Introductory Lecture

Deep Learning Verification

Kumar Madhukar

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

IITD Winter Systems School 2023

December 5, 2023

Program Verification

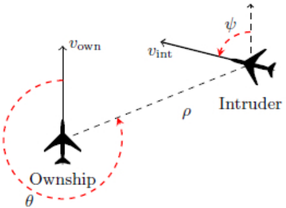
```
int fac (int x)

  int y = 1;
  int z = 0;

  while (z != x)
    z = z + 1;
    y = y * z;


  return y;
```

- property: if $(x > 0)$, then y $=$ `factorial`(x)

Neural Networks

- no notion of functional correctness
- consider the task of recognizing digits from images
- formalizing what exactly distinguishes the digit 2 from a 7, in a way that captures all common handwriting styles, is impossible
- which is why we use neural networks in the first place
- but, then, how can we ensure that the networks are well-behaved?

Aircraft Collision Avoidance System



- trained nets instead of look-up tables
- but what about behavioral guarantees?

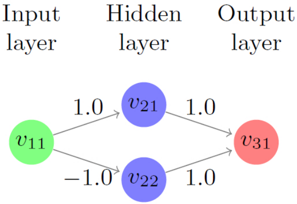
DNN Properties

- Safety
  - the network won't do anything undesirable
- Privacy
  - the network hasn't memorized the training examples
- Robustness
  - small changes won't affect the output of the network
- Consistency
  - network's predictions are consistent with certain laws (of physics, nature, etc.)

Formal Verification of DNNs

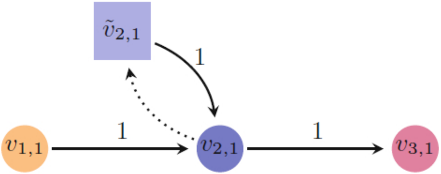
- we can define and study these properties formally
    - is there an input (say, a cat's image), classified as *cat* by the network N, such that the same image with reduced brightness is classified as *not cat* by N?
    - can be modelled as an SMT query (solved by an SMT solver, e.g. Z3)
- networks are large in size; non-linear activation functions

A toy example

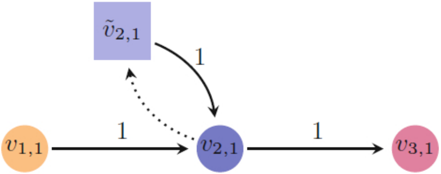


```
toy-network (real v11)
  real v21, v22, v31;

  v21 = 1.0 * v11;
  if (v21 <= 0)
     v21 = 0;


  v22 = -1.0 * v11;
  if (v22 <= 0)
     v22 = 0;


  v31 = (1.0 * v21) + (1.0 * v22);
  return v31;
```

Recurrent Neural Networks

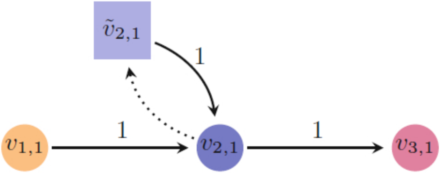


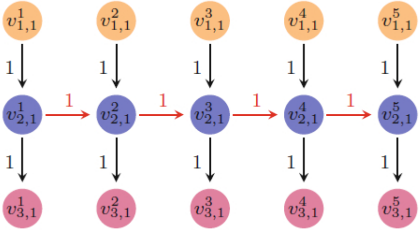
Recurrent Neural Networks



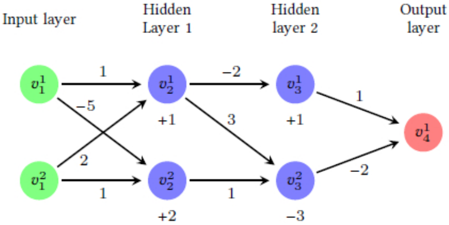
Time Step	$v_{1,1}$	$v_{2,1}$	$\tilde{v}_{2,1}$	$v_{3,1}$
1	0.5	0.5	0	0.5
2	1.5	2	0.5	2
3	-1	1	2	1
4	-3	0	1	0

Recurrent Neural Networks

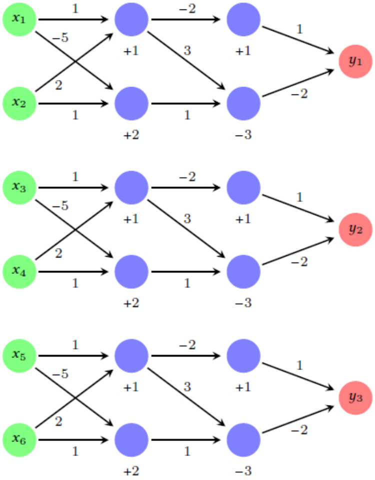




Deep Reinforcement Learning



Deep Reinforcement Learning



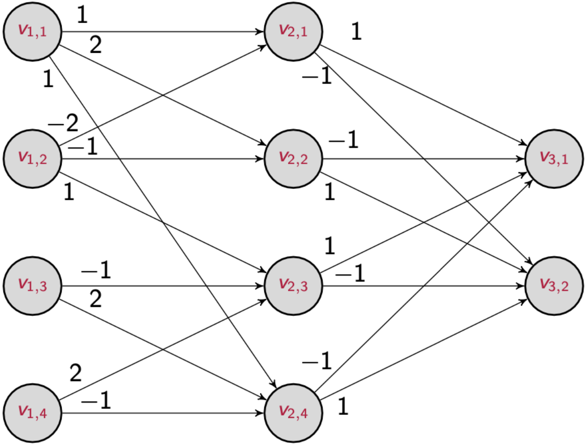
Interesting problems, active research area

- minimizing DNNs without affecting their functionality
- abstraction-refinement techniques for verification
- repairing networks
- explainability of DNNs

Lab

- Install Marabou: `https://github.com/NeuralNetworkVerification/Marabou`
- Marabou documentation: `https://neuralnetworkverification.github.io/Marabou/`
- Task: to run Marabou on some ACAS benchmarks

Lab



Thank you!