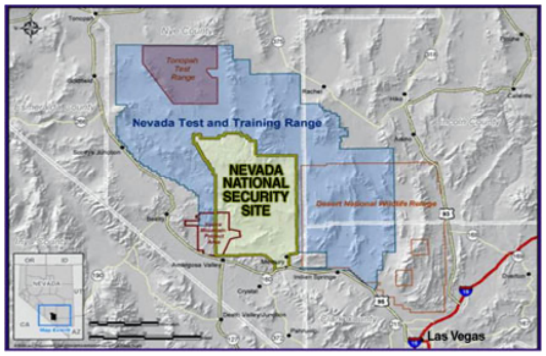
ERAU Industrial Math Project Proposal: Ensemble Deep Learning

The Engineering Technologies Division of the Science and Technology Directorate at the Nevada National Security Site (NNSS) is pleased to partner with students at Embry-Riddle Aeronautical University (ERAU) to develop methods for quantifying uncertainty in neural networks to help improve network interpretability, efficiency, and correctness, using publicly available data.

NNSS Background


Tonopah Test Range
Nevada Test and Training Range
NEVADA NATIONAL SECURITY SITE
Desert National Wildlife Range
Las Vegas

The Nevada National Security Site (NNSS) is a U.S. Department of Energy research and development complex larger than the state of Rhode Island, located in the Nevada desert, approximately 65 miles north of Las Vegas. The NNSS has a multi-faceted mission, with its two primary thrusts being Defense Nuclear Nonproliferation and Stewardship Science and Experimentation (SSE). The SSE mission centers around the National Nuclear Security Administration's (NNSA) science-based Stockpile Stewardship program, which according to the NNSA, "combines nonnuclear experiments, highly accurate physics modeling, and improved computational power to simulate and predict nuclear weapon performance over a wide range of conditions and scenarios."

One of the primary laboratories for SSE experimentation is the U1a underground complex at the NNSS. At almost one thousand feet underground, this facility supports the nation's subcritical experimentation program through experiments in which plutonium and other materials are subjected to high pressures and shocks, replicating conditions similar to nuclear explosions (`nnss.gov`).

Cygnus X-ray diagnostic machines (`nnsa.energy.gov`)

X-ray radiography is one method by which data is collected from these high-energy tests at U1a and other nuclear physics facilities. The Cygnus dual x-ray machines at U1a capture high-resolution snapshot images at prespecified times during experiments that take place on the order of less than a second. While the amount of information contained in these images is immeasurable, significant post-processing is required to extract the "truth" from the images. The basic post-processing is a roughly 10-step sequence, and we are constantly reevaluating our algorithms to improve this process. Beyond this, the analysis is highly subjective.

Experiments such as these are incredibly rare (the only place the US conducts subcritical nuclear tests is at our

Jesse Adams, Senior Scientist, Nevada National Security Site,
P.O. Box 98521 M/S NLV039, Las Vegas, NV, 89193-8521
702-295-5352 adamsjj@nv.doe.gov

This project proposal was prepared by Mission Support and Test Services, LLC, under Contract No. DE-AC52-06NA25946 with the U.S. Department of Energy.

facility), and the necessary analysis is not available in journal articles or through internet searches; it requires the development of new ideas and methods tailored to specific data examples. Many high-impact, multi-million dollar decisions are made based on assessments that come from data analyses such as these.

The project presented here is part of a new movement within the nuclear National Labs to approach radiographic image analysis in a revolutionary way using machine learning, or more specifically, using deep neural networks (DNNs). This area of research, more generally called deep learning, is at the forefront of cutting-edge mathematics, but much remains to be understood about implementing them within physics applications with results being physically interpretable.

Neural Network Background

Advances in machine learning, specifically neural networks (NNs) and DNNs, have opened new doors of opportunity to data modeling and prediction. In particular, we are interested in using convolutional neural networks (CNNs) to analyze image data to extract information and determine values of interest about an experiment. We want our results to be physically meaningful within the specific physics application.

A neural network is an algorithm modeled after the network of neurons in the human brain, which, when trained, is able to identify underlying relationships in a set of data. In terms of the human brain, consider how children learn to identify cats. Rather than memorizing images of cats, children learn to associate them with having fur, pointy ears, long whiskers, four paws, and a tail. Then when a child encounters a new cat, or even a cartoon image of a cat, they know it is a cat without having to ask or be told, simply because they have learned what features represent a cat. The neural network similarly attempts to learn, from input data and corresponding labels, how to label new inputs. We describe some very basic terminology for neural networks below, but we acknowledge that various aspects of this project should be comfortably approachable with minimal understanding of the neural network architecture.

Neural network diagrams, like the one below, are traditionally read from left to right. Sample data is fed into a network through an input layer. Networks can be designed to handle all sorts of numerical data from single values, to vectors, to 2D images, and even images with several layers, such as RBG images or a stack of frames from a video. The network below shows 3 yellow nodes in its input layer, so the input data would be samples of 3 values, such as a 3x1 vector. From the input layer, data is passed through hidden layers which are all the internal layers in a network. A DNN is a neural network with more than one hidden layer; the network diagram below shows 3 hidden layers.

Information is passed from each layer to the next through a series of weighted sums. The gray lines connecting the nodes in the diagram each have a weight associated with them, representing the strength of the connection between the nodes. Each node value can be computed by taking the weighted sum of the values in the previous layer, and then plugging this number into an activation function, usually a simple ReLU or sigmoid function. Each layer in the network extracts higher and higher level patterns from the data. The final layer of the network is the output layer. This is typically a single value, or vector of values.

Jesse Adams, Senior Scientist, Nevada National Security Site,
P.O. Box 98521 M/S NLV039, Las Vegas, NV, 89193-8521
702-295-5352 adamsjj@nv.doe.gov

This project proposal was prepared by Mission Support and
Test Services, LLC, under Contract No. DE-AC52-06NA25946
with the U.S. Department of Energy.

A network is trained using a set of data with known desired output values, called training data. The weights in the network are initialized to small, random values. Samples of the training data are fed through the network from left to right, called a forward pass and the network output is compared to the known, desired output value. Based on the difference between the actual and desired outputs, all the weights in the network are updated through a process called back propagation. The details of this are not important for this project, but a general understanding of the training process is useful.

After many iterations of forward and backward passes, the weights will be sufficiently trained and the network will have learned patterns in the data, enabling it to give the correct output values for the training data. At this point, the network is ready to have new data fed into it, where the desired output is not known. For example, we could train a network to predict the next day's temperature. Our input values could be the previous day's temperature, the month of the year, and the number of cloudy days in the preceding week. We might imagine that the previous day's temperature is a stronger predictor of the next day's temperature than the other input values, so the weights associated with the first input value will probably be larger than the weights associated with the other input values. We could train the network on data collected from the last 30 days, and then use the trained network to predict tomorrow's temperature.

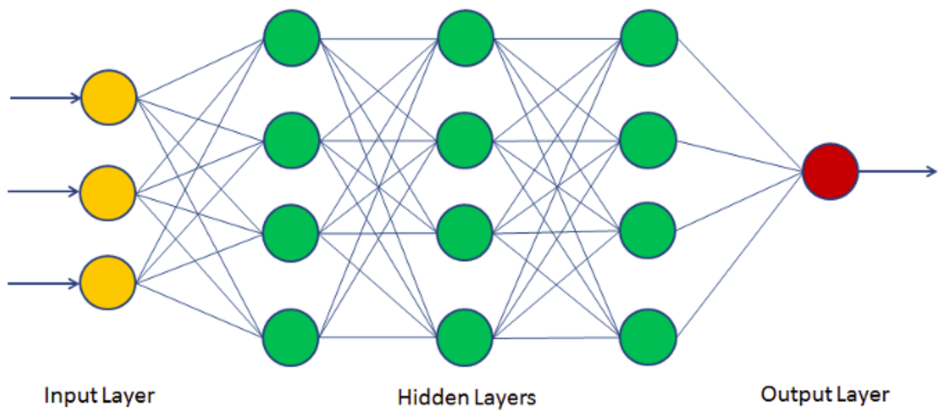


Diagram of a deep neural network with 3 hidden layers

For this project, we are interested in ensemble neural networks (ENNs) shown in the figure below. ENNs, as the name suggests, involve taking several NNs or DNNs and combining them to produce an output. There are a variety of ways to construct ENNs. The simplest is to take a single network architecture and train it several times, with different training data. This will result in networks that will provide slightly different outputs given the same inputs. These outputs can then be combined to produce a single output value e.g. by averaging, along with some statistical information such as a standard deviation.

More sophisticated ENNs can involve several more complex changes. Firstly, the training data used for the neural networks can be chosen in different ways. One popular method for this is bootstrap aggregation, or bagging, which involves a specific scheme for sampling the training data. Secondly, the NNs the ensemble is comprised of can be altered. This may mean altering the network architecture for the same input type, but could also resemble breaking the input into meaningful

Jesse Adams, Senior Scientist, Nevada National Security Site,
P.O. Box 98521 M/S NLV039, Las Vegas, NV, 89193-8521
702-295-5352 adamsjj@nv.doe.gov

This project proposal was prepared by Mission Support and
Test Services, LLC, under Contract No. DE-AC52-06NA25946
with the U.S. Department of Energy.

chunks, and having the NNs function on these different portions of the input. For example, if your input is comprised of multimodal data such as temperature, pressure, location, etc., the neural networks in the model might take only one or two of these as an input. A third change that can be made is the method used to combine the outputs from the ensemble members. This can involve a weighting scheme where the weights are decided by hold-out validation, or even using another network to combine the predictions. More sophisticated methods such as boosting, have been shown to produce better generalized results.

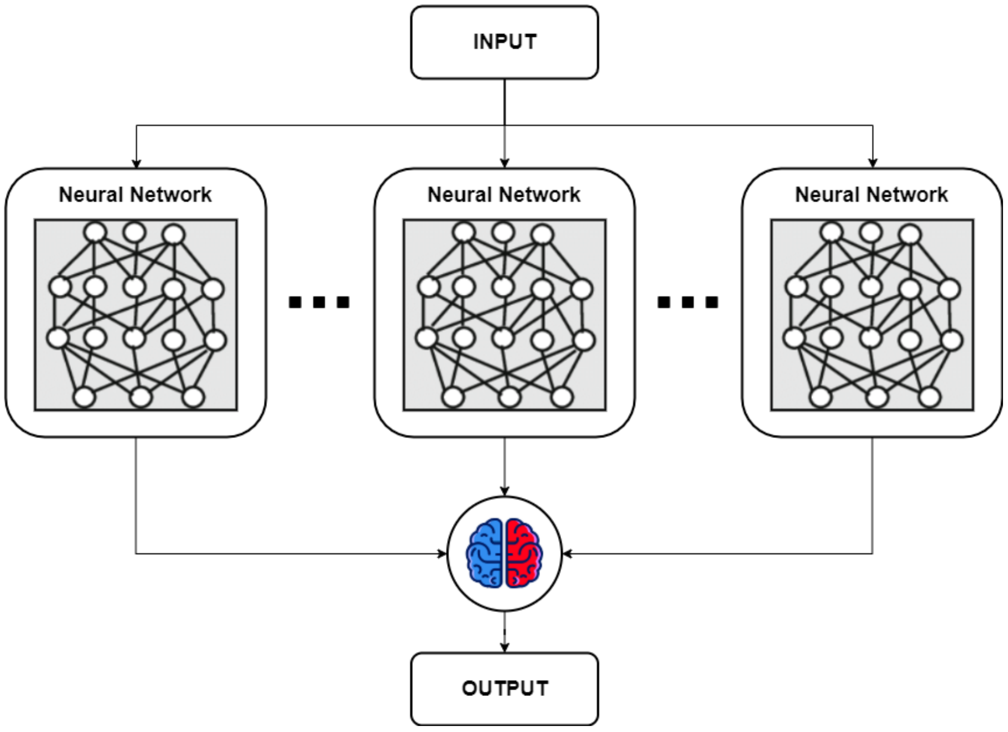


Diagram of an ensemble neural network.

Uncertainty Quantification Background

Although NNs have proven to be quite useful for solving a wide array of problems, a common criticism is that they are effectively a "black box" where an input is provided, some magic happens, and an output is produced. It is possible to test a network for accuracy, by supplying known data, but understanding how a network might react to different data is a complicated problem. One way to attempt to address this issue is via Uncertainty Quantification (UQ): given sources of uncertainty in the input data and the models used, is it possible to quantify the effect this has on the output? UQ involves assigning probability distributions to characterize the sources of uncertainty, and propagating these through the model to gain an output distribution via many computer simulations.

Jesse Adams, Senior Scientist, Nevada National Security Site,
P.O. Box 98521 M/S NLV039, Las Vegas, NV, 89193-8521
702-295-5352 adamsjj@nv.doe.gov

This project proposal was prepared by Mission Support and
Test Services, LLC, under Contract No. DE-AC52-06NA25946
with the U.S. Department of Energy.

Project Description

This may sound familiar to some of you who participated in this course last year. Last year, our team developed the variable stride method with the goal of training a specific subclass of neural networks quickly and efficiently. The ultimate goal is to implement the variable stride method in an ENN, and then develop UQ for this network. This is a lofty goal though, and computationally expensive with our dataset of images. So first we want to develop methods using smaller 1D datasets that require less computation, and then see if we can scale those techniques to our larger problem. We are hoping that the students at ERAU can help us investigate possible ENN architectures to facilitate this task.

Data and Project Package

- Publicly available multivariate dataset to regress on and quantify uncertainties (there are several possible data sets we can discuss)
- Document on uncertainty quantification
- Document on "variable stride" pooling method
- Python codes for "variable stride" pooling method

Potentail Project Tasks

We understand that for most students, this class will be their first introduction to neural networks, or even to machine learning in general. We suggest that the ERAU team take some time to learn the basics of computer image processing, machine learning, and neural networks. There are many excellent free online resources that students are welcome to explore to find what is useful to them. As a starting place, we recommend lectures 1 and 3 of MIT's Introduction to Deep Learning course. All lectures and course materials are available at `http://introtodeeplearning.com/`. A basic understanding of neural networks, in particular, deep neural nets, would be beneficial, but we intend that this project is still tractable without a complete understanding of deep learning. Below is a suggested semester-long project outline. Actual project direction and decisions are up to the discretion of Dr. Berezovski.

1. Background research: look into references and other online sources to become comfortable with the language and concepts of neural networks and uncertainty quantification. Please reach out if resources are needed.
2. After students feel comfortable with the language and concepts of neural networks, implement small NNs in Python using Keras/Tensor flow and play around with network architectures on the provided data set.
3. Decide on a single architecture, and train the architecture n times ($n \geq 10$). Produce confidence interval(s) on the output(s) from the n different trained networks.

Jesse Adams, Senior Scientist, Nevada National Security Site,
P.O. Box 98521 M/S NLV039, Las Vegas, NV, 89193-8521
702-295-5352 adamsjj@nv.doe.gov

This project proposal was prepared by Mission Support and Test Services, LLC, under Contract No. DE-AC52-06NA25946 with the U.S. Department of Energy.

4. Implement a more sophisticated ENN architecture, and see if we can produce improved output results, but via accuracy and statistics.
5. Develop a probability model for the architecture and provide UQ for the model.

Industrial Liaison Contact Information

Jesse Adams, PhD
Senior Scientist
Office: 702/295-5352
Cell: 702/506-7886
adamsjj@nv.doe.gov

Maggie Lund, PhD (Project PI)
Senior Scientist
Office: 702/295-3155
Cell: 609/556-0664
lundmc@nv.doe.gov

References and resources

Intro to Deep Learning, MIT:
http://introtodeeplearning.com/

A mathematical explanation of Neural Networks with code example:
https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9

Kaggle resources:
https://www.kaggle.com/learn/intro-to-deep-learning

Ensemble methods:
https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/
https://towardsdatascience.com/neural-networks-ensemble-33f33bea7df3

This project proposal was prepared by Mission Support and
Test Services, LLC, under Contract No. DE-AC52-06NA25946
with the U.S. Department of Energy.