



Anfertigen eines CAT bzw. PIKE

Autor: Andreas Schau - AS

Kann ... // Algorithmus x // exact ... // (Kunden)-Problem ... berechnen / lösen?

Ja, ein neuronales Netz kann die Klassifikation mit einer Genauigkeit von bis zu 90% durchführen.

Data Science Kernaussage:

(P) roblem):

[Welcher Frage hat für die Lösung des Kunden / Auftraggeber die größte Bedeutung?]:

Kann mit Hilfe eines neuronalen Netzes zuverlässig eine Ziffernerkennung von Google Streetview Hausnummern durchgeführt werden? Dabei geht es vorerst nur um die Erkennung der Ziffern von 0 bis 9 welche jedoch nur vorrangig in der Mitte der Bilder stehen und von weiteren Ziffern der Hausnummern umgeben sein können. Erschwerend kommen außerdem noch Verzerrungs-Effekte durch seitliche und gedrehte Aufnahmen, sowie einen Fish-Eye Effekt auf diesen hinzu.

(I) ntervention:

(Bibliotheken- und Algorithmen-Auswahl, ... z.B. pandas für Finanzdaten ...)

[Welche Berechnung erwäge ich vornehmlich?]:

Für das Neuronale Netz wurde die Pytorch Bibliothek in der Version 2.2.0+rocm5.7 verwendet. (Also Version 2.2 mit Support für AMD Grafikkarten.) Für die Datenvorbereitung und Analyse wurden weiterhin „os“, „collections“, „matplotlib“, „numpy“, „torchvision“, „scipy.io“ und „tqdm“ verwendet. Dabei wurde hauptsächlich ein Convolutional Network und ein paar lineare Hidden Layer verwendet. Das Laden der Daten geschah via der „scipy.io“ Bibliothek, welche einen Matlab-Loader enthält, mit dem die in dem Matlab Format *.mat vorliegenden Daten in das Script geladen wurden. Numpy wurde für die Datenaufbereitung und Formatsanpassung verwendet und „matplotlib“ um die jeweiligen Fortschritte zu verbildlichen.

(K) ontrollintervention

(falls erforderlich: Bibliotheken- und Algorithmen-Auswahl ... z.B. scikit-learn für Finanzdaten ...)

[Was ist die andere Möglichkeit?]:

Man hätte auch „tensorflow“ oder eine eigene Implementierung eines Neuronalen Netzes zurückgreifen können. Da es sich in diesem Projekt jedoch um Bilddaten handelt war es ratsam auf eine möglichst leistungsfähige Implementierung zu setzen, die auf die Unterstützung der Grafikkarte zugreifen kann.

Man hätte bei der Vorverarbeitung sicher auch auf reines Python und bei der Diagramm-Erstellung Seaborn oder andere Bibliotheken verwenden können. Das wäre jedoch nicht unbedingt effektiver.

(E) rgebnismaß (Zielgröße(n)) – Die Evidence

[Was möchte ich / der Kunde erreichen? Z.B. Prädiktor oder Klassifikator erstellen ...]:

Eine Klassifizierung mit hoher Genauigkeit (über 80%) soll erreicht werden.

Anmerkungen

Literaturhinweise

Die Suche nach der besten Evidenz

1. Problem

Eine Unterscheidung von undeutlich sichtbaren, in Farbe aufgenommenen Ziffern von Null bis Neun (0 - 9) soll mittels eines neuronalen Netzes durchgeführt werden.

2. Definition einer wichtigen suchbaren Frage

Kann die Klassifizierung mittels eines neuronalen Netzes mit hoher Genauigkeit von über 80 Prozent durchgeführt werden?

3. Auswahl der wahrscheinlichsten Quelle für diese Evidenz

Während des Trainings wird am Ende jeder Epoche mit einem Testdatensatz berechnet, wie genau das Modell schon bei Daten, auf die nicht trainiert wurde reagiert. Am Ende des Trainings kann man auch noch einmal separat den Trainingserfolg validieren. Sowie ihn mit ein paar Stichproben bildlich darstellen. Innerhalb jeder Epoche werden außerdem auch der Verlust und die Genauigkeit des aktuellen Laufes gespeichert, welche man sich am Ende in einer Loss- und Accuracy-Kurve visualisieren kann.

4. Erstellung einer Suchstrategie

Um eine möglichst hohe Genauigkeit bei der Validierung des Modells zu erreichen werden einige unterschiedliche Modelausprägungen untersucht. Dabei wurde hauptsächlich die Dropout-Strategie verwendet um Overfitting zu vermeiden und es wurde die Tiefe des Netzes variiert um zu evaluieren ob so eine bessere Feature Verallgemeinerung erreicht werden kann.

5.0 Zusammenstellung der Evidenzausbeute

5.1 Falle Ausbeute schlecht

**Auswahl der zweit-wahrscheinlichsten
Quelle für diese Evidenz
Erstellung einer Suchstrategie
Zusammenfassung der Evidenz
Anwendung der Evidenz**

Das Modell besteht aus einer Input-Schicht mit 3x32 Parametern die in ein Convolutional Layer geht. Dessen 32 Ausgänge wieder als 32 Inputs in eine weitere Convolution-Schicht gehen, die ebenfalls 32 Ausgänge hat. Diese gehen wiederum in mehrere Lineare-Schichten mit 8192 Eingängen auf 512 Ausgänge, 512 Eingängen auf 128 Ausgänge und 128 Eingängen auf 32 Ausgänge. Und diese gehen in eine Logarithmische Softmax Schicht, mit 32 Eingängen und den 10 Klassifikationen als Ausgang. An den Ausgängen der Linearen Schichten gibt es jeweils einen Dropout zwischen 0,2 und 0,1.

6. Anwendung der Evidenz

Dieses Modell wurde über 200 Epochen mit einer Batch-Größe von 10 über 13861 von 18960 Samples bzw. 73% der Daten trainiert. Die anderen 27% wurden als Validierungs-Datensatz verwendet. Mithilfe dieses Modells und dem SGD Optimizer und der Cross-Entropy-Loss-Funktion konnte eine Genauigkeit von bis zu 90% in einem günstigen Lauf erreicht werden.

(In späteren Versuchen wurde noch einmal mit unterschiedlichen Epochen und Sample-Größen gearbeitet.)