UNIVERSITY OF BIRMINGHAM

Fusion Neutron **Acti**vation Spectra **U**nfolding by **N**eural **N**etworks
(FACTIUNN)

author:	Ocean Wong
	(Hoi Yeung Wong)
supervisor:	Ross Worrall
Submitted in fulfilment of the requirement for:	MSc. Physics and Technology of Nuclear Reactors
date:	June-September 2019
student ID:	1625143

I warrant that the content of this dissertation is the direct result of my own work and that any use made in it of published or unpublished materials is fully and correctly referenced.

Abstract

Include these things:

- Attempted this
- Got this result
- advice for the future

Keywords: activation, neutronics, fusion

Contents

List of Figures

1 Introduction

In a fusion reactor, the neutron fluence can go up to as high as 1.6×10^{21}[citation needed]. (For JET, [citation needed]; for ITER, [citation needed]). This leads to an unprecedented need of shielding against neutrons of up to 14.1 MeV or higher energies, which has not been experienced in fission reactors before [citation needed].

Neutrons are notoriously difficult to shield against due to their uncharged nature, and therefore low propensity to interact with matter[citation needed]. To develop effective shielding for various components of the reactor from these high energy neutrons, the energy spectrum of the neutrons created inside the nuclear reactor has to be well understood [3]. It is also important to understand the neutron spectrum inside the reactor in order to develop Tritium breeding modules, which is essential for making fusion a sustainable source of clean energy [12]. Last but not least, the power output of future fusion power plants can only be quantified when the neutron spectrum is characterised [31]. An accurate measurement of the neutron spectrum is required to properly model the energy distribution of neutrons to be used in neutron transport simulations for the above purposes.

Therefore, neutron energy measurement is a key focus [rewording needed] in the diagnostic systems in all fusion reactors.

Ironically, for the same reason that they are difficult to shield against, neutron energy is also difficult to measure. Neutrons, especially high energy neutrons such as the 14.1 MeV neutrons created in fusion reactors, do not easily deposit their full energy into a sufficiently small detection volume to allow direct measurement[citation needed]. Various neutron detectors has been developed to deal with this problem[citation needed] ;however, most of them cannot stand this high neutron fluence that is found at the first wall of fusion reactors without additional shielding that changes the flux profile, defeating the objective of trying to measure neutrons energy distribution with minimal disturbance to the spectrum itself.[citation needed] The extreme temperature and magnetic fields inside the nuclear fusion reactor compounds the difficulty of employing other means of neutron measurement as most electronics will not be able to function in such environments effectively. [20]

This is where the technique of neutron activation stands out:

By analyzing the level of activations in various elements induced by neutrons, relying on the fact that different reactions has different sizes of reaction cross-sections, each with varying sensitivities to neutrons of different energies, one can infer the neutron spectra that was previously present at the first wall.

This is a very robust method as it does not require any active components, thus can be employed for very high neutron fluxes and fluences[5], as the total number of neutron activation reactions can be controlled by changing the thickness of the activation foils used [7] according to the anticipated neutron fluence in the next irradiation period, so not to paralyze the γ radiation detector. It is also insensitive to γ rays, thus removing most of the challenges facing mixed-field spectrometry. [2]

The disadvantage of this method is that it has to be time-integrated (over the whole irradiation period), i.e. no information about the temporal variation in the neutron spectrum can be extracted.

Another disadvantage of using neutron activation as the means of measuring the neutron spectrum in a fusion reactor is that it is an indirect method of measurement, requiring the measured reaction rates to be 'unfolded' back into reaction rates. This is a 'mathematically incorrectly posed ' problem[13], as will be further explained in the next section (2.3), requiring an *a priori* spectrum to be provided before the unfolding procedure can take place. This is because the number of activities recorded (usually denoted as M) is fewer than the number of neutron groups (usually denoted as N) of whose activity we would like to know, i.e. M<N, thus the problem is underdetermined (the number of contraints is fewer than the number of variables). The *a priori* has to be used in order to

introduce extra information into the problem. However, if this *a priori* spectrum deviates too much from the actual spectrum, then the result of the unfolding will be inaccurate.

To address this problem, an investigation into using neural networks for the purpose of unfolding is presented in this thesis. Neural networks excels in incorperating previous spectra as *a priori* information, without requiring users to explicitly input an *a priori*. Two approaches are proposed. The first one is to use neural networks directly as an unfolding tool; and the second one is to use them as an *a priori* generator, which is then fed into an existing unfolding code, where the actual neutrons spectra is then calculated out of.

2 Theory

When a nuclide is placed in the activation module at the irradiation position inside a nuclear fusion reactor (or any other neutron sources), it is activated via one or more nuclear reactions with the incoming neutrons. The probability of interacting with the incoming neutron via reaction j is proportional to the microscopic cross-section $\sigma_j(E)$, where E is the neutron's energy, and reaction j is a neutron-induced reaction, i.e. (n,??) reaction.

By measuring the activity of reaction j's daughter nuclide in the activation foil (which has a known amount of the initial nuclide) after irradiation, and multiplying it by a correction factor of

$$\frac{1}{1 - exp(\lambda_j T)} \tag{1}$$

the reaction rate Z_{0j} can be obtained. This correction factor accounts for the decay of the daughter nuclide of reaction j which has a half-life of λ_j, over the period T which is the duration between irradiation and measurement. A more complicated correction factor is required if the irradiation period is comparable to the half-life λ_j, or if the population of the parent nuclides for reaction j changes over the course of the irradiation. This can be done using FISPACT-II, detailed in [26].

The total reaction rate of the j^{th} reaction can then be expressed as a Fredholm integral as follows:

$$Z_{0j} = \int_0^\infty R_j(E)\phi_0(E)dE \tag{2}$$

where the reaction rate Z_{0j} has the unit of s^{-1}, ϕ_0 is the neutron flux (unit: $cm^{-2}s^{-1}eV^{-1}$), which is a function of energy E. The unfolding process aims to find a solution spectrum ϕ which approximates the actual spectrum ϕ_0 as closely as possible.

As for R in the equation above, (which has dimension of area)

$$R_j(E) = \sigma_j(E)\frac{N_A}{A}F_j\rho V \tag{3}$$

assuming that there is no self-shielding/down-scattering inside the foil. N_A is the Advogadro's constant (unit: mol^{-1}), A is the molar mass of the parent nuclide for reaction j (unit: g mol^{-1}), F_j is reaction j's parent isotope's mass fraction in the foil's constituent material (unit: dimensionless), ρ is the density of the alloy (unit: g $barn^{-1}$ cm^{-1}), V is the volume of the foil (unit: cm^3) Note that $\sigma(E)$ (unit: $barn$)is the only energy dependent component in R.

The neutron spectrum can be discretized into N energy bins:

$$Z_{0j} = \sum_{i=1}^{N} R_{ij}\phi_{0i} \tag{4}$$

where ϕ_{0i} is the scalar flux integrated over the energy bin's range

$$\phi_{0i} = \int_{E_{i-1}}^{E_i} \phi_0 d(E) \tag{5}$$

, thus having a unit of $cm^{-2}s^{-1}$.

By assuming that the scalar flux distribution inside each energy bin is relatively flat, equation 4 calculates Z_{0j} by replacing $(R_j(E), E_{i-1} \leq E \leq E_i)$ with

$$R_{ij} = R_j(E_{i-1}) \tag{6}$$

Let there be M neutron-induced reactions whose reaction rate was measured,

$$\begin{aligned} \forall j \in & \{1, ..., M\}, \\ \exists Z_{0j} \in & \mathbb{R}_{\geq 0} \end{aligned} \tag{7}$$

Collecting all reaction rates into a vector $\boldsymbol{Z_0}$ of M-dimensions, one can express eq. 4 as a matrix multiplication equation:

$$\boldsymbol{Z_0} = \underline{\underline{\mathbf{R}}}\boldsymbol{\phi_0} \tag{8}$$

where $\underline{\underline{\mathbf{R}}}$ is a $M \times N$ matrix, termed the *response matrix*. $\boldsymbol{\phi_0}$ is an N-dimensional vector containing the neutron flux in the each of the N bins. The subscripts 0's denotes that they are the measured/known quantity, as opposed to the conjectured solutions which will appear later in this text.

For nuclear fusion applications, the number of possible reaction investigated M is very limited [15], as the parent nuclide of each of these reactions must exist in solids which:

- does not melt in the reactor,
- can be machined into specified shape and thickness,
- are safe to be handled,
- has a cutoff/threshold energy in the region of interest (in the MeV range),
- has well-characterised cross-section values in nuclear data libraries (see [6])
- has sifficiently stable parent and daughter isotopes for the activation and γ measurement to be carried out respectively.

in practice, very few types of metals/alloys can be used in these systems. For the ACT in JET in particular, in recent experiments, only 7 types of foil materials and 11 reactions can be examined. [26].

Meanwhile, the number of bins, N, can be arbitrarily high. For some investigations, such at the one in [24] it goes up to 709 bins. This makes the unfolding problem a very underdetermined one.

In the mathematical sense of the problem, an inverse does not exist. This is because, theoretically, multiple neutron spectra, say $\boldsymbol{\phi_0}$, $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$, can give the same set of reaction rates $\boldsymbol{Z_0}$, so there is no correct, unique choice of mapping of $\boldsymbol{Z_0}$ back to $\boldsymbol{\phi_0}$, $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$. [13] A conceivable situation is detailed below as an example:

The three spectra $\boldsymbol{\phi_0}$, $\boldsymbol{\phi_1}$, $\boldsymbol{\phi_2}$ has identical flux values in all but the first two energy bins:

- $\boldsymbol{\phi_0}$ has a flux of $0cm^{-1}s^{-1}$ in the 1^{st} bin and a flux of $2 \times 10^{10}cm^{-2}s^{-1}$ in the 2^{nd} bin;
- $\boldsymbol{\phi_1}$ has a flux of $2 \times 10^{10}cm^{-2}s^{-1}$ in the 1^{st} bin and a flux of $1 \times 10^{10}cm^{-2}s^{-1}$ in the 2^{nd} bin;
- $\boldsymbol{\phi_2}$ has a flux of $4 \times 10^{10}cm^{-2}s^{-1}$ in the 1^{st} bin and a flux of $0cm^{-2}s^{-1}$ in the 2^{nd} bin;

And the reaction cross-sections in this energy range (very low neutron energy/thermal energy) for all but the 1^{st} reaction is vanishingly small, as all other reactions than the 1^{st} reaction are threshold reactions.

If the first two columns of the response matrix $\underline{\underline{\mathbf{R}}}$ are given as follows:

$$R_{1,j} = \delta_{1j}(5 \times 10^{-11})cm^2 \tag{9}$$

$$R_{2,j} = \delta_{1j}(1 \times 10^{-10})cm^2 \tag{10}$$

where the δ used is the Kronecker delta,

then one can see that $\boldsymbol{\phi_0}$, $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$ will all lead to the same reaction rate Z_0. This is because, in each of these cases, the first two bins of each of the $\boldsymbol{\phi}$ contributes the same amount of reaction rate of 2 counts s^{-1} to the 1^{st} reaction rate (Z_{0j} where j=1), ultimately resulting in the same measured reaction rate of $\boldsymbol{Z_0}$.

Such a problem is termed 'mathematically incorrectly posed'. [13]

2.1 General unfolding methods

The most straight-forward way of getting back a solution ϕ is by using the Moore-Penrose inverse matrix. This matrix inversion operation generalizes the usual matrix inversion operation for square matrices, where the $M \times N$ response matrix $\underline{\underline{\mathbf{R}}}$ in equation 8 is inverted into an $N \times M$ matrix $\underline{\underline{\mathbf{R}}}^{-1}$, so that ϕ can be obtained by $\boldsymbol{\phi} = \underline{\underline{\mathbf{R}}}^{-1}\boldsymbol{Z_0}$. However, this method is the equivalent of rotating a 2-D photo of a 3-D object from a horizontal position to an upright/tilted position: the solution is still "trapped" in a flat, M-dimensional manifold within the N-dimensional solution space.

Therefore to start the unfolding process, extra information has to be given to the program. This is termed the *a priori* spectrum.

The most general unfolding program can, ideally, find a solution $\boldsymbol{Z}$, $\underline{\underline{\mathbf{R}}}$ and $\boldsymbol{\phi}$ [18], such that their overall deviation from the measured reaction rates ($\boldsymbol{Z_0}$), expected response matrix ($\underline{\underline{\mathbf{R_0}}}$), and the initial guessed neutron spectrum ($\boldsymbol{\phi_0}$), is minimized. The deviation of the solution reaction rates from the measured reaction rate is calculated from its covariance matrix $\underline{\underline{\mathbf{S_Z}}}$, as the $(\chi^2)_Z = Z^T\underline{\underline{\mathbf{S_Z}}}^{-1}Z$. Equivalently the deviation of $\boldsymbol{\phi}$ from $\boldsymbol{\phi_0}$ and $\underline{\underline{\mathbf{R}}}$ from $\underline{\underline{\mathbf{R_0}}}$ can be calculated from their respective covariance matrix..

2.2 Current practice

In practice, the ambiguity in the response matrix is nearly always ignored, by assuming that the response matrix $\underline{\underline{\mathbf{R_0}}}$ is accurately and precisely defined, fixing the response matrix during the solution search. This reduce the number of dimensions in the solution search by $M \times N$, massively reducing the computational complexity. It also assumes that the covariance matrix of the reaction rates is diagonal, i.e. there are no covariance matrix across different reaction rates.

Some programs, such as GRAVEL[17] and SAND-II[19], simply start their iterative solution search from this *a priori* spectrum, with the aim of minimizing the χ^2 (which measures the deviation of $\boldsymbol{Z}$ from $\boldsymbol{Z_0}$); while others, such as MAXED [27] add the deviation of the solution spectrum from the *a priori* spectrum ($\boldsymbol{\phi}$ from $\boldsymbol{\phi_0}$) **on top of** the deviation of the solution reaction rates from the measured reaction rates ($\boldsymbol{Z}$ from $\boldsymbol{Z_0}$) when evaluating the χ^2.

Current fusion neutron measurements relies on MCNP simulations heavily to supplement their unfolding procedure. They use MCNP model of thre reactor to obtain a guess neutron spectrum, which is used as the *a priori* [16] [14], and the response matrix is usually obtained in the same way as well [7].

2.3 Neural Networks

Neuralnetworks, on the other hand, learns the relationship between reaction rates and the original neutron spectrum. Ideally it will make use of information in previous neutron spectra, effectively bypassing the problem of underdetermination.

A typical neural network learns the relationship between the inputs (the two nodes in the leftmost layer in Figure 1) and outputs (the node in the rightmost layer in Figure 1) of a function via training, thus becoming an approximator for that function.

2.3.1 Forward Propagation

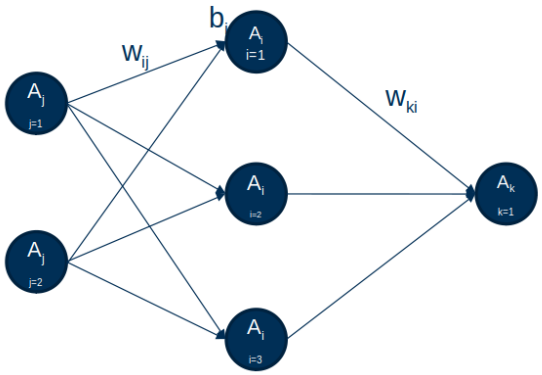


Figure 1: Illustration of the topology of a typical neural network

The inputs to the neural network are known as "features" and the outputs are known as the "labels".

In the context of neutron spectrum unfolding using neural networks, there are M features (reaction rates Z_i for $1 \leq i \leq M$) and N labels (neutron flux in each bin ϕ_j for $1 \leq j \leq N$).

The "activation" A_i of the i^{th} node refers to the value that it takes. w_{ij} denotes the "weight" of each connection from the j^{th} node to the i^{th}.

When the activations in the input layer (A_i) are known, the activation in the next layer (in this case, the first hidden layer) is calculated as follows:

$$A_i = \sigma_i \left(\sum_j (w_{ij} A_j) + b_i \right) \tag{11}$$

b_i denotes a "bias" value which will be added onto the sums in front of each node before it is parsed through the activation function σ_i. The activation function is usually denoted as σ_i, i.e. it is possible to use different activation functions for different nodes i; however the common practice is to use the same type of activation function across the whole layer, or even across all nodes and all layers of the neural network. The typical function chosen is the ReLU function (Figure 2), i.e. for all layers, and for all values of i, as it is one of the simplest non-linear function whose gradient can be computed quickly.

$$\sigma_i(x) = ReLU(x) = \frac{|x| + x}{2} \tag{12}$$

Equation 11 is applied recursively to calculate the activations in the immediate next layer. For example, to calculate the activations in second layer (i.e. the output layer) in Figure 1 simply by swapping the indices in for the indices of the next layer: $i \mapsto h$, $j \mapsto i$. This process is known as forward propagation.

2.3.2 Backpropagation

The weights w and biases b are known as the parameters of the neural network. This is in contrast with the term "hyperparameters", which are the numbers that describes the topology of the neural network, i.e. number of layers, number of nodes in each layer,

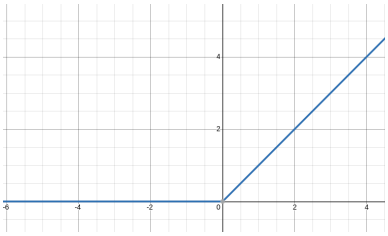


Figure 2: A ReLU function (a rectifying function)
Abscissa=function's input; ordinate=function's output.

learning rate (see section 2.3.4 below), etc. During the training phase of the neural network, these parameters are adjusted so that the neural network's predicted output values align with the true output values more closely. This deviation of the predicted label from the true label is termed the "loss value", and can be calculated in a variety of manner (see Section 13 for the loss value metrics considered in this investigation). For the moment let's assume it is calculated as the mean-squared value, i.e. same as the χ^2 value familiar to physicists.

The process of adjusting parameters to reduce the loss value is known as backpropagation, as the 'desired' change to each weight and bias (calculated from the gradient of the loss value with respect to w or b, i.e. $\frac{\partial(loss)}{\partial w}$ or $\frac{\partial(loss)}{\partial b}$) is obtained by tracing the change in the output layer back to the weight and biases of each layer.

For the neural network to converge on a stable set of parameters (i.e. a minimum value of the loss value in the parameter space), features are usually normalized before they are given to the neural network. This reduces the difference in variance across each feature, allowing the neural network to take a more direct path when gradient-descending to the set of parameters that achieves minimum loss value, instead of an oscillatory approach to the minimum loss value[21], thus reducing the number of steps required to train the neural network.

2.3.3 Universal approximation theorem

Before diving into the details of neural network training, it is beneficial to see how a neural network can approximate any function.

The key to its ability of approximating functions lies in the non-linear activation function.

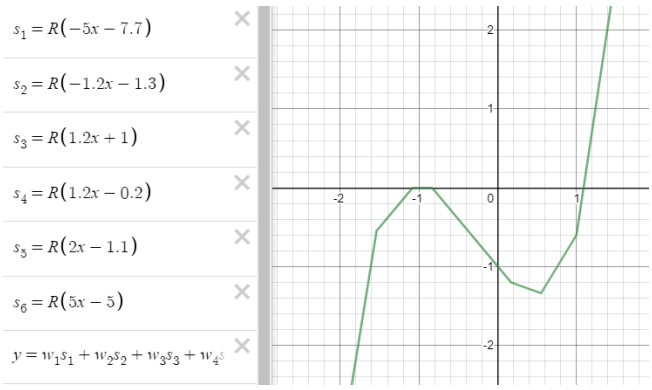


Figure 3: A cubic function approximated by a neural network

This neural network has 1 hidden layer containing 6 neurons. Here, R is the alias for the ReLU function (see Figure 2), abscissa is the input layer neuron's activation value (feature); and the ordinate is the output layer neuron's activation value (label), obtained by summing over the product of the activation of the i^{th} neuron (aliased as s_i) with the weight of its connection to the final layer (aliased as w_i). The weights and bias to the first layer is already defined on-screen inside the brackets of the first six lines; while the weights and bias to the second layer (output layer) is defined off screen.

Figure 3 is a crude representation of how a 1-hidden-layer neural network can approximate a cubic function. A single hidden layer neural network with one scalar input and one scalar output is able to approximate any non-linear functions, provided that there are enough neurons in the hidden layer.

The weights w_i scales each ReLU function; whereas the bias to the first layer (defined as the second term inside each of the bracket in the first six lines) changes the horizontal offset of each ReLU function. The bias to the second layer controls the vertical offset of the whole function. Summing them up leads to the output in Figure 3.

Even with only six hidden neurons (only 19 parameters), it is able to reasonably approximate a cubic function within the visualized domain. Obviously this approximation to the cubic function can be improved by increasing the number of neurons available in the neural network, provided that there are enough training data to cover the domain densely.

Armed with this intuition, the notion of a neural network being able to approximate any function becomes conceivable, even if the function has vectorial inputs and outputs.

2.3.4 Training the neural network

Before adjusting the parameters, a fraction of the data is drawn out and reserved for testing. The remaining is used as the training dataset. These "training" and "testing" data are chosen in such a way that they cover the same range of domain and co-domain in the feature- and label-space respectively.

The parameters are adjusted in iterative steps to minimize the loss value. Each step requires calculation of the gradient value over the entire dataset, known as an "epoch", obtained by calculating the average $\frac{\partial(loss)}{\partial w}$ or $\frac{\partial(loss)}{\partial b}$ over the entire training dataset. When further training no longer improves the loss value over the training set (i.e. the "training loss"), training can be stopped, and the parameters are then fixed at their final values. The size of each step is calculated as (learning rate)$\times$(gradient of the loss value in parameter space)

The performance of the neural network is then evaluated over the testing set to obtain an average loss value, known as the "testing loss". If the testing loss is much higher than the training loss, it signifies that the neural network was "overfitting", i.e. it reached a

minimum training loss by "memorizing" the relationship between training features and training labels, and is unable to generalize these relationships to the testing set. This may suggest that the neural network is too complex, i.e. has too many nodes or neurons.

Apart from reducing the complexity of the model, various techniques exist to reduces overfitting, including weight-regularization and dropout [1]. Weight regularization ensures that the numerical values of weights w remains small; while dropout effectively removes a specified fraction of the connections at each layer. However, these techniques are not applied.

However, one of the most widely used method of reducing overfitting is by measuring the validation loss. A small subset of the training data is reserved and not used for backpropagation during the training, but its loss value (known as the "validation loss") is calculated at each epoch also. This amounts to calculating the loss value of the neural network's prediction on a set of data that it has never seen before as well. When the validation error stops decreasing, then one can be sure that the neural network has stopped identifying general patterns which applies across both the validation set and the training set, and begin memorization. The training can be stopped at this point.

This method is called "Early Stopping"; it catches the neural network before it begins overfitting aggressively.

2.3.5 Applying neural network to the unfolding problem

To apply neural networks as unfolding tools, we will want neutron spectra as the output and reaction rates as the input, i.e. M features (reaction rates Z_i for $1 \leq i \leq M$) and N labels (neutron flux in each bin ϕ_j for $1 \leq j \leq N$).

By using a neural network to do the unfolding, we are assuming that the inverese equation (below) exist,

$$\boldsymbol{Z} = \underline{\underline{\mathbf{R}}}^{-\mathbf{1}}\boldsymbol{\phi} \tag{13}$$

i.e. all reaction rates can be unfolding back to one and only one unique solution spectrum. Ideally, the set of all possible solution for the neutron spectrum $\boldsymbol{\phi}$ is expected to be confined in an M- (or fewer-) dimensional manifold in the N-dimensional solution space, by various physical constraints. The role of this neural network is to identify this M-dimensional manifold co-domain in the solution space.

Several metrics were considered for the neural networks. Since the neural networks' goal is to predict a set of labels (solution spectrum) ϕ_{pred} that is identical to the true spectrum ϕ_0 when given the set of features Z_0 corresponding to the set of labels ϕ_0, the loss function must have a minimum at $\phi_{pred} = \phi 0$.

This loss value is also expected to scale its penalization according to the true flux ϕ_0. Large deviation when ϕ_0 is large should be penalized by the same amount as with small deviations when ϕ_0 is small. For example, over-predicting the to flux at the 14.1 MeV peak by, say, 10%, in a DD-operation, should be given the same penalty as over-predicting the 14.1 MeV peak flux in a DT-operation by 10%, despite the fact that $\phi_0(E = 14.1MeV)$ is much smaller for the same TOKAMAK in a DD campaign than in a DT campaign.

Several of such functions comes to mind; they include:

- cross entropy, $H(\phi_{pred}, \phi_0) = \sum_i^N \left(\phi_{pred}(E_i)(ln(\phi_{pred}(E_i)) - ln(\phi_0(E_i))) \right)$ (See [29])
- Average distance in L^P log-space $= \left(\sum_i^N (log(\phi_{pred}) - log(\phi_0))^p \right)^{\frac{1}{p}}$ which is a generalization of mean squared error and mean absolute error.
- mean fractional deviation, $MFD(\phi_{pred}, \phi_0) = \sum_i^N \left| \frac{\phi_{pred}(E_i) - \phi_0(E_i)}{\phi_0(E_i)} \right|$

In the end, the following functions were chosen as they were the default functions available from tensorflow; using these functions minimizes the room for human error and development time.

Let there be L features-labels pairs in the datatset. The loss values are defined as:

- mean squared error:

$$MSE(\boldsymbol{\phi_{pred}}, \boldsymbol{\phi_0}) = \frac{1}{L}\sum_{k}^{L}\sum_{i}^{N}\left(log_{10}\big(\phi_{pred,k}(E_i)\big) - log_{10}\big(\phi_{0,k}(E_i)\big)\right) \tag{14}$$

- mean pairwise squared error:

$$MPSE(\boldsymbol{\phi_{pred}}, \boldsymbol{\phi_0}) = \frac{1}{L}\sum_{k}^{L}\sum_{i}^{N}\sum_{q}^{N}\left(log_{10}\left(\frac{\phi_{pred,k}(E_i)}{\phi_{pred,k}(E_q)}\right) - log_{10}\left(\frac{\phi_{0,k}(E_i)}{\phi_{0,k}(E_q)}\right)\right) \tag{15}$$

The neural networks in this investigation differ from the typical neural network, in that the latter has fewer labels than features output ($N \leq M$); and that, since the features is related to the labels via a physical process, the inverse function for turning labels back into features exist (equation 8), and is assumed to be deterministic.

This allows for an additional information to be supplied to the neural network during the training stage:

- mean squared error including folded reaction rates:

$$MSE_{\text{including\_folded\_reaction\_rates}} = MSE(\boldsymbol{\phi'_{pred}}, \boldsymbol{\phi'_0}) \tag{16}$$

- mean pairwise squared error including folded reaction rates:

$$MPSE_{\text{including\_folded\_reaction\_rates}} = MPSE(\boldsymbol{\phi'_{pred}}, \boldsymbol{\phi'_0}) \tag{17}$$

Where ϕ' is the $\boldsymbol{\phi}$ and $\boldsymbol{Z}$ vector concatenated together,

$$\boldsymbol{\phi'} = [\phi_1, ..., \phi_N, Z_1, ..., Z_M] \tag{18}$$

and Z is, in turn, obtained by equation 8:

$$\boldsymbol{Z_{pred}} = \underline{\underline{\mathbf{R}}}\boldsymbol{\phi_{pred}}$$
$$\boldsymbol{Z_0} = \underline{\underline{\mathbf{R}}}\boldsymbol{\phi_0}$$

This is analogous to the technique of regularization[**?**] in normal unfolding procedures, where both deviation from the *a priori* spectrum and the reaction rates are calculated and used as the χ^2 value. In this case, the regularization constnat (weight of the neutron flux's deviation relative to the reaction rates' deviation) is simply chosen as 1.

These two metrics will give loss value = 0 when ϕ_{pred} and ϕ_0 matches perfectly; but the neural network will be penalized by an additional amount if it makes a mistakes in the spectrum that leads to a greater deviation of the Z_{pred} from the Z_0 (which is a mistake that other linear/non-linear least-square unfolding codes such as MAXED and GRAVEL will not make. This effectively incorperate some physics into the neural network with the hopes of improving its accuracy.

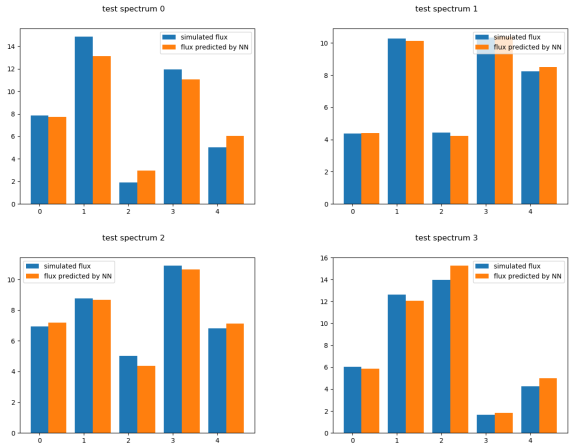


Figure 4: The spectra predicted by a 2-hidden-layers neural network (with 16 nodes per layer) on a fully determined system.

3 Literature review

Previous attempts of machine learning techniques applied to unfolding only include:

- Bonner spheres
- genetic algorithms

Problem: we don't have as much data as in fission. [11] [23]

[8] [10] [4] [30] [28] Therefore the neural network approach to unfolding the few-channel case for fusion purpose is novel.

4 Proof of concept on simulated spectra

For the simple 5×5 case, did perfectly well.

But once we take log of both sides, it's become an impossible problem because we can't take log of a matrix that's singular. See Figure 4.

4.1 Creating simulated spectra by data augmentation

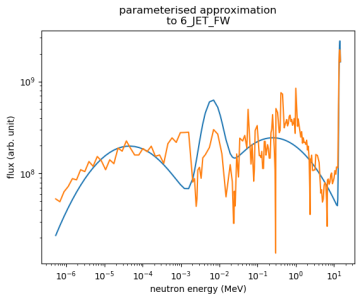
To demonstrate that the neural network is capable of doing the unfolding procedure in an underdetermined condition, 1800 spectra were made to train the neural network. Further increase in complexities (number of layers and numebr of nodes per layers) yield no significant improvement in the fit. This is because increasing the complexity further will simply lead to over-fitting, as the neural network 'memorizes' the relationship between features and labels. This was reflected by the continual decrease in the loss value over the training dataset, while the loss value over the validation dataset stops improving.

After being trained on variants of the same spectrum (only with slighly different peak heights), along with the perturbed spectra of 6 other fusion neutron sources.

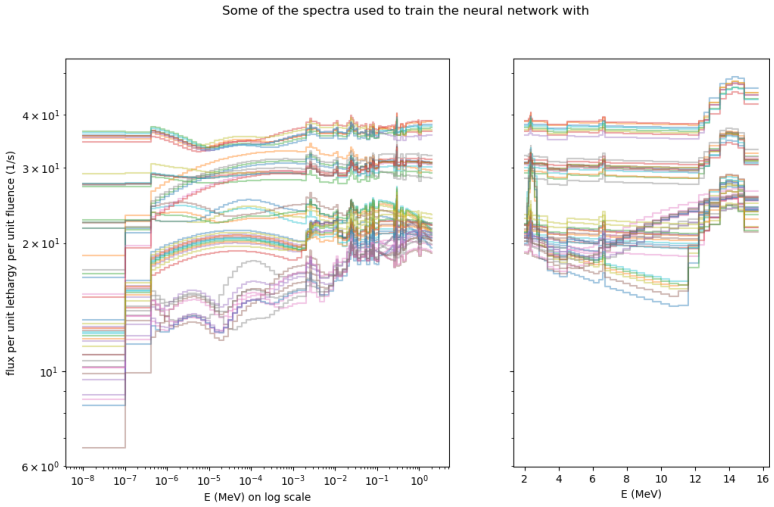
It should be trying to approximate the underlying pattern, In deed it does quite well. But it's still not perfect. Though that doesn't matter too much for our purpose. We deem it as good enough; move on.

5 Neural networks trained on Real spectra

IF there's an underlying pattern, it will pick it out. It performs just as well.

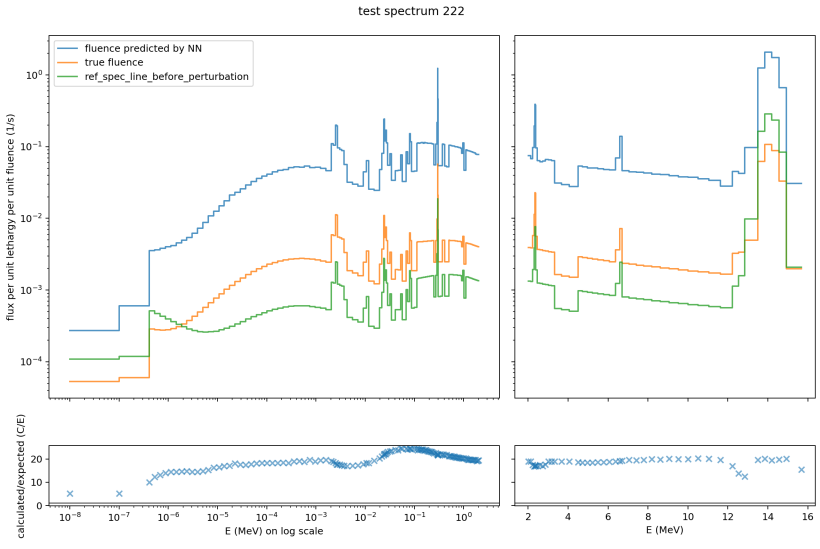


(a) An example of parametrisation performed on the JET spectra. These flux values are the total flux inside each energy bin, *not* divided by the lethargy span of each bin, so they are higher/lower in wider/narrower energy bins.

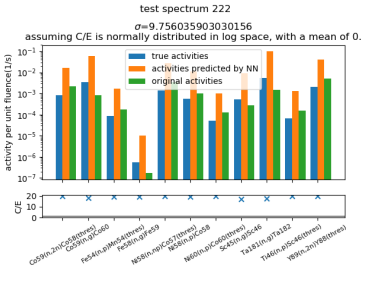


(b) 300 × perturbed spectra were generated for each of the 6 original fusion spectra and plotted here, in flux per unit lethargy.

Figure 5: Data augmentation performed to create simulated spectra.



(a) The neural network's predicted fluence was not normalized unlike the other two, therefore it was scaled up by a constant factor relative to the true fluence.



(b) The activities (features) used. The neural network was given the true activities, and asked to predict the fluence (6a).
Note that the colour scheme was reversed, i.e. the blue bars denote the reaction rates predicted by the neural network instead of the true reaction rates, and vice versa.

Figure 6: The neural network's attempt at predicting a perturbed JET first wall spectrum.

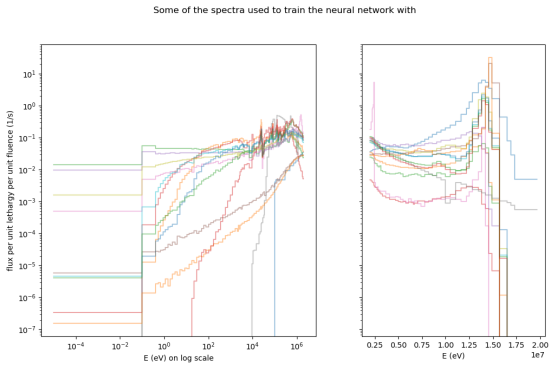


Figure 7: All fusion spectra used, obtained from [**?**]

5.1 How Neural Networks are applied in this investigation

Since the of each neural network varies according to the hyperparameters used and the data that it is trained on, when investigating the real fusion spectra in section 5, multiple neural networks were generated, each with different hyperparameters, to investigate the combination of optimal hyperparameters which may be applied onto this problem.

The following hyperparameters/variables were considered:

- activation function used
- strategies applied to prevent overfitting
  - weight regularization
  - dropout
- number of layers
- number of nodes in each layer
- number of epochs trained
- learning rate
- Normalization techniques applied
-
- Take log of features
- metric used to evaluate the loss value (See Section 13)
- Training set/testing set

After some consideration, the following choices were made for each of the hyperparameter:

- The choice of number of nodes in each layer was chosen as 32 for the first hidden layer, and then logarithmically increased up to 256. This number takes into consideration the results in [22], where a 7 reaction rates input, 10 hidden neurons, 75 output energy bins were found to be optimal for a single layer neural network unfolding; and the fact that most typical neural network research numbers were.

The However, if a grid search for optimal hyperparametres were considered, then an optimal grid search over all

Acquired from the IAEA+UKAEA compendium[25].

5.2 Results of predicting using the real spectra

Rebinning

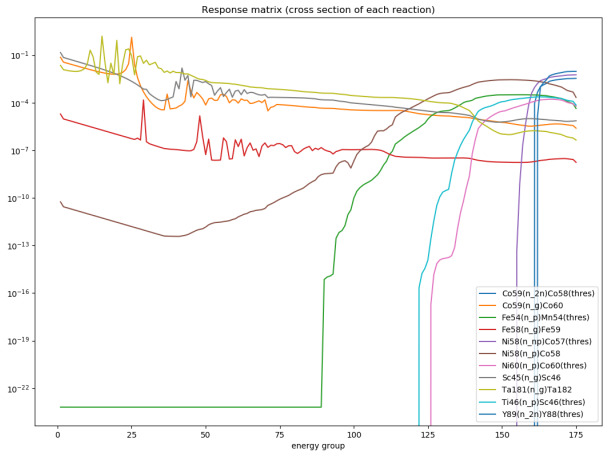


Figure 8: Microscopic cross-section of each reaction
These values are obtained from TENDL15 via FISPACT-II [9] at the left edge of each bin in the Vitamin-J group structure.

The neural networks were trained on 19 fusion neutron spectra, which were all rebinned into the Vitamin-J group structure. How did you sort it

By assuming that the flux per unit lethargy inside each bin are relatively flat (energy independent), the reaction rates contributed by the neutron flux in each bin is then proportional to the product of neutron flux with the microscopic. Symbolically,

$$Z_j \propto \sum_i \sigma_{ij}\phi_i \tag{19}$$

Akin to the equation 4. Therefore these microscopic cross-section values were used in place of the response function for each of the reaction, assuming the constant of proportionality in equation 19 is unity.

The experiment was repeated using other metrics as the loss values. The complete list of tested loss values are listed below:

- mean-squared-error (rms) = $\sum_i$
- mean-pairwise-squared-error
- mean-squared-error-including-folded-reaction-rates
- mean-pairwise-squared-error-including-folded-reaction-rates

This shows that Figure 10 likely only achieves the above average performance serendipiteously by re-tracing the same average spectrum. This hypothesis is supported by it replicating a very similar spectrum when it attempts to deduce the spectra corresponding to the other two test data.

- What work has been done to find the optimal
- How to quantify optimal

5.3 potential future improvements

1. In the future we can calculate the spectral index, and plug THOSE values in, instead of plugging in the direct values, it might be better at picking out these differences, because it may be more obvious, and can pick it out even under so few data.

Use Orthogonal Arrays instead of grid searching the entire hyperparameter space, as well as fractional factorial instead of full factorial combinations, as proposed in [22], when performing the optimization. This can reduce the amount of time required for

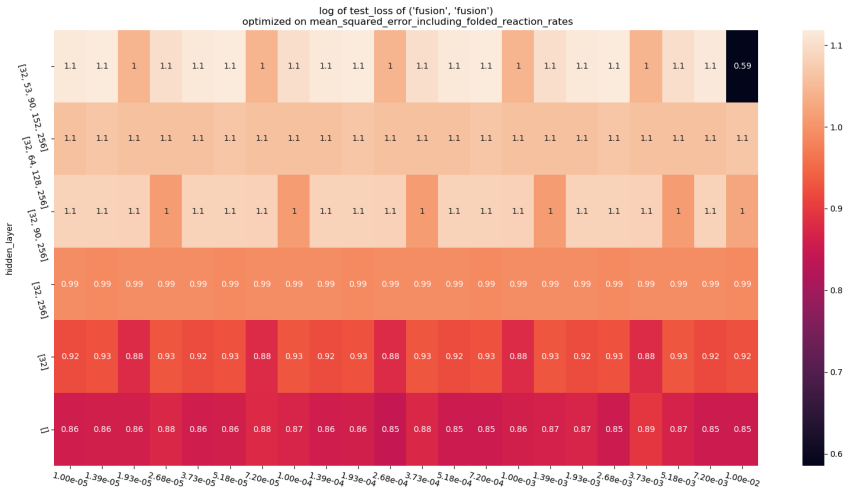


Figure 9: Heatmap visualizing the loss values of the nerual networks' prediction on the test dataset.

Each square represents a neural network with a particular set of hyperparameters, i.e. learning rate and number of layers. The learning rate increases logarithmically across the x-axis; while the number of layers increase linearly across the y-axis. The number of nodes per layer is increased logarithmically from 32 to 256 (if the number of layers ≥ 2). Neural networks which performs better has lower loss values, and are represented with darker colours.

At the top right hand corner, i.e. the neural network with the hyperparameters of (learning rate=0.01, number of nodes in each layer=[32,53,90,152,256]), a particularly loss value is obtained. Therefore this neural network is regarded as the neural network with the optimal hyperparameter, and further investigation into the predictions of this neural network is conducted.

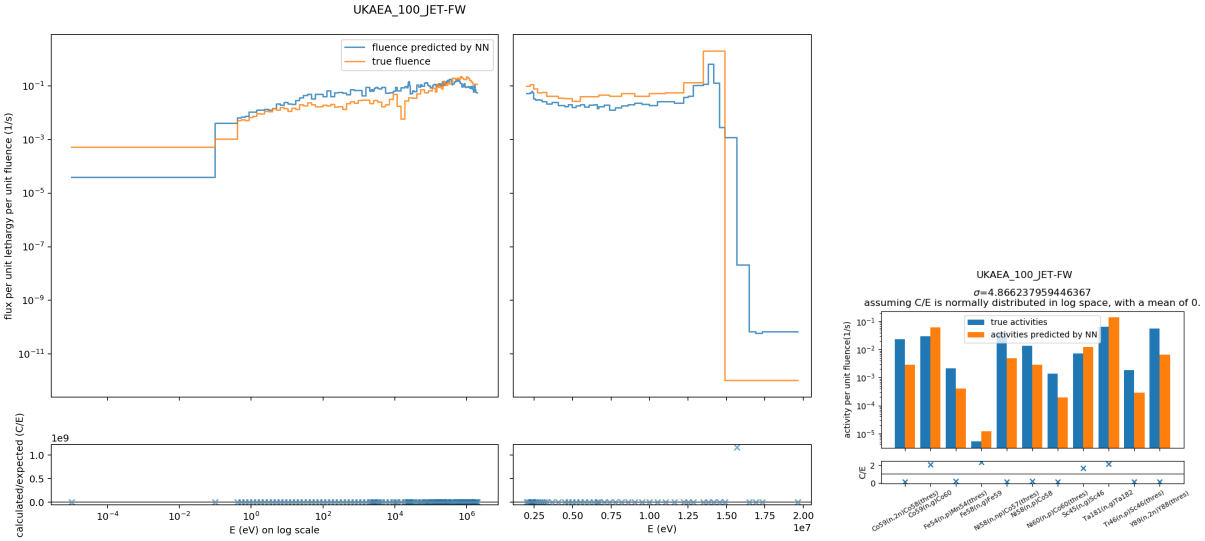


Figure 10: JET first wall spectrum as predicted by the optimally performing NN among all NN trained on fusion data.

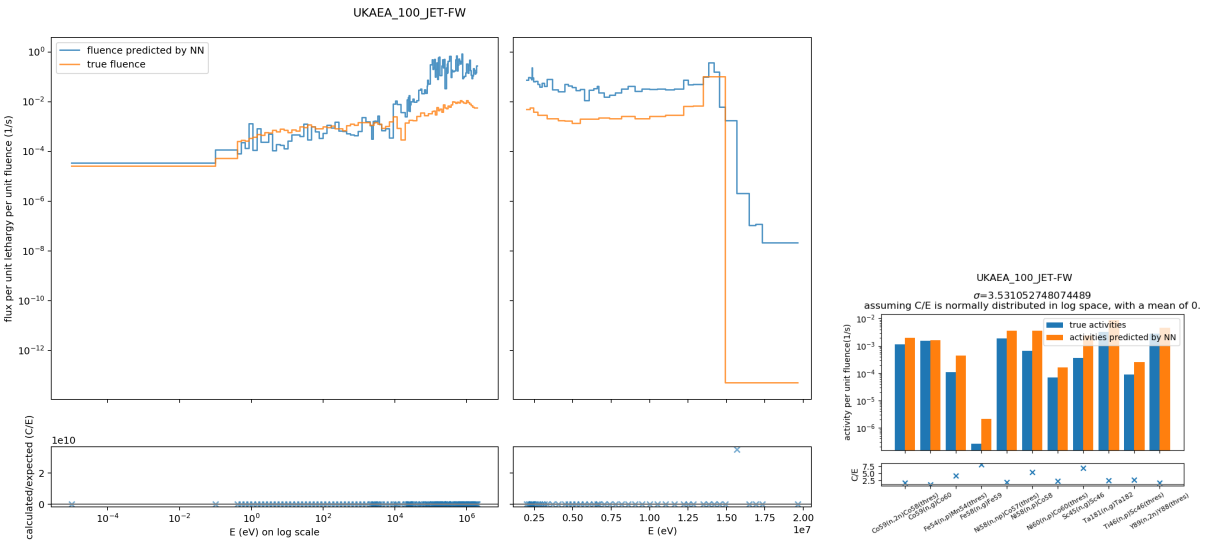


Figure 11: JET first wall spectrum as predicted by the second best performing NN among all NN trained on fusion data.

the experimentation; or extend the range of hyperparameter space searched in the same amount of time; multiple dimensional space can be search through in the same manner as well.

Use RBF-NN.

6 Benchmarking against existing codes

6.1 As an unfolding tool

If they would like to use it directly as an unfolding tool, then they can incorperate the whole folding process into the loss function; but this method requires:

- (optional) The response matrix to already been known → better results?

*Gotta make a fair comparison between a neural network unfolded against an a priori unfolded one.

The more exciting aspect arises from the fact that it can be used as an a priori generator code:

6.2 As an a priori generator

Can be used as the a priori generator?

But it also sucks as an a priori generator, giving , which is only a marginal improvement on the 10.188233 stated in Figure 13

Which is almost as bad as using a naive prior, i.e. using a flat a priori and thus giving no meaningful information to gravel before unfolding (Figure 13). In this case it achieves an average mean squared error of

- the user doesn't want to commit to hours of MCNP model generation (cite a paper where Lee Packer's group has used a whole MCNP model to get the response matrix and the reaction rates);
- and already has a few similar neutron spectra to pick from;
- want a higher reproducibility/credibility than hand-drawing an a priori with reference to the previous spectra/ averaging over the existing spectra.

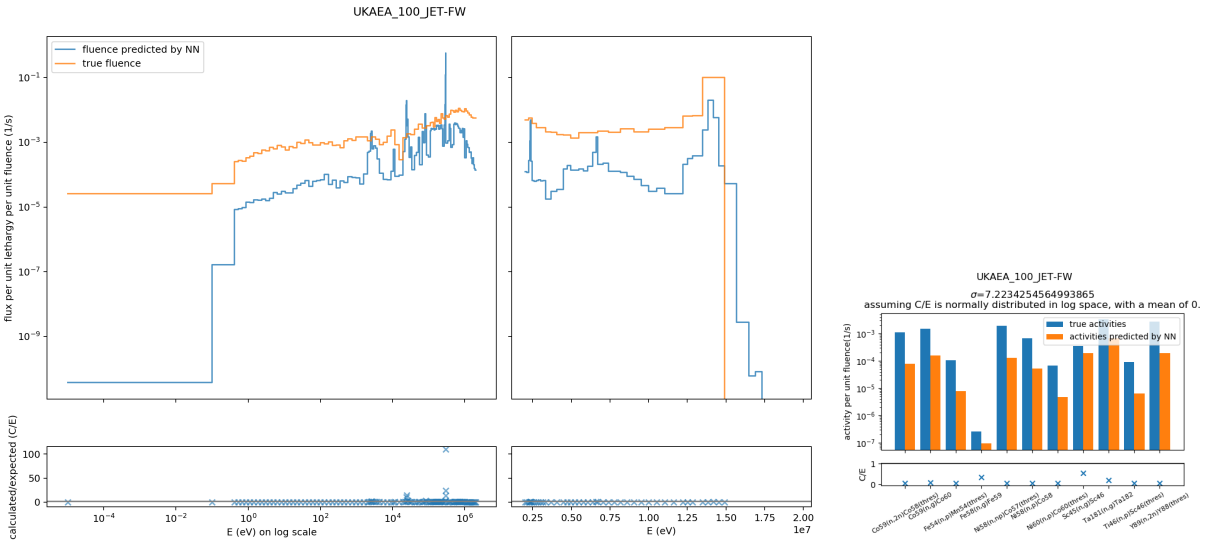


Figure 12: JET first wall spectrum as unfolded by GRAVEL upon using the NN's output as the *a priori* spectrum.

An average rms value of 9.89936 was achieved when using the neural network's prediction as the *a priori* for unfolding by gravel.

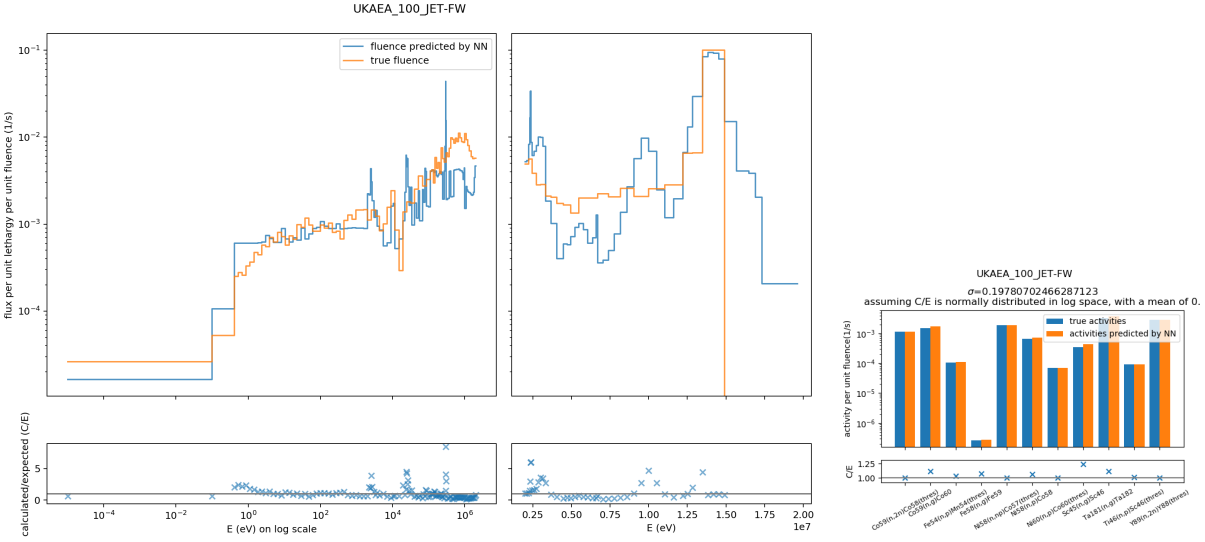


Figure 13: JET first wall spectrum as unfolded by GRAVEL upon using a flat a priori as the *a priori*

A rms value of 10.188233 when comparing the unfolded spectra against the true spectra in log space.

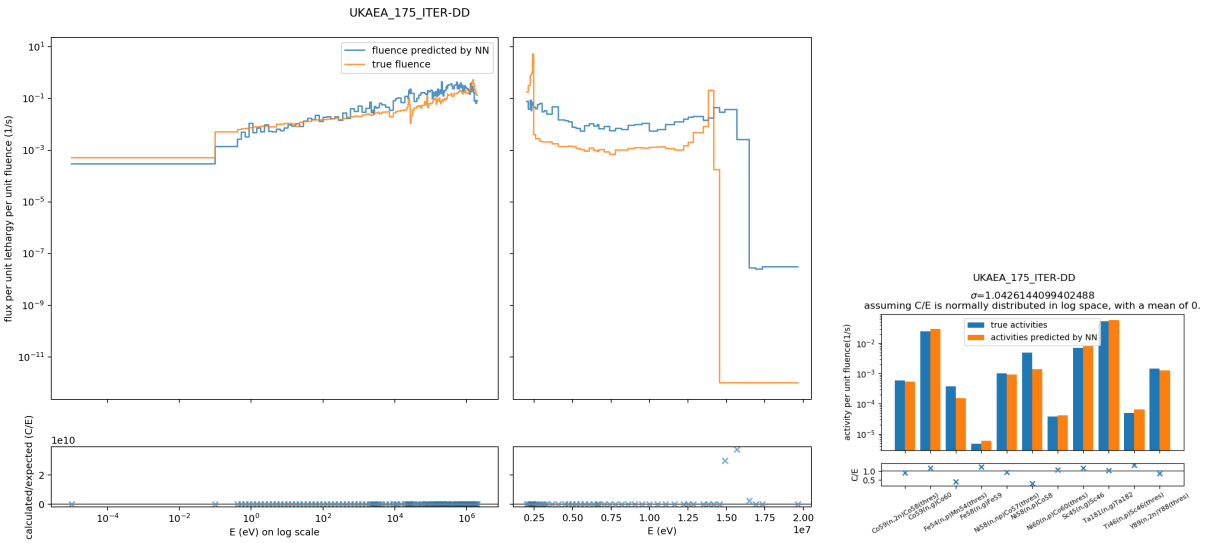


Figure 14: (calculated) ITER spectrum as predicted by the optimal NN among all NN trained on fission spectra

EVEN if the response matrix is not known.

Allows for a probability distribution of weights? does that account for the variance and covariance between the labels and features? * But this is beyond the scope of this paper, which is to demonstrate that the idea of NN works.

7 An attempt at using fission data to predict fusion data

Explain why: we have so many fission spectra... but only very few fusion spectra

But even the best neural network trained on fission spectra and obtained the lowest loss value when tested on fusion spectra gave very poor results:

For the record, this optimal neural network has the hyperparameters of (learning rate = 0.01, number of nodes in each layer=[32,53,90,152,256]).

8 Future improvements

- Transfer learning: start with fission data, fix the weights of the second half of the matrix as it gives the connection
- Use RBF NN or GRNN, which are known to perform better under low sample number conditions, though it is more complicated to implement.
- infer the uncertainty (σ) associated with the neural network's prediction using Monte Carlo method

9 Conclusion

- What's the loss values
- achieved using what topology of NN
- trained upon what data
- How does it compare to neutron spectrum unfolding using other methods

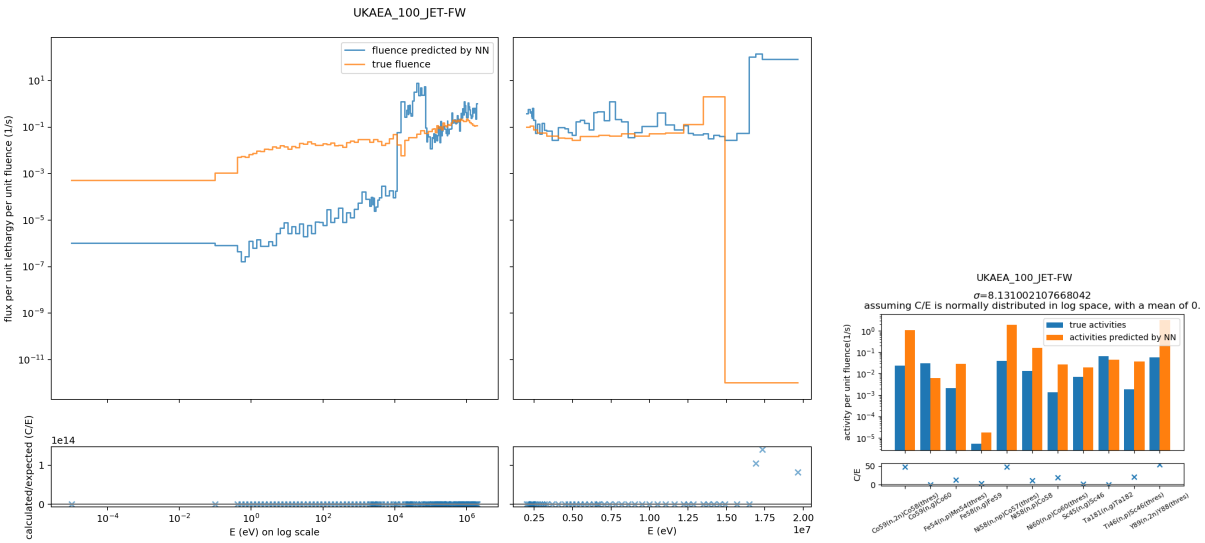


Figure 15: JET first wall spectrum as predicted by the optimal NN among all NN trained on fission spectra

- What's the significance on the unfolding community: should they use it more? Should they improve upon it?
- what additional observation did you find regarding training on different dataset.

References

[1] explore overfitting and underfitting tensorflow core 2019, 2019.

[2] AV Alevra and DJ Thomas. Neutron spectrometry in mixed fields: multisphere spectrometers. *Radiation Protection Dosimetry*, 107(1-3):33–68, 2003.

[3] P Batistoni, D Campling, Sean Conroy, D Croft, T Giegerich, T Huddleston, X Lefebvre, I Lengar, S Lilley, A Peacock, et al. Technological exploitation of deuterium–tritium operations at jet in support of iter design, operation and safety. *Fusion Engineering and Design*, 109:278–285, 2016.

[4] Cláudia C Braga and Mauro S Dias. Application of neural networks for unfolding neutron spectra measured by means of bonner spheres. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 476(1-2):252–255, 2002.

[5] FD Brooks and H Klein. Neutron spectrometryhistorical review and present status. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 476(1-2):1–11, 2002.

[6] Roberto Capote, Konstantin I. Zolotarev, Vladimir G. Pronyaev, and Andrej Trkov. chapter Updating and Extending the IRDF-2002 Dosimetry Library, pages 197–209. ASTM International, West Conshohocken, PA, Aug 2012.

[7] Bethany Colling, P. Batistoni, S.C. Bradnam, Z. Ghani, M.R. Gilbert, C.R. Nobs, L.W. Packer, M. Pillon, and S. Popovichev. Testing of tritium breeder blanket activation foil spectrometer during jet operations. *Fusion Engineering and Design*, 136:258–264, 2018.

[8] G Fehrenbacher, R Schütz, K Hahn, M Sprunck, E Cordes, JP Biersack, and W Wahl. Proposal of a new method for neutron dosimetry based on spectral information obtained by application of artificial neural networks. *Radiation protection dosimetry*, 83(4):293–301, 1999.

[9] Culham Centre for Fusion Energy. Ukaea: Fistact-ii, 2017.

[10] Seyed Abolfazl Hosseini. Neutron spectrum unfolding using artificial neural network and modified least square method. *Radiation Physics and Chemistry*, 126:75–84, 2016.

[11] T. Kin, Y. Sanzen, M. Kamida, K. Aoki, N. Araki, and Y. Watanabe. Artificial neural network for unfolding accelerator-based neutron spectrum by means of multiple-foil activation method. In *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2017 - Conference Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2018.

[12] A Klix, A Domula, U Fischer, D Gehre, P Pereslavtsev, and I Rovni. Neutronics diagnostics for european iter tbms: Activation foil spectrometer for short measurement cycles. *Fusion Engineering and Design*, 87(7-8):1301–1306, 2012.

[13] Slobodan Krevinac. *Neutron energy spectrum unfolding method*. PhD thesis, University of Birmingham, Birmingham, 1971.

[14] C. R. Nobs S. Bradnam B. Colling K. Drozdowicz S. Jednorog M. R Gilbert E. Laszynska D. Leichtle J.W.Mietelski M. Pillon I.E. Stamatelatos T. Vasilopoulou A. Wjcik-Gargula L. W. Packer, P. Batistoni and JET Contributors. Act - activation of real iter materials report on deliverables f21-29. 6 2019.

[15] Igor Lengar, Alja Ufar, Vladimir Radulovi, Paola Batistoni, Sergey Popovichev, Lee Packer, Zamir Ghani, Ivan A. Kodeli, Sean Conroy, and Luka Snoj. Activation material selection for multiple foil activation detectors in jet tt campaign. *Fusion Engineering and Design*, 136:988–992, 2018.

[16] M. Matzke. Unfolding procedures. *Radiation Protection Dosimetry*, 107(1-3):155–174, 2003.

[17] Manfred Matzke. Unfolding of pulse height spectra: the hepro program system. Technical report, SCAN-9501291, 1994.

[18] Manfred Matzke. Unfolding methods. *Physikalisch-Technische Bundesanstalt, Germany*, 2003.

[19] WN McElroy, S Berg, T Crockett, and RG Hawkins. A computer-automated iterative method for neutron flux spectra determination by foil activation. volume 1. a study of the iterative method. Technical report, ATOMICS INTERNATIONAL CANOGA PARK CA, 1967.

[20] Alberto Milocco. Neutron radiation damage in ccd cameras at joint european torus (jet). *Radiation protection dosimetry*, 180(1-4), 2017.

[21] Andrew Ng. Normalizing inputs (c2w1l09), 2017.

[22] Jose Manuel Ortiz-Rodriguez, Ma del Rosario Martinez-Blanco, Jose Manuel Cervantes Viramontes, and Hector Rene Vega-Carrillo. Robust design of artificial neural networks methodology in neutron spectrometry. In *Artificial Neural Networks-Architectures and Applications*. IntechOpen, 2013.

[23] J.M. Ortiz-Rodrguez, A. Reyes Alfaro, A. Reyes Haro, J.M. Cervantes Viramontes, and H.R. Vega-Carrillo. A neutron spectrum unfolding computer code based on artificial neural networks. *Radiation Physics and Chemistry*, 95:428–431, 2014.

[24] L. W. Packer, P. Batistoni, S. C. Bradnam, S. Conroy, Z. Ghani, M. R Gilbert, E. Laszyska, I. Lengar, C.R. Nobs, M. Pillon, S. Popovichev, P. Raj, I.E. Stamatelatos, T. Vasilopoulou, A. Wojcik-Gargula, R. Worrall, and JET contributors. Neutron spectrum and fluence determination at the iter material irradiation stations at jet. 5 2019.

[25] Lee W. Packer, Mark R. Gilbert, Jonathan Naish, and Steven Bradnam. Ukaea-r-17-nt1 unfolding code support: progress report. 3 2017.

[26] LW Packer, P Batistoni, SC Bradnam, B Colling, Sean Conroy, Z Ghani, MR Gilbert, S Jednorog, E Łaszyńska, D Leichtle, et al. Activation of iter materials in jet: nuclear characterisation experiments for the long-term irradiation station. *Nuclear Fusion*, 58(9):096013, 2018.

[27] M. Reginatto and P. Goldhagen. Maxed, a computer code for maximum entropy deconvolution of multisphere neutron spectrometer data. *Health Phys.*, 77(5):579–83, 1999.

[28] H. Shahabinejad and M. Sohrabpour. A novel neutron energy spectrum unfolding code using particle swarm optimization. *Radiation Physics and Chemistry*, 136, 2017.

[29] John Shore and Rodney Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on information theory*, 26(1):26–37, 1980.

[30] Vitisha Suman and P.K. Sarkar. Neutron spectrum unfolding using genetic algorithm in a monte carlo simulation. *Nuclear Inst. and Methods in Physics Research, A*, 737:76–86, 2014.

[31] D.B. Syme, S. Popovichev, S. Conroy, I. Lengar, and L. Snoj. Fusion yield measurements on jet and their calibration. *Nuclear Engineering and Design*, 246(C):185–190, 2012.