Frequency Decomposition of NMR Spectra

An educational application of deep learning principles

Sebastian W. Atalla | APS DSECOP Fellow

THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

01

MOTIVATION

Machine Learning in Physics Education

Machine Learning and Physics

- Physics & Astronomy is a rapidly evolving discipline requiring knowledge of multiple domains.
  - Mathematics, chemistry, biology, computer science
- Machine learning (ML) research is heavily pursued in every physics-related field.
- Physics education must adapt to include ML education to prepare physics students for future academic and professional appointments.

Machine Learning and Physics

- ML is utilized in all corners of physics research due to its generalizability.
  - Solving differential equations, protein folding, image analysis, simulations
- Many physics curricula include programming electives
  - MATLAB, Mathematica, Python, R
- Programming curriculum teaches students to solve computational problems
- ML is the next evolution of solving problems involving physical systems.

02

IMPLEMENTATION

Implementing Educational ML Modules

Modules in the Classroom

- DSECOP's goal is to create ML modules to be taught against existing curriculum.
- Modules augment material taught by instructor.
- May be taught over a period of 1 or more class periods.
  - Includes self-study and examination material
- Should be exciting and relevant, with enough information to inspire independent learning.
  - Not a barrage of information

Deep Learning Applications in NMR Spectroscopy

- First module in a series meant to introduce physics students to signal processing techniques via ML.
- Focus is not just on implementation, but also understanding the problems being solved by ML.

Module Learning Goals

- Module teaches users to differentiate between good and bad applications of ML, and how to reimagine problems to meet criteria for good applications.
- Users should consider, “What problem am I solving?”

Bad

- Computing DFT of signal acquired from spectrometer using a neural network.
- Still need to integrate the peaks!

Good

- Using network to learn best-fit parameters for peak at resonant frequency in spectrum of signal acquired from spectrometer.

Module Curriculum

- Brief overview of Fourier transform and NMR spectroscopy
- Overview of neural network architecture
- Estimating the DFT of an acquired signal using a neural network
  - Homework questions
- Brief overview of NMR line-shape functions and curve-fitting
- Estimating curve-fitting parameters of resonance peaks in an FID's spectrum
  - Homework questions

Example Instructional Material

Activation Functions

Activation functions are some of the most important components of a network. These functions may be linear or nonlinear; the nonlinear functions are what allow the network to learn complex relationships within the training data. Activation functions are also responsible for constraining the data within physically meaningful bounds. As a simple example, classification networks tend to be concerned with percentages describing the confidence at which a network can classify an input as belonging to a particular group. This means that the activations $h_n^{(k)}$ should ideally be constrained on the interval $[0, 1]$. For a regression problem, the boundaries could be defined on any interval meaningful to the data. Thus, different activation functions are better suited for different problems, and using the wrong activation function can yield poor or meaningless output from the network. With the inclusion of the activation function, the node's activation may now be expressed as

$$h_n^{(k+1)} = f\left(b + \sum_{n=1}^{N} h_n^{(k)} w_{n,k}\right)$$

where the activation $h_n^{(k)}$ is the argument of the activation function f.

- Module covers layer architecture, activation functions, regularization, learning rate, and optimizers with mathematical rigor.

Example Exercise

Exercise 2

Unlike the linear transformation from an FID to its frequency spectrum in Exercise 1, there exist much more complex relationships between the input and output data in the following ANN due to the addition of a curve-fitting task which the network must learn.

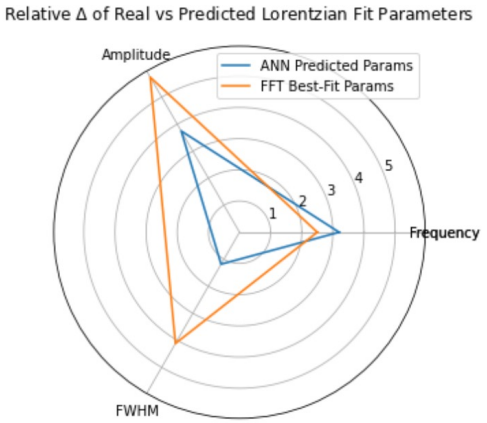
Activation functions are especially important for elucidating these complex relationships. Without them, the model behaves as if it only has a single layer and important weights cannot be updated as quickly. In short, the activation functions introduce non-linearity.

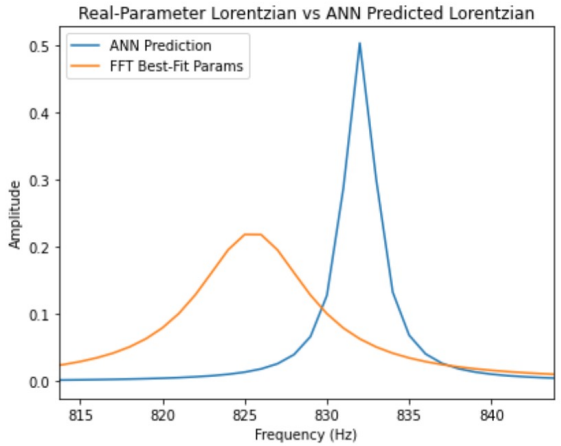
Exercise 2.1 Revisit the activation functions introduced in Exercise 1. Tune the `activation` hyperparameter using some of the more advanced functions available in the Keras Activation Function API documentation. Note how these functions alter the model's ability to converge upon a solution.

Aside from the activation function, all of the same hyperparameters from the network in Exercise 1 exist in this network, e.g. the number of layers and the number of nodes within each layer, the learning rate, momentum, etc.

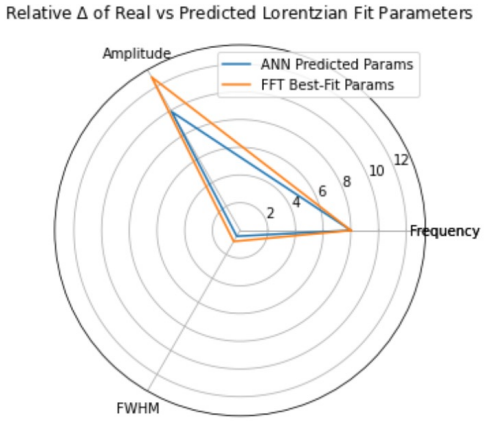
Exercise 2.2 This model is overfitting. Revisit the discussion on **Regularization** in Exercise 1, and tune the hyperparameters to seek some improvement. Change the `monitor` option in the early stopping callback to monitor `val_loss` instead of `loss`. In addition to the suggestions in Exercise 1, the model may be overly complex or more samples may be needed for training. Again, these are all tunable hyperparameters that may need adjustment over the course of designing the network. Check the tuning results against the figure at the end of this section - run it several times to compare against several randomly selected FIDs. If the results are unsatisfactory, tune the parameters again and continue to attempt to minimize `val_loss`.

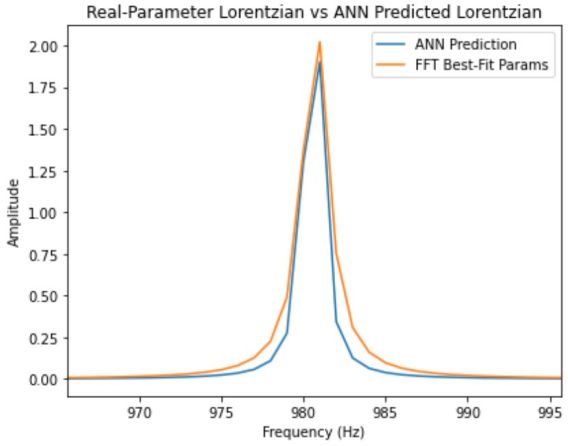
Example Exercise





- Module includes graphical output for qualitative inspection of network output after hyperparameter tuning.





Module Curriculum

- Module is applicable to a variety of courses.
  - Numerical Techniques — UNC PHYS 331
  - Introduction to Medical Physics — UNC PHYS 461
  - Imaging Science — UNC PHYS 585
  - Intro. to Biomedical Imaging Science — UNC PHYS 586

Module Curriculum

- Module is not intended to replace formal ML instruction.
  - Motivates users to seek out further education in ML
  - Requires proficiency in elementary statistics and calculus
- Implemented in Python as a Jupyter notebook.
  - Executable on cloud computing platforms such as Google's Colaboratory
- Covers basic TensorFlow/Keras API usage.
  - Layer architecture, activation functions, regularization, learning rate, optimizers

03

FUTURE

Module Roadmap

FUTURE

Roadmap

Module 1

- Basic introduction to ML
- Basic feedforward neural networks
- 1D NMR spectroscopy

Module 2

- Convolutional neural networks
- More detailed examination of hyperparameter tuning
- 2D FFT/image reconstruction from *k*-space data

04

CONCLUSION

Final Remarks

CONCLUSION

Final Remarks

- This project is sponsored by a fellowship from the American Physical Society's Data Science Education Community of Practice.
  - APS DSECOP website
- Improvements to these modules are driven by feedback.
  - bit.ly/DSECOP-feedback

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL