MODE

Automated Neural Network Model Debugging via State Differential Analysis and Input Selection

REPLICATION / EXTENSION PROJECT

•••

Fabrice Harel-Canada

Roadmap

- Overview
  - Replication
  - Extensions
- Challenges
- Demo
- Results
- Future Work
- Q&A

Replication + Extensions

- Treat the paper like a technical spec...
- Parameterize the similarity function and try all kinds of metrics to see:
  - If they produce intuitively reasonable “superlative” images
  - If they improve performance outcomes
- Analysis of whether target layer selection is really helpful
  - If not, let’s save time and just make heat maps from the final layer’s output every time.

Challenges

- Basic Implementation
  - Learning a lot about Keras and Tensorflow, cloning models, copying weights, etc.
  - Dissimilarities in model performance
- Hyper-parameter ambiguities
- Implementation ambiguities
  - Target Layer Selection
  - Heat maps

Hyperparameter ambiguities

- Bhattacharyya distance
  - Never touched upon in paper
- Theta - underfitting threshold
  - Empirical - decided to set at 0.92
- Gamma - overfitting threshold
  - Empirical - decided to set at 0.10
  - No overfitting encountered in experiments

Target Layer Selection?

- Do we prioritize layer selection for underfitting or overfitting first?
- How often do you select a faulty target layer?
- How do you handle the wide variety of layers when creating feature submodels?
  - Do dropout, pooling, flattening layers count as layers?
  - If a layer is not fully connected, how do we associate it's outputs to the final layer?

Heat maps?

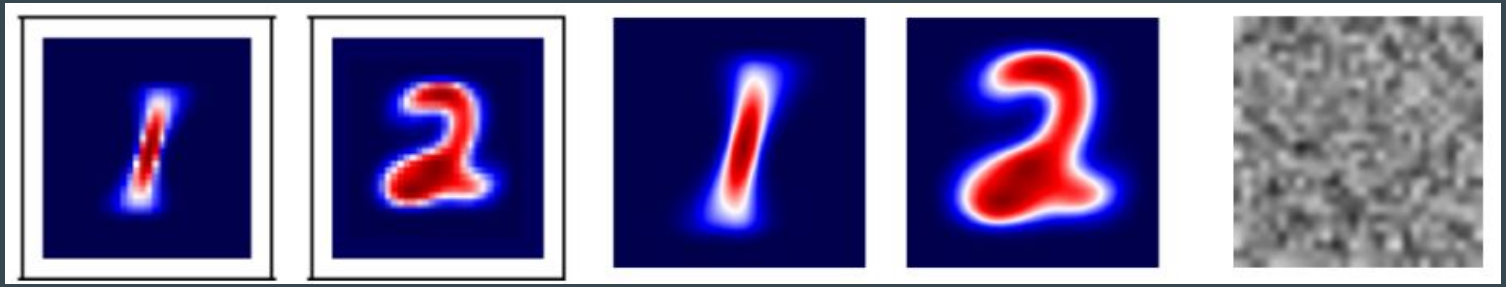
- The paper states that a heat map is:

  “an image whose size equals to the number of neurons and the color of a pixel represents the importance of a neuron.”

- Based on neuron weights?
- How can an arbitrarily sized layer of neurons be used to assess similarity across all potential bug fixing samples?

Heat maps?

- All signs point to averaging of correct / incorrectly classified images as the source of the heat maps.



Their heat maps
“based on neurons”

My heat maps based
on image averaging

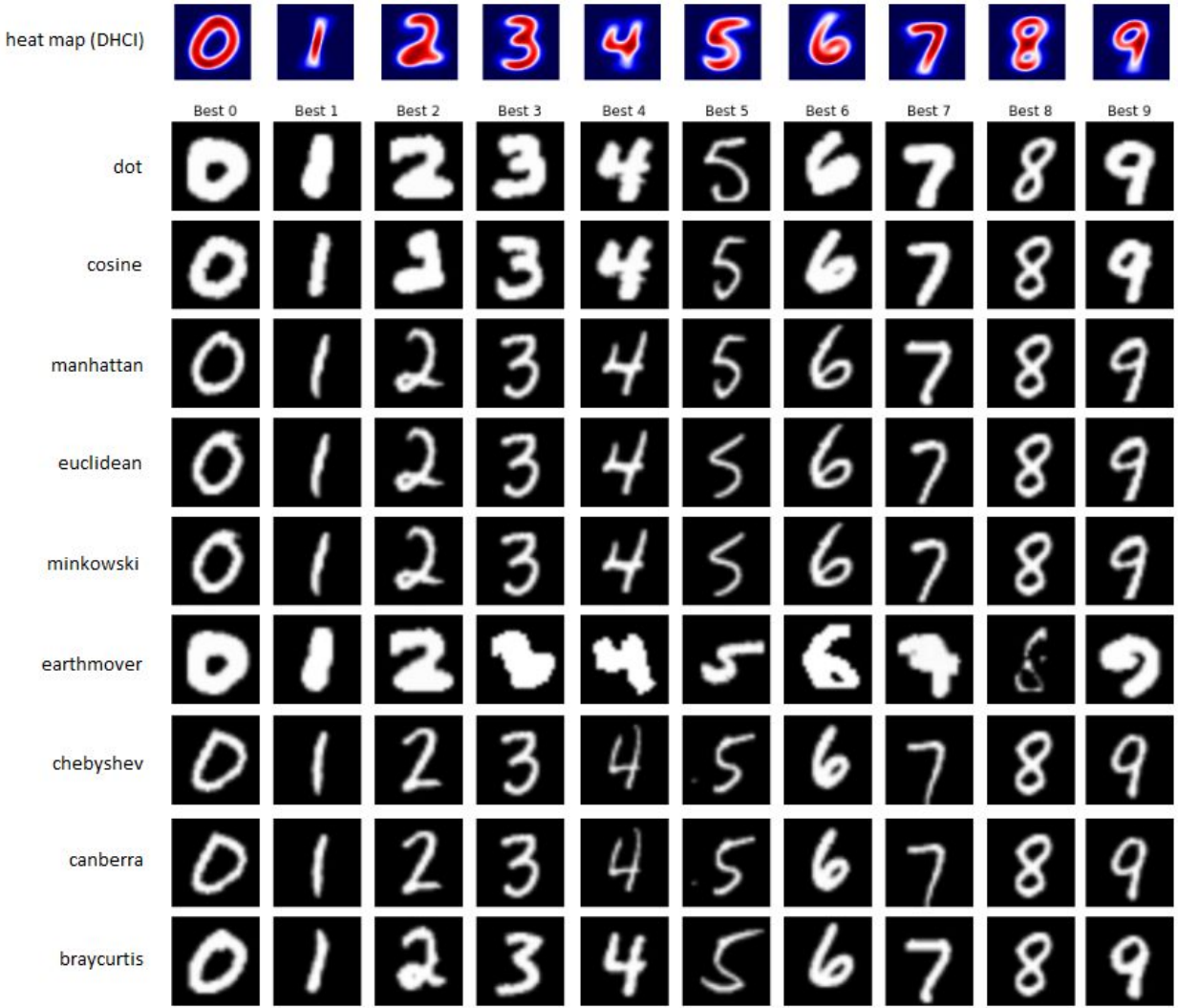
My heat maps based
on neurons

DEMO TIME

Results | Metrics

Most similar per metric

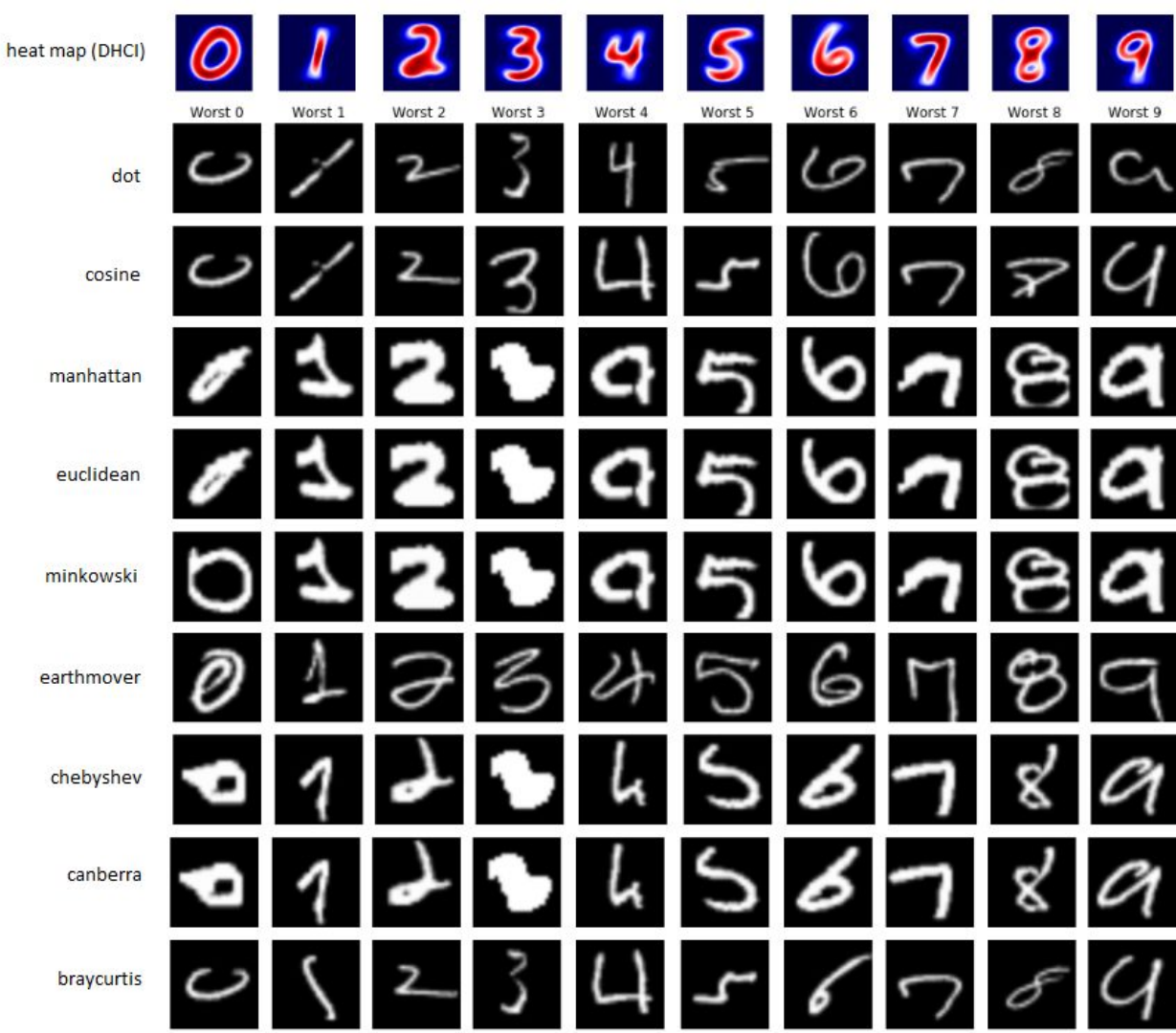
Which one would you think would do the best?



Results | Metrics

Least similar per metric

Does your hypothesis change at all?



Results | Metrics

Metric	Initial Acc	Final Acc	Control Acc	Finished Early	Diff I-F	Diff F-C
dot product	0.9436	0.9438	0.9483	0	0.0002	-0.0045
cosine	0.9481	0.952	0.9481	0	0.0039	0.0039
manhattan	0.9452	0.9356	0.9439	0	-0.0096	-0.0083
euclidean	0.9445	0.9458	0.9438	0	0.0013	0.0020
minkowski	0.945	0.9405	0.9438	0	-0.0045	-0.0033
chebyshev	0.9439	0.9426	0.9439	0	-0.0013	-0.0013
earth mover	0.9468	0.9435	0.9485	1	-0.0033	-0.0050
canberra	0.9502	0.9435	0.9490	0	-0.0067	-0.0055
bray curtis	0.9462	0.9435	0.9434	1	-0.0027	0.0001

- Earth Mover did the best
- Results are not statistically significant.
- Model performance was already high.
- Testing time was significant, limiting reruns

Results | Forgoing Layer Selection

Metric	Initial Acc	Final Acc	Control Acc	Finished Early	Diff I-F	Diff F-C
dot product	0.9436	0.9438	0.9483	0	0.0002	-0.0045
cosine	0.9481	0.952	0.9481	0	0.0039	0.0039
manhattan	0.9452	0.9356	0.9439	0	-0.0096	-0.0083
euclidean	0.9445	0.9458	0.9438	0	0.0013	0.0020
minkowski	0.945	0.9405	0.9438	0	-0.0045	-0.0033
chebyshev	0.9439	0.9426	0.9439	0	-0.0013	-0.0013
earth mover	0.9468	0.9435	0.9485	1	-0.0033	-0.0050
canberra	0.9502	0.9435	0.9490	0	-0.0067	-0.0055
bray curtis	0.9462	0.9435	0.9434	1	-0.0027	0.0001

- No statistically significant difference between having layer selection and not having it.
- Cut it and save the time.

Future Work

- Extending layer selection to handle the nuances of different layer types and their combinations.
- Investigating whether using the least similar images could be used to address overfitting in lieu of using the other heatmaps.
- Investigating the relationship between the number of classes and the ratio of selected-to- random inputs.


QUESTIONS?