Regadío Carretero, A., Esteban, L. & Sánchez Prieto, S. 2021, "Unfolding using deep learning and its application on pulse height analysis and pile-up management", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1005, art. no. 165403.

Available at https://doi.org/10.1016/j.nima.2021.165403

Unfolding using deep learning and its application on pulse height analysis and pile-up management

Alberto Regadío[a,*], Luis Esteban[b], Sebastián Sánchez-Prieto[c]

[a] *Department of Space Programs, Instituto Nacional de Técnica Aeroespacial, 28850 Torrejón de Ardoz, Spain*
[b] *WiYo Technologies, Caléndula 95, 28109 Alcobendas, Spain*
[c] *Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain*

Abstract

Traditionally, electronics for pulse processing can be modeled as linear transfer functions. In contrast, due to the fact that artificial Neural Networks (NNs) are generally non-linear systems, their behaviour against noise is significantly different as in linear systems. We take advantage of this non-linearity to achieve acceptable Signal-to-Noise Ratios (SNR) with a extremely short shaping time. This article shows an approach to a concrete NN named U-net as pulse shaper. It filters the pulses and return them unfolded solving the pile-up problem, and even estimates the height of the pulses when there has been saturation in the detector. In this article, the NN architecture and results using simulated pulses and real pulses from scintillators are shown. The results clearly show the effectiveness of the approach.

Keywords: Digital pulse processing, Instrumentation, Unfolding, Pile-up, Deep Learning, Neural Networks

1. Introduction

When particles interact with detectors, pulses of current or charge are generated. In order to be analyzed, these pulses are converted to voltage at the output of a preamplifying stage and shaped afterwards. The ideal shape for a detector depends on the shape of the pulse coming from the preamplifier and the noise type of the entire system. Thus, specific techniques are used to synthesize various shapes to maximize their Signal-to-Noise Ratio (SNR) [1, 2].

One of the most common type of noise in spectroscopy systems is white noise, whose spectrum is the same in all frequencies. According to [3–6] the impact of this type of noise in measurements is inversely proportional to the shaping time. Thus, a common practice to mitigate it, is making it longer with the consequent risk of pile-up. As it is known, the pile-up problem is twofold. On the one hand, the pulse processing complexity must be increased to get the information (e.g. the height) of the incident particles

*Corresponding Author
Email addresses: `regadioca@inta.es` (Alberto Regadío), `luis.esteban@die.upm.es` (Luis Esteban), `sebastian.sanchez@uah.es` (Sebastián Sánchez-Prieto)

contained in the pulses. On the other hand, the shaping stage can become saturated, thus invalidating the height measurements.

When pile-up occurs in detectors, many algorithms detect and discard the piled-up pulses with the consequent loss of information. However, there are algorithms that try to analyze the signals even when they are piled-up (e.g. [7, 8]). Apart from these algorithms, a common technique to deal with pile-up is unfolding.

The unfolding (or deconvolution) technique allows the transformation of the input digitized signal $x[n]$ into a unit impulse $\delta[n]$ of length equal to one in the discrete-time domain (see [9] and the references therein), according to the following equation:

$$x[n] * h[n] = \delta[n-d],\ d \in \{0, 1, 2, \ldots\}\,, \tag{1}$$

where $h[n]$ is the transfer function of the shaper that acts as unfolder and d is the delay of the unit impulse in cycles.

Despite the fact that using unfolders is the optimal shaping to manage the pile-up where the height of the pulse is obtained directly, unfolding is not always used because its short shaping time increases the effect of the white noise [4]. To avoid this problem, one option is to replace the shaper whose transfer function is $h[n]$ by a non-linear shaper, which would be able to unfold pulses while keeping white noise low.

However, the design of a non-linear filter can be tedious. Neural Networks (NNs) usually have non-linear response and are automatically configured from input and output patterns. For these reasons, this article proposes the use of NNs to overcome the problems of designing a non-linear filter. In addition, the use of these nets, as they have a non-linear response, could allow to improve the S/N ratio beyond what a linear system allows. Figure 1 shows a diagram of the behaviour of each type of shaping against noise and pile-up.

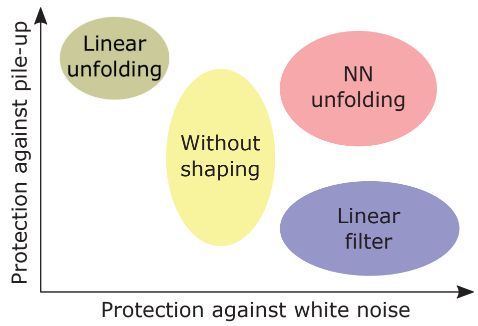

Figure 1: Diagram of the behaviour of each type of shaping against noise and pile-up. Without processing, the pile-up probability depends on the pulse length.

Noise filtering using NNs has already been applied in fields such as electrocardiograms [10] or automatic speech recognition [11]. The use of NNs and deep learning on particle detectors and spectroscopy is not new, they have been used for alpha/gamma [12] and gamma/neutron [13–15] discrimination, the three last using convolutional NNs. These nets have also been used for nuclide identification from radiactive facilities [16], accelerators [17] and cosmic rays [18, 19]. The cited articles have in common that the presented NNs allow the identification of particles. However, the shaping of pulses from particle detectors using deep NNs has not yet been used to our knowledge.

This article shows the topology and training of a neural network that is used to perform unfolding, eliminating the pile-up problem without losing sight of the SNR. The presented NN, apart from unfolding while increasing the SNR, has a series of characteristics that make the training process more laborious. One of them is that the network should learn how to approximate the height of the pulse chopped when the preamplifier is saturated. Another is the lack of homogeneity both in the arrival time and in the height of pulses.

The rest of the paper is structured as follows: Section 2 explains both the NN architecture and training. The results of using these NNs with simulated and real pulses are exposed in Section 3. Finally, Section 4 summarizes the conclusions of this work.

2. Pulse shaper architecture

When dealing with a problem involving deep NNs and signal processing, two options are typically chosen: (a) generate a spectrogram of the pulse and process it as if the signal were a two-dimensional image; (b) process the signal as a temporal series (one dimension). For simplicity and because the results have not been significantly improved using a spectrogram, this last option has been chosen in this work.

The first option considered was to implement the filter using Fully Connected (FC) layers, such as a multilayer perceptron. However, we have used Convolutional Neural Networks (CNN) layers because in the latter, neurons in one layer do not connect to all the neurons in the next layer, saving resources and training time. In CNNs, each set of neurons focuses on one part of the signal and analyzes a specific feature.

Once all the features of the input signal are detected, the next step is to get the desired signal based on them in the same way as in Deep Convolutional Inverse Networks [20]. After trying with simple NN topologies without success, U-Net [21] was chosen because it is already used in signal filtering as Wave-U-Net [22]. Specifically, this first architecture has been taken as a model. However, since we are working with a time dimension, the central core of our NN is a Long Short-Term Memory (LSTM) layer, that works better in time-processing. After this layer, a dense linear layer as in [23] is placed. A complete topology of the NN is shown in Figure 2. The concatenation of the decoder part with blocks from the left (grey horizontal arrows) significantly improves the obtained results.

This NN was trained with a sequence of samples that contains pulses at random time intervals and that can be piled-up. This sequence of samples was mixed with white noise. The target output was the unfolded pulses (i.e. Dirac delta pulses). Once trained, it was tested with sequences of samples different from the

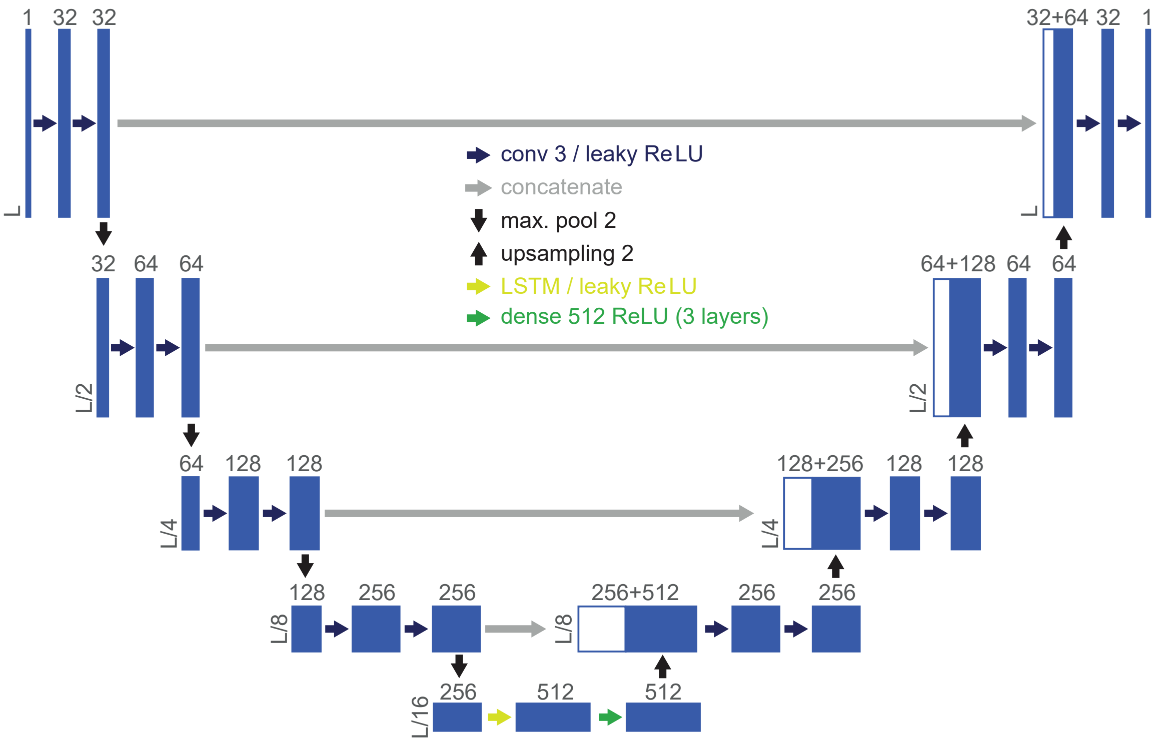

Figure 2: U-net architecture (example for $L = 2048$ samples). Each blue box corresponds to a layer. The number of channels is denoted on top of the box. The window of the signal is provided at the lower left edge of the box. White boxes represent copied features using concatenation. The arrows denote the different operations.

training set.

The implementation of the NN was programmed in Python using Tensorflow [24] and Keras [25] packages[1]. The pulse processing has been performed off-line. However, the size of this NN (3,274,305 units) implies that it can also be inserted in an embedded computer with enough memory and carry out the processing on-line. For comparison, an alternative linear pulse shaping was performed using the `scipy.signal` package [26]. Concretely, pulses were also unfolded using the `deconvolve` function. The pulse height to create histograms was measured using the `find_peaks` function in all cases.

3. Results

3.1. Results with simulated pulses

In this section, we used as source step or Heaviside pulses whose pulse height is a random uniform distribution between 0 and 1. A complete configuration of the simulation is shown in Figure 3.

[1] The code of this U-net is accessible online via: https://github.com/arc140181/unet2pulseprocessing. It has been tested in the Spyder (www.spyder-ide.org) environment.

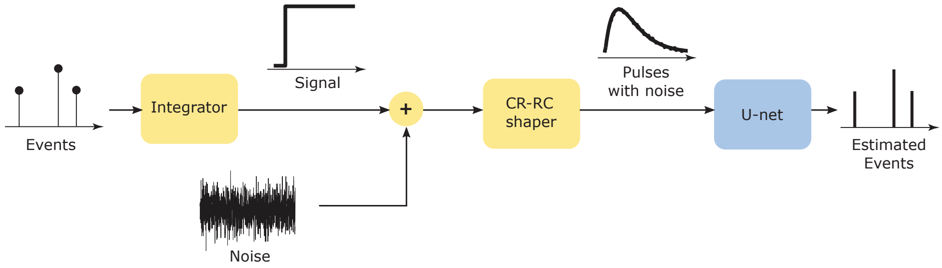

Figure 3: Generation of the input pulses and configuration of the simulation.

As decided in Section 2, pulses were generated at random time intervals and some of them are piled-up. These pulses were mixed with white noise that, as explained in Section 1, affects unfolders the most. The white noise was created with a random number generator whose amplitude was set to 0.05. So, the SNR is equal to 20 in both training and testing data. Afterwards, the noisy pulses were shaped to CR–RC ones whose parameters are $\tau_1 = 1$ μs, $\tau_2 = 0.1$ μs [1, p. 630], mean count rate that varies from $10^4$to $2 \cdot 10^5$ s^{-1} and sampling frequency equal to 50 MHz (20 ns). A sample of this specific train of pulses is shown in the top panel of Figure 5. During the training process, these noisy CR–RC pulses were used as input of the U-net whereas unfolded pulses without noise were used at the output.

The loss function to train the network chosen was the mean squared error, concretely:

$$J = (x - x^*)^2 \tag{2}$$

where x is the output signal and x^* the desired output signal. A total of 20 training epochs have been used with Adam optimizer. Figure 4 shows the loss function along the training process. Analyzing this Figure, we conclude that it converges and there is no overfitting. During each of these epochs, a set 320 sequences of 1024 samples each was used. The pulse length is 128 samples. When the number of samples approaches to the pulse length, the loss function is lowered only up to a point. In contrast, when the length of the training samples is in the order of 16000, the training process is dramatically increased. For the validation process, another set of 80 sequences also of length equal to 1024 was used. In each of the sequences, pulses with an uniformly distribution amplitude between 0 and 1 were randomly generated. In the same way that training sequences, these pulses can give rise to overlaps that saturate the signal when it is greater than 1.

The training process took 690 seconds on Google Colab with Graphics processing unit (GPU) enabled. As the sequences have been generated by simulation, we can obtain as many as we want to improve the training of the NN at the cost of increasing the training time. However, an additional application of this filter may be to regenerate the shaper when the features of the detector have changed as a consequence of radiation, as for example in [27] as it can be the case with silicon detectors installed on payloads. Therefore for these experiments a tradeoff between training time and performance was chosen.

In order to evaluate the performance of the proposed NN, a set of preliminary simulations in time-domain

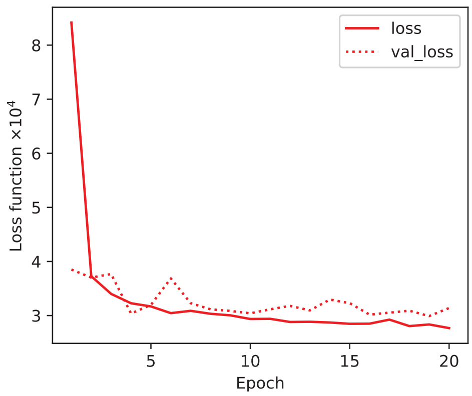

Figure 4: Training process of the U-net whose output are depicted in the bottom panel of Figure 5. The results obtained with the training pulses (solid line) and validation pulses (dotted line) were evaluated with the loss function of Eq. (2). We can observe that both functions decrease at the same time.

were carried out. An example of such results is shown in Figure 5. Note that the pulse height is detected even when the signals are piled-up. It even approximates the height of the pulse when, due to pile-up, the input signal is saturated. The threshold of the pulses was set to 0.03.

As mentioned in Section 1, this NN is a non-linear system, so the noise equations to calculate the noise impact or the Equivalent Noise Charge (ENC) [4–6] cannot be applied. Apart from comparing individual pulse heights as in Figure 5, an alternative method to evaluate the NN is to compare the Full Width at Half Maximum (FWHM). In this article, the FWHM has been calculated applying the function `peak_widths` from the `scipy.signal` package [26] on the histograms with the parameter `rel_height`=0.5. In Figure 6, histograms generated with the proposed NN, with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1, 1, 1, 1, 1)$, with linear unfolding (using `deconvolve`) and without filtering are shown. In this test, a low pulse arrival rate was used to avoid pile-up and thus the effects on FWHM due to white noise filtering were observed.

It can be seen that the FWHM is lowered when the proposed NN is used with respect to the CR–RC shaper and the CR–RC + FIR shaper. This has also been confirmed for other types of noise such as $1/f$, but has not been included here for brevity. For $1/f^{\alpha}$ noise type, where $\alpha > 1$, the proposed NN begins to lose efficiency, the FWHM increases and the training process does not converge as quickly. It must be taken into account that for a NN to be protected against a type of noise, it should be trained with that type of noise as was performed in [28].

In order to test the NN dealing with pile-up, we increased the pulse arrival frequency $\times 10$ to get a pulse rate similar to that shown in Figure 5. The resulting histograms are depicted in Figure 7. We can observe that using the proposed NN we obtain the lowest FWHM.

To measure the performance of the NN compared to the others filters, we generated a set of pulses

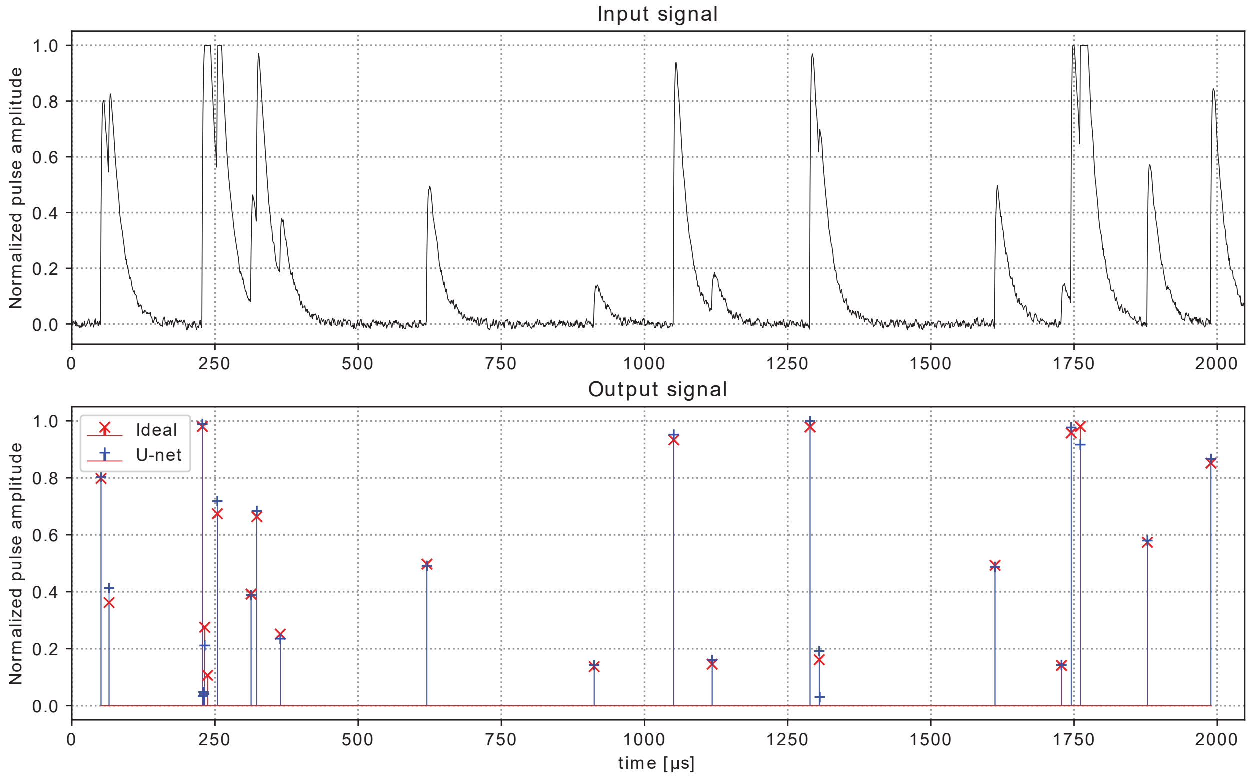

Figure 5: Top panel: Input CR–RC pulses from a test sample with random heights and arrival time; the saturation value was normalized to 1. Bottom panel: Height of the top panel pulses before adding noise (red) and pulse height obtained with the U-net of Figure 2 (blue). In both panels the sequence was sampled at 1 μs/sample.

of equal height. These pulses can be piled-up depending of the pulse rate arrival. We also used the four methods to detect and measure the pulses included in this Section. The results of the number of pulses detected with each method is presented in the first half of Table 1. We observe that the linear unfolder without noise and the NN in any case are the shapers that allow to detect the most pulses. The FWHM of the histogram in number of channels is presented in the second half of Table 1, we see that the FHWM of CR–RC and CR–RC + FIR remains lower than unfolders, but this is because the formers do not take many pulses into account due to the high pulse rate and the preamplifier's saturation. We also observe that the FWHM of the linear unfolder explodes as the noise is increased. Therefore, we can conclude that the NN take into account all the pulses and keeps the noise effect low to calculate the pulse height accurately. Similar results are obtained with trapezoidal, cusp-like and triangular instead CR–RC shapers.

3.2. Results with pulses from a scintillator

Finally, a group of tests to check the proposed NN with real pulses were performed. The main objective of these tests is to check that the NN works and try to improve the results obtained with a linear shaper.

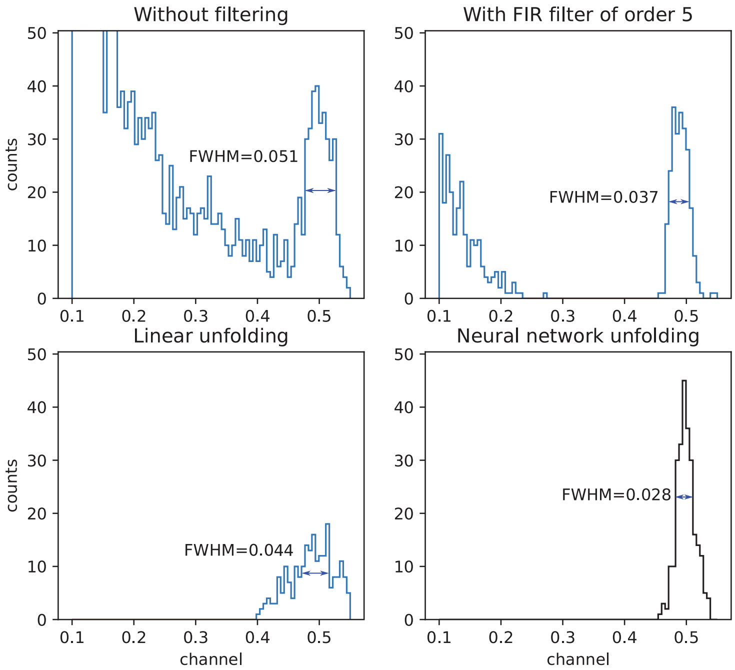

Figure 6: Histograms for a low pulse arrival. Top left panel: without filtering (CR–RC pulses); top right panel: with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1,1,1,1,1)$; bottom left panel: with linear unfolding (using `deconvolve`); bottom right panel: with the NN unfolder.

The pulses were collected in the Radiation Physics Laboratory located in Santiago de Compostela University (Spain) using a scintillator. A diagram of the detection chain used in the experimental test is shown in Figure 8. The scintillator model of NaI is 1M1/1.5 and worked at +475 V, with an integrated preamplifier PA-12. The amplifier N968 (with a shaping of 2 μs and gain $\times 14$ was connected to a Digital Phosphor Oscilloscope Tektronix TDS 3014B. An amount of 500 points were taken for each pulse at a frequency of 1 GS/s.

This oscilloscope performs the function of data acquisition system, receiving the raw data from the amplifier and storing it in a laptop. The resolution of the signal amplitude is 8 bits (256 levels) for a range between -5 and 5 V. The scintillator received radiation from a source of ^{22}Na whose activity is 105 kBq and produces a peak at 511 keV and from a source of ^{137}Cs whose activity is 8.71 kBq and produces a peak at 661.6 keV. The raw data was stored in text files, this allows data reusing without recapturing new samples and ensures that changes in the results obtained during the test are exclusively due to the pulse processing.

To train the NN, a set 400 different pulses distributed throughout a sequence of 204800 samples were filtered. These samples have been divided into 200 sequences of length 1028 each. This has been done so that each sample is long enough to accurately fit the weights of the LSTM layer, but not long enough to cause the count time to increase excessively as explained in Section 3.1. However, tests have been carried out in a range of lengths between 2048 and 8192 and no significant variations have been observed, neither in

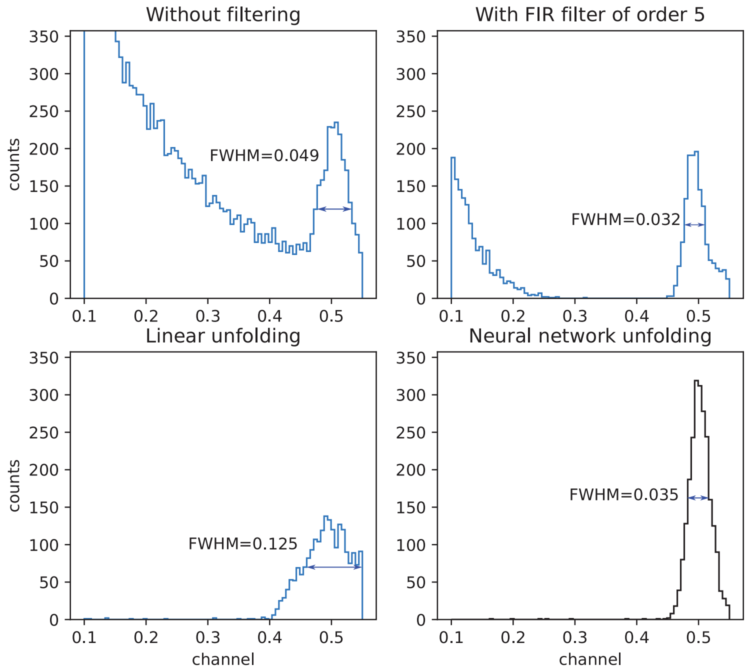

Figure 7: Histograms for a high pulse arrival. Top left panel: without filtering (CR–RC pulses); top right panel: with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1, 1, 1, 1, 1)$; bottom left panel: with linear unfolding (using `deconvolve`); bottom right panel: with the NN unfolder.

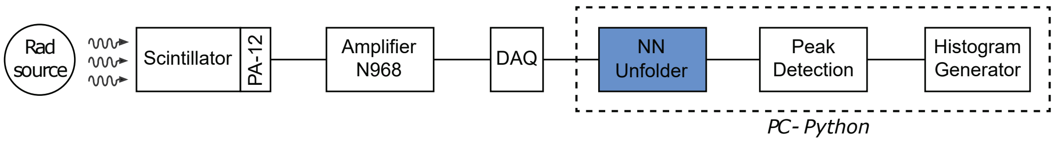

Figure 8: Diagram of the detection chain used for the experimental test.

the computation time, nor in the result. The number of epochs was 200 and the optimizer used was Adam.

One of the drawbacks when training the NN with these samples was that since the NN is non-linear, there must be pulses of the entire spectrum of amplitudes. With a uniform range of amplitudes, the pulses used in Section 3.1 were generated. However, the height of pulse from particle detectors are not uniform, so a sufficient number of pulses from each height is needed for the network to be trained correctly. This set the minimum number of sequences. Fortunately, the regions of interest are usually determined by peaks, so that in these regions the number of samples is assured.

Once trained, a different sequence with 10500 pulses was filtered. For comparison purposes, the same histograms that in Section 3.1 were generated. They are depicted in Figures 9 and 10. As it can be observed, the non-linearity of the NN hardly distorts the histogram. Besides, the achieved FWHM is in the order FIR filters but with the advantage that the NNs solve the pile-up problem much better than the FIR because the former generate unfolded pulses.

	Noise amplitude											
	0				0.05				0.10			
Pulse rate	(1)	(2)	(3)	**(4)**	(1)	(2)	(3)	**(4)**	(1)	(2)	(3)	**(4)**
$1 \cdot 10^4$ s^{-1}	1.000	0.988	1.000	**1.000**	0.976	0.968	1.000	**1.000**	0.995	0.991	1.000	**1.000**
$2 \cdot 10^4$ s^{-1}	0.979	0.967	0.997	**0.999**	0.979	1.000	1.000	**1.000**	0.889	0.877	0.983	**0.984**
$3 \cdot 10^4$ s^{-1}	0.959	0.931	0.997	**1.000**	0.835	0.812	0.998	**0.999**	0.829	0.814	0.977	**0.980**
$4 \cdot 10^4$ s^{-1}	0.936	0.896	0.993	**1.000**	0.741	0.702	0.994	**0.999**	0.716	0.677	0.970	**0.975**
$1 \cdot 10^4$ s^{-1}	6	7	6	**7**	25	19	47	**21**	33	26	90	**38**
$2 \cdot 10^4$ s^{-1}	6	11	6	**7**	21	22	51	**25**	31	32	99	**41**
$3 \cdot 10^4$ s^{-1}	—	—	6	**7**	—	—	51	**26**	—	—	92	**51**
$4 \cdot 10^4$ s^{-1}	—	—	6	**7**	—	—	44	**25**	—	—	91	**46**

Table 1: Top values: Fraction of number of pulses detected for each method: (1) without filtering (CR–RC pulses); (2) with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1,1,1,1,1)$; (3) with linear unfolding (using `deconvolve`); (4) bottom right entries (in **bold**): with the NN unfolder. Bottom values: FWHMs $\times 0.001$ of the histogram generated. Simulation carried out along 240960 time steps (4.8192 ms)

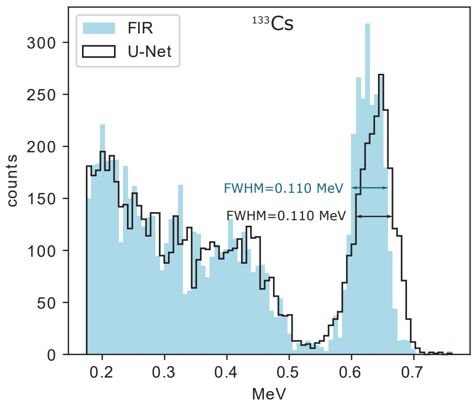

Figure 9: Histograms and result of GA for ^{137}Cs. Left panel: with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1,1,1,1,1)$. Right panel: with the NN unfolder.

4. Conclusions

We have created a specific type of U-net that filters the pulses from particle detectors (using non-linear filtering), returns their height, corrects the pile-up and even estimates the height of the pulses when there has been saturation in the detector. According to results, when there is no noise, the number of pulses detected is the same as that of the optimal pile-up processing: the linear unfolder. When this noise is increased up to a SNR equal to 10, the NN detects 97.5% of the pulses, a value similar to that of the unfolder. This pulse detection occurs without a considerable lowering of its resolution (expressed in this article as FWHM)

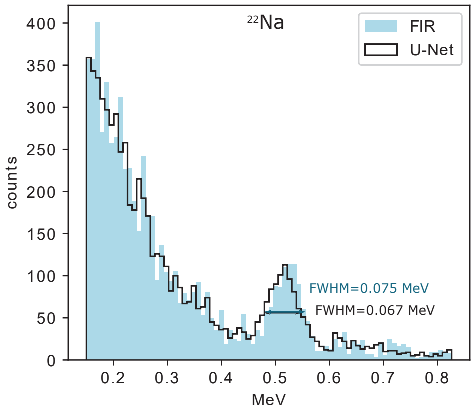

Figure 10: Histograms and result of GA for ^{22}Na. Left panel: with a FIR filter whose transfer function is $h[n] = \frac{1}{5}(1, 1, 1, 1, 1)$. Right panel: with the NN unfolder. In each of them the `peak` function was used to calculate the pulse height.

with respect to other methods such as FIR shaping or linear unfolding. Thus, when there is no pile-up, the U-net has a resolution similar to a FIR filter. However, when pile-up occurs, the resolution of the FIR filtering drops dramatically and the resolution of the U-net is similar to that obtained with the unfolder. When the noise is increased up to an SNR equal to 10, the resolution of the latter becomes up to 50% that obtained with the U-net. This has been possible due to the non-linearity presented by filters based on NNs. The architecture presented here is more flexible than a simple linear shaper. On the whole, this network provides a superior performance compared to more traditional shaping methods. The NN presented in this article can be trained in a time of the order of minutes. It has been tested using simulated pulses and real pulses from scintillators. Therefore, this approach can be used for analysis of pulses coming from particle detectors.

Acknowledgements

The authors thank the Radiation Physics Laboratory located in Santiago de Compostela University (Spain) for providing the scintillator detector and its associated electronics. The authors also thank Dr Violeta Monasterio for their valuable comments and advices.

References

[1] Glenn G. Knoll, Radiation Detection and Measurement (4th Edition), 2010.

[2] M. Nakhostin, Signal processing for radiation detectors, John Wiley & Sons, 2017.

[3] V. Radeka, Optimum signal-processing for pulse-amplitude spectrometry in the presence of high-rate effects and noise, IEEE Transactions on Nuclear Science 15 (3) (1968) 455–470.

[4] F. S. Goulding, Pulse-shaping in low-noise nuclear amplifiers: A physical approach to noise analysis, Nuclear instruments and methods 100 (3) (1972) 493–504.

[5] P. D. Group, P. Zyla, R. Barnett, J. Beringer, O. Dahl, D. Dwyer, D. Groom, C.-J. Lin, K. Lugovsky, E. Pianori, et al., Review of particle physics, Progress of Theoretical and Experimental Physics 2020 (8) (2020) 083C01.

[6] A. Regadío, J. Tabero, S. Sánchez-Prieto, Impact of colored noise in pulse amplitude measurements: A time-domain approach using differintegrals, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 811 (2016) 25–29.

[7] C. Imperiale, A. Imperiale, On nuclear spectrometry pulses digital shaping and processing, Measurement 30 (1) (2001) 49–73.

[8] M. Kafaee, M. M. Goodarzi, Pile-up correction in spectroscopic signals using regularized sparse reconstruction, IEEE Transactions on Nuclear Science 67 (5) (2020) 858–862.

[9] V. T. Jordanov, Unfolding-synthesis technique for digital pulse processing. part 1: Unfolding, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 805 (2016) 63–71.

[10] K. Antczak, Deep recurrent neural networks for ecg signal denoising, arXiv preprint arXiv:1807.11551.

[11] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, 2013, pp. 6645–6649.

[12] C. Yang, C. Feng, W. Dong, D. Jiang, Z. Shen, S. Liu, Q. An, Alpha–gamma discrimination in baf 2 using fpga-based feedforward neural network, IEEE Transactions on Nuclear Science 64 (6) (2017) 1350–1356.

[13] G. Liu, M. D. Aspinall, X. Ma, M. J. Joyce, An investigation of the digital discrimination of neutrons and γ rays with organic scintillation detectors using an artificial neural network, Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipmentdoi:10.1016/j.nima.2009.06.027.

[14] J. Griffiths, S. Kleinegesse, D. Saunders, R. Taylor, A. Vacheret, Pulse shape discrimination and exploration of scintillation signals using convolutional neural networks, Machine Learning: Science and Technology 1 (4) (2020) 045022. doi:10.1088/2632-2153/abb781.
URL https://doi.org/10.1088/2632-2153/abb781

[15] X. Fabian, G. Baulieu, L. Ducroux, O. Stézowski, A. Boujrad, E. Clément, S. Coudert, G. de France, N. Erduran, S. Ertürk, V. González, G. Jaworski, J. Nyberg, D. Ralet, E. Sanchis, R. Wadsworth, Artificial neural networks for neutron/gamma discrimination in the neutron detectors of neda, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 986 (2021) 164750. doi:https://doi.org/10.1016/j.nima.2020.164750.
URL http://www.sciencedirect.com/science/article/pii/S0168900220311475

[16] R. Jimenez, M. Sanchez-Raya, J. Gomez-Galan, J. Flores, J. Duenas, I. Martel, Implementation of a neural network for digital pulse shape analysis on a fpga for on-line identification of heavy ions, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 674 (2012) 99–104.

[17] J. Flores, I. Martel, R. Jiménez, J. Galán, P. Salmerón, Application of neural networks to digital pulse shape analysis for an array of silicon strip detectors, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 830 (2016) 287–293.

[18] R. Scrimaglio, N. Finetti, L. D'Altorio, E. Rantucci, M. Raso, E. Segreto, A. Tassoni, G. Cardarilli, A neural network device for on-line particle identification in cosmic ray experiments, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 524 (1-3) (2004) 152–161.

[19] Z. Szadkowski, D. Głas, K. Pytel, M. Wiedeński, Optimization of an fpga trigger based on an artificial neural network for the detection of neutrino-induced air showers, IEEE Transactions on Nuclear Science 64 (6) (2017) 1271–1281.

[20] T. D. Kulkarni, W. F. Whitney, P. Kohli, J. Tenenbaum, Deep convolutional inverse graphics network, in: Advances in neural information processing systems, 2015, pp. 2539–2547.

[21] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[22] D. Stoller, S. Ewert, S. Dixon, Wave-u-net: A multi-scale neural network for end-to-end audio source separation, arXiv preprint arXiv:1806.03185.

[23] A. Défossez, N. Usunier, L. Bottou, F. Bach, Music source separation in the waveform domain, arXiv preprint arXiv:1911.13254.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL `https://www.tensorflow.org/`

[25] F. Chollet, et al., Keras, `https://keras.io` (2015).

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. `doi:10.1038/s41592-019-0686-2`.

[27] J. Lanchares, O. Garnica, J. L. Risco-Martín, J. I. Hidalgo, A. Regadio, Real-time evolvable pulse shaper for radiation measurements, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 727 (2013) 73–83.

[28] A. Regadio, S. Sanchez-Prieto, L. Esteban, Filtering of pulses from particle detectors using neural networks by dimensionality reduction, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 942 (2019) 162372.