❮ WOMANIUM | QUANTUM ❯

Comparison of Tornado Damage Prediction Accuracy between Classical and Quantum Neural Networks

Jorge O. Cedeño, Omar Alsaid Sulaiman,
Sirikarn Phuangthong, Suhani Sundar

Problem Statement: Tornado Damage Prediction

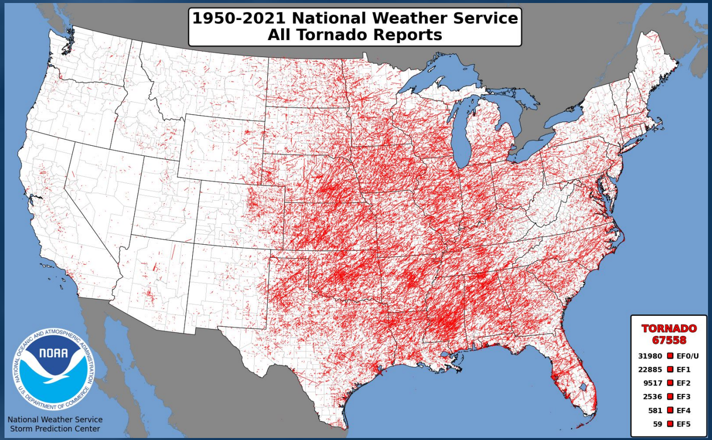
- Tornadoes are among the deadliest natural disaster worldwide
    - Approximately 1,000 tornadoes occur annually in the United States, with an average of 80 deaths, 1,500 injuries, and significant architectural damages
- Advancements in tornado prediction using machine learning is one of the most effective methods
    - CSU-MLP by Colorado State University [1]
    - Deep CNN by McGuire and Moore [2]
- To enhance model accuracy, we want to implement quantum computing into ML model
    - Reduce computational cost for analyzing large dataset
    - Investigate how quantum properties (such as superposition and entanglements) benefit complex data analysis

https://www.electronic-sirens.com/the-most-twisted-tornadoes-in-united-states/

[1] Hill, A. J., R. S. Schumacher, and I. L. Jirak. A New Paradigm for Medium-Range Severe Weather Forecasts: Probabilistic Random Forest–Based Predictions. Wea. Forecasting, 38, 251–272, https://doi.org/10.1175/WAF-D-22-0143.1, 2023.
[2]McGuire, M. P., and T. W. Moore. Prediction of tornado days in the United States with deep convolutional neural networks. Computers & Geosciences, 159, 104990, https://doi.org/10.1016/j.cageo.2021.104990, 2022.

Methodology & Project Solution

- Data Management
  - Obtain data from Storm Prediction Centre (SPC) [1]
  - Analysing and cleaning the data
  - Data pre-processing and encoding
- Creating multi-output classical and quantum neural networks
  - Create neural network model using TensorFlow and Keras
  - Implement Quantum Computing properties to the NN model using TensorFlow-Quantum
- Evaluating the models and updating to enhance performance
- Comparing the two models and identifying the better option based on its accuracy

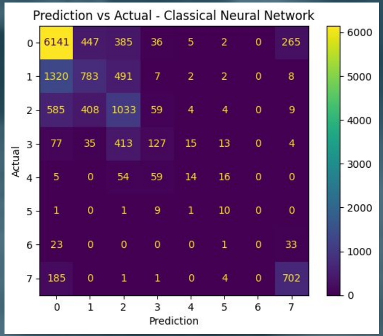


https://www.spc.noaa.gov/gis/svrgis/

[1] https://www.spc.noaa.gov/gis/svrgis/

Results: Classical Neural Network


```
Classification report:
              precision    recall  f1-score   support

         0.0       0.74      0.84      0.79      7281
         1.0       0.47      0.30      0.37      2613
         2.0       0.43      0.49      0.46      2102
         3.0       0.43      0.19      0.26       684
         4.0       0.34      0.09      0.15       148
         5.0       0.19      0.45      0.27        22
         6.0       0.00      0.00      0.00        57
         7.0       0.69      0.79      0.73       893

    accuracy                           0.64     13800
   macro avg       0.41      0.39      0.38     13800
weighted avg       0.61      0.64      0.62     13800
```

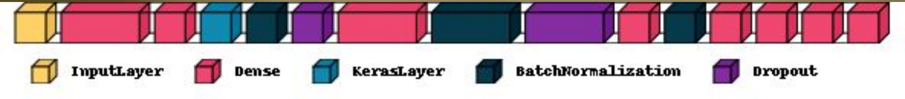



```
fat_mae: 0.1802 - inj_mae: 1.2820 - loss: 6.6499 - loss_accuracy: 0.6360 - ns_mae: 0.0187
```

Layer (type)	Output Shape	Param #	Connected to
input_layer_21 (InputLayer)	(None, 6)	0	-
dense_66 (Dense)	(None, 512)	3,584	input_layer_21[0][0]
batch_normalization_66 (BatchNormalization)	(None, 512)	2,048	dense_66[0][0]
dropout_45 (Dropout)	(None, 512)	0	batch_normalization_6…
dense_67 (Dense)	(None, 256)	131,328	dropout_45[0][0]
batch_normalization_67 (BatchNormalization)	(None, 256)	1,024	dense_67[0][0]
dropout_46 (Dropout)	(None, 256)	0	batch_normalization_6…
dense_68 (Dense)	(None, 64)	16,448	dropout_46[0][0]
batch_normalization_68 (BatchNormalization)	(None, 64)	256	dense_68[0][0]
ns (Dense)	(None, 1)	65	batch_normalization_6…
fat (Dense)	(None, 1)	65	batch_normalization_6…
inj (Dense)	(None, 1)	65	batch_normalization_6…
loss (Dense)	(None, 8)	520	batch_normalization_6…

Total params: 155,403 (607.04 KB)
Trainable params: 153,739 (600.54 KB)
Non-trainable params: 1,664 (6.50 KB)

Results: Hybrid Neural Network



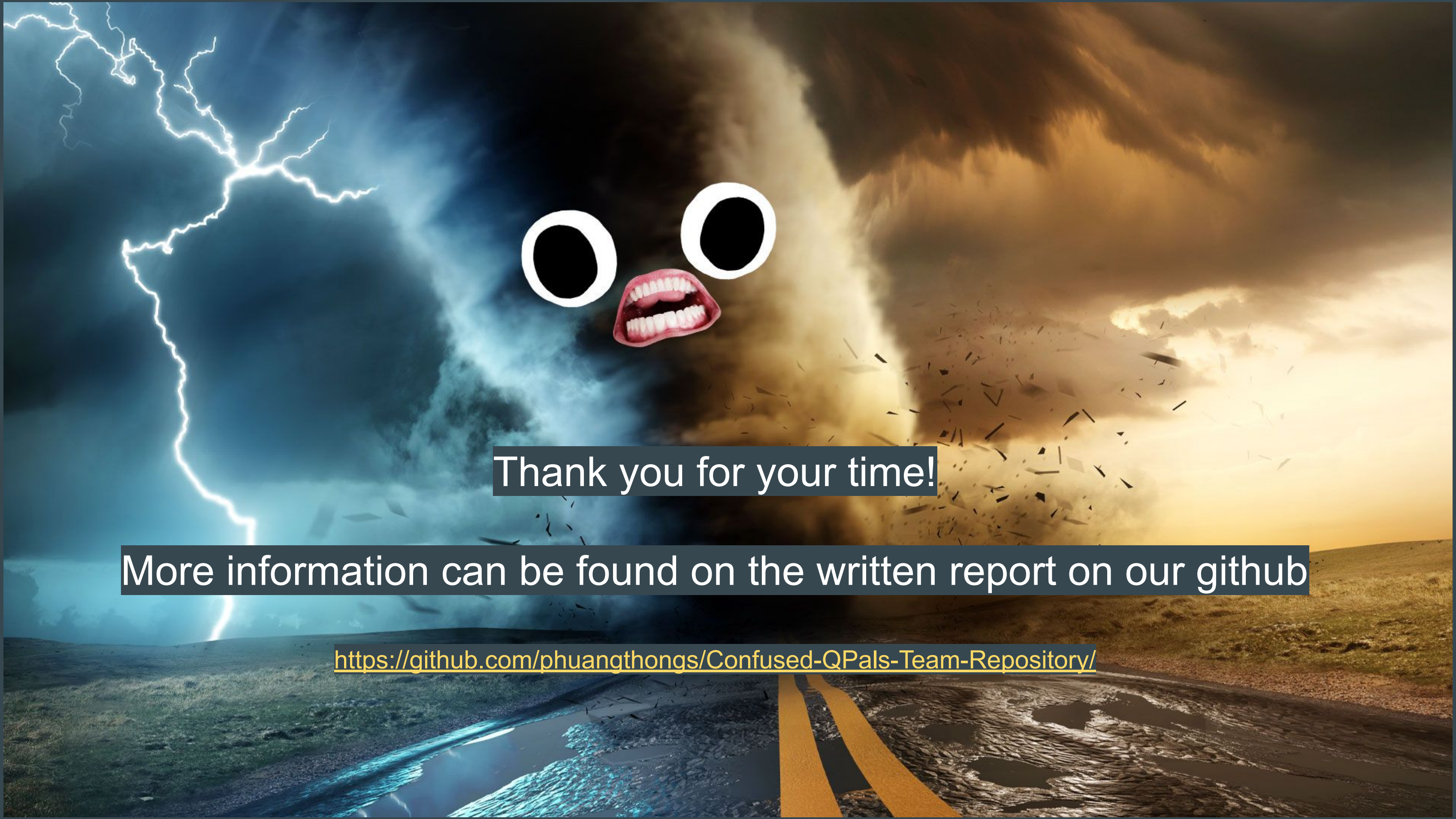
- The hybrid (quantum+classical) neural network has performed better than its classical counterpart
- The loss accuracy reached was .67 during the 4th epoch
- Unfortunately we were unable to capture the performance due to frequent runtime session issues (running out of computation units, etc...)

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 59)]         0           []

 dense (Dense)                  (None, 512)          30720       ['input_1[0][0]']

 dense_1 (Dense)                (None, 2)            1026        ['dense[0][0]']

 keras_layer (KerasLayer)       (None, 2)            4           ['dense_1[0][0]']

 batch_normalization (BatchNorm  (None, 2)           8           ['keras_layer[0][0]']
 alization)

 dropout (Dropout)              (None, 2)            0           ['batch_normalization[0][0]']

 dense_2 (Dense)                (None, 512)          1536        ['dropout[0][0]']

 batch_normalization_1 (BatchNo  (None, 512)         2048        ['dense_2[0][0]']
 rmalization)

 dropout_1 (Dropout)            (None, 512)          0           ['batch_normalization_1[0][0]']

 dense_3 (Dense)                (None, 64)           32832       ['dropout_1[0][0]']

 batch_normalization_2 (BatchNo  (None, 64)          256         ['dense_3[0][0]']
 rmalization)

 ns (Dense)                     (None, 1)            65          ['batch_normalization_2[0][0]']

 fat (Dense)                    (None, 1)            65          ['batch_normalization_2[0][0]']

 inj (Dense)                    (None, 1)            65          ['batch_normalization_2[0][0]']

 loss (Dense)                   (None, 4)            260         ['batch_normalization_2[0][0]']

==================================================================================================
Total params: 68,885
Trainable params: 67,729
Non-trainable params: 1,156
```

Conclusion

- While we successfully created both classical and quantum-classical neural network models, we were unable to make a fair comparisons between the models to investigate if the quantum-hybrid alternative does better than the classical neural network due lack of time and computational costs

- In the future, if the computational cost issue has been addressed, we plan to run the quantum-classical neural network models with 20 epochs and 8 qubits in order to compare both neural network models more fairly



Thank you for your time!

More information can be found on the written report on our github

https://github.com/phuangthongs/Confused-QPals-Team-Repository/

https://www.beano.com/posts/tornado-facts