Deep Neural Networks for Understanding and Diagnosing Partial Discharge Data

V. M. Catterson, B. Sheng
Institute for Energy and Environment, University of Strathclyde,
Glasgow, United Kingdom.
Email: v.m.catterson@strath.ac.uk

***Abstract*—Artificial neural networks have been investigated for many years as a technique for automated diagnosis of defects causing partial discharge (PD). While good levels of accuracy have been reported, disadvantages include the difficulty of explaining results, and the need to hand-craft appropriate features for standard two-layer networks. Recent advances in the design and training of deep neural networks, which contain more than two layers of hidden neurons, have resulted in improved results in speech and image recognition tasks. This paper investigates the use of deep neural networks for PD diagnosis. Defect samples constructed in mineral oil were used to generate data for training and testing. The paper demonstrates the improvements in accuracy and visualization of learning which can be gained from deep learning.**

***Keywords*—Artificial neural networks; deep learning; partial discharge; diagnostics; UHF monitoring; defects in oil**

I. Introduction

Partial Discharge (PD) is a much-studied phenomenon associated with insulation weakness and breakdown. In HV assets, understanding the nature of the defect causing PD is critical for scheduling appropriate maintenance. One facet of smart grids is increased online condition monitoring and in-field processing capabilities, bringing a corresponding need for automated systems to analyze the large volumes of data captured by PD monitoring systems.

Machine learning techniques have for many years been demonstrated as being capable of automatically diagnosing defects causing PD. These techniques, such as artificial neural networks (ANNs) [1], [2] and support vector machines (SVMs) [3], [4], are so-called *shallow architectures*, where much engineering effort is required up-front to define a feature vector for diagnosis by one or two layers of simple computational units [5]. A variety of features have been trialled for PD diagnosis, including statistical [6], [3] and shape descriptors [7], with some attempt to compare their relative diagnostic powers in specific contexts [8]. However, the level of expertise required in selecting and calculating appropriate features can be a barrier to deploying automated diagnostic systems within utilities.

This paper investigates the use of deep neural networks (DNNs) for diagnosis of PD. Deep networks, which comprise more than two layers of computational units, have been shown to outperform shallow architectures with hand-crafted features for a range of speech and image recognition tasks [9]. This paper presents the results of applying deep learning to a PD dataset previously used for shallow learning [2], and demonstrates the accuracy and visualization improvements which can be gained.

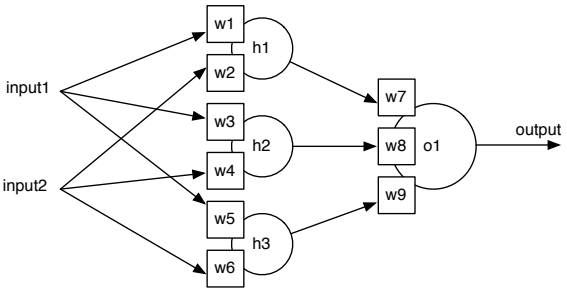

Fig. 1. A two layer network with three hidden neurons and one output neuron

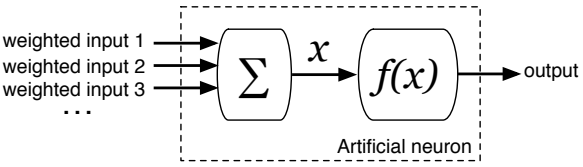

Fig. 2. Each neuron calculates a given function on a weighted sum of inputs

II. Deep neural networks

The standard architecture for a neural network comprises multiple layers of computational units (neurons) (see Fig. 1), with each neuron performing a simple function on the weighted sum of its inputs (see Fig. 2). Each layer is fully connected to the next layer, with those between the inputs and the output layer referred to as hidden layers. The weights between pairs of neurons allow adjustment of the strength of each connection. The network is trained to find the appropriate weights, such that the output of the whole network matches the desired target in the majority of cases.

Training of neural networks was enabled by the backpropagation algorithm, first introduced in 1986 [10]. Thereafter, a variety of network architectures were investigated, testing the effects of number of layers, and number of neurons in each layer. Extra neurons in a layer add linearly to training time, while extra layers of neurons add combinatorially to training times. As a result, after it was shown that any function could be approximated by a two layer network in 1989 [11], ANN work focused on two-layer shallow architectures.

The number of neurons in the output layer is dictated by the type of function being learnt, with classification networks tending to have one output neuron per class. There is no clear heuristic for selecting the number of neurons in the hidden layer. The inputs to the network dictate how many weights on the first layer need to be trained, so the size and format of the input has a significant effect on the training time of the network.

In the past, raw PD data was generally considered too large in size to produce an efficient two-layer network. An alternative to raw data is to extract information-rich features from the raw data [1], [2], [8]. A feature vector may contain tens of values, compared to hundreds or thousands of raw data points, with a significant effect on training. Since the feature vector is also more information-dense than the raw data, it is computationally easier to train an accurate network.

However, as computational power has grown over the years, network size has become less of a constraint than before. In 2006, it was demonstrated that networks with more than two layers of neurons could be trained to extract features automatically from raw images [12]. Further work has shown methods of visualization of what each neuron has actually learned to recognize [13], with intriguing parallels with biological vision.

Initially, deep networks were composed of neurons using the same types of activation functions as those used in shallow architectures. Popular choices are bounded and non-linear, such as the sigmoid (bounded between zero and one), and the hyperbolic tangent (bounded between negative and positive one). The non-linearity allows complexity in the learnable function, while bounds can simplify learning by limiting the output range.

However, such functions also suffer from various problems. Neurons can easily saturate, and take many, many training epochs to return to a useful range of operation. Deep networks compound this problem, since saturated neurons in one layer will block good training of the next layer until they return to non-saturated operation.

Advances in neuroscience suggest that biological neurons do not saturate, but instead perform the leaky-integrate-and-fire function [14]. A simplified version of the biological function was introduced for artificial neural networks, called the Rectified Linear Unit (ReLU):

$$f(x) = max(0, x) \tag{1}$$

ReLU is computationally simpler than sigmoid or hyperbolic tangent, while still being non-linear. It does not saturate in the positive direction, and therefore leads to faster training times, while still retaining accuracy [14].

As a result of these advances, deep neural networks are now out-performing techniques such as SVMs for image and speech recognition tasks [15]. In addition to simply giving higher accuracy, advances in visualizing the function learned by a given neuron mean that they are less of a "black-box" technique than before, and can potentially give a fresh new perspective on a classification problem.

The following section demonstrates how to apply these techniques to PD classification.

III. Applying DNNs to PD diagnosis

The aim of this work is to use deep neural networks to diagnose defects causing PD. Six different types of defect were constructed in oil, and PD measured using a UHF sensor [16]. The defect types are:

- Bad electrical contact (BC)
- Object at a floating potential (FL)
- Metallic protrusion, configuration 1 (PRO1)
- Metallic protrusion, configuration 2 (PRO2)
- Freely rolling particle (RP)
- Surface discharge (SD).

Data was captured in $1s$ bursts and phase resolved, with a phase window of $5.625°$. This gives a phase resolved PD (PRPD) pattern containing $50 \times 64 = 3200$ values, where each value represents the relative amplitude of any PD recorded during that window. This can be represented visually as a pixel intensity in a 50×64 pixel image, as shown in Fig. 3.

Approximately 250–300 PRPD patterns were recorded from each defect type. Due to the higher number of neurons and weights in a deep network, learning tends to be most accurate with an order of magnitude more examples than this. As a result, the original dataset was synthetically increased, as described below.

A. Generating more data

The original data was recorded with a given level of amplification, resulting in all PRPD patterns from one defect type having approximately the same maximum PD amplitude. The first measure to generate more data was simply to scale the existing PRPD patterns by a variable amount, to give a basic simulation of varying amplification in the sensor hardware.

For a given pattern, all values were multiplied by a scale factor randomly selected from a reasonable range. This was done multiple times for each pattern until the total number of patterns for a given defect class was greater than 1000.

A second step involved making slight adjustments to the phase window of PDs within a scaled pattern. Up to half the values in a given pattern were swapped with a value from a neighboring phase window within the same cycle. This had the effect of slightly altering the pattern, while not significantly changing the relationship between phase position and PD amplitude.

In total, these steps increased the dataset from 1341 to 6776 patterns, giving over 1000 examples for each defect type. Examples of original and generated patterns are shown in Fig. 3. This full dataset was then split randomly, with 75% of patterns used for training, and 25% used for testing.

B. Two layer networks

Initially, a shallow two-layer network was constructed, to test the effect of number of neurons on accuracy. The input to the network was the data from a PRPD pattern, i.e. 3200

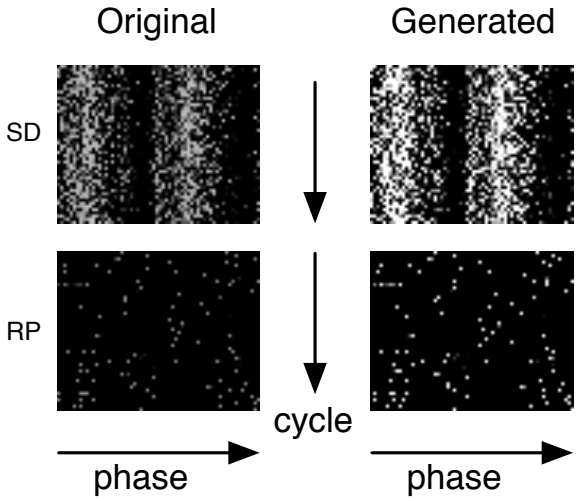

Fig. 3. Examples of original and generated PRPDs, as 50×64 pixel matrices with color intensity representing PD amplitude

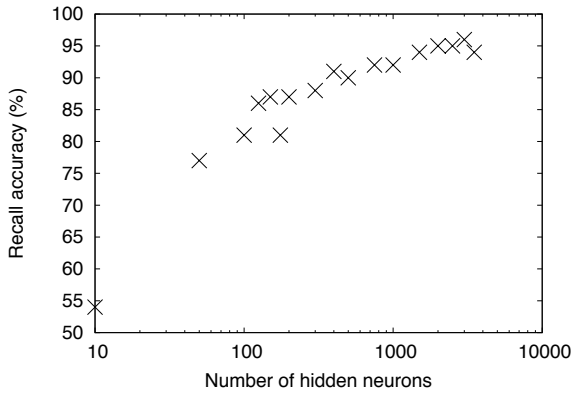

Fig. 4. Recall accuracy of two layer ReLU networks

values. The number of output neurons was six: one for each of the six defect classes, utlizing the softmax activation function.

The number of hidden neurons was varied between 10 and 3500, using the ReLU activation function. The network was trained for up to 10 epochs, and the recall accuracy calculated.

The results (Fig. 4) show good accuracy beginning around 75 neurons, with an overall peak at 3000 neurons. This is not surprising, as any function can be learned with two layers given enough training time and neurons. A reasonable trade-off between accuracy and number of neurons occurs at the 'knee-point' on Fig. 4, around a layer size of 100 to 150 neurons.

C. Deep networks

Next, layers of neurons were added to investigate the effect on accuracy. As before, there were 3200 inputs and six output neurons. The size for every hidden layer was fixed at 100 neurons using the ReLU activation function, and the number of hidden layers was varied from one to seven.

Fig. 5 shows that adding layers does improve accuracy up to a certain point. Beyond five hidden layers, there is a drop

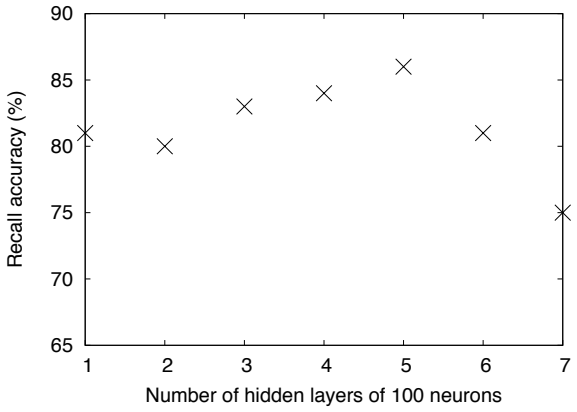

Fig. 5. Recall accuracy of deep ReLU networks

TABLE I
CONFUSION MATRIX FOR RELU NETWORK WITH FIVE HIDDEN LAYERS

	Predicted					
Actual	BC	FL	PRO1	PRO2	RP	SD
BC	311	0	0	0	0	4
FL	1	204	12	22	27	3
PRO1	0	11	224	6	9	1
PRO2	0	17	4	229	24	12
RP	0	29	4	30	214	0
SD	4	3	1	4	1	283

off in accuracy, suggesting that five is the appropriate number for this particular architecture. The confusion matrix for the best performing network is shown in Table I.

D. Comparing activation function

Finally, the choice of activation function was investigated. Deep networks of 100-neuron layers were constructed as in the previous experiment, but with the hidden layer activation function chosen to be sigmoid instead of ReLU.

The results (Table II) show that, as expected, the sigmoid network struggles to learn appropriately with more than one hidden layer. The accuracy falls sharply at two hidden layers, and networks with more than this show a random guess level of accuracy.

This strongly suggests that the ReLU function does indeed enable deep learning to take place. In addition, even in the networks with one hidden layer of 100 neurons, the ReLU network has an improved accuracy of 81% over the sigmoid accuracy of 72%. These new approaches to neural network design offer improved accuracy for PD classification tasks.

TABLE II
RELATIVE ACCURACY OF SIGMOID VERSUS RELU DEEP NETWORKS

Number of hidden layers	1	2	3	4	5	6
Sigmoid accuracy	72%	41%	17%	18%	17%	18%
ReLU accuracy	81%	80%	83%	84%	86%	81%

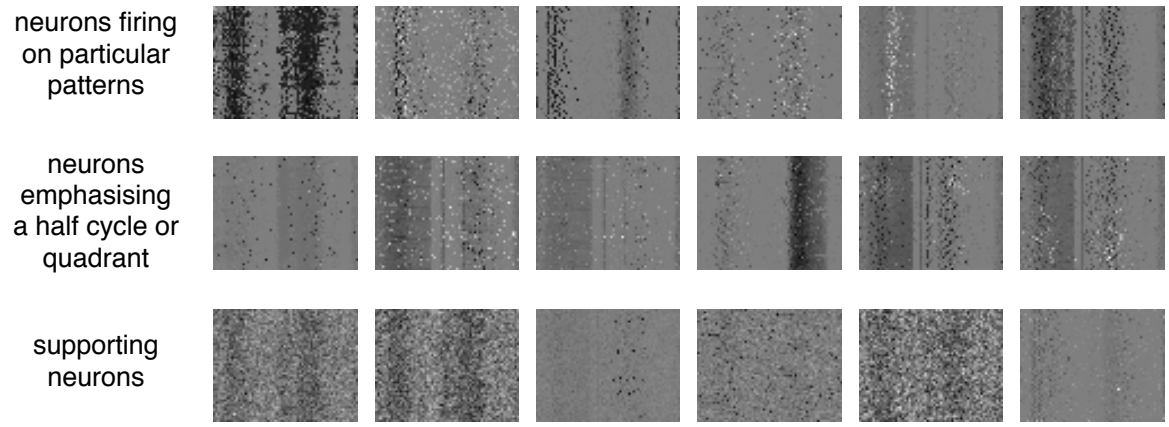

Fig. 6. Examples of neuron activation, visualized by input weightings. Each image represents one neuron.

E. Visualizing the learning

An interesting benefit of the deep network is that it tends to result in sparse networks, where a relatively few number of neurons are activated by a single input [14]. Previously this was considered to be wasteful of training resources compared to a compact network. With current computing resources this is less of a concern, and holds a significant advantage for visualizing the learning that has occurred. The input weightings reveal the specific pattern that has been learnt by that neuron.

In the case of PD diagnosis, this allows a novel way of examining the core properties of a given defect's PRPD pattern. Some neurons respond very clearly to a particular defect type, and the pattern of weights identifies the critical information being used to make the diagnosis. Others perform more of a supporting role, emphasising the effects of half or quarter cycles, or other parts of the PRPD. Some examples are shown in Fig. 6. To generate these images, the input weights to a given neuron have been scaled to values between 0 and 255. A mid grey tone represents a weight close to zero, while black and white represent very strong weights.

IV. Conclusions

This paper has introduced the use of deep neural networks for diagnosis of phase resolved PD data. Data was captured from defect samples in oil, using a UHF sensor. The effect on the diagnosis of the number of layers, and the rectified linear unit activation function have been explored. Compared to shallow networks with a sigmoid activation function, accuracy of diagnosis can be increased from 72% to 86%.

However, accuracy is only one of the benefits of deep architectures. The increased ability to visualize the learning that has taken place means that neural networks are less obscure than previously thought. Examination of a neuron's activation can reveal interesting information about the invariant properties of a PRPD pattern for a given defect type, as well as giving increased confidence in the diagnosis itself.

Acknowledgements

This work was supported by the EPRSC through the Supergen HubNet project EP/I013636/1.

The authors thank Mark Waters and Callum Ferguson for discussions held in the preparation of this paper.

References

[1] E. Gulski and A. Krivda, "Neural networks as a tool for recognition of partial discharges," *IEEE Trans. Electr. Insul.*, vol. 28, no. 6, pp. 984–1001, Dec. 1993.

[2] S. D. J. McArthur, S. M. Strachan, and G. Jahn, "The design of a multi-agent transformer condition monitoring system," *IEEE Transactions on Power Systems*, vol. 19, no. 4, Nov. 2004.

[3] L. Hao, P. L. Lewin, Y. Tian, and S. J. Dodd, "Partial discharge identification using a support vector machine," in *Conference on Electrical Insulation and Dielectric Phenomena (CEIDP '05)*, Oct. 2005.

[4] J. Hunter, P. Lewin, L. Hao, C. Walton, and M. Michel, "Autonomous classification of PD sources within three-phase 11 kV PILC cables," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 20, no. 6, Dec. 2013.

[5] Y. Bengio and Y. LeCun, "Scaling Learning Algorithms towards AI," in *Large-Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds. MIT Press, 2007.

[6] E. Gulski, *Computer-Aided Recognition of Partial Discharges using Statistical Tools*, 1991, Ph.D. dissertation, Delft Univ. Press, Delft, The Netherlands.

[7] S. Rudd, S. D. J. McArthur, and M. D. Judd, "A generic knowledge-based approach to the analysis of partial discharge data," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 17, no. 1, 2010.

[8] V. Catterson, S. Bahadoorsingh, S. Rudd, S. McArthur, and S. Rowland, "Identifying Harmonic Attributes From Online Partial Discharge Data," *IEEE Transactions on Power Delivery*, vol. 26, no. 3, Jul. 2011.

[9] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, Oct. 1986.

[11] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, 1989.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, Jul. 2006.

[13] D. Erhan, A. Courville, and Y. Bengio, "Understanding Representations Learned in Deep Architectures," Oct. 2010, Université de Montréal Technical Report 1355.

[14] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, Apr. 2011.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012.

[16] G. P. Cleary and M. D. Judd, "An Investigation of Discharges in Oil Insulation using UHF PD Detection," in *Proceedings of the 14th IEEE Int. Conf. on Dielectric Liquids (GRAZ)*, Jul. 2002, pp. 341–344.