RUTGERS
WINLAB | Wireless Information Network Laboratory

Neural Networks For Feature Analysis

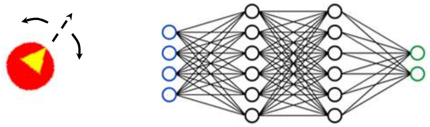
Daksh Khetarpaul, Mayank Barad, Katherine Lew

Overview

- Animal behavioral studies: Hypothesize some response to a stimulus
- Use of statistical methods to test the hypothesis
- **Issue**: Someone needs to guess what the behavior will look like in the first place
- **Solution**: we can use machine learning (ML) to analyze behaviors
- ML can extract features in the behavior that we would not expect
- Key question: **How well can neural networks analyze behavioral response to stimuli?**

Methodology

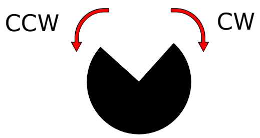
- Design a simulation with randomized behavior
- Example: at each timestep, rotate and walk forward by a certain amount.

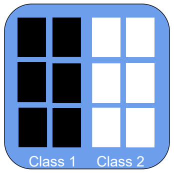


- Train a neural network to differentiate samples that are *biased* or *unbiased* in some chosen feature
  - The biased case represents behavioral response to a stimulus
- The accuracy of the network is an estimate of how well the network learned the behavioral response

Testing The Stack

- We trained the model using simple test cases
- This was to validate that the software stack works as expected
- Two main test cases:
  - Black/white for class differentiation
  - CW/CCW for time-varying features

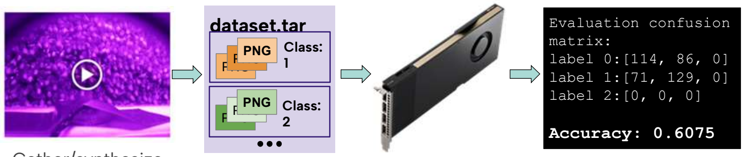




Software

ML stack

- Same one used in Beehive Monitoring; uses PyTorch for ML models
  - https://github.com/bfirner/bee_analysis
- WebDataset tarfiles: storing datasets



```
Evaluation confusion
matrix:
label 0:[114, 86, 0]
label 1:[71, 129, 0]
label 2:[0, 0, 0]

Accuracy: 0.6075
```

Gather/synthesize video frames

Compile frames into WebDataset tarfile

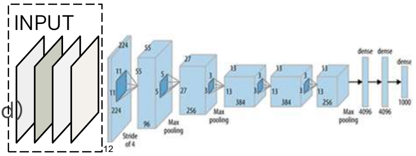
Train model using iLab GPUs

Log the testing dataset accuracy after training

- We generate multiple datasets where each dataset contains a testing and training set
- For each dataset, train a completely new model
- Record the accuracy of each model after

Model:

- Network used: AlexNet
- Input: Sequence of frames
- Output: Either class 1 (unbiased) or class 2 (biased)



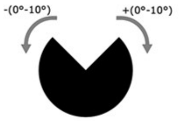
Simulation

- We synthesize our dataset by preparing simulations using the Pygame library
- The simulation is a Pac-Man shaped figure that rotates at each timestep
- We generate a simulation that exhibits an unexpected behaviour
- We use ML on that simulation to see if those behaviours can be detected

Generating the Datasets

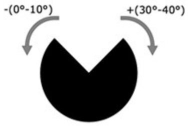
- Each example: 4 frames + class
- 2 classes: **unbiased** and **biased**
- 20k samples per class
- The 1st frame of each sample is randomly reset

Unbiased



After each frame, the Pac-Man will rotate either clockwise or counter-clockwise at any angle in (0°,10°).

Biased (30° used as an example)

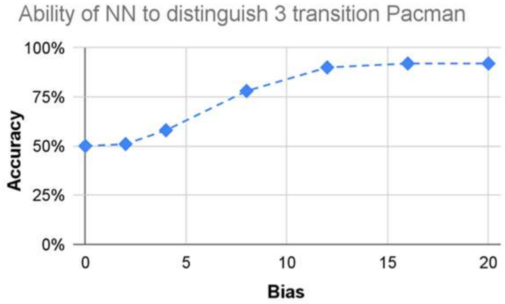


Same as random case, except in the case of turning right, add an extra 30° of rotation.

This work was supported in part by the NSF REU program and the donation from nVERSES CAPITAL

Results

We obtain the accuracy of the model for bias values ranging from 0 to 20 degrees.



- As expected, the accuracy starts out at 50% (the biased and unbiased sets are the same)
- We see a nonlinear relationship between bias and accuracy
- This shows that machine learning can detect the unexpected behaviour

Future Work

- Diversify the simulations and features for further testing
- Use saliency graphs to analyze the features our model picked up
- Use a more universal metric of bias(eg: entropy)

Use this link to view our website and current progress ➡

Acknowledgements

We would like to thank our advisors **Dr. Richard Howard** and **Dr. Richard Martin** for their guidance. We would also like to thank **Dr. Bernhard Firner** for the machine learning software stack.

WINLAB