Deep Learning in Resource and Data Constrained Edge Computing Systems

Pranav Sharma, Marcus Rüb, Daniel Gaida, Heiko Lutz, and Axel Sikora

Hahn-Schickard-Gesellschaft für angewandte Forschung e.V.,
Wilhelm-Schickard-Str. 10, 78052 Villingen-Schwenningen, Germany
{firstname.lastname}@hahn-schickard.de

Abstract. To demonstrate how deep learning can be applied to industrial applications with limited training data, deep learning methodologies are used in three different applications. In this paper, we perform unsupervised deep learning utilizing variational autoencoders and demonstrate that federated learning is a communication efficient concept for machine learning that protects data privacy. As an example, variational autoencoders are utilized to cluster and visualize data from a microelectromechanical systems foundry. Federated learning is used in a predictive maintenance scenario using the C-MAPSS dataset.

Keywords: Variational Autoencoders · Federated Learning · Unsupervised Learning · Predictive Maintenance

1 Introduction

Usually, deep learning methods are in need of a lot of labeled training data and computing resources to exploit their full potential. In most industrial applications, labeled training data is very expensive and time-consuming to collect. With the ongoing trend of bringing artificial intelligence (AI) on edge and embedded devices, also known as edge AI, the computational power is limited too. In this paper, methodologies that counteract the scarcity of labeled data are presented and exemplified by selected applications from production industry. These methods are variational autoencoders and federated learning [3], which are applied to the following applications:

1. Clustering and visualization of wafermap patterns
2. Anomaly detection for sensor data of a furnace
3. Predictive maintenance using federated learning

In the first two applications unsupervised learning is employed, which is a classical methodology to detect patterns in data without the need of labeling the data. In the latter application, federated learning is used to demonstrate its use in the case of edge AI. All of these applications are used as examples, to demonstrate the usage of aforementioned techniques.

2 Methods & Related Work

In this section, variational autoencoders and federated learning are introduced.

2.1 Variational Autoencoder

Autoencoders belong to the family of unsupervised machine learning methods and are used for dimensionality reduction. An autoencoder encodes high-dimensional input data to a lower dimensional latent space and then decodes this back to its original dimension to restore the input data. A variational autoencoder (VAE) encodes the input to corresponding mean and variance, which means that the input data is assumed to come from (or) generated from a statistical process [2]. These mean and variance are used to reconstruct the input during training. Doing this, forces the encoding of the latent space to be meaningful everywhere. For both methods, the lower dimensional latent space is used to e.g. analyse or visualize the original data distribution.

2.2 Federated Learning

The AI market is dominated by tech giants like Google, Amazon and Microsoft, which provide cloud solutions and APIs (application programming interface) for AI. This monopolization of data, develops mistrust, especially in small and medium-sized companies to make their data available for AI or to use it themselves. Instead of collecting data and sharing it in a data center, the data should be kept on the embedded devices where it is collected. To be able to use AI in this scenario, McMahan et al. [3] introduced a learning algorithm in 2017 that allows any number of clients with local training to improve the model parameters of a global model shared with all other devices. This algorithm is called federated learning that follows the approach of *“bringing code to data instead of data to code”*.

Imagine a production chain in which several motors and heating elements are in operation. In order to avoid production downtimes, the machines are equipped with sensors that allow to do condition monitoring. Predictive maintenance algorithms estimate the next maintenance date based on the results of this monitoring. It is evident that such sensitive production data should not leave *“the house”*. To prevent this from happening, the machine learning model is trained with the locally kept data in the company and only the model parameter changes are forwarded to the server. The server collects the parameters of each production line and aggregates them by the federated averaging algorithm. The updated model is then redistributed to all clients.

The use of federated learning in the application case of mechanical manufacturers has some differences compared to the original intended application by Google. The main difference is the number of clients and therefore less possibility to compensate for outliers.

3 Results

3.1 Clustering and Visualization of Wafermap Patterns

Fig. 1: Comparison of input wafermap image with reconstructed wafermap image.

Production of chips from silicon wafer requires optimum performance checks for each chip, which are typically electrical measurements. The electrical measurements for all chips of a wafer results in a wafermap (see Fig. 1). A wafermap visualizes the measured values of one electrical measurement as color-coded values. Wafermaps produced in a production process, may contain patterns that are result of production and material changes over a time horizon. The observed patterns can be utilized to gain insight into their cause of production.

Fig. 2: Structure of encoder and decoder section of wafermap VAE.

The wafermaps studied in this paper come from the microelectromechanical systems (MEMS) foundry at Hahn-Schickard. A diverse range of patterns are available in these wafermaps. This requires identification of patterns on all of the available wafermaps. To avoid to have to go through all of the wafermaps manually and keep track of all the patterns along with adding labels, we chose

unsupervised learning techniques to help cluster all wafermaps with similar patterns in the same cluster.

Fig. 3: Visualization of representations for the learned latent space.

Following [4] and [6], it was chosen to use a variational autoencoder to learn a lower dimensional latent space for the wafermaps of the available wafers. A two dimensional latent space representation was utilized, to make the encodings more human interpretable. This lack in dimensionality led to a bias in reconstruction of the wafermaps. This bias can be seen in Fig. 1b. But this wafer reconstruction did not change the patterns appearing in the wafermaps. The architecture of encoder and decoder subsection of the variational autoencoder is shown in Fig. 2a and Fig. 2b, respectively. One can view the latent space based reconstructions produced by trained variational autoencoder for wafermaps in Fig. 3.

Fig. 4: Clustering of encodings formed in the latent space using k-means clustering and visualization of wafermaps that are representative for their clusters (best viewed in color).

Once the encodings are generated, they are utilized to perform clustering of patterns available in wafermaps. A k-means clustering method was utilized to identify the clusters in the given latent space as seen in Fig. 4. From Fig. 4 one can see how many different patterns there are and how often each pattern appears.

With this information, one can deal with the most frequent patterns and try to identify the processes that produce these patterns, to avoid the patterns in the future.

3.2 Anomaly Detection for Sensor Data of a Furnace

Fig. 5: Raw visualization of latent space encodings.

Fig. 6: Visualization of latent space encodings with fitted Bayesian distribution.

Measurements of various sensors (eight temperature sensors, a couple of gaseous concentrations, timestamps, etc.) were recorded during the manufacturing process in a furnace. Proper detection of anomalies in this recorded high dimensional space of timeseries data is difficult and error prone. To deal with this, unsupervised deep learning was used for dimensionality reduction.

The data of such a process has to be processed properly, as it is time dependent. First, a difference between all of the consecutive datapoints of all measurements (including time) is calculated and is appended to the state as input too. Then all data is normalized.

In this paper, a variational autoencoder was utilized to reduce the multi-dimensional process parameters to a two dimensional latent space as shown in Fig. 5. The architecture of encoder and decoder section of the VAE can be seen in Fig. 7a and Fig. 7b, respectively. Once the network is trained, one can see cluster of points for all process steps, such as heating up, processing and cooling

Fig. 7: Structure of encoder and decoder section of VAE for sensor data of furnace.

down. If one sees the need, then the encoder network can be further tuned to separate the clusters even more by means of labeled process information.

Once the final latent encodings for all different process steps are produced, one can fit them into individual bayesian mixture distributions, see Fig. 6. Then, distance from the observed distribution is used to detect anomalies. Fig. 8 shows a well performing batch, which can be observed via the distance from observed distribution.

Fig. 8: Visualization of a well performing batch for the process.

3.3 Predictive Maintenance using Federated Learning on Edge Devices

With the use of federated learning, a use case is presented with a machine manufacturer offering a predictive maintenance service to its customers. Each customer updates a machine learning model using the data of its machine and sends

the model update back to the machine manufacturer. The developed network ar-

Fig. 9: Architecture for federated learning; Comparison of learning results for federated learning, one big centralized dataset and one worker standalone, respectively.

chitecture is shown in Fig. 9a. The architecture can consist of a large number of workers and one server. Workers are edge devices at different performance levels, which are installed at the different customer locations, where they remotely train neural networks on the respective data. Here Nvidia Jetson Nano, Raspberry Pi 4 and Intel NUC were used as workers and Raspberry Pi 3 as server. All workers consist of a neural network model and a transmitter/receiver unit that controls the transmission and reception of shared data. The server consists of an aggregator part that combines the individual weights in a global model and a sender/receiver which connects the server to the network.

MQTT is commonly used as network protocol in the field of internet of things (IoT) due to several reasons. Among others it has small overhead, it can be scaled easily, it is easy to implement and preserves privacy, thus it is used in this federated learning scenario as well.

Till date, we have tested federated learning in a network with four to 30 workers on predictive maintenance datasets. As a result, each subdataset of the well-known C-MAPSS dataset [5] was distributed unevenly among these workers. To improve the federated learning results on each subdataset, the learning rate of the optimizer was adjusted as well as the number of learning epochs on the customer data. Taking the training on a centralized storage as benchmark with the complete training dataset stored at one place, the federated averaging (FedAvg) [1] algorithm performs very well. Fig. 9b shows losses from training of centralized model and the single worker model over the learning epochs. The loss incurred for federated learning is shown above the “communication rounds” axis. A single “communication round” contains the four steps: (1) Distribution of the model from server to workers, (2) Training of models on worker’s data for several epochs, (3) Sending weight updates from workers to the server, and (4) Aggregating weight updates by federated averaging into a new global model.

Fig. 9a shows that learning on distributed datasets achieves almost the same accuracy as learning centralized. This comparison is made only to assess the

performance in terms of prediction. Centralized learning can only be seen as a benchmark, not as a real alternative as privacy preservation is not sufficient. The learning results of federated learning, when compared with results attained by a single worker (without connection to federated learning), are better, because one single worker operates on less amount of data. The amount of data available to one single worker, was often not sufficient to achieve clean convergence.

4 Conclusion

In this paper, it was shown that deep learning can be used for industrial processes, where data is scarce and data privacy is important. Mainly, unsupervised learning methods such as variational autoencoders can be used to cluster and visualize high-dimensional data as was shown with clustering of wafermaps and visualization of process data of a furnace. For a sequential and fixed duration process, one can use such clustering of encodings for monitoring the undergoing process. Also, for monitoring the final production of any process, latent space encoding of samples can give insight into issues and opportunities for the process.

Furthermore, the basic ideas of federated learning were introduced, which make them predestined for industrial use cases. It was shown that the accuracy of the predictions from a federated learning model is in a similar range to the prediction results of centralised training, based on same inputs. In federated learning, any increase (with required data privacy) in the data volume should thus lead to an increase in the quality of predictions. The customer of a machine with a predictive maintenance model, that is optimized via federated learning, will benefit by reducing production downtime through intelligent algorithms.

References

1. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kidon, C., Konecny, J., Mazzocchi, S., McMahan, H.B., Van Overveldt, T., Petrou, D., Ramage, D., Roselander, J.: Towards Federated Learning at Scale: System Design (2019), <http://arxiv.org/abs/1902.01046>
2. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: *ICLR* 2. p. 10 (2014)
3. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-Efficient Learning of Deep Networks from Decentralized Data. In: *AISTATS*. vol. 54, pp. 1273–1282 (2017)
4. Santos, T., Kern, R.: Understanding wafer patterns in semiconductor production with variational auto-encoders. In: *ESANN*. pp. 141–146. No. April, Bruges, Belgium (2018)
5. Saxena, A., Goebel, K.: Turbofan Engine Degradation Simulation Data Set (2008), <http://ti.arc.nasa.gov/project/prognostic-data-repository>
6. Tulala, P., Mahyar, H., Ghalebi, E., Grosu, R.: Unsupervised Wafermap Patterns Clustering via Variational Autoencoders. In: 2018 International Joint Conference on Neural Networks (IJCNN). vol. 2018-July,

pp. 1–8. IEEE (jul 2018). <https://doi.org/10.1109/IJCNN.2018.8489422>,
<https://ieeexplore.ieee.org/document/8489422/>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.