Machine Learning in sicherheitskritischen Systemen

Aktuelle Probleme und Möglichkeiten der Absicherung

Adrian Schwaiger, Fraunhofer IKS, Abteilung für zertifizierbare Künstliche Intelligenz

Der Einsatz von Machine Learning (ML), insbesondere von Deep Learning, ermöglicht erst viele hochkomplexe Anwendungen, beispielsweise in der Medizintechnik oder bei autonomen Systemen. Derzeit gibt es beim Einsatz in solchen sicherheitskritischen Systemen jedoch noch einige Herausforderungen. Drei dieser Probleme und Möglichkeiten, wie diese in Zukunft gehandhabt werden können, sollen im Nachfolgenden am Beispiel des autonomen Fahrens vorgestellt werden.

Autonome Fahrzeuge sind mit einer Vielzahl verschiedener Sensoren ausgestattet – von Kameras über Radar- und Lidarsysteme hin zu Ultraschallsensoren. Diese liefern kontinuierlich große Mengen an Daten über ihre Umgebung. Allerdings können diese Daten, in ihrer unverarbeiteten Form, nicht ohne Weiteres für die Steuerung der Fahrzeuge genutzt werden. Hier kommt die Perzeption ins Spiel, das heißt die Interpretation dieser Sensordaten. Die präzise und verlässliche Wahrnehmung der Umwelt ist dabei essenziell für autonome Fahrzeuge: Von ihr hängt die nachfolgende Steuerung des Fahrzeugs maßgeblich ab.

Perzeption als Grundlage autonomer Systeme

Dabei ist verlässliche Perzeption nicht trivial, vor allem nicht, wenn die Systeme in verschiedensten Situationen eingesetzt werden sollen [1]. Dort ist zu berücksichtigen, dass die Sensoren beeinträchtigt werden können, etwa durch unterschiedliche Wetterbedingungen. Ebenso ist es möglich, dass Situationen und Objekte auftreten, welche so zuvor noch nicht beobachtet wurden. Mit klassischen, handgefertigten Algorithmen lässt sich diese Komplexität jedoch nicht erfassen. Deshalb kommt hier vor allem ML zum Einsatz, insbesondere tiefe neuronale Netze (Deep Neural Networks, DNN). Diese müssen selbstverständlich ebenfalls abgesichert werden. Das wiederum wirft aufgrund der datengetriebenen Natur der Algorithmen neue Herausforderungen auf. Drei dieser Herausforderungen im Kontext der Perzeption werden nachfolgend vorgestellt.

Zuverlässige Quantifizierung der Unsicherheit

Moderne DNNs sind sich häufig zu sicher in ihren Vorhersagen. Dies ist unter anderem dadurch bedingt, dass durch die immer größeren Netzarchitekturen insgesamt mehr und mehr Evidenzen bezüglich der einzelnen Vorhersagemöglichkeiten gesammelt werden. Im abschließenden Schritt wird die Wahrscheinlichkeitsverteilung über diese Vorhersagemöglichkeiten erstellt. Durch den Einsatz der Softmax-Funktion kommt es schließlich dazu, dass, den Vorhersagemöglichkeiten mit den meisten Evidenzen eine überproportional hohe Wahrscheinlichkeit zugewiesen wird. Der Grund dafür liegt in der exponentiellen Natur der Softmax-Funktion.

Gerade in sicherheitskritischen Bereichen ist es jedoch erforderlich, dass neuronale Netze zuverlässig ihre Unsicherheit ausdrücken. Das ermöglicht es, präventiv zu handeln und potentiell kritische Situationen zu vermeiden.

Es gibt dabei einige vielversprechende Lösungsansätze, darunter vor allem Bayesian Neural Networks (BNNs) [2]. Hierbei werden für die Neuronengewichte, anstelle einzelner Skalare, Wahrscheinlichkeitsverteilungen gelernt. Dadurch stellt die Konfidenz bei der Prädiktion keinen Punktschätzer sondern ebenfalls eine Wahrscheinlichkeitsverteilung dar. Ein großes Problem bei BNNs ist jedoch, dass das Training bedeutend mehr Rechenaufwand erfordert, ebenso die Inferenz, da hier mehrmals aus den Verteilungen der Netzparameter gezogen werden muss, um eine einzelne Prädiktion zu erhalten.

Für die Praxis relevanter ist hierfür der Ansatz der Deep Ensembles [3]. Die Idee ist dabei – wie auch bei anderen Ensembling-Methoden im ML –, dass mehrere DNNs zunächst separat trainiert werden und zu einem Ensemble zusammengefasst werden. Für die Inferenz werden schließlich alle Einzelprädiktionen der Teilnetzwerke kombiniert und durch Mittelwert- und Varianzbildung lässt sich die Unsicherheit quantifizieren, welche sich dabei bedeutend zuverlässiger als bei Standard-DNNs [4] zeigt. Deep Ensembles sind dabei zwar deutlich effizienter als BNNs, aber gerade im Bereich eingebetteter Systeme noch mit einem zu hohen Rechenaufwand verbunden. Für diesen Bereich sind vor allem Ansätze der Ensemble-Destillation [5] oder One-Shot-Verfahren [6] interessant.

Erkennung unbekannter Eingaben

Ein weiteres Problem tritt auf, wenn einem DNN ein Konzept präsentiert wird, welches es während der Trainingsphase nicht gesehen hat. Denn auch hier wird das Netzwerk versuchen, Muster in der unbekannten Eingabe zu finden, welche zu den bekannten Konzepten passen. Beispielsweise würde ein Klassifikator – trainiert darauf, Hunde von Katzen zu unterscheiden – auch bei einem Elefanten den beiden trainierten Klassen Wahrscheinlichkeiten zuweisen, welche sich zu 1 aufsummieren.

Der Grund hierfür ist, dass das Netzwerk auch in dem Elefantenbild Evidenzen in Form von Mustern findet, welche es gelernt hat, um Hunde von Katzen zu unterscheiden. Auch wenn dabei womöglich insgesamt weniger übereinstimmende Muster gefunden werden, wird am Ende dennoch eine Wahrscheinlichkeitsverteilung über die bekannten Kategorien gebildet, aus welcher diese Tatsache nicht mehr erkenntlich wird. Das Erkennen solcher unbekannten Eingaben, auch als Out-of-Distribution Detection oder Novelty Detection bekannt, ist notwendig, da es sonst zu einer fehlerhaften Wahrnehmung der Umwelt kommen kann. Weiterführende Entscheidungen basieren damit auf falschen Annahmen, was schließlich ein Fehlverhalten des Gesamtsystems auslösen kann.

Der naheliegende Ansatz – die Einführung einer eigenen Kategorie für unbekannte Konzepte – ist allerdings nur schwer realisierbar, da hierfür die Muster aller unbekann-

ten Konzepte gelernt werden müssten. Vielversprechender sind Ansätze, die zur Laufzeit beobachten, welche Neuronen zu einem gegebenen Zeitpunkt aktiviert sind [7]. Dieses Aktivierungsmuster wird verglichen mit den während des Trainings beobachteten Mustern. Unterscheiden sich diese zu sehr, so kann es sich um ein nicht gelerntes Konzept handeln.

Eine weitere Möglichkeit ist, ein Netzwerk zu trainieren, welches eine stark vereinfachte Repräsentation der bekannten Konzepte erstellen und aus diesen Repräsentationen wieder die Ausgangsbilder rekonstruieren kann [8]. Bei bekannten Konzepten ist der Unterschied zwischen den Ein- und Ausgabebildern dabei gering, neuartige Konzepte hingegen weisen einen deutlich höheren Fehler bei der Rekonstruktion auf.

Interpretierbarkeit Neuronaler Netze

Die letzte in diesem Beitrag betrachte Herausforderung ist die unzureichende Erklärbarkeit von DNNs.

Diese haben, gerade im Bereich der Perzeption, üblicherweise mehrere Millionen Parameter. In Folge ist es einem Menschen nicht mehr möglich, zu beurteilen, auf welchen Grundlagen das DNN eine Entscheidung getroffen hat. In diesem Zusammenhang spricht man auch von einem DNN als Black Box. Für eine Argumentation bezüglich der Sicherheit eines Systems, welches ein DNN einsetzt, wäre es jedoch äußert hilfreich, den Prädiktionsprozess transparenter zu gestalten. Damit könnte sichergestellt werden, dass ein Netzwerk auch die richtigen Merkmale für die Vorhersage nutzt und nicht beispielsweise Eisbären von Grizzlies aufgrund des häufigeren Vorkommens von Schnee bei ersteren unterscheidet.

Im Bereich der erklärbaren KI gibt es dabei qualitative und quantitative Lösungsansätze. Erstere befassen sich beispielsweise damit, auf dem Eingabebild zu visualisieren, wie viel Einfluss jeder einzelne Pixel auf das Ergebnis hatte [9]. Man spricht dabei vom Heatmapping oder von Saliency Maps. Eine weitere Methode ist das Visualisieren der gelernten Muster und wie diese über die verschiedenen Ebenen des Netzwerkes miteinander interagieren [10]. Für eine vollständige Sicherheitsargumentation erweisen sich qualitative Methoden jedoch nicht als ausreichend. Sie führen jedoch zu Erkenntnissen, welche bei der Entwicklung neuronaler Netze und bei der Analyse von Fehlern helfen, die nie vollständig ausgeschlossen werden können.

Im Gegensatz dazu geht es bei quantitativen Ansätzen unter anderem um das Lernen von Konzepten und Mustern, die auch für den Menschen interpretierbar sind. So könnte ein DNN beispielsweise trainiert werden, Fahrzeuge anhand konkreter Bestandteile wie Reifen oder Windschutzscheiben zu erkennen. Diese Bestandteile wiederum können ebenfalls in konkrete Komponenten aufgegliedert werden.

Herkömmliche neuronale Netze lernen zwar ebenfalls Hierarchien von Mustern und Konzepten, der Unterschied liegt darin, dass bei diesen Ansätzen des erklärbaren Lernens explizit festgelegt wird, welche Komponenten gelernt werden sollen oder bereits gelernte Muster zu konkreten, interpretierbaren Konzepten zusammengefasst werden [12].

Einbettung in ein sicheres Gesamtsystem

Die vorgestellten Lösungsansätze für die Probleme neuronaler Netze sollen diese robuster machen und stellen weitere Informationen zur Verfügung, um ihren Zustand zu beurteilen. Zusätzlich sind jedoch sichere Vorgehensweisen und Architekturen um die DNNs herum notwendig. So müssen beispielsweise neue Ansätze gefunden werden, um Sicherheitsargumentationen auch für datengetriebenen ML-Ansätze zu ermöglichen [13].

Auch das umgebende System muss entsprechend angepasst werden, wenn ML zum Einsatz kommen soll. So müssen die zusätzlichen Informationen, beispielsweise über die Unsicherheit einer Prädiktion, auch sinnvoll weiterverarbeitet werden. Ein Konzept hierfür sind die Safety Envelopes [14]. Diese können eine potentiell unsichere ML-Komponente abschotten und dynamisch Ihren Einsatz nur bei gegebener Zuverlässigkeit erlauben. Für den Fall, dass der ML-Komponente zu einem Zeitpunkt nicht vertraut wird, wird ein Sicherheitspfad ausgeführt, welcher durch eine geringere Komplexität leistungsschwächer, dafür jedoch einfacher absicherbar ist. Abbildung 1 verdeutlicht dieses Konzept.

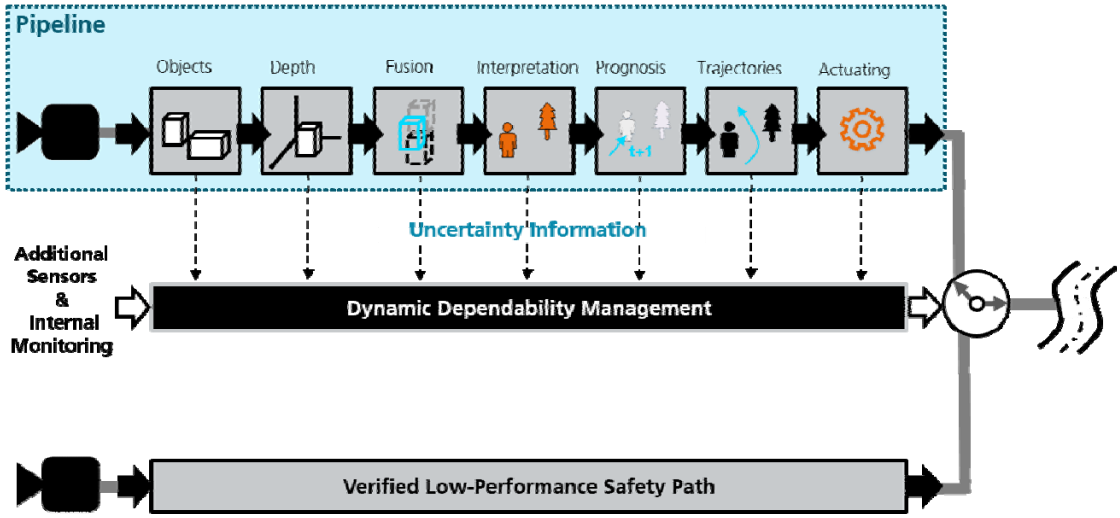

Abbildung 1: Integration von ML in ein dynamisches Sicherheitsmanagement

Fazit

Aktuell ist der Einsatz von ML, insbesondere von DNNs, in sicherheitskritischen Systemen nur schwer möglich. Vor allem in den letzten Jahren gibt es jedoch immer mehr Ansätze, um DNNs selbst sicherer zu machen und um geeignete Architekturen und Vorgehensweisen zu entwerfen, die den Einsatz in Anwendungen mit hohen Sicherheitsanforderungen erlauben. Auch wenn es in dieser Richtung noch großen Forschungsbedarf gibt, so ist deutlich abzusehen, dass ML zunehmend auch Einzug in sicherheitskritischen Bereichen finden wird.

Literatur- und Quellenverzeichnis

[1] K. Czarnecki and R. Salay, “Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving,” in *Computer Safety, Reliability, and Security*, Cham, 2018, pp. 439–445.

[2] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, Lille, France, Jul. 2015, pp. 1613–1622.

[3] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Proc. NeurIPS 30*, Curran Associates, Inc., 2017, pp. 6402–6413.

[4] M. Henne, A. Schwaiger, K. Roscher, and G. Weiss, “Benchmarking Uncertainty Estimation Methods for Deep Learning With Safety-Related Metrics,” in *Proc. SafeAI@AAAI 2020*, 2020, vol. 2560, pp. 83–90.

[5] A. Malinin, B. Mlodozeniec, and M. Gales, “Ensemble Distribution Distillation,” *arXiv:1905.00076 [cs, stat]*, Nov. 2019.

[6] A. Malinin and M. Gales, “Predictive Uncertainty Estimation via Prior Networks,” *arXiv:1802.10501 [cs, stat]*, Nov. 2018.

[7] C.-H. Cheng, G. Nührenberg, and H. Yasuoka, “Runtime Monitoring Neuron Activation Patterns,” in *2019 Design, Automation Test in Europe Conference Exhibition*, Mar. 2019, pp. 300–303.

[8] S. Pidhorskyi, R. Almohsen, and G. Doretto, “Generative Probabilistic Novelty Detection with Adversarial Autoencoders,” in *Proc. NeurIPS 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 6822–6833.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization,” 2017, pp. 618–626.

[10] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter, “Zoom In: An Introduction to Circuits,” *Distill*, vol. 5, no. 3, p. 10.23915/distill.00024.001, Mar. 2020.

[11] Q. Zhang, Y. N. Wu, and S.-C. Zhu, “Interpretable Convolutional Neural Networks,” in *Proc. CVPR* 2018, pp. 8827–8836.

[12] B. Kim *et al.*, “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV),” in *Proc. ICML 2018*, Jul. 2018, pp. 2668–2677.

[13] S. Burton, L. Gauerhof, B. B. Sethy, I. Habli, and R. Hawkins, “Confidence Arguments for Evidence of Performance in Machine Learning for Highly Automated Driving Functions,” in *Computer Safety, Reliability, and Security*, Cham, 2019, pp. 365–377.

[14] G. Weiss, P. Schleiss, D. Schneider, and M. Trapp, “Towards integrating undependable self-adaptive systems in safety-critical environments,” in *Proc. SEAMS 2018,* New York, NY, USA, May 2018, pp. 26–32.

Autor

Nach Abschluss seines Studiums in angewandter Informatik mit Fokus auf Machine Learning, trat Adrian Schwaiger eine Stelle als wissenschaftlicher Mitarbeiter beim Fraunhofer Institut für kognitive Systeme an. Sein Forschungsschwerpunkt liegt dabei auf der Absicherung von Machine-Learning-Verfahren, etwa durch die Quantifizierung der Unsicherheiten in der Prädiktion oder durch die Erkennung von neuartigen Eingaben. Anwendungsgebiete sind dabei die Bilderkennung in autonomen Fahrzeugen, unter anderem im Rahmen des ADA Lovelace Centers, und automatisierte Analysen auf strukturierten Daten.

Kontakt

Internet: https://www.iks.fraunhofer.de/
Email: adrian.schwaiger@iks.fraunhofer.de