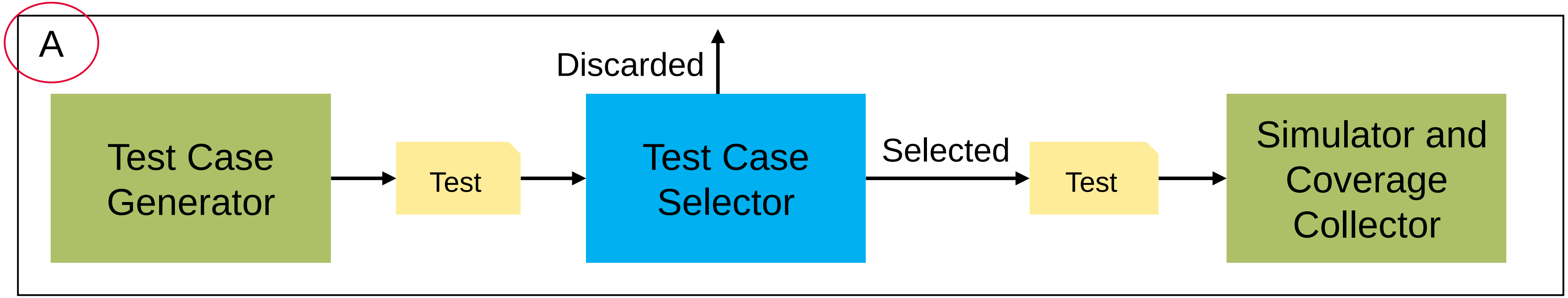
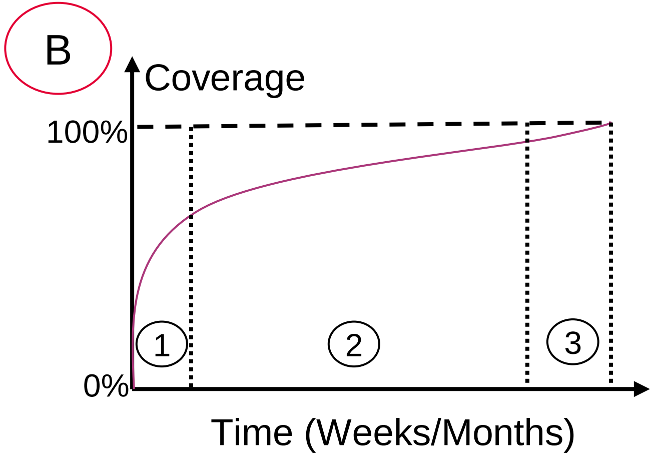
Using Neural Networks to Select Test Cases for Coverage Closure

Tim Blackmore, *Infineon Technologies*
Xuan Zheng, *University of Bristol*
Kerstin Eder, *University of Bristol*

Test Case Selection and Coverage Closure

Test Selection for Coverage Closure





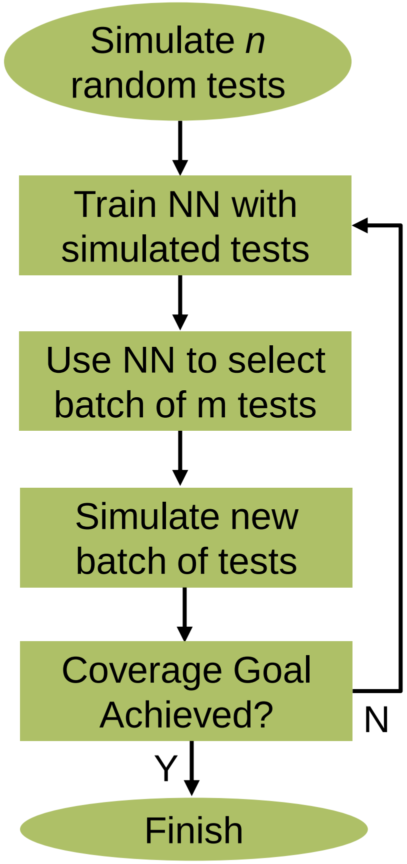
A. Test selector introduced between generator and simulator
   - Generation cheap, simulation expensive
   - At time of selection no coverage information is known about non-simulated tests

B. Coverage closure split into 3 phases
   1. Quick coverage growth – any easy-to-hit bins covered
   2. Coverage growth slows – many generated tests do not add to coverage
   3. Manual biasing of tests typically needed to hit new coverage

Goal is to select tests that hit new coverage to reduce time spent in phases 2 and 3

Commonality in the two approaches

- Both approaches try to select tests that are novel with regard to (or contrast with) tests already simulated
- Both approaches use a Neural Network (NN) in the test selector
  - The feature set (input layer) for the neural networks are the same
    - The fields whose values are generated by the generator
  - The output layer (and the other layers) are different
  - (Both approaches can also use other ML models)
- Both approaches use same flow



Test Selector 1

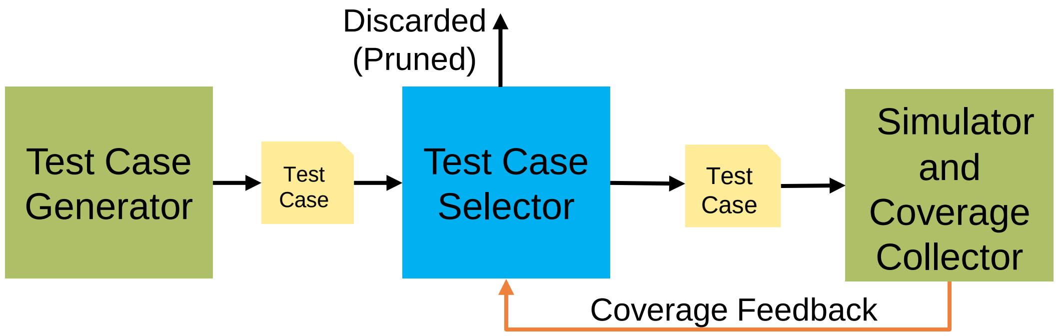
Machine Learning-Guided Stimulus Generation for Functional Verification

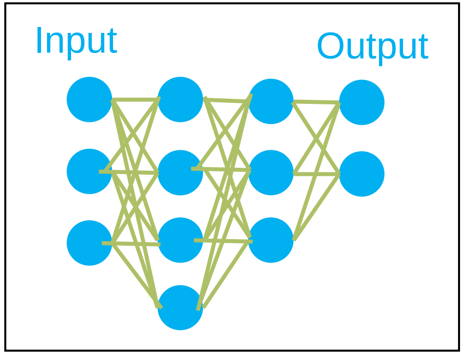
Saumil Gogri, Jiang Hu, Aakash Tyagi, Mike Quinn, Swati Ramachandran, Fazia Batool, and Amrutha Jagadeesh

Texas A&M University

DVCon U.S. 2020

Overview





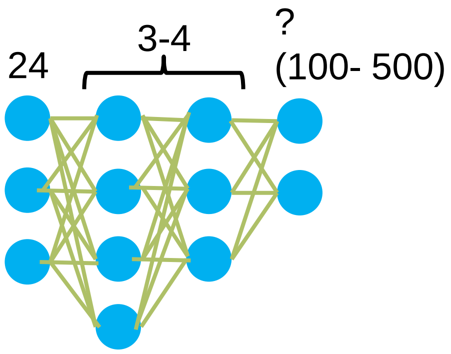
- Testcase selector uses NNs to predict whether test will hit coverage bins
  - NN trained via feedback loop from coverage on previously simulated tests
  - Neuron in output layer gives probability of coverage bin being hit (*p*)
- Uses of ternary classification of output neurons
  - Decided-1 ($p>\alpha$), Decided-0 ($p<\beta$), Undecided ($\beta<p<\alpha$)
- Tests are selected if the classifier either predicts Decided-1 on not-hit coverage or have a ‘fair number’ of undecideds
  - In practice this comes down to ‘fair number’ of undecideds
- ‘Fair number of undecideds’ on a well-trained network is interpreted as meaning that test has higher odds of having stimulus ‘contrasting’ with the simulated tests used to train the NN i.e. it is *novel*

Experiment

- Scope of experiment
  - 1738 coverage bins
    - Group A - 827 bins 'easy to reach by applying right test constraints'
    - Group A predicted across 6 NNs
    - Group B - 911 bins that 'do not have any obvious correlation to any test constraint'
    - Group B predicted across 2 NNs
  - Generation based on 24 'test constraints'
    - Some binary, some integers
    - 24 neurons in input layer of each NN
  - Batch size of 10 used
  - Group-A coverage hit by 587 random tests
  - Group-B coverage hit by circa 750 random tests

- Results with test selector
  - Group A coverage hit by 137 tests (77% saving)
  - But only 'little benefit' for Group B coverage (~10% saving?)
    - 'most bins fell into undecided category … only a few tests were pruned'
  - Naïve summing across both groups gives ~60% saving

- Is the divergence in results due to coverage type, or to number of nodes in NN relative to number of tests?
- Do the results scale?



Number of Tests	Group A	Group B
Random	587	~750
Test Selector	137	~680

Test Selector 2

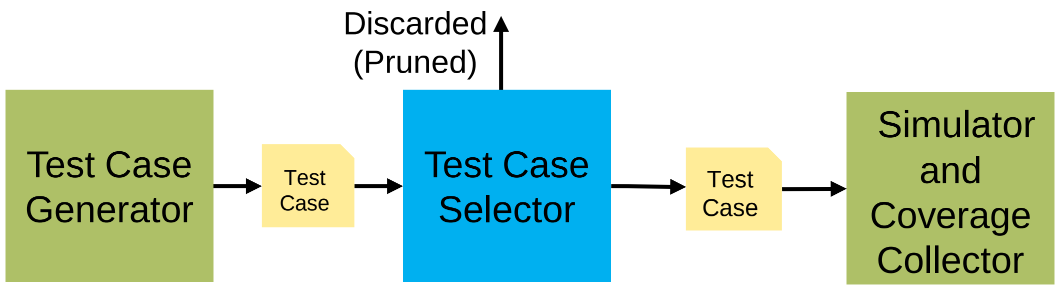
Novelty-Driven Verification: Using Machine Learning to Identify Novel Stimuli and Close Coverage

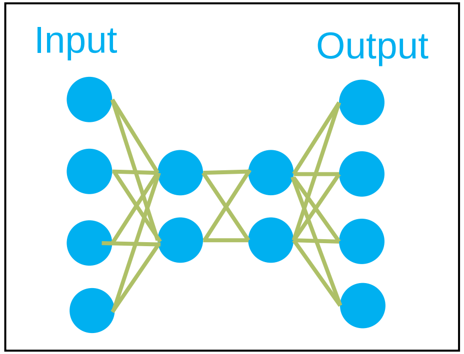
Tim Blackmore, Rhys Hodson, Sebastian Schaal

Infineon Technologies and Luminovo

DVCon U.S. 2021

Overview



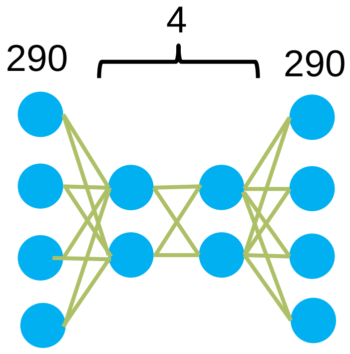


- Testcase selector uses NNs to compress and decompress the feature set (Autoencoder)
  - The output layer is now the same size as the input layer
  - NN trained to recover input values as output values (despite lossy compression in hidden layers)
  - No feedback loop to NN from coverage collection

- On a well-trained autoencoder a high loss between output nodes and input nodes is an indicator of novelty
- Tests with the highest loss are selected

Experiment

- Scope of experiment
  - 5992 coverage bins
    - All white-box functional coverage from industrial coverage model
  - 290 test features
    - After feature engineering
    - All binary (non-binary features one-hot encoded)
  - 85470 tests
    - Mix of a golden regression (3076 tests) and random tests (82644 tests)
    - In reality >2M random tests needed to hit all coverage
  - Batch size of 1000 used



- Results with test selector
  - 60% saving in number of tests to achieve 99% and even 99.5% coverage

Number of Tests	99%	99.5%
Random	52350	63500
Test Selector	21300	25400

- Autoencoder may be easier to train than coverage predictor
  - Building a simpler function using fewer neurons

Comparison on a 3rd data set

Experiment on 3rd Data Set

- Both approaches run on a 3rd data set
  - Implementation of first approach (Ternary Predictor) dependent on interpretation
  - Little effort spent on optimising either approach
- Data Set
  - Stored in SQLite database to avoid running simulations multiple times
  - Similar DUV to Experiment 2 (Autoencoder-based Test Selector)
  - 8409 white-box coverage bins (compare 5992)
  - 265 binary test features (compare 290) after feature engineering
  - Still approx. 85500 tests

Number of Tests and % Saving	97%	99%	99.5%	99.95%
Random	19236	41133	54202	83590
Test Selector 1 (Ternary Prediction)	12061	28193	46081	79626
% Saving	37%	31%	15%	5%
Test Selector 2 (Autoencoder)	9153	27599	38304	61079
% Saving	52%	33%	29%	27%

Conclusion

Conclusions

- Simulating tests that are most novel with regard to already simulated tests can lead to faster coverage closure
  - Novelty is a cheap(er), reasonable proxy for coverage
  - Perhaps …
- Many publications on use of machine learning in verification
  - Use very small data sets (scale)
  - Use single or few data sets (generalise)
  - Use proprietary data sets and code (reproducibility)
  - Compare single samples (interpretability)
- Adoption of machine learning techniques for verification would benefit from
  - Relevant, accessible (anonymised) public data sets and code
  - Use of standard methods (e.g. statistical) for presenting results

Part of your life. Part of tomorrow.