Accuracy Correlation in Neutron Resonance Reclassification

Ian Snider, SULI intern, Department of Physics, Truman State University, Kirksville, MO 63501
Gustavo Nobre, Mentor, National Nuclear Data Center, Brookhaven National Lab, Upton, NY 11973

Abstract

Current methods for determining the resonance quantum numbers associated with angular momenta and spin are difficult, time consuming, and my not be fully reproducible, often leading to incorrect spin assignments. To solve this problem, we have employed a machine learning (ML) based method to train an algorithm for identifying and reclassifying incorrect neutron spin assignments. We build synthetic data that mimics the statistical properties of real resonances to train the algorithm. Then, we validate the trained algorithm with a set of real In-115 polarized data. **The goal of this project is to analyze the correlation between the training and validation accuracies.** We can then attempt to improve the validation accuracy by adjusting the ML classifier's hyperparameters. We also explored an iterative method in which, under certain conditions, successive reclassifications could incrementally improve the quality of any misclassified resonance sequence.

Introduction

NEUTRON RESONANCES

- A neutron reaction occurs when a nuclei absorbs a neutron, forming a compound nuclei. The compound nuclei then decays, often emitting a gamma ray (capture channel) or neutron (elastic channel).
- In certain energy regions, this will lead to a cross-section resonance.
- In fig. 1, a **resonance peak** is observed when the incident energy from a capture reaction approaches the excitation state of an In-115 energy level.

NUCLEAR DATA & ENDF

- From experiments, scientists can measure cross section resonances and estimate their spin assignments through shape analysis**.**
- The peaks represent different types of waves: s, p, etc, corresponding to resonances with L = 0, 1, etc, respectively.
- These resonance assignments are then compiled into an Evaluated Nuclear Data File (ENDF). This data is then used in many nuclear applications.
- Fitting the peaks is a manual process often done in an irreproducible manner, and thus prone to inaccuracies. This is an opportunity to develop a new approach.

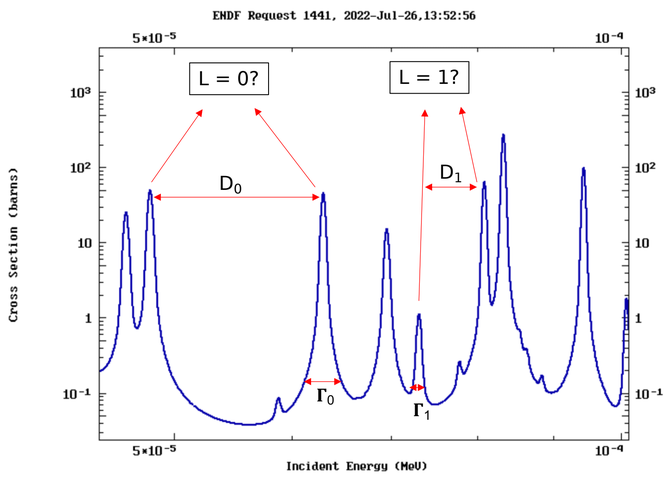


Figure 1: Snippet of nuclear cross section plot for In-115 neutron capture. D and **Γ** are the resonance spacing and mean width, respectively. **Source:** Multi-platform EXFOR-CINDA-ENDF **Credit:** V. Zerkin, IAE-NDS,1999-2022

MACHINE LEARNING & BRR

- Machine learning (ML) is a process that attempts to take data, learn patterns, and make predictions.
- At the National Nuclear Data Center (NNDC), we devised a machine learning method to reclassify misassigned neutron resonances: Bayesian Resonance Reclassifier (BRR).

Methods

ML TRAINING

- To train the algorithm, we need resonances for which we know the correct assignments. So, we train BRR on synthetic data based on real nuclei.
- A perfect data set does not correspond to what we often see in nature, so we purposely misassign a certain percentage of the synthetic data in a process called "jumbling".

POLARIZED INDIUM-115

- Ultimately, the goal is to know the correct spin assignments of the resonances. There are experimental methods that allow the researcher to know the spin assignments in advance, such as polarization.
- However, polarizing compound nuclei requires expensive magnets and is not usually worth the cost. But we do have some polarized In-115 data, so we can validate the training accuracy on what we accept as accurate resonance assignments.

CLASSIFIERS & HYPERPARAMETERS

- Scikit-learn offers many ML classifiers
- Focus on **Neural Net** and **Random Forest** classifiers
- These classifiers have hyperparameters that can be adjusted to fit our data.
- Fig. 2 is an example of regularization, a hyperparameter that helps avoid overfitting for weights with large magnitudes.
- In this project, we test several combinations of Random Forest hyperparameters and observe the validation/training accuracy correlation.

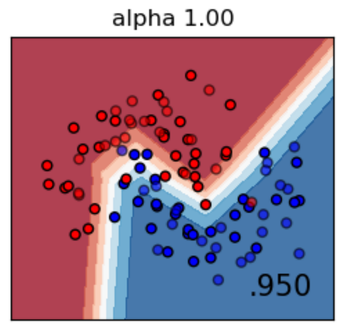


Figure 2: Visual of regularization **Source:** Scikit-learn 1.17.4. Regularization **Credit:** Scikit-learn

VALIDATING TRAINING ACCURACY

- We are most interested in observing how a validation set performs on a trained algorithm. We want to see if the validation data performs consistently better or worse at multiple training jumbleness values.

JUMBLING THE VALIDATION DATA

- As is generally the case, polarized validation data will not be available, so we jumble the validation data and observe the performance of the algorithm.

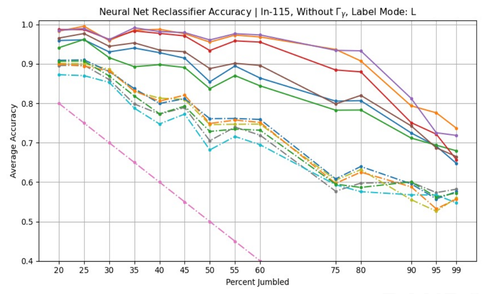
ITERATIVE RECLASSIFICATION

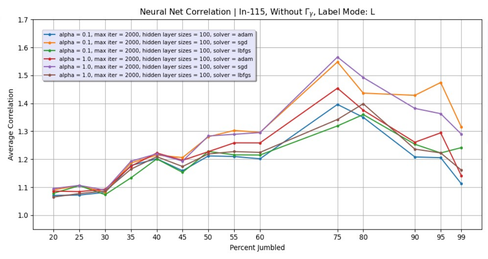
- We start with jumbled validation data and use successive reclassifications in attempt to incrementally improve the validation accuracy.

Results

COMPARING HYPERPARAMETERS

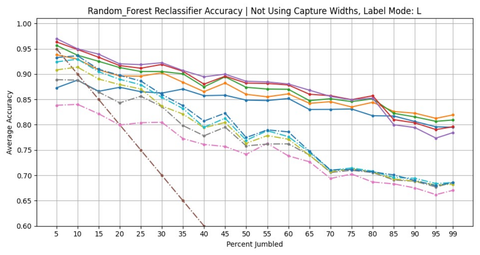
NEURAL NET

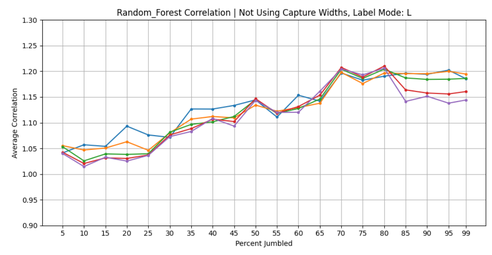




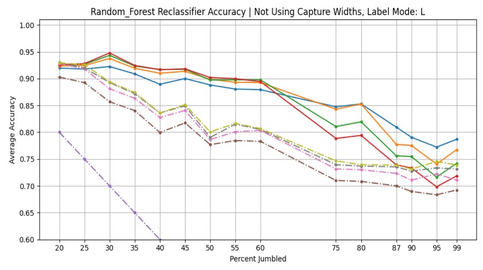
- alpha = 0.1, solver = adam
- alpha = 0.1, solver = sgd
- alpha = 0.1, solver = lbfgs
- alpha = 1.0, solver = adam
- alpha = 1.0, solver = sgd
- alpha = 1.0, solver = sgd
- No Classification

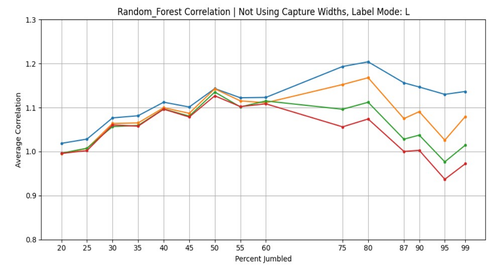
RANDOM FOREST





- max depth = 1
- max depth = 2
- max depth = 3
- max depth = 4
- max depth = 5
- No Classification



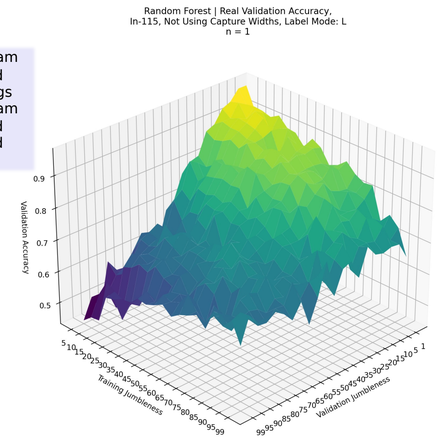


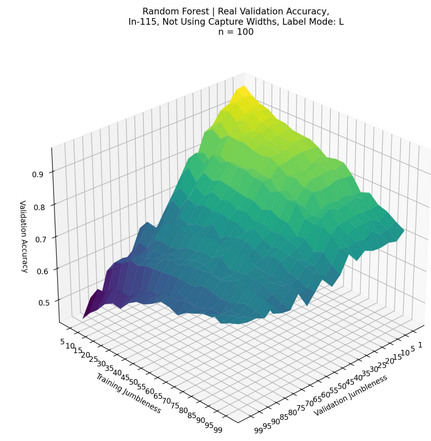
- max features = 1
- max features = 2
- max features = 3
- max features = 4
- No Classification

Plots of Average Accuracy (left) & Correlation (right) vs % Jumbleness.

- In these plots, the training accuracy is plotted as a dashed line and the validation (solid line) data comes from unjumbled In-115 data. Correlation is calculated as $correlation = \frac{validation\ accuracy}{training\ accuracy}$. Here we have tested various hyperparameters.
- The solver is a function that attempts to optimize efficiency of fitting the data. Sgd (stochastic gradient descent) generally performed the best. Max depth is the maximum depth of the Random Forest tree, determines the number of times a new node is created. Max features specifies the number of features to be used when looking for the best decision tree split.
- We see that in most cases, the correlation degrades at high training jumbleness values.

JUMBLED VALIDATION DATA





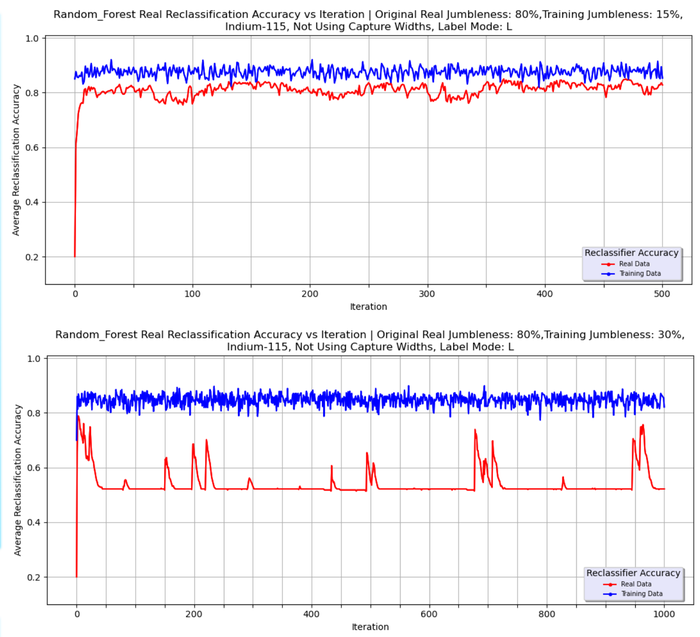
Change in Validation Accuracy as a function of Training & Validation jumbleness.

- Generally, a validation jumbleness around 50% will perform about the same for any training jumbleness.
- These plots also illustrate the importance of averaging cycles. Left: 1 cycle, right: 100 cycles.
- With these plots we can examine optimal paths of iterative reclassification.

ITERATIVE RECLASSIFICATION

Plots of successive reclassifications for a single Training jumbleness.

- On the top plot, the original validation jumbleness is 80% with a training jumbleness of 15%. The data is then successively reclassified.
- From this successive reclassification, we observe a convergence around 85% accuracy.
- On the bottom plot, the original validation jumbleness is again 80%, but with a training jumbleness of 30%.
- There is a clear convergence around 53%, but there are also sudden spikes in accuracy.
- Currently, we can only speculate as to the cause of the reassignment spikes, but we think it may result from the fact that we currently cannot average reclassified data when iterating. If we could build a reclassified sequence that considers information from many classification cycles, the spikes and strong fluctuations may disappear.



Future Work

- **Successive iterations** has proved interesting for this project.
- We would like to design a system that averages the reassignment statuses and creates a data set reflecting that average for the next iteration.
- We could potentially also design a system that rewards the algorithm each time a reassignment peak occurs in the successive iterations.

ACKNOWLEDGEMENTS

I would like to sincerely thank my mentor, Dr. Gustavo Nobre, for his insight and guidance throughout the summer SULI program. I would also like to thank my fellow intern, Cole Fritsch, all past interns, and Dr. David Brown for their combined efforts and contributions to this project. This project was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

REFERENCES

- NOBRE, GUSTAVO, Nobre, G, Brown, D, Scoville, S, Fucci, M, Ruiz, S, Crawford, R, Coles, A, and Vorabbi, M. *Expansion of Machine-Learning Method for Classifying Neutron Resonances*. United States: N. p., 2021. Web. doi:10.2172/1823638.
- Brown, David, Nobre, G. P. A., Hollick, S. J., Rodriguez, P., and Scoville, s. *Machine learning applied to classifying neutron resonances*. United States: N. p., 2020. Web. doi:10.2172/1673302.
- Pedregosa *et al.*, *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830, 2011.

www.bnl.gov