Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Survey Track

Recent Advances in Concept Drift Adaptation Methods for Deep Learning

Liheng Yuan , Heng Li , Beihao Xia , Cuiying Gao , Mingyue Liu , Wei Yuan* , Xinge You
Huazhong University of Science and Technology
{ylh,liheng,xbh_hust,gaocy,lmy_hust,youxg}@hust.edu.cn,yuanwei@mail.hust.edu.cn

Abstract

In the "Big Data" age, the amount and distribution of data have increased wildly and changed over time in various time-series-based tasks, e.g weather prediction, network intrusion detection. However, deep learning models may become outdated facing variable input data distribution, which is called **concept drift**. To address this problem, large number of samples are usually required to update deep learning models, which is impractical in many realistic applications. This challenge drives researchers to explore the effective ways to adapt deep learning models to concept drift. In this paper, we first mathematically describe the categories of concept drift including *abrupt drift*, *gradual drift*, *recurrent drift*, *incremental drift*. We then divide existing studies into two categories (i.e., *model parameter updating* and *model structure updating*), and analyze the pros and cons of representative methods in each category. Finally, we evaluate the performance of these methods, and point out the future directions of concept drift adaptation for deep learning.

1 Introduction

In many real-world tasks, data arrive continuously in a stream manner, such as weather prediction [Elwell and Polikar, 2011], call record analysis [Charles *et al.*, 1977], network intrusion detection [Andresini *et al.*, 2021] and real-time stock market trading data analysis [Hu *et al.*, 2015]. The distribution of these data often changes over time. For instance, the weather prediction rules may vary radically depending on seasons. The characteristics of spam may occasionally change, leading to the redefinition of spam. The above phenomenon is called concept drift. In [Ghomeshi *et al.*, 2019], concept drift is divided into four categories, i.e., abrupt drift, gradual drift, incremental drift and recurrent drift. As shown in Fig. 1, classification performance will deteriorate when concept drift occurs, since classification models usually cannot spontaneously adapt to the new data distribution. To address this problem, a series of concept drift adaptation methods have been proposed to update models online to handle four different concept drift situations.

There have been some good reviews of concept drift adaptation methods [Lu *et al.*, 2020] [Gopu and Godandapani, 2015] [Barros and Santos, 2018]. They mainly consider traditional machine learning models and seldom involve deep learning ones. Inspired by this, we study the recent advances in concept drift adaptation methods developed for deep learning models. In addition to the well-known *stability-plasticity dilemma*[1] [Ditzler and Polikar, 2013], deep learning oriented concept drift adaptation faces two main challenges: heavier sample shortage and higher latency of model updating. In fact, updating deep learning models requires lots of samples, which is difficult to satisfy in practice. Moreover, updating deep learning models usually consumes longer time and results in higher latency of concept drift adaptation [Jan *et al.*, 2020] [Kantchelian *et al.*, 2013] [Pendlebury *et al.*, 2018]. Up to now, a variety of concept drift adaptation methods have been developed through exploiting the complexity and the flexibility of deep learning models, which have attracted increasing attention from the research community.

In this survey, we thoroughly investigate the existing concept drift adaptation methods for deep learning. Our main contributions are summarized as follows:

(1) We analyze mathematically different concept drift situations, and divide the existing adaptation methods for deep learning into two main categories and four subcategories.

(2) We examine the representative methods in each subcategory, analyze their pros and cons, and compare their performance on widely used datasets under different concept drift.

(3) We propose the main challenges faced by existing studies, and point out the future directions of deep learning model oriented concept drift adaptation studies.

2 Background

2.1 The Definition of Concept Drift

Concept drift is a phenomenon in which the statistical characteristics of the target variable change over time [Baena-Garc *et al.*, 2006]. Suppose X is the feature space and y is the label. Then the emergence of concept drift between times t and

*Corresponding author: yuanwei@mail.hust.edu.cn

[1]Concept drift adaptation should balance the learning focus between the past information and the incoming distribution changes.

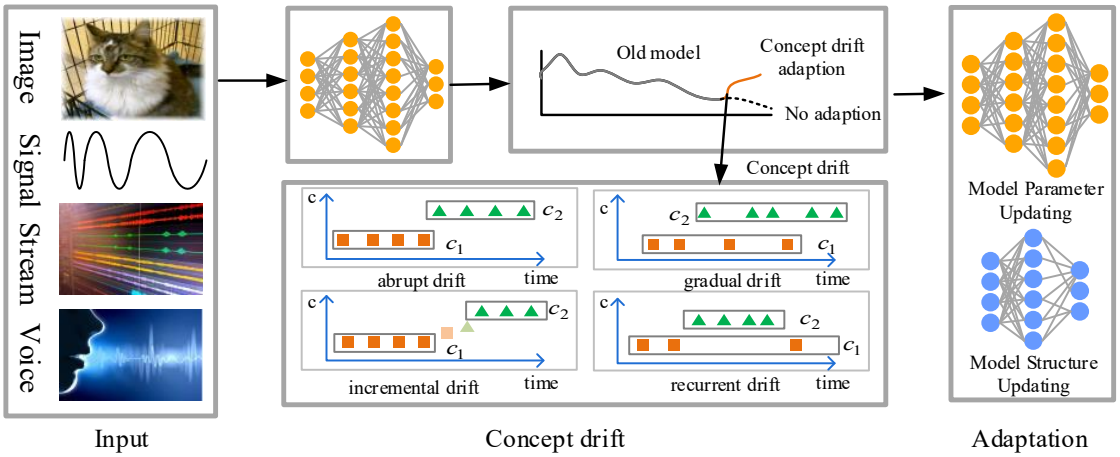

Figure 1: Overview of concept drift adaptation methods.

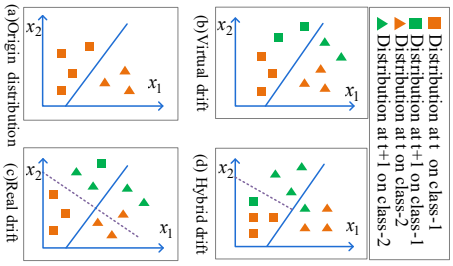

Figure 2: The sources of concept drift. (x_1, x_2) represents a two-dimensional feature space. Class-1 and class-2 denote two different classes. The solid line shows the original decision boundary, and the dotted describes the new one.

$t+1$ can be described by

$$\exists X : P_t(X, y) \neq P_{t+1}(X, y), \tag{1}$$

where $P_t(X, y)$ and $P_{t+1}(X, y)$ represent the data distribution at time t and $t+1$, respectively.

2.2 The Sources of Concept Drift

Considering $P_t(X, y) = P_t(X)P_t(y \mid X)$, we attribute concept drift to the following three reasons depicted in Fig. 2.

(1) $P_t(X)$ changes and $P_t(y \mid X)$ remains unchanged. As described in Fig. 2(b), the feature space is accordingly changed, but the decision boundary is not impacted. This is known as virtual drift.

(2) $P_t(X)$ remains unchanged and $P_t(y \mid X)$ changes. As shown in Fig. 2(c), the decision boundary also changes accordingly. This is called real drift. Since we have

$$P_t(y \mid X) = \frac{P_t(X \mid y)P_t(y)}{P_t(X)}, \tag{2}$$

the probabilistic sources of drift can be attributed to: (1) label prior drift $P_t(y) \neq P_{t+\triangle t}(y)$, and (2) likelihood drift $P_t(X|y) \neq P_{t+\triangle t}(X|y)$.

(3) Both $P_t(X)$ and $P_t(y \mid X)$ changes over time. As depicted in Fig. 2(d), virtual and real concept drift exist simultaneously in this case. In practice, this case is more common than the previous ones.

2.3 The Categories of Concept Drift

Concept drift fall into four categories: abrupt, gradual, incremental and recurrent, which are depicted in Fig. 1.

• Abrupt drift. It indicates data distribution changes at a precise point in time, which is defined by the inequality below. For example, some people's hobbies may significantly change suddenly for some reason.

$$\exists X : P_t(X, y) \neq P_{t+\triangle t}(X, y), \quad \triangle t < \delta, \tag{3}$$

where δ is a threshold defined.

• Incremental drift. It indicates data distribution changes from one to another over a period of time, characterized by the following inequalities. It includes multiple sources, but the difference among sources is very small. Furthermore, incremental drift is not necessarily monotonically increasing.

$$\begin{aligned} &\exists X : P_t(X, y) \neq P_{t+\triangle t}(X, y), \\ &P_t(X, y) < P_m(X, y) < P_{t+\triangle t}(X, y) \quad t < m < t + \triangle t. \end{aligned} \tag{4}$$

• Gradual drift. As defined by the inequalities below, gradual drift means both source of samples are active at some point. As time goes by, the probability of sampling from the source c_1 becomes lower and sampling from c_2 source becomes higher. For example, the function of a device fails occasionally, until a new failure mode completely takes over.

$$\begin{aligned} &\exists X : P_t(X, y) \neq P_{t+\triangle t}(X, y), \\ &P_m(X, y) = \alpha(t)P_t(X, y) + (1 - \alpha(t))P_{t+\triangle t}(X, y) \\ &t < m < t + \triangle t. \end{aligned} \tag{5}$$

where $\alpha(t) = \beta(c(t))$ holds. Here β denotes the Bernoulli distribution. $\alpha(t)$ has a probability $c(t)$ of 0 and $1 - c(t)$ of 1. $c(t)$ represents an incremental drift. $P_{t+\triangle t}(X, y)$ will become dominant over time.

• Recurrent drift. It reveals that the data distribution reverts back to the original distribution over time, as shown in the following formula. For instance, some rare weather phenomena, like hurricanes, can occur repeatedly in certain places.

$$\begin{aligned} &\exists X : P_t(X, y) \neq P_{t+\triangle t}(X, y), \\ &\forall X : P_t(X, y) = P_{t+\triangle m}(X, y) \quad \triangle m > \triangle t. \end{aligned} \tag{6}$$

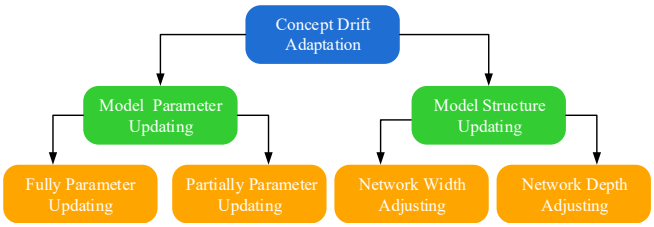

Figure 3: Basic classification framework.

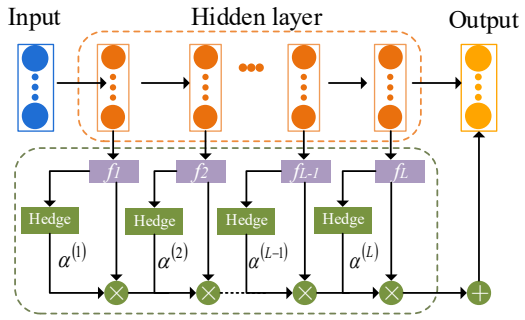

Figure 4: Hedge Back propagation.

3 Methods

As shown in Fig. 3, we divide the existing concept drift adaptation methods for deep learning into two categories, i.e., *model parameter updating* and *model structure updating*.

3.1 Model Parameter Updating

The model parameter updating methods preserve network structure but update the parameters, i.e., weights of neural network, to adapt to concept drift.

Fully Parameter Updating

Fully parameter updating means that all parameters are updated through training model with new data. [Soleymani and Paquet, 2020] proposes a pre-trained Convolutional Neural Networks (CNN) model through initial training over historical data, and then updates model parameters (i.e., weights) with an online learning scheme for the purpose of concept drift adaptation. In [Ryan *et al.*, 2019], model parameters are fully updated with new data and a new meta learning method is introduced to accelerate model retraining.

The authors of [Saadallah and Morik, 2021] leverage a meta learning method to learn the weights of their ensemble model to adapt to concept drift. Adadelta optimizer-based deep neural networks (ADODNN) proposed in [Priya and Uthra, 2021] uses Adadelta optimizer to update network weights, and optimizes hyper parameters to further improve classification performance.

DAREM [Darem *et al.*, 2021] employs an incremental learning method to update model parameters for concept drift adaptation. Since its training data contains both new and old data, this method alleviates the catastrophic forgetting problem. DAREM can be used to handle gradual drift and incremental drift. However, it suffers a slow convergence rate and cannot tackle abrupt drift.

In [Sahoo *et al.*, 2018], the authors focus on DNNs and propose Hedge Back propagation (HBP) to counter concept drift. As described in Fig. 4, this method connects hidden layers to the output layer, each of which represents different concept at different level. It assigns a weight to every hidden layer, and its prediction function can be expressed as

$$F(x) = \sum_{l=0}^{L} \alpha^{(l)} f^{(l)}, \tag{7}$$

where $\alpha^{(l)}$ is the weight of the l-th layer, f is the output of the l-th layer, and L is the number of hidden layers. Weights can be updated with back propagation to adapt to concept drift.

Furthermore, Recurrent Neural Networks (RNNs) are also studied by existing concept drift adaptation methods. Causal ConvLSTM [Yen *et al.*, 2019] uses BPTT (Back-Propagation Through Time) to update model parameters with recent samples. In [Fekri *et al.*, 2020], the tree-structured Parzen Estimator (TPE) is used to optimize hyper parameters, and the weights of a RNN are then updated for concept drift adaptation. This method neither retains historical data nor requires retraining.

Partially Parameter Updating

Many neural networks adapt to concept drift by using the BPTT method on the latest samples, so that their parameters can be redefined on the latest samples. Unfortunately, these methods may lead to catastrophic forgetting, in which new learning is followed by almost complete forgetting of what was previously learned. A common solution is to update only a part of network parameters to accommodate concept drift, which is called partially parameter updating in this survey.

Selective Ensemble-Based Online Adaptive Deep Neural Network (SEOA) [Guo *et al.*, 2021] selects shallow classifiers from frozen base classifiers and adds them into the ensemble model to adapt to concept drift. When the data fluctuation ratio is small, the base classifier whose weight is lower than a specified threshold is frozen.

[Yang *et al.*, 2019] and [Kirkpatrick *et al.*, 2016] employ the fisher information matrix to prevent the weight of the new task from moving away from the old one. In [Yang *et al.*, 2019], the authors exploit the attention based fisher information matrix to overcome the catastrophic forgetting problem.

In [Disabato and Roveri, 2019], the authors study learning CNNs under concept drift. They take the two-layer hypothesis test [Yu *et al.*, 2019] to find the layer where concept drift occurs first, and then updates the subsequent layers for concept drift adaptation. To overcome the limited data challenge, [Diez-Olivan *et al.*, 2021] generates synthetic data via a kernel density estimation method. And these data can be used to fine-tune the last layer, which helps quickly adapting to concept drift.

It is worth noting that model training is accelerated when only a part of model parameters are updated. Hence the above methods can deal with the abrupt drift.

3.2 Model Structure Updating

The model structure updating methods adjust network structure (e.g., network width or depth) to adapt to concept drift.

Network Width Adjusting

The network width adjusting methods modify network width to accommodate concept drift. The original part of their network remains unchanged, and their model can be incrementally learned by adding new branches or units for the purpose of concept drift adaptation. Here we introduce two key methods for network width adjusting.

Adding Branches. Progressive Neural Networks (PNN) [Rusu *et al.*, 2016] has been proposed as a new network structure for concept drift adaptation. [Budiman *et al.*, 2016] designs an Adaptive Convolutional ELM method (ACNNELM) which enhances CNNs with a hybrid Extreme Learning Machine (ELM) model. Unfortunately, the number of parameters increases sharply with the number of tasks (i.e., concept drift adaptation), and the design of different tasks requires prior knowledge. The highly complex network structure will lead to slow convergence of model training, so they cannot well adapt to concept drift.

To solve the above problem, [Kauschke *et al.*, 2019] constructs a discriminative classifier to identify the area where the classifier is incorrectly classified, and then trains a new classifier (called patch network) on the misclassified data. The patch network takes advantage of the middle layers of the original neural network to extract features and representations that may be critical to classification.

Adding Units. This kind of methods adjusts neural networks by adding some hidden units. Dynamically Expandable Networks (DEN) [Lee *et al.*, 2017] selectively retrains old tasks and adapts to new tasks by adding new hidden units. However, its network complexity increases with the number of learning tasks. To overcome this challenge, Deep Evolving Denoising Autoencoder (DEVDAN) [Ashfahani *et al.*, 2020] chooses to explore the flexible network structure, whose hidden units can be inserted or deleted for the purpose of concept drift adaptation. The authors of DEVDAN point out that Mean Square Error (MSE) has some problems for concept drift adaptation: (1) it calls for all data to understand the reconstruction capability, and (2) it cannot examine the reconstruction power on unseen samples. Therefor, DEVDAN exploits the *NS* formula to evaluate the generalization power of network structure under the assumption on normal distribution:

$$NS = \int_{-\infty}^{\infty} (X - z)^2 P(\tilde{X}) d\widetilde{X}, \tag{8}$$

where $P(\widetilde{X})$ is the probability density function, X and z are the input and reconstruct, respectively. It can be turned into a bias-variance formula:

$$NS = Bias(z)^2 + Var(z), \tag{9}$$

where z is a random variable, $Bias(z)^2$ and the $Var(z)$ can be expressed as $(X - E[z])^2$ and $E\left[z^2\right] - E[z]^2$, respectively.

In practice, the term $Bias(z)^2$ keeps decreasing with the increase of training sample number. DEVDAN alarms the occurrence of concept drift, once a rise of $Bias^2$ is captured. At this moment, DEVDAN can adapt to concept drift through adding new hidden units or deleting old hidden units. However, DEVDAN relies on the strong assumption that the incoming data follows normal distribution. To relax this assumption, [Pratama *et al.*, 2019b] and [Pratama *et al.*, 2019a] exploit the Autonomous Gaussian Mixture Model (AGMM) to capture concept drift.

Network Depth Adjusting

It is shown in [Eldan and Shamir, 2015] that increasing network depth better enhances the generalization performance of neural networks than increasing network width. Therefore, the network depth adjusting methods usually can better adapt to concept drift. Fast Hoeffding Drift Detection Method (FHDDM) [Pratama *et al.*, 2019c] builds a deep stack network with the feature enhancement method, and adds a hidden layer for concept drift adaptation. The hidden layer is built on the new incoming data between warnings and determinations on the occurrence of concept drift. Each hidden layer has an output that is weighted to get the final label. What's more, FHDDM sets a decaying factor ρ_d to adjust the weight of each layer:

$$\rho_d = \rho_d \pm \pi, \tag{10}$$

where π is the step size. When ρ_d is low, FHDDM can adapt to the gradual or incremental drift, but result in slow adaptation to abrupt drift. If the hidden layer makes a wrong prediction, a penalty is imposed as follows:

$$X_d = X_d * \pi, \tag{11}$$

where X_d is the prediction of the d-th layer.

Another popular method is Autonomous Deep Learning (ADL) [Ashfahani and Pratama, 2018]. This method combines network width and network depth adaptation. Moreover, ADL utilizes the local error derivative back propagated from the layer most related to the current concept. Thus ADL can adapt to abrupt drift. However, ADL assigns a softmax layer to every layer of neural network, making it limited to classification applications.

Network with Dynamically Evolved Capacity (NADINE) [Pratama *et al.*, 2019d] is proposed to handle concept drift in the non-classification applications. NADINE constructs a Multilayer Perceptron (MLP) model whose network depth can be adjusted to adapt to concept drift. Moreover, NADINE freezes the parameters of some layers and updates the other layers with Stochastic Gradient Descent (SGD), in order to avoid the catastrophic forgetting. Its learning rate can be adjusted adaptively, and the value of its learning rate is related to the correlation between the current concept and the hidden layers. However, the error back propagation used in this method has a major drawback, i.e., the slow convergence of model training.

To accelerate convergence, Stacked Auto Encoder-Deep Neural Network (SAE-DNN) [Zhang *et al.*, 2020] expands the neural network by Random Vector Functional Link (RVFL) structure, and the parameters of the expanded layers are dynamically assigned by the new incoming data with the group lasso regularization and the L2 regularization.

Dataset	Type	Name	#Fea.	#Class	#Sam.
Syn.	abrupt	SEA	3	2	100k
		LED abrupt	24	10	100k
		Tree	10	6	100k
		R.MNIST	784	10	70k
	gradual	LED gradual	24	10	100k
	incremental	Hyperplane	10	2	120k
	recurrent	SEA	3	2	100k
	mixed	P.MNIST	784	10	65k
Real	mixed	Weather	8	2	18k
		CoverType	54	7	581k
		Elec	5	2	45.3k
		KDDCUP	41	2	500k

Table 1: Properties of datasets. "Syn.", "Fea.", "Sam." are short for synthetic, features and sample, respectively. "mixed" means there are several kinds of concept drift in the corresponding datasets.

4 Datasets and Evaluation

4.1 Available Datasets

The datasets used in concept drift studies include the synthetic dataset and the real dataset[2]. Representative datasets are given in Table 1, which gives the basic properties about drift types, features, category number and sample amount.

• SEA [Ashfahani and Pratama, 2018]: This dataset contains the samples of two classes, and every sample has three features (or attributes). The relationship between the first two features can be characterized by a binary classification, described as $f_1 + f_2 < \theta$. In this problem, f_1 and f_2 are two relevant features, and θ is the threshold value for distinguishing these two classes. Through changing θ from 4 to 7 and then back to 4, abrupt drift and recurrent drift can be implemented.

• Hyperplane [Dong *et al.*, 2015]: This dataset contains the samples of two classes, and every sample has 10 features. The decision boundary is given by a hyperplane $\sum_{j=1}^{d} w_j x_j = w_o$. The position and orientation of that hyperplane can be changed by continuously conducting $w_o = w_o + \epsilon$, where ϵ is a small number. In this way, incremental drift can take place.

• Rotated MNIST [Lopez-Paz and Ranzato, 2017]: This dataset is built by rotating the handwritten digits in the traditional MNIST dataset to arbitrary angles from $-\pi$ to π. Accordingly, abrupt drift can be introduced into this dataset.

• Permuted MNIST [Ashfahani *et al.*, 2020]: Similar to Rotated MNIST, this dataset is also generated from MNIST. More specifically, every sample in this dataset is obtained by permuting some pixels in an image from MNIST. In this way, abrupt drift and recurrent drift occur in this dataset.

• LED [Guo *et al.*, 2021]: This dataset contains the data used to predict the numbers on the seven-segment LED display. It contains 24 attributes and 17 of them are irrelevant. It can generate gradual and abrupt concept drift dataset which called LED gradual and LED abrupt, respectively.

• Tree [Brzezinski and Stefanowski, 2014]: The Tree dataset is a fast changing dataset, whose data are generated from a random tree generator. This dataset contains 15 abrupt drift over 100000 instances.

The above datasets are all constructed with synthetic data. There exist some datasets for concept drift studies containing real data, e.g., Weather [Elwell and Polikar, 2011], Elec [Wang *et al.*, 2020] and KDDCUP [Yang *et al.*, 2019]. Weather is a real dataset which contains the real data used for weather prediction. Every sample in Weather has 8 features. Elec contains the real data that are collected to predict the electricity price of Australian New South Wales Electricity Market. KDDCUP is composed of the real data that can be used to predict network intrusion events. CoverType [Gama *et al.*, 2003] contains 54 features and 7 classes. It collects the US Forest Service (USFS) Region 2 Resource Information System (RIS) data. These datasets often contain two or more types of concept drift, e.g., abrupt and recurrent drift.

4.2 The Evaluation Metric

The studies on concept drift mainly use the following metrics to evaluate their methods: 1) classification accuracy, 2) execution time (ET), 3) the number of parameters (NoP), 4) the number of hidden layers per time step (NoHL) and 5) the number of hidden nodes per time step (NoHN).

4.3 Performance Analysis

We summarize the performance of representative concept drift adaptation methods in Tables 2 and 3 [3], respectively. For fair comparison, we only consider the methods evaluated on the same datasets in existing literature. We aim to analyze the influencing factors for different types of concept drift.

The partially parameter updating methods can mitigate the problems of limited data and catastrophic forgetting. They are able to adapt to abrupt drift owing to their fast convergence of model training. However, when the new data distribution is remarkably different from the old one, their adaptability will be limited. For instance, the performance of partially parameter updating methods is usually not very good on KDDCUP dataset.

Although the adding branch based methods can adapt to concept drift, their convergence is slow due to their complex network structure. So they cannot handle both gradual and abrupt drift. As shown in Table 3, the ET of these methods (e.g.,PNN) is long. This accounts for why some experimental results of ET (denoted by "-" in Table 3) cannot be obtained after running the corresponding methods for days due to the computational constraint. What's more, the number of their parameters grows rapidly with the increase of the added branches, making them difficult to work in the rapidly changing environment. The adding unit based methods in Table 3 adjust network structure based on NS formula. Therefore, they can adapt to the mixed concept drift (e.g., the concept drift in P. MNIST).

The network depth adjusting methods can deal with the abrupt and recurrent drift through adapting their network depth. Moreover, the flexibility of their network structure makes it possible to cope with gradual drift. As shown in Table 3, the ET of these methods is shorter than that of the

[2]https://github.com/vlosing/driftDatasets/tree/master

[3]All experimental results are from the original references.

Category	Method	SEA	Hyperplane	P.MNIST	R.MNIST	Weather	KDDCUP
Parameter Updating	HBP[Sahoo *et al.*, 2018]	80.56	89.69	-	-	81.39	98.23
	SEOA[Guo *et al.*, 2021]	80.68	89.77	-	-	**88.55**	96.54
Adding Branches	PNN [Rusu *et al.*, 2016]	83.20	85.55	64.42	60.94	68.46	99.00
Adding Units	DEVDAN [Ashfahani *et al.*, 2020]	91.12	91.19	76.67	**76.48**	70.75	99.83
	ATL[Pratama *et al.*, 2019b]	91.12	91.40	-	-	71.53	99.52
	Parsnet [Pratama *et al.*, 2019a]	91.41	92.28	**83.91**	64.32	72.58	-
Depth Adjusting	DEVFNN [Pratama *et al.*, 2019c]	91.70	91.57	-	-	80.00	99.00
	ADL [Ashfahani and Pratama, 2018]	92.13	**92.33**	68.40	72.90	74.48	**99.84**
	NADINE [Pratama *et al.*, 2019d]	**92.24**	-	77.65	74.51	-	**99.84**

Table 2: Classification accuracy of partial classical methods (higher is better).

Dataset	Method	ET(s)	NoHL	NoHN	NoP
SEA	DEN	-	1	6	38
	PNN	-	3	33	347
	ADL	14	1	11.42	71.18
	NADINE	15	1.19	12.91	114
R. MNIST	DEN	-	3	440	290K
	PNN	-	3	750	503K
	ADL	199	1.9	8.73	7.4K
	NADINE	192	1	15.69	12.67K
P. MNIST	DEN	-	2	440	290K
	PNN	-	3	705	503K
	ADL	212	1.39	19.81	16.2K
	NADINE	202	1.19	22.33	12.96K
KDDCUP	DEN	-	1	20	860
	PNN	-	3	375	41.9K
	ADL	115	1	12.68	561.36
	NADINE	98	1	12.33	545.41

Table 3: ET, NoP, NoHL, and NoHN of partial classical methods (lower is better).

adding branch based methods. However, it is difficult for these methods to work in the scenarios where multiple types of concept drift exist simultaneously. For instance, [Pratama *et al.*, 2019c] and [Ashfahani and Pratama, 2018] control the process of weight updating with a decaying factor. When the decaying factor is small, it can adapt to gradual drift, but cannot adapt to abrupt drift. This makes them perform poorly on the real-data datasets (e.g.,Weather) .

5 Discussion

5.1 Major Challenges

The research of deep learning model oriented concept drift adaptation is still in its early stage and is faced with the following challenges:

(1) When concept drift occurs, the number of samples with new distribution is limited. However, deep learning requires a large amount of data for model training. Accordingly, deep learning models are unable to rapidly converge and quickly adapt to the new distribution. Under this situation, the performance of deep learning models will inevitably deteriorate.

(2) Existing studies on concept drift adaptation usually deal with only one type of concept drift. However, various concept drift often occur simultaneously in practice. How to use a method to tackle different types of concept drift simultaneously is still an open issue.

5.2 Future Directions

• *Relying on fewer samples with new distribution*. As mentioned above, a main challenge in concept drift adaptation is the shortage of the data with new distribution. Few-shot learning provides a promising solution to this problem. Few-shot learning is a machine learning method where its training dataset contains limited information. Few-shot learning attempts to build accurate models with less training data. Hence, introducing few-shot learning may help quickly adapting to new data distribution, especially when training data are insufficient.

• *Integrating lightweight classifiers to counter concept drift*. Ensemble algorithms can reduce the deviation and variance of neural networks. Moreover, they can prevent overfitting by combining multiple individual models. In general, the prediction ability of ensemble classifier is better than that of single classifier. Applying ensemble learning to concept drift adaptation seems to be a promising and interesting direction. A simple way is to adjust the weights of base classifiers to react to new data distribution. One can add new base classifiers trained over new data into the ensemble model, then remove or decrease the weights of outdated base classifiers. In this situation, how to choose lightweight and diverse base classifiers is of great importance. More efforts should be devoted to reducing the computational complexity of this method.

6 Conclusion

How to handle concept drift is an important research topic in deep learning field. In this article, we review the general methods of concept drift adaptation for deep learning models. We summarize the sources of concept drift, analyze the deep learning oriented concept drift adaptation methods, and point out the future research directions. We hope this survey can provide researches with the state art of the knowledge of concept drift in deep learning and provide guidelines for the future.

Acknowledgments

This work was supported partially by the National Natural Science Foundation of China (Grant No. 62172177 and 61571205), and in part by the Fundamental Research Funds for the Central Universities (Grant No. 2021JYCXJJ037).

References

[Andresini *et al.*, 2021] Giuseppina Andresini, Annalisa Appice, Corrado Loglisci, Vincenzo Belvedere, Domenico Redavid, and Donato Malerba. A network intrusion detection system for concept drifting network traffic data. In Carlos Soares and Luis Torgo, editors, *Discovery Science*, pages 111–121, Cham, 2021. Springer International Publishing.

[Ashfahani and Pratama, 2018] Andri Ashfahani and Mahardhika Pratama. Autonomous deep learning: Continual learning approach for dynamic environments. In *Siam International Conference on Data Mining*, 2018.

[Ashfahani *et al.*, 2020] Andri Ashfahani, Mahardhika Pratama, Edwin Lughofer, and Yew-Soon Ong. Devdan: Deep evolving denoising autoencoder. *Neurocomputing*, 390:297–314, 2020.

[Baena-Garc *et al.*, 2006] Manuel Baena-Garc, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, and Rafael Morales-Bueno. Early drift detection method. *International Workshop on Knowledge Discovery from Data Streams*, 2006.

[Barros and Santos, 2018] RSMD Barros and Silas Garrido Teixeira De Carvalho Santos. A large-scale comparison of concept drift detectors. *Information Sciences*, 451-452(C), 2018.

[Brzezinski and Stefanowski, 2014] Dariusz Brzezinski and Jerzy Stefanowski. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1):81–94, 2014.

[Budiman *et al.*, 2016] Arif Budiman, Mohamad Ivan Fanany, and Chan Basaruddin. Adaptive convolutional ELM for concept drift handling in online stream data. *CoRR*, abs/1610.02348, 2016.

[Charles *et al.*, 1977] Charles, Swithinbank, Paul, McClain, Patricia, and Little. Drift tracks of antarctic icebergs. *Polar Record*, 18(116):495–501, 1977.

[Darem *et al.*, 2021] Abdulbasit A. Darem, Fuad A. Ghaleb, Asma A. Al-Hashmi, Jemal H. Abawajy, Sultan M. Alanazi, and Afrah Y. Al-Rezami. An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning. *IEEE Access*, 9:97180–97196, 2021.

[Diez-Olivan *et al.*, 2021] Alberto Diez-Olivan, Patxi Ortego, Javier Del Ser, Itziar Landa-Torres, Diego Galar, David Camacho, and Basilio Sierra. Adaptive dendritic cell-deep learning approach for industrial prognosis under changing conditions. *IEEE Transactions on Industrial Informatics*, 17(11):7760–7770, 2021.

[Disabato and Roveri, 2019] Simone Disabato and Manuel Roveri. Learning convolutional neural networks in presence of concept drift. In *2019 International Joint Conference on Neural Networks*, pages 1–8, 2019.

[Ditzler and Polikar, 2013] Gregory Ditzler and Robi Polikar. Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2283–2301, 2013.

[Dong *et al.*, 2015] Fan Dong, Jie Lu, Guangquan Zhang, and Kan Li. A modified learn++.nse algorithm for dealing with concept drift. In *International Flins Conference*, 2015.

[Eldan and Shamir, 2015] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *Conference on Learning Theory*, 12 2015.

[Elwell and Polikar, 2011] Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011.

[Fekri *et al.*, 2020] Mohammad Fekri, Harsh Patel, Katarina Grolinger, and Vinay Sharma. Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Applied Energy*, 282(3), 2020.

[Gama *et al.*, 2003] João Gama, Ricardo Rocha, and Pedro Medas. Accurate decision trees for mining high-speed data streams. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, 2003.

[Ghomeshi *et al.*, 2019] Hossein Ghomeshi, Mohamed Medhat Gaber, and Yevgeniya Kovalchuk. *Ensemble Dynamics in Non-stationary Data Stream Classification*, pages 123–153. Springer International Publishing, Cham, 2019.

[Gopu and Godandapani, 2015] Akila Gopu and Zayaraz Godandapani. A brief survey on concept drift. In Lakhmi C. Jain, Srikanta Patnaik, and Nikhil Ichalkaranje, editors, *Intelligent Computing, Communication and Devices*, pages 293–302, 2015.

[Guo *et al.*, 2021] Husheng Guo, Shuai Zhang, and Wenjian Wang. Selective ensemble-based online adaptive deep neural networks for streaming data with concept drift. *Neural Networks*, 2021.

[Hu *et al.*, 2015] Yong Hu, Xiangzhou Zhang, Bin Feng, Kang Xie, Mei Liu, and David Taniar. itrade: A mobile data-driven stock trading system with concept drift adaptation. *International Journal of Data Warehousing & Mining*, 11(1):66–83, 2015.

[Jan *et al.*, 2020] Steve Jan, Qingying Hao, Tianrui Hu, Jiameng Pu, Sonal Oswal, Gang Wang, and Bimal Viswanath. Throwing darts in the dark? detecting bots with limited data using neural data augmentation. In *2020 IEEE Symposium on Security and Privacy (SP)*, 2020.

[Kantchelian *et al.*, 2013] Alex Kantchelian, Sadia Afroz, Ling Huang, Aylin Caliskan Islam, Brad Miller, Michael Carl Tschantz, Rachel Greenstadt, Anthony Joseph Joseph, and Doug Tygar. Approaches to adversarial drift. In *Acm Workshop on Artificial Intelligence & Security*, pages 99–110, 2013.

[Kauschke *et al.*, 2019] Sebastian Kauschke, David H. Lehmann, and Johannes Fürnkranz. Patching deep

neural networks for nonstationary environments. In *2019 International Joint Conference on Neural Networks*, pages 1–8, 2019.

[Kirkpatrick *et al.*, 2016] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei 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. *Proceedings of the National Academy of Sciences*, 114, 12 2016.

[Lee *et al.*, 2017] Jeongtae Lee, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *CoRR*, abs/1708.01547, 2017.

[Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *neural information processing systems*, 2017.

[Lu *et al.*, 2020] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1, 2020.

[Pendlebury *et al.*, 2018] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. TESSERACT: eliminating experimental bias in malware classification across space and time. *CoRR*, abs/1807.07838, 2018.

[Pratama *et al.*, 2019a] Mahardhika Pratama, Andri Ashfahani, and Mohamad Abdul Hady. Weakly supervised deep learning approach in streaming environments. *2019 IEEE International Conference on Big Data*, pages 1195–1202, 2019.

[Pratama *et al.*, 2019b] Mahardhika Pratama, Marcus de Carvalho, Renchunzi Xie, Edwin Lughofer, and Jie Lu. Atl: Autonomous knowledge transfer from many streaming processes. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, page 269–278. Association for Computing Machinery, 2019.

[Pratama *et al.*, 2019c] Mahardhika Pratama, Witold Pedrycz, and Geoffrey Webb. An incremental construction of deep neuro fuzzy system for continual learning of non-stationary data streams. *IEEE Transactions on Fuzzy Systems*, PP, 08 2019.

[Pratama *et al.*, 2019d] Mahardhika Pratama, Choiru Za'in, Andri Ashfahani, Yew-Soon Ong, and Weiping Ding. Automatic construction of multi-layer perceptron network from streaming examples. *CoRR*, abs/1910.03437, 2019.

[Priya and Uthra, 2021] Swahini Priya and Annie Uthra. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex & Intelligent Systems*, 07 2021.

[Rusu *et al.*, 2016] Andrei Rusu, Neil Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. 06 2016.

[Ryan *et al.*, 2019] Sid Ryan, Roberto Corizzo, Iluju Kiringa, and Nathalie Japkowicz. Deep learning versus conventional learning in data streams with concept drifts. In *2019 18th IEEE International Conference On Machine Learning And Applications*, pages 1306–1313, 2019.

[Saadallah and Morik, 2021] Amal Saadallah and Katharina Morik. Online ensemble aggregation using deep reinforcement learning for time series forecasting. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics*, pages 1–8, 2021.

[Sahoo *et al.*, 2018] Doyen Sahoo, Quang Pham, Jing Lu, and Steven C. H. Hoi. Online deep learning: Learning deep neural networks on the fly. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence,*, pages 2660–2666. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[Soleymani and Paquet, 2020] Farzan Soleymani and Eric Paquet. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder - deepbreath. *Expert Systems with Applications*, 156:113456, 2020.

[Wang *et al.*, 2020] XueSong Wang, Qi Kang, MengChu Zhou, Le Pan, and Abdullah Abusorrah. Multiscale drift detection test to enable fast learning in nonstationary environments. *IEEE Transactions on Cybernetics*, PP(99):1–13, 2020.

[Yang *et al.*, 2019] Yang Yang, Da Wei Zhou, De Chuan Zhan, Hui Xiong, and Yuan Jiang. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In *the 25th ACM SIGKDD International Conference*, 2019.

[Yen *et al.*, 2019] Steven Yen, Melody Moh, and Teng-Sheng Moh. Causalconvlstm: Semi-supervised log anomaly detection through sequence modeling. In *2019 18th IEEE International Conference On Machine Learning And Applications*, pages 1334–1341, 2019.

[Yu *et al.*, 2019] Shujian Yu, Zubin Abraham, Heng Wang, Mohak Shah, Yantao Wei, and José C. Príncipe. Concept drift detection and adaptation with hierarchical hypothesis testing. *Journal of the Franklin Institute*, 356(5):3187–3215, 2019.

[Zhang *et al.*, 2020] Xu Zhang, Yuanyuan Zou, and Shaoyuan Li. Enhancing incremental deep learning for fccu end-point quality prediction. *Information Sciences*, 530, 2020.