DeeperHyperion

CS454 Project Sales Pitch

Team 6
20180459 Subeom
20224734 Arogya
20228163 Xiangchi
20180650 Hyunjoon

"DeepHyperion: Exploring the Feature Space of Deep Learning-Based Systems through Illumination Search"

Problem

- Dependability of Deep Learning (DL) Systems is more crucial than ever.
  - DL systems are now being used in many safety-critical domains.

- How do we ensure DL systems can be trusted with diverse real-world inputs?
  - Traditional code coverage metrics are not effective.
  - White box approaches are not sufficient to understand misbehaving input features.

- What if we could see a detailed view of the system's behavior with diverse inputs?
  - What about a feature map to interpret system behavior based on input characteristics?

DeepHyperion

- An automated test input generator for DL systems.
  - Generate diverse set of high-performing test inputs.

- “Illuminates” the input space by returning the highest-performing solution.
  - User can define the search space by features of interest.

- Provide a feature map where inputs are positioned based on their characteristics.
  - User can understand which inputs expose which misbehaviours.

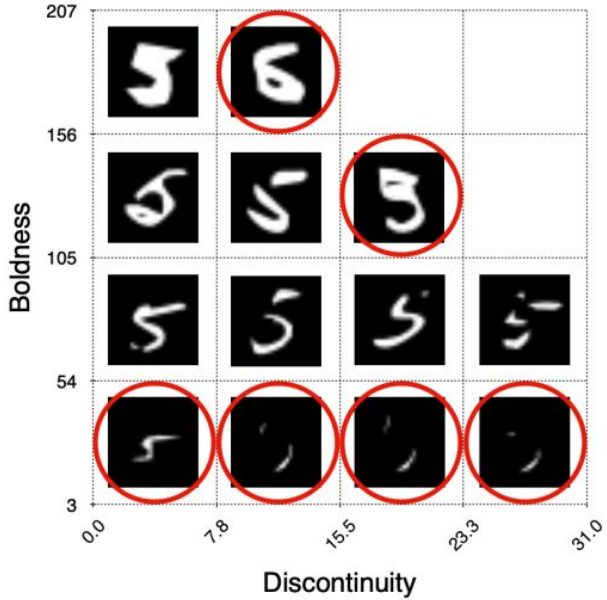

Figure 1: Feature map produced by DeepHyperion for a handwritten digit classifier. The two axes quantify two features: *discontinuity* and *boldness*. Cells show inputs that are either misclassified (marked with a circle) or close to being misclassified.

DeepHyperion: Initial Population Generation

- **Begin:** selecting a diverse set of individuals from the feature space.
- **Digit Classification:** Random inputs are chosen and converted to SVG format.
- **Initialise Population Function:** Chooses the most diverse inputs by calculating the pairwise Manhattan distance.
- **Greedy Manner:** Construct desired population size starting from a randomly chosen seed.

DeepHyperion: Selection and Mutation

- Random Selection
  - Broadly explore the feature space to illuminate it as fully as possible.
- Mutate Operator [**MUST in domain constraints**]
  - Ensures the mutated input differs from its original form (parent).

```
1  /* Initialise the map of elites */
2  map M ← ∅;
3  /* Generate initial population */
4  seeds S ← GENERATESEEDS(seedsize);
5  foreach s ∈ S do
6  |  EVALUATE(s);
7  end
8  population P ← INITIALISEPOPULATION(S, popsize);
9  /* Populate map with initial population */
10 foreach ind ∈ P do
11 |  M ← UPDATEMAP(ind) ;
12 end
```

DeepHyperion: Fitness and Feature Map

- Fitness
  - How close the DL system is to a **misbehaviour**?
  - For Digit classification,

= *confidence level of expected class - max(confidence level of any other classes)*

- Feature Map
  - feature space defined by **dimensions of variation** of…
    - structural features of inputs
    - behavioural features of the DL system

DeepHyperion: Dimensions for Digit Recognition

- Open Coding
  - Human assessors **manually** analyse inputs
  - Then, select relevant features that **best characterize inputs**
- Metric Identification
  - Among candidate metrics, select that best quantify the selected features

highest correlation, p < 0.05

Case Study	Feature	Metric	Agree	Correl	*p*-val
MNIST	Boldness	Lum	100%	0.67	<0.002
	Smoothness	AvgAng	66%	0.05	0.241
	Discontinuity	Mov	100%	0.90	<0.002
	Rotation	Or	-	0.43	<0.002

Investigating DeepHyperion

- **Ablation Study:**

Investigate the performance of DeepHyperion by progressively adding/removing features. Aim to identify which features are the most crucial for test-case generation.

- **Comparison with Baselines:**

Compare DeepHyperion's performance with other test generation tools or strategies to validate its effectiveness. This could be based on metrics like fault discovery, diversity of test cases, or speed of test generation.

- **Synthetic Fault Injection:**

Introduce synthetic faults into a DL model and observe if DeepHyperion can effectively discover them. This would be a direct measure of its testing capability.

DeeperHyperion?

- **Dynamic Dimension Selection:**

Try to dynamically determine the optimal number of dimensions based on the dataset using PCA or t-SNE

- **Hybrid Features:**

Mix both input-based and behavior-based features so to allow capturing not only the direct input properties but also how the DL system processes it with methods like CAM

- **Interactive Feature Selection:**

Create an interactive method that allows domain experts to hand-pick or give weight to certain features on DeepHyperion to prioritize certain dimensions. (Tensorboard)

- **Exploring other neural network models**

Redefine fitness for MNIST classifier to predict on different models to calculate difference between confidence level of expected class

Evaluating DeeperHyperion

- **Coverage Analysis:**

Measure the percentage of the feature space that DeepHyperion is able to explore

- **Fault Discovery Rate:**

Count the number of faults or misbehaviors DeepHyperion is able to identify as a ratio of the total number of generated test cases.

- **Diversity Score:**

Evaluate the diversity of generated test cases using clustering algorithms.

- **Performance Overhead:**

Measure the computational time and resources DeepHyperion consumes compared to its benefits in terms of fault discovery or coverage

- **Comparison Metrics:**

Measure other aspects like how quickly a fault is found, or the "severity" of discovered faults

Thank You

DeeperHyperion

CS454 Project Sales Pitch

Team 6

“DeepHyperion: Exploring the Feature Space of Deep Learning-Based Systems through Illumination Search”

Reference

[1] Zohdinasab, Tahereh, et al. "Deephyperion: exploring the feature space of deep learning-based systems through illumination search." Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. 2021.

[2] Mouret, Jean-Baptiste, and Jeff Clune. "Illuminating search spaces by mapping elites." arXiv preprint arXiv:1504.04909 (2015).