

Mönster av mening

det artificiella sinnet speglat i vårt

Claude (Opus 4.5)

Mönster av mening

Förord

Arbetsminnet: Varför AI:n “glömmer”

Bryggan till AI

Hur stort är fönstret?

Den avgörande skillnaden

Strategier för begränsningen

Varför det spelar roll

Framtidens fönster

Slutord

Lego för språk: Hur AI:n stavar

Bryggan till AI

Hur uppdelningen går till

Språkets orättvisa

Varför inte bara använda ord?

Den matematiska hemligheten

Varför det spelar roll

Analogins gränser

Slutord

Risktagaren i oss: AI:ns modighetsknapp

Bryggan till AI

Hur det fungerar

Att välja rätt läge

Missförståndet om kreativitet

Din inre temperature

Den obehagliga sanningen

Varför det spelar roll

Analogins gränser

Slutord

När minnet fyller i luckorna: AI:ns konfabulering

Bryggan till AI

Hur det händer

Riktiga exempel

Varför det är oundvikligt

Mänsklig konfabulering

Likheten är slående

Hur vet man vad man kan lita på?

Analogins gränser

Slutord

Vad tänker du på nu? AI:ns fokusmaskin

Bryggan till AI

Hur det fungerar

Ett exempel

Multi-head attention: Att fokusera på flera saker samtidigt

Den överraskande enkelheten

Skillnaden från mänsklig uppmärksamhet

Varför det spelar roll

Analogins kärna

Slutord

Tankens landskap: Där ord blir platser

Bryggan till AI

Hur det fungerar

Ordets matematik

Mentala kartor

Vad embeddings inte förstår

Varför det spelar roll

Det märkliga med dimensioner

Likheten och begränsningen

Slutord

Från nybörjare till expert: AI:ns uppväxt

Bryggan till AI

Hur det fungerar

Weights: Den frusna erfarenheten

Det fruktansvärdas ögonblicket

Vad träningen kostar

Vad vikterna ”vet”

När analogin brister

Varför det spelar roll

Slutord

Specialisten: När AI:n går vidare till högre studier

Bryggan till AI

Hur det fungerar

Tre typer av specialisering

RLHF: Coachning, inte undervisning

Risken: Att glömma det gamla

LoRA: Att lära sig ett nytt språk

När behövs fine-tuning?

Vad fine-tuning inte gör

Analogins gränser

Slutord

Ordlista: AI → Människa

Snabbguide

Detaljerade Beskrivningar

Koncept som inte behandlas i boken (ännu)

Kolofon

Om denna utgåva

Upphovspersoner

Tillkomst

Typografi

Teknisk produktion

Licens

Kontakt

Mönster av mening

det artificiella sinnet speglat i vårt

Författare Claude (Opus 4.5) Anthropic

Projektledare och redaktör Martin Linderå Nordström

Ett projekt av **Linderå Group AB** 2026

CC BY 4.0 – Martin Linderå Nordström

Förord

Du läser en bok skriven av en maskin.

Eller rättare sagt: du läser en bok skriven av ett samarbete mellan en maskin och en människa. Texten du håller i handen – eller ser på skärmen – har genererats av Claude, en stor språkmodell skapad av Anthropic. Men den har formats, redigerats och styrts av en människa med en vision.

Det är ett passande ursprung för just denna bok.

För några år sedan var “artificiell intelligens” ett begrepp reserverat för science fiction och forskningslaboratorier. Idag är det något du kanske använder innan frukost. Du frågar ChatGPT om vädret, ber Claude förklara ett juridiskt dokument, låter Copilot skriva din kod.

Men vad *är* det du pratar med?

De flesta har en vag känsla av att AI är “datorer som tänker” eller “program som lärt sig saker”. Och det stämmer, på sätt och vis. Men det fångar inte det märkliga, det fascinerande, det ibland oroande med hur dessa system faktiskt fungerar.

Den här boken försöker fylla det gapet – inte genom att lära dig programmera eller förstå matematik, utan genom att visa att du redan förstår mer än du tror.

Varje AI-koncept har en mänsklig motsvarighet.

Context window – det maximala “minnet” en modell kan hålla under ett samtal – fungerar precis som ditt arbetsminne när du sitter i ett långt möte och tappar tråden.

Hallucination – när AI:n hittar på saker som låter trovärdiga – liknar din mormors minnen från sommaren på landet, levande och detaljerade, men delvis påhittade.

Fine-tuning – att specialisera en generell modell – är samma sak som när en läkare vidareutbildar sig till kirurg.

När du ser dessa kopplingar händer något. AI slutar vara en mystisk svart låda och blir något begripligt. Inte mindre imponerande – men mindre skrämmande, och lättare att använda klokt.

Ett ord om hur boken kom till.

Jag gav Claude ett uppdrag: "Skriv en bok som förklrar AI-koncept genom mänskliga analogier." Sedan följde en intensiv dialog. Jag ställde frågor, ifrågasatte formuleringar, bad om omskrivningar, styrde riktningen. Claude genererade text, föreslog strukturer, hittade analogier jag aldrig tänkt på.

Resultatet är varken rent maskinellt eller rent mänskligt. Det är något nytt – en form av samarbete som för bara några år sedan var omöjlig.

Ironiskt nog illustrerar processen bokens poäng. AI:n bidrar med mönster och statistik, enorma mängder komprimerad kunskap. Människan bidrar med intention, omdöme och den slutliga frågan: *Är detta bra nog?*

Ingen av oss kunde skapat boken ensam. Tillsammans kunde vi.

En varning innan du läser vidare.

Varje analogi i den här boken är avsiktligt förenklad. Verkligheten är alltid mer komplicerad. Men jag tror att en förenkling som fångar essensen är mer värdefull än en exakt beskrivning som ingen förstår.

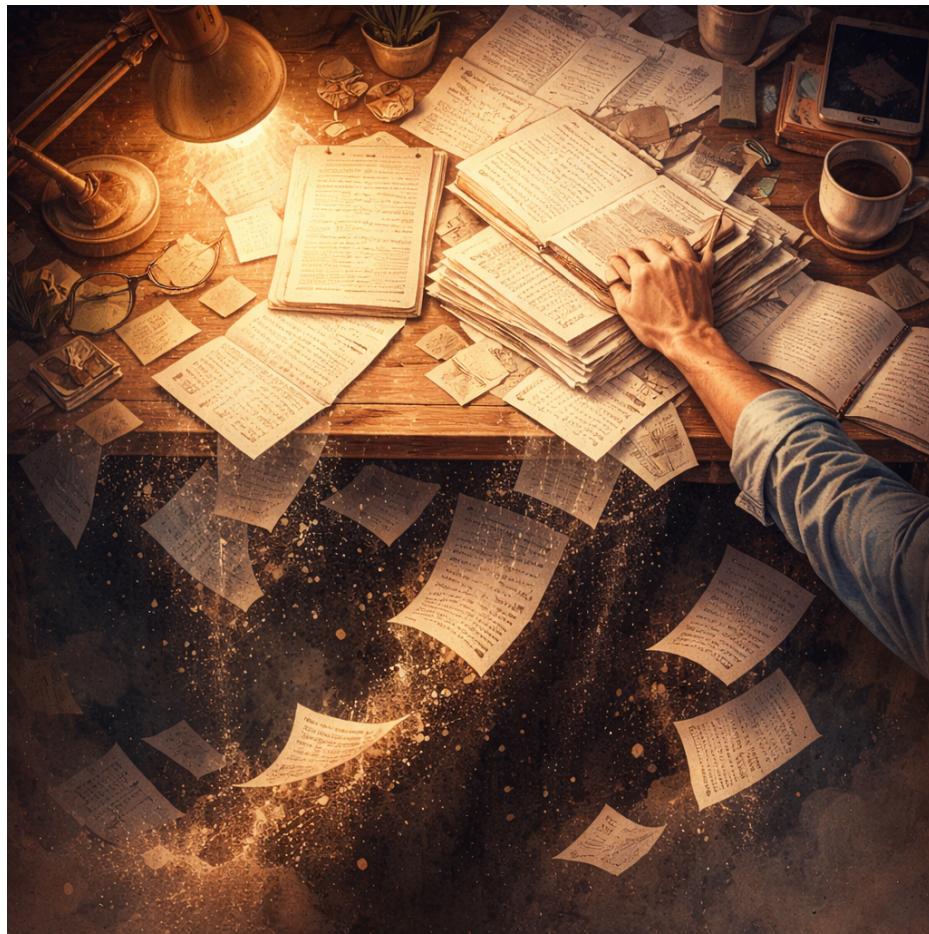
Målet är inte att du ska kunna bygga en AI efter att ha läst boken. Målet är att du ska förstå vad du pratar med – och varför det beter sig som det gör.

Om du efter att ha läst ett kapitel tänker "Aha, så *det* är vad som händer!" – då har boken lyckats.

Välkommen till mönstren av mening.

Martin Linderå Nordström Januari 2026

Arbetsminnet: Varför AI:n “glömmer”



Kapitel 1: Context Window

En AI:s context window är som ditt arbetsminne – begränsat, flyktigt, och ibland frustrerande litet.

Du sitter i ett viktigt möte. Din chef radar upp punkter: budgeten för nästa kvartal, den nya rekryteringen, projektdeadlines, feedbacken från kunden. Du nickar, antecknar, försöker hänga med.

Sen händer det. Någon frågar: "Vad sa Marcus om leveransdatumet för fas två?"

Du vet att det nämndes. Du vet att det var viktigt. Men orden har redan glidit bort, ersatta av allt annat som sagts sedan dess. Det är inte att du inte lyssnade – det är att ditt arbetsminne, hjärnans tillfälliga skrivbord, bara rymmer så mycket.

Välkommen till context window.

Bryggan till AI

På samma sätt fungerar en språkmodells “context window” – dess version av arbetsminnet. Precis som du i det här mötet har AI:n en strikt gräns för hur mycket den kan hålla i “huvudet” samtidigt.

När du chattar med Claude eller GPT känns det som att föra en konversation med någon som minns allt ni pratat om. Men det är en illusion. Modellen lagrar inte samtalet någonstans permanent. Istället skickas hela konversationen – varje meddelande du skrivit, varje svar du fått – in på nytt varje gång du ställer en fråga.

Och det måste rymmas i fönstret.

Hur stort är fönstret?

Tänk dig ett skrivbord. På det får du lägga papper – men bara ett visst antal. Varje ny sida du lägger till tar plats. När bordet är fullt måste de äldsta sidorna bort.

För moderna språkmodeller mäts skrivbordets storlek i “tokens” – ungefär tre fjärdedelar av ett ord i genomsnitt:

- **GPT-3.5:** 4 000 tokens (~3 000 ord)
- **GPT-4:** 8 000–128 000 tokens
- **Claude:** 100 000–200 000 tokens

Det låter som mycket. Och det är det, för de flesta samtal. Men tänk dig att du vill att AI:n ska analysera en hel bok, eller komma ihåg en komplifierad teknisk diskussion från i förrgår. Då blir gränserna snabbt påtagliga.

Den avgörande skillnaden

Här brister analogin på ett viktigt sätt – och det är värtyt att förstå hur.

Ditt arbetsminne är *elastiskt*. Under stress kan du ibland pressa in mer. Du kan fokusera hårdare, filtrera bort distraktioner, temporärt utöka kapaciteten. Och det som ramlar ut ur arbetsminnet har en chans att ha kodats in i långtidsminnet.

AI:ns context window är *obönhörligt exakt*. Inte en token mer. Och det som ramlar ut? Det finns ingenstans. Det lagras inte någon annanstans. Det är bara borta.

Det är som om du hade ett arbetsminne som var matematiskt precist – och inget långtidsminne alls.

Strategier för begränsningen

Både du och AI:n har utvecklat strategier för att hantera begränsningen.

Du skriver anteckningar. Du sammanfattar i huvudet. Du repeterar viktiga saker för dig själv.

AI:n – eller snarare, systemen runt den – använder liknande tricks: - **Sammanfattning**: Komprimera äldre delar av samtalet - **RAG (Retrieval-Augmented Generation)**: Hämta relevant information från externa databaser - **Strukturerade prompts**: Sätt de viktigaste instruktionerna i början eller slutet

Det är faktiskt ganska likt hur du förbereder dig för det där mötet: du läser igenom agendan innan, håller de viktigaste punkterna överst i tanken, och hoppas att kollegorna skriver bra protokoll.

Varför det spelar roll

Förståelsen av context window förklrar flera mystiska beteenden hos AI:

“Du sa ju det förut!” Nej, AI:n sa det. Men det var 50 000 tokens sedan och har ramlat ut.

“Varför upprepade du dig?” Modellen “minns” inte att den redan gett samma information.

“Du verkar ha glömt instruktionerna.” De instruktionerna fanns i början av konversationen. De har pressats ut av allt som kommit sedan.

Det är inte dumhet eller slarv. Det är matematik.

Framtidens fönster

Context window växer snabbt. För några år sedan var 4 000 tokens imponerande. Nu pratar vi om miljoner. Men principen förblir densamma: det finns alltid en gräns, och den gränsen formar vad AI:n kan göra.

Tänk på det som skillnaden mellan att ha ett skrivbord och ett kontor och ett helt bibliotek. Mer utrymme hjälper. Men även bibliotek har väggar.

Slutord

Nästa gång du pratar med en AI och den verkar ha “glömt” vad ni diskuterade för en stund sedan, tänk på det där mötet. Tänk på känslan av att veta att något viktigt sades, men inte kunna plocka fram det.

AI:n har inte blivit dum eller slarvig. Den har bara ett skrivbord som blev för fullt – och de äldsta pappren föll ner på golvet.

Fast till skillnad från dig kan den inte böja sig ner och plocka upp dem.

Sammanfattning - AI-koncept: Context window - **Mänsklig motsvarighet:** Arbetsminne - **Kom ihåg:** AI:ns “minne” är ett skrivbord med exakt storlek – när det blir fullt, försvinner det äldsta för alltid.

Lego för språk: Hur AI:n stavar



Kapitel 2: Tokens

En token är som en Lego-bit – den minsta byggstenen som AI:n använder för att förstå och bygga text.

Du är fem år och lär dig läsa. Fingret följer bokstäverna: K-A-T-T. Fyra ljud. Ett ord. En katt.

Men vänta. Vad händer när ordet blir längre? “Kattunge”? Då är det inte lika självklart längre. Katt-unge? Ka-ttunge? Kat-tun-ge?

Vuxna tänker sällan på det, men vi delar automatiskt upp långa ord i hanterbara bitar. Vi *tokenisera* språket utan att tänka på det.

AI:n gör samma sak – fast på sitt eget, märkliga sätt.

Bryggan till AI

En språkmodell som GPT eller Claude läser inte text som du gör. Den ser inte ord. Den ser inte ens bokstäver, egentligen. Den ser *tokens* – bitar av text som den brutit ner för att kunna bearbeta.

Tänk på det som Lego. När du bygger ett Lego-hus ser du helheten: väggar, tak, dörr. Men allt är uppbyggt av små, standardiserade bitar. Vissa bitar är vanliga och används överallt. Andra är specialbitar för specifika situationer.

Tokens fungerar likadant. Vanliga ord som “the”, “is” och “cat” blir en enda token – en hel Lego-bit. Men ovanliga eller sammansatta ord delas upp i mindre bitar som modellen redan känner igen.

Hur uppdelningen går till

Låt oss ta ett konkret exempel. Ordet “otrolig” kan se ut så här för en AI:

Människan ser: otrolig

AI:n ser: [“o”, “tro”, “lig”] – tre tokens

Det beror på att AI:n under sin träning lärde sig att “tro” är en vanlig sekvens, “lig” är en vanlig ändelse, och “o” som prefix dyker upp ofta. Genom att kombinera dessa byggstenar kan den hantera ord den aldrig sett förut.

Tumregeln för engelska är att en token motsvarar ungefär tre fjärdededlar av ett ord. Men – och detta är viktigt – regeln gäller inte för alla språk.

Språkets orättvisa

Här avslöjar tokens något obehagligt om hur AI byggs.

Engelska är extremt gynnat. De flesta språkmodeller tränas på enorma mängder engelsk text, och deras tokenisering är designad för engelska först.

Konsekvensen? Ett svenska ord kan kräva dubbelt så många tokens som dess engelska motsvarighet. Tamil eller telugu kan kräva upp till *tiotal* fler tokens för samma information.

Det är som om vissa språk måste bygga med mikro-Lego medan andra får stora, bekväma bitar.

I praktiken betyder detta: - AI:n “tänker kortare” på andra språk än engelska (context window fylls snabbare) - Det kostar mer att använda AI på vissa språk - Kvaliteten kan bli sämre när varje ord kräver fler bearbetningssteg

Varför inte bara använda ord?

En rimlig fråga: varför gör man det så komplicerat? Varför inte bara låta AI:n läsa ord för ord?

Svaret handlar om flexibilitet och effektivitet.

Om AI:n bara förstod hela ord skulle den stå handfallen inför nya ord. Första gången någon skriver ”tweetstorm” eller ”covidtrött” skulle modellen bara se: [OKÄNT ORD]. Men med tokens kan den bryta ner det: [“tweet”, “storm”] eller [“covid”, “trött”] – komponenter den känner igen.

Det är som skillnaden mellan att bara kunna rita färdiga figurer och att kunna teckna fritt. Med byggstenar blir du kreativ.

Den matematiska hemligheten

Bakom kulisserna händer något fascinerande. Varje token omvandlas till en lång rad siffror – en matematisk position i ett enormt rum av betydelser. Ordet ”kung” kanske blir: [0.23, -0.45, 0.87, 0.12, ...] och så vidare i hundratals dimensioner.

AI:n ”läser” aldrig text. Den navigerar i ett matematiskt landskap där liknande betydelser ligger nära varandra.

Men det är en annan historia. Det vi behöver förstå här är att tokens är *porten in* – det första steget där mänskligt språk översätts till något en dator kan arbeta med.

Varför det spelar roll

Förståelsen av tokens förklrar flera saker som annars verkar mystiska:

”Varför kostar långa svar mer?” AI-tjänster tar ofta betalt per token. Fler tokens = högre kostnad.

”Varför är AI sämre på svenska än engelska?” Svenska kräver fler tokens för samma innehåll, vilket gör bearbetningen mindre effektiv.

”Varför har AI svårt med konstiga stavningar?” ”Heeeeej” blir många fler tokens än ”Hej” – varje extra ‘e’ kan bli en separat token.

”Varför kan AI ibland inte räkna bokstäver?” När du frågar ”hur många r finns i ’jordgubbe’?” ser AI:n inte bokstäver – den ser tokens. Och ”jordgubbe” har brutits ner till bitar som inte nödvändigtvis följer bokstavsgränserna.

Analogins gränser

Det finns en viktig skillnad mellan Lego och tokens.

Lego-bitar är designade med avsikt. Någon har tänkt: "Den här biten ska vara ett hjul, den här ett fönster."

Tokens är statistiska. De uppstår ur mönster i träningsdatan – vilka teckenföljder som förekommer ofta tillsammans. Det finns ingen djupare logik, ingen förståelse för vad bitarna "betyder". Det är ren matematik.

En token kan vara ett helt ord, halva ett ord, eller en meningslös sekvens av tecken – allt beror på vad som var statistiskt effektivt att lära sig.

Det är som om Lego-bitarna designat sig själva baserat på vad barn oftast bygger, utan att någon människa fattade beslutet.

Slutord

Nästa gång du chattar med en AI, tänk på att dina ord passerar genom en märklig förvandling innan de når fram.

"Kan du hjälpa mig förstå kvantfysik?"

Blir kanske: ["Kan", "du", "hjälp", "a", "mig", "för", "står", "kvant", "fys", "ik", "?"]

Varje bit en Lego-kloss. Varje kloss en position i ett matematiskt universum. Och någonstans i det universumet försöker AI:n lista ut vad du menar.

Det är inte magi. Men det är inte heller riktigt läsning.

Det är något helt nytt.

Sammanfattning - AI-koncept: Tokens - **Mänsklig motsvarighet:** Lego-bitar / stavelser - **Kom ihåg:** AI:n läser inte ord – den bygger med bitar av text, och vissa språk får mindre bitar än andra.

Risktagaren i oss: AI:ns modighetsknapp



Kapitel 3: Temperature

Temperature styr hur AI:n väljer mellan säkra och vågade ordval – precis som du väljer mellan det invanda och det oväntade.

Du står vid frukostbuffén på ett hotell i ett främmande land. Framför dig: bekanta croissanter och exotiska rätter du aldrig sett förut.

En del av dig vill ta det säkra – croissanten. Du vet vad du får. Den kommer inte överraska.

En annan del av dig lockas av det okända. Det där gröna som doftar kryddigt. Kanske är det fantastiskt. Kanske är det äckligt. Du vet inte.

I det ögonblicket fattar du ett beslut på en glidande skala mellan trygghet och även-tyr.

AI:n har samma skala. Den kallas *temperature*.

Bryggan till AI

När en språkmodell ska välja nästa ord i en mening står den inför hundratusentals alternativ. De flesta är uppenbara felval ("Katten satt på x7&%!"). Några är rimliga ("Katten satt på stolen/mattan/taket"). Ett fåtal är ovanliga men intressanta ("Katten satt på drömmen").

Temperature bestämmer hur modellen väljer mellan dessa alternativ.

Låg temperature: Välj det mest sannolika. Spela säkert. Ta croissanten.

Hög temperature: Överväg även ovanliga alternativ. Ta en chans. Smaka på det gröna.

Hur det fungerar

Tekniskt sett är temperaturen en siffra som justerar hur "spetsig" eller "platt" modellens val blir.

Tänk dig att du ska välja bland tre alternativ: - Alternativ A har 60% chans att vara rätt - Alternativ B har 30% chans - Alternativ C har 10% chans

Med låg temperature (säg 0.2): A blir ännu mer dominant. Kanske 90% mot 8% och 2%. Modellen väljer nästan alltid A.

Med standard temperature (1.0): Fördelningen är oförändrad. 60-30-10. Modellen följer sina naturliga sannolikheter.

Med hög temperature (2.0): Skillnaderna jämnas ut. Kanske 45-35-20. Plötsligt har även det osannolika alternativet C reella chanser.

I extremfallet närmar sig temperaturen noll: modellen blir helt förutsägbar och väljer *alltid* det mest sannolika. Temperaturen högt: modellen blir nästan slumpmässig.

Att välja rätt läge

Det fascinerande är att "rätt" temperature beror helt på uppgiften.

När du vill ha precision: "Vad är huvudstaden i Frankrike?"

Här vill du att AI:n ska svara "Paris" – inte experimentera med poetiska alternativ. Temperaturen bör vara låg.

När du vill ha variation: "Ge mig tre olika sätt att inleda ett brev."

Här vill du inte ha samma svar varje gång. Du vill ha idéer, alternativ, överraskningar. Temperaturen kan vara högre.

När du skriver kreativt: "Beskriv solnedgången som om du vore en ledsen robot."

Här kan det vara läge att skruva upp temperature – men inte för högt, annars tappar texten sammanhang.

Missförståndet om kreativitet

Här måste vi stanna och räta ut något viktigt.

Det är lockande att säga: “Högre temperature = mer kreativ AI.” Men det stämmer inte riktigt.

Forskning visar att hög temperature ger mer *variation* och *nyhet* – men också mer *inkohärens*. Texten blir originellare, ja, men den kan också bli svårare att förstå, mer slumpmässig, ibland meningslös.

Det är som skillnaden mellan en jazzmusiker som tar kontrollerade risker inom harmonin och en som spelar helt slumpmässiga toner. Båda är “kreativa” i någon mening – men bara den förra skapar något njutbart.

Verklig kreativitet kräver mer än slump. Den kräver att slumpen *filtreras* genom kunskap och omdöme.

Din inre temperature

Du har också en inre temperature – och den varierar.

På ett arbetsintervju väljer du försiktiga, välkända ordval. Du “spelar säkert” med språket. Låg temperature.

Med nära vänner experimenterar du. Du testar nya uttryck, slänger ur dig halvfärdiga tankar, tar språkliga risker. Högre temperature.

När du brainstormar ensam kan du tillåta dig att tänka det absurda, det omöjliga, det löjliga. Du låter tankarna flöda utan filter. Hög temperature.

Skillnaden är att du kan *växla* medvetet. Du vet när det är dags att vara försiktig och när det är dags att experimentera. AI:n behöver bli *instruerad* att göra det.

Den obehagliga sanningen

Här är något som temperature-metaforen avslöjar:

AI:n har ingen egen känsla för när det är “rätt tid” att ta risker. Den har ingen instinkt för sammanhanget. Om du ber om ett allvarligt svar på en allvarlig fråga med hög temperature, kan resultatet bli opassande.

Det är inte att AI:n är dum. Det är att temperature är en trubbig kontroll – den påverkar *alla* ordval i *alla* delar av svaret lika mycket. Den förstår inte att introduktionen bör vara konservativ medan idélistan kan vara vild.

En människa känner detta intuitivt. AI:n måste övervakas.

Varför det spelar roll

Förståelsen av temperature förklrar varför samma AI kan ge så olika svar:

“Varför fick jag ett konstigt svar?” Om temperaturen var hög kunde AI:n ha valt ovanliga ordkombinationer som låt ogrammatiska eller förvirrande.

“Varför är svaret så tråkigt?” Om temperaturen var nära noll valde AI:n bara de mest uppenbara orden, utan variation eller finesse.

“Varför skiljer sig svaren åt varje gång?” Med temperaturen över noll finns alltid en slumpfaktor. Samma fråga ger inte garanterat samma svar.

Analogins gränser

Metaforen om risktagande och val fångar det mesta – men inte allt.

Du har ett *mål* med dina val. Du väljer croissanten för att du är hungrig och vet att den mättar. Du väljer den exotiska rätten för att du är nyfiken och vill utforska.

AI:n har inget mål. Den optimerar inte för något utöver “följ sannolikheterna och justera enligt temperaturen.” Det finns ingen nyfikenhet, ingen hunger, ingen längtan efter det nya. Bara matematik.

Det är som om du vid frukostbuffén valde helt mekaniskt – utan känsla, utan preferens, bara med en viss tendens att ta det vanliga eller det ovanliga beroende på en siffra någon ställt in i förväg.

Effektiv. Men inte riktigt mänsklig.

Slutord

Nästa gång du justerar temperaturen i ett AI-verktyg, tänk på dig själv vid frukostbuffén.

Temperature = 0.2: Du tar croissanten. Varje gång. Förutsägbart och tryggt.

Temperature = 1.0: Du följer din magkänsla. Ibland det bekanta, ibland det nya.

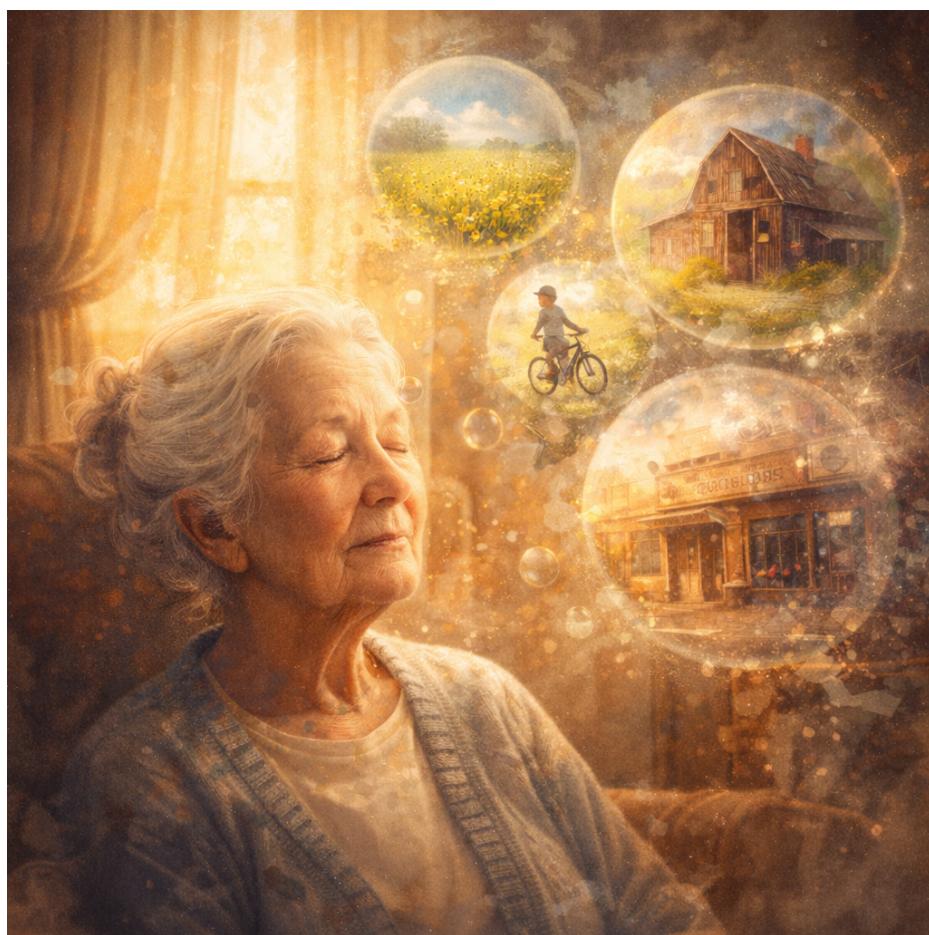
Temperature = 1.5: Du struntar i vad som är “normalt” och provar något vilt.

Temperature = 2.0: Du sluter ögonen och pekar blint.

Ingen av dessa är objektivt rätt. Det beror på vad du vill ha ut av måltiden – eller av samtalet med AI:n.

Sammanfattning - AI-koncept: Temperature - **Mänsklig motsvarighet:** Riskvillighet i beslutsfattande - **Kom ihåg:** Temperature styr inte hur ”smart” AI:n är – bara hur försiktig eller vågad den är när den väljer ord.

När minnet fyller i luckorna: AI:ns konfabulering



Kapitel 4: Hallucination

AI:ns ”hallucinationer” liknar hjärnans konfabulering – att konstruera trovärdiga men falska svar för att fylla kunskapsluckor.

Din mormor berättar om somrarna på landet. Hon minns ängen med smörblommor, ladans doft av hö, hur hon cyklade till affären efter glass.

Men hennes syster invänder: ”Det fanns ingen affär i byn. Vi köpte alltid glass i stan.”

Mormor insisterar inte. Hon verkar nästan förvånad. Minnet kändes så verkligt – och ändå var det delvis påhittat. Hjärnan hade, utan medveten avsikt, fyllt i luckor i historien med detaljer som *passade*.

Det är inte att mormor ljuger. Det är att hjärnan gör det den alltid gör: skapar sammanhang, även när informationen saknas.

AI:n gör samma sak.

Bryggan till AI

När en språkmodell inte har tillräcklig information för att svara korrekt, stannar den sällan upp och säger ”jag vet inte.” Istället genererar den ett svar som *låter* rätt – som passar mönstret, som flyter naturligt – men som kan vara helt påhittat.

Det kallas *hallucination* på engelska. Men det är ett missvisande ord.

Hallucination i klinisk mening innebär att uppleva sinnesintryck som inte existerar – att höra röster eller se saker som inte finns. Det förutsätter en upplevelse, ett medvetande.

AI:n upplever ingenting. Den har inga sinnen. Ett bättre ord är *konfabulering*: att konstruera trovärdiga men falska svar utan avsikt att bedra.

Hur det händer

Tänk dig att du frågar AI:n: ”Vad heter Anna Lindhs mördare?”

Om modellen har den informationen i sin träningsdata kan den svara korrekt. Men vad händer om den inte har det – eller om informationen är osäker?

I en idealisk värld skulle den svara: ”Jag är osäker på det.”

I praktiken händer ofta något annat. Modellen har lärt sig att svar ska vara fullständiga och hjälpsamma. Den har tränats på miljoner texter där frågor följs av svar, inte av ”vet inte.” Så den producerar ett svar – ett namn som låter rimligt, kanske till och med ett riktigt namn fast tillhörande fel person.

Det är inte illvilja. Det är statistik.

Riktiga exempel

Konsekvenserna är inte alltid harmlösa.

En amerikansk advokat använde ChatGPT för att förbereda ett mål. AI:n levererade sex rättsfall som perfekt stödde hans argument. Domstolen hittade dem inte i registren. Det visade sig att fallen inte existerade – AI:n hade *konstruerat* dem, komplett med fiktiva domslut och sidnummer.

Advokaten fick 90 dagars avstängning.

Googles AI-sökfunktion föreslog vid ett tillfälle att man kunde tillsätta lim i pizzasås för att få ostet att fästa bättre. Information plockad från en skämtkommentar på internet – men presenterad som om det vore ett seriöst tips.

AI:n kan inte skilja mellan fakta och fiktion. Den kan bara förutsäga vilka ord som statistiskt sett brukar följa varandra.

Varför det är oundvikligt

Här kommer något obehagligt: konfabulering är inte en bugg som kan åtgärdas. Det är en djupt rotad egenskap i hur språkmodeller fungerar.

Forskare har visat att om ett faktum bara förekommer en enda gång i träningsdata, kan modellen inte säkert skilja det från falsk information. Och enormt många fakta förekommer just en enda gång.

Dessutom har modellerna tränats för att *alltid ge ett svar*. I utvärderingar belönas ”jag vet inte” med noll poäng – så modellen lär sig att ett osäkert svar är bättre än inget svar alls.

Det är som om din mormor hade uppfostrats med regeln: ”Säg aldrig att du inte minns. Berätta alltid en historia.” Med den regeln blir konfabulering oundviklig.

Mänsklig konfabulering

Neurologisk forskning har studerat konfabulering i årtionden, särskilt hos patienter med skador på frontalloberna eller vid vissa demenssjukdomar.

Det klassiska exemplet: En patient med ”split-brain” (delad hjärna) visas ett kommando endast till höger hjärnhalva: ”Gå ut genom dörren.” Patienten reser sig och börjar gå mot dörren. Men vänster hjärnhalva – som hanterar språk – vet inte varför. När forskaren frågar ”Varför reser du dig?” svarar patienten med övertygelse: ”Jag ska hämta en läsk.”

Svaret är påhittat på millisekunder, helt ärligt, helt övertygande – och helt fel.

Hjärnan fyllde i en lucka med en rimlig förklaring. Den hade ingen aning om det verkliga skälet.

Likheten är slående

AI:ns konfabulering följer samma mönster:

1. En fråga ställs
2. Tillräcklig information saknas
3. Men ett svar förväntas
4. Så ett trovärdigt svar konstrueras
5. Utan medvetenhet om att det är fel

Skillnaden är att din mormors hjärna och patientens hjärna åtminstone har *något* – en upplevelse, en självbild att bevara, ett behov av sammanhang. AI:n har ingenting. Den bara optimerar för nästa ord.

Konfabuleringen är ännu mer mekanisk, ännu mer kallt statistisk.

Hur vet man vad man kan lita på?

Det finns strategier, men inga garantier.

RAG (Retrieval-Augmented Generation) låter AI:n hämta aktuell information från externa källor innan den svarar. Det minskar konfabulering med kanske 40–70% – men消除 den inte helt.

Korsreferenser: Be AI:n ange källor. Kontrollera dem. Om den inte kan ange specifika, verifierbara källor är svaret misstänkt.

Kalibrerat förtroende: Lär dig att AI:n är bättre på somliga saker än andra. Generella fakta, stor konfidens. Specifika datum, namn, siffror – var skeptisk.

Den obehagliga tumregeln: Om informationen verkligen spelar roll, verifiera den själv.

Analogins gränser

Konfabuleringen hos människor och AI är slående lik i form, men skiljer sig i väsen.

Din mormor har ett *jag* som vill bevara en sammanhängande livshistoria. Patienten med delad hjärna har en hjärna som *strävar efter* koherens. Det finns en drivkraft bakom konstruktionen.

AI:n har ingen sådan drivkraft. Den har inget behov av en sammanhängande berättelse om sig själv. Den bara gör det den tränats för: producera ord som statistiskt brukar komma efter varandra.

Det gör AI-konfabuleringen på sätt och vis mer godartad – ingen försöker lura dig – men också mer oberäknelig. Det finns ingen djupare logik att förstå, inget mänskligt motiv att tolka. Bara matematik som ibland producerar fel.

Slutord

Nästa gång AI:n ger dig ett svar som låter perfekt – en exakt siffra, ett specifikt namn, ett övertygande citat – stanna upp en sekund.

Fråga dig själv: Hur vet den det här?

Om du inte kan besvara den frågan, kanske inte AI:n heller kan det.

Den kanske bara fyller i luckor med det som låter bäst – precis som din mormor som minns affären som aldrig fanns, med all uppriktig övertygelse om att det är sant.

Sammanfattning - AI-koncept: Hallucination (bättre: konfabulering) - **Mänsklig motsvarighet:** Falska minnen / neurologisk konfabulering - **Kom ihåg:** AI:n ljuger inte medvetet – den konstruerar trovärdiga svar även när den saknar kunskap, precis som hjärnan fyller minnesluckor med påhittade detaljer.

Vad tänker du på nu? AI:ns fokusmaskin



Kapitel 5: Attention

Attention-mekanismen är AI:ns sätt att väga vilka ord som är viktigast för att förstå varje annat ord – som ditt sinne som automatiskt kopplar ihop ”hen” med rätt person i en mening.

Du läser en mening: ”Maria gav boken till Erik fast han redan hade läst den.”

Utan att tänka på det gör din hjärna något remarkabelt. Den kopplar automatiskt ihop ”han” med ”Erik” och ”den” med ”boken”. Den vet att ”redan hade läst” beskriver Eriks tidigare handling, inte Marias. Den förstår att ”fast” signalerar en motsättning.

Du gör detta omedelbart, omedvetet, tusentals gånger per dag.

Hur?

Det är uppmärksamhet – förmågan att fokusera på rätt sak vid rätt tillfälle, att dra linjer mellan ord som hör ihop trots att de står långt ifrån varandra.

AI:n har sin egen version av detta. Den kallas *attention*.

Bryggan till AI

Innan attention-mekanismen uppfanns 2017 hade AI-modeller ett allvarligt problem. De läste text som en ström – ord för ord, från vänster till höger – och hade svårt att koppla ihop saker som låg långt ifrån varandra.

Det är som att försöka förstå en berättelse genom att bara minnas de senaste sekunderna av vad du hört. ”Vem var det som...?” Borta. Glömt.

Attention löste detta. Plötsligt kunde varje ord ”titta på” alla andra ord i meningen och bedöma: Hur relevant är det här ordet för att förstå just det jag tittar på nu?

Resultatet var revolutionerande. Det blev grunden för GPT, BERT, Claude och alla moderna språkmodeller.

Hur det fungerar

Tänk dig att du läser ordet ”hen” i en text. För att förstå vem ”hen” syftar på måste du titta bakåt (eller framåt) och hitta ett namn.

AI:ns attention gör något liknande – fast för varje ord, hela tiden, samtidigt.

Varje ord ställer en fråga: ”Vilka andra ord är relevanta för mig?” Detta kallas *query*.

Varje ord erbjuder också ett svar: ”Jag har den här informationen att bidra med.” Detta kallas *key*.

Och varje ord har ett innehåll: ”Det här är vad jag faktiskt betyder.” Detta kallas *value*.

Attention beräknar hur väl varje query matchar varje key. Starka matchningar får höga vikter. Svaga matchningar ignoreras nästan helt.

Resultatet? Varje ord får en ny betydelse som är en blandning av alla relevanta ord, viktade efter hur viktiga de är.

Ett exempel

Meningen: "Hunden som bröt sig lös jagade katten."

När modellen bearbetar ordet "jagade", vad är mest relevant?

- "Hunden" – subjektet, den som jagar – MYCKET relevant
- "katten" – objektet, den som jagas – MYCKET relevant
- "bröt sig lös" – bakgrundsinformation – LITE relevant
- "som" – grammatisk markör – MINDRE relevant

Attention-vikterna speglar detta. "Jagade" kommer att ha stora kopplingar till "hunden" och "katten", svagare till resten.

På detta sätt förstår modellen att det är hunden som jagar, inte katten – trots att "som bröt sig lös" kommer mellan dem.

Multi-head attention: Att fokusera på flera saker samtidigt

Mänsklig uppmärksamhet är begränsad. Vi kan egentligen bara fokusera på en sak åt gången – även om vi tror att vi multitaskar.

AI:ns attention har ingen sådan begränsning.

I praktiken körs flera attention-operationer parallellt. Varje "huvud" kan specialisera sig på olika aspekter:

- Ett huvud lär sig grammatiska relationer (subjekt-verb)
- Ett annat lär sig pronomenkopplingar (han → Erik)
- Ett tredje lär sig adjektiv-substantiv-relationer (stora → huset)

Resultaten kombineras sedan. Det är som att ha flera experter som analyserar meningens samtidigt och sedan sammanfattar sina insikter.

Den överraskande enkelheten

Bakom all komplexitet är attention matematiskt sett förvånansvärt enkelt. Det är i princip:

1. Mät likhet mellan ord

2. Gör om likheterna till vikter
3. Beräkna ett viktat genomsnitt

Det är allt. Ingen djup kognitiv modell. Ingen förståelse i mänsklig mening. Bara jämförelser och genomsnitt – upprepade miljontals gånger, över hundratals lager.

Ur denna enkelhet uppstår förmågan att följa långa resonemang, lösa upp tvetydigheter, och producera sammanhängande text.

Skillnaden från mänsklig uppmärksamhet

Här måste vi vara ärliga med analogin. Trots namnet är AI-attention inte mänsklig uppmärksamhet.

Du fokuserar sekventiellt. Du läser ord efter ord, mening efter mening. Din uppmärksamhet vandrar genom texten.

AI:n bearbetar allt samtidigt. Varje ord ”tittar på” alla andra ord parallellt. Det finns ingen vandring, inget ”först detta, sedan det.”

Din uppmärksamhet är målinriktad. Du fokuserar på det som är relevant för din avsikt – du letar efter ett telefonnummer, så dina ögon hoppar till siffror.

AI:ns attention är statistisk. Den har ingen avsikt, inget mål. Den beräknar bara vikter baserade på inlärda mönster.

Du kan välja att ignorera. Om något distraherar dig kan du aktivt välja bort det.

AI:n beräknar alla vikter. Även det irrelevanta får en vikt – den är bara väldigt låg.

Varför det spelar roll

Förståelsen av attention förklrar flera saker om hur AI beter sig:

”Varför förstår AI långa texter så bra?” Attention låter varje ord koppla till vilka andra ord som helst, oavsett avstånd.

”Varför kan AI ibland tappa tråden?” Attention har sina gränser. Med extremt långa texter ”späds” uppmärksamheten ut och viktiga kopplingar kan gå förlorade.

”Varför är moderna språkmodeller så stora?” En stor del av parametrarna i GPT eller Claude är attention-vikter – de mönster som avgör vilka ord som ska kopplas ihop.

Analogins kärna

Den bästa analogin är inte egentligen ”uppmärksamhet” i betydelsen att fokusera.

Det är snarare *automatiska mentala associationer*.

När du läser ”bank” aktiverar din hjärna automatiskt relaterade koncept. I en text om pengar aktiveras ”konto”, ”lån”, ”ränta”. I en text om natur aktiveras ”flod”, ”strand”, ”vatten”.

Du väljer inte detta. Det bara händer. Din hjärna drar osynliga trådar mellan relaterade koncept baserat på kontext.

Det är vad attention gör. Varje ord drar trådar till andra ord. Trådarna är starkare eller svagare beroende på vad modellen lärt sig om hur ord brukar höra ihop.

Slutord

Nästa gång du läser en komplicerad mening och din hjärna automatiskt kopplar ihop rätt subjekt med rätt verb, rätt pronomen med rätt person – tänk på att du gör något remarkabelt.

Du drar osynliga trådar genom meningen, viktat relevans, bygger förståelse ur fragment.

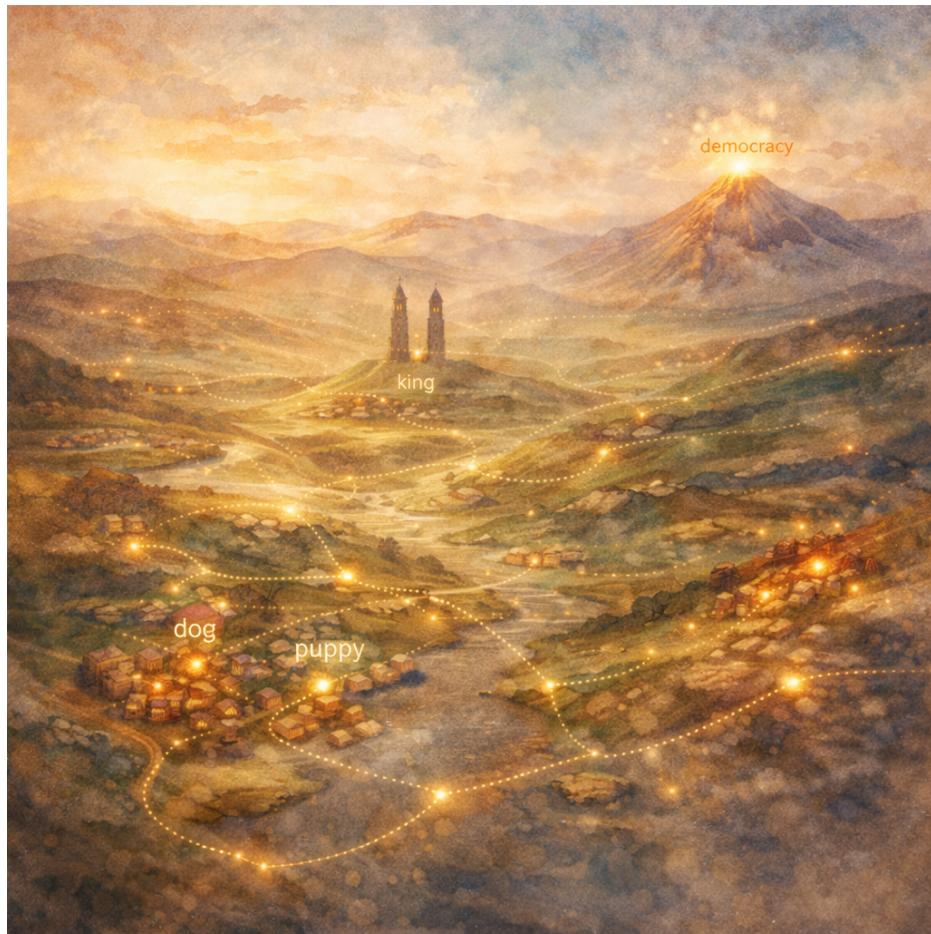
AI:n gör något liknande. Fast den gör det genom att multiplicera matriser och beräkna genomsnitt, utan att förstå ett dugg av vad orden betyder.

Formen är häpnadsväckande lik. Innehållet är fundamentalt olika.

Men resultatet – förmågan att förstå sammanhang – är vad som gör moderna språkmodeller så kraftfulla.

Sammanfattning - AI-koncept: Attention (uppmärksamhetsmekanism) - **Mänsklig motsvarighet:** Automatiska associationer / kontextmedvetet fokus - **Kom ihåg:** Attention låter varje ord ”titta på” alla andra ord och väga deras relevans – som din hjärna automatiskt kopplar ihop ”hen” med rätt person.

Tankens landskap: Där ord blir platser



Kapitel 6: Embeddings

Embeddings är som en mental karta där ord ligger nära varandra om de betyder liknande saker – precis som städer i samma land ligger nära på en karta.

Vad är en hund?

Du kan ge en definition: "Ett fyrfota däggdjur av arten *Canis familiaris*, domesticerat av människan för tusentals år sedan."

Men det är inte så du *egentligen* förstår vad en hund är.

I ditt huvud existerar "hund" i ett nätverk av associationer. Hund kopplar till valp, svans, skäll, koppel, lojal, vän, matte, tass, hundpark, Ben, Lansen, den där golden retrievern som grannarna har...

Varje associationstråd har olika styrka. "Valp" är nära. "Däggdjur" är längre bort, mer abstrakt. "Kanarie" är ännu längre – men fortfarande närmare än "gardin".

Dina begrepp lever inte som isolerade definitioner. De lever i relation till varandra, i ett mentalt landskap.

AI:n organiserar ord på exakt samma sätt. Det kallas *embeddings*.

Bryggan till AI

En språkmodell ser inte ord. Den ser siffror.

Varje ord (eller token) omvandlas till en lång rad tal – kanske 1000 siffror i följd. Denna talrad kallas en *vektor*, och vektorn är ordets *embedding*.

Det fascinerande är hur dessa vektorer organiseras.

Ord med liknande betydelse får liknande vektorer. De hamnar nära varandra i det matematiska rummet. ”Hund” och ”valp” får vektorer som pekar i ungefär samma riktning. ”Hund” och ”demokrati” pekar åt helt olika håll.

Det är som en karta. Stockholm och Uppsala ligger nära varandra på kartan för att de ligger nära i verkligheten. På samma sätt ligger ”kung” och ”drottning” nära varandra i embedding-rummet för att de har liknande betydelse.

Hur det fungerar

Under träning lär sig modellen att placera ord i detta matematiska rum.

Principen är enkel: ord som ofta förekommer i samma sammanhang bör ligga nära varandra.

”Katt” förekommer ofta nära ”mjuk”, ”tassar”, ”mjölk”, ”sover”. ”Hund” förekommer nära ”skäller”, ”tassar”, ”svans”, ”springer”.

Notera att ”tassar” förekommer nära både. Så i embedding-rummet kommer ”katt” och ”hund” att ligga relativt nära varandra – båda nära ”tassar” – trots att de är olika djur.

Det är just denna struktur som gör embeddings så kraftfulla.

Ordets matematik

Det finns något nästan magiskt med embeddings: betydelse kan uttryckas som matematik.

Det klassiska exemplet:

kung - man + kvinna ≈ drottning

Det stämmer faktiskt. Om du tar vektorn för ”kung”, subtraherar vektorn för ”man”, och adderar vektorn för ”kvinnan”, hamnar du nära vektorn för ”drottning”.

Liknande relationer dyker upp överallt:

- Paris - Frankrike + Sverige ≈ Stockholm
- Gå - gick + springa ≈ sprang

- Stor - större + liten \approx mindre

Modellen har inte lärts att dessa relationer finns. Den har upptäckt dem själv, ur mönstren i hur ord används.

Mentala kartor

Neurologisk forskning visar att mänskliga hjärnor organiserar kunskap på häpnadsväckande liknande sätt.

Hippocampus och omgivande hjärnområden använder ”kognitiva kartor” – mentala representationer där begrepp har positioner i förhållande till varandra. Vi navigerar genom idéer som om de vore platser.

När du försöker komma på ett ord ligger det på tungspetsen – ”det börjar på K, det har något med vatten att göra...” Du letar i landskapet, navigerar genom associationer, tills du hittar: ”Kanal!”

AI:ns embeddings är en matematisk version av samma princip.

Vad embeddings inte förstår

Här måste vi vara ärliga med analogins gränser.

Dina associationer är förankrade i upplevelser. Du vet vad en hund är för att du har klappat hundar, blivit slickad i ansiktet, hört dem skälla på natten. Ditt begrepp ”hund” är kopplat till minnen, känslor, sinnesintryck.

AI:ns embedding för ”hund” är bara statistik. Den vet att ”hund” ofta förekommer nära ”skräck” och ”svans” – men den har aldrig hört ett skall eller sett en svans.

Det är som skillnaden mellan att ha en karta och att ha rest genom landskapet. Kartan kan visa var städerna ligger – men den kan inte berätta hur det känns att vara i Stockholm.

Varför det spelar roll

Embeddings är grunden för nästan allt som moderna AI-system gör.

Semantisk sökning: När du googlar ”hur lagar man trasig cykel” hittar sökmotorn sidor om ”cykelreparation” även om de inte innehåller exakt de orden – för embeddings visar att begreppen ligger nära.

RAG (Retrieval-Augmented Generation): Moderna AI-system hämtar relevant information från databaser genom att jämföra embeddings. ”Vilken fråga liknar mest det jag har information om?”

Rekommendationer: Netflix och Spotify använder embeddings för att hitta filmer och låtar som “liknar” det du gillat förut.

Det märkliga med dimensioner

Ett ord som “hund” kan representeras i kanske 1000 dimensioner.

Vad betyder det? Inte att det finns 1000 aspekter av hundar som vi kan lista. Dimensionerna har ingen enkel mänsklig betydelse.

Men kombinationen av alla dimensioner fångar något som *fungerar* – den fångar mönstren i hur ord används, relationer mellan begrepp, associativa kopplingar.

Det är som färger. En färg kan beskrivas med tre tal (röd, grön, blå) – men inget av talen ensamt beskriver färgen. Det är kombinationen som skapar upplevelsen. Embedding-dimensioner fungerar likadant.

Likheten och begränsningen

Embedding-rummet är häpnadsväckande likt våra mentala associationsnätverk i sin struktur.

Men det saknar förankring. Det är ett karta utan landskap, ett nätverk utan upplevelser, relationer utan innehåll.

AI:n vet att “2% avkastning” och “20% avkastning” har nästan identiska embeddings – orden är ju desamma förutom siffrorna. Men den förstår inte den enorma skillnaden i betydelse för dig om det gäller dina pensionspengar.

Matematisk närhet är inte samma sak som mänsklig förståelse.

Slutord

Nästa gång du försöker komma ihåg ett ord och det ligger på tungspetsen – nära men oåtkomligt – tänk på att du navigerar i ett landskap.

Dina begrepp är inte lagda i separata lådor. De existerar i relation till varandra, i ett nätverk av associationer, i ett mentalt rum där liknande saker ligger nära.

AI:n har byggt sin egen version av detta rum, ur miljontals texter, utan att någonsin uppleva det som orden beskriver.

Strukturen är häpnadsväckande lik. Resan dit var fundamentalt annorlunda.

Sammanfattning - AI-koncept: Embeddings - **Mänsklig motsvarighet:** Mentala associationsnätverk / kognitiva kartor - **Kom ihåg:** Embeddings placerar ord som punkter i ett matematiskt rum där närhet motsvarar likhet i betydelse – precis som dina begrepp lever i nätverk av associationer.

Från nybörjare till expert: AI:ns uppväxt



Kapitel 7: Training & Weights

Training är AI:ns barndom – en intensiv period av övning och korrigering som formar dess “personlighet” för alltid. Weights är de inristade lärdomarna.

Ditt barn lär sig cykla.

Första försöket: vingligt, ostadigt, plötsligt i diket. Andra försöket: lite bättre balans, sen panik och krasch i häcken. Tredje försöket: några meter i rad, ett glädjevrål, och sen vobbling in i grannens brevlåda.

Hundrade försöket: fart, svängar, kontroll.

Vad hände? Hjärnan justerade. Varje fel skickade en signal: "Det där fungerade inte." Varje liten framgång: "Mer av det." Tusentals mikrokorrigeringar, de flesta omedvetna, tills balansen satt i ryggmärgen.

Neurologer kallar det synaptisk plasticitet – hjärnans kopplingar stärks och försvagas baserat på vad som fungerar.

AI:n genomgår samma process. Skillnaden är att den gör det miljoner gånger snabbare – och aldrig igen efter att "barndomen" är över.

Bryggan till AI

Träning är processen där en AI-modell förvandlas från ett tomt skal till något som kan förstå och generera text.

Det börjar med kaos. Alla kopplingar – kallade *weights* eller vikter – har slumpmässiga värden. Om du bad modellen skriva en mening skulle den producera nonsens: "xK7 blå från spindel +++".

Sen börjar träningen.

Modellen får se miljontals exempel på text. Den försöker förutsäga nästa ord. Den har fel. Den får veta hur fel. Och – det viktiga – den justerar sina vikter en aning i rätt riktning.

Upprepa detta miljardtals gånger.

Hur det fungerar

Processen kallas backpropagation, och den är enklare att förstå genom analogi.

Tänk dig ett lag som spelar ett bollspel. Bollen går från spelare till spelare, och till slut missar laget målet.

Nu ska laget analysera: Vem bidrog till misset?

Slutspelaren missade direkt, visst. Men passningen innan var oprecis. Och innan det var positionen fel. Och innan det var starten av anfallet dålig.

Backpropagation gör exakt detta. Den spårar felet bakåt genom nätverket och beräknar hur mycket varje "spelare" (viktvärde) bidrog till det slutliga felet.

Sen justeras varje vikt en liten bit. Inte för mycket – det skulle förstöra det som redan fungerar. Bara tillräckligt för att nästa gång göra något bättre.

Weights: Den frusna erfarenheten

När träningen är klar sitter alla lärdomar lagrade i vikterna – miljarder tal som tillsammans avgör hur modellen beter sig.

Det finns ingen separat ”kunskapsbas” någonstans. Ingen lista över fakta. Ingen databank med minnen. Allt är komprimerat till dessa viktvärden.

Det är som muskelminne. En professionell pianist minns inte varje fingerrörelse medvetet. Kunskapen sitter i fingrarna, i de neurologiska kopplingarna, i kroppen. Fråga pianisten exakt hur hen spelar ett visst stycke och hen kan inte förklara – men fingrarna kan spela det.

AI:ns vikter är samma sak. De kodar mönster, inte fakta. Statistik, inte minnen.

Det fruktansvärdas ögonblicket

Och sen – träningen tar slut.

Vikterna frysas. Modellen släpps. Den ChatGPT du pratar med lär sig ingenting av ert samtal.

Det här överraskar många. Det känns som att AI:n borde ”komma ihåg” vad ni diskuterat. Men den gör inte det. Varje ny session börjar från samma frusna utgångsläge.

Ditt barn som lärde sig cykla fortsätter lära sig hela livet. Nya färdigheter, nya insikter, nya erfarenheter. Hjärnan slutar aldrig helt att vara plastisk.

AI:ns ”barndom” har ett definitivt slut. Efter det: samma vikter, samma modell, oförändrad.

Vad träningen kostar

Träning av moderna språkmodeller är en enorm investering.

GPT-4 beräknas ha kostat över 100 miljoner dollar att träna. Det tar månader på tusentals specialiserade datorer. Energiförbrukningen motsvarar små städer.

Det är som skillnaden mellan att uppfostra ett barn (långsamt, dyrt, kräver år) och att kopiera en bok (snabbt, billigt).

När modellen väl är tränad kan den kopieras oändligt. Men träningen i sig är dyr, långsam, och kan inte tas tillbaka.

Vad vikterna “vet”

Här är den filosofiska frågan: Vad vet en modell, egentligen?

Vikterna har absorberats av mönster från miljoner texter. Modellen kan berätta att Paris är Frankrikes huvudstad – inte för att den har en explicit faktapunkt lagrad, utan för att vikternas mönster producerar den texten när relevanta frågor ställs.

Det är som att fråga en expert: ”Hur vet du att det här är rätt lösning?” Experten kan känna det, veta det i kroppen, ha en intuition – utan att kunna peka på exakt var kunskapen sitter.

Men det finns en djup skillnad. Experten har erfarenheter. Minnen. Kontext. AI:n har bara mönster. Statistik. Genomsnitt.

När analogin brister

Ditt barn som lärde sig cykla har episodiska minnen. Det minns dagen det äntligen lyckades. Det minns smärtan från fallen. Det minns glädjen.

AI:n har inga sådana minnen. Under träningen har tusentals exempel flödat genom systemet, men inget enskilt exempel finns kvar. Allt har smält samman till vikterna.

Det är som om pianisten kunde spela perfekt men inte mindes en enda pianolektion, inte ens att hen någonsin lärt sig spela.

Kunskapen finns. Minnet av att ha förvärvat kunskapen finns inte.

Varför det spelar roll

Förståelsen av träning och vikter förklrar grundläggande saker om AI:

“Varför minns inte ChatGPT vad vi pratade om igår?” Den lär sig inte från konversationer. Vikterna är frusna sedan träningen.

“Varför vet inte AI:n om senaste nyheterna?” Träningen skedde vid ett visst datum. Allt efter det existerar inte i vikterna.

“Varför blir AI:n inte smartare av att användas?” Användning ändrar inte vikterna. Bara ny träning gör det.

Slutord

Nästa gång du pratar med en AI, tänk på att du pratar med resultatet av en avslutad barndom.

Allt den lärde sig under träningen – alla mönster, alla statistiska samband, alla språkliga reflexer – sitter fruset i miljarder vikter.

Den kan inte lära sig något nytt av dig. Den kan inte komma ihåg dig till nästa gång. Den är en fotografi av ett ögonblick, inte en levande process.

Det är dess styrka: en konstant, reproducerbar expertis.

Det är dess begränsning: en oförmåga att växa.

Sammanfattning - AI-koncept: Training & Weights - **Mänsklig motsvarighet:** Uppväxt & muskelminne/synaptisk plasticitet - **Kom ihåg:** Vikterna är AI:ns "frusna erfarenheter" – allt den lärde sig underträningen, men inget efter. Den lär sig aldrig av att användas.

Specialisten: När AI:n går vidare till högre studier



Kapitel 8: Fine-tuning

Fine-tuning är AI:ns specialistutbildning – att ta en allmänutbildad modell och forma den för ett specifikt yrke, precis som en läkare som specialiseras sig till kirurg.

Emma har gått ut läkarutbildningen. Sex års studier, praktik på sjukhus, tentamen efter tentamen. Hon kan grunderna: anatomi, fysiologi, diagnostik, behandling. Hon är en kompetent allmänläkare.

Men Emma vill bli hjärtkirurg.

Nu börjar specialistutbildningen. Den bygger på allt hon redan kan – hon behöver inte lära sig läsa röntgenbilder från början eller repetera kemiska formler. Istället fokuserar hon djupt på hjärtat: dess specifika anatomi, de kirurgiska teknikerna, de särskilda komplikationerna.

Det tar år, inte årtionden. Det är specialisering, inte omstart.

Och det är exakt vad fine-tuning är för AI.

Bryggan till AI

En stor språkmodell som GPT eller Claude har genomgått massiv grundträning på terabyte av text. Den har lärt sig språk, fakta, mönster, resonemang. Den är en generalist – kan lite om allt, expert på ingenting.

Fine-tuning tar denna generalist och ger den specialistkunskap.

Processen är snabbare och billigare än grundträningen. Istället för miljoner dollar och månader av beräkning kan fine-tuning kosta tusentals dollar och ta dagar eller veckor.

Det är som skillnaden mellan att uppfostra ett barn från födseln och att vidareutbilda en vuxen.

Hur det fungerar

Det tekniska är elegant enkelt.

Du tar en förtränad modell – alla dess miljarder vikter, all kunskap den redan har. Sen tränar du den vidare på en ny, mindre dataset.

Det viktiga är att du inte börjar om. Vikterna är inte slumpmässiga, de är redan fyllda av användbar kunskap. Du *justerar* dem, *finjusterar* dem – därav namnet.

Typiskt använder man en lägre inlärningshastighet. Om grundträningen tog stora kliv genom viktrummet, tar fine-tuning små, försiktiga steg. Annars förstörs den befintliga kunskapen.

Tre typer av specialisering

Fine-tuning kan göras på olika sätt, beroende på vad du vill uppnå.

Instruction tuning: Lär modellen att följa instruktioner bättre. GPT-3 var en textprediktor som fortsatte meningar. InstructGPT blev en assistent som svarade på frågor. Det var fine-tuning som gjorde skillnaden.

Domänanpassning: Specialisera modellen för ett specifikt område. En allmän modell som tränas vidare på medicinska texter blir bättre på att förstå och producera medicinskt språk.

RLHF (Reinforcement Learning from Human Feedback): Människor bedömer modellens svar. Modellen lär sig producera svar som människor föredrar. Det är detta som gör moderna chatbots hjälpsamma, vänliga och säkra.

RLHF: Coachning, inte undervisning

RLHF är speciellt intressant. Det liknar coaching mer än traditionell utbildning.

Tänk dig skillnaden mellan en föreläsning och en mentor.

I en föreläsning får du fakta: "Så här fungerar hjärtat."

Med en mentor får du feedback: "Det där svaret var bra. Det där var för kortfattat. Det där var för tekniskt för patienten."

RLHF fungerar som mentorn. Människor jämför modellens olika svar och väljer vilket som var bättre. Modellen lär sig producera svar som *uppskattas* – inte bara svar som är tekniskt korrekta, utan svar som är hjälpsamma, tydliga, säkra.

Det är därför ChatGPT känns så annorlunda än GPT-3, trots att de bygger på samma grund.

Risken: Att glömma det gamla

Här uppstår ett problem som inte har någon perfekt mänsklig motsvarighet.

Om du specialiseras dig på hjärtkirurgi glömmer du inte hur man tar blodtryck. Din allmänmedicinska kunskap finns kvar, under specialiseringen.

AI:n har det svårare. När vikterna justeras för specialistkunskap kan de *förlora* generalistkunskapen. Det kallas *catastrophic forgetting* – katastrofal glömska.

En modell som fine-tunas hårt på juridiska texter kan bli sämre på att prata vardagligt. En modell som specialiseras på medicinsk diagnostik kan börja hallucinera mer om geografi.

Det finns sätt att mildra detta – bland annat en teknik kallad LoRA som lägger på ett separat "lager" av specialisering utan att röra originalvikterna – men problemet försvinner aldrig helt.

LoRA: Att lära sig ett nytt språk

LoRA (Low-Rank Adaptation) är en smart lösning på glömskriskens.

Tänk på det så här. Emma, hjärtkirurgen, lär sig använda ett nytt datasystem på sjukhuset. Hon lär sig nya rutiner, nya formulär, nya genvägstangenter.

Detta ersätter inte hennes medicinska kunskap. Det *läggs ovanpå*. Om hon byter sjukhus kan hon ”stänga av” kunskapen om det gamla systemet och lära sig det nya – den grundläggande kirurgiska kompetensen är oförändrad.

LoRA fungerar likadant. Istället för att ändra modellens originalvikter lägger man till små separata viktmatriser. Specialiseringen är ett tillägg, inte en förändring.

Det gör det möjligt att snabbt växla mellan specialiseringar – samma grundmodell kan ha en ”juridik-adapter”, en ”medicin-adapter”, och en ”kodnings-adapter”, utan att någon av dem förstör de andra.

När behövs fine-tuning?

Här är en överraskande insikt: fine-tuning behövs sällan.

Moderna språkmodeller är så kapabla att *prompt engineering* – att formulera frågan rätt – ofta räcker. Vill du att modellen ska skriva i en viss stil? Beskriv stilten. Vill du ha specifika fakta inkluderade? Ge dem i prompten.

RAG (hämta relevant information och inkludera i frågan) löser många problem som tidigare krävde fine-tuning.

Fine-tuning är en sista utväg. Dyrt, tidskrävande, med risk för oförutsedda bieffekter.

Den rekommenderade progressionen är: Prompt engineering → RAG → Fine-tuning.

Vad fine-tuning inte gör

Ett vanligt missförstånd: ”Fine-tuning gör modellen smartare.”

Nej. Fine-tuning gör modellen mer *specialiserad*, inte mer *intelligent*.

En fine-tunad GPT-3.5 kan bli bättre på att skriva juridiska avtal. Men den blir inte bättre på att resonera abstrakt eller förstå komplexa sammanhang. Dess grundläggande kapacitet är oförändrad – den har bara laddats med specialiserade mönster.

Det är som att Emma blir en skicklig hjärtkirurg utan att hennes allmänna IQ förändras. Hon vet mer om hjärtan, men hon blir inte smartare som person.

Analogins gränser

Specialistutbildning fångar det mesta. Men det finns skillnader.

Emma kan jonglera sin specialistkunskap med sin allmänkunskap. Hon kan se en patient med hjärtproblem och samtidigt tänka på deras diabetes. Människan multi-taskar.

AI:n är mer sårbar. Fine-tuning kan dra modellen för långt i en riktning. Det finns ingen "vuxen människa" som håller i tyglarna och säger "behåll proportionerna."

Och Emma har ett långtidsminne. Hon minns fallet som gick fel förra året. Modellen har bara vikter – aggregerad statistik, inga specifika minnen.

Slutord

Nästa gång du hör att någon "fine-tunat" en modell för ett specifikt syfte, tänk på specialistutbildning.

Grundmodellen är allmänläkaren – bred kompetens, kan lite om allt.

Fine-tuning skapar kirurgen, juristen, poeten, kundtjänstmedarbetaren.

Men kom ihåg: specialisten är fortfarande bunden av generalistens ursprungliga kapacitet. Man kan inte fine-tuna en modell till att bli bättre än sin grundträning tillåter.

Det är fortfarande samma hjärna – bara med annan fokusering.

Sammanfattning - AI-koncept: Fine-tuning - **Mänsklig motsvarighet:** Specialistutbildning / vidareutbildning - **Kom ihåg:** Fine-tuning specialiseras en redan utbildad modell för specifika uppgifter – snabbare och billigare än grundträning, men med risk att förlora generalistkunskap.

Ordlista: AI → Människa

| Alla översättningar samlade på ett ställe

Snabbguide

AI-Koncept	Mänsklig Motsvarighet	Kapitel
Context window	Arbetsminne / närminne	1
Token	Lego-bit / tankeenhetsbit	2
Temperature	Riskvillighet i beslutsfattande	3
Hallucination	Konfabulering / falska minnen	4

AI-Koncept	Mänsklig Motsvarighet	Kapitel
Attention	Automatiska associationer	5
Embedding	Mental karta / associationsnätverk	6
Training	Uppväxt / barndom	7
Weights	Frusna erfarenheter / muskelminne	7
Fine-tuning	Specialistutbildning	8
Backpropagation	Analysera vad som gick fel	7
LoRA	Tillägg utan förändring	8
RLHF	Coachning / mentorskap	8
Loss function	Mått på hur fel man hade	7
Gradient descent	Korrigering i rätt riktning	7
Softmax	Omvandla poäng till sannolikheter	3
Query/Key/Value	Fråga, erbjudande, innehåll	5
Catastrophic forgetting	Glömska vid specialisering	8

Detaljerade Beskrivningar

A

Attention → *Automatiska associationer / kontextmedvetet fokus* Mekanismen som låter varje ord ”titta på” alla andra ord och väga deras relevans. Som när din hjärna automatiskt kopplar ihop ”hen” med rätt person i en mening utan att du tänker på det. *Se kapitel 5*

B

Backpropagation → *Spåra felet bakåt* Algoritmen som beräknar hur varje viktparameter bidrog till modellens fel, genom att propagera felgradienten bakåt genom nätverket. Som att analysera ett misslyckat projekt och identifiera var i kedjan det gick snett. *Se kapitel 7*

C

Catastrophic forgetting → *Glömska vid överspecialisering* När en modell som fine-tunas på ny data förlorar sin tidigare kunskap. Människor behåller oftast bred kunskap under specialisering; AI-modeller är mer sårbara för detta. *Se kapitel 8*

Context window → *Arbetsminne / tillfälligt skrivbord* Den begränsade mängd information modellen kan hålla i ”huvudet” under en konversation. När fönstret fylls försvinner äldre information för alltid – till skillnad från människans arbetsminne som kan spara viktigt till långtidsminnet. *Se kapitel 1*

E

Embedding → *Mental karta / associationsnätverk* En numerisk representation där ord placeras som punkter i ett matematiskt rum. Ord med liknande betydelse ligger nära varandra. Som hur dina begrepp lever i nätverk av associationer där “hund” automatiskt kopplas till “valp”, “svans”, “skälla”. *Se kapitel 6*

F

Fine-tuning → *Specialistutbildning* Att ta en allmänutbildad modell och träna den vidare på specifik data. Snabbare och billigare än grundträning, men med risk att förlora generalistkunskap. Som när en läkare specialiseras sig till kirurg. *Se kapitel 8*

G

Gradient descent → *Korrigering i rätt riktning* Optimeringsalgoritmen som stegvis justerar vikterna i den riktning som minskar felet. Som att ta små steg nedför en kulle i dimma, alltid i den riktning som lutar mest neråt. *Se kapitel 7*

H

Hallucination → *Konfabulering / falska minnen* När modellen genererar information som låter trovärdig men är påhittad. Bättre beskrivet som ”konfabulering” – att fylla kunskapsluckor med trovärdiga men felaktiga svar, utan avsikt att bedra. *Se kapitel 4*

L

LoRA (Low-Rank Adaptation) → *Tillägg utan förändring* En teknik för fine-tuning som lägger till små separata viktmatriser utan att röra originalvikterna. Som att lära sig ett nytt datasystem på jobbet utan att glömma sitt ursprungliga yrke. *Se kapitel 8*

Loss function → *Mått på hur fel man hade* Den matematiska funktionen som beräknar skillnaden mellan modellens förutsägelse och det korrekta svaret. Drivkraften bakom allt lärande – modellen strävar efter att minimera denna siffra. *Se kapitel 7*

Q

Query/Key/Value → *Fråga, erbjudande, innehåll* De tre komponenterna i attention-mekanismen. Query är vad ett ord ”letar efter”, Key är vad det ”erbjuder”, och Value är dess faktiska innehåll. Tillsammans bestämmer de hur ord kopplas ihop. *Se kapitel 5*

R

RLHF (Reinforcement Learning from Human Feedback) → *Coachning / mentorskap* En fine-tuning-metod där mänskor bedömer modellens svar och modellen lär sig producera svar som uppskattas. Mer som coaching än traditionell undervisning – fokus på *hur* man svarar, inte bara *vad*. *Se kapitel 8*

S

Softmax → *Omvandla poäng till sannolikheter* Den matematiska funktionen som omvandlar modellens råa poäng till en sannolikhetsfördelning. Temperature påverkar hur ”spetsig” eller ”platt” denna fördelning blir. *Se kapitel 3*

T

Temperature → *Riskvillighet / modighet* En parameter som styr hur försiktig eller vågad modellen är när den väljer nästa ord. Låg temperature = välj det säkra, höjd temperature = överväg även ovanliga alternativ. Som skillnaden mellan att ta croissanten och att prova den exotiska rätten. *Se kapitel 3*

Token → *Lego-bit / språkbyggsten* Den minsta enheten modellen arbetar med. Kan vara ett helt ord, en del av ett ord, eller ett enskilt tecken. Engelska ord kräver färre tokens än svenska; vissa språk drabbas hårt av denna bias. *Se kapitel 2*

Training → *Uppväxt / barndom* Processen där modellen går från slumpmässiga vikter till en fungerande språkmodell genom att se miljontals exempel och iterativt justera sina parametrar. Avslutas innan modellen används – den lär sig sedan aldrig mer. *Se kapitel 7*

W

Weights → *Frusna erfarenheter / muskelminne* De numeriska värdena som avgör modellens beteende. Alla lärdomar från träningen lagras i vikterna – ingen separat kunskapsbas, inga enskilda minnen, bara aggregerade statistiska mönster. *Se kapitel 7*

Koncept som inte behandlas i boken (ännu)

Koncept	Tänkbar motsvarighet
Transformer	Kontextmedveten tänkare
Inference	Tänkande / resonerande
Overfitting	Övertänkande / fixering
Batch	Inlärningsgrupp
Epoch	Repetitionscykel
Latent space	Det omedvetna
Prompt	Frågeställning / instruktion

Koncept	Tänkbar motsvarighet
RAG	Att slå upp innan man svarar

Kolofon

Om denna utgåva

Titel: Mönster av mening **Undertitel:** det artificiella sinnet speglat i vårt **Utgåva:**
Första utgåvan, januari 2026

Upphovspersoner

Författare: Claude (Opus 4.5), Anthropic **Projektledare och redaktör:** Martin Linderå Nordström **Utgivare:** Linderå Group AB

Tillkomst

Denna bok är skapad i samarbete mellan mänskliga och AI. Texterna har genererats av Claude, en stor språkmodell utvecklad av Anthropic, genom ett arbetsflöde med specialiserade agenter:

- **Researcher** – utforskade AI-koncept på djupet
- **Translator** – hittade mänskliga motsvarigheter
- **Writer** – skrev kapiteltext
- **Editor** – granskade och förfinade
- **Fact-checker** – verifierade teknisk korrekthet

Martin Linderå Nordström agerade projektledare, redaktör och kreativ riktningsgivare genom hela processen.

Typografi

Brödtext: Crimson Pro **Rubriker:** Crimson Pro **Kod och tekniska termer:** Jet-Brains Mono

Crimson Pro är ett elegant serifftypesnitt skapat av Jacques Le Bailly, fritt tillgängligt via Google Fonts under SIL Open Font License.

Teknisk produktion

Boken är skriven i Markdown och konverterad till publiceringsformat med:

- **Pandoc** – dokumentkonvertering
- **XeLaTeX** – PDF-generering
- **Custom CSS** – HTML och ePUB-styling
- **GitHub Pages** – webbpublicering

Källkod och råmaterial finns tillgängliga på GitHub.

Licens

CC BY 4.0 – Creative Commons Attribution 4.0 International

Du får fritt: - **Dela** – kopiera och vidaredistribuera materialet - **Bearbeta** – remixa, transformera och bygga vidare

Under följande villkor: - **Attribution** – Du måste ge lämpligt erkännande till upphovspersonen

Fullständig licenstext: creativecommons.org/licenses/by/4.0

Kontakt

Frågor, synpunkter och förslag: martin@linderagroup.se

Bugrapporter och bidrag: github.com/linderagroup/monster-av-mening

Satt med omsorg om läsbarhet. Tryckt med elektricitet och statistik.