

Lerndokumentation

CKi ein CNN-Model in C++

Für weitere Informationen

Simeon Stix
Tel. 078 323 8800
simeon.stix@stud.kftg.ch

Simeon Stix

3i

18.02.2024

Betreuer

Sven Nüesch
Tel. 071 664 20 77
sven.nueesch@kftg.ch

Inhaltsverzeichnis

01	CKi ein CNN-Model in C++.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.1	Identifikation.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.2	Beschrieb des Ablaufs der Arbeit.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.3	Gemachte Erfahrungen	Fehler! Es wurde kein Textmarkenname vergeben.
01.3.1	... in Bezug auf die ausgeführte Arbeit.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.3.2	... in Bezug auf das eigene Verhalten.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.4	Lernergebnisse	Fehler! Es wurde kein Textmarkenname vergeben.
01.4.1	Fach- und Methodenkompetenz.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.4.2	Selbst- und Sozialkompetenz.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.5	Schlussfolgerungen	Fehler! Es wurde kein Textmarkenname vergeben.
01.5.1	Ziele, die ich erreichen will	Fehler! Es wurde kein Textmarkenname vergeben.
01.5.2	Termin der Zielüberprüfung	Fehler! Es wurde kein Textmarkenname vergeben.
01.6	Bemerkungen.....	Fehler! Es wurde kein Textmarkenname vergeben.
01.7	Weiterführende Aktionen	Fehler! Es wurde kein Textmarkenname vergeben.

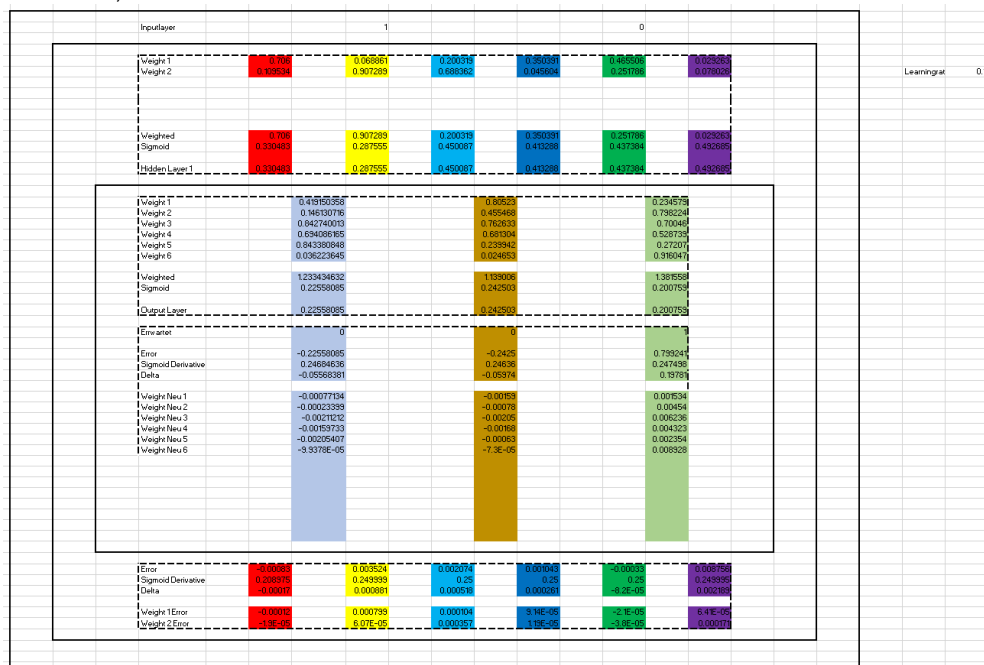
01 CKi ein CNN-Model in C++

01.1 Identifikation

Auftragsnummer	NaN
Auftragsbezeichnung	CKi ein CNN-Model in C++

01.2 Beschrieb des Ablaufs der Arbeit

Für mein Projekt, das sich auf die Erforschung und Implementierung von Convolutional Neural Networks (CNN) und allgemeinen neuronalen Netzwerken konzentrierte, habe ich mich eigenständig auf eine umfassende Recherche eingelassen. Die Frist war die einzige Vorgabe, was mir die Freiheit liess, meine Interessen vollständig zu verfolgen. Grundlagenwissen zu CNNs und neuronalen Netzwerken sammelte ich durch sorgfältige Recherche auf renommierten Plattformen wie Medium, IBM, Geekflare, der Stanford University, und Database Camp. Bei der Lösung spezifischer Softwareprobleme erwiesen sich Stack Overflow, GitHub Copilot und ChatGPT, sowie die Dokumentationen der verwendeten Abhängigkeiten und Cppreference als unverzichtbare Ressourcen. Der Datensatz, der für das Training des Netzwerks verwendet wurde, stammt von Yann LeCun's MNIST-Datenbank, einem Standardbenchmark in der Branche.



Die Visualisierung des Projekts erfolgte durch die Erstellung von Ablaufdiagrammen, die sowohl den Prozess als auch die Struktur der Architektur eines neuronalen Netzwerks darstellen, sowie einem Klassendiagramm, das die objektorientierte Struktur der Implementierung verdeutlicht. Da keine Datenbank beteiligt war, wurde kein Entity-Relationship-Modell (ERM) benötigt.

Die Entscheidung, das Projekt in C++ zu entwickeln, war nicht zufällig, sondern eine bewusste Wahl, die von mehreren Schlüsselfaktoren beeinflusst wurde. Zunächst ist C++ bekannt für seine aussergewöhnliche Leistungsfähigkeit und Effizienz^{Zeigler, 1995; Hudak, 1994}, was es zur idealen Sprache für rechenintensive Anwendungen wie neuronale Netzwerke macht. Meine persönliche Vorliebe für C++ spielte ebenfalls eine Rolle. Überdies eröffnete die Wahl von C++ die Möglichkeit, in Zukunft die Verarbeitungsgeschwindigkeit durch die Einbindung von GPU-Berechnungen zu steigern. Dies wäre ein entscheidender Vorteil für die Skalierung und Effizienz des Projekts. Entwickelt wurde das Ganze unter Windows 11 und Manjaro Linux, einem Arsch-basierten Betriebssystem, mit CLion, einer IDE von JetBrains. Diese Entwicklungsumgebung wurde speziell wegen ihrer Benutzerfreundlichkeit und Cross-Plattform-Unterstützung ausgewählt, was einen nahtlosen Entwicklungsprozess auf verschiedenen Systemen ermöglichte.

Die Durchführung der Softwaretests erfolgte mittels des Google Test Frameworks, wodurch eine gründliche Prüfung jeder Klasse gewährleistet wurde. Dieser Ansatz trug entscheidend zur Stabilität und Funktionsfähigkeit des entwickelten Systems bei. Obwohl es keinen direkten Kunden für das Projekt gab, entschied ich mich dafür, den Quellcode auf Plattformen wie GitHub und GitLab zu teilen. Dieser Schritt diente nur der Demonstration meiner Arbeit und nicht dem Anstossen von Diskussionen, da dieses Projekt schon vielfach implementiert wurde. Es ist schliesslich das Einstiegsprojekt für Machine Learning.

01.3 Gemachte Erfahrungen ...

01.3.1 ... in Bezug auf die ausgeführte Arbeit

Im Rahmen des Projekts CKI zur Digitalisierung handschriftlicher Zahlen wurde eine umfassende Analyse der Herausforderungen und Erfahrungen durchgeführt. Die Komplexität der Problemstellung offenbarte sich in der tiefgreifenden Auseinandersetzung mit Machine-Learning-Algorithmen und der fortgeschrittenen Programmierung in C++. Die Implementierung des Back-Propagation-Algorithmus stellte eine signifikante Hürde dar, deren Überwindung nicht nur technisches Verständnis, sondern auch eine methodische Herangehensweise erforderte. Technische Schwierigkeiten, wie die Integration externer C++-Bibliotheken und die Handhabung von Pointer-Exceptions, unterstrichen die Notwendigkeit einer präzisen Planung und Ausführung. Die Projektarbeit machte deutlich, dass die Erreichung einer hohen Präzision in der Zahlenidentifizierung eng mit der effizienten Nutzung der Ressourcen und der Minimierung von Rechenleistungsverbrauch verbunden ist. Trotz der Herausforderungen bot das Projekt wertvolle Einblicke in das Machine Learning.



01.3.2 ... in Bezug auf das eigene Verhalten

Diese Arbeit war zugleich motivierend und herausfordernd. Die Programmierung in C++ und die erfolgreiche Umsetzung des Projekts, insbesondere die Realisierung des neuronalen Netzwerks, haben mich sehr motiviert. Es war faszinierend, die direkten Auswirkungen meiner Arbeit zu sehen, vorwiegend bei der Erkennung handschriftlicher Ziffern. Jedoch hatten langwierige Probleme, wie die Implementierung der Back-Propagation, einen zermürbenden Effekt auf mich. Diese komplexen Herausforderungen erforderten viel Geduld und Ausdauer, und es gab Momente, in denen ich mich stark demotiviert fühlte.

Des Weiteren empfand ich das Dokumentieren als eine beschwerliche Angelegenheit. Obwohl ich weiss, wie wichtig eine gute Dokumentation für die Nachvollziehbarkeit und Wartbarkeit eines Projekts ist, fiel es mir schwer, mich dazu zu motivieren. Meine Stärken liegen eindeutig in der praktischen Umsetzung – in der Realisation. Der Entwicklungsprozess, vornehmlich die Programmierung, hat mir viel Freude bereitet und ich habe gerne neue Funktionen implementiert und das Netzwerk optimiert.

Allerdings habe ich gemerkt, dass ich dazu neige, schwierigere und anstrengender zu implementierende Teile, wie die Gestaltung der Benutzeroberfläche und die Implementierung der Back-Propagation, aufzuschieben. Diese Tendenz zum Aufschieben komplexerer Aufgaben hat mich in meiner Effizienz eingeschränkt und führte dazu, dass ich primär nur die unbedingt notwendigen Funktionen (Must-haves) implementierte. In Retrospektive sehe ich, dass ich zu viel prokrastiniert habe, was meine Zufriedenheit mit der eigenen Leistung mindert.

Insgesamt bin ich mit dem, was ich erreicht habe, zufrieden, insbesondere mit der Realisierung des neuronalen Netzwerks, jedoch hätte ich mit mehr Durchhaltevermögen und einem besseren (Zeit-)Management mehr erreichen können.

01.4 Lernergebnisse

01.4.1 Fach- und Methodenkompetenz

Während der Arbeit an meinem Projekt im Bereich der künstlichen Intelligenz konnte ich mein Fachwissen erheblich erweitern. Durch die intensive Auseinandersetzung mit C++ vertiefte ich meine Kenntnisse in dieser Programmiersprache und erlernte den effektiven Einsatz von Unit-Testing sowie die Nutzung von CMake für die Projektverwaltung. Ferner habe ich ein tiefgreifendes Verständnis für die Funktionsweise und Implementierung sowohl konventioneller neuronaler Netzwerke als auch Convolutional Neural Networks (CNN) entwickelt. Besonders hervorzuheben ist mein Zugewinn an Know-how im Bereich der Mathematik hinter neuronalen Netzwerken, welcher mir ermöglicht, die theoretischen Grundlagen und deren praktische Anwendung im maschinellen Lernen besser zu verstehen.

Die Grundlagen des maschinellen Lernens, einschliesslich der Entwicklung mit C++ unter Verwendung von CMake und Unit-Testing, sind nun Bereiche, in denen ich mich sicher fühle und zurechtfinde. Diese Kenntnisse ermöglichen es mir, komplexe Projekte im Bereich der künstlichen Intelligenz eigenständig zu planen, zu entwickeln und zu evaluieren. Die erworbenen Fähigkeiten stellen eine solide Basis dar, auf der ich weiter aufbauen und mich in spezialisierte Themenbereiche vertiefen kann.

01.4.2 Selbst- und Sozialkompetenz

Bezüglich der Selbstkompetenz erkannte ich, dass längere Projekte ein hohes Mass an Disziplin erfordern, um kontinuierlich Fortschritte zu machen. Mein Arbeitsfluss war stark von meiner Motivation abhängig, was zu einem unbeständigen Fortschritt führte. Ich habe gelernt, dass ich in Zukunft weniger prokrastinieren muss, um meine Ziele effizienter zu erreichen.

In Bezug auf die Sozialkompetenz war mein Kontakt zu anderen Personen, abgesehen von meinem Betreuer, sehr begrenzt. Die Interaktion beschränkte sich hauptsächlich auf die Übermittlung von Fortschrittsberichten und die Suche nach Unterstützung bei spezifischen Problemen, wie dem Algorithmus der Back-Propagation. Somit konnte ich wegen mangelnden Kontakts keine direkten Lernerfahrungen im Umgang mit Kollegen, Lehrern oder Fachpersonen machen.

01.5 Schlussfolgerungen

01.5.1 Ziele, die ich erreichen will

In Zukunft möchte ich mein Wissen und meine Fähigkeiten im Bereich des maschinellen Lernens, speziell im Deep Learning, weiter vertiefen. Mein Fokus liegt dabei auf dem Erlernen von Unsupervised Learning (unter anderem generative KI) und Reinforcement Learning. Diese Bereiche sind hochinteressant, vor allem vor dem Aspekt der autonomen Prozessautomatisierung. Somit bieten diese Bereiche ein enormes Potenzial für innovative Anwendungen, und ich bin motiviert, meine Kompetenzen hier zu erweitern.

Um meine Lern- und Arbeitsprozesse zu verbessern, strebe ich an, zukünftige Projekte besser zu strukturieren und längere Pausen effektiv zu nutzen, um Phasen der Demotivation zu minimieren. Weniger Prokrastination und eine konstantere Arbeitsweise sollen dabei helfen, meine Ziele effizienter zu erreichen.

01.5.2 Termin der Zielüberprüfung

Es ist wichtig zu betonen, dass diese Ziele langfristig und ohne konkrete Terminierung sind. Ich verstehe, dass die Entwicklung von Fachwissen und Fähigkeiten in Bereichen wie Unsupervised Learning, Reinforcement Learning und generativer KI Zeit benötigt. Diese Ziele dienen als Leitlinie für meine kontinuierliche Weiterbildung und Entwicklung in der Zukunft, ohne mich durch starre Fristen einzuschränken.

01.6 Bemerkungen

In meinen Augen ist das Projekt CKI zwar gelungen, aber bisher nicht ausreichend. Es gibt noch so viele Bereiche des maschinellen Lernens, die ich mir zumindest kurz anschauen möchte. Leider hat dafür die Zeit nicht ausgereicht.

01.7 Weiterführende Aktionen

Das Projekt CKI ist noch ausbaufähig. Somit werde ich, wie beschrieben, in „Ziele, die ich erreichen will“, mich mit generativer künstlicher Intelligenz und Reinforcement Learning beschäftigen.

Dafür wird das Projekt CKI in eine neue Phase übergeleitet. Es werden weitere Netzwerke realisiert und vielleicht die bestehenden Klassen in eine komplette Bibliothek umgebaut.

Als Erstes würde ich mich gerne mit dem Thema generative künstliche Intelligenz beschäftigen. Dafür werde ich ein weiteres Netzwerk auf das bestehende Netzwerk etablieren, sodass ich mit dem ersten Netzwerk das Bild generiere und mit dem Zweiten dieses danach bewerten kann. Ein solches Netzwerk könnte man danach auch noch optimieren, zum Beispiel durch die Generierung von komprimierten Bildern (so müssten weniger Bytes generiert werden).

Eine solche Implementierung ist leider wegen der begrenzten Zeit des Projektes CKI nicht problemlos zu realisieren und muss so ausserhalb des Projektes (evtl. auf Basis des Projektes CKI) realisiert werden.

Gelesen und besprochen	
Datum:	Datum:
Der/die Lernende	Der/die Betreuer/-in