Testing A Moving Target: How Do We Test Machine Learning Systems?

Peter Varhol
Technology Strategy Research, USA
Testing a Moving Target

How Do We Test Machine Learning Systems?

Peter Varhol, Technology Strategy Research
About Me

• International speaker and writer
• Degrees in Math, CS, Psychology
• Evangelist at Dynatrace
• Former university professor, tech journalist
Gerie Owen

- Test Manager, Tester and as such experienced bug finder and bug misser
- Subject expert on testing for TechTarget’s SearchSoftwareQuality.com
- International and Domestic Conference Presenter
- Marathon Runner & Running Coach
- Cat Mom

www.gerieowen.com
gerie.owen@gerieowen.com
What You Will Learn

• What kind of systems produce nondeterministic results
• Why we can’t test these systems using traditional techniques
• How we can assess, measure, and communicate quality with learning and adaptive systems
Agenda

• What are machine learning and adaptive systems?
• How are these systems evaluated?
• Challenges in testing these systems
• What constitutes a bug?
• Summary and conclusions
We Think We Know Testing

• We test deterministic systems
  • For a given input, the output is always the same
  • And we know what the output is supposed to be
  • If the output is something else
  • We may have a bug

• We know nothing
Machine Learning and Adaptive Systems

• We are now building a different kind of software
  • It never returns the same result
  • That doesn’t make it wrong
• How can we assess the quality?
  • How do we know if there is a bug?
How Does This Happen?

- The problem domain is ambiguous
- There is no single “right” answer
  - “Close enough” is good
- We don’t know quite why the software responds as it does
  - We can’t easily trace code paths
What Technologies Are Involved?

• Neural networks
• Genetic algorithms
• Rules engines
• Feedback mechanisms
• Sometimes hardware
Neural Networks

• Set of layered algorithms whose variables can be adjusted via a learning process
• The learning process involves training with known inputs and outputs
• The algorithms adjust coefficients to converge on the correct answer (or not)
• You freeze the algorithms and coefficients, and deploy
A Sample Neural Network
Genetic Algorithms

- Use the principle of natural selection
- Create a range of possible solutions
- Try out each of them
- Choose and combine two of the better alternatives
- Rinse and repeat as necessary
Rules Engines

- Layers of if-then rules, with likelihoods associated
- With complex inputs, the results can be different
- Determining what rules/probabilities should be changed is almost impossible
- How do we measure quality?
How Are These Systems Used?

• Transportation
  • Self-driving cars
  • Aircraft

• Ecommerce
  • Recommendation engines

• Finance
  • Stock trading systems
A Practical Example

• Electric wind sensor
  • Determines wind speed and direction
  • Based on the cooling of filaments
• Several hundred data points of known results
• Designed a three-layer neural network
  • Then used the known data to train it
Another Practical Example

- Retail recommendation engines
  - Other people bought this
  - You may also be interested in that
- They don’t have to be perfect
  - But they can bring in additional revenue
Challenges to Validating Requirements

• What does it mean to be correct?
  • The result will be different every time
  • There is no one single right answer

• How will this really work in production?
• How do I test it at all?
Possible Answers

• Only look at outputs for given inputs
  • And set accuracy parameters

• Don’t look at the outputs at all
  • Focus on performance/usability/other features
  • We can’t test accuracy

• Throw up our hands and go home
Testing Machine Learning Systems

• Have objective acceptance criteria
• Test with new data
• Don’t count on all results being accurate
• Understand the architecture of the network as a part of the testing process
• Communicate the level of confidence you have in the results to management and users
What About Adaptive Systems?

• Adaptive systems are very similar to machine learning
  • The problems solved are slightly different
  • Neural algorithms are used, and trained
  • But the algorithms aren’t frozen in production
Machine Learning and Adaptive Systems

• These are two different things
  • Machine learning systems get training, but are static after deployment
  • Adaptive systems continue to adapt in production
    • They dynamically optimize
    • They require feedback
Adaptive Systems

- **Airline pricing**
  - Ticket prices change three times a day based on demand
  - It can cost less to go farther
  - It can cost less later

- **Ecommerce systems**
  - Recommendations try to discern what else you might want
  - Can I incentivize you to fill up the plane?
Recommendation Engines Can Be Very Wrong

• Brooks Ghost running shoes
  • Versus ghost costumes
  • We don’t take context into account
• But do they make money?
  • Well, probably
Considerations for Testing Adaptive Systems

• You need test scenarios
  • Best case, average case, and worst case
• You will not reach mathematical optimization
  • Determine what level of outcomes are acceptable for each scenario
• Defects will be reflected in the inability of the model to achieve goals
What Does Being Correct Mean?

• Are we making money?
  • Is the adaptive system more efficient?
• Are recommendations being picked up?
  • Is it worthwhile to test recommendations?
  • How would you score that?
These Are Very Different Measures

• We have never tested these characteristics before
  • Can we learn?
• How to we make quality recommendations?
  • Consistency?
  • Value?
• Does it matter?
Objections

• I will never encounter this type of application!
  • You might be surprised

• I will do what I’ve always done
  • Um, no you won’t

• My goals will be defined by others
  • Unless they’re not

• You may be the one
How Do We Test These Things?

• Multiple inputs at one time
  • Inputs may be ambiguous or approximate

• The output may be different each time
  • Testing accuracy is a fool’s game

• Past data
  • We know how different pricing strategies turned out
  • We made recommendations in the past
What is a Bug?

• A mismatch between inputs and outputs?
  • It supposed to be that way!
  • Not every recommendation will be a good one
  • But that doesn’t mean it’s a bug
• Too many wrong answers
  • Define too many
We Found a Bug, Now What?

• The bug could be unrelated to the neural network
  • Treat it as a normal bug

• If the neural network is involved
  • Determine a definition of inaccurate
  • Determine the likelihood of an inaccurate answer
  • This may involve serious redevelopment
Conclusions

• We have little experience with learning and adaptive systems
  • Requirements have to be very different
• We need to understand the difference between correct and accurate
• We need objective requirements
  • And the ability to measure them
  • And the ability to communicate what they mean
Thank You

Peter Varhol
Dynatrace LLC
peter.varhol@Dynatrace.com