Thesis:

In the unfolding section:

(…We’re trying to find the inverse function.) In the mathematical sense of the problem, an inverse does not exist. This is because, theoretically, multiple neutron spectra, say **φ0** , **φ1** and **φ2** , can still give the same set of reaction rates **Z0**; so there is no correct, unique choice of mapping of **Z0** back to **φ0** , **φ1** and **φ2**.

~~(However, if the correct response matrix is chosen, and there are less than M bases vector…)~~

Neural Network:

A feed-forward neural network predicts label(s) when given a set of features. This is equivalent to a function, where the features are equivalent to the inputs of the function; and the labels are equivalent to the outputs of the function.

For example, let’s say that we have a neural network that predicts the likelihood of rain in a city in the next hour from its temperature, humidity and cloud cover in the last hour. It has one label (i.e. probability of rain in that city, Prain, where 0≤P≤1 ) and, three features (temperature, cloud cover, humidity in the past hour). This is analogous to a scalar field (i.e. a scalar function with a 3-D vector input) which takes the value between 0 and 1.

This can be further generalized to multiple labels, e.g. likelihood of rain, likelihood of snow, and likelihood of sunshine in the next hour. Then this neural network will be equivalent to a vector field (i.e. a 3-D vector function with a 3-D vector input).

A neural network can predict the labels associated with a given set of features by approximating the function that generates these features.

Going back to the weather prediction example, assuming that there is a single, deterministic function that predicts the likelihood of rain (, snow and sunshine) in the next hour, then the neural network can approximate this function by having multiple layers. (Explain the layers, weights, and biases -> universal approximation theorem)

The parameters, namely the weights and biases, are tuned to their respective optimal values iteratively, at the training stage. These iterative steps are named epochs. At each epoch the gradient in $chi^2$ with respect to the parameters $(frac\{d \chi^2}{d **w**} \intertext{and} \frac{d \chi^2}{d **b**} )$ are calculated. Then each of these parameters takes a step “downhill”, by a step size $= a \frac{d \chi^2}{d **w**} \intertext{or} a \frac{d \chi^2}{d **b**} $, respectively, where a is the learning rate. After that it can be tested on data that it has never processed before, to evaluate its accuracy.

(Overtraining and undertraining)

(Unlikely to get stuck in local minima)

(But the biggest obstacle is plateaus, therefore people prefer to use ReLU to minimize the problem of diminishing gradients)

Applied to the unfolding problem

Equivalently, assuming that the inverse function which turns reaction rates back into neutron spectra EXIST, then …