

.. .

3ocMethods

3



**Advanced
Reactors &
Fuel
Cycles**

ADVANCED REACTORS AND FUEL CYCLES GROUP
Technical Report Series
ARFC-NPRE-17-00

Inception Neural Networks for Isotope Identification

Prepared for:
INSTITUTE OF REPORTS
CONTRACT NN-NNNN

Prepared by:
Samuel DOTSON
Mark KAMUDA
Prof. Kathryn HUFF

June 5, 2019

Urbana, IL
Department of Nuclear, Plasma, and Radiological Engineering
University of Illinois at Urbana-Champaign

1 Introduction

Developing algorithms that accurately identify the isotope sources of low-resolution gamma-ray spectra will be an important advance over the current isotope identification workflow (Sullivan, 2010). Previous work has shown that artificial neural networks can perform isotope identification using low-resolution gamma-ray spectrometers (Cite: [3], [4], [5]). The purpose of this paper is to introduce a new feature to the existing architecture of the Artificial Neural Network for Spectroscopic Analysis (ANNSA) package and to improve the training data simulations to better emulate background radiation found in real measurements. The new feature is known as an Inception Neural Network (INN) that implements wide convolutional layers with several filters, rather than the typical single-filter layer in a simple convolutional neural network (CNN). The features of a gamma-ray spectrum vary depending on the full width at half max (FWHM) of the photopeaks. Simultaneously applying multiple filters of different sizes allows an INN to capture more features during a single layer than a CNN. We hypothesize that an INN will also be robust to changes in background radiation thereby generalizing the ANNSA framework to more scenarios. We will compare the INN to a simple CNN to determine if the improvement in accuracy is enough to warrant the increased computational complexity. Finally, new training data will be obtained through simulations with GADRAS-DRF (Mitchell and Harding, 2014) software.

2 Theory – Artificial Neural Networks

An artificial neural network (ANN) is a function that maps values from \mathbb{R}_N to \mathbb{R}_K by mimicking biological neurons. Examples of an arbitrary neural net and a single neuron are shown in Fig. 1 and Fig. 2. The sum of the inputs times the weights pointing to a neuron are passed through an activation function, here it is a rectified linear unit,

$$Relu = \operatorname{argmax}(0, x), \quad (1)$$

and then used as the input for the next layer, as shown in Fig. 2. An ANN may be trained by iteratively updating the weights of a network that minimizes an error function E . The weights are updated through back propagation by taking the derivative of E with respect to the weights. The error function minimized during training of the INN is cross-entropy,

$$E = - \sum_c^M y_{o,c} \ln p_{o,c} \quad (2)$$

Eq. 2 is the cross-entropy for a multiclass classification, where there are more than two possible labels for a given input. M is the total number of labels for a given model, in this case it corresponds to 29 radionuclides (ANSI, 2015). Variable $y_{o,c}$ is binary, indicating whether observation, o , has the correct label, c . Variable $p_{o,c}$ is the probability that o is a member of c . The complete INN model is shown in Fig. 3 and Fig. 4. The input for the INN is a 2"x2" NaI spectrum of 1024 channels and the final output is a softmax given by,

$$\text{softmax}(z_j) = \frac{\exp(z_j)}{\sum_{k=1}^k}. \quad (3)$$

The input data is passed through three ‘inception’ layers and, after flattening, the output is passed to a dense layer which gives the final softmax output. Each ‘inception’ layer has a bottleneck, a convolution, and a concatenation (Szegedy, et. al, 2014). The bottleneck is performed to reduce the computational complexity before large convolution filters are applied. A large filter can be factorized into several smaller filters to reduce the number of required multiplications. The convolution layer uses filters to pick features of a spectrum that have local spatial significance, but no long-range relationships. Once the convolution has been done with several filters, in parallel, the outputs of each of those convolutions are concatenated into a single tensor that is passed to the next ‘inception’ layer, or dense layer if the last layer has been reached. Just like a typical CNN (Fig. [5]) the final step after feature identification, is classification. Classification is performed by using a dense, or fully connected, layer that learns weights that correspond to a probability for a certain label. Here, the corresponding labels are radioactive isotopes.

3 Methods

3.1 Training set creation

3.2 Network Structure and Hyperparameters

3.3 Benchmark Techniques

4 Conclusion

References

[1]

[2]

[3]

[4]

[5]

[6]