City Research Online

City, University of London Institutional Repository

Making Good on LSTMs' Unfulfilled Promise

Daniel Philps
Rothko Investment Strategies
Department of Computer Science
City, University of London
daniel.philps@city.ac.uk

Artur d'Avila Garcez
Department of Computer Science
City, University of London
a.garcez@city.ac.uk

Tillman Weyde
Department of Computer Science
City, University of London
t.e.weyde@city.ac.uk

Abstract

LSTMs promise much to financial time-series analysis, temporal and cross-sectional inference, but we find that they do not deliver in a real-world financial management task. We examine an alternative called Continual Learning (CL), a memory-augmented approach, which can provide transparent explanations, i.e. which memory did what and when. This work has implications for many financial applications including credit, time-varying fairness in decision making and more. We make three important new observations. Firstly, as well as being more explainable, time-series CL approaches outperform LSTMs as well as a simple sliding window learner using feed-forward neural networks (FFNN). Secondly, we show that CL based on a sliding window learner (FFNN) is more effective than CL based on a sequential learner (LSTM). Thirdly, we examine how real-world, time-series noise impacts several similarity approaches used in CL memory addressing. We provide these insights using an approach called Continual Learning Augmentation (CLA) tested on a complex real-world problem, emerging market equities investment decision making. CLA provides a test-bed as it can be based on different types of time-series learners, allowing testing of LSTM and FFNN learners side by side. CLA is also used to test several distance approaches used in a memory recall-gate: Euclidean distance (ED), dynamic time warping (DTW), auto-encoders (AE) and a novel hybrid approach, *warp-AE*. We find that ED under-performs DTW and AE but *warp-AE* shows the best overall performance in a real-world financial task.

Keywords: Continual learning, time-series, LSTM, similarity, DTW, auto-encoder

1 Introduction

Both LSTMs [20] and a wide range of time-series *sliding window* approaches suffer from a common problem: how to deal with long versus short term dependencies [30, 38, 13]. This problem is a manifestation of *catastrophic forgetting* (CF), one of the major impediments to the development of artificial general intelligence (AGI), where the ability of a learner to generalise to older tasks is corrupted by learning newer tasks [11, 37]. To address this problem, Continual Learning (CL) has been developed. Although very few time-series CL approaches exist, some have the advantage of having interpretable memory addressing [42], in contrast to LSTMs. The advantages of better interpretability are significant for real-world financial problems but there are a number of open questions for time-series CL. Firstly, should a financial time-series CL approach be based on recurrent

(e.g. LSTM) or a sliding window architecture (e.g. feed-forward neural net (FFNN))? What impact does time-series noise have on the memory functions of a CL approach? How would these choices translate to performance in a real-world financial problem when compared with LSTMs and a simple sliding window approach (FFNN)? This study empirically examines these questions in the context of the complex real-world problem of stock selection investment decisions in emerging market equities.

This paper is organised as follows. Section 2 introduces common design choices for time-series CL. Section 3 reviews related work while section 4 discusses *Continual Learning Augmentation* remember-gates, recall-gates and memory balancing. Section 5 describes the experimental setup, results and interpretability of the complex, real-world tests conducted, while section 6 concludes this paper.

2   Background

Machine learning based time-series approaches can be broadly separated into *sliding window*: dividing a time-series into a number of discrete modelling steps, e.g. FFNN; and *sequential*: attempting to model a time-series process as a sequence of values, e.g. LSTM.

A sliding window allows the choice of a wide range of learners, such as OLS regression [8] or feed forward neural networks (FFNN) applied in a step forwards fashion. Specialist models have also been developed, such as time delayed neural nets (TDNN) [54] but these are still constrained by choices relating to the time-delays to use. The major shortcoming of sliding-window approaches is that, as time-steps on, all information that moves out of the sliding window is typically forgotten.

Sequential approaches, while still requiring a sliding window for longer series, attempt to avoid the arbitrary, window sizing problem. However, this comes with a greater degree of complexity [59], as cross-sectional, temporal and short versus long-term dependencies need to be established sequentially. While there are many types of sequential learner [5], LSTMs are probably the most popular. While they are able to solve time-series problems that sliding window approaches cannot, sliding window approaches have outperformed LSTMs on seemingly more simple time-series problems [13]. Interpreting LSTMs is also challenging [18]. In either case, FFNNs and LSTMs applied to a time-evolving data-set, are exposed to CF [45]; occurring when a learner is applied in a $step\,forwards$ fashion. For example, a FFNN that is first trained to accurately approximate a model in a time-period A, and is then trained in time-period B, may see a deterioration in accuracy when applied to time-period A again. CL has been proposed to address CF, using implicit or explicit memory for the purpose. In many cases $similarity$ plays a part in driving this [33, 9, 47] but in the real-world of complex, noisy time-series similarity can be difficult to define. Temporal dependencies, changing modalities and other issues [46] all complicate gauging time-varying similarity and generally add computational expense. The impact of these temporal dynamics on a learner is sometimes called *concept drift* [46, 56].

This paper addresses whether a sequential or sliding window approach is best in a noisy real-world financial task and examines time-series similarity in a CL context. We test a sliding window approach (FFNN) and then a sequential learner (LSTM) finding that the FFNN applied as a sliding window is the best performer. Secondly, we test different time-series similarity approaches, which are used to drive CLA's memory recall-gate. We find simple Euclidean distance (ED) under-performs noise invariant similarity approaches; dynamic time warping (DTW) and auto-encoders (AE). We find the best performing similarity approach is a novel hybrid; *warp-AE*.

3   Related Work

While CF remains an open problem, many techniques have been developed to address it including gated neural networks [20], explicit memory structures [55], prototypical addressing [50], weight adaptation [19, 51], task rehearsal [49] and encoder based lifelong learning [52] to name a few. As researchers have addressed the initial challenges of CL, other problems have emerged, such as the overhead of external memory structures [43], problems with weight saturation [28], transfer learning [34] and the drawbacks of outright complexity [58]. While most CL approaches aim to learn sequentially, only a fraction of CL approaches have been focused on time-series [25, 16, 32, 10]. It is also unclear how effective these approaches would be when applied to open-world, state-based, temporal learning in long term, noisy, non-stationary time-series, particularly those commonly found

in finance. As most CL approaches have been developed for well-defined, generally labelled and typically stylized tasks [34] a strong motivation exists to develop time-series specific CL.

3.1 Remembering

Regime switching models [27] and change point detection [41] provide a simplified answer to identifying changing states in time-series with the major disadvantage that change points between regimes (or states) are notoriously difficult to identify out of sample [7] and existing econometric approaches are limited by long term, parametric assumptions in their attempts [6, 61, 48]. There is also no guarantee that a change point represents a significant change in the accuracy of an applied model, a more useful perspective for learning different states.

Residual change aims to observe change in the absolute error of a learner, aiming to capture as much information as possible regarding changes in the relation between independent and dependent variables. Different forms of residual change have been developed [2, 24, 23, 35, 22, 12]. However, most approaches assume a single or known number of change points in a series and are less applicable to a priori change points or multivariate series [57]. *Drift adaptation* approaches attempt to address some of these issues [12] but are of limited value to CL as they are focused on classification [21], tend to use simple and generally instance-based memory [56, 36, 29, 14], while tending to neglect recurring concepts [21] and to suffer from *gradual forgetting* [31]. This contrasts with explicit, task-oriented, memory structures of the sort used by CL to address CF. With the advent of time-series CL, residual change has another interesting application.

3.2 Recalling

CL approaches that use external memory structures require an appropriate addressing mechanism to store and recall memories. Memory addressing is generally based on a similarity measure such as cosine similarity [15, 17, 40] kernel weighting [53], use of linear models [50] or instance-based similarities, many using K-nearest neighbours [26, 51]. More recently, auto-encoders (AE) have been used to gauge similarity in the context of multitask learning (MTL) [1] and for memory consolidation [52]. However, these methods, as they have been applied, are not obviously well suited to assessing similarity in noisy multivariate time-series. In contrast, *data mining* researchers have extensively researched noise invariant time-series distance measures [3], generally for time-series classification (TSC). While simple Euclidean distance (ED) offers a rudimentary approach for comparing time-series it has a high sensitivity to the timing of data-points, something that has been addressed by dynamic time warping (DTW) [44]. However, DTW requires normalized data and is computationally expensive, although some mitigating measures have been developed [60]. A relatively small subset of data-mining research has used deep learning based approaches, such as convolutional neural nets (CNN) [62]. While results have been encouraging, interpretability is still an open question. Another interesting possibility is to use AEs to cope with noise by varying manifold dimensionality and by using simple activation functions to introduce sparsity (i.e. ReLU).

4 Continual Learning Augmentation

Continual learning augmentation (CLA) augments a conventional learner for time-series regression with memory. The aim is to allow well understood learners to be used in a CL framework in an interpretable way. CLA's memory functions are applied as a sliding window stepping forward through time, over input data of one or more time-series. The approach is initialized with an empty memory structure, M, and a chosen base learner, ϕ, parameterized by θ_B. This base learner can be a sequential approach or a sliding window approach applied to a multivariate input series, X, with K variables over T time-steps. The chosen base learner produces a forecast value $\hat{y}_{t+1}$ in each period as time steps forward. A *remember-gate*, j, appends a new memory, M_t^m, to M, on a *remember cue* defined by the change in the base learner's absolute error at time-point t. A *recall-gate*, g, balances a mixture of current and memory-based forecasts to result in the final outcome, $\hat{y_{t+1}}$. Figure 1 shows the functional steps of remembering, recalling and balancing learner-memories.
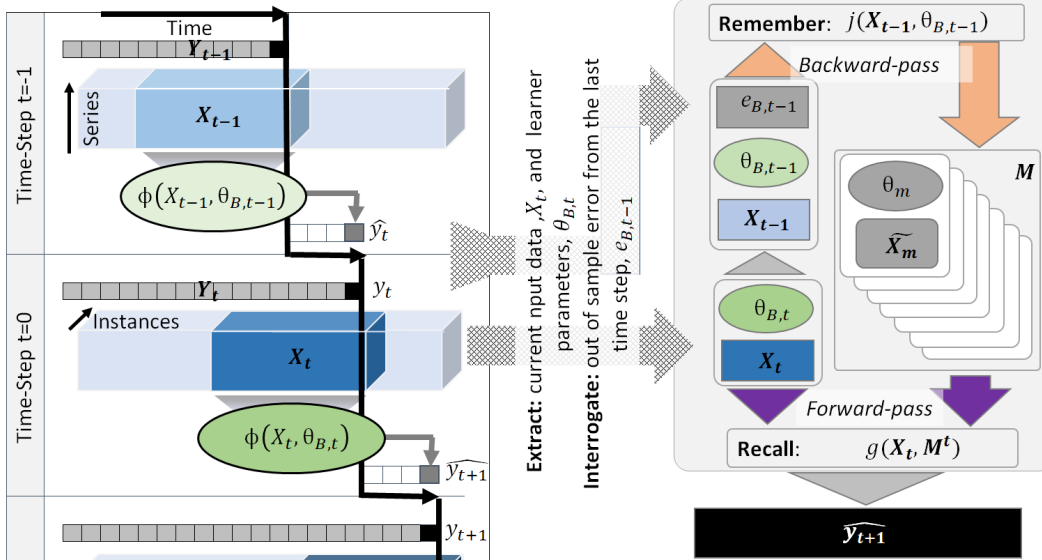
Figure 1: Continual learning augmentation architecture. Backward-pass: When y_t becomes observable, *remember-gate*, j, assesses $|\epsilon_{B,t-1}|$ for change and if change has occurred, extracts $\theta_{B,t-1}$ adding a memory column to M. Forward-pass: output of learner-memories and base learner are balanced by the *recall-gate*, g.

4.1 Memory management

Repeating patterns are required in sub-sequences of the input data to provide memory cues to remember and recall different past states. Learner parameters, θ_m, trained in a given past state can then be applied if that state approximately reoccurs in the future. When CLA forms a memory, it is stored as a column in an explicit memory structure, similar to [4], which changes in size over time as new memories are remembered and old ones forgotten. Each memory column consists of a copy of a past base learner parameterization, θ_m, and a representation, $\widetilde{X_m}$, of the training data used to learn those parameters. As the sliding window steps into a new time period, CLA recalls one or more learner-memories by comparing the latest input data, X_t, with a representation of the training data stored in each memory column, $\widetilde{X_m}$. Memories with training data that are more similar to the current input series will have a higher weight applied to their output, $\hat{y}_{m,t+1}$, and therefore make a greater contribution to the final CLA output, $\hat{y}_{t+1}$.

4.2 Remember-Gate

Remembering is triggered by changes in the absolute error series, ϵ_B, of the base learner as the approach steps forward through time:

$$\epsilon_B = \left\{ |(\hat{y}_t - y_t)|, \ldots, |(\hat{y}_T - y_T)| \right\} \tag{1}$$

CLA interrogates the base learner to determine changes in out-of-sample error, $\epsilon_{B,t-1}$, which are assumed to be associated with changes in state. The remember-gate, j, both learns to define and trigger a change which stores a pairing of the parameterization of the base learner, $\theta_{B,t-1}$, and a contextual reference, $\widetilde{X_{t-1}}$. Figure 1 shows how a change is detected by j, which then results in a new memory column being appended to M:

$$M = \left\{ (\widetilde{X_1}, \theta_1), \ldots, (\widetilde{X_M}, \theta_M) \right\} \tag{2}$$

Immediately after the new memory column has been stored a new base learner is trained on the current input, overwriting θ_B.

Theoretically, for a fair model of a state, ϵ_B would be approximately $i.i.d.$ with a zero valued mean. Therefore the current base model would cease to be a fair representation of the current state when

ϵ_B exceeds a certain confidence interval, in turn implying a change in state. J_{Crit} is a threshold representing a critical level for ϵ_B. When the observed absolute error series,ϵ_B, spikes above this critical level, it is interpreted as a change in state and also as a remember-cue.

J_{Crit} is estimated by the hyperparameter j_{Crit}. This is learned at every time-step to result in a level of sensitivity to remembering that forms an external memory, $\boldsymbol{M}$, resulting in the lowest empirical forecasting error for the CLA approach over the study term up until the current time, t:

$$J_{Crit} = \operatorname*{argmin}_{j_{Crit} \in jgrid} f(\boldsymbol{X_t}, j_{Crit}) \tag{3}$$

Where f is the CLA approach expressed as a function of the input series, $\boldsymbol{X_t}$, and j_{Crit}, yielding $\epsilon_{B,t}$ (the absolute error of the base learner at time t). $jgrid$ is an equidistant set, between the minimum and the maximum values of ϵ_B up until time t. This represents a discretization of the empirical distribution of ϵ_B from which to empirically solve for J_{Crit}. In our testing $jgrid$ was initialised to a value of 20, representing five-percent buckets in the empirical range of ϵ_B.

4.3 Recall-Gate

The recall of memories takes place in the recall-gate, g, which calculates $\hat{y}_{(t+1)}$, a mixture of the predictions from the current base learner and from learners stored in memory:

$$\hat{y}_{(t+1)} = g(\boldsymbol{X_t}, \boldsymbol{M^t}) \tag{4}$$

The mixture coefficients are derived by comparing the similarity of the current time varying context $\boldsymbol{X_t}$ with the contextual references, $\widetilde{\boldsymbol{X_m}}$, stored with each individual memory. Memories that are more similar to the current context have a greater weight in CLA's final outcome.

4.4 Recall-gate Similarity Choices

Several approaches for calculating contextual similarity are tested separately, using the CLA approach. Each is used to define $\widetilde{\boldsymbol{X_m}}$, either by simply storing past training examples or by using a process of contextual learning; essentially learning a representation of base learner training data.

Firstly, ED and then DTW are applied. Both approaches require $\widetilde{\boldsymbol{X_m}}$ to be raw training examples, stored in each respective memory column, making both approaches relatively resource hungry. Secondly, AE similarity is used through a process of contextual learning. Rather than needing to store many training examples, only the AE parameters are needed to form a reconstruction of the training data with the disadvantage that an AE must be trained in every time-step, reducing machine-memory usage versus DTW but increasing processing time. Thirdly we introduce a DTW filtered AE similarity, which is intended to phase adjust the AE reconstruction, we call this *warp-AE*. Again, an AE needs to be trained at every time-step but DTW processing expense is less because N in Equation 8 is equal to the number of securities, while in Equation 6, N is generally larger, to provide an effective sample over many security pairings. We describe each approach in turn.

ED and DTW are applied to N randomly sampled instances from $\widetilde{\boldsymbol{X_m}}$ and $\boldsymbol{X_t}$, sampling over rows (i.e. cross-sectional data), each of which represents a different security at a given point in time:

$$\hat{D}_{ED}(\widetilde{\boldsymbol{X_m}}, \boldsymbol{X_t}) = 1/N \sum_N ED(\widetilde{X}_{m,r_1(D)}, X_{t,r_2(D)}) \tag{5}$$

$$\hat{D}_{DTW}(\widetilde{\boldsymbol{X_m}}, \boldsymbol{X_t}) = 1/N \sum_N DTW(\widetilde{X}_{m,r_1(D)}, X_{t,r_2(D)}) \tag{6}$$

Where $\hat{D}$ is the dissimilarity, N is the number of samples to take and $r_1(D), r_2(D)$ are random integers between 1 and D.

AE similarity is used in a similar fashion to Aljundi et al 2017, [1], using ReLU activations to avoid over-fit. However CLA's use of AEs is specifcally designed for state-based CL in noisy, real-world, multivariate time-series. To facilitate this application CLA uses additional sparsity loss [39] and pure-

		a) CLA Result: Total Return					b) Augmentation Benefit: Relative to Base Learner				
		TR (% ann.)	SD (% ann.)	Sharpe	t-stat	p-val	RR (% ann.)	SD (% ann.)	Info Ratio	t-stat	p-val
		1	2	3	4		5	6	7		
CLA-LSTM	ED	-2.86%	-2.86% -	0.03 -	0.03	-2.86%	0.62%	4.37%	0.14	0.50	62.10%
	DTW	-1.75%	7.94% -	0.22 -	0.77	44.32%	4.07%	4.43%	0.92	3.20	0.17%
	AE	-0.55%	7.97% -	0.07 -	0.24	81.05%	4.65%	4.17%	1.11	3.89	0.02%
	w-AE	0.87%	8.33%	0.10	0.36	71.62%	5.87%	4.37%	1.34	4.69	0.00%
CLA-FFNN	ED	0.57%	6.54%	0.09	0.31	75.99%	0.75%	2.97%	0.25	0.89	37.76%
	DTW	2.34%	6.56%	0.36	1.24	21.53%	3.18%	4.11%	0.77	2.70	0.78%
	AE	3.59%	7.23%	0.50	1.73	8.58%	3.23%	5.38%	0.60	2.09	3.81%
	w-AE	2.07%	6.47%	0.32	1.11	26.71%	4.01%	4.98%	0.81	2.81	0.56%

Figure 2: Long-short investment simulation results (median over 5 simulation runs). Simulations are of long/short emerging market equities, between 2006-2017. No friction costs considered, as analysis of relative (not absolute) performance is the intention. 1) TR: annualized total return, 2) SD: annualized standard deviation, 3) Sharpe ratio: TR/SD, 4) p-value of the t-stat for the Sharpe ratio (see below), 5) RR: relative return of CLA versus the base learner performance from same simulation, 6) SD: annualized standard deviation of RR, 7) Information ratio: RR/SD. Results indicate CLA provides an augmentation benefit to FFNN and LSTM. This is concluded from the t-statistics of the Information Ratios, tested to determine a significant difference from zero.

linear decoder activations to reconstruct the approximate time-series and cross-sectional distributions. CLA then uses AE distances to balance memory-recall weightings:

$$\hat{D}_{AE}(\widetilde{\boldsymbol{X}_m}, \boldsymbol{X_t}) = 1/N \ ED(\boldsymbol{X}_t, a(h(\boldsymbol{X}_t))) \tag{7}$$

$ED(\boldsymbol{X}_t, a(h(\boldsymbol{X}_t)))$ is the Euclidean reconstruction loss of the current input, $\boldsymbol{X}_t$. a and h are the encoder and decoder functions respectively, N is the number of securities in $\boldsymbol{X_t}$.

warp-AE is designed to benefit from AE's lower memory usage while benefiting from the invariance to time-deformation provided by DTW:

$$\hat{D}_{wAE}(\widetilde{\boldsymbol{X}_m}, \boldsymbol{X_t}) = 1/N \ DTW(\boldsymbol{X}_t, a(h(\boldsymbol{X}_t))) \tag{8}$$

Each of these four (dis)similarity approaches can be used to determine which memories to recall from $\boldsymbol{M}$ and also how to weight the contribution of each to CLA's final outcome, $\hat{y}_{(t+1)}$. Each similarity approach was tested in turn, in CLA's memory recall-gate, aiming to gain new insights into time-series CL as it is applied to a complex real-world problem.

4.5 Balancing

The base learner and all recalled memories are weighted by similarity to produce CLA's final outcome, using the recall-gate, $g(\boldsymbol{X_t}, \boldsymbol{M})$:

$$\hat{y}_{t+1} = \sum_{m=1}^{M} \phi(\boldsymbol{X_t}, \theta_m) \cdot \left[1 - \frac{\hat{D}(\widetilde{\boldsymbol{X}_m}, \boldsymbol{X_t})}{\sum_{m=1}^{M} \hat{D}(\widetilde{\boldsymbol{X}_m}, \boldsymbol{X_t})}\right] \tag{9}$$

Where M is the number of memories in the CLA memory structure, $\boldsymbol{M}$. Previous research indicated balancing was the most powerful approach over selecting the *best* single memory [42]. (Notably, both these balancing approaches significantly outperform equal weighing of all memories, indicating CLA is gaining significantly more than a simple ensemble effect).

5 Experimental Results

5.1 Setup

CLA is used as a test bed for different learners and similarity approaches in a regression task to forecast future expected returns of individual equity securities. This is used to drive equities

investment simulations, a real-world task using noisy time-series. The data set consists of stock level characteristics at each time-step, for many stocks over many time-steps, as detailed below. Tests were conducted to show the relative performance of a sliding window base learner, FFNN, and a sequential base learner, LSTM. Different similarity approaches were also used to drive the memory recall-gate; ED, DTW, AE and *warp-AE*. All combinations of learner and similarity were tested.

Base learners were batch trained over all stocks at each time-step, forecasting US$ total returns 12months ahead for each stock. For the sliding window learner a year long, fixed length sliding window of four quarters was used for training and for the sequential learner all historic data up to the current time, was used for training. A stock level forecast in the top (bottom) decile of the stocks in a time-period was interpreted as a buy (sell) signal.

Although CLA is designed to use non-traditional driver variables, stock level characteristics are commonly expressed using *factor-loadings*. These were estimated, in-sample at each time-step by regressing style factor excess returns against each stock level US$ excess return stream over a 36month window: $r_{i,t} = \alpha_{i,t} + \beta_{MKT,i,t} x_{MKT,t} + \beta_{VAL,i,t} x_{VAL,i,t} + \epsilon_{i,t}$, where $r_{i,t}$ is the excess return of stock i in period t, $x_{MKT,t}$ is the excess return of the Emerging Market Equities Index, $x_{VAL,t}$ is the relative return of the Emerging Market Value Equities Index.

Stock level factor loadings populated a matrix, X, which comprised the input data. Each row represented a stock appearing in the index at time t (up to 5,500 stocks) and each column related to a coefficient calculated on a specific time lag. X resulted from a fifth and ninety-fifth percentile winsorizing of the raw input to eliminate outliers.

Long/short model portfolios were constructed (i.e. rebalanced) every six months over the study term, using equal weighted long positions (buys) and shorts (sells). The simulation encompassed 5,500 equities in total, covering 26 countries across emerging markets, corresponding to an Emerging Market Equities Index between 2006-2017. To account for the sampling approach used for ED and DTW similarities and differences in random initialisation of neural components, several simulations were carried out per test.

5.2 Simulation Results

CLA results showed a significant augmentation benefit for both base learners (see Figure 2 b), while all similarity approaches performed better than ED. Unaugmented base learners showed mixed results.

Sliding window learner tests, CLA-FFNN, outperformed all the equivalent sequential learner tests, CLA-LSTM, in terms of total return (TR) while Sharpe ratios (see Figure 2 a) were superior also (although no positive Sharpe ratio was significant at the 5% level). However, augmentation benefit, gauged by relative return (RR) and information ratio (Info Ratio), was superior for CLA-LSTMs (Figure 2 b), with most augmentation tests for both learners statistically significant at the 5% level. In these tests, although CLA-LSTM saw a better augmentation benefit (RR), CLA-FFNN saw the strongest outright performance (TR), followed by unaugmented FFNN (given by TR-RR), then CLA-LSTM. By far the weakest outright performer (TR) was unaugmented LSTM (given by TR-RR).

Tests of different similarity approaches, used in the recall-gate, saw ED under-perform DTW in TR terms and also in terms of augmentation benefit. This was true for both learners tested. This would imply that the invariance to time-deformation DTW provides, is an important consideration in a real-world context. AE similarity tests showed higher TRs than DTW and demonstrated statistically significant augmentation benefits at the 5% level for both learners, indicating that AE similarity is an appropriate approach to use in this context. *warp-AE* generated the highest RR and information ratios of all similarity tests, implying that adding a DTW filter to AE similarity was the most interesting similarity approach tested.

5.3 Interpretable Memory

CLA produces outcomes that can be attributed to specific memories. Figure 3 shows an example of one of the simulation runs, CLA-FFNN with AE similarity. The upper panel shows hypothetical investment returns of the unaugmented base learner (line chart) compared to those of the CLA augmented base learner (area). The lower panel shows CLA's expanding-memory-triangle, expanding by one row at each time-step forwards, indicating the possibility of the system adding a new memory at each step. Note the black horizontal lines, showing when certain memories were recalled and applied.
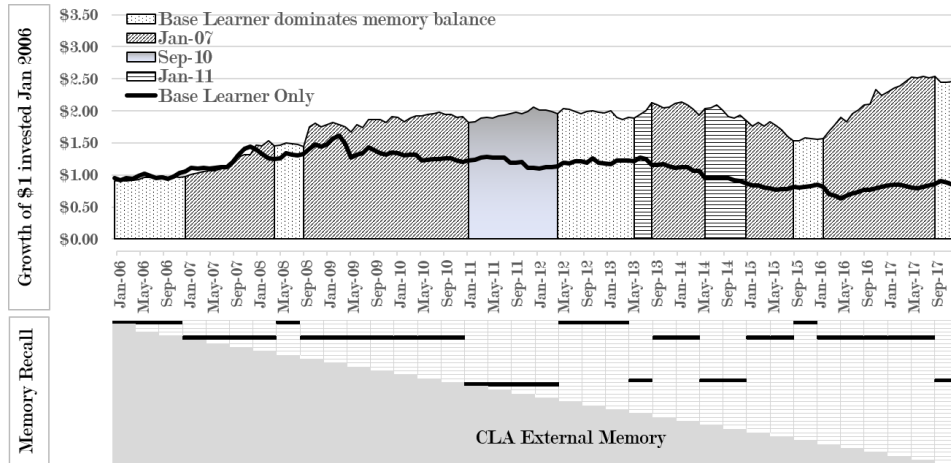
Interpretable Memory

Figure 3: Interpretable memory. Top: Investment returns of an unaugmented FFNN base learner (line) shown next to CLA-FFNN returns (area), from the same CL instance, separated by memory contribution. Lower chart: representation of the CLA memory structure, where each row in the expanding memory-triangle represents a step-forward and therefore a potential remember-cue. Highest weighted recalled memories are show as horizontal bold lines.

In this example, at least three memories are remembered and recalled at later times. Qualitatively, a memory remembered in January 2007, a period of turbulence in financial markets, adds materially to the strategy return in later periods, when recalled. It proves more appropriate than the base learner in the period of the 2008 financial crisis and its aftermath involving concerted fiscal stimulus (Sept 2008-Dec 2010). It was again recalled in 2013 and in 2015/2016, periods where fiscal stimulus dominated market returns (in Europe and China respectively). The sparsity of the memory structure is also notable, with only three principle memories recalled/used. As a rule of thumb, fewer memories will be remembered/recalled for noisier data-sets. This occurs through either a lower learned sensitivity to remembering, from a higher learned value of j_{Crit}, or through less recalling, from less discernible contexts.

6 Conclusion

We have empirically demonstrated that when applied to a real-world financial task involving noisy time-series with a large cross-sectonal component, a CL augmented sliding window learner (CLA-FFNN) is superior to both an LSTM learner and to a CL augmented LSTM learner (CLA-LSTM).

Testing of different similarity approaches, applied to a recall-gate, showed poor performance of simple Euclidean distance (ED) when compared to dynamic time warping (DTW). This strongly implies that the timing of data-points is important in this task and likely in other real-world problems involving noisy time-series. Simulation tests also showed that AE similarity is a good alternative to DTW. *warp-AE* was proposed to benefit from both DTW's time-deformation invariance and AE's lower machine-memory requirements, an approach that produced the strongest augmentation benefit. We also show that time-series CL not only outperforms LSTM and FFNN base learners but can provide a transparent explanation for which memory did what and when.

In summary, the most successful CL augmentation choice was found to be a sliding window CLA-FFNN learner combined with a recall-gate using *warp-AE* similarity. These tests also affirm Continual Learning Augmentation (CLA) as a real-world time-series CL approach, with the flexibility to augment different types of learners.

7 Bibliography

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. "Expert Gate: Lifelong Learning with a Network of Experts". In: *CoRR* abs/1611.06194 (2017). arXiv: 1611.06194. URL: http://arxiv.org/abs/1611.06194.

[2] Evans J Brown R Durbin J. "Techniques for testing the constancy of regression relationships over time". In: *Journal of the Royal Statistical Society Series B (methodological)* 37.2 (1975), pp. 149–192.

[3] Sung-Hyuk Cha. "Comprehensive Survey on Distance/Similarity Measures Between Probability Density Functions". In: *Int. J. Math. Model. Meth. Appl. Sci.* 1 (Jan. 2007).

[4] Dan C. Ciresan, Ueli Meier, and Jurgen Schmidhuber. "Multi-column Deep Neural Networks for Image Classification". In: *CoRR* abs/1202.2745 (2012). arXiv: 1202.2745. URL: http://arxiv.org/abs/1202.2745.

[5] Thomas G. Dietterich. "Machine Learning for Sequential Data: A Review". In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ed. by Terry Caelli et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 15–30. ISBN: 978-3-540-70659-5.

[6] Smith A Engle R. "Stochastic Permanent Breaks". In: *The Review of Economics and Statistics* 81.4 (1999), pp. 553–574.

[7] F.J. Fabozzi, S.M. Focardi, and P.N. Kolm. *Quantitative Equity Investing: Techniques and Strategies*. Frank J. Fabozzi Series. Wiley, 2010. ISBN: 9780470262474. URL: https://books.google.se/books?id=l2LCDAEACAAJ.

[8] Eugene F. Fama and Kenneth R. French. "Common Risk Factors in the Returns On Stocks And Bonds". In: *Journal of Financial Economics* 33 (1993), pp. 3–56.

[9] Geli Fei, Shuai Wang, and Bing Liu. "Learning Cumulatively to Become More Knowledgeable". In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, 2016, pp. 1565–1574. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939835. URL: http://doi.acm.org/10.1145/2939672.2939835.

[10] Thomas Fischer and Christopher Krauss. *Deep learning with long short-term memory networks for financial market predictions*. FAU Discussion Papers in Economics 11/2017. Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics, 2017. URL: https://ideas.repec.org/p/zbw/iwqwdp/112017.html.

[11] French. "Catastrophic forgetting in connectionist networks." In: *Trends in cognitive sciences* 3 4 (1999), pp. 128–135.

[12] João Gama et al. "A Survey on Concept Drift Adaptation". In: *ACM Comput. Surv.* 46.4 (Mar. 2014), 44:1–44:37. ISSN: 0360-0300. DOI: 10.1145/2523813. URL: http://doi.acm.org/10.1145/2523813.

[13] Felix A. Gers, Douglas Eck, and Jürgen Schmidhuber. "Applying LSTM to Time Series Predictable through Time-Window Approaches". In: *Artificial Neural Networks — ICANN 2001*. Ed. by Georg Dorffner, Horst Bischof, and Kurt Hornik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 669–676. ISBN: 978-3-540-44668-2.

[14] João Bártolo Gomes, Ernestina Menasalvas, and Pedro A. C. Sousa. "CALDS: Context-aware Learning from Data Streams". In: *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques*. StreamKDD '10. Washington, D.C.: ACM, 2010, pp. 16–24. ISBN: 978-1-4503-0226-5. DOI: 10.1145/1833280.1833283. URL: http://doi.acm.org/10.1145/1833280.1833283.

[15] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *CoRR* abs/1410.5401 (2014). arXiv: 1410.5401. URL: http://arxiv.org/abs/1410.5401.

[16] Alex Graves et al. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, 2006, pp. 369–376. ISBN: 1-59593-383-2. DOI: 10.1145/1143844.1143891. URL: http://doi.acm.org/10.1145/1143844.1143891.

[17] Alex Graves et al. "Hybrid computing using a neural network with dynamic external memory". In: *Nature* 538.7626 (Oct. 2016), pp. 471–476. ISSN: 00280836. URL: http://dx.doi.org/10.1038/nature20101.

[18] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. "Exploring Interpretable LSTM Neural Networks over Multi-Variable Data". In: *CoRR* abs/1905.12034 (2019). arXiv: 1905.12034. URL: http://arxiv.org/abs/1905.12034.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. "Distilling the Knowledge in a Neural Network". In: *NIPS Deep Learning and Representation Learning Workshop*. 2015. URL: http://arxiv.org/abs/1503.02531.

[20] Schmidhuber J Hochreiter S. "Long short term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[21] A. S. Iwashita and J. P. Papa. "An Overview on Concept Drift Learning". In: *IEEE Access* 7 (2019), pp. 1532–1547. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2886026.

[22] Bai J. "On the partial sums of residuals in autoregressive and moving average models". In: *Journal of Timeseries Analysis* 14.3 (1991).

[23] V. K. Jandhyala and I. B. MacNeill. "Residual partial sum limit process for regression models with applications to detecting parameter changes at unknown times". In: *Stochastic Processes and their Applications* 33.2 (1989), pp. 309–323. URL: https://EconPapers.repec.org/RePEc:eee:spapps:v:33:y:1989:i:2:p:309-323.

[24] MacNeill I Jandhyala V. "The change point problem: a review of applications". In: *Developments in water science* 27 (1986), pp. 381–387.

[25] M.W. Kadous, Mohammed Waleed Kadous, and Supervisor Claude Sammut. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. Tech. rep. Unknown, 2002.

[26] Lukasz Kaiser et al. "Learning to Remember Rare Events". In: *CoRR* abs/1703.03129 (2017). arXiv: 1703.03129. URL: http://arxiv.org/abs/1703.03129.

[27] Chang-Jin Kim and Charles Nelson. *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*. 1st ed. Vol. 1. The MIT Press, 1999. URL: https://EconPapers.repec.org/RePEc:mtp:titles:0262112388.

[28] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *CoRR* abs/1612.00796 (2016). arXiv: 1612.00796. URL: http://arxiv.org/abs/1612.00796.

[29] Ralf Klinkenberg. "Learning drifting concepts: Example selection vs. example weighting". In: *Intell. Data Anal.* 8 (Aug. 2004), pp. 281–300. DOI: 10.3233/IDA-2004-8305.

[30] Jan Koutník et al. "A Clockwork RNN". In: *CoRR* abs/1402.3511 (2014). arXiv: 1402.3511. URL: http://arxiv.org/abs/1402.3511.

[31] Ivan Koychev. "Gradual Forgetting for Adaptation to Concept Drift". In: *In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*. 2000, pp. 101–106.

[32] Zachary Chase Lipton et al. "Learning to Diagnose with LSTM Recurrent Neural Networks". In: *CoRR* abs/1511.03677 (2015). arXiv: 1511.03677. URL: http://arxiv.org/abs/1511.03677.

[33] Xiaodong Liu et al. "Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 912–921. DOI: 10.3115/v1/N15-1092. URL: https://www.aclweb.org/anthology/N15-1092.

[34] David Lopez-Paz and Marc'Aurelio Ranzato. "Gradient Episodic Memory for Continuum Learning". In: *CoRR* abs/1706.08840 (2017). arXiv: 1706.08840. URL: http://arxiv.org/abs/1706.08840.

[35] Ian B. MacNeilt. "Detecting unknown interventions with application to forecasting hydrological data". In: *JAWRA Journal of the American Water Resources Association* 21.5 (1985), pp. 785–796. DOI: 10.1111/j.1752-1688.1985.tb00172.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1752-1688.1985.tb00172.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1752-1688.1985.tb00172.x.

[36] Marcus A. Maloof and Ryszard S. Michalski. "Selecting Examples for Partial Memory Learning". In: *Machine Learning* 41.1 (Oct. 2000), pp. 27–52. ISSN: 1573-0565. DOI: 10.1023/A:1007661119649. URL: https://doi.org/10.1023/A:1007661119649.

[37] Michael McCloskey and Neal J. Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". In: *Psychology of Learning and Motivation*. Ed. by Gordon H. Bower. Vol. 24. Psychology of Learning and Motivation. Academic Press, 1989, pp. 109 –165. DOI: `https://doi.org/10.1016/S0079-7421(08)60536-8`. URL: `http://www.sciencedirect.com/science/article/pii/S0079742108605368`.

[38] Michael C. Mozer. "Induction of Multiscale Temporal Structure". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS'91. Denver, Colorado: Morgan Kaufmann Publishers Inc., 1991, pp. 275–282. ISBN: 1-55860-222-4. URL: `http://dl.acm.org/citation.cfm?id=2986916.2986950`.

[39] Bruno A. Olshausen and David J. Field. "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision Research* 37.23 (1997), pp. 3311 –3325. ISSN: 0042-6989. DOI: `https://doi.org/10.1016/S0042-6989(97)00169-7`. URL: `http://www.sciencedirect.com/science/article/pii/S0042698997001697`.

[40] Seongsik Park et al. "Quantized Memory-Augmented Neural Networks". In: *CoRR* abs/1711.03712 (2017). arXiv: 1711.03712. URL: `http://arxiv.org/abs/1711.03712`.

[41] A. N. Pettitt. "A Non-Parametric Approach to the Change-Point Problem". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28.2 (1979), pp. 126–135. DOI: `10.2307/2346729`. eprint: `https://rss.onlinelibrary.wiley.com/doi/pdf/10.2307/2346729`. URL: `https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2346729`.

[42] Daniel Philps et al. "Continual Learning Augmented Investment Decisions". In: *CoRR* abs/1812.02340 (2018). arXiv: 1812.02340. URL: `http://arxiv.org/abs/1812.02340`.

[43] Jack W. Rae et al. "Scaling Memory-Augmented Neural Networks with Sparse Reads and Writes". In: *CoRR* abs/1610.09027 (2016). arXiv: 1610.09027. URL: `http://arxiv.org/abs/1610.09027`.

[44] H Sakoe and S Chiba. "Dynamic-programming algorithm optimization for spoken word recognition". English. In: *IEEE TRANSACTIONS ON ACOUSTICS SPEECH AND SIGNAL PROCESSING* 26.1 (1978), pp. 43–49.

[45] Monika Schak and Alexander Gepperth. "A study on catastrophic forgetting in deep LSTM networks". In: *Working Paper, University of Applied Sciences Fulda, 36037 Fulda, Germany*. 2019.

[46] Jeffrey C. Schlimmer and Richard H. Granger. "Incremental learning from noisy data". In: *Machine Learning* 1.3 (Sept. 1986), pp. 317–354. ISSN: 1573-0565. DOI: `10.1007/BF00116895`. URL: `https://doi.org/10.1007/BF00116895`.

[47] Lei Shu, Hu Xu, and Bing Liu. "Unseen Class Discovery in Open-world Classification". In: *CoRR* abs/1801.05609 (2018). arXiv: 1801.05609. URL: `http://arxiv.org/abs/1801.05609`.

[48] D Siegmund. "Change-points: from sequential detection to biology and back". In: *Sequential Analysis* 32.2-14 (2013), pp. 43–46.

[49] Daniel L. Silver and Robert E. Mercer. "The Task Rehearsal Method of Life-Long Learning: Overcoming Impoverished Data". In: *Advances in Artificial Intelligence*. Ed. by Robin Cohen and Bruce Spencer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 90–101. ISBN: 978-3-540-47922-2.

[50] Jake Snell, Kevin Swersky, and Richard S. Zemel. "Prototypical Networks for Few-shot Learning". In: *CoRR* abs/1703.05175 (2017). arXiv: 1703.05175. URL: `http://arxiv.org/abs/1703.05175`.

[51] Pablo Sprechmann et al. "Memory-based Parameter Adaptation". In: *CoRR* abs/1802.10542 (2018). arXiv: 1802.10542. URL: `http://arxiv.org/abs/1802.10542`.

[52] Amal Rannen Triki et al. "Encoder Based Lifelong Learning". In: *CoRR* abs/1704.01920 (2017). arXiv: 1704.01920. URL: `http://arxiv.org/abs/1704.01920`.

[53] Oriol Vinyals et al. "Matching Networks for One Shot Learning". In: *CoRR* abs/1606.04080 (2016). arXiv: 1606.04080. URL: `http://arxiv.org/abs/1606.04080`.

[54] A. Waibel et al. "Phoneme recognition using time-delay neural networks". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3 (Mar. 1989), pp. 328–339. DOI: `10.1109/29.21701`.

[55]   Jason Weston, Sumit Chopra, and Antoine Bordes. "Memory Networks". In: *CoRR* abs/1410.3916 (2014). arXiv: 1410.3916. URL: http://arxiv.org/abs/1410.3916.

[56]   Gerhard Widmer and Miroslav Kubat. "Learning in the presence of concept drift and hidden contexts". In: *Machine Learning* 23.1 (Apr. 1996), pp. 69–101. ISSN: 1573-0565. DOI: 10.1007/BF00116900. URL: https://doi.org/10.1007/BF00116900.

[57]   H Yu. "High moment partial sum processes of residuals in ARMA models and their applications". In: *Journal of time series analysis* 28.1 (2007), pp. 72–91.

[58]   Wojciech Zaremba and Ilya Sutskever. "Reinforcement Learning Neural Turing Machines". In: *CoRR* abs/1505.00521 (2015). arXiv: 1505.00521. URL: http://arxiv.org/abs/1505.00521.

[59]   Saizheng Zhang et al. "Architectural Complexity Measures of Recurrent Neural Networks". In: *CoRR* abs/1602.08210 (2016). arXiv: 1602.08210. URL: http://arxiv.org/abs/1602.08210.

[60]   Zheng Zhang et al. "Dynamic Time Warping Under Limited Warping Path Length". In: *Inf. Sci.* 393.C (July 2017), pp. 91–107. ISSN: 0020-0255. DOI: 10.1016/j.ins.2017.02.018. URL: https://doi.org/10.1016/j.ins.2017.02.018.

[61]   Ji Hanlee Zhang N Siegmund D and Li J. "Detecting simultaneous change-points in multiple sequences". In: *Biometrika* 97 (2010), pp. 631–646.

[62]   Yi Zheng et al. "Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks". In: *Web-Age Information Management*. Ed. by Feifei Li et al. Cham: Springer International Publishing, 2014, pp. 298–310. ISBN: 978-3-319-08010-9.