PROJECTING 3D POINTS ONTO A UNIT SPHERE USING NEURAL NETWORKS

Cameron Richards

Undergraduate BSc Victoria University of Wellington

Google Collab - https://colab.research.google.com/drive/18kt5n6Rmnh4wk4DxEa6pGDJQQbcGpcEi?usp=sharing

1. INTRODUCTION

Neural Networks have been a central development within the field of Artificial Intelligence as a Machine Learning technique, with relatively powerful capabilities in modelling both classification and regression relationships across a range of real-world domains. This project aims to utilize a conventional fully connected neural network to model the projection of points in 3-dimensional space onto the surface of a unit sphere.

2. THEORY

2.1. Task Definition

The aim of this project was to build a conventional fully connected neural network which would seek to model the relationship between a point in 3-Space and its corresponding cartesian coordinates when projected onto the surface of a unit sphere (a sphere about the origin, with a radius of 1), such that the polar and azimuthal angle are preserved.

2.2. Technology

This project, including the data generation, implementation of the neural network and graphing was completed using python, along with Matplotlib, PyTorch and Numpy libraries.

2.2. Data

The data used for the project was artificially generated as 500 samples from a trivariate standard, normal, gaussian distribution ($Z \sim N(0,1)$).

Similarly, validation and testing data comprised of 250 samples each from identical normal, standard gaussian distributions.

2.3. Model

The model used is a standard conventional fully connected neural network of four layers. The input and output layers are both three neurons wide each, as dictated by the 3-dimensionality of the attributes which describe a point's location in 3-space.

The model contains 2 hidden layers, both 16 neurons wide each. A ReLU (Rectified Linear Unit) activation function is used in between the first and second, and the second and third layers, whilst the final layer is linear. Biases are used alongside weights for all layers in the network.

The model used the Adam optimization algorithm applied in minibatches to update its weights and biases, alongside a Mean Squared Error Loss function; typical for a regression task such as this.

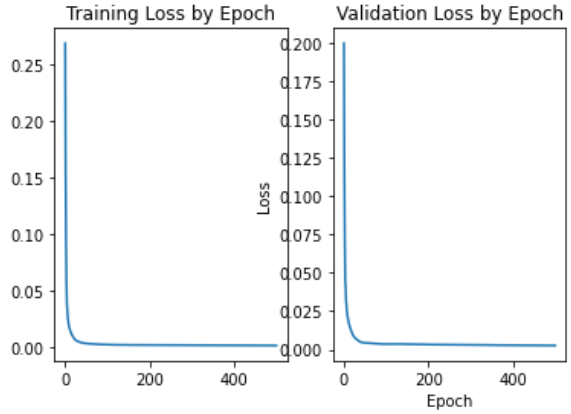
3. RESULTS

3.1. Parameterization

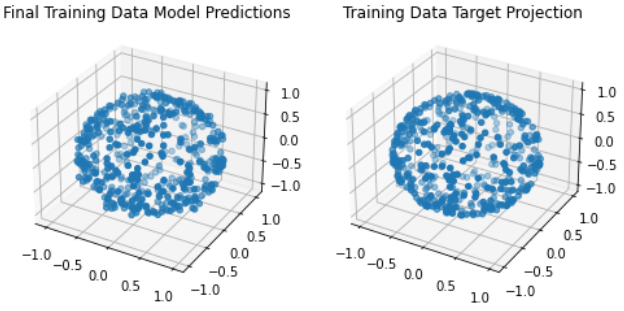
After experimentation, alongside the afore mentioned training data of 500 samples, a learning rate of 0.005 along with 10 minibatches, consisting of 50 samples each were found to be suitable parameters. The model was run for 500 epochs, at which point convergence had been achieved.

3.2. Training and Validation Results

After Training for 500 epochs the following graphs showing average loss by function could be produced for the training and validation data.

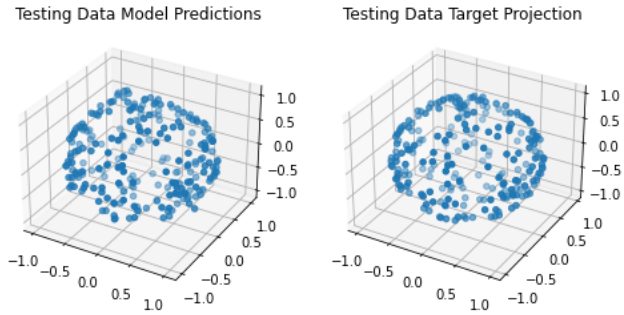


Gladly, the graphs show similar performance, indicating good generalization. After training was complete, the final average MSE of the training data was **0.00189(3sf**) and **0.00244(3sf**) for the validation. Below shows a comparison of the plotted target projection for training data and the final output predications by the model, which shoe robust and clearly recognizable performance in modelling the projection for the training data.



3.3. Testing Results

After training, the model predicated the projection locations for 250 identically distributed generated samples, which are plotted alongside their target projections below.



Visually, the model appears to perform very similarly well on unseen testing data which indicates high quality generalization of the model, for a robust set of examples. The associated average MSE for this training data was **0.00248(3sf).** This correlates with an average Euclidean distance of **0.0726(3sf)** from its target location; an approximate average error of 7% of is radius of 1 from the origin, indeed satisfactory performance.

4. CONCLUSIONS

To conclude, this paper explored the used of a neural network, to model the projection of standard, normal gaussian distributed datapoints onto the surface of a unit sphere. After the trained model found success which generalized to unseen testing data well, confirming that a neural network such as this is an effective process to model this relationship.