Проверка моделей как средство верификации нейронных сетей

Соколов Павел Павлович

Зачем нужна верификация?[1]

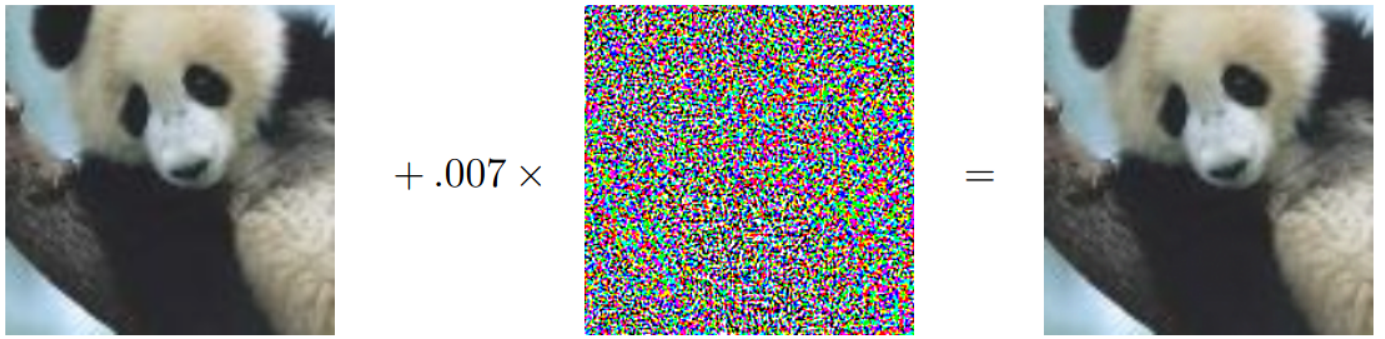


Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

[1]EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES. Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy

Зачем нужна верификация — II[2]

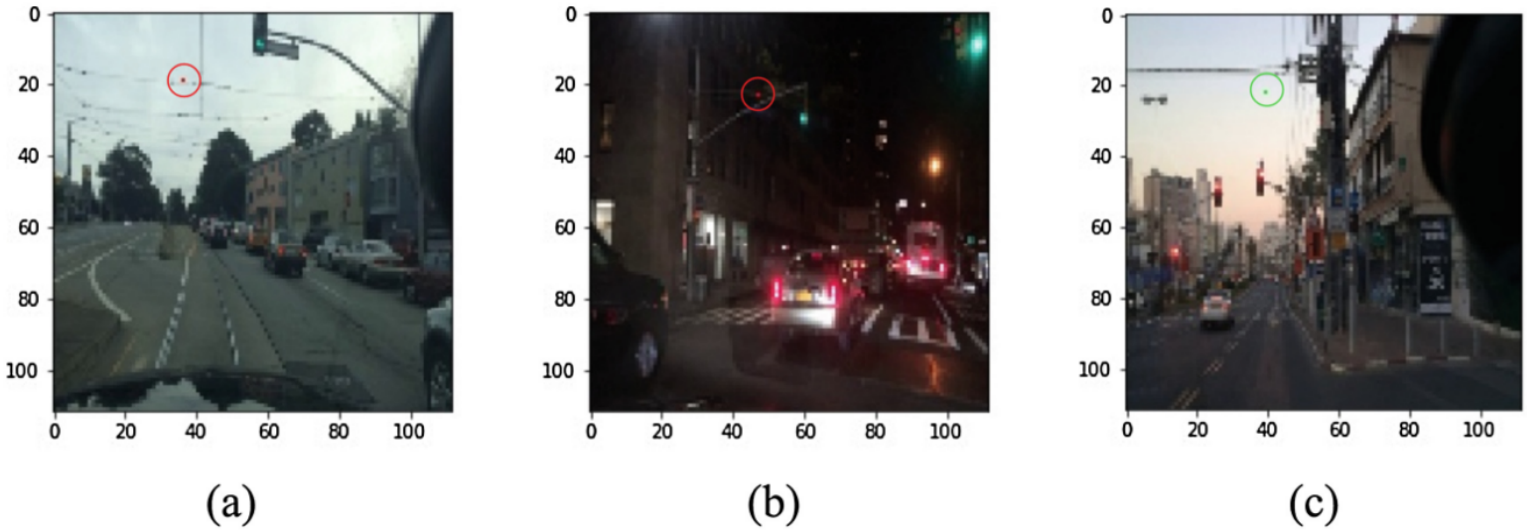


Fig. 4. Adversarial examples generated on Nexar data demonstrate a lack of robustness. (a) Green light classified as red with confidence 56% after one pixel change. (b) Green light classified as red with confidence 76% after one pixel change. (c) Red light classified as green with 90% confidence after one pixel change. (Color figure online)

[2]Feature-Guided Black-Box Safety Testing of Deep Neural Networks. Matthew Wicker, Xiaowei Huang & Marta Kwiatkowska

Задачи верификации нейронных сетей

- В задаче классификации: робастность — малые изменения входных данных не должны изменять результат.
- В самоуправляемых устройствах: безопасность — устройство не сталкивается с препятствиями; витальность — устройство доезжает до точки назначения.
- В языковых моделях — ???
- ...

Робастность классификации — I[3]

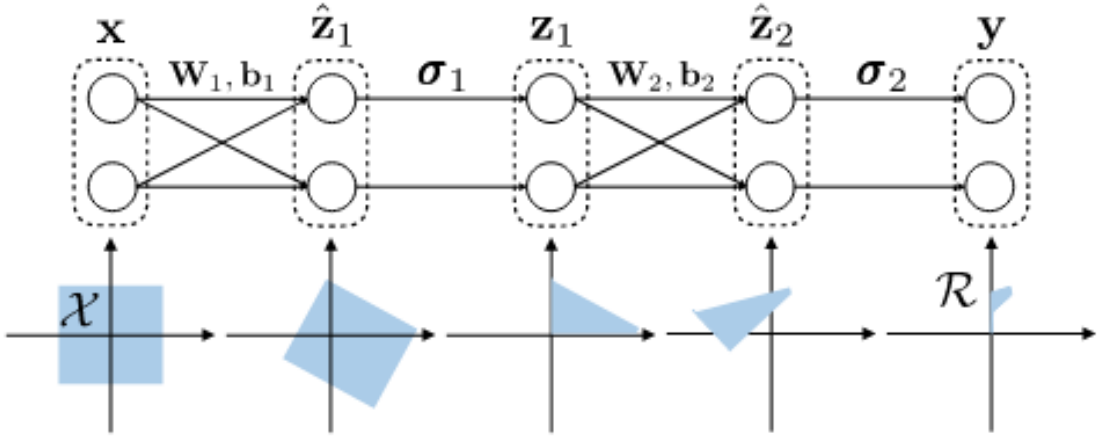
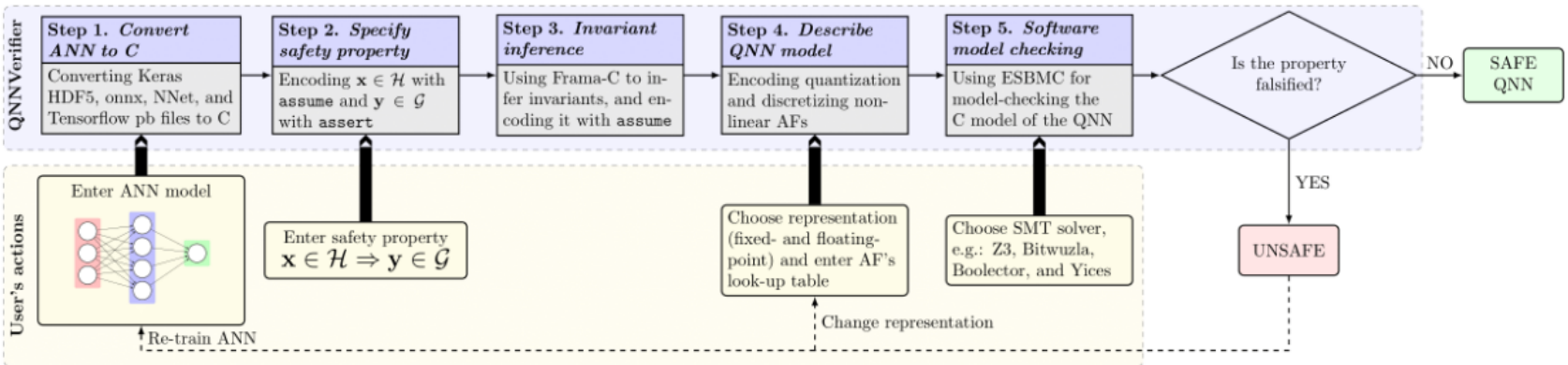


Figure 5.1: Illustration of reachability methods. The network in the illustration only contains one hidden layer. The input set $\mathcal{X}$ is first passed through the linear mapping defined by $\mathbf{W}_1$ and $\mathbf{b}_1$. Then it goes through the nonlinear mapping defined by $\boldsymbol{\sigma}_1$ (ReLU is considered). The corresponding reachable sets are illustrated in the shaded area. The process is repeated for the next layer and the output reachable set is then obtained.

[3] Algorithms for Verifying Deep Neural Networks.
Changliu Liu, Tomer Arnon et al.

Робастность классификации — II[4]



[4]QNNVerifier: A Tool for Verifying Neural Networks using SMT-Based Model Checking. Xidan Song, Luiz Sena, Mikel Luján et al.

Робастность классификации — III[5]

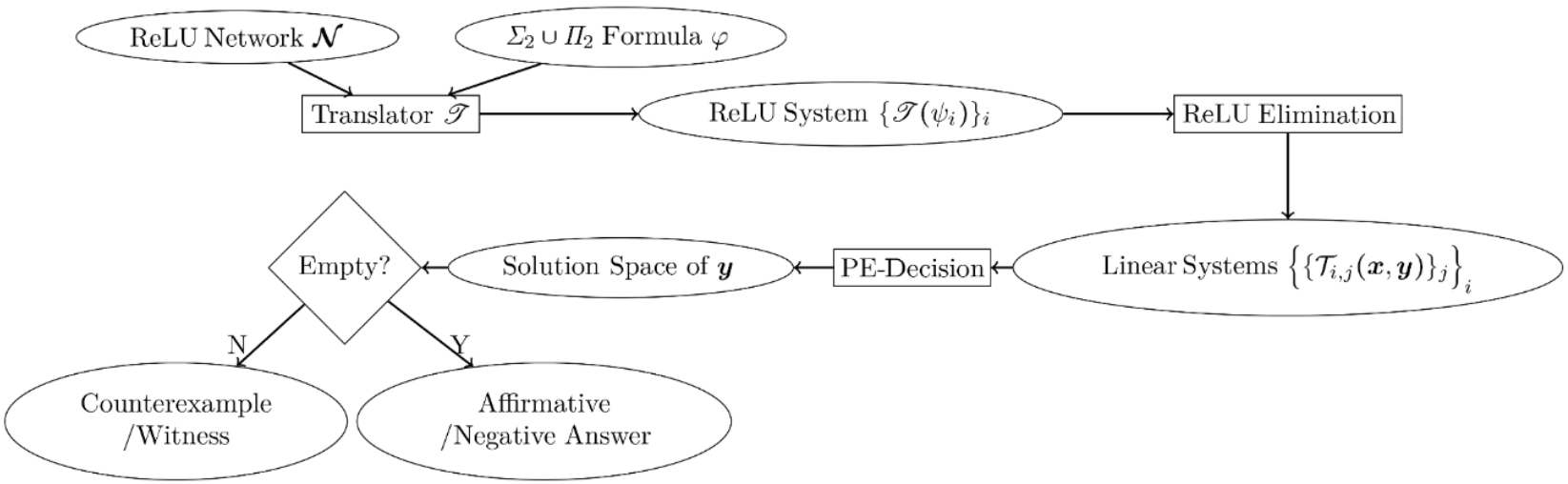


Fig.1. Framework for $\Sigma_2 \cup \Pi_2$ formula model checking.

[5]Verifying ReLU Neural Networks from a Model Checking Perspective. Wan-Wei Liu, Fu Song, Tang-Hao-Ran Zhang and Ji Wang

Темпоральная логика ReTL

Термы и формулы

$$\mathrm{t} ::= \mathrm{v} \mid \mathrm{x} \mid \bigcirc^{\mathrm{k}} \mathrm{t} \mid \mathrm{t}[\mathrm{i}] \mid \mathrm{M}\mathrm{t} \mid \mathrm{t} + \mathrm{t}$$

$$\phi ::= \mathrm{t}_1 \sim \mathrm{t}_2 \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \mathrm{X}\phi \mid \phi_1 \mathrm{U} \phi_2 \mid \exists \mathrm{x} \in \mathbb{R}^{\mathrm{n}}.\phi \mid \forall \mathrm{x} \in \mathbb{R}^{\mathrm{n}}.\phi$$

Элиминация ReLU

$$\mathrm{ReLU}(\mathrm{t}) = \mathrm{E}_{\mathrm{U}}\mathrm{t}$$

при условии

$$\begin{cases} \mathrm{E}_{\mathrm{U}}\mathrm{t} \geqslant 0 \\ (\mathrm{I} - \mathrm{E}_{\mathrm{U}})\mathrm{t} \leqslant 0 \end{cases}$$

Deep statistical model checking[6]

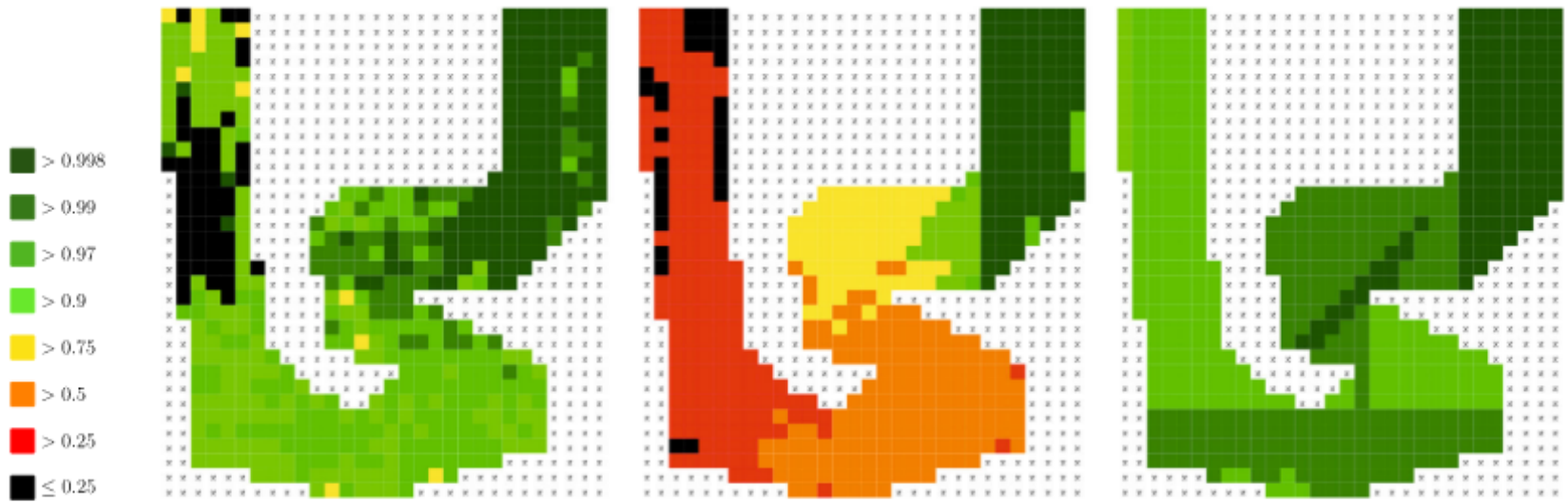


Fig. 4 Goal probability of NN oracle on the Barto-big benchmark trained and executed with 20% noise versus stress-test executed with 50% noise using the same NN (middle) versus optimal policies obtained by probabilistic model checking with 50% noise (right)

[6] Analyzing neural network behavior through deep statistical model checking. Timo P. Gros, Holger Hermanns, Jörg Hoffmann et al.

Deep statistical model checking — I

Определение

Марковский процесс принятия решений — кортеж $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathrm{s}_0 \rangle$, состоящий из конечного множества состояний $\mathcal{S}$, конечного множества действий $\mathcal{A}$, частичной функции вероятности перехода $\mathcal{T} : \mathcal{S} \times \mathcal{A} \hookrightarrow \mathcal{D}(\mathcal{S})$ (где $\mathcal{D}$ — множество распределений вероятности) и начального состояния $\mathrm{s}_0 \in \mathcal{S}$.

Определение

(Детерминированная, зависящая от истории) политика действий — функция $\sigma : \mathcal{S}^+ \to \mathcal{A}$ такая, что $\forall \mathrm{w} \in \mathcal{S}^*, \mathrm{s} \in \mathcal{S} : \sigma(\mathrm{ws}) \in \mathcal{A}(\mathrm{s})$.

("Беспамятная" политика — такая, что для любых w,w′ $\sigma(\mathrm{ws}) = \sigma(\mathrm{w}'\mathrm{s})$)

Deep statistical model checking — II

Определение

Цепь Маркова — кортеж $\mathcal{C} = \langle \mathcal{S}, \mathcal{T}, \mathrm{s}_0 \rangle$, состоящий из множества состояний $\mathcal{S}$, функции вероятности перехода $\mathcal{T} : \mathcal{S} \to \mathcal{D}(\mathcal{S})$ и начального состояния $\mathrm{s}_0 \in \mathcal{S}$.

Заметим, что применение политики действий $\sigma : \mathcal{S}^+ \to \mathcal{A}$ к Марковскому процессу принятия решений $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathrm{s}_0 \rangle$ порождает цепь Маркова на множестве состояний $\mathcal{S}^+$. (Беспамятная политика позволяет сузить множество состояний до $\mathcal{S}$). Как формулировать и верифицировать свойства такой цепи?

Statistical model checking

Синтаксис PCTL

$$\phi ::= \top \mid \mathrm{a} \mid \neg\phi \mid \phi \wedge \phi' \mid \mathrm{P}_{\geqslant\theta}(\psi)$$

$$\psi ::= \phi \mid \mathrm{X}\phi \mid \phi \mathrm{U}^{\leqslant \mathrm{t}} \phi' \mid \phi \mathrm{U} \phi'$$

Техники верификации

На данный момент только численными методами.[7]

[7] A Survey of Statistical Model Checking. Gul Agha and Karl Palmskog