

Candy WALTER, Hildesheim

„Lesende Maschinen“ – Maschinelle Lernalgorithmen im Einsatz der Mathematikdidaktik

Ein häufiges Problem beim Auswerten offener Fragen

Frei formulierte Antworten sind eine wertvolle Quelle für Informationen. Sie liefern Hinweise, wie Personen über einen bestimmten Sachverhalt denken. Offene Fragen lassen in der Regel tiefere Einblicke auf vorhandene Kompetenzen, kontextbasierte Vorstellungen und Meinungen zu, weil sie im Gegensatz zu geschlossenen Fragen die Wahrscheinlichkeit herabsetzen, eine vorgegebene Antwort durch Raten richtig zu ermitteln (Krosnick & Presser, 2010). An der Universität Hildesheim müssen alle Lehramtsstudierende in den Vorlesungen zur Mathematikdidaktik für den Bachelor- und Masterstudiengang als Studienleistungen eine didaktische Aufgabe pro Woche schriftlich bearbeiten. Jede Bearbeitung wird den Studierenden mit bestanden oder nicht bestanden zurückgemeldet. In diesem Rahmen müssen bei 13 Sitzungen pro didaktische Veranstaltung mit je 120 Studierenden 1560 bearbeitete Aufgaben gelesen und bewertet werden. Im Vergleich zu geschlossenen Fragen ist die Auswertung offener Fragen dementsprechend *zeitaufwendig und kostenintensiv*.

Maschinelles Lernen bietet hierfür eine komfortable Lösung unter Verwendung modernster linguistischer Technologie. Trainierte Algorithmen erzeugen *künstliche Intelligenzen*, mit denen es möglich ist, eine vollautomatische Textanalyse vornehmen zu lassen.

Maschinelles Lernen – Automatisierte Textanalyse mithilfe trainierter Algorithmen

Beim *supervised machine learning* (dt. überwacht maschinelles Lernen) wird ein Algorithmus so trainiert, dass er Zusammenhänge in den eingespeisten annotierten (gelabelten) Daten erkennt und auf neue (ihm unbekannte) Daten anwendet. Hierdurch lassen sich beispielsweise frei formulierte Antworten schnell auf richtig oder falsch bzw. in bestanden oder nichtbestanden klassifizieren. Das heißt, dass beim Training für jede Eingabe (des Trainingsdatensatzes) die richtige Antwort vorliegt. Mithilfe dieser annotierten Trainingsdaten wird nun ein Algorithmus trainiert, der anhand der Trainingsdaten lernt und automatisch für jede neue Eingabe eine möglichst korrekte „Antwort“ zurückgibt. Der Algorithmus muss beim Training selbständig ein Muster in den Daten erkennen und daraus eine Regel ableiten.

Das Projekt „Content classification of open tasks“ an der Universität Hildesheim

Durch einen mit *Python* geschriebenen *Deep Learning Algorithmus* sollen offene Fragestellungen aus den Mathematikdidaktischen Vorlesungen vollautomatisch mithilfe des *supervised learning* maschinell ausgewertet werden. Abbildung 1 verdeutlicht das supervised learning am Beispiel einer binären Textklassifikation, wie sie im Projekt realisiert wird.

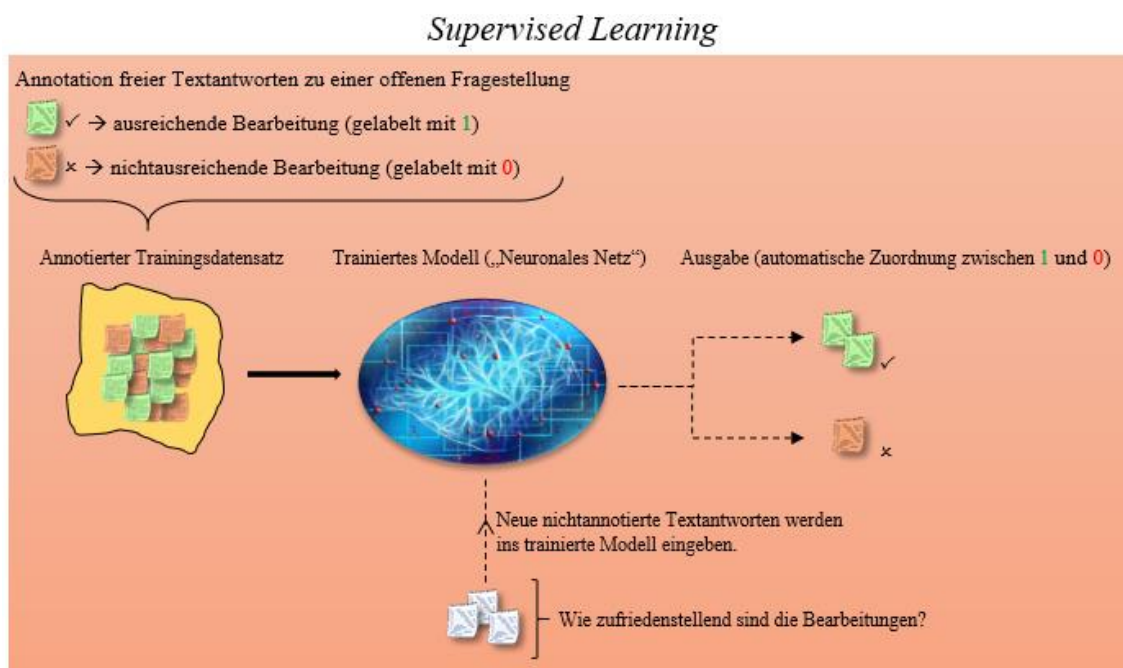


Abb. 1: Verdeutlichung einer vollautomatischen Textklassifikation in bestanden ✓ und nicht bestanden x

„Lesende Maschinen“ – Neuronale Modelle zur Textklassifikation nutzen

Maschinelle Textklassifizierung beschreibt eine ganze Klasse von Problemen, wie Sentimentanalysen oder die Klassifizierung von E-Mails als Spam oder nicht Spam. Maschinelle-Lern-Algorithmen erweisen sich als äußerst effizient bei Anwendungen, die das *Natural Language Processing* (dt. Verarbeitung natürlicher Sprache) betreffen, also auch für die Klassifikation von Textdaten. Der Modus Operandi für die Textklassifizierung beinhaltet für gewöhnlich die Verwendung von *Word Embeddings* (zur Repräsentation der Wörter im Vokabular durch einen Vektor) und eines *Convolutional Neural Networks* (Brownlee, 2017, S. 146 f.).

Wie lernt ein neuronales Netz?

Nachfolgend wird exemplarisch beschrieben, wie eines unserer neuronalen Netze lernt und bezogen auf die Aufgabenstellung „Erklären Sie, wofür dass

nach J. Bruner bekannte EIS-Prinzip steht“ eine vollautomatische Textklassifikation vornimmt.

- Zunächst wurde die Struktur des künstlichen neuronalen Netzes festgelegt. Dabei wurden neun Dichte-Schichten (dense-layers) eingesetzt, so dass ein *convolutional neural network* (CNN) entstand (Deru & Ndiaye, 2019, S. 80). Der Modellaufbau erfolgte *sequential* mithilfe von Googles Framework *TensorFlow* und der in TensorFlow integrierten High-Level-API *Keras* (ebd., Kap. 6).
- Für den Lernprozess wurden folgende Trainingsparameter festgelegt: Bei einer *Lernrate* von $1 \cdot 10^{-6}$ (Adam-Optimierer) wurden 20000 Trainingseinheiten durchgeführt. Neben einem *Validationssplit* von 0.1 wurden von der Menge des vorhandenen Datensatzes zusätzlich 10 Prozent zum Modelltesten zurückgehalten. Zum momentanen Zeitpunkt bestand der gesamte Datensatz aus 401 Studentenantworten.
- Anschließend erfolgte die Einspeisung der Trainingsdatensets in das neuronale Netz.
- Während des Trainings wurde eine Fehlerevaluation mithilfe der in Keras integrierten Kostenfunktion *binary-crossentropy-Function* durchgeführt. Der Fehler sollte möglichst minimal sein. Im Projekt wurde der Optimierungsgradient *Adam* verwendet (Géron, 2017, S. 300-302).
- Um zu gewährleisten, dass das neuronale Netz hinreichend verallgemeinert, wurden 10 Prozent der Trainingsdaten in das trainierte Netz eingespeist und die Präzision des Modells gemessen. Hierbei wurde verlangt, dass das Modell nahezu fehlerfrei arbeitet. Unter Berücksichtigung der geringen Datenmenge konnte eine Genauigkeit von ca. 0.97 Prozent bei einem Fehler von ca. 0.23 Prozent erreicht werden (vgl. Abb. 2).

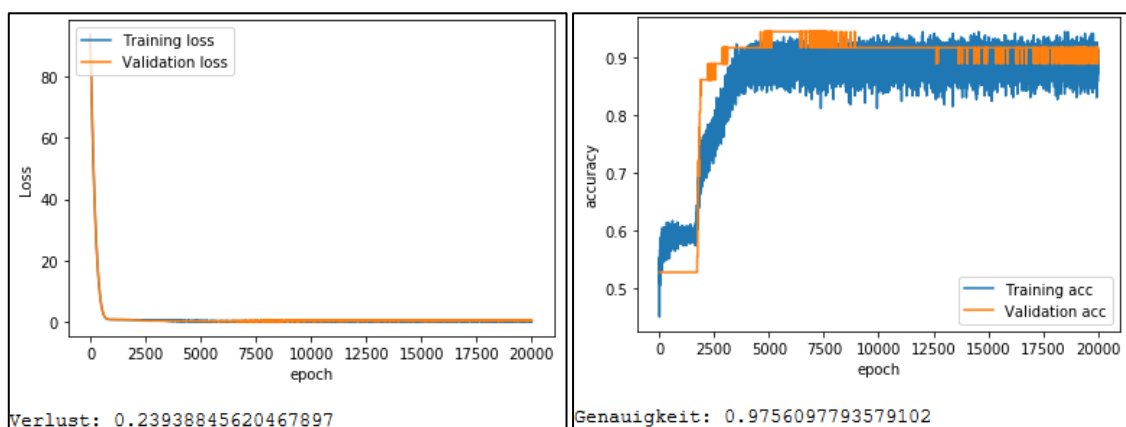
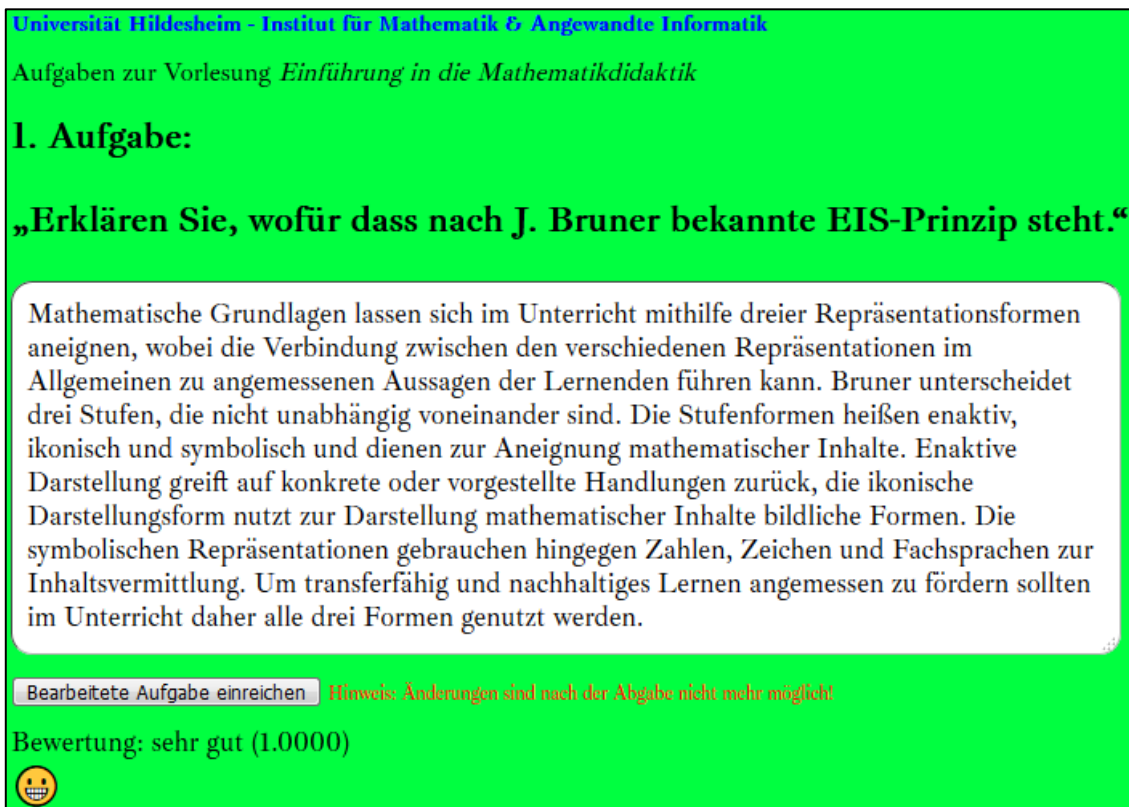


Abb. 2: Lernkurven nach 20000 Trainingseinheiten. Der Fehler ist mit ca. 23% Verlust sehr hoch

Aufgabenbearbeitung und Analyse im Browser

Damit die Aufgaben durch die Studierenden am Rechner, Laptop etc. bearbeitet werden können, muss das im Vorfeld trainierte Modell mit *TensorFlow.js* in eine HTML fähige Seite eingebunden werden (u. a. Deru & Ndiaye, 2019, Kap. 8). Die Studierenden tippen ihre Antworten ins Textfeld. Die Bewertung erfolgt vollautomatisch im Hintergrund. Abbildung 3 verdeutlicht die Analyse einer Studentenantwort im Firefox-Browser. Die Zahl 1,0000 gibt neben der Bewertung „sehr gut“ die prozentuale Genauigkeit der analysierten Antwort zurück.



Universität Hildesheim - Institut für Mathematik & Angewandte Informatik

Aufgaben zur Vorlesung *Einführung in die Mathematikdidaktik*

I. Aufgabe:

„Erklären Sie, wofür das nach J. Bruner bekannte EIS-Prinzip steht.“

Mathematische Grundlagen lassen sich im Unterricht mithilfe dreier Repräsentationsformen aneignen, wobei die Verbindung zwischen den verschiedenen Repräsentationen im Allgemeinen zu angemessenen Aussagen der Lernenden führen kann. Bruner unterscheidet drei Stufen, die nicht unabhängig voneinander sind. Die Stufenformen heißen enaktiv, ikonisch und symbolisch und dienen zur Aneignung mathematischer Inhalte. Enaktive Darstellung greift auf konkrete oder vorgestellte Handlungen zurück, die ikonische Darstellungsform nutzt zur Darstellung mathematischer Inhalte bildliche Formen. Die symbolischen Repräsentationen gebrauchen hingegen Zahlen, Zeichen und Fachsprachen zur Inhaltsvermittlung. Um transferfähig und nachhaltiges Lernen angemessen zu fördern sollten im Unterricht daher alle drei Formen genutzt werden.

Bearbeitete Aufgabe einreichen Hinweis: Änderungen sind nach der Abgabe nicht mehr möglich!

Bewertung: sehr gut (1.0000)

😊

Abb. 3: Vollautomatische Bewertung einer Studentenantwort mithilfe des machine learning

Literatur:

- Brownlee, J. (2017). *Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems*.
- Deru, M. & Ndiaye, A. (2019). *Deep Learning mit TensorFlow, Keras und TensorFlow.js*. Bonn: Reihnwerk.
- Géron, A. (2017). *Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme*. Köln: O'Reilly.
- Krosnick, J. A. & Presser, S. (2010). Question and questionnaire design. In P. V. Marsden & J. D. Wright (Hrsg.), *Handbook of survey research* (Vol. 2, S. 263–314). Bingley, UK: Emerald Group Publishing Limited.