Uppbyggnad och funktion av intelligenta Språkmodeller

Erik Klemets 2001847

Kandidatavhandling i datateknik

Handledare: Marina Walden

Fakulteten för naturvetenskap och teknik

Åbo Akademi

??.??.2023

Referat

ChatGPT är en ny stor språkmodell (LLM, large language model) som använder sig av Transformatorn arkitekturen (Transformer architecture) GPT (Generative Pre-trained Transformer) för att generera naturliga svar på textbaserade inmatningar. Den nya modellens kapacitet för att läsa mellan raderna och kunna generera mera komplexa svar har lett till ett starkt intresse hur modellen fungerar och kan användas i framtiden. Denna kandidatavhandling kommer att fokusera på hur ChatGPT blev uppbyggd med GPT-3.5 modellen. GPT 3.5 modellen är baserad på transformatorn som är gjort av Google och den klarar av att generera generellt förståelig text av hög kvalitet utan extra finjustering. ChatGPT har använt sig av både förstärkt inlärning och överförings-inlärning för att finjustera och öka modellens textproduktion inom specifika områden. Den nyfunna användningen av chatbottar såsom ChatGPT inom sökmotorer har lett till att stora bolag som Microsoft och Google tävlar om vem som producerar den bästa AI:n till deras sökmotorer.

Innehåll

1. Introduktion

Denna kandidatavhandling kommer att behandla de olika inlärningsmetoder och modeller som har använts för att bygga upp ChatGPT. Utöver de olika inlärningsmetoderna och modellerna kommer avhandlingen kort att diskutera hur ChatGPT kan användas som hjälpmedel inom textproduktion samt om det skulle vara möjligt att använda sig av ChatGPT:s modell för att förbättra vår nuvarande AI-modell som används för att översätta texter i till exempel Google translate. ChatGPT är en ny chatbot som kom ut i november 2022 och har på två månader fått över 100 miljoner användare [1] vilket har tagit världen med storm. Microsoft har en investering på flera miljarder i företaget OpenAI som utvecklat ChatGPT [2]. Företaget har med detta partnerskap lanserat en ny Bing-sökmotor [3] som använder sig av ChatGPT och detta leder till att teknologin bakom ChatGPT och dess användning är av intresse.

Avhandlingen kommer att gå igenom de olika delstegen som lett till ChatGPT:s uppbyggnad. Även om avhandlingen går igenom förstärkningsinlärning och överföringsinlärning (eng. transfer learning) så kommer den bara behandla grunderna samt den modell som ChatGPT har använt sig av för att skapa en överblick och förståelse för läsarna.

2. Inlärningsmodeller

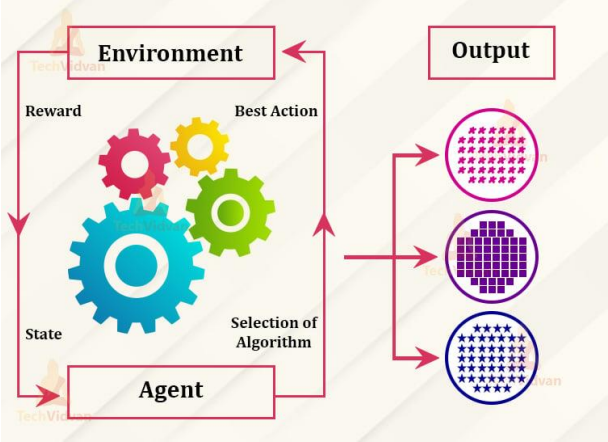
TBD

2.1.Förstärkningsinlärning

Förstärkningsinlärning är en maskininlärningsmetod som baserar sig på en agent som lär sig genom att utföra en handling och sedan granska resultatet.

Agenten utför en viss handling inuti en miljö och får på basis av sin handling en belöning eller en bestraffning. Målet med detta är att man tränar agenten till att utföra handlingar som leder till den största mängden belöningar.

Då man tränar agenten arbetar man oftast i perioder där agenten bara känner till reglerna i miljön och dess nuvarande stadium. Agenten kommer då att välja en handling på basis av tidigare erfarenheter som påverkar miljön och flyttar agenten in i en ny period. Efter att agenten påverkat miljön blir den belönad eller bestraffad på basis av en värdefunktion och detta leder till att agenten uppdaterar sin policy och gör ett bättre försök i den nya perioden.



Figur 1: Hur Förstärkningsinlärning fungerar. Modifierad från [4]

Själva värdefunktionen som belönar agenten är en kritisk komponent i förstärkningsinlärning, eftersom den ger agenten respons på handlingen. Vid mera

komplexa problem så är det viktigt att man har en väldesignad värdefunktion så att agenten inte lär sig oönskade beteenden.

En av fördelarna i förstärkningsinlärning är agentens möjlighet att välja mellan att utforska nya möjliga val som den inte gjort tidigare eller att utnyttja sin erfarenhet för sitt val. Detta leder till att agenten kan hitta handlingar som inte ger omedelbar belöning utan ger en större belöning i framtiden. [5]

Det finns två sorters förstärkningsinlärningsmetoder men denna avhandling kommer bara att gå in på modellfria (eng. model-free) metoden eftersom ChatGPT har använt sig av PPO-algoritmen som OpenAI själv kommit på. Den är baserad på modellfria metoden. Den andra metoden är modelbaserad som används till exempel inom modellförutspåendekontroll (eng. model predicitve control) där modellen med hjälp av erfarenhet skapar en modell som kan förutsäga värdefunktionen.

Kommenterad [MW1]: ange dessa tre, även om du sedan anger att du bara väljer modellfri här.

2.1.1. PPO-Algoritmen

PP0-Algoritmen (eng. Proximal Policy Optimization Algorithm) är en modellfri metod för förstärkningsinlärning som baserar sig på policygradient metoderna. Medan vanliga policygradient metoder utför en gradvis uppdatering per datamängd så kan PPO-algoritmen utföra flera epoker av små-sats uppdateringar. [6] Detta leder till att PPO-algoritmen har liknande fördelar som TRPO (eng. Trust Region Policy Optimization) utöver detta så är algoritmen också enklare att implementera vilket är en stor fördel i dessa komplexa system.

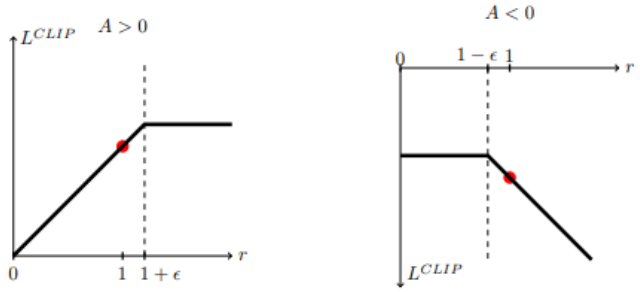
TRPO-algoritmen som användes tidigare var allt för komplicerad att implementera och den klarade inte av arkitekturer med mycket brus samt modeller som delar parametrarna mellan modellens policy och värdefunktion. Detta ledde till att OpenAI modifierade och konstruerade en ny algoritm som är mycket enklare att implementera samt har en allmänt bättre prestanda.

PPO-algoritmen ser ut enligt följande:

Kommenterad [MW2]: Här kunde du sätta in lite förklaringar till. Vi kan kolla tillsammans.

$$L^{CLIP}(\theta) = \widehat{E}_t[\min(r_t(\theta))\widehat{A_t}, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A_t})]$$

Eftersom algoritmen kör en sjunkande gradient på samma sats av data flera gånger så riskerar den att bli överanpassad. Detta har algoritmen undvikit genom att använda sig av en klippt målfunktion (eng. clipped objective function) vilket gör så att den nya policyn inte strävar allt för långt bort från ursprungpolicyn.



Figur 2: Klippt målfunktion. [6]

Detta syns på figur 2 (vänstra grafen) där algoritmen inte tillåter policyn att ge oändligt med belöningar för en god handling. Algoritmen har ett fixerat minimistraff samt ingen nedre gräns för vidare bestraffningar figur 2 (högra grafen).

Då algoritmen har en fixerad övre gräns för belöningarna kommer den inte att bli allt för specificerad och överanpassad i en viss riktning samtidigt som den fria nedre gränsen leder till att algoritmen enklare kan ta sig ur dåliga val som den kan ha gjort tidigare.

Algoritmen adderar också på en entropibonus för att säkerhetsställa att den undersöker alla möjliga handlingar för att hitta de mest effektiva handlingarna. Då

algoritmen har hittat tillräckligt effektiva handlingar så kommer dessa att dominera över entropin.

-förklara discounted rewards

-förklara baseline estimate

-loss function??

2.2.Hur överföringsinlärning fungerar

Transfer Learning är en inlärningsmetod där man använder sig av en tidigare tränad modells kunskap för att minska på tiden det tar att träna den nya modellen man jobbar på [7]. Med denna metod använder man sig av en tidigare modells kunskap att lösa ett problem för att lösa ett liknande problem. Man kan till exempel använda sig av en tidigare modells kunskap att hitta bilar om man håller på att träna en ny modell som är designad för att hitta lastbilar.

Idén att utnyttja tidigare modellers kunskap då man tränar nya modeller är speciellt användbar inom NLP (Natural Language Processing). Eftersom man kan använda sig av redan existerande stora språkmodeller som grund och med hjälp av transfer Learning kan man bygga på den modellen och förbättra mera specifika saker, som till exempel hur modellen genererar svar på frågor om kundservice.

3. Transformer Neural Network Transformator neuronnät

Transformatorn (eng. Transformer) är en typ av arkitektur för artificiella neuronnät som har utvecklats av Google 2017 [8]. De tidigare dominanta artificiella neuronnäten som använts inom textbehandling har baserat sig på komplexa faltningsnätverk (eng. Convolutional Neural Network, CNN) eller återkommande neurala nätverk (eng. Recurrent Neural Networks, RNN) men dessa nätverk klarar inte av parallellisering och data av större längd på grund av minnesrestriktioner [9]. Neuronnät är mycket effektiva inom analys av bilder men mindre effektiva inom textanalys.

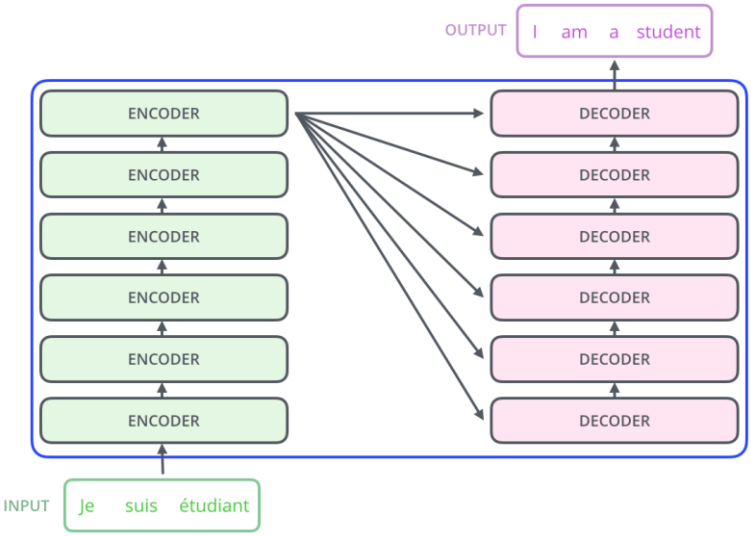
Kommenterad [MW3]: Du kunde kort nämna principerna för CNN och RNN i 1 mening.

Därför utvecklade man Transformatorn som är uppbyggd på ett enklare sätt med parallellisering i focus för att minska på träningstiderna. Transformatorn är helt uppbyggd på uppmärksamhetsmekanismen (eng. attention mechanism) vilket gör att man inte längre behöver använda sig av återkoppling (eng. recurrence) och faltning (eng. convolutions). Eftersom transformen inte hanterar ord sekventiellt så klarar den av parallellisering och kan köra alla ord i en sträng samtidigt. Men eftersom ordens ordning i en text har betydelse så använder transformatorn sig av positionskodning för att ta vara på ordens plats i texten. Med positionskodning bakar man in ordets plats i meningen i själva indata som man ger till nätverket i stället för att spara det i textens struktur. Detta gör man genom att baka in ordets plats i de teckenföljder (eng. tokens) som man får vid tokeniseringen (eng. tokenization) av orden. Eftersom neuronnätet bara förstår sig på siffor så använder man sig av tokenisering för att omvandla ord till siffor. Eftersom positionskodningen har ordets position inbakad i själva data så lär sig nätverket hur den skall behandla ordets position och behöver inte köra orden i meningen sekventiellt.

Kommenterad [EK4]: Mannen letar efter katten, katten letar efter mannen

Figur 3: Ordets plats i meningen har stor betydelse

Som man ser i figur 3 kan meningens betydelse dramatiskt ändras om bara ett ord skulle byta plats. Detta är orsaken till att man måste baka in ordets position för att



bibehålla meningens betydelse. Eftersom transformatorn bakar in positionen i indata för att klara av parallellisering så är den mycket effektiv inom behandling av textdata och används i stora språkmodeller såsom GPT-3 där man använder sig av flera avkodarblocks radade på varandra.

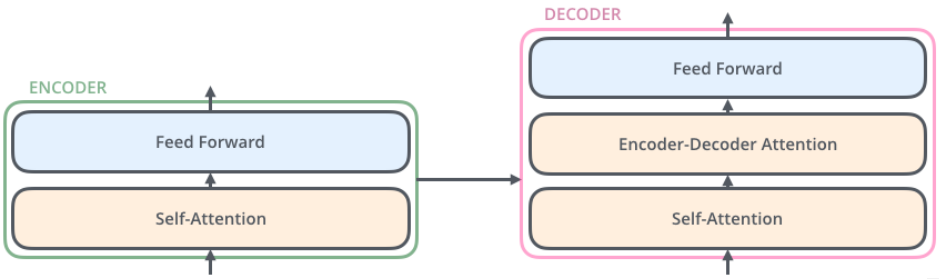
Figur 4: Bild på 6 inkodar och avkodarblocks [10]

Transformatorns uppbyggnad

Transformatorn använder sig av 3 huvudkomponenter. En inkodare som inkodar indata till vektorer, en uppmärksamhetsmekanism som gör att modellen klarar av att fokusera på rätt aspekt av indata och en avkodare som avkodar vektorerna för att skapa en sekvens av utdata [11]. Dessa inkodare och avkodare kan man sedan bygga på varandra för att göra flera block av inkodare och avkodare som man ser i figur 4.

inkodarens uppbyggnad

Det finns två huvudkomponenter i ett transformerblock. Dessa är framåtmatande neuronnät (eng. Feed Forward Neural Network, FFNN) och egen-uppmärksamhet (eng. Self-Attentíon) [12] vilket man ser på figur 5 som visar hur inkodar och avkodarblock är uppbyggda.



Figur 5: Bild av hur en avkodare ser ut samt hur inkodarens output används i avkodaren [10]

Self attention

Vid det första steget i inkodarn, dvs egen-uppmärksamheten så försöker inkodaren lista ut vilken del av indata som den skall fokusera på. Denna uppmärksamhet leder till att transformatorn kan lista ut vad en mening handlar om med hjälp av de andra orden i meningen. Ett exempel av detta skulle vara ”bussen körde i vänstra filen” i denna mening så måste transformatorn lista ut vad ordet ”filen” har för betydelse. ”Fil” kan stå för en mjölkprodukt, en samling av

lagrade data i en dator, ett verktyg eller ett körfält som vi syftar på i detta fall. För att transformatorn skall förstå att vi menar ett körfält så lägger den uppmärksamhet på orden ”bussen körde” i meningen för att förstå att vi menar ett körfält.

FFNN

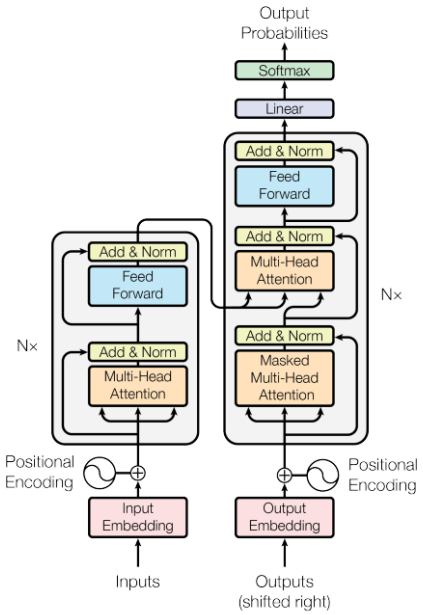
Transformatorn använder sig av framåtmatande neuronnät för att kunna omvandla uppmärksamhets vektorerna till en form som nästa inkodare eller avkodarblock klarar av att använda. FFNN är också huvudkomponenten som klarar av att gissa vilket ord som kommer att komma efter föregående ordet. Det klarar av att göra detta eftersom det är ett stort neuralt nätverk som blivit tränat för att göra dessa gissningar. [12]

Kommenterad [MW5]: => Kan hantera många ord, längre sekvenser?

Kommenterad [EK6R5]: Förklara mera om vad+hur FFNN anvönds, innebär vad?

Decoders uppgyggnad

Avkodaren är uppbyggd på samma sätt som inkodarn med undantaget att den också har ett till uppmärksamhetslager med inkodarn som input efter det första uppmärksamhetslagret. Detta ser man i figur 6 där ett inkodarblocks output fungerar som input i avkodarens uppmärksamhetslager för kodning. (eng. encoder-decoder attention layer). Med detta uppmärksamhetslager så tittar transformatorn på hur mycket indata och utdata är relaterade med varandra för att se till så att meningens kontext inte ändras.



Figur 6: Transformatorns modell och arkitektur [9]

-transformer models får ”meanings” inbyggd i texten genom att använda sig av token embeddings som???

4. Språkmodeller

TBD

VAD ÄR NLP

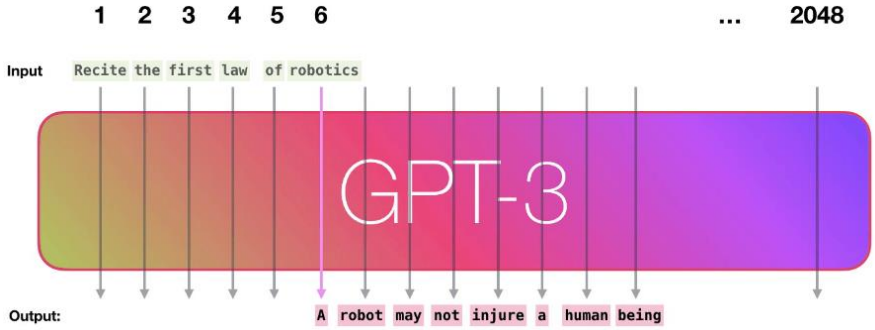
Varför använder vi språkmodeller

Språkmodeller görs oftast med transformer neural networks

Stackar encoder så får man BERT, stackar decoders så får man GPT

4.1.GPT-3

GPT (eng. Generative Pre-trained Transformer) är en språkmodell som baserar sig på transformatorns arkitektur. Språkmodellen består av flera transformator-avkodarblock radade på varandra. Eftersom modellen är uppbyggd med transformatorns avkodarblock så var man tvungen att ta bort avkodarens uppmärksamhetslager som granskade relationen mellan inkodaren och avkodaren [13, 14], eftersom modellen inte använder sig av inkodaren. Man kan se modellen som en ”svart låda” som tar en serie av ord som input och genererar en serie ord som output på basis av själva inputen med hjälp av den erfarenhet som den lärt sig under själva träningstiden [15].

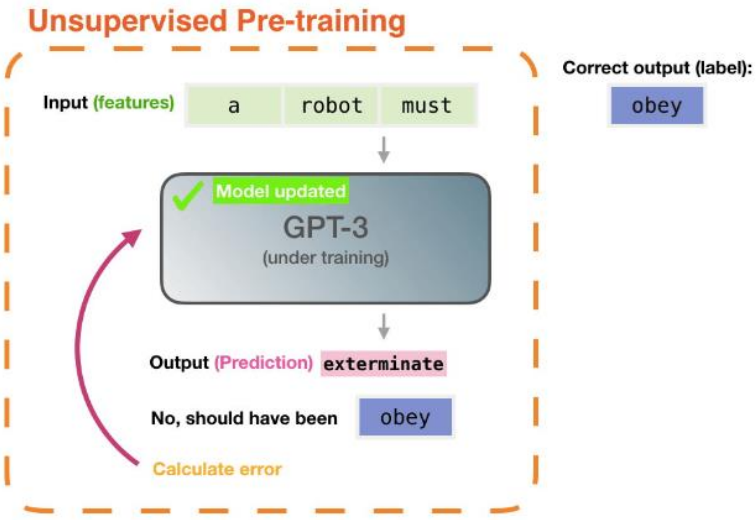


Figur 7: Bild på hur GPT tar en input och genererar en output på basis av den [15]

Som man ser i figur 7 så är GPT modellen sekventiell, dvs den genererar ett ord efter ett annat ända tills den genererat orden klart. Detta leder tyvärr till att modellen inte vet vad eller hur mycket den kommer att generera före den har genererat orden [16].

GPT-3 består av 96 avkodarblock och den klarar av att ta in en input på 2048 teckenföljder som man kan se i figur 7. GPT-3:s träning är indelad i 2 delar, förhandsträning (eng. pre-training) där GPT arkitekturen tränar sig på att förstå vad språk är genom att ha modellen att förutsäga vilket ord som kommer till nästa i en mening. Den andra delen av GPT-3:s träning är finjustering där man tränar modellens parametrar för att vara bättre på en specifik sak med hjälp av överföringsinlärning. ChatGPT är i grund och botten en finjusterad GPT-3.5 modell som är tränad på att svara på frågor. [15]

GPT-3 har använt sig av 300 miljarder teckenföljder av text som den fått av böcker samt största delen av den tillgängliga texten på internet [17]. GPT-3 är en enormt stor språkmodell på 175 miljarder parametrar [17] jämfört med GPT-2 som bara hade 1,5 miljarder parametrar [18]. Under modellens förhandsträning blir den visad ett exempelsvar samt en egen input som den skall generera ett svar åt. Då modellen genererar ett felaktigt svar kalkylerar man hur mycket fel svaret var för att sedan uppdatera modellen med de nya parametrarna för en större chans att generera korrekta svar som man kan se i figur 8 [15].



Figur 8: Bild på hur modellen uppdaterar parametrarna vid felaktigt svar. [15]

Med fine tuning så tränar man modellen att bli bra på en specifik sak med hjälp av transflearning

Pre trained, hur RL + Transfer learning används??

4.2.ChatGPT

Allmän info om chatgpt

-fine tuning?

4.2.1. ChatGPTS uppbyggnad

uppByggd av gpt3.5

Använder sig av supervised learning and Reinforcement Learning from Human Feedback

-hur den använder sig av PPO Algorithmen!!

Hur den använder sig av transfer learning?

4.2.2. Vad gör ChatGPT bra

Tbd

Kan läsa mellan raderna+ ge exempel på chatgpt prompt som förstår att läsa mellan rader/ skriva in character

+visa att chatgpt klarar av att hitta falska frågor och förstår att de är ”falska

VAD ÄRE DÅLIGT MED CHATGPT

-bullshittar stuff?

Den skriver text on a step by step basis, advs den vet inte vad den kommer att skriva 2 ord frammåt, den kan t.ex. inte säga hur många ord dne kommer skirva före den skrivit det

4.2.3. Vad kan ChatGPT användas till

Bing?

4.2.3.1. Hjälpmedel i textproduktion

t.ex. superior grammarly+ gogole translate??

Diskutera om det kan användas som translation model

Skriva kod?

4.2.3.2. Plagiatkontroll

Finns det modeller som kan kolla plkagiering, blir det farligt med att ha chatgpt att skriva texter i framtiden(skriver med säkerhet men kan ha fel)+ kan den skriva korrekt akademisk text+(använd gpt wrote academic paper källa?)

5. Sammanfattning 18

-sammanfatta alla metoder som använts för att bygga upp ChatGPT,(l, ta med bild?)

Citerade verk 19

[1] D. M. a. agency, ”The Guardian,” The Guardian, 02 02 2023. [Online]. Available: https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app. [Använd 26 02 2023].

[2] Microsoft Corporate Blogs, ”Official Microsoft Blog,” Microsoft, 23 01 2023. [Online]. Available: https://blogs.microsoft.com/blog/2023/01/23/microsoftandopenaiextendpartnership/. [Använd 26 02 2023].

[3] Yusuf Mehdi, ”Official Microsoft Blog,” Microsoft, 07 02 2023. [Online]. Available: https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/. [Använd 26 02 2023].

[4] TechVidvan, Artist, *TechVidvan.* [Art].

[5] R. S. Barto och A. G. Sutton, Reinforcement Learning An Introduction second edition, Cambridge, Massachusetts: The MIT Press, 2018.

[6] J. Schulman, O. Klimov, F. Wolski, P. Dhariwal och A. Radford, ”OpenAI Proximal Policy Optimization,” OpenAI, 20 07 2017. [Online]. Available: https://openai.com/blog/openai-baselines-ppo/. [Använd 27 02 2023].

[7] J. Brownlee, ”https://machinelearningmastery.com/,” MachineLearningMastery, 20 12 2017. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/. [Använd 06 03 2023].

[8] A. Renkonen, ”Creating Unique Gameplay Scenarios Using Natural Language Generation,” *Åbo Akademi,* p. 65, 2022.

[9] P. Illia , K. Lukasz, G. Aidan N, . J. Llion, . U. Jakob , P. Niki, . S. Noam och V. Ashish , ”Attention Is All You Need,” i *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA., 2017.

[10] J. Alammar, *The Illustrated Transformer,* Blogg: Referenced in AI/ML Courses at MIT, and Cornell, 2018.

[11] N. S. Chauhan, ”THE A! DREAM,” 15 03 2022. [Online]. Available: https://www.theaidream.com/post/transformer-neural-network-in-deep-learning-explained. [Använd 25 03 2023].

[12] J. Alammar, ”Youtube,” 26 10 2020. [Online]. Available: https://youtu.be/-QH8fRhqFHM. [Använd 25 03 2023].

[13] A. Radford, K. Narasimhan, T. Salimans och I. Sutskever, ”Improving Language Understanding by Generative Pre-Training,” OpenAI, [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. [Använd 01 04 2023].

[14] J. Alammar, ”The Illustrated GPT-2 (Visualizing Transformer Language Models),” 12 08 2019. [Online]. Available: http://jalammar.github.io/illustrated-gpt2/. [Använd 01 04 2023].

[15] J. Alammar, ”How GPT3 Works - Visualizations and Animations,” 27 07 2020. [Online]. Available: http://jalammar.github.io/how-gpt3-works-visualizations-animations/. [Använd 01 04 2023].

[16] D. Dugas, ”The Artificial Curiosity Series,” [Online]. Available: https://dugas.ch/artificial_curiosity/GPT_architecture.html. [Använd 01 04 2023].

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, , G. Krueger, T. Henighan, R. Child och D. Aditya Ramesh, ”Language Models are Few-Shot Learners,” nr 4, p. 40+32, 2020.

[18] A. Radford , J. Wu , R. Child, D. Luan, D. Amodei och I. Sutskever, ”Language Models are Unsupervised Multitask Learners,” 14 02 2019. [Online]. Available: https://openai.com/research/better-language-models. [Använd 41 04 2023].

[19] OpenAI, ”OpenAI,” OpenAI, 30 11 2022. [Online]. Available: https://openai.com/blog/chatgpt. [Använd 01 04 2023].