Kodierung und Dekodierung im Zusammenhang mit Sprachmodellierung

Konversation mit OpenAI ChatGPT : März 2024

Es geht um **Kodierung** und **Dekodierung** im Zusammenhang mit Sprachmodellierung.

Was nun genau ist das Encoder-Only-Verfahren, das Decoder-Only-Verfahren und das Encoder-Decoder-Verfahren?

In der Sprachmodellierung und im maschinellen Lernen gibt es drei gängige Architekturen, die im Zusammenhang mit Kodierung (Encoding) und Dekodierung (Decoding) verwendet werden:

das Encoder-Only-Verfahren,
das Decoder-Only-Verfahren
das Encoder-Decoder-Verfahren.

Diese Architekturen werden häufig in neuronalen Netzen verwendet, insbesondere in Transformern, um Text zu verarbeiten und zu generieren.

Jede dieser Architekturen hat einen spezifischen Anwendungsbereich und eine bestimmte Funktionsweise:

1. Encoder-Only-Verfahren

Beispiel: BERT (*Bidirectional Encoder Representations from Transformers*)

Anwendung: Verständnis von Text (z.B. Textklassifikation, Sentimentanalyse)

Funktionsweise: Im Encoder-Only-Verfahren wird nur der Encoder verwendet. Der Encoder ist dafür zuständig, eine Eingabesequenz (z.B. einen Text) zu verarbeiten und eine Repräsentation (ein sogenanntes "Embedding") zu erstellen, die Informationen über den Text enthält.

Dieses Verfahren arbeitet in der Regel **bidirektional**: Das Modell betrachtet den Gesamtkontext eines Textes zugleich, also sowohl die vorhergehenden als auch die nachfolgenden Wörter.

Es ist besonders nützlich für Aufgaben, bei denen das Verständnis eines Textes wichtig ist, z.B. bei der Extraktion von Bedeutungen, der Beantwortung von Fragen oder der Textklassifikation.

Beispiel:
Im Fall von BERT wird ein Satz eingegeben, und das Modell erzeugt für jedes Wort eine Einbettung (word-embedding), die sowohl die links als auch die rechts liegenden Wörter berücksichtigt.

2. Decoder-Only-Verfahren

Beispiel: GPT (*Generative Pretrained Transformer*)

Anwendung: Texterzeugung (z.B. Textvorhersage, Sprachgenerierung)
Funktionsweise:
Das Decoder-Only-Verfahren verwendet nur den Decoder, der auf der autoregressiven Modellierung basiert. Das bedeutet, dass der Decoder Wörter nacheinander generiert, wobei er nur auf die vorhergehenden Wörter schaut, um das nächste Wort vorherzusagen.

Dieses Verfahren ist für die **Textgenerierung** ideal, da es in der Lage ist, schrittweise den nächsten Teil eines Satzes zu erzeugen.

Der **unidirektionale** Ansatz sorgt dafür, dass das Modell nur auf die Wörter links vom aktuellen Wort schaut (im Gegensatz zum bidirektionalen Ansatz des Encoders).

Beispiel:
Bei GPT wird ein Anfangssatz eingegeben, und das Modell generiert auf Basis der bereits generierten Wörter fortlaufend neue Wörter, bis der gewünschte Text vollständig ist.

3. Encoder-Decoder-Verfahren

Beispiel: T5 (*Text-to-Text Transfer Transformer*), Seq2Seq mit Attention (z.B. für maschinelle Übersetzung)

Anwendung: Aufgaben, bei denen eine Sequenz (z.B. ein Satz) in eine andere Sequenz umgewandelt wird (z.B. maschinelle Übersetzung, Zusammenfassung)

Funktionsweise:
Hier arbeiten sowohl der **Encoder** als auch der **Decoder** zusammen. Der Encoder verarbeitet die Eingabesequenz und erstellt eine Repräsentation dieser Sequenz. Der Decoder verwendet diese Repräsentation, um eine neue Sequenz zu erzeugen, z.B. eine Übersetzung in eine andere Sprache.

Diese Architektur wird typischerweise in Aufgaben wie **Sequenz-zu-Sequenz-Transformationen** eingesetzt, bei denen es darum geht, aus einem Eingabetext eine andere Form von Text zu erzeugen.

Der Encoder extrahiert die Bedeutung aus der Eingabesequenz, und der Decoder gibt die entsprechende Ausgabe zurück.

Beispiel:
Bei der maschinellen Übersetzung wird ein Satz in einer Ausgangssprache durch den Encoder in eine Repräsentation umgewandelt, und der Decoder erzeugt auf Basis dieser Repräsentation den übersetzten Satz in der Zielsprache.

Zusammenfassung:

- **Encoder-Only:** Verarbeitung und Verständnis des Textes (z.B. BERT).

- **Decoder-Only:** Generierung von Text, autoregressiv (z.B. GPT).

- Encoder-Decoder: Umwandlung von Sequenzen (z.B. maschinelle Übersetzung, T5).

Jede dieser Architekturen hat spezifische Stärken, je nachdem, ob es darum geht, Text zu verstehen, zu erzeugen oder eine Transformation durchzuführen.