

Kenan Khauto
7592047
B.Sc Informatik
Studienfachkombination / Schwerpunkt
6
kenan.khauto@stud.uni-frankfurt.de

Seminararbeit Text Analytics

Image Understanding

Lernen visueller Konzepte beim Inferieren ohne finetuning

Kenan Khauto

Abgabedatum: <Datum>

Goethe-Universität Frankfurt am Main
Prof. Alexander Mehler

Erklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen Quellen oder Hilfsmittel als die in dieser Arbeit angegebenen verwendet habe.

Ort, Datum

Unterschrift

Inhaltsverzeichnis

1. Einleitung	4
2. Hauptteil	5
2.1. Theoretischer Hintergrund	5
2.1.1. Grundlagen des maschinellen Lernens	5
2.1.2. Deep Learning und neuronale Netzwerke	5
2.1.3. Kontrastives Lernen und CLIP	7
2.2. Analyse von CLIP	7
2.2.1. Architektur	7
2.2.2. Integration und gemeinsamer Einbettungsraum	8
2.2.3. Kontrastives Lernen im Detail	9
2.2.4. Zero-Shot Learning in CLIP	10
2.2.5. Vergleichsanalyse	11
2.3. Fallstudien / Anwendungen	12
2.4. Diskussion von Herausforderungen und Grenzen	12
2.5. Zukünftige Richtungen und mögliche Verbesserungen	12
Literatur	13
A. Anhang 1	14

1. Einleitung

Im Bereich der künstlichen Intelligenz und des maschinellen Lernens hat sich in den letzten Jahren eine bemerkenswerte Entwicklung vollzogen, insbesondere im Verständnis visueller Konzepte durch Modelle wie CLIP (Contrastive Language–Image Pretraining). Diese Modelle haben das traditionelle Paradigma, das umfangreiche Fine-Tuning und spezialisierte Datensätze erforderte, herausgefordert und bieten neue Wege, wie Maschinen Bilder und Texte in einem zusammenhängenden Rahmen verstehen können.

Die Kernfrage, die wir in diesem Seminar untersuchen, lautet: "Wie können KI-Modelle wie CLIP visuelle Konzepte effektiv durch Inferenz verstehen und interpretieren, ohne dass ein umfangreiches Fine-Tuning erforderlich ist?" Diese Frage berührt die Grundlagen der Art und Weise, wie maschinelle Lernmodelle trainiert und angewendet werden, insbesondere im Kontext der Integration von visuellen und textuellen Daten.

CLIP, ein Produkt von OpenAI, repräsentiert einen Durchbruch in der Art und Weise, wie Maschinen lernen, Bilder und Texte zu verbinden. Anstatt sich auf umfangreiche, spezialisierte Datensätze zu verlassen, nutzt CLIP ein breites Spektrum an Internetdaten und lernt, visuelle Konzepte direkt aus einer Vielzahl von Bildern und den dazugehörigen Beschreibungen zu extrahieren. Dieser Ansatz ermöglicht es dem Modell, eine breite Palette von visuellen Konzepten zu verstehen, ohne für jedes neue Konzept speziell angepasst zu werden.

In diesem Seminar werden wir die Mechanismen hinter CLIP und ähnlichen Modellen erforschen. Wir werden untersuchen, wie diese Modelle trainiert werden, ihre Fähigkeit, visuelle Daten zu interpretieren, und die Herausforderungen, denen sie gegenüberstehen, wie zum Beispiel die Behandlung von Verzerrungen und die Generalisierbarkeit ihrer Erkenntnisse. Darüber hinaus werden wir diskutieren, wie diese Technologien in verschiedenen Anwendungsfeldern eingesetzt werden könnten, von der automatisierten Bildbeschreibung bis hin zur Verbesserung der Mensch-Maschine-Interaktion.

2. Hauptteil

2.1. Theoretischer Hintergrund

Dieser Abschnitt stellt die theoretischen Grundlagen vor, die für das Verständnis der Funktionsweise von KI-Modellen wie CLIP (Contrastive Language–Image Pretraining) essentiell sind. Hier werden die Kernelemente des maschinellen Lernens, die Besonderheiten des Deep Learning und die Bedeutung des kontrastiven Lernens für die Verarbeitung von Bild- und Textdaten erörtert.

2.1.1. Grundlagen des maschinellen Lernens

Maschinelles Lernen ist ein Teilgebiet der künstlichen Intelligenz, das sich mit der Entwicklung von Algorithmen beschäftigt, die Computern das Lernen aus Daten ermöglichen. Die Hauptzielsetzung des maschinellen Lernens ist es, Muster in Daten zu erkennen und auf Basis dieser Muster Vorhersagen oder Entscheidungen zu treffen.

Überwachtes vs. unüberwachtes Lernen: Überwachtes Lernen bezieht sich auf Lernprozesse, bei denen das Modell anhand von Beispieldaten und bekannten Ausgabewerten trainiert wird. Unüberwachtes Lernen hingegen befasst sich mit dem Finden von Mustern oder Strukturen in Daten, ohne vorherige Kenntnis der Ausgabewerte.

Verstärkendes Lernen: Eine weitere wichtige Lernmethode ist das verstärkende Lernen, bei dem ein Modell durch Belohnungen lernt, bestimmte Aktionen in einer Umgebung auszuführen, um ein bestimmtes Ziel zu erreichen.

2.1.2. Deep Learning und neuronale Netzwerke

Deep Learning, eine Unterklasse des maschinellen Lernens, basiert auf künstlichen neuronalen Netzwerken mit vielen Schichten (sogenannten "tiefen" Netzwerken). Diese Netzwerke sind in der Lage, komplexe Muster in großen Datenmengen zu erkennen.

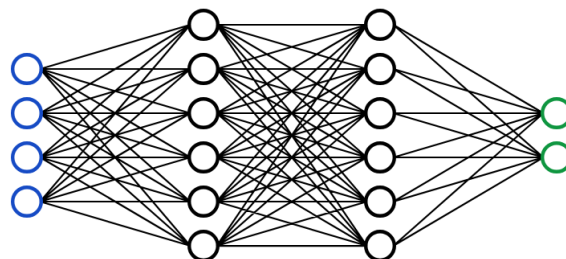


Abbildung 2.1.: A simple fully connected neural network, see <https://victorzhou.com/series/neural-networks-from-scratch/>

Convolutional Neural Networks (CNNs): Speziell für die Bildverarbeitung sind CNNs von entscheidender Bedeutung. Sie sind darauf ausgelegt, hierarchische Muster in Bildern zu erkennen, was sie ideal für Aufgaben wie Bildklassifikation und Objekterkennung macht.

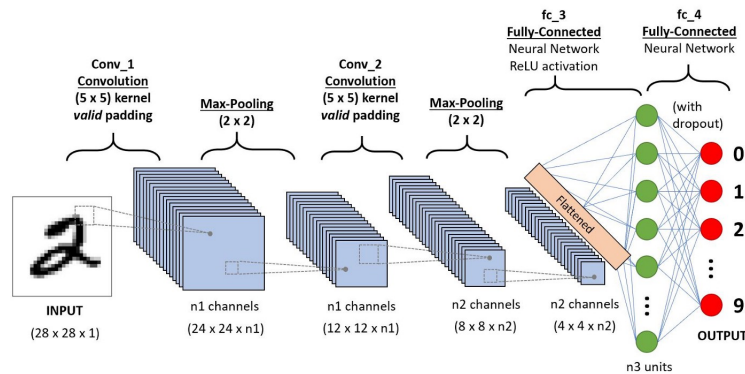


Abbildung 2.2.: Convolutional neural network (CNN), see <https://paperswithcode.com/methods/category/convolutional-neural-networks>

Transformer-Modelle: Ursprünglich in der Verarbeitung natürlicher Sprache eingesetzt, haben Transformer-Modelle aufgrund ihrer Fähigkeit, langfristige Abhängigkeiten in Daten zu erkennen, zunehmend Anwendung in anderen Bereichen, einschließlich der Bildverarbeitung, gefunden.

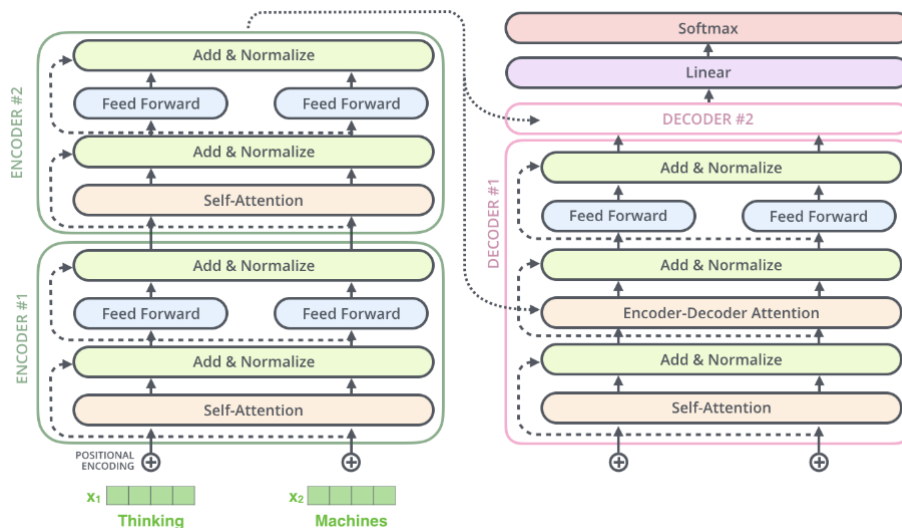


Abbildung 2.3.: Transformer with two encoders, 2 decoders and a fully connected layer for prediction, see <http://jalammar.github.io/illustrated-transformer/>

2.1.3. Kontrastives Lernen und CLIP

Kontrastives Lernen ist eine Technik, die darauf abzielt, ähnliche Datenpunkte näher zusammenzubringen und unähnliche weiter voneinander zu entfernen. CLIP verwendet kontrastives Lernen, um die Beziehungen zwischen Bildern und Text zu verstehen.

Funktionsweise von CLIP: CLIP wird mit einer Vielzahl von Bildern und den dazugehörigen Textbeschreibungen trainiert. Das Modell lernt, die Verbindungen zwischen Bildern und Texten zu verstehen, was es ihm ermöglicht, eine breite Palette von Bildinhalten effektiv zu interpretieren.

Vorteile gegenüber traditionellen Ansätzen: Im Gegensatz zu traditionellen Bilderkennungsmodellen, die oft umfangreiches Fine-Tuning für spezifische Aufgaben benötigen, kann CLIP vielfältige visuelle Konzepte anhand seiner Trainingsdaten erkennen und interpretieren, was eine größere Flexibilität und Anpassungsfähigkeit bedeutet.

2.2. Analyse von CLIP

CLIP ist ein neuronales Netzwerk, das anhand einer Vielzahl von Bildern und den zugehörigen Textbeschreibungen trainiert wurde. Dieses Training ermöglicht es ihm, sowohl visuelle als auch textuelle Eingaben zu verstehen und zu interpretieren. Im Gegensatz zu traditionellen Modellen, die eine aufgabenspezifische Feinabstimmung erfordern, ist CLIP für eine breite Palette visueller Aufgaben direkt einsetzbar.

2.2.1. Architektur

Das Modell besteht aus zwei Hauptkomponenten: einem Text-Encoder und einem Bild-Encoder. Der Text-Encoder verarbeitet textuelle Eingaben, während der Bild-Encoder sich um visuelle Eingaben kümmert. Beide Encoder wandeln ihre jeweiligen Eingaben in einen gemeinsamen Einbettungsraum um, was dem Modell ermöglicht, die beiden unterschiedlichen Datentypen direkt zu vergleichen und zu verknüpfen.

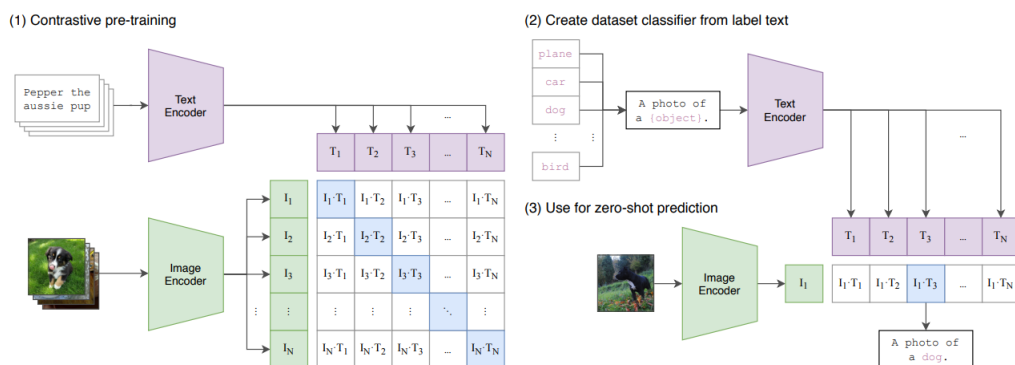


Abbildung 2.4.: CLIP, Bild von (Radford u. a. 2021)

Text-Encoder

Der Text-Encoder in CLIP basiert auf einer Transformer-Architektur. Jedes Wort im Text wird in eine Vektordarstellung umgewandelt. Der Transformer verwendet Selbst-Attention, um die Bedeutung jedes Wortes im Kontext des gesamten Textes zu verstehen. Ziel des Text-Encoders ist es, eine Darstellung des Textes zu erzeugen, die dessen semantischen Inhalt in Bezug auf mögliche Bildinhalte widerspiegelt.

Bild-Encoder

Der Bild-Encoder in CLIP ist nicht ausschließlich auf Convolutional Neural Networks (CNNs) beschränkt, sondern kann auch Vision Transformers (ViTs) umfassen. Die Wahl des spezifischen Netzwerktyps hängt von der gewünschten Leistung und den Eigenschaften der Aufgabe ab:

- **CNN-basierte Encoder:** Traditionelle CNN-Architekturen sind effektiv in der Erkennung lokaler Muster und Texturen in Bildern. Sie wandeln das Bild in eine Serie von Vektorrepräsentationen um, die die visuellen Inhalte des Bildes kodieren.
- **Vision Transformers:** ViTs nähern sich der Bildverarbeitung auf eine andere Weise an, indem sie das Bild in eine Sequenz von Patches zerlegen und diese Patches ähnlich wie Wörter in einem Satz behandeln. Dies ermöglicht es ViTs, sowohl lokale als auch globale Kontextinformationen aus dem Bild zu erfassen.

Der Hauptzweck des Bild-Encoders, unabhängig von der gewählten Architektur, besteht darin, eine Darstellung des Bildes zu erzeugen, die in demselben Vektorraum wie der Text-Encoder liegt, um eine direkte Vergleichbarkeit und Verknüpfung von Text und Bild zu ermöglichen.

Für Leser, die sich tiefergehend mit der Theorie und Anwendung von Transformern beschäftigen möchten, empfehle ich die grundlegenden Papers (Vaswani u. a. 2017) und (Dosovitskiy u. a. 2020). Diese zwei Werke bieten eine umfassende Einführung in das Konzept der Transformer und ViTs und legt den Grundstein für viele moderne Ansätze in der Verarbeitung natürlicher Sprache und Bilderkennung.

2.2.2. Integration und gemeinsamer Einbettungsraum

Die zentrale Innovation von CLIP besteht darin, dass beide Encoder - der Text- und der Bild-Encoder - darauf trainiert sind, ihre Ausgaben in einem gemeinsamen Einbettungsraum zu repräsentieren. Dies ermöglicht es dem Modell, die Beziehung zwischen Texten und Bildern effektiv zu verstehen und auf dieser Basis präzise Entscheidungen zu treffen.

- **Vektorraumbildung:** CLIP lernt, Bilder und Texte in einen hochdimensionalen Vektorraum zu transformieren. Dabei werden Bild- und Textrepräsentationen so angepasst, dass korrespondierende Paare nahe beieinander liegen.

- **Kontrastives Lernen:** Das Modell verwendet einen kontrastiven Lernansatz, um die Distanz zwischen übereinstimmenden Bild-Text-Paaren zu minimieren und die Distanz zwischen nicht übereinstimmenden Paaren zu maximieren. Dieses Training fördert eine effektive Abbildung beider Modalitäten in den gemeinsamen Raum.

2.2.3. Kontrastives Lernen im Detail

Der kontrastive Lernansatz von CLIP basiert darauf, dass das Modell lernt, korrespondierende Text-Bild-Paare miteinander zu verknüpfen, während es gleichzeitig nicht passende Paare unterscheidet. Dieser Ansatz erlaubt es dem Modell, eine umfassende und nuancierte Sicht auf die Beziehungen zwischen Texten und Bildern zu entwickeln, was für die Zero-Shot-Lernfähigkeiten von CLIP entscheidend ist.

Vektoreinbettungen

Sei \mathbf{I} ein Bild und \mathbf{T} ein Text, dann werden die Einbettungen durch die Encoder-Funktionen wie folgt generiert:

$$\begin{aligned}\text{Bild-Einbettung: } \mathbf{v}_I &= f_{\text{Bild}}(\mathbf{I}) \\ \text{Text-Einbettung: } \mathbf{v}_T &= f_{\text{Text}}(\mathbf{T})\end{aligned}$$

wobei f_{Bild} und f_{Text} die Funktionen des Bild- bzw. Text-Encoders sind.

Ähnlichkeitsberechnung

Die Ähnlichkeit zwischen einem Bild- und einem Textvektor wird durch das Skalarprodukt ihrer normalisierten Vektoren berechnet:

$$\text{sim}(\mathbf{I}, \mathbf{T}) = \frac{\mathbf{v}_I \cdot \mathbf{v}_T}{\|\mathbf{v}_I\| \|\mathbf{v}_T\|}$$

Das kann man in 2.4 sehen.

Training und kontrastiver Verlust

Das Training von CLIP erfolgt durch Minimierung des kontrastiven Verlustes. Für ein Batch von N Bild-Text-Paaren wird der Verlust für ein Paar (i, j) berechnet als:

$$L_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{v}_{I_i}, \mathbf{v}_{T_j})/\tau)}{\sum_{k=1}^N \exp(\text{sim}(\mathbf{v}_{I_i}, \mathbf{v}_{T_k})/\tau)}$$

Hierbei ist:

- τ ein Temperaturparameter ist, der die Schärfe der Wahrscheinlichkeitsverteilung beeinflusst,
- \mathbf{v}_{T_k} die Embeddings der negativen Beispiele (nicht korrespondierende Bilder),

Diese Formel zielt darauf ab, die Wahrscheinlichkeit zu maximieren, dass das korrekte Bild-Text-Paar unter Berücksichtigung aller anderen Paare im Batch als das ähnlichste Paar ausgewählt wird. Der Logarithmus im Zähler bewirkt, dass das korrekte Paar einen hohen Beitrag zum Gesamtverlust leistet, wenn es nicht als ähnlichstes Paar identifiziert wird. Im Nenner summiert sich der Beitrag aller Paare im Batch, was die Unterscheidung zwischen korrekten und inkorrekten Paaren fördert.

2.2.4. Zero-Shot Learning in CLIP

Konzept

Zero-Shot Learning bezieht sich auf die Fähigkeit des Modells, Aufgaben zu verstehen und durchzuführen, für die es nicht explizit trainiert wurde. Im Kontext von CLIP bedeutet dies, Bilder und Konzepte zu erkennen und zu interpretieren, die es während des Trainings nie gesehen hat.

Mechanismus

Dies wird durch das generalisierte Verständnis des Modells von Bildern und Texten erreicht. Durch das Training mit einem vielfältigen Datensatz lernt CLIP eine breite Palette von visuellen Stilen, Objekten und Konzepten, was ihm eine gute Generalisierung auf neue, ungesehene Aufgaben ermöglicht.

Anwendungen

Diese Eigenschaft macht CLIP außerordentlich vielseitig. Es kann für verschiedene Aufgaben wie Objekterkennung, Inhaltsmoderation und sogar kreative Anwendungen wie das Generieren von Bildern aus Textbeschreibungen verwendet werden.

```
model_id = "openai/clip-vit-base-patch32"
processor = CLIPProcessor.from_pretrained(model_id)
model = CLIPModel.from_pretrained(model_id)

image = read_image(image_name="2.jpg")
image = np.expand_dims(image, axis=0)

labels = ["A photo of a piano",
          "Someone playing the piano",
          "A photo of a guitar",
          "A photo of a piano in a white background",
          "A very big dog eating hotdogs",
          "A fluffy cat",
          "A photo of the earth from the dark space"]
```

```

labels = processor(
text=labels,
images=None,
padding=True,
return_tensors="pt"
).to(device)

text_emb = model.get_text_features(**labels)
text_emb = text_emb.detach().cpu().numpy()
text_emb = text_emb / np.linalg.norm(text_emb, axis=0)

image = processor(
text=None,
images=image,
return_tensors="pt"
)["pixel_values"].to(device)

image_emb = model.get_image_features(image)
image_emb = image_emb.detach().cpu().numpy()

similarities = np.dot(image_emb, text_emb.T)

index = np.argmax(similarities, axis=1).item()

result = labels[index]

```

Im obigen Codestück sieht man grob, wie man CLIP für image Classification ohne fine-tuning nutzen kann. Der komplette Code findet man auf <https://github.com/KenanKhauto/zero-shot-learning>. Die Labels habe ich selber manuell erstellt. Und das Bild, was ich als Eingabe benutzt habe, ist ein Klavier mit einem weißen Hintergrund. Und die Ausgabe des Models ist auch diese Beschreibung. Das zeigt vor allem, dass das Model nicht nur Objekte klassifiziert, sondern auch das komplette Bild versteht. Das war ein einfaches Beispiel über Image Classification. Wir werden aber im Section-2.3 noch ein detailliertes Beispiel sehen, wie man Object Localization und Object Detection mithilfe von CLIP machen kann.

2.2.5. Vergleichsanalyse

Traditionelle Vision-Modelle

Frühere Modelle in der Computer Vision waren in der Regel aufgabenspezifisch (z.B. Modelle zur Objekterkennung, Bildsegmentierung) und erforderten umfangreiche Feinabstimmungen mit beschrifteten Daten für jede Aufgabe.

Multimodale Modelle

Andere zeitgenössische Modelle wie Googles BERT und OpenAIs GPT-3 haben fortgeschrittene Fähigkeiten in ihren jeweiligen Bereichen (Text- und Sprachverarbeitung), sind jedoch nicht inhärent darauf ausgelegt, die Beziehung zwischen Text und Bildern zu verstehen, wie es bei CLIP der Fall ist.

Vorteile von CLIP

CLIPs Fähigkeit, textuelle und visuelle Informationen zu verstehen und zu verknüpfen, ohne aufgabenspezifisches Training, hebt es ab. Seine Fähigkeiten zum Zero-Shot Learning reduzieren deutlich den Bedarf an großen annotierten Datensätzen für neue Aufgaben.

2.3. Fallstudien / Anwendungen

Hier erwähnen wo und wie CLIP in image understanding paper benutzt wurde.

2.4. Diskussion von Herausforderungen und Grenzen

2.5. Zukünftige Richtungen und mögliche Verbesserungen

Füge Attention is all you need und Vits papers als Quellen fürs Verstehen, wie Transformer genau funktionieren.

Literatur

- Dosovitskiy, Alexey u. a. (2020). „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale“. In: *arXiv preprint arXiv:2010.11929*.
- Radford, Alec u. a. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. arXiv: [2103.00020 \[cs.CV\]](#).
- Vaswani, Ashish u. a. (2017). „Attention is all you need“. In: *Advances in neural information processing systems*, S. 5998–6008.

A. Anhang 1

```
^^I^^Iimport torch
^^I^^Iimport torchvision
^^I^^Iimport torchvision.transforms as transforms
^^I^^Ifrom transformers import CLIPProcessor, CLIPModel
^^I^^Iimport matplotlib.pyplot as plt
^^I^^Ifrom torch.utils.data import DataLoader
^^I^^Ifrom datasets import load_dataset
^^I^^Iimport numpy as np
^^I^^I
^^I^^Iimagenette = load_dataset(
^^I^^I^^I"frgfm/imagenette",
^^I^^I^^I"320px",
^^I^^I^^Isplit="validation",
^^I^^I^^Irevision="4d512db"
^^I^^I)
^^I^^I
^^I^^Ilabels = imagenette.info.features["label"].names
^^I^^Iclip_labels = [f"A photo of {x}" for x in labels]
^^I^^Iclip_labels
```