# A Brief on CNN-based Particle Incident Detection

Erjin Bao
Institute of Automation, Chinese academy of Science

September 15, 2020

## 1 Algorithm Design and Method Descriptions

### 1.1 Network Structure

Advances in neural networks have brought breakthroughs in various domains like Computer Vision and Natural Language Processing. As an efficient composition with weighted additions and pointwise nonlinearities, the method has prevailed against most of the traditional algorithms in pattern recognition tasks. In our experiment, we introduced a multi-layer convolutional neural network to process time-sensitive signals from detector outputs. Based on the Physics nature that a detector signal (PMT outputs) do only have a local correlation with recent incoming particles (photons incidents), we chose convolutional setups with a moderate total width coverage. From a view of information, a broader coverage by an output neuron's receptive field on its related area will contribute to higher accuracy, while redundant connections could only lead to excessive computation and lower processing speed. As a result, a wise setup should try to cover all relevant signal areas exclusively while cutting off every connection unnecessary, balancing the speed and accuracy from the two mentioned effects.

A similar trade-off also exists in the depth of the neural network. A deeper network could provide more complex combinations to all detected features, and the computation complexity of result calculation changes linearly with the depth. With the existence of advanced structure like residual connection and strategies like batch-norm, depth in training is never a problem. However, to cut the computation in massive experimental data, a network should be relatively shallow. Detector signals are relatively similar and peak-shaped. Such a simple pattern does not require many layers to recognize. A good design for the task is a network with 4 to 6 layers. Deeper structures may bring a slight improvement in precision but will introduce a considerable increase in processing cost.

### 1.2 Processing Workflow

The complete workflow of data processing consists of two stages, data training and predicting. Data training is a task of supervised learning. Based on paired data examples, the goal is to find an efficient mapping from detector waves to particle incidents with back-propagation methods. In the training process, a loss function judges the difference between training outputs and their referenced truth, and the training process is to minimize the loss. The product of network training is a function that creates desired outputs from inputs. By directly putting waveforms as inputs, one can get demanded outputs from a trained network in prediction.

## 1.3 Loss Function

A critical issue matters the network training is the loss function. Particle incidence only happens in a small proportion of time channels within a recorded sequence, which means the outputs and related truth are always sparse. While operating with ineffective loss functions, training processes always ends up in local optimal of constant output due to the sparsity. Also, in practice, the form of loss function influence heavily on the final prediction performance.

A well-designed loss function should meet two requirements as follows: it should handle the sparsity, and it should give a fair judgement to the output in training. A qualified loss function could work on either sequence values or normalized distributions. In pursuit of time precision, a good algorithm should encourage correct predictions in the close neighbourhoods of their corresponding truth, while punishing outputs that are incorrect or inaccurate in time. While working on the sequence outputs, such a mechanism is easy to implement but hard to arrange appropriately. It is hard to find a fair arrangement of punishment weight matching all circumstance. On the other hand, statistical distances and divergences could assess the difference in distribution but are hard to implement in a trainable form for backpropagation.

To tackle the mentioned difficulties, efforts of our work have established feasible, robust solutions with guaranteed performance and convergence for each case of processing. We will describe the details of the method in the following section.

### 1.3.1 Value-based Reconstruction Loss

To cope with sparsity in value-based processing, we have introduced a reconstruction procedure to build up a waveform based on its corresponding particle incidences. By introducing a reverse process after neural network computing, one could define the loss function on a denser wave domain instead of the original incidence domain. Expression of the reconstruction is arbitrary but should fit the paired relationship expressed by data samples. Typically, one should add a training process to fit the parameters in construction expressions before training the predicting neural networks.

### 1.3.2 CDF Wasserstein Loss

A direct implementation of judgement metrics into loss function is often difficult. To deal with sparsity and encourage time accuracy, we established the loss function on the cumulative distribution functions(CDF) of normalized outputs. The form of cumulative sums could build a connection between sequence prediction and its previous values, making optimization through history possible. Moreover, the first Wasserstein distance between two 1D distributions is equivalent to the norm-1 difference of two corresponding CDFs. The optimal transport amount described by this metric is a reliable measurement standard to all distribution differences, including cases in our application.

## 1.4 Method Details and Results