Arithmetisches Kodieren

Vito Schopper - 7503386
Mario Navarro -

Pro-Seminar
Datenkompression WS 2022

Dozent: Dr.- Ing. The Anh Vuong
Graphische Daten Verarbeitung, Informatik Institut
Goethe Universität , Frankfurt am Main

Inhaltsverzeichnis

1 Thema

Hier könnte ein kurzer Satz zur Einleitung stehen.

2 Theoretische Grundlagen

Hier könnte ein kurzer Satz zur Einleitung stehen.

3 Verfahren-Beschreibung

Hier könnte ein kurzer Satz zur Einleitung stehen.

4 Anwendungsgebiet

Hier könnte ein kurzer Satz zur Einleitung stehen.

5 Qualitätsbewertung über das Verfahren

Hier könnte ein kurzer Satz zur Einleitung stehen.

6 Präsentation / Demonstration Kit

Wir haben bei der Visualisierung/der Umsetzung des Demonstrations Kit zwischen 2 Fällen unterschieden. Da es bei den Berechnungen während der Kodierung/Dekodierung zu Schwierigkeiten mit den Intervallgrenzen kommt (siehe Verfahren Beschreibung), haben wir ein **naives** und ein **genaues** Kit erstellt.

Das naive Kit erstellt eine gut verständliche, visuelle Animation. Jedoch funktioniert dieses Verfahren nur für relativ kleine Eingaben wie z.B "Hello" oder "Laterne". Bei größeren Eingaben benötigt die Erstellung der Animation erheblich länger. Da dieses Verfahren auch nicht das TODO umsetzt, kommt es hier bei größeren Eingaben zu den in Kapitel 3 erwähnten Rundungsfehlern. Auf kurzen Eingaben ist jedoch eine korrekte Kodierung/Dekodierung, sowie eine Erstellung einer visuell ansprechenden Animation kein Problem.

Hierfür haben wir auf das python-Modul **manim** zurückgegriffen. Dabei handelt es sich um ein Opensource-Projekt des Youtubers "3Blue1Brown", welcher es vor mehreren Jahren geschrieben hat, um visuell ansprechende Animationen ifür den Bereich der Mathematik zu erstellen.
Mittlerweile wird dieses Model jedoch durch die Community erweitert und bietet viele Möglichkeiten Themen aus dem MINT-Bereich visuell darzustellen.

6.1 naives Verfahren

6.1.1 Installation

Es wird das python-modul **manim**, sowie viele kleine weitere module wie z.B. **ffmpeg** benötigt, welche jedoch automatisch mit **manim** heruntergeladen werden.

Der Installationsvorgang wird unter https://docs.manim.community/en/stable/installation.html genauer beschrieben.

6.1.2 Kodierung

Für die Kodierung wird die Datei **encoding.py** mithilfe des Befehls

```
manim -pql -v critical encoding.py
```

im terminal ausgeführt.
Anschließend wird als Eingabe das zu kodierende Wort erfragt. Hier wie oben erwähnt, kein zu langes Wort eingeben. Daraufhin kann es je nach länge des Wortes 1-2 min dauern, bis die Animation erstellt wurde, welche sich direkt in einem neuen Fenster öffnet und angeschaut werden kann. Alternativ wurde die Animation unter

```
.\SeminarDatenkompression-2022\pythno\media\videos%encodin\480p15\Encoding.mp4
```

gespeichert.

6.1.3 Dekodierung

Für die Dekodierung wird die Datei **decoding.py** mithilfe des Befehls

```
manim -pql -v critical decoding.py
```

im terminal ausgeführt. Anschließend wird als Eingabe zuerst die zu dekodierende Binärzahl erfragt. Daraufhin wird eine Häufigkeitsverteilung der vorkommenden Buchstaben erfragt. Diese muss im Format eines python-Dictionaries eingegeben werden. z.B: TODO: ist Reihenfolge egal?

```
{"H": 1, "e": 1, "l": 2, "o": 1}
{"A": 1, "e": 1, "f": 2}
```

Daraufhin kann es je nach länge des Wortes 1-2 min dauern, bis die Animation erstellt wurde, welche sich direkt in einem neuen Fenster öffnet und angeschaut werden kann. Alternativ wurde die Animation unter

```
.\Seminar-Datenkompression-2022\python%\media\videos\decodin\480p15\Decoding.mp4
```

gespeichert.

6.2 genaues Verfahren

6.3 Kodierung

Für die Kodierung wird eine Eingabe im Eingabefeld erwartet. Falls gewünscht kann man sich mit dem Button **Generiere Wahrscheinlichkeiten**, die Wahrscheinlichkeitsverteilung der Eingabe anzeigen lassen.

7 temp

wird am Ende entfernt. Dient nur zur Referenz für gewisse commands
Man könnte sich auf Abschnitt 7.1 beziehen. Oder auf Formeln (1) und (2) und auf Abbildung 1.

7.1 Ein Unterabschnitt

Dies ist der Text von einem Unterabschnitt.

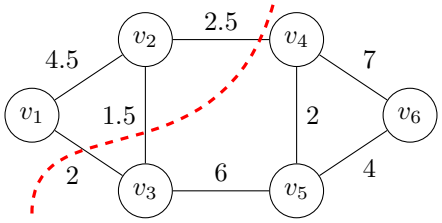

Abbildung 1: Ein Beispielgraph mit eingezeichneter gestrichelter Linie.

Hier könnte auch noch etwas Text stehen.

7.2 Ein weiterer Unterabschnitt

Hier könnten Formeln stehen, z.B.

$$n^2 + 2n + 40 = \mathcal{O}(n^2)$$

Manchmal möchte man auch Formeln nummerieren, zum Beispiel

$$V = \{v_1, \ldots, v_n\} \tag{1}$$

$$E = \{\{v_i, v_j\} \mid j \in \{1, \ldots, n-1\},\ i = j + 1\} \tag{2}$$

Bemerkung. "Pythagoras (geb. um 570 v.Chr.) gilt traditionell als der Entdecker des als Satz des Pythagoras bekannten Lehrsatzes der Euklidischen Geometrie über das rechtwinklige Dreieck. Dieser Satz war schon Jahrhunderte vor Pythagoras den Babyloniern bekannt. Ob sie aber einen Beweis für den Satz kannten, ist unbekannt. Zhmud meint, Pythagoras habe einen Beweis gefunden, während Burkert im Sinne der Schamanismusthese argumentiert, dafür gebe es keinen Beleg und Pythagoras habe sich für mathematische Beweisführung gar nicht interessiert." [3]

7.3 Der nächste Abschnitt

Vielleicht möchte man hierfür eine neue Seite beginnen und Bezug auf [1] oder [2] nehmen.

Literatur

[1] Namen der Autoren, *Name des Artikels*, Name der Veröffentlichung, Datum, S. 42-46

[2] Namen der anderen Autoren, *Name des anderen Artikels*, Name der anderen Veröffentlichung, Datum, S. 10-20

[3] Wikipedia