WSTĘP

Ta sekcja stanowi dynamiczne wprowadzenie do generatywnej sztucznej inteligencji (AI), koncentrując się na mechanizmie uwagi, który jest kluczowym elementem architektury transformatorowej. Dowiesz się również o kilku wpływowych firmach z branży AI. Pierwsza część wprowadza Cię w generatywną AI, w tym jej najważniejsze cechy i techniki. Dowiesz się również o różnicy między konwersacyjną a generatywną AI. Druga część rozpoczyna się krótkim wprowadzeniem do kilku firm wnoszących znaczący wkład w AI i przetwarzanie języka naturalnego (NLP). Będziesz dobrze zaznajomiony z tymi firmami, jeśli planujesz karierę w NLP. Trzecia część wprowadza koncepcję LLM (dużych modeli językowych). Czwarta część wprowadza koncepcję uwagi, która jest potężnym mechanizmem generowania osadzeń słów, zawierających informacje specyficzne dla kontekstu dla słów w zdaniach. Koncepcja iloczynu skalarnego wektorów leży u podstaw głównej zasady uwagi, a także word2vec i maszyn wektorów nośnych (czyli tzw. „sztuczki jądra").

CZYM JEST GENERATYWNA SZTUCZNA INTELIGENCJA?

Generatywna sztuczna inteligencja odnosi się do podzbioru modeli i technik sztucznej inteligencji zaprojektowanych w celu generowania nowych próbek danych, które są podobne w swej naturze do danego zestawu danych wejściowych. Celem jest wytworzenie treści lub danych, które nie były częścią oryginalnego zestawu treningowego, ale są spójne, istotne kontekstowo i utrzymane w tym samym stylu lub strukturze. Sztuczna inteligencja generatywna wyróżnia się swoją zdolnością do tworzenia i innowacji, w przeciwieństwie do samego analizowania lub klasyfikowania. Postępy w tej dziedzinie doprowadziły do przełomów w dziedzinach kreatywnych i praktycznych zastosowaniach, czyniąc ją pionierskim obszarem badań i rozwoju w dziedzinie sztucznej inteligencji.

Kluczowe cechy sztucznej inteligencji generatywnej

Poniższa lista zawiera najważniejsze cechy sztucznej inteligencji generatywnej, a następnie krótki opis każdej z nich:

- Generowanie danych
- Synteza
- Dystrybucje uczenia

Generowanie danych odnosi się do możliwości tworzenia nowych punktów danych, które nie są częścią danych treningowych, ale je przypominają. Może to obejmować tekst, obrazy, muzykę, filmy lub dowolną inną formę danych. Synteza oznacza, że modele generatywne mogą łączyć różne dane wejściowe, aby generować dane wyjściowe, które zawierają cechy z każdego z nich, na przykład łączenie stylów dwóch obrazów. Dystrybucje uczenia oznaczają, że modele sztucznej inteligencji generatywnej dążą do poznania rozkładu prawdopodobieństwa danych treningowych, aby móc generować nowe próbki na podstawie tego rozkładu.

Popularne techniki w generatywnej sztucznej inteligencji (AI)

Generatywne sieci przeciwstawne (GAN) składają się z dwóch sieci, generatora i dyskryminatora, które są trenowane jednocześnie. Generator stara się generować fałszywe dane, podczas gdy dyskryminator stara się odróżnić dane rzeczywiste od fałszywych. Z czasem generator staje się coraz lepszy w generowaniu realistycznych danych. Autoenkodery wariacyjne (VAE) to modele probabilistyczne, które uczą się kodować i dekodować dane, tak aby zakodowane reprezentacje mogły być wykorzystane do generowania nowych próbek danych. Rekurencyjne sieci neuronowe (RNN) są wykorzystywane głównie do generowania sekwencji, takich jak tekst czy muzyka.

Co wyróżnia generatywną sztuczną inteligencję

Tworzenie a klasyfikacja: Podczas gdy większość tradycyjnych modeli AI ma na celu klasyfikowanie danych wejściowych do predefiniowanych kategorii, modele generatywne mają na celu tworzenie nowych danych.

Uczenie bez nadzoru: Wiele modeli generatywnych, zwłaszcza sieci GAN i VAE, działa w sposób nienadzorowany, co oznacza, że nie wymagają one danych z etykietami do treningu. Różnorodne wyniki: Modele generatywne mogą generować szeroką gamę wyników w oparciu o wyuczone rozkłady, co czyni je idealnymi do zadań takich jak generowanie dzieł sztuki i transfer stylu.

Wyzwania: Generatywna sztuczna inteligencja stawia przed nami wyjątkowe wyzwania, takie jak załamanie trybu w sieciach GAN czy zapewnienie spójności generowanych treści. Istnieje wiele obszarów, w których wykorzystuje się generatywną sztuczną inteligencję, z których niektóre wymieniono poniżej:

• Tworzenie sztuki i muzyki

• Rozszerzanie danych

• Transfer stylu

• Generowanie tekstu

• Synteza obrazu

• Odkrywanie leków

Tworzenie sztuki i muzyki obejmuje generowanie obrazów, muzyki lub innych form sztuki. Rozszerzanie danych polega na tworzeniu dodatkowych danych dla modeli treningowych, zwłaszcza gdy oryginalny zbiór danych jest ograniczony. Transfer stylu odnosi się do zastosowania stylu jednego obrazu do treści innego. Generowanie tekstu polega na tworzeniu spójnego i kontekstowo istotnego tekstu. Synteza obrazu obejmuje generowanie realistycznych obrazów, twarzy, a nawet tworzenie scen do gier wideo. Odkrywanie leków dotyczy generowania struktur molekularnych dla nowych, potencjalnych leków.

SZTUKA KONWERSACYJNA A SZTUKA GENERATYWNA

Zarówno sztuczna inteligencja konwersacyjna, jak i sztuczna inteligencja generatywna to ważne poddziedziny w ramach szerszej dziedziny sztucznej inteligencji. Jednak te poddziedziny koncentrują się na innym aspekcie, jeśli chodzi o ich główny cel, wykorzystywane technologie i zastosowania. Główne różnice między tymi dwiema poddziedzinami przedstawiają się następująco:

• Cel główny

• Zastosowania

• Wykorzystywane technologie

• Szkolenie i interakcja

• Ewaluacja

• Wymagania dotyczące danych

Cel główny

Głównym celem sztucznej inteligencji konwersacyjnej jest ułatwienie interakcji między maszynami a ludźmi, przypominających interakcje ludzkie. Obejmuje to chatboty, wirtualnych asystentów i inne systemy, które angażują się w dialog z użytkownikami. Głównym celem sztucznej inteligencji generatywnej jest tworzenie nowych treści lub danych, które nie znajdowały się w zestawie treningowym, ale mają podobną strukturę i styl. Może to obejmować generowanie obrazów, muzyki i tekstu, a także bardziej złożone zadania, takie jak synteza wideo.

Zastosowania

Typowe zastosowania sztucznej inteligencji konwersacyjnej obejmują chatboty obsługi klienta, wirtualnych asystentów sterowanych głosem (takich jak Siri lub Alexa) oraz systemy interaktywnej odpowiedzi głosowej (IVR). Istnieje wiele zastosowań sztucznej inteligencji generatywnej, takich jak tworzenie grafiki lub muzyki, generowanie realistycznych środowisk gier wideo, synteza głosów oraz produkcja realistycznych obrazów lub deep fake'ów.

Wykorzystane technologie

Sztuczna inteligencja konwersacyjna często opiera się na technikach przetwarzania języka naturalnego (NLP) w celu zrozumienia i generowania języka ludzkiego. Obejmuje to rozpoznawanie intencji, ekstrakcję encji i zarządzanie dialogiem. Generatywna sztuczna inteligencja (AI) powszechnie wykorzystuje generatywne sieci przeciwstawne (GAN), autoenkodery wariacyjne (VAE) i inne modele generatywne do tworzenia nowej treści, takie jak ChatGPT i GPT-4.

Szkolenie i interakcja

Chociaż szkolenie może odbywać się w trybie nadzorowanym, półnadzorowanym lub nienadzorowanym, podstawowym trybem interakcji w przypadku konwersacyjnej AI jest dialog lub rozmowa. Proces szkolenia generatywnej AI, zwłaszcza w przypadku modeli takich jak GAN, obejmuje procesy iteracyjne, w których model uczy się generować dane, próbując oszukać dyskryminator i wmówić mu, że wygenerowane dane są prawdziwe.

Ewaluacja

Metryki oceny sztucznej inteligencji konwersacyjnej często koncentrują się na zrozumieniu i dokładności odpowiedzi, satysfakcji użytkownika oraz płynności generowanych odpowiedzi. Metryki oceny sztucznej inteligencji generatywnej dla modeli takich jak sieci GAN mogą być trudne i mogą wymagać wykorzystania połączenia metryk ilościowych i ludzkiej oceny w celu oceny jakości generowanych treści.

Wymagania dotyczące danych

Wymagania dotyczące danych dla sztucznej inteligencji konwersacyjnej zazwyczaj obejmują dane dialogowe, dotyczące rozmów między ludźmi lub między ludźmi a botami. Niezbędne są duże zbiory danych dotyczące rodzaju treści, które sztuczna inteligencja ma generować (takich jak obrazy, tekst i muzyka). Chociaż zarówno sztuczna inteligencja konwersacyjna, jak i sztuczna inteligencja generatywna zajmują się generowaniem wyników, ich główne cele, zastosowania i metodologie mogą się znacznie różnić. Sztuczna inteligencja konwersacyjna została zaprojektowana z myślą o interaktywnej komunikacji z użytkownikami, podczas gdy sztuczna inteligencja generatywna koncentruje się na tworzeniu nowych, oryginalnych treści.

CZY DALL-E JEST CZĘŚCIĄ GENERATYWNEJ SZTUCZNEJ INTELIGENCJI?

DALL-E i podobne narzędzia generujące grafikę z tekstu są przykładami generatywnej sztucznej inteligencji. W rzeczywistości DALL-E jest jednym z najbardziej znanych przykładów generatywnej sztucznej inteligencji w dziedzinie syntezy obrazu.

Oto lista cech generatywnych DALL-E wraz z krótkim opisem każdego elementu:

• Generowanie obrazu

• Dystrybucje uczenia

• Innowacyjne kombinacje

• Szerokie zastosowanie

• Architektura transformatorowa

Generowanie obrazu to kluczowa cecha DALL-E, który został zaprojektowany do generowania obrazów na podstawie opisów tekstowych. Po otrzymaniu polecenia, takiego jak „dwugłowy flaming", DALL-E może wygenerować nowy obraz, który pasuje do opisu, nawet jeśli taki obraz nie został uwzględniony w danych treningowych.

Dystrybucje uczenia: Podobnie jak inne modele generatywne, DALL-E uczy się rozkładu prawdopodobieństwa swoich danych treningowych. Podczas generowania obrazu pobiera próbki z tego wyuczonego rozkładu, aby wygenerować wizualizacje, które są wiarygodne w oparciu o wyniki treningu.

Innowacyjne kombinacje: DALL-E potrafi generować obrazy reprezentujące nowe lub abstrakcyjne koncepcje, prezentując swoją zdolność do łączenia i rekombinacji poznanych elementów w innowacyjny sposób. Oprócz syntezy obrazu, DALL-E zapewnia szerokie wsparcie w obszarach takich jak generowanie grafiki, mieszanie stylów i tworzenie obrazów o określonych atrybutach lub motywach, podkreślając jego wszechstronność jako narzędzia generatywnego. DALL-E wykorzystuje wariant architektury transformatorowej, podobny do modeli takich jak GPT-3, ale dostosowany do zadań generowania obrazu. Inne narzędzia, które generują grafikę, grafikę lub dowolną formę treści wizualnej na podstawie danych wejściowych (czy to tekstu, innego obrazu, czy jakiejkolwiek innej formy danych) i mogą generować wyniki, które nie są jawnie obecne w danych treningowych, są również uważane za generatywną sztuczną inteligencję. Prezentują one zdolność modeli sztucznej inteligencji nie tylko do analizy i klasyfikacji, ale także do tworzenia i innowacji.

CZY CHATGPT-3 I GPT-4 SĄ CZĘŚCIĄ GENERATYWNEJ AI?

Zarówno ChatGPT-3, jak i GPT-4 to modele LLM i przykłady generatywnej AI. Należą do klasy modeli zwanych „transformatorami", które są szczególnie skuteczne w obsłudze sekwencji danych, takich jak zadania związane z tekstem. Poniższa lista przedstawia powody, dla których te modele LLM są uważane za generatywne, a następnie krótki opis każdego z nich:

• Generowanie tekstu

• Dystrybucje uczenia się

• Szerokie zastosowania

• Uczenie się bez nadzoru

Generowanie tekstu: Modele te mogą generować spójne, kontekstowo istotne i często wysoce zaawansowane sekwencje tekstu w oparciu o podane wskazówki. Generują odpowiedzi, które nie były

jawnie obecne w ich danych treningowych, ale są konstruowane w oparciu o wzorce i struktury, których nauczyły się podczas treningu. Dystrybucje uczenia się: Modele GPT-3, GPT-4 i podobne uczą się rozkładu prawdopodobieństwa swoich danych treningowych. Podczas generowania tekstu zasadniczo pobierają próbki z tego wyuczonego rozkładu, aby wygenerować sekwencje, które są prawdopodobne na podstawie ich treningu.

Szerokie zastosowania: Poza czatem lub rozmową tekstową, modele te mogą być wykorzystywane do różnorodnych zadań generatywnych, takich jak pisanie opowiadań, generowanie kodu, poezja oraz tworzenie treści w określonych stylach lub naśladowanie konkretnych autorów, prezentując ich możliwości generatywne.

Uczenie bez nadzoru: Chociaż można je dostrajać za pomocą konkretnych zestawów danych, modele takie jak GPT-3 są trenowane głównie w sposób bez nadzoru na ogromnych ilościach tekstu, ucząc się generować treści bez konieczności jawnego etykietowania danych dla każdej możliwej odpowiedzi. W istocie ChatGPT-3, GPT-4 i podobne modele firmy OpenAI są kwintesencją generatywnej sztucznej inteligencji w dziedzinie przetwarzania i generowania języka naturalnego. W kolejnych sekcjach pokrótce przedstawiono niektóre firmy o silnej pozycji w świecie sztucznej inteligencji.

DEEPMIND

Firma DeepMind wniosła znaczący wkład w rozwój sztucznej inteligencji (AI), w tym w tworzenie różnych systemów AI. Firma DeepMind powstała w 2010 roku i w 2014 roku stała się spółką zależną Google. DeepMind stworzył model językowy Gopher o pojemności 280 GB, który znacznie przewyższa konkurencję, w tym GPT-3, J1-Jumbo i MT-NLG. DeepMind opracował również AlphaFold, który w 30 minut rozwiązał problem zwijania białek, z którym badacze borykali się przez dziesięć lat. DeepMind udostępnił AlphaFold za darmo dla wszystkich w lipcu 2021 roku. DeepMind wniósł znaczący wkład w rozwój systemów gier AI, z których niektóre omówiono w następnej sekcji.

DeepMind i gry

DeepMind jest siłą napędową systemów AI AlphaStar, który gra w StarCrafta, oraz AlphaGo, który pokonał najlepszych graczy w Go (znacznie trudniejszym niż szachy). Te gry dostarczają „doskonałych informacji", podczas gdy gry z „niedoskonałymi informacjami" (takie jak poker) stanowiły wyzwanie dla modeli uczenia maszynowego (ML). AlphaGo Zero (następca AlphaGo) opanował grę dzięki samodzielnej grze w krótszym czasie i przy mniejszej mocy obliczeniowej. AlphaGo Zero wykazał się niezwykłą wydajnością, pokonując AlphaGo 100–0. Innym potężnym systemem jest AlphaZero, który również wykorzystywał technikę samodzielnej gry w Go, szachy i shogi; system osiągnął wyniki SOTA („State Of The Art"). Dla porównania, modele uczenia maszynowego wykorzystujące przeszukiwanie drzewa dobrze sprawdzają się w grach z doskonałą informacją. Z kolei gry z niedoskonałą informacją (takie jak poker) zawierają ukryte informacje, które można wykorzystać do opracowania strategii kontrujących strategie przeciwników. W szczególności AlphaStar jest w stanie grać z najlepszymi graczami w StarCraft II i stał się pierwszą sztuczną inteligencją, która osiągnęła wyniki SOTA w grze wymagającej „strategicznych zdolności w świecie niedoskonałej informacji".

Gracz w Gry (PoG)

Zespół DeepMind w Google opracował uniwersalny algorytm PoG (Gracz w Gry), który opiera się na następujących technikach:

• CFR (minimalizacja kontrfaktycznego żalu)

• CVPN (kontrfaktyczna sieć wartości i polityki)

• GT-CFT (rosnące drzewo CFR)

• CVPN

Kontrfaktyczna sieć wartości i polityki (CVPN) to sieć neuronowa, która oblicza kontrfaktyczne wartości dla każdego przekonania o stanie w grze. Ma to kluczowe znaczenie dla oceny różnych wariantów gry w dowolnym momencie. Growing Tree CFR (GT-CFR) to odmiana CFR zoptymalizowana pod kątem drzew gry, które rosną w czasie. GT-CFR opiera się na dwóch podstawowych fazach, które omówiono bardziej szczegółowo online

OPENAI

OpenAI to firma badawcza zajmująca się sztuczną inteligencją, która wniosła znaczący wkład w rozwój sztucznej inteligencji, w tym DALL-E i ChatGPT. OpenAI została założona w San Francisco przez Elona Muska i Sama Altmana (a także innych), a jednym z jej deklarowanych celów jest rozwój sztucznej inteligencji przynoszącej korzyści ludzkości. Biorąc pod uwagę ogromne inwestycje Microsoftu i głęboką współpracę z tą organizacją, OpenAI można postrzegać jako część Microsoftu. OpenAI jest twórcą serii GPT-x LLM, a także ChatGPT, który został udostępniony 30 listopada 2022 roku. Firma OpenAI udostępniła GPT-3 komercyjnie za pośrednictwem API do użytku w różnych aplikacjach, pobierając opłaty za słowo. GPT-3 został ogłoszony w lipcu 2020 roku i udostępniony w ramach programu beta. Następnie, w listopadzie 2021 roku, OpenAI udostępniło GPT-3 wszystkim użytkownikom. OpenAI opracowało również DALL-E, który generuje obrazy z tekstu. Początkowo OpenAI nie pozwalało użytkownikom na przesyłanie obrazów zawierających realistyczne twarze. Później OpenAI zmieniło swoją politykę, zezwalając użytkownikom na przesyłanie twarzy do swojego systemu online. Nawiasem mówiąc, modele dyfuzji zastąpiły testy porównawcze DALL-E. Firma OpenAI wydała publiczną wersję beta Embeddings, formatu danych odpowiedniego do różnych typów zadań z wykorzystaniem uczenia maszynowego, zgodnie z opisem online. OpenAI jest twórcą Codex, który udostępnia zestaw modeli wytrenowanych w zakresie przetwarzania języka naturalnego (NLP). Początkowa wersja Codex była w fazie prywatnej wersji beta; OpenAI udostępnia cztery modele, zwane łącznie „Instruct Models", które obsługują funkcję GPT-3 w zakresie generowania języka naturalnego. Modele te zostaną wycofane na początku stycznia 2024 r. i zastąpione zaktualizowanymi wersjami GPT-3, ChatGPT i GPT-4.

COHERE

Cohere to startup i konkurent OpenAI. Cohere rozwija najnowocześniejszą technologię przetwarzania języka naturalnego (NLP), która jest komercyjnie dostępna dla wielu branż. Cohere koncentruje się na modelach, które przeprowadzają analizę tekstu, a nie na modelach do generowania tekstu (takich jak modele oparte na GPT). Zespół założycielski Cohere jest imponujący: prezes Aidan Gomez jest jednym ze współtwórców architektury transformatorowej, a dyrektor techniczny Nick Frosst jest protegowanym Geoffa Hintona.

HUGGING FACE

Hugging Face to popularne, społecznościowe repozytorium technologii NLP o otwartym kodzie źródłowym. W przeciwieństwie do OpenAI czy Cohere, Hugging Face nie tworzy własnych modeli NLP. Zamiast tego Hugging Face to platforma zarządzająca mnóstwem modeli NLP o otwartym kodzie źródłowym, które klienci mogą dostrajać, a następnie wdrażać. Hugging Face stał się wiodącym miejscem współpracy nad modelami NLP i jest czasami określany jako „GitHub dla uczenia maszynowego i NLP".

Biblioteki Hugging Face

Hugging Face udostępnia trzy ważne biblioteki: zbiory danych, tokenizatory i transformatory. Biblioteka Accelerate obsługuje modele PyTorch. Biblioteka datasets udostępnia zestaw bibliotek dla NLP. Biblioteka tokenizers umożliwia konwersję danych tekstowych na wartości liczbowe. Prawdopodobnie najbardziej imponującą biblioteką jest biblioteka Transformers, która oferuje ogromny zestaw wstępnie wytrenowanych modeli opartych na BERT do wykonywania szerokiej gamy zadań NLP.

Hugging Face Model Hub

Hugging Face udostępnia centrum modelowe, które udostępnia mnóstwo modeli dostępnych online. Ponadto strona internetowa obsługuje testowanie modeli online, co obejmuje następujące zadania:

• Maskowane uzupełnianie słów z BERT

• Rozpoznawanie jednostek nazwanych z Electra

• Wnioskowanie w języku naturalnym z RoBERTa

• Odpowiadanie na pytania z DistilBERT

• Podsumowanie z BART

• Generowanie tekstu z GPT-2

• Tłumaczenie z T5

Przejdź do poniższego linku, aby zapoznać się z możliwościami generowania tekstu w narzędziu „write with transformer": https://transformer.huggingface.co. W kolejnej części zobaczysz przykłady kodu w Pythonie, które pokazują, jak wyświetlić wszystkie dostępne zbiory danych Hugging Face i jak załadować konkretny zbiór danych.

AI21

AI21 to firma dostarczająca zastrzeżone, rozbudowane modele językowe za pośrednictwem API, wspierające aplikacje swoich klientów. Obecny model SOTA AI21 nosi nazwę Jurassic-1 (mniej więcej tej samej wielkości co GPT-3), a AI21 tworzy również własne aplikacje bazujące na Jurassic-1 i innych modelach. Obecny pakiet aplikacji AI21 obejmuje narzędzia, które mogą wspomagać czytanie i pisanie. Primer to starszy konkurent w tej dziedzinie, założony dwa lata przed wynalezieniem transformatora. Firma obsługuje głównie klientów z sektora rządowego i obronnego.

INFLECTIONAI

Nowszą firmą w branży sztucznej inteligencji jest InflectionAI, której imponujący zespół założycielski obejmuje:

• Reid Hoffman (LinkedIn)

• Mustafa Suleyman, współzałożyciel DeepMind

• Karen Simonyan, badaczka DeepMind

InflectionAI stawia sobie ambitne zadanie: umożliwienie ludziom interakcji z komputerami w taki sam sposób, w jaki komunikują się między sobą.

ANTHROPIC

Anthropic został stworzony w 2021 roku przez byłych pracowników OpenAI. Anthropic korzysta ze znacznego wsparcia finansowego ze strony wielu firm, w tym Google i Salesforce. Claude 2 umożliwia streszczanie do 75 000 słów tekstu, podczas gdy ChatGPT ma obecnie limit 3000 słów. Co więcej, Claude 2 uzyskał wynik 76,5% w częściach egzaminu adwokackiego i 71% w teście z programowania w Pythonie. Claude 2 ma również wyższy wskaźnik niż ChatGPT pod względem udzielania „czystych" odpowiedzi na zapytania użytkowników. To kończy część dotyczącą firm z branży AI, które wnoszą istotny wkład w rozwój tej dziedziny. Kolejna sekcja stanowi ogólne wprowadzenie do studiów LLM.

CZYM SĄ LLMS?

Duże modele językowe, LLM, oparte są na architekturze transformatorowej. Istnieje wiele modeli LLM o różnej wielkości, z których wiele jest większych niż modele oparte na BERT. W związku z tym, niniejsza część zawiera bardzo krótki przegląd środowiska LLM. Modele LLM są znane ze swojej wielkości, która zazwyczaj wynosi co najmniej 10 miliardów parametrów (BERT ma „tylko" 1,5 miliarda parametrów). Co więcej, wymagają one bardzo dużych zbiorów danych na etapie uczenia, co może wymagać tygodni czasu szkolenia i kosztować miliony dolarów. Niestety, wiążą się z tym koszty środowiskowe. Według niektórych, ślad węglowy szkolenia GPT-3 jest porównywalny z podróżą samochodu na Księżyc w obie strony. Poza BERT i rodziną BERT, prawdopodobnie najbardziej znanym modelem LLM, który wzbudził ogromne zainteresowanie, jest GPT-3. GPT-3 składa się ze 175 miliardów parametrów, czyli prawie 120 razy więcej niż BERT. Istnieją jednak modele o znacznie większej liczbie parametrów: Switch wykorzystał 540 miliardów parametrów, a DAO ponad 1,2 biliona parametrów. Niedawno (listopad 2022 r.) ChatGPT wywarł sensacyjny wpływ na opinię publiczną (100 milionów zarejestrowanych użytkowników w ciągu dwóch miesięcy).

Rozmiar modelu a rozmiar zestawu treningowego

Chociaż kuszące może być stwierdzenie, że sam rozmiar jest najważniejszym czynnikiem, rozmiar zestawu danych treningowych ma większe znaczenie. To stwierdzenie przeczy wynikom Kaplana i jego zespołu, którzy badali rozmiar zestawu treningowego w porównaniu z rozmiarem modelu. Przyjrzyjmy się pokrótce niektórym opracowanym modelom LLM. Model Chinchilla LLM firmy DeepMind składa się z 70 miliardów parametrów i jest w stanie przewyższyć GPT-3, Jurassic-1 (178 miliardów) i Megatron-Turing (530 miliardów), ponieważ jego zestaw danych treningowych był pięć razy większy niż zestawy danych treningowych dla innych modeli LLM. Pomimo imponujących wyników LLM-ów i długo oczekiwanej funkcjonalności GPT-4 (wydanego 14 marca 2023 r.), LLM-y nie są w stanie rozumieć języka w sposób, w jaki rozumieją go ludzie. Zdolność jednostki do dokonywania inteligentnych wyborów porównywalnych z tymi dokonywanymi przez ludzi nie dowodzi, że jednostka ta rzeczywiście rozumie te wybory w taki sam sposób jak człowiek. Czy LLM-y rozumieją język? Jako zabawną i częściowo powiązaną analogię, rozważmy następującą historię o dwóch arcymistrzach szachowych, człowieku zaufania i 12-letnim chłopcu, którzy podróżują statkiem transatlantyckim na początku XX wieku. Gdy statek był kilka godzin od celu, człowiek zaufania podjął zuchwały zakład, że w ciągu dwóch godzin nauczy młodego chłopca gry w szachy tak, aby mecze zakończyły się remisem lub zwycięstwem chłopca. Jednakże arcymistrzowie i chłopiec musieli grać w zadaszonym pomieszczeniu, przypominającym szafę, a trzej uczestnicy nie mogli się ze sobą w żaden sposób komunikować. Arcymistrzowie przyjęli wyzwanie, spodziewając się, że wykorzystają swoją ogromną wiedzę przeciwko młodemu zawodnikowi. Jednak w miarę postępu partii arcymistrzowie byli zszokowani szybkością i wyrafinowaniem ruchów szachowych chłopca. Ich pewność siebie szybko ustąpiła miejsca obawie, a następnie desperacji. W końcu jeden arcymistrz zaproponował remis, a drugi arcymistrz poddał mecz. Oszustwo było niezwykle proste: za każdym razem, gdy jeden arcymistrz wykonywał ruch szachowy, chłopiec wykonywał ten sam ruch przeciwko drugiemu arcymistrzowi, co w praktyce oznaczało, że arcymistrzowie rywalizowali ze sobą. Na szczęście dla oszusta, statek dotarł do celu, a on zdołał

odebrać nagrodę i chłopca, a następnie wysiąść, zanim arcymistrzowie zdali sobie sprawę, że zostali oszukani. Sednem poprzedniej historii jest to, że chłopiec wykonał niezwykle inteligentne ruchy szachowe, ale niekoniecznie rozumiał logikę ich wykonania. Zatem, jeśli człowiek nie rozumie logiki stojącej za działaniem, sugeruje to, że jest jeszcze mniej prawdopodobne, aby LLM rozumiał powody swoich zaleceń na poziomie ludzkim.

Zastrzeżenia dotyczące modeli LLM

Chociaż modele LLM mogą dawać imponujące wyniki, należy pamiętać o następujących kwestiach:

• mniejsze modele mogą przewyższać większe modele (w niektórych zadaniach)

• modele generują fałszywe wyniki („halucynacje")

• niektóre modele przetwarzają długie dokumenty, a inne są bardziej konwersacyjne.

• rosnące znaczenie wektorowych baz danych

• większość modeli nie jest trenowana na aktualnych zestawach danych (tylko do pewnego momentu).

Na przykład ChatGPT został wytrenowany na danych sprzed września 2021 r. Dwie funkcje, których znaczenie rośnie, to

1. możliwość przechowywania historii poprzednich konwersacji

2. możliwość wyszukiwania informacji w internecie na bieżąco

Co więcej, warto porównać cechy modeli open source z modelami zamkniętymi. Yann LeCunn rzekomo zauważył, że „open source ostatecznie wygra wyścig sztucznej inteligencji".

Funkcje straty

Modele głębokiego uczenia obejmują modele oparte na architekturze transformatorowej. Takie modele opierają się na funkcji straty w procesie trenowania. Dokładniej rzecz biorąc, funkcja straty to funkcja różniczkowalna, która określa błąd wynikający z przewidywań danego modelu. Podczas każdego przebiegu wstecznego (zwanego „wsteczną propagacją błędu" lub „wsteczną propagacją") w modelu, funkcja straty oblicza gradient (tj. wektor pochodnych cząstkowych) parametrów modelu, aby zaktualizować wagi parametrów w celu poprawy dokładności modelu. W trakcie procesu trenowania sekwencja liczb dotyczących dokładności modelu nie zawsze jest monotonicznie rosnącym zbiorem liczb: niektóre iteracje działają gorzej niż ich bezpośrednie poprzedniczki, a czasami nie ma żadnej poprawy.

CZYM JEST DRYFT AI?

Termin dryft AI odnosi się do scenariusza, w którym modele LLM reagują na dane wejściowe w nieoczekiwany sposób. Dryft AI może różnić się od halucynacji:

Pierwszy z nich wiąże się z regularnym lub stale nieoczekiwanym zachowaniem, podczas gdy halucynacje mogą występować losowo, ale niekoniecznie według spójnego wzorca. W rzeczywistości można nawet „wywołać" zachowania przypominające halucynacje w modelach LLM, wywołując metodę Completion.create() , w której parametr temperatury jest ustawiony na wartość większą niż 1,0. Jedną z potencjalnych przyczyn dryftu AI może być niezamierzony skutek poprawy jednego obszaru modeli LLM, który nieumyślnie prowadzi do pogorszenia wydajności w innych obszarach modelu LLM. Inną potencjalną przyczyną dryftu jest dryft modelu, który może wystąpić, gdy dane produkcyjne znacząco różnią się od danych treningowych. W takim scenariuszu należy monitorować dane

produkcyjne i treningowe, a także prognozy. Konsekwencje dryftu AI mogą być znaczące. Na przykład proces decyzyjny jest podatny na dryft AI, który może prowadzić do podejmowania decyzji na podstawie błędnych założeń. To z kolei może prowadzić do niższej jakości prognoz i rekomendacji, a także negatywnie wpłynąć na zadowolenie klienta.

UCZENIE MASZYNOWE I DRYF (OPCJA)

W terminologii uczenia maszynowego dryf odnosi się do dowolnego rodzaju zmiany w rozkładzie w danym okresie. Dryf modelu odnosi się do zmiany (dryfu) dokładności prognozy modelu, natomiast dryf danych odnosi się do zmiany rodzaju gromadzonych danych. (Należy pamiętać, że dryf danych jest również nazywany dryfem danych wejściowych, dryfem cech lub dryfem kowariancji). Istnieje kilka czynników wpływających na wartość danych, takich jak dokładność, trafność i wiek. Na przykład sklepy stacjonarne sprzedające telefony komórkowe znacznie częściej sprzedają nowsze modele telefonów niż starsze. W niektórych przypadkach dryf danych występuje przez pewien czas, a w innych wynika z faktu, że niektóre dane nie są już istotne ze względu na zmiany związane z cechami w aplikacji. Na dryf danych w określonym zestawie danych może wpływać wiele czynników. Dwie techniki radzenia sobie z dryftem danych to klasyfikator domen i detektor przesunięć czarnej skrzynki. Oprócz powyższych typów dryftu, w zbiorze danych mogą wystąpić inne rodzaje zmian:

• Przesunięcie koncepcji

• Przesunięcie kowariancji

• Przesunięcie domeny

• Przesunięcie prawdopodobieństwa a priori

• Przesunięcie korelacji pozornej

• Przesunięcie subpopulacji

• Przesunięcie w czasie

Przeprowadź wyszukiwanie online, aby znaleźć więcej informacji na tematy z powyższej listy. Poniższa lista zawiera strony internetowe z narzędziami open source opartymi na Pythonie, które umożliwiają wykrywanie dryfu:

• alibi-detect (https://github.com/SeldonIO/alibi-detect)

• evidently (https://github.com/evidentlyai/evidently)

• Torchdrift (http://torchdrift.org/)

Mając na uwadze powyższe tematy, skupimy się na koncepcji uwagi, która jest tematem kolejnej sekcji.

CZYM JEST UWAGA?

Uwaga to mechanizm w architekturze transformatorowej, za pomocą którego określane są kontekstowe osadzenia słów w korpusie. W przeciwieństwie do word2vec lub gloVe, mechanizm uwagi uwzględnia wszystkie słowa w zdaniu podczas tworzenia osadzenia słów dla danego słowa w danym zdaniu. Rozważmy na przykład następującą listę zdań:

I went to the bank.

I sat on the river bank.

The road will bank to the right

Jak widać, słowo „bank” jest przeciążone w tym sensie, że ma inne znaczenie w każdym z trzech zdań (dwa razy jako rzeczownik i raz jako czasownik). Mechanizm uwagi w architekturze transformatorowej generuje inny wektor kontekstowy (tj. jednowymiarowy wektor z liczbami zmiennoprzecinkowymi). W związku z tym to samo słowo w różnych (i odrębnych) zdaniach będzie miało inne osadzenie słowa w każdym z tych zdań. Co ciekawe, mechanizm uwagi istniał przed architekturą transformatorową: wykorzystanie uwagi w tej drugiej to właśnie tam mechanizm uwagi zyskał na znaczeniu. Z kolei algorytm word2vec (opracowany w 2013 roku przez Google) tworzy pojedyncze osadzenie słowa (wektor) dla słowa „bank” (a także dla każdego innego słowa w danym zdaniu). Nie jest to krytyka word2vec, który był ważnym przełomem dla przetwarzania języka naturalnego w momencie jego powstania.

Pochodzenie uwagi

Zanim opracowano mechanizm uwagi, popularne architektury opierały się na innych architekturach głębokiego uczenia, takich jak sieci neuronowe RNN, sieci LSTM lub bi-LSTM. W rzeczywistości mechanizm uwagi był początkowo używany w połączeniu z sieciami neuronowymi RNN lub sieciami LSTM. Zespół Google przeprowadził jednak kilka eksperymentów z zadaniami tłumaczenia maszynowego na modelach, które opierały się wyłącznie na mechanizmie uwagi i architekturze transformatora, i odkrył, że modele te osiągały wyższą wydajność niż modele zawierające sieci CNN, RNN lub LSTM. Wynik ten doprowadził do sformułowania „attention is all you need” (uwaga to wszystko, czego potrzebujesz), które jest tytułem przełomowego artykułu dotyczącego architektury transformatora. Co ciekawe, usunięcie sieci RNN przyniosło dwie dodatkowe korzyści. Po pierwsze, sieci RNN były stosunkowo powolne ze względu na obliczenia sekwencyjne, których nie można zrównoleglić. Po drugie, sieci RNN borykały się z problemem zanikającego gradientu. W konsekwencji usunięcie sieci RNN wyeliminowało obie te wady. W dużym uproszczeniu, transformator składa się z kodera i dekodera, z których każdy zawiera stos warstw uwagi, wykonujących operacje związane z uwagą.

Samouwaga

Samouwaga może być zrównoleglona, dzięki czemu obliczenia są wykonywane niezależnie. W związku z tym implementacje samouwagi wymagają złożoności czasowej i pamięciowej rzędu O(N**2). Naukowcy z Google opracowali algorytm, który znacząco redukuje złożoność pamięciową do O(log N). Chociaż standardowe implementacje samouwagi charakteryzują się złożonością czasową i pamięciową, najnowsze prace naukowców z Google pokazują, że złożoność pamięciową można zredukować poprzez prostą zmianę kolejności operacji. Jeśli znasz sieci neuronowe RNN i LSTM, wiesz, że obie architektury przetwarzają tokeny sekwencyjnie i mogą śledzić kolejność, w jakiej tokeny pojawiają się w sekwencji wejściowej. Z kolei architektura transformatorowa może przetwarzać słowa w zdaniu w sposób równoległy, a kolejność, w jakiej słowa pojawiają się w sekwencji wejściowej, nie jest zachowywana. Architektura transformatorowa wykorzystuje mechanizm zwany kodowaniem pozycyjnym (omówionym w rozdziale 3) do śledzenia sekwencji słów w sekwencji wejściowej.

GAtt (Ghost Attention)

GAtt to algorytm używany do precyzyjnego dostrajania czatu LLM LlaMa 2 firmy Meta . GAtt został opracowany w celu rozwiązania problemu, w którym ChatGPT ma tendencję do „zapominania” instrukcji podczas długich konwersacji.

Rodzaje uwagi i algorytmy

Oprócz uwagi hierarchicznej istnieją dodatkowe rodzaje mechanizmów uwagi, z których trzy to:

1. samouwaga

2. globalny/miękki

3. lokalny/twardy

Samouwaga próbuje określić, w jaki sposób słowa w zdaniu są ze sobą powiązane. Uwaga wielogłowicowa wykorzystuje blok wielokrotnej samouwagi zamiast tylko jednej samouwagi. Jednak każda głowica przetwarza inną sekcję wektora osadzania. Oprócz powyższych mechanizmów uwagi dostępnych jest również kilka algorytmów uwagi:

• Addytywny

• Oparty na treści

• Iloczyn skalarny

• Ogólny

• Bazujący na lokalizacji

• Skalowany iloczyn skalarny <= – ten algorytm wykorzystuje transformator

Wzory mechanizmów uwagi dzielą się na dwie kategorie: wzory, które wymagają iloczynu skalarnego wektorów (czasami ze współczynnikiem skalowania) oraz wzory, które stosują funkcję softmax() lub tanh() do iloczynów macierzy i wektorów. Architektura transformatorowa wykorzystuje mechanizm skalowanego iloczynu skalarnego do obliczania uwagi. Dostępnych jest kilka typów uwagi. Zaletą uwagi jest zrównoleglanie obliczeń obejmujących iloczyny skalarne między parami wektorów słów, podczas gdy sieci neuronowe RNN i LSTM wymagają obliczeń sekwencyjnych zamiast zrównoleglania. Uwaga: architektura transformatorowa wykorzystuje mechanizm skalowanego iloczynu skalarnego do obliczania uwagi.

Uwaga w GPT-2 a BERT

LLM-y mogą różnić się mechanizmem samouwagi. Na przykład samouwaga w komponencie dekodera GPT-2 różni się od samouwagi w komponencie kodera BERT .

OBLICZANIE UWAGI: UJĘCIE WYSOKIEGO POZIOMU

Obliczanie wartości opartych na uwadze obejmuje wiele kroków. Aby ułatwić zrozumienie tego procesu, przyjrzyjmy się krokowi propagacji w przód w sieciach neuronowych.

Propagacja w przód w sieciach neuronowych

Etap propagacji w przód w sieciach neuronowych może pomóc nam zrozumieć rolę macierzy Q, K i V w mechanizmie uwagi dla architektury transformatora. Przypomnijmy, że sieć neuronowa składa się z warstwy wejściowej, jednej lub więcej warstw ukrytych oraz warstwy wyjściowej. Dla dowolnej pary sąsiednich warstw w sieci neuronowej wagi parametrów (tj. krawędzi) między sąsiednimi warstwami są reprezentowane przez macierz W. Oczywiście, zawartość macierzy W jest różna dla różnych par sąsiednich warstw w sieci neuronowej. Przejście w przód rozpoczyna się od warstwy wejściowej i przebiega następująco, aż do warstwy wyjściowej:

1. warstwa wejściowa to wektor v1

2. v2 = macierz W pomnożona przez wektor v1

3. v3 = funkcja aktywacji zastosowana do v2

4. powtórz kroki 2, 3 i 4

Macierz W można zainicjować małymi, losowymi wartościami (lub użyć algorytmu He), a wartości te są aktualizowane podczas wstecznej propagacji błędów (która ma miejsce po zakończeniu przejścia w przód). Zatem wartości w macierzy W są dostrajane podczas procesu uczenia z udziałem sieci neuronowej.

Macierze Q, K i V transformatora

Można postrzegać te trzy macierze jako ich rolę w przeprowadzaniu wyszukiwania internetowego za pomocą wyszukiwarki. Typowy przypadek użycia obejmuje następujące czynności:

• Określenie ciągu wyszukiwania (zapytanie)

• Wybranie jednej z opcji zwróconych przez wyszukiwarkę (klucz)

• Wyświetlenie zawartości wybranej opcji (wartość)

Chociaż możesz nie myśleć o środkowym kroku podczas wyszukiwania, ważne jest, aby postrzegać wyszukiwanie jako trzy kroki z poprzedniej listy. Wyobraźmy sobie zapytanie, klucz i wartość jako wektory na płaszczyźnie euklidesowej (dla uproszczenia). Po wyświetleniu wyników zwróconych dla ciągu zapytania zobaczysz listę wielu linków (i wiele dodatkowych stron zawierających wiele linków). Wyobraź sobie każdy link jako klucz, któremu odpowiada wektor na płaszczyźnie euklidesowej. Koncepcyjnie, wybrany link odpowiada wektorowi najbliższemu wektorowi zapytania. Co więcej, czynność wybrania (tj. kliknięcia) tego linku powoduje, że wyszukiwarka pobiera wartość tego linku (klucz), która jest treścią powiązanego linku. Podsumowując, ciąg wejściowy (zapytanie) jest dopasowywany do najbliższego wektora (klucza) ze zbioru wektorów (kluczy), aby pobrać zawartość (wartość) skojarzoną z wybranym wektorem (kluczem).

Kroki obliczania uwagi

Chociaż proces obliczania wartości uwagi różni się od obliczeń macierzy W w przykładzie z sieciami neuronowymi omówionym we wcześniejszej sekcji, odpowiednik macierzy W obejmuje trzy macierze Q, K i V, z których każda jest inicjowana losowo. Sekwencja kroków obliczania macierzy uwagi jest wymieniona poniżej:

• Tokenizacja zdania wejściowego.

• Wygenerowanie wektora osadzania dla każdego tokena.

• Utworzenie macierzy X z wektorów osadzania.

• Zainicjowanie trzech macierzy: Q, K i V.

• Obliczenie Q*Kt z wartościami parami.

• Podzielenie przez skalar dk2 = sqrt (wymiarowość wektora klucza).

• Zastosowanie funkcji softmax() do Q*Kt/dk2.

• Oblicz macierz uwagi Z.

Alternatywny (i krótszy) sposób opisu sekwencji kroków obliczania macierzy uwagi, uwzględniający wymiarowość macierzy, przedstawiono poniżej:

```
   X      Wq/Wk/Wv   = Q/K/V
4x512 *   512x64    = 4x64

1) Q*Kt                      [4x64 * 64x4 = 4x4]
2) Q*Kt/sqrt(dk)             [4x4]
3) softmax[Q*Kt/sqrt(dk)]    [4x4]
4) softmax[Q*Kt/sqrt(dk)]*V  [4x4 * 4x64 = 4x64]
```

Następna sekcja zawiera przykład wykonania każdego kroku z poprzedniej listy.

PRZYKŁAD AUTOMATYCZNEJ UWAGI

Rozważmy zdanie wejściowe „I love Chicago pizza", które zawiera cztery tokeny, i oblicz osadzenia słów (tj. wektory zmiennoprzecinkowe) dla każdego tokenu. Wartości początkowe dla każdego osadzenia słowa mogą być liczbami losowymi, a poprawne wartości liczb zmiennoprzecinkowych w każdym wektorze słów są określane na etapie uczenia. Przypomnijmy, że wagi parametrów w dwóch sąsiednich warstwach sieci neuronowej można zainicjować liczbami losowymi, a wartości te są aktualizowane podczas wstecznej propagacji błędów w fazie uczenia. Załóżmy, że szerokość każdego wektora wynosi 512 i przypiszmy dowolnie osadzenia y1, y2, y3 i y4 odpowiednio do słów „I", „love", „Chicago" i „pizza" w następujący sposób

```
y1 = [5.32 7.56 . . . 0.234]
y2 = [8.65 0.58 . . . 2.085]
y3 = [0.95 3.47 . . . 6.813]
y4 = [3.91 5.72 . . . 7.055]
```

Teraz możemy utworzyć macierz X, której wiersze składają się z wektorów y1, y2, y3 i y4, jak pokazano poniżej:

```
     |5.32 7.56 . . . 0.234|
X =  |8.65 0.58 . . . 4.085|
     |0.95 3.47 . . . 6.813|
     |3.91 5.72 . . . 7.055|
```

Macierz X to macierz 4x512 liczb zmiennoprzecinkowych, a jeśli zdanie wejściowe zawiera P słów, to macierz X jest macierzą Px512. Generujemy trzy macierze z losowo zainicjowanymi wartościami, którym nadajemy nazwy Wq, Wk i Wv, po czym możemy zainicjować trzy macierze Q, K i V, które są używane w samouwadze, jak pokazano poniżej:

```
•  a query matrix Q = X*Wq
•  a key matrix   K = X*Wk
•  a value matrix V = X*Wv
```

Macierze Q, K i V są w pewnym stopniu analogiczne do macierzy W z poprzedniej sekcji: są inicjowane losowymi wartościami i aktualizowane podczas etapu uczenia. Kolejne szczegóły, na które należy zwrócić uwagę, są następujące:

- The 1st row in Q/K/V: the query/key/value vectors of the word I
- The 2nd row in Q/K/V: the query/key/value vectors of the word love
- The 3rd row in Q/K/V: the query/key/value vectors of the word Chicago
- The 4th row in Q/K/V: the query/key/value vectors of the word pizza 6.

Oblicz macierz Q*Kt

Teraz jesteśmy gotowi obliczyć macierz równą iloczynowi macierzy Q i macierzy Kt (transpozycji macierzy K). Iloczyn tych dwóch macierzy daje macierz, której wartości komórek zawierają wszystkie możliwe iloczyny skalarne (skaliste) między każdą parą słów w zdaniu oryginalnym. Przypomnijmy, że iloczyn skalarny dwóch wektorów stanowi miarę podobieństwa między nimi. Na przykład, pierwszy wiersz Q*Kt jest iloczynem skalarnym wektora zapytania „I" z wektorami kluczowymi I, love, Chicago i pizza. Drugi wiersz Q*Kt jest iloczynem skalarnym wektora zapytania „love" z wektorami kluczowymi I, love, Chicago i pizza. Trzeci wiersz Q*Kt jest iloczynem skalarnym wektora zapytania „Chicago" z wektorami kluczowymi I, love, Chicago i pizza. Czwarty wiersz Q*Kt jest iloczynem skalarnym wektora zapytania „pizza" z wektorami kluczowymi I, love, Chicago i pizza. Aby podkreślić wcześniejsze stwierdzenie dotyczące zawartości Q*Kt: macierz iloczynu zawiera iloczyn skalarny osadzenia słów dla każdego tokena (słowa) w oryginalnym zdaniu „I love Chicago pizza" przy każdym innym osadzeniu słów w tym zdaniu. Powyższe obliczenie ma złożoność O(N**2), gdzie N jest liczbą tokenów w zdaniu wejściowym. Jednakże, te obliczenia są niezależne od siebie, co oznacza, że mogą być wykonywane równolegle, co nie jest możliwe w przypadku RNN i LSTM. Ponadto istnieją warianty architektury transformatorowej, takie jak architektura Reformer, które osiągnęły złożoność niższą niż kwadratowa. Podziel przez pierwiastek kwadratowy wymiarowości wektora Wymiarowość wektora dk jest równa 64, więc jej pierwiastek kwadratowy wynosi 8. Musimy podzielić Q*Kt przez tę wartość, aby „zmniejszyć" jego wartości. Należy zauważyć, że człon dk został określony empirycznie i dobrze sprawdza się jako współczynnik skalowania. Zastosuj funkcję Softmax do Q*Kt/dk. Zastosowanie funkcji softmax() do wierszy macierzy Q*Kt/dk pozwala osiągnąć dwa cele: 1) wszystkie wartości mieszczą się w zakresie od 0 do 1 oraz 2) każdy wiersz jest dyskretnym rozkładem prawdopodobieństwa. Wartości w macierzy Q*Kt/dk stanowią względną miarę powiązania dowolnej pary słów. Załóżmy na przykład, że pierwszy wiersz zawiera następujące wartości:

[0,88 0,06 0,03 0,03]

Pamiętając, że wiersze i kolumny są oznaczone tokenami w zdaniu „Uwielbiam pizzę z Chicago", wiemy teraz, że wartości w pierwszym wierszu dostarczają następujących informacji:

0,88 to korelacja między I a I

0,06 to korelacja między I a love

0,03 to korelacja między „I" a „Chicago"

0,03 to korelacja między „I" a „pizzą"

Podobne wyniki można uzyskać z wierszy 2, 3 i 4 macierzy Q*Kt/dk.

Oblicz macierz uwagi Z

Możemy teraz zdefiniować macierz uwagi Z, która jest iloczynem dwóch macierzy:

Z = softmax [(Q*Kt/dk)] * V

Etykiety wierszy i kolumn w macierzy Q*Kt/dk i macierzy V są takie same: składają się one z tokenów w zdaniu „I love Chicago pizza”. Wartości samouwagi dla każdego słowa są równe ważonej sumie iloczynu każdego wiersza w macierzy Q*Kt/dk z wartościami w macierzy V. Mechanizm samouwagi jest również znany jako skalowany iloczyn skalarny uwagi, ponieważ jest równy skalowanym wartościom iloczynu skalarnego Q i Kt. Na przykład niech y1, y2, y3 i y4 będą osadzeniami słów dla tokenów „I”, „love”, „Chicago” i „pizza”, którym przypisano arbitralnie następujące wartości.

```
y1: [0.823, 4.432, 2.789, . . ., 7.123]
y2: [2.382, 3.432, 8.786, . . ., 6.459]
y3: [3.569, 2.432, 5.783, . . ., 4.684]
y4: [5.277, 0.426, 0.123, . . ., 9.001]
```

Chociaż wektory dla y1, y2, y3 i y4 składają się z 512 składowych, a rzeczywiste wartości będą się różnić od wartości podanych powyżej, nadal są one przydatne do celów ilustracyjnych. Następnie utwórzmy macierz z tymi samymi czterema wektorami, jak pokazano poniżej:

```
     |0.823, 4.432, 2.789, . . ., 7.123|
X =  |2.382, 3.432, 8.786, . . ., 6.459|
     |3.569, 2.432, 5.783, . . ., 4.684|
     |5.277, 0.426, 0.123, . . ., 9.001|
```

Utworzymy również trzy macierze wag: Wq, Wk i Wv, które są inicjowane losowymi wartościami aktualizowanymi w trakcie procesu uczenia, co jest nieco analogiczne do etapu uczenia dla MLP (perceptronu wielowarstwowego). Tworzymy trzy macierze: macierz zapytań Q, macierz kluczy K i macierz wartości V, mnożąc X przez ich poprzednie odpowiedniki, jak pokazano poniżej:

```
Q = X*Wq
K = X*Wk
V = X*Wv
```

Macierze Q, K i V mają wymiar 4x64, ponieważ są iloczynem macierzy X o wymiarach 4x512 i macierzy 512x64: a zatem iloczyn jest macierzą 4x64. Ponadto każdy wiersz w macierzach Q, K i V jest powiązany z odpowiadającym mu słowem z naszego zdania wejściowego „Uwielbiam pizzę z Chicago”. Jednak zawartość macierzy Q, K i V różni się od zawartości macierzy X. Iloczyn Q*Kt to macierz samouwagi o wymiarach 4x4, która zawiera podobieństwo każdego tokenu wejściowego y1, y2, y3 i y4 do wszystkich tokenów w zdaniu wejściowym, jak pokazano poniżej:

```
     y1      y2      y3      y4
y1 |0.823, 4.432, 2.789, 123|
y2 |2.382, 3.432, 8.786, 456|
y3 |3.569, 2.432, 5.783, 789|
y4 |5.277, 0.426, 0.123, 432|
```

Zatem pozycje diagonalne w poprzedniej macierzy reprezentują podobieństwo każdego tokena wejściowego do siebie. Następny krok polega na skalowaniu pozycji w poprzedniej macierzy i, zgodnie z oryginalną pracą, wartości macierzy są zmniejszane poprzez podzielenie przez pierwiastek kwadratowy z dk, który jest wymiarowością wektora kluczowego. W naszym przypadku dk wynosi 64, więc pierwiastek kwadratowy wynosi 8. Nowa macierz jest pokazana tutaj:

```
     y1      y2      y3      y4
y1 |0.104, 0.554, 0.348, 15.375|
y2 |0.297, 0.422, 1.098, 57.000|
y3 |0.446, 0.304, 0.724, 98.625|
y4 |0.659, 0.053, 0.015, 54.000|
```

UWAGA WIELOGŁOWICOWA (MHA)

Wcześniej omówiono kroki obliczania samouwagi dla każdego słowa w zdaniu wejściowym. Uwaga wielogłowicowa polega na obliczaniu wielu macierzy uwagi zamiast jednej macierzy uwagi. Pojedyncza głowica przetwarza jeden „stos" przepływu, który rozpoczyna się od wektora osadzania i przechodzi przez każdy komponent kodera (który zazwyczaj obejmuje 6 lub 12 takich komponentów). Proces tworzenia wielu macierzy uwagi obejmuje powtórzenie kroków tworzenia początkowej macierzy uwagi, co oznacza utworzenie nowego zestawu macierzy Q, K i V, które są mnożone przez nowy zestaw macierzy Wq, Wk i Wv. Oryginalny artykuł badawczy dotyczący uwagi określa 8 macierzy uwagi, co oznacza, że obliczamy macierze uwagi Z1, Z2, …, Z8. Następnym krokiem jest połączenie macierzy Z1, Z2, …, Z8 w celu utworzenia macierzy końcowej Zf, a następnie pomnożenie Zf przez połączenie zestawu macierzy W1, W2, …, W8, co daje pojedynczą macierz, którą nazywamy MHA (uwaga wielogłowicowa). Podsumowując, skalowana uwaga iloczynu skalarnego oblicza iloczyn skalarny dla każdego zapytania Q ze wszystkimi kluczami K, którego wynik jest dzielony przez dk (pierwiastek kwadratowy z wymiaru d), który następnie jest przepuszczany przez funkcję aktywacji softmax(). Wynikiem poprzedniego zdania jest macierz pomnożona przez macierz wartości V, co daje macierz będącą skalowaną wersją macierzy V. Sprawdź diagram architektury transformacji i zauważ, że istnieją trzy wejścia dla uwagi wielogłowicowej: są to wektory Q, K i V. Ponadto wzór na uwagę jednogłowicową nie uwzględnia wag dla uwagi softmax(), a w przypadku uwagi wielogłowicowej wagi pojawiają się przed wykonaniem faktycznej skalowanej uwagi iloczynu skalarnego. W rzeczywistości wagi pojawiają się w warstwach liniowych, które są umieszczone niżej na diagramie i przed obliczeniem skalowanej uwagi iloczynu skalarnego. Należy zauważyć, że iloczyny skalarne mogą być duże i prowadzić do efektu „zanikającego gradientu" (stąd wprowadzenie współczynnika skalowania dk w celu uzyskania skalowanych wartości, które unikają tego efektu). W poprzednich sekcjach pokazano, jak obliczyć wartości uwagi z macierzy Q, K i V. Możemy również obliczyć wiele macierzy uwagi, z których każda opiera się na innym zestawie wartości macierzy Q, K i V. Chociaż może się to wydawać zbędne, rozważmy analogiczną sytuację w sieciach neuronowych CNN (splotowych sieciach neuronowych), która zostanie omówiona bardziej szczegółowo w następnej sekcji. Chociaż może się to wydawać zbędne, rozważmy analogiczną operację w sieciach neuronowych CNN (splotowych sieciach neuronowych), która polega na wykonywaniu splotów wielu niezależnych małych, losowo generowanych filtrów (zwykle 3x3) z tablicą wartości pikseli dla obrazu PNG.

Filtry CNN i uwaga wielogłowicowa

Przypomnijmy, że proces splotu polega na określeniu liczby filtrów, które są macierzami kwadratowymi inicjowanymi losowymi wartościami. Następnie stosujemy splot, funkcję aktywacji ReLU i maksymalne łączenie, które można wykonać wielokrotnie. Ta sekwencja operacji jest wykonywana dla każdego filtra, a po ich zakończeniu scalamy wyniki uzyskane z każdego filtra. Scalony wynik można przekazać do jednej lub więcej w pełni połączonych warstw, a ostatni krok polega na zastosowaniu funkcji softmax() w celu późniejszego uzyskania danych wyjściowych. Rozważmy teraz ponownie uwagę wielogłowicową. Załóżmy, że chcemy obliczyć 8 wartości uwagi wielogłowicowej. Analogicznie do filtrów w sieci neuronowej (CNN), musimy obliczyć 8 macierzy uwagi {Z1, Z2, . . . , Z8}, gdzie każda z tych wartości jest obliczana w taki sam sposób, jak pokazano wcześniej. W przypadku sieci neuronowych (CNN) wymagałoby to 8 różnych filtrów, a w przypadku wartości uwagi 8 zestawów

trójek {Q, K, V}, z których wszystkie są niezależnie inicjowane losowymi wartościami. Ostatni krok polega na scalaniu macierzy w {Z1, Z2, . . . , Z8}, a następnie mnożąc przez macierz W, aby uzyskać ostateczną macierz uwagi, jak pokazano tutaj:

MHA = konkatenacja(Z1, Z2, . . ., Z8) * W

PODSUMOWANIE

Ta część rozpoczęła się od krótkiego opisu kilku firm, takich jak DeepMind, OpenAI i Hugging Face, które są liderami w dziedzinie sztucznej inteligencji. Następnie zapoznałeś się z modelami LLM (modelami dużego języka) oraz zależnością między rozmiarem modelu a rozmiarem zbioru treningowego. Poznałeś mechanizm uwagi, który jest niezwykle ważnym elementem architektury Transformer , oraz jego przydatność w uzyskiwaniu informacji kontekstowych z tekstu. Poznałeś również samouwagę i niektóre szczegóły dotyczące obliczania uwagi. Na koniec zapoznałeś się z wykorzystaniem funkcji softmax() w połączeniu z mechanizmem uwagi w architekturze Transformer.