Incremental Stock Volume Prediction with Gradient Distillation and Diversified Memory Selection

Shicheng Li[1] , **Zhiyuan Zhang**[1] , **Lei Li**[1] , **Ruihan Bao**[2*] , **Keiko Harimoto**[2] and **Xu Sun**[1]
[1]MOE Key Lab of Computational Linguistics, School of Computer Science, Peking University
[2]Mizuho Securities Co., Ltd.
{lisc99, zzy1210}@pku.edu.cn, lilei@stu.pku.edu.cn, {ruihan.bao, keiko.harimoto}@mizuho-sc.com, xusun@pku.edu.cn

Abstract

Stock volume forecasting is a typical time series regression task, which aims to predict the trading volume according to historical transaction data. In this paper, we explore an incremental learning scenario of volume prediction, which is a more practical setting as new data comes in over time. Traditional incremental framework based on memory prediction consistency is primarily targeted at classification tasks and neglects the characteristics of regression problems, resulting in poor knowledge transfer efficiency of the memorized samples in incremental volume prediction. To remedy this problem, we incorporate a gradient distillation term during the model update stage to fully exploit the information contained in the memory. We also propose a diversified memory construction method during the memory update stage to further improve memory utilization. Experiments on real-world stock data and further analyses demonstrate the superiority of our proposed method to existing incremental learning approaches.

1 Introduction

As a classical time series regression task, stock volume prediction aims at forecasting the trading volume of the next time period given historical transaction data, which plays a significant role in many financial applications including quantitative trading and market regulation. Traditionally, historical averages are often used to estimate future volumes. With the prevalence of deep learning, neural networks have been increasingly applied to the task of volume prediction and prove a great success in modeling the complex factors behind volume trends [Libman *et al.*, 2019; Zhao *et al.*, 2021].

One desideratum for volume prediction models is the ability to perform incremental updates as new data constantly come in. In the stock market, volume trends may move gradually under economic situations or undergo drastic changes due to sudden events. New patterns may emerge from each day's incoming data, while old patterns may also recur at some future point, making it a necessity to retain the knowledge about previous experiences. In this work, we focus on such an incremental setting where the model is trained sequentially on a data stream and required to predict future volumes during the progress, which is a more practical and challenging setting compared to the conventional single-task scenario.

Traditional neural networks trained with gradient-based optimization are known to suffer from a problem called *Catastrophic Forgetting* [McCloskey and Cohen, 1989; Ratcliff, 1990], the phenomenon that the performance on previous tasks will decrease dramatically after the model is trained on a new task or data distribution. This makes it a challenge to incrementally update the volume prediction model to adapt to the massive influx of transaction data in such a non-stationary environment as the stock market. To alleviate this problem, a common approach [Rebuffi *et al.*, 2017; Chaudhry *et al.*, 2019; Buzzega *et al.*, 2020] is to maintain a memory of previously seen samples and encourage the prediction consistency on these stored samples between the old model and the new one. However, these methods primarily focus on classification tasks and adopt a random memory selection method, which exhibit a performance gap due to the under-utilization of the memory when adapted to the incremental volume prediction task.

Our work tackles the limitation of previous studies by improving memory utilization from the following two aspects.

First, we encourage the consistency of input gradients between the old model and the new one to improve the efficiency of knowledge transfer. Previous memory-based incremental learning methods on classification use the probabilities predicted by a model to facilitate knowledge transfer which convey its knowledge of the relationship between classes. As this information is unavailable in the context of regression, we propose to use the model's gradient with respect to the input as a surrogate for the model's knowledge since it describes how the model's prediction changes within the neighborhood of the input. Based on this notion, we encourage consistency between the input gradients by incorporating a gradient distillation term into the training objective. Compared to naive experience replay, our gradient distillation term is more informative about the decision process of previous models and therefore provides more resistance against catastrophic forgetting.

*Contact Author

Second, we propose to update the memory with samples that are representative of the model's knowledge. Considering the large data size and relatively low signal-to-noise ratio, we design a memory selection method that maintains a diversified collection of data samples by comprehensively considering a sample's market information and the knowledge used by the model in its prediction. To prevent the memory from stagnating, a simple approach is introduced to balance the diversity and fluidity of memorized samples. Our diversified memory selection improves upon traditional reservoir sampling by updating the memory with samples that facilitate future learning and offers a way to inject domain knowledge into the incremental learning process.

We conduct experiments on the 30 largest stocks in the Tokyo Exchange that constitute the TOPIX Core30 Index by simulating the incremental training and testing scenario from 2013 to 2018. The results demonstrate that our volume prediction model can outperform existing incremental learning methods by improving the efficiency of knowledge transfer.

2 Related Work

Human can accumulate knowledge throughout their entire life. Deep learning, however, though outperforming human beings in some single-task scenarios, has been shown to experience the problem of *catastrophic forgetting* [McCloskey and Cohen, 1989; Ratcliff, 1990] when faced with a sequence of learning tasks. In other words, training models on a new task results in a drastic deterioration of performance on previous tasks. As a step towards artificial general intelligence, great efforts have been made in the area of *incremental learning* with the aim of endowing neural networks with the human-like ability to acquire new knowledge in an incremental manner [Lange *et al.*, 2021; Mai *et al.*, 2022; Awasthi and Sarawagi, 2019].

As a general term describing machine learning on a data stream, incremental learning includes many different settings. Two of the settings that have drawn the most attention from the research community are task-incremental learning and class-incremental learning. They both assume clear task boundaries with an inclination for classification tasks and mainly differ in whether task identities are known during inference. In this work, we focus on a slightly different setting, *i.e.*, performing regression on a stream of time series data where no natural task boundaries are available.

Existing work on incremental learning broadly falls into three categories: regularization-based methods, memory-based methods and architectural methods.

Regularization-based methods constrain the deviation of network parameters that are important to previous tasks by incorporating a regularization term into the training objective. The regularizer is typically computed as the weighted l_2 norm of parameter deviation from previous models where the weights correspond to the parameter importance. Different methods to estimate parameter importance have been proposed. For example, Elastic Weight Consolidation (EWC) [Kirkpatrick *et al.*, 2016; Schwarz *et al.*, 2018] measures parameter importance using the diagonal of the Fisher information matrix from a Bayesian perspective; Synaptic Intelligence (SI) [Zenke *et al.*, 2017] estimates the parameter importance as the cumulative parameter-specific contribution to changes in the total loss; Memory-Aware Synapses (MAS) [Aljundi *et al.*, 2018] measures the parameter importance using the gradients of the squared l_2 norm of model outputs, which makes it applicable to unsupervised scenarios.

Memory-based methods tackle the catastrophic forgetting problem by making use of a memory buffer. The memory is filled with samples from previous tasks during past training processes and is combined with the incoming data to update the model. For example, Experience Replay (ER) [Chaudhry *et al.*, 2019] simply interleaves new training data with samples in the memory during training. More sophisticated methods including iCaRL [Rebuffi *et al.*, 2017] and Dark Experience Replay [Buzzega *et al.*, 2020] attempt to achieve more efficient knowledge transfer through knowledge distillation. Gradient Episodic Memory (GEM) [Lopez-Paz and Ranzato, 2017] and its online variant, Averaged Gradient Episodic Memory (A-GEM) [Chaudhry *et al.*, 2019], keep track of the parameter gradients on the memorized samples and use them to enforce the constraint that the loss on previous tasks does not drop. Another line of research in this paradigm including Gradient-based Sample Selection (GSS) [Aljundi *et al.*, 2019b] and Maximally-Interfered Retrieval (MIR) [Aljundi *et al.*, 2019a] focuses on how to update the memory and which samples to retrieve from the memory. As a general approach to alleviating catastrophic forgetting, memory-based methods can be applied to a wide range of scenarios. We also build our work upon the memory-based framework and devise novel methods for incrementally updating the model and the memory in the task of incremental stock volume prediction.

Architectural methods [Rusu *et al.*, 2016; Serrà *et al.*, 2018; Mallya and Lazebnik, 2018] take a more direct approach by using different sets of parameters for different tasks. For example, Progressive Neural Networks [Rusu *et al.*, 2016] propose to freeze previously learned parameters and instantiate a new network to accommodate each new task. The new network is connected to the old one via lateral connections to achieve knowledge transfer. These methods, however, require well-defined task boundaries and may incur an excessive computational cost and memory requirement, and therefore are not suitable for our purpose.

Related Learning Scenarios

In volume prediction, our primary goal is to accurately forecast the trading volume for the upcoming hour/day. Therefore, we only require the model to yield good performance on future data, as opposed to most incremental learning literature which also measures the model performance on previous tasks. Our scenario also slightly differs from online learning which requires the model to learn from a single pass of the data stream. Since volume prediction models do not need to be updated so frequently, we temporally split the data stream into small subsets and allow the model to go over each subset for multiple epochs.

3 Method

In this section, we first introduce the incremental volume prediction problem and the memory-based incremental learning

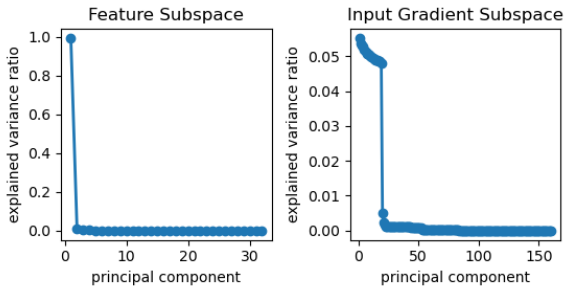

Figure 1: The explained variance ratios of the principal components for the feature subspace and the input gradient subspace. The number of non-zero values can be used to approximate the dimensionality of the subspaces.

framework. Then we elaborate on the two proposed methods for maximizing memory utilization: gradient distillation and diversified memory selection.

3.1 Preliminaries

An incremental learning problem typically consists of a collection of tasks $\{\mathcal{T}_i\}_{i=1}^{N}$ with varied distributions. The model is trained on these tasks sequentially and expected to retain its knowledge about previous tasks after training on a new task.

In our volume prediction problem, there are no natural task boundaries. So we create tasks by artificially determining the boundaries to simulate incremental model updates. Since we do not care about the performance on previous tasks, we test the model on a future time period after training on each task to see whether the model has learned to combine the new knowledge with the old one to better predict volume trends.

Memory replay methods are a family of popular incremental learning algorithms. They utilize a memory $\mathcal{M}$ of fixed size M which contains training samples stored during previous tasks. In each incremental step, the model iterates over the data from the current task and goes through the following stages [Mai *et al.*, 2022]:

1. **Memory Retrieval**: Select some samples from the memory for the model update.
2. **Model Update**: Jointly use the data from the current task and the data retrieved from the memory to update the model.
3. **Memory Update**: Update the memory using the data from the current task.

Various approaches have been proposed that target the three stages to improve the overall performance. In our work, we focus on the model update stage and the memory update stage and propose novel methods to maximize the utilization of the memory in the volume prediction task.

3.2 Gradient Distillation

From the viewpoint of knowledge retention, the memorized samples serve as the medium through which we can extract the knowledge of previously trained models and transfer it to the current model. To maximize the utilization of the memory during the model update, it is necessary to increase the information contained in the transferring mechanism.

A naïve but strong baseline in the incremental learning literature is Experience Replay [Chaudhry *et al.*, 2019], which simply combines the data from the memory and the new data to update the model. Formally, Experience Replay adopts the following training objective when training on the i-th task,

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\ell(f(\mathbf{x}),y)\right] + \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{M}}\left[\ell(f(\mathbf{x}),y)\right] \quad (1)$$

where $\mathcal{D}_i$ is the distribution of task i. Prior work focusing on incremental classification tasks use output probabilities to distil knowledge from previous models to the current one by incorporating a regularization term into the training objective as follows [Rebuffi *et al.*, 2017],

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\ell(f(\mathbf{x}),y)\right] \\ &+ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{M}}\left[\ell(f(\mathbf{x}),y) + \lambda\cdot \mathrm{KL}(f(\mathbf{x})\|\tilde{f}(\mathbf{x}))\right]\end{aligned} \quad (2)$$

In this part, we use $\tilde{\cdot}(\mathbf{x})$ to denote the model output after training on the task containing the labeled data $(\mathbf{x}, y)$.

Since such information is unavailable in regression tasks, a natural substitute is to use the hidden states before the final linear layer to perform distillation. Assuming the model f consists of a representation encoder $h : \mathcal{X} \to \mathcal{Z} \subset \mathbb{R}^d$ and a linear prediction layer $g : \mathcal{Z} \to \mathbb{R}$, this feature distillation term can be written as

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\ell(f(\mathbf{x}),y)\right] \\ &+ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{M}}\left[\ell(f(\mathbf{x}),y) + \lambda\|h(\mathbf{x}) - \tilde{h}(\mathbf{x})\|^2\right]\end{aligned} \quad (3)$$

However, we observe that both naive experience replay and feature distillation yield suboptimal performance for the incremental volume prediction task. We argue that this is due to the limited information they provide about previous model knowledge. Naive experience replay uses only the scalar ground-truth value as the supervision signal. On the other hand, although feature distillation uses a high-dimensional vector, this latent space is also approximately one-dimensional, as shown by the PCA results in Figure 1.

To remedy this problem, we propose to use the gradient of the model output with respect to the input vector, namely $\nabla_{\mathbf{x}} f(\mathbf{x};\theta)$, as the carrier of model knowledge. The gradient with respect to $\mathbf{x}$ demonstrates how the model responds to local changes in the neighborhood of $\mathbf{x}$. As a result, it characterizes the model's knowledge about the decision process it has learned about previously seen samples. This insight has been exploited in previous work on model interpretability under the name of gradient-based attribution [Sundararajan *et al.*, 2017; Ancona *et al.*, 2018]. In this work, we demonstrate that we can also take advantage of this idea for more efficient knowledge transfer in the incremental setting.

To be more specific, during the memory update stage, alongside the selected sample $(\mathbf{x}_i, y_i)$, we also store the input gradient $\nabla_{\mathbf{x}}\tilde{f}(\mathbf{x}_i;\theta)$ into the memory. During the model update stage, we encourage the model to minimize the distance between the input gradient of the current model and that of the old model on memorized samples. Since the model $\tilde{f}$ may produce inaccurate predictions for some of the previous samples, we only apply our gradient distillation term to the samples where the current model underperforms the old

model. In short, we modify the training objective as follows,

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\left[\ell(f(\mathbf{x}),y)\right]\\&+\mathbb{E}_{(\mathbf{x},y)\sim\tilde{\mathcal{M}}}\left[\ell(f(\mathbf{x}),y)+\lambda\|\nabla f(\mathbf{x})-\nabla\tilde{f}(\mathbf{x})\|^2\right]\end{aligned} \tag{4}$$

where $\tilde{\mathcal{M}}=\{(\mathbf{x},y)\in\mathcal{M}||f(\mathbf{x})-y|>|\tilde{f}(\mathbf{x})-y|\}$.

Similar to the feature space, We conduct PCA on the space of input gradients as shown in Figure 1. The results show that the input gradients occupy a much larger subspace than experience replay and feature distillation. Consequently, our gradient distillation supplies the current model with more information about previously learned knowledge.

Gradient Distillation as Regularization

Previous research has looked into the reason behind the success of output distillation in the context of model compression [Hinton *et al.*, 2015; Yuan *et al.*, 2020]. They claim that the probabilities that a model predicts can encode its knowledge about the relationship between different classes. In the case of hand-written digit recognition, for example, the predicted probabilities convey information about which 2's look more like 3 and which 2's look more like 7. Output distillation thus acts as a soft label regularization term similar to label smoothing, which provides guidance as to how the model should generalize.

In the incremental scenario, generalization remains a big concern for memory-based approaches. When input distribution varies violently between tasks, the current task may have little information about certain parts of the input space. The only supervision signals it can receive are the relatively few samples stored in the memory. Therefore, without appropriate regularization, the model is likely to overfit previous distributions, particularly when the memory is small.

In the absence of logit information for regression tasks such as time series prediction, we argue that our gradient distillation term also has a regularization effect. To see why, consider the Taylor expansion of the model around some memorized sample $\mathbf{x}_0$.

$$f(\mathbf{x})=f(\mathbf{x}_0)+\nabla_{\mathbf{x}}f(\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)+o(\|\mathbf{x}-\mathbf{x}_0\|_2) \tag{5}$$

Besides the zeroth-order term $f(x_0)$, our gradient distillation term also encourages the current model to imitate the first-order behavior on memorized samples. This information imposes a more stringent constraint on the local landscape of the model on previous distributions. Therefore, gradient distillation serves as a form of implicit regularization and can prevent the overfitting of the samples in the memory.

3.3 Diversified Memory Selection

Most previous incremental learning methods use reservoir sampling [Vitter, 1985] for memory update, which replaces samples in the memory with new samples in a manner such that each of them has the same opportunity of staying in the memory. We seek to improve upon this strategy by diversifying the memorized samples. By increasing memory diversity, the selected samples are more representative of the knowledge of previous models and can facilitate the knowledge transfer process.

Formally, we first design a similarity function $\sigma(\mathbf{x}_1,\mathbf{x}_2)$ between any two input samples. Generally, this similarity function can be used to incorporate the expertise of specific applications as well as the prior knowledge of the model. In our case, we calculate the similarity as the sum of the following three terms:

- **Stock correlation:** the correlation between the two associated stocks computed using historical returns;
- **Time indicator:** a binary indicator of whether the two samples are obtained at the same time of the day;
- **Model knowledge:** the cosine similarity between the two samples' input gradients.

Given several candidate samples x, we solve the following quadratic programming problem,

$$\begin{aligned}\min_{w_j\in[0,1],1\le j\le J}\ &\mathbf{w}^T\mathbf{\Sigma}\mathbf{w}\\ \text{s.t.}\quad &\mathbf{w}^T\mathbf{1}=1\end{aligned} \tag{6}$$

where $\Sigma_{i,j}=\sigma(\mathbf{x}_i,\mathbf{x}_j)$ and $\mathbf{w}$ is the weight vector where the i-th component w_i corresponds to the weight assigned to the i-th sample x_i.

Each entry of w corresponds to the weight assigned to the corresponding candidate sample. To decide which sample should be selected into the memory, one way is to select samples with the top-M weights. This, however, may result in a stagnant memory as training proceeds, where the majority of the memory remains the same across tasks. To remedy this problem, we propose to sample the candidates according to the weights and introduce a tuning parameter T to achieve balance between diversity and fluidity. The modified weights are calculated as

$$\tilde{w_i}\sim w_i^{\frac{1}{T}} \tag{7}$$

Note that $\log w_i^{\frac{1}{T}}=\frac{\log w_i}{T}$. Therefore, the hyper-parameter T is equivalent to the temperature in softmax that is widely used in classification tasks. A large T pushes all the weights towards 1 and resembles uniform sampling, while a small T close to 0 behaves more like choosing the top-M samples.

4 Experiments

In this section, we present our empirical results on a real-world incremental volume prediction dataset.

4.1 Incremental Volume Prediction

We conduct experiments on the task of incremental volume prediction by simulating the incremental scenario on stock transaction data from the Tokyo Stock Exchange to demonstrate the effectiveness of our method.

Dataset and Input Format

The dataset we use consists of stock transaction data from 2013-01-04 to 2018-02-08 on the 30 largest stocks of the Tokyo Stock Exchange that constitute the TOPIX Core30 Index. It contains the five-dimensional OHLCV (open/high/low/close/volume) data on a five-minute scale. For each time step, the input is composed of two sequences: the data from the previous 12 time steps, and the data from the previous 20 trading days at the same time. All input values are transformed onto the log scale.

Method	MSE (-)	RMSE (-)	MAE (-)	Accuracy (+)
Mean20	0.4560	0.6753	0.5128	0.6804
Finetune	0.3593 (±0.0075)	0.5994 (±0.0063)	0.4549 (±0.0054)	0.6898 (±0.0028)
EWC	0.3513 (±0.0034)	0.5927 (±0.0029)	0.4489 (±0.0025)	0.6925 (±0.0003)
SI	0.3649 (±0.0073)	0.6040 (±0.0061)	0.4586 (±0.0048)	0.6883 (±0.0019)
MAS	0.3615 (±0.0089)	0.6012 (±0.0074)	0.4560 (±0.0064)	0.6897 (±0.0027)
ER	0.3559 (±0.0038)	0.5966 (±0.0032)	0.4502 (±0.0029)	0.6913 (±0.0013)
A-GEM	0.3530 (±0.0029)	0.5941 (±0.0024)	0.4485 (±0.0026)	0.6921 (±0.0023)
FeatDistil	0.3432 (±0.0041)	0.5858 (±0.0035)	0.4424 (±0.0034)	0.6931 (±0.0033)
Ours	**0.3373 (±0.0055)**	**0.5808 (±0.0047)**	**0.4384 (±0.0036)**	**0.6969 (±0.0013)**

Table 1: Results on the incremental stock volume prediction experiment. The results are averaged over five runs with different random seeds (except for Mean20) and the figures in brackets denote the standard deviation. (-) means the lower the better and (+) means the higher the better.

Evaluation Protocol

We adopt the training and evaluation protocol similar to the one in Chaudhry *et al.* [2019]. Specifically, we temporally split available data into two time periods with equal length: the validation phase and the test phase. The validation phase is used to select the hyper-parameters, while the test phase is used to compare the performance between different models. We keep the two phases fully independent of each other by re-initializing the model at the beginning of the test phase and training it from scratch. This ensures that the hyper-parameters are not optimized for test performance.

To simulate the incremental learning scenario, each phase is further divided into smaller subsets by time as the incremental steps. The first subset is larger than the others and serves as a warmup stage, while the subsequent ones are all evenly spaced. Within each phase, the model is trained sequentially on these subsets. After updating the model on one subset, we immediately test the model on the next subset. The average score over the subsets is then used to measure the performance of the model in this phase.

In our experiments, we set the length of the warmup subset as 120 days and the subsequent subsets as 10 days each. The model is trained on each subset for three epochs before testing on the next subset. The number of subsets is 51 for the validation test phase each (including the warmup stage).

Model and Hyper-parameters

For the base learning model, we choose two separate one-layer bi-directional LSTMs [Hochreiter and Schmidhuber, 1997] to encode the two input sequences, respectively. We then apply attentive pooling to the outputs of each LSTM and concatenate the two pooled vectors into one vector, which is fed into a two-layer MLP to produce the predicted volume.

During training, we use the Adam [Kingma and Ba, 2015] optimizer with a learning rate of 1e-3 and a batch size of 64. The size of the memory is set to 64 as well. When updating the memory, due to the excessive computational cost of quadratic programming in the case of a large candidate set, we perform memory update by dividing the set into small batches containing 64 new samples each. The memory is then updated by sampling from the 128 candidates (64 from the memory and 64 from the new data) for each batch.

Baselines

We compare our approach against several non-incremental and incremental baselines. The non-incremental methods we use include

- **Mean20**: Use the average volume of the previous 20 days at the same time as the prediction.
- **Finetune**: Train the model sequentially with no regularization terms or memories.

For regularization-based incremental learning methods, we use the online version of **Elastic Weight Consolidation (EWC)**, **Synaptic Intelligence (SI)** and **Memory-Aware Synapses (MAS)**. For memory-based approaches, we use naive **Experience Replay (ER)** and **Averaged Gradient Episodic Memory (A-GEM)**. We also compare our method against a simple adaptation of distillation methods as introduced in Section 3.2, which we term **Feature Distillation (FeatDistil)**. All the memory-based baselines above use reservoir sampling for the memory update stage.

Results

We measure the mean squared error (MSE), the root mean squared error (RMSE), and the mean absolute error (MAE) of the proposed method and the baselines. We also compute the accuracy of these methods where we say a model makes a correct prediction if the predicted volume changes in the same direction as the ground truth value compared to the last time step. We test each method (except Mean20) five times with different random seeds and present the results in Table 1. As shown in Table 1, some regularization-based methods including SI and MAS even underperform the non-incremental finetuning approach. The improvements brought by previous memory-based methods are also less than satisfactory as they are mainly focused on classification tasks and fail to achieve adequate memory utilization. These results demonstrate that previous incremental learning methods may not work well for the task of incremental time series regression. In contrast, our method fills this gap by designing a more efficient transfer mechanism and a better memory selection algorithm, which take the nature of regression tasks into consideration. Consequently, our method not only significantly outperforms the non-incremental approaches, but

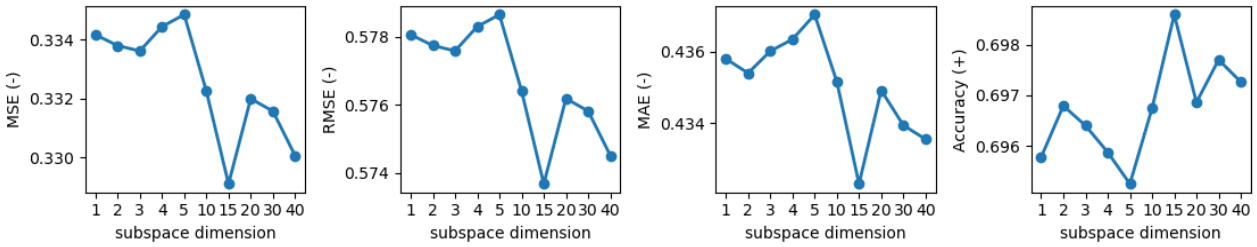

Figure 2: Results on incremental volume prediction using distillation of input gradients projected to subspaces of different dimensionality.

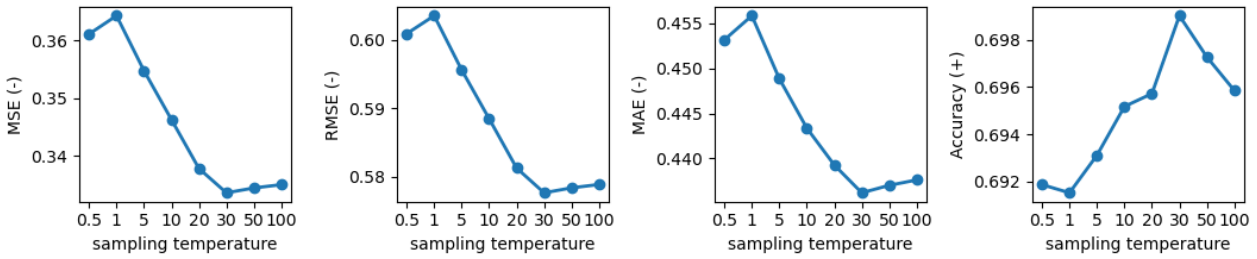

Figure 3: Results on incremental volume prediction using different sampling temperature values for memory update.

also yields better results compared to common regularization-based and memory-based approaches.

4.2 Effect of Subspace Dimension on Distillation

In Section 3.2, we claim that our gradient distillation term provides more information for transferring the knowledge of previous models to the current one. Here we aim to verify this claim. To be specific, we estimate the mean and the covariance matrix of the input gradients during the memory update stage in an online fashion. We then perform eigendecomposition on the covariance matrix to get the principal components of the input gradient space. Before distilling the gradient, we project the input gradients into a k-dimensional space spanned by the first k principal components. We then vary the number of k and compare the results. As shown in Figure 2, increasing the number of dimensions is generally beneficial to the incremental learning process while reducing the dimensionality of the gradient space hurts the performance due to information loss. This result validates our hypothesis that gradient distillation can improve the efficiency of knowledge transfer by supplying richer information to the model.

4.3 Trade-off between Sample Diversity and Fluidity

We also experiment with different values of memory sampling temperature T to investigate the trade-off between sample diversity and fluidity. The results are presented in Figure 3. As can be seen from Figure 3, a small T has a relatively large negative effect on the model performance since it prevents the memory from admitting new samples. A very large T is also suboptimal since it does not consider the diversity of the memory and effectively samples each incoming input uniformly at random. The best performance is achieved in the case of a moderate T which achieves a balance between sample diversity and fluidity. Figure 3 also suggests that our memory-based approach is more sensitive to the absence of sample fluidity than that of diversity, partly explaining why reservoir sampling performs reasonably well in a wide range of scenarios.

5 Conclusion

In this work, we propose a novel incremental time series regression method based on the existing memory-based incremental learning framework. Our approach aims to maximize the utilization of the memory by using input gradient distillation for efficient knowledge transfer and by maintaining a diversified memory. We apply our approach to the task of incremental volume prediction and demonstrate its superiority to existing methods. Besides the practical value, we also point out the gap when applying previous incremental learning algorithms to regression tasks and offer some new insights on the relationship between feature subspace dimensionality and model performance. We hope our work will draw more attention to the research on incremental time series regression.

References

[Aljundi *et al.*, 2018] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV (3)*, volume 11207 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2018.

[Aljundi *et al.*, 2019a] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *CoRR*, abs/1908.04742, 2019.

[Aljundi *et al.*, 2019b] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019.

[Ancona *et al.*, 2018] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR (Poster)*. OpenReview.net, 2018.

[Awasthi and Sarawagi, 2019] Abhijeet Awasthi and Sunita Sarawagi. Continual learning with neural networks: A review. *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2019.

[Buzzega *et al.*, 2020] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020.

[Chaudhry *et al.*, 2019] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *ICLR (Poster)*. OpenReview.net, 2019.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[Kirkpatrick *et al.*, 2016] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

[Lange *et al.*, 2021] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ale Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2021.

[Libman *et al.*, 2019] Daniel S. Libman, Simi Haber, and Mary Schaps. Volume prediction with neural networks. *Frontiers in Artificial Intelligence*, 2, 2019.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, pages 6467–6476, 2017.

[Mai *et al.*, 2022] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo J. Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[Mallya and Lazebnik, 2018] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773. Computer Vision Foundation / IEEE Computer Society, 2018.

[McCloskey and Cohen, 1989] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

[Ratcliff, 1990] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97 2:285–308, 1990.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542. IEEE Computer Society, 2017.

[Rusu *et al.*, 2016] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[Schwarz *et al.*, 2018] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 4535–4544. PMLR, 2018.

[Serrà *et al.*, 2018] Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 4555–4564. PMLR, 2018.

[Sundararajan *et al.*, 2017] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.

[Vitter, 1985] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11:37–57, 1985.

[Yuan *et al.*, 2020] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, pages 3902–3910. Computer Vision Foundation / IEEE, 2020.

[Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR, 2017.

[Zhao *et al.*, 2021] Liang Zhao, Wei Li, Ruihan Bao, Keiko Harimoto, Yunfang Wu, and Xu Sun. Long-term, short-term and sudden event: Trading volume movement prediction with graph-based multi-view modeling. In *IJCAI*, pages 3764–3770. ijcai.org, 2021.