Ensemble Deep Learning: Neural Networks

In connection with the Research Experience for Undergraduates (REU) Program

Support for the program has been provided by the National Science Foundation (NSF) through REU Award Number DMS - 2050754.

Rhiannon Hicks – Program Correspondent
Dr. Mihhail Berezovski – Program Director
Embry-Riddle Aeronautical University

Dr. Jesse Adams and Dr. Margaret Lund – Senior Scientists
Nevada National Security Site

Abstract

Data science is a broad field that encompasses studying and extracting meaning from data, and a subset of this category is machine learning. Machine learning is the tactic of using various techniques to build computer models that are able to learn by themselves using data. A neural network is a type of machine learning that is modeled after how the human brain learns, and takes a given input and predicts an output from this value. Neural networks are extremely applicable in the field of data science, but also come with one large drawback: no method exists for quantifying the amount of error in a neural networks output prediction. This begs the question of what methods we can attempt to implement or create to effectively measure the uncertainty in neural networks so that we can better comprehend and dissect the meaning of their predictions. The Research Experience for Undergraduates program students will attempt to create an ensemble neural network with a valid uncertainty quantification method. Throughout this paper we will explore the complete method that the REU program has attempted in order to combat this problem.

Introduction

Embry-Riddle Aeronautical University's REU program is partnering with the Nevada National Security Site (NNSS) with funding from the National Science Foundation (NSF) in order to develop methods for quantifying and understanding uncertainty in the accuracy of neural network algorithms. The REU students will use publicly available datasets to develop a neural network, and will also create an ensemble with some uncertainty quantification (UQ) approach, such as Monte Carlo (MC) dropout, in order to better comprehend the results of the neural network and the uncertainty factor of their accuracy (Adams & Lund, 2022).

Background

The NNSS, formerly known as the Nevada Test Site (NTS), is a part of the United States Department of Energy (DOE). Their complex is larger than the state of Rhode Island and focuses mainly on Defense Nuclear Nonproliferation and Stewardship Science & Experimentation. Although there are many divisions within the NNSS, REU students will be working directly with the Weapons' Program, whose goal is to ensure the United States' nuclear stockpile remains safe, secure, and effective. In order to do this, they conduct classified nuclear experiments, high-tech computer modeling, and detailed engineering analysis according to their website, (*About the NNSS,* n.d.). At the U1a underground facility high energy tests are conducted so that radiographers can capture high resolution images of the explosions and interpret if things are working the way that they should be. This is where neural networks are widely used. There are thousands of images to sort through and interpret, and there is a lot of post processing to complete to learn anything from the images, so therefore neural networks are used to speed up the process and essentially extract the truth. But when using neural networks to determine if nuclear weapons are working as they should, a high amount of accuracy is needed since the safety of our nation depends on these tests. Without accurate error bars on these neural networks' predictions, detrimental explosions or mishaps could occur because of a slightly off uncertainty value. Therefore, it is necessary to research implementation methods for better quantifying uncertainty levels in neural networks.

Scope of the Problem

Neural networks are computer algorithms modeled after the neuron connections in a human brain and are used to detect relationships and patterns in large sets of data to better understand it. They consist of an input layer, multiple hidden layers, and an output layer. Given some input data a neural network will go through multiple different layers and pathways until it decides on the most likely output value. Another important aspect of neural networks is exactly how they learn to predict a certain output. A given data set will be split into training data and testing data, and a typical split is 70% training and 30% testing. The neural network learns while using the training data, and then attempts to predict the testing data as accurately as possible based on what it has learned. In essence, neural networks can be used as a type of prediction model, but they are also categorized as a 'black box' of information. In other terms, no prediction model has 100 percent accuracy all the time; there is always some error. A widely used example of how neural networks

can have high uncertainty is as follows. Say you have a neural network that takes in images and gives an output of whether they are a cat or a dog, but then you input an image of a zebra. The neural network will still predict this image as either a cat or a dog because it has not been trained to recognize zebras as an output, and therefore has a high uncertainty value and subsequent error bar for that specific image. Learning how to accurately quantify the level of uncertainty in neural networks by accounting for situations like these is what the REU students will be focusing on for the duration of their research.

Uncertainty Quantification (UQ) is the science of using various methods to quantify and reduce the amount of uncertainty in a data set. An ensemble neural network combines multiple similar but individual neural networks to decide on the most likely cumulative output based on each networks individual results. An ensemble is a type of UQ approach because it adds more checks to the neural network to ensure it is as certain as possible about its output value. REU students will attempt to implement an ensemble neural network for their selected dataset alongside another UQ approach to see how low the uncertainty factor in the accuracy can get without overtraining the network. Students will decide on their specific UQ approach individually, and combine individual efforts to develop the best neural network possible.

Data Visualization/Description

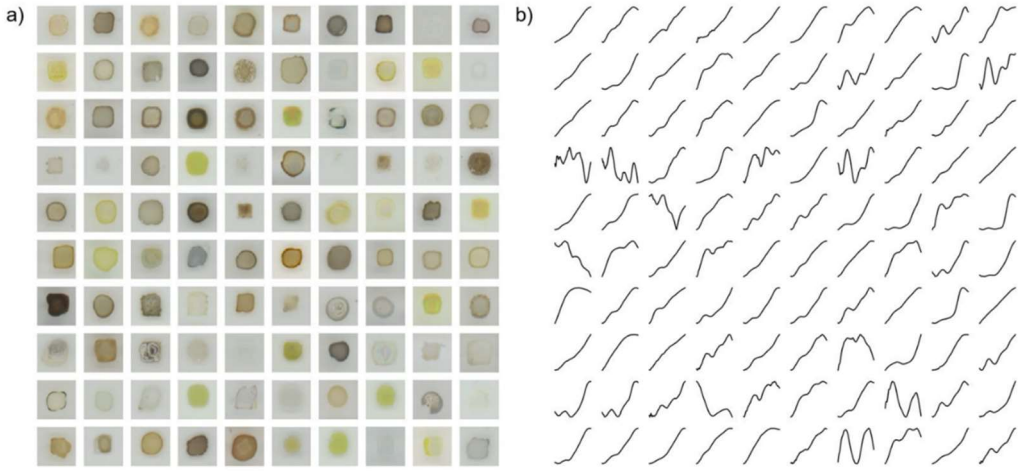
The selected data set is JCAP images and absorption spectra for 179072 metal oxides, and is provided by the US DOE and described by Figure 1, (Gregoire et al., 2018). Inputs are characterized as images and outputs are characterized as spectra. There are 220 output values for each input. Images were taken using a consumer flatbed scanner and scaled to 64 x 64 pixels. The colored imaging in the center is the printed material, whereas the coffee-ring-like structure on the outer edge is due to drying of the printed solutions and you can see these different structures in numerous images within Figure 2. All of the absorption spectra were measured using a dual sphere spectrometer, and represents 220 photon energies between the 1.31 to 3.1 eV energy range. The reported outputs are “fractional optical absorbance, which is the product of the absorptions coefficient and effective material thickness,” (Stein et al., 2019). These are represented by the graphed line data in Figure 2.

Figure 1:

Dataset	Content Description	Data Range	Data Size	Physical Units	Method
Images	Sample images	0–1 for every channel	(64,64,3,180902)	Color values for RGB	platebead scanner
spectra	fractional optical absorbance spectrum	0–ca. 0.5	(220,180902)	fractional absorb. coefficient	dual-sphere optical spectrometer

Description of JCAP Dataset. Note that images are normalized between zero and one, and are routed through three channels (RGB), (Stein et al., 2019).

Figure 2:



Correlation of input images and there corresponding output spectra line graph, (Stein et al., 2019).

Initial Strategy

The first step in our solution is to create a neural network. It was decided that only $1/11^{th}$ of the total output points of the spectra will be used. This simply means that to create the spectra graph every 11^{th} point, for a total of 20 points, will be used instead of 220 points in order to create the line. The reason for this is computational simplicity and to decrease some recurrent oscillations we were experiencing in our trials. In addition, we decided that we would be using 80% training data and 20% testing data. Lastly, in order to quantitatively measure the differences between tweaks to our neural network we created a root mean square error graph. Ideally, the root mean square error would be equal to zero, which would correspond to the actual data and the predicted value being exactly the same. Essentially, this allows us to argue that the closer to zero the model is, the better it is. Root mean square error was chosen because it is one of the most common methods for evaluating prediction models, and gives an answer in the units desired.

In order to expedite creation of the neural network, we found a sample code on Kaggle for a convolutional neural network to use as a starting point. This sample code is shown in Figure 3, (Antounes, 2022). This network is not specific to our data set, so therefore we just used it as a guide to then manipulate and build our own convolutional neural network from it. However, what exactly is the difference between a neural network and a convolutional neural network? A basic neural network has been explained previously in this paper, and a convolutional neural network is just a regular network with at least one convolutional layer included in it. In Figure 3 the convolutional layer is seen as “Conv2D”, and all convolutional layers do is essentially add a filter to the original input to prepare it for the rest of the layers of the network.

Figure 3:

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu', inp
ut_shape=(IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=1, activation=None)
])
```

Starting code base from Kaggle to build new convolutional neural network off of, (Antounes, 2022).

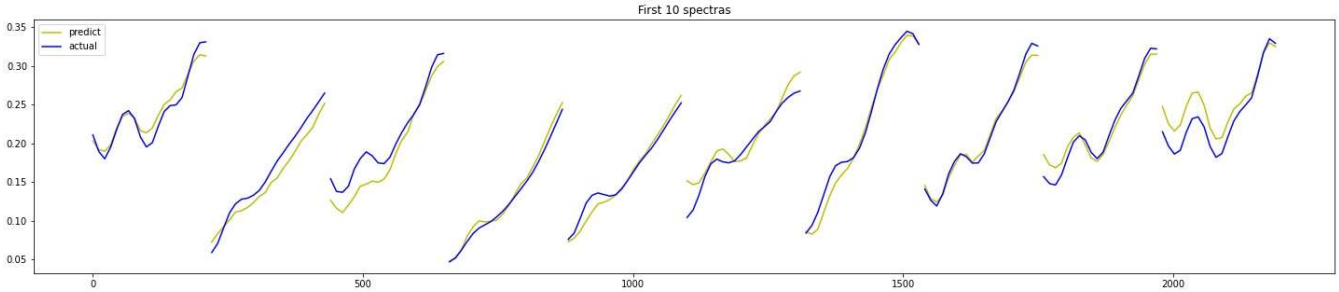
The base code in Figure 3 was altered to create a neural network that is better suited to our data set. For each of the following runs 10,000 samples were used with 50 epochs and a batch size of 32 unless otherwise stated. The sample size is the total number of images, the epochs represent how many passes the entire training dataset goes through, and the batch size represents the number of samples processed before the network is updated.

Initially we used the following layers in our network: Convolutional, Dense, Max Pooling, Convolutional, Dense, Max Pooling, Convolutional, Convolutional, Max Pooling, Flatten, Dense, Dense, Dense. This did not provide a successful model, so after other testing we settled on the following final order of layers: Convolutional, Dense, Max Pooling, Flatten, Dense, Dense, Dense. When running this network, it produced decent correlation between the training and validation data as shown in Figure 4. In addition, a big change between the networks was changing the activation function of the last Dense layer from none to softsign. In simple terms, softsign helps with the smoothing of data, which was needed because we were getting a lot of oscillation in our prediction with the initial model. By implementing softsign our graphs with the new network were much more visually accurate.

In an attempt to increase the accuracy of the network, an ensemble was created with 5 neurons to better improve the correlation, and this is shown in Figure 5. An ensemble is a method used to improve predictive performance in machine learning, and therefore is one type of uncertainty quantification. By using an ensemble in a neural network, instead of all the data being run through one network one time and producing an output, it will go through an ensemble of networks, or essentially multiple neural networks, before producing an output. These neural networks have almost exactly the same architecture, but have slightly different weighting in each so that multiple different outcomes are provided. This effectively allows the data to be dissected on different paths and then a final output can be decided from these options. For reference, Figure 7 effectively shows how an ensemble is implemented into a neural network. For our ensemble specifically, in order to decide the final output, the geometric average of all the neurons was calculated and this decision process is shown in Figure 8. The geometric average was used because it supposedly is better fit for data that could have large fluctuations between points, which we were experiencing initially. The formula for geometric mean is $(x_1 * x_2 * x_n)^{\frac{1}{n}}$.

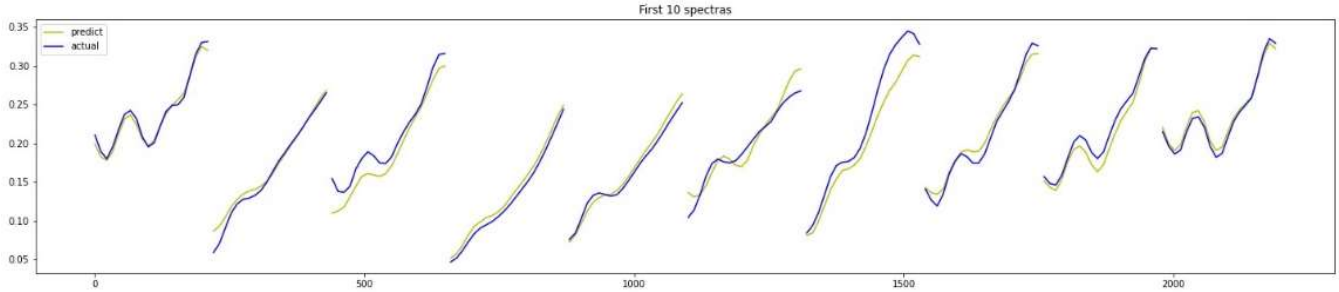
Furthermore, after playing around with the number of epochs and the batch size, it was found that the network had been slightly overtrained at 50 epochs so this was decreased to 40, and the batch size was decreased to 20 in order to train the network more often. This resulted in a slightly more visually accurate display of graphs, shown in Figure 6.

Figure 4:



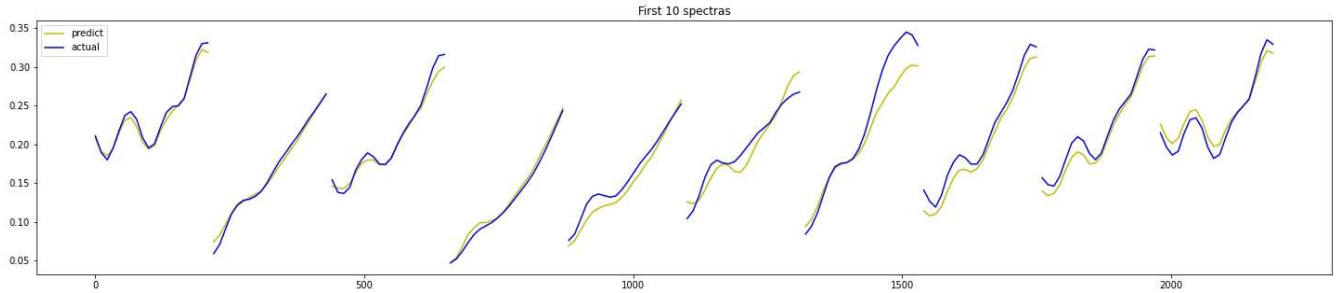
The first 10 spectra graphs with the final neural network architecture, 10,000 images, 50 epochs, a batch size of 32, and 1 neuron (without ensemble).

Figure 5:



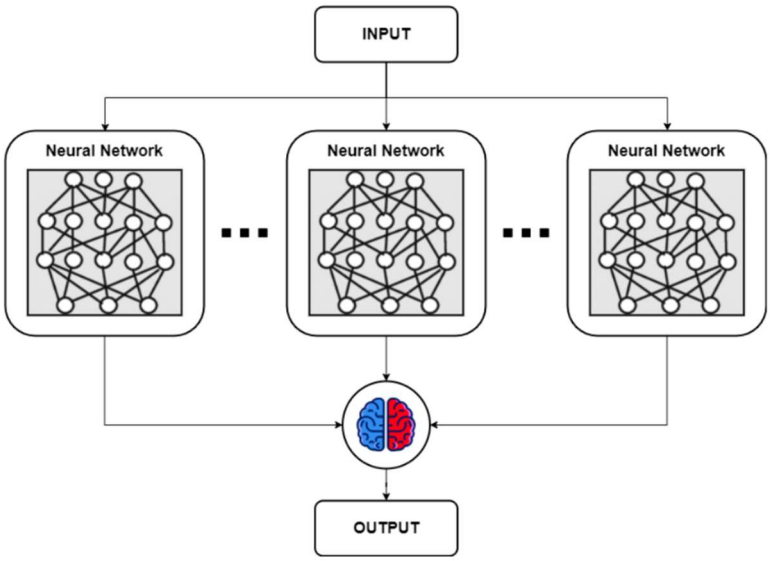
The first 10 spectra graphs with the final neural network architecture, 10,000 images, 50 epochs, a batch size of 32, and 5 neurons (with ensemble).

Figure 6:



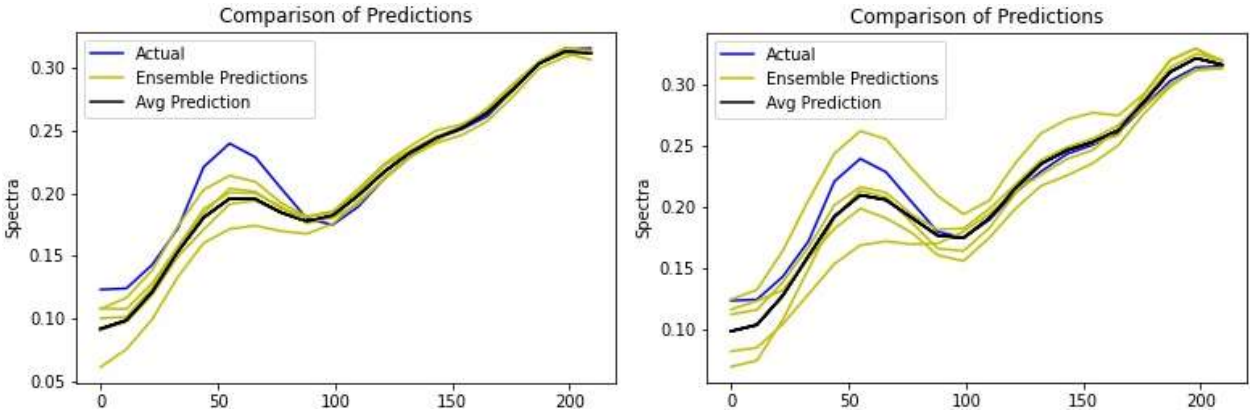
The first 10 spectra graphs with the final neural network architecture, 10,000 images, 40 epochs, a batch size of 20, and 5 neurons (with ensemble).

Figure 7:



Visualization of ensemble neural network including three neurons, (Adams & Lund, 2022).

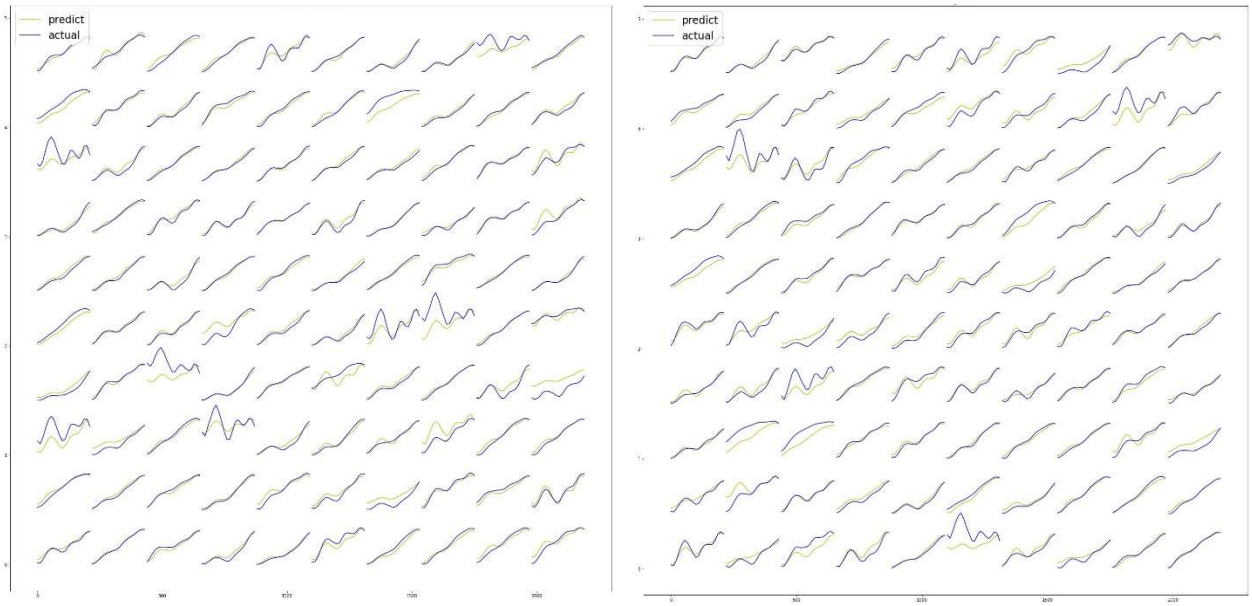
Figure 8:



Comparison of ensemble predictions (yellow), the final predicted selection (black), and the actual desired spectra graph (blue). Left: Figure 5's network. Right: Figure 6's network.

Figure 8 shows the progression of change from Figure 5 to Figure 6's predictions. It is clear from these images that the predictions in the left image (from Figure 5) are closer together and less varied than the predictions in the right image (from Figure 6). This suggests that the five networks of the ensemble were not changed enough in Figure 5 to produce varied predictions. It is also noticeable that all the predictions in the left image are underestimated, whereas the predictions on the right have more variance. All of these facts suggest that visually, Figure 6's network is a better prediction model than Figure 5. To further prove this, Figure 9 shows the comparison of two images of 100 randomized spectra graphs.

Figure 9:



Comparison of randomized 100 spectra graphs with predicted spectra in yellow and actual spectra in blue. Left: Figure 5's network. Right: Figure 6's network.

Currently we are working on methods to numerically determine the accuracy of our networks in order to compare them. After we accomplish this, the final step is to actually research different uncertainty quantification methods and implement them on our finalized network.

References

Adams, J., & Lund, M. (2022). ERAU REU Project Proposal: Ensemble Deep Learning. Nevada National Security Site.

Antounes. (2022, January 5). *Convolutional Neural Network for Image Regression*. Kaggle. Retrieved May 30, 2022, from https://www.kaggle.com/code/matthieubritoantunes/convolutional-neural-network-for-image-regression

Gregoire, J. M., Stein, H. S., Soedarmadji, E., Newhouse, P. F., & Guevarra, D. (2018, October 29). *JCAP images and absorption spectra for 179072 Metal oxides*. Office of Scientific and Technical Information. Retrieved May 25, 2022, from https://www.osti.gov/dataexplorer/biblio/dataset/1479489

Mission Support and Test Services. (n.d.). *About the NNSS*. Nevada National Security Site. Retrieved May 25, 2022, from https://www.nnss.gov/pages/about.html

Stein, H. S., Soedarmadji, E., Newhouse, P. F., Guevarra, D., & Gregoire, J. M. (2019, March 27). *Synthesis, optical imaging, and absorption spectroscopy data for 179072 metal oxides*. Nature News. Retrieved May 25, 2022, from https://www.nature.com/articles/s41597-019-0019-4