# 神经网络形式化验证算法调研

智能系统的安全可信性问题不只是在人工智能或者机器学习领域得到关注，也 引起了形式化方法领域的关注。传统上,形式化方法分析和验证主要针对软硬件系统。尽管以神经网络为代表的智能系统也可以实现为软件或者硬件，一个显著的不同在于,传统形式化方法研究的软硬件系统一般有严格的逻辑描述或者结构化描述，也就是符号系统。而智能网络缺乏逻辑结构，神经网络是通过大量的神经元之间的连接构造起 来的系统，并通过优化算法来学习系统参数。从符号到亚符号系统的变迁使得形式化方法领域半个世纪来发展的大量技术并不能直接被用于智能系统的可信性研究。

下面我们从形式化方法的角度出发观察过去几年基于形式化方法的智能系统安全可信研究的进展。

# 1 方法分类

从方法所运用到的技术来分，大致可以分为三类：

1. 以可达性为基础方法
2. 以优化为基础的方法
3. 以搜索为基础的方法

# 2 基于可达性的方法

**可达性描述**：

在给定某个神经网络和输入集合的前提下，其对应的输出值会有一个可达集，称之为输出可达集。此类方法的原理是，一方面可以计算出这个输出可达集的某个近似估计，另一方面也可以检查这一输出可达集与某些安全规格（specification）的非（negation）的交集是否为空。

**相关论文及工具**：

【ExactReach】Xiang, W., H.-D. Tran, and T. T. Johnson. 2017. “Reachable set computation and

safety verification for neural networks with ReLU activations”. ArXiv. (1712.08163).

【Ai2】Gehr, T., M. Mirman, D. Drashsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev.

2018. “Ai2: Safety and robustness certification of neural networks with abstract

interpretation”. In: IEEE Symposium on Security and Privacy (SP).

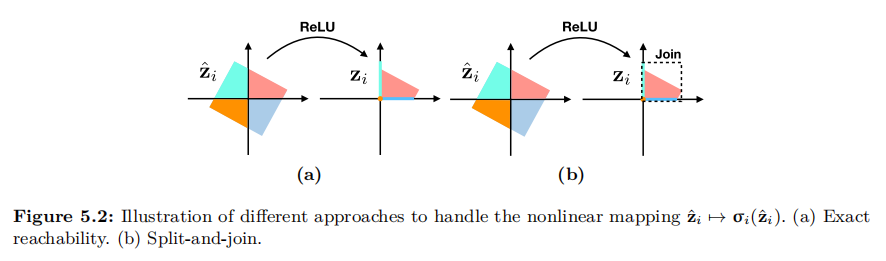
【MaxSens】Xiang, W., H. Tran, and T. T. Johnson. 2018a. “Output Reachable Set Estimation and

Verification for Multilayer Neural Networks”. IEEE Transactions on Neural Networks

and Learning Systems. 29(11): 5777–5783.

## 2.1 ExactReach

ExactReach 采用了几何计算解决了基于可达性的神经网络验证问题。ExactReach对具有线性或ReLU激活函数的网络执行精确的可达性分析。对于任何ReLU函数，如果输入集是多面体的并集，则输出可达集也是多面体的并集，如图5.2a所示。该方法可以用H-多面体或星集实现。



特点：对于一个输入多面体，ExactReach方法每经过一个隐藏层就会生成2ki个多边形（ki是第i层的约束数量）。因此多边形的数量随着深度的增加呈指数级增长。虽然空集可以在这个过程中被删除，但跟踪大型神经网络的精确可达集仍然显得效率低下。

解决这个问题的方法是并行计算个别几何对象，或者开发新的数据结构来处理线性和非线性映射。例如之后的星集。

## 2.2 Ai2

ExactReach所采用的实现方法有明显的缺陷，在许多情况下，精确可达集是难以处理的。因此，在Ai2中使用了可达集的估计，它满足原精确可达集是估计可达集的子集。

Ai2使用抽象域来近似每一层的可达集，该集合由捕获某些几何形状的一组逻辑公式表示，例如半空间多面体、超矩形、全对称多面体、星集、半空间等。抽象域的选择需要在精度和可扩展性之间取得平衡。 例如，在ExactReach中使用的多面体是精确的，但可扩展性差，而在MaxSens中使用的超矩形是可缩放的，但太松散。

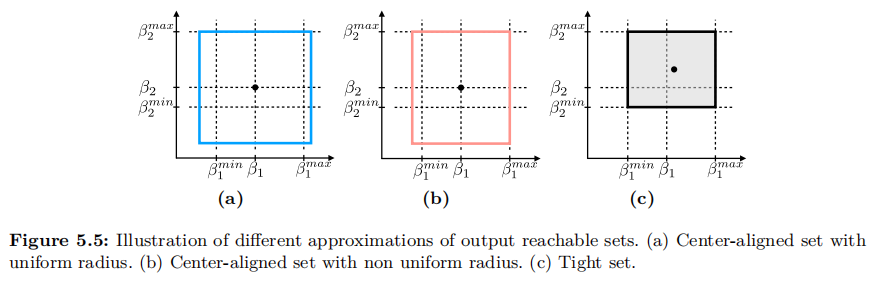
特点：Ai2的原始实现方式使用了zonotopes，即中心对称的凸封闭多面体，它比多面体更具可伸缩性，比超矩形更紧密。相比于其他两种方法，这种方法显得更加均衡。

## 2.3 MaxSens

MaxSens是一种使用过近似的可达性方法。它适用于具有单调激活函数以及低维输入输出的网络。MaxSens的关键思想是对输入空间进行网格化，并计算每个网格单元的可达集。 网格单元越精细，则过逼近度越小。 计算不同小区的可达集可以并行完成。

有三种方法来定义过近似：

1. 均匀半径的中心对齐集合，对应a) 图
2. 非均匀半径的中心对齐集合，对应b) 图
3. 紧密集合，对应c) 图



特点：与其他可达性方法相比，MaxSens的优势在于，在逐层传播期间，几何对象的数量不会增加。 超矩形的总数仅取决于初始分区。尽管在逐层传播期间超矩形的数目不会增加，但是相对于层数，过近似造成的误差将迅速累积。对于具有许多输入节点的网络，在严格估计时，初始分区上的超矩形数量可能会过大。否则，使用稀疏分区的计算将过于保守。

# 3 基于优化的方法

**优化问题的描述**

优化方法试图伪造断言。为输入集，为合理的输出集。在优化方法中，网络结构是需要考虑的一个约束。现有方法仅适用于ReLU激活函数。利用ReLU中的分段线性，将网络编码为一组线性约束或混合整数线性约束的。返回的结果可以是反例(CounterExample)，也可以是最大允许扰动(Adversarial)，具体取决于优化的目标。

**相关论文及工具**：

【ILP】Bastani, O., Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi. 2016. “Measuring neural net robustness with constraints”. In: Advances in Neural Information Processing Systems (NIPS).

【NSVerify】Lomuscio, A. and L. Maganti. 2017. “An approach to reachability analysis for feedforward relu neural networks”. ArXiv. (1706.07351)

【MIPVerify】Tjeng, V., K. Xiao, and R. Tedrake. 2017. “Evaluating robustness of neural networks with mixed integer programming”. ArXiv. (1711.07356).

## ILP

ILP（迭代线性规划）通过在参考点处对网络进行线性化，将激活函数为ReLU的神经网络编码为一组线性约束。ILP中的优化问题是一个对抗性问题，试图估计输入侧的最大允许干扰。 迭代地解决优化问题。由于ILP只考虑网络中激活模式相同的一段线性部分，该方法不完备。

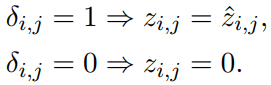
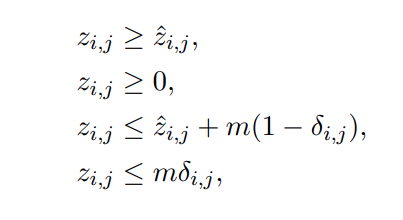
图3.1.1

图3.1.2

在迭代方法中，求解器首先使用3.1.1中的松弛线性约束对网络进行编码，然后求解松弛问题。 如果结果解决方案违反了任何不等式约束3.1.2，则将约束添加到问题中，然后再次解决问题。 重复该过程，直到满足所有约束条件。 因为只有有限的许多约束，所以该过程可以收敛于有限的步骤。 约束的数量等于神经元的数量。

## NSVerify

NSVerify是一个基于混合整形线性规划（ MILP ）的神经网络形式化验证工具。它将全连接(fully-connected)神经网络编码成混合整数规划，譬如对于神经网络的某个隐藏层来说，其上的ReLU激活函数就可以被表达成一系列混合整数规划，便计算输出范围(output range)。网络中的节点关系编码成了基于Big-M的约束，由于ReLU节点的存在，引入了整形变量，从而使用混合整形线性规划求解器求解。编码方式如下：

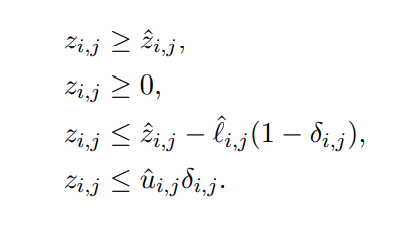


m必须是一个足够大的数，如果m的取值小于最大上界，这种编码会导致约束出现错误。

特点：由于每一个ReLU节点就会引入一组二元变量，因此这种编码方式能解决的网络规模较小，效率不高，并且很难估计m的值。但是这种方法是绝对完备的。

## MIPVerify

MIPVerify 可以看作NSVerify的直接的延伸。它同样把网络编码成混合整形的约束。但是它的编码效率更高，因为它在NSVerify基础上引入了区间算术，在求解之前对网络中节点的边界进行了预处理，预计算出了隐藏层中节点的近似上界和近似下界，从而有效地束紧了m的范围



特点：MIPVerify和NSVerify都是绝对完备的并且MIPVerify在NSVerify的基础上做了有效的优化。

# 4基于搜索的方法

基于搜索的方法会结合可达性方法或优化方法一起使用，从而降低整体的时间复杂度，常见的基于搜索的方法有：分支定界法、SAT、SMT等。

相关论文：

【sherlock】 Dutta, S., S. Jha, S. Sanakaranarayanan, and A. Tiwari. 2017. “Output range analysisfor deep neural networks”. ArXiv. (1709.09130).

【planet】Ehlers, R. 2017. “Formal verification of piece-wise linear feed-forward neural networks”.In: International Symposium on Automated Technology for Verification and Analysis.

【BaB】Bunel, R. R., I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda. 2018. “A unified view of piecewise linear neural network verification”. In: Advances in Neural Information Processing Systems.

【DLV】Huang, X., M. Kwiatkowska, S. Wang, and M. Wu. 2017. “Safety verification of deep

neural networks”. In: International Conference on Computer Aided Verification.

【ReLUVal】Wang, S., K. Pei, J. Whitehouse, J. Yang, and S. Jana. 2018b. “Formal Security Analysis of Neural Networks using Symbolic Intervals”. In: USENIX Security Symposiu.

【Neurify】Wang, S., K. Pei, J. Whitehouse, J. Yang, and S. Jana. 2018a. “Efficient formal safety analysis of neural networks”. In: Advances in Neural Information Processing Systems.

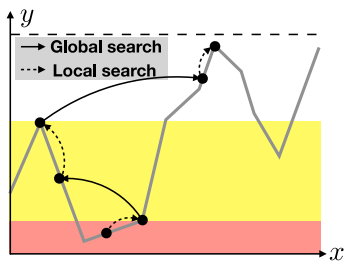
【Reluplex】Katz, G., C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. 2017. “Reluplex: An efficient SMT solver for verifying deep neural networks”. In: International Conference on Computer Aided Verification.

【FastLin/FastLip】Weng, L., H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon. 2018a. “Towards Fast Computation of Certified Robustness for ReLU Networks”. In: International Conference on Machine Learning (ICML). Vol. 80. Proceedings of Machine Learning Research.

## Sherlock

Sherlock一次只能计算出一个输出节点的上下界，通过结合局部搜索和全局搜索来计算输出节点的上下界。在全局搜索中，它求解混合整形线性规划问题，以检查是否可以优化当前的局部最优边界，从而脱离局部极值。在局部搜索中，求解器从一个样本点开始，求解一个线性规划问题，在f的相同线段上找到一个与x处激活模式相同且更好的可行解。为了找到全局最优边界，求解器迭代执行局部搜索和全局搜索。

但sherlock的缺点是，这个方法是不完备的，它有可能无法判断安全属性是满足的还是不满足的。



## Planet

Planet结合SAT以及线性规划问题，并提出了以下四点思路：

1. 对神经网络的整体行为进行近似。

2. 基于弹性滤波的线性约束的不可约不可行子集分析。

3. 从部分节点相位估值的可行性检验中推断出可能的安全节点相位选择

4. 对节点相位进行单元传播类推理。

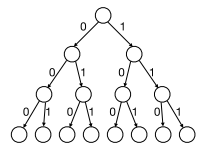
并且与MILP不同的是，Planet提出了一种近似的描述约束的方法——三角松弛。这种方法需要提前知道变量的上下界。对于神经网络的输入节点，比如图像处理中，输入向量中一维的范围一般在[0, 1]之间。在其他网络中，通常也需要将输入值归一化，在验证的时候也是如此，因此需要再网络上使用经典的区间算法。

Planet还使用了弹性过滤。在这种方法中，所有由于节点相位选择而添加的约束都将被松弛变量所削弱，其中每个节点都有一个对应的松弛变量。当以最小化松弛变量的加权总和为目标再次运行线性规划求解器时，将根据节点对冲突的贡献程度对节点进行排名，其中一些节点根本没有贡献（因为它们的松弛变量的值为0）。然后，将最大值的松弛变量固定为0，从而使相应的约束变得严格，并重复搜索过程，直到生成的LP实例变得不可行为止。然后可以知道，在此过程中严格执行的节点相位固定装置在一起已经不可行，并且建立了仅包含它们的冲突子句。最后可以观察到，这些冲突子句比没有应用弹性过滤时要短得多。

特点: Planet集成了SAT求解器，可以在状态空间中进行树搜索。搜索的目的是找到一组ReLU网络的激活模式，该模式将X中的输入映射到输出。它在搜索过程中结合了基于优化的过滤和简直。并且Planet是完备的。

## BaB

BaB使用分支定界法来计算网络的输出范围。它具有模块化设计，可以用作支持其他方法（例如Reluplex和Planet）的统一框架。BaB与Sherlock是比较相似的，都估计网络的输出边界，但是估计边界的方法是不同的。

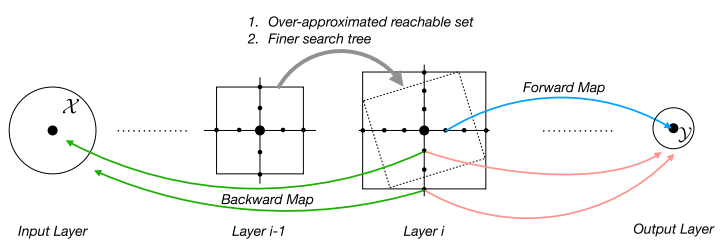


BaB提出了concrete bounds 和 approximated bound，BaB在整个输入域上进行二分，并不断更新concrete bounds和approximated bound的值，如果说真实上界的上界与下界的差小于某个指定的阈值，那么就推断上界被找到。

特点：BaB的缺点是这种方法是不完备的。

## DLV

DLV是一种特殊的方法，它不仅仅在输入空间进行搜索，而且还在在隐藏层中逐层搜索对抗性输入。



DLV使用逐层分析。 在第i层，它根据第i−1层中的可达集和搜索树，计算出一个过近似的可达集和一个更精细的搜索树。对于搜索树上的每个点，DLV首先使用前向映射检查其是否违反输出约束Y。对于违反输出约束的点，DLV然后使用后向映射检查在给定输入约束X的情况下是否可到达。如果可以到达该点，则将相应的输入值作为反例。

特点：DLV的优势是可以支持所有的激活函数。

## ReLUVal

Reluval 是一个将搜索和可达性方法结合的工具。它使用了一种新的方式来计算节点的边界，当某个节点的输入为负时，通过ReLu节点时不能继续用线性方程来保持符号区间。这时，我们具体化它的上限和下限。这种具体化是有误差的，为了最小化这种情况导致的误差，该方法使用了迭代精细。通过结合这两种方法可以获得非常紧密的边界。对于利普希兹连续的函数来说依赖误差随着区间宽度的下降而下降。同时具有限层的DNN是利普希兹连续的。利用这一事实，可以分割输入的区间，将区间平均分为两个子区间的结合，减少区间范围的高估。最终通过重复的分割输入区间来精细输出区间。这些操作是可以高度并行化的。它的效率比较之前的工具有了极大的提升。但是只适用于全连接网络。

## Neurify

Nurify是Reluval的延伸版本。和Reluval一样，它的算法结构也是通过符号来进行可达性分析，通过迭代的进行区间精细来进行搜索。和Reluval相比，主要有两个修改：第一，符号分析中Neurify使用的具体化技术为符号的线性松弛，符号线性松弛和符号传播过程类似，唯一的区别就是传播过ReLu节点时，符号松弛会对ReLu节点进行三角松弛。在计算可达集时这种松弛会比符号区间分析紧密很多。第二，相对于Reluval的迭代精细它使用了直接的约束精细，迭代的去精细化超近似集合，它会对处于非线性部分ReLu激活函数进行分割，而不仅仅对输入进行分割，大大提升搜索效率。该方法是完备的，效率是Reluval的数倍，但是依旧只适用于ReLu网络。

## Reluplex

Reluplex是一个基于SMT的求解器。在该方法中，每个ReLu节点都被分为两个节点：激活前节点和激活后节点。在Reluplex中允许变量临时超出它的边界，因为它会迭代的寻找一个合适的赋值,然后迭代的选择超出边界或者违反ReLu的节点，最后使用单纯形方法的Pivot和Update操作纠正它们的值。直到找到一个反例(UNSAT)或者找不到反例(SAT)。该方法还使用了浮点运算，边界推理，冲突分析等技术来提升其性能和可拓展性。这个工具是完备的。但是因为它会遍历所有的可能赋值，所以只支持小型的网络，用该工具验证一个包含300个节点的全连接网络需要数个小时。效率不高。

## FastLin

通过计算出最大允许扰动的下界来验证网络是否安全。这个方法结合可达性分析和二分搜索来估计边界，从而返回一个最大允许扰动(Adversarial Result).

## 4.9 FastLip

FastLip依赖Fastlin去计算节点的边界，然后估计网络的局部李普希兹常量。一般来说，FastLin比FastLip延展性更好，但是后者对于边界能够提供更好的解决方法。