

1. Gruppentreffen

Datum:

28.02.2024

Zusammenfassung des Treffens:

- erste Besprechung via Discord im Sprachchat
- kurzes Kennenlernen mit Vorstellung:
 - Namen,
 - Herkunft
 - was hat man vorher gemacht
 - wie sind die Gruppenmitglieder bisher mit den KI-Aufgaben zurecht gekommen
 - welche Vorerfahrungen gab es auf diesem Gebiet
- Absprache, wer die Position des Gruppensprechers übernimmt, nach kurzer Überlegung übernimmt Kai diese Aufgabe
- nach Vorschlag von Nikita informieren sich alle bis zum nächsten Treffen eingehend über das Thema:
 - welche bisherigen Bemühungen gab es den Datensatz mithilfe von CNN zu analysieren
 - sammeln von Papern, Foreneinträgen, Artikeln etc.
 - wie stellt sich jeder einzelne die Gestaltung des Portfolios vor
- Einigung auf einen regelmäßigen wöchentlichen Besprechungstermin, um Erkenntnisse zusammenzutragen
- nächstes Treffen jedoch erst in zwei Wochen, am 13.03.

2. Gruppentreffen

Datum:

13.03.2024

Zusammenfassung des Treffens:

- Nikita hat erstes Modell erstellt
 - Modell basiert auf Erfahrungsberichten der MNIST-Competition auf kaggle
- daran orientiert haben wir uns dieses Ziel gesetzt:
 - Das CNN soll eine Genauigkeit von über 99,5 % auf den Testdatensatz erreichen, bei gleichzeitiger Vermeidung von Overfitting und ohne erheblichen Rechenaufwand
- Besprechung, wie weiter mit dem Modell verfahren wird:
 - nacheinander soll jeder verschiedene Ideen zur Verbesserung des Modells ausprobieren
 - dabei soll jedes Gruppenmitglied für sich entscheiden, in welche Richtung die Verbesserung gehen soll
 - hohe accuracy, geringe Fehlerquote
 - möglichst einfaches Modell
 - wenige Epochen
- Die nächsten Tage versucht Martin mögliche Ansätze.
- Kenan widmet sich zunächst der explorativen Datenanalyse.

- Wenn ein Gruppenmitglied schon Zeit und Lust hat, kann es sich derjenige auch schon mit der Visualisierung befassen.
 - genauere Vorstellungen dazu wurden noch nicht besprochen
- Nächstes Treffen wurde für Sonntag 17.3. 21:00 Uhr festgelegt.

3. Gruppentreffen

Datum:

17.03.2024

Zusammenfassung des Treffens:

- Martin hat vieles getestet
 - verschiedenste Parameter -> eine Kombination brachte 0,997 accuracy, leider sind diese Parameter nicht mehr identifizierbar
 - zusätzliche Layer -> Knicke in der Kurve werden weniger, schönere Lernkurve
 - Batch Normalisierung -> keine signifikante Auswirkung
 - verschiedene Aktivierungsfunktionen: sigmoid (lief gar nicht gut), gelu, tanh
 - unterschiedliche Anwendung der Aktivierungsfunktionen:
 - Aktivierungsfunktion immer die Gleiche bis zum Dropout -> konsistente Ergebnisse
 - gestapelte Aktivierungsfunktionen für jedes Mal zu schlechteren Ergebnissen
 - min Lernrate um Faktor 10 herabgesetzt -> weniger Fehleranfällig
 - Test-size:
 - größer als 0,3 sehr viel schlechter
 - kleiner kaum Auswirkung
 - Batch-size 64 -> Meldung Daten nicht ausreichend
 - early stop disabled -> man bewegt sich eher von guten Ergebnissen weg
- kommt mit dem aktuellen Modell stetig auf 0,996 accuracy
- Feststellung: bei seinem Modell steigt die accuracy der Validierungsdaten schneller an, als die der Testdaten
- Kai hat etwas explorative Datenanalyse betrieben
- bis 20.03. setzt Kai sich an das Modell und versucht noch bessere Ergebnisse durch Feature Engineering zu erzeugen

4. Gruppentreffen

Datum:

20.03.2024

Zusammenfassung des Treffens:

- Kai hat ein zusätzliches Modell entworfen, welches zusätzliche Merkmale (Feature Engineering) mit einbezieht:
 - Gesamtintensität der Bilder
 - Symmetrie der Ziffern
 - Anzahl geschlossener Schleifen in einer Ziffer

- Das Modell kombiniert am Ende die Ergebnisse der beiden Modelle und erstellt ein Gesamtergebnis
 - liefert immer Ergebnisse im Bereich um 0,9870 accuracy -> schlechter als mit Martins Modell ohne additional features
- nun weitere Fragen, wie zB: Wie viele falsche Schleifen wurden gezählt? -> möglicherweise einige plotten um zu sehen, ob die vorhergesagte Ziffer auch tatsächlich die richtige ist
- Nikita hat weitere Verbesserungen am ursprünglichen Modell vorgenommen
 - Batchsize auf 64
 - Testsize auf 0,1
 - nach jedem convolutional layer Batch Normalisierung
 - Modell langsamer aber konstant über 0,9955 accuracy
- Erfahrungsaustausch inwiefern die verwendete Recheneinheit Einfluss hat:
 - Nikita und Martin nutzen ältere Nvidia Grafikkarten mit zufriedenstellenden Ergebnissen
 - Kai verwendet Nvidia Grafikkarte und erreicht damit zumindest bei Nikitas letztem Modell erheblich bessere Rechengeschwindigkeiten
 - Kenan nutzt eine AMD Grafikkarte und hat von Beginn an Probleme mit Abstürzen bei der Berechnung
- Besprechung weiteres Vorgehen:
 - Modell ist soweit zufriedenstellend -> wer möchte kann aber noch immer daran herumprobieren
 - Portfolio muss fertig gestaltet werden
 - Kenan Betrachtung des Datensatzes mit dessen Beschreibung, wie sich dieser zusammensetzt, was ihn auszeichnet, wo er herkommt
 - Nikita Dokumentation des Ergebnisses
 - Martin schaut welche Möglichkeiten der Visualisierung noch bestehen
- nächstes Treffen Mittwoch, 27.03.

5. Gruppentreffen

Datum:

27.03.2024

Zusammenfassung des Treffens:

- Vorstellung der ersten Version des Endergebnisses durch Nikita
 - Anmerkung: es fehlt Hinweis auf die Unmöglichkeit der 100% korrekten Klassifizierung, da im Datensatz auch Fehler enthalten sind
- nach letzter Aktualisierung des Modells, läuft dieses nur noch auf Nikitas GPU problemlos
 - jedoch nicht mehr weiter relevant, da das Modell soweit ausgereift ist und keiner großen Anpassungen mehr bedarf
- nun stellt sich die Frage: Wie könnte man mit dem Modell weiter verfahren?
 - besonders Interessant wäre es zu sehen, wie es sich mit anderen Datensätzen verhält -> mehr europäische Ziffernschreibweise?
 - es könnte in der Praxis getestet werden -> werden zB live geschriebene Ziffern korrekt erkannt
- Kenan hat eine Beschreibung vom Datensatz erstellt

- zudem wird versucht das Modell auf dem Großrechner des DKRZ (Deutsches Klima Rechenzentrum) laufen zu lassen -> leider aktuell noch Versionsprobleme
- bis zum nächsten Treffen am 10.04.
 - Nikita schreibt soweit Ergebnis fertig
 - jeder schreibt individuelles Logbuch (gegebenenfalls untereinander teilen, damit nichts vergessen geht)