Hybrid Intelligent Systems Based on Fuzzy Logic and Deep Learning

Alexey Averkin(✉)

Federal Research Centre of Informatics and Computer Science of RAS, Moscow,
Vavilova, 42, Moscow, Russia
averkin2003@inbox.ru

Abstract. The purpose of this lecture is to establish the fundamental links between two important areas of artificial intelligence - fuzzy logic and deep learning. This approach will allow researchers in the field of fuzzy logic to develop application systems in the field of strong artificial intelligence, which are also of interest to specialists in the field of machine learning. The lecture also examines how neuro-fuzzy networks make it possible to establish a link between symbolic and connectionist schools of artificial intelligence. A lot of methods of rule extraction from neural networks are also investigated.

Keywords: Deep learning · Neural networks · Rule extraction · Convolutional neural network · Machine learning · Artificial intelligence

1 Introduction

This lection introduces the terms and definitions of machine learning that are relevant to the context of extracting rules from classical and deep neural networks. It includes the problem of classification as a whole, as well as rule-based teaching methods and neural networks. Then, we will look at the current state of rule extraction from neural networks. Here we define the problem as well as the main approaches to its solution and present some of the existing rules extraction algorithms. The last part discusses specific problems when working with deep neural networks. At this stage, we also propose some algorithms that can successfully extract rules from these more complex neural networks.

Artificial Neural Networks (ANN) are widely known parallel computing models that exhibit excellent behavior in solving complex problems of artificial intelligence. However, many researchers refuse to use them due to their being a “black box”. This means that determining why a neural network makes a specific decision is a difficult task.

This is a significant drawback, since it is difficult to trust the reliability of the network that solves real problems without the ability to make acceptable decisions. For example, this is the case in critical, in terms of safety, applications where hidden failure can lead to life-threatening actions or huge economic losses.

In addition, studying how neural networks extract, store and transform knowledge can be useful for future machine learning methods. For example, increasing the transparency of neural networks can help detect the so-called hidden dependencies that

G. S. Osipov et al. (Eds.): Artificial Intelligence, LNAI 11866, pp. 3–12, 2019.
https://doi.org/10.1007/978-3-030-33274-7_1

are not present in the input data, but appear as a result of their integration into the neural network.

To overcome this lack of neural networks, researchers came up with the idea of extracting rules from neural networks, which can became a bridge between symbolic and connectionist models of knowledge representation in artificial intelligence.

Most authors focus on extracting the most understandable rules, and at the same time they should mimic the behavior of the neural network as precisely as possible, right up to an isomorphic representation of fuzzy rules in the form of a neuro-fuzzy system. Since 1992, since Chang's doctoral thesis on neuro-fuzzy networks, much work has been done in this area, which ended with the creation of the direction of soft computing. Since then, many methods for extracting rules from neural networks have been developed and evaluated, and excellent results have been obtained for many approaches.

However, despite the fact that there are quite a few available algorithms, none of them has ever been explicitly tested in deep neural networks. In addition, most authors focus on networks with only a small number of hidden layers.

Only in the last few years pioneering work has appeared on the analysis of specific methods for extracting rules from deep-seated networks and new algorithms are presented that are capable of performing this task.

2 Methods for Extracting Rules from the Neural Network

In artificial intelligence, neural networks and rule-based learning methods are two approaches to solving classification problems. Both methods are known variants of studying models that predict classes for new data. For many tasks, NN rules-based teaching methods excel in accuracy.

However, neural networks have one major drawback: the ability to understand what a trained concept models, NN is not as strong as for rule based approaches. The concepts learned by neural networks are difficult to understand because they are represented using a large set of parameters [1].

Increasing the transparency of neural networks by extracting rules has two main advantages. First, it gives the user some insight into how the neural network uses input variables to make a decision—and can even reveal hidden functions in NN when the rules are used to explain individual neurons. Identification of particularly important attributes or identification of the causes of neural network errors can be part of the understanding. Trying to make opaque neural networks more understandable, methods for extracting rules eliminate the gap between accuracy and clarity [2–4].

A more comprehensible form is required if, for example, a neural network is to be used in safety-critical applications, such as aircraft and power plants. In these cases, it is extremely important that the system user have the opportunity to check the output of the artificial neural network under all possible input conditions [5].

To formalize the task of extracting rules from a neural network, we give Craven's definition: "Given the trained neural network and the data on which it was trained, create a description of the network hypothesis that is understandable, but comes close to the network prediction behavior" [6].

To distinguish between different approaches for extracting rules from neural networks, Andrews introduced the widely used multidimensional taxonomy [5]. The first dimension they describe is a form (for example, IF-THEN rules or a variant of fuzzy production rules).

The third dimension is the quality of the rules extracted. Since quality is a broad term, it is divided into several criteria, namely, accuracy, fidelity, consistency and comprehensibility. While accuracy measures the ability to "correctly classify previously unseen examples", validity measures the degree to which rules can "imitate" the behavior of a neural network well [2].

The second dimension is called transparency and describes the strategy followed by the algorithm for extracting rules. If the method uses a neural network only as a black box, regardless of the architecture of NN, we call the pedagogical approach. If, instead, the algorithm takes into account the internal structure of the neural network, we call this approach decompositional. If the algorithm uses components of both pedagogical and decomposition methods, then this approach is called eclectic.

The third dimension is the quality of the rules extracted. Since quality is a broad term, it is divided into several criteria, namely, accuracy, fidelity, consistency and comprehensibility. While accuracy measures the ability to correctly classify previously unseen examples, validity measures the degree to which rules can imitate the behavior of a neural network well [2].

Fidelity can be considered as accuracy relative to the output of NN. Consistency can only be measured when the rule extraction algorithm involves learning the neural network instead of processing the already trained NN: The extracted rule set is considered consistent when the neural network generates rule sets that correctly classify test data for different training sessions. Comprehensibility is considered here as a measure of the size of the rules, that is, short and few rules are considered more understandable [5].

In this review we will focus only on the three criteria described. In accordance with [7], we focus on methods that do not impose special requirements on how the neural network was trained before the rules were extracted. In addition, only algorithms that are capable of extracting rules from direct propagation neural networks, despite any other characteristics of the architecture, are analyzed. According to [3] we want the algorithm to offer a high level of generality.

Let us analyze some methods for extracting rules that meet the above characteristics. We start with decomposition approaches. As mentioned earlier, decomposition approaches for extracting rules from neural networks operate at the neuron level. Usually, the decomposition method analyzes each neuron, and forms rules that imitate the behavior of this neuron. For various reasons, we do not take into account all available decomposition approaches in the subsequent review. We consider here the KT algorithm, Tsukimoto's approach and rule extraction through decision tree induction (CRED algorithm).

The KT algorithm was one of the first decomposition approaches for extracting rules from neural networks was presented in [29]. The KT algorithm describes each neuron (layer by layer) with the IF-THEN rules by heuristically searching for combinations of input attributes that exceed the threshold of the neuron. The rewrite module is used to obtain rules that refer to the original input attributes, and not to the outputs of the previous level. To find suitable combinations, the KT method applies a search on the tree, that is, a rule (represented as a node in the tree) at this level generates its child nodes by adding an additional available attribute [8]. In addition, the algorithm uses a number of heuristics to stop the growth of a tree in situations where further improvement is impossible. The algorithm searches for both confirming and non-confirming rules. These are the rules that predict the triggering of a neuron if there is a certain input configuration.

Polinomial Tsukimoto's approach to extracting rules from a neural network is very similar to the KT method. It also uses a layered decomposition algorithm to extract the IF-THEN rules for each neuron, and also monitors the strategy for finding input configurations that exceed the threshold of the neuron. The main advantage of the Tsukimoto method is its computational complexity, which is polynomial, while the KT method is exponential [9]. The algorithm achieves polynomial complexity by searching for relevant terms using the space of multilinear functions. In the second stage, these terms are used to create IF-THEN rules. Subsequently, if any, training data is used to improve the accuracy of the rules. In the last step, the Tsukimoto algorithm attempts to optimize clarity by removing non-essential attributes from the rules.

Another method for extracting rules through decision tree induction was introduced in [10]. Their CRED algorithm converts each output unit of a neural network into a solution, where tree nodes are tested using nodes of a hidden layer, and leaves represent a class. After this, intermediate rules are extracted from this step. Then for each split point used in these rules, another decision tree is created using split points on the input layer of the neural network. In the new trees, the leaves do not directly choose the class. Extracting the rules from the second decision tree leads us to the description of the state of hidden neurons, consisting of input variables. As a final step, intermediate rules that describe the output layer through the hidden layer and those that describe the hidden layer based on the inputs of the neural network are replaced. Then they are combined into construction rules that describe the output of the neural network based on its input data.

The next group of pedagogical approaches of rule extraction based on validity interval analysis, approaches for rule extraction using sampling and rule extraction by reverse engineering the neural network.

Pedagogical approaches, as opposed to decomposition, do not take into account the internal structure of the neural network. The motive in pedagogical approaches is to treat trained NN at the lowest possible level of granularity, that is, as a whole or, alternatively, as a black border [11]. Their idea is to extract the rules by directly matching the input data with the output data [12]. More formally, we can assume that pedagogical approaches have access only to the function of the neural network. This function returns the output of the neural network for random input, but offers no understanding of the internal structure of NN or any weights (except for the number of inputs and outputs in NN). Having NN, this class of algorithms tries to find coherence between possible input variations and outputs created by the neural network, while some of them use specified training data, and some do not. As in the previous section, we will not discuss all possible algorithms, but only briefly go over the main ones.

Rule extraction based on interval analysis approach uses the interval confidence analysis (VIA), a kind of sensitivity analysis, to extract rules that mimic the behavior of a neural network [13]. The main idea of this method is to find the input intervals in which the output signal NN is stable, that is, the predicted class is the same for slightly changing input configurations. As a result, VIA provides the basis for reliably correct rules.

Retrieving rules using sampling represents several methods that follow more or less the same strategy for extracting rules from a neural network using sampling, that is, they create an extensive set of data as a basis for extracting rules. After that, the selected data set is submitted to a standard learning algorithm for generating rules that simulate the behavior of a neural network. In [2] it is proved that the use of sample data exceeds the use of training data in the problems of extracting rules.

One of the first methods that followed this strategy was the Trepan algorithm [14]. Trephine works very much like C4.5 by searching for split points on training data for individual instances of different classes. The main differences from C4.5 are the best strategy for expanding the tree structure, additional split points in the M-of-N style and the ability to choose additional learning examples at deeper points of the tree. As a result, the algorithm also creates a decision tree, which, however, can be transformed into a set of rules, if necessary.

Another of these very general pedagogical approaches that use sampling to extract rules from the neural network is presented in [13]. The algorithm, called Binarized Input-Output Rule Extraction (BIO-RE), is capable of processing only NN with binary or binarized input attributes. BIO-RE creates all possible input combinations and requests them from the neural network. Using the NN output, a truth table is created for each example. From the truth table it is just as easy to go to the rules, if necessary.

ANN-DT is another decision-based sampling method for describing the behavior of a neural network [14]. The overall algorithm is based on CART with some variations in the initial implementation. ANN-DT uses the sampling method to expand the training set so that most of the training sample is still representative. This is “achieved using the nearest neighbor method, in which the distance from the sample point to the nearest point in the training data set is calculated” [14] and compared with the reference value.

The idea of creating a large set of instances at the first stage is also implemented by the STARE algorithm [15]. Like BIO-RE, STARE also forms extensive truth tables for learning. The advantage of STARE is its ability not only to handle binary and discrete attributes, but also to work with continuous input data. For the formation of truth tables, the algorithm rearranges the input data, while for each continuous attribute “it is necessary to sample it over the entire range of values with a high frequency”.

The last pedagogical approach using a sample of educational data that we want to briefly present here is KDRuleEx [4]. Like Trepan, the Sethi algorithm also generates additional learning cases where the basis for the following separation points is too small. KDRuleEx uses a genetic algorithm to create new training examples. The technique leads to a decision table that can be converted, for example, into IF-THEN rules, if desired.

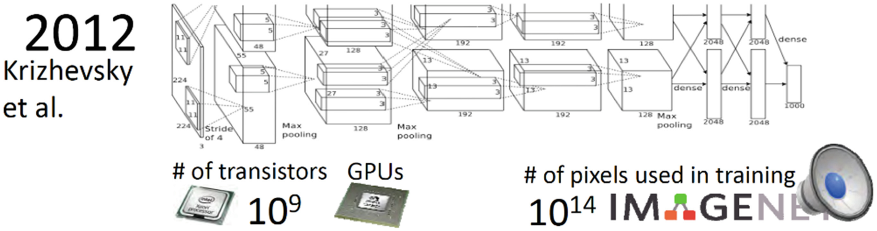
Eclectic approach are the methods for extracting rules include elements of both pedagogical and decompositional, then such methods are known as eclectic [3]. In particular, eclectic approaches use knowledge of the internal architecture and/or weight vectors in the neural network to complement the symbolic learning algorithm [5].

The fast retrieval of rules from a neural network approach includes the FERNN approach, which first tries to identify the corresponding hidden neurons, as well as the corresponding inputs to the network. For this step, a decision tree is constructed using the well-known algorithm C4.5. The rule extraction process leads to the generation of M-of-N and IF-THEN rules. Having a set of properly classified teaching examples, FERNN analyzes the activation values of each hidden unit. For each hidden unit, activation values are sorted in ascending order. Then use the C4.5 algorithm to find the best split point to form the decision tree. The most interesting from the point of view of this study is the extraction of rules using neuro-fuzzy models On the other hand, systems based on fuzzy rules (FRBS), developed using fuzzy logic, have become a field of active research over the past few years. These algorithms have proven their strengths in tasks such as managing complex systems, creating fuzzy controls. The relationship between both worlds (ANN and FRBS) has been carefully studied. Indeed, this is a close relationship, since equivalence results were obtained [17].

This link gives two immediate and important conclusions. First, we can apply what was discovered for one of the models to the other. Secondly, we can translate the knowledge embedded in the neural network into a more cognitively acceptable language - fuzzy rules. In other words, we get a clear interpretation of neural networks [18–20].

3 Extracting Rules from Deep Neural Networks

Since 2012, the revolution began networks of deep learning. Consider one of the first and probably the most cited works in convolutional NN- Alexnet - it has 7 hidden layers, 650,000 neurons, 60,000,000 parameters. Studied at 2 GPU for 1 week (Picture 1). Where do we get enough images to train her?



Picture 1. Alexnet - first deep learning champion in image recognition

In 2010 dataset Imagenet has appeared. The emergence of Imagenet brought the learning of neural networks to a whole new level. Parallel rapidly developed computing power, which led computer vision to the kind that we know and love it now. Since 2010, the annual Imagenet competition has also been held, where for the first time in 2012, the Alexnet convolutional neural network won and, since then, the National Assembly has not lost its positions. The last winner, the National Assembly presented by scientists from China, contained 269 layers (Picture 2).

IMAGENET

2016 competition winner:

269 layer R-CNN network

Gated Bi-directional CNN for Object Detection
X.Zeng et al., The Chinese University of Hong Kong, Sensetime Group

Picture 2. Dataset Imagenet.

To semantic intepretation of the deep learning blackbox neuro-fuzzy networks can be use used instead of full connection layer (Fig. 3). For example, ANFIS (adaptive neuro-fuzzy system) [25] is a multilayer feed forward network. This architecture has five layers such as fuzzy layer, product layer, normalized layer, de - fuzzy layer and total output. The fixed nodes are represented by circle and the node s represented by square are the adapted nodes. ANFIS gives the advantages of the mixture of neural network and fuzzy logic.

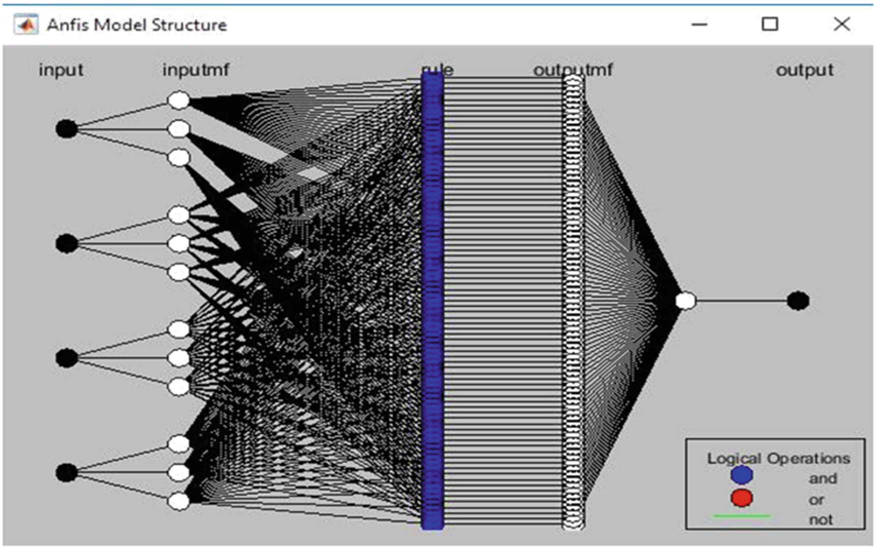


Fig. 3. Structure of ANFIS model for time series forecasting

The aim of mixing fuzzy logic and neural networks is to de-sign an architecture, which uses a fuzzy logic to show knowledge in fantastic way, while the learning nature of neural network to maximize its parameters. ANFIS put forward by Jang in 1993 integrate the advantages of both neural network and fuzzy systems, which not only have good learning capability, but can be interpreted easily also. ANFIS has been used in many applications in many areas, such as function approximation, intelligent control and time series prediction.

A hypothetical system can be created using two components [26]. The first is deep learning feature generation which can be used to create representative features from text directly. The deep learning system would initially be trained on unlabeled data. Once these features are extracted from the deep learning system, they will be integrated into fuzzy-inference systems. These systems can incorporate both the features detected from the deep learning as well as subjective information from an analysts as a method of biasing the system. These two pieces together can be used for classification purposes. The final system would therefore be able to report both classification results and the specific features and rules that were activated for the system to arrive at its conclusion. Additionally, the final system could be further biased by an analyst as a form of feedback.

In previous part has shown that there is a wide variety of algorithms to describe a one-hidden-layer neural network's behavior by rules.. In [21] it was made a first step in investigating the problem of extracting rules from deep neural networks. DNNs have a few characteristics that, compared to one-hidden-layer neural networks, can complicate the rule extraction process. For deep neural networks, the rule extraction problem is very complex. We must not only to explain an output's behaviour, but also to describe the behaviour of a neuron in a hidden layer. In [21] authors give an overview of how decompositional rule extraction algorithms like DeepRED or pedagogical approaches can be used to analyse and explain the behaviour of a DNN.

Very interesting approach is suggested in [22, 23], where the author established a fundamental connection between two important fields in artificial intelligence i.e. deep learning and fuzzy logic. He shows, how deep learning could benefit from the comparative research by re-examining many trail-and-error heuristics in the lens of fuzzy logic, and consequently, distilling the essential ingredients with rigorous foundations. The author proposed deep generalized hamming network (GHN) as such not only lends itself to rigorous analysis and interpretation within the fuzzy logic theory but also demonstrates fast learning speed, well-controlled behaviour and state-of-the-art performances on a variety of learning tasks. In [24] it is presented another approach for incorporating such rule based methodology into neural networks by embedding fuzzy inference systems into deep learning networks.

4 Conclusion

The study of fuzzy logic culminated in the end of the 20th century, and since then has begun to decrease [27]. This decrease can be partly attributed to the lack of results in machine learning. Extracting rules is one way to help understand neural networks. These studies will pave the way for fuzzy logic researchers to develop applications in artificial intelligence and solve complex problems that are also of interest to the machine learning community. Experience and knowledge in the field of fuzzy logic are well suited for modeling ambiguities in big data, modeling uncertainty in the representation of knowledge and providing transmission training with non-inductive inference, etc.

References

1. Craven, M., Shavlik, J.W.: Using sampling and queries to extract rules from trained neural networks. In: ICML, pp. 37–45 (1994)
2. Johansson, U., Lofstrom, T., Konig, R., Sonstrod, C., Niklasson, L.: Rule extraction from opaque models–a slightly different perspective. In: 5th International Conference on Machine Learning and Applications. ICMLA 2006, pp. 22–27 (2006)
3. Craven, M., Shavlik, J.: Rule extraction: where do we go from here. In: University of Wisconsin Machine Learning Research Group Working Paper, pp. 99–108 (1999)
4. Sethi, K.K., Mishra, D.K., Mishra, B.: KDRuleEx: a novel approach for enhancing user comprehensibility using rule extraction. In: 2012 Third International Conference Intelligent Systems, Modelling and Simulation (ISMS), pp. 55–60 (2012)
5. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl.-Based Syst. **8**(6), 373–389 (1995)
6. Craven, M.W.: Extracting comprehensible models from trained neural networks. Ph.D. thesis, University of Wisconsin-Madison (1996)
7. Thrun, S.: Extracting provably correct rules from artificial neural networks. Technical report, University of Bonn, Institut für Informatik III (1993)
8. Fu, L.: Rule generation from neural networks. IEEE Trans. Syst. Man Cybern. **24**(8), 1114–1124 (1994)
9. Tsukimoto, H.: Extracting rules from trained neural networks. IEEE Trans. Neural Networks **11**(2), 377–389 (2000)
10. Sato, M., Tsukimoto, H.: Rule extraction from neural networks via decision tree induction. In: International Joint Conference on Neural Networks. Proceedings. IJCNN 2001, vol. 3, pp. 1870–1875 (2001)
11. Tickle, A.B., Andrews, R., Golea, M., Diederich, J.: The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Trans. Neural Networks **9**(6), 1057–1068 (1998)
12. .
13. Thrun, S.: Extracting rules from artificial neural networks with distributed representations. In: Advances in Neural Information Processing Systems, pp. 505–512 (1995)
14. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in Neural Information Processing Systems, pp. 24–30 (1996)
15. Taha, I.A., Ghosh, J.: Symbolic interpretation of artificial neural networks. IEEE Trans. Knowl. Data Eng. **11**(3), 448–463 (1999)
16. Towell, G.G., Shavlik, J.W.: Extracting refined rules from knowledge-based neural networks. Mach. Learn. **13**(1), 71–101 (1993)
17. Setiono, R., Leow, W.K.: FERNN: an algorithm for fast extraction of rules from neural networks. Appl. Intell. **12**(1–2), 15–25 (2000)
18. Averkin, A., Yarushev, S.: Hybrid neural networks for time series forecasting. In: Kuznetsov, S.O., Osipov, G.S., Stefanuk, V.L. (eds.) RCAI 2018. CCIS, vol. 934, pp. 230–239. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00617-4_21

19. Pilato, G., Yarushev, S.A., Averkin, A.N.: Prediction and detection of user emotions based on neuro-fuzzy neural networks in social networks. In: Abraham, A., Kovalev, S., Tarassov, V., Snasel, V., Sukhanov, A. (eds.) IITI'18 2018. AISC, vol. 875, pp. 118–125. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-01821-4_13
20. Averkin, G.P., Yarushev, S.A.: An approach for prediction of user emotions based on ANFIS in social networks. In: Second International Scientific and Practical Conference Fuzzy Technologies in the Industry, FTI 2018–CEUR Workshop Proceedings, pp. 126–134 (2018)
21. Zilke, J.R., Loza Mencía, E., Janssen, F.: DeepRED – rule extraction from deep neural networks. In: Calders, T., Ceci, M., Malerba, D. (eds.) DS 2016. LNCS (LNAI), vol. 9956, pp. 457–473. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46307-0_29
22. Fan, L.: Revisit fuzzy neural network: demystifying batch normalization and ReLU with generalized hamming network. In: NIPS 2017 (2017)
23. Fan, L.: Revisit Fuzzy Neural Network: Bridging the Gap between Fuzzy Ljgic and Deep Learning. Technical Report (2017)
24. Bonanno, D., Nock, K., Smith, L., Elmore, P., Petry, F.: An approach to explainable deep learning using fuzzy inference. In: Proceedings of the SPIE 10207, Next-Generation Analyst V, 102070D (2017)
25. Jang, S.R.: ANFIS: adaptive-network-based fuzzy inference systems. IEEE Trans. Syst. Man Cybernet. **23**, 665–685 (1992)
26. Bonanno, D., Nock, K., Smith, L., Elmore, P., Petry, F.: An approach to explainable deep learning using fuzzy inference. In: Hanratty, T.P., Llinas, J. (eds.) Next-Generation Analyst V. Proceedings of the SPIE, vol. 10207 (2017)
27. Goodfellow, I., et al.: Generative ad- versarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence,N. D., Weinberger, K.Q., éditeurs: Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014)