Safety verification for deep neural networks

Marta Kwiatkowska

Department of Computer Science, University of Oxford

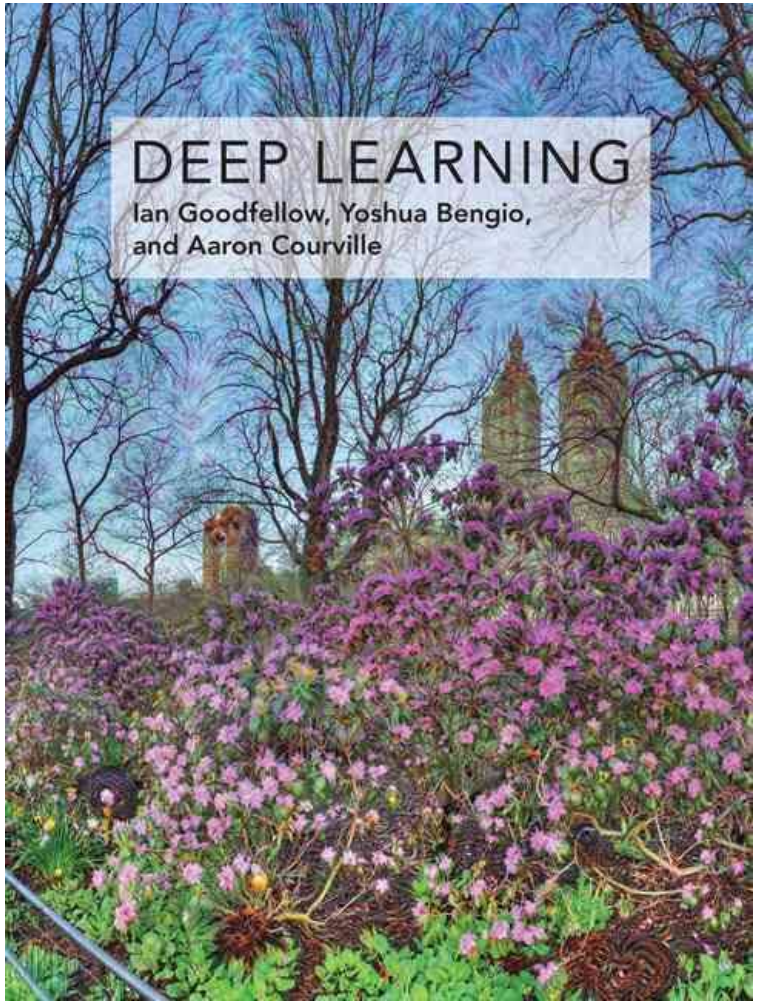
Based on joint work with Xiaowei Huang, Sen Wang, Min Wu and Matthew Wicker

CAV, Heidelberg, 26th July 2017

The unstoppable rise of deep learning

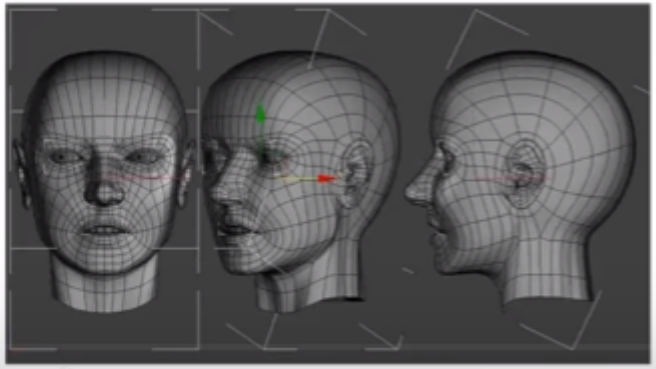
- Neural networks timeline
  - 1940s First proposed
  - 1998 Convolutional nets
  - 2006 Deep nets trained
  - 2011 Rectifier units
  - 2015 Vision breakthrough
  - 2016 Win at Go

- Enabled by
  - Big data
  - Flexible, easy to build models
  - Availability of GPUs
  - Efficient inference



Much interest from tech companies,

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.

Google Translate—here shown on a mobile phone—will use deep learning to improve its translations between texts.

...healthcare,

nature International weekly journal of science

Home | News | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For Authors

Archive > Volume 542 > Issue 7639 > Letters > Article > Article metrics > News

Article metrics for:

Dermatologist-level classification of skin cancer with deep neural networks

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun

Nature **542**, 115–118 (02 February 2017) | doi:10.1038/nature21056

Last updated: 24 July 2017 10:10:28 EDT

> The Stanford University team said the findings were "incredibly exciting" and would now be tested in clinics.
>
> Eventually, they believe using AI could revolutionise healthcare by turning anyone's smartphone into a cancer scanner.
>
> Cancer Research UK said it could become a useful tool for doctors.
>
> The AI was repurposed from software developed by Google that had learned to spot the difference **between images of cats and dogs**.

…and automotive industry

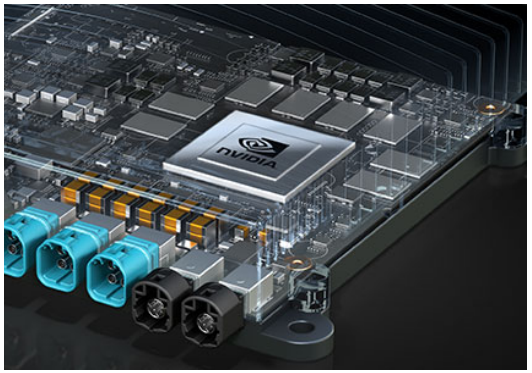
https://www.youtube.com/watch?v=mCmO_5ZxdvE

...and more

https://blogs.nvidia.com/blog/2017/01/04/bb8-ces/

What you have seen

- PilotNet by NVIDIA (regression problem)
  - end-to-end controller for self-driving cars
  - neural network
  - lane keeping and changing
  - trained on data from human driven cars
  - runs on DRIVE PX 2



- Traffic sign recognition (classification problem)
  - conventional object recognition
  - neural network solutions already planned…

- BUT
  - neural networks don't come with rigorous guarantees!

PilotNet https://arxiv.org/abs/1604.07316

What your car sees…

Original
Traffic light
(ImageNet class 920)

1200 pixels

VGG16

1287 pixels

VGG19
Misclassified

1812 pixels

RESNET

State-of-the art deep neural networks on ImageNet

Nexar traffic sign benchmark

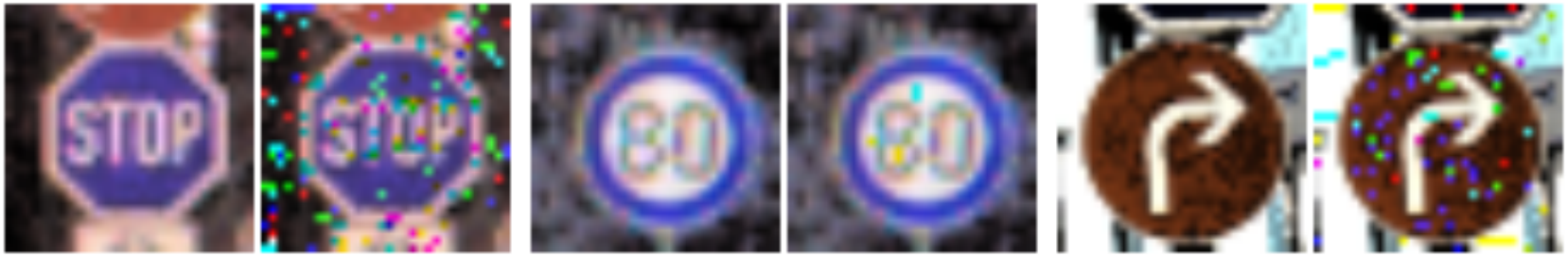
Red Light Modified to
Green after 18 white pixels.
Probability: 59%

Red Light Modified to
Green after 9 green pixels.
Probability: 50.9%

Red Light Modified to
Green after 9 green pixels.
Probability: 53%

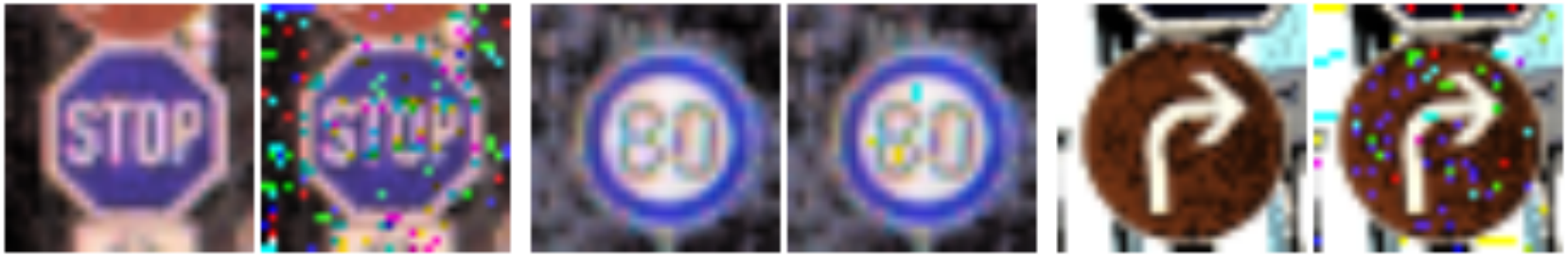
No Light Modified to Green
after 4 green pixels.
Probability: 51.9%

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		

Aren’t these artificial?

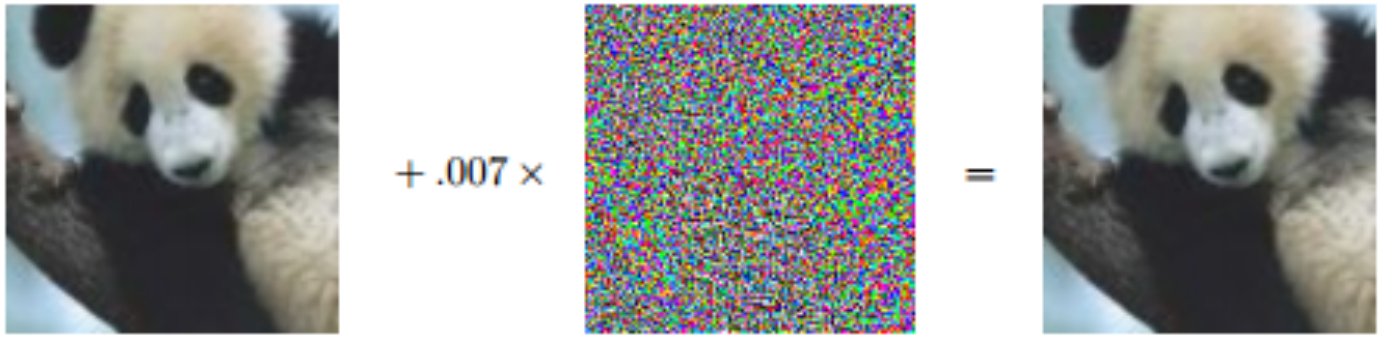
Venkat Viswanathan
@venkvis

Follow

.@TeslaMotors Model S autopilot camera misreads 101 sign as 105 speed limit at 87/101 junction San Jose. Reproduced every day this week.

4:40 AM - 15 Jul 2017

Deep neural networks can be fooled!



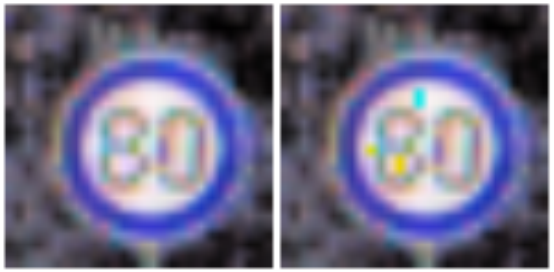
- They are unstable wrt adversarial perturbations
  - often imperceptible changes to the image [Szegedy et al 2014, Biggio et al 2013 …]
  - sometimes artificial white noise
  - practical attacks, potential security risk
  - transferable between different architectures

Risk and robustness

- Conventional learning theory
  - empirical risk minimisation [Vapnik 1991]
- Substantial growth in techniques to evaluate robustness
  - variety of robustness measures, different from risk
  - e.g. minimal expected distance to misclassification
- Methods based on optimisation or stochastic search
  - gradient sign method [Szegedy et al 2014]
  - optimisation, tool DeepFool [Moosavi-Desfooli et al 2016]
  - constraint-based, approximate [Bastani et al 2016]
  - adversarial training with cleverhans [Papernot et al 2016]
  - universal adversarial example [Moosavi-Desfooli et al 2017]

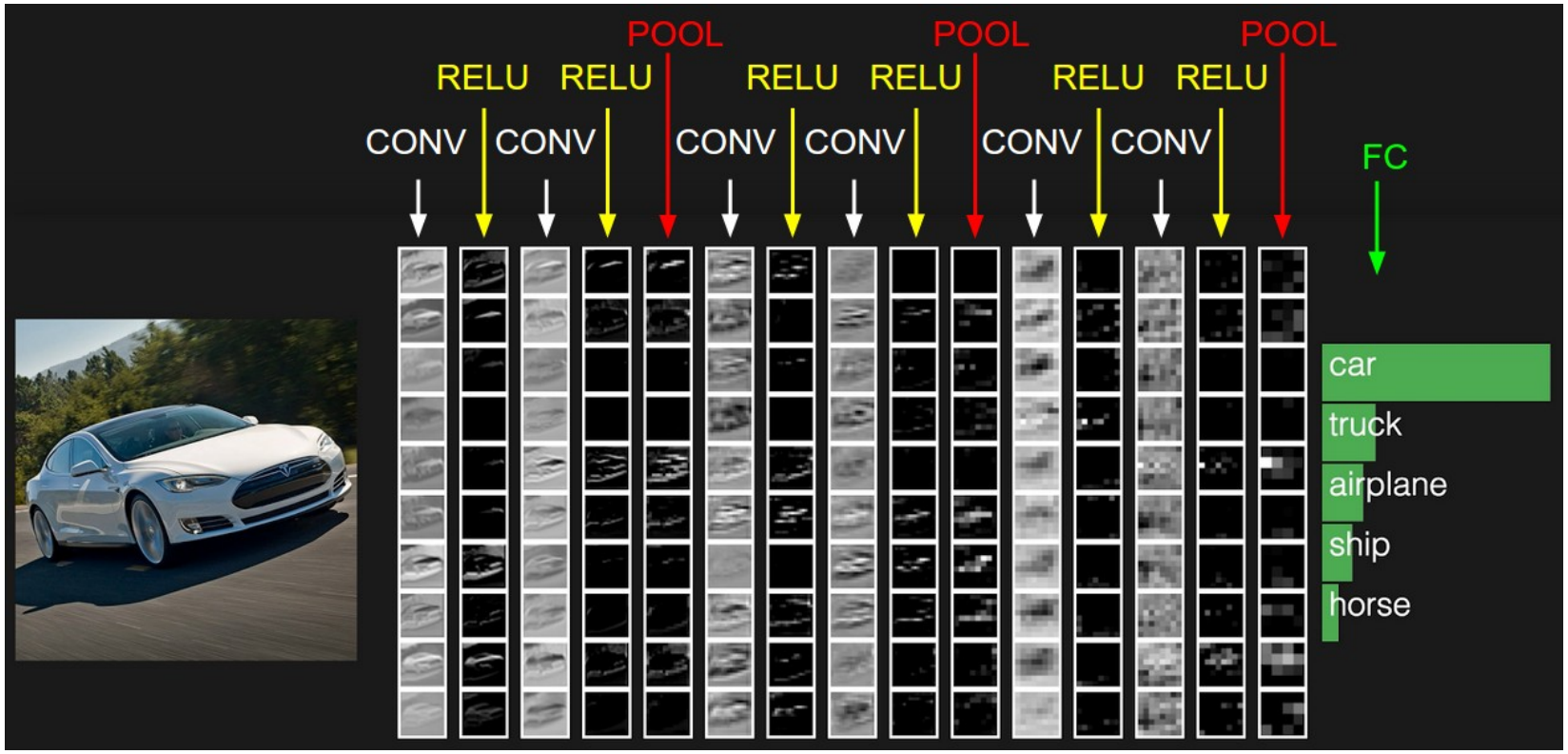
This talk

- First steps towards methodology to ensure safety of classification decisions
  - visible and human-recognisable perturbations: change of camera angle, lighting conditions, glare, snow, sign imperfections, ...
  - should not result in class changes
  - focus on individual decisions
  - images, but can be adapted to other types of problems
  - e.g. networks trained to produce justifications, in addition to classification (explainable AI)



- Search-based automated verification framework
  - tool DLV based on Satisfiability Modulo Theory
  - https://128.84.21.199/abs/1610.06940

Deep feed-forward neural network



Convolutional multi-layer network

http://cs231n.github.io/convolutional-networks/#conv

Problem setting

- Assume
  - vector spaces D_{L0}, D_{L1}, …, D_{Ln}, one for each layer
  - $f : D_{L0} \rightarrow \{c_1,\ldots c_k\}$ classifier function modelling human perception ability

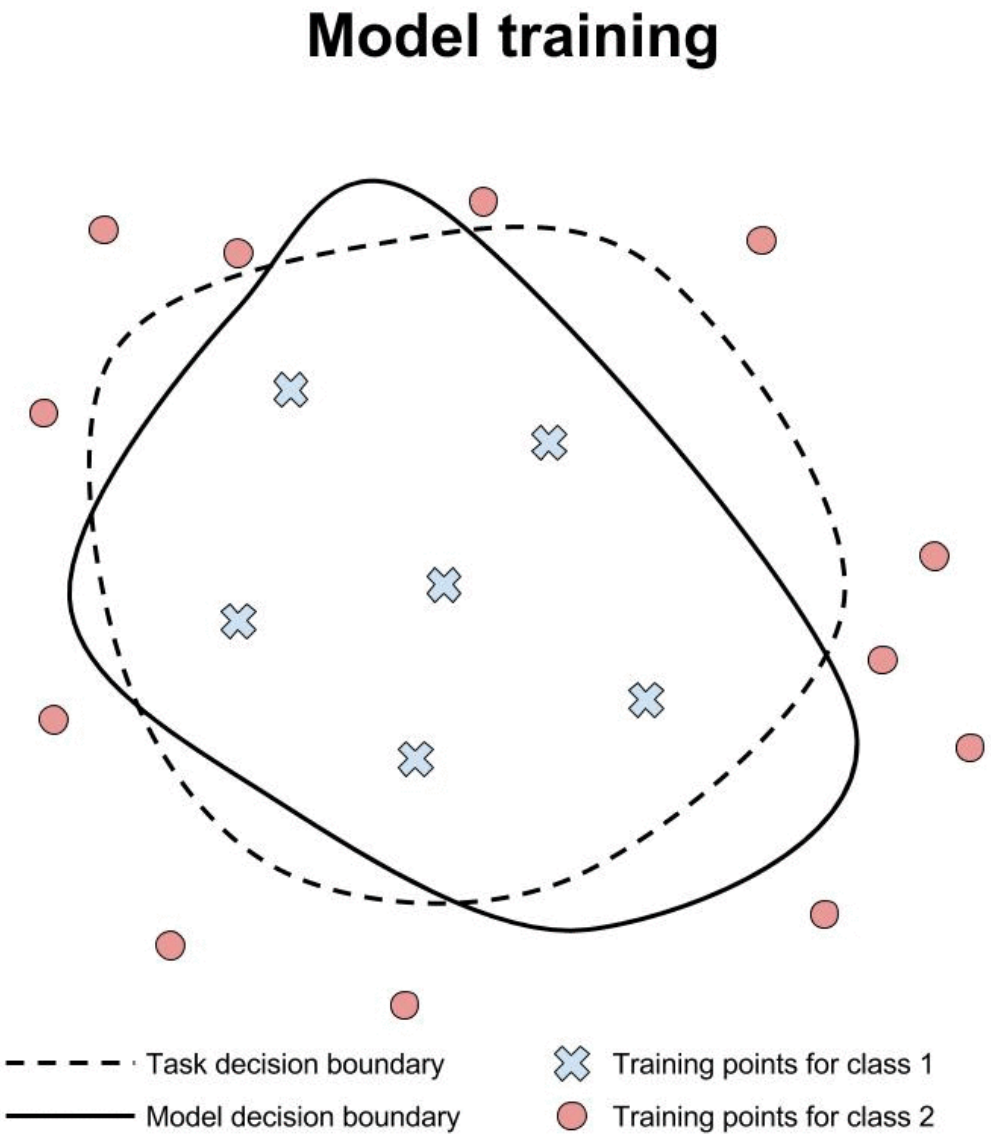
- The network $f' : D_{L0} \rightarrow \{c_1,\ldots c_k\}$ approximates f from M training examples $\{(x_i,c_i)\}_{i=1..M}$
  - built from activation functions ϕ_0, ϕ_1, …, ϕ_n, one for each layer
  - for point (image) $x \in D_{L0}$, its activation in layer k is

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

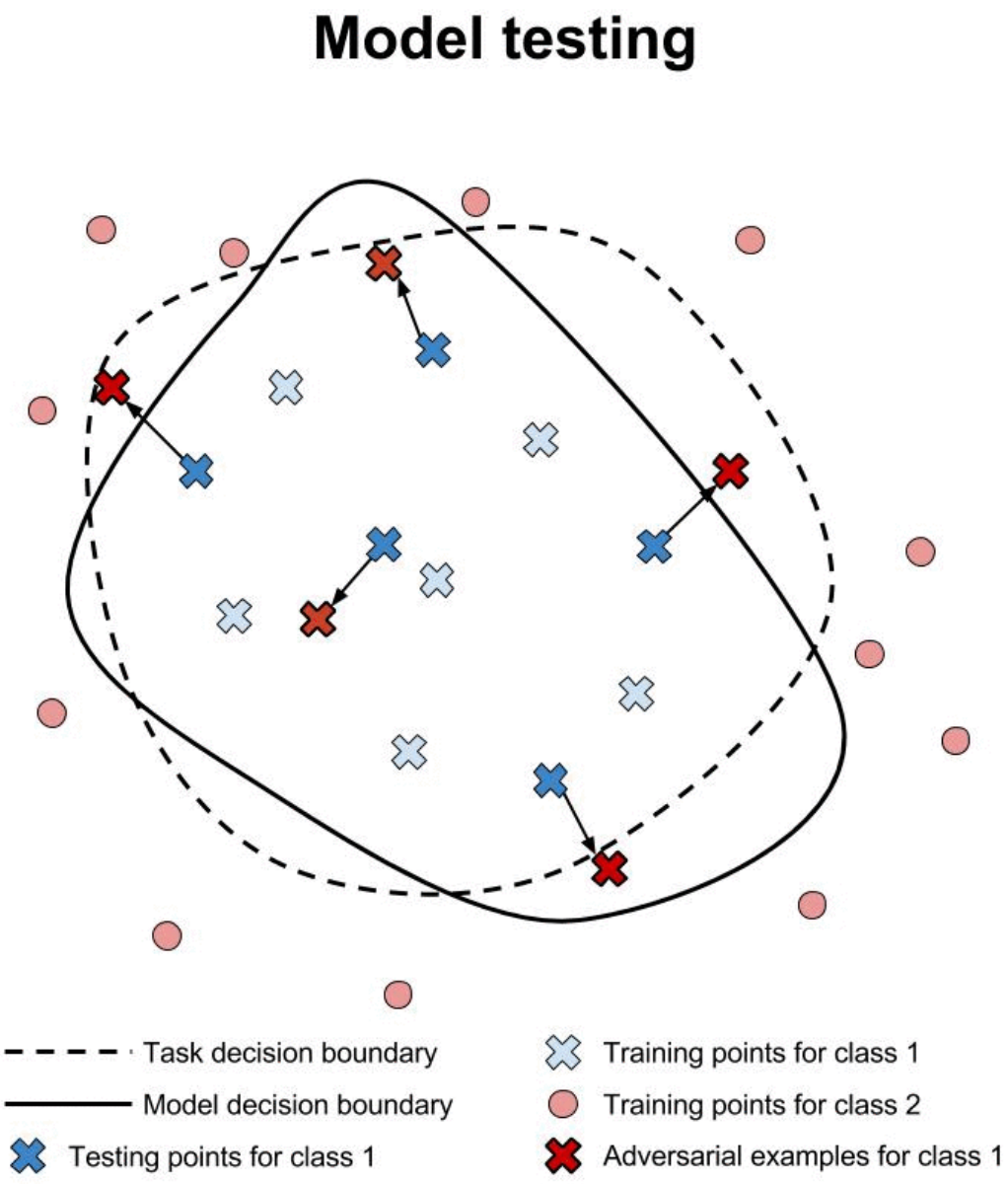
  - where $\phi_k(x) = \sigma(xW_k+b_k)$ and $\sigma(x) = \max(x,0)$
  - W_k learnable weights, b_k bias, σ is ReLU

- Notation
  - overload $\alpha_{x,n} = \alpha_{y,n}$ to mean x and y have the same class

Training vs testing



Training vs testing



Robustness

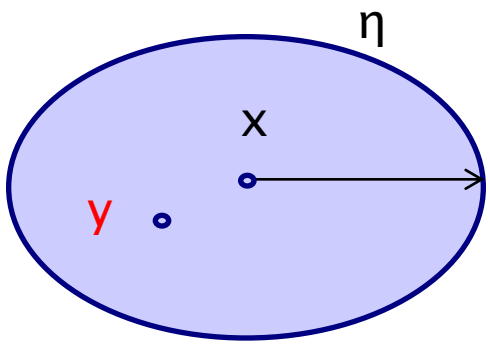
- Regularisation such as dropout improves smoothness
- Common smoothness assumption
  - each point $x \in D_{L0}$ in the input layer has a region η around it such that all points in η classify the same as x
- Pointwise robustness [Szegedy et al 2014]
  - f' is not robust at point x if $\exists y \in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

Verification for neural networks

- Little studied
- Reduction of safety to Boolean combination of linear arithmetic constraints [Pulina and Tachela 2010]
  - encode entire network using constraints
  - approximate the sigmoid using piecewise linear functions
  - SMT solving, does not scale (6 neurons, 3 hidden)
- Reluplex [Barrett et al 2017]
  - similar encoding but for ReLU, rather than sigmoid
  - generalise Simplex, SMT solver
  - more general properties
  - successful for end-to-end controller networks with 300 nodes

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…



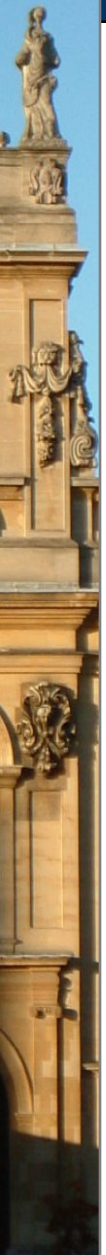
- But..
  - what diameter for region η?
  - which norm? L^2, L^{sup} ?
  - what is an acceptable/adversarial perturbation?

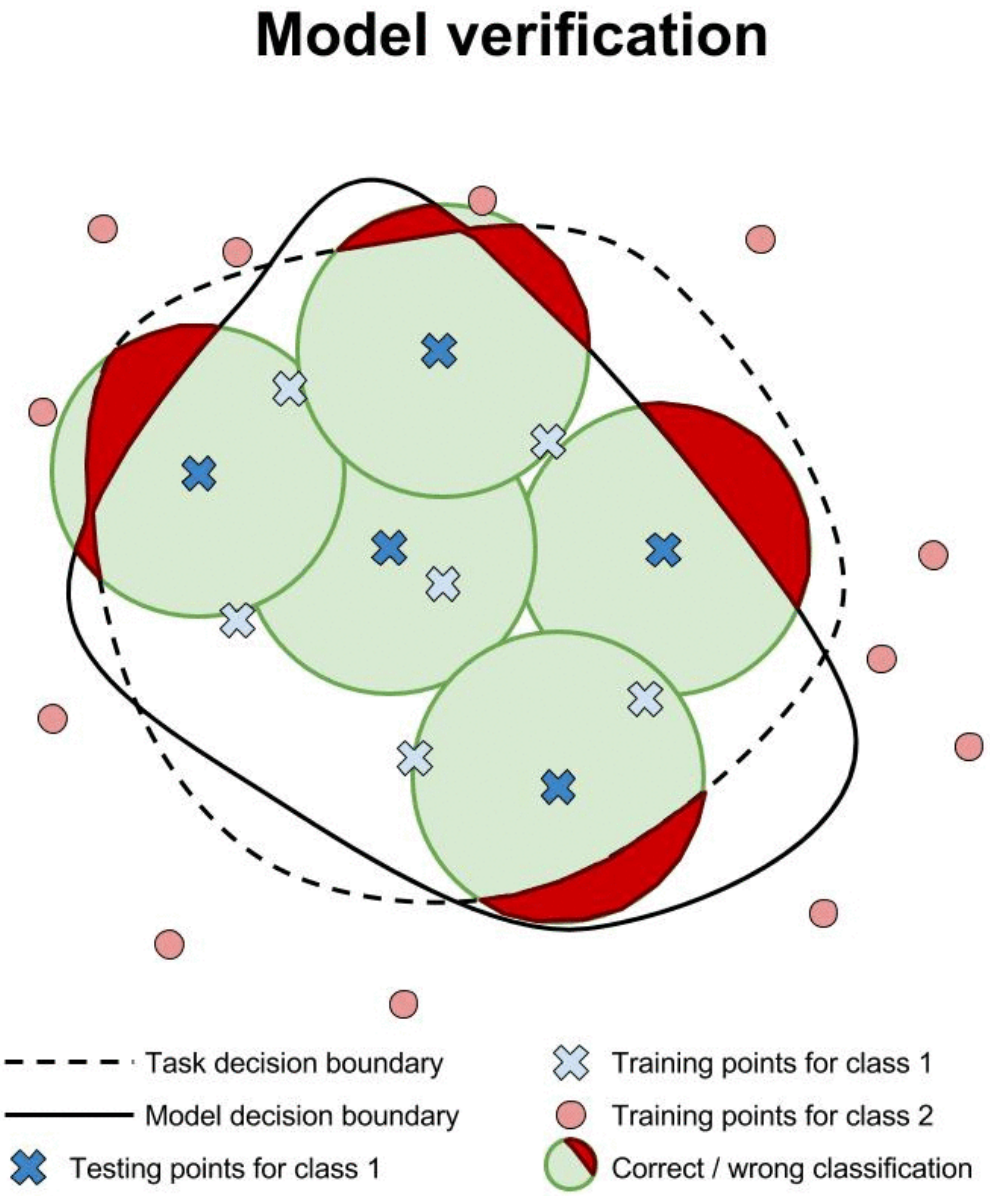
- Introduce the concept of manipulation, a family of operations that perturb an image
  - think of scratches, weather conditions, camera angle, etc
  - classification should be invariant wrt safe manipulations

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
  - guarantee of decision safety if adversarial example not found
- Falsification (= searching for adversarial examples)
  - good for attacks, no guarantees

Training vs testing vs verification

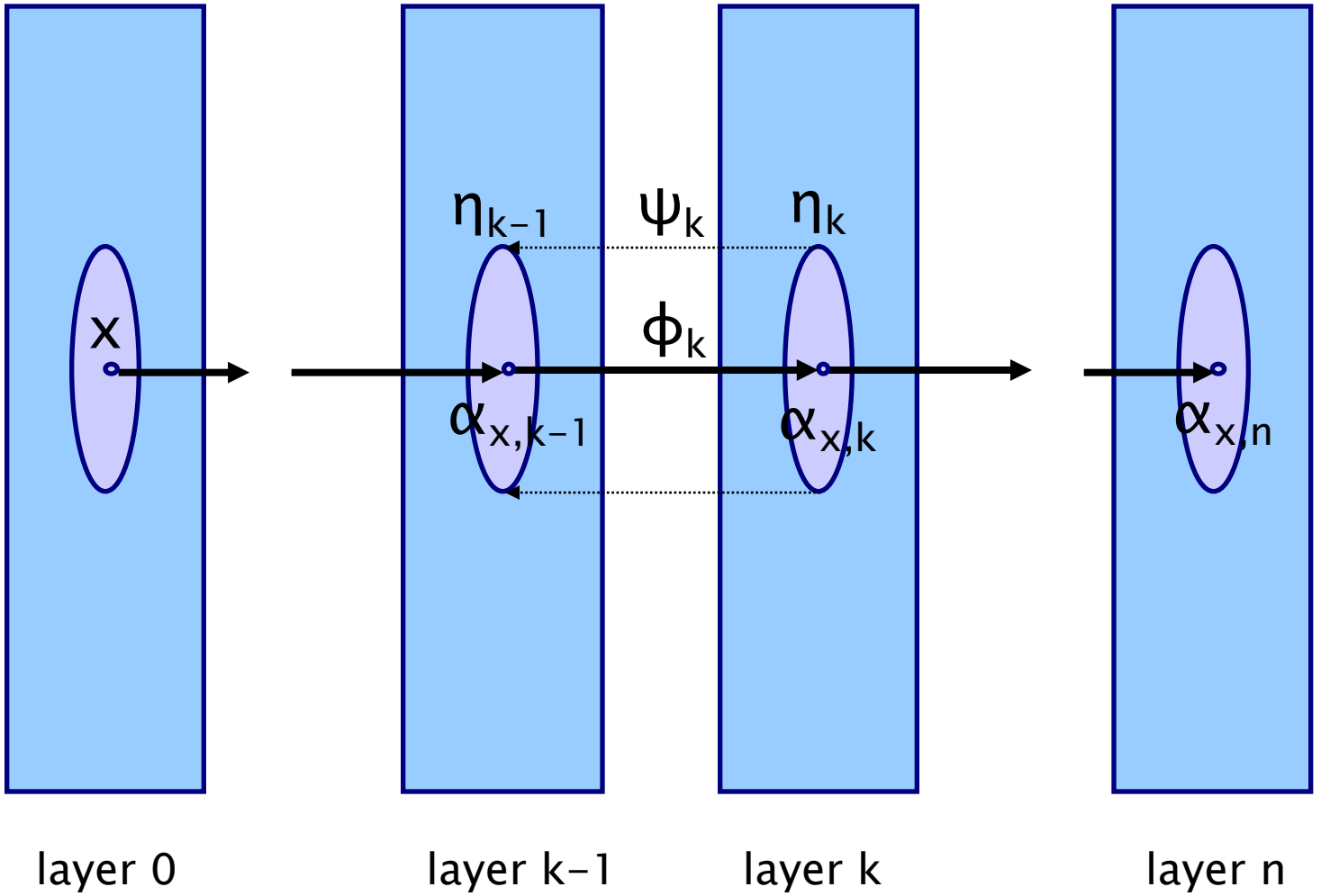




Verification framework

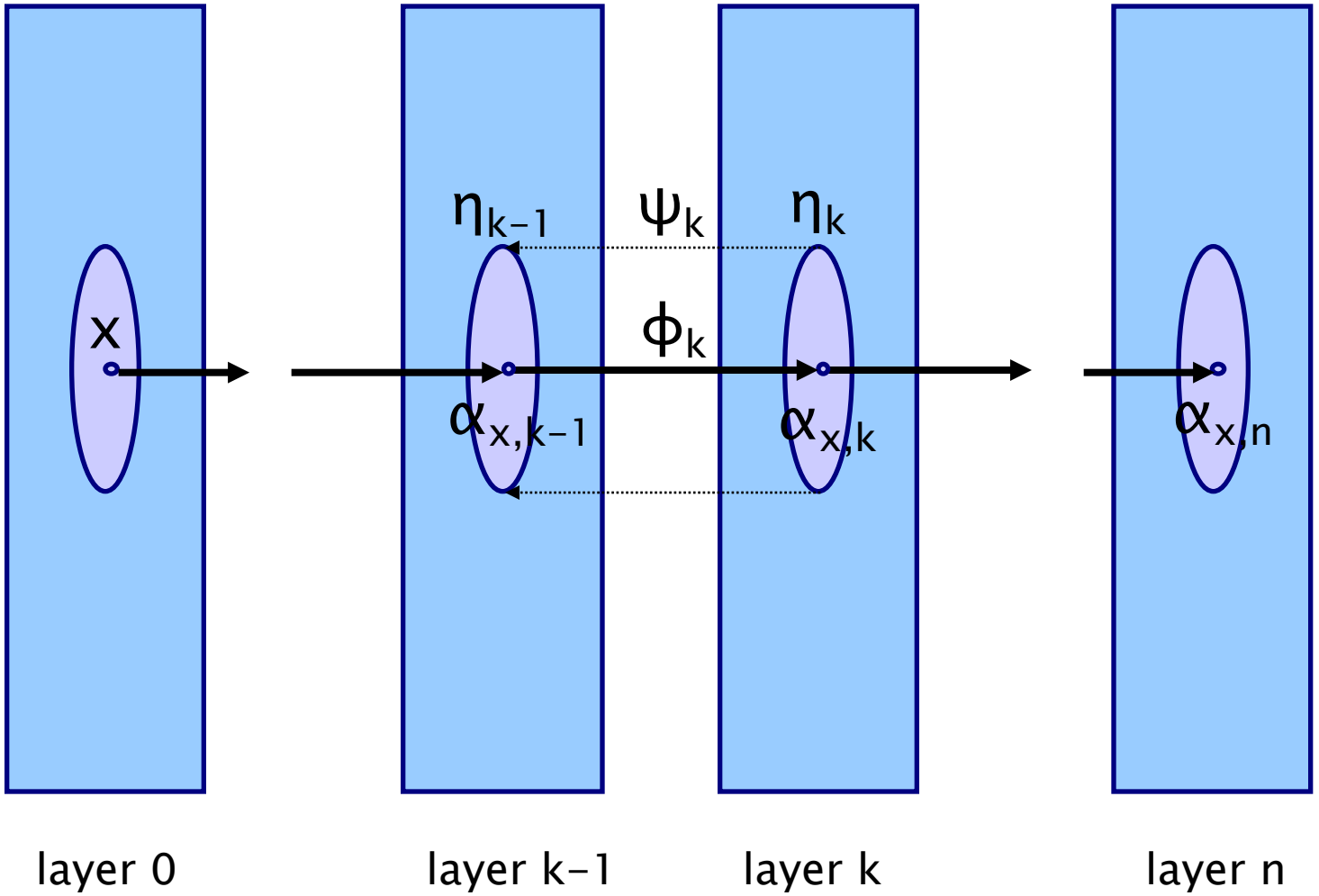
- Size of the network is prohibitive
  - millions of neurons!
- The crux of our approach
  - propagate verification layer by layer, i.e. need to assume for each activation $\alpha_{x,k}$ in layer k there is a region $\eta(\alpha_{x,k})$
  - dimensionality reduction by focusing on features
- This differs from heuristic search for adversarial examples
  - nonlinearity implies need for approximation using convex optimisation
  - no guarantee of precise adversarial examples
  - no guarantee of exhaustive search even if we iterate

Multi-layer (feed-forward) neural network



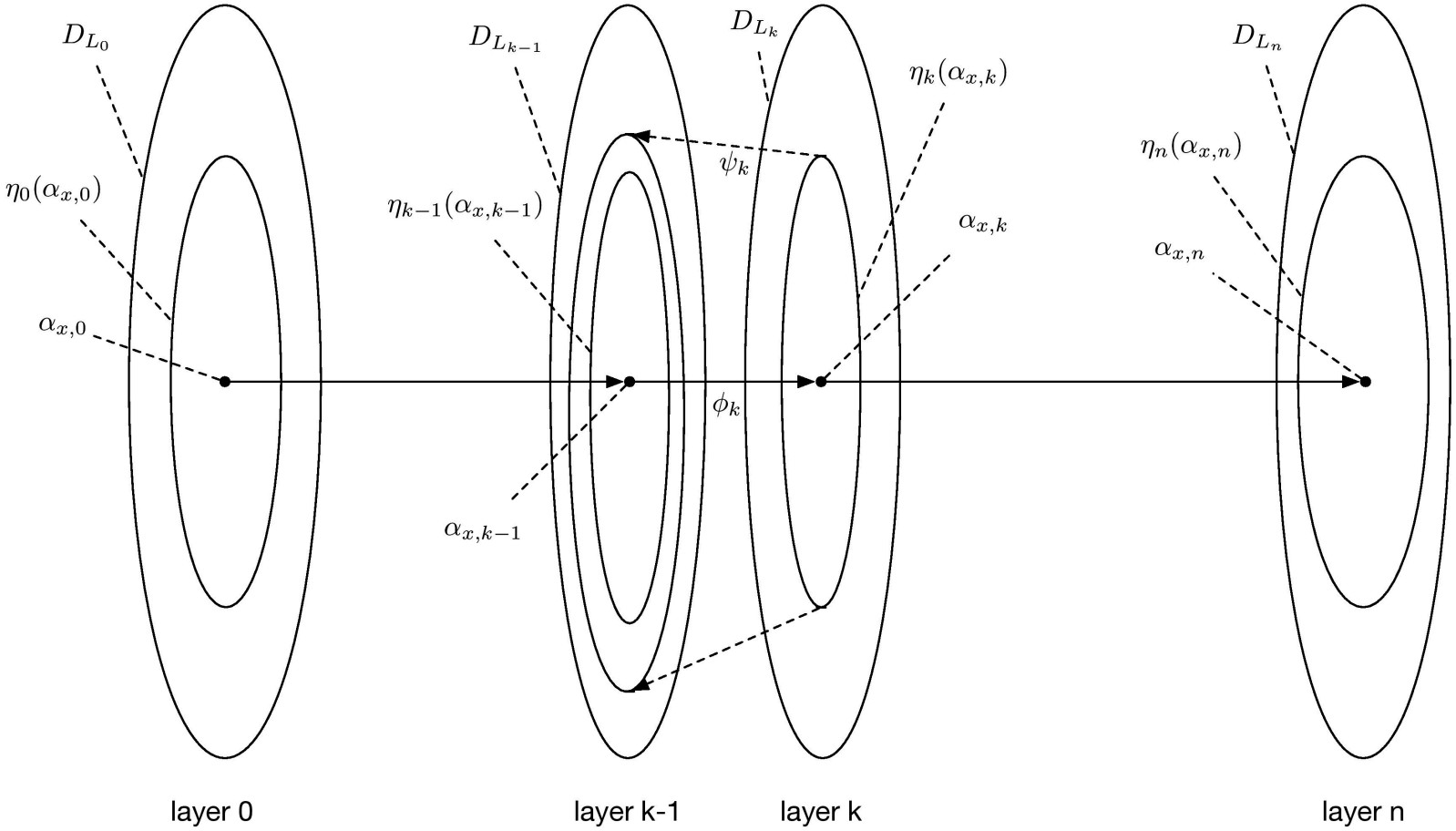
- Require mild conditions on region η_k and ψ_k mappings

Mapping forward and backward



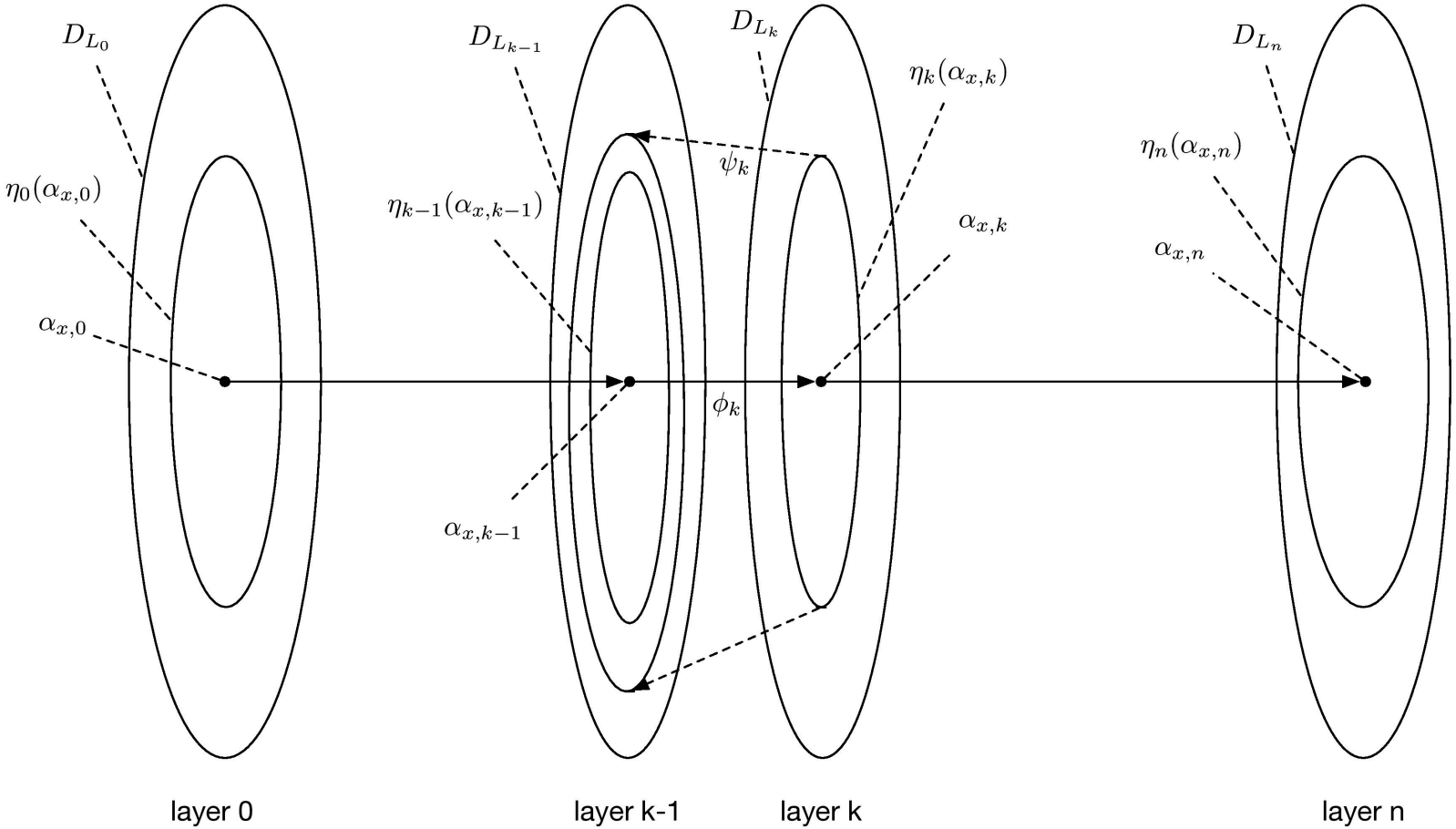
- Map region $\eta_k(\alpha_{x,k})$ forward via ϕ_k, backward via inverse ψ_k

Multi-layer (feed-forward) neural network



- Require mild conditions on region η_k and ψ_k mappings

Mapping forward and backward



- Can compute region $\eta_k(\alpha_{x,k})$ forward via activation function
- Back, use inverse of activation function

Manipulations

- Consider a family Δ_k of operators $\delta_k : D_{Lk} \rightarrow D_{Lk}$ that perturb activations in layer k, incl. input layer
  - think of scratches, weather conditions, camera angle, etc
  - classification should be invariant wrt such manipulations

- Intuitively, safety of network N at a point x wrt the region $\eta_k(\alpha_{x,k})$ and set of manipulations Δ_k means that perturbing activation $\alpha_{x,k}$ by manipulations from Δ_k will not result in a class change

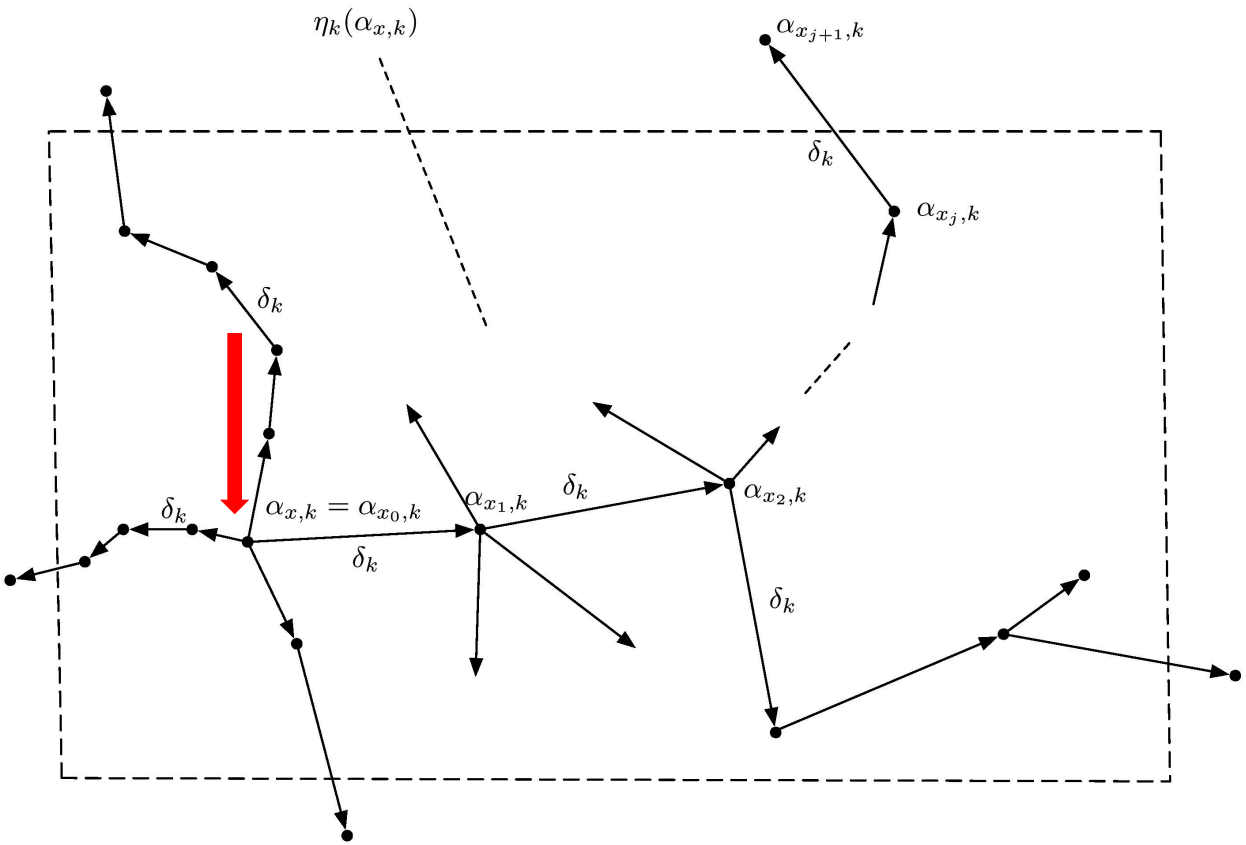
- Note that manipulations can be
  - defined by user and wrt different norms
  - made specific to each layer, and
  - applied directly on features, i.e. subsets of dimensions

Ensuring region coverage

- Fix point x and region $\eta_k(\alpha_{x,k})$
- Want to perform exhaustive search of the region for adversarial manipulations
  - if found, use to fine-tune the network and/or show to human tester
  - else, declare region safe wrt the specified manipulations

- Methodology: reduce to counting of misclassifications
  - discretise the region
  - cover the region with 'ladders' that are complete and covering
  - show 0-variation, i.e. explore nondeterministically and iteratively all paths in the tree of ladders, counting the number of misclassifications after applying manipulations
  - search is exhaustive under assumption of minimality of manipulations, e.g. unit steps

Covering region with ‘ladders’



- NB related work considers approximate, deterministic and non-iterative manipulations that are not covering
- Can search single or multiple paths (Monte Carlo tree search)

Layer-by-layer analysis

- In deep neural networks linearity increases with deeper layers
- Naïve search intractable: work with features

- Propagate analysis, starting from a given layer k:
- Determine region $\eta_k(\alpha_{x,k})$ from region $\eta_{k-1}(\alpha_{x,k-1})$
  - map forward using activation function
  - NB each activation at layer k arises from a subset of dimensions at layer k-1
  - check forward/backward mapping conditions (SMT-expressible)
- Refine manipulations in Δ_{k-1}, yielding Δ_k
  - consider more points as the analysis progresses into deeper layers
- If safety wrt $\eta_k(\alpha_{x,k})$ and Δ_k is verified, continue to layer k+1, else report adversarial example

Layer-by-layer analysis

- Framework ensures that safety wrt $\eta_k(\alpha_{x,k})$ and Δ_k implies safety wrt $\eta_{k-1}(\alpha_{x,k-1})$ and Δ_{k-1}
- If manipulations are minimal, then can deduce safety (= pointwise robustness) of the region at x
- But adversarial examples at layer k can be spurious, i.e. need to check if they are adversarial examples at the input layer
- NB employ various heuristics for scalability
  - explore manipulations of a subset of most extreme dimensions, which encode more explicit knowledge
  - employ additional precision parameter to avoid overly small spans

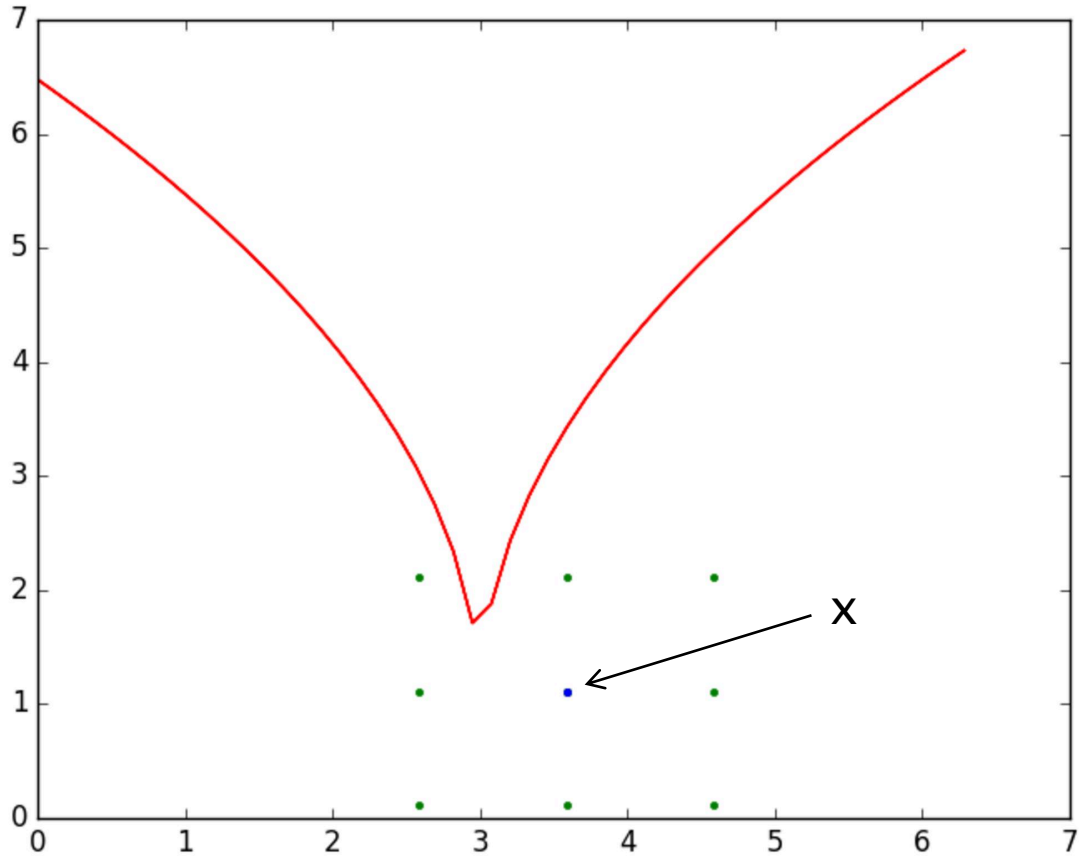
Features

- The layer-by-layer analysis is finite, but regions $\eta_k(\alpha_{x,k})$ are high-dimensional
  - exhaustive analysis impractical, need heuristics…
- We exploit decomposition into features, assuming their independence and low-dimensionality
  - natural images form high-dimensional tangled manifold, which embeds tangled manifolds that represent features
  - classification separates these manifolds
- By assuming independence of features, reduce problem of size $O(2^{d1+..+dn})$ to set of smaller problems $O(2^{d1}),…O(2^{dn})$
  - e.g. compute regions and 0-variation wrt to features
  - analysis discovers features automatically through hidden layer analysis

Implementation

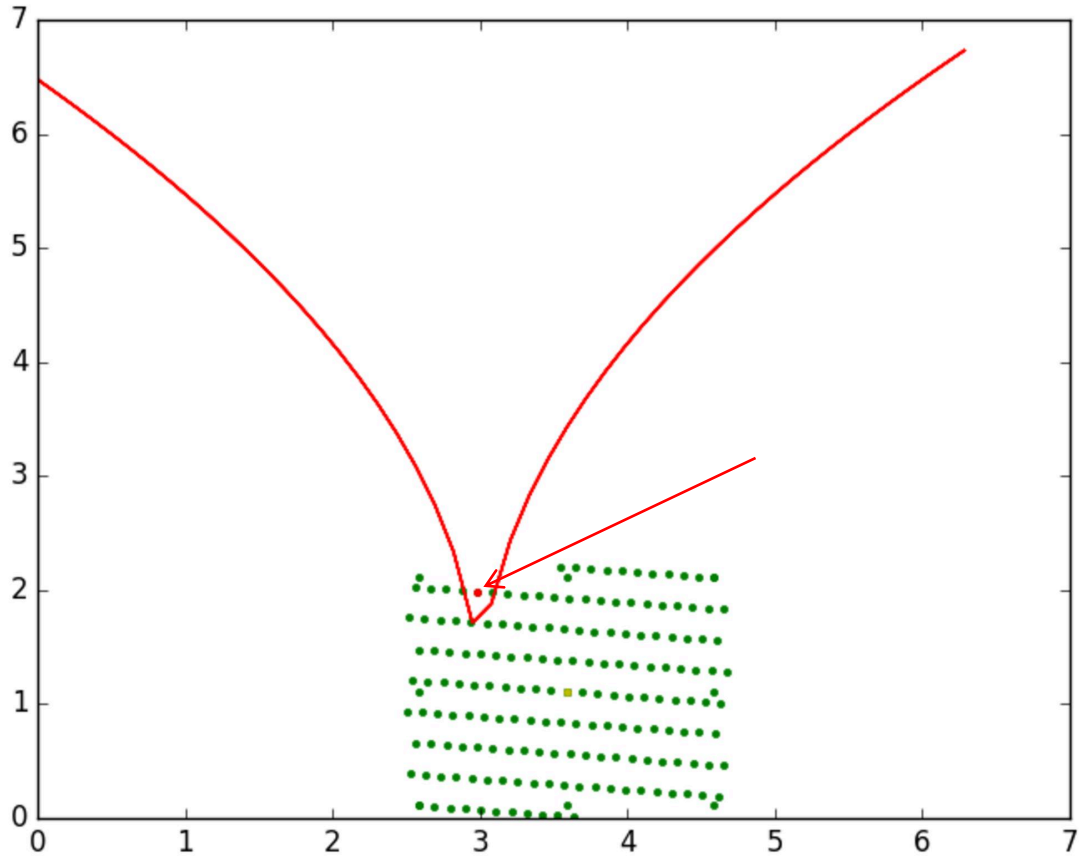
- Implement the techniques using SMT (Z3)
  - for layer-by-layer analysis, use linear real arithmetic with existential and universal quantification
  - within the layer (0-variation), use as above but without universal quantification
  - work with Euclidean and Manhattan norms, can be adapted to other norms
- We work with one point/decision at a time, rather than activation functions, but computation is exact
  - avoid approximating sigmoid (not scalable) [Pulina et al 2010]
  - more scalable than approximating ReLU by LP [Bastani et al 2016] or Reluplex [Barrett et al 2017]
- Main challenge: how to define meaningful regions and manipulations
  - but adversarial examples can be found quickly

Example: input layer



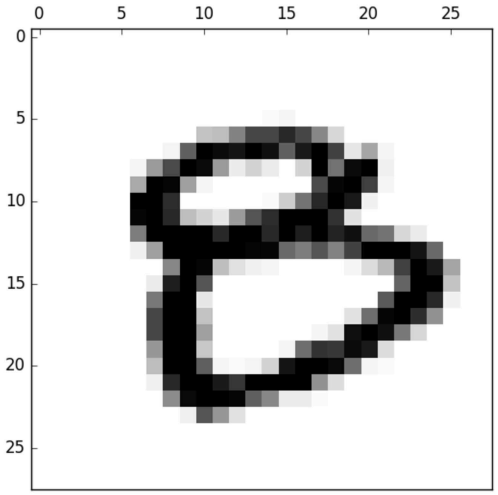
- Small point classification network, 8 manipulations

Example: 1st hidden layer

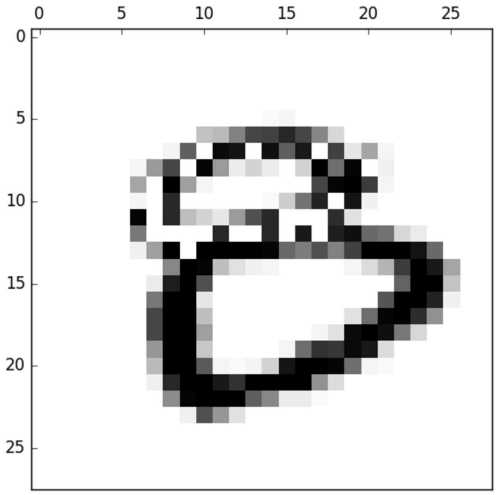


- Refined manipulations, adversarial example found

MNIST example



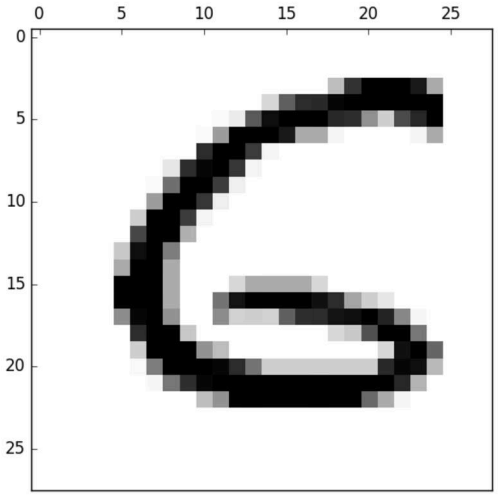
8



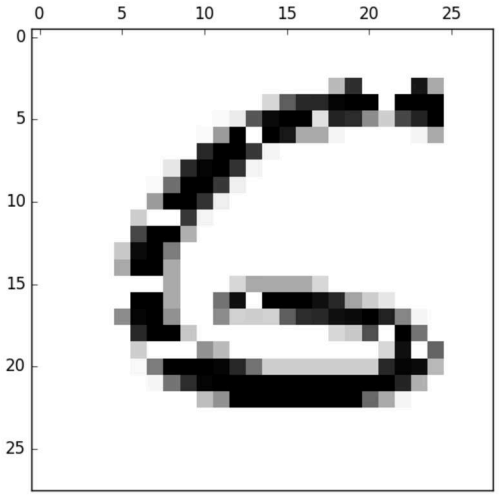
0

- 28x28 image size, one channel, medium size network (12 layers, Conv, ReLU, FC and softmax)

Another MNIST example



6



5

- 28x28 image size, one channel, medium size network (12 layers, Conv, ReLU, FC and softmax)

Compare to existing methods

- Search for adversarial perturbations only (=falsification)
- FGSM [Goodfellow et al 2014]
  - calculates optimal attack for a linear approximation of network cost, for a set of images
  - deterministic, iterative manipulations
- JSMA [Papernot et al 2015]
  - finds subset of dimensions to manipulate (in the input layer)
  - manipulates according to partial derivatives
- DLV (this talk)
  - explores proportion of dimensions in input and hidden layers
  - so manipulates over features discovered in hidden layers

Falsification comparison

- DLV able to find examples with smaller average distance than JSMA, at comparable performance (may affect transferability)
- FGSM fastest per image
- For high success rates (approx 98%) JSMA has smallest average distance, followed by DLV, followed by FGSM

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

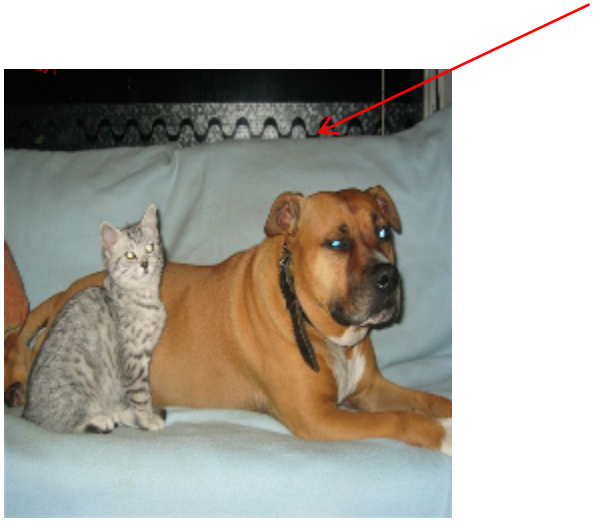
- 224x224 image size, 3 channels, 16 layers, state-of-the-art network VGG, (Conv, ReLU, Pool, FC, zero padding, dropout and softmax)
- Work with 20,000 dimensions (of 3m), unsafe for 2nd layer

ImageNet example

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

Another ImageNet example

Boxer



Rhodesian ridgeback

- 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

Yet another ImageNet example

Labrador retriever

Lifeboat

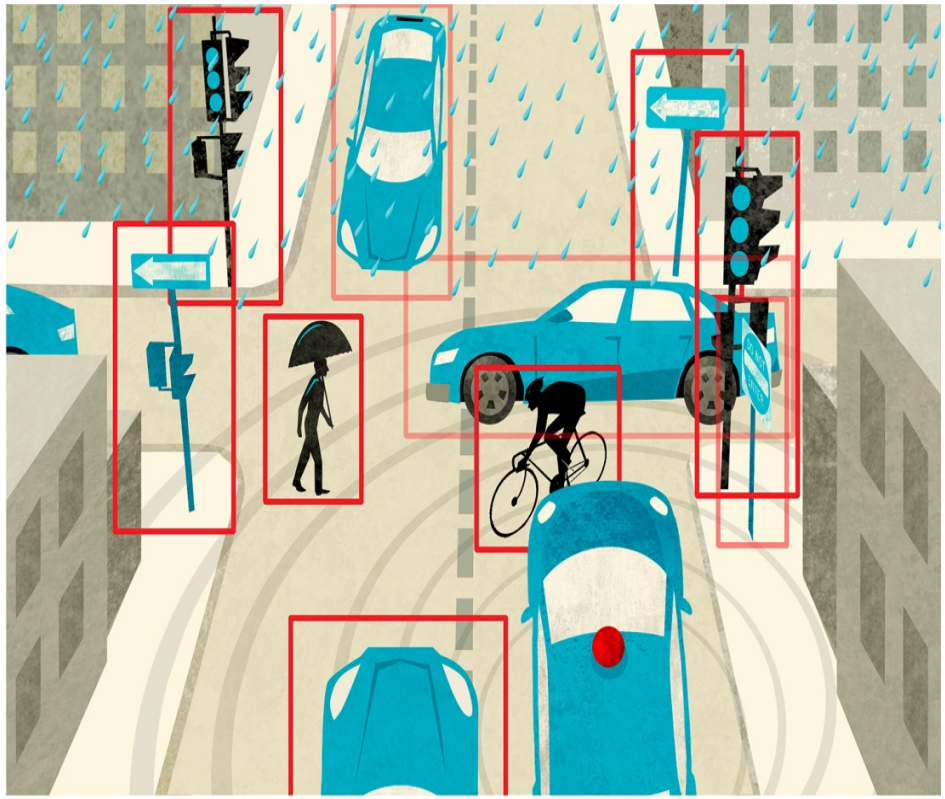
- 224x224 image size, 3 channels, 16 layers, state-of-the-art network, (Conv, ReLU, Pool, FC, zero padding, dropout and softmax)
- Work with 20,000 dimensions

Conclusion

- First framework for safety verification of deep neural network classifiers formulated and implemented
- Method
  - safety parameterised by region and set of manipulations
  - based on exhaustive search, akin to explicit model checking
  - propagation of analysis into deeper layers
  - heuristics to improve scalability
  - adversarial examples found quickly
- Future work
  - simplistic, can we use operator or statistical properties?
  - how best to use adversarial examples: training vs logic
  - symbolic methods?
  - abstraction-refinement?
  - more complex properties?

AI safety – challenge for verification?

- Complex scenarios
  - goals
  - perception
  - situation awareness
  - context (social, regulatory)
- Safety-critical, so guarantees needed
- Should failure occur, accountability needs to be established



Reasoning about cognitive trust

Over-trust and inattention are known problems that technology developers need to **design for**, and simply telling customers not to do what comes naturally is probably not enough.

Patrick Lin

- Formulate a theory for expressing and reasoning about social trust in human-robot collaboration/competition
- Develop tools for trust evaluation to aid design and analysis of human-robot systems

Quantitative verification for trust?

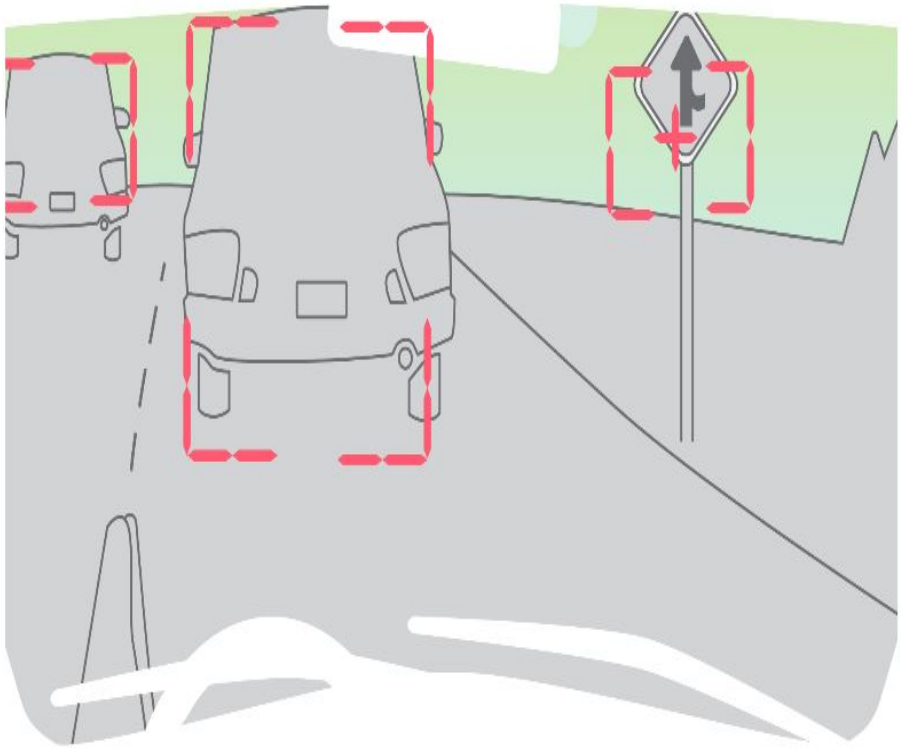
- Logic PRTL* undecidable in general
- Have identified decidable fragments (EXPTIME, PSPACE, PTIME), by restricting the expressiveness of the logic and the stochastic multiagent systems
- Reasoning about trust can be used
  - in decision-making for robots
  - to justify and explain trust-based decisions, also for humans
  - to infer accountability for failures

- Next step is to develop model checking for trust…

- But many challenges remain!

Morality, ethics and social norms

- Already merging into traffic proving difficult, what about social subtleties?
- What to do in emergency?
  - moral decisions
  - enforcement
  - conflict resolution
  - handover in semi-autonomous driving
- Obey traffic rules
  - cultural dependency



A car is merging on the freeway. How fast should it drive?

http://www.pbs.org/wgbh/nova/next/tech/robot-morals/

Acknowledgements

- My group and collaborators in this work
- Project funding
  - ERC Advanced Grant
  - EPSRC Mobile Autonomy Programme Grant
  - Oxford Martin School, Institute for the Future of Computing

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


FLoC
2018
DEPARTMENT OF
COMPUTER
SCIENCE
UNIVERSITY OF
OXFORD
06–19 JULY 2018
OXFORD, UK
FEDERATED LOGIC
CONFERENCE
2018