Perspectives on Neural Network Verification

Patrick Musau

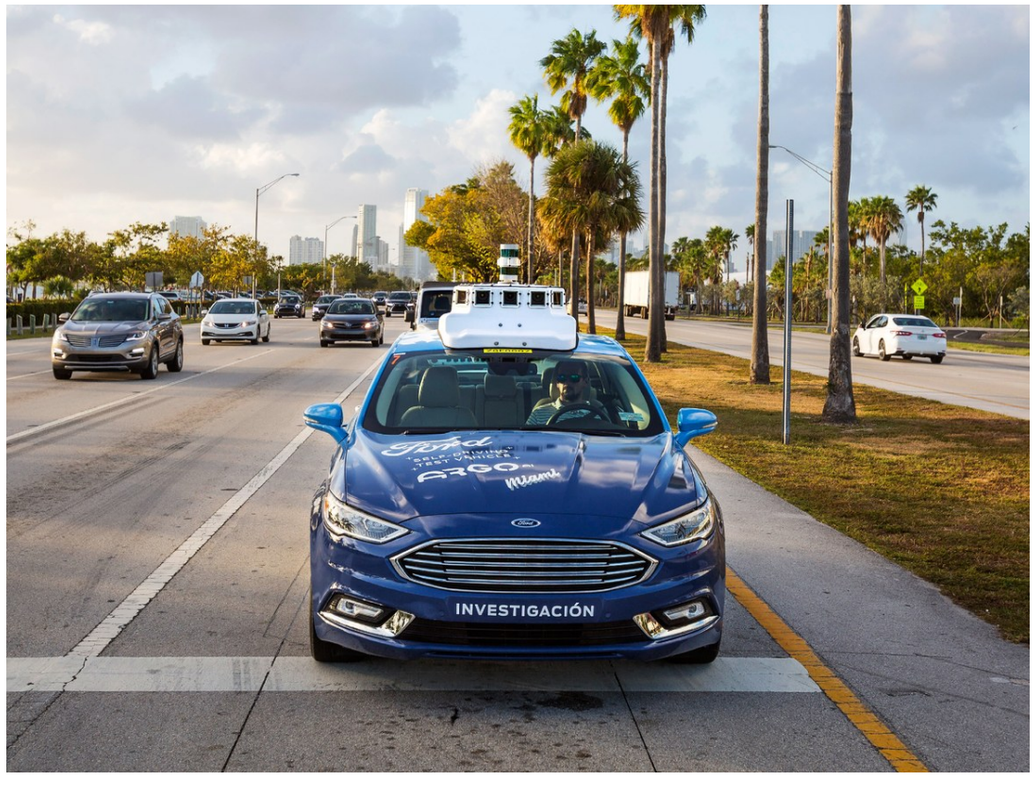
Department of Electrical Engineering and Computer Science
Vanderbilt University

October 30, 2018

Context and Origins

Artificial Neural Networks have demonstrated remarkable utility in numerous application domains including:

- Autonomous Vehicles
- Image and Pattern Recognition
- Machine Translation
- Non-linear system identification and control
- Natural Language Processing



Uber Self-Driving Car (Marshall, 2018 [1])

Adversarial Examples [3]

- Susceptible to errant behavior via a slight perturbation to their inputs
  - These inputs are known as *Adversarial Examples*
  - Can be tailored to cause a misclassification to a specific class

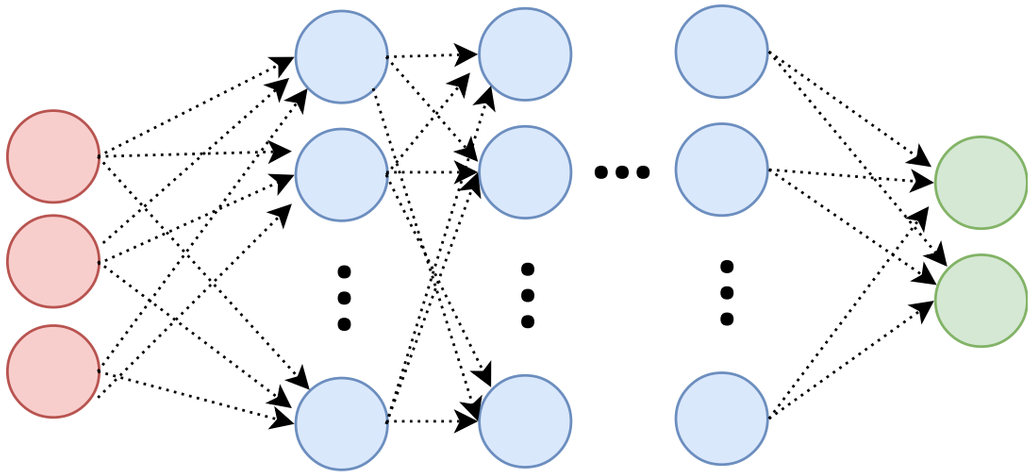
Adversarial Example discovered for self-driving car applications (Pei et al. 2018 [2])

Impersonation Attacks, Shariff et al.

- Sharif et al. fooled a facial recognition system using a printed a pair of eyeglass frames
- Students were able to impersonate celebrities as well as evade being recongnized

Physical impersonation attacks designed for state-of-the-art facial biometric systems. (Sharif, 2016 [4])

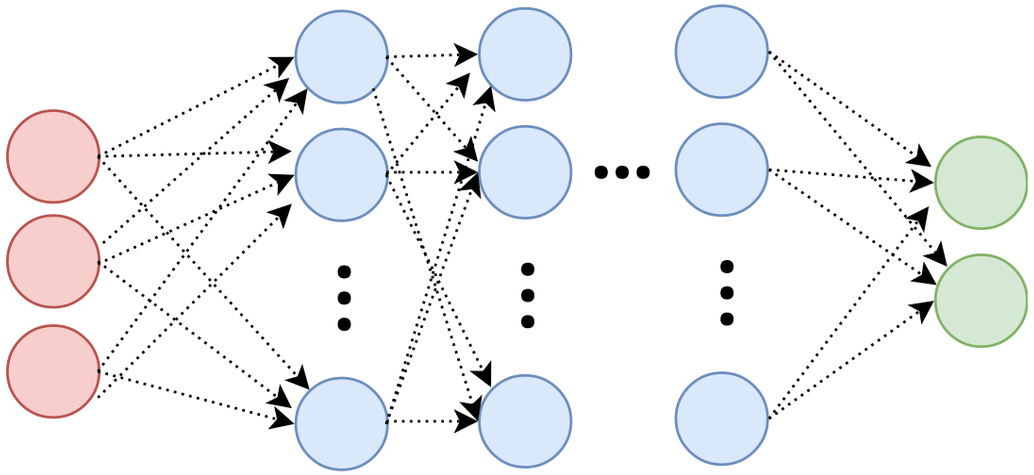
Verification Challenge



- Given a neural network, N and a property P, we wish to demonstrate:

$$N \models P$$

Verification Challenge

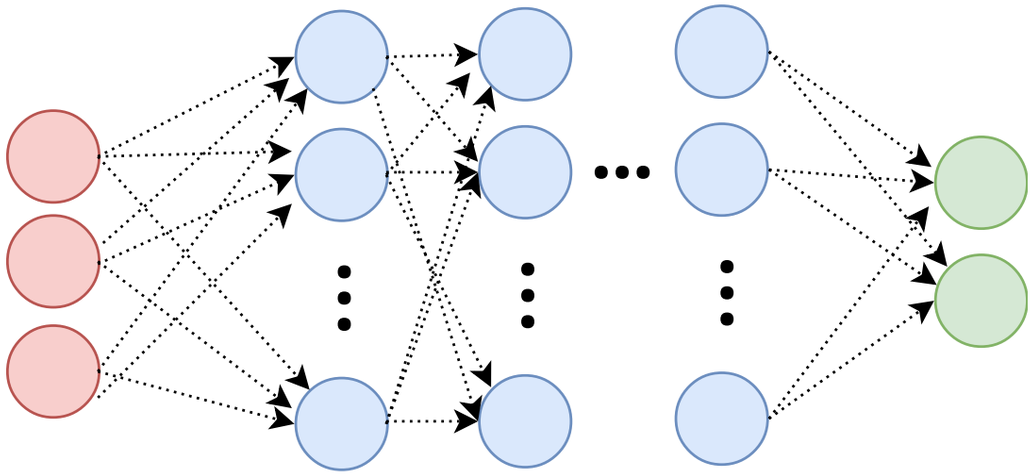


- Given a neural network, N and a property P, we wish to demonstrate:

$$N \models P$$

- Otherwise demonstrate why $N \not\models P$ by generating a counterexample

Verification Challenge

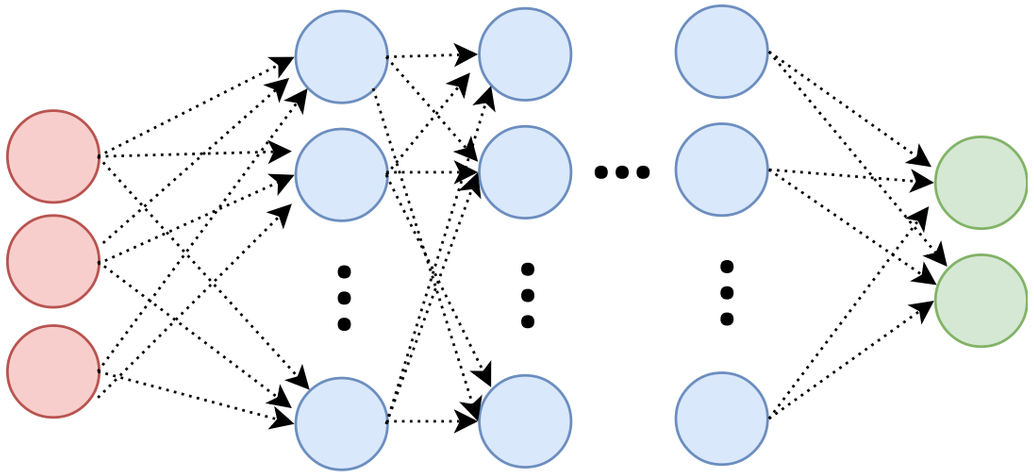


- Given a neural network, N and a property P, we wish to demonstrate:

$$N \models P$$

- Otherwise demonstrate why $N \not\models P$ by generating a counterexample
- Notoriously difficult problem

Verification Challenge

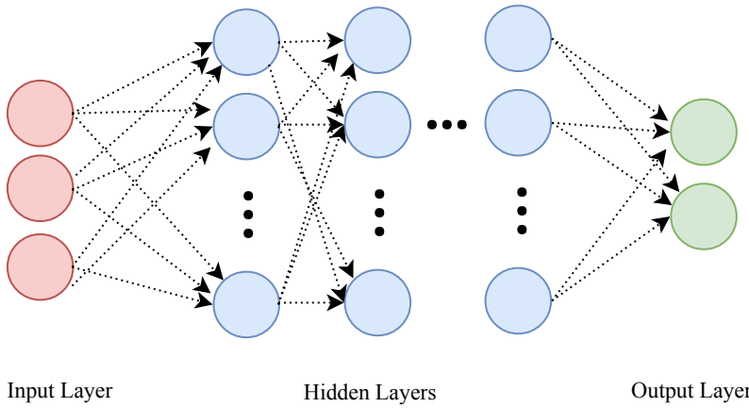


- Given a neural network, N and a property P, we wish to demonstrate:

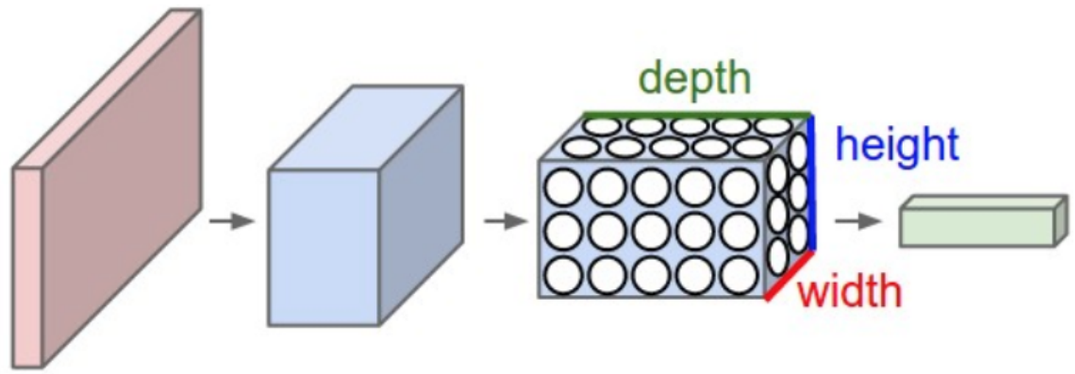
$$N \models P$$

- Otherwise demonstrate why $N \not\models P$ by generating a counterexample
- Notoriously difficult problem
  - Verifying even simple properties is NP Complete [5]

Neural Network Preliminaries



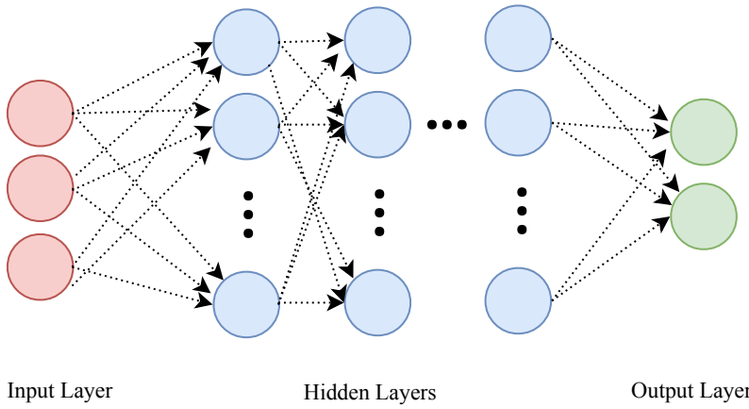
Feed-forward Architecture. (Karpathy, 2017 [6])



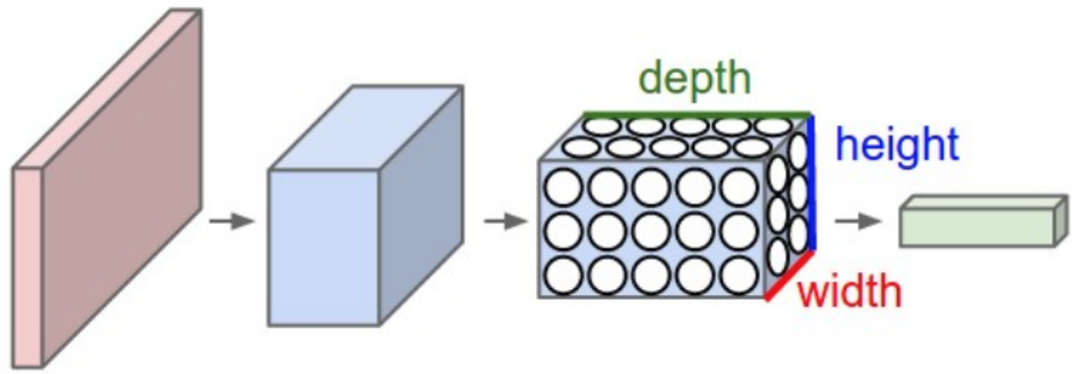
Convolutional Architecture. (Karpathy, 2017 [6])

- Organized into three types of layers

Neural Network Preliminaries



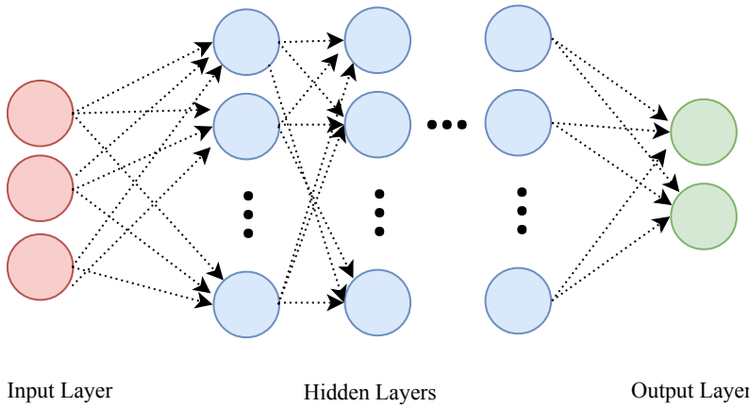
Feed-forward Architecture. (Karpathy, 2017 [6])



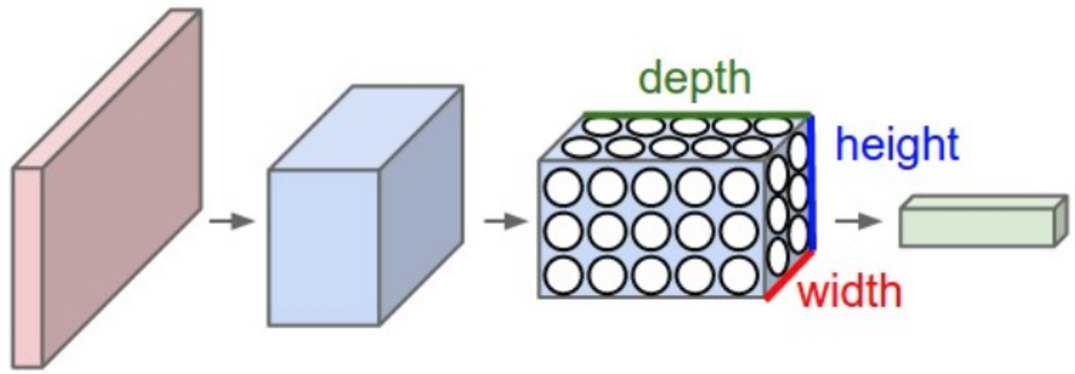
Convolutional Architecture. (Karpathy, 2017 [6])

- Organized into three types of layers
- Variety of different architectures
  - Feed-forward, Convolutional

Neural Network Preliminaries



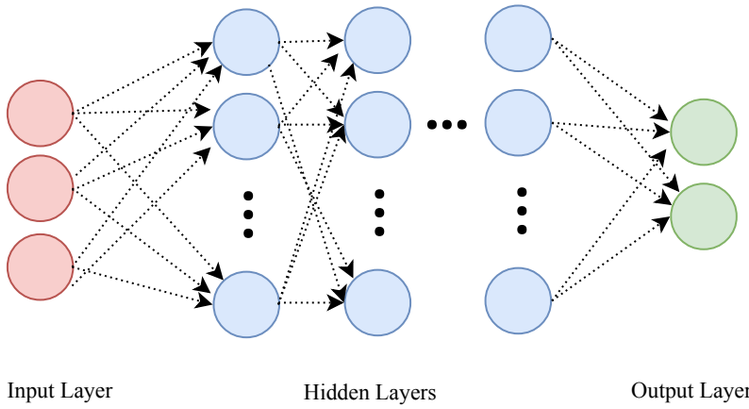
Feed-forward Architecture. (Karpathy, 2017 [6])



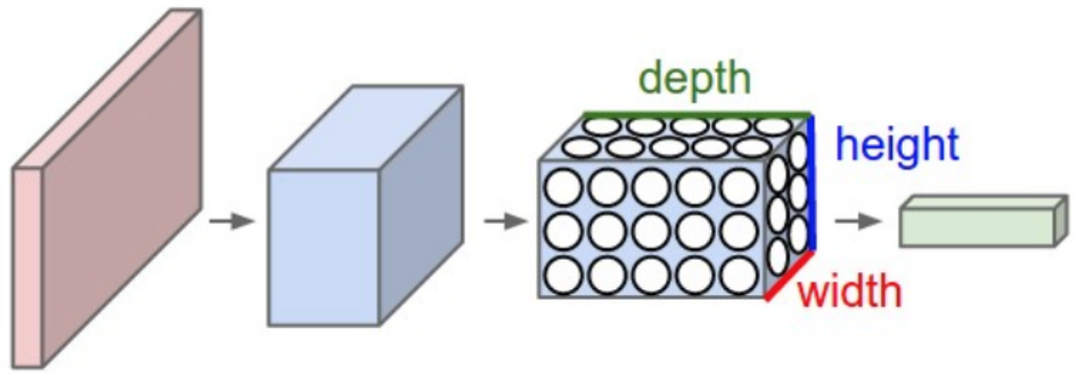
Convolutional Architecture. (Karpathy, 2017 [6])

- Organized into three types of layers
- Variety of different architectures
  - Feed-forward, Convolutional
- Each connection is assigned a weight
  - weights define the network's behavior and are determined during a training phase

Neural Network Preliminaries



Feed-forward Architecture. (Karpathy, 2017 [6])



Convolutional Architecture. (Karpathy, 2017 [6])

- Organized into three types of layers
- Variety of different architectures
  - Feed-forward, Convolutional
- Each connection is assigned a weight
  - weights define the network's behavior and are determined during a training phase
- Trained on finite set of inputs expected to generalize

Neural Network Preliminaries

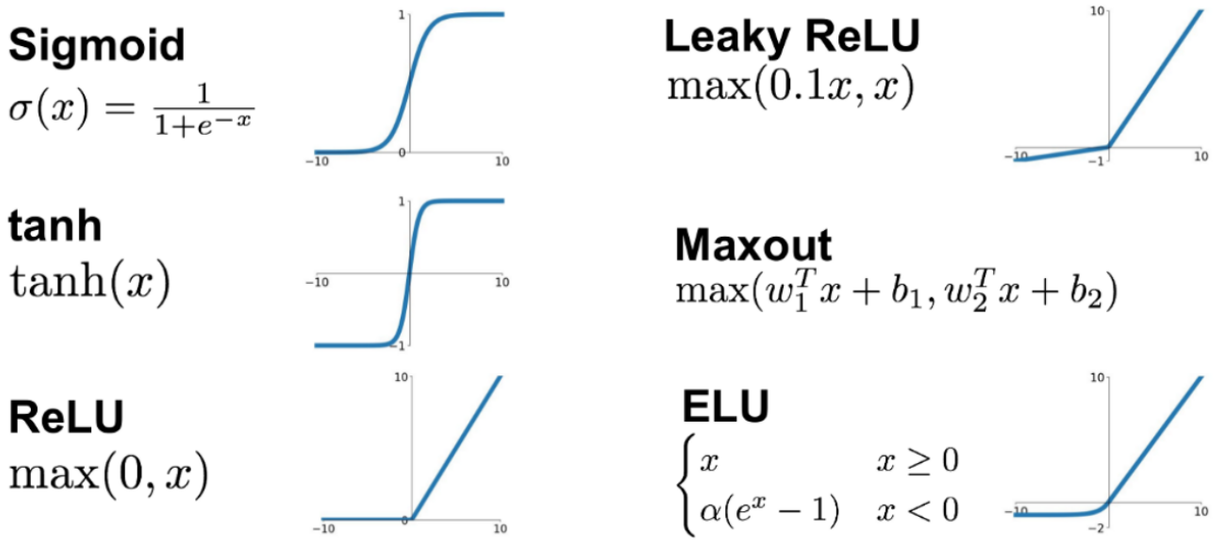
- Output of the the network is determined layer by layer

Neural Network Preliminaries

- Output of the the network is determined layer by layer
  - Each layer's output is determined from its predecessor, using the edge weights and a non-linear activation function

Neural Network Preliminaries

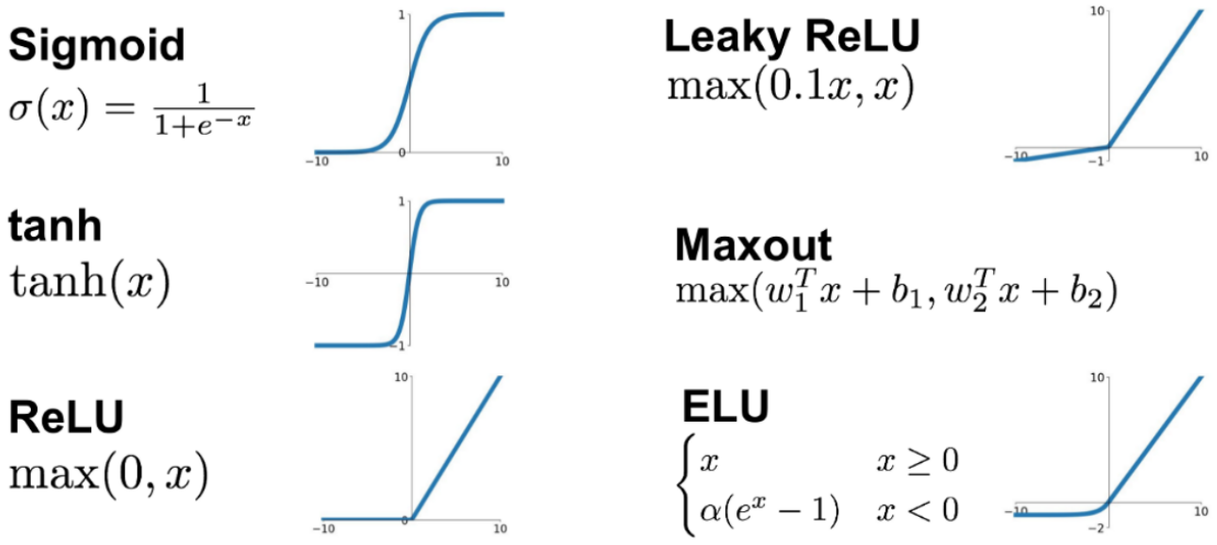
- Output of the the network is determined layer by layer
  - Each layer's output is determined from its predecessor, using the edge weights and a non-linear activation function
- Activation Functions introduce non-linearities that allow neural networks to learn complex behavior



Various Activation Functions. (Jadon, 2018 [7])

Neural Network Preliminaries

- Output of the the network is determined layer by layer
  - Each layer's output is determined from its predecessor, using the edge weights and a non-linear activation function
- Activation Functions introduce non-linearities that allow neural networks to learn complex behavior
- Can be viewed as a function, $v : I^n \rightarrow O^m$, mapping an n-dimensional input, I^n, to an m-dimensional output, O^m



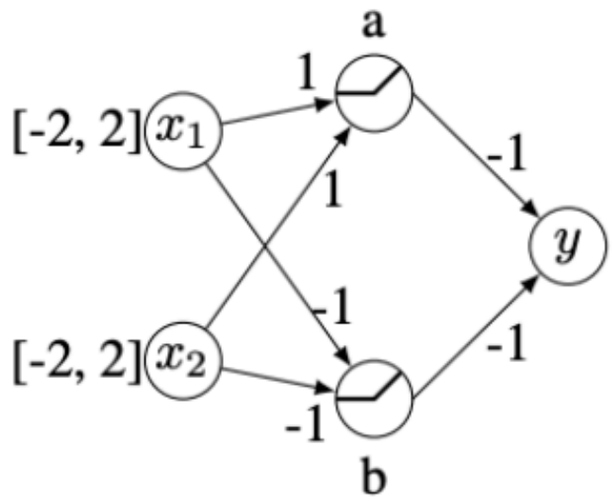
Various Activation Functions. (Jadon, 2018 [7])

Neural Network Verification [8]

- Given an input domain, C, and a predicate, P, we wish to prove:

$$\forall x_1 \in C,\ u_1 = v(x_1) \Rightarrow P(u_1)$$

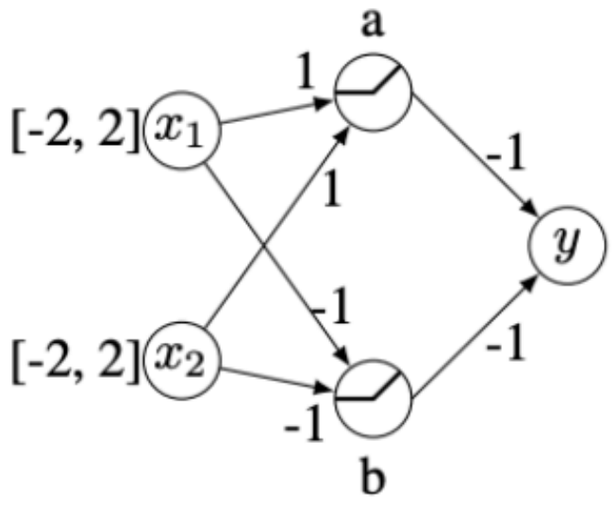
- Does there exist an example x_1 such that $\neg P(u_1)$
  - If we cannot find such an input we can conclude $N \models P$
  - Otherwise $N \not\models P$ and x_1 is a counterexample



Prove that $y > -5$

Example Neural Network Verification Problem. (Bunel, 2018 [8])

Encoding The Network [8]



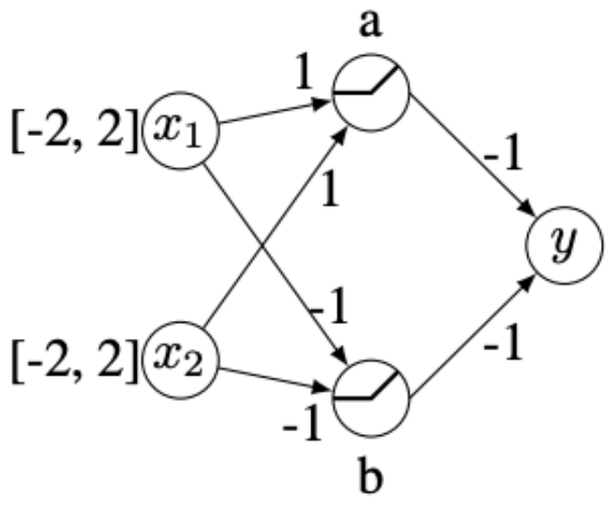
Prove that $y > -5$

Example Neural Network Verification Problem. (Bunel, 2018 [8])

- Goal: Find $\mathbf{x} \in C = [-2, 2] \times [-2, 2]$ such that $y \leq -5$

$$-2 \leq x_1 \leq 2 \qquad -2 \leq x_2 \leq 2$$
$$v_1 = x_1 + x_2 \qquad v_2 = -x_1 - x_2$$
$$a = \max(v_1, 0) \quad b = \max(v_2, 0)$$
$$y = -a - b$$

Decision Procedures [9]

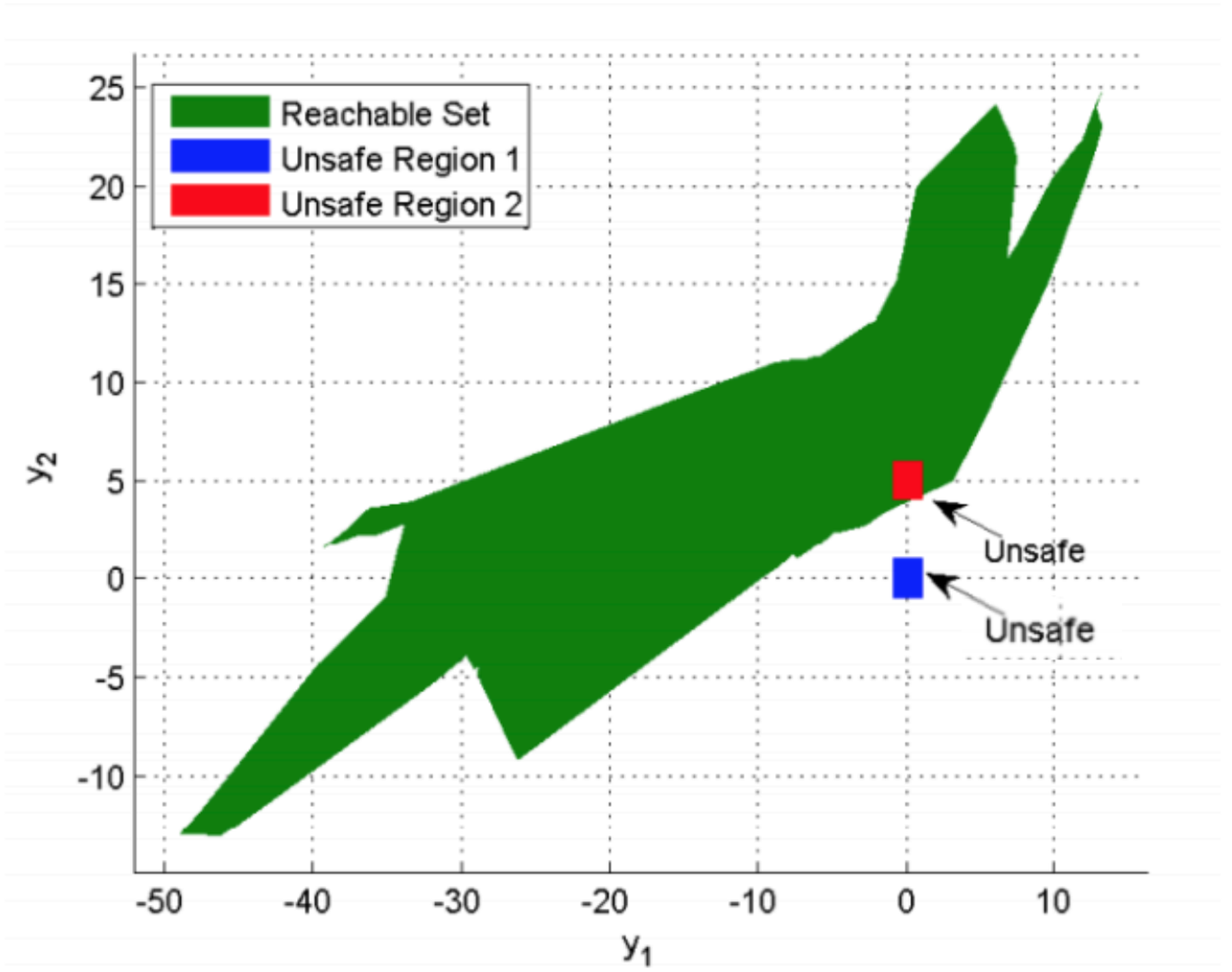


Prove that $y > -5$

Example Neural Network Verification Problem. (Bunel, 2018 [8])

- Satisfiability Modulo Theories (SMT)
- Mixed Integer Programming (MIP)

Reachability Analysis Methods [10]



Reachable Set Estimation. (Xiang, 2018 [10])

Leofante et al. [9]

- Categorized verification approaches into three distinct classes

Leofante et al. [9]

- Categorized verification approaches into three distinct classes
  - Invariance: Global and Local

Leofante et al. [9]

- Categorized verification approaches into three distinct classes
  - Invariance: Global and Local
  - Invertibility

Leofante et al. [9]

- Categorized verification approaches into three distinct classes
  - Invariance: Global and Local
  - Invertibility
  - Equivalence

Leofante et al. [9]

- Categorized verification approaches into three distinct classes
  - Invariance: Global and Local
  - Invertibility
  - Equivalence
- Majority of verification approaches deal with some notion of invariance

Leofante et al. [9]

- Categorized verification approaches into three distinct classes
  - Invariance: Global and Local
  - Invertibility
  - Equivalence
- Majority of verification approaches deal with some notion of invariance
- Very few approaches deal with network equivalence
  - Leofante et el. cite this as an interesting problem for further investigation.

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem

[1] https://github.com/guykatzz/ReluplexCav2017

[2] https://github.com/progirep/planet

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem
- Provided experimental evaluation of two state of the art verification tools

[1] https://github.com/guykatzz/ReluplexCav2017

[2] https://github.com/progirep/planet

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem
- Provided experimental evaluation of two state of the art verification tools
  - Reluplex[1][5]: Specialized SMT solver for ReLU neural networks

[1] https://github.com/guykatzz/ReluplexCav2017
[2] https://github.com/progirep/planet

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem
- Provided experimental evaluation of two state of the art verification tools
  - Reluplex[1][5]: Specialized SMT solver for ReLU neural networks
  - Planet[2][10]: Solver that leverages both SMT and MIP

[1] https://github.com/guykatzz/ReluplexCav2017

[2] https://github.com/progirep/planet

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem
- Provided experimental evaluation of two state of the art verification tools
  - Reluplex[1][5]: Specialized SMT solver for ReLU neural networks
  - Planet[2][10]: Solver that leverages both SMT and MIP
- Acheived a speed-up of over two orders of magnitude in some cases compared to Reluplex and Planet

[1] https://github.com/guykatzz/ReluplexCav2017

[2] https://github.com/progirep/planet

Rudy Bunel et al. [8]

- Posed the verification of neural networks with piecewise-linear activation functions as a Branch and Bound Optimization problem
- Provided experimental evaluation of two state of the art verification tools
  - Reluplex[1][5]: Specialized SMT solver for ReLU neural networks
  - Planet[2][10]: Solver that leverages both SMT and MIP
- Acheived a speed-up of over two orders of magnitude in some cases compared to Reluplex and Planet
  - Reluplex [5]: Did not provide Scalability Analysis

[1] https://github.com/guykatzz/ReluplexCav2017

[2] https://github.com/progirep/planet

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Sound and Complete, Sound and Incomplete

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Sound and Complete, Sound and Incomplete

- Sound and Complete
  - Guaranteed to find a solution if one exists
  - Limited Scalablility: thousands of neurons

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Sound and Complete, Sound and Incomplete

- Sound and Complete
  - Guaranteed to find a solution if one exists
  - Limited Scalablility: thousands of neurons
- Sound and Incomplete
  - Achieves better scalability: tens of thousands of neuorns

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Sound and Complete, Sound and Incomplete

- Sound and Complete
  - Guaranteed to find a solution if one exists
  - Limited Scalablility: thousands of neurons
- Sound and Incomplete
  - Achieves better scalability: tens of thousands of neuorns
  - can return "I don't know"

Modes of thought [11]

Classified Verification Approaches into two modes of thought

Sound and Complete, Sound and Incomplete

- Sound and Complete
  - Guaranteed to find a solution if one exists
  - Limited Scalablility: thousands of neurons
- Sound and Incomplete
  - Achieves better scalability: tens of thousands of neuorns
  - can return "I don't know"
  - Work well in practice

Summary of Verification Tools

- Majority of verification methods are for FNNs with piece-wise linear activation functions
- State-of-the-art image classification networks are currently out of reach.
  - VGGNet ([12]): 1.38×10^8 parameters
  - AlexNet ([13]): 6.0×10^7 parameters
  - GoogLeNet ([14]): 4.0×10^6 parameters

Tool Name	Network Type	Verification Approach	Largest Network Considered	Completeness	Type of Property
Reluplex [5]	FNN	SMT	1800 neurons	Complete	Invariance
Planet [10]	FNN, CNN	SMT, MIP	1341 neurons	Complete	Invariance, Invertibility
PLNN [8]	FNN, CNN	Branch and Bound	1485	Complete	Invariance
Sherlock [15]	FNN	MIP	3822	Complete	Invariance
VeriDeep/DLV[16]	FNN, CNN	SMT	1.25×10^6 parameters	Incomplete	Invariance
NNAF [17]	CNN	LP	60000 parameters	Incomplete	Invariance
MIPVerify.jl [18]	FNN, CNN	MIP	500 neurons	Incomplete	Invariance, Invertibility
AI^2 [19]	FNN, CNN	Abstract Interpretation	53000 neurons	Incomplete	Invariance

Summary of several available software tools for network verification present in the research literature. A more comprehensive list of tools can be found in [20].

Challenges

- Main Challenge is Scalability
  - Need for effective treatment of activation functions

Challenges

- Main Challenge is Scalability
  - Need for effective treatment of activation functions
- Comparing the existing verification techniques is difficult due to the diversity of:
  - Underlying SAT and SMT Solvers,
  - Verification modes of thought
  - Verification specifications considered
  - Benchmark sets and tool input formats

Challenges

- Main Challenge is Scalability
  - Need for effective treatment of activation functions
- Comparing the existing verification techniques is difficult due to the diversity of:
  - Underlying SAT and SMT Solvers,
  - Verification modes of thought
  - Verification specifications considered
  - Benchmark sets and tool input formats
- Shortage of methods for correcting erroneous behavior detected in neural network models

Future Work

- There are several interesting directions for future work

Future Work

- There are several interesting directions for future work
  - Creation of standardized benchmark sets and tool input formats

Future Work

- There are several interesting directions for future work
  - Creation of standardized benchmark sets and tool input formats
  - Comprehensive experimental evaluation of state-of-the-art tools

Future Work

- There are several interesting directions for future work
  - Creation of standardized benchmark sets and tool input formats
  - Comprehensive experimental evaluation of state-of-the-art tools
- Interdisciplinary studies from domain experts within the Automated Verification and Artificial Intelligence communities

Future Work

- There are several interesting directions for future work
  - Creation of standardized benchmark sets and tool input formats
  - Comprehensive experimental evaluation of state-of-the-art tools
- Interdisciplinary studies from domain experts within the Automated Verification and Artificial Intelligence communities
  - Development of explainable neural network models

Future Work

- There are several interesting directions for future work
  - Creation of standardized benchmark sets and tool input formats
  - Comprehensive experimental evaluation of state-of-the-art tools
- Interdisciplinary studies from domain experts within the Automated Verification and Artificial Intelligence communities
  - Development of explainable neural network models
- Consideration of other verification strategies such as Test Coverage Methods, Run-time Monitoring

Research Direction

- Currently working on creating a translation tool for neural network models[3]

What is ONNX?

ONNX is a open format to represent deep learning models. With ONNX, AI developers can more easily move models between state-of-the-art tools and choose the combination that is best for them. ONNX is developed and supported by a community of partners.

Open Neural Network Exchange Format. ([22])

[3] https://github.com/verivital/NeuralNetworkToolParsers

Research Direction

- Currently working on creating a translation tool for neural network models[3]
- Considering Verification of
  - Recurrent Neural Networks
  - Feed-forward networks with more general activation functions

What is ONNX?

ONNX is a open format to represent deep learning models. With ONNX, AI developers can more easily move models between state-of-the-art tools and choose the combination that is best for them. ONNX is developed and supported by a community of partners.

Open Neural Network Exchange Format. ([22])

[3] https://github.com/verivital/NeuralNetworkToolParsers

Conclusions

- Neural Network Verification is hard

Conclusions

- Neural Network Verification is hard
  - Obtaining efficient methods is worth the effort. Especially for safety-critical systems

Conclusions

- Neural Network Verification is hard
  - Obtaining efficient methods is worth the effort. Especially for safety-critical systems
- Numerous promising verification approaches have been proposed

Conclusions

- Neural Network Verification is hard
  - Obtaining efficient methods is worth the effort. Especially for safety-critical systems
- Numerous promising verification approaches have been proposed
  - Majority of these methods deal with feed-forward neural network with piece-wise linear activation functions

Conclusions

- Neural Network Verification is hard
  - Obtaining efficient methods is worth the effort. Especially for safety-critical systems
- Numerous promising verification approaches have been proposed
  - Majority of these methods deal with feed-forward neural network with piece-wise linear activation functions
- Main challenge is scalability but there are several other open problems

Conclusions

- Neural Network Verification is hard
  - Obtaining efficient methods is worth the effort. Especially for safety-critical systems
- Numerous promising verification approaches have been proposed
  - Majority of these methods deal with feed-forward neural network with piece-wise linear activation functions
- Main challenge is scalability but there are several other open problems
- Field is growing rapidly
  - Obtaining successful methods will have a large impact in numerous application domains

Thank you!

Questions?

References

A. Marshall, "The lose-lose ethics of testing self-driving cars in public," Mar 2018. [Online]. Available: https://www.wired.com/story/lose-lose-ethics-self-driving-public/

K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," *CoRR*, vol. abs/1705.06640, 2017. [Online]. Available: http://arxiv.org/abs/1705.06640

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1528–1540. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978392

G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," *CoRR*, vol. abs/1702.01135, 2017. [Online]. Available: http://arxiv.org/abs/1702.01135

A. Karpathy, "Convolutional Neural Networks (CNNs / ConvNets)," http://cs231n.github.io/convolutional-networks/. [Online]. Available: http://cs231n.github.io/convolutional-networks/

S. Jadon, "Introduction to different activation functions for deep learning," Mar 2018. [Online]. Available: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

R. Bunel, I. Turkaslan, P. H. S. Torr, P. Kohli, and M. P. Kumar, "Piecewise Linear Neural Network verification: A Comparative Study," *CoRR*, vol. abs/1711.00455, 2017. [Online]. Available: http://arxiv.org/abs/1711.00455

References

F. Leofante, N. Narodytska, L. Pulina, and A. Tacchella, "Automated Verification of Neural Networks: Advances, Challenges and Perspectives," *CoRR*, vol. abs/1805.09938, 2018. [Online]. Available: http://arxiv.org/abs/1805.09938

R. Ehlers, "Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks," *CoRR*, vol. abs/1705.01320, 2017. [Online]. Available: http://arxiv.org/abs/1705.01320

G. Katz, ""Verification of Machine Learning Programs"." St Anne's College (Oxford): 2nd School on Foundations of Programming and Software Systems, 7 2018, st Anne's College (Oxford), Oxford, England, July 4, 2018.

K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output Range Analysis for Deep Neural Networks," *CoRR*, vol. abs/1709.09130, 2017. [Online]. Available: http://arxiv.org/abs/1709.09130

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety Verification of Deep Neural Networks," *CoRR*, vol. abs/1610.06940, 2016. [Online]. Available: http://arxiv.org/abs/1610.06940

References

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety Verification of Deep Neural Networks," *CoRR*, vol. abs/1610.06940, 2016. [Online]. Available: http://arxiv.org/abs/1610.06940

O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. V. Nori, and A. Criminisi, "Measuring Neural Net Robustness with Constraints," *CoRR*, vol. abs/1605.07262, 2016. [Online]. Available: http://arxiv.org/abs/1605.07262

V. Tjeng and R. Tedrake, "Verifying Neural Networks with Mixed Integer Programming," *CoRR*, vol. abs/1711.07356, 2017. [Online]. Available: http://arxiv.org/abs/1711.07356

T. Gehr, M. Mirman, D. D. Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "AI^2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation," *IEEE Symposium on Security and Privacy*, vol. 39, May 2018. [Online]. Available: https://www.srl.inf.ethz.ch/publications.php

W. Xiang, P. Musau, A. A. Wild, D. M. Lopez, N. Hamilton, X. Yang, J. Rosenfeld, and T. T. Johnson, "Verification for Machine Learning, Autonomy, and Neural Networks Survey," *CoRR*, vol. abs/1810.01989, 2018. [Online]. Available: https://arxiv.org/abs/1810.01989

"Open neural network exchange format." [Online]. Available: http://onnx.ai/getting-started

S. A. Seshia, A. Desai, T. Dreossi, D. J. Fremont, S. Ghosh, E. Kim, S. Shivakumar, M. Vazquez-Chanlatte, and X. Yue, "Formal specification for deep neural networks," in *Automated Technology for Verification and Analysis*, S. K. Lahiri and C. Wang, Eds. Cham: Springer International Publishing, 2018, pp. 20–34.