CONTINUAL LEARNING FOR ANOMALY DETECTION WITH VARIATIONAL AUTOENCODER

Felix Wiewel and Bin Yang

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

ABSTRACT

Detecting anomalies using a variational autoencoder (VAE) suffers from catastrophic forgetting when trained on a continually growing set of normal data where only the most recently added data is available. Solving this problem would allow the use of the VAE for anomaly detection in settings where it is difficult or even impossible to retain all normal data at the same time. We propose an efficient extension of a method for continual learning which alleviates catastrophic forgetting for anomaly detection using a VAE. We show on some anomaly detection problems that the definition of normal data can be continually expanded without requiring all previously seen data.

***Index Terms*—** Continual Learning, Anomaly Detection, Variational Autoencoder, Generative Replay

1. INTRODUCTION

Neural networks achieve state-of-the-art performance and in some cases even surpass humans on machine learning tasks. Despite their success they lack an important ability compared to human learning, namely the ability to continually learn tasks even if only training examples of the most recent task are available [1, 2]. Currently neural networks suffer from a phenomenon called *catastrophic forgetting*, a rapid decrease in performance on previously learned tasks when trained on a new task [3]. Alleviating this problem is of great interest, for many current and future applications. Retraining a neural network with newly acquired data, for example, typically requires all previously used training data as well. This traditional method of retraining is hence limited by the available memory and computational resources. Continual learning, on the other hand, only uses the newly acquired data and can be more memory efficient and faster. Sometimes the training dataset consists of many small datasets which might not be available all at the same time. It might be illegal or even impossible to keep all datasets. An example for this is medical data, which has to be deleted after a certain amount of time, or data which can not be stored due to memory constraints. In these cases, continual learning is the only way to still train a neural network.

There exists previous work for anomaly detection using a VAE [4, 5, 6, 7]. But none of the proposed methods was shown to be useful in the setting of continual learning. On the other hand, there has been a great interest in solving the problem of catastrophic forgetting in other settings like supervised classification, reinforcement learning and generative models. Recent works include *elastic weight consolidation* (EWC) [8] where a regularization term, whose purpose is to protect the weights of a neural network that are most important for solving the previous tasks, is determined after every task and used for training of the next task. *Synaptic intelligence* [9] is a similar method which only differs from EWC in the way how the regularization term is determined. Other methods include *Learning without forgetting* [10] which uses the output of a neural network to data of a current task in order to preserve the responses of previous tasks. *Variational continual learning* [11] uses variational inference in combination with a coreset while *Dynamically Expandable Networks* expands the neural network, keeps the weights learned on previous tasks fixed and then combines similar neurons in order to avoid redundant computations. The inspiring work of Shin et al. [12] first proposed *generative replay* (GR) for supervised classification and is used as a base for our method.

In this paper, we propose a simple but effective extension of continual learning to the anomaly detection problem. The proposed method is based on the VAE. We utilize its capability to generate data, which is currently unused in anomaly detection, in order to enable continual learning. The proposed method is evaluated on the MNIST [13] and KDDCup99 [14] datasets. We further study a degeneration effect that can be observed when the capacity of the VAE is limited.

2. PROBLEM FORMULATION

Anomalies are patterns characterised by a noticeable deviation from so called normal data, where normal means compliance with some typical or expected features. The detection of anomalies requires the definition of a decision boundary which precisely separates normal and anormalous data in a suitable feature space. This poses several challenges [15]. First, since the training dataset is often limited, the desired

Fig. 1. Proposed method: The marked decoder of the VAE trained on the previous task is used to generate replayed samples from previous tasks. These are then mixed with the data of the current task and used for training of the VAE.

decision boundary can only be approximated. Thus samples close to the decision boundary can be wrongly classified. Second, the definition of normal data might change over time. This requires a change of the decision boundary over time as well. In addition, noise in the data can be confused with anomalies, which makes it difficult to distinguish between actual anomalies and noise.

In this work, we focus on temporal changes of the definition of normal data. Consider a system for anomaly detection which was trained on a particular set of normal data. This could be a system for the detection of anomalous traffic in a computer network. An anomaly could indicate an attack on the network, which requires an immediate action. As new types of normal traffic are introduced into the network, the considered system could wrongly classify them as anomalous. Incorporating this new normal data into the normal data would require a retraining and hence all previous normal data. With continual learning new data could be incorporated into a growing set of normal data without a complete retraining of the system and without the storage of all previous normal data.

3. ANOMALY DETECTION USING VARIATIONAL AUTOENCODER

The VAE is a generative model trained to approximate the data generating distribution $p(\mathbf{x})$ of an observed random vector $\mathbf{x}$ from a given dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. The VAE is a directed probabilistic model, which combines the concept of an autoencoder with the method of variational inference [16]. It is well suited for high dimensional data with highly nonlinear dependencies among its elements. Using the VAE for anomaly detection is a well known technique in the literature [4, 5, 6, 7]. Assuming i.i.d. samples, the marginal log-likelihood $\ln p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ of the dataset can be decomposed as a sum over the individual samples $\mathbf{x}_i$ as $\ln p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i=1}^{N} \ln p(\mathbf{x}_i)$, where each term can be lower bound by the so called *evidence lower bound* (ELBO)

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{1}$$

$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is given by an encoder network with the parameter vector $\boldsymbol{\phi}$, $p(\mathbf{z})$ is the prior distribution of $\mathbf{z}$ in the latent space and $\ln p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is the log-likelihood of the sample $\mathbf{x}$ given by a decoder network with the parameter vector $\boldsymbol{\theta}$. The encoder and decoder form the known structure of an autoencoder as illustrated in Fig. 1.

During training of the VAE, the negative ELBO is minimized, which is equivalent to maximizing the ELBO. By inspection of eq. (1), we can interpret the training process as twofold. By maximizing the expectation of the log-likelihood $\ln p(\mathbf{x}|\mathbf{z})$ w.r.t. $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ the encoder and decoder are trained to reconstruct the sample $\mathbf{x}_i$ as good as possible. Since the KL divergence D_{KL} is non-negative, a maximization of the ELBO forces the KL divergence to approach zero. This means, the distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ parameterized by the encoder approaches the prior distribution $p(\mathbf{z})$.

For the detection of anomalies, a threshold-based method is used. An anomaly score $AI(\mathbf{x})$ is defined and compared with a threshold γ. If $AI(\mathbf{x}) < \gamma$, the sample is considered to be an anomaly. Otherwise, it is a normal sample. While [4, 5, 6, 7] use the reconstruction probability $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x_i})}\left[\ln p_{\boldsymbol{\theta}}(\mathbf{x_i}|\mathbf{z})\right]$ as the anomaly score, we use the ELBO as the anomaly score because it gives slightly better results in our experiments.

4. PROPOSED METHOD

If the definition of normal samples in an anomaly detection application is expanding over time, the VAE has to be retrained in order to adapt to these changes. In this setting, incorporating a new dataset $\mathcal{D}^i$ into the existing set $\{\mathcal{D}^1, \ldots, \mathcal{D}^{i-1}\}$ of normal samples for training is defined as the i-th task. Training the VAE on a sequence of tasks with datasets $\mathcal{D}^1, \ldots, \mathcal{D}^i$ where only $\mathcal{D}^i$ is available on the i-th task leads to catastrophic forgetting of what was learnt on all $\mathcal{D}^j,\ j < i$. In order to prevent this, we use GR as proposed in [12]. GR uses a task solving network, the solver, and a generative model, the generator, for overcoming catastrophic forgetting. First the solver is trained to solve the first task based on $\mathcal{D}^1$ while the generator is trained to approximate its data generating distribution. On the following tasks, the training process is two-fold. The generator is used to reproduce the data of all previously solved tasks. The current solver is then used to infer the labels of this generated data. This data is combined with the currently available dataset

$\mathcal{D}^i$ to form an expanded training dataset, which contains not only the data of the current task but also generated data of all previously solved tasks. The ratio between the number of generated samples and the currently available samples determines the importance of the current task. Finally, the solver and the generator are trained on the expanded training dataset.

We propose to use a VAE as the generative model for replay in anomaly detection. This is a natural choice since the generative capabilities of the VAE are unused in this application. Our approach leads to an efficient system for continually expanding the definition of normal samples where the solver and the generator are implemented in one neural network. Instead of generating all previous data before the training, we use a copy of the decoder to generate data on a batch basis as shown in Fig. 1. This means, that for each batch of current data from $\mathcal{D}^i$ we generate a corresponding amount of replay data. In this way, we only need to store one additional set of weights for the decoder instead of all previous normal data.

Although generative adversarial networks (GAN) [17] are known to generate data with more details, e.g. sharper images, using them for generative replay for anomaly detection based on the VAE is unrewarding. First, training the VAE on the normal data and additionally training a GAN on the same data clearly is computationally more expensive. Second, using a VAE for generative replay leads to generated data, which can certainly be reconstructed by itself. While using a GAN for generative replay might lead to more detailed data, the VAE is unable to reconstruct all those details and hence these finer features are not considered when detecting anomalies. Generating more detailed images as those generated by the VAE is not necessary in this application. The more detailed data generated by a GAN might, however, help to mitigate the degeneration effect discussed in section 5.3. This is left open for future research.

5. RESULTS

We evaluate our method on two different datasets using a VAE with $400, 300, 200, 100$ densely connected hidden units in the encoder, a latent space with dimension 50 and a decoder symmetric to the encoder. For the prior $p(\mathbf{z})$ we choose a standard normal distribution. We use a sigmoid output layer for the decoder to predict the mean of a Bernoulli distribution in our signal model similar to [18]. For each dataset, we first define an anomaly class, which is selected from all available classes in the dataset. This anomaly class can either be one single class or a set of classes. All other classes of the dataset are considered to be normal classes. Then, starting with only one normal class, a sequence of tasks is defined, where each task expands the definition of normal data by one class. For training the VAE, only the normal data is used and all results are averaged over 10 runs. We report the area under the receiver operating curve (AUC) as a performance metric. The normal data is split into a training and an evaluation set while all anomalous data is only used during evaluation. We train the VAE for 25 epochs with a learning rate of 0.001 with the *Adam* optimizer and use 100 samples to evaluate the expectation in eq. (1). As the batch size we use 16384 for KDDCup99 and 128 for MNIST. We compare our generative replay method (GR) with another recently proposed method against catastrophic forgetting called *Elastic Weight Consolidation* (EWC), an upper bound (UB) given by a VAE trained with all data available all the time and a lower bound (LB) given by a VAE trained only on the most recent data.

Table 1. Attacks contained in KDDCup99 dataset and their corresponding types

Attack	Type	Attack	Type
back	DOS	bufferoverflow	U2R
ftpwrite	R2L	guesspasswd	R2L
imap	R2L	ipsweep	PROBE
land	DOS	loadmodule	U2R
multihop	R2L	neptune	DOS
nmap	PROBE	perl	U2R
phf	R2L	pod	DOS
portsweep	PROBE	rootkit	U2R
satan	PROBE	smurf	DOS
spy	R2L	teardrop	DOS
warezclient	R2L	warezmaster	R2L

5.1. KDD Cup 1999

The KDD Cup 1999 dataset [14] consists of approximately 4.9 million tcpdumps and was prepared by Stolfo et al. [19]. It was originally used in the KDD Cup competition for training and evaluating classifiers for intrusion detection in computer networks. Since the data consists of raw tcpdumps, it contains not only discrete values like the individual parts of an IP address but also categorical values like the transport and application protocols. In total this leads to 41 features for each sample. In order to train the VAE with such data, it has to be preprocessed. We first map each categorical feature to an integer starting from 0. After this we normalize all features to the interval $[0, 1]$. The resulting feature vectors can then be used to train and evaluate the VAE. We use approximately 10% of the data for evaluation and the rest for training. The KDDCup99 dataset contains 22 uniquely labeled attacks and one class considered normal. Although there are many different attacks, they can be grouped in four types as illustrated in Table 1. We define the anomaly class as all attacks of type DOS. For creating a sequence of tasks, we start with the normal class and in each task the definition of normal data is expanded by one attack selected from all types other than DOS. As can be seen in Fig. 2, our method (GR) achieves

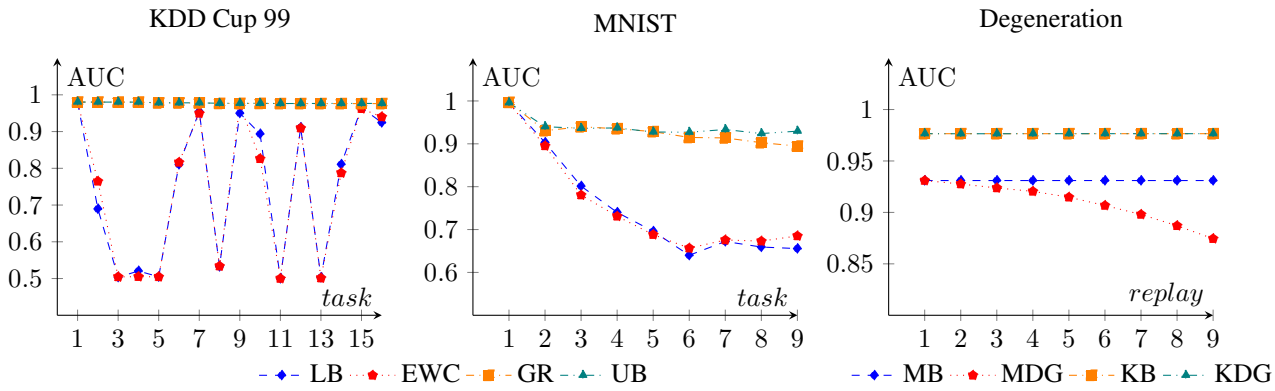

Fig. 2. AUC on KDDCup99, MNIST and the degeneration due to repeated GR on both

practically the same results as the upper bound (UB). EWC completely fails to prevent the VAE from catastrophic forgetting. It is worth noting that EWC as well as the lower bound on some tasks show comparable performance as GR while on the previous and next task they completely fail. This is because the current tasks data encompasses the previous tasks data and hence the VAE generalizes well on previous tasks. If the VAE is trained on a sharply defined task, it is only able to identify the current tasks data as normal while data of previous tasks is classified as anomalous.

5.2. MNIST

The MNIST dataset consists of 70000 grayscale images of hand-written digits with a resolution of 28×28 pixel [13]. Although it is a popular dataset for classification, it can also be used for anomaly detection. For this we first preprocess the data by normalizing each pixel to $[0, 1]$. We use 60000 images for training and 10000 for evaluation and define the digit 0 to be the anomaly class. For creating a sequence of tasks, we first start with the digit 1 as the normal class and expand this definition in each task by the next higher digit. The results are illustrated in Fig. 2. While our method using GR still achieves almost the same performance as the upper bound (UB), a notable decrease in performance can be observed for GR on tasks 5 to 9. This phenomenon is discussed in section 5.3. In contrast to the KDDCup99 dataset, a decrease in performance of both UB and GR is notable after task one. This phenomenon was also observed with a different ordering of the tasks. While defining the normal class with two digits significantly hurts performance, adding even more digits seems to have a negligible effect. EWC again fails and performs the same as the lower bound (LB).

5.3. Replay Degeneration

Due to the well known inability of the VAE to generate highly detailed data [20], i.e. it generates blurry images, repeated GR causes the generated data to represent the original data with an ever decreasing precision. This has a direct effect on continual learning since the replayed data is used for training in each new task and hence leads to a degeneration of the VAEs ability to solve the task. We study this effect by first training a VAE on the last task of the KDDCup99 as well as the MNIST dataset and repeatedly use GR to train on the same task. This task uses the widest definition of the normal class according to the task definition of sections 5.1 and 5.2. By using this setup, we can observe how long the VAE is able to generate useful data without suffering from forgetting. The results of this experiment are illustrated in Fig. 2. We use the VAEs performance on the last task as a baseline (KDDCup99: KB, MNIST: MB) and compare it with repeated replay (KDDCup99: KDG, MNIST: MDG). On the KDDCup99 dataset the VAE does not suffer from forgetting over the tested 9 replays while on MNIST forgetting starts right after the first replay and increases with every replay. This is due to the much higher complexity of the MNIST data compared to KDDCup99 data. The generated data lacks details, when compared to the original data, and leads to ever increasing forgetting when using repeated GR. One way of alleviating this could be to incorporate the continued replay into the training of the VAE. This could be done by training the VAE not only to reconstruct the training data but to also reconstruct the repeated replays of this data. Training a VAE like this would be similar to training a chain of VAEs with shared weights. We leave this open for future research.

6. CONCLUSION

We study the extension of GR for continual learning to the problem of anomaly detection using VAEs and propose a simple but effective system to mitigate catastrophic forgetting. While achieving results comparable to an upper bound, the method requires no additional components other than the VAE. It instead makes use of the generative capability of the VAE, which is otherwise unused in the context of anomaly detection.

7. REFERENCES

[1] Stephan Lewandowsky and Shu-Chen Li, "Catastrophic interference in neural networks: causes, solutions, and data," in *Interference and inhibition in cognition*, pp. 329–361. Elsevier, 1995.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436, 2015.

[3] Robert M French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[4] Jinwon An and Sungzoon Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, pp. 1–18, 2015.

[5] Suwon Suh, Daniel H Chae, Hyon-Goo Kang, and Seungjin Choi, "Echo-state conditional variational autoencoder for anomaly detection," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1015–1022.

[6] Yuta Kawachi, Yuma Koizumi, and Noboru Harada, "Complementary set variational autoencoder for supervised anomaly detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2366–2370.

[7] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 187–196.

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, p. 201611835, 2017.

[9] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds., International Convention Centre, Sydney, Australia, 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, PMLR.

[10] Zhizhong Li and Derek Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[11] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner, "Variational continual learning," in *International Conference on Learning Representations*, 2018.

[12] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems*, 2017, pp. 2990–2999.

[13] Yann LeCun and Corinna Cortes, "MNIST handwritten digit database," 2010.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, July 2009, pp. 1–6.

[15] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 15, 2009.

[16] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680. Curran Associates, Inc., 2014.

[18] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther, "Ladder variational autoencoders," in *Advances in neural information processing systems*, 2016, pp. 3738–3746.

[19] Salvatore J Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," Tech. Rep., COLUMBIA UNIV NEW YORK DEPT OF COMPUTER SCIENCE, 2000.

[20] X. Hou, L. Shen, K. Sun, and G. Qiu, "Deep feature consistent variational autoencoder," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 1133–1141.