PhysNet

Cause why not

Goal:

Given a spectrum, design a geometry to provide it.

Empirically test this as well.

We have a script in matlab that can generate the scattering/ absorption spectrum for a given geometric arrangment. The idea is to use this script to generate training data, then train a neural network with this data.

Thus the neural network will be able to produce the spectrum given a geometric arrangement, and also be able to further predict spectrums for objects it has not seen.

The major contribution of this work is that it allows us to find a way to match a spectrum. Aka we can take a desired spectrum, feed it into the output of the neural network, then find out the geometric arrangement suggested to create such a spectrum.

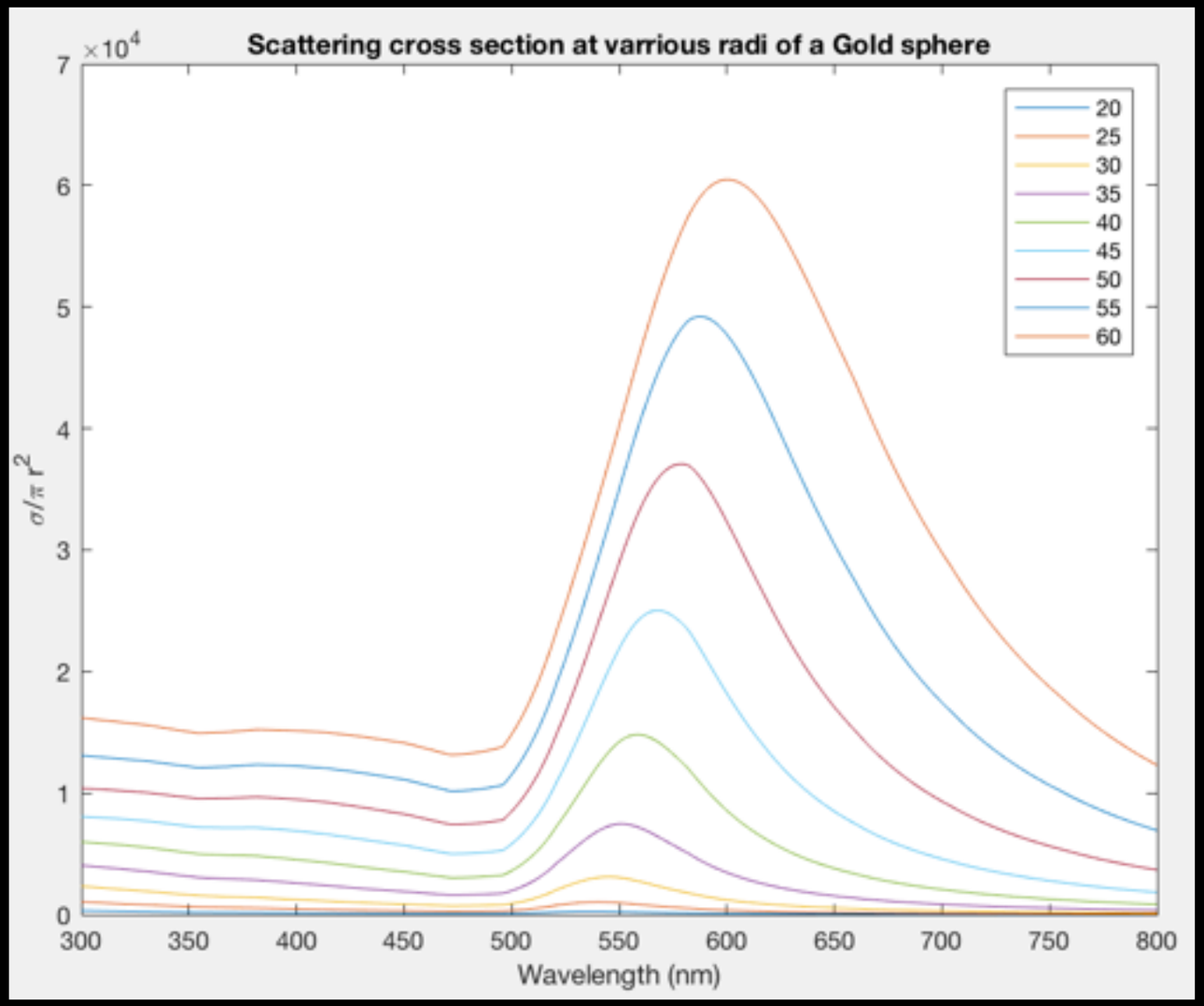
The data we generated was:
Silver/Gold spheres of radius:

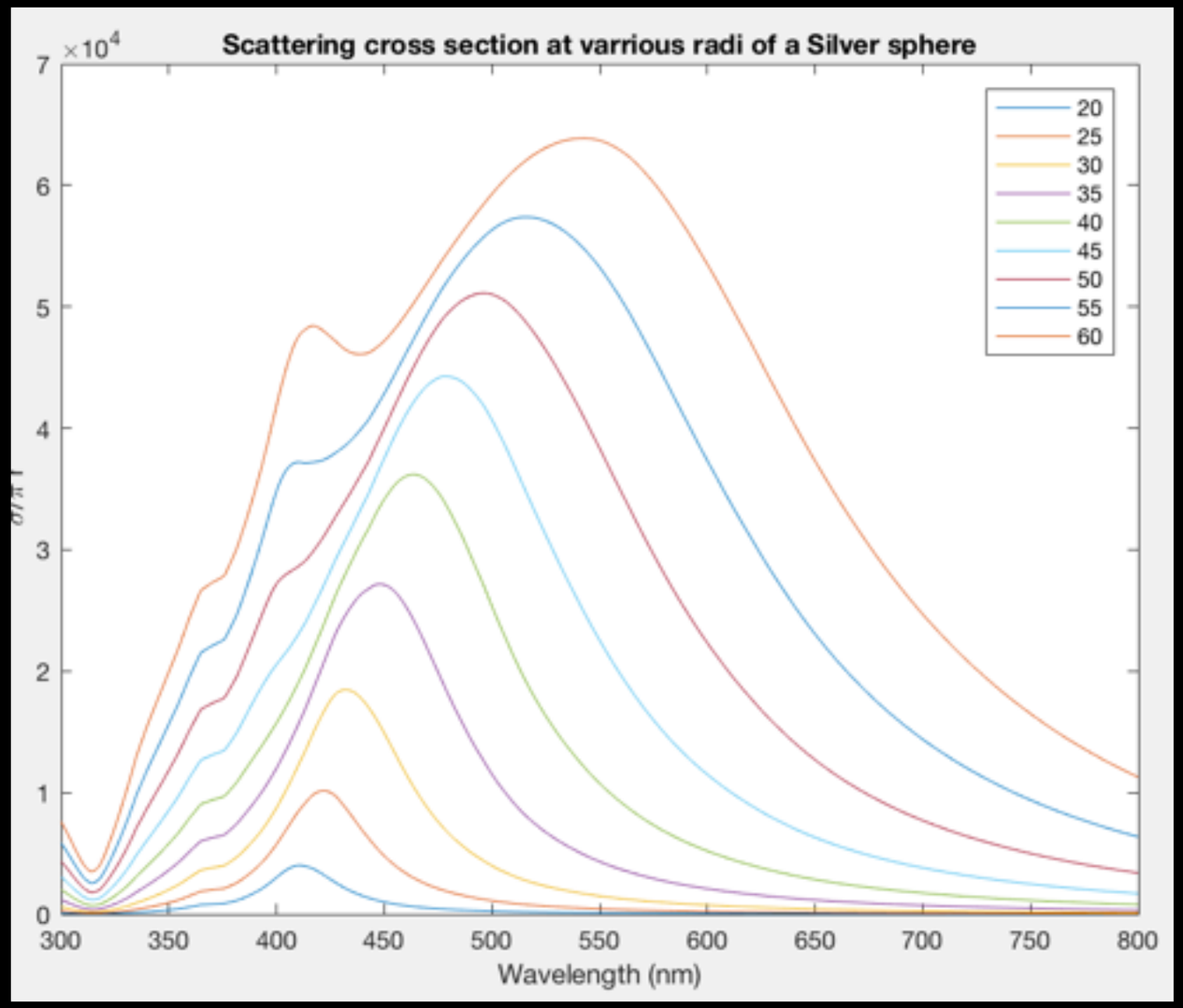
20:2:80

20:2:80

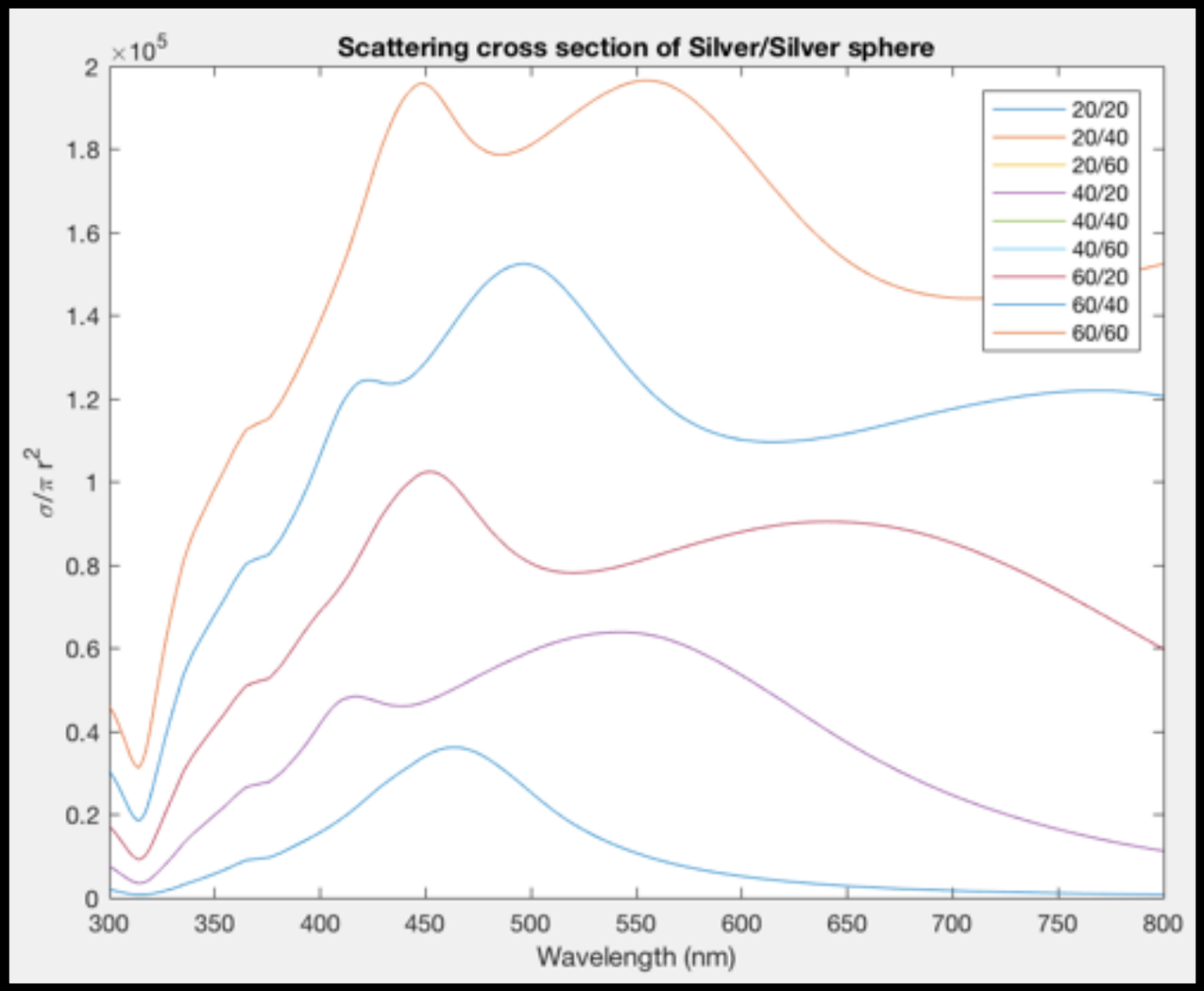
Gold/Silver

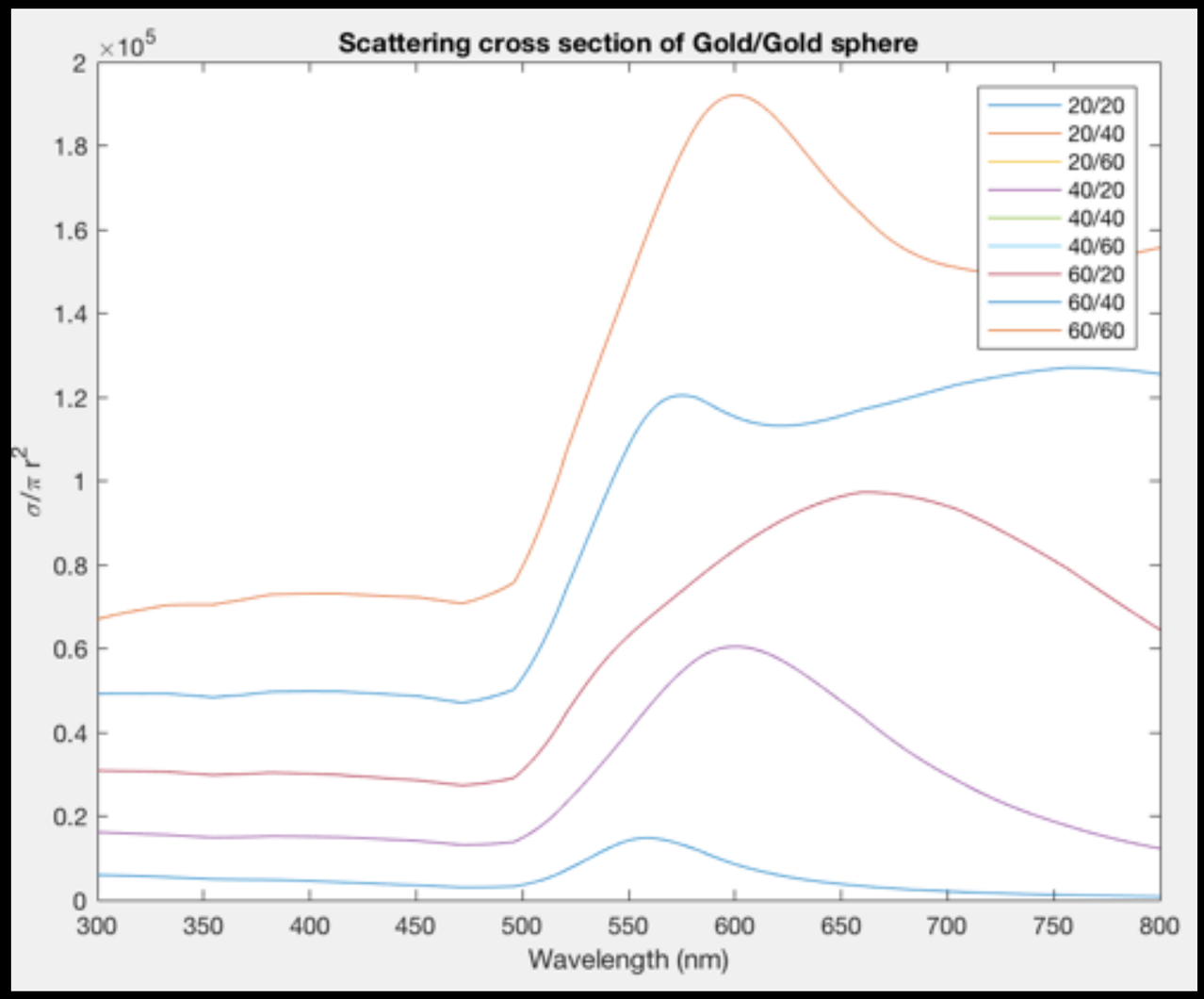
Training Data: Spectrums from Matlab





Training Data: Spectrums from Matlab

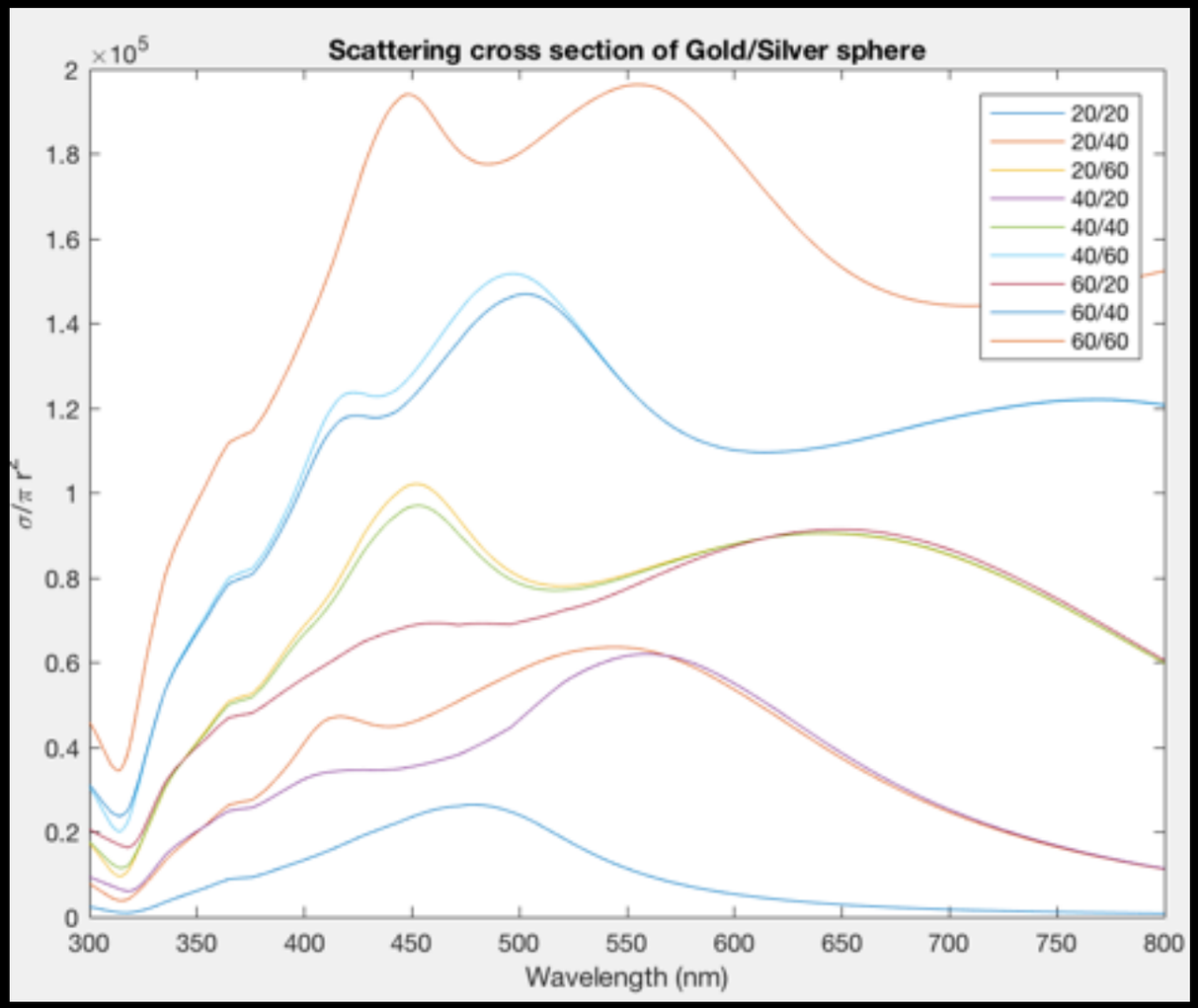


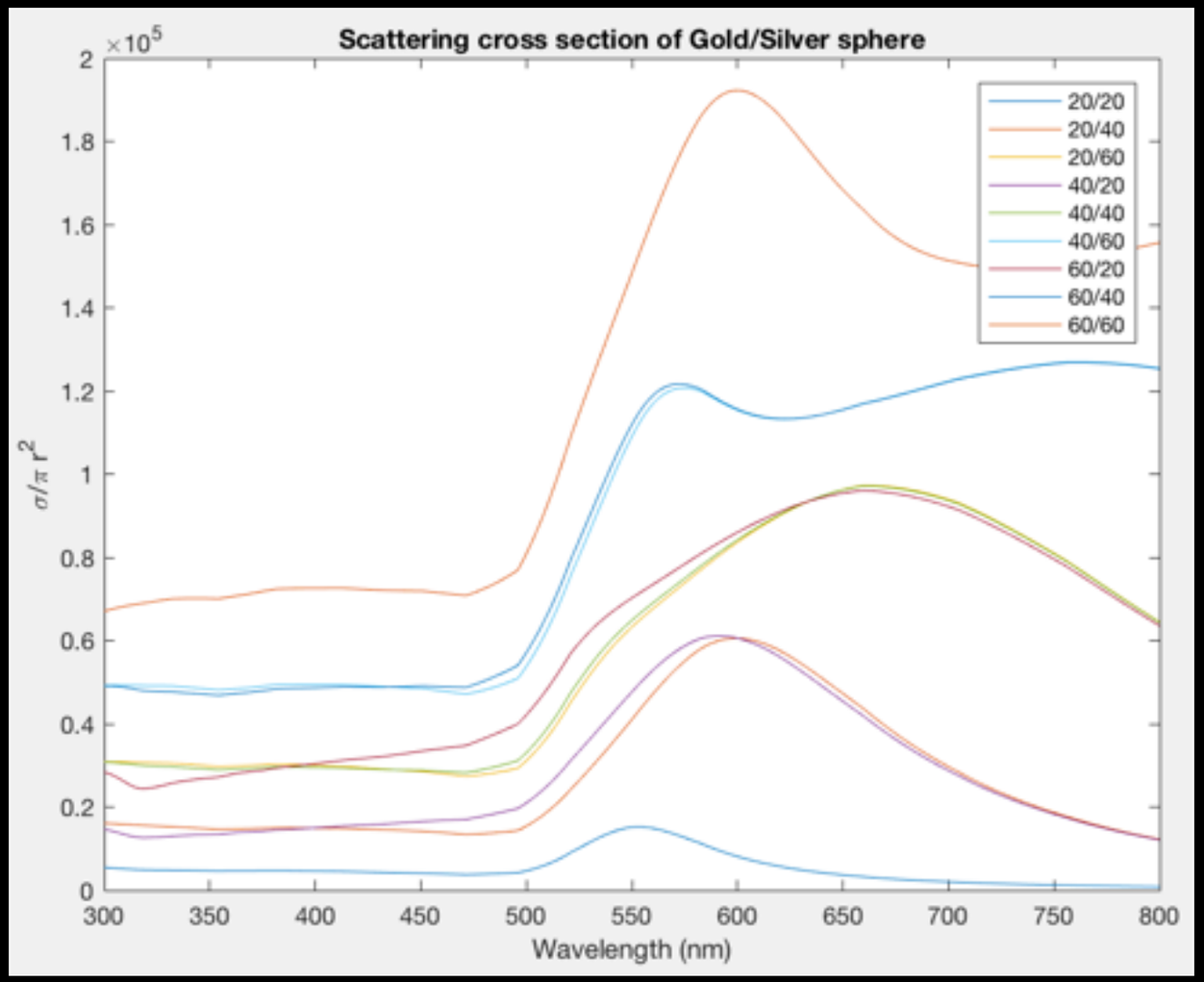


Note that these should match the cases for the pure silver/gold sphere.

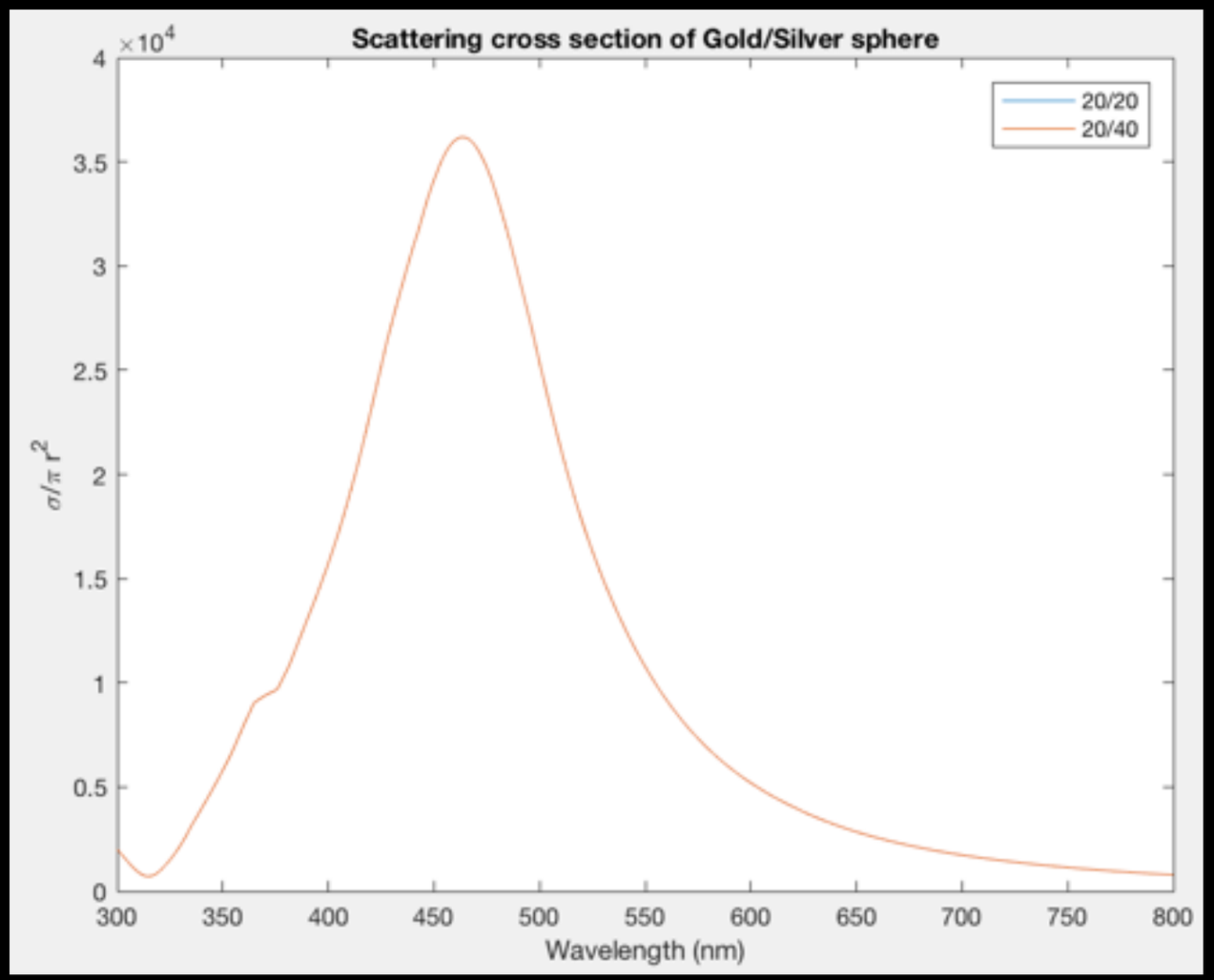
The differences are likely from the size differences.

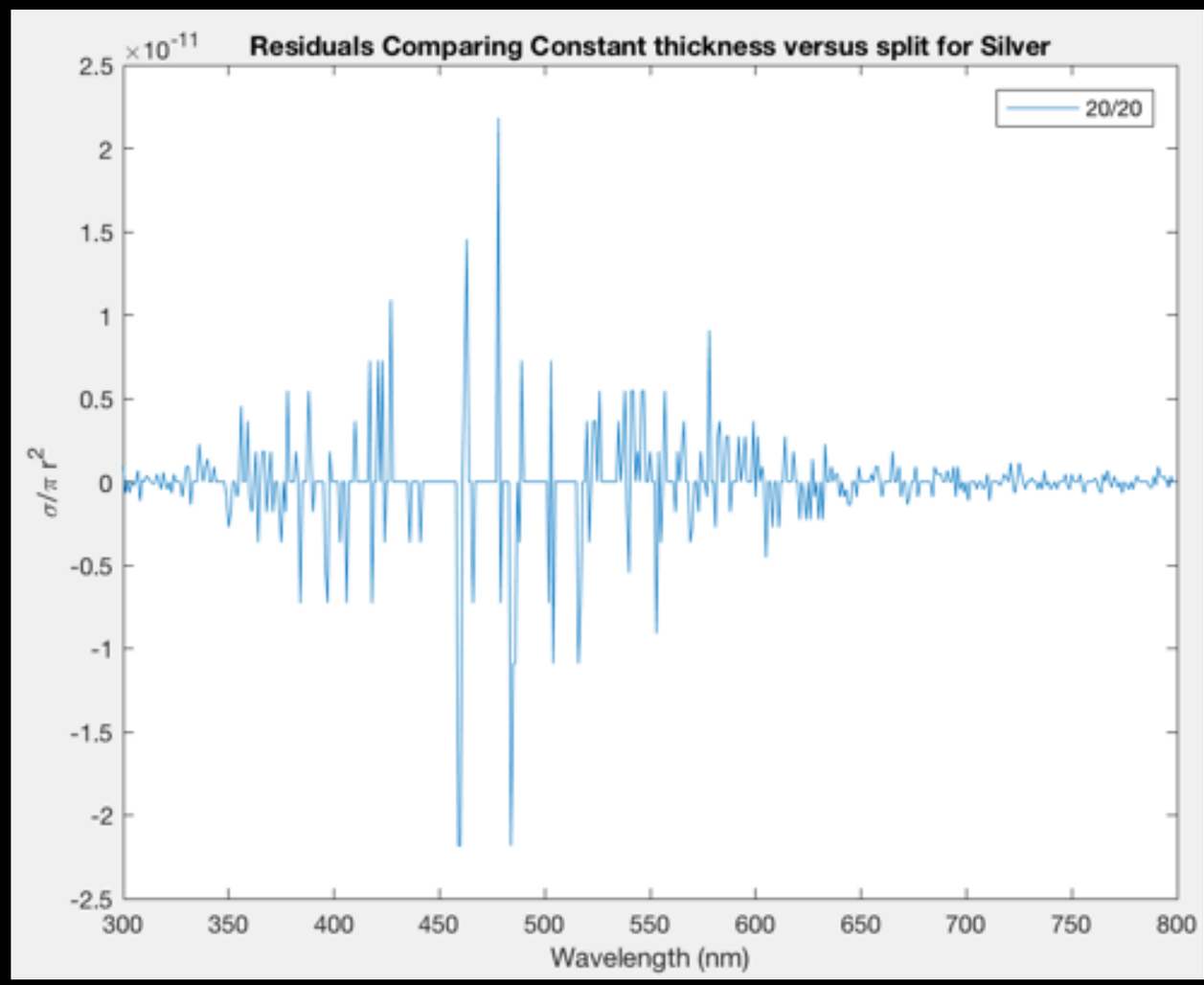
Training Data: Spectrums from Matlab





Testing Training Data





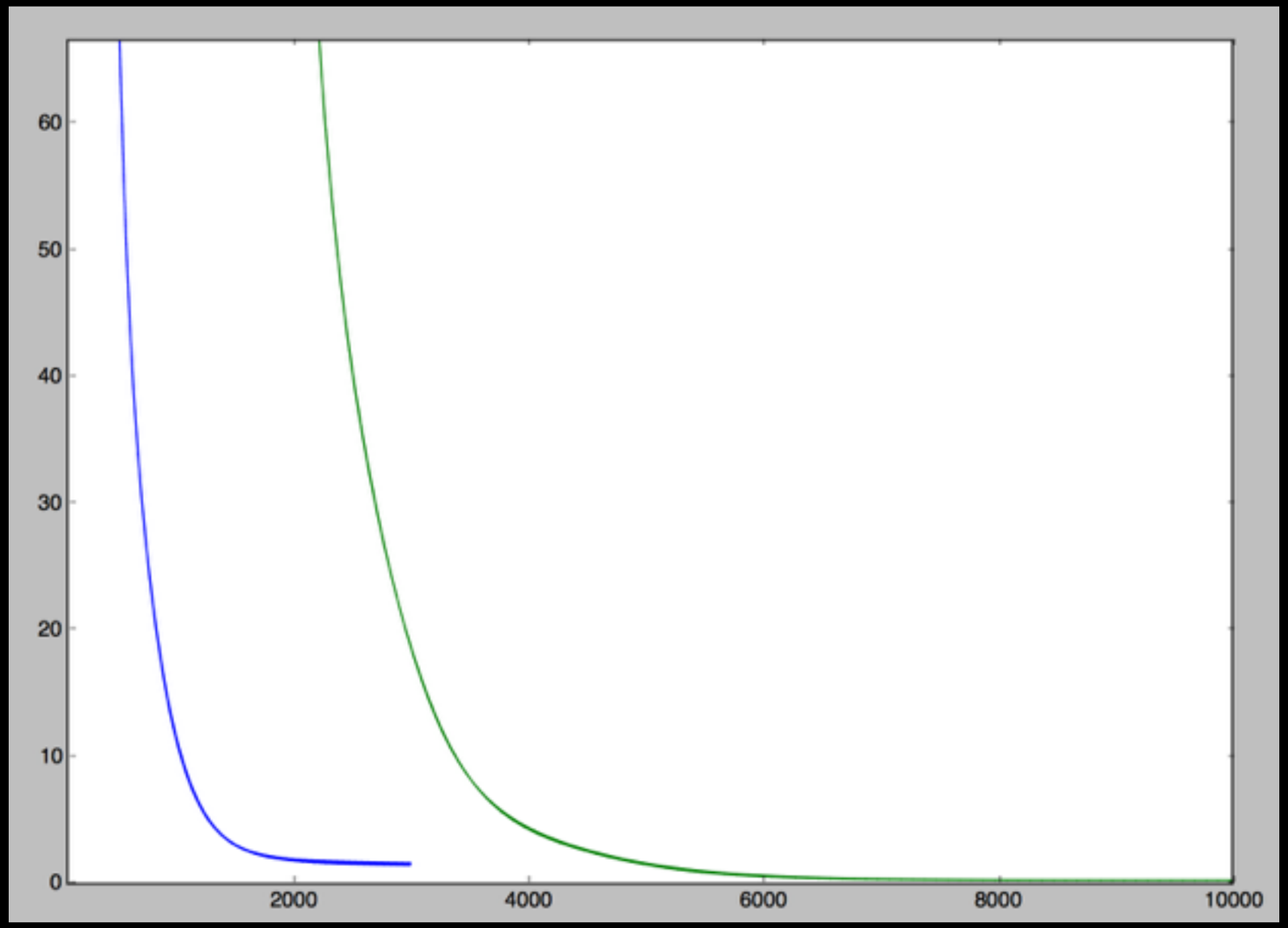
The residuals are very small (10^-11) compared to the spectrum (10^4).
The algorithm works well in matlab!

Now we generate the data and train a neural network.

Step 1: Single Sphere with variable radius

- Architecture:
    - 1 in, 20x20x20x20x20x100 out
    - 3620 parameters
- Training Data:
    - 1800 training points, 100 spectrum points per
        - Spectrum is 300-800nm, sampled every 5.
        - Training is 10-> 100nm, .05 increments
    - ~180,000 data points
- Training results:
    - ~21 loss per epoch
    - .01 error per point of the spectrum.
- Reversing:
    - Had to do it manually
    - 23.5275 -> The program finds: 23.5396

Very accurate

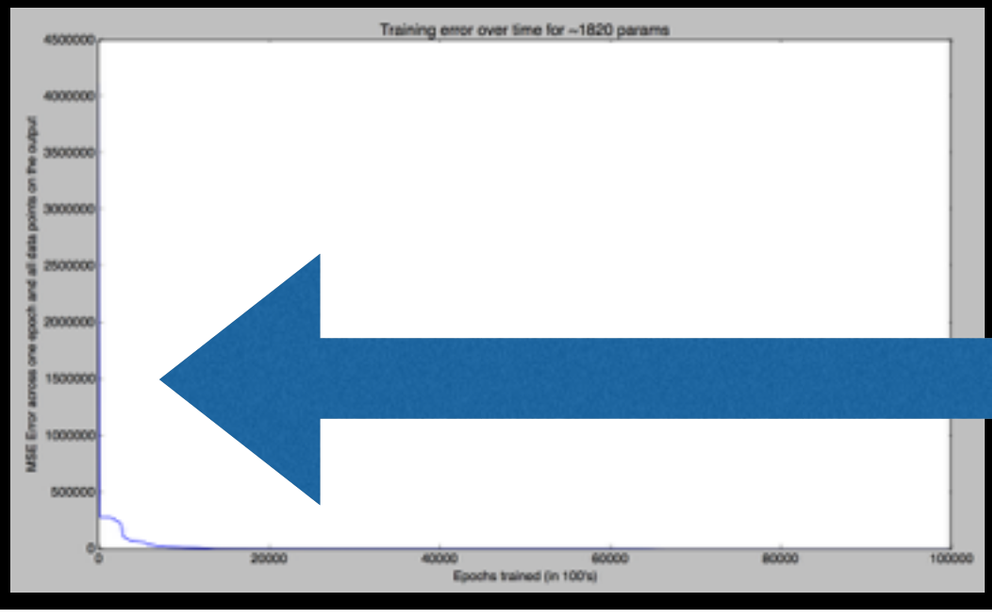


Training loss over iterations.
By around 6,00 iterations, it is approximately its final value

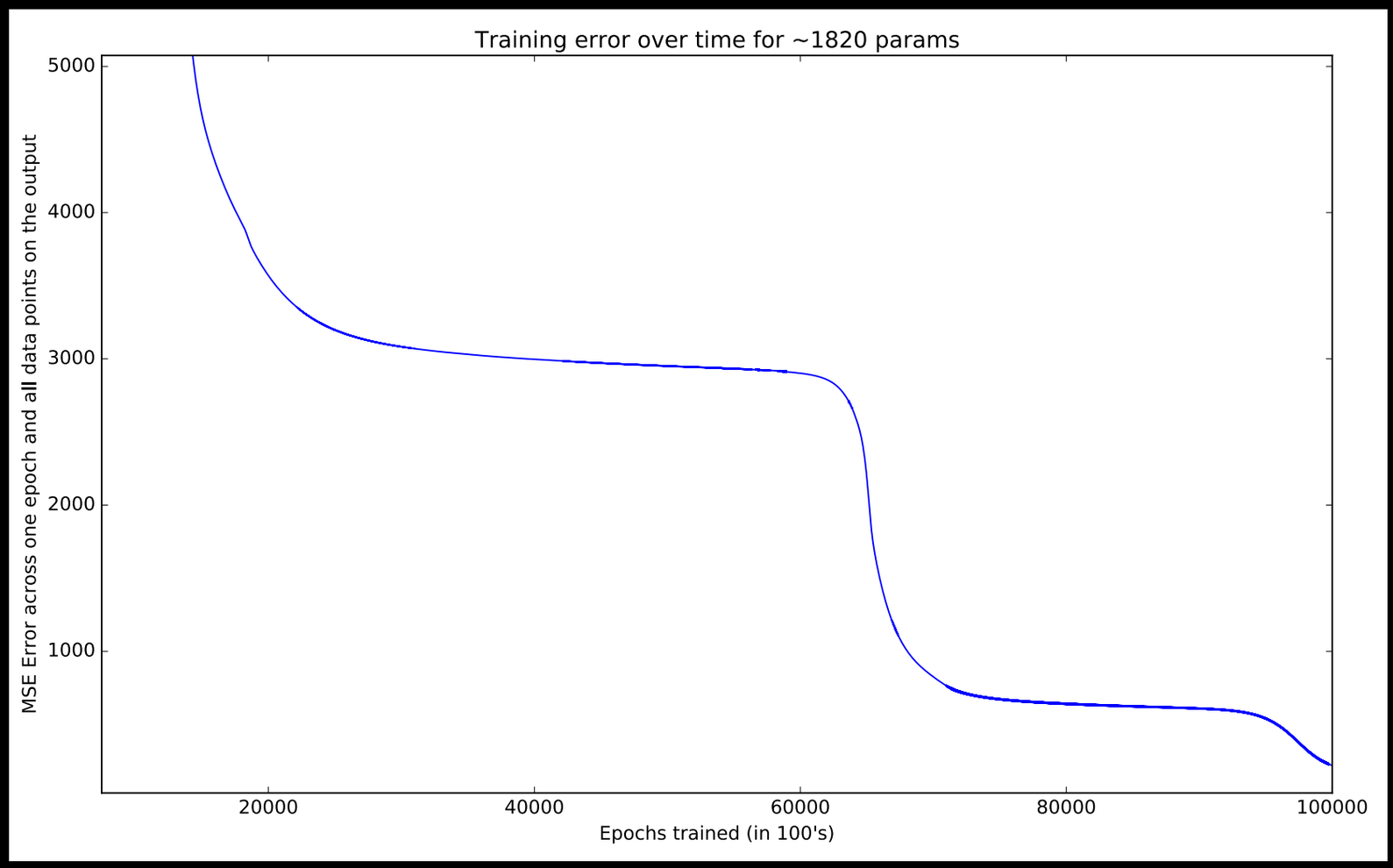
Step 2: Two-Layer Sphere with different materials (work-in-progress)

- Architecture:
    - 1 in, 20x20x20x20x20x100 out
    - 3620 parameters
- Training Data:
    - 3844 training points, 100 spectrum points per
        - Spectrum is 300-800nm, sampled every 5.
        - Training is 20:2:80, 20:2:80, gold/silver.
        - ~384,400
- Training results:
    - 48701 epochs, ~2956 loss per epoch.
    - 90101 epochs, ~200 loss per epoch.
    - .023 error per point of the spectrum.
- Reversing:
    - Had to do it manually
    - 23.5275 -> The program finds: 23.5396

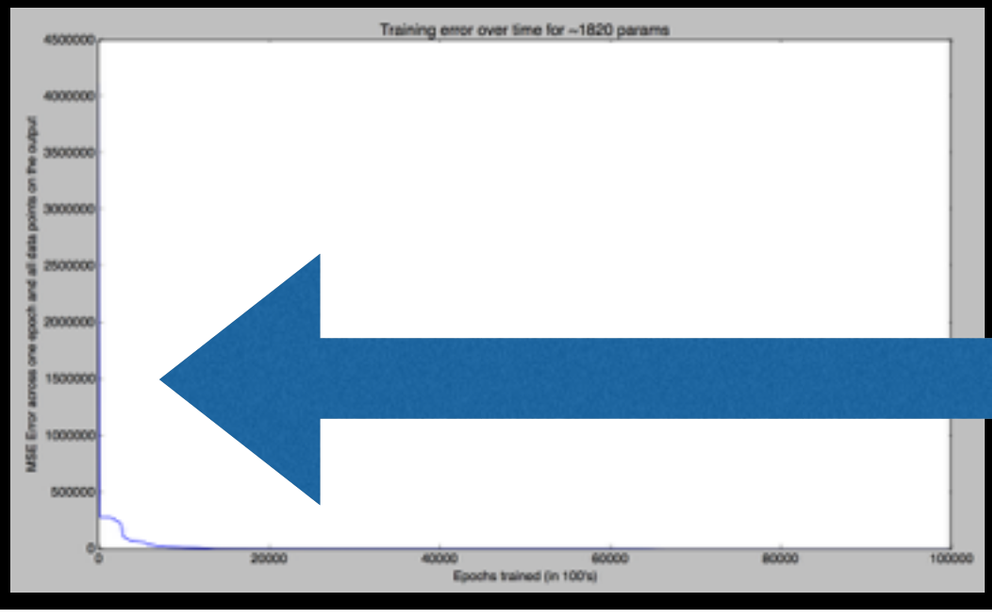
Very accurate



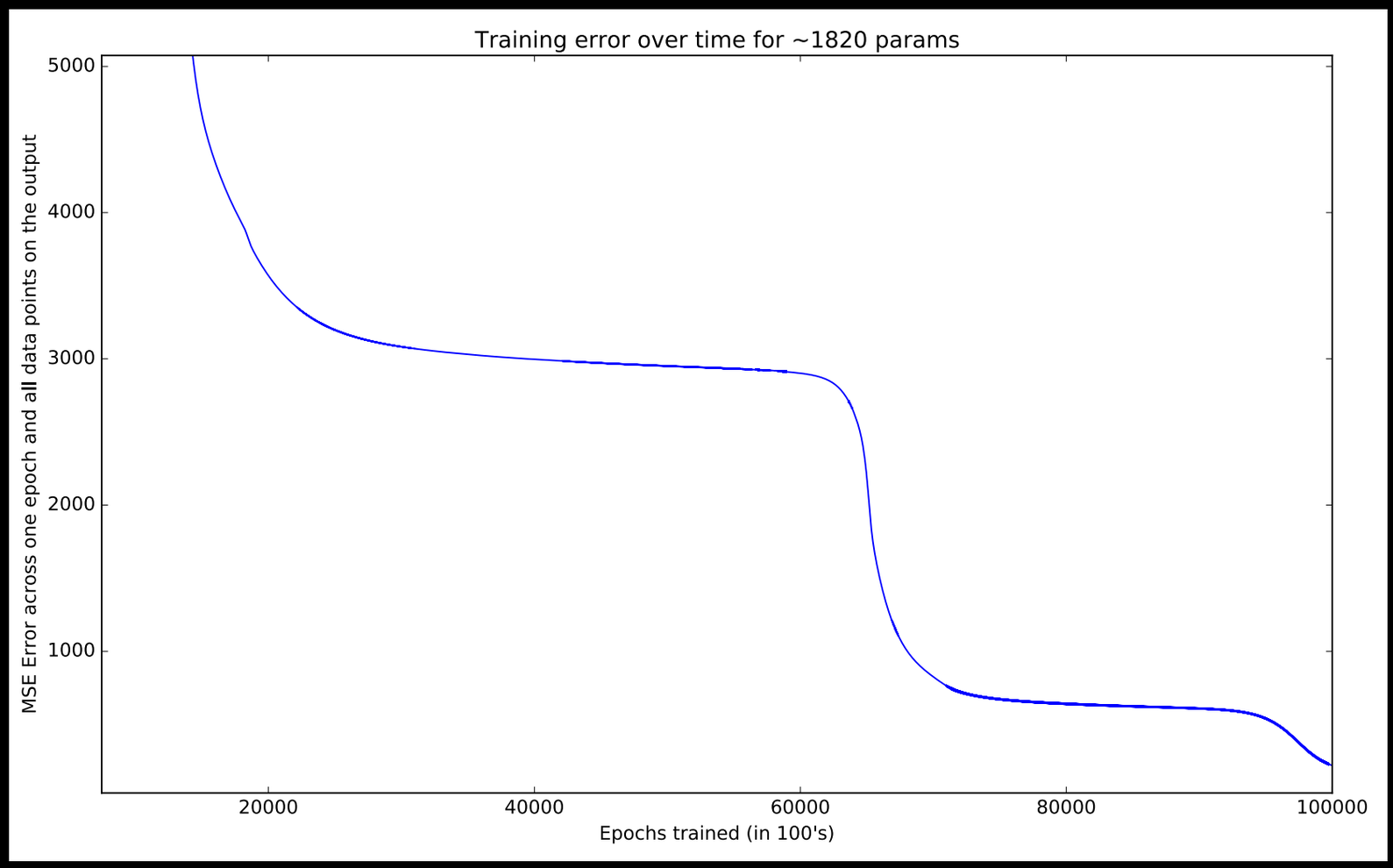
It trains that well



Training loss over iterations.
Still at 100,000 it has not converged.



It trains that well



Training loss over iterations.
Still at 100,000 it has not converged.