A Computability Perspective on (Verified) Machine Learning

Tonicha Crook, Jay Morgan, Arno Pauly and Markus Roggenbach

Swansea University

November 2021

Introduction

- Deep Neural Networks work well in image classification.

Introduction

- Deep Neural Networks work well in image classification.
- Learn with examples and non-examples.

Introduction

- Deep Neural Networks work well in image classification.
- Learn with examples and non-examples.
- Verification tasks of finding adversarial examples or preventing them.

Outline

Background

- Machine Learning is used in various domians

Background

- Machine Learning is used in various domians
- How much trust can we put into the responses of Machine Learning Models?

Background

- Machine Learning is used in various domians
- How much trust can we put into the responses of Machine Learning Models?
- Most verification techniques are hard to apply

Computable Analysis

- Most Machine Learning notions are based on real numbers

Computable Analysis

- Most Machine Learning notions are based on real numbers
- Computable Analysis is developed as the theory of functions on the real numbers and other sets from analysis, which can be computed by machines.

Computable Analysis

- Most Machine Learning notions are based on real numbers
- Computable Analysis is developed as the theory of functions on the real numbers and other sets from analysis, which can be computed by machines.
- What properties of the domain are actually needed to obtain the fundamental results?

Computable Analysis

- Most Machine Learning notions are based on real numbers
- Computable Analysis is developed as the theory of functions on the real numbers and other sets from analysis, which can be computed by machines.
- What properties of the domain are actually needed to obtain the fundamental results?
- What kind of verification questions are answerable about Machine Learning models?

Adversarial Examples

Adversarial Examples

An adversarial example is the result of a small change or perturbation to the original input that results in a change of classification made by the DNN. I.e. given the classifier f and an input x, an adversarial example is $f(x) \neq f(x+r)$ for $||r|| \leq \epsilon$ and $\epsilon > 0$.

Adversarial Examples

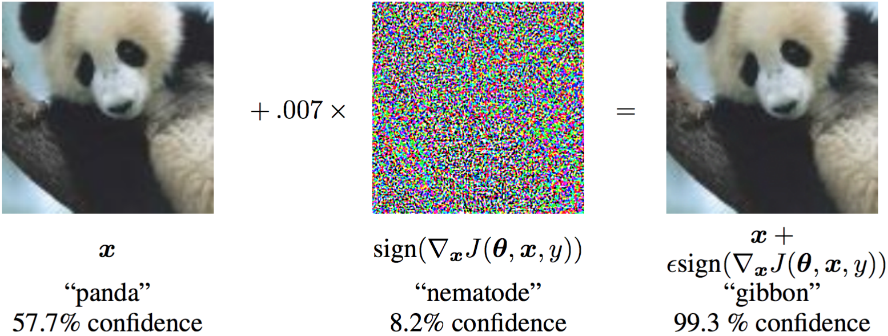

Figure: Adverserial Example - PyTorch

[1] Adversarial Example Generation. Available at https://pytorch.org/tutorials/beginner/fgsm_tutorial.html

Classifiers

Properties of a classifier

Simple examples of properties that such a classifier f might exhibit, include:

- Can f output a specific colour for points stemming from a given region?
- Is f constant on a given region?
- Are there two 'close' points that f maps to distinct colours?

Basic Verification Questions

Computable Verification Questions

The following verification questions are computable:

1. Is there any point that a given function maps to a particular class?

Basic Verification Questions

Computable Verification Questions

The following verification questions are computable:

1. Is there any point that a given function maps to a particular class?
2. Are all inputs mapped to a particular class?

Basic Verification Questions

Computable Verification Questions

The following verification questions are computable:

1. Is there any point that a given function maps to a particular class?
2. Are all inputs mapped to a particular class?
3. Do all things map to a specific answer, n, or does something map to a different colour to n?

Basic Verification Questions

Computable Verification Questions

The following verification questions are computable:

1. Is there any point that a given function maps to a particular class?
2. Are all inputs mapped to a particular class?
3. Do all things map to a specific answer, n, or does something map to a different colour to n?
4. Does everything map to the same answer or is there clear variation?

Choosing the Distance Parameter

Locally Constant

LocallyConstant is a map which has a point, a radius of a ball around the point and a function.

Choosing the Distance Parameter

Locally Constant

LocallyConstant is a map which has a point, a radius of a ball around the point and a function.

Locally Constant and Adversarial Examples

- Are their adversarial examples in the vicinity of the point?

Choosing the Distance Parameter

Locally Constant

LocallyConstant is a map which has a point, a radius of a ball around the point and a function.

Locally Constant and Adversarial Examples

- Are their adversarial examples in the vicinity of the point?
- How small the do the perturbations need to be to count as adversarial examples?

Choosing the Distance Parameter

Locally Constant

LocallyConstant is a map which has a point, a radius of a ball around the point and a function.

Locally Constant and Adversarial Examples

- Are their adversarial examples in the vicinity of the point?
- How small the do the perturbations need to be to count as adversarial examples?
- How much would we need to disturb a given point in order to get an adversarial example?

Choosing the Distance Parameter

Locally Constant

LocallyConstant is a map which has a point, a radius of a ball around the point and a function.

Locally Constant and Adversarial Examples

- Are their adversarial examples in the vicinity of the point?
- How small the do the perturbations need to be to count as adversarial examples?
- How much would we need to disturb a given point in order to get an adversarial example?

Optimal Radius

OptimalRadius is a map which shows the optimal radius needed for the closed ball in order for the point to become an adversarial example.

Learners and their Robustness

Learner

A learner is a map from finite sequences of labelled points to classifiers.

Learners and their Robustness

Learner

A learner is a map from finite sequences of labelled points to classifiers.

Robust Points

How robust is a classifier under small additions to the training data?
A basic version of this is a map which can give one of three responses, 1, 0 or no answer.

Conclusions

- Removing conditions usually leads to non-computability.

Conclusions

- Removing conditions usually leads to non-computability.
- The efficiency of the algorithms will be crucial for practical relevance.

Conclusions

- Removing conditions usually leads to non-computability.
- The efficiency of the algorithms will be crucial for practical relevance.
- What if we changed the questions we asked?

Thank You for Listening

Tonicha Crook, Jay Morgan, Arno Pauly & Markus Roggenbach:
A Computability Perspective on (Verified) Machine Learning.
arXiv:2102.06585