold strategy.

Chen & Shih, (2006), apply neural network models to six Asian stock markets, including Nikkei, Hang Seng, Kospi, All Ordinaries, Straits Times, and Taiwan Weighted indices. Features included five technical analysis indicators. Models included fast learning support vector machines and the slow learning backpropagation. Results show that the neural models outperformed autoregression models, especially with respect to risk. The fast learning models also appear to outperform the slow learning models.

Ko & Lin, (2008), apply a modified slow learning backpropagation model to a portfolio optimization problem on 21 companies from the Taiwan Stock Exchange for five years. Results show that their resource allocation neural network outperformed the buy-and-hold considerably, averaging 15% to 5% yearly gains.

Freitas, et al. (2009), apply an enhanced slow learning context sensitive model to weekly closing prices for 52 Brazilian stocks for 8 years. Their model uses recurrence to increase emphasis towards more recent data. Results show that their model produced results in excess of the mean-variance model and the market index with similar levels of risk.

In all remaining cases, the neural networks appear outperform random walk or buying-and-holding approaches to the financial time series. The results appear robust regardless of network learning rule or context sensitivity. The bias towards slow learning networks probably reflects their earlier availability (Rumelhart, Hinton, & Williams, 1986). Of the studies employing context sensitive models, all relied on slow learning rules incorporated in Jordan and Elman networks (e.g. Versace et al, 2004; Yu, Wang, & Lai, 2008; Freitas, Souza, & Almeida, 2009; Jordan, 1986; Elman, 1990). Studies directly comparing fast learning, slow learning, and slow learning context sensitive networks have found no significant differences in empirical results (e.g. Saad et al, 1998).

This paper explores the disagreement between the intuition supporting the importance of context sensitivity and the empirical results showing no differential benefit relative to existing neural network models. The bulk of the studies indicate existing models tend not to incorporate fast learning with context in finance. Therefore, this paper introduces a novel context sensitive fast learning network, Echo ARTMAP, for transparent analysis (Moody & Darken, 1989; Carpenter, Grossberg, & Reynolds, 1991; Parsons & Carpenter, 2003). The base fast learning component model is ARTMAP (Amis & Carpenter, 2007) from the Adaptive Resonance Theory class of models. While ARTMAP is not a perfect blend of all existing fast learning characteristics (e.g. it differs in learning vs. Radial Basis Function networks), it can be regarded as a general purpose, default network that automatically adapts and scales its topology to a generic dataset (e.g. Carpenter, 2003). For this paper, benchmarks include random walk, regression, auto-regression, a slow learning backpropagation (Rumelhart, Hinton, & Williams, 1986), a fast learning ARTMAP (Amis & Carpenter, 2007), and a slow learning context sensitive model (Jordan, 1986).

 Kernel Review for a Typical Fast Learning Model

Slow learning networks possess hidden layers that have opaque representations relating inputs to outputs. In contrast, fast learning allows immediate and transparent convergence for independent storage layer

nodes. ARTMAP is a type of fast learning network that was inspired by biological constraints and can be adapted to a variety of uses. Extensive literature shows its capabilities and interprets its mathematical bases (e.g. Amis & Carpenter, 2007; Parsons & Carpenter, 2003). Figure 2 (left) shows the default ARTMAP flow diagram.

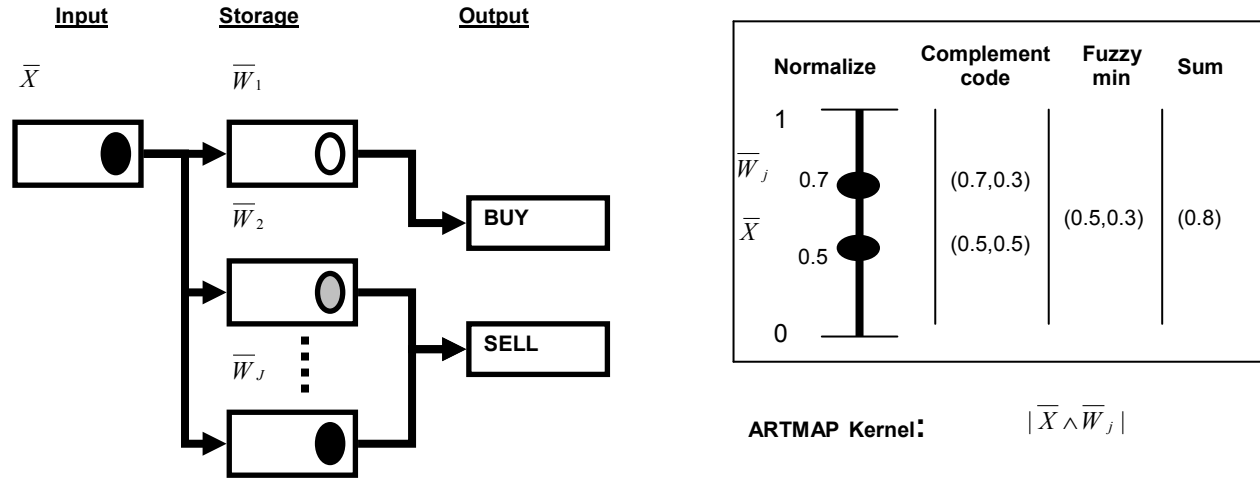Figure 2: A Typical General Purpose Fast Learning Neural Model, ARTMAP

Figure 2: (left) A default ARTMAP network diagram showing the three-layer architecture. For a particular input pattern, $\overline{X}$ *the ARTMAP kernel finds the most similar stored pattern,* $\overline{W}_j$ *which maps to a specific output node. Boxes, or nodes, represent patterns (responses in the output layer) and circles represent individual components of the pattern. (right) This is an example of the ARTMAP kernel calculating the similarity between an input pattern and a single stored pattern. See the text for details.*

There are three layers in the default ARTMAP network. The input layer receives input patterns, each represented by a vector with one or more components, $\overline{X}$. Given this vector, the network finds the most similar vector $\overline{W}_j$ from the storage layer, where $j = \{1...J\}$ and J is the number of storage nodes. The output layer node associated with the most similar storage node dictates the network response. The ARTMAP kernel, which is a function that determines the similarity between vectors (Bishop, 2006), models pattern recognition as per equation (1):

$$T_j = |\overline{X} \wedge \overline{W}_j|, \tag{1}$$

where T_j is the similarity score for storage node j. The kernel procedure has four steps: normalize all vector component values to between 0 and 1; complement code both vectors such that $\overline{X} = (x_1, 1 - x_1)$; apply the fuzzy min operator (^) on the vectors; and sum (‖). For example, given a normalized input value of 0.5 and a particular normalized storage node of 0.7, the complement codes would be (0.5, 0.5) and (0.7, 0.3). The fuzzy min would be the lesser of each component, or (0.5, 0.3) and their sum would be 0.8, which as a normalized value can also be read as 80% similar. The default ARTMAP learning rules that update and add storage layer nodes with their associated output nodes are not modified and are not treated here. For references on previously published ARTMAP models, please see http://techlab.bu.edu. The following section provides the theoretical modifications to this kernel.

Extracting Context

The approach taken here explores fast learning network rules with context sensitivity. Figure 3 shows how a fast learning ARTMAP model can be modified to process time patterns in the data with three steps via input delays, decays, and output-to-input recurrence to create the novel Echo ARTMAP model.

Figure 3: Echo ARTMAP Extends the General Purpose Neural Model for Context

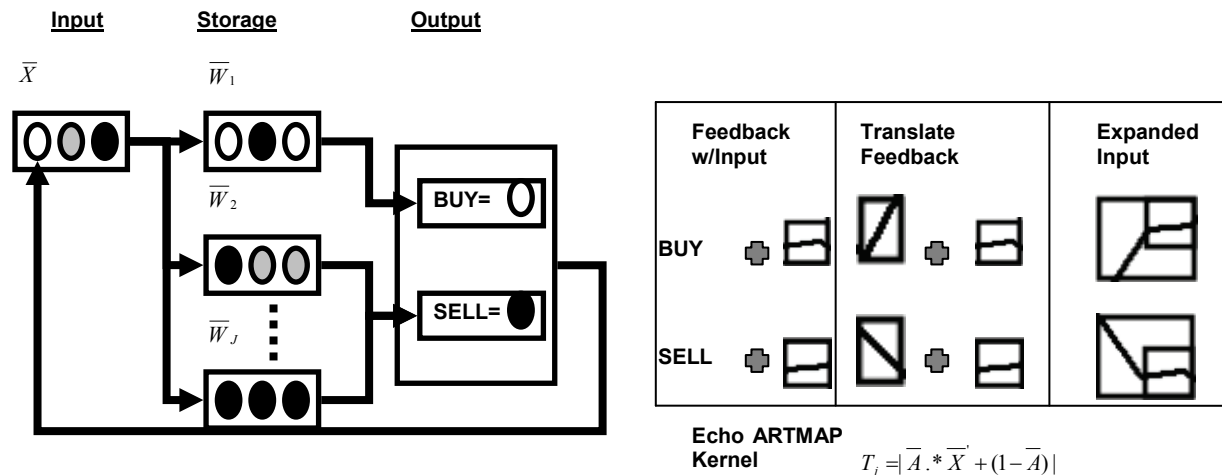$$T_j = | \overline{A} .* \overline{X}' + (1 - \overline{A}) |$$

Figure 3: (left) The full Echo ARTMAP architecture with time delay, decay, and recurrence. See text for the breakdown of the three steps. (right) Excerpted from figure 1, the feedback provides additional input information from past storage values. Translating the feedback back into its component pattern allows more information to be input into the network. This example assumes the patterns in Figure 1, left, have already been stored, for instance allowing the feedback Buy value to be translated into an uptrend. This process can be repeated infinitely, allowing greatly expanded inputs.

Implementing input time delays allows an ARTMAP network to model one aspect of working memory. Figure 3 (left) shows the ARTMAP network from figure 2 with multiple components in each node, the right two being the same feature at different points in time. Similarity proceeds from equation (1), but depends on multiple points in time. Figure 4 (a and b) compares the influence of a given input over time when introducing input time delay. With no delay, the network at time *t* can only consider inputs from time *t*.

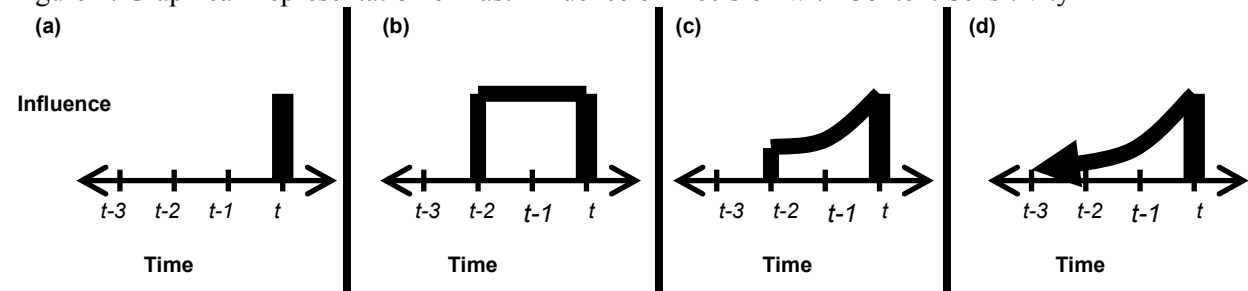Figure 4: Graphical Representation of Past Influence on Decision with Context Sensitivity

Figure 4: The influence of past inputs at time t. (a) At time t, a model with no time delay only considers the current input from t. (b) A model with a delay of 2 considers both the input from t and the past two inputs equally. (c) A model with delay of 2 with decay considers the input from t and the past two inputs, but with more emphasis on more current inputs. (d) A model with delay of 2 with decay and output-to-input recurrence theoretically considers all prior inputs albeit with very little emphasis on distant inputs in time.

Implementing time decay allows an ARTMAP network to model a more complex, non-stationary working memory. In a non-stationary data set, proximal points in time should have more influence than distal points in time (Hamilton, 1994). The underlying state or context is shifting over time, such that

feature values within the current state are more relevant. Equation (2) shows how to scale the contextual importance:

$$T_j = | \overline{A} .* \overline{X}' + (1 - \overline{A}) |, \tag{2}$$

where $\overline{X}' = \overline{X} \wedge \overline{W}_j$, or the component-wise collection after the first three steps in equation (1), $\overline{A} = (a_1, a_2, ... a_M)$, $0 <= a_m < \infty$, a is monotonically decreasing with m, $m = \{1..M\}$, and M is the number of time delays. The $(.*)$ operator indicates an array multiplication that performs element-by-element factoring. The vector $\overline{A}$ scales the importance of the components of $\overline{X}'$. Figure 4 (c) shows the scaling effect of monotonically decreasing A components as the influence of inputs from time t diminishes after time t. The $(1 - \overline{A})$ term ensures the T_j score has a maximum of 1. Although negative scores are possible, the default ARTMAP matching rule selects the highest positive T_j score (Amis & Carpenter, 2007).

Finally, implementing output-to-input recurrence is useful since it allows a compressed version of past inputs (i.e. as implied by the output decision based on these inputs) to remain available to influence decisions at future times. For example, a fixed small window in time may not fully capture a large pattern (Figure 1, right). A fixed large window captures non-pattern noise that leads to over-fitting. Output-to-input recurrence can simulate a dynamic window size. Figure 3 (right) shows a short pattern preceded by a recurrent output label. Since each output label associates with a specific storage pattern, using the label as a further input creates sensitivity to patterns that are composites of past short window patterns as needed, generating the profile in Figure 4(d). This combines the flexibility of long windows with the lower susceptibility to over-fitting of short windows. The novel Echo ARTMAP model performs this implicit composition of past inputs.

DATA AND METHODOLOGY

Financial Decision Making on a Random Sample of Stocks

Figure 5 shows the overall study framework for collecting and evaluating benchmark performance via risk-adjusted rates of return for the Echo ARTMAP network and benchmarks.

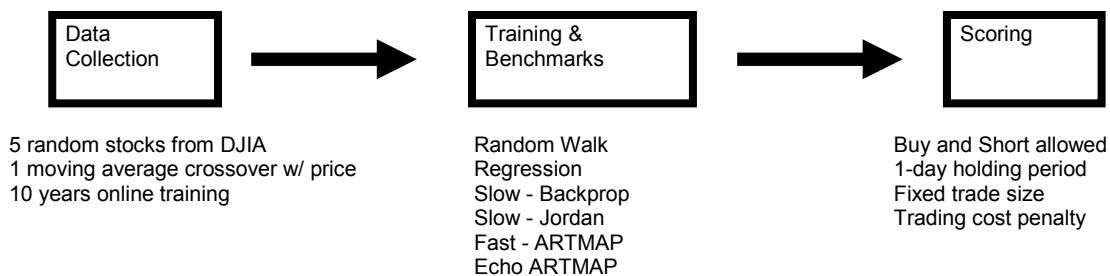Figure 5: Data and Methodology Overview

Data Collection	Training & Benchmarks	Scoring
5 random stocks from DJIA 1 moving average crossover w/ price 10 years online training	Random Walk Regression Slow - Backprop Slow - Jordan Fast - ARTMAP Echo ARTMAP	Buy and Short allowed 1-day holding period Fixed trade size Trading cost penalty

Figure 5: The outline process flow for evaluating the Echo ARTMAP and benchmarks. Data collection proceeds from http://finance.yahoo.com. This study uses ten years of online training. One moving average difference with the price provides the basic input. Trading costs and risk adjustment are included in fixed trading sizes (no trades are assumed to yield zero percent; gains and losses are replaced to maintain fixed trading size).

This study randomly selects five stocks from the Dow Jones Industrial Average index as of 1999 (American Express, Exxon-Mobil, IBM, JP Morgan Chase, and United Technologies Corporation) for daily online training spanning ten years from 2000-2009. This generates 12,500 sample data points,

which exceeds the sample size required for 95% confidence given estimated population standard deviation under both normal and non-normal assumptions (Higgins, 2004). Online means the fast learning networks continually expand their training set after testing on each trading day; the slow learning networks replicate this process by using rolling training window sizes of two years and averaging the results. Supervised classes derive from whether the forward one-day price change is positive, negative, or neutral. The single input feature uses a moving average period of 10 days subtracted from the current price. From this single feature, each benchmark model receives up to an 11-dimensional derived input set for each stock: 10 input delays from the single feature plus one from the benchmark's output-to-input recurrence where applicable.

The benchmarks include: an industry standard random walk; regression and auto-regression (Box & Jenkins, 1970); a slow learning static input neural network, backpropagation (Rumelhart, Hinton, & Williams, 1986); a fast learning static input neural network, ARTMAP (Amis & Carpenter, 2007); a slow learning context sensitive neural network, the Jordan network (Jordan, 1986); and the novel Echo ARTMAP fast learning context sensitive network.

For scoring purposes, buying, not trading, and selling short are allowed (i.e. 3-class predictor). Each decision lasts for one day. Position sizes are fixed, with gains being removed and losses being replaced. Not trading is valued at zero gains and zero costs. Round trip trading costs deduct 0.1% per active trading decision. To counter this trading cost, the supervised learning classifies trading days with daily variance of less than 1% as not trading. In addition, the Sharpe Ratio (Chartered Financial Analyst Institute, 2010) divides the average return by the standard deviation of the returns. This provides an additional, singular, and objective measure of the risk/reward profiles for each benchmark.

School Seminar Detection Based on Human Traffic Patterns

This paper uses the same six benchmarks to attempt to detect when a school seminar is taking place at a classroom building based on human foot traffic into and out of the building. The data consists of six-months of human foot traffic data from the University of California, Irvine, machine learning repository with ground truth event listing (http://ics.uci.edu). The goal of all benchmarks is to accurately detect the presence or absence of the seminars (i.e. a 2-class predictor). The static benchmarks (random walk, regression, backpropagation, and ARTMAP) can generate predictions at time t based only on the observations at time t. The context sensitive benchmarks (autoregression, Jordan network, and the novel Echo ARTMAP) can also consider previous observations. For evaluation purposes, this paper uses the receiver operating characteristic curve to provide a distribution-free metric of signal utility (Witten & Frank, 2005). This factors the frequency of the events into the final accuracy function.

The utility of this seminar detection problem relies on the fact that it is human driven, has a clear ground truth, has discrete states, and can be followed intuitively for parallel analysis with the financial data set.

Two Class Non-Temporal Circle-in-the-Square Data Set

This paper also uses the six benchmarks to explore context sensitive models on the two-class circle-in-the-square problem, which is a purely spatial, context-blind data set. Given a unit circle occupying exactly 50% of the area of a bounding square, the benchmarks need to predict if a randomly given point inside the square is also inside the circle. Intuitively, since each point is not related to the prior points, context sensitive models may attempt to establish a non-existent context and therefore perform poorly. This problem set allows empirical assessment of this intuition.

RESULTS

Table 1 shows the six benchmarks' average annualized performance over five random Dow Jones Industrial Average stocks over ten years. For reporting purposes, the ten-year period divides into three periods of 3.33 years each to demonstrate a possible range of results. The Sharpe Ratios provide single, numerical measures of risk and reward for each benchmark.

Table 1: The 10-Year Financial Data Set Annualized Gains per Benchmark

Decision Method	Annualized Gains			Sharpe Ratio
	2000-2003	2003-2006	2006-2009	
Random Walk	-1.4%	-0.6%	-2.1%	-1.8
(Auto) Regression	-10.9%	-4.7%	-3.3%	-1.6
Slow - Backpropagation	-0.5%	-4.2%	8.5%	0.2
Slow - Jordan	3.3%	-3.9%	7.0%	0.4
Fast - ARTMAP	-3.3%	0.6%	7.7%	0.3
Echo ARTMAP	15.7%	3.8%	7.6%	1.5

Table 1: The 10-year financial data set annualized gains per benchmark. The 10-year average for each benchmark is broken into three equal reporting periods for further granularity. Regression and auto-regression provided similar results and are combined for simplicity. All results include trading costs. The Sharpe Ratio is the average return divided by the standard deviation of returns. Typical mutual fund Sharpe Ratios range from -1.7 to 2.5 per www.morningstar.com.

Trading costs penalize each trade, which accounts for the random walk having a slightly negative annual rate. Without trading costs a random walk should consistently generate near zero average gains due to perfect hedging of buying and shorting.   In agreement with Yu et al. (2008), the regressive benchmarks both had more difficulty than neural networks due to the non-linear nature of financial time series data and the penalties incurred from the trading costs. Results are combined in Table 1 for simplicity since both benchmarks had similar performances. In agreement with Saad et al. (1998), the Slow-Backprop (backpropagation), Slow-Jordan, and Fast-ARTMAP networks all had similar risk-adjusted performances that outperformed the random walk. The networks can generate high gains, but the transaction costs and the volatility reduce much of the benefits. The prior studies reviewed did not typically include transaction costs or risk adjustment in their analyses.

The novel Echo ARTMAP network strongly outperformed all other benchmarks on the sample of five stocks, with a mean 10-year annual gain net of costs of 9% and a Sharpe Ratio of 1.5. This quantitatively shows that neural network topology can have significant empirical effects and that adding context can greatly improve fast learning neural network performance on financial decision making. To examine the effects of context sensitivity further, Figure 6 shows a detailed comparison between Fast-ARTMAP and Echo ARTMAP behaviors on an excerpt of trading data.

The Echo ARTMAP model can more accurately determine the underlying context and modulate its prediction behavior accordingly, which leads to better cumulative gains. To further quantify these effects this paper re-runs the simulation for Echo ARTMAP and Fast-ARTMAP all 30 current members of the Dow Jones Industrial Average over the ten years period 2000-2009. The Sharpe ratio for Fast-ARTMAP remained unchanged. The Echo ARTMAP average annual rate falls to 6%, but the Sharpe ratio increases to 2.1. This shows that the results from Table 1 are likely to be replicated in a larger portfolio of stocks.

To more closely examine the effects of combining context sensitivity with fast learning, this paper breaks the final discussion into three parts: the individual quantitative effects of delay, the effects of decay, and the distinct properties of output-to-input recurrence with fast learning.

Time delay relates to the size of the pattern, or how many periods are required to determine the trend or context. If the time delay is too short, it may not capture an existing pattern. If the delay is too long, it may capture non-pattern noise and over-fit. The Echo ARTMAP results in Table 1 were based on an arbitrary delay period of 10 trading days. Figure 7 shows Echo ARTMAP with varying delay periods, absent decay or output-to-input recurrence, averaged on all five stocks over ten years.

Figure 6: Detailed Comparison view of ARTMAP vs. Echo ARTMAP
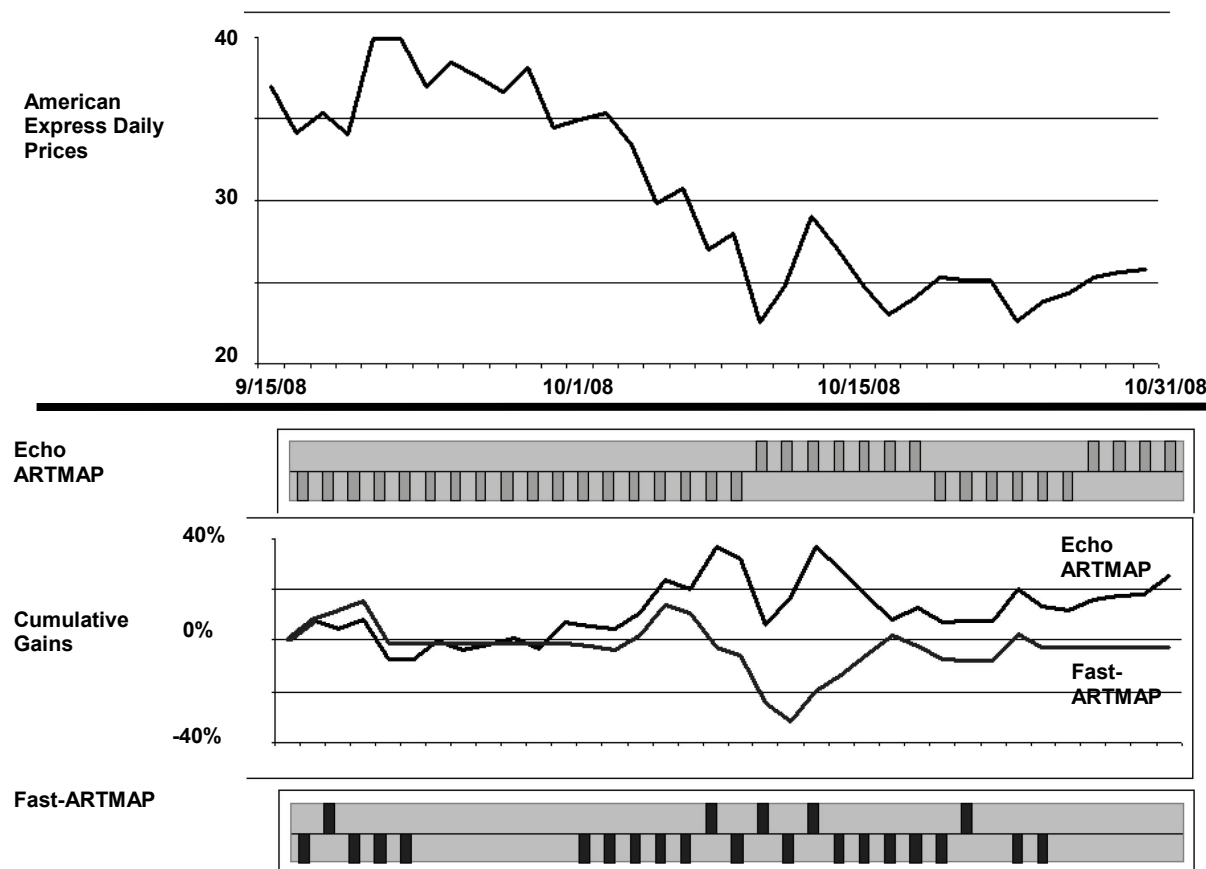
Figure 6: Excerpt of daily decisions to compare Fast-ARTMAP vs. Echo ARTMAP, which differs only in allowing the network to base decisions on past history of the same input features, as per Figure 4(d). (top) These are the daily prices for American Express, showing a rough downtrend with mini-uptrends in mid and late October. (bottom) The Echo ARTMAP decisions above, in blue, and the Fast-ARTMAP decisions below, in red, their cumulative respective gains from the decisions. A spike above the line indicates Buy, a spike below the line indicates a Sell, and no spike indicates No Trade. Note the more consistent trading decisions in Echo ARTMAP as it tracks downtrending and uptrending contexts, albeit imperfectly. The cumulative gains also quantitatively support that tracking context can improve prediction performance. Trading costs are included. In other time periods, Echo ARTMAP can predict No Trade.

The performance appears to show multiple peaks at the 12- and 24-period delays. This roughly coincides with prior empirical financial research favoring 12- and 26-period delays via moving averages (Appel, 1999). The peaks indicate that as the delay increases from zero to approximately 12, the period size better captures a small scale pattern. Further increases begin to capture non-pattern noise until a larger scale pattern manifests itself.

Time decay allows the network to capture multiple pattern scales simultaneously. Traders often need to consider both short term periods (e.g. 12-day periods) and longer term periods (e.g. 26-day periods) (Schwager, 1994; Appel, 1999). While the context may have changed on the short time scale, the context may remain the same on a longer time scale. To combine these two scales with differential influences, decay can reduce the influence over time. The Echo ARTMAP results in Table 1 are based on a fixed,

slow decay value of 0.9. Figure 8 shows Echo ARTMAP with varying decay rates, with fixed 10-day delay and absent output-to-input recurrence, averaged on all five stocks.

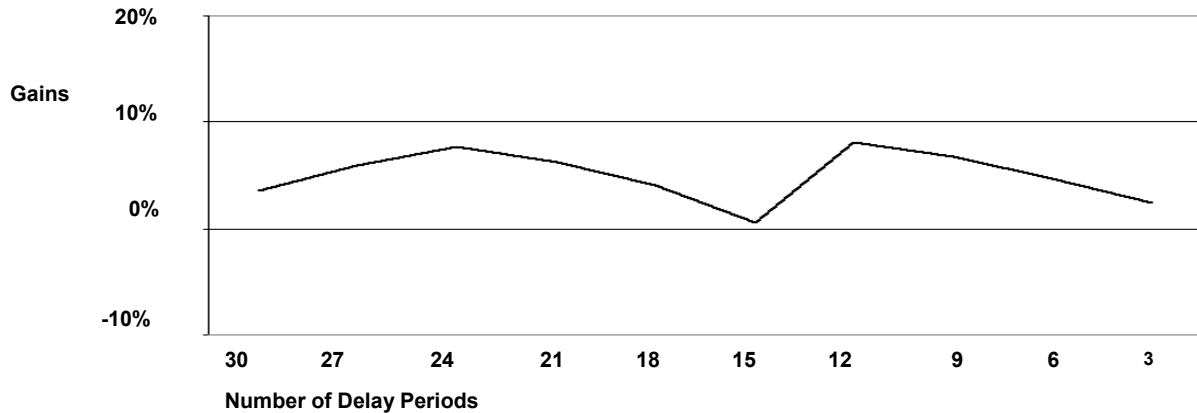Figure 7: The 10-year Echo ARTMAP Average Annual Gains by Period Delay

Figure 7: Echo ARTMAP cost-adjusted gains with varying delay parameters. There is neither feedback nor decay for this plot. The gains appear to show an inverse-U shape, with minor peaks. The smaller values on the right indicate smaller delay windows and are more suitable for data with short temporal patterns.

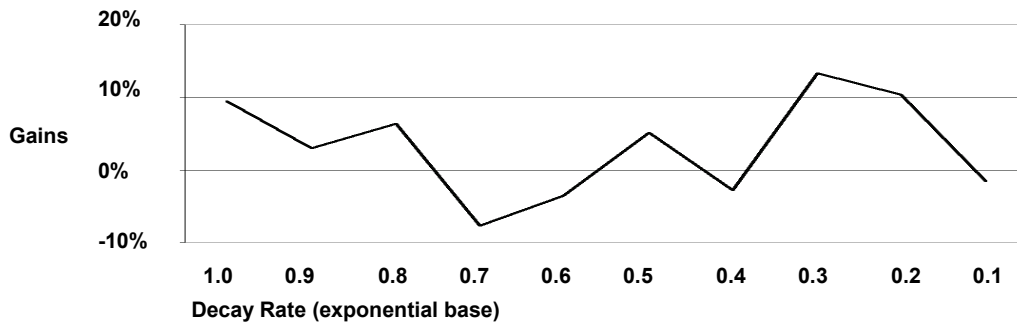Figure 8: The 10-Year Echo ARTMAP Average Annual Gains by Decay Rate

Figure 8: Echo ARTMAP cost-adjusted gains with varying decay rates. The delay period is fixed at 10 and there is no feedback. The gains appear highly variable. The smaller values on the right indicate faster decay and are more suitable for more heteroskedastic data.

The performance exhibits a highly variable and chaotic behavior over varying decay rates, with peaks near the non-decaying base of 1.0 and near a fast decay base of 0.3. The peaks indicate that combining different pattern scales for different stocks for different years may be a delicate procedure that is not very amenable to pre-selected, static values. If an automatic and adaptive method can successfully apply time decay rates, then the potential returns may be greatly improved.

Output-to-input feedback on a fast learning neural network allows automatic and dynamic adaptation to multiple pattern sizes without the need for pre-selected delay and decay values. The strong Echo ARTMAP performance with output-to-input feedback in Figure 6 shows it can capture some of this effect with delay and decay values of 10 and 0.9, respectively. To explore why this feedback has empirically not performed as well on a slow learning Jordan network per Table 1, this paper examines this issue with a seminar detection problem. Figure 9 shows the benchmark performances on correctly detecting seminars while minimizing the number of false detections. Detecting a seminar can be thought more generally as detecting an underlying context.

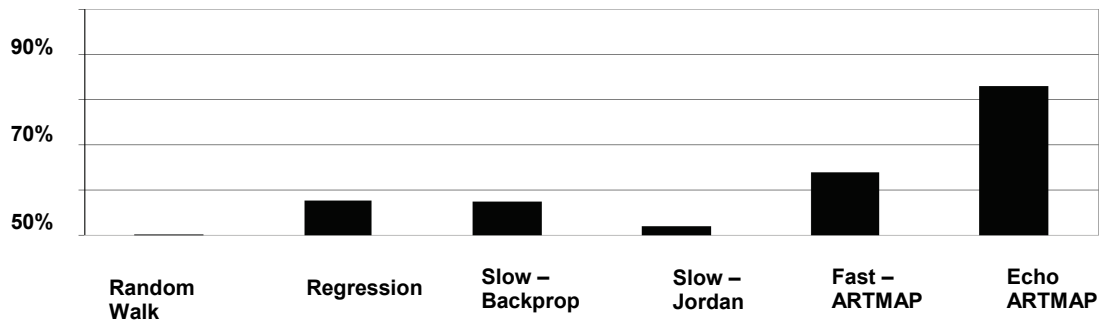Figure 9: The Event Detection Data Set Information Values by Benchmark

Figure 9: The six-month, two-class seminar detection data set as a function of area under the receiver operating characteristic curve. 50% indicates the benchmark is equivalent to constantly predicting one class or a random guess, as per random walk. Similar to the financial decision problem, Echo ARTMAP strongly outperforms all other benchmarks. The regressive models are combined for simplicity due to similar results.

The results from figure 9 again highlight the strong performance of Echo ARTMAP above all the other benchmarks, particularly that of the slow learning Jordan network. Figure 10 contrasts the Jordan network with the Echo ARTMAP.

Figure 10: Theoretical Differences between Fast Learning Echo ARTMAP and Slow Learning Jordan Model
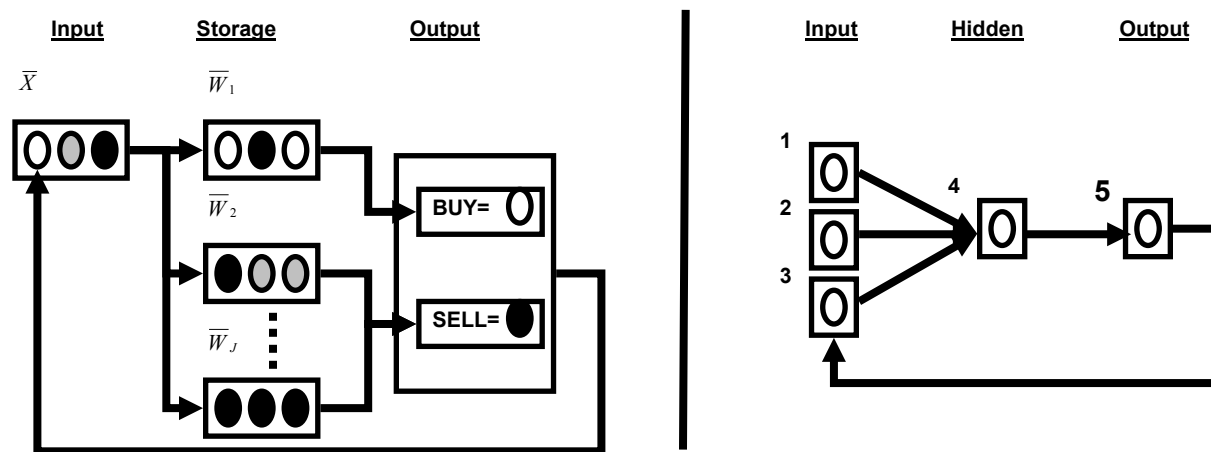

Figure 10: A comparison of output-to-input feedback network topologies. (left) The fast learning Echo ARTMAP, repeated from Figure 3 for convenience. (right) The equivalent slow learning Jordan network. Boxes represent nodes and circles represent pattern components.

While the two networks superficially appear similar, there are fundamental differences. Echo ARTMAP has a storage layer, each node of which contains a complete, separate pattern. The kernel matches each input to the most similar storage node. The storage node maps to a specific output node. Per equation (2), the $\overline{A}$ vector modifies the kernel. If a vector component approaches infinity, a storage node would be rejected if its related component differs even slightly from that of the input node. The storage nodes become more discerning.

In contrast, Jordan networks possess a hidden layer, each node of which contains one component of a pattern (for simplicity, only one hidden node is shown). Each input node also contains only one component. The input values multiply with the node connections to generate the hidden node value,

which multiplies with its node connection to generate the output node value. The output node value dictates the response.

If the absolute value of the product of the node connection multipliers from node 3 to 4 to 5 and back to 3 exceeds one, this introduces vulnerability to positive feedback and network saturation. As in the positive feedback loop with a speaker outputting to a microphone (Armstrong et al, 2009), output intensities continually increase towards infinity regardless of other microphone inputs, with the practical result being that the speaker produces its maximum output – a maximum volume screech. Similarly, in a Jordan network with a typical thresholded sigmoid transfer function for each node, each node transmits the product of its maximum thresholded value (e.g. typically one) and their node connection (e.g. up to infinity) to the next node.

When the output node does this, the classifications remain static (e.g. output of one) regardless of the other input values. The saturated Jordan network becomes biased towards the same class from prior time steps and cannot react quickly or at all to changes in the input. The only solution in this case is the use of notch filters, dampeners, and their biologically inspired network counterparts of negative feedback inhibition to prevent saturation (Haykin, 2001; Kandel et al, 2000). This has the effect of constraining the viable node connection multipliers to near zero. A zero value in the loop means there is no feedback. To demonstrate these differences in output-to-input recurrence with slow learning vs. fast learning, figure 11 shows a detailed view of the traffic pattern over two days, one of which contains an event.

Figure 11: Empirical Differences between Fast Learning Echo ARTMAP and Slow Learning Jordan Model
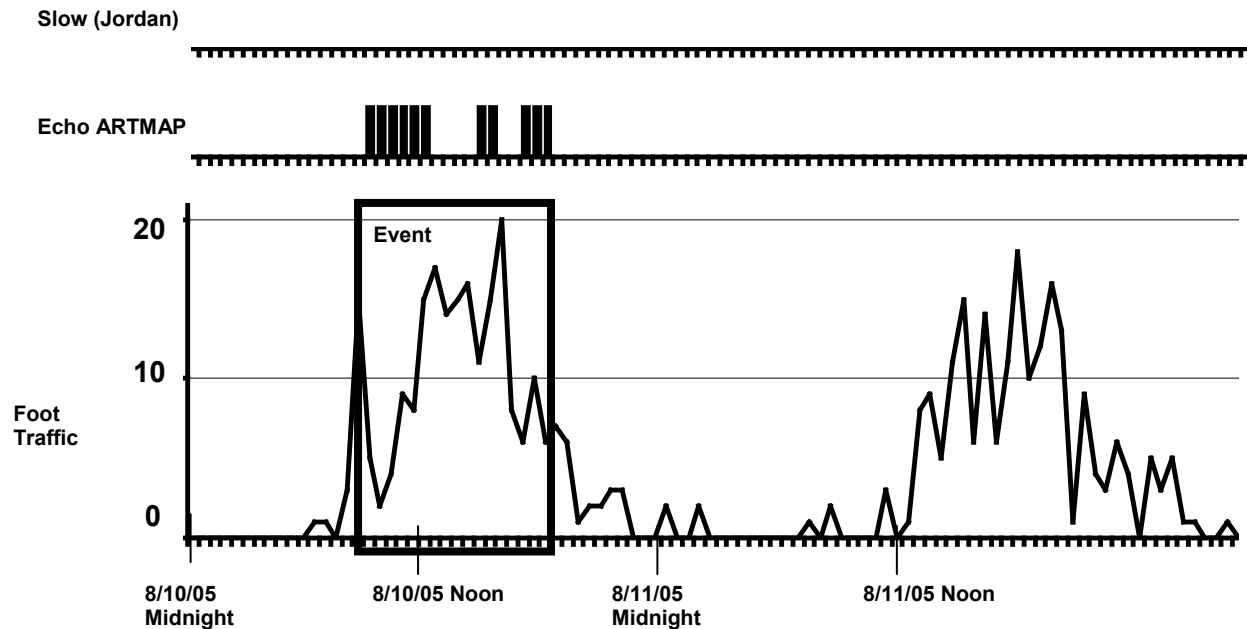

Figure 11: A two-day excerpt of the seminar detection problem. On 8/10/05 at 11:00 AM, there was a three-hour event, indicated by the box. The foot traffic exhibits a specific pattern that the benchmarks need to distinguish from the normal daily traffic, as shown on 8/11/05 in absence of an event. At top are the event predictions from the slow learning Jordan network and the fast learning Echo ARTMAP. A spike indicates the network predicts an event is in progress.

The Jordan network consistently predicts no event since the feedback continually biases the network towards prior periods with no events. Events are relatively uncommon. Echo ARTMAP, in comparison, can and does react rapidly by correctly indicating the presence of an event. The $\overline{A}$ vector values operate

on the kernel rather than on the input values directly. Echo ARTMAP has fewer constraints regarding positive feedback loops vs. slow learning networks and can therefore more fully explore optimal output-to-input feedback connections.

As a final note on context sensitive models, Figure 12 shows the benchmark performances on a context-free, purely spatial data set. Temporally context sensitive models should exhibit difficulties attempting to track non-existent temporal patterns in the circle-in-the-square problem.

Figure 12: The Non-Temporal Data Set Information Values by Benchmark
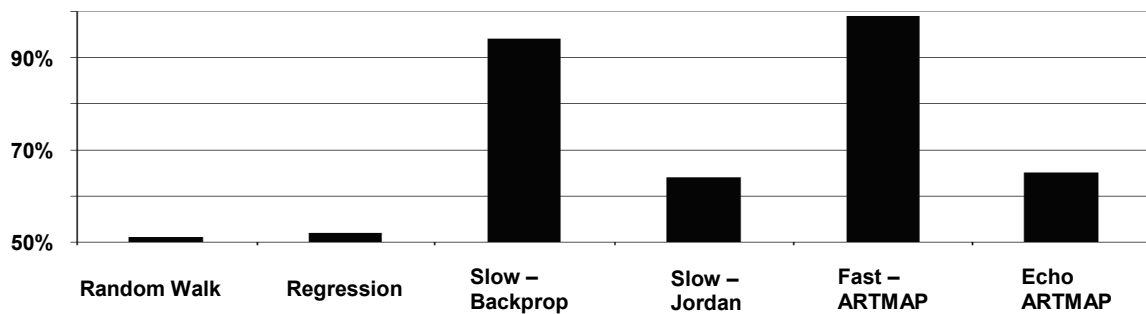

Figure 12: The benchmarks and their information values on the purely spatial circle-in-the-square data set. Regression and auto-regression showed only slight differences.

The Slow-Jordan and Echo ARTMAP perform poorly compared to their context blind counterparts (Slow-Backprop and Fast-ARTMAP, respectively). These context sensitive networks are unable to automatically adapt to the fact that each input in the data set is completely independent and there is no context. The network settings were identical as for the financial data set; that is, Echo ARTMAP was pre-selected with a delay period of 10, a decay base of 0.9, and with output-to-input feedback. Regression and auto-regression still perform poorly on this data set because the circle-in-the-square data set is non-linear.

CONCLUDING COMMENTS

In a financial time series, decision makers are best served by being cognizant of past and current indicators. This builds context into trading decisions. For automated systems like neural networks to emulate and assist in the decision-making process, they should be context sensitive. For neural networks to be adaptive and reactive to fluid changes in the environment, they should also rely on fast learning rules. The goal of this paper is to develop a novel fast learning context sensitive Echo ARTMAP neural model that quickly and transparently incorporates the current market conditions into its decisions.

To empirically test this novel model, this paper uses five randomly selected stocks from the Dow Jones Industrial Average over ten years of post-selection data. Trading costs are included into the risk-adjusted, annualized performance measures. For comparison, this paper applies six industry standard alternatives on the same data including random walk, regression, auto-regression, a slow learning backpropagation neural model, a slow learning context sensitive Jordan network model, and a fast learning ARTMAP.

Echo ARTMAP empirically outperformed all alternatives over the ten year study, under varying market conditions. While context-blind models cannot modulate their decisions based on extant environments and slow learning models react very slowly and poorly to ever-changing environments, the theory behind the enhancements in a fast learning, context sensitive model supports the Echo ARTMAP empirical findings. This supports the concept of working memory as a means of extracting the context that disambiguates feature inputs over time and leads to more intelligent decision-making.

More research is needed for exploring the effects of varying working memory spans. While this paper found periodicities corresponding to prior research on the effects of varying time delayed input data, it remains to be seen if this is a general finding across longer memory spans, different input features, and with different scales (e.g. hourly, real-time, weekly, etc.). There is also a general dearth of research into examining the effects of varying levels of time decay to measure how rapidly the information contained in a current data point loses value. Future work will focus on how feedback and neural model learning rules can dynamically adapt and adjust these contextual parameters to real-life data.

APPENDIX

Appendix I. Detailed Example of Echo ARTMAP Input Decay Scaling Using the same 1-dimensional example from Figure 2, Figure 13 details the effects of different *A* vectors.

The Echo ARTMAP kernel (equation (2)) follows the first three steps of equation (1), namely to normalize, complement code, and fuzzy min. Collecting terms for each component assigns individual similarity scores per dimension. Since this example uses one dimension, there is one similarity score.

The $\overline{A}.*\overline{X}'$ term applies the A vector to these dimensional similarity scores. Vector *A* values larger than 1 increase the influence of a dimension such that Echo ARTMAP becomes more discerning and only accepts very similar matches between the input and storage vectors. If $A = (2)$, for example, then $\overline{A}.*\overline{X}'$ $= (2)(0.8) = (1.6)$.

To complete the process, term $(1 - \overline{A}) = (1-2) = (-1)$. Adding the two terms together yields an Echo ARTMAP similarity of (0.6). This similarity is less than the original (0.8) from Figure 2, making the output associated with this storage node less likely to form the response. Closer matches between input and storage are required.

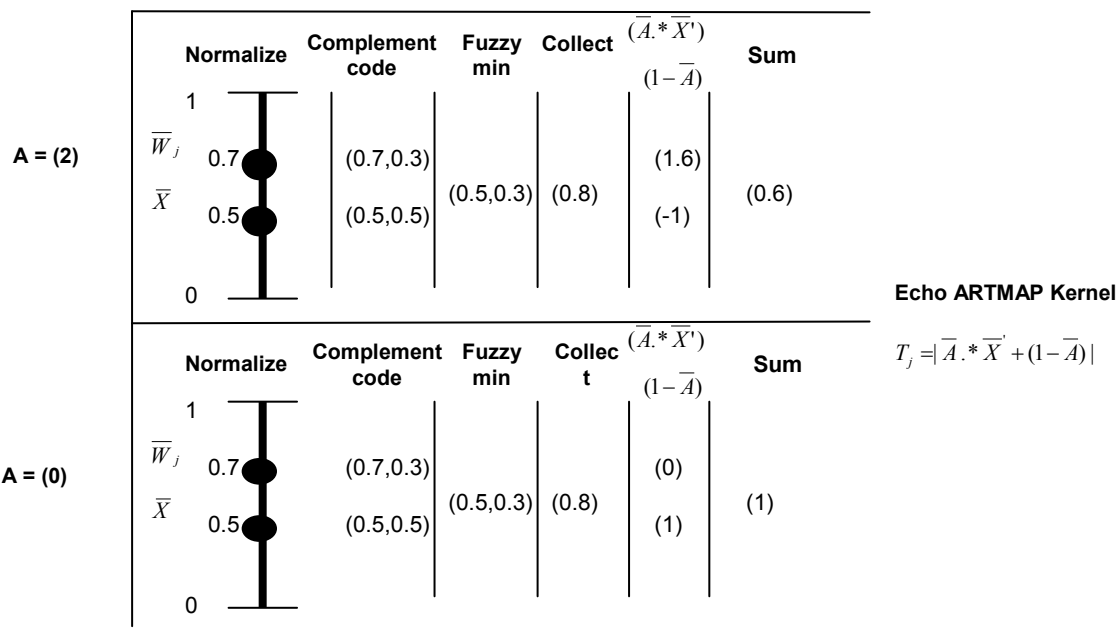Figure 13: Detailed Example of the Echo ARTMAP Kernel

Figure 13: The example from Figure 1 demonstrated with the Echo ARTMAP kernel using two different A vectors. See text for details.

Vice versa, *A* values less than 1 decrease the influence of a dimension such that Echo ARTMAP is less discerning and tends to accept any storage node. If $A = (0)$, then $\overline{A}^T \overline{X}' = (0)$ and $(1 - \overline{A}) = (1)$, which together sum to 1 regardless of input or storage values. In essence, this dimension has no effect and is ignored.

REFERENCES

Akman, V. (2002). "Context in Artificial Intelligence: A Fleeting Overview," *In: La Svolta Contestuale, C. Penco, ed.*, McGraw-Hill, Milano.

Amis, G. & Carpenter, G. (2007). "Default ARTMAP," *Proceedings of the International Joint Conference on Neural Networks (IJCNN,07) Orlando, Florida*, p. 777-782.

Appel, G. (1999). *Technical analysis power tools for active investors*, Financial Times Prentice Hall.

Armstrong, S., Sudduth, J., & McGinty, D. (2009). "Adaptive Feedback Cancellation: Digital Dynamo," *Advance for Audiologists,* vol. 11(4), p. 24.

Baddeley, A. D., & Logie, R. (1999). "Working Memory: The Multiple Component Model," in *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, Cambridge University Press, New York.

Bishop, C. (2006). *Pattern Recognition and Machine Learning,* Springer Science.

Bjorklund, D. & Rosenblum, K. (2002). "Context Effects in Children's Selection and Use of Simple Arithmetic Strategies, *Journal of Cognition & Development,* vol. 3, p. 225-242.

Box, G. & Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco.

Carpenter, G. (2003). "Default ARTMAP," *Proceedings of the International Joint Conference on Neural Networks, Portland, Oregon,* p. 1396-1401.

Carpenter, G., Grossberg, S., & Reynolds, J. (1991), "ARTMAP: Supervised Real-time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network," *Neural Networks*, vol. 4, p. 565-588.

Carraher, T., Carraher, D., & Schliemann, A. (1985). "Mathematics in the Streets and in Schools," *British Journal of Developmental Psychology,* vol.3, p. 21-29.

Chartered Financial Analyst Institute (2010). *CFA program curriculum*, CFA Institute, Pearson.

Chen, W. & Shih, J. (2006). "Comparison of Support-Vector Machines and Back Propagation Neural Networks in Forecasting the Six Major Asian Stock Markets," *International Journal of Electronic Finance,* vol. 1(1), p. 49-67.

Elman, J. (1990). "Finding Structure in Time," *Cognitive Science,* vol. 14, p. 179-211.

Freitas, F., Souza, A., & Almeida, A. (2009). "Prediction-Based Portfolio Optimization Model Using

Neural Networks," *Neurocomputing,* vol. 72(10), p. 2155-2170.

Gaganis, C., Pasiouras, F., & Doumpos, M. (2007). "Probabilistic Neural Networks for the Identification of Qualified Audit Opinions," *Expert Systems with Applications*, vol. 32, p. 114–124.

Hamilton, J. (1994). *Time Series Analysis*, Princeton University Press.

Haykin, S. (2001). *Adaptive Filter Theory*, Prentice Hall.

Higgins, J. (2004). *Introduction to Modern Nonparametric Statistics*, Brooks/Cole-Thomson Learning.

Jordan, M. (1986). "Serial Order: A Parallel Distributed Processing Approach," *Institute for Cognitive Science Report 8604*, University of California, San Diego.

Kandel, E., Schwartz, J., & Jessell, T. (2000). *Principles of neuroscience,* McGraw Hill.

Kane, M. & Engle, R. (2002). "The Role of Prefrontal Cortex in Working-Memory Capacity, Executive Attention, and General Fluid Intelligence: An Individual-Differences Perspective," *Psychonomic Bulletin & Review,* vol. 9(4), p. 637-671.

Ko, P. & Lin, P. (2008). "Resource Allocation Neural Network in Portfolio Selection," *Expert Systems with Applications,* vol. 35, p. 330–337.

Lo, A. (2001). "Bubble, Rubble, Finance in Trouble?", *Journal of Psychology and Financial Markets,* vol. 3, p. 76-86.

Lo, A. (2007). "The Efficient Markets Hypothesis," in *The Palgrave Dictionary of Economics, Palgrave Macmillan*

Lo, A. & Repin, D. (2002). "The Psychophysiology of Real-Time Financial Risk Processing," *Journal of Cognitive Neuroscience,* vol. 14(3), p. 323- 339.

Moody, J. & Darken, C. (1989). "Fast Learning in Networks of Locally Tuned Processing Units," *Neural Computation,* vol. 1, p. 281-294.

Morning Star (2010). *Retrieved from* http://www.morningstar.com

Parsons, O. & Carpenter, G. (2003). "ARTMAP Neural Networks for Information Fusion and Data Mining: Map Production and Target Recognition Methodologies," *Neural Networks*, vol. 16, p. 1075-1089.

Rumelhart, D., Hinton, G., & Williams, R. (1986). "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* vol. 1, MIT Press.

Saad, E., Prokhorov, E., & Wunsch, D. (1998). "Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks," *IEEE Transactions on Neural Networks,* vol. 9(6), p. 1456-1470.

Schwager, J. (1995). *The New Market Wizards: Conversations with America's Top Traders*, Wiley.

Sun, Y., Liang, Y., Zhang, W., Lee, H., Lin, W., & Cao, L. (2005). "Optimal Partition Algorithm of the RBF Neural Network and its Application to Financial Time Series Forecasting," *Neural Computing & Applications,* vol. 14, p. 36–44.

Versace, M., Bhatt, R., Hinds, O., & Schiffer, M. (2004). "Predicting the exchange traded fund DIA with a combination of Genetic Algorithms and Neural Networks," *Expert Systems with Applications,* Vol. 27(3), P. 417-425.

Witten, I. & Frank, E. (2002). *Data Mining*, Morgan Kaufman Publishers, San Francisco.

Wong, C., & Versace, M. (2011a). "Rethinking Neural Networks in Financial Decision-Making Studies: Seven Cardinal Confounds," *Global Conference on Business and Finance Proceedings, Las Vegas, Nevada.*

Yu, L., Wang, S., & Lai, K. (2008). "Neural Network-Based Mean–Variance–Skewness Model for Portfolio Selection," *Computers & Operations Research,* vol. 35, p. 34– 46.

Zhang, D., Jiang, Q., & Li, X. (2005). "A Heuristic Forecasting Model for Stock Decision Making," *Mathware & Soft Computing,* vol. 12, p. 33-39.

BIOGRAPHY

Charles Wong is a PhD candidate at the Cognitive and Neural Systems program, Boston University. He has previously worked for Deutsche Bank AG and KPMG LLP in New York. He can be contacted at ccwong@bu.edu.

Massimiliano Versace (PhD, Cognitive and Neural Systems, Boston University, 2007) is a Senior Research Scientist at the Department of Cognitive and Neural Systems at Boston University, Director of Neuromorphics Lab, and co-Director of Technology Outreach at the NSF Science of Learning Center CELEST: Center of Excellence for Learning in Education, Science, and Technology. He is a co-PI of the Boston University subcontract with Hewlett Packard in the DARPA Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SyNAPSE) project. He can be contacted at Versace@cns.bu.edu or at www.maxversace.com.