Fachhochschule Aachen
Campus Jülich

Fachbereich: Medizintechnik und Technomathematik
Studiengang: Scientific Programming

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

Eine Bachelorarbeit von Christian Peters

Jülich, den June 22, 2018

Contents

1 Introduction

2 Deep Learning Fundamentals

2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are a class of machine learning algorithms that are loosely inspired by the structure of biological nervous systems. To be precise, each ANN consists of a collection of artificial neurons that are connected with each other. The neurons are able to exchange information along their connections. A common way to arrange artificial neurons within a network is to organize them in layers as depicted in figure 2.1.

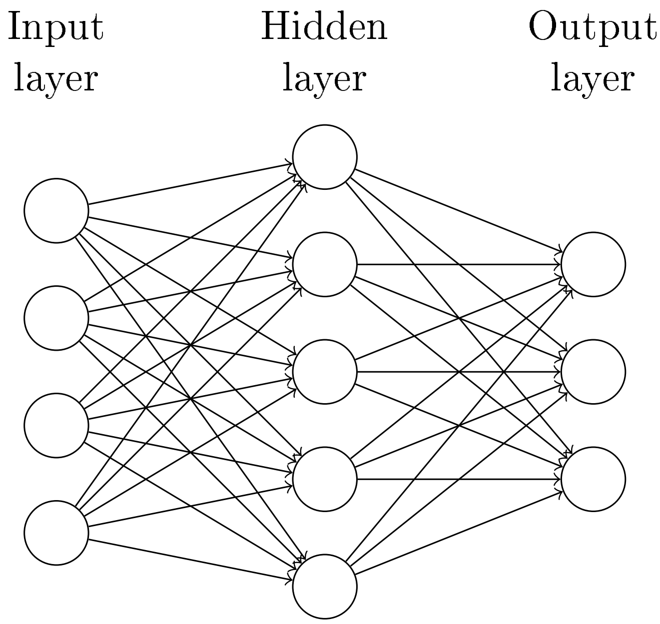

Figure 2.1: The structure of a simple ANN. The nodes represent the neurons, the edges represent their connections, also indicating the flow of information.

When an artificial neuron receives a signal on some of its incoming connections, it may elect to become active.[1] In this state it also influences all neurons it has an outgoing

[1] The details of this process are further illustrated in section 2.1.1 on the next page.

connection to by passing a signal along their channel. Those other neurons in turn may also elect to become active - this way a signal can propagate through the network along the connecting edges.

Usually, each ANN consists of at least one layer of neurons that is responsible for receiving signals from the environment - we call this an *input layer* (see figure 2.1 on the preceding page). When these neurons receive a signal from the environment, they propagate it to their connected neighbors in the next layer. This process repeats until the *output layer* is reached. The neurons in this layer represent the output of the whole network. Each layer in between is called a *hidden layer* because there is no direct communication between the neurons in this layer and the environment.

The goal behind this procedure usually is to convert an input signal into a meaningful output by feeding it through the network. If the network is able to detect relevant features or patterns in the input signal, it can be used to perform tasks such as classification or regression (i.e. approximate discrete or continuous functions). In order for this to be possible, some kind of learning has to take place which enables the network to capture the essence of the data it is confronted with. We will take a further look at these aspects as well as the mathematical model of a neural network in the following sections.

2.1.1 Modeling Artificial Neurons

To fully understand how each neuron processes the signals it receives, it is necessary to develop a mathematical model that describes all the operations taking place. The following descriptions are partially based on the explanations that are provided in [Hay08].[2] As shown in figure 2.2 on the next page, each artificial neuron basically consists of three components:

1. **A set of weighted inputs:** Each connection that is leading into the neuron has a weight w_{kj} associated with it where k denotes the neuron in question and j denotes the index of the neuron that delivers its input to the current neuron k.[3] The signal that passes the connection is multiplied by the related weight of that connection before arriving at the next component.

2. **A summation unit:** This component adds up all the weighted signals that arrive at the neuron as well as a bias value b_k that is independent of the inputs.

[2] See chapter I.3: *Models of a Neuron* for more details.

[3] There might arise the question why the indexing of a weight from neuron j to neuron k is w_{kj} and *not* w_{jk}. This is the case because the weights are usually stored in matrices where each row corresponds to a neuron k and each column corresponds to an input j which allows for much faster computations by heavily utilizing matrix-multiplication.

3. **An activation function:** The activation function $\phi(\cdot)$ takes the output of the summation unit and applies a transformation to it that is usually non-linear. The value of the activation function is the output of the neuron which will travel further through the network alongside the corresponding connections. In section 2.1.2 on the following page, a more detailed explanation of activation functions as well as some commonly used examples will be provided.

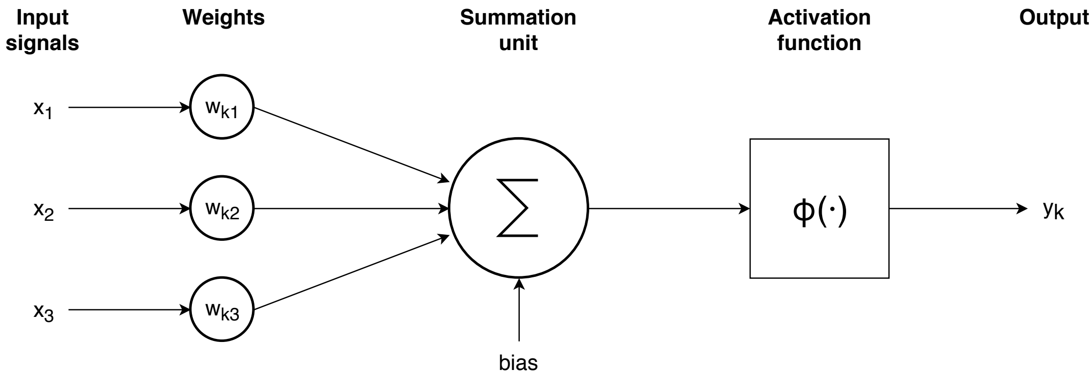

Figure 2.2: The components of a single artificial neuron. This neuron k receives three input signals that are first multiplied by the associated weights, summed up including a bias and then fed into an activation function that will determine the ouput signal.

As a result, the output of the summation unit of a particular neuron k with n input signals x_j can be described by the following equation:

$$z_k = \sum_{j=1}^{n} x_j \cdot w_{kj} + b_k \tag{2.1}$$

where b_k denotes the bias term of neuron k and z_k describes the result of the summation unit.

As a consequence, the output signal y_k of neuron k can be computed by applying the activation function $\phi(\cdot)$ to the output of the summation unit which can be described by the following expression:

$$y_k = \phi(z_k) \tag{2.2}$$

2.1.2 Activation Functions

2.2 The Multilayer Perceptron

2.3 Deep Networks

3 Convolutional Neural Networks

4 The CLAS12 Particle Detector

5 Implementing and Testing a CNN-Model in DL4J

6 Discussion

7 Conclusion

Bibliography

[Hay08] Simon Haykin. *Neural Networks and Learning Machines*. 3rd ed. Prentice Hall International, 2008. ISBN: 978-0131471399.

List of Figures