Quantifying Uncertainty in Ensemble Deep Learning

Emily Diegel*, Rhiannon Hicks†, Max Prilutsky‡, Rachel Swan§

Project Advsior: Mihhail Berezovski¶

August 2022

1 Abstract

Neural networks are an emerging topic in the data science industry due to their high versatility and efficiency with large data sets. Past research has utilized machine learning on experimental data in the material sciences and chemistry field to predict properties of metal oxides. Neural networks can determine underlying optical properties in complex images of metal oxides and capture essential features which are unrecognizable by observation. However, neural networks are often referred to as a "black box algorithm" due to the underlying process during the training of the model. This poses a concern on how robust and reliable the prediction model actually is. To solve this ensemble neural networks were created. By utilizing multiple networks instead of one the robustness of the model was increased and points of uncertainty were identified. Overall, ensemble neural networks outperform singular networks and demonstrate areas of uncertainty and robustness in the model.

*Embry Riddle Aeronautical University, diegele@my.erau.edu
†Embry Riddle Aeronautical University, hicksr10@my.erau.edu
‡San Diego State University, mprilutsky7577@sdsu.edu
§Embry Riddle Aeronautical University, swanr3@my.erau.edu
¶Embry Riddle Aeronautical University, berezovm@erau.edu

2 Introduction

As neural networks are a newly developed algorithm, there has been hesitation from industries and national laboratories to use this method. The key concept that these companies are seeking is understanding the rationale behind a system's decision-making or the interpretability of the model. It is necessary to define a method to quantify neural network uncertainty in order to evaluate how the model is functioning and whether or not it can be trusted to generate reliable predictions.

In the Chemical Science Journal[2], it is investigated how neural networks operate by utilizing a diverse absorption spectroscopy data set [3]. This data set is made up of of 178,994 unique materials samples spanning 78 different composition spaces, which includes 45 components, and contains more than 80,000 unique quinary oxide and 67,000 unique quaternary oxide compositions [3]. This data set is backed by years of extensive materials synthesis and optical characteristics.

The purpose of the research published in the journal article is to compare three different neural networks predictions of the same data set to each other and highlight the similarities and differences between them. To achieve this they used a variational autoencoder, deep neural network, and conditional variational autoencoder to predict absorption spectra from an image of a metal oxide. Our project is focused on improving their model 2, which is a hybrid dense and convolutional deep neural network model that was trained independently from other models [2]. In order to improve upon their deep neural network model our project involves introducing an ensemble neural network to decrease the overall uncertainty in the prediction algorithm.

The main results from the article are that the deep neural network produced results with an R^2 value of approximately 0.6. The goal is to improve the R^2 value of an ensemble neural network utilizing the same data set. After implementing an ensemble neural network, the overall R^2 improved from 0.56252 of a single network to 0.65422 with 10 networks working in parallel.

3 Background

Science and applied mathematics have been significantly impacted by artificial intelligence. Due to its deep and nonlinear structure, deep neural networks in particular exhibit a remarkable potential for grasping complicated scientific relationships. They have been effectively used for a variety of applications, including machine learning, computational imaging, and language processing. These networks work together to predict patterns from any data set that is translated into numerical values, and they operate by mimicking the neuron connections in a brain. In order to train neural networks, a collection of inputs and associated outputs are given to the model. The network then establishes appropriate weights and biases to connect the input to the output values. They predict an outcome with a specific probability after being given an input, but we are not

aware of their underlying thought processes. The most significant drawbacks of deep learning systems, however, are arguably their "black-box" nature and the accompanying lack of reliability. There are numerous instances where trained networks behave in an irrational manner. For example, if a network is trained on a certain amount of images and focuses on the left corner of each image, it may be missing a lot of information that is in the entirety of the image.

In order to use these models, it is essential to provide error bars alongside predictions to be clear on how trustworthy the predictions are. Therefore, quantifying uncertainty is a crucial component in strengthening the reliability of predictions generated by a trained network.

4 Data

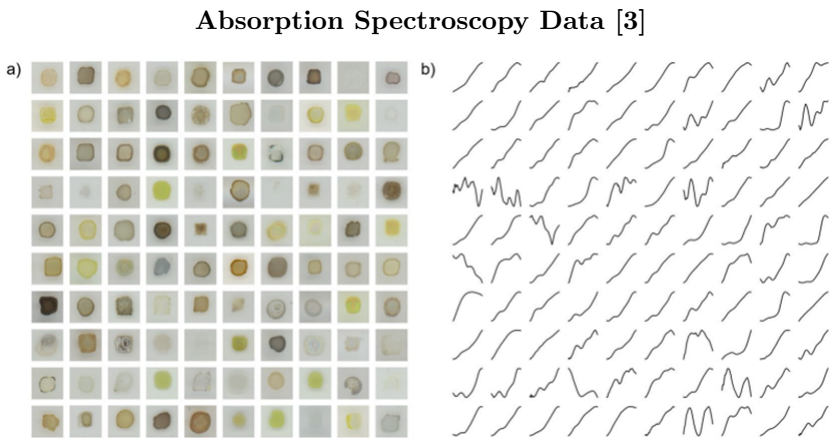

Figure 1: Materials images of metal oxides (a) with their corresponding fractional optical absorbance spectra (b).

The Absorption Spectroscopy Data for Metal Oxides consists of 180902 distinct samples. 1830 of these are blank samples, meaning that no material was placed on the substrate, leaving 179072 material samples. The images of the dried metal oxides were made by inkjet printing a mixture of elemental precursors, followed by thermal processing in an atmosphere including oxygen, as shown in Figure 1. The multiple samples of metal oxides include a range of 1-5 cation element combinations. Each sample image is 64 by 64 pixels with 3 red-green-blue channels with a corresponding absorption spectrum, consisting of 220 points, each representing an energy between 1.31 and 3.1 electron volts. Multiple compositions occur in the database as a result of duplication of compositions, sometimes with varied synthesis circumstances, to enable exploration of various synthesis conditions, give internal standards, and assess reproducibility.

The data set includes samples with a range of 1-5 metals in various combi-

Table of Metal Composition Comparison of 179,072 Samples

Number of Metals in Composition	Total samples out of 179,072	Percentage
1	9250	5.17%
2	19216	10.73%
3	66427	37.10%
4	82241	45.93%
5	1938	1.08%

Figure 2: Encompassing table numerically describing how often there are a specific number of elements in a combination in a given sample from the entire data set.

nations in each sample. Figure 2 shows a full breakdown of the data set.

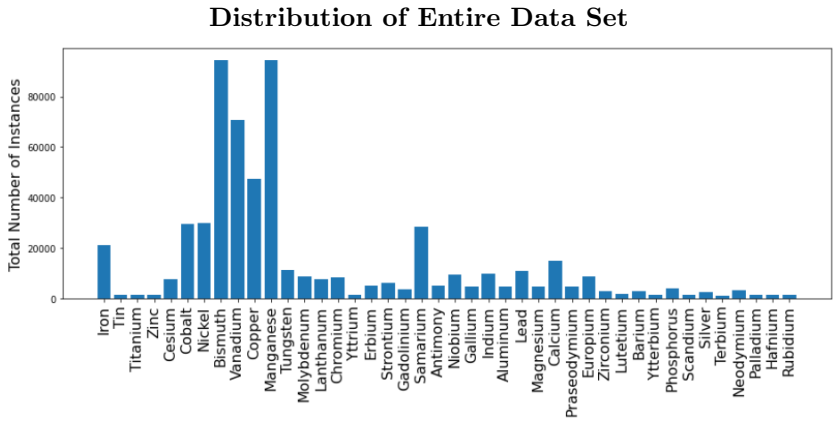

Figure 3: Representation of number of instances each element appears over the entire data set.

Figure 3 represents the number of times each metal appears in the data set, regardless of what concentration it appears with. Bismuth and manganese appear the most in the data set at roughly 90,000 instances, followed by vanadium at approximately 70,000 instances. Most of the metals appear much less than 20,000 times in the data set with the smallest at 1077.

The heat map in Figure 4 displays how often each metal appears in certain concentration amounts over the entire data set. Lighter colors represent a higher quantity of a metal appearing at a certain concentration, meanwhile darker colors represent a lower quantity of a metal appearing at a certain concentration. Typically when looking at the data sets' columns there are lighter colors at the lower concentrations down to 0.05, and there are darker colors at the higher concentration values. The 0.0 block represents where the concentration is less than 0.05 and it is apparent that there is a mix of elements who do and do not appear at this level of concentration. This provides insight that most metals

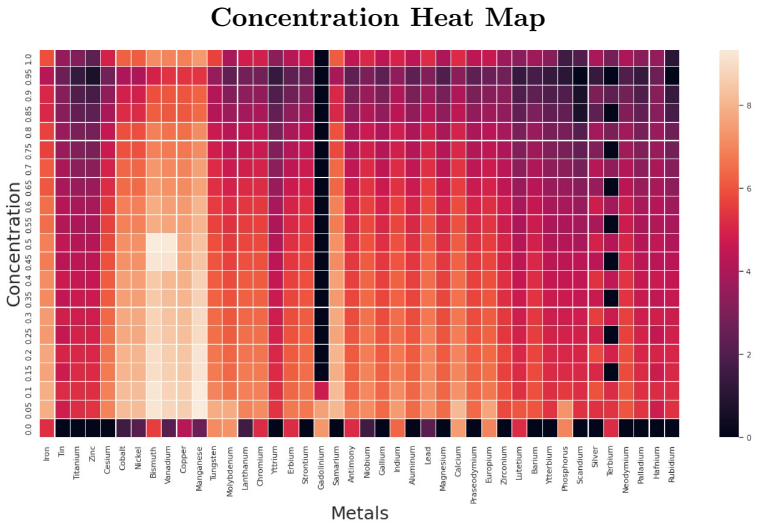

Figure 4: This heat map is completely symmetrical along the line y = x and provides a visualization of how often specific metals are combined together.

are only represented in a sample at a very low concentration when they are in combinations. It is also shown that the columns for bismuth and manganese are much lighter overall than other columns in this heat map, showing they appear more in the data set than other elements, no matter what concentration they appear at.

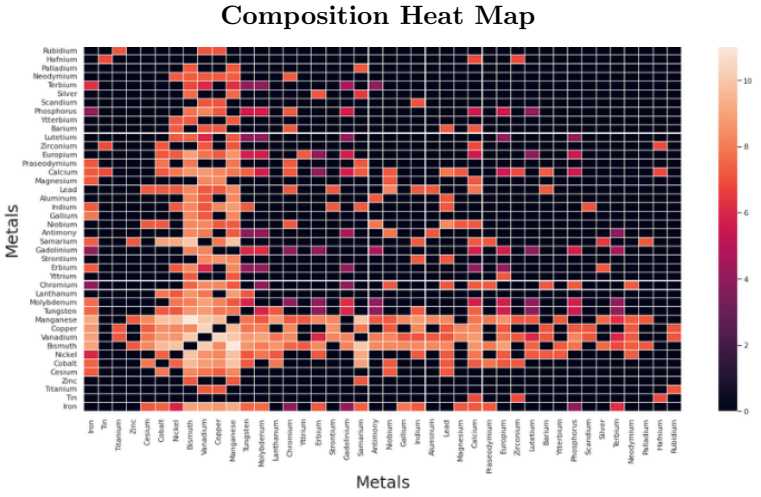

Figure 5: This heat map is completely symmetrical along the diagonal and provides a visualization of how often specific metals are combined together.

The heat map in Figure 5 is a visualization of how many times two metals appear together in the same combination in a sample. Lighter colors represent higher quantities of combinations, while darker colors represent a quantity closer to zero. It is clear that black is the most dominant color on this heat map, which demonstrates that there are a lot of metals that never appear in combination with each other. However, if we look at bismuth and manganese specifically, we see that this is the lightest square on the heat map and suggests these two elements often appear in combination with each other. Thus the conclusions from Figure 3 regarding bismuth and manganese are fortified by this graph.

4.1 Stratified Sampling

Due to limitations in available RAM and other computational limitations, our research was not able to be conducted on the entire data set and therefore we had to take a subset. This could lead to problems in training as our data is very unevenly distributed. Since certain metals appear more in combination, the training data set needs to be stratified or it will be skewed towards these metals. A stratified random sample of the data must be taken instead of a completely randomized sample to ensure the model is being trained well. When

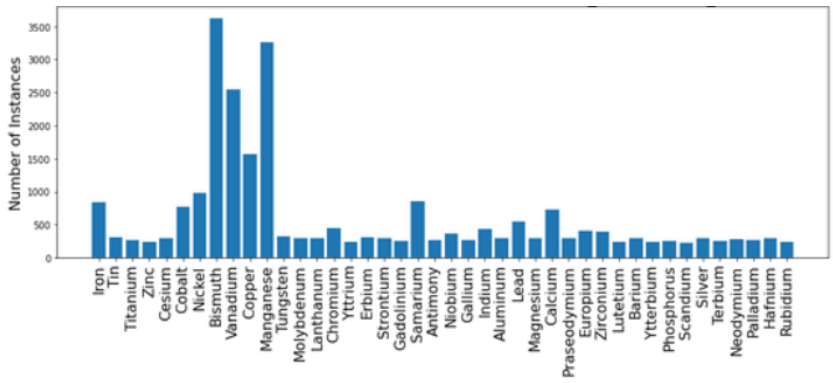

Figure 6: Representation of number of instances each element appears over the stratified data set of 7 825 images.

training a neural network, it is important to have a balanced data set with outputs being represented equally. If a training data set is unbalanced, the model will be better at predicting outputs that have a larger representation in the data set because these outputs were seen more during training.

Therefore, we created an algorithm that ensured every element in the data set was represented at least 300 times. The algorithm also incentivized elements to not exceed 900 instances to keep representation similar. However, because of how often bismuth and manganese are in combination with other low occurring elements, their number of instances had to increase well above 900 in order to

maintain the 300 minimum value of low occurring elements. As shown in Figure 6, this creates a stratified data set that can be used to train our neural network.

5 Method

Now, with a usable data set, a neural network was developed to translate a 2D image into an absorption spectrum. To reduce computational complexity every 11^{th} point of the 220 points was used for the absorption spectrum. Resulting in a total of 20 points. As shown in Figure 7, the shape of the spectra graph is maintained, thus the accuracy lost when using only 20 points is minimal.

Single Absorption Spectrum

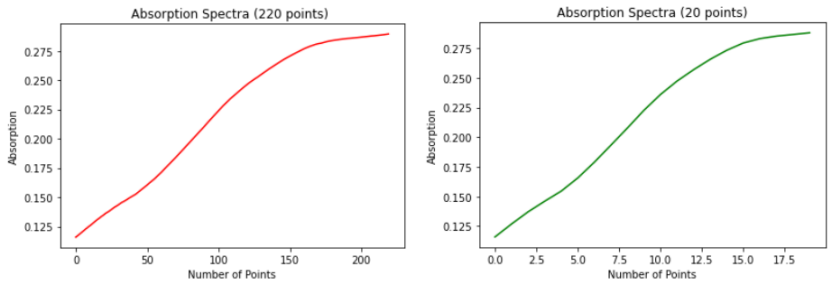

Figure 7: A comparison of these graphs shows that even thought the amount of output points is drastically decreased in the right graph, the shape of the spectra graph is maintained when compared to the left graph.

5.1 Neural Network Architecture

After analyzing the data and choosing the ideal training/testing subset, we built a neural network with 8 layers [2].

Each layer in the model contributes to the overall training of the network. The layer applies a filter using a kernel to a portion of the image and uses that filter to calculate a dot product which is fed through the output array. The filter shifts by strides of 1 by 1 using a 3 by 3 kernel in order to reduce the image size. The dense layer computes a weighted average of the input and runs it through an activation function, known as a nonlinear function. The max pooling layer determines the highest value for each patch in each feature map and focuses attention on the patch's key characteristic by pooling down into a smaller matrix. The dropout layer randomly sets neuron values to zero in deep layers to prevent overfitting. Finally, the flatten layer converts the data into a 1D array for inputting into the next layer.

Neural Network Architecture

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 64)        1792

dense (Dense)                (None, 62, 62, 128)       8320

max_pooling2d (MaxPooling2D) (None, 31, 31, 128)       0

dropout (Dropout)            (None, 31, 31, 128)       0

flatten (Flatten)            (None, 123008)            0

dense_1 (Dense)              (None, 128)               15745152

dense_2 (Dense)              (None, 64)                8256

dense_3 (Dense)              (None, 20)                1300

=================================================================
Total params: 15,764,820
Trainable params: 15,764,820
Non-trainable params: 0
```

Figure 8: Description of single neural network model architecture.

5.2 Training

For our implemented neural network, we incorporated early stopping into our model to prevent overfitting. T his f unction s tops t raining t he n etwork a t the most optimal point where the difference b etween t raining l oss a nd validation loss begins to increase. The maximum number of epochs is capped at 50 as any more would generally result in severe overtraining. We used a batch size of 32, meaning 32 images were shown before the weights updated. The loss function used to optimize the training was mean squared error. As the model is trained, it is tested in parallel to evaluate if the network is training properly. The subset size used to train this model was 7,825 images, as shown in Figure 6 of the stratified data s et. 80 percent of t he images were used for training and 20 percent were used for testing. Testing data is used to ensure that the network can predict values from images that it has not seen before. The images that are used to train the data should not be the same to test if the network is able to function as intended. The purpose of validation data is to introduce unseen inputs to the network and see how accurate the prediction is. In our model 2,000 random validation images were used in making predictions.

5.3 Ensemble Techniques

Ensemble neural networks are multiple individual networks training in parallel. Each network trains slightly different due to the random initialized weights and biases. Even if a network had the same architecture and same training data, it would never train the model the same due to the initialization. Therefore, in an ensemble network, each individual network produces slightly varying results. In order to combine the results of the ensemble, various approaches can be applied such as averaging the results or using another algorithm to combine the predictions. These approaches will, in general, provide a better result. This better result comes at the cost of a much longer training period. As shown in figure 9, the separate networks think in parallel and combine in the end for a final prediction.

Diagram of Ensemble Neural Network [1]

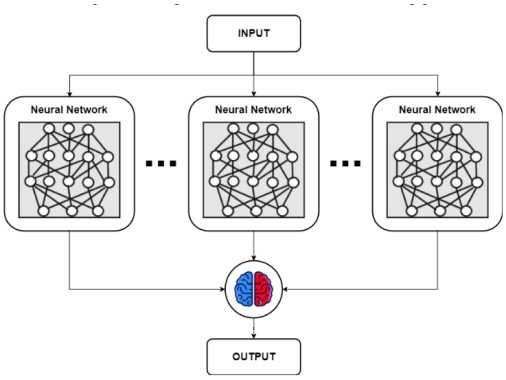

Figure 9: Diagram of the flow of an ensemble neural network. Each individual network receives the same input but produces their own output which is later combined for a final prediction.

In addition to higher accuracy, ensemble networks are a technique to quantify the uncertainty in a network. They highlight regions of uncertainty by displaying the unique variations in each network's predictions. The model is more certain if all the networks predictions agree on a value. When the separate model projections are drastically different from one another and do not exhibit agreement, there are areas of uncertainty present. This method evaluates the model's dependability and robustness, allowing for a better understanding of how well the model is performing.

5.4 Ensemble Implementation

For our ensemble, ten individual networks with the same architecture were trained in parallel. The architecture can be referenced in Figure 8. The predictions of the 10 networks were combined using an average and the package

shuffle split was used to split the data 80/20. This function randomizes the spread of the training and testing data based on the number of splits needed. For instance, in an ensemble, the training and testing will be from various parts of the data and will have overlap. The individual networks train on similar data with approximately an 85 percent overlap between any two individual networks. The effect of this can be seen in Figure 10 below.

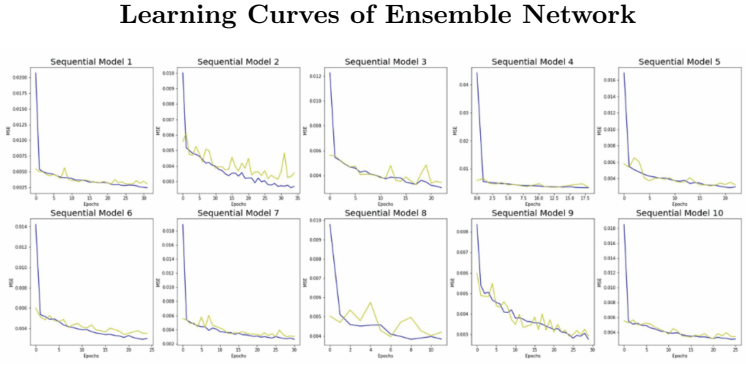

Figure 10: Mean squared error of training (blue) and testing (yellow) data of ensemble. Visual representation of how each individual network is being trained differently.

Figure 10 demonstrates how the mean squared error changes for each of the 10 networks in the ensemble. This loss function was used to optimize the model during the training. The yellow line represents the testing data, and the blue line represents the training data. Each network was trained using early stopping, altering the number of epochs used to train each model. As the number of epochs increases, the mean squared error decreases and this is because weights and biases are adjusted according to the value of mean squared error. The smaller the mean squared error, the better the overall network is performing. Figure 10 shows how much variance there is in training the individual networks, which will directly affect the predictions of the networks.

6 Results

6.1 Singular Neural Network

Figure 11 depicts a specific example of what our model is doing. The image on the left gives an example of a spectra image, and the graph on the right compares the corresponding actual spectra graph and single neural network prediction. Looking at this graph it is clear there are some oscillations and jagged points in the prediction. Overall this would be considered a good prediction since it is so close to the actual prediction, follows the same general shape, and has a relatively low mean squared error value.

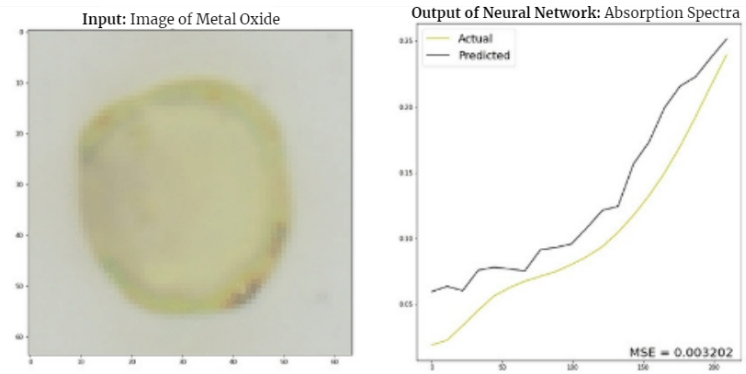

Figure 11: Singular example of one neural network prediction with comparison to actual image spectra and actual spectra graph.

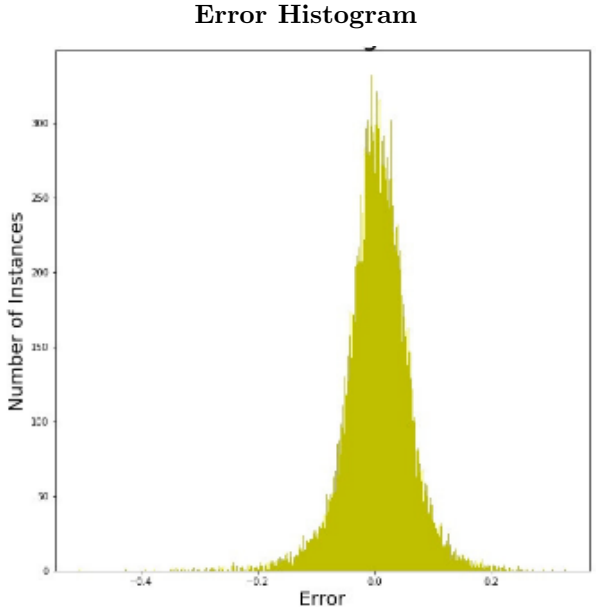

Figure 12: Histogram distance error visualization for single neural network predictions with overall mean squared error value.

In order to get an encompassing picture of how the network is performing over 8000 samples, a histogram was created as shown in Figure 12. This graph provides a representation of the distance error between the actual spectra line graph and the actual line graph, and shows how often specific values of error occur. Visually, it is clear that Figure 12 has a bell shaped distribution curve,

which is expected. This reinforces the idea that our single neural network model is making good predictions. There are higher number of instances for smaller error values. In fact, most of our predictions fall between ± 0.2, which is an acceptable level or accuracy for our model. Something else to notice is that the mean squared error value for the entire single neural network is 0.003202. Since the ideal value for this metric is zero, this also supports the claim that our network is making mostly accurate predictions in terms of the distance between actual and predicted spectra graphs.

Figure 12 helps us understand how well our model is predicting, but to further verify our claim that our model is working well we created a scatter plot of the actual versus predicted value of the network seen in Figure 13.

Single Network Scatter Plot Error

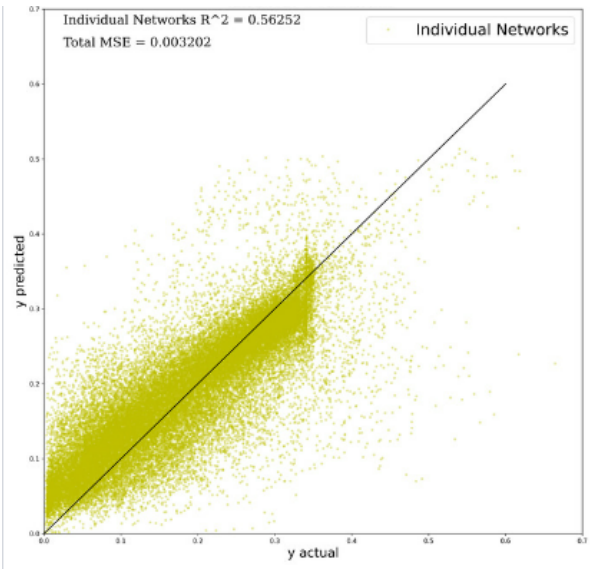

Figure 13: Scatter plot error visualization for single neural network predictions with R^2 value.

Figure 13 depicts a scatter plot with every prediction from the single neural network. The predictions are plotted such that that the line y = x (plotted in black) represents the most optimal location for plotted points as the predicted spectra exactly matches the actual spectra value. It can be seen that most of our predictions are clustered around this line. The distance error is represented in the y direction, so points above y = x are overapproximations and vice versa.

The purpose of this graph is to view the results in a different w ay. From this graph we can see that at the beginning of the line y = x in the lower left corner there are a lot of overapproximations, however, when looking at the upper right portion of the cluster we can see that there are more underapproximations at that point. Another benefit of this graph is that since we have a specific line

that ideally the data would be clustered around, we can fit an R^2 value to our predictions. As shown in Figure 13 the R^2 value for the single neural network is 0.56252. Since an ideal R^2 is 1.00, this suggests that while our predictions are generally good, there are definitely some consistent outliers in out data that if improved upon would decrease the uncertainty in our network.

6.2 Ensemble Neural Network

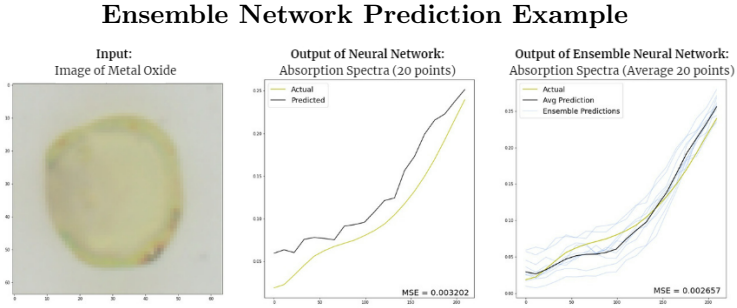

Figure 14: Comparison of 10 individual network predictions, the averaged ensemble prediction, and ground truth spectra for a singular spectra image.

Figure 14 shows a sample metal oxide from the data set on the left, the prediction of our singular network in the center for reference, and the ensemble predictions and finalized prediction on the right. There are 10 predictions from the individual neural networks, which are averaged in the ensemble to produce a single output. From Figure 14 alone we can see that the ensemble has already improved our model. First, the averaged predicted spectra is much smoother than any of the single network predictions. Second, it is clear that the mean squared error of the ensemble prediction model is even lower than the single network's mean squared error, which was already low. This fortifies that our prediction model is giving accurate graphs.

Figure 15 represents the error of 2,000 predictions of the ensemble neural network. This graph provides a representation of the raw error between the actual spectra line graph and the actual line graph, and shows how often specific values of error occur. In comparison to Figure 12, the ensemble neural network predicts outcomes closer to the desired value. Most of the error occurs between ± 0.1 The data ranges from 0-1, we can observe that most of the models predictions have less than 10 percent error.

A second graph was created to show the distribution of error in the ensemble neural network and provide another method of measuring the error in the model's predictions. The characteristics of how to read the graph are the same as in Figure 13. Figure 16 depicts a scatter plot with every prediction from an ensemble neural network whereas before, Figure 13 was just for a single network. In the bottom left of the graph, an overestimation is observed as the majority

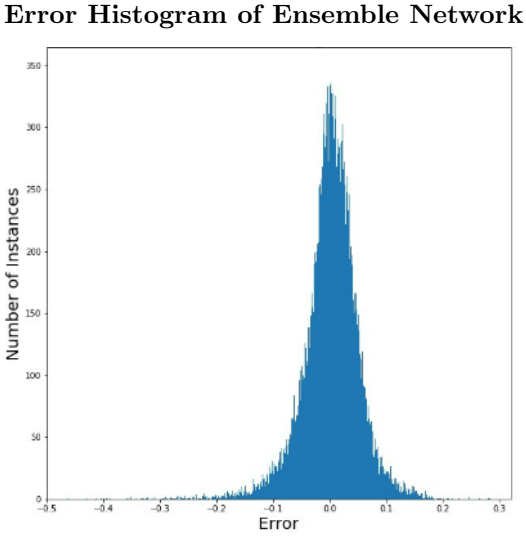

Figure 15: Histogram distance error visualization for an ensemble neural network predictions.

of the points fall above the desired output. The overall R^2 value of the error of an ensemble network was 0.65422. Ideally 1.0 is a model with zero error, yet after implementing an ensemble neural network, the overall R^2 improved from 0.56252 of a single network to 0.65422 with 10 networks working in parallel.

Figure 17 depicts the differences in error of the single network and an ensemble network. The mean squared error for the ensemble network is smaller than the neural network, which indicates that the ensemble overall performed better than the single network. The R^2 value is closer to 1.0 for the ensemble network, therefore the model has less observed error. Overall, ensemble neural networks have a smaller mean squared error and larger R^2 value over 2,000 predictions, demonstrating the advantages of using ensemble networks.

With the ensemble made, it is important to find a way to evaluate how well the model performed. Since the output is regression and not categorical, this method is not immediately apparent. To start, we recreated the scatter plot error graph that we made for the single neural network in Figure 13 to include all of the individual predictions as well as the final ensemble output. This is shown in Figure 18. The characteristics of how to read the graph are the exact same as in Figure 13.

In Figure 18 the yellow points represent every prediction of all 10 individual neural networks. For this reason the R^2 value of the individual networks increased to 0.65422, which is the average of all the yellow points. On this graph the averaged prediction from the ensemble is plotted in blue, and it is visually apparent that the blue cloud is more clustered around the line $y = x$ than the yellow cloud. This is expected since the ensemble prediction is the average of

Ensemble Neural Network Scatter Plot Error

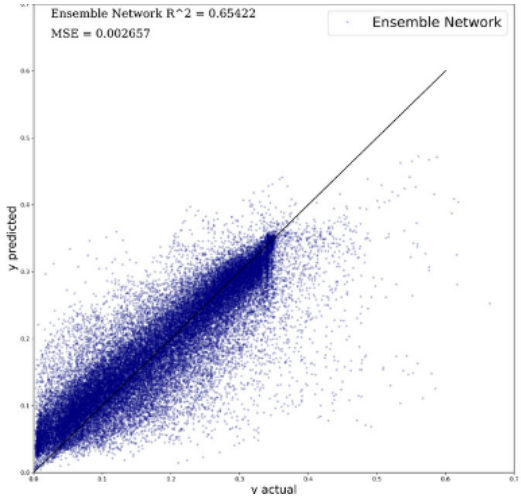

Figure 16: Scatter plot error visualization for an ensemble neural network predictions with R^2 value.

Comparison of Error

	Mean Squared Error	R Squared Value
Single Neural Network	0.003202	0.56252
Ensemble Neural Network	0.002657	0.65422

Figure 17: Individual mean squared error and R^2 values for both the singular network and the ensemble network.

all 10 individual predictions. Also, the R^2 value of the ensemble is 0.65422, which is higher than the individual networks' R^2. Based on how these numbers changed we are able to conclude that using an ensemble improve the accuracy of our overall predictions by lowering the amount of uncertainty in the model.

6.3 Implemented Uncertainty Quantification

Along with providing better predictions, ensemble networks themselves are a method of Uncertainty Quantification. Since multiple models have contributed to the final output it is important to investigate the distribution of the individual networks predictions. As mentioned earlier, the 10 individual networks in the ensemble will produce slightly different predictions and the variance in these predictions demonstrate areas of uncertainty in a model. To accomplish this we made a 95 percent confidence interval around the 20 points predicted for each spectra. This can be seen in Figure 19 and 20 below. Above we can see that the spread of the ensemble predictions in Figure 19 match the changing area of the confidence interval. By creating these confidence intervals we can

Ensemble Neural Network vs Neural Network Scatter Plot Error

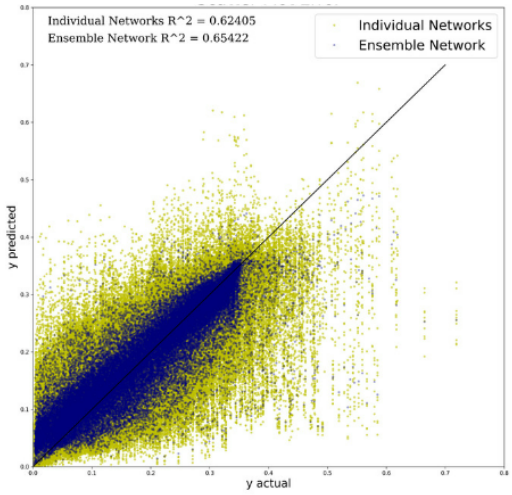

Figure 18: Scatter plot error visualization comparison of individual network predictions and the averaged ensemble neural network's final prediction with associated R^2 and mean squared error values.

Single Ensemble Prediction with 95 Percent Confidence Interval

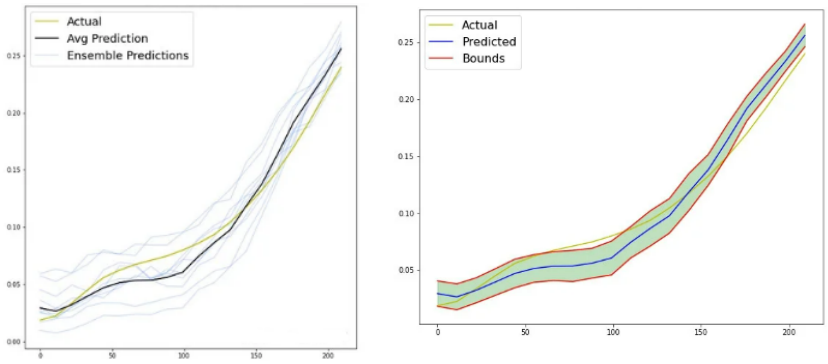

Figure 19: Comparison of individual network predictions (left) with changing area of a confidence interval (right).

see that the later points of the spectra are closely distributed since the area is relatively small. Conversely, the earlier points are more widely distributed since the area is relatively large. Areas that are larger, demonstrate areas of less certainty, because the networks individual predictions are farther off from each other. The closer the predictions are to each other, the more certain the model is about that prediction. More certainty is observed closer to the 220^{th} point in Figure 19. Now, let's look at this idea across the entire data set.

In Figure 20 below we separate the data by each of the 20 points of the spectra to see the relationship between error and uncertainty. Standard deviation is measured in Figure 20 as it is the sole factor in determining the size of the confidence interval.

Pointwise Error and Pointwise Standard Deviation of Ensemble Neural Network

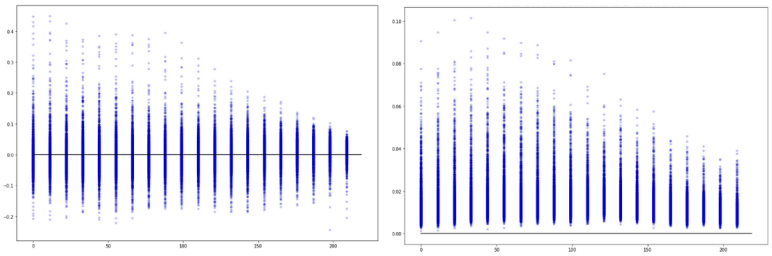

Figure 20: Comparison of point-wise error in each of the 20 points in each 2,000 predictions (left) to the point wise standard deviation of the 20 points in each 2,0000 predictions (right) in the ensemble network.

In Figure 20, the point wise error starts out widely distributed but becomes closer as we get closer to the 220^{th} point. This type of behavior is also demonstrated in the standard deviation graph. This shows that the error of the network is seemingly in proportion with the uncertainty. This is important as the error plot can only be made when the true value is known. However, if the true value is known then we have no need for a neural network. The plot of the standard deviations can be made even if the actual value is unknown as it is made from the network outputs. This will help us determine a standard deviation cutoff for which predictions to trust and which ones to remove. Overall, our model demonstrated more error and more uncertainty in predicting the points on the left of the spectrum in comparison to the points on the right of the spectrum.

7 Conclusions

The singular networks were able to make spectra graph predictions consistently within 10 percent error to the actual spectra. Additionally, adding an ensemble reduced the mean squared error of the model while increasing the R^2 value of the predictions with respect to the actual spectra.

In comparison to the previous research done by Chemical Science[2], our data produced superior results. The deep neural network produced an R^2 value of 0.6 when trained on 100,000 samples, whereas our ensemble neural network produced an R^2 value of 0.65422 when trained on only 7,000 samples. This shows that our model has a more accurate predictive process when trained on less than $\frac{1}{10}th$ of the amount of samples used in the original research.

Finally, we were able to use uncertainty quantification techniques such as confidence intervals and point wise error graphs to determine the robustness of our model. Overall, our model demonstrated more certainty in predicting points on the right side of spectrum than on the left side.

8 Further Research

For future research, we would be interested in the impacts of using the full data set to train the network rather than a subset. Secondly, we could compare impacts of using 220 points on the absorption spectrum versus only 20. One other factor that could be altered and investigated is the neural network architecture and how that may impact the error of the network. Finally we could investigate model sensitivity towards changes in training data and introducing noise to the model.

9 Acknowledgements

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

This project was proposed by the Nevada National Security Site.

10 References

1. cerliani2020 M. Cerliani, *Neural Networks Ensemble*, Medium, (2020).

2. stein2019H. S. Stein, J. M. Gregoire, et al, *Machine learning of optical properties of materials – predicting spectra from images and images from spectra*, Chemical Science, vol. 10, (2019), pp. 47–55.

3. stein2019 H. S. Stein, E. Soedarmadji, P. F. Newhouse, Dan Guevarra and J. M. Gregoire, *Synthesis, optical imaging, and absorption spectroscopy data for 179072 metal oxides*, Scientific Data, vol. 6, no. 1, (2019).