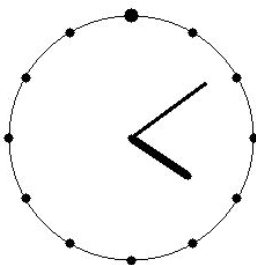


CAS Applied Data Science Modul 5, Peer Consulting

Marco Bassi, Juni 2020 (Version 1.0)

Projekthinhalt

Der Inhalt des Peer Consultings ist das Projekt «Uhren Lesen» von Gerhard Binder; siehe <https://github.com/gerbin56/uhr.git>. G.B. verwendet 3000 Bilder (s/w) des Ziffernblatts einer einfachen Analoguhr mit verschiedenen Zeigerständen, um seinem Neuronalen Netzwerk das Uhrenlesen beizubringen. Bsp:



Die Bilder variieren nach Grösse des Ziffernblatts, und nach Position auf dem Bild.

Die Labels zu den Bildern enthalten pro Zeile Stunde und Minute, die das entsprechende Uhrbild anzeigt. Im obigen Beispiel «4,9». Insgesamt finden sich bei den Labels 713 verschiedene Uhrzeiten. Eine Analoguhr kann – bei einer Auflösung von 1 Minute – 720 verschiedene Zeiten anzeigen.

Das Neuronale Netzwerk

Das Neuronale Netzwerk (NN) ist mit dem Tensorflow Keras-API implementiert. Es umfasst 3 Ebenen mit je einem Convolutional Layer, einem Maxpooling Layer und einem Batch-Normalisierungslayer. Der erste Convolutional Layer hat 50 Filter, der zweite 100 und der dritte 150 Layer.

Nach diesen drei Schichten folgt nochmals ein Convolutional Layer mit 200 Schichten, danach ein Dropout-Layer (Rate 0.4) und ein Flatten-Layer. Die Activation bei allen vier Convolutional Layern ist ReLu.

Der Output für Stunde und Minute wird getrennt berechnet. Für die Stunde gibt es 2 Fully Connected Layer mit Activation ReLu, und ein Fully Connected Layer mit Activation Softmax, Output-Dimension 12. Und für die Minute hat es analoge Layer, wobei die letzte keine Activation und Output-Dimension 1 hat.

Was mit gut gefällt

Definition der Aufgabe

Die Aufgabenstellung ist klar umrissen, und die Datenbasis ist definiert.

Wahl der Methode und des NN-Modells

Die Wahl eines NN für die Lösung der Aufgabe ist naheliegend. Der Aufbau des NN hält sich weitgehend an die «Best Practices» gemäss Keras-Dokumentation für die

Bildklassifikation. Für jemanden, der mit dem API nicht vertraut ist, ist das sehr hilfreich für das Verständnis.

Die Implementierung des NN ist logisch nachvollziehbar, gut strukturiert und zweckmässig. Auch beim Setzen der Hidden Parameters orientiert sich das NN an den zur Verfügung stehenden Beispielen.

Manuelle Datenaufbereitung

Das Keras-API bietet Funktionen zum Einlesen und Aufbereiten (Standardisierung) der Daten, und für die Definition der Trainings- und Test-Datensets. Der Nachteil dieser Funktionen ist, dass sie weitgehend «Black Boxes» sind, d.h. wenn man wissen will, wie diese Funktionen zu ihrem Ergebnis kommen, muss man sich in die Dokumentation einlesen und auch selbst experimentieren.

G.B. implementiert die Datenaufbereitung und die Erstellung der Datensets selbst. Das macht die Ergebnisse direkt nachvollziehbar und überprüfbar.

Implementierung im Jupyter-Notebook

Die Implementierung im Jupyter-Notebook ist gut strukturiert und gut nachvollziehbar. Die Inlinedokumentation ist ausreichend und sehr hilfreich für das Verständnis. Dasselbe gilt für die Ausgabe von Zwischenergebnissen, Datenstrukturen und Plots.

Verbesserungsmöglichkeiten und Probleme

Readme

Im Projekt fehlt ein Readme, welches die Aufgabenstellung und den Lösungsansatz dokumentiert. Und auch im Jupyter-Notebook selbst wäre eine Einleitung sinnvoll. Die Einarbeitung in die Aufgabenstellung muss deshalb mittels mündlicher Information und studieren des Jupyter-Notebooks erfolgen.

Dokumentation von Designentscheidungen

Eine Einleitung im Notebook wäre auch der richtige Ort gewesen, um einige Designentscheidungen zu dokumentieren. Eine offene Frage meinerseits zum Design ist z.B.: Warum wird für die Stunde das Bild klassifiziert, aber für die Minute eine Regression berechnet? Und wieso wird dieser zweigleisige Ansatz gewählt, anstatt entweder eine Klassifikation oder eine Regression für die Minuten seit Mitternacht (Modulo 720)?

Overfitting

Der Vergleich der Ergebnisse von Test-Datenset gegenüber Trainings-Datenset (Loss, Accuracy bzw. Mean Absolute Error) gibt deutliche Hinweise auf Overfitting des NN.

Ein Grund dafür dürfte die weitgehend fehlende «Augmentation» der Datenbasis sein. Die üblichen bildbearbeitenden Verfahren wie Rotation und Spiegelung kommen für die Aufgabenstellung nicht in Frage, da die Orientierung des Ziffernblattes wichtig ist. Es gibt jedoch weitere Möglichkeiten, wie z.B.

- Perspektivische Verzerrung
- Überlagerung mit einem Muster
- Weichzeichnen

Für das skriptbasierte bildbearbeiten stellt z.B. die Software ImageMagick zahlreiche Schnittstellen zu Programmiersprachen zur Verfügung

Weitere Gründe für das Overfitting können in den Hidden Parameters liegen. Das Thema wird leider nicht weiter behandelt. . Mit etwas Programmieraufwand könnten die Bilder auch dynamisch erzeugt werden, auch mit variierenden Zifferblättern und Zeigern. Damit könnte zusätzlich zum Trainings- und Test-Datenset auch ein Dev-Datenset zum Optimieren der Hidden Parameters zur Verfügung gestellt werden.

Trainingsdauer

Für dieses doch sehr übersichtliche Projekt dauert das Training des NN sehr lange. Das kann u.U. daran liegen, dass das NN für die Aufgabenstellung überdimensioniert ist. Die Bilder haben viel weniger Features als z.B. die Katzenbilder, mit denen wir die Bildklassifikation trainiert haben.

Eine weitere Möglichkeit, das Training zu beschleunigen, wäre ein Inception Layer, das bereits auf die Erkennung von Bildstrukturen trainiert ist. Nicht berücksichtigt ist auch die Verwendung von GPU-Cores, wofür das Keras API ebenfalls Funktionen zur Verfügung stellt.

Verwendung von `numpy.random.choice`

Bei der Verwendung von `numpy.random.choice` in Zelle 5 gibt es eine Reihe von Problemen:

1. Diese Funktion soll nicht mehr verwendet werden. Auszug aus der NumPy Version 1.18-Dokumentation:
New code should use the `choice` method of a `default_rng()` instance instead; see [random-quick-start](#). `randint`, `shuffle`, `permutation` should be used in new code.
2. Die Funktion wird nicht explizit mit einem Seed/Salt initialisiert. Damit können Resultate nicht reproduziert werden.
3. Beim Aufrufen der Funktion sollte `replace=False` gesetzt werden, damit ein Element höchstens 1 ausgewählt werden kann. Der Default ist `True`, so dass die Wahrscheinlichkeit gross ist, dass ein oder mehrere Elemente mehrfach ausgewählt werden.