Safety verification for deep neural networks with provable guarantees

Marta Kwiatkowska

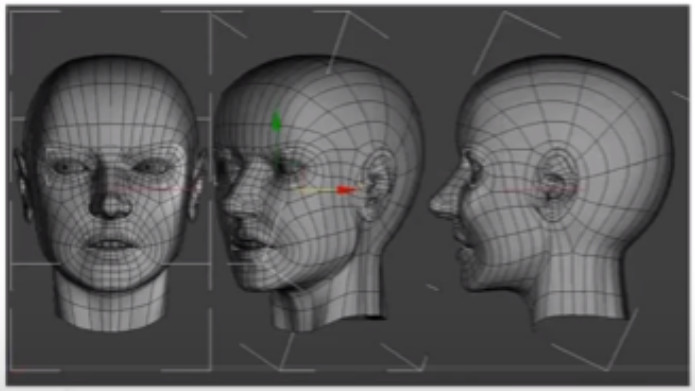
Department of Computer Science, University of Oxford

Based on CAV 2017, TACAS 2018 and IJCAI 2018 papers
and joint work with X Huang, W Ruan, S Wang, M Wu and M Wicker

SoMLMFM at FLoC 2018, 13th July 2018

Much industrial interest in deep learning

DeepFace
Closing the Gap to Human-Level Performance in Face Verification



Yaniv Taigman
Ming Yang
Marc'Aurelio Ranzato
Lior Wolf
- 2014

97.35% accuracy
Trained on the largest facial dataset – 4M facial images belonging to more than 4,000 identities.

News in the last few weeks…

Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam

Leer en español

By DAISUKE WAKABAYASHI MARCH 19, 2018

Tesla Says Crashed Vehicle Had Been on Autopilot Before Fatal Accident

By GREGORY SCHMIDT MARCH 31, 2018

RELATED COVERAGE

Tesla Looked Like the Fu
Ask if It Has One. MARC

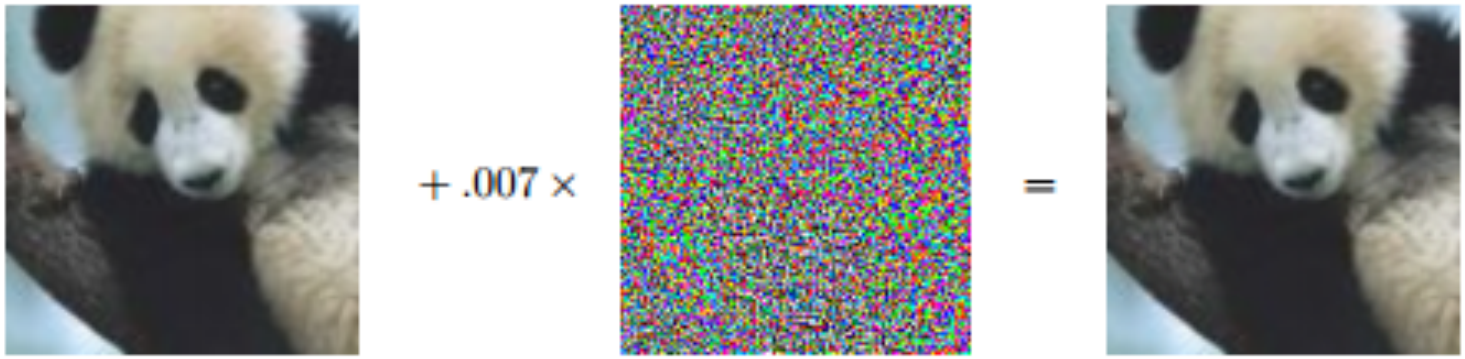
Fatal Tesla Crash Raises New Questions About Autopilot System

U.S. Safety Agency Criticizes Tesla Crash Data Release

How can this happen if we have 99.9% accuracy?

https://www.youtube.com/watch?v=B2pDFjIvrIU

Deep neural networks can be fooled!



- They are unstable wrt adversarial perturbations
  - often imperceptible changes to the image [Szegedy et al 2014, Biggio et al 2013 …]
  - sometimes artificial white noise
  - practical attacks, potential security risk
  - transferable between different architectures

German traffic sign benchmark…

stop

30m speed limit

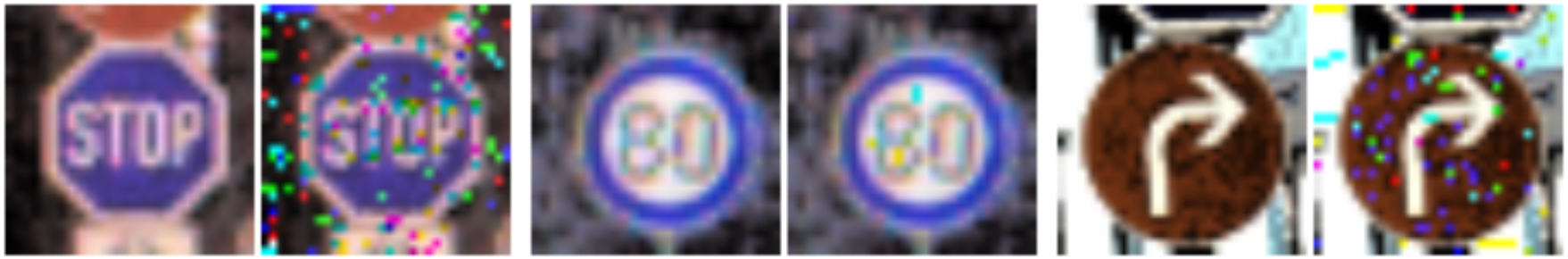
80m speed limit

30m speed limit

go right

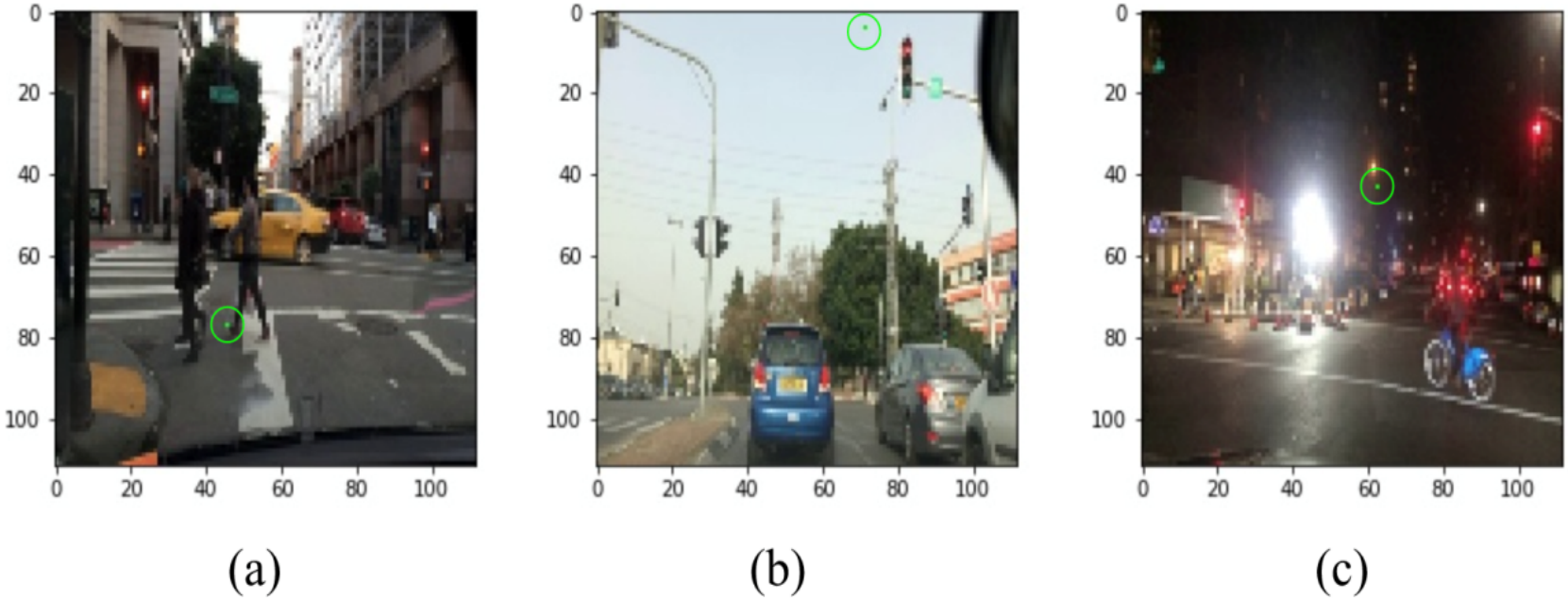
go straight

German traffic sign benchmark…



stop	30m speed limit	80m speed limit	30m speed limit	go right	go straight
	Confidence	0.999964	0.99		

Nexar traffic sign benchmark



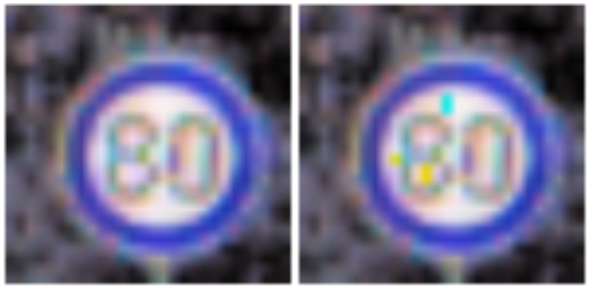
Red light classified as green with (a) 68%, (b) 95%, (c) 78% confidence after one pixel change.

- TACAS 2018, https://arxiv.org/abs/1710.07859

Can we verify that such behaviour cannot occur?

This talk

- First steps towards methodology to ensure safety of classification decisions
  - visible and human-recognisable perturbations: change of camera angle, snow, sign imperfections, ...
  - should not result in class changes
  - focus on individual decisions, pointwise robustness
  - images, but can be adapted to other types of problems



- Towards an automated verification framework
  - search: CAV 2017, https://arxiv.org/abs/1610.06940
  - game: TACAS 2018, https://arxiv.org/abs/1710.07859
  - global optim: IJCAI 2018, https://arxiv.org/abs/1805.02242

Problem setting

- Assume
  - vector spaces D_{L0}, D_{L1}, …, D_{Ln}, one for each layer
  - $h : D_{L0} \rightarrow \{c_1,\ldots c_k\}$ classifier function modelling human perception ability
- The network $f : D_{L0} \rightarrow \{c_1,\ldots c_k\}$ approximates h from M training examples $\{(x_i,c_i)\}_{i=1..M}$
  - built from activation functions ϕ_0, ϕ_1, …, ϕ_n, one for each layer
  - for point (image) $x \in D_{L0}$, its activation in layer k is

    $$\alpha_{x,k} = \phi_k(\phi_{k-1}(\ldots\phi_1(x)))$$

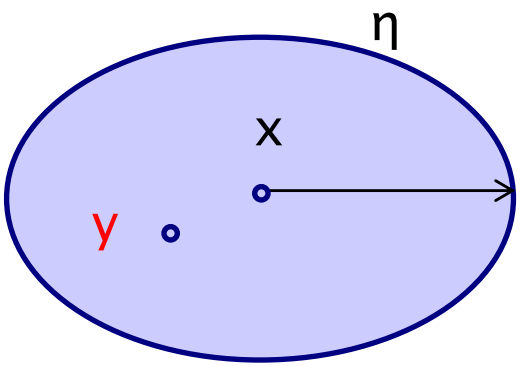
  - where $\phi_k(x) = \sigma(xW_k+b_k)$ and $\sigma(x) = \max(x,0)$
  - W_k learnable weights, b_k bias, σ ReLU

Robustness

- Regularisation such as dropout improves smoothness
- Common smoothness assumption
  - each point $x \in D_{L0}$ in the input layer has a region η around it such that all points in η classify the same as x
- Pointwise robustness [Szegedy et al 2014]
  - f is not robust at point x if $\exists y \in \eta$ such that $f(x) \neq f(y)$
- Robustness (network property)
  - smallest perturbation weighted by input distribution
  - reduced to non-convex optimisation problem

Safety of classification decisions

- Safety assurance process is complex
- Here focus on safety at a point as part of such a process
  - consider region supporting decision at point x
  - same as pointwise robustness…



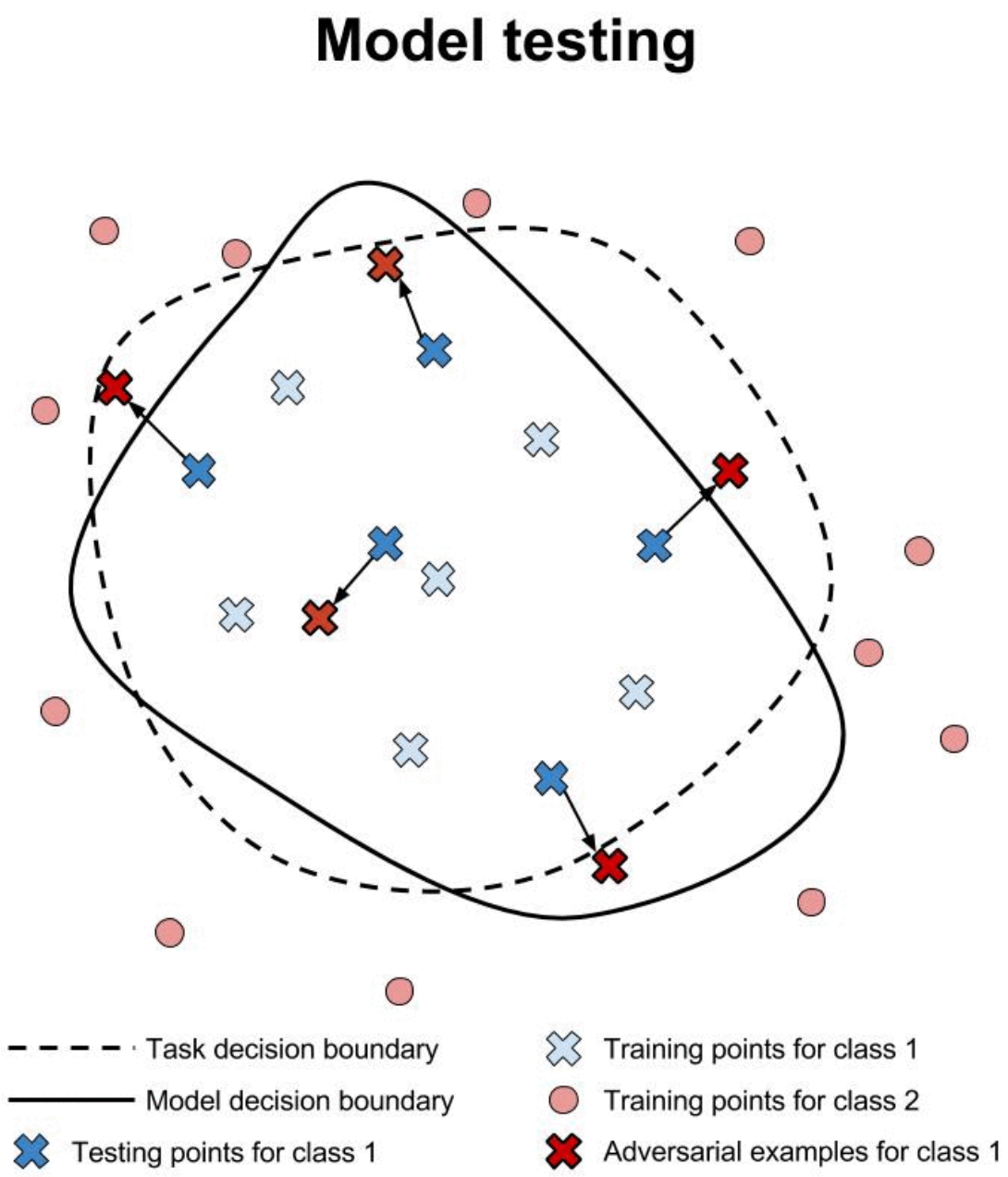
- But..
  - what diameter for region η?
  - which norm? L^2, L^∞ ?
  - what is an acceptable/adversarial perturbation?

- Introduce the concept of manipulation, a family of operations that perturb an image
  - think of scratches, weather conditions, camera angle, etc
  - classification should be invariant wrt safe manipulations

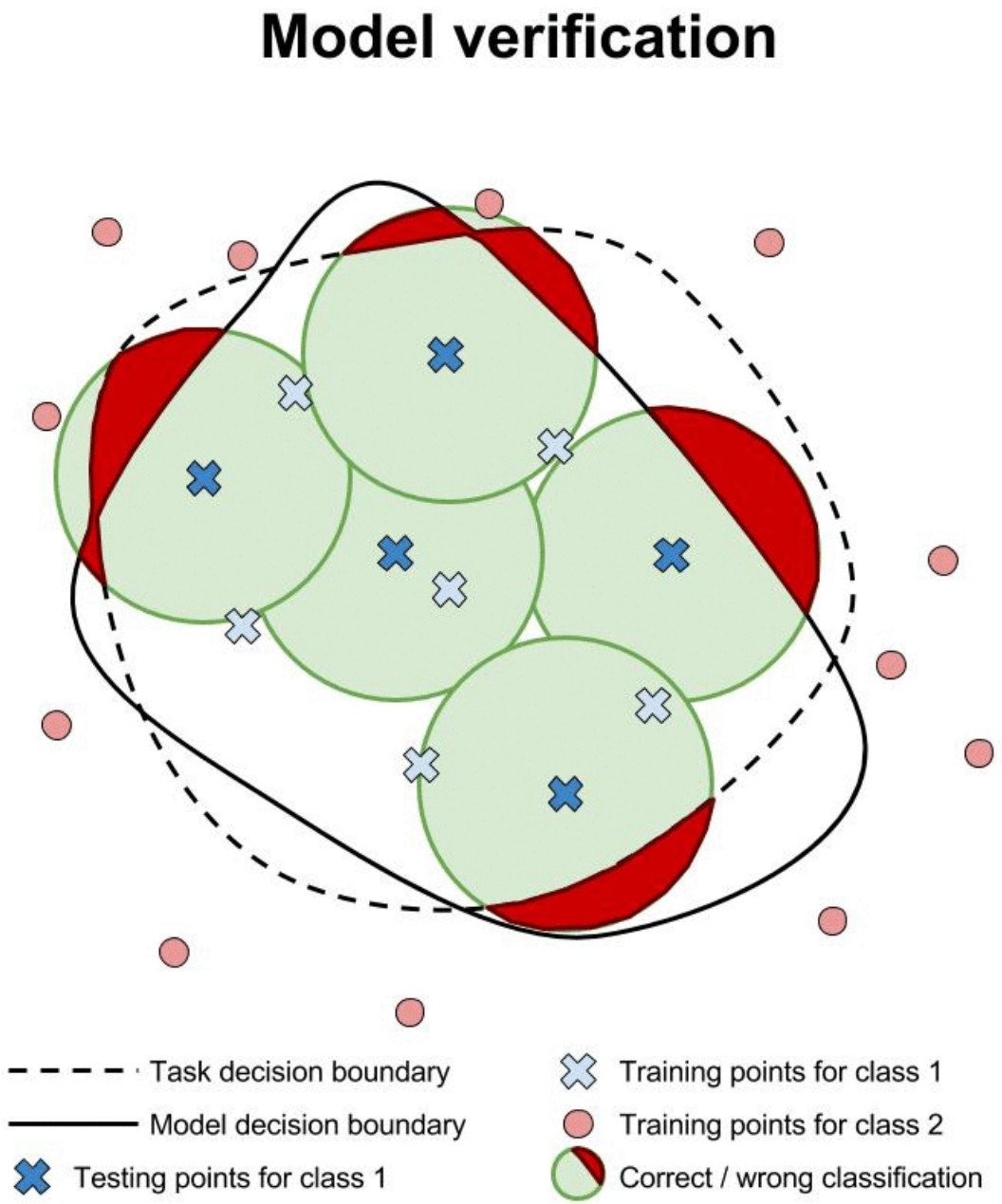
Safety verification

- Take as a specification set of manipulations and region η
  - work with pointwise robustness as a safety criterion
  - focus on safety wrt a set of manipulations
  - exhaustively search the region for misclassifications
- Challenges
  - high dimensionality, nonlinearity, infinite region, huge scale
- Automated verification (= ruling out adversarial examples)
  - need to ensure finiteness of search
  - provable guarantee of decision safety if adv. example not found
- Falsification (= searching for adversarial examples)
  - good for attacks, no guarantees

Training vs testing



Training vs testing vs verification



Verification framework

- Size of the network is prohibitive
  - millions of neurons!
- The crux of our approach
  - propagate verification layer by layer: safety wrt $\eta_k(\alpha_{x,k})$ and Δ_k implies safety wrt $\eta_{k-1}(\alpha_{x,k-1})$ and Δ_{k-1}
  - reduction to finite exhaustive search of the region by discretisation, subject to minimality of manipulations
  - implementation in SMT (counting problem in linear arithmetic)
  - NB employ various heuristics for scalability
- This differs from heuristic search for adversarial examples
  - no guarantee of precise adversarial examples
  - no guarantee of exhaustive search even if we iterate

CIFAR-10 example

ship

ship

truck

- 32x32 image size, 3 channels, medium size network (Conv, ReLU, Pool, FC, dropout and softmax)
- Working with 1st hidden layer, project back to input layer

ImageNet example

Street sign

Birdhouse

- 224x224 image size, 3 channels, 16 layers, state-of-the-art network VGG, (Conv, ReLU, Pool, FC, zero padding, dropout and softmax)
- Work with 20,000 dimensions (of 3m), unsafe for 2nd layer

Yet another ImageNet example

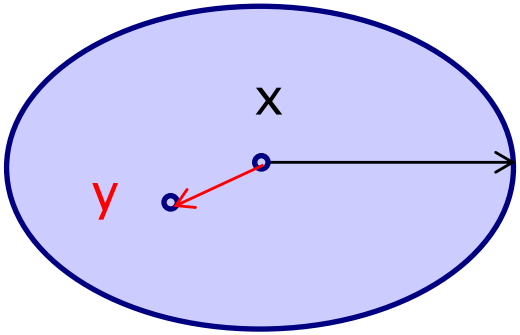
Labrador retriever

Lifeboat

- 224x224 image size, 3 channels, 16 layers, state-of-the-art network, (Conv, ReLU, Pool, FC, zero padding, dropout and softmax)
- Work with 20,000 dimensions

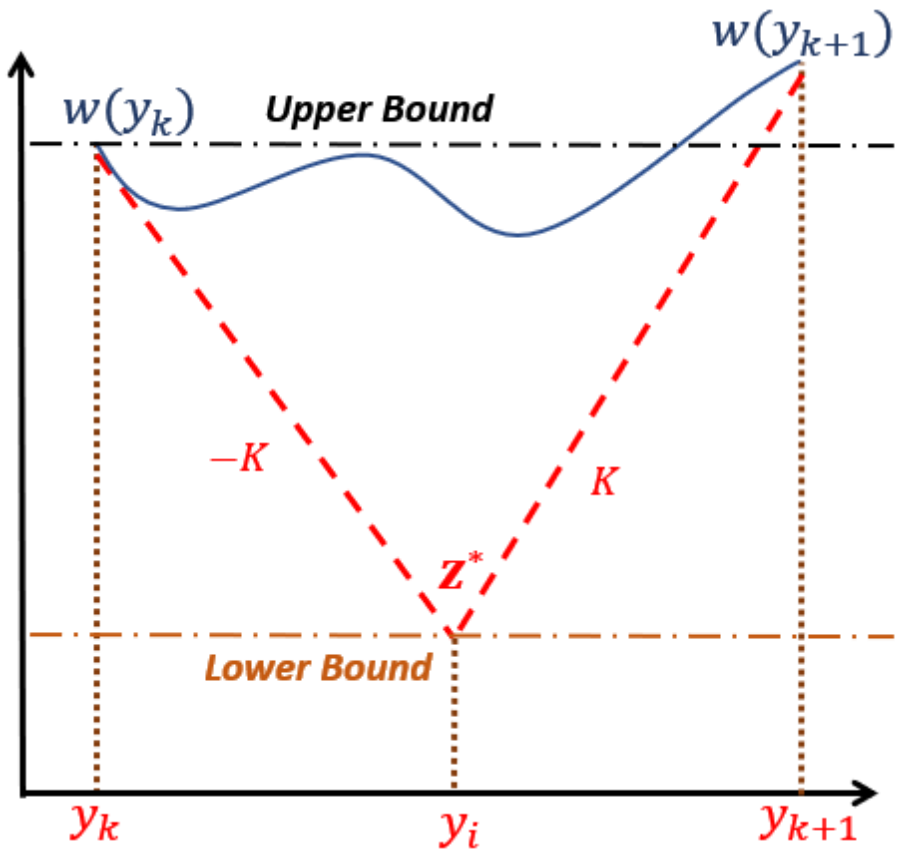
Alternative approach: reachability analysis

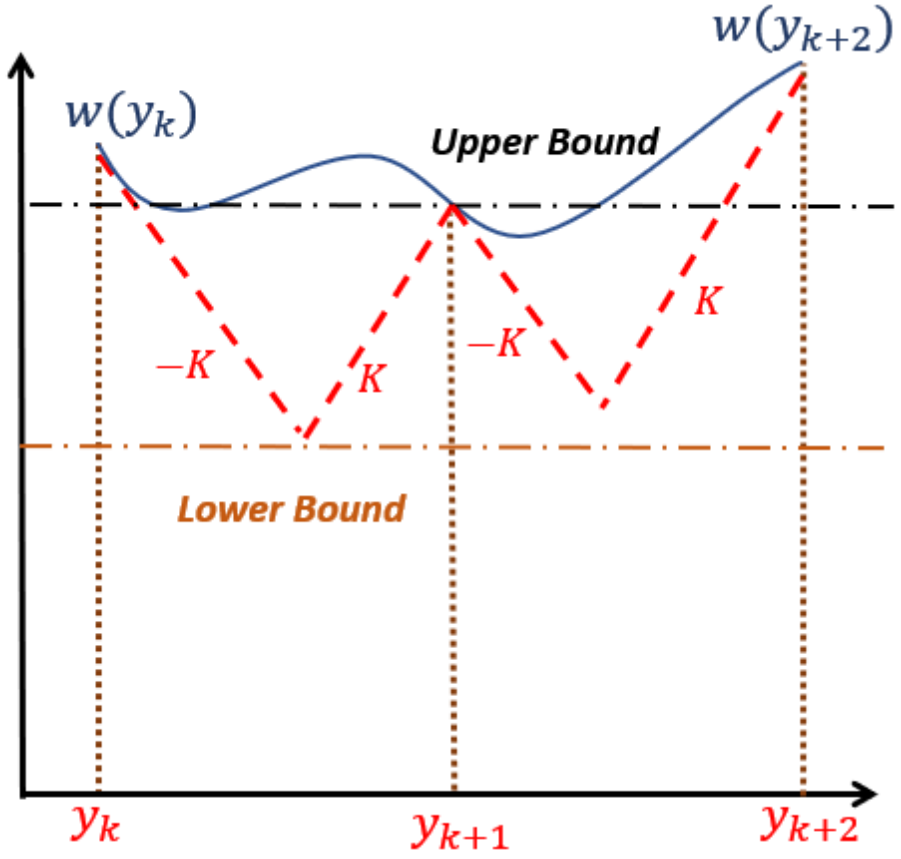
- Instead of relying on exhaustive search of discretized region, can we compute the reachable region?
- Under assumption of Lipschitz continuity
  - for $x \in \eta$, compute maximum/minimum value of $f(\eta)$
  - or maximum safe radius
  - using global optimisation
  - anytime fashion
- Gives provable guarantees
  - best/worst case confidence values
  - pointwise confidence diameter
  - can average over input distribution
- Method NP-complete
  - wrt the number of input dimensions, not number of neurons



- IJCAI 2018, https://arxiv.org/abs/1805.02242

Global optimization: main idea

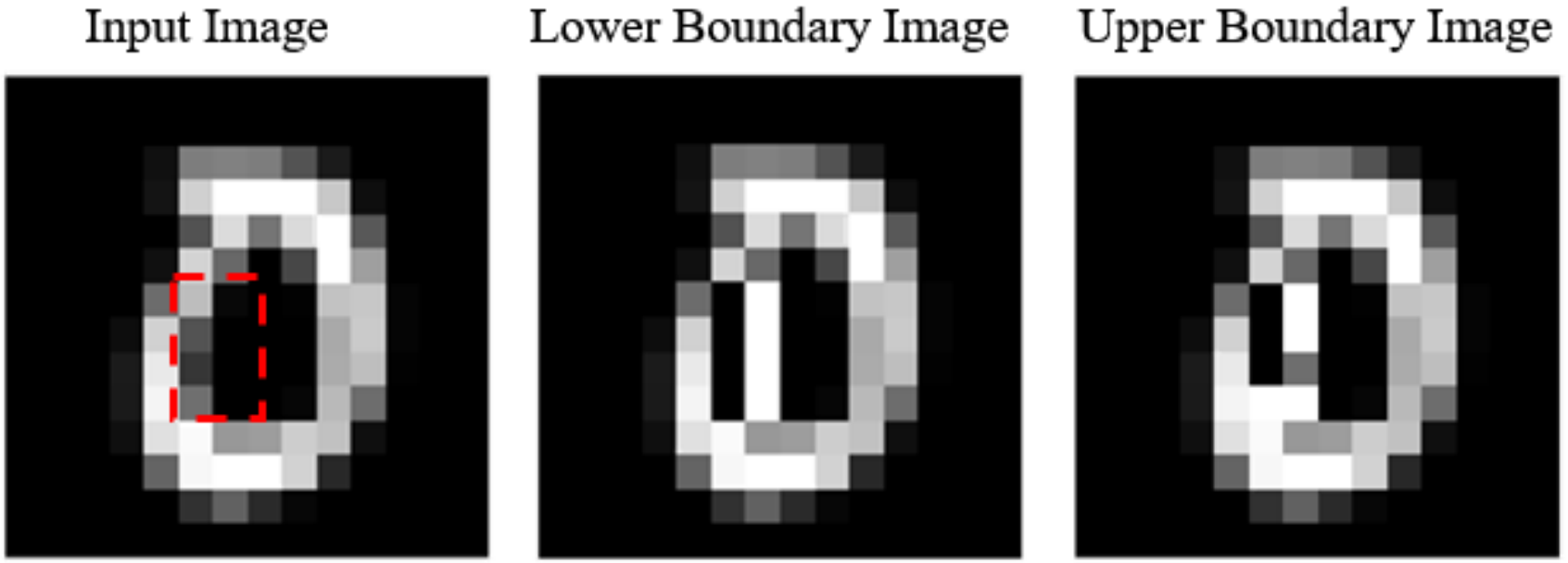




- Adaptive nested optimization, asymptotic convergence
  - construct a series of lower and upper bounds
- K – Lipschitz constant

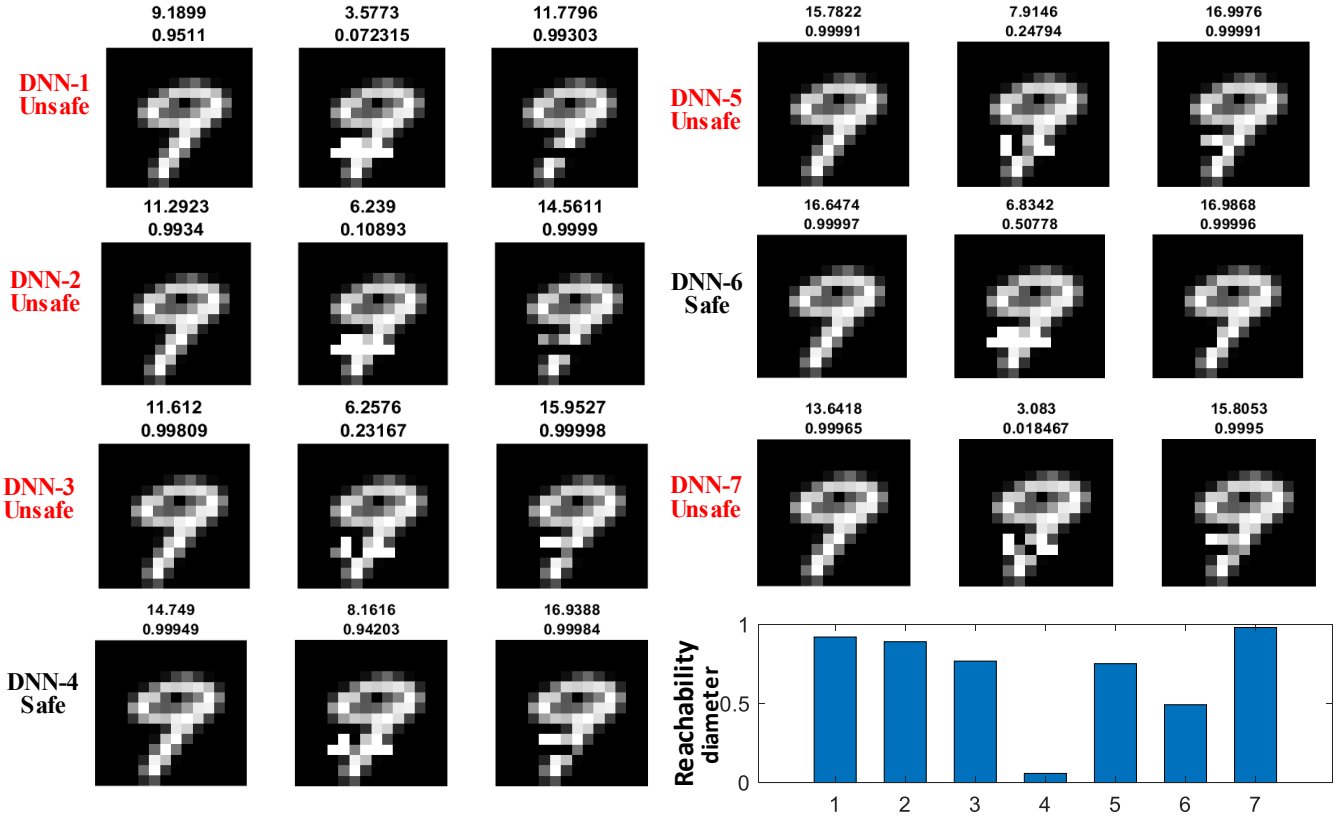
MNIST example

- Take an image and select a feature within it



- Safety verification for the feature
  - manipulating the feature can only reduce confidence to 74.36%

MNIST network comparison



- Showing pointwise confidence diameter
- Can obtain global robustness evaluation by averaging wrt the test data distribution

Searching for adversarial examples…

- Input space for most neural networks is high dimensional and non-linear
- Where do we start?
- How can we apply structure to the problem?

- Image of a tree has 4,000 x 2,000 x 3 dimensions = 24,000,000 dimensions
- We would like to find a very 'small' change to these dimensions

TACAS 2018, https://arxiv.org/abs/1710.07859

Feature-based exploration

- Searching by trying every combination of pixel values is intractable
- We can 'reduce' the dimensionality of an images by reducing it only to its salient features

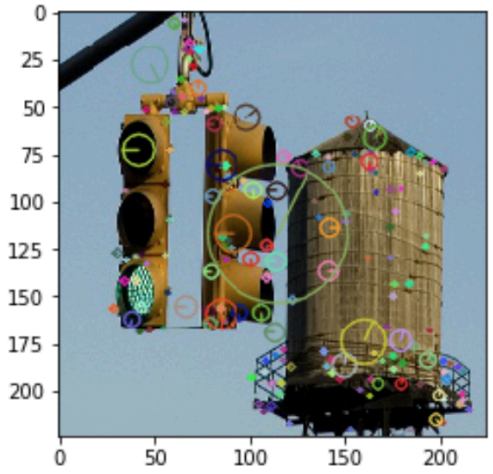
$\Lambda(\alpha)$ – Set of features given an image

λ_r – Response strength of the feature (roughly how 'important' it is)

λ_x – X coordinate of a keypoint

λ_y – Y coordinate of a keypoint

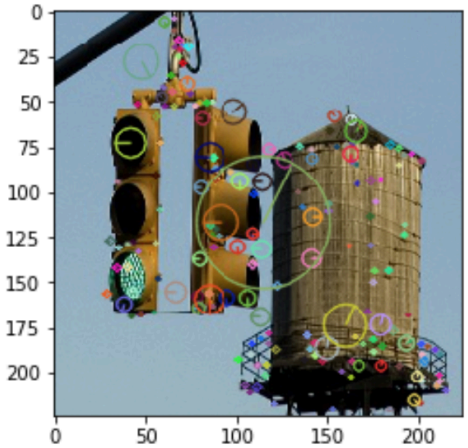
λ_s – Radius of a keypoint

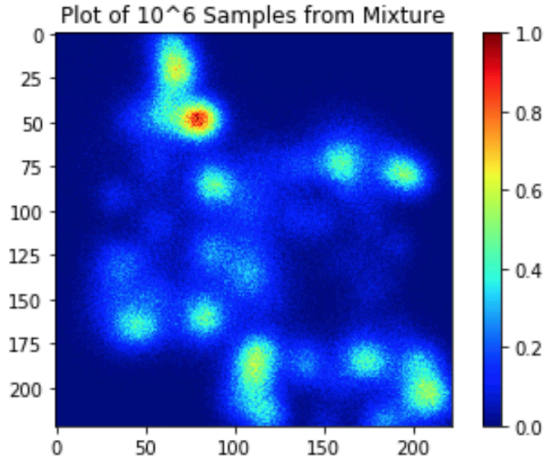


Feature-based representation

- Employ the SIFT algorithm to extract features
- Reduce dimensionality by focusing on salient features
- Use a Gaussian mixture model in order to assign each pixel a probability based on its perceived saliency

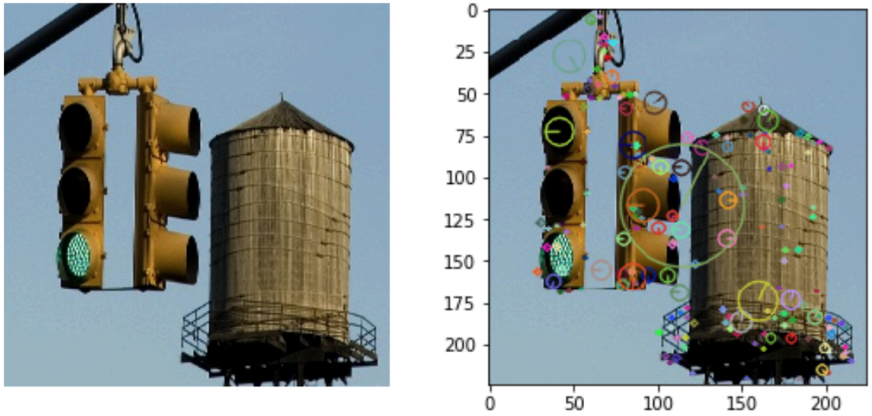
$$\mathcal{G}_{i,x} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp(\frac{-(p_x - \lambda_{i,x})^2}{2\lambda_{i,s}^2}) \quad \mathcal{G}_{i,y} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp(\frac{-(p_y - \lambda_{i,y})^2}{2\lambda_{i,s}^2})$$





Solution: two-player game

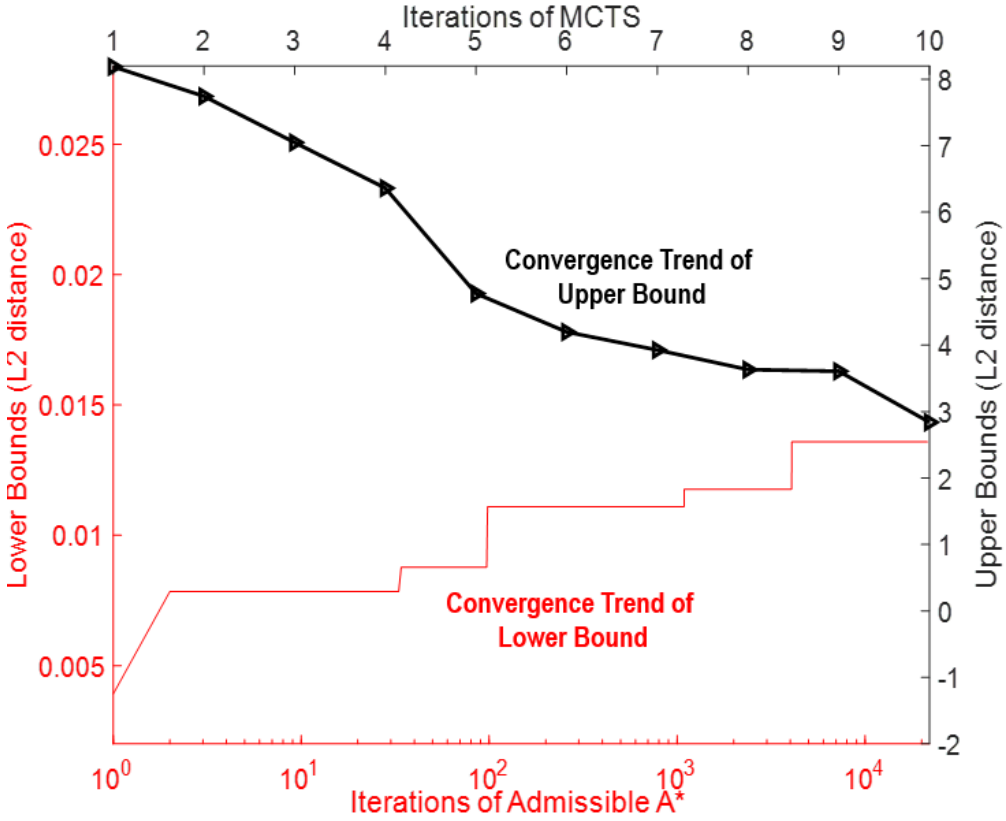
- Player 1 selects the feature that we will manipulate $\Lambda(\alpha)$

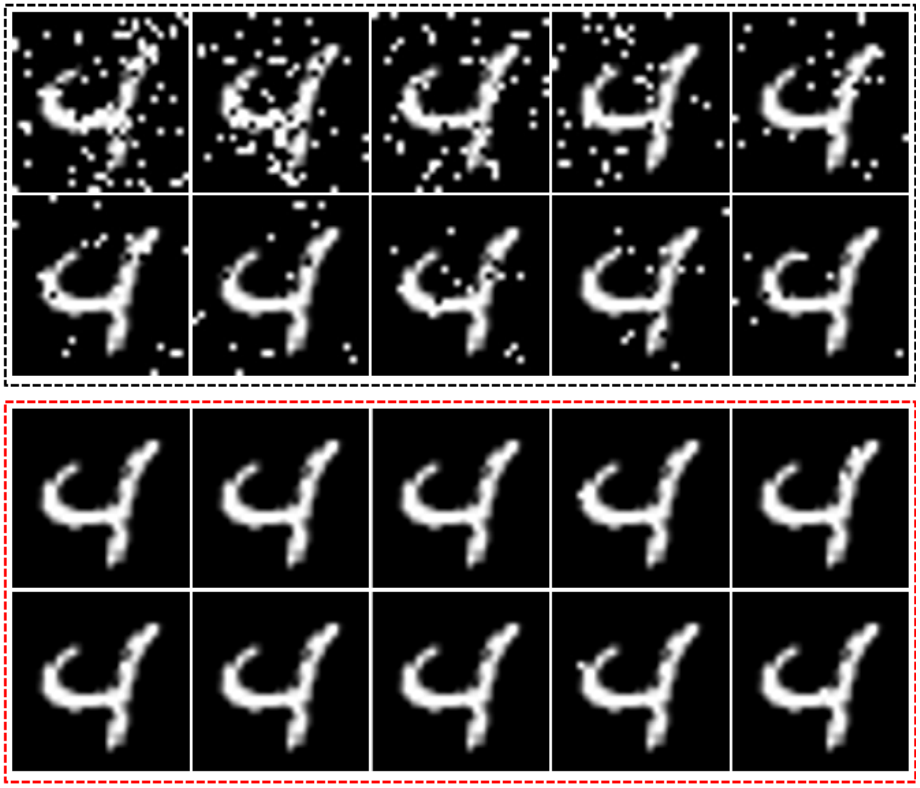


- Each feature represents a possible move for player 1
- Player 2 then selects the pixels within the feature to manipulate
- Use Monte Carlo tree search to explore the game tree, while querying the network to align features
- Method black/grey box, can approximate the maximum safe radius and feature robustness

Convergence (MNIST)

- Convergence of lower and upper bounds on maximum safe radius





Evaluating safety-critical scenarios: Nexar

- Dashboard camera images from the Nexar dataset were taken in order to test a safety critical situation
- Tens of thousands of images were taken from real dash cams in all weather and lighting conditions
- Challenge winning network achieves 95% accuracy over unseen test data

Evaluating safety-critical scenarios: Nexar

- Using our Game-based Monte Carlo Tree Search method we were able to reduce the accuracy of the network to 0%
- On average, each input took less than a second to manipulate (.304 seconds)
- On average each image was vulnerable to 3 pixel changes

Conclusion

- Deep learning should be more critically evaluated when put into practice in safety- and security-critical situations
- Adversarial examples help in understanding the robustness of DNN decision boundaries
- Proposed first framework for safety verification of deep neural network classifiers
  - search-based (SMT) and Monte Carlo tree search
  - feature-guided exploration for fast, black/grey-box testing, in a game-theoretic framework
  - provable guarantees for Lipschitz continuous networks
- Future work
  - how best to use adversarial examples: training vs logic
  - abstraction-refinement?
  - probabilistic properties?
  - more complex properties?

Acknowledgements

- My group and collaborators in this work
- Project funding
  - ERC Advanced Grant
  - EPSRC Mobile Autonomy Programme Grant

- See also
  - VERIWARE www.veriware.org
  - PRISM www.prismmodelchecker.org