Дедуктивная верификация искусственного интеллекта

Кондратьев Дмитрий Александрович

Институт систем информатики им. А. П. Ершова СО РАН

Проблема: надежность и корректность искусственного интеллекта

В настоящее время искусственный интеллект, основанный на нейронных сетях, стал активно применяться в программном обеспечении таких систем, к надежности и корректности которых предъявляются повышенные требования.

В качестве примера таких систем можно привести

- беспилотные авиационные системы
- беспилотные транспортные системы
- роботоризированные системы
- экспертные системы в банковской сфере

и т.д.

От тестирования программного обеспечения к формальной верификации программ

Известно, что тестирование не может гарантировать корректность и надежность программного обеспечения.

Следовательно, тестирование не может гарантировать корректность и надежность такого программного обеспечения, которое реализует искусственный интеллект.

Гарантировать корректность и надежности программного обеспечения может только формальная верификация программ.

Формальная верификация программ

Входными данными формальной верификации являются программа и ее спецификации, описывающие те свойства программы, выполнение которых нужно проверить.

Формальная верификация позволяет доказать, что программа корректна относительно своих спецификаций.

Основные виды формальной верификации:

- дедуктивная верификация
- проверка на модели (model checking).

Актуальность формальной верификации искусственного интеллекта

Формальную верификацию можно применить и к программному обеспечению, реализующему искусственный интеллект, чтобы сделать искусственный интеллект доверенным.

В отличие от тематики формальной верификации программного обеспечения общего назначения, тематика формальной верификации искусственного интеллекта стала активно развиваться только последние три года.

Итого, проблема формальной верификации искусственного интеллекта, основанного на нейронных сетях, является актуальной.

Дедуктивная верификация программного обеспечения

Дедуктивная верификация позволяет свести задачу проверки корректности программы относительно спецификаций к задаче проверки истинности утверждений о том, что программа корректна относительно спецификаций

Такие утверждения являются логическими формулами и называются условиями корректности программы

Если все условия корректности являются истинными, то программа корректна относительно спецификаций

Дедуктивная верификация основана на генерации условий корректности верифицируемой программы

Входными данными для генерации условий корректности являются верифицируемая программа и ее спецификации

Истинность условий корректности проверяется с помощью программных систем для доказательства теорем

Этапы дедуктивной верификации программ

1. Задание спецификаций верифицируемой программы
2. Генерация условий корректности для верифицируемой программы и ее спецификаций
3. Проверка истинности условий корректности с помощью программных систем для доказательства теорем (например, с помощью SMT-решателей Z3, CVC5 или с помощью интерактивных систем доказательства теорем Coq, ACL2, Isabelle/HOL)
4. Если все условия корректности истинны, то верифицируемая программа корректна отнносительно спецификаций.
5. Если истинность какого-либо из условий корректности не удается доказать или удается доказать ложность какого-либо из условий корректности, то наступает этап локализации ошибок.

Рассмотрим основной этап дедуктивной верификации программ, этап генерации условий корректности

Генерация условий корректности программ

Генерация условий корректности программ основана на логике Хоара

В логике Хоара программа и ее спецификации представляются в виде тройки Хоара:

$$\{P\}\, S\, \{Q\}$$

где

- P — предусловие (логическая формула)
- S — программа
- Q — постусловие (логическая формула)

Частичная корректность тройки Хоара $\{P\}\, S\, \{Q\}$ означает, что "если предусловие P истинно перед исполнением фрагмента программы S, и, если исполнение S завершилось, тогда постусловие Q выполняется после его завершения".

Логика Хоара

Генерация условий корректности основана на правилах вывода логики Хоара. Схема правила вывода в логике Хоара:

$$\frac{\{P_1\}\, S_1\, \{Q_1\},\ \ldots\ \{P_n\}\, S_n\, \{Q_n\},\ \gamma_1,\ \ldots\ \gamma_m}{\{P\}\, S\, \{Q\}}$$

где

- $\{P_1\}\, S_1\, \{Q_1\},\ \ldots\ \{P_n\}\, S_n\, \{Q_n\}$ — n посылок правила вывода в виде троек Хоара
- $\gamma_1,\ \ldots\ \gamma_m$ — m посылок правила вывода в виде условий корректности (логических формул).
- $\{P\}\, S\, \{Q\}$ — заключение правила вывода (тройка Хоара)

Генерация условий корректности происходит в направлении от заключений правил вывода к посылкам правил вывода. Генерация условий корректности порождает дерево и останавливается, когда все листья полученного дерева представляют собой условия корректности.

Правила вывода для базовых конструкций языка

Правило вывода для пустой программы:

$$\frac{P \rightarrow Q}{\{P\}\text{emptyProgram}\{Q\}}$$

где $\rightarrow$ обозначает импликацию.

Правило вывода для присваивания переменной:

$$\frac{\{P\}\ \text{prog};\ \{Q(var \leftarrow expr)\}}{\{P\}\ \text{prog};\ \text{var=expr}\ \{Q\}}$$

где $Q(var \leftarrow expr)$ обозначает замену всех вхождений var в Q на $expr$, prog обозначает остальную часть программы.

Правило вывода для *if*:

$$\frac{\{P \wedge B\}\ \text{S}_1;\ \text{prog}\ \{Q\},\ \{P \wedge \neg B\}\ \text{S}_2;\ \text{prog}\ \{Q\}}{\{P\}\ \text{if B then S}_1\ \text{else S}_2;\ \text{prog}\ \{\text{Q}\}}$$

где $\wedge$ обозначает конъюнкцию, $\neg$ – отрицание.

Такие правила вывода задают формальную семантику конструкций языка программирования

Пример генерации условий корректности

$$\{true\}\ \text{if}\ (\mathsf{x} == 0)\ \{\mathsf{y} = 1;\}\ \text{else}\ \{\mathsf{y} = 2;\}$$
$$\{((x = 0) \to (y = 1)) \wedge ((x \neq 0) \to (y = 2))\}$$

Используем правило вывода для *if*

- $\{true \wedge (x = 0)\}\ \mathsf{y} = 1;$
  $\{((x = 0) \to (y = 1)) \wedge ((x \neq 0) \to (y = 2))\}$
  Используем правило вывода для присваивания
  $\{true \wedge (x = 0)\}\{((x = 0) \to (1 = 1)) \wedge ((x \neq 0) \to (1 = 2))\}$
  Используем правило вывода для пустой программы и получим условие корректности: $(true \wedge (x = 0)) \to (((x = 0) \to (1 = 1)) \wedge ((x \neq 0) \to (1 = 2)))$
- $\{true \wedge (x \neq 0)\}\ \mathsf{y} = 2;$
  $\{((x = 0) \to (y = 1)) \wedge ((x \neq 0) \to (y = 2))\}$
  Используем правило вывода для присваивания
  $\{true \wedge (x \neq 0)\}\{((x = 0) \to (2 = 1)) \wedge ((x \neq 0) \to (2 = 2))\}$
  Используем правило вывода для пустой программы и получим условие корректности: $(true \wedge (x \neq 0)) \to (((x = 0) \to (2 = 1)) \wedge ((x \neq 0) \to (2 = 2)))$

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Xie X., Kersting K., Neider D. Neuro-Symbolic Verification of Deep Neural Networks. 2022. DOI: `https://doi.org/10.48550/arXiv.2203.00938`

Важность исследования, описанного в данной статьи:

- В данной статье описан прототип языка NeSAL для задания спецификаций, описывающих функциональные свойства нейронных сетей
- В данной статье впервые описан подход к дедуктивной верификации нейронных сетей. Данный подход назван нейросимволическим подходом.

Рассмотрим данную статью

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

В целях дедуктивной верификации рассмотрим нейронную сеть $f : \mathbb{R}^m \to \mathbb{R}^n$ как расширенный граф $G_f = (V, V_I, V_O, E, \alpha)$, где

- V — конечное множество вершин (т.е., нейронов),
- V_I являются входными нейронами,
- V_O являются выходными нейронами ($V_I \cap V_O = \emptyset$),
- $E \subseteq V \times \mathbb{R} \times V$ — отношение для дуг с весами,
- α является отображением, назначающим функцию активации (например, линейный выпрямитель, сигмоид, и д.р.) каждому нейрону в $V \setminus V_I$.

Без потери общности, мы можем мы опускаем смещения, поскольку их легко включить в определения функций активации. Более того, мы полагаем, что входные нейроны $V_I = \{v_{in,1}, \ldots, v_{in,m}\}$ и выходные нейроны $V_O = \{v_{out,1}, \ldots, v_{out,n}\}$ неявно упорядочены, отражая порядок входов и выходов сети.

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Тройка Хоара с нейронной сетью

$$P \coloneqq \left\{\varphi_{pre}(\vec{x})\right\}\ \varphi_{assign}(\vec{x},\vec{y})\ \left\{\varphi_{post}(\vec{x},\vec{y})\right\},$$

где

- $\vec{x} = (x_1, \ldots, x_m)$,
- $\vec{y} = (y_1, \ldots, y_n)$,
- $\varphi_{assign}(\vec{x},\vec{y}) \coloneqq \vec{y} \leftarrow f(\vec{x})$,
- $f : \mathbb{R}^m \to \mathbb{R}^n$ — нейронная сеть,
- $G_f = (V, V_I, V_O, E, \alpha)$ — графовое представление f.

Далее будем определять правило вывода для тройки Хоара с нейронной сетью

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Пусть выходному значению каждого нейрона $v \in V$ соответствует переменная X_v. Тогда каждому нейрону (кроме входных) соответствует формула φ_v:

$$\varphi_v \;:=\; X_v = \alpha(v)\left(\sum_{(v',w,v)\in E} w \cdot X_{v'}\right),$$

В случае такой функции активации, как линейный выпрямитель, зададим φ_v с помощью вспомогательной Y_v:

$$\left(Y_v = \sum_{(v',w,v)\in E} w \cdot X_{v'}\right) \wedge \left(\left(Y_v \le 0 \to X_v = 0\right) \wedge \left(Y_v > 0 \to X_v = Y_v\right)\right),$$

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Формула, соответствующая всей нейронной сети:

$$\varphi_f \;:=\; \bigwedge_{v \in V \setminus V_I} \varphi_v$$

Правило вывода, задающее формальную семантику нейронной сети:

$$\frac{\left(\varphi_{pre}[\vec{x} \,/\, \vec{X}_{V_I}] \wedge \varphi_f\right) \rightarrow \varphi_{post}[\vec{x} \,/\, \vec{X}_{V_I}, \vec{y} \,/\, \vec{X}_{V_O}]}{\left\{\varphi_{pre}(\vec{x})\right\} \vec{y} \leftarrow f(\vec{x}) \left\{\varphi_{post}(\vec{x}, \vec{y})\right\}}$$

где $\varphi[\vec{z}_1 \,/\, \vec{z}_2]$ обозначает результат замены в φ вектора переменных z_1 на z_2.

Условие корректности нейронной сети:

$$\psi \;:=\; \left(\varphi_{pre}[\vec{x} \,/\, \vec{X}_{V_I}] \wedge \varphi_f\right) \rightarrow \varphi_{post}[\vec{x} \,/\, \vec{X}_{V_I}, \vec{y} \,/\, \vec{X}_{V_O}],$$

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Случай нескольких нейронных сетей:

$$\left\{\varphi_{pre}(\vec{x}_1,\ldots,\vec{x}_\ell)\right\}\ \vec{y}_1 \leftarrow h_1(\vec{x}_1)\wedge\cdots\wedge\vec{y}_\ell \leftarrow h_\ell(\vec{x}_\ell)\ \left\{\varphi_{post}(\vec{x}_1,\ldots,\vec{x}_\ell,\vec{y}_1,\ldots,\vec{y}_\ell)\right\},$$

где $h_i \in \{f, g_1, \ldots, g_k\}$ для $i \in \{1,\ldots,\ell\}$ являются функциональными символами, соответствующими не только сети f, но и другим нейронным сетям. Правило вывода:

$$\frac{\begin{array}{c}\left(\varphi_{pre}[\vec{x}_1,\ldots,\vec{x}_\ell \,/\, \vec{X}_{V_I^1},\ldots,\vec{X}_{V_I^\ell}]\wedge\varphi_f\wedge\varphi_{g_1}\wedge\cdots\wedge\varphi_{g_\ell}\right)\rightarrow \\ \varphi_{post}[\vec{x}_1,\ldots,\vec{x}_\ell \,/\, \vec{X}_{V_I^1},\ldots,\vec{X}_{V_I^\ell},\vec{y}_1,\ldots,\vec{y}_\ell \,/\, \vec{X}_{V_O^1},\ldots,\vec{X}_{V_O^\ell}]\end{array}}{\left\{\varphi_{pre}(\vec{x}_1,\ldots,\vec{x}_\ell)\right\}\ \vec{y}_1 \leftarrow h_1(\vec{x}_1)\wedge\cdots\wedge\vec{y}_\ell \leftarrow h_\ell(\vec{x}_\ell)\ \left\{\varphi_{post}(\vec{x}_1,\ldots,\vec{x}_\ell,\vec{y}_1,\ldots,\vec{y}_\ell)\right\}}$$

Условие корректности:

$$\left(\varphi_{pre}[\vec{x}_1,\ldots,\vec{x}_\ell \,/\, \vec{X}_{V_I^1},\ldots,\vec{X}_{V_I^\ell}]\wedge\varphi_f\wedge\varphi_{g_1}\wedge\cdots\wedge\varphi_{g_\ell}\right)\rightarrow \varphi_{post}[\vec{x}_1,\ldots,\vec{x}_\ell \,/\, \vec{X}_{V_I^1},\ldots,\vec{X}_{V_I^\ell},\vec{y}_1,\ldots,\vec{y}_\ell \,/\, \vec{X}_{V_O^1},\ldots,\vec{X}_{V_O^\ell}]$$

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Свойство состязательной устойчивости (adversarial robustness). Необходимо доказать, что нейронная сеть устойчива к небольшим изменениям входных данных (т. е. небольшие изменения входных данных не меняют выходные данные). Чтобы сделать это математически точно, предположим, что нам дана многоклассовая нейронная сеть $f : \mathbb{R}^m \to \{c_1, \ldots, c_n\}$ с m признаками и n классами, конкретным входом $\vec{x}^\star \in \mathbb{R}^m$, функция расстояния $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$, и расстояние $\varepsilon \geq 0$. Тогда задача — доказать, что

$$d(\vec{x}^\star, \vec{x}) \leq \varepsilon \text{ implies } f(\vec{x}^\star) = f(\vec{x})$$

для всех входных данных $\vec{x} \in \mathbb{R}^m$. Другими словами, классы $\vec{x}^\star$ и каждого входа на расстоянии не более ε от $\vec{x}^\star$ должны совпадать. Вход $\vec{x} \in \mathbb{R}^m$, нарушающий свойство состязательной устойчивости, называется состязательным примером (adversarial example) и свидетельствует о том, что f не является состязательно устойчивой.

Обзор статьи о нейросимволическом подходе к дедуктивной верификации искусственного интеллекта

Запись свойства состязательной устойчивости (adversarial robustness) на языке NeSAL:

$$\left\{d(\vec{x}^\star, \vec{x}) \leq \varepsilon\right\} \vec{y}^\star \leftarrow f(\vec{x}^\star) \wedge \vec{y} \leftarrow f(\vec{x}) \left\{\vec{y}^\star = \vec{y}\right\}$$

где $\vec{x}^\star \in \mathbb{R}^m$ является конкретным входом, $\varepsilon \geq 0$.

Преимущества такого подхода:

- Если сгенерированное для такой тройки Хоара условие корректности удастся доказать, то выполнение свойства состязательной устойчивости гарантируется.
- Если система доказательства обнаружила контрпример к условию корректности, сгенерированному для данной тройки Хоара, то в таком случае мы автоматически получаем состязательный пример (adversarial example).
- Язык NeSAL позволяет записывать спецификации, описывающие свойства нейронных сетей

Планируемое исследование

Цель исследования состоит в разработке комплексного подхода к формальной верификации нейронных сетей и реализации данного подхода в программной системе для формальной верификации нейронных сетей.

Комплексный подход будет включать

- Язык для задания спецификаций нейронных сетей (набор конструкций языка)
- Формальная семантика нейронных сетей (набор правил вывода)
- Набор методов задания модели

Программная система, в которой будет реализован комплексный подход, позволит пользователям задавать на полученном языке спецификации нейронных сетей и автоматически проверять корректность нейронных сетей относительно данных спецификаций.

Перспективы

- Исследование формальной семантики искусственного интеллекта
- Доверенный искусственный интеллект на основе формальной верификации
- Гарантированные свойства искусственного интеллекта

Дедуктивная верификация искусственного интеллекта

Кондратьев Дмитрий Александрович

Институт систем информатики им. А. П. Ершова СО РАН