Autonomous Fault Detection Using Artificial Intelligence Applied to CLAS12 Drift Chamber Data

August 13, 2018

Christian Peters

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Motivation

- Most crucial elements of a physical experiment?
  - Methods of measurement, e.g. drift chamber at CLAS12
  - Need to be highly precise
  - Essential for success
- Problem: Extreme conditions often lead to faults
  - Distortions in measurement accuracy
  - Have to be detected and filtered out during runtime
- Too much data to be processed by a human
  - An *autonomous* approach of fault detection is required

Motivation

- Emerging field lending itself particularly well to the task:
  - The domain of Artificial Intelligence (AI)
  - Deep Learning, Convolutional Neural Networks (CNNs)
- Goal: Apply methods of AI to the problem of fault detection
  - Experimental context: CLAS12 drift chamber
- Baseline software: deeplearning4j (DL4J) library
  - Will be used to implement the fault detection system

The CLAS12 Drift Chamber

- Subsystem of the CLAS12 particle detector
  - Electron beam hits target inside the detector's center
  - Drift Chamber (DC) is used to measure the results (particle momentum)
- Hierarchical arrangement of multiple wires grouped together as wire chambers
  - Wires are used to detect particle presence
  - Particle hits wire $\rightarrow$ wire gets activated

The CLAS12 Drift Chamber

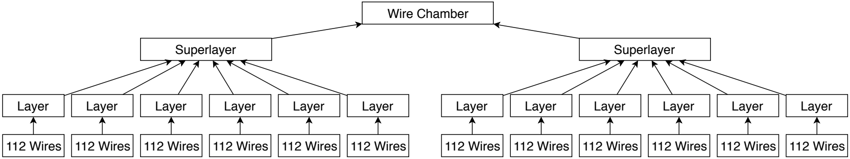
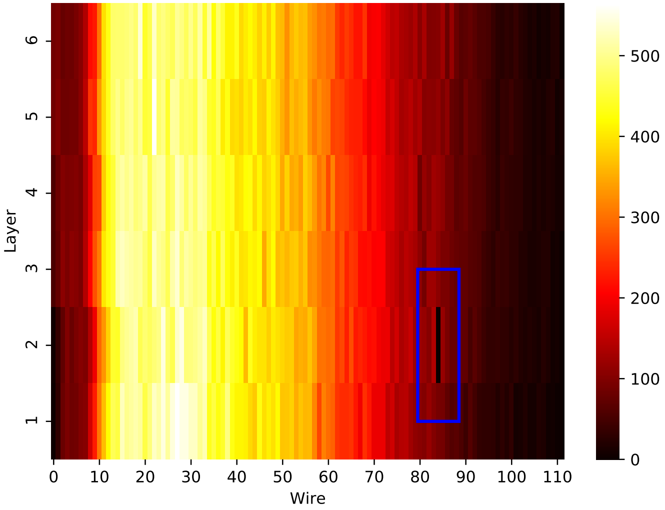


Figure: The hierarchical structure of a single wire chamber.

Drift Chamber Faults

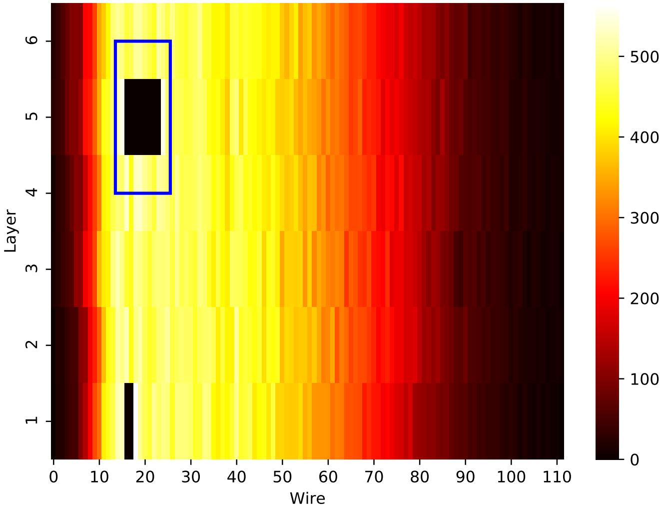
- Drift chamber operates under extreme conditions
  - Huge amounts of radiation
  - Components can get damaged during an experiment
  - Single wires or collections thereof stop working
- Wire activations can be visualized as heatmaps
  - Easier to detect faults

Dead Wire



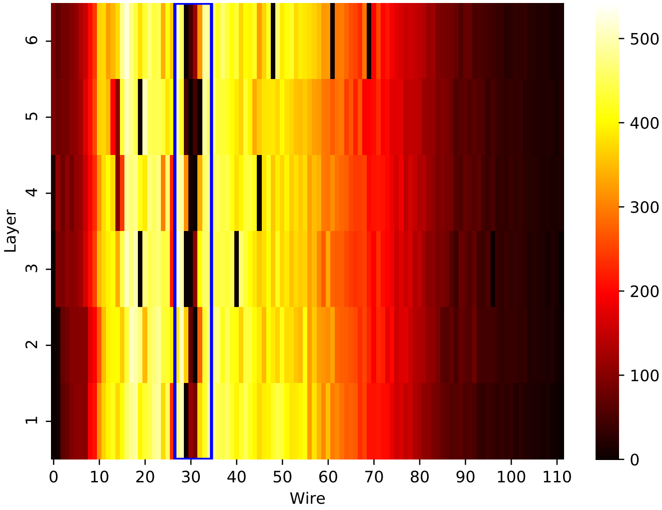
Layer
Wire
1
2
3
4
5
6
0
10
20
30
40
50
60
70
80
90
100
110
0
100
200
300
400
500

Dead Pin

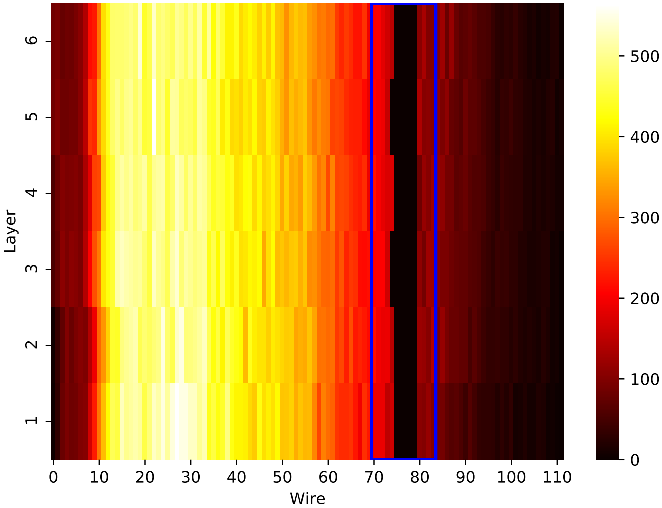


Layer
1
2
3
4
5
6
Wire
0
10
20
30
40
50
60
70
80
90
100
110
0
100
200
300
400
500

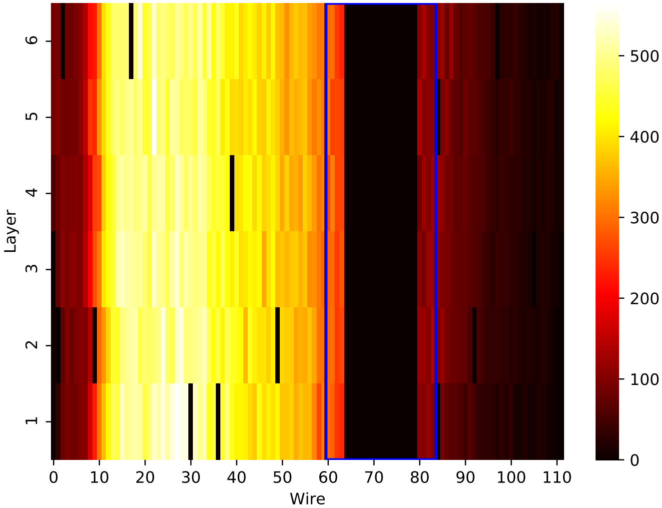
Dead Connector



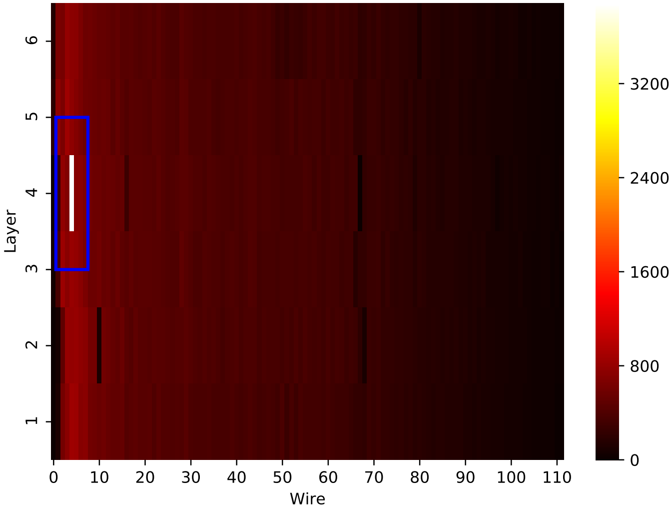
Dead Fuse


Layer
6
5
4
3
2
1
Wire
0
10
20
30
40
50
60
70
80
90
100
110
500
400
300
200
100
0

Dead Channel



Hot Wire



Layer
Wire
0
10
20
30
40
50
60
70
80
90
100
110
1
2
3
4
5
6
3200
2400
1600
800
0

Artificial Neural Networks

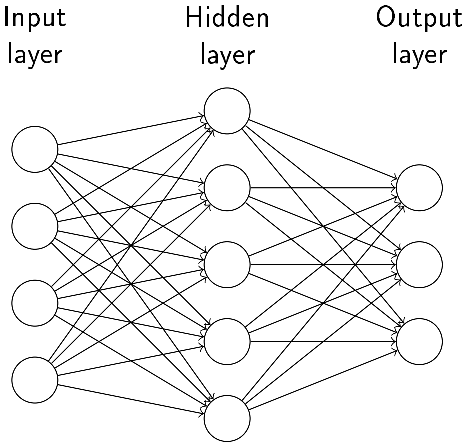


Figure: A common ANN-structure represented by a directed graph.

Artificial Neural Networks

- Class of machine learning algorithms
  - Loosely inspired by biological nervous systems
- Collection of artificial neurons that are connected with each other
  - Enables them to exchange signals along their connections
  - Can be represented by a directed graph
- Usually arranged in layers
  - *Input Layer* collects input signals and passes them on
  - *Hidden Layers* apply transformations to incoming signals and pass the outcomes further into the network
  - *Output Layer* applies a final transformation representing the networks' result
- Goal: Convert input into meaningful output by applying multiple transformations

Modeling Artificial Neurons

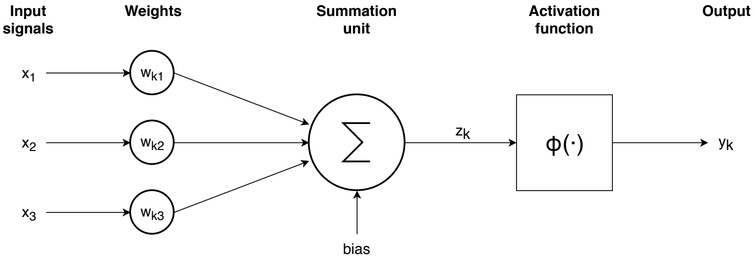


Figure: The components of a single artificial neuron k.

Components of the neural model

- A set of weighted inputs
  - Each input originating from neuron j and traveling into neuron k is first multiplied by a weight w_{kj}
- A summation unit
  - All the weighted inputs are summed and a constant value, the *bias*, is added to yield the result z_k
- An activation function
  - Applies a non-linear transformation $\phi(\cdot)$ to the output of the summation unit
  - This result, called y_k, is propagated further into the network alongside the connections

Y. Bengio. "Practical recommendations for gradient-based training of deep architectures". In: *ArXiv e-prints* (June 2012). arXiv: `1206.5533 [cs.LG]`.

Léon Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks: Tricks of the Trade.* Springer, Berlin, Heidelberg, 2012. ISBN: 978-3-642-35288-1.

Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. PMLR, 13–15 May 2010, pp. 249–256.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pp. 315–323.

Simon Haykin. *Neural Networks and Learning Machines*. 3rd ed. Prentice Hall International, 2008. ISBN: 978-0131471399.

D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *ArXiv e-prints* (Dec. 2014). arXiv: 1412.6980 [cs.LG].

Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. 1st ed. O'Reilly Media, 2017. ISBN: 978-1491914250.

J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: *ArXiv e-prints* (Apr. 2018). arXiv: 1804.02767 [cs.CV].

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *nature* 323 (1986).

O. Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *ArXiv e-prints* (Sept. 2014). arXiv: 1409.0575 [cs.CV].