Supplementing BNS Simulations with Artificial Neural Networks

Peter Hammond

NP3M Postdoctoral Fellow
Pennsylvania State University
Institute for Gravitation and the Cosmos

pph5189@psu.edu

13 September, 2023

Overview

- What are ANNs?
- How can they be useful for us?
- What does an ANN implementation look like?
- How does an ANN perform?

What is an Artificial Neural Network?

- Artificial Neural Networks (ANNs) are a collection of nodes connected by directed edges, arranged into layers.
- Each edge uses an activation function to transform the value at it's root to the value at it's head.
- Each node (other than those in the input layer) calculates a weighted sum of it's input edges and passes the result to it's output edges.

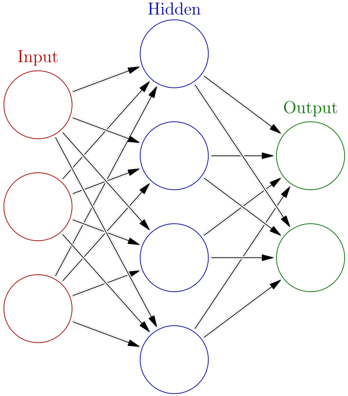

Figure 1: Schematic of an ANN with one hidden layer.

What is an Artificial Neural Network?

- Common activation functions are the ReLU, sigmoid, and tanh.
- The network can be trained by calculating an error on some training set, then using the derivative of that error with respect to the parameters in the network to calculate a new set of parameters.

Figure 2: ReLU, sigmoid, and tanh activation functions.

How can we use ANNs?

- A Multi-Layer Perceptron (MLP) (feed-forward, fully-connected ANN with non-linear activation functions) can be used as a universal function approximator.
- We can therefore use an MLP to approximate functions that are either too expensive to calculate "on-the-fly", or which cannot be expressed algebraically.
- I will use as an example the problem of determining the equilibrium conditions in neutrino-opaque matter within a BNS simulation.

How can we use ANNs?

- Example problem:
  - We want to calculate the equilibrium conditions that some NS fluid + neutrinos will reach.
  - The simulation can give us the rest mass, total-energy, and lepton number densities:
    - $\rho = \rho_{\mathrm{fluid}}$,
    - $e_{\mathrm{total}} = e_{\mathrm{fluid}} + e_{\mathrm{radiation}}$,
    - $Y_{\mathrm{l}} = Y_{\mathrm{q}} + \sum_x \left(n_{\nu_x} - n_{\overline{\nu}_x}\right)/n_{\mathrm{b}}$.
  - We need to calculate the temperature and species fractions.
  - For simplicity we will consider only the electron-ic lepton species.

How can we use ANNs?

- If we know $(n_{\mathrm{b}}, T, Y_{\mathrm{q}} = Y_{\mathrm{e}})$ then we can calculate $(\rho, e_{\mathrm{total}}, Y_{\mathrm{l}} = Y_{\mathrm{l,e}})$ from the EoS and some simple functions, but the inverse is not possible.
- A typical approach might be to use Newton-Raphson to solve for the temperature and species fractions, however this can be unstable and/or slow, especially where there is not a good initial guess.
- We could use an ANN to:
  1. obtain a solution directly,
  2. obtain a good initial guess to stabilise the NR.

What does an ANN implementation look like?

- Basic structure:
  - Inputs will be the rest mass density, total energy density, and lepton electron number.
  - Outputs will be the temperature and electron fraction.
- ANNs work best when inputs and outputs are in the range $-1 \leq x \leq 1$, so we apply the following transformations:
  - $I_1 = \log(\rho)$
  - $I_2 = \log(e_{\mathrm{tot}}/\rho)$
  - $I_3 = Y_{\mathrm{l,e}}$
  - $O_1 = \log(T)$
  - $O_2 = Y_{\mathrm{e}}$

  and re-scale each of these to the above interval using the entire EoS table as input.

What does an ANN implementation look like?

- Other considerations (hyperparameters):
  - number of hidden layers,
  - size of hidden layers,
  - activation functions,
  - optimisation method,
  - learning rate,
  - etc..
- For this example we use 3 hidden layers with 100 neurons each, the LeakyRLU activation function, and a limiter on the inputs and outputs to stop the values going beyond those in the table.

How does an ANN perform?

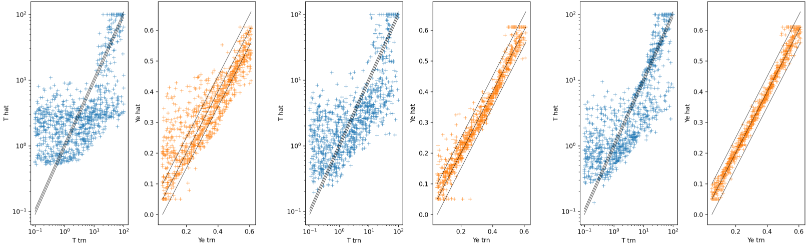

Figure 3: Output of ANN after 30 seconds, 5 minutes, and 1 hour of training on a consumer grade GPU. The error in Y_{e} is good enough for an initial guess, but not so much for $T \leq 10$ MeV.

How does an ANN perform?

- All is not lost!
- By using the calculated Y_e we can reduce the dimensionality of the problem to 1D.
- We can then use a stable method, e.g. bisection, to get our T guess.
- Because we have two equations and only one unknown we are free to pick for which function to find the root (or minimise a combination of both).
- Here for simplicity I will use $f(T) = Y_{\mathrm{l,e}}(T) - Y_{\mathrm{l,e}} = 0$

How does an ANN perform?

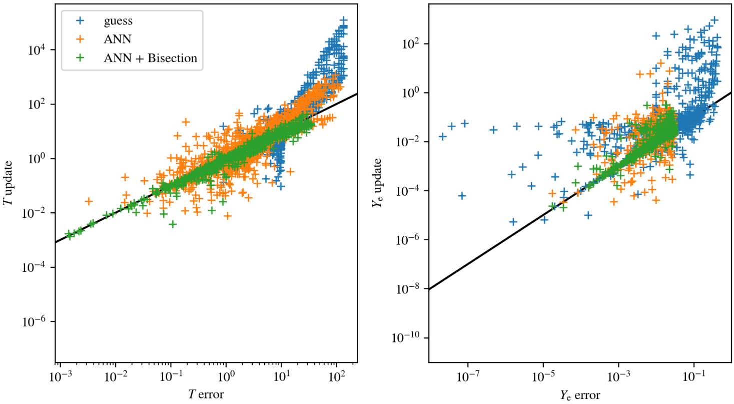

Figure 4: Update of first NR iteration against error.

Summary

- ANNs can be used to find approximate solutions to calculations that are too expensive to do on the fly, or which cannot be expressed algebraically.
- We have used an MLP to obtain better initial guesses for an unstable method like Newton-Raphson.
- Other applicable problems include Con2Prim, EoSs, ν-oscillation, etc..
- Last but not least, they are fast!