International Journal of All Research Education and Scientific Methods (IJARESM)
ISSN: 2455-6211, Volume 5, Issue 9, September- 2017, Impact Factor: 2.287

An Algorithm for Power System Fault Analysis based on Convolutional Deep Learning Neural Networks

Florian Rudin[1], Guo-Jie Li[2], Keyou Wang[3]

[1,2,3]Department of Electrical Engineering/Shanghai Jiaotong, University/China

ABSTRACT

This paper discusses the possibility of using deep learning architecture using convolutional neural networks (CNN) for real-time power system fault classification. This work is about fault classification only and not about localization. It aims to classify power system voltage signal samples in real time and determine whether they belong to a faulted or non-faulted state. The data is produced by simulating a simple two-bus power system with a three -phase balanced load. The voltage signal is measured at the beginning of the line between the two buses while the fault occurs at half of the line length between the two buses. In a first step, Wavelet transform is used to extract fault harmonics using db4 Daubechies mother wavelets. A sample window of fixed size is slid over the wavelet detail at decomposition level 4 which seems to be a suitable choice. After normalization, the generated training samples are fed into the CNN for learning procedure. The CNN learns fault features of the power system through training by faulted and non-faulted samples to finally classify samples from a test set.

Keywords: Power System Fault Analysis, Convolutional Neural Networks, Wavelet Transform.

I. INTRODUCTION

As power grids grow large and highly interconnected, they are increasingly often subject to complex dynamics and unpredictable changes. This includes the effects of occurring power system faults. In order to guarantee a satisfactory degree of power quality accordingly, measures must be taken in minimum time following such a fault. Data availability has increased as well, which is an important asset for the fault analysis of today.

There has been researching done about the use of machine intelligence to analyze power system faults. Power system fault features have been used to train several neural network architectures such as probabilistic neural nets in combination with Concordia patterns [1] and multi-resolution analysis using wavelets for power system fault feature extraction has been used thoroughly [2]–[4]. Full automation for detecting power system faults and wavelet decomposition for fault analysis are specifically encouraged [5]. Stacked auto-encoders have been applied for fault analysis using a greedy layer-wise approach to extract features training one auto encoder at a time through unsupervised learning [6]. The researchers used only two hidden layers as feature extractors, which is one reason for the accuracy to be low. Furthermore, the practice of only stacking fully connected layers will likely not provide any good results for complex problems and are an old-fashioned approach to deep learning.

A promising fault detection paper proposed Fuzzy K-NN Logic [7] to solve the problem. While their research is super solid and results are good, there are generalization issues with the Fuzzy approach to classifying data. Another mentionable approach proposed the use of feature energies [8] using Wavelet transform for training neural nets and support vector machines. They distinguish two classification stages, the first indicates the faulted phases while the second determines if ground is involved. They do seem to struggle with the complex decision boundaries during stage one, but the separation of classification stages is valuable and can be helpful for future studies. Facing the complexity of modern power grids, researchers are still struggling with efficiently using machine intelligence technologies.

[1]This work was supported in part by the National Natural Science Foundation of China under Grant 51477098 and National Key R&D Program of China (2016YFB0900504) and State Grid science and technology project (SGTYHT/16-JS-198).

Convolutional neural networks have been widely used in image recognition tasks [9] and are building blocks for the world's best image recognition architectures such as ResNet [10] or GoogLeNet [11]. They use convolution kernels to extract features from data through supervised learning and backpropagation. Their ability for self-learned hierarchical feature extraction for 2D inputs in image detection can potentially should be applicable for 1D signals such as a phase voltage signal. To the author's knowledge, there exists no reference that uses deep learning with convolutional neural networks for power system fault analysis.

It is not surprising that neural networks with only a few layers have never shown any good results for learning complex problems. This is due to their inability of learning a high-level abstraction of the data so they can not generalize well.

Training deep neural networks (many hidden layers) achieves better performance than training shallow networks. They show clear advantages in feature extraction and they have many nice properties such as not falling into bad local optima [12] and many available algorithms that solve problems which stacked non-linearities bring, such as Batch Normalization [13]. Fundamental requirements for training deep networks are the availability of large amounts of data and powerful computers (usually GPU instances) to train them. These requirements are among the reasons why those networks had been discredited for a long time and labeled inefficient to use [14]. This was before the reignition in popularity of neural networks. In 2012 deep neural networks were proposed to be used again in speech recognition [15] and nowadays provide state of the art technology in this domain as well as in image recognition.

Nowadays a vast amount of power system data is available. Also, drastically improved computational speeds using GPU instances for deep learning can be relied on and there are many sophisticated programmatical frameworks that further smoothen the implementation. Training times that were weeks have become days, and what used to need days has become hours of training. The standard method for updating the weights is still backpropagation.

II. BRIEF THEORY

A. *Wavelet transform and Multi Resolution Analysis*

Multiresolution analysis is to analyze a time-domain signal at different frequency resolutions. Wavelet transform allows us to do a wavelet decomposition which decomposes a time domain signal into two signals, one contains the low frequencies (approximation), one contains the high frequencies (detail, wavelet) as shown in Fig. 1. This can be done multiple times to analyze different frequency spectra of the original signal. During each decomposition the signal is downsampled to half the resolution, meaning that the more the frequency window is narrowed down, the more information is lost about time resolution. This is according to Heisenberg's uncertainty principle.

From previous research [16] it is known that Daubechies db4 is a good choice as mother wavelet for feature extraction purposes.

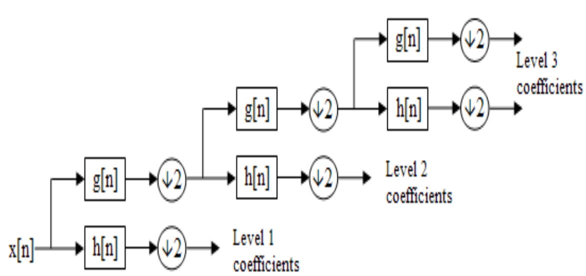


Fig. 1: Wavelet decomposition

B. *Convolutional Neural Networks (CNN)*

CNNs are a class of deep learning neural networks that consist of multiple consecutive convolutional layers usually in combination with a down sampling operation called pooling. Fig. 2 shows a CNN used for image recognition.

Convolutional layers represent filters whose parameters are learned during the learning process of the network. Each filter creates an activation map by striding through the input space and convoluting the data with the filter at each position. The output depth of each layer is according to the number of filters applied. This is a way of learning features from the input signal. The classification part usually is done with a handful of fully connected layers at the end of the convolutional layers.

Pooling is an operation that down samples the input data by combining a certain number of pixels at a time in the activation map. This results in the same number of activation channels but with a decreased spatial size. Its main effect is the diffusion of the exact location of each feature, meaning that information about where exactly on the input signal the feature has been detected is thrown away. For example in image recognition, looking at a picture of a cat, pooling allows a network to still detect the cat on the picture caring less about its exact location on the picture.

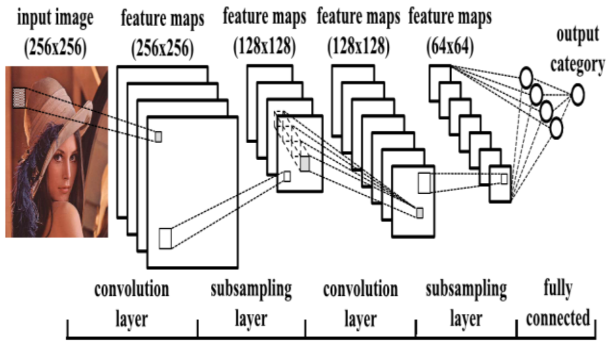


Fig.2: Structure of a convolutional used for image recognition

III. SIMULATION EXPERIMENT

A. *In brief*

The system under consideration is shown in Fig. 3. It is a simple two- bus transmission line system where the two buses are interconnected by a three-phase line. A power source connected to bus 1 is supplying the three-phase impedance connected on the other end at bus 2.

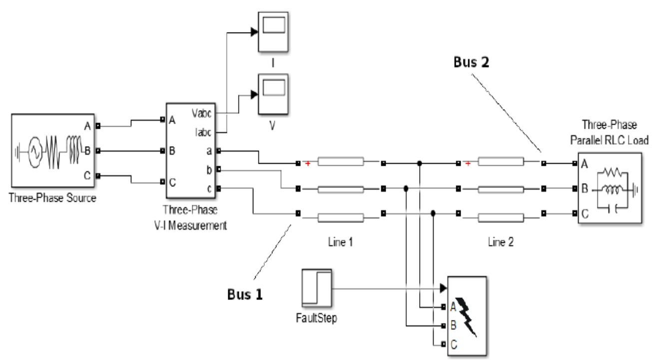


Fig.3: Simulink model the two bus transmission system

Different types of faults such as AB, BC, CA, ABG, BCG, CAG, AG, BG, CG, and ABC are simulated for different fault times.

The simulated data is transformed using Wavelet decomposition and the level 4 details is used that lie within the range of 9.375kHz and 18.750kHz where the fault response seems to have an interesting impact in terms of magnitude and its asymmetric and decaying properties.

A sample window of 256 samples is slid over the extracted signal and thresholds are set such that a signal is classified faulty if the first 0.01 seconds (a half cycle of grid frequency) of the fault is within the sample window.

This generated roughly 50,000 three- dimensional data samples (50,000 for each phase) and labels for faulted and not faulted states with which the neural net could be trained and tested. Prior to training, 20% of the data was reserved for testing.

The proposed method continuously analyzes samples from a three-phase voltage signal and tries to learn how to classify a fault type correctly if one occurs. The fault classification includes the information about which phase is faulted and if ground is involved. A fault label is a four dimensional binary vector f = [A,B,C,G] where values {0, 1} indicate if phases A, B and C or ground G are involved (1) or not (0). i.e. [0 0 0 0] stands for no fault and [1 1 0 1] for ABG fault. For the final classification the following threshold was used:

$$f(x) = \begin{cases} 1, & \text{if } x \in [0.5, \infty] \\ -1, & \text{if } x \in [-\infty, 0.5] \\ x, & \text{otherwise} \end{cases}$$

B. *The neural net*

The network consists of a feature extraction part and a classification part, former being a consecutive stack of convolutions and pooling operations for down sampling. The classification part is a sequence of three fully connected layers with batch normalization and dropout for further performance enhancement. The input depth of the network is 3 channels, one for each phase. A detailed description of the layers used is shown in Table 1. The input depth of the network is 3 channels, one for each phase.

Padding and stride are parameters of how the kernel is applied. The network consists of the following types of layers:

- CONV: A convolution layer.
- MAXPOOL: Max-pooling o = max(i_1, …, i_n).
- RELU: A rectifying linear unit that sets everything smaller than zero to 0 when propagating forward through the network.
- FC: a Fully-Connected layer, connects all activations from a previous layer to the neurons of the FC layer.
- BN: stands for Batch Normalization layer to normalize input distributions between layers.
- DROPOUT: applied to any layer during training it randomly sets a fixed amount of neuron inputs to zero. Prevents co-adaptations of neurons and improves the generalization ability of the network.

C. *Tech specs*

For data generation, MATLAB and Simulink have been used. The neural network was implemented with Py Torch using Python 3.5. It was trained on a GPU instance hosted on floydhub.com.

Table 1: CNN layers, input size is 256x1x3 (3 phase channels)

Layer	Kernel HxWxD	Padding	Stride
CONV16	(3,1,3)	(1,0)	1
RELU			
CONV16	(3,1,16)	(1,0)	1
RELU			
MAXPOOL	(2,1,16)	0	2
CONV32	(3,1,16)	(1,0)	1
RELU			
CONV32	(3,1,32)	(1,0)	1
RELU			
MAXPOOL	(2,1,32)	0	2
CONV64	(3,1,32)	(1,0)	1
RELU			
CONV64	(3,1,64)	(1,0)	1
RELU			
MAXPOOL	(2,1,64)	0	2

CONV128	(3,1,64)	(1,0)	1
RELU			
CONV128	(3,1,128)	(1,0)	1
RELU			
MAXPOOL	(2,1,128)	0	2
FC & RELU	(2048 in,512 out)		
BN	(512 in,512 out)		
DROPOUT	p=0.2		
FC & RELU	(512 in, 64 out)		
BN	(64 in, 64 out)		
DROPOUT	p=0.2		
FC & RELU	(64 in, 4 out)		

IV. SIMULATION RESULTS

Various results are obtained from the simulations using fault samples as shown in Fig. 4 for training and testing. To show that the network is capable of learning anything it is first overfitted with only a single fault type. The network easily converges to zero training loss and the testing accuracy is 100%. It is found that learning occurs with using multiple types of faults during training. Figures 5a) and 5b) show how the network converges for different types of fault samples.

For a network trained with all faults, a per phase classification accuracy of over 85% is reached, see Table 2. Testing accuracy per phase is above 56% for a network trained with all faults only involving phase A, see Table 3. The classification accuracy by fault is very low and strongly overfitted for ABCG or ACG faults, see Table 4.

A network trained with all available data (including no-fault data) shows a devastating 0 % testing accuracy for all samples containing a fault. The network clearly seems to be overfitted by no-fault types because the majority of the samples represent no-fault states. This is shown by a testing accuracy of nearly 100 % for no-fault samples.

Table 2: Classification accuracy per phase for all fault types by involved phase or ground

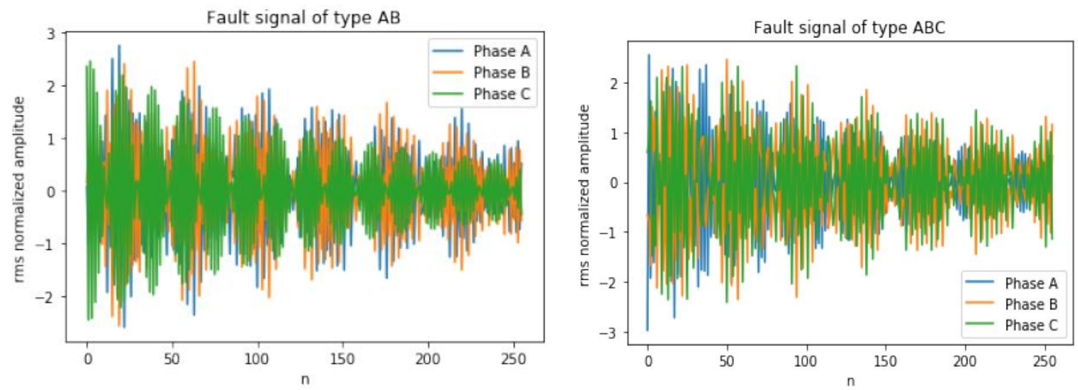
	Classification accuracy
A faulted	85.59 %
B faulted	85.67 %
C faulted	85.58 %
Ground involved	85.42 %

Table 3: Classification accuracy per phase trained with fault samples where A is faulted

	Classification accuracy
A faulted	99.79 %
B faulted	56.51 %
C faulted	56.65 %
Ground involved	57.08 %

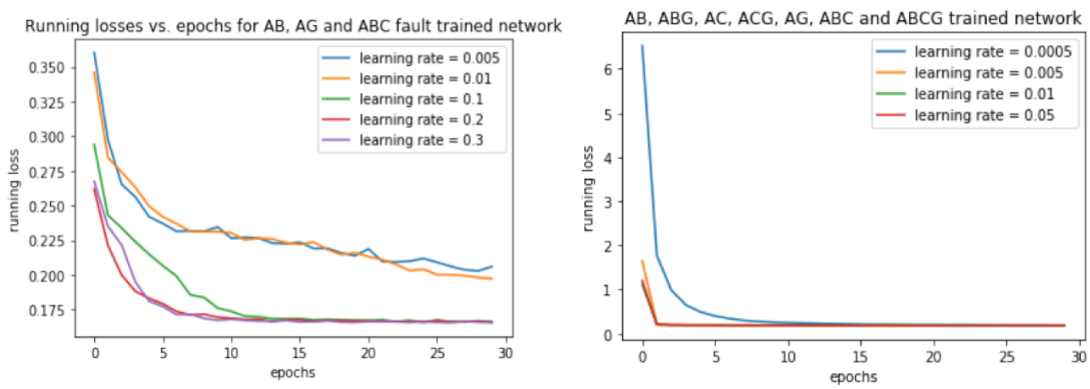
Table 4: Overall classification accuracy trained with fault samples where A is faulted

	Classification accuracy
AG	0.0 %
AC	1.02 %
ACG	17.82 %
AB	0.0 %
ABG	0.0 %
ABC	0.0 %
ABCG	91.19 %



(a) AB fault (b) ABC fault

Fig. 4: Fault samples as fed into the network, each phase RMS normalized



(a) Training with only AB, AG and ABC faults (b) Training with all faults that include phase A

Fig. 5: Epochs vs. losses

V. Discussion

Although there is a sign of learning, the achieved accuracy levels are clearly not satisfactory for use in practice yet. The results could be improved in many ways.

A deeper network architecture could help to extract finer fault details. The network is still pretty shallow with its 4 convolutional layers and might simply not be able to learn the necessary details. Usually CNNs have up to 20 layers, and more modern approaches even more. This might be also a reason why it behaved quite unstable with testing data. Sometimes there were overflows in the output which hints to bad regularization. Also, one could experiment more with system parameters and how to implement performance enhancement layers such as batch norm. It could be interesting to see how ResNet performs on the given task.

Three depth channels are used for the different phases. This idea originates from the intuition that there might be an analogy between a three-phased power signal and an image with three color layers (in image recognition often a depth of 3 is used for RGB images). Both types of data are combined layers of similar data with similar characteristics representing a whole thing. It is yet to be confirmed if it would be better to use one single channel input and train the network one phase at a time, but this would make the spatial dimensions larger and computational speeds might increase instead. It is also believed that using all three phases in three channels will take into account all three phases at fault time when the filters are learned, which can be an advantage for more accurate fault feature extraction. It is observed that individual phase or ground accuracy is generally better than the accuracy of classifying one particular fault correctly, see tables 3 and 4. This could be again a hint that using one network per phase could yield improved performance.

The network was trained using roughly 40’000 samples and tested with 10’000 separate samples. This amount of data seems not enough to assure a satisfactory training effect and testing accuracy. There is a comparatively large amount of no-fault data (roughly 3/4 of all data) for training which might influence overall system accuracy as well and lead to overfitting. Results are bad when using the given no-fault data, which might be improved if no-fault data is about the same number than others. But of course it could be also due to no-fault data looking just too similar to fault data and the network is often not able to find the difference in its current state.

Learning rates ranging from 0.0005 to 0.1 all performed about the same well above 10 epochs during training. The running loss never converged lower than 0.175 per epoch. This threshold of training accuracy could be due to poor data quality. Improved data quality could be achieved by only refining the way of thresholding a sample to be classified as faulted or not, which is currently done rudimentary. Data is currently normalized by dividing each phase signal with its RMS, which might not be the best choice either since input data is relatively un-similar in terms of amplitude. It is generally believed though that normalized amplitudes do not play a crucial role for feature extraction but rather would improve regularization of the network.

It is common to use a training batch size larger than 1 to make use of batch normalization's nice regularization properties. The number of epochs was set to 20 to 30, but more epochs might improve the fitting ability of the network.

This work has only addressed fault classification and not location, the possibility of latter still being subject to further research. Also, it is not only to be researched more on the effects of more network layers but also on how CNNs would perform in more complex systems with challenges such as strong presence of harmonics and system dynamics.

CONCLUSION

It has been shown that CNNs can successfully learn power system faults features and classify those correctly. For certain training scenarios (only faulted test samples), a per phase testing accuracy of over 85% is achieved. This has been validated by a simulation of a two-bus power system with balanced load. The network performs well on three-phase faults as shown in Table 4.

The appropriate setting of hyper parameters of the network used in this paper has yet to be fully understood and adjusted eventually. The indication of successful learning is motivating to pursue further research on the topic, being said that a CNN with a series of consecutive layers of convolutions is not even the most advanced type of deep learning network. Recurrent Neural Networks as well as several Inception architectures with a way bigger depth may have the potential for a better fault analysis performance, especially in a more complicated system with multiple buses and unbalanced loads.

REFERENCES

[1]. S. Mishra and A. Baral,"Classification of power system faults using voltage Concordia pattern feature aided PNN[C]",*2016 IEEE 6th International Conference on Power Systems (ICPS)*, March 2016, pp. 1-6.

[2]. M. Patel and R.N. Patel, "Fault detection and classification on a transmission line using wavelet multi resolution analysis and neural network[J]",*International Journal of Computer Applications*, vol.47, no.22,2012, pp.27-33.

[3]. K. Kasinathan, "Power system fault detection and classification by wavelet transforms and adaptive resonance theory neural networks[D]",*University of Kentucky Master's Theses*, 2007.

[4]. M. Jamil, M. A. Kalam, A.Q. Ansari, M.Rizwan, "Wavelet-FFNN based fault location estimation of a transmission line[J]",*Electrical engineering research*, vol.1,no.3, 2013,pp.77-82.

[5]. A. Borghetti, M. Bosetti, C. A. Nucci, M.Paolone, A.Abur, "Integrated use of time-frequency wavelet decompositions for fault location in distribution networks: Theory and experimental validation[J]",*IEEE Trans. on Power Delivery*, vol.25, no.4, 2010, pp.3139-3146.

[6]. Y. Wang, M. Liu, and Z. Bao, "Deep learning neural network for power system fault diagnosis[C]",*2016 35th Chinese Control Conference (CCC)*, July 2016, pp. 6678-6683.

[7]. S. Vasilic and M. Kezunovic,"Fuzzy art neural network algorithm for classifying the power system faults[J]",*IEEE Trans. on Power Delivery*, vol.20, no.2, 2005, pp.1306-1314.

[8]. N. S. Coleman, C. Schegan, and K. N. Miu, "A study of power distribution system fault classification with machine learning techniques[C]", *2015 North American Power Symposium (NAPS)*, Oct. 2015, pp. 1-6.

[9]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning[J]" *Nature*, vol. 521, no.7553, 2015, pp.436-444.

[10]. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition[C]",*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016,pp.770-778.

[11]. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, et al, "Going deeper with convolutions[C]", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp.1-9.

[12]. G. Swirszcz, W. M. Czarnecki, and R. Pascanu, "Local minima in training of neural networks[J]",*arXiv:1611.06310v2*, 2017, pp.1-12.

[13]. S. Ioffe and C. Szegedy,"Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]",*International Conference on Machine Learning*, 2015, pp.448-456.

[14]. Q. H. Alsafasfeh, I. Abdel-Qader, and A. M. Harb, "Fault Classification and Localization in Power Systems Using Fault Signatures and Principal Components Analysis[J]",*Energy and Power Engineering*, vol.4, no.6, 2012, pp.506-522.

[15]. G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, et al, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]", *IEEE Signal Processing Magazine*, vol.29, no.6, 2012, pp.82-97.

[16]. C. M. Schegan, "A hardware/software platform for fault detection and identification in electric power distribution systems for testing various detection schemes[D]",*Drexel University: Theses, Dissertations, and Projects*, 2008.