Adaptive Deep Models for Incremental Learning: Considering Capacity Scalability and Sustainability*

Yang Yang
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023
yangy@lamda.nju.edu.cn

Da-Wei Zhou
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023
zhoudw@lamda.nju.edu.cn

De-Chuan Zhan*
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023
zhandc@nju.edu.cn

Hui Xiong*
Rutgers University
hxiong@rutgers.edu

Yuan Jiang
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023
jiangy@lamda.nju.edu.cn

ABSTRACT

Recent years have witnessed growing interests in developing deep models for incremental learning. However, existing approaches often utilize the fixed structure and online backpropagation for deep model optimization, which is difficult to be implemented for incremental data scenarios. Indeed, for streaming data, there are two main challenges for building deep incremental models. First, there is a requirement to develop deep incremental models with *Capacity Scalability*. In other words, the entire training data are not available before learning the task. It is a challenge to make the deep model structure scaling with streaming data for flexible model evolution and faster convergence. Second, since the stream data distribution usually changes in nature (concept drift), there is a constraint for *Capacity Sustainability*. That is, how to update the model while preserving previous knowledge for overcoming the catastrophic forgetting. To this end, in this paper, we develop an incremental adaptive deep model (IADM) for dealing with the above two capacity challenges in real-world incremental data scenarios. Specifically, IADM provides an extra attention model for the hidden layers, which aims to learn deep models with adaptive depth from streaming data and enables capacity scalability. Also, we address capacity sustainability by exploiting the attention based fisher information matrix, which can prevent the forgetting in consequence. Finally, we conduct extensive experiments on real-world data and show that IADM outperforms the state-of-the-art methods with a substantial margin. Moreover, we show that IADM has better capacity scalability and sustainability in incremental learning scenarios.

KEYWORDS

Deep Incremental Learning, Capacity Scalability, Capacity Sustainability

ACM Reference Format:
Yang Yang, Da-Wei Zhou, De-Chuan Zhan*, Hui Xiong*, and Yuan Jiang. 2019. Adaptive Deep Models for Incremental Learning: Considering Capacity Scalability and Sustainability. In *KDD '19: The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 3–7, 2018, Anchorage, Alaska.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Nowadays, a large amount of the streaming data, such as traffic flows, sensor data, and query logs, have been accumulated in many application scenarios. As a result, there is a critical need for developing methods for incremental learning [36]. Indeed, tremendous efforts have been made for incremental learning in different application domains, such as incremental recommendation [5], demand prediction [33], and graphlet matching [7]. However, most existing incremental learning methods are with shallow structures (e.g., linear or kernel) [16, 39], which were not designed to learn complex nonlinear functions.

Deep learning techniques have achieved a wide range of successes with powerful nonlinear models, such as biomedical detection [30], article analysis [35], and semantic representation [11]. However, existing deep models are trained in a batch learning setting with the entire training data and are not designed for incremental learning tasks. Therefore, there is a need to perform *Deep Incremental Learning* (DIL). A direct way to do DIL is applying the standard backpropagation training for the pre-fixed model with only single instance at each round. Such an approach is simple but has several limitations, particularly for solving the capacity of the model. First, different from off-line learning requiring the

** is the corresponding author

KDD '19, August 3–7, 2018, Anchorage, Alaska

ACM ISBN 978-1-4503-9999-9/18/06. . . $15.00
https://doi.org/10.1145/1122445.1122456

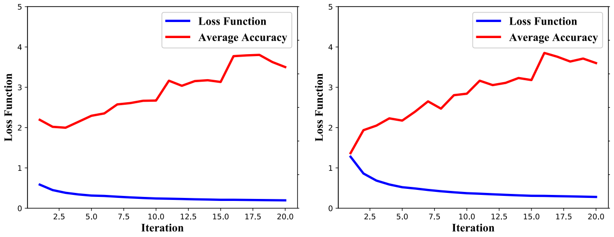
entire training data available in prior, incremental learning manages to optimize classifiers over the stream of data. This requires the models should have flexible structures, which can scale with the stream data for convergence and performance improvement. This challenge is defined as the "*Capacity Scalability*". As to deep incremental models, it is significantly more challenging at choosing a proper model structure (i.e., depth of the deep model) with the stream data, as shown in Figure 1. Indeed, the learning process will converge slowly if the model is too complex, while the capacity is restricted if the model is simple. In addition, no validation data are available in incremental settings, thus it is not realistic to address this issue as batch learning.

Furthermore, it is notable that many data stream is evolving in nature. That is, the joint distribution between the input feature and the ground truth will change as the concept drift [12]. If we ignore the distribution change, the performance of previous distribution will dramatically drop down as the catastrophic forgetting phenomenon [29]. For instance, in Figure 2, we can find that the knowledge learnt from the previous distribution (X_1) will lost when information relevant to the current distribution (X_2) is incorporated. This challenge is defined as the "*Capacity Sustainability*". However, previous DIL methods rarely consider this crucial problem. Recently, fisher information matrix is introduced for preventing this problem [4, 19, 26], while these methods concentrate on the life-long multi-task learning with obvious task conversion. Moreover, these methods ignore that the importance of different parts in the fisher information matrix are also adapting with the evolution of the model structure evolution. These two problems are concurrent, and impose the challenge to exploit these methods for developing deep incremental learning models.

To this end, in this paper, we design the "Incremental Adaptive Deep Model" (IADM) framework, and propose a novel end-to-end adaptable deep model considering both capacity scalability and sustainability challenges. Specifically, IADM can evolve from a shallow network (fast convergence) to deep model (large capacity) adaptively with the stream data, which can effectively improve overall prediction performance. Meanwhile, IADM is knowledgable about the past and present data distribution with the adaptive fisher regularization. As a matter of fact, it can accurately reflect whether the algorithm utilizes the model capacity efficiently. Finally, we provide an extensive analysis for understanding the performances of incremental learning algorithms on various real-world streaming data. The results show that IADM outperforms the state-of-the-art methods with a substantial margin. Moreover, we show that IADM has better capacity scalability and sustainability in incremental scenarios.

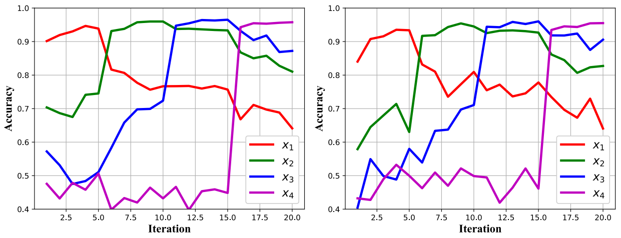
2 RELATED WORK

The exploitation of deep incremental learning has attracted much attentions recently. Considering the heavy storage memory, we can only acquire the real-time data or fixed transitory period data, which expresses the requirements on the scalability of the model capacity. Moverover, note that joint distribution is evolving with the stream data, which causes the forgetting of previous learned knowledge, thus the capacity sustainability is also a challenge. These two problems are always concurrent in deep incremental learning.



(a) Shallow Model (b) Deep Model

Figure 1: Performance measure (Loss vs. Accuracy) with different network structures on MNIST. (a) network with 1 fully connected hidden layer; (b) network with 5 fully connected hidden layers.



(a) Shallow Model (b) Deep Model

Figure 2: Catastrophic forgetting phenomenon. In detail, we construct 4 stages from MNIST dataset, and each stage removes $\frac{1}{4}$ of the image (top left as X_1, top right as X_2, bottom left as X_3, bottom right as X_4), which is submitted to the concept drift scenario. It reveals that with the stream data, the previous stages will appear forgetting as the accuracy decreasing for both shallow and deep models.

Incremental Learning Considering Scalability

Incremental learning aims to update the models from data stream sequentially, and has achieved many successes in both application and theory [17, 37]. However, previous models are designed with linear function or kernel metric, which are hardly to be extended to non-linear models with high capacity. With the development of deep learning, it shows that the incremental learning setting can be directly applied in deep models with online backpropagation, yet with many drawbacks, i.e., convergence limitation (gradient vanishing and diminishing feature reuse). Thus, Lee et al. proposed a dual memory architecture that process slow-changing global patterns [27]; Zhou et al. proposed an incremental feature learning algorithm to determine the optimal model complexity based on the autoencoder [38]. However, they operate sliding window approach with batch training stage, making them unsuitable for the streaming data. Besides, considering the limitations of the fixed model structure, which also cannot be validated easily in the incremental setting. The most relevant work to our approach is the [31], which proposes a novel framework for deep models in the incremental setting, and adapts the model capacity from simple to complex incrementally, combining the merits of both incremental learning and deep learning consequently. However, the weight delay weight of the proposed Hedge Backpropagation (HBP) causes lower layers to be difficult to train, making it difficult to adaptively update parameters.

Incremental Learning Considering Sustainability

Concept drift caused by the distribution evolution is a well-recognized research direction in incremental learning and has wide applications [12, 14]. Previous methods always assume that there are some useful knowledge for future prediction in previous data, and only concentrate on the current task. These methods can be divided into three categories: sliding window based approaches, which maintain the nearest data items and discard old items [21]; evolving based approaches, which downweight previous data items according to time series [20]; ensemble based approaches, which can adaptively add or delete classifiers and dynamically adjust weights [2]. However, these methods ignore an important phenomenon in incremental learning, that is the catastrophic forgetting, which is the tendency for losing the learnt knowledge of previous distribution. To mitigate the catastrophic forgetting, there are many attempts, including ensemble methods combine multiple classifiers for final prediction [9]; rehearsal methods mix data from earlier sessions [13]; dual-memory models store memories in two distinct neural networks [13]; sparse-coding methods reduce the forgetting by learning sparse representation [8], readers can refer to the introduction for further information [18]. Recently, many researches are concentrate on utilizing the fisher information matrix and have achieved excellent performance. Fisher information is a way of measuring the amount of information that an unknown parameter Θ of a distribution models the data X, which is related to relative entropy and can be represented as Kullback-Leibler divergence form. Kirkpatrick et al. proposed the elastic weight consolidation to reduce catastrophic forgetting in artificial neural networks [19]. Lee et al. proposed to incrementally match the moment of the posterior distribution of the neural network [26]. Lee et al. dynamically decided the network capacity for lifelong learning [24]. While these methods are multi-task methods, which require clear task segmentation, and can not be directly applied to the incremental learning setting.

Therefore, to solve these crucial two challenges, we propose an Incremental Adaptive Deep Model (IADM) with attention mechanism. In detail, IADM utilizes an extra shallow attention network to learn the attention weights for the hidden layers. As a result, IADM can concentrate on the shallow layers with large attention weights firstly for fast convergence, while acquiring high capacity by considering deep structure with the increase of data. On the other hand, different from only considering the fisher information matrix of last task, IADM incorporates the learned attention weight of each hidden layer into the corresponding parameters in fisher matrix. Consequently, IADM can adaptively update both the deep structure and fisher information matrix.

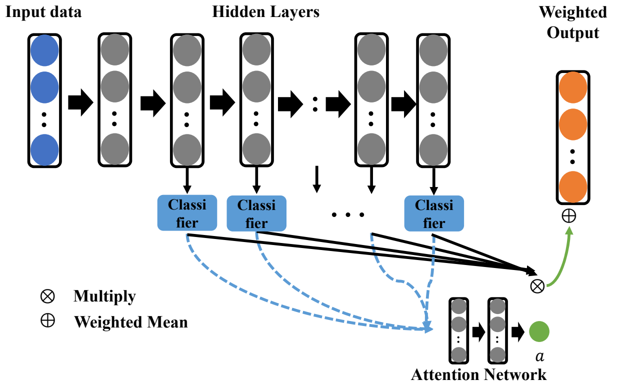


Figure 3: Illustration of the proposed IADM. Specifically, with the stream data, we build independent classifiers for the hidden layers, and utilize an extra attention model to calculate the attention weights for these classifiers for the final ensemble. Meanwhile, we also integrate the weights with the fisher information matrix as the adaptive regularization for relieving forgetting.

3 PROPOSED METHOD

3.1 Notations

In this paper, we consider the problem of incrementally training the deep model considering the concept drift with the stream data. Specifically, our goal is to learn an adaptive model $\mathbf{f}: \mathbb{R}^d \to \mathbb{R}^C$ for specific task with sequence instances. $X = \{X_1, X_2, \cdots, X_t, \cdots, X_T\}$ denotes stream data with unbounded T, where the instance stage X_t comes with training data $\mathcal{D}_t = \{\mathbf{x}_{t,i}, \mathbf{y}_{t,i}\}_{i=1}^{N_t}$, $N_t \geq 1$ is the number of examples in stage t. Without any loss of generality, note that the instance stage X_t is set manually, i.e., real-time single instance; a transitory period data (fixed time); a fixed number of data. $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional instance representation, $\mathbf{y} \in \{0, 1\}^C$, C is the number of classes. The main challenge in the incremental learning setting is that all the previous training data are not available at the current time t (only the model parameters of the previous stages are accessible).

3.2 Capacity Scalability and Sustainability

We now develop a deeper understanding for the deep incremental learning with concept drift, in which the capacity scalability and sustainability problems are concurrent, while previous methods have rarely considered.

It is notable that using deep incremental model faces several issues. Previous methods always fix the structure of the neural network in prior, and cannot be changed during the training process. Therefore, it is a difficult task to determine the deep model in advance, e.g., the depth, while in the incremental setting, different depths are suitable for different numbers of instances, e.g., from the Figure 1, shallow network is with fast convergence, while with restricted learning capacity. Correspondingly, deep network is with larger capacity, yet the learning process will converge slowly. In conclusion, our framework aims to exploit the fast convergence of shallow network at the initial stage, and utilize the power of deep representation gradually in the following stages.

On the other hand, considering the objective of the incremental learning is to keep on learning with stream data, it should be evaluated on both the past and present examples of the learned model. However, in real application, the distribution usually change with the data collection sequence, which is referred to as concept drift. From the Figure 2, in the concept drift scenario, it reveals that the accuracy of the model over different stages X_t will continue to decrease over future stages. Thus, there is additional crucial component that need to be considered: catastrophic forgetting, which is how much an algorithm forgets what it has learned in the past data. Intuitively, we want to build a model that considers both the ability of preventing forgetting and learning new distribution instances, thus efficiently reflecting the model capacity sustainability.

Without any loss of generality, with a given θ, which is the conditional likelihood distribution learned by the model. A prediction can be defined as the sample obtained from the likelihood distribution $p_\theta(\mathbf{y}|\mathbf{x})$. F_θ, known as the "Empirical Fisher Information Matrix" [1, 28] at θ, is defined as: $F_\theta = E_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\big[(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial\theta}\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial\theta}^\top)\big]$, where $\mathcal{D}$ is the instance domain. It is notable that the log-likelihood $\log p_\theta(\mathbf{y}|\mathbf{x})$ is the same as the negative of the cross-entropy loss function in deep model for simplicity. Thus, F_θ can be seen as the expected loss of gradient covariance matrix. F_θ has 3 key properties [28]: 1) is equivalent to the second derivative of the loss near a minimum; 2) can be computed from first-order derivatives alone and is thus easy for large models; 3) is guaranteed to be positive semi-definite. On the other hand, let $D_{KL}(p_\theta \| p_{\theta+\triangle\theta})$ be the KL-divergence [34] between the conditional likelihood of the model at θ and $\theta+\triangle\theta$, when $\triangle\theta \to 0$, the second-order Taylor approximation of KL-divergence can be written as $D_{KL}(p_\theta \| p_{\theta+\triangle\theta}) \approx \frac{1}{2}\triangle\theta^\top F_\theta \triangle\theta$, which is also equivalent to computing distance in a Riemannian manifold [25]. Since $F_\theta \in \mathbb{R}^{d_\theta \times d_\theta}$ and d_θ is usually with millions for neural networks, it is practically infeasible to store F_θ. To handle this problem, according to [19], we assume parameters to be independent of each other (only using the diagonal parameters of F_θ), which results in the following approximation:

$$L = L_t + \frac{\lambda}{2}\sum_i F_{\theta_{t-1_i}}(\theta_{t_i} - \theta^*_{t-1_i}) \tag{1}$$

where L_t is the loss for $t-$th stage only, θ_{t_i} is the $i-$th entry of θ at stage t, λ represents how important the last stage compares to the new one. It is notable the fisher regularization will try to keep the important parameters close to the learned parameters of previous stage.

3.3 Incremental Adaptive Deep Model

In this section, we address the incrmental adaptive deep model (IADM) in an unified framework. IADM ingeniously illustrate both the capacity scalability and sustainability problems in designing, i.e., attention based model expansion for capacity scalability, and weighted fisher regularization for capacity sustainability. Specifically, we adopt an additional shallow network to learn the attention weights for the classifiers built by the middle hidden layers, then fuse the multiple weighted hidden classifiers for the final prediction. Besides, for the fisher information matrix in different stages, we embed the learned attention weights to the corresponding elements in the fisher information matrix, which matches the moments of overall posterior distributions in an incremental way. Basically, our framework consists of two modules: 1) capacity scalability by evolutive deep network: IADM builds the adaptive model with extra attention weights for the hidden layers. Thus, we can exploit the shallow networks at the initial stage, and the deep representation at later stage; 2) capacity sustainability by weighted fisher regularization: IADM embeds hierarchical attention weights into fisher information matrix of different stages, which aims to match the the posterior distribution on all stages incrementally.

Evolutive Deep Network

Without any loss of generality, the deep neural network is with L hidden layers, i.e., the fully connected network is with L fully connected layers, the CNN is with L hidden blocks, here we assume that the maximum capacity of the network is with L hidden layers considering the existing computing ability. Different from the original network using the final feature representation h_L for prediction, in IADM, as shown in Figure 3, the final prediction is the weighted combination of outputs learnt using the middle hidden layer feature representations from $\{h_1, h_2, \cdots, h_L\}$. Following is the prediction function using attention based pooling:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{l=1}^{L} \alpha_l f_l \\ f_l &= softmax(h_l \Theta_l) \qquad \forall l = 1, 2, \cdots, L \end{aligned} \tag{2}$$

where f_l is the classifier using $l-$th hidden layer feature representations h_l, Θ_l is the parameters for f_l. $\alpha_l = g(f_l)$, $g(\cdot)$ is a shallow neural network (i.e., fully connected network) for calculating the weights for each output of the hidden layers, which aims to discover the relationships among hierarchical classifiers. At the end of every round, the weights α_l are normalized as $\sum \alpha_l = 1$. Therefore, the loss is:

$$L_t(f(\mathbf{x}), \mathbf{y}) = \ell_t(\sum_{l=1}^{L} \alpha_l f_l(\mathbf{x}), \mathbf{y}) \tag{3}$$

the loss function can be any convex function here, and we utilize the cross-entropy loss for simplicity. During the incremental learning procedure, we need to learn the $g(\cdot), \Theta_l, W_l$, W_l is the parameters for learning h_l. Different from the original backpropagation, where the error derivatives are backpropagated from the last output layer. In Eq. 3, the error derivatives are backpropagated from each classifier f_l, i.e., $W_l^{t+1} \leftarrow W_l^t - \eta \nabla_{W_l} \ell_t(\sum_{l=j}^{L} \alpha_l f_l(\mathbf{x}), \mathbf{y})$. We compute the gradient of the final prediction with respect to each depth parameters. Note that the summation can be started at $l = j$ in deep network, because the shallower blocks can be regarded for the basic feature extraction. Consequently, with the intuition that shallow models converge faster than deep models [6], using the attention mechanism will concentrate on the shallower layers with larger α at the initial stage, while with the increase of the data, larger α is learned for the deeper layers, which conforms to the capacity scalability. This gives an effective approach to learn the optimal network depth automatically in sequence.

Weighted Fisher Regularization

Here, we believe that the distribution of instances will not change drastically in a transitory stage X_t, i.e., users' interest will not change in a short time when following an online news stream. Furthermore, even for more complicated situations, we can adopt the drift detection algorithm to split the data stream into epoches in which the underlying distribution is relatively smooth. Thus, we regularize over the conditional likelihood distribution $p_\theta(\mathbf{y}|\mathbf{x})$ of every stages using the fisher information matrix, as Eq. 1, for the forgetting measure. Intuitively, using F_θ facilitates the network to learn parameters such that considering both the new and previous distributions.

It can be found that Eq. 1 only consider the fisher information matrix of the last stage, but has not considered all the previous stages. Thus there still will be a phenomenon of interval forgetting. This can be enforced either with multiple separate penalties, or as the sum of the quadratic penalties over different stages. While in incremental setting, with the network structure evolution with the attention mechanism, in other words, different layers of the network have different importance. Similarly, different parts of the fisher information matrix have different importance in sequential stages. Therefore, to incrementally matching the posterior distribution of the neural network trained on all stages, we embed the attention weights to the corresponding parameters of the fisher regularization, and the adaptive regularization can be represented as following:

$$R = \frac{1}{T} \sum_{t=2}^{T} \sum_{i} \alpha_{t-1} \odot F_{\theta_{t-1_i}} (\theta_{t_i} - \theta^*_{t-1_i})^2 \tag{4}$$

where $\alpha_t = [\alpha_{t,1}, \alpha_{t,2}, \cdots, \alpha_{t,L}]^\top$, $\odot$ means multiplying the α_l to the parameters of the corresponding layer in the fisher information matrix. This continuous averaging leads that the stages learned in previous are with less influence than the stages in recent.

Thus, considering both the Eq. 3 and Eq. 4 comprehensively, the whole loss function can be represented as:

$$L = \ell_t(\sum_{l=1}^{L} \alpha_l f_l(\mathbf{x}), \mathbf{y}) + \frac{\lambda}{T} \sum_{t=1}^{T} \alpha_{t-1} \odot F_{\theta_{t-1}} (\theta_t - \theta^*_{t-1})^2 \tag{5}$$

Furthermore, when θ_{t-1} is at a local minimum, gradients would be nearly zero, making $F_{\theta_{t-1}}$ very small. Theoretically, the regularization is negligible, which would result in catastrophic forgetting. However, experimentally we observed that this can be circumvented by using a very high value ($\approx 10^4$) for the hyperparameter λ.

4 EXPERIMENTS AND DISCUSSION

4.1 Datasets and Configurations

Previous incremental datasets considering concept drift are always with limited size, which cannot be adopted for deep models efficiently. Therefore, We first experiment on 1 synthetic (*Hyperplane* [10]) and 2 constructed image incremental datasets (*Incremental MNIST* [23], *Incremental CIFAR10* [22]), then give the analysis on 1 real-world datasets as action recognition (*Incremental UCF101* [32]), all the datasets are stream data with concept drift as [12]. In detail, **Hyperplane**: It is generated uniformly in a 10 dimensional hyperplane with 30,000 instances in total over 3 different stages for binary classification, which is a benchmark synthesis dataset for regression scenario; **Incremental MNIST**: The standard MNIST dataset is split into 4 stages considering concept drift, i.e., the instances in each stage remove $\frac{1}{4}$ of the images; **Incremental CIFAR10**: We extend CIFAR-10 dataset into 3 disjoint stages with more complex concept drift setting, the first stage is the raw data, and remaining two stages add gaussian noise with different level of intensity, i.e., (0.1, 0.02) for second stage and (0.3, 0.04) for third stage; **Incremental UCF101**: To validate the real application, we further evaluate on the real-world action recognition dataset, UCF101 is an action recognition dataset of realistic action videos, collected from YouTube, having 101 action categories, and can be grouped into 25 groups, where each group can consist of 4-7 videos of an action. For the incremental setting, we select the 5 longest groups.

For synthetic and UCF101 dataset, we randomly sample 30% of the examples at each stage for testing, remaining for training; for MNIST and CIFAR10 datasets, they have standard testing sets. To comprehensively evaluate IADM, for the synthetic and MNIST datasets, we adopt the fully connected network, and utilize the representative DCNN architecture for remianing 2 datasets, i.e., resnet18 [15]. The images are randomly flipped before passing into the network and no other data augmentation is utilized. The base learning rate is set to 0.001 and optimize with Adam. When the variation between the objective values of Eq. 5 is less than 10^{-5} in iterations, we consider IADM converges. We run the following experiments with the implementation of an environment on NVIDIA K80 GPUs server.

Finally, 5 criteria, i.e., average Accuracy, average Precision, average Recall, average F1, average AUC are taken to measure the performance, e.g., let $acc_{k,j}$ be the accuracy evaluated on the held-out set of the $j-$th stage ($j \le k$) after training the network incrementally from stage 1 to k, the average accuracy at stage k is defined as: $A_k = \frac{1}{k} \sum_{j=1}^{k} acc_{k,j}$ as [4], the higher of the A_k, the better of the classifier. Similarly, other average criteria can also be calculated. To validate the capacity scalability, we calculate the evolution of the

Table 1: Comparison results of IADM with both compared methods on 3 benchmark datasets an 1 real-world dataset. The best performance for each criterion is bolded. ↑ / ↓ indicate the larger/smaller the better.

Methods	Average Accuracy ↑				Average Precision ↑			
	Hyperplane	MNIST	CIFAR10	UCF101	Hyperplane	MNIST	CIFAR10	UCF101
Adwin	0.684	0.504	0.619	0.771	0.681	0.497	0.612	0.760
DNN-SGD	0.607	0.819	0.601	0.702	0.607	0.822	0.602	0.702
ODLD	0.616	0.817	0.655	0.847	0.616	0.820	0.656	0.846
DNN-Base	0.615	0.828	0.630	0.556	0.614	0.830	0.621	0.554
DNN-L2	0.608	0.826	0.611	0.571	0.607	0.826	0.664	0.568
DNN-EWC	0.638	0.874	0.622	0.717	0.639	0.874	0.621	0.711
Mean-IMM	0.611	0.863	0.654	0.684	0.612	0.863	0.654	0.684
Mode-IMM	0.654	0.876	0.655	0.619	0.653	0.874	0.651	0.620
DEN	0.645	0.717	0.646	0.749	0.644	0.713	0.647	0.749
IADM	**0.687**	**0.892**	**0.680**	**0.927**	**0.687**	**0.892**	**0.674**	**0.926**

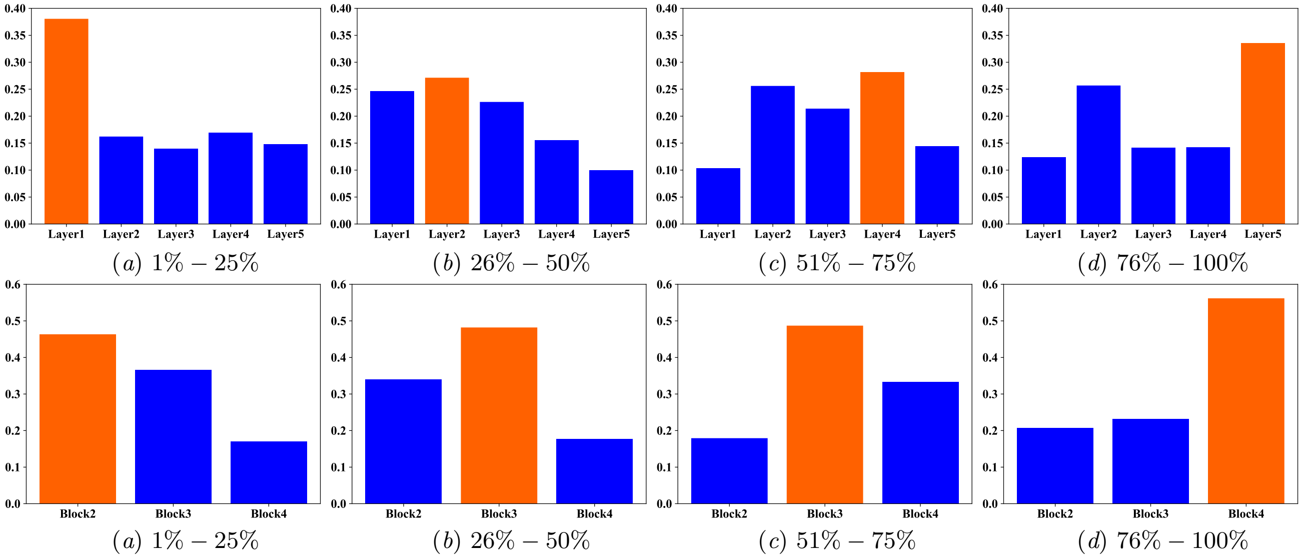
Methods	Average Recall ↑				Average F1 ↑			
	Hyperplane	MNIST	CIFAR10	UCF101	Hyperplane	MNIST	CIFAR10	UCF101
Adwin	0.684	0.642	0.625	0.796	0.682	0.441	0.618	0.758
DNN-SGD	0.614	0.866	0.687	0.668	0.601	0.811	0.610	0.669
ODLD	0.621	0.864	0.725	0.788	0.612	0.809	0.662	0.806
DNN-Base	0.616	0.867	0.699	0.536	0.613	0.821	0.638	0.507
DNN-L2	0.615	0.864	0.720	0.576	0.601	0.820	0.669	0.546
DNN-EWC	0.663	0.891	0.686	0.651	0.615	0.873	0.614	0.657
Mean-IMM	0.645	0.881	0.690	0.779	0.587	0.861	0.652	0.634
Mode-IMM	0.656	0.890	0.688	0.742	0.653	0.874	0.650	0.556
DEN	0.658	0.803	0.680	0.681	0.636	0.704	0.641	0.692
IADM	**0.713**	**0.901**	**0.714**	**0.880**	**0.696**	**0.891**	**0.683**	**0.900**

Methods	Average AUC ↑				Forgetting↓			
	Hyperplane	MNIST	CIFAR10	UCF101	Hyperplane	MNIST	CIFAR10	UCF101
Adwin	0.684	0.721	0.789	0.832	N/A	N/A	N/A	N/A
DNN-SGD	0.610	0.901	0.808	0.813	0.183	0.145	0.184	0.297
ODLD	0.621	0.899	0.778	0.904	0.159	0.156	0.167	0.153
DNN-Base	0.616	0.906	0.794	0.722	0.181	0.143	0.194	0.444
DNN-L2	0.608	0.903	0.813	0.732	0.175	0.077	0.128	0.428
DNN-EWC	0.640	0.930	0.790	0.823	0.145	0.067	0.203	0.282
Mean-IMM	0.612	0.924	0.808	0.802	0.103	0.064	0.138	0.212
Mode-IMM	0.655	0.931	0.809	0.761	0.034	0.047	0.137	0.231
DEN	0.645	0.841	0.803	0.843	0.120	0.153	0.123	0.247
IADM	**0.687**	**0.940**	**0.822**	**0.954**	**-0.011**	**0.043**	**0.111**	**0.073**

parameter α. Moveover, to validate the capacity sustainability, we calculate the performance about the forgetting profile of different learning algorithms as [4], i.e., the $\frac{A^*-mean(A)}{A^*}$, A^* is the optimal accuracy with the entire data.

4.2 Compared methods

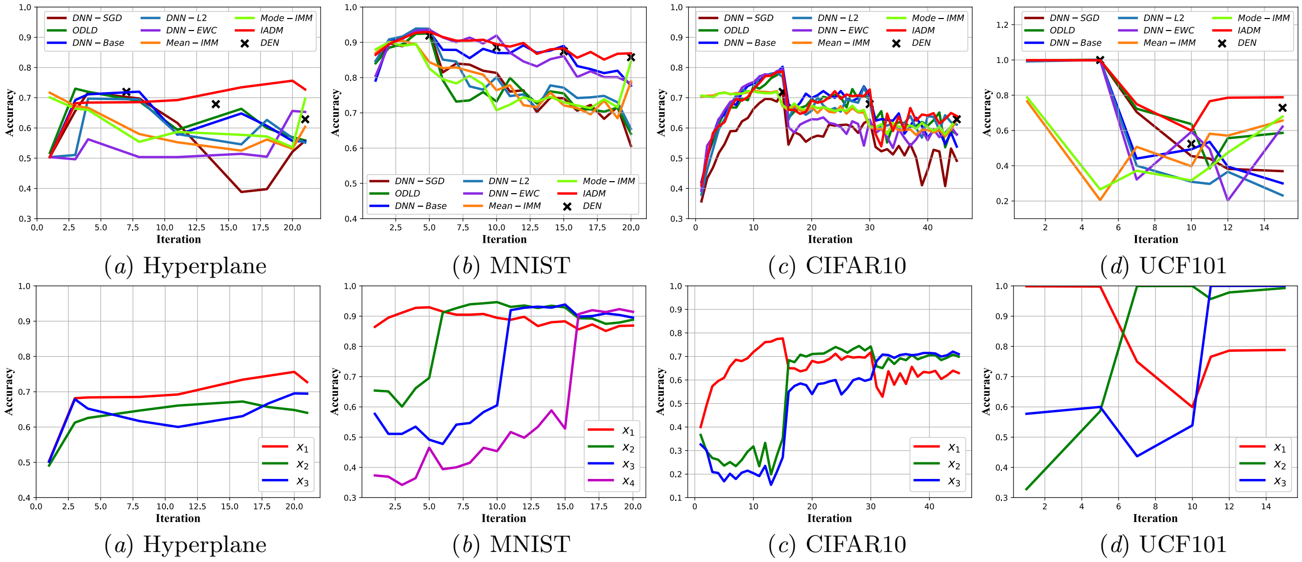
Considering IADM is related to the deep incremental learning with concept drift, several state-of-the-art methods are compared, i.e., DNN-SGD, Adwin [3], ODL [31]. Besides, in our experiments, IADM can be degenerated into catastrophic forgetting setting, therefore, several modified forgetting methods, i.e., DNN-Base, DNN-L2, DNN-EWC [19], IMM [26], DEN [24] are also compared, each stage is regarded as a task in these methods. In detail, the compared methods are: **Adwin:** A concept drift method, using adaptive online sliding windows according to the rate of change observed from the data; **DNN-SGD:** Base DNN with online backpropagation; **ODLD:** A online deep learning framework learns DNN models of adaptive depth from a sequence of training data in an incremental learning setting; **DNN-Base:** Base DNN



(a) 1% – 25% (b) 26% – 50% (c) 51% – 75% (d) 76% – 100%

(a) 1% – 25% (b) 26% – 50% (c) 51% – 75% (d) 76% – 100%

Figure 4: Evolution of weight distribution over various stages. Top row is MNIST, bottom row is CIFAR10.



(a) Hyperplane (b) MNIST (c) CIFAR10 (d) UCF101

(a) Hyperplane (b) MNIST (c) CIFAR10 (d) UCF101

Figure 5: Accuracy performance of different models and stages. The top row is the results of first stage about different methods over sequential stages, the bottom row is the results of different stages about IADM over sequential stages.

with L_2-regularizations; **DNN-L2:** Base DNN, where at each stage t, Θ_t is initialized as Θ_{t-1} and continuously trained with L_2-regularization between Θ_t and Θ_{t-1}; **DNN-EWC:** Deep network trained with elastic weight consolidation for regularization, which remembers old stages by selectively slowing down learning on the weights important for those stages; **IMM:** A incremental moment matching method with two extensions: Mean-IMM and Mode-IMM, which incrementally matches the posterior distribution of the neural network trained on the previous stages; **DEN:** A deep network architecture for incremental learning, which can dynamically decide its network structure with a sequence of stages, and learns the overlapping knowledge among stages.

4.3 Performance Measure

We report the results of all the datasets about the 5 criteria and the forgetting profile in Table 1. From the results, it can be obviously found that our IADM approach can achieve

Table 2: Comparison for fisher regularization, which drops different ratio of low energy parameters in the fisher information matrix. The best performance are bolded.

Methods	Hyperplane				MNIST			
	20%	40%	60%	80%	20%	40%	60%	80%
DNN-EWC	0.634	0.643	0.646	0.610	0.874	0.873	0.869	0.854
IADM	**0.649**	**0.669**	**0.660**	**0.645**	**0.886**	**0.887**	**0.889**	**0.859**
Methods	CIFAR10				UCF101			
	20%	40%	60%	80%	20%	40%	60%	80%
DNN-EWC	0.654	0.656	0.656	0.661	0.777	0.638	0.722	0.644
IADM	**0.661**	**0.677**	**0.662**	**0.679**	**0.900**	**0.892**	**0.887**	**0.877**

the best performance on all the datasets with different performance measures, which reveals that the IADM approach is a high-competitive method handling both the capacity scalability and sustainability challenges. For a more intuitive measurement of forgetting, we study the degree of forgetting among different models, which defines the forgetting for a particular task as the difference between the maximum knowledge gained about that task throughout the learning process and the knowledge we currently have about it, the lower the better. It shows that IADM is with the least forgetting. The N/A in Adwin because it is unable to get the intermediate result in the training process, thus the forgetting profile cannot be calculated. Negative value in forgetting profile means not only without forgetting, but also has the positive influence for the future classification.

To validate the effectiveness of learned fisher information matrix, we conduct more experiments comparing fisher regularization based methods. In detail, we drop the low energy parameters with low values in the fisher information matrix from 20% to 80%, 20% as the interval and record the performance after removing the low energy parameters in Table 2. From the results, it reveals that the performance will still be competitive instead after removing the low energy parameters similar to the dropout in the traditional deep learning, which further illustrates the effectiveness of the important parameters calculated by the adaptive fisher information matrix.

4.4 Capacity Adaptation

In this section, we evaluate the weight distribution (parameter α) learnt by IADM over different stages. We extract data from different stages at intervals of 25%, and analyse the mean weight distribution in different stages in Figure 4 on the MNIST and CIFAR10 dataset. The block 1 in CIFAR10 network is used for the basic feature extraction as mentioned before. From the results, it reveals that in the initial phase (first stage), the maximum weight locates at the shallow classifier. In the second stage, slightly deeper classifiers have picked up some weight, and in the following stages, deeper classifiers get more weight. Thus, the weight evolution shows that IADM has the ability to perform model selection. Meanwhile, different stages are with different depth indicates that IADM learns more discriminative features with more data, in other words, IADM uses the deeper classifiers to learn better features.

4.5 Evaluation of Forgetting

Due to page limitation, we report the performance of first stage for different datasets in top row of Figure 5, and the performance of different stages of IADM in the bottom row. For compare methods, note that the DEN utilized the timestamp to save the model of each stage for prediction, while the testing data are always unpredictable of the source stage in real applications as our setting, so we only use the latest model of the DEN for testing. From the top row, it reveals that the methods without considering the forgetting regularization (e.g., DNN-SGD, ODLD), the performance will steady decline, while IADM shows stable performances on almost all the datasets with slowly forgetting, and superior to other fisher regularization based method with the adaptive attention mechanism, IMM methods need to add multi-task layer for further adjustment after training all stages training, which leads to decreasing performance in the early stage (i.e., using SGD for training), and rising at the end (the last point is the results using fine-tuned IMM methods). From the bottom row, it reveals that at the transition of different stages, the performance of previous stages will not fall rapidly, which shows that IADM can prevent forgetting efficiently. Considering that the background of the examples in the first stage of the UCF101 dataset is very easy to be classified, thus the initial accuracy is very high.

5 CONCLUSION

In this paper, we investigated how to develop deep models for incremental learning. Indeed, there are two major concurrent challenges for building deep incremental models. First, if the entire training data are not available before learning the task, it is necessary to make the deep model structure scaling with streaming data for flexible model evolution and faster convergence. In other words, we need to develop deep incremental models with capacity scalability. Second, due to concept drift in streaming data, it is important to update the model while preserving previous knowledge for overcoming the catastrophic forgetting. Here, we aim to deal with these two challenges in one unified framework. Along this line, we developed an incremental adaptive deep model (IADM), which has a carefully designed attention model for the hidden layer and enables capacity scalability by learning deep models with adaptive depth from shallow to deep. Moreover, IADM has the ability in embedding the attention weights into fisher information matrix, which can incrementally match the the posterior distribution of the neural network trained on all stages and prevent the forgetting in consequence. Finally, experiments on numerous real-world data showed the effectiveness of IADM for incremental learning.

6 ACKNOWLEDGMENTS

This research was partially supported by the National Key R&D Program of China (2018YFB1004300), NSFC (61773198, 61632004) and NSF IIS-1814510.

REFERENCES

[1] Shunichi Amari. 1998. Natural Gradient Works Efficiently in Learning. *Neural Computation* 10, 2 (1998), 251–276.

[2] Alina Beygelzimer, Satyen Kale, and Haipeng Luo. 2015. Optimal and Adaptive Algorithms for Online Boosting. In *ICDM*. Lille, France, 2323–2331.

[3] Albert Bifet and Ricard Gavalda. 2007. Learning from Time-Changing Data with Adaptive Windowing. In *ICDM*. Minneapolis, Minnesota, 443–448.

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. *CoRR* abs/1801.10112 (2018).

[5] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, and Hai-Hong Tang. 2018. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *KDD*. London, UK, 1187–1196.

[6] Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. 2015. Net2Net: Accelerating Learning via Knowledge Transfer. *CoRR* abs/1511.05641 (2015).

[7] Xiaowei Chen and John C. S. Lui. 2016. Mining Graphlet Counts in Online Social Networks. In *ICDM*. Barcelona, Spain, 71–80.

[8] Robert Coop, Mishtal, and Itamar Arel. 2013. Ensemble Learning in Fixed Expansion Layer Networks for Mitigating Catastrophic Forgetting. *Trans. on Neural Netw. and Learning Syst.* 24, 10 (2013), 1623–1634.

[9] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *ICML*. Corvallis, Oregon, 193–200.

[10] Wei Fan. 2004. Systematic data selection to mine concept-drifting data streams. In *KDD*. Seattle, Washington, 128–137.

[11] Ji Feng, Yang Yu, and Zhi-Hua Zhou. 2018. Multi-Layered Gradient Boosting Decision Trees. *CoRR* abs/1806.00007 (2018).

[12] Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Computer Survey* 46, 4 (2014), 44:1–44:37.

[13] Alexander Gepperth and Cem Karaoguz. 2016. A Bio-Inspired Incremental Learning Architecture for Applied Perceptual Problems. *Cognitive Computation* 8, 5 (2016), 924–934.

[14] Heitor Murilo Gomes, Jean Paul Barddal, Fabricio Enembreck, and Albert Bifet. 2017. A Survey on Ensemble Learning for Data Stream Classification. *ACM Computer Survey* 50, 2 (2017), 23:1–23:36.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. Las Vegas, NV, 770–778.

[16] Steven C. H. Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. 2013. Online Multiple Kernel Classification. *ML* 90, 2 (2013), 289–316.

[17] Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2018. Online Learning: A Comprehensive Survey. *CoRR* abs/1802.02871 (2018).

[18] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L. Hayes, and Christopher Kanan. 2018. Measuring Catastrophic Forgetting in Neural Networks. In *AAAI*. New Orleans, Louisiana, 3390–3398.

[19] 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. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR* abs/1612.00796 (2016).

[20] Ralf Klinkenberg. 2004. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis* 8, 3 (2004), 281–300.

[21] Ralf Klinkenberg and Thorsten Joachims. 2000. Detecting Concept Drift with Support Vector Machines. In *ICML*. Stanford, CA, 487–494.

[22] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report.

[23] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. of the IEEE* 86, 11 (1998), 2278–2324.

[24] Jeongtae Lee, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. 2017. Lifelong Learning with Dynamically Expandable Networks. *CoRR* abs/1708.01547 (2017).

[25] John M Lee. 2006. *Riemannian manifolds: an introduction to curvature*. Vol. 176. Springer Science Business Media.

[26] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Overcoming Catastrophic Forgetting by Incremental Moment Matching. In *NIPS*. Long Beach, CA, 4655–4665.

[27] Sang-Woo Lee, Chung-yeon Lee, Dong-Hyun Kwak, Jiwon Kim, Jeonghee Kim, and Byoung-Tak Zhang. 2016. Dual-Memory Deep Learning Architectures for Lifelong Learning of Everyday Human Behaviors. In *IJCAI*. New York, NY, 1669–1675.

[28] Razvan Pascanu and Yoshua Bengio. 2013. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584* (2013).

[29] Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychol. Review* 97, 2 (1990), 285.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*. Munich, Germany, 234–241.

[31] Doyen Sahoo, Quang Pham, Jing Lu, and Steven C. H. Hoi. 2018. Online Deep Learning: Learning Deep Neural Networks on the Fly. In *IJCAI*. Stockholm, Sweden, 2660–2666.

[32] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012).

[33] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. 2017. The Simpler The Better: A Unified Approach to Predicting Original Taxi Demands based on Large-Scale Online Platforms. In *KDD*. Halifax, Canada, 1653–1662.

[34] Ramanarayanan Viswanathan. 1993. A note on distributed estimation and sufficiency. *TIT* 39, 5 (1993), 1765–1767.

[35] Yang Yang, Yi-Feng Wu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. 2018. Complex Object Classification: A Multi-Modal Multi-Instance Multi-Label Deep Network with Optimal Transport. In *KDD*. London, UK, 2594–2603.

[36] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. 2017. TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams. In *KDD*. Halifax, NS, 595–604.

[37] Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-Hua Zhou. 2016. Online Stochastic Linear Optimization under One-bit Feedback. In *ICML*. New York City, NY, 392–401.

[38] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. 2012. Online Incremental Feature Learning with Denoising Autoencoders. In *AISTATS*. La Palma, Canary Islands, 1453–1461.

[39] Yue Zhu, Kai Ming Ting, and Zhi-Hua Zhou. 2017. New Class Adaptation Via Instance Generation in One-Pass Class Incremental Learning. In *ICDM*. New Orleans, LA, 1207–1212.