DEFINITIONS

Artificial neural network (ANN): An analytic tool modeled after biologic neural pathways. The typical structure consists of input neurons, a hidden layer of neurons, and output neurons. A neuron, in this sense, simply holds a value of off or on, 0 or 1. Each neuron is triggered by the sum of the weights and biases of the neurons feeding it.

Flight Data Monitoring/Management (FDM): The collection of data from sensors aboard an aircraft. Typically includes aircraft positional, power-plant, and navigation/communication equipment status.

Flight training device (FTD): an approved aviation simulator. These devices accurately depict flight physics and are used to train students in both visual and instrument conditions.

Go around: a common maneuver in aviation where an approach to landing is aborted and the aircraft climbs away from the runway.

Loss of control: a condition where an aircraft is disrupted into an unusual attitude and the pilot may be unable to recover. Frequently the result of abnormal maneuvers or loss of situational awareness.

Stabilized approach: condition where an aircraft is positioned well to land. Identifiable by constant airspeed, descent rate, and pitch. Antonym: unstable approach or unstabilized approach- a condition that may result in aircraft mishap on landing.

INTRODUCTION

Loss of control events are the leading cause of fatal aviation accidents in our region, with most of these accidents occurring during the landing phase of flight [1,2]. A principal contributor to these accidents is failure to recognize an unstable approach [2]. The purpose of this study is to use historic flight data to develop a decision-support tool for recognizing unstable approaches and providing a recommendation to either land or go-around.

LITERATURE REVIEW

The use of neural networks in flight data analysis is somewhat sparse. Artificial neural networks have shown promise in predicting time-series data in studies conducted at University of North Dakota [3]. In these studies, the predictive tool is used for near-term data—predicting aircraft status for a second following a given flight-status.

The development of a real-time device for stabilized approach recognition is ongoing in aviation. A patent search reveals a mechanical device from 1980 attempting to accomplish this task and software from the early 2000s developed by Honeywell to recognize unstable characteristics in flight and alert the pilot [4,5].

The use of a neural network in identifying unstabilized approaches appears novel.

METHODS

Overview:

This project is accomplished at a large scale in two phases and three steps. The first phase is to collect and analyze the flight data. The next phase is to develop a model to predict behaviors seen in the flight data, and to test that model. An outline of the process is seen below.

1) Collect data and identify go-arounds and landings. An assumption is made that go arounds are largely the result of unstabilized approaches, rather than external variables such as traffic conflicts or runway incursions. This top-down process should provide the least biased data for the neural network to develop its definition of an unstable approach.
2) Develop artificial neural network. This is an iterative process, where different topologies will be tested.
3) Develop software. The neural network's structure will be copied into a usable interface for testing.

Data collection and selection:

Data is collected from the Department of Aviation Management and Flight's 5 Garmin G1000 equipped Cessna 172R aircraft. This 4-seat aircraft is ubiquitous in aviation- it frequently serves as a flight trainer and personal passenger aircraft; it is the most common aircraft in the world. Flight data is stored as a CSV at 1hz intervals and includes performance characteristics: airspeed, power settings, rate of climb; positional status: pitch, bank, GPS lat/long; among other variables for a total of 64 data points per second.

The flight data set is enormous, over 10GB of data spanning 5 years. Identification of go-around events within this data was attempted using several methodologies. The first, manually scanning through Google Earth data plots of individual flights proved highly accurate but slow. The next method of go-around identification required identifying flight characteristics common to the go-around condition and using search algorithms to identify those cases. This method proved efficient, but required manual observation to identify and isolate outlier cases (instances that were not go arounds, but instead similar maneuvers). The result of this process is the identification of nearly 300 discrete go-around events.

Next, a random sample of normal-approach landing events is collected. This set consists of approach phases that resulted in normal landings (non-go-arounds). The data set now consists of a snapshot of an aircraft on approach, at 200' above ground level, and the outcome: landing or go around, coded as 1 or 0, respectively. An example of this data is seen below (T1). Note that airspeed, vertical speed, and pitch have been selected as the flight characteristics for this research.

AS	VS	Pitch	Land
61.96	-317	1.816	0
58.7	-567	-1.45	1
64.9	-625	-3.47	1

Table 1. Example data set showing three cases.

The flight data is then normalized to values between 0 and 1. This helps with the development of the neural network by reducing input bias.

Neural network development and testing

To rapidly develop and test neural network structures, Multiple Back-Propagation v2.2.5 was utilized [6]. This software allows for fast configuring and development of neural networks in C—saving extraordinary time in the trial phase of this research.

The data set is randomly subdivided into a training set and a testing set. The training set is used to develop the neural network. In short, the ANN randomly weights the strength of each input variable (airspeed, vertical speed, and pitch) and compares the sum of those products to the output variable (feed-forward). The difference is used to iteratively change the weights and bias (back propagation) until the output from the ANN and the output variable match. The recursive function is the primary subject of change in testing neural network structures, as is the excitation function of each neuron. Figure 1 below schematically represents the ANN structure.

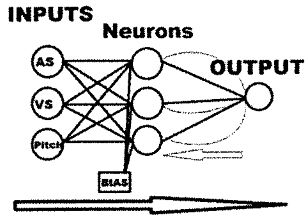

Figure 1. Artificial neural network schematic. Black arrow represents the feed-forward progression from inputs to outputs, the narrow gray arrow represents the back-propagation function used to reduce network error.

After the neural network has achieved a satisfactorily low error rate on the training data, it is tested with a portion of the testing data set. Because it has not been given this data during training, the error rate with this data serves as an excellent measure of the network's utility.

Program development:

A simple program is developed in C. The program uses the weights and biases developed by training the neural network. The program receives manually entered inputs and uses the ANN structure to recommend to the user to go around or land. This program can be used later in near real-time to evaluate the model's predictive ability.

FINDINGS

ANN Structure:

Topology has a profound impact on the run-time of training sets. Pyramid structures (fewer inner neurons than input neurons) run significantly faster. Increasing the number of hidden layers decreased error rate significantly. These observations are unremarkable and expected in terms of neural network development [8]. Neural network training averaged 20,000 iterative cycles on a set of 10-20 samples. Run time on a moderate-spec workstation was less than a minute.

ANN Error Rate:

Training root mean square error (RMSE) values were as low as .0009. This signifies the neural net learned the input/output relationships for that data very well. Accuracy to this degree, however, may indicate overlearning or overfitting of the training set and subject the model to greater interference from noise in testing. Different topologies with a more general fit and lower training RMSEs may function better with real-world noise.

Testing RMSE rates were near .06. At face value, this shows success of the model. However, this is orders of magnitude different than the training data RMSE, indicating a possible overlearning scenario, similar to a force-fit of the neural network [7]. While still a significant value that demonstrates success, it may be desirable for training and testing error rates to be closer.

SUMMARY

The use of neural networks as a predictive tool in go around decision making appears successful. Low error rates indicate the model's ability to recognize and predict trends with historic data. The success of neural networks in predicting behavior within this complex time-series multi-variable data is certain, but further study is necessary before developing any decision aide.

Despite the success of this model, there are some inherent flaws to this methodology that warrant further research. For example, it is unknowable, given the data, as to the true cause of the go around. It is assumed that in most cases it is due to an unstable approach; although in some cases there may be other factors (traffic avoidance, wildlife on runway, etc.). To minimize the error caused by such indeterminate variables requires active experimentation. In theory, the existing neural network may treat these artifacts as noise and their impact on the model itself is minimal, but an overlearned network may give value to this erroneous data if so trained.

A further assumption is that the cases categorized as landings ought to have been landings--it is assumed that all historic landings were satisfactory. This assumption, especially using flight training data, is another inherent weakness. A method to minimize that error is to use a larger data set, or one external to a flight training environment.

Other compounding limitations include the inability of this model to deal with challenging/a-typical scenarios. Heavy and capricious winds, terrain, and field conditions may dictate an augmented approach which requires human judgment to override a stabilized approach. With limited data, it would be difficult to adapt the model to fit each of these scenarios.

FURTHER STUDY

The next step of data validation is to continue testing in two phases. The first phase is to continue using test data derived from ongoing FDM analysis to ensure the results are consistent across a larger sample size. If the current model proves inaccurate, a more general model may be applied. When error rates are consistent and low, the next step is to test the model experimentally.

The experimental testing could be implemented using the department's flight training devices. In this testing, different approaches could be flown by experienced pilots while the decision tool program is used to evaluate the approach. In conditions where a go around is recommended, the pilot will continue the approach despite the recommendation and landing outcome evaluated. The purpose of this study is to test for false-positives in go-around recommendation.

The most hazardous outcome of this model would be to predict falsely that an aircraft can land when it should not. A false-negative such as this could be disastrous. By achieving low error rates with the model and biasing the output towards go-arounds this risk is minimized.

References

1. FAA. (2016, January 5). Fly Safe: Prevent Loss of Control Accidents. Retrieved from https://www.faa.gov/news/updates/?newsId=84506

2. NTSB. (2015). Prevent Loss of Control in Flight in General Aviation. Retrieved from https://www.ntsb.gov/safety/mwl/Documents/MWL_2015_Factsheet_07.pdf

3. Desell T., Clachar S., Higgins J., Wild B. (2014) Evolving Neural Network Weights for Time-Series Prediction of General Aviation Flight Data. In: Bartz-Beielstein T., Branke J., Filipič B., Smith J. (eds) Parallel Problem Solving from Nature – PPSN XIII. PPSN 2014. Lecture Notes in Computer Science, vol 8672. Springer, Cham

4. Rein-weston, Karl J. Automatic Approach Landing and Go-around Control System for Aircraft. The Boeing Company, assignee. Patent US 4319219 A. 30 June 1980. https://www.google.com/patents/US4319219

5. Method, Apparatus and Computer Program Product for Unstabilized Approach Alerting. Yasuo Ishihara, assignee. Patent US 20010052562 A1. 20 Dec. 2001. https://www.google.com/patents/US7068187

6. Multi-back propagation software.2016. Noel de Jesus Mendoca Lopes. http://mbp.sourceforge.net.

7. Lessons in Neural Network Training: Overfitting May be Harder than Expected, Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, Menlo Park, California, pp. 540–545, 1997.

8. Lawrence, S. (1996).What size neural network gives optimal generalization and convergence properties of back propagation. Institute of Advanced Computer Studies, University of Maryland. Available online at url: https://clgiles.ist.psu.edu/papers/UMD-CS-TR-3617.what.size.neural.net.to.use.pdf