



Original Article

Neutron spectrum unfolding using two architectures of convolutional neural networks

Maha Bouhadida ^{a,*}, Asmae Mazzi ^a, Mariya Brovchenko ^a, Thibaut Vinchon ^b, Mokhtar Z. Alaya ^c, Wilfried Monange ^a, François Trompier ^a^a Institut de Radioprotection et de Sûreté Nucléaire (IRSN), Fontenay-aux-Roses, 92260, France^b Institut de Radioprotection et de Sûreté Nucléaire (IRSN), Cadarache, 13115, France^c LMCA EA222, Université de Technologie de Compiègne, 60200, Compiègne, France

ARTICLE INFO

Article history:

Received 20 October 2022
 Received in revised form
 17 March 2023
 Accepted 20 March 2023
 Available online 21 March 2023

Keywords:

Neutron spectrum unfolding
 Machine learning
 Convolutional neural networks

ABSTRACT

We deploy artificial neural networks to unfold neutron spectra from measured energy-integrated quantities. These neutron spectra represent an important parameter allowing to compute the absorbed dose and the kerma to serve radiation protection in addition to nuclear safety. The built architectures are inspired from convolutional neural networks. The first architecture is made up of residual transposed convolution's blocks while the second is a modified version of the U-net architecture. A large and balanced dataset is simulated following "realistic" physical constraints to train the architectures in an efficient way. Results show a high accuracy prediction of neutron spectra ranging from thermal up to fast spectrum. The dataset processing, the attention paid to performances' metrics and the hyper-optimization are behind the architectures' robustness.

© 2023 Korean Nuclear Society, Published by Elsevier Korea LLC. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Neutron flux or fluence (flux integrated over time) energy distribution (also known as neutron spectrum) plays a key role for radiation protection, nuclear safety, reactors design and others. Despite the progress in neutron detector systems, assessing the neutron spectrum remains a hard task essentially for two reasons. First, the variety of shapes is almost infinite and depends on the neutron sources as well as the neutron interactions along their path. Second, no measurement method provides a direct assessment of the neutron flux distribution in the energy range encountered in radiation protection scenarios, which ranges from GeV evaporation sources down to meV thermalized neutron spectra. Indeed, multiple detectors are needed to cover this whole energy range and data processing is required to elaborate the fluence energy distribution. Therefore, the estimation of the neutron spectrum remains a hard task.

The most explored approaches to assess a neutron spectrum for radioprotection applications remain the so-called Bonner spheres spectroscopy (BSS) and the multi-foils activation methods [1]. The

basic principle of these approaches is identical and is based on multiple detectors having different responses functions against the neutron's energy. Knowing the response function of all detectors and the measured data (counting rates in detectors), an unfolding algorithm is in principle able to assess the neutron spectrum. Most of the unfolding algorithm are based on iteration, maximum entropy, and matrix inversion [2–4]. The feedback from the use of these unfolding algorithms highlights a number of limitations, the main one being the necessity to provide a prior spectrum, the output of the unfolding process being dependent on the initial solution proposed.

To get more accurate neutron spectra unfolding without depending on a priori knowledge of the measured neutrons, alternatives using artificial neural networks (ANNs) have been proposed [5,6]. An ANN can learn and model complex and non-linear relationships between inputs and outputs. It is based on a training process, which overcomes the requirement of any prior knowledge of the neutron spectrum. However, a large and balanced training dataset is mandatory to deploy a robust ANN. Besides, the design of these architectures is rather challenging, and special attention should be paid to model choice, training dataset processing (feature engineering, data augmentation, scaling, ...), performance metrics, loss functions, and hyper-parameters tuning.

Different unfolding ANN architectures have been proposed in

* Corresponding author.

E-mail address: maha.bouhadida@gmail.com (M. Bouhadida).

recent years. We can cite multilayer perceptron (MLP) [7], generalized regression neural network (GRNN) [8], and radial basis function network (RBF) [9]. In this paper we investigate an innovative approach by using a convolutional neural network (CNN), which, to the authors' knowledge, has not yet been considered for neutron spectra unfolding. As a matter of fact, CNNs are mainly credited for their efficiency in image processing since they use linear algebra as a powerful tool to identify image features. Recently, CNN has also started to be used in signal prediction, which makes them a promising candidate for neutron spectra reconstruction [10].

In this paper, we propose to investigate the use of two architectures that derive from CNN. The first one is based on residual transposed convolution blocks [11] and the second one is an adapted version of U-NET architecture recently investigated in the biomedical field [12]. The methods' performances are detailed and compared opening the way to a robust unfolding solution for different applications in the nuclear field.

2. Methodology

2.1. Training dataset

To perform efficient and robust solutions for neutron spectra reconstruction, a large and balanced dataset is necessary for the neural network training. This dataset should include neutron spectra and their reaction rates over a detector. The term "balanced" refers to an equiprobable proportion of the different types of neutron spectra. The dataset is based on neutronic simulation, since the neutron spectrum is data that cannot be measured directly. The simulations are based on a simple physical model (neutron source, material's geometry, and detector) enabling to generate as many samples as needed. This dataset generation is original and should cover large range of neutron spectrum that are representative of the real system neutron spectra shapes. The representativity of the spectrum in the dataset compared to real systems is an essential information to keep in mind when using the trained networks. For instance, the generated dataset is limited to fission neutron sources, that excludes its application to fusion systems.

The set of the neutron spectra, (flux distributed over 1000 energy bins), with the shape of (rows = number of samples, columns = 1000) corresponds to the neural network output. It contains 19 000 samples where each one is a neutron spectrum generated via a Monte-Carlo based SERPENT software (http://montecarlo.vtt.fi/download/Serpent_manual.pdf). The generated default SERPENT spectra are normalized by the total neutron flux. The considered geometry is described in Fig. 1. It is about a central sphere s1 of a radius R1 and two spherical "shells" s2 and s3 surrounding it and having respectively radii R2 and R3. For the spectra generation, the radii R1, R2 and R3 are randomly sampled within a given range. Each cell of this geometry (s1, s2 and s3) is made up of a mixture of 16 basic materials that absorb or slow down the neutrons with random fractions. The list of the materials is the following: H2O, D2O, graphite, B, B4C, CH2, concrete, steel, UO2, U8, Cd, Pb, Gd, Ag–In–Cd, Xe and Hf. Vacuum is also added to the list to enable the modification of the materials' densities. These fractions are found by deploying a genetic algorithm guaranteeing a complete exploration of the sampling spaces and a conception of a balanced dataset.

The temperature of $S(\alpha, \beta)$ (symmetric form of the thermal scattering law where α is momentum transfer and β is energy transfer) related to materials moderators (H2O, D2O, graphite ...) is randomly chosen for each neutron spectrum within the set [21°, 51°, 101 °C, 151 °C, 201°, 251°, 300°, 350°, 373°, 523°, 726°] expressed in Celsius.

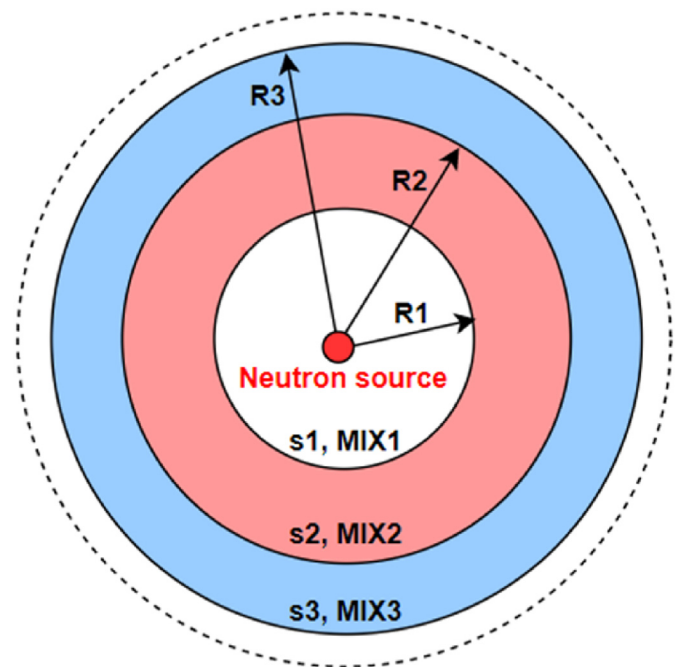


Fig. 1. Geometry context used for generating the spectra dataset with Serpent; $s_{1,2,3}$ represent the surfaces; $R_{1,2,3}$ represent the radii; $MIX_{1,2,3}$ represent the material mixtures; dashed sphere represents the detector.

Finally, the neutrons source is generated in the center of s1 and randomly chosen from the fission neutron distribution of typical isotopes. The obtained neutron flux at the external surface of s3 is scored in 1000 energy bins between 1.10^{-9} MeV and 20 MeV with equal lethargy. The SERPENT output files are processed as detailed in Ref. [13].

The neural network input is the reaction rates associated to the neutron spectra set. The reaction rates are computed for given radioisotopes in different activation foils included in a detector. We consider, as a detector, the multiple foils neutron activation spectrometer (SNAC2) described in Ref. [14]. Since SNAC2 is composed of 8 activation foils, the shape of the neural network input is (rows = number of samples, columns = 8). The reaction rates are computed as explained in Ref. [13] using Equation (1) where ϕ_i is the flux at the i th bin with $i = 1 \dots n = 1000$, RF_{ji} is the j th response function value at the i th bin with $j = 1..8$ and R_j is the j th reaction rate. The response functions are "pre-calculated" by SERPENT simulation using a detailed SNAC2 activation foils modeling. Using this approach, we have the possibility to re-use the neutron spectrum dataset, which generation requested some CPU and analysis investment, to other detectors, by simply replacing the response functions.

$$R_j = \sum_{i=1}^n RF_{ji} * \phi_i \tag{1}$$

Equation (1) assumes that the detector response is linear, allowing the response function matrix and generated spectra to convolute in order to generate the reaction rates. Such an assumption is valid in the simulation case. For experimental data and depending on the detector and the intensity of the incident flux, this assumption should be re-evaluated. In the present case of SNAC2 detector, we have checked that no saturation effect was observed as well as no detection limits.

The total number of resulting neutron spectra is 16 240. A feature engineering is performed by computing the ratios between

all the reactions rates pairs. Then, the input becomes with a matrix shape equivalent to (number of samples, $64 = 8 +$ all computed ratios). For the network output, a padding of 24 zeros at the beginning of the neutron spectra vectors is done to fit the built architectures and to preserve factors of 2 for the input shape of 64 up to 1024 for the output layer. Fig. 2 shows the architecture's general structure from the input (reaction rates) to the output (neutron spectrum).

2.2. Convolutional proposed architectures

Several intrinsic properties of CNN make it an efficient alternative for neutron spectra reconstruction. In fact, properties of a CNN are based on weight sharing which means that it makes use of local spatial coherence providing the same weight to some of the connections. This local spatial coherence exists in images as well as in the neutron spectra, where the flux in one energy bin is highly correlated to its adjacent values. Besides, the neuron connections in CNNs are inspired from the animal visual cortex where only a patch of neurons from one layer is connected to a single neuron of the next layer contrary to the fully connected networks. Since the problem to optimize is “inverse”, we can explore the translation of invariance to guarantee an accurate correlation of bins with their nearest neighbors. In this paper, we propose two varieties of CNNs: ResConvT which includes residual blocks of transposed convolutions [11] and UnetUnfold which is an adapted version of the original U-net architecture [12].

To develop both architectures, we are based on Python scientific libraries for dataset processing and on *Tensorflow* (version 2.11) (Releases · tensorflow/tensorflow (github.com) libraries for training and evaluation processes. We use the callback *TensorBoard* (version 2.11.2) (<https://pypi.python.org/pypi/tensorflow-tensorboard>) to track the training in real time, and we use the *mlflow* (version 2.0) (<https://learn.microsoft.com/en-us/azure/machine-learning/how-to-track-experiments-mlflow>) library to easily save all our results.

2.2.1. CNN1: ResConvT

Since the output size is larger than the input one, we need to expand the input data to reach this final dimension which corresponds to the neutron spectra vector length. One way to do so is to explore transposed convolution layers ConvT [11]. ConvT has no deconvolution layers but, unlike a regular convolution, it broadcasts input elements to produce a larger output via kernels and by tuning a stride. A $\text{ConvT}_{k,s,f}$ is characterized by a kernel k that extracts the data features, a stride s that modifies the amount of the

kernel movement over the data and the number of filters f within each ConvT unit. Parameters k , s , and f can be adjusted to adapt the proposed architecture to a target application.

We propose a residual block based on transposed convolution and upsampling layers as described in Fig. 3 (a). The residual alternative avoids the vanishing gradient in case of a high number of layers and creates “highways” to overcome some inefficient layers and to reduce training running time.

This residual block includes two branches: The first branch is based on $\text{ConvT}_{1,1,256}$ applied to the up-sampling part has a kernel size of 1, a stride of 1, and 256 filters. The other branch, which is $\text{ConvT}_{k,2,256}$, uses a stride of 2, 256 filters and a kernel size of k depending on the input data. The two parts are summed, and an activation function is applied to this sum. The considered activation function is the rectified linear function ReLU which will output the input directly if it is positive and zero if it is negative.

An architecture based on such blocks have the form shown in Fig. 3 (b). The input is injected into some dense layers (MLP) followed by residual blocks. The dense layers represent the encoding part, and the data dimension does not change at the output of the MLP, which is the input of the first ResConvT block. After 4 residual blocks, the final output is a 1024-length vector, and it represents the predicted neutron spectrum.

2.2.2. CNN2: UnetUnfold

Another way to reconstruct the neutron spectra efficiently is to propose an adapted version of a novel architecture named UnetUnfold. This architecture is a CNN developed for biomedical image segmentation at the Computer Science Department of University of Freiburg [12]. U-net has several advantages such as better handling of data noise, a more precise output estimation thanks to its layers' combination and preservation of the structural integrity of the input data. Fig. 4 shows the detailed structure of the adapted U-net version that we propose.

U-Net architecture looks like a ‘U’ justifying its name. It is composed of a contracting path (encoder) and an expansive path (decoder). Both paths are connected through a bottleneck section. These fundamental sections are kept in the proposed adapted version of the architecture but with key modifications in the choice of layers, data dimensions, and filter numbers and dimensions. In fact, the width of the architecture shape is based on the symmetry of the data set (respecting powers of two from 64 to 1024). These modifications make the architecture suitable for our case of study. The contracting path (left side) consists of the repeated application of two one-dimension convolutions denoted conv1D. Each conv1D is followed by a ReLU activation function. Then a 2×2 max-pooling operation is applied to reduce the input dimensions for example from 64 to 32 in the first and the second blocks. As the neural network gets deeper and since we move to forward layers, we double the number of filters for better extraction of features. For this adapted architecture, we have a filter size of 2×2 .

For the bottleneck section, there is only one block. As input, it takes the output of the last encoder block with 128 filters and 8 data dimensions. Then, it feeds it to two conv1D layers and then to a ResConvT block defined in the section above. Therefore, the result is combined with a concatenation layer to give an output with 256 filters and 8 data vector length.

In the decoder expansive path, every step includes an up-sampling of the feature map followed by a 2×2 transpose convolution block as defined before. Such a structure enables to divide by two the number of feature channels. We also add concatenations with the corresponding feature map from the contracting path, and conv1D layers. At the final step, a “flatten” layer transforms the conv1D output to a vector of 1024 bins which is fed to a dense layer that gives the corresponding target vector of 1000

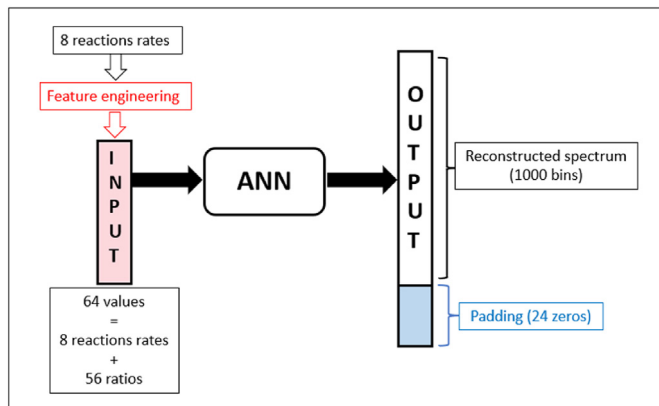


Fig. 2. ANN general structure.

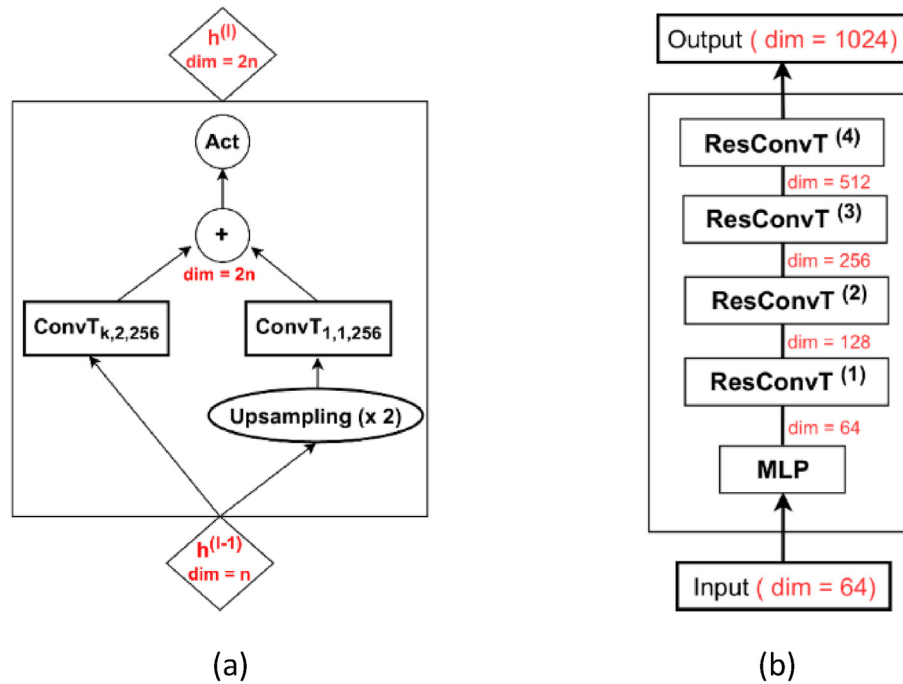


Fig. 3. (a) ResConvT block scheme: Two parallel ConvT blocks with an upsampling; (b) ResConvT architecture scheme: MLP block followed by 4 ResConvT blocks; h represents a hidden layer.

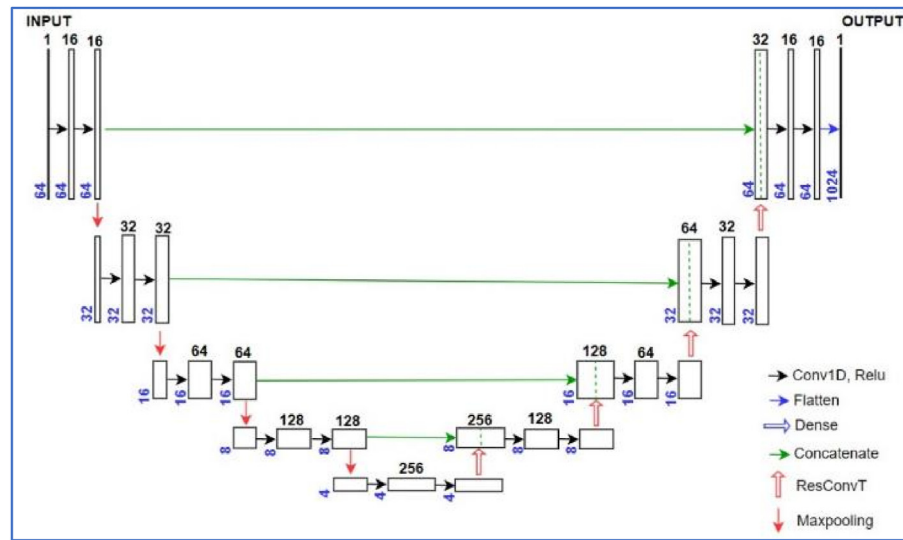


Fig. 4. UnetUnfold architecture scheme: contraction, bottleneck and expansion blocks; numbers above rectangular shapes refer to the number of filters; numbers on the side of rectangular shapes refer to data dimensions.

bins.

3. Results and discussion

To adapt the data to the architectures, scaling the input and the output is mandatory. We have tested all the possible combinations of the predefined scalers called *MaxScaler*, *MinMaxScaler* and *StandardScaler* from *scikit learn* library respectively for the neutron spectra and reaction rates. The best scaling combination is applying *MinMaxScaler* to neutron spectra and *StandardScaler* to reaction rates. In fact, *MinMaxScaler* translates each sample individually to a given range (between zero and one in this case). *StandardScaler*

consists of scaling each input vector separately while subtracting the mean (called centering) and dividing by the standard deviation to shift the distribution. Then, the mean and the standard deviation will be respectively equal to zero and one. We consider 75% of the dataset for training and the other 25% for validation. The training process is performed on an *Nvidia Tesla V100 GPU* (12 GB of GPU memory).

The hyper-parameters have a huge impact on neutron spectra unfolding and they include the number of hidden layers, the number of neurons, learning rate, batch size, normalization batch, and pooling. We decide to use *optuna* to ensure the hyper-optimization. What makes *optuna* efficient is its ability to

Table 1
Hyperparameters values for both architectures.

	ResConvT	UnetUnfold
Batch size	256	128
Regularization	Dropout of 0.2	L1 type

automate the optimization process of the hyperparameters cited above. It gives us the best combination of the optimal hyperparameter values. In fact, *optuna* is based on the history record of trials to determine the hyperparameter values for the next trial [1]. Exploring this data, it becomes able to estimate a promising area for optimal architecture. The process is repeated until reaching the optimization goal. The algorithm behind *optuna* [15] uses a Bayesian approach and is called Tree-structured Parzen Estimator. *optuna* is implemented in the ML script and adapted to the project context. We detail in Table 1 the values of the batch size and the chosen regularization fixed after the hyper-optimization. These two hyper-parameters have a significant impact on the predicted spectra smoothness. For example, in the UnetUnfold case; decreasing the batch size from 256 to 128 makes the relative error between the predicted and the real spectra decrease from an interval of [−5.2%, 5.9%] to an interval of [−2%, 2%] for Spectrum 1. Besides, applying a dropout of 0.2; in the ResConvT case; reduces the relative error from an interval of [−3.8%, 4.4%] to an interval of [−2%, 1.5%] for the same spectrum.

The performances of the prediction are measured via physically significant metrics. Since the neutron spectra are normalized, we decided to record a loss which is the mean square error MSE and a performance metric which is the spectral quality QS. MSE is a loss whose formula is detailed in Equation (2). It characterizes the degree of difference between the predicted neutron spectrum y_{pred} and the SERPENT generated neutron spectrum y_{true} . The metric QS represents the closeness of the predicted neutron spectrum y_{pred} to

the true one y_{true} . In an ideal case, both methods are close to 0. Its definition is shown in Equation (3).

$$MSE = \frac{1}{n} \sum_1^n (y_{pred} - y_{true})^2 \tag{2}$$

$$QS = \sqrt{\frac{\sum_1^n (y_{pred} - y_{true})^2}{\sum_1^n (y_{true})^2}} \tag{3}$$

Three examples of neutron spectrum reconstructions with both architectures are proposed in Fig. 5. The upper part represents a thermal spectrum denoted Spectrum 1, the medium one is a fast spectrum denoted Spectrum 2 and an epithermal one (Spectrum 3) is represented in the lower part. The default SERPENT neutron flux normalized by the total flux is plotted as a function of the energy bins. The energy bins axis is log-scaled. We notice a significant agreement between the predicted and the real spectra with both methods and for the different tested spectrum types.

We have also plotted, in Fig. 6 the relative error between the predicted and the real spectra as a function of energy bins for the three neutron spectra shown above. The relative error doesn't exceed ±2.5% for the three spectrum types along the energy bins vector. These observations are confirmed for both architectures.

For both methods, no gradient explosion effect is observed. The optimization algorithm well converges in the two cases and the gradients have low values. Fig. 7 describes the evolution of train and test QS averages as a function of epochs in the case of ResConvT. As observed, the stopping epoch is 421. UnetUnfold curves have the same trends with a stopping epoch equal to 435.

To compute MSE average of the test samples, we restore the original spectra values range. Test-MSE average is 1.16e-09 with a standard deviation of 5.85e-10 for ResConvT and 5.87e-09 with a

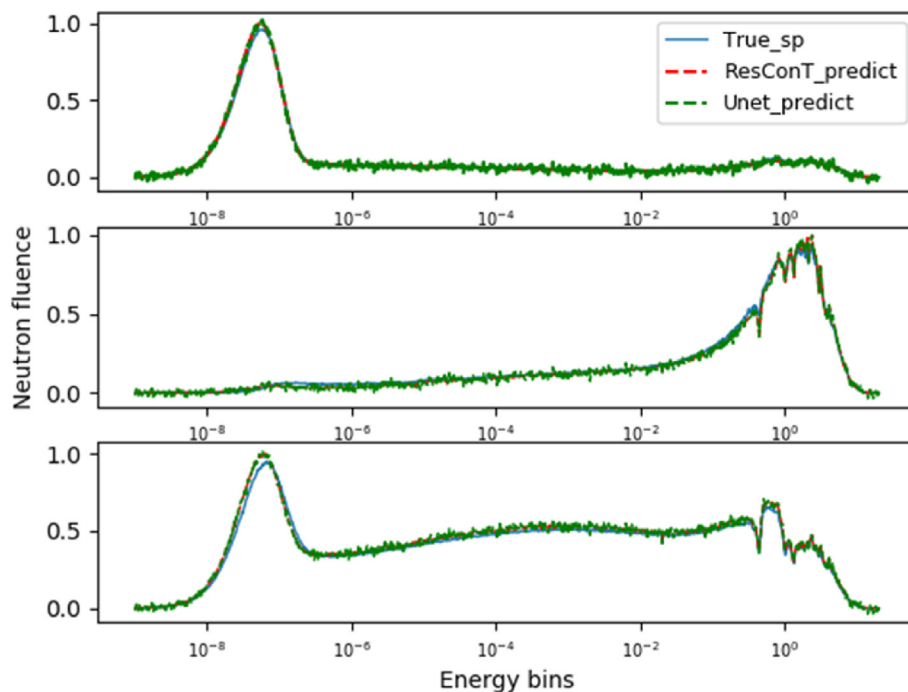


Fig. 5. Spectra reconstruction using ResConvT and UnetUnfold: Blue lines correspond the true spectra; dashed red and green lines represent respectively ResConvT and UnetUnfold predictions; Three examples of neutron spectra (thermal (Spectrum 1), fast (Spectrum 2) and epithermal (Spectrum 3)) are shown. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

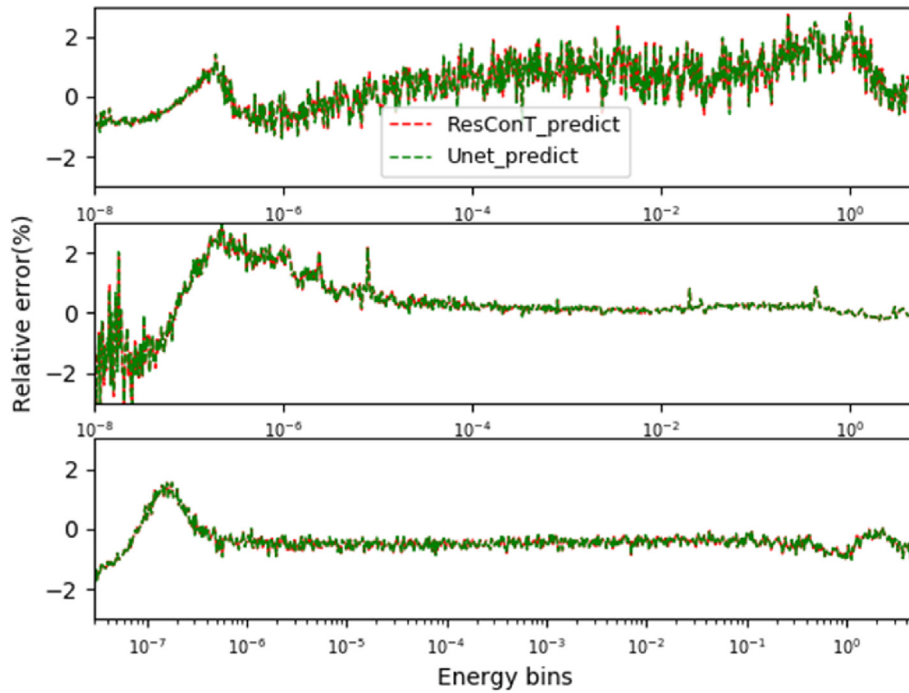


Fig. 6. Relative error between the predicted and the real spectra as a function of energy bins for Spectrum 1, Spectrum 2, and Spectrum 3.

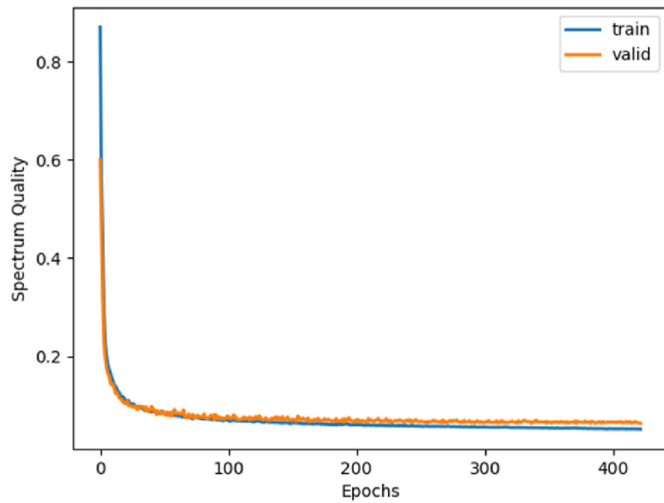


Fig. 7. Train and test QS as functions of epochs.

standard deviation of $3.95e-09$ for UnetUnfold. These losses values prove the robustness of the proposed methods. These high-quality predictions, with both methods, are found for the different types of neutron spectra (the thermal, the fast, and even the epithermal ones). The average of test-QS is 0.059 with a standard deviation of 0.034 for ResConvT architecture and 0.064 with a standard deviation of 0.041 for UnetUnfold. These average values, very close to zero, confirm the high accuracy of the neutron spectra predictions. Histograms of QS and MSE sets covering 100 samples are respectively represented in Fig. 8 (a) and 8 (b) and giving an idea about the metrics distribution for both methods. We can see that QS distributions is equivalent for the two techniques while MSE ones are different since the MSE histogram represents a maximal bar around $1e-9$ with 78 of redundancy over 100 samples.

Table 2 shows the examples of test-QS and test-MSE values associated to the specific plotted cases Spectrum 1, Spectrum 2, and Spectrum 3 in Fig. 5 and after training and after restoring the original spectra values. These spectra are, as mentioned above, among the test dataset.

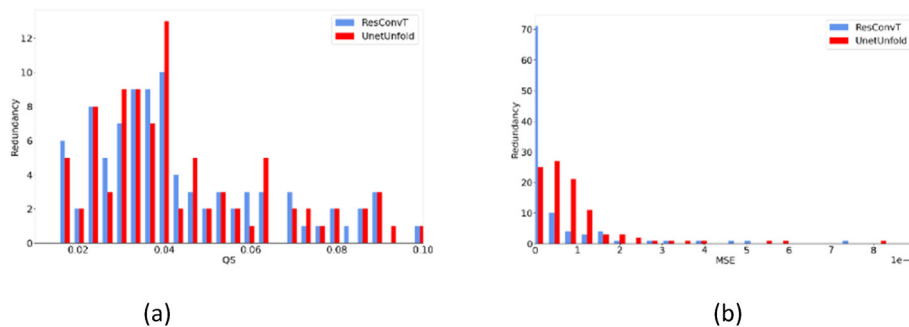


Fig. 8. Test QS and MSE histograms (100 samples).

Table 2
Validation QS and MSE for both architectures.

		ResConvT	UnetUnfold
QS	<i>Spectrum 1</i>	0.057	0.066
	<i>Spectrum 2</i>	0.063	0.07
	<i>Spectrum 3</i>	0.058	0.061
MSE	<i>Spectrum 1</i>	1.2×10^{-9}	5.1×10^{-9}
	<i>Spectrum 2</i>	2.3×10^{-9}	4.3×10^{-9}
	<i>Spectrum 3</i>	1.9×10^{-9}	6.7×10^{-9}

4. Conclusion

In this paper, we detailed two implemented ANNs for neutron spectra unfolding. Both networks include convolutional layers enabling a precise reconstruction of signals. Compared to other ANN architectures, the CNN, with both proposed versions, allows accurate neutron spectra unfolding even for a high number of energy bins (1000 bins). The spectral quality is significantly high (around 0.065) and the different types of neutron spectra are efficiently predicted giving a relative loss of an order of 10^{-9} . ResConvT is less time-consuming than UnetUnfold. Then, it is a better alternative for applications dealing with long-dimension data vectors. It also has slightly better performance metrics. However, UnetUnfold has higher performance than ResConvT for smaller training datasets (same performance metrics with only 70% of the original training dataset). So, it is an efficient option for applications suffering from a lack of available dataset on which the prediction is based. We built a large dataset including in a balanced way the different types of spectra for robust training. The suitable dataset processing (scaling and feature engineering) is one of the powerful keys to enhance accuracy. The results show that we can rely on ANNs to overcome the limitation of classical unfolding techniques, which is the necessity of the prior information and to get a robust unfolding.

The adaptation of the feature engineering to the experimental activation measurements issued from gamma spectrometry is in progress. This step requires an accurate normalization of the input measurement to the simulated training dataset. We have done some preliminary tests with real spectra resulting from CALIBAN, SILENE and GODIVA IV reactors. The first step of this approach consists in finding a normalization factor using the mean value of gold data in the training dataset divided by the value of gold activation foil in the measurement input part. Since some other deconvolution approaches and simulations have shown that the impact of neutron diffusion has not been highlighted in simulation leading to a nonrealistic computed response function for the back-copper foil, a second step is to replace the back-copper response function and the associated back-copper counting respectively by the front-copper response function and the front copper counting. In complement, we have done tests with different binning (1024 bins, 512 bins and 128 bins) in order to see the impact of this parameter on real spectra unfolding. This approach leads to promising results on case of SILENE, where we can compare with a precise simulation of the reactor and its environment. A detailed comparison of the results of ANN, with these simulations and classic deconvolution techniques results will be the object of a future publication.

It is also important to investigate the impact of neutron spectra and reaction rates' uncertainties on the CNN model performances and on the spectra prediction. We aim to integrate them for example, by adding new input features by modeling a set of Gaussians where the mean is the main input information and the standard deviation is the uncertainties set.

Finally, expanding the applicability of the proposed models to different fields is possible by making the networks allow to compute the absorbed dose and the kerma which are key parameters for radiation protection for example.

In the future, the genetic algorithm could be more accurate and can take more parameters and constraints into account to fit as much as possible the real data case and this is part of a new exploratory project having for goal to characterize the neutron spectra induced in laser installations. Our actual dataset is up to now oriented to the state of art of expected SNAC2 spectra and should of course be improved for this new project.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank Stéphane Gaiffas for his valuable and constructive suggestions regarding the chosen methodology. They would also like to thank Marcel Reginatto for his generous support to this research project.

References

- [1] M. Reginatto, Bayesian approach for quantifying the uncertainty of neutron doses derived from spectrometric measurements, *Radiat. Protect. Dosim.* 121 (1) (2006) 64–69.
- [2] S. Itoh, A fundamental study of neutron spectra unfolding based on the maximum likelihood method, *Nucl. Instrum. Methods A* 251 (1) (1986) 144–155.
- [3] M. Reginatto, Overview of spectral unfolding techniques and uncertainty estimation, *Radiat. Meas.* 45 (10) (2010) 1323–1329.
- [4] H.N. Mülthei, B. Schorr, On an iterative method for the unfolding of spectra, *Nucl. Instrum. Methods A* 257 (2) (1987) 371–377.
- [5] C. Cao, Q. Gan, J. Song, K. Yang, L. Hu, F. Wang, T. Zhou, An adaptive deviation-resistant neutron spectrum unfolding method based on transfer learning, *Nucl. Eng. Technol.* 52 (2020) 2452–2459.
- [6] S.A. Hosseini, Neutron spectrum unfolding using artificial neural network and modified least square method, *Radiat. Phys. Chem.* 126 (2016) 75–84.
- [7] C. Cao, Q. Gan, J. Song, P. Long, B. Wu, Y. Wu, A two-step neutron spectrum unfolding method for fission reactors based on artificial neural network, *Ann. Nucl. Energy* 139 (2020), 107219.
- [8] J. Wang, Z. Guo, X. Chen, Y. Zhou, Neutron spectrum unfolding based on generalized regression neural networks for neutron fluence and neutron ambient dose equivalent estimations, *Appl. Radiat. Isot.* 154 (2019), 108856.
- [9] A.A. Alvar, M.R. Deevband, M. Ashtiyani, Neutron spectrum unfolding using radial basis function neural networks, *Appl. Radiat. Isot.* 129 (2017) 35–41.
- [10] R. Yamashita, M. Nishio, R.K.G. Do, Convolutional neural networks: an overview and application in radiology, *Insights Imaging* 9 (2018) 611–629.
- [11] D. Im, D. Han, S. Choi, S. Kang, H.-J. Yoo, D.T.-CNN, Dilated and transposed convolution neural network accelerator for real-time image segmentation on mobile devices, in: *Proc. Of the IEEE International Symposium of Circuits and Systems (ISCAS)*, May 26–29 2019, pp. 1–5. Sapor, Japan.
- [12] O. Ronneberger, P. Fischer, T. Brox, U-Net, Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. Wells, A. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, MICCAI 2015, Lecture Notes in Computer Science, vol. 9351, Springer, Cham, 2015, https://doi.org/10.1007/978-3-319-24574-4_28.
- [13] Maha Bouhadida, Mariya Brovchenko, Vinchon Thibaut, Wilfried Monange, Francois Tromprier, Neutron spectra reconstruction based on an artificial neural network trained with a large built dataset, in: *ICRS 14/RPSD 2022 (14th International Conference on Radiation Shielding and 21st Topical Meeting of the Radiation Protection and Shielding Division)*, ANS, Seattle, United States, Sep 2022. (hal-03900950).
- [14] F. Tromprier, C. Huet, R. Medioni, I. Robbes, B. Asselineau, Dosimetry of the mixed field irradiation facility CALIBAN, *Radiat. Meas.* 43 (2008) 1077–1080.
- [15] T. Akiba, Optuna: a next-generation hyperparameter optimization framework, in: *Proc. Of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, USA, August 4–8, 2019, pp. 2623–2631.