Gamma Enrichment Analysis Algorithm Based on Physics-Informed Neural Networks

Marcus J. Neuer[1,2] and Christian Henke[1]

[1] innoRIID GmbH, Merowingerstr. 1, 40237 Duesseldorf, Germany, http://www.innoriid.eu
[2] RWTH Aachen,Turmstrae 46, 52064 Aachen, Germany, http://www.rwth.de

Abstract. A gamma enrichment analysis is presented that utilises physics-informed neural networks (PINN). Such networks exploit existing mathematical formulas and analytic transformations, to provide the best possible starting for a machine learning training - they hybridise the machine learning perspective with established analytic relationships. Autoregressive-PINN structures are presented. They are a tool for nonlinear preparation of the spectral data and are used for extracting characteristic enrichment features.
Primary reason for using a physics-informed approach was the reduced amount of training data needed by this learning method, which de-facto yields good results even on sparse data sets. Tests of the algorithm were performed with Sodium Iodide (NaI:Tl), Lanthanum Bromide (LaBr3:Ce,Sr) and Cadmium Zinc Telluride (CZT) spectra. The algorithm was applied to measurement data from real sources and to simulated data conducted with GEANT4. The results for the different detectors are compared.
The physics-informed approach yields advantages, because it can include complex peak shapes and the variation of the shape during the training. Interpreting the peak shape as probability distribution, the variational autoencoding represents an effective uncertainty quantification of the moments of this distribution. This includes kurtosis and skewness in the context of asymmetrical CZT peak structures. The tested PINN is studied by sensitivity analysis methods. It allows us to quantify the influence of each input data contribution: scattering, background, spectrum content in specific regions-of-interest or temperature.

Keywords: machine learning, uncertainty quantification, stochastic dynamics

1 Introduction

Machine learning (ML) has brought diverse solutions to different industrial areas. Most people consider these techniques as theory-agnostic black-boxes. This might be true for some applications, were the influence of the input data plays only a smaller role. A perspective that allows deeper understanding of the learning algorithm, clearer explaination of its decision making and a direct provision of prior knowledge is physics-informed machine learning, as it is e.g. showed by Karniadakis et al. [1], Raissi et al. [2] and Nascimento et al. [3]. The latter works include rigorous integration e.g. differential equations in the topology of a neural network.

The basic tenor behind this flavour of machine learning is: Why should we hide knowledge from our machine learning codes, that scientists and experts worked out for decades?

This question emphasizes the idea of physics-informed learning, that we should provide prior knowledge to the ML system rather than hide it. With the prior knowledge, training approaches are more robust, converge faster and require less amount of training data. This can be understood by the fact, that the algorithm does no longer have to train *everything*, instead some important fragments of a solution are supplied by the user.

We apply this approach to the problem of gamma enrichment analysis. This has been a longstanding field of research with several publications, among which the work of Gunnink et al. [4] or Berlizov [5] are just two examples of high precision gamma estimation. Especially the work of Berlizov yields several theoretical cornerstones to tackle a good reproduction of the Uranium enrichment level. In our paper, we restrict to multi-purpose solution for enrichment estimates. These estimates are important for handheld radiation detection instrumentation, as well, as portal solutions. They do not necessarily offer specialised geometries or optimization for their measurements. Yet, the users of those instruments still wish reasonable accurate enrichment information, in case they find special nuclear material (SNM) like Uranium.

We think that our solution can be combined with those methods that already exist, in order to increase precision further or simply, to adapt quicker to new detector types.

2 AI Gamma enrichment analysis algorithm

Our solution approach consists of two mayor elements: a) an Autoencoder network and b) a neural network regressor. Each is trained separately, but the regressors requires the AE to be trained in advance. In the following, we will discuss both architectures in a bit more detail.

2.1 Autoencoder (AE)

A first element of our enrichment code is based on autoencoding. Autoencoders are a special form of neural networks. They are unsupervised learning paradigms, thus do not need labels. The training data is used as input and as desired output (label). The AE is therefore a **1**-mapping of its input. Formally we define it as follows:

Definition 1 (Autoencoder). *We say, a neural network $\mathcal{A}$ operator is called an Autoencoder, if*

$$\mathbf{1}\mu = \mathcal{A}\mu, \tag{1}$$

for all spectra $\mu \in \mathbb{T}$ of the training set $\mathbb{T}$.

If we now design the network $\mathcal{A}$ in a way, that the first and the last layers have the dimension of the input data and the middle layer is substantially smaller, all information that is processed by the AE must pass the middle layer bottleneck. If the reconstruction in the output layer is good, this refers to a situation, where all information is conserved, although a severe reduction of dimensionality has happened. In that sense, the transformation in the AE is highly nonlinear and typically outperforms related methods like the PCA, as any complex dependency can be integrated into the network.

We formalise this concept further, as we would like to explicitly use the middle layer, also known as the latent layer, in our subsequent processing. Therefore we split the operation of the network in two logical partitions:

Definition 2 (Decomposition). *We can decompose an autoencoder $\mathcal{A}$ into an encoding partition $\mathcal{E}$ and an decoding partition $\mathcal{D}$, such that*

$$\mu = \mathcal{D}\left[\mathcal{E}(\mu)\right] \tag{2}$$

.

We say $\tilde{\mu} = \mathcal{E}(\mu)$ are the latent or encoded channels of the spectrum μ.

As mentioned before, due to the specific topology of the Autoencoder, the encoded dimension is lower, than the original spectrum dimension. In our case, 300 channels per spectra are used for μ and those channels are effectively reduced to 7 latent channels. The information contained in the original spectrum is therefore compressed into these 7 channels, where, if the AE is designed correctly, a maximum of information is still conserved.

2.2 Neural network regressor

The AE is the preprocessor stage for a neural regressor. This regressor is a physics-informed approach that intermixes the AE latent channels with information about the peak counts in the 186keV, the background counts and the variance of the spectrum. The input coming from the AE reflects the shape of the spectrum. Together with the other information, the regressor has all necessary base information to predict the enrichment r.

Definition 3 (Enrichment regressor). *The neural network regressor $\mathcal{N}$ performs the calculation of the enrichment r,*

$$r = \mathcal{N}(\tilde{\mu}, A_{186}, A_{BG}, \sigma_\mu). \tag{3}$$

This is the primary predictor algorithm. It works together with the AE.

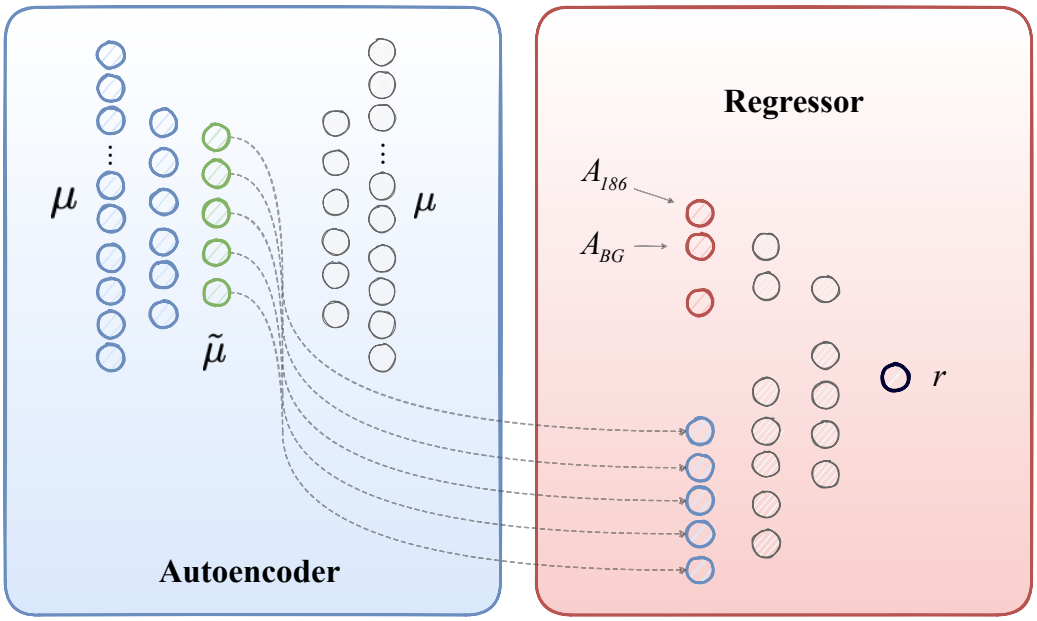

Fig. 1: Concept of AI-GEM algorithm. Left: Autoencoder prestage, with dimensionality reduction of the U-spectrum. Right: Regressor stage to predict the enrichment ratio r.

In Fig. 1 the algorithm is illustrated, yet with simplified topologies. The AE acts as prestage. The latent channels $\tilde{\mu}$ are used as intermediate result. They are the input for the neural regressor.

We tested different forms of regressors including also recurrent networks with long-short-term-memory (LSTM) [6] and mixture-density networks [7], among which the latter yield the favourable advantage that they also predict the uncertainties of their calculation. For the given presentation, we restrict to a normal regressor, as it explains the actual concept already completely.

We also developed a variational Autoencoder in addition to the demonstrated common variant of an AE. The differences in the final performance depend on the statistics of the input data. For low statistics in the input spectra, variational AE and mixture-density networks have an advantage compared to the normal variants. But as our input data have a sufficiently high number of counts, we demonstrate the application with common variants of AE and regressor.

At the input stage of the neural networks, also multiple additional information can be added to the processing: e.g. parameters for the specific Pearson-IV peak shaping for CZT

spectra or details on the resolution-energy dependency. In this case, skewness and kurtosis of the CZT peak distribution are provided to the AE in two additional neurons of the input (and symmetrically also to the output layer).

2.3 Training and test data

We gathered spectra of Uranium in different enrichment states. This data collection took multiple years, where we attended different data taking exercises. The enrichment levels we primarily measured were: Depleted Uranium (DU) with 0.3% ^{235}U, Natural Uranium, Lightly Enriched Uranium (LEU) with 4.4% ^{235}U, Lightly Enriched Uranium (LEU) with 7% ^{235}U, Lightly Enriched Uranium (LEU) with 14.4% ^{235}U, Lightly Enriched Uranium (LEU) with 16% ^{235}U, High Enriched Uranium (LEU) with 20% ^{235}U and Highly Enriched Uranium (LEU) with 80% ^{235}U.

For each detector type we use ca. 40 base measurements of these enrichment degrees. These 40 real measurements are extended by Monte-Carlo simulations based on the measurements and mixture compositions. Based on these data we can calculate interpolations, so that a smooth range of enrichment values are covered. In the course of this work, we restrict to a scale from 0.3% to 20% enrichment.

Two data sets are prepared, a training data set with 3800 spectra and a test data set with 100 spectra. All spectra are calibrated using the 186keV peak of ^{235}U prior to evaluation.

2.4 Implementation

Both networks are set in Tensorflow [8] and Keras [9] in a Python environment. We use an AE with the following topology [300, 133, 78, 11, 78, 133, 300] and a neural network with [5, 25, 25, 25, 1]. All layers are ReLU-activated except the final layers, which use a Leaky-ReLU activation. We train with a learning rate of 0.001 and use the ADAM optimizer. Bias was switched on for all layers as well.

The successful training requires ca. 1000 epochs on our 3800 training spectra in $\mathbb{T}$. We also use cross-validation with 20 folds, applying the tools from Scikit-Learn. Both networks are trained successively. First, we use the training set to train the the AE. Once this network converged into a good mapping, we test the AE with the dedicated test data set.

We then process the full training data set with the AE and introduce the according enrichment rate r as desired output label for the regressor. Next, the regressor is trained with these data. Note, we also encoded all test data, as we need the test data set in the cross-validation of the regressor.

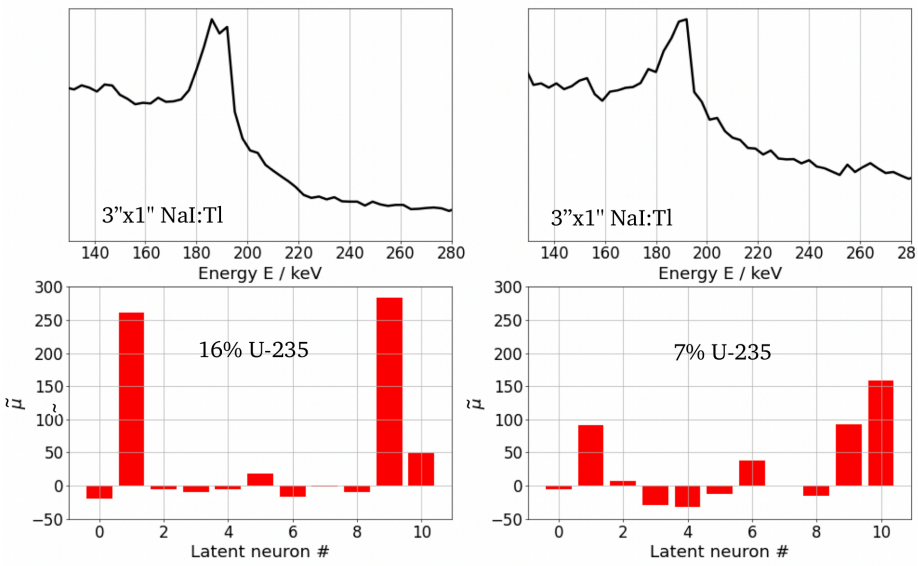

Fig. 2: Top: Two spectrum snippets of the 186keV line area for 17.8% and 20% enrichment, measured with a CZT detector. Bottom: Latent spectra of those examples. The spectra are both normalised. The AE just works on the shape deviations. Differences can be seen in latent neurons 5, 6, 7, 8, 9 and 10. Similarities are encoded in the first 3 latent neurons.

3 Results

As a first result, we show the autoencoder transformation $\tilde{\mu}$ of the measurement. In Fig. 2 $\tilde{\mu}$ for the spectra of a NaI:Tl detector are shown. The spectra that are transformed are real measurements of a 16% enrichment situation (left) and a measurement of 7% enrichment with ^{235}U.

The latent spectrum has 11 channels, respectively neurons. Note the strong difference in the latent space: the two transformations have strong deviations, that give an impressions, how well the AE maps the differences in the original spectra. Please also consider the influence of the axis scaling, which is different in both cases. In fact, the original spectra are normalised. The AE derives its encoding only from the shaping of the region around the 186keV peak.

Figure 3 shows two examples of encoding CZT spectra. In this case, enrichment degrees of 17.8% and 20% ^{235}U were selected. Although the two original spectra are nearly identical (they are not, as they have different peak areas) and normalised, the AE works out a clear difference. The scaling of the

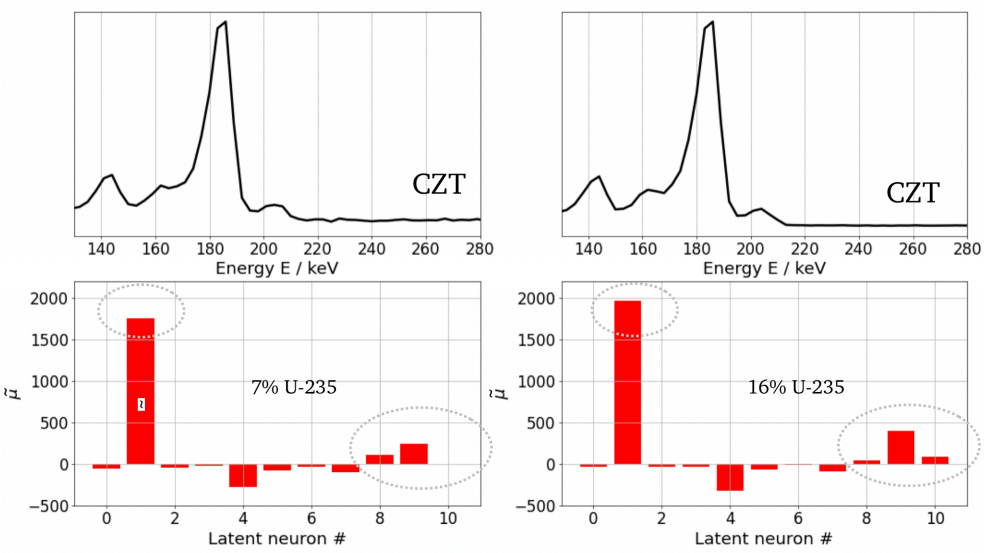

Fig. 3: Top: Two spectrum snippets of the 186keV line area for 17.8% and 20% enrichment, measured with a CZT detector. Bottom: Latent spectra of those examples. The spectra are both normalised. The AE just works on the shape deviations. Differences can be seen in latent neurons 5, 6, 7, 8, 9 and 10. Similarities are encoded in the first 3 latent neurons.

3.1 Determination of the enrichment ratio

Figure 4 shows the results of the combination of AE and regressor for the NaI:Tl detector. Similar results are obtained with for $LaBr_3$ and CZT detectors. The reproduction is of enrichment values r is very good, except some cases where the model deviates slightly. For one specific real world test case (No. 47), the technique predicts a slightly higher contribution of ^{235}U than expected. Some further points also depart from the reference line in Fig. 4. Nonetheless, the results are promising and provide a good technique for calculating the enrichment with handheld instrumentation.

3.2 Discussion

All measurements would benefit from a larger data set. Also Gunnink et al. [4] refer to specific measurement geometries for high precision measurements. In those situations, a tailored tungsten collimation is used for improving the quantification of ^{235}U. As we use arbitrary measurements that were not optimized for the actual enrichment analysis, we

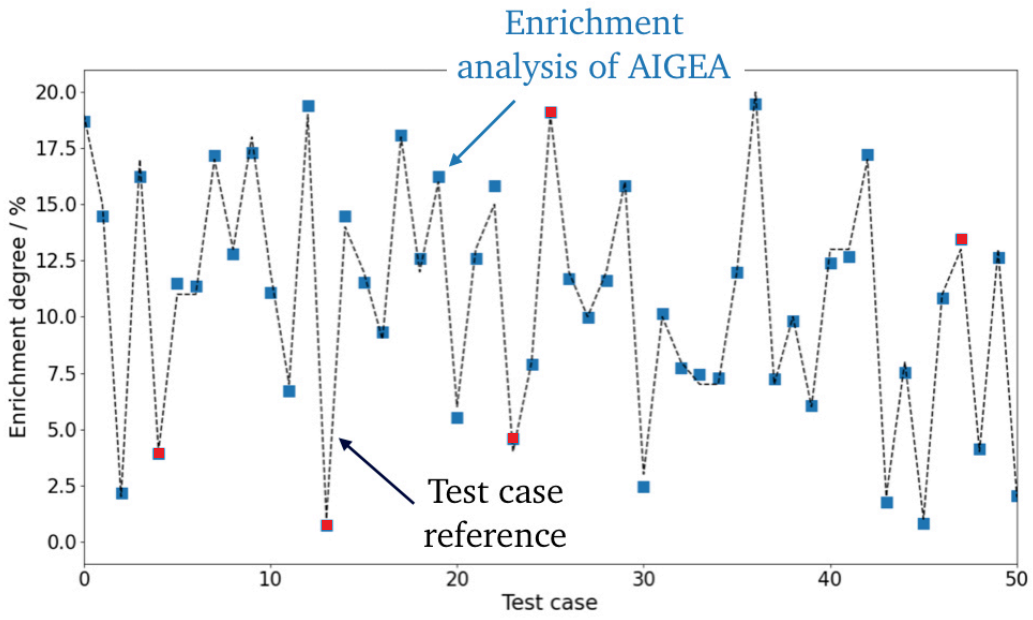

Fig. 4: Calculated enrichment degrees r compared to the real enrichment value of the test case (black line). Red squares indicate predictions for a real measurement, directly from a detector. Blue squares indicate the prediction for a synthetic mixture obtained by Monte-Carlo simulations.

can assume that even better results can be achieved upon using standardised measurement situations.

4 Summary

We showed an enrichment analysis based on machine learning. The algorithm we propose uses an autoencoder prestage, that nonlinearly maps the information stored in the spectrum. It reduces the original spectrum onto a limited number of values (in our case 11). We referred to these values as the transformed, latent spectrum $\tilde{\mu}$. Once this transformed state was achieved, we trained a regression network. This network maps the output of the AE to a floating point value r that reflects the enrichment contribution of ^{235}U. For the separated test set, we find good agreement with the actual enrichment values. The tests also show the sensitivity of the autoencoder to smallest deviations in the original spectrum. The AE emphasizes the most important differences and thus contains the best representation for our

data. The method is integrated in a commercial handheld and delivered to a series of key customers for testing.

References

1. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021.
2. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
3. R. G. Nascimento, K. Fricke, and F. A. Viana, "A tutorial on solving ordinary differential equations using python and hybrid physics-informed neural network," *Engineering Applications of Artificial Intelligence*, vol. 96, p. 103996, 2020.
4. R. Gunnink, R. Arlt, and R. Bernt, "New ge and nai analysis methods for measuring 235u enrichments," *roceeding of the 19th Annual ESARDA Symposium on European Safe- guards Research and Development Association*, 1997.
5. A. Berlizov, "Gem: A next-generation gamma enrichment measurements code," *Journal of Materials Management*, vol. 50, no. 1, pp. 110–120, 2022.
6. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
7. B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1 ed., 2008.
8. 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, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.
9. F. Chollet *et al.*, "Keras," 2015.