Telescope Array FD Weather Classification using Machine Learning

Greg Furlich[*] **for the Telescope Array Collaboration**[†]

High Energy Astrophysics Institute and Department of Physics and Astronomy, University of Utah, Salt Lake City, UT, USA

E-mail: greg.furlich@cosmic.utah.edu

Telescope Array (TA) cosmic ray observatory fluorescence detector (FD) sites have passed 10 years of operation. In order to select nights for further analysis, clear nights should be distinguished from cloudy or noisy nights in the FD data. Weather observations during the night are recorded every hour while the FDs are collecting data by on-site operators at the Middle Drum site. However a more robust and uniform weather classification method is desired for flagging bad weather in the FD data. A series of snapshots of the night sky was created using the detector's photomultiplier tubes (PMTs) pedestals. We classified the night's weather using the PMTs pedestal snapshots using a Recurrent Convolution Neural Network (RCNN).

36th International Cosmic Ray Conference -ICRC2019-
July 24th - August 1st, 2019
Madison, WI, U.S.A.

---

[*]Speaker.

[†]for collaboration list see PoS(ICRC2019)1177

1. Introduction

The Telescope Array (TA) cosmic ray observatory located in Millard County, Utah is the largest cosmic ray observatory in the northern hemisphere. It operates 507 surface detectors and 3 Fluorescence Detector (FDs) sites , Black Rock (BR), Long Ridge (LR), and Middle Drum (MD) to detect ultra high energy cosmic rays (UHECR). The FDs operate on clear, moonless nights to best observe the cosmic ray Extensive Air Shower that excites the Nitrogen in the atmosphere. However, sometimes the detector operates when the night is cloudy and this effects the scattering fluorescence light in the atmosphere diminishing our ability to properly reconstruct or simulate the cosmic ray event. In order to flag and remove cloudy weather from the FD data, neural networks were trained on snapshots of the night sky created using the FADC pedestals of each PMT at BR. Starting with simple neural networks and building up complexity, we were able to achieve high accuracy of weather classification and by classifying each part allows for better time resolution of the operation night's weather progression.

2. FADC Minute Snapshots of the Night Sky

Black Rock and Long Ridge FD sites at TA were built using Fast Analog Digital Converters (FADCs)[1]. Every minute of operation, the pedestal average of each PMT is readout and stored. FD data is collected in data parts for every 1000 triggers of the detectors and the pedestal values are stored corresponding to each part. Using these pedestal averages and the PMT's configuration of nominal pointing on the nights sky, a snapshot, such as Figure 1a, can be produced. To enhance the moving objects in the snapshots, the subsequent frames were differenced from the proceeding frame. For each data part, these snapshots can be used as a frame in a temporal sequence to animate the night sky progression. In these animations we see stars move across the field of view (FOV) or clouds obscuring them.

## 3. Weather Classification Using Machine Learning

The current method of weather classification at TA is operators at MD going out every hour to check the weather visually. However this introduces nonuniformity with different operators and the weather can change more quickly than an hour, thus a more robust method is desired. Using the FADC pedestal animations of each FD data part, a neural network was trained to classify the weather. The snapshots were fed into the neural network with their associated weather label, the neural network adjusted it's weights over each training epoch to best match the mapping of the inputs to the desired outputs by optimizing it's cost function[2][3]. These neural networks were constucted using the Keras machine learning framework[4] and run on a University of Utah Center for High Performance Computing GPU Node.

3.1 Training Data

Twenty percent of the FD data parts were sampled and selected uniformly throughout all operation to reduce bias from seasonal effects or other temporal effects to the FDs. These sampled data parts were classified by eye using the snapshot animation with the breakdown of classes in Figure 2.These snapshots are then input through the neural network for weather classification. The FD data parts were classified into three categories; clear if a good amount of stars were going across the FOV (Figure 1a), cloudy if flowing nebulous shapes obstructed stars across the FOV (Figure

1
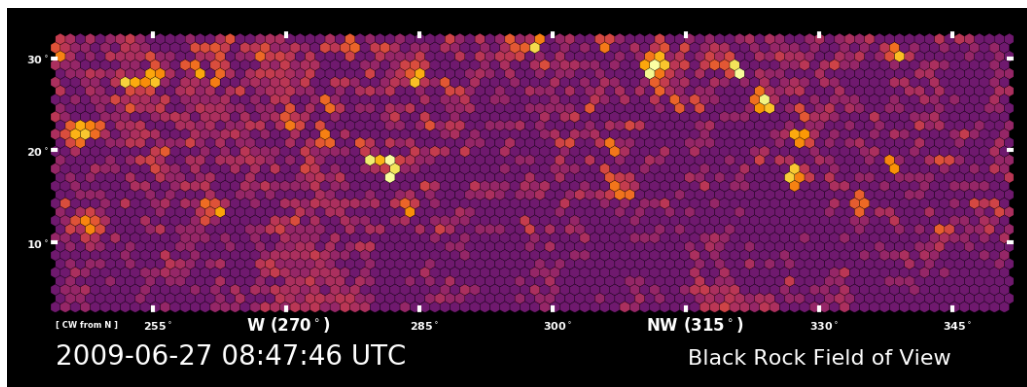
(a) Clear data part
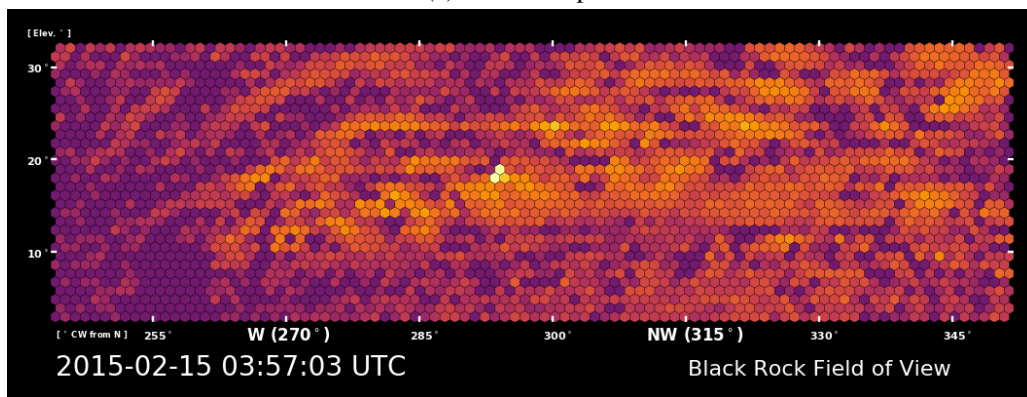
(b) Cloudy data part
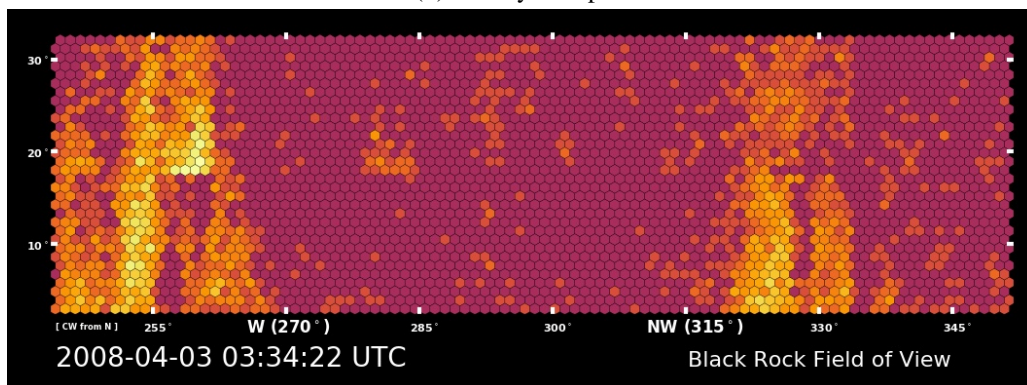
(c) Noisy Data Part

Figure 1: Weather classes distinguished in training set to be learned by neural network.

1b), and noisy if the instruments were not behaving properly and no apparent shapes or points could be identified (Figure 1c).

The training set was further split into two-thirds and a third for the third to be used as a validation set for the neural network to be tested with after training on the remaining two-thirds to confirm if generalization had been achieved. The validation training set does not adjust the neural network weights at all and is a good metric to see how well the neural network preforms on data it has not seen before.
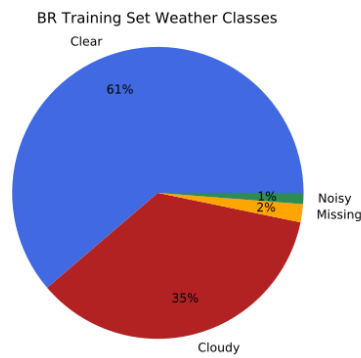
Figure 2: Weather classes breakdown of training set

3.2 Deep Neural Network (DNN)

The first attempt to classify the snapshots was using a simple deep neural network (DNN) which is useful for learning general patterns in a data set. The input for the DNN was the last frame of each data part flattened into a one dimensional array where each PMT pedestal corresponded to a input node. The input was processed through three hidden layers of densely connected nodes and to three outputs corresponding to each of the three weather classes.
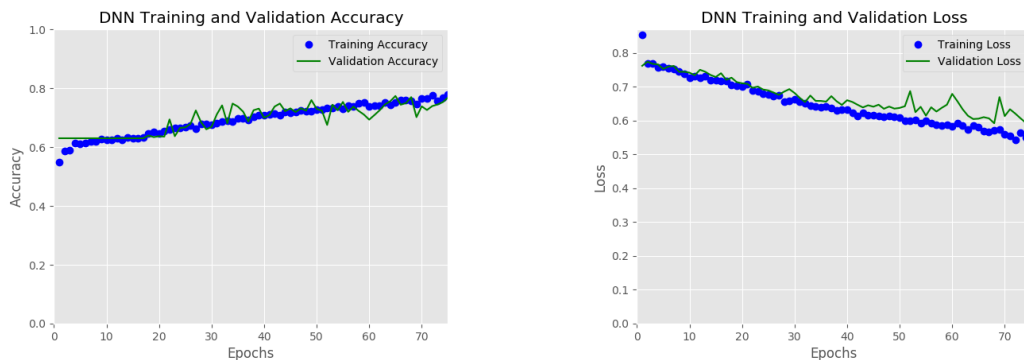
(a) DNN accuracy after each training epoch.

(b) DNN cross entropy after each training epoch.
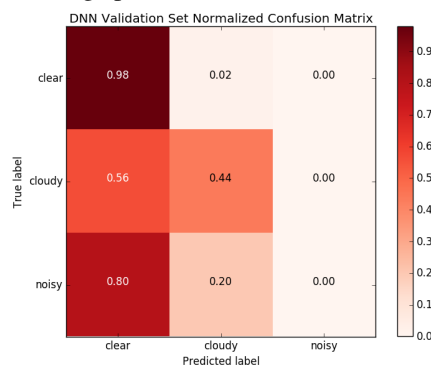
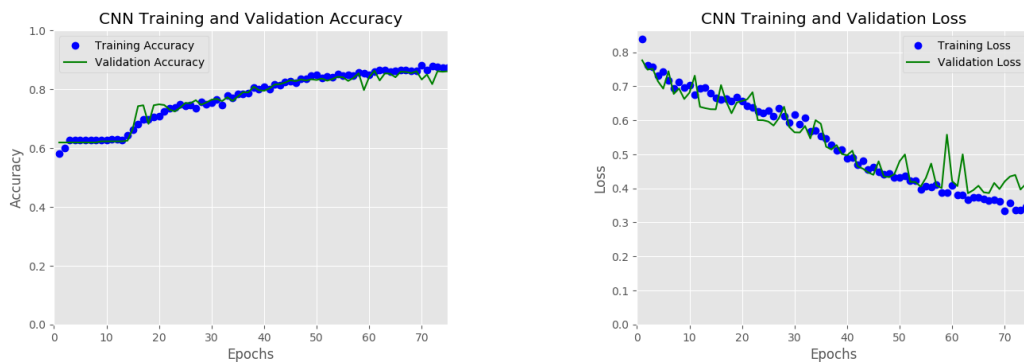(c) DNN normalized validation set confusion matrix

Figure 3: DNN performance

The DNN showed capacity for learning as it's accuracy increased in Figure 3a and cross entropy decreased in Figure 3b. Cross entropy is a metric for comparing the distributions of the neural

network predicted labels and human classified labels. By minimizing cross entropy, the likelihood that the two distributions are indistinguishable is maximized. The validation data confusion matrix shows how the neural network predicted weather classes compare to the true labels classified by a human. In Figure 3c the validation data confusion matrix showed the DNN predicted most of the data parts were clear while they were labeled differently by a human. This resulted in poor classification due to the limitation of the inputs being one dimensional with a flattened array of each PMT pedestal value for the last frame of the data part. The DNN could not take advantage of all the features of the data.

3.3 Convolution Neural Network (CNN)

The next neural network used was a convolution neural network (CNN) which is useful for learning patterns from a spatially correlated data set. A two dimensional image can be input for this neural network and a series of convolution filters are run over the input image to discern key features in the image. Convolutions are used in image processing to distinguish features whereas in a CNN, the neural network is learning which image features are important and weights them accordingly. The last frame of each part was input through three convolution layers followed by three densely connected layers to three output nodes.

(a) CNN accuracy after each training epoch.

(b) CNN cross entropy after each training epoch.
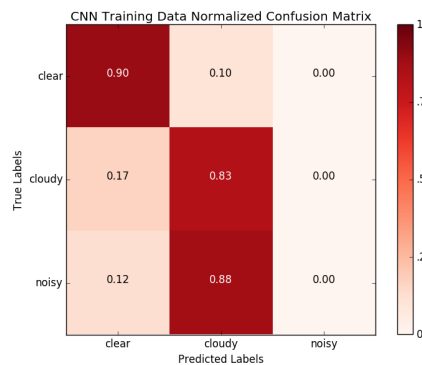
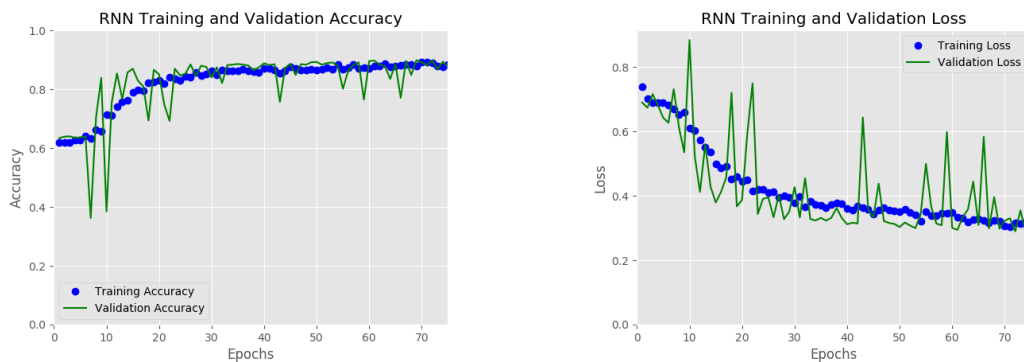(c) CNN normalized validation set confusion matrix

Figure 4: CNN performance

The CNN showed better learning ability in the higher accuracy and lower cross entropy in Figures 4a and 4b. In In Figure 4c the validation data confusion matrix, the CNN started to distinguish between clear and not clear but was not yet achieving the desired diagonalization of generalization.

The neural network was again limited to the last frame of each part but could recognize spatial features but could be improved upon by including all snapshots of a data part.

3.4 Recurrent Neural Network (RNN)

To include the temporal feature of the input data being a time series of snapshots, a recurrent neural network (RNN) was constructed. A RNN is useful for learning patterns from sequentially correlated inputs. It retains a state or memory as sequential data is feed through and the RNN tries to optimize towards the desired classification. However the sequential data is limited to a one dimensional array like the DNN, but all the frames of the data part were flattened and fed through for learning. The flattened array of all frames in a part were input through three Long Short Term Memory (LSTM) layers to three output nodes.

(a) RNN accuracy after each training epoch.
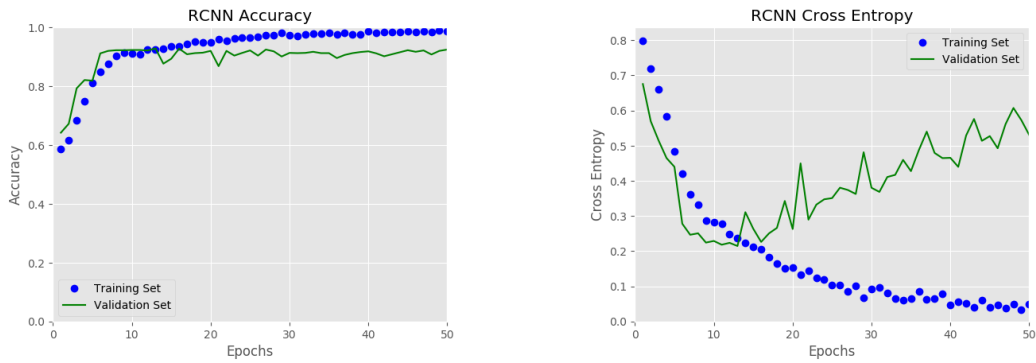

(b) RNN cross entropy after each training epoch.
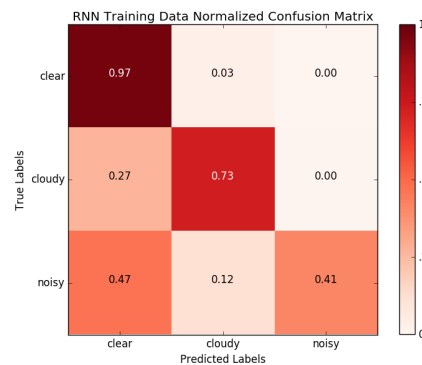
(c) RNN normalized validation set confusion matrix
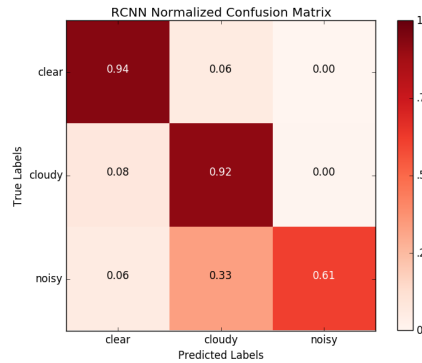
Figure 5: RNN performance

Including the sequential reasoning by constructing a RNN lead to better results in the accuracy and cross entropy (Figures 5a and 5b). The validation data confusion matrix in Figure 5c shows further diagonalization where a majority of predicted classes matched their true labels.

3.5 Recurrent Convolution Neural Network (RCNN)

We combined the features of the CNN and RNN to allow for spatial and temporal reasoning to take full advantage of the features of the training data set. The neural network can read in frame by frame and learn temporal and spatial patterns.

(a) RCNN accuracy after each training epoch.

(b) RCNN cross entropy after each training epoch.

(c) RCNN normalized validation set confusion matrix

Figure 6: RCNN performance

The RCNN has good training accuracy and cross entropy but now there is divergence of the validation and training data in Figures 6a and 6b which shows the model over-training and pre-forming well on the training set but not generalizing to the unseen validation data. However the validation confusion matrix(Figure 6c) shows good diagonalization which means if the issue of over-training can be minimized, the model will preform well and can be used as a good method for classifying weather.

3.6 Model Comparison

Model	Training Epochs	Optimizer	Comp. Time	Val. Accuracy	Val. Cross Entropy
DNN	75	Adadelta	0:21:49 hours	76.79 %	.58
CNN	75	Adadelta	0:09:06 hours	86.09 %	.42
RNN	75	Adadelta	1:06:02 hours	87.65 %	.35
RCNN	50	Adadelta	2:06:55 hours	92.53 %	.52

Table 1: Comparing the performance of each neural network model

The RCNN takes full advantage of the FADC pedestal snapshot animation of each data part to learn to classify weather. However it is more computationally expensive as it takes much longer to iterate through it's training epochs.

4. Summary and Plans

A uniform method to classify weather during FD operation was developed using neural networks. A RCNN was trained on FADC pedestal snapshots which illustrate the night sky in the FD's FOV. The RCNN was shown to preform well but with an issue of over-training. Minimizing the over-training would allow for robust weather classification at TA. This weather classification could be used to enhance further analyses such as a FD Monocular Energy Spectrum[5] or FD Stereo Composition Analysis.

5. Acknowledgments

References

[1] H. Tokuno *et al.* for the Telescope Array Collaboration, Nuclear Instruments A 676, 2012

[2] Chollet, François, Deep Learning with Python, Manning Publishing Co., 2018

[3] Chrisptopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

[4] Chollet, François *et al.*, Keras, 2015, *https://keras.io*

[5] G. Furlich and D.R. Bergman for the Telescope Array Collaboration, these proceedings.