Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
May 4, 2023
A Regional Conference of the Society for Industrial and Systems Engineering

Design of a DLNN Classification Tool to Gain Productivity Insights from Data for Non-Profits

Paige La, Máire Shrimpton, Martin Yanakiev, and Basmalh Hamam

George Mason University
Volgenau School of Engineering
Fairfax, VA 22030

Corresponding author's Email: pla4@gmu.edu

Abstract: A generic no-code tool to construct trained models for deep learning neural networks (DLNN) was developed for nonprofits and small businesses. The tool is known as Easy Insights. The tool is coded in Python and can be accessed via the web. The tool accepts the following inputs from simple pull-down menus and text boxes: the number of nodes, layers, epochs, and the testing/training split. The tool generates a DLNN and outputs a trained model, along with an accuracy on its performance. The performance metric is the overall accuracy of the trained model evaluated with Hold-Out validation. The tool was demonstrated in two case studies: Fairfax Fire & Rescue and for GMU Parking Department.

Keywords: Machine Learning, Accuracy, Deep Learning Neural Networks, Classification Analysis, Nonprofits

1. Context Analysis

Small business and non-profits can have complex processes and massive amounts of data on performance and their customer preferences. This data is largely not exploited by these enterprises due to the lack of analytical staff and analytical expertise. Stakeholder interviews with 30 small business and non-profits identified an opportunity to improve profits and efficiency by analyzing the enterprise data. Advances in technology related to Machine Learning and cloud data services provide the potential for deploying a no-code tool that small businesses and non-profits without expert analytical skills can leverage.

A generic no-code tool to construct trained models for deep learning neural networks was developed. The tool is known as Easy Insights. This tool is coded in Python and can be accessed via the web. The tool accepts the following inputs from simple pull-down menus: the number of nodes, epochs, runs, layers, testing/training split, hold out data percentage, and feature/target training data. The tool generates a DLNN and outputs a trained model, along with an accuracy on its performance. The accuracy is measured against hold-out data (i.e. data not used in training or testing).

A case study for Fairfax Fire & Rescue (FFR) was conducted for the task of assigning incident fire codes to the National Fire Incident Reporting System (NFIRS). The current manual process uses report narratives to assign codes. The manual code assignment is only 50% accurate. The target prediction is the NFIRS-IncidentTypeCode. There are 36 possible NFIRS codes. The features are: UnitType, UnitType-Code, NFIRS-Apparatus-Primary-Action-Taken, and NFIRS-Apparatus-Primary-Action-Taken-Code. There are 63 possible actions taken and 30 possible unit types. Example rules for assignment of the NFIRS code: (1) NFIRS code is 162 when the Unit Type is Brush, Action Taken is Standby, and NFIRS Apparatus Primary Action Taken Code = 1; (2) NFIRS Code is 100 when Unit Type = Battalion Chief and NFIRS Apparatus Primary Action Taken Code = 1,2,3,4,6,7,9,12, or 15. A DLNN with inputs of 3 hidden layers with 32 nodes each and 500,000 runs was developed. The DLNN trained with an 80/20 training/testing split for 1000 epochs. The DLNN initially achieved an accuracy of 32%, which was improved to an accuracy of 45% by adjusting the number of epochs and duplicating low occurrence data.

Another case study for GMU Parking was conducted for the task of predicting future space availability of outdoor parking lots at the GMU Fairfax campus. The goal is to maximize permit sales and predict future lot capacity. The current manual insights report is only 70% accurate. The classification problem is predicting lot capacity given features: year, month, weekday, time code, and lot code. The target prediction is lot capacity, shown in range increments of 25. A DLNN with inputs of 3 hidden layers with 50 nodes each was developed. The DLNN trained with an 80/20 testing/training split for 100 epochs. The DLNN model achieved an accuracy of 85%.

2. Stakeholder Analysis

ISBN: 97819384964-4-8

2.1 Fairfax Fire and Rescue Use Case

The main tension is that there is constant correcting and reiterating of the fire reports between the analytics team and reporting authorities. In the fire department, the analytics team is required to classify every fire emergency according to an NFIRS code (national fire incident response system) that is required by the US Fire Administration. The problem is that they rely on firsthand reports which are not always accurate. Firefighters are responsible but are not trained to classify NFIRS codes so 50% of the original codes are incorrect resulting in multiple rounds of corrections. This creates a delay in funding, or incorrect funding and equipment.

2.2 GMU Parking Use Case

The main tensions for the GMU Parking stakeholders are between the parking services and the University Business Consultants (UBS). GMU Parking does not have analytics skills, so they contract this work for the UBS, but it takes too long to receive updates and costs too much to train the consultants on the parking practices. A no-code tool would take the place of the UBS so that their work is no longer needed to visualize the lot capacity. UBS is not paid by the parking services directly because they are an auxiliary service, meaning self-funded. Since UBS is paid by George Mason University and not parking services, they are not incentivized to work any faster and find better ways to analyze data. The current manual process to get an updated analysis report takes 7 days - a process that must first go through the Manual Parking Collector, to the consultants, then to the Parking Services.

3. Problem and Need Statement

The problem is that nonprofits and low-tech enterprises produce large datasets that are not being exploited because they don't have their own analytics method and lack the coding knowledge to create DLNN prediction models.

There is a need for a no-code tool that allows nonprofits to create prediction models for their businesses to increase profits and efficiencies without having to code.

For GMU Parking, the most important need is to have a tool that reduces time and increases accuracy. For the Fairfax Fire and Rescue Department, the most important need is to increase accuracy of NFIRS classification.

4. Concept of Operations

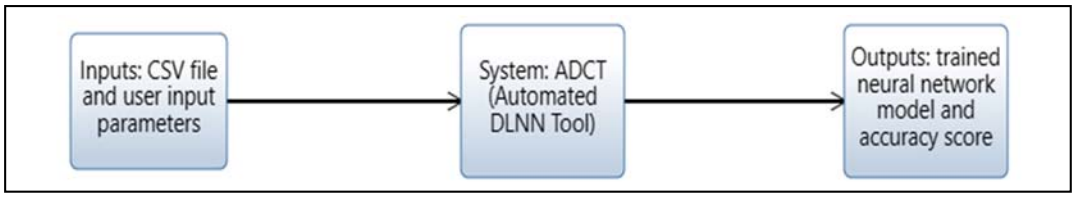

Figure 3. High level system inputs and outputs

In the high-level inputs and outputs of the system, (Figure 3) the input is a processed CSV file and user input parameters. The output is a trained neural network model and the model accuracy score, which is the percentage of correctly predicted data against Hold-Out-Data.

User Input Parameter Definitions:

1) Nodes: This algorithm relies on data input combined with coefficients or weights to amplify or dampen that input in order to assign significance to inputs for the task it is trying to perform.
2) Layers: As input is fed through the net, a row of neuron-like switches turns on or off. Taking the data as an initial input, each layer's output becomes the next layer's input.
3) Epochs: It specifies how many times the algorithm should train on the entire dataset. Data type and task determine how many epochs are needed.
4) Test/Train split: How much of the processed data will be trained. The remaining percentage of data will be used to test how accurately the model was trained.

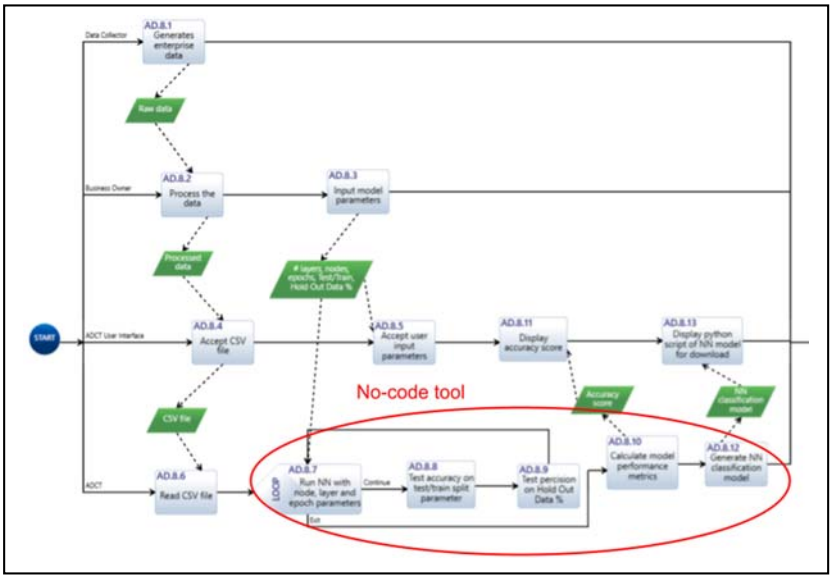

Figure 4. CONOPS action diagram

Easy Insights shall fulfill a need for all generic use cases. In the process labeled DT.2 Data Cleaning and DT.3 Data Sorting/Compression is where changes will occur from the current processes that exist now. Currently, the processes for data sorting and compression occur is done manually or is too expensive and time consuming for nonprofits. This is where the tool provides a unique value for users, seen in process DT.3 Classification. The TO-BE changes are the generic no-code tool capabilities that can be applied to all use cases. The red circle in Figure 4 outlines that Easy Insights would be able to run the neural network model with personalized tuning parameters nodes, layers, test/train split, and generate an accuracy of the testing data without the user needing to code anything.

5. Mission Requirements

There are 11 total mission requirements broken down into four top level mission requirements - generating a classification model, data formal requirements, data size requirements and tool setup time. The mission requirements are:

- MR.1 Easy Insights shall generate a DLNN model to a desired accuracy.
    - MR.1.1. Easy Insights shall offer hold-out validation parameters options for users.
        - MR.1.1.1 Easy Insights shall offer training to testing percentage breakdowns 90/10, 80/20, and 70/30.
    - MR.1.2 Easy Insights shall offer neural network construction parameter options for users.
        - MR.1.2.1 Easy Insights shall offer a fill in value for the number of nodes.
        - MR.1.2.2 Easy Insights shall offer a fill in value for the number of layers.
        - MR.1.2.3 Easy Insights shall offer a fill in value for the number of epochs.
    - MR.1.3 Easy Insights shall provide instructions describing each input parameter on the interface.
- MR.2 Easy Insights shall read only CSV files.

ISBN: 97819384964-4-8

- MR.3 Easy Insights shall only accept files within 100 KB to 10 MB in size.
- MR.4 Easy Insights shall take less than 30 minutes to set up a trained model.

6. Functional Architecture

There are 11 total functional requirements broken down into three top level requirements. These are load data files, apply user input parameters, and model evaluation metrics. There are five main functions of the decomposed P.2 Classification Model Delivery, and each functional requirement falls under the function that satisfies how the system will deliver a neural network model.

In the decomposed functional overview of the P2 Classification Model Delivery, the functions within this decomposed view show how the neural network takes input data to create an untrained model, then training the model and testing it against the hold out data. The P.2.5 Model Evaluation function happens after the trained model is constructed, tested, and should display an accuracy for how well it can predict data it has never "seen" before. This supports the Hold Out Validation technique because it was split from the raw data to begin with in P.2.2 Split Training Material.

7. Design

The graphical user interface (GUI) is organized by functionality to satisfy all mission and design requirements. The GUI was designed to follow the F-web pattern which support human factoring engineering by leveraging a top to bottom and left to right pattern that human eyes naturally do. The steps for inputting training parameters are all on the left side of the GUI while all output and the accuracy is shown on the right side of the GUI. (See Figure 6) The main window is organized into four widgets: DataUpload, ModelTraining, OutputWindow, and RunModel. The names of these widgets correspond to its functionality. The GUI follows each step in the concept of operations chronologically for what the user must accomplish to use Easy Insights.

7.1 User Steps for Easy Insights Tool

1) Business collects and aggregates data.
   a) Assume the input data is already pre-processed.
2) End user uploads the processed data to Easy Insights as a CSV file.
3) End user inputs the desired parameters using no-code fill-in and pull-down options.
   a) Number of nodes, layers, epochs, Test/Train split (Hold-Out-Data %).
4) Model is trained using the chosen parameters.
5) Easy Insights performs Classification Analysis.
   a) Insert steps to perform classification analysis using DLNN.
6) The trained model contains output data and performance measure.
   a) Accuracy score
   b) Python script available for download.
7) End user is presented with a trained model and may rerun it with different parameters.

ISBN: 97819384964-4-8

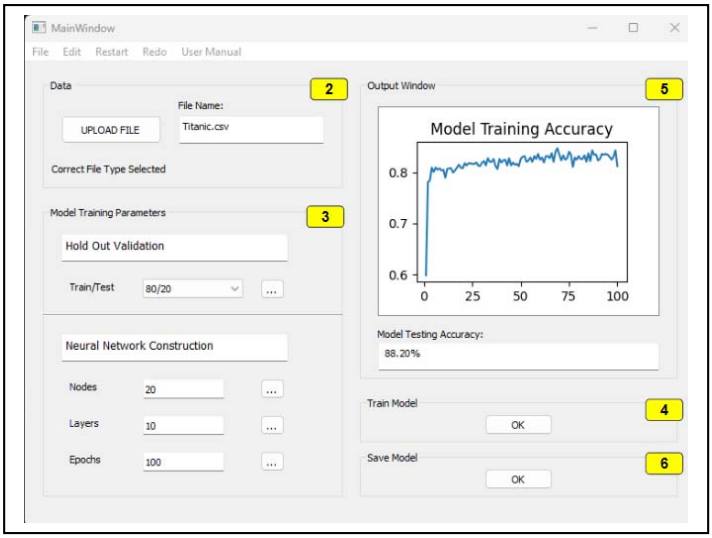

Figure 6. GUI with CONOPS task number label

8. Validation

8.1 Fairfax Fire and Rescue Validation

In the pre-processed data there is 30,839 lines in the CSV file. There are 5211 total fires from 2019-2022. There are 36 total NFRIS codes. There are 30 Unit Types deployed, such as fire trucks, medics, and fire marshals. There are 63 Actions Taken, such as fire control, investigation, and enforce codes. The sum of each unit type and action taken for the NFIRS incident report provided generates 480,000 data points. The task is to compare the final assigned NFIRS code.

A DLNN with inputs of 3 hidden layers with 32 nodes each and 500,000 runs was developed. The DLNN trained with an 80/20 training/testing split for 1000 epochs. The DLNN initially achieved an accuracy of 32%, which was improved to an accuracy of 45% by adjusting the number of epochs and balancing the data by duplicating low occurrence data and removing single occurrence actions taken.

Unfortunately, this type of tabulated data loses context when reducing the data down to a single line for the neural network to process. As a result, the trained model was unable to achieve accuracies above 40%. Unless there is a significant amount of domain knowledge to help pre-process the data, general tabulated data will not work on a neural network training algorithm. This is a limitation found in DLNN training data and machine learning overall.

8.2 GMU Parking Validation

Version 1 of the Easy Insights tool requires pre-processed data. Features of the neural network are Lot Code, Year, Month, Weekday, and Time Code which are the independent variables (X) for the accuracy prediction. The column Avail Code is the data that shall be predicted in the neural network, also called the target. There are 28 total lots coded from 1 to 28 labelled as the Lot Code. The Spots Available column from the raw data was assigned numerical values from 0 to 1240 in power increments of 5 (0,5,10, … 1240). Then each increment grouping was coded from 0 to 41. This is how the Avail Code label was created.

Raw parking data was processed to construct the neural network model. Columns Lot Code, Year, Month, Weekday, and Time Code are the features (independent variables) that will be the X variables for accuracy prediction. The column Avail Code (label) is the data that shall be predicted in the neural network.

There are 28 total lots coded from 1 to 28 (Lot Code). The Spots Available column from the raw data was assigned numerical values from 0 to 1280+ in power increments starting from 0, 5, 10, 20, etc.. Then each group was coded from 0 to 9. This is how the Avail Code label was created. This type of power distribution was chosen because it categorizes the data in a logical way, where a handful of spots available means more in context to each other than hundreds of available spots.

This type of data structure yielded neural network accuracies over 80% when used with a neural network structure of 15 layers of 50 nodes each, over 100 epochs, and trained on an 80/20 training/testing split.

9. Business Plan

Additional case studies show that Easy Insights can reduce time spent on manual processes by 20%, saves $92,500 annually on consulting services, and reduces human errors by 35% per business.

The target market is small to medium sized enterprises (SMEs) and nonprofits in Fairfax County. The target market possesses a large volume of data, approximately more than 50,000 data points, an ideal dataset size to create a neural network, and lacks a current process for data analytics. The end user of Easy Insights is business owners, and the total number of end users is 40,000, estimated from the 10% of small businesses in the US (Doré, 2019). Easy Insights will be distributed as a subscription service to business owners. Price is determined by data set size and company size: $750 per user per year for small companies and $1,500 per user per year for large companies. The small to large company split is 70 to 30.

The total market value is $39 million calculated by the product of price and total possible sales. Given a 10% annual market penetration, the five-year projection for revenue is $51.2 million, profit is $41.6 million, and operating costs is $9.6 million. Operating costs include the five-year salary for three systems engineers, two software engineers, server hosting, and marketing. The ROI is expected to be 425% and break-even in Year 1.

10. References

Awan, A. A. (2019, December 9). *Building Neural Network (NN) models in R*. DataCamp. Retrieved March 21, 2023, from https://www.datacamp.com/tutorial/neural-network-models-r.

Banoula, M. (2023, February 13). *Classification in Machine Learning: What it is & Classification Models*. Simplilearn. Retrieved March 21, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning.

CompTIA. (2022, September). *Business Technology Adoption and Skills Trends*. Retrieved March 21, 2023, from https://www.comptia.org/content/research/2022-international-business-technology-adoption-and-skills-trends.

Di Pietro, M. (2021, December 17). *Deep learning with python: Neural networks (complete tutorial)*. Medium. Retrieved March 21, 2023, from https://towardsdatascience.com/deep-learning-with-python-neural-networks-complete-tutorial-6b53c0b06af0.

Doré, J. (2019, January 30). *Small Businesses Generate 44 Percent Of U.S. Economic Activity*. US Small Business Administration Office of Advocacy . Retrieved March 21, 2023, from https://advocacy.sba.gov/2019/01/30/small-businesses-generate-44-percent-of-u-s-economic-activity/.

GeeksforGeeks. (2020, August 20). *Building a simple neural network in R programming*. Retrieved March 21, 2023, from https://www.geeksforgeeks.org/building-a-simple-neural-network-in-r-programming/.

Indicative. (n.d.). *What Is Classification Analysis?* Retrieved March 21, 2023, from https://www.indicative.com/resource/classification-analysis/.

Pozo Ramos, L. (2022). *Qt designer and python: Build your GUI applications faster*. Real Python. Retrieved March 21, 2023, from https://realpython.com/qt-designer-python/.

Seth, N. (2021, April 26). *Estimation of Neurons and Forward Propagation in Neural Net*. Analytics Vidhya. Retrieved March 21, 2023, from https://www.analyticsvidhya.com/blog/2021/04/estimation-of-neurons-and-forward-propagation-in-neural-net/#:~:text=The%20equation%20for%20the%20neural%20network%20is%20a,denotation%20of%20the%20above%20graphical%20representation%20of%20ANN.

United States Census Bureau. (2022, October 6). *Small Business Pulse Survey*. Retrieved March 21, 2023, from https://portal.census.gov/pulse/data/.