The What-If Tool (WIT)
Interactive Probing of Machine Learning Models

James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viegas, and Jimbo Wilson
Presented on Nov 19, by Patrick Huber

Why? - Initial Analysis
Proof-of-concept
Evaluate technical suitability and compatibility of InfoVis solution
Workshop
2 usability studies at different scales and with different user groups
Application builds on insights from usability studies
Authors derive 5 distinct user needs

Why? - User Needs
Need 1: Test multiple hypotheses with minimal code
Interact with trained model through graphical interface (no code)
Comprehend relationships between data and models

Need 2: Use visualizations as a medium for model understanding
Generate explanations for model behavior
Problem: Visual complexity, hard to find meaningful insights
Solution: Provide multiple, complementary visualizations

Need 3: Test hypotheticals without having access to the inner workings of a model
Trust models as black boxes
Generate explanations for end-to-end model behavior
Solution: Interactive visual "what-if" exploration

Need 4: Conduct exploratory intersectional analysis of model performance
Users often interested in subsets of data on which models perform unexpectedly
False positive and false negative rates can be wildly different
No access to model internals
Solution: Interactive visual "what-if" exploration

Need 5: Evaluate potential performance improvements for multiple models
Track impact of changes in model hyperparameters (e.g., changing a threshold)
Interactively debug model performance by testing strategies

What? - The Tool
Build using Tensorboard, a code-free and installation-free visualization framework
No custom coding (N1)
Help developers and practitioners to understand ML systems
Covers many standpoints (inputs / single data points / models)
Basic layout: 2 main panels → control panel & visualization panel
https://pair-code.github.io/what-if-tool/iris.html

How? - Tailoring 3 Tasks to Satisfy User Needs
Closely related to user needs
Example of the UCI Census dataset
Solve prediction task
Closely related to user needs
Train 2 models
Multi-layer neural network
Simple linear classifier

How? - Task 1: Exploring the Data
Customizable Analysis
Generate & test hypotheses about how model treats data
Edit data points
Identify counterfactuals
Observe partial dependencies
Apply carefully chosen input modifications (edit, add or delete feature values)
Result of changing income from $3,000 to $20,000 (edit data point)

Why? - Task 2: Investigating What-If Hypothesis
Generate & test hypotheses about how model treats data
Edit data points
Identify counterfactuals
Observe partial dependencies
Apply carefully chosen input modifications (edit, add or delete feature values)
Result of changing income from $3,000 to $20,000 (edit data point)

What? - The Tool
Data
Machine Learning Model
What-If Tool

Model Understanding Frameworks
Black-Box:
Does not rely on internals
Probing depending on in- and outputs
General - used in many applications
WIT
White-Box:
Illuminates internal workings
Specific for a model
Often not applicable

What? - The Tool
Build using Tensorboard, a code-free and installation-free visualization framework
No custom coding (N1)
Help developers and practitioners to understand ML systems
Covers many standpoints (inputs / single data points / models)
Basic layout: 2 main panels → control panel & visualization panel
https://pair-code.github.io/what-if-tool/iris.html

Problem & Objective
Problem:
Machine Learning models (e.g. deep learning) are "black-boxes"
 Responses of models to different inputs cannot be easily foreseen
Big step in AI: Explainability

Objective:
Gain understanding of a model’s capabilities
○ when does it perform well/poorly
○ how is a change in the input reflected in the output (diversity)

Solution:
Interactive visual "what-if" exploration

Feature Analysis: Dataset Summary

Why? - Task 1: Exploring the Data
Customizable Analysis
Generate & test hypotheses about how model treats data
Edit data points
Identify counterfactuals
Observe partial dependencies
Apply carefully chosen input modifications (edit, add or delete feature values)
Result of changing income from $3,000 to $20,000 (edit data point)
How? - Task 3: Evaluate Performance and Fairness

- Slice data by feature values
- Perform measures on the subset
  - ROC
  - Confusion Matrix
  - Cost Ratio
- Measures can also be applied to compare models

Data Scaling

- Assumption: Standard laptop
- Computational restrictions:
  - Tabular Data:
    - # Features: 10-100
    - # Datapoints: ~100,000
  - Image Data:
    - Pixel dimensions: 78x64
    - # Datapoints: 2,000

- Comment:
  - As seen before, occlusion already a problem with less data

Evaluation

- 3 case studies executed
  - 2 studies in a large software company
  - 1 study in a university environment
- Showing the potential of WIT to:
  - Uncover bugs
  - Explore the data
  - Find partial dependencies

Analysis Summary

- What data:
  - User data & machine learning models
- What derived:
  - Inference of the model (on the data)
- What shown:
  - Dataset- and datapoint-level results of ML models
  - Giving a better understanding of the capabilities and possible adversarial attacks

Strength and Weaknesses

Strengths:
- Versatile tool
- Many useful real-world applications
- Greatly reducing workload compared to creating own visualizations

Weaknesses:
- Only easily compatible with Tensorflow (one deep-learning library)
- Occlusion is a problem, already with small datasets (150 data points, see example)
- Strict computational restrictions (100,000 data points is not a lot)

Thank You

Questions?