A Dual Approach to Scalable Verification of Deep Networks

Krishnamurthy (Dj) Dvijotham,[*] Robert Stanforth, Sven Gowal, Timothy Mann, Pushmeet Kohli
DeepMind
London, N1C 4AG, UK

Abstract

This paper addresses the problem of formally verifying desirable properties of neural networks, *i.e.*, obtaining provable guarantees that neural networks satisfy specifications relating their inputs and outputs (robustness to bounded norm adversarial perturbations, for example). Most previous work on this topic was limited in its applicability by the size of the network, network architecture and the complexity of properties to be verified. In contrast, our framework applies to a general class of activation functions and specifications on neural network inputs and outputs. We formulate verification as an optimization problem (seeking to find the largest violation of the specification) and solve a Lagrangian relaxation of the optimization problem to obtain an upper bound on the worst case violation of the specification being verified. Our approach is *anytime* i.e. it can be stopped at any time and a valid bound on the maximum violation can be obtained. We develop specialized verification algorithms with provable tightness guarantees under special assumptions and demonstrate the practical significance of our general verification approach on a variety of verification tasks.

本文讨论了形式验证神经网络的理想特性的问题，即获得可证明的保证，即神经网络满足有关其输入和输出的规范（例如对有界范数对抗性摄动的鲁棒性）。以前有关此主题的大多数工作都受到网络规模，网络体系结构以及要验证的属性的复杂性的限制。相反，我们的框架适用于神经网络输入和输出的一类通用的激活函数和规范。我们将验证公式化为一个优化问题（试图找到最大的违反规范的问题），并解决优化问题的拉格朗日松弛，以获取最坏情况下正在验证的违反规范的上限。我们的方法是随时随地，即可以随时停止，并且可以获得最大违规的有效界限。我们在特殊的假设下开发了具有可证明的紧密性保证的专用验证算法，并展示了我们的通用验证方法对各种验证任务的实际意义。

1 INTRODUCTION

Neural networks and deep learning have revolutionized machine learning achieving state of the art performance on a wide range of complex prediction tasks [Krizhevsky et al., 2012, Goodfellow et al., 2016]. However, in recent years, researchers have observed that even state of the art networks can be easily fooled into changing their predictions by making small but carefully chosen modifications to the input data (known as *adversarial perturbations*) [Szegedy et al., 2013, Kurakin et al., 2016, Carlini and Wagner, 2017a, Goodfellow et al., 2014, Carlini and Wagner, 2017b]. While modifications to neural network training algorithms have been proposed to mitigate this phenomenonMadry et al. [2018], a comprehensive solution that is fully robust to adversarial attacks remains elusive [Carlini and Wagner, 2017b, Uesato et al., 2018].

神经网络和深度学习已经彻底改变了机器学习，在各种复杂的预测任务上实现了最先进的性能[]。然而，近年来，研究人员观察到，即使是最先进的网络，也可以通过对输入数据进行小规模但经过仔细选择的修改（称为对抗性扰动）来轻易地愚弄改变其预测[]。虽然已经提出了对神经网络训练算法的修改来缓解这种现象Madry等人[2018]，但对对抗性攻击完全稳健的全面解决方案仍然难以实现神经网络通常使用标准的机器学习范式进行测试：如果网络的性能（准确率）在网络在训练时无法访问的保持（测试）集上足够高，则认为网络可以接受。这是由基于数据生成机制假设的统计学论证来证明的，即每个输入输出对都是独立于相同（未知）数据分布生成的。然而，这种评估协议在具有关键安全约束的领域中是不够的[Marston和Baca，2015]。在这些情况下，我们可能需要更强的测试：例如，我们可能要求网络在一定的界限内对对抗性扰动具有鲁棒性。

Neural networks are typically tested using the standard machine learning paradigm: If the performance (accuracy) of the network is sufficiently high on a holdout (test) set that the network did not have access to while training, the network is deemed acceptable. This is justified by statistical arguments based on an *i.i.d.* assumption on the data generating mechanism, that is each input output pair is generated independently from the same (unknown) data distribution. However, this evaluation protocol is not sufficient in domains with critical safety constraints [Marston and Baca, 2015]. In these cases, we may require a stronger test: for example, we may require that the network is robust against adversarial perturbations within certain bounds.

对抗性评估

Adversarial evaluation. In the context of adversarial examples, a natural idea is to test neural networks by checking if it is possible to generate an adversarial attack to change the label predicted by the neural network [Kurakin et al., 2016] and train them to be robust to these examples Madry et al. [2018]. Generating adversarial examples is a challenging computational task itself, and the attack generated by a specific attack algorithm may be far from optimal. This may lead one to falsely conclude that a given model is robust to attacks even though a stronger adversary may have broken the robustness. Recent work [Athalye et al., 2018, Uesato et al., 2018] has shown that evaluating models against weak adversaries can lead to incorrect conclusions regarding the robustness of the model. Thus, there is a need to go beyond evaluation us-

在对抗样本的上下文中，自然的想法是通过检查是否可能产生对抗性攻击来更改神经网络预测的标签并训练它们对这些示例具有鲁棒性来测试神经网络。 生成对抗样本本身就是一项艰巨的计算任务，并且由特定攻击算法生成的攻击可能远非最佳。这可能会导致人们错误地得出结论，即使强大的对抗可能破坏了鲁棒性，给定的模型也对攻击具有鲁棒性。 最近的工作表明，针对弱势对抗评估模型可能导致有关模型鲁棒性的错误结论。 因此，需要超越使用特定对抗攻击的评估，并找到能够为任何对抗攻击提供可证明保证的方法。

[*] dvij@cs.washington.edu

ing specific adversarial attacks and find approaches that provide provable guarantees against attacks by *any adversary*.

建立可验证的模型。

Towards verifiable models. Verification of neural networks has seen significant research interest in recent years. In the formal verification community, Satisfiability Modulo Theory (SMT) solvers have been adapted for verification of neural networks [Ehlers, 2017, Huang et al., 2017, Katz et al., 2017]. While SMT solvers have been successfully applied to several domains, applying them to large neural networks remains a challenge due to the scale of the resulting SMT problem instances. Furthermore, these approaches have been largely limited to networks with piecewise linear activation functions since most SMT solvers are unable to deal efficiently with nonlinear arithmetic. More recently, researchers have proposed a set of approaches that make use of branch and bound algorithms either directly or via mixed-integer programming solvers [Bunel et al., 2017, Cheng et al., 2017, Tjeng and Tedrake, 2017]. While these approaches achieve strong results on smaller networks, scaling them to large networks remains an open challenge. These approaches also rely heavily on the piecewise linear structure of networks where the only nonlinearities are maxpooling and ReLUs.

近年来,神经网络的验证引起了广泛的研究兴趣。在正式的验证社区中,可满足性模理论（SMT）求解器已被用于神经网络的验证[Ehlers,2017; Huang等,2017; Katz等,2017]。尽管SMT求解器已成功应用于多个领域,但由于生成的SMT问题实例的规模巨大,将其应用于大型神经网络仍然是一个挑战。此外,由于大多数SMT求解器无法有效地处理非线性算术,因此这些方法在很大程度上限于具有分段线性激活函数的网络。最近,研究人员提出了一套直接通过混合整数规划求解器使用分支定界算法的方法[Bunel等,2017;Cheng等,2017;Tjeng和Tedrake,2017]。尽管这些方法在较小的网络上取得了不错的结果,但将其扩展到大型网络仍然是一个开放的挑战。这些方法还严重依赖于网络的分段线性结构,其中唯一的非线性是maxpooling和ReLU。

建立通用模型的可扩展验证。

Towards scalable verification of general models. In this paper, we develop a novel approach to neural network verification based on optimization and duality. The approach consists of formulating the verification problem as an optimization problem that tries to find the largest violation of the property being verified. If the largest violation is smaller than zero, we can conclude that the property being verified is true. By using ideas from duality in optimization, we can obtain bounds on the optimal value of this problem in a computationally tractable manner. Note that this approach is *sound but incomplete*, in that there may be cases where the property of interest is true, but the bound computed by our algorithm is not tight enough to prove the property. This strategy has been used in prior work as well [Kolter and Wong, 2018, Raghunathan et al., 2018]. However, our results improve upon prior work in the following ways:

在本文中,我们开发了一种基于优化和对偶性的新型神经网络验证方法。该方法包括将验证问题表述为优化问题,以试图找到对所验证属性的最大违规。如果最大违规小于零,我们可以得出结论,正在验证的属性为true。通过使用对偶性的思想进行优化,我们可以以可计算的方式获得此问题的最优值的界限。请注意,这种方法是合理的,但并不完备,因为在某些情况下,感兴趣的属性为true,但是我们的算法计算出的边界不够严格,无法证明该属性。该策略也已用于先前的工作中[KolterandWong,2018,ghunathanetal。,2018]。但是,我们的结果在以下方面改进了先前的工作：

1. Our verification approach applies to *arbitrary* feed-forward neural networks with *any architecture* and *any activation function* and our framework recovers previous results [Ehlers, 2017] when applied to the special case of piecewise linear activation functions.

   我们的验证方法适用于具有任何体系结构和任何激活函数的任意前馈神经网络，并且当应用于分段线性激活函数的特殊情况时，我们的框架可以恢复以前的结果[Ehlers，2017]。

2. We can handle verification of systems with discrete inputs and combinatorial constraints on the input space, including cardinality constraints.

   我们可以处理对具有离散输入和输入空间组合约束（包括基数约束）的系统的验证。

3. The computation involved only requires solving an unconstrained convex optimization problem (of size linear in the number of neurons in the network), which can be done using a subgradient method efficiently. Further, our approach is *anytime*, in the sense that the computation can be stopped at any time and a valid bound on the verification objective can be obtained.

   所涉及的计算仅需要解决无约束的凸优化问题（网络中神经元数量的大小线性），可以使用次梯度方法有效地完成。 而且，我们的方法是随时随地，从某种意义上说，可以随时停止计算，并且可以获得验证目标的有效界限。

4. For the special case of single hidden layer networks, we develop specialized verification algorithms with provable tightness guarantees.

   对于单隐藏层网络的特殊情况，我们开发了具有可证明的紧密性保证的专用验证算法。

5. We attain state of the art verified bounds on adversarial error rates on image classifiers trained on MNIST and CIFAR-10 under adversarial perturbations in the infinity norm.

   我们在无穷范数的对抗性扰动下，在MNIST和CIFAR-10上训练的图像分类器上，获得了对抗错误率的最新验证边界。

2 Related Work

Certifiable training and verification of neural networks A separate but related thread of work is on *certifiable training*, ie, training neural networks so that they are guaranteed to satisfy a desired property (for example, robustness to adversarial examples within a certain radius) [Kolter and Wong, 2018, Raghunathan et al., 2018]. These approaches use ideas from convex optimization and duality to construct bounds on an optimization formulation of verification. However, these approaches are limited to either a class of activation functions (piecewise linear models) or architectures (single hidden layer, as in Raghunathan et al. [2018]). Further, in Kolter and Wong [2018], the dual problem starts with a constrained convex formulation but is then converted into an unconstrained but nonconvex optimization problem to allow for easy optimization via a backprop-style algorithm. In contrast, our formulation allows for an unconstrained dual convex optimization problem so that for *any choice of dual variables*, we obtain a valid bound on the adversarial objective and this dual problem can be solved efficiently using subgradient methods.

单独但相关的工作是在可认证的训练上，即训练神经网络，以确保它们满足所需的属性（例如，在一定半径内对对抗示例的鲁棒性）[Kolter and Wong, 2018，Raghunathan等。。，2018]。这些方法使用凸优化和对偶的思想来构造验证优化公式的界限。但是，这些方法仅限于一类激活函数（分段线性模型）或架构（如Raghunathan等人[2018]所述的单个隐藏层）。此外，在Kolter和Wong [2018]中，对偶问题从约束凸公式开始，然后转换为无约束但非凸的优化问题，从而可以通过反向传播算法轻松实现优化。相反，我们的公式允许无约束的双凸优化问题，因此对于任何对偶变量的选择，我们都能获得对抗目标的有效边界，并且可以使用次梯度方法有效地解决该对偶问题。

We also note that the ultimate goals of [Kolter and Wong, 2018, Raghunathan et al., 2018] are different from our paper: they modify the training procedure of the neural network so that the network is trained to be easily verifiable. In contrast, our work focuses on extending verification algorithms to apply to a broader class of architectures, activation functions and - in this sense, we view our work as complementary to [Kolter and Wong, 2018, Raghunathan et al., 2018]. In fact, since the objective of our dual optimization is differentiable with respect to the network weights, we can extend our approach to training verifiable networks easily by simultaneously optimizing the network weights and dual variables to minimize the

我们还注意到，[Kolter and Wong, 2018; Raghunathan et al。，2018]的最终目标与我们的论文不同：它们修改了神经网络的训练过程，因此训练了该网络易于验证。 相比之下，我们的工作重点是扩展验证算法，以将其应用于更广泛的架构，激活功能，从这个意义上讲，我们认为我们的工作是对[Kolter和Wong，2018；Raghunathan等，2018]的补充。 实际上，由于双重优化的目标相对于网络权重是可区分的，因此我们可以通过同时优化网络权重和双重变量以最小化双重目标，将我们的方法扩展到轻松培训可验证网络。 我们将这种扩展的研究留给以后的工作。

dual objective. We leave the study of such an extension for future work.

Theoretical analysis of robustness Another related line of work has to do with theoretical analysis of adversarial examples. It has been shown that feedforward ReLU networks cannot learn to distinguish between points on two concentric spheres without necessarily being vulnerable to adversarial examples within a small radius [Gilmer et al., 2018]. Under a different set of assumptions, the existence of adversarial examples with high probability is also established in Fawzi et al. [2018]. In Wang et al. [2017], the authors study robustness of nearest neighbor classifiers to adversarial examples. As opposed to these theoretical analyses, our approach searches computationally for proofs of existence or non-existence of adversarial examples. The approach does not say anything a-priori about the existence of adversarial examples, but can be used to investigate their existence for a given network and compare strategies to guard against adversarial attacks.

另一个相关的工作领域是与对抗性样本进行理论分析。已有研究表明，前馈ReLU网络无法学习区分两个同心球上的点，而在小半径内必然容易受到对抗性例子的影响。在一组不同的假设下，Fawzi等人也确定了具有高概率的对抗性例子的存在。作者研究了最近邻分类器对对抗性示例的鲁棒性。与这些理论分析相反，我们的方法在计算上搜索对抗性示例的存在或不存在的证据。该方法并没有说出对抗性示例的存在先验性，但是可以用于调查给定网络的存在性，并比较防范对抗性攻击的策略。

3 VERIFICATION AS OPTIMIZATION

3.1 NOTATION

Our techniques apply to general feedforward architectures and recurrent networks, but we focus on layered architectures for the development in this paper. The input layer is numbered 0, the hidden layers are numbered $1, \ldots, L-1$ and the output layer is numbered L. The size of layer l is denoted n_l

我们的技术适用于一般的前馈体系结构和递归网络，但是在本文中，我们专注于分层体系结构的开发。输入层编号为0，隐藏层编号为1到L-1，输出层编号为L。层l的大小表示为nl

We denote by x^{in} the input to the neural network, by z^l the pre-activations of neurons at layer l *before* application of the activation function and by x^l the vector of neural activations *after* application of the activation function (to z^{l-1}). For convenience, we define $x^0 = x^{in}$. We use $x^l(x^{in}), z^l(x^{in})$ to denote the activations at the l-th layer as a function of the input x^{in}. Upper and lower bounds on the pre/post activations are denoted by $\underline{x}^l, \overline{x}^l, \underline{z}^l, \overline{z}^l$ respectively. The activation function at layer l is denote h^l and is assumed to be applied component-wise, ie,

$$[h^l(z^l)]_k = h^l_k(z^l_k)$$

Note that max-pooling is an exception to this rule - we discuss how max-pooling is handled separately in the Appendix section 6.2.1. The weights of the network at layer l are denoted W^l and the bias is denoted b^l, $z^l = W^l x^l + b^l$.

3.2 VERIFICATION PROBLEM

As mentioned earlier, *verification* refers to the process of checking that the output of the neural network satisfies a certain desirable property for all choices of the input within a certain set. Formally, this can be stated as follows:

如前所述，验证是指检查神经网络的输出满足特定集合中输入的所有选择的特定期望属性的过程。 形式化的，可以这样表示：

$$\forall x^{in} \in \mathcal{S}_{in}\left(x^{nom}\right) \quad x^L(x^{in}) \in \mathcal{S}_{out} \tag{1}$$

where x^{in} denotes the input to the network, x^{nom} denotes a nominal input, $\mathcal{S}_{in}(x^{nom})$ defines the constrained subset of inputs induced by the nominal input, and $\mathcal{S}_{out}$ denotes the constraints on the output that we would like to verify are true for all inputs in $\mathcal{S}_{in}(x^{nom})$. In the case of adversarial perturbations in image classification, x^{nom} would refer to the nominal (unperturbed image), $\mathcal{S}_{in}(x^{nom})$ would refer to all the images that can be obtained by adding bounded perturbations to x^{nom}, and x^{in} would refer to a perturbed image.

Sin (xnom) 定义由标称输入引起的约束输入子集

对于图像分类中的对抗性扰动，x_nom将指代名义（未扰动的图像），S_in (x_nom) 将指代通过将有限扰动添加到x_nom可获得的所有图像，而x_in将指代扰动的图像 。

In this paper, we will assume that: $\mathcal{S}_{out}$ is always a described by a finite set of linear constraints on the values of final layer *ie.* $\mathcal{S}_{out} = \cap_{i=1}^{m}\{x^L : \left(c^i\right)^T x^L + d^i \leq 0\}$, and $\mathcal{S}_{in}\left(x^{nom}\right)$ is any bounded set such that any linear optimization problem of the form

$$\max_{x^{in} \in \mathcal{S}_{in}(x^{nom})} c^T x$$

can be solved efficiently. This includes convex sets and also sets describing combinatorial structures like spanning trees, cuts in a graph and cardinality constraints.

这包括凸集，还包括描述组合结构的集，例如生成树，图形中的割和基数约束。

See the following examples for a concrete illustration of the formulation of the problem:

Robustness to targeted adversarial attacks. Consider an adversarial attack that seeks to perturb an input x^{nom} to an input x^{in} subject to a constraint on the perturbation $\left\|x^{in} - x^{nom}\right\| \leq \epsilon$ to change the label from the true label i to a target label j. We can map this to (1) as follows:

考虑一个对抗性攻击，该攻击试图在对扰动p的约束下，将一个输入xn扰动为一个输入xi(给xn添加扰动，变为xi)。

$$\mathcal{S}_{in}\left(x^{nom}\right) = \{x^{in} : \left\|x^{in} - x^{nom}\right\| \leq \epsilon\}, \tag{2a}$$

$$\mathcal{S}_{out} = \{z : c^T z \leq 0\} \tag{2b}$$

where c is a vector with $c_j = 1, c_i = -1$ and all other components 0. Thus, $\mathcal{S}_{out}$ denotes the set of outputs for which the true label i has a higher logit value than the target label j (implying that the targeted adversarial attack did not succeed).

单调预测因子。

Monotonic predictors. Consider a network with a single real valued output and we are interested in ensuring that the output is monotonically increasing wrt each dimension of the input x^{in}. We can state this as a verifica-

考虑一个具有单个实值输出的网络，我们有兴趣确保输出随输入x_in的每个维度单调增加。 我们可以说这是一个验证问题。

tion problem:

$$\mathcal{S}_{in}\left(x^{nom}\right) = \{x^{in} : x^{in} \geq x^{nom}\} \tag{3a}$$

$$\mathcal{S}_{out} = \{x^L : x^L\left(x^{nom}\right) - x^L \leq 0\} \tag{3b}$$

Thus, $\mathcal{S}_{out}$ denotes the set of outputs which are large than the network output at x^{nom}. If this is true for each value of x^{nom}, then the network is monotone.

单调

基数约束

在某些情况下，不仅在范式上而且在可以被扰动的输入维数方面，都应该限制扰动。 我们可以这样说

Cardinality constraints. In several cases, it makes sense to constrain a perturbation not just in norm but also in terms of the number of dimensions of the input that can be perturbed. We can state this as:

$$\mathcal{S}_{in}\left(x^{nom}\right) = \\ \{x^{in} : \left\|x^{in} - x^{nom}\right\|_0 \leq k, \left\|x^{in} - x^{nom}\right\|_\infty \leq \epsilon\} \tag{4a}$$

$$\mathcal{S}_{out} = \{z : c^T z \leq 0\} \tag{4b}$$

where $\|x\|_0$ denotes the number of non-zero entries in x. Thus, $\mathcal{S}_{out}$ denotes the set of outputs which are larger than the network output at x^{nom}. If this is true for each value of x^{nom}, then the network is monotone.

3.3 OPTIMIZATION PROBLEM FOR VERIFICATION

Once we have a verification problem formulated in the form (1), we can easily turn the verification procedure into an optimization problem. This is similar to the optimization based search for adversarial examples [Szegedy et al., 2013] when the property being verified is adversarial robustness. For brevity, we only consider the case where $\mathcal{S}_{out}$ is defined by a single linear constraint $c^T z + d \leq 0$. If there are multiple constraints, each one can be verified separately.

$$\max_{\substack{z^0,\ldots,z^{L-1} \\ x^0,\ldots,x^L}} c^T x^L + d \tag{5a}$$

$$\text{s.t } x^{l+1} = h^l\left(z^l\right), l = 0, 1, \ldots, L-1 \tag{5b}$$

$$z^l = W^l x^l + b^l, l = 0, 1, \ldots, L-1 \tag{5c}$$

$$x^0 = x^{in}, x^{in} \in \mathcal{S}_{in}\left(x^{nom}\right) \tag{5d}$$

If the optimal value of this problem is smaller than 0 (for each c, d in the set of linear constraints defining $\mathcal{S}_{out}$), we have verified the property (1). This is a nonconvex optimization problem and finding the global optimum in general is NP-hard (see Appendix section 6.4.1 for a proof). However, if we can compute *upper bounds* on the value of the optimization problem and the upper bound is smaller than 0, we have successfully verified the property. In the following section, we describe our main approach for computing bounds on the optimal value of (5).

3.4 BOUNDING THE VALUE OF THE OPTIMIZATION PROBLEM

We assume that bounds on the activations $z^l, x^l, l = 0, \ldots, L-1$ are available. Section 6.1 discusses details of how such bounds may be obtained given the constraints on the input layer $\mathcal{S}_{in}\left(x^{nom}\right)$. We can bound the optimal value of (5) using a Lagrangian relaxation of the constraints:

$$\begin{aligned} \max_{\substack{z^0,\ldots,z^{L-1} \\ x^0,x^1,\ldots,x^{L-1}}} & c^T\left(h^{L-1}\left(z^{L-1}\right)\right) + d \\ & + \sum_{l=0}^{L-1} \left(\mu^l\right)^T\left(z^l - W^l x^l - b^l\right) \\ & + \sum_{l=0}^{L-2} \left(\lambda^l\right)^T\left(x^{l+1} - h^l\left(z^l\right)\right) \end{aligned} \tag{6a}$$

$$\text{s.t. } \underline{z}^l \leq z^l \leq \overline{z}^l \quad, l = 0, 1, \ldots, L-1 \tag{6b}$$

$$\underline{x}^l \leq x^l \leq \overline{x}^l \quad, l = 0, 1, \ldots, L-1 \tag{6c}$$

$$x^0 \in \mathcal{S}_{in}\left(x^{nom}\right) \tag{6d}$$

Note that any feasible solution for the original problem (5) is feasible for the above problem, and for any such solution, the terms involving λ, μ become 0 (since the terms multiplying λ, μ are 0 for every feasible solution). Thus, for any choice of λ, μ, the above optimization problem provides a valid upper bound on the optimal value of (5) (this property is known as weak duality [Vandenberghe and Boyd, 2004]).

We now look at solving the above optimization problem. Since the objective and constraints are separable in the layers, the variables in each layer can be optimized independently. For $l = 1, \ldots, L-1$, we have

$$f_l\left(\lambda^{l-1}, \mu^l\right) = \\ \max_{x^l \in [\underline{x}^l, \overline{x}^l]} \left(\lambda^{l-1} - \left(W^l\right)^T \mu^l\right)^T x^l - \left(b^l\right)^T \mu^l$$

which can be solved trivially by setting each component of x^l to its upper or lower bound depending on whether the corresponding entry in $\lambda^{l-1} - \left(W^l\right)^T \mu^l$ is non-negative. Thus,

$$f_l\left(\lambda^{l-1}, \mu^l\right) = \\ \left[\lambda^{l-1} - \left(W^l\right)^T \mu^l\right]_+^T \overline{x}^l \\ + \left[\lambda^{l-1} - \left(W^l\right)^T \mu^l\right]_-^T \underline{x}^l - \left(b^l\right)^T \mu^l$$

where $[x]_+ = \max(x, 0), [x]_- = \min(x, 0)$ denote the positive and negative parts of x.

Similarly, collecting the terms involving z^l, we have, for $l = 0, \ldots, L-1$

$$\tilde{f}_l(\lambda^l, \mu^l) = \max_{z^l \in [\underline{z}^l, \overline{z}^l]} {\mu^l}^T z^l - \left(\lambda^l\right)^T h^l\left(z^l\right)$$

where $\lambda_{L-1} = -c$.

Since h^l is a component-wise nonlinearity, each dimension of z^l can be optimized independently. For the k-th dimension, we obtain

$$\tilde{f}_{l,k}\left(\lambda^l_k, \mu^l_k\right) = \max_{z^l_k \in [\underline{z}^l_k, \overline{z}^l_k]} \mu^l_k z^l_k - \lambda^l_k h^l_k\left(z^l_k\right)$$

This is a one-dimensional optimization problem and can be solved easily- for common activation functions (ReLU, tanh, sigmoid, maxpool), it can be solved analytically, as discussed in appendix section 6.2. Finally, we need to solve

$$f_0(\mu^0) = \max_{x^0 \in \mathcal{S}_{in}(x^{nom})} \left(-\left(W^0\right)^T \mu_0\right)^T x^0 - \left(b^0\right)^T \mu^0$$

which can also be solved easily given the assumption on $\mathcal{S}_{in}$. We work out some concrete cases in 6.3.

Once these problems are solved, we can construct the dual optimization problem:

$$\min_{\lambda,\mu} \sum_{k=0}^{n_{L-1}} \tilde{f}_{L-1,k}\left(-c_k, \mu^l_k\right) + \sum_{l=0}^{L-2} \sum_{k=0}^{n_l} \tilde{f}_{l,k}\left(\lambda^l_k, \mu^l_k\right) + \sum_{l=1}^{L-1} f_l(\lambda^{l-1}, \mu^l) + f_0\left(\mu^0\right) + d \tag{7}$$

This seeks to choose the values of λ, μ so as to minimize the upper bound on the verification objective, thereby obtaining the tightest bound on the verification objective.

This optimization can be solved using a subgradient method on λ, μ.

Theorem 1. *For any values of λ, μ, the objective of* (7) *is an upper bound on the optimal value of* (5). *Hence, the optimal value of* (7) *is also an upper bound. Further,* (7) *is a convex optimization problem in* (λ, μ).

Proof. The upper bound property follows from weak duality [Vandenberghe and Boyd, 2004]. The fact that (7) is a convex optimization problem can be seen as each term $f_l, \tilde{f}_{l,k}$ is expressed as a maximum overa set of linear functions of λ, μ [Vandenberghe and Boyd, 2004]. □

Theorem 2. *If each h is a ReLU function, then* (7) *is equivalent to the dual of the LP described in Ehlers [2017].*

Proof. See section 6.4. □

The LP formulation from Ehlers [2017] is also used in Kolter and Wong [2018]. The dual of the LP is derived in Kolter and Wong [2018] - however this dual is different from (7) and ends up with a constrained optimization formulation for the dual (the details of this can be found in appendix section 6.4.2). To allow for an unconstrained formulation, this dual LP is transformed to a backpropagation-like computation. While this allows for folding the verification into training, it also introduces nonconvexity in the verification optimization - our formulation of the dual differs from Kolter and Wong [2018] in that we directly solve an unconstrained dual formulation, allowing us to circumvent the need to solve a nonconvex optimization for verification.

3.5 TOWARDS THEORETICAL GUARANTEES FOR VERIFICATION

The bounds computed by solving (7) could be loose in general, since (5) is an NP-hard optimization problem (section 6.4.1). SMT solvers and MIP solvers are guaranteed to find the exact optimum for piecewise linear neural networks, however, they may take exponential time to do so. Thus, an open question remains: Are there cases where it is possible to perform *exact verification* efficiently? If not, can we approximate the verification objective to within a certain factor (known a-priori)? We develop results answering these questions in the following sections.

Prior work: For any linear classifier, the scores of each label are a linear function of the inputs $w_i^T x + b_i$. Thus, the difference between the predictions of two classes j (target class for an adversary) and class i (true label) is $(w_i - w_j)^T x$. Maximizing this subject to $\left\|x - x^0\right\|_2 \leq \epsilon$ can be solved analytically to obtain the value $(w_i - w_j)^T x^0 + \|w_i - w_j\|_2 \epsilon$. This observation formed the basis for algorithms in [Raghunathan et al., 2018] and [Hein and Andriushchenko, 2017]. However, once we move to nonlinear classifiers, the situation is not so simple and computing the worst case adversarial example, even in the 2-norm case, becomes a challenging task. In [Hein and Andriushchenko, 2017], the special case of kernel methods and single hidden layer classifiers are considered, but the approaches developed are only upper bounds on the verification objective (just like those computed by our dual relaxation approach). Similarly, in Raghunathan et al. [2018], a semidefinite programming approach is developed to compute bounds on the verification objective for the special case of adversarial perturbations on the infinity norm. However, none of these approaches come with a-priori guarantees on the quality of the bound, that is, before actually running the verification algorithm, one cannot predict how tight the bound on the verification objective would be. In this section, we develop novel theoretical results that quantify when the verification problem (5) can be solved *either exactly or with a-priori approximation guarantees*. Our results require strong assumptions and do not immediately apply to most practical situations. However, we believe that they shed some understanding on the conditions under which exact verification can be performed tractably and lead to specialized verification algorithms that merit further study.

We assume the following for all results in this section:
1) We study networks with a single hidden layer, *i.e.* $L = 2$, with activation function $h^0 = h$ and a linear mapping from the penultimate to the output layer $x^2 = h^1\left(z^1\right) = z^1 = W^1 x^1 + b^1$.
2) The network has a differentiable activation function h with Lipschitz-continuous derivatives denoted h' (tanh, sigmoid, ELU, polynomials satisfy this requirement).
3) $\mathcal{S}_{in}\left(x^{nom}\right) = \{x^{in} : \left\|x^{in} - x^{nom}\right\|_2 \leq \epsilon\}$.

Since the output layer is a linear function of the penultimate layer x^1, we have

$$\begin{aligned}
c^T x^2 &= \left(\left(W^1\right)^T c\right)^T x^1 + c^T b^1 \\
&= \left(\left(W^1\right)^T c\right)^T h^0\left(z^0\right) + c^T b^1
\end{aligned}$$

For brevity, we simply denote $\left(W^1\right)^T c$ as c, drop the constant term $c^T b^1$ and let $W = W^0, b = b^0, z^{nom} = W x^{nom} + b$ and W_i denote the i-th row of W. Then, (5) reduces to:

$$\max_{x^{in} : \|x^{in} - x^{nom}\|_2 \leq \epsilon} \sum_i c_i h_i\left(W_i x^{in} + b_i\right) \qquad (8)$$

Theorem 3. *Suppose that h has a Lipschitz continuous first derivative:*

$$h_i'(t) - h_i'(\tilde{t}) \leq \gamma_i |t - \tilde{t}|$$

Let

$$\begin{aligned}
\nu &= \left\|\operatorname{diag}(c)\, W^T h'\left(z^{nom}\right)\right\|_2 \\
L &= \sigma_{\max}\left(\operatorname{diag}(c)\, W^T\right) \sigma_{\max}\left(\operatorname{diag}(\gamma)\, W\right)
\end{aligned}$$

Then $\forall\, \epsilon \in (0, \frac{\nu}{2L})$*, the iteration:*

$$x^{k+1} \leftarrow x^{nom} + \epsilon\left(\frac{W^T \operatorname{diag}(c)\, h'(W x^k + b)}{\|W^T \operatorname{diag}(c)\, h'(W x^k + b)\|_2}\right)$$

starting at $x^0 = x^{nom}$ *converges to the global optimum* $x^\star$ *of* (8) *at the rate* $\left\|x^k - x^\star\right\| \leq \left(\frac{\epsilon L}{\nu - \epsilon L}\right)^k$

Proof. Section 6.4 □

Thus, when ϵ is small enough, a simple algorithm exists to find the global optimum of the verification objective. However, even when ϵ is larger, one can obtain a good approximation of the verification objective. In order to do this, consider the following quadratic approxiimation of the objective from (8):

$$\begin{aligned}
\max \sum_i c_i\left(h_i\left(z_i^{nom}\right) + h_i'\left(z_i^{nom}\right)\left(W_i z\right)\right) \\
+ \sum_i \frac{c_i}{2} h_i''\left(z_i^{nom}\right)\left(W_i z\right)^2 \qquad (9a) \\
\text{s.t. } \|z\|_2 \leq \epsilon \qquad (9b)
\end{aligned}$$

This optimization problem corresponds to a trust region problem that can be solved to global optimality using semidefinite programming [Yakubovic, 1971]:

$$\begin{aligned}
\max_{z,Z} \sum_i c_i\left(h_i\left(z_i^{nom}\right) + h_i'\left(z_i^{nom}\right)\left(W_i z\right)\right) \\
+ \sum_i \frac{c_i}{2} h_i''\left(z_i^{nom}\right) \operatorname{tr}\left(W_i^T W_i Z\right) \qquad (10a) \\
\text{s.t. } \operatorname{tr}(Z) \leq \epsilon, \begin{pmatrix} 1 & z^T \\ z & Z \end{pmatrix} \succeq 0 \qquad (10b)
\end{aligned}$$

where $X \succeq 0$ denotes that X is constrained to be a positive semidefinite matrix. While this can be solved using general semidefinite programming solvers, several special purpose algorithms exist for this trust region problem that can exploit its particular structure for efficient solution, [Hazan and Koren, 2016]

Theorem 4. *Suppose that h is thrice-differentiable with a globally bounded third derivative. Let*

$$\zeta_i = \|W_i\|_2, \eta_i = \sup_t |h_i'''(t)|, \kappa = \frac{1}{6}\left(\sum_i \eta_i c_i \zeta_i^3\right)$$

For each $\epsilon > 0$, the difference between the optimal values of (10), (8) *is at most* $\kappa\epsilon^3$.

Proof. See Section 6.4 □

4 EXPERIMENTS

In this section, we present numerical studies validation our approach on three sets of verification tasks:
Image classification on MNIST and CIFAR: We use our approach to obtain guaranteed lower bounds on the accuracy of image classifers trained on MNIST and CIFAR-10 under adversarial attack with varying sizes of the perturbation radius. We compare the bounds obtained by our method with prior work (in cases where prior work is applicable) and also with the best attacks found by various approaches.

Classifier stability on GitHub data: We train networks on sequences of commits on GitHub over a collection of 10K repositories - the prediction task consists of predicting whether a given repository will reach more than 40 commits within 250 days given data observed until a certain day. Input features consist a value between 0 and 1 indicating the number of days left until the 250th day, as well as another value indicating the progress of commits towards the total of 40. As the features evolve, the prediction of the classifier changes (for example, predictions should become more accurate as we move closer to the 250th day). In this situation, it is desirable that the classifier provides consistent predictions and that the number of times its prediction switches is as small as possible. It is also desirable that this switching frequency cannot be easily be changed by perturbing input features. We use our verification approach combined with dynamic programming to compute a bound on the maximum number of switches in the classifier prediction over time.
Digit sum task: We consider a more complex verification task here: Given a pair of MNIST digits, the goal is to bound how much the sum of predictions of a classifier can differ from the true sum of those digits under adversarial perturbation subject to a total budget on the perturbation across the two digits.

4.1 IMAGE CLASSIFICATION: MNIST AND CIFAR

We study adversarial attacks subject to an l_∞ bound on the input perturbation. An adversarial example (AE) is a pertubation of an input of the neural network such that the output of the neural network differs from the correct label for that input. An AE is said to be within radius ϵ if the ℓ_∞ norm of the difference between the AE and the original input is smaller than ϵ. We are interested in the *adversarial error rate*, that is,

$$\frac{\#\text{ Test examples that have an AE within radius } \epsilon}{\text{Size of test set}}$$

Computing this quantity precisely requires solving the NP-hard problem (5) for each test example, but we can obtain upper bounds on it using our (and other) verification methods and lower bounds using a fixed attack algorithm (in this paper we use a bound constrained LBFGS algorithm similar to [Carlini and Wagner, 2017b]). Since theorem 2 shows that for the special case of piecewise linear neural networks, our approach reduces to the basic LP relaxation from Ehlers [2017] (which also is the basis for the algorithms in Bunel et al. [2017] and Kolter and Wong [2018]), we focus on networks with smooth nonlinearities like tanh and sigmoid. We compare our approach with The SDP formulation from Raghunathan et al. [2018] (note that this approach only works for single hidden layer networks, so we just show it as producing vacuous bounds for other networks).

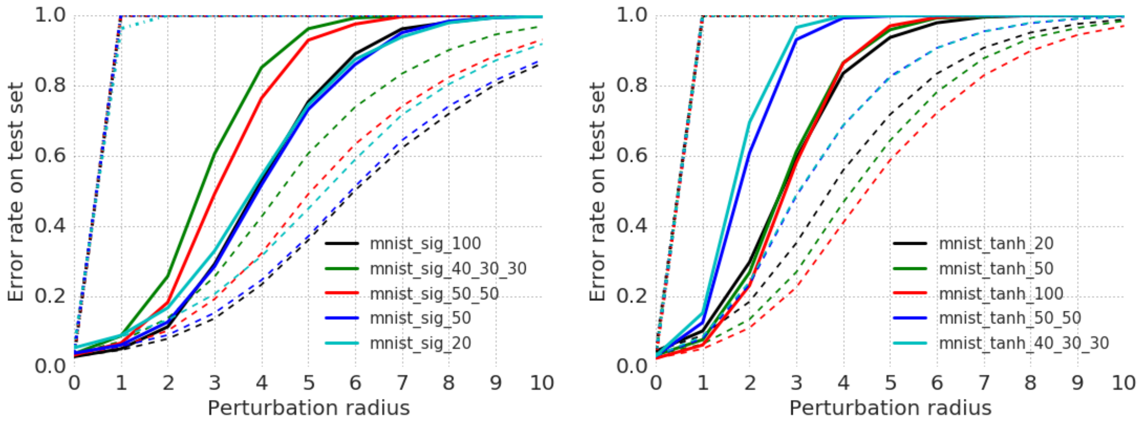
Each approach gets a budget of 300 s per verifiation problem (choice of test example and target label). Since the SDP solver from [Raghunathan et al., 2018] only needs to be run once per label pair (and not per test example), its running time is amortized appropriately.

Results on smooth activation functions: Figures 1a,1b show that our approach is able to compute nearly tight bounds (bounds that match the upper bound) for small perturbation radii (up to 2 pixel units) and our bounds significantly outperform those from the SDP approach [Raghunathan et al., 2018] (which is only able to compute nontrivial bounds for the smallest model with 20 hidden units).

Results on models trained adversarially: We use the adversarial training approach of [Uesato et al., 2018] and train models on MNIST and CIFAR that are robust to perturbations from the LBFGS-style attack on the training set. We then apply our verification algorithm to these robust models and obtain bounds on the adversarial error rate on the test set. These models are all multilayer models, so the SDP approach from [Raghunathan et al., 2018] does not apply and we do not plot it here. We simply plot the attack versus the bound from our approach. The results for MNIST are plotted in figure 2b and for CIFAR in figure 2a. On networks trained using a different procedure, the approaches from Raghunathan et al. [2018] and Kolter and Wong [2018] are able to achieve stronger results for larger values of ϵ (they work with $\epsilon = .1$ in real units, which corresponds to $\epsilon = 26$ in pixel units we use here). However, we note that our verification procedure is agnostic to the training procedure, and can be used to obtain fairly tight bounds for any network and any training procedure. In comparison, the results in Kolter and Wong [2018] and Raghunathan et al. [2018] rely on the training procedure optimizing the verification bound. Since we do not rely on a particular adversarial training procedure, we were also able to obtain the first non-trivial verification bounds on CIFAR-10 (to the best of our knowledge) shown in figure 2a. While the model quality is rather poor, the results indicate that our approach could scale to more complicated models.

4.2 GITHUB CLASSIFIER STABILITY

We allow the adversary to modify input features by up to 3% (at each timestep) and our goal is to bound the maximum number of prediction switches induced by each attack over time. We can model this within our verification framework as follows: Given a sequence of input features, we compute the maximum number of switches



(a) Sigmoid

(b) Tanh

Figure 1: Figures show three curves per model - Dashed line: Lower bound from LBFGS attack. Solid line: Our verified upper bound. Dash-dot line: SDP verified upper bound from [Raghunathan et al., 2018]. Each color represents a different network. The dashed lines at the bottom are lower bounds on the error rate computed using the best attack found using the LBFGS algorithm.

achievable by first computing the target classes that are reachable through an adversarial attack at each timestep (using (7)), and then running a dynamic program to compute the choices of target classes over time (from within the reachable target classes) to maximize the number of switches over time.

Figure 3a shows how initially predictions are easily attackable (as little information is available to make predictions), and also shows how the gap between our approach and the best attack found using the LBFGS algorithm evolves over time.

4.3 COMPLEX VERIFICATION TASK: DIGIT SUM

In order to test our approach on a more complex specification, we study the following task: Given a pair of MNIST digits, we ask the question: Can an attacker perturb each image, subject to a constraint on the total perturbation across both digits, such that the sum of the digits predicted by the classifier differs from the true sum of those digits by as large an amount as possible? Answering this question requires solving the following optimization problem:

$$\begin{aligned}
&\max_{\substack{x_a^{in}, x_b^{in} \\ \epsilon_a, \epsilon_b}} \left| \operatorname{argmax}\left(x^L\left(x_a^{in}\right)\right) + \operatorname{argmax}\left(x^L\left(x_b^{in}\right)\right) - s \right| \\
&\quad \text{s.t. } \left\| x_a^{in} - x_a^{nom} \right\| \leq \epsilon_a, \left\| x_b^{in} - x_b^{nom} \right\| \leq \epsilon_b \\
&\qquad \epsilon_a + \epsilon_b \leq \epsilon
\end{aligned}$$

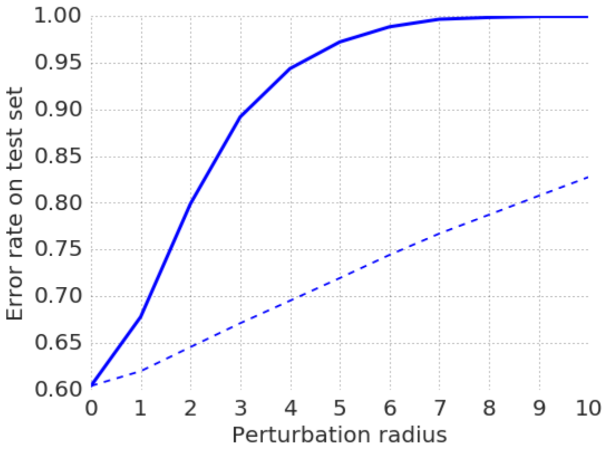
where s is the true sum of the two digits. Thus, the adversary has to decide on both the perturbation to each digit, as well as the size of the perturbation. We can encode this within our framework (we skip the details here). The upper bound on the maximum error in the predicted sum from the verification and the lower bound on the maximum error computed from an attack for this problem (on an adversarially trained two hidden layer sigmoid network) is plotted in figure 3b. The results show that even on this rather complex verification task, our approach is able to compute tight bounds.

5 CONCLUSIONS

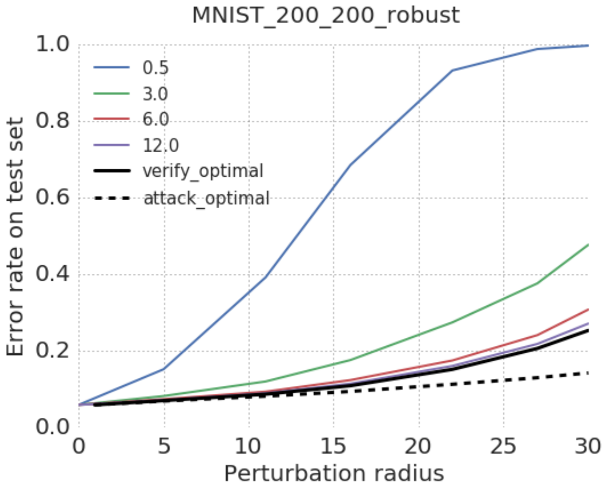
We have presented a novel framework for verification of neural networks. Our approach extends the applicability of verification algorithms to arbitrary feedforward networks with any architecture and activation function and to more general classes of input constraints than those considered previously (like cardinality constraints). The verification procedure is both efficient (given that it solves an unconstrained convex optimization problem) and practically scalable (given its anytime nature only required gradient like steps). We proved the first known (to the best of our knowledge) theorems showing that under special assumptions, nonlinear neural networks can be verified tractably. Numerical experiments demonstrate the practical performance of our approach on several classes of verification tasks.

ACKNOWLEDGEMENTS

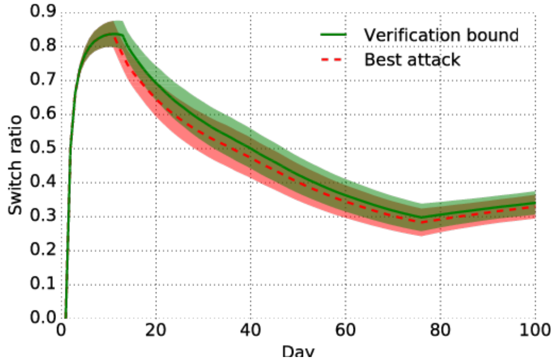
The authors would like to thank Brendan O'Donoghue, Csaba Szepesvari, Rudy Bunel, Jonathan Uesato and



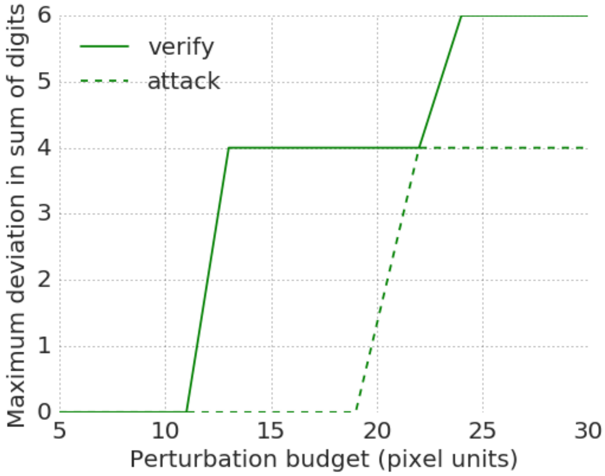
(a) Verification of robust CIFAR-10 model as a function of perturbation radius ϵ.



(b) Verification of robust models as a function of perturbation radius ϵ. Solid lines:verified upper bounds and dashed lines: attack lower bounds.



(a) Maximum number of switches (over number of days evaluated so far) over different days (upper bound from our verification approach and lower bound from the best attack found). Results are averaged over 100 unseen repositories. Shaded areas are 95% confidence intervals.



(b) Maximum difference between predicted and actual sum of digits vs the perturbation budget for a pair of randomly chosen MNIST images.

Shane Legg for helpful comments and feedback on this paper. Jonathan Uesato's help on adversarial training of neural networks is also gratefully acknowledged.

References

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

R. Bunel, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar. Piecewise linear neural network verification: A comparative study. *arXiv preprint arXiv:1711.00455*, 2017.

N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017a.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017b.

C.-H. Cheng, G. Nührenberg, and H. Ruess. Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 251–268. Springer, 2017.

R. Ehlers. Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks. *ArXiv e-prints*, May 2017.

A. Fawzi, H. Fawzi, and O. Fawzi. Adversarial vulnerability for any classifier. *arXiv preprint arXiv:1802.08686*, 2018.

J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres. *ArXiv e-prints*, Jan. 2018.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

E. Hazan and T. Koren. A linear-time algorithm for trust region problems. *Mathematical Programming*, 158(1-2):363–381, 2016.

M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2263–2273, 2017.

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In R. Majumdar and V. Kunčak, editors, *Computer Aided Verification*, pages 97–117, Cham, 2017. Springer International Publishing. ISBN 978-3-319-63387-9.

J. Z. Kolter and E. Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 2018.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

M. Marston and G. Baca. Acas-xu initial self-separation flight tests. *NASA Armstrong Flight Research Center Flight Report*, 2015.

B. Polyak. The convexity principle and its applications. *Bulletin of the Brazilian Mathematical Society*, 34(1): 59–75, 2003.

A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Bys4ob-Rb.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

V. Tjeng and R. Tedrake. Verifying neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

M. Udell and S. Boyd. Maximizing a sum of sigmoids. *Optimization and Engineering*, 2013.

J. Uesato, B. O'Donoghue, A. van den Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, 2018.

L. Vandenberghe and S. Boyd. *Convex optimization*, volume 1. Cambridge University Press Cambridge, 2004.

Y. Wang, S. Jha, and K. Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. *arXiv preprint arXiv:1706.03922*, 2017.

V. Yakubovic. S-procedure in nonlinear control theory. *Vestnik Leningrad Univ.*, 1:62–77, 1971.