WOGAN at the SBST 2022 CPS Tool Competition

Jarkko Peltomäki

Information Technology
Åbo Akademi University

9.5.2022

Joint work with F. Spencer and I. Porres

WOGAN

- WOGAN is a general test suite generation algorithm which utilizes generative adversarial networks and neural networks.

Inputs and Outputs for BeamNG.research

- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).

Inputs and Outputs for BeamNG.research

- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).
  - Makes random search possible.

Inputs and Outputs for BeamNG.research

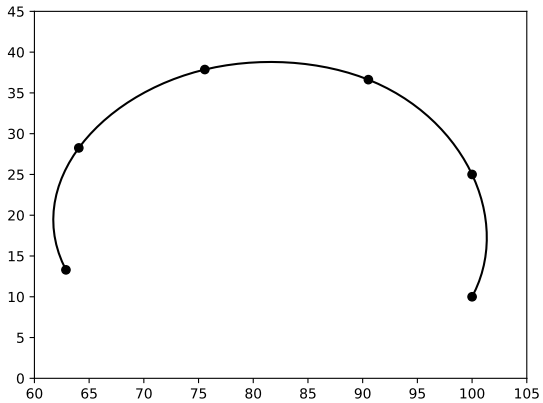
- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).
  - Makes random search possible.
- All roads start from the middle of the bottom of the map and point initially up.

Inputs and Outputs for BeamNG.research

- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).
  - Makes random search possible.
- All roads start from the middle of the bottom of the map and point initially up.
  - The output of the simulation should be rotation and translation invariant.

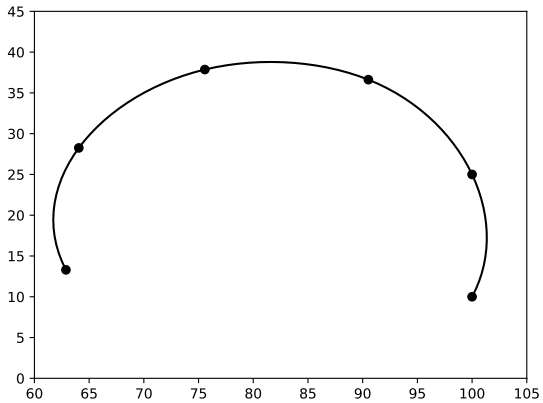
Inputs and Outputs for BeamNG.research

- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).
  - Makes random search possible.
- All roads start from the middle of the bottom of the map and point initially up.
  - The output of the simulation should be rotation and translation invariant.
- We fix the roads to have 6 points (5 curvature values).



Inputs and Outputs for BeamNG.research

- We use the curvature representation for roads as in Frenetic 2021 with step size 15 (map 200×200).
  - Makes random search possible.
- All roads start from the middle of the bottom of the map and point initially up.
  - The output of the simulation should be rotation and translation invariant.
- We fix the roads to have 6 points (5 curvature values).



  - This is an arbitrary decision. Small dimension makes learning with NNs easier, but higher dimensions allow more varied roads.

Inputs and Outputs for BeamNG.research

- We use as the output the maximum body out of lane percentage (BOLP) during the simulation.

Inputs and Outputs for BeamNG.research

- We use as the output the maximum body out of lane percentage (BOLP) during the simulation.
  - Signal output and distances to lane edges also available, but we did not use them.

Generators

- Consider the set $A \subseteq \mathcal{I}$ of inputs.

Generators

- Consider the set $A \subseteq \mathcal{I}$ of inputs.
- A generator on A is a model which can sample from the uniform distribution on A.

Generators

- Consider the set $A \subseteq \mathcal{I}$ of inputs.
- A generator on A is a model which can sample from the uniform distribution on A.
- Such a generator is trained based on a sample from A.

Generators

- Consider the set $A \subseteq \mathcal{I}$ of inputs.
- A generator on A is a model which can sample from the uniform distribution on A.
- Such a generator is trained based on a sample from A.
- WOGAN trains online a generator on

$$\{t \in \mathcal{I} : \mathrm{BOLP}(t) > 1 - \varepsilon\}$$

for some small ε.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \text{BOLP}(t)$.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \mathrm{BOLP}(t)$.
  - Sample G for tests and estimate their BOLP using A.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \text{BOLP}(t)$.
  - Sample G for tests and estimate their BOLP using A.
  - Select the test t with best estimated BOLP.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \text{BOLP}(t)$.
  - Sample G for tests and estimate their BOLP using A.
  - Select the test t with best estimated BOLP.
  - Execute t on the SUT to learn its true BOLP.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \text{BOLP}(t)$.
  - Sample G for tests and estimate their BOLP using A.
  - Select the test t with best estimated BOLP.
  - Execute t on the SUT to learn its true BOLP.
  - Add t to T.

The WOGAN Algorithm

- Sample N random tests T.
- Repeat while $|T| <$ budget:
  - Train generator G on high-BOLP samples of T.
  - Train analyzer A on T for the mapping $t \mapsto \text{BOLP}(t)$.
  - Sample G for tests and estimate their BOLP using A.
  - Select the test t with best estimated BOLP.
  - Execute t on the SUT to learn its true BOLP.
  - Add t to T.
- N.B. We execute the best test t on the SUT in order to find more training data for A. Without this the estimates of A can be unreliable.

Intuition

- The intuition is that over time A gets more accurate and thus more high-BOLP tests get included in the training batch of G. Thus G should be able to learn a distribution on high-BOLP tests.

Some Further Details

- We train the generator as a Wasserstein generative adversarial network (WGAN) as it is empirically known to be able to produce varied samples (should be good for road diversity).

Some Further Details

- We train the generator as a Wasserstein generative adversarial network (WGAN) as it is empirically known to be able to produce varied samples (should be good for road diversity).
- WOGAN is a general-purpose algorithm the only domain knowledge we use is the input representation.

Experimental Validation

- WOGAN among the best performing entries (better in BeamNG.AI and not so good in DAVE-2).

Experimental Validation

- WOGAN among the best performing entries (better in BeamNG.AI and not so good in DAVE-2).
- We are quite satisfied: WGAN enabled us diverse tests and we got good results without much domain knowledge.