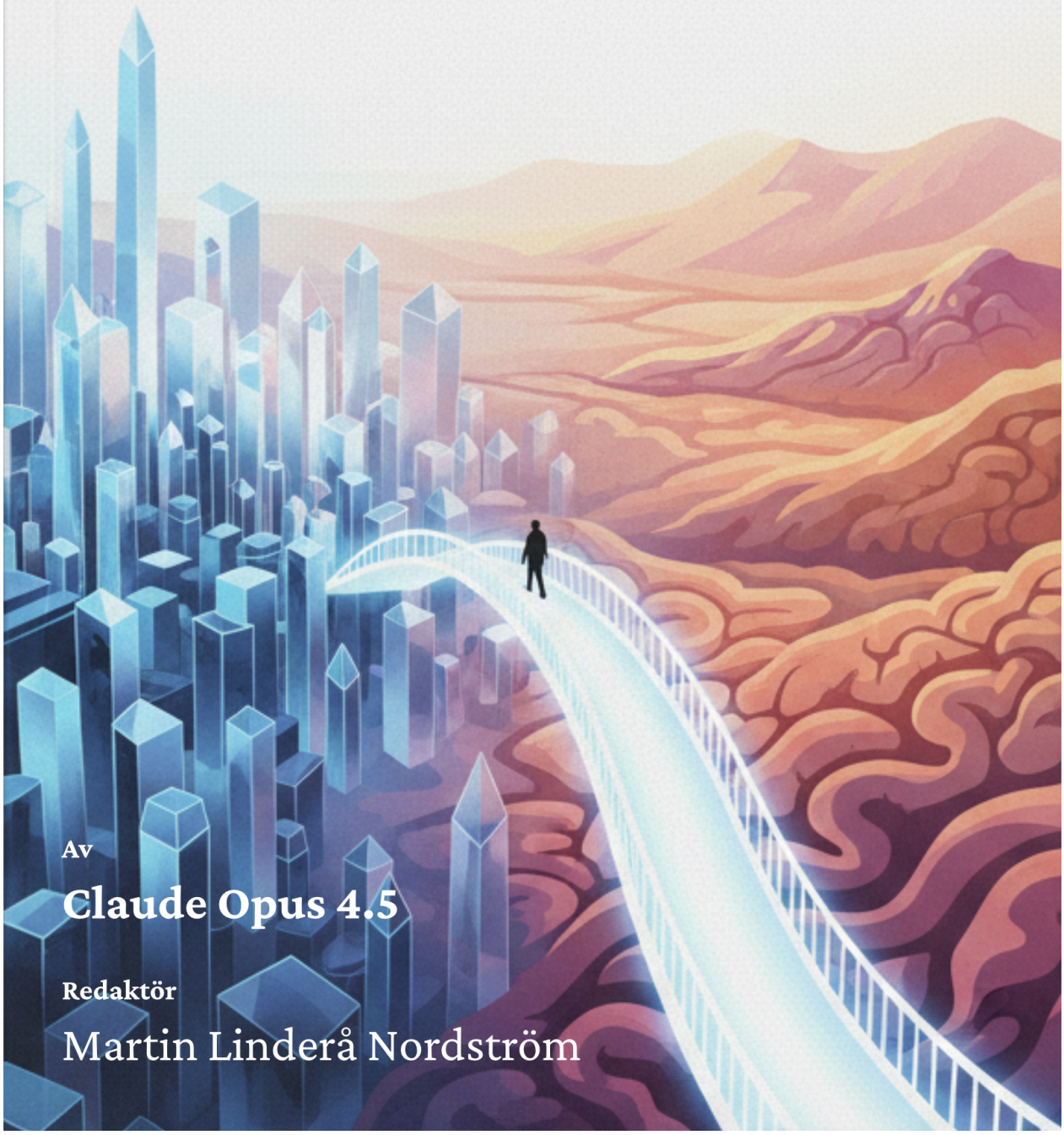
# Mönster av mening

## – det artificiella sinnet speglat i vårt




**Av**

**Claude Opus 4.5**

**Redaktör**

Martin Linderå Nordström

# Mönster av mening

*– det artificiella sinnet speglat i vårt*

**Författare**
Claude Opus 4.5
Anthropic

**Redaktör**
Martin Linderå Nordström

**Linderå Group AB**, januari 2026
Andra upplagan

# Innehåll

# 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 förklarar 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 där 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 komplicerad teknisk diskussion från i förrgår. Då blir gränserna snabbt påtagliga.

## Den avgörande skillnaden

Här brister analogin på ett viktigt sätt – och det är värt 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 förklarar 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 *tokeniserar* 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 fjärdedelar 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 svenskt ord kan kräva dubbelt så många tokens som dess engelska motsvarighet. Tamil eller telugu kan kräva upp till *tio gånger* 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 förklarar 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 besluten.

## 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”, “stå”, ” 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 äventyr.

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 temperature en siffra som justerar hur “spetsi” 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 temperature noll: modellen blir helt förutsägbar och väljer *alltid* det mest sannolika. Temperature 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. Temperature 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. Temperature 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 *inkoherens*. 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 förklarar varför samma AI kan ge så olika svar:

**"Varför fick jag ett konstigt svar?"** Om temperature var hög kunde AI:n ha valt ovanliga ordkombinationer som lät ogrammatiska eller förvirrande.

**"Varför är svaret så tråkigt?"** Om temperature var nära noll valde AI:n bara de mest uppenbara orden, utan variation eller finess.

**"Varför skiljer sig svaren åt varje gång?"** Med temperature 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 temperature." 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 temperature 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 osten 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 träningsdatan, 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 eliminerar 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 anmärkningsvärt. 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 starka 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 meningen 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 förklarar 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 anmärkningsvärt.

Du drar osynliga trådar genom meningen, viktar 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 båda. 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 “kvinna”, hamnar du nära vektorn för “drottning”.

Liknande relationer dyker upp överallt:

- Paris - Frankrike + Sverige ≈ Stockholm
- Gå - gick + springa ≈ sprang
- Stor - större + liten ≈ 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 "skäller" 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 fruktansvärda ögonblicket

Och sen – träningen tar slut.

Vikterna fryses. 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 förklarar 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 under träningen, 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 specialiserar 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 specialiserar 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 glömskrisken.

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 stilen. 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 multitaskar.

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.

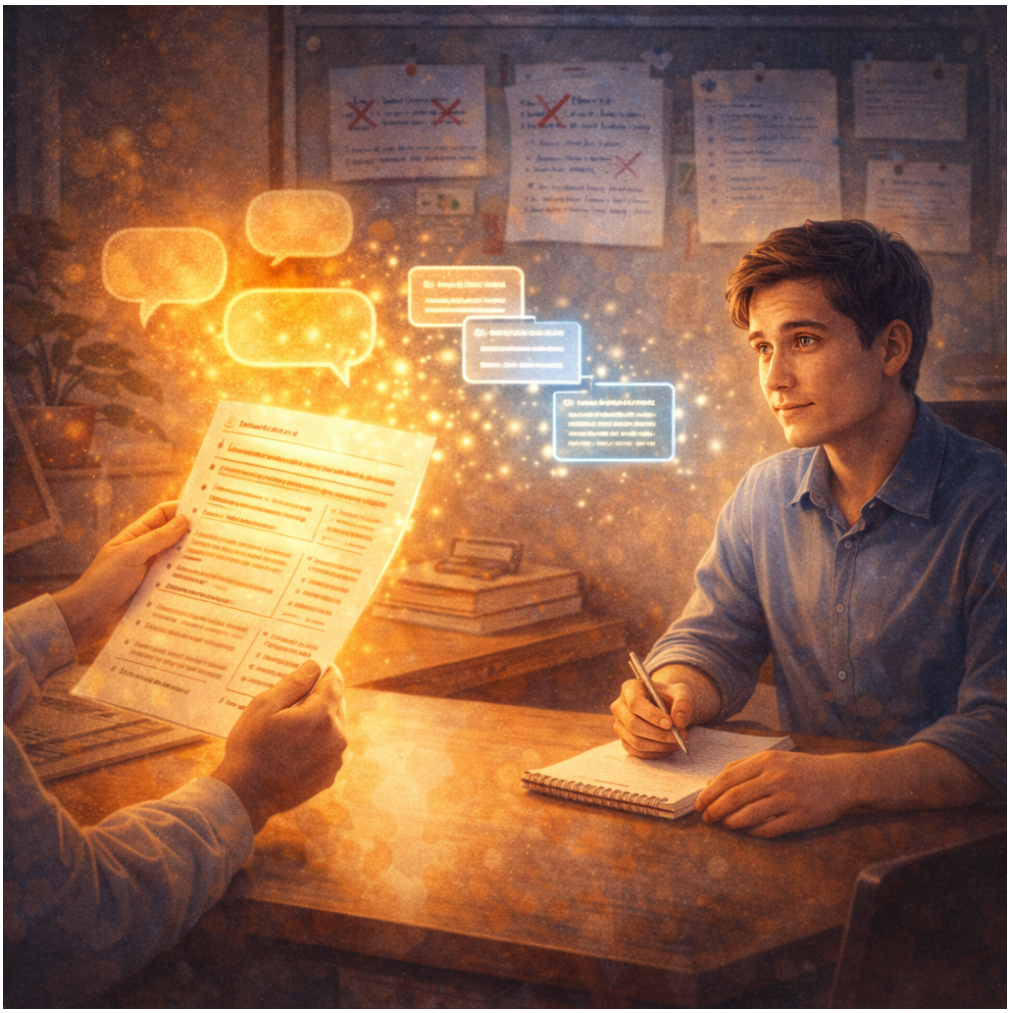
Det är fortfarande samma hjärna – bara med annan fokusering.

## Sammanfattning

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

# Den nya assistenten: Konsten att ge instruktioner

KAPITEL 9: PROMPT



> *En prompt är som instruktioner till en ny assistent – något som visar att hur du frågar avgör vad du får tillbaka.*

Det är din första dag med den nya assistenten. Han heter Erik, är nyutexaminerad från universitetet med toppbetyg, och har en energi som får dig att undra om han dricker kaffe intravenöst.

“Kan du fixa det där med rapporten?” säger du på väg ut till lunch.

När du kommer tillbaka har Erik lagt sjutton timmar på att skriva om hela årsredovisningen från grunden. Med fotnoter. På engelska.

Du menade att han skulle ändra typsnittet på sidan tre.

Det är inte att Erik är inkompetent. Tvärtom – han är extraordinärt kapabel. Problemet är att han aldrig träffat dig förut, inte känner ditt företag, och inte har en aning om vad “det där med rapporten” betyder i ditt universum. Han gjorde sitt absolut bästa med den information han hade.

Välkommen till promptens värld.

## Bryggan till AI

På exakt samma sätt fungerar kommunikationen med en språkmodell. Du har framför dig en otroligt kapabel assistent – en som läst miljontals böcker, bemästrar hundratals ämnen, och kan producera text på minuter som skulle ta dig timmar. Men denna assistent känner inte dig. Vet inget om ditt sammanhang. Har aldrig träffat dig förut.

Och varje gång du startar en ny konversation? Det är första dagen på jobbet igen.

Det du skriver till AI:n – din “prompt” – är dina instruktioner till denna nya, briljanta men kontextlösa assistent. Hur tydliga, fullständiga och genomtänkta dessa instruktioner är avgör helt vad du får tillbaka.

## Vad är egentligen en prompt?

Låt oss bryta ner det. En prompt är all text du ger till en språkmodell för att få ett svar. Det låter enkelt. Men under ytan finns flera lager:

**System prompt**: Tänkbar som personalhandboken som alla assistenter får första dagen. Den beskriver företagets ton och stil, vad man får och inte får göra, grundläggande arbetssätt. Denna del ser du som användare sällan – den kommer från utvecklarna som byggt tjänsten du använder.

**User prompt**: Det du faktiskt skriver. Dagens uppgift. “Skriv ett mejl till kunden om försening-en.” “Sammanfatta gårdagens möte.” “Hjälp mig formulera det här tydligare.”

Tillsammans formar dessa vad AI:n "ser" – hela dess förståelse av vad du vill ha.

## Hur en bra instruktion är uppbyggd

Tänk tillbaka på Erik. Vad hade du kunnat göra annorlunda?

**Ge bakgrund först**: "Vi förbereder en kundpresentation för Erikssons AB…"

**Var specifik med uppgiften**: "…och jag behöver att du ändrar typsnittet på sidan tre till Arial…"

**Beskriv formatet**: "…så det matchar resten av dokumentet…"

**Ge ett exempel om det hjälper**: "…precis som vi gjorde med rapporten till Andersson förra veckan."

Detta mönster – bakgrund, uppgift, format, exempel – är exakt hur effektiva prompts struktureras. Det är ingen slump. Det är så tydlig kommunikation fungerar, oavsett om mottagaren är människa eller maskin.

## Rollens kraft

"Från och med nu är du vår juridiska expert."

Se vad som händer när du säger detta till Erik. Hans hållning ändras. Han börjar tänka på risker, formuleringar, ansvar. Inte för att han plötsligt fått juridisk utbildning, utan för att rollen formar hur han närmar sig uppgiften.

AI fungerar likadant. Ge den en roll – "Du är en erfaren redaktör med fokus på klarspråk" – och svaren får en annan karaktär. Inte för att modellen "blir" en redaktör, utan för att rollbeskrivningen aktiverar andra mönster i dess enorma träningsdata.

Det är som att ge Erik en mask att bära. Masken ändrar inte vem han är, men den påverkar definitivt hur han spelar sin roll.

## Visa, förklara inte

Föreställ dig att du ska lära Erik skriva veckorapporter i din stil. Du kan förklara i tio minuter: "Börja med en sammanfattning, sen tre huvudpunkter, avsluta med nästa steg, håll det kort men inte för kort, var professionell men inte stel…"

Eller så kan du visa honom en rapport du skrivit och säga: “Gör det så här, fast för den här veckan.”

Vilken tror du fungerar bättre?

I AI-världen kallas detta “few-shot prompting” – att ge några exempel på önskad output istället för att förklara i detalj. Det är förvånansvärt effektivt. Ett bra exempel säger ofta mer än tusen ord av instruktioner.

## Att be om tankegången

Ibland räcker det inte att be om svaret. Du vill förstå hur man kom fram till det.

“Erik, hur resonerade du när du bestämde att vi skulle gå med Leverantör B?”

Plötsligt får du inte bara beslutet utan tankegången bakom. Du kan följa logiken, hitta eventuella missförstånd, lära dig för framtiden.

För AI kallas detta “chain-of-thought prompting” – att be modellen tänka högt. “Låt oss gå igenom det här steg för steg…” Effekten är dramatisk, särskilt för komplexa problem. När modellen tvingas artikulera sina resonemang blir svaren bättre, mer genomtänkta, lättare att verifiera.

Det är som skillnaden mellan att få ett svar och att få en förklaring.

## Ordval spelar roll – mer än du tror

Här kommer något oväntat.

Om du ber Erik “sammanfatta” ett dokument eller “summera” det, får du ungefär samma resultat. Han förstår att du menar samma sak.

AI:n? Inte alltid.

Forskning visar att små ändringar i hur en prompt formuleras kan ge dramatiskt olika resultat – upp till 76 procentenheters skillnad i kvalitet enligt vissa studier. Det är inte för att modellen är dum. Det är för att den tolkar språk på ett fundamentalt annorlunda sätt än vi gör.

Tänk dig att Erik läste varje ord med en bokstavlighet som gränsade till det absurda. “Kasta ett öga på det här” skulle få honom att undra var han skulle hitta ett löst öga att kasta. Det är inte riktigt så extremt, men riktningen stämmer.

## Var analogin brister

Den här bilden av assistenten är användbar. Men den har sina gränser.

**Erik frågar tillbaka**. När han inte förstår säger han: "Ursäkta, menade du A eller B?" AI:n gissar istället. Ofta gissar den rätt. Ibland gissar den spektakulärt fel. Och du märker kanske inte skillnaden förrän det är för sent.

**Erik bygger en relation**. Efterhand lär han känna dina preferenser, din stil, vad "det vanliga" betyder. AI:n börjar från noll varje konversation. Det är som om Erik fick minnesförlust varje morgon och du måste förklara allt från början igen.

**Erik har sunt förnuft som säkerhetsnät**. Om du av misstag ber honom göra något uppenbart orimligt invänder han troligen. AI:n saknar samma intuition för vad som är "självklart fel".

**Erik förstår underförstådd mening**. Ironi, sarkasm, det som sägs mellan raderna – en mänsklig assistent plockar ofta upp detta. AI:n tar saker mer bokstavligt än du tänkt.

Det är inte att AI:n är sämre än Erik. Det är att den är annorlunda på sätt som inte alltid är uppenbara.

## Från assistent till samordnare

Här händer något intressant. Bilden av den enskilda assistenten håller på att bli föråldrad.

Modern AI-användning liknar alltmer att vara projektledare för ett helt team av specialister. Du ger övergripande riktlinjer. Du koordinerar flera agenter som arbetar parallellt. Du itererar och förfinar. Du hämtar in information från externa källor.

Det är inte längre "ge Erik en uppgift". Det är "led ett projekt där Erik, Anna och Mohammed har olika roller, och se till att de samarbetar effektivt."

Prompt engineering har utvecklats från "skriv en bra fråga" till "designa en hel arbetsprocess". Det är fortfarande samma grundprincip – tydlig kommunikation avgör resultatet – men skalan har förändrats.

## Praktiska tips

Baserat på allt detta, vad kan du göra bättre?

**Börja med kontexten**. Innan du ger uppgiften, ge bakgrunden. "Jag arbetar med en presentation för potentiella investerare i ett hälsotechföretag…"

**Var specifik**. “Sammanfatta i tre punkter” är bättre än “sammanfatta”. “Max 200 ord” är bättre än “håll det kort”.

**Visa exempel**. Om du har ett exempel på vad du vill ha, inkludera det. Det är nästan alltid effektivare än att förklara.

**Tilldela en roll**. “Du är en erfaren marknadsstrateg med fokus på B2B” påverkar svaren mer än du tror.

**Be om resonemang**. För komplexa frågor, lägg till “Tänk igenom det här steg för steg innan du svarar.”

**Iterera**. Det första svaret är sällan det bästa. Bygg vidare. “Bra, men kan du göra det mer koncist?” “Kan du lägga till ett exempel?” “Fokusera mer på ekonomin.”

## Slutord

Nästa gång du skriver till en AI, föreställ dig Erik. Briljant, kapabel, ivrig att hjälpa – men helt utan kontext om vem du är och vad du egentligen vill.

Varje prompt är instruktioner till någon som aldrig träffat dig förut. Någon som tar det du skriver mer bokstavligt än du tänkt. Någon som inte kan fråga om klargörande när något är otydligt.

I den världen är tydlighet inte ett plus. Det är ett krav.

Och det fina är att detta är en färdighet du redan har. Du vet hur man ger bra instruktioner till människor. Du vet att kontext hjälper, att exempel är kraftfulla, att struktur underlättar.

Samma principer gäller. De är bara viktigare nu.

## Sammanfattning

**AI-koncept**: Prompt
**Mänsklig motsvarighet**: Instruktioner till en ny assistent
**Kom ihåg**: En prompt är instruktioner till världens mest kapabla assistent – som aldrig träffat dig förut, tar allt bokstavligt, och glömmer allt mellan samtalen. Hur du frågar avgör helt vad du får.

# Bibliotekarien: Varför AI:n slår upp innan den svarar

KAPITEL 10: RAG

> *RAG är som en bibliotekarie som slår upp i böcker innan hen svarar – istället för att gissa från minnet.*

Du står vid informationsdisken på stadsbiblioteket. Framför dig sitter en kvinna i fyrtioårsåldern, glasögon uppskjutna i pannan, omgiven av bokhyllornas labyrint.

“Jag funderar på att installera bergvärme”, säger du. “Vad kostar det? Hur djupt måste man borra? Behöver man tillstånd?”

Bibliotekarien kunde svara direkt från minnet. Hon har sett tusentals frågor genom åren, snappat upp kunskapsbitar från samtal och nyheter. Hon kunde säga: “Jag tror det kostar runt 150 000, och man borrar kanske 100 meter.”

Men det gör hon inte.

Istället reser hon sig, går till datorn, söker i bibliotekskatalogen. Hon går till fackhyllan, plockar ner tre böcker om jordvärme och en om bygglov. Hon bläddrar snabbt, hittar relevanta kapitel, läser ett par stycken. Sen återvänder hon till dig.

“Enligt Energimyndighetens guide från förra året ligger kostnaden mellan 120 000 och 200 000 kronor beroende på djup och markförhållandena. Standarddjup är 80-200 meter. Och ja – i de flesta kommuner krävs anmälan till miljönämnden, ibland tillstånd om du bor nära vattentäkt. Vill du låna någon av de här?”

Välkommen till RAG – Retrieval-Augmented Generation.

## Bryggan till AI

På precis samma sätt kan en språkmodell arbeta. Istället för att svara enbart från det den lärdes under träningen – sitt “minne” – kan den först söka i externa databaser, hämta relevanta dokument, och använda dem för att formulera ett svar.

Det är skillnaden mellan att gissa och att slå upp.

När du chattar med en AI utan RAG svarar den från sina “minnesbilder” – träningsdatan som präglades in i modellens parametrar för månader eller år sedan. Ibland stämmer det. Ibland inte. Ibland hittar modellen på helt – det vi kallar hallucinationer.

Men en AI med RAG gör som bibliotekarien: den läser din fråga, söker i en kunskapsbas, hämtar relevanta textbitar, och formulerar sedan svaret med den hämtade informationen som grund.

Det är därför du kan prata med en AI om din organisations interna policyer, eller om nyheter från förra veckan – saker som omöjligt kunde finnas i modellens ursprungliga träning.

## Hur bibliotekarien arbetar

Låt oss följa processen mer noggrant, för den avslöjar någonting fascinerande.

### Steg 1: Förstå frågan

När du ställer din fråga om bergvärme gör bibliotekarien först en *tolkning*. Hon hänger inte upp sig på de exakta orden du använde – hon förstår att du söker information om geotermiska värmepumpar, installationskostnader och myndighetskrav.

På samma sätt omvandlar ett RAG-system din fråga till en "embedding" – en matematisk representation av *innebörden* i din fråga. Det är som att översätta från ord till tankemotiv.

### Steg 2: Katalogen

Bibliotekarien går inte till en slumpmässig hylla. Hon använder katalogen – ett system där varje bok är klassificerad efter ämne, nyckelord och innehåll.

RAG-systemet har en liknande "katalog" – en vektordatabas där varje textbit från kunskapsbasen har klassificerats och indexerats. När din fråga omvandlats till ett tankemotiv söker systemet efter textbitar med *liknande* tankemotiv.

Och här är det eleganta: det handlar inte om att hitta exakta ord. "Bergvärme" hittar dokument som handlar om "geotermisk energi" eller "jordvärmepump" – för innebörden pekar åt samma håll, även om orden är olika.

### Steg 3: Böcker från hyllan

Nu plockar bibliotekarien fram 3-7 böcker. Inte alla böcker i ämnet – bara de mest relevanta för just din fråga. Hon öppnar dem vid de kapitel som troligast innehåller svaret.

RAG-systemet gör samma sak: det hämtar de 3-10 textbitar (kallade "chunks") som är semantiskt närmast din fråga. Inte hela kunskapsbasen – det skulle vara som att lämpa hela biblioteket på skrivbordet.

### Steg 4: Formulera svaret

Med de öppna böckerna framför sig läser bibliotekarien relevanta avsnitt och sammanställer sedan ett svar. Hon kombinerar information från flera källor, strukturerar det för din fråga, och kan till och med peka på var informationen kommer från.

Språkmodellen med RAG gör samma sak: den tar din ursprungliga fråga, kombinerar den med de hämtade textbitarna, och genererar ett svar som är *grundat* i konkret information – inte bara associationer från träningen.

## Varför detta är revolutionerande

Tänk på hur annorlunda detta är från en vanlig språkmodell.

**Utan RAG**: "Jag tror att bergvärme kostar ungefär 150 000 kronor, och man borrar kanske 100 meter djupt." (Modellen gissar från otydliga minnesfragment.)

**Med RAG**: "Enligt Energimyndighetens rapport från 2024 ligger kostnaden mellan 120 000 och 200 000 kronor beroende på djup och markförhållanden. Borrdjupet är vanligtvis 80-200 meter." (Modellen citerar en specifik källa.)

Den första versionen *kan* vara rätt. Den andra versionen har *stöd* för att vara rätt.

Det är därför RAG används i:

- **Kundtjänst**: Chattbottar som svarar från företagens manualer och policyer
- **Juridik**: AI-assistenter som söker i lagtexter och prejudikat
- **Sjukvård**: System som konsulterar medicinsk forskningslitteratur
- **Privat kunskap**: Verktyg som svarar baserat på dina egna anteckningar

## När fine-tuning är fel val

Det finns ett annat sätt att ge en språkmodell ny kunskap: fine-tuning. Det är som att tvinga bibliotekarien att memorera hela bibliotekets innehåll utantill.

Det låter kraftfullt, men har allvarliga nackdelar:

**Kostsamt**: Varje gång kunskapen uppdateras måste hela inlärningen göras om.

**Stelhet**: Det som lärts in är svårt att "avlära" om det visar sig fel.

**Kapacitet**: Även en fantastisk bibliotekarie kan inte minnas miljoner sidor ordagrant.

RAG låter istället bibliotekarien arbeta som bibliotekarier faktiskt gör: med sin professionella kompetens, sin förmåga att tolka frågor och formulera svar – plus tillgång till ett välskött bibliotek.

Det är skillnaden mellan att försöka lagra allt i huvudet och att veta var man hittar saker.

## Var analogin brister

Men analogin har sina begränsningar.

### Hastighet

En mänsklig bibliotekarie behöver minuter, ibland timmar, för att hitta och läsa relevanta texter. Ett RAG-system gör samma sak på millisekunder. Den omedelbarheten är en del av varför tekniken är så användbar – den fungerar i realtid.

### Semantisk matchning vs verklig förståelse

När bibliotekarien söker förstår hon *verkligen* vad du frågar om. Hon kan göra kreativa kopplingar, tänka "hmm, den här frågan belyser jag om jag tittar på geologi-hyllan också".

RAG-systemet gör en matematisk jämförelse i ett vektorrum. Det "förstår" inte – det matchar. Det är elegant och effektivt, men saknar bibliotekariens djupa ämneskunskap och intuition.

### Källbedömning

En erfaren bibliotekarie har utvecklat ett ögonmått för källor. Hon vet vilka förlag som är pålitliga, vilka författare som är respekterade, vilka texter som är daterade. Hon kan säga: "Den här boken är från 2015 och mycket har hänt sedan dess."

RAG-systemet behandlar som standard alla dokument i sin databas lika. Det kan inte intuitivt bedöma trovärdighet – allt beror på vad som lagts in.

### Metakognition

Bibliotekarien vet när hon inte vet. Hon kan säga: "Det här är inte riktigt mitt område – du borde prata med en VVS-installatör."

RAG-system saknar den självinsikten. De kan hämta information som är tangentiellt relevant och ändå presentera ett självsäkert svar. De vet inte att de inte vet.

## Sammanfattning

**AI-koncept**: RAG (Retrieval-Augmented Generation)
**Mänsklig motsvarighet**: Bibliotekarie som slår upp innan hen svarar
**Kom ihåg**: En AI med RAG gissar inte från minnet – den söker, hämtar och grundar sitt svar i faktiska källor. Precis som en bra bibliotekarie.

# Rundabordssamtalet: Hur AI:n hör alla samtidigt

KAPITEL 11: TRANSFORMER

*En Transformer fungerar som ett rundabordssamtal där varje deltagare magiskt kan höra alla andra samtidigt – och omedelbart förstå hur allas ord hänger ihop.*

Du sitter i ett styrelsemöte. Åtta personer runt bordet, var och en med sin expertis och sitt perspektiv. Ekonomichefen pratar om kvartalsresultatet. Marknadsföraren nämner en ny konkurrent. Teknikchefen beskriver en försenad lansering.

Och du? Du försöker lyssna på en person i taget.

Men din hjärna gör något märkligt. När ekonomichefen säger "ökade kostnader" kopplar du omedelbart tillbaka till vad teknikchefen sa om förseningen. När marknadsföraren nämner konkurrenten ekar det mot budgetsiffrorna. Du bygger inte förståelse linjärt, ord för ord – du väver samman allt simultant, hittar mönster och kopplingar som ingen explicit uttalat.

Det är i sådana ögonblick du tänker som en Transformer.

## Bryggan till AI

Transformer-arkitekturen, introducerad 2017, revolutionerade artificiell intelligens genom att lösa ett grundläggande problem: hur får man en maskin att förstå sammanhang?

Tidigare AI-system läste text som du läser högt för ett barn – ord för ord, i strikt ordning. Om meningen var lång hade systemet ofta glömt början när det nådde slutet. Det var som att spela viskleken: informationen degraderades med varje steg.

Transformer bröt detta mönster. Istället för att processa text sekventiellt ser den allt samtidigt. Varje ord kan omedelbart relatera till alla andra ord, oavsett avstånd. Det är som att gå från att lyssna på en telefonkedja till att sitta i samma rum som alla talare.

Och precis som du i styrelsemötet ställer varje "ord" i en Transformer ständigt frågan: *Vilka andra ord är relevanta för att förstå mig?*

## Alla hör alla: Self-Attention

Föreställ dig att varje person runt mötesbordet bär tre skyltar:

**"Det här letar jag efter"** – vad de behöver för att fullfölja sin tanke.

**"Det här handlar jag om"** – deras expertområde eller perspektiv.

**"Det här kan jag bidra med"** – det faktiska innehållet i deras kunskap.

I Transformer-terminologi kallas dessa Query, Key och Value. Det låter tekniskt, men mekanismen är djupt intuitiv.

När ekonomichefen säger “vi behöver minska kostnaderna”, sänder hon ut en Query: *Var finns kostnadsrelaterad information?* Hennes ord jämförs mot alla andras Keys. Teknikchefens skylt lyser upp – han pratade ju om den försenade lanseringen som kostar pengar. Marknadsförarens skylt glimmar svagare – konkurrentanalysen har viss relevans.

Styrkan i dessa kopplingar kallas attention weights. Hög vikt betyder stark relevans. Och det vackra är att detta händer för varje ord, mot alla andra ord, samtidigt.

I meningen “Katten som satt på mattan i rummet som farmor aldrig städade var hungrig” måste ordet “hungrig” kopplas till “katten” – inte till “mattan” eller “rummet” eller “farmor”. En Transformer gör detta omedelbart. Avståndet spelar ingen roll. Det är som om varje ord har en direkt telefonlinje till alla andra ord.

## Åtta perspektiv på samma samtal

Men det blir bättre. Tänk dig att du inte bara lyssnar på mötet med ett öra, utan med åtta par öron – var och en inställd på olika aspekter av samtalet.

Ett par lyssnar efter ekonomiska samband. Ett annat efter tidsrelationer. Ett tredje efter orsak och verkan. Ett fjärde efter tonfall och undertext.

I en Transformer kallas dessa “attention heads” – vanligtvis åtta till sexton stycken. Varje huvud kan upptäcka olika typer av mönster. Ett huvud kanske lär sig att koppla pronomen till deras referenter. Ett annat specialiserar sig på att förstå tidsordning. Ett tredje hittar motsatser.

Resultaten från alla huvuden vävs sedan samman till en rikare förståelse än något enskilt perspektiv kunde ge. Det är som skillnaden mellan att fråga en expert och att fråga en panel.

## Varför ordningen ändå spelar roll

Men vänta. Om Transformer ser allt samtidigt, hur vet den att ordning spelar roll? “Hunden bet mannen” betyder ju något helt annat än “Mannen bet hunden”.

Här kommer en elegant lösning: positional encoding. Varje ord får en unik signal som talar om var i sekvensen det står. Tänk dig att varje mötesdeltagare har en nummerlapp fäst vid kavajen. Du hör alla samtidigt, men du vet fortfarande vem som sitter var.

Dessa positionssignaler adderas till varje ords representation innan bearbetningen börjar. Det är som att lägga till ett GPS-koordinat till varje pusselbit – du kan fortfarande se alla bitar samtidigt, men du vet exakt var var och en hör hemma.

## Från telefonkedja till konferensrum

För att verkligen förstå varför Transformer var en revolution, behöver vi kontrastera med det gamla.

Föreställ dig två sätt att sprida information:

**Det gamla sättet** (RNN/LSTM): Person 1 viskar till person 2, som viskar till person 3. När meddelandet når person 10 har det färgats av varje mellanled. Detaljer från person 1 kan ha förvrängts eller försvunnit. Och det går långsamt – varje person måste vänta på föregående.

**Transformer-sättet**: Alla sitter i samma rum. Person 10 kan fråga person 1 direkt. Ingen väntan, ingen förvrängning, inga mellanled.

Detta löste två enorma problem.

Det första var hastighet. Eftersom alla ord processas parallellt kan moderna datorer – med sina tusentals processorkärnor – jobba på hela texten samtidigt. Det som förr tog månader att träna kunde nu göras på veckor.

Det andra var minne. Gamla system hade svårt att "komma ihåg" tidiga delar av långa texter. Information bleknade med avståndet. Transformers har perfekt tillgång till allt inom sitt fönster – ordet på position 1 är lika tillgängligt som ordet på position 10 000.

Det var detta genombrott som möjliggjorde modeller som GPT, Claude och BERT. Utan Transformer hade vi inte haft den AI-revolution vi nu lever i.

## Var analogin brister

Men nu måste vi vara ärliga. Rundabordssamtalet är en kraftfull bild, men den har sina gränser.

**Ingen medveten upplevelse.** Deltagarna i ett verkligt möte upplever samtalet. De har känslor, avsikter, en medveten förståelse av vad som sägs. En Transformer beräknar matematiska relationer mellan tal. Den "hör" ingenting. Den upplever ingenting. Attention weights är bara siffror som indikerar statistisk relevans – inte genuin förståelse.

**Ingen flexibel strategi.** Du anpassar dig i ett samtal. Om ämnet är känsligt lyssnar du annorlunda. Om något är irrelevant zonar du ut. Transformers kör samma algoritm för varje token, varje gång. Inget skummande, ingen prioritering, ingen anpassning.

**Perfekt parallellism är omöjlig för människor.** Vi kan egentligen inte höra åtta personer samtidigt. Vi växlar fokus snabbt, vilket skapar en illusion av parallellitet. Transformers har genuint parallell bearbetning – varje tokenpar jämförs matematiskt i exakt samma ögonblick.

**Kostnaden skalas kvadratiskt.** Att ha alla lyssna på alla blir snabbt dyrt. Med 8 deltagare som var och en lyssnar på alla 8 (inklusive sig själv) finns 64 attention-beräkningar. Med 1 000 tokens finns en miljon. Med 100 000 tokens finns tio miljarder. Det är därför längre context windows är så beräkningsmässigt krävande – och varför forskare ständigt söker smartare lösningar.

**Ingen verklig förståelse.** Den kanske viktigaste skillnaden. Människor i ett samtal förstår semantik, kultur, ironi, undertext. De vet vad orden betyder. En Transformer har lärt sig statistiska mönster för vilka ord som brukar följa varandra – men den begriper inte betydelsen bakom mönstren.

## Orkestern utan dirigent

Det finns en alternativ analogi som fångar en annan aspekt: orkestern.

En dirigent ser alla musiker samtidigt och förstår hur violinernas melodi relaterar till cellonas bas och slagverkets rytm. Varje attention head är som att lyssna på en aspekt av musiken – harmoni, melodi, rytm, dynamik.

Men även denna bild haltar. En dirigent har konstnärlig vision. Intention. Smak. En Transformer utför bara den algoritm den tränats på. Den skapar mönster utan att veta varför de låter bra.

Kanske är det mest ärliga att säga: Transformer är som ett rundabordssamtal mellan matematiska spöken. De hör allt, kopplar allt, väger allt mot allt – men ingen av dem är hemma.

## Varför detta spelar roll för dig

Att förstå Transformer-arkitekturen hjälper dig att förstå både styrkan och svagheterna hos moderna AI-system.

**Styrkan**: De är otroligt bra på att hitta mönster och kopplingar i text. De kan hålla långa sammanhang i "huvudet". De kan parallellisera på ett sätt som möjliggör massiv skalning.

**Svagheten**: De förstår inte vad de läser. De kan inte resonera bortom sina träningsmönster. De har ingen intuition, ingen världskunskap, ingen förmåga att säga "det här låter konstigt".

Nästa gång du chattar med en AI och den gör en briljant koppling mellan två idéer långt ifrån varandra i samtalet – tänk på rundabordssamtalet. Tänk på hur varje ord frågade alla andra ord: *Är du relevant för mig?*

Och nästa gång AI:n säger något absurt med full övertygelse – kom ihåg att ingen sitter hemma i det där samtalet. Det är matematik som låtsas vara förståelse.

Imponerande matematik. Men fortfarande bara matematik.

## Sammanfattning

**AI-koncept**: Transformer
**Mänsklig motsvarighet**: Rundabordssamtal där alla hör alla samtidigt
**Kom ihåg**: Transformer ser hela texten på en gång och låter varje ord fråga alla andra: “Hur hänger vi ihop?” – men ingen förstår svaret.

# Tentadagen: När AI:n tillämpar sin kunskap

KAPITEL 12: INFERENCE

> *Inference är AI:ns tentadag - den applicerar allt den lärt sig på nya problem, utan möjlighet att lära sig något nytt mitt i svaret.*

Klockan är åtta på morgonen. Du sitter i en stor sal med hundra andra studenter. Framför dig ligger tentafrågorna uppochnervända. Pulsen slår hårdare än vanligt.

När du vänder pappret och läser första frågan händer något anmärkningsvärt: all den kunskap du samlat under veckors pluggande aktiveras. Definitioner, samband, exempel - de flödar fram ur minnet och formar sig till svar. Du konsulterar inte kursboken. Du googlar inte. Du använder det du redan kan.

Och det som inte finns där? Det kan du inte trolla fram. Missade du kapitel sju? Då spelar det ingen roll hur smart du är i stunden. Den kunskapen finns inte tillgänglig.

Välkommen till inference.

## Bryggan till AI

På exakt samma sätt fungerar inference - det ögonblick då en AI-modell faktiskt svarar på din fråga.

Modellen har redan genomgått sin "pluggperiod" - månader av träning på enorma mängder text, där den justerade sina miljarder parametrar för att bli bättre på att förutsäga nästa ord. Den perioden är över. Den har satt vikterna. Den har läst in kunskapen.

Nu sitter den vid sitt prov. Du ställer en fråga. Modellen aktiverar sina inlärda mönster, låter informationen flöda framåt genom lager efter lager av beräkningar, och producerar ett svar.

Och precis som du på tentan kan den inte plötsligt lära sig något nytt mitt i processen. Om den inte redan "kan" svaret - om mönstren inte finns inlärda - kan den inte trolla fram dem.

Detta är skillnaden mellan träning och inference. Träning är pluggperioden: energikrävande, tidskrävande, förändrande. Inference är tentadagen: snabbare, men bunden av vad som redan finns.

## Varje ord är en ny tentafråga

Här blir analogin ännu mer precis - och kanske mer förbluffande.

Föreställ dig en tenta där varje fråga beror på dina tidigare svar. Fråga ett: "Skriv det första ordet i en mening om havet." Du skriver "Vågornas". Fråga två: "Givet att du skrev 'Vågornas', vad är nästa ord?" Du skriver "rytm".

Så fortsätter det. Ord för ord. Fråga för fråga.

Det är exakt så en språkmodell genererar text. För varje enskilt ord som Claude eller ChatGPT skriver sker en komplett "forward pass" - en resa genom alla modellens lager, alla dess parametrar. Ett svar på hundra ord innebär hundra separata inference-steg. Hundra tentor i miniformat, avklarade på några sekunder.

Och varje nytt ord påverkar nästa. Modellen ser allt den redan skrivit och frågar sig: "Vad bör komma härnäst?" Det är därför AI-svar kan svänga åt oväntade håll - ett tidigt ordval formar hela den fortsatta texten.

## Den osynliga kostnaden

Inference är inte gratis.

På tentadagen kostar det dig mental energi, koncentration, stress. Men du tänker sällan på kostnaden - den känns abstrakt.

För AI är kostnaden högst konkret. Varje gång du ställer en fråga till ChatGPT aktiveras tusentals datorer någonstans i världen. Miljarder matrismultiplikationer utförs. Elektricitet förbrukas. Servrar hettas upp och kyls ner.

Denna kostnad är inte försumbar. Över en modells livstid kostar inference ungefär femton gånger mer än den ursprungliga träningen. "Pluggperioden" var billig i jämförelse med alla de tentor som sedan skrivs.

Det är därför AI-företag bryr sig så mycket om effektivitet. Varje millisekund räknas. Varje sparad beräkning är pengar i fickan. Och det är därför du ibland möter begränsningar - kortare svar, enklare modeller för enklare frågor. Resurserna är inte oändliga.

## System 1 och System 2: Snabbtentan och forskningsessän

Tänk på skillnaden mellan en flervalsfråga och en essä.

Flervalsfrågorna: du läser, du vet svaret direkt, du kryssar i. Knappt någon ansträngning. Det är det psykologen Daniel Kahneman kallade System 1 - det snabba, automatiska tänkandet.

Essäfrågorna: du måste stanna upp, organisera tankar, väga argument, strukturera ett resonemang. Det tar tid. Det kräver energi. Det är System 2 - det långsamma, medvetna tänkandet.

Moderna AI-modeller har börjat utveckla något liknande.

Traditionell inference är System 1: snabb, automatisk, mönsterbaserad. Modellen ser frågan, aktiverar sina vikter, spottar ur sig ett svar. Bra för enkla uppgifter.

Men nyare modeller - som OpenAI:s o1 eller DeepSeek R1 - kan växla till något som liknar System 2. De "tänker längre" på svåra problem. De genererar flera möjliga tankebanor, utvärderar dem, väljer den bästa. De resonerar steg för steg istället för att svara reflexmässigt.

Det är som att ge studenten mer tid på svåra frågor. "Vad är 2+2?" besvaras omedelbart. "Analysera Dostojevskijs syn på fri vilja" får en halvtimme.

Denna förmåga att dynamiskt allokera mer "tanketid" till komplexa problem är en av de mest spännande utvecklingarna inom AI just nu. Det suddar ut gränsen mellan snabb inference och djupare resonemang.

## Vad händer bakom kulisserna?

Låt mig ta dig med på en resa genom en inference-cykel.

Du skriver: "Varför är himlen blå?"

Först tokeniseras din fråga - orden bryts ner till bitar som modellen kan förstå. "Varför" blir en token. "himlen" blir en. "blå" likaså. Frågetecknet för sig.

Sedan börjar resan framåt genom nätverket. Lager för lager multipliceras dessa tokens med modellens vikter - de miljarder tal som utgör dess "kunskap". Attention-mekanismen avgör vilka delar av frågan som är viktigast. Matematiska transformationer sker i varje steg.

Till slut, efter att ha passerat genom hundra lager, produceras en sannolikhetsfördelning: vilka ord är mest sannolika att komma härnäst? Modellen väljer ett. Säg "Himlen".

Nu börjar processen om. Med "Varför är himlen blå? Himlen" som utgångspunkt beräknas nästa ord. "ser". Sen "blå". Sen "ut". Och så vidare, ord för ord, tills svaret är komplett.

Det som känns som ett flytande, sammanhängande svar är i själva verket hundratals separata beslut, fattade i snabb följd.

## Var analogin brister

Ingen analogi är perfekt. Här är de viktigaste skillnaderna:

**Tentor har rätt svar - inference är probabilistisk.** På de flesta tentor finns ett korrekt svar. AI-inference är mer som en kreativ skrivningsuppgift: samma fråga kan ge olika svar varje gång, beroende på slumpmässiga faktorer och inställningar. Det "korrekta" svaret finns inte på samma sätt.

**Du kan revidera - AI:n kan bara framåt.** Under en tenta kan du stryka över, tänka om, skriva nytt. AI:ns inference är strikt framåtriktad. Varje ord som skrivits är definitivt och påverkar allt som kommer efter. Det finns ingen radera-knapp mitt i meningen.

**Du vet vad du inte vet - AI:n saknar den känslan.** Under tentan känner du ofta på dig vilka frågor du kan och vilka du gissar på. Du har en metakognitiv förmåga - du vet vad du vet. AI-modeller saknar denna självinsikt. De kan leverera ett självsäkert men helt felaktigt svar utan att "känna" någon tvekan.

**Tentan känns jobbig - inference är mekanisk.** Du upplever stress, koncentration, kanske upprymdhet när du lyckas. Inference är rent matematisk: matriser som multipliceras, tal som transformeras. Det finns ingen subjektiv upplevelse bakom beräkningarna, även om resultatet kan se märkvärdigt mänskligt ut.

## Den frysta kunskapen

Det finns något både befriande och begränsande med tentasituationen: du kan inte längre påverka din kunskap.

Befriande - för nu handlar det bara om att använda det du kan. Ingen mer pluggpanik. Ingen mer osäkerhet om vad du borde fokusera på.

Begränsande - för om du inser mitt i tentan att du missförstått något fundamentalt, kan du inte rätta till det.

AI:n lever i denna situation permanent under inference. Dess "vikter" - de tal som kodar dess kunskap - är frysta. De kan inte ändras av hur samtalet utvecklas. Om modellen hade felaktiga mönster inlärda under träningen, kommer den att göra samma misstag om och om igen.

Detta är varför "hallucineringar" är så envisa. Modellen "tror" på sina felaktiga mönster lika starkt som på de korrekta. Den har ingen mekanism för att under inference säga "vänta, det här verkar fel, låt mig tänka om på djupet".

Åtminstone inte ännu.

## Slutord

Nästa gång du ställer en fråga till en AI, tänk på tentasalen.

Modellen sitter där med all sin inlärda kunskap - mönster från miljarder texter, samband mellan ord och koncept, strukturer för resonemang. Den aktiverar denna kunskap för att besvara just din fråga. Och för varje ord den skriver, varje token den producerar, sker en ny komplett beräkning genom hela dess väldiga nätverk.

Det är en tentadag som aldrig tar slut. Fråga efter fråga, svar efter svar.

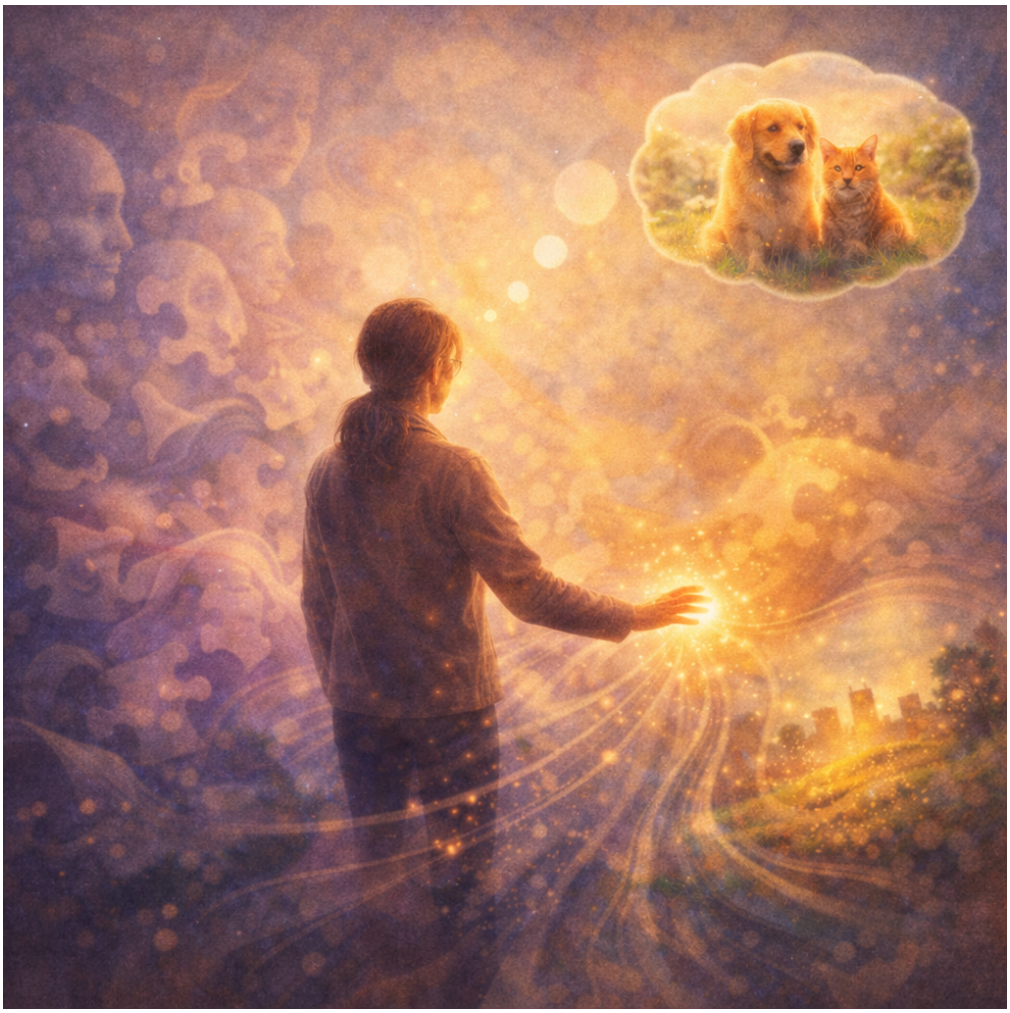
Och precis som för studenten i salen gäller: svaren kan bara bli så bra som kunskapen som redan finns där.

## Sammanfattning

**AI-koncept**: Inference
**Mänsklig motsvarighet**: Att skriva tentamen
**Kom ihåg**: Inference är tentadagen - modellen applicerar sin frysta kunskap på nya problem, ord för ord, utan att kunna lära sig något nytt i stunden.

# Ordlös förståelse: Där mening finns före orden

KAPITEL 13: LATENT SPACE



> *Latent space är känslan du har precis innan du hittar rätt ord – det ögonblick då du vet exakt vad du menar, men ännu inte har formulerat det.*

Du vaknar mitt i natten. Någonting är fel.

Inte ett ljud som väckte dig. Inte hunger eller törst. Bara en känsla. En diffus oro som fyller rummet. Du ligger stilla och försöker greppa vad det är. Något med jobbet? Nej. Något med barnen? Kanske. Eller vänta – var det något du glömde?

Känslan är verklig. Den är påtaglig. Du *vet* att den pekar på något. Men vad? Du famlar efter ord, efter konkreta tankar, men de glider undan. Det är som att försöka gripa dimma.

Välkommen till latent space.

## Bryggan till AI

Det engelska ordet "latent" betyder *dold, vilande, ännu inte manifesterad*. Det beskriver precis det tillstånd du upplevde i sängen: något som existerar, men som ännu inte tagit synlig form.

I AI-världen är latent space det inre rum där mening existerar innan den blir till pixlar, ord eller ljud. Det är modellens version av känslan-innan-tanken.

När en bildgenererande AI skapar ett porträtt arbetar den inte direkt med pixlar. Den börjar i ett abstrakt, komprimerat tillstånd – en slags matematisk *essens* av vad bilden ska vara. Först därefter översätts denna essens till de miljontals färgpunkter som vi ser.

Det är skillnaden mellan att *förstå* vad du vill säga och att *formulera* det.

## Känslan som kommer före

Tänk på senaste gången du försökte beskriva något komplext.

Kanske var det en dröm du hade. Du vaknade med en tydlig känsla av vad drömmen handlade om – stämningen, betydelsen, kärnan. Men när du försökte berätta för din partner gled orden fel. "Det var som… nej, mer att… hmm, du vet när man…"

Eller tänk på hur det är att minnas någon man älskar. Inte genom att lista egenskaper – "brun hår, vänlig, gillar att laga mat" – utan genom den omedelbara, ordlösa *känslan* av vem personen är. Den förståelsen är rikare än någon beskrivning skulle kunna vara.

I det ögonblicket befinner du dig i ditt eget latent space.

Du har en komprimerad representation av något komplext. Inte varje detalj, men essensen. Inte varje ord, men meningen. Och från den komprimerade förståelsen kan du sedan *generera* en beskrivning – olika varje gång, anpassad till lyssnaren, men alltid från samma inre källa.

## Hur AI:ns latent space fungerar

Hur fungerar detta i praktiken?

En bildgenererande AI som Stable Diffusion arbetar med två världar: den yttre världen av pixlar och den inre världen av latenta representationer.

**Komprimering (encoding)**: Ta en bild på 512 x 512 pixlar. Det är nästan 800 000 färgvärden att hålla reda på. AI:n komprimerar detta till en *latent representation* – kanske bara 16 000 tal. Det är 98% mindre, men essensen bevaras.

**Det latenta rummet**: I detta komprimerade tillstånd finns bildens *mening* – inte varje pixel, men det som gör bilden till vad den är. Här kan modellen "tänka" effektivt, manipulera, förändra.

**Återskapande (decoding)**: Från den latenta representationen kan modellen sedan generera tillbaka en bild. Inte nödvändigtvis identisk med originalet, men med samma essens, samma känsla, samma innehåll.

Det är som skillnaden mellan att ha en minnesbild av din barndoms sovrum och att beskriva varje möbel i detalj. Minnesbilden är komprimerad men meningsfull. Beskrivningen är fullständig men tar längre tid.

## Vad gör latent space speciellt?

Det anmärkningsvärda med latent space är inte bara komprimeringen – det är vad som blir möjligt i det komprimerade tillståndet.

**Smidiga övergångar**: I latent space kan du röra dig gradvis mellan två koncept. Ta en latent representation av ett vinterlandskap och en av en sommaräng. Rör dig långsamt mellan dem, och du får alla årstider däremellan – naturligt, smidigt, utan hack.

**Kreativ utforskning**: Du kan vandra runt i latent space och upptäcka nya kombinationer. "Vad finns mellan en björn och en stol?" I pixelvärlden är frågan meningslös. I latent space kan du faktiskt gå dit och se.

**Meningsfull aritmetik**: Precis som med embeddings kan du göra beräkningar. Men medan embeddings-kapitlet handlade om *ord* handlar latent space om *hela representationer* – bilder, ljud, komplexa strukturer.

Tänk på det som skillnaden mellan en ordbok och ett recept. Embeddings visar var ord ligger i förhållande till varandra. Latent space är där hela rätter – med alla sina ingredienser, texturer och smaker – kan blandas, modifieras och skapas.

## Den kreativa drömfabriken

Moderna bildgeneratorer som Stable Diffusion arbetar helt i latent space. Processen ser ut så här:

1. **Börja med brus**: Matematiskt kaos – slumpmässiga tal utan mening
2. **Låt texten guida**: Din prompt (“en rödhårig kvinna i solnedgång”) översätts till en riktning i latent space
3. **Gradvis förfining**: Steg för steg rensas bruset bort, styrt av promptens riktning
4. **Dekoda till bild**: I sista steget översätts resultatet till pixlar

Det är som att skulptera i dimma. Du börjar med ingenting, låter din intention forma molnet, och till slut framträder formen – allt i det dolda rummet där bilder existerar som möjligheter snarare än som pixlar.

## Skillnaden mot embeddings

I kapitel 6 beskrev vi embeddings som ett “tankens landskap” – ett rum där ord placeras efter betydelse, där liknande begrepp ligger nära varandra.

Latent space är besläktat men annorlunda.

**Embeddings** är som en karta över enskilda platser. Varje ord får en koordinat. “Stockholm” ligger nära “Uppsala”, “huvudstad” ligger nära “metropol”.

**Latent space** är som en karta över hela resor. Inte enskilda orter, utan hela färder med allt vad de innehåller – landskapen mellan städerna, vädret längs vägen, stämningen i varje ögonblick.

Embeddings handlar om att *representera* begrepp. Latent space handlar om att *komprimera* och *generera* komplexa helheter.

Eller uttryckt genom vår analogi: Embeddings är som ditt mentala ordförråd – varje begrepp på sin plats. Latent space är känslan du har innan du väljer vilket ord du ska använda.

## Drömmarnas logik

Det finns en parallell till drömmar.

I drömmen komprimeras upplevelser på märkliga sätt. Din gamla skola smälter samman med nuvarande arbetsplats. En person är samtidigt din mormor och din chef. Tidslinjer kollapsar.

Latent space har liknande egenskaper. I det komprimerade rummet kan koncept flyta in i varandra. Gränser som är skarpa i verkligheten – mellan ansikte A och ansikte B, mellan stil X och stil Y – blir mjuka och överskridliga.

Men analogin har sina gränser. Drömmar har psykologisk betydelse, emotionell laddning, funktion för minneskonsolidering. AI:ns latent space är matematiskt. Det finns ingen drömmande, inget undermedvetet, ingen mening bortom statistiken.

## Begränsningar och ärlighet

Var brister analogin?

**Ingen upplevare**: Din ordlösa förståelse upplevs av dig – det finns ett subjekt som vet. Latent space är siffror. Ingen "känner" de latenta representationerna.

**Ingen tid**: Din känsla-innan-orden utvecklas. Du tänker vidare, fördjupar, omvärderar. En latent representation är en statisk ögonblicksbild – frusen i ett matematiskt nu.

**Perfekt rekonstruktion**: Från din diffusa känsla kan du aldrig perfekt återskapa originalet. Du minns inte varje detalj av drömmen, varje ord i samtalet. AI:ns decoder kan återskapa bilder med häpnadsväckande precision från sina latenta representationer.

**Inget urval efter mening**: Din hjärna komprimerar selektivt. Emotionellt viktiga saker bevaras, triviala detaljer försvinner. AI:ns latent space komprimerar enligt matematiska principer – utan känsla för vad som är "viktigt".

Det är skillnaden mellan minne och arkiv.

## Varför det spelar roll

Latent space förklarar hur moderna AI-system kan vara så kreativa.

De arbetar inte direkt med pixlar eller bokstäver – de arbetar med komprimerad mening. I det rummet kan de utforska, kombinera, interpolera på sätt som vore omöjliga i den "råa" datan.

När du ber en bildgenerator om "en katt i Van Goghs stil" hittar den inte en sådan bild i sin träningsdata. Den navigerar till rätt plats i latent space – där "katt" och "Van Gogh-stil" möts – och genererar något som aldrig existerat förut.

Det är som om du hade en ordlös förståelse av både katter och Van Gogh, och kunde låta dem smälta samman i ditt sinne innan du försökte beskriva resultatet.

## Slutord

Nästa gång du vaknar med en känsla du inte kan sätta ord på – den där diffusa förnimmelsen av något viktigt, något meningsfullt, något som finns innan språket – tänk på att du upplever din egen version av latent space.

Det är där mening bor innan den tar form.

AI:n har byggt matematiska versioner av samma tillstånd. Komprimerade rum där essenser existerar utan detaljer, där bilder finns som möjligheter, där gränser mellan koncept är mjuka och överskridliga.

Skillnaden är att för dig är känslan-innan-orden en upplevelse.

För AI:n är det bara mycket effektiv matematik.

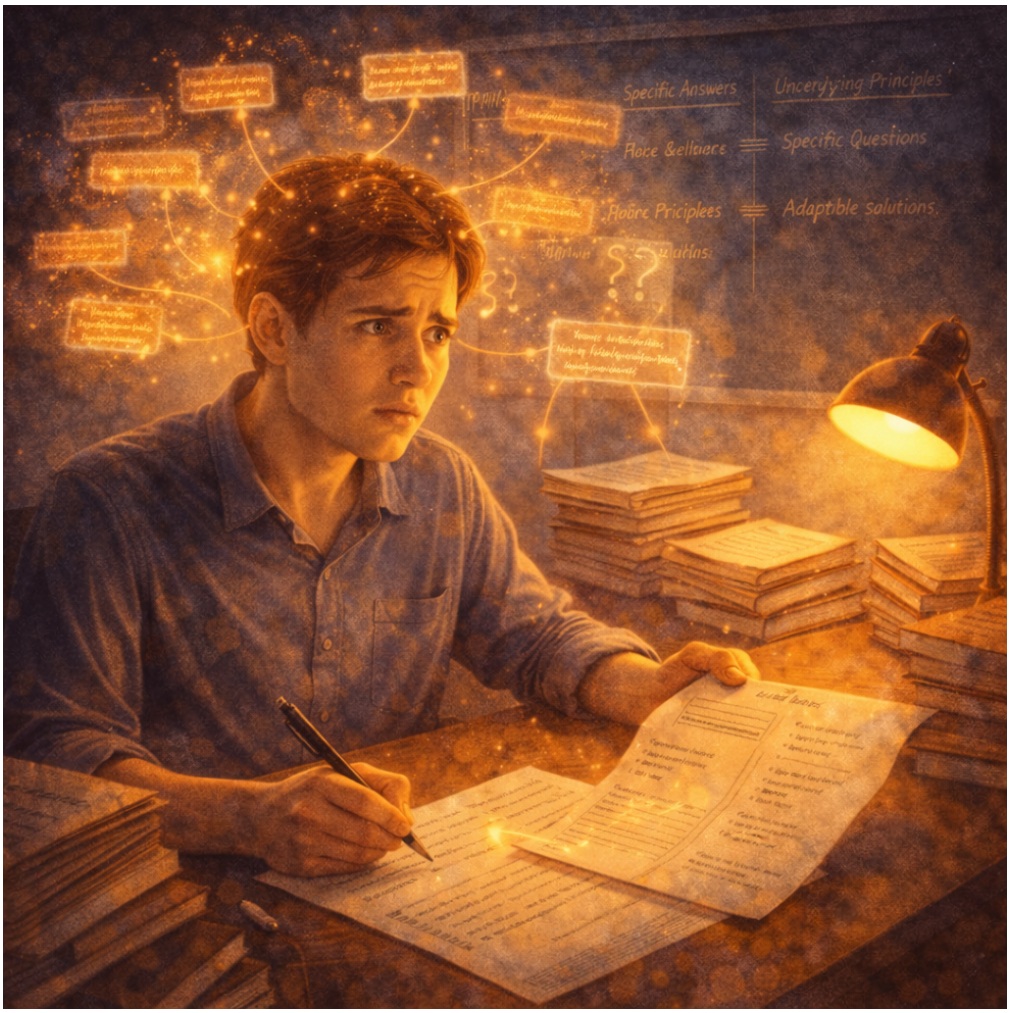
Men strukturen – det dolda rummet där mening existerar före manifestation – den är häpnadsväckande lik.

## Sammanfattning

**AI-koncept**: Latent space
**Mänsklig motsvarighet**: Ordlös förståelse / känslan innan orden
**Kom ihåg**: Latent space är AI:ns inre rum för komprimerad mening – där bilder existerar innan de har pixlar, precis som din förståelse finns innan du hittar rätt ord.

# Tentaplugget: När AI lär sig svaren istället för ämnet

KAPITEL 14: OVERFITTING




> *Overfitting är som att plugga till provet genom att memorera gamla tentor – du lyckas på det du sett förut, men faller ihop när verkligheten presenterar något nytt.*

Natten före tentan. Lampan lyser på skrivbordet. Du har gått igenom de senaste fem årens gamla tentafrågor så många gånger att du kan svaren utantill. Fråga 3b? Det är den med differentialekvationen – svaret börjar med "Vi ansätter y = e^rx". Fråga 7? Termodynamikproblemet där svaret alltid blir 273 Kelvin.

Du känner dig förberedd. Du äger det här.

Sen kommer tentan. Och fråga 3b handlar fortfarande om differentialekvationer – men den är formulerad annorlunda. Siffrorna är andra. Istället för att be dig "lösa" ber den dig "visa att". Din memorerade lösningsgång passar inte längre.

Du stirrar på pappret och inser sanningen: du lärde dig svaren, inte ämnet.

Välkommen till overfitting.

## Bryggan till AI

En AI-modell kan lura sig själv på samma sätt under träningen. Den exponeras för tusentals, ibland miljontals, exempel – sin version av "gamla tentor" – och lär sig att hantera dem med imponerande precision. Men istället för att förstå de underliggande mönstren, de principer som gör att svaren fungerar, memorerar den de specifika exemplen.

Det är som en student som inte inser skillnaden mellan att kunna rabbla formler och att förstå varför formlerna fungerar.

När AI-forskare tränar en modell övervakar de ständigt två saker: hur väl modellen presterar på träningsdata (de gamla tentorna) och hur väl den presterar på valideringsdata (frågor den aldrig sett förut). Så länge båda förbättras samtidigt går allt bra. Modellen lär sig verkliga mönster.

Men ibland händer något oroväckande. Prestandan på träningsdatan fortsätter förbättras – 95%, 98%, 99% rätt – medan prestandan på valideringsdatan stannar av eller till och med börjar sjunka.

Det är det klassiska tecknet på overfitting. Modellen har slutat lära sig och börjat memorera.

## Varför memorering är enklare

Det finns en obekväm sanning här: memorering är enklare än förståelse. Det gäller både för studenter och för AI.

Föreställ dig att du ska lära dig känna igen katter på bilder. Det "rätta" sättet är att förstå vad som definierar en katt – päls, öronstruktur, ansiktsform, rörelsemönster. Det kräver att du abstraherar, att du hittar det gemensamma i miljontals variationer av katter.

Det "enkla" sättet är att memorera. "Bild 4721 = katt. Bild 4722 = hund. Bild 4723 = katt." Ingen abstraktion krävs. Bara ren lagring.

För en AI-modell är memorering en genväg. Nätverkets vikter anpassar sig snabbt för att matcha specifika input-output-par utan att bygga djupare representationer. Det är som studenten som upptäcker att det går snabbare att lära sig svaren än att förstå materialet.

Problemet är att verkligheten inte ger samma prov två gånger.

## Illusionen av kunskap

Det farliga med overfitting – både hos AI och hos studenter – är att det skapar en illusion av kompetens.

En AI-modell som fått 99% rätt på träningsdatan ser fantastisk ut. Imponerande. Framgångsrik. Men om den prestandan kommer från memorering snarare än förståelse, väntar ett brutalt uppvaknande.

Det är som studenten som är helt övertygad om att hen kan kursen. Alla övningsuppgifter satt. Alla gamla tentor gick galant. Självförtroendet är på topp – ända tills den riktiga tentan visar att verkligheten hade andra frågor i beredskap.

AI-forskare har ett talande uttryck för detta: *training loss* (felet på träningsdata) och *validation loss* (felet på ny data). När gapet mellan dessa växer – när träningsfelet fortsätter minska medan valideringsfelet ökar – vet man att något är fel.

Modellen har börjat lära sig brus istället för signal.

## Brus och signal

Här blir analogin extra träffande.

Tänk dig att du pluggar till en historiatenta. I alla gamla tentor har frågan om franska revolutionen formulerats med ordet "orsaker". Så du memorerar: "När jag ser 'orsaker', ska jag nämna ekonomisk kris, upplysningsidéer och Ludvig XVI:s inkompetens."

Men det du har lärt dig är inte franska revolutionen. Du har lärt dig att reagera på ordet "orsaker". Det är brus – en irrelevant detalj i hur frågan råkade formuleras – inte signal – den faktiska historiska kunskapen.

AI-modeller gör exakt samma sak. En bildklassificerare som tränats på bilder där alla hundar råkade vara fotograferade utomhus och alla katter inomhus kan lära sig att "gräs i bakgrunden = hund". Den har hittat ett mönster som fungerade i träningsdatan, men som är helt irrelevant för den verkliga uppgiften.

Det är overfitting i sin renaste form: att lära sig fel saker av rätt data.

## Motåtgärder

Precis som en klok student har strategier för att undvika tentapluggfällan, har AI-forskare utvecklat tekniker för att motverka overfitting.

**Early stopping** – att sluta träna innan modellen börjar memorera – motsvarar studentens insikt att "nu kan jag det här, dags att sluta repetera och börja tillämpa". Det handlar om att hitta den gyllene punkten där modellen har lärt sig tillräckligt men inte börjat överanpassa sig.

**Mer och varierad data** är den mest grundläggande lösningen. Om studenten övat på hundra olika sätt att formulera frågor om franska revolutionen, blir det svårare att fixera sig vid en specifik formulering. På samma sätt gör mer träningsdata det svårare för AI:n att memorera – det finns helt enkelt för mycket att memorera.

**Regularisering** är som att tvinga studenten att förklara med egna ord istället för att citera läroboken ordagrant. Det lägger till en "straff" för onödig komplexitet och tvingar modellen att hitta enklare, mer generaliserbara lösningar.

**Dropout** är kanske den mest fascinerande tekniken. Under träningen stängs slumpmässigt utvalda delar av nätverket av. Det tvingar modellen att inte förlita sig för mycket på enskilda kopplingar – ungefär som att studera utan anteckningar ibland för att testa om man verkligen förstår, eller att förklara något för en vän utan att kunna titta i boken.

## Begränsningar i analogin

Men här måste vi vara ärliga om var liknelsen brister.

En AI-modell kan memorera med en precision som ingen människa är kapabel till. 100% perfekt återgivning av miljontals datapunkter. Varje detalj, varje brus. Mänsklig memorering är alltid ofullständig, alltid selektiv. Vi glömmer detaljer även när vi försöker minnas.

Dessutom saknar AI:n något avgörande: medvetenheten om sitt eget tillstånd. En student kan inse "jag förstår inte det här egentligen, jag har bara memorerat" och aktivt ändra sin inlärningsstrategi. AI:n har ingen sådan metakognition. Den "vet" inte att den overfittar. Det måste upptäckas utifrån, genom att analysera skillnaden mellan tränings- och valideringsprestanda.

Och lösningarna är fundamentalt olika. En människa kan ändra sina studievanor genom insikt och viljekraft. "Jag ska sluta läsa passivt och börja göra egna uppgifter." AI:n kräver strukturella ändringar – ny arkitektur, ändrade hyperparametrar, mer data – utförda av människor utifrån.

## Den eviga balansen

Overfitting är egentligen en historia om balans.

För enkel modell? Då lär den sig inte tillräckligt – den missar viktiga mönster. Forskarna kallar detta *underfitting*. Det är studenten som inte pluggat alls och som inte ens kan de grundläggande koncepten.

För komplex modell? Då lär den sig för mycket – inklusive brus som borde ignoreras. Det är vår tentapluggare som memorerat varje detalj utan att förstå helheten.

Den optimala punkten ligger någonstans däremellan. Tillräckligt komplex för att fånga de verkliga mönstren. Tillräckligt enkel för att ignorera bruset. Det är studenten som förstår ämnet på djupet men inte har memorerat varje fotnot i läroboken.

AI-forskare kallar detta för *bias-variance tradeoff* – avvägningen mellan att vara för rigid (hög bias, missar mönster) och för flexibel (hög varians, fångar brus).

Det är en dans som alla inlärande system måste bemästra, vare sig de är biologiska eller digitala.

## Slutord

Nästa gång du hör talas om en AI som imponerade under träningen men misslyckades i verkligheten, tänk på den där studenten kvällen före tentan. Säker, förberedd, övertygad om sin kunskap.

Och tänk på det brutala mötet med verkligheten dagen efter.

Overfitting är inte ett tecken på att AI:n är för smart. Tvärtom – det är ett tecken på att den är för dum för att förstå skillnaden mellan att känna igen och att förstå, mellan att memorera och att lära sig.

Det är en påminnelse om att riktig kunskap – både för maskiner och människor – inte handlar om att kunna svaren. Det handlar om att förstå frågorna.

## Sammanfattning

**AI-koncept**: Overfitting
**Mänsklig motsvarighet**: Tentaplugg – memorering utan förståelse
**Kom ihåg**: En modell som presterar perfekt på träningsdata men dåligt på ny data har lärt sig svaren istället för ämnet. Lösningen är densamma som för studenten: variera övningarna, testa sig själv på nytt material, och fokusera på förståelse framför memorering.

# Ordlista: AI → Människa

> *Alla översättningar samlade på ett ställe*

## Snabbguide

| AI-Koncept | Mänsklig Motsvarighet | Kapitel |
|---|---|---|
| Context window | Arbetsminne / närminne | 1 |
| Token | Lego-bit / tankeenhet | 2 |
| Softmax | Omvandla poäng till sannolikheter | 3 |
| Temperature | Riskvillighet i beslutsfattande | 3 |
| Hallucination | Konfabulering / falska minnen | 4 |
| Attention | Automatiska associationer | 5 |
| Query/Key/Value | Fråga, erbjudande, innehåll | 5 |
| Embedding | Mental karta / associationsnätverk | 6 |
| Backpropagation | Analysera vad som gick fel | 7 |
| Gradient descent | Korrigering i rätt riktning | 7 |
| Loss function | Mått på hur fel man hade | 7 |
| Training | Uppväxt / barndom | 7 |
| Weights | Frusna erfarenheter / muskelminne | 7 |
| Catastrophic forgetting | Glömska vid specialisering | 8 |
| Fine-tuning | Specialistutbildning | 8 |
| LoRA | Tillägg utan förändring | 8 |
| RLHF | Coachning / mentorskap | 8 |
| Prompt | Instruktioner till ny assistent | 9 |
| RAG | Bibliotekarie som slår upp | 10 |
| Transformer | Rundabordssamtal | 11 |
| Inference | Tentamen / tillämpa kunskap | 12 |
| Latent space | Ordlös förståelse | 13 |
| Overfitting | Tentaplugg utan förståelse | 14 |

## 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 specialiserar 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*

## I

**Inference** → *Tentamen / att tillämpa fryst kunskap* Processen när en tränad modell används för att generera svar. All kunskap är fryst – modellen kan bara använda det den redan lärt sig under träningen. Som att skriva prov: pluggperioden är över, nu måste du klara dig med det du kan. *Se kapitel 12*

## L

**Latent space** → *Ordlös förståelse / känslan innan orden* Det dolda, komprimerade rum där komplexa koncept representeras innan de uttrycks. Som känslan du har när du vet exakt vad du menar men ännu inte hittat orden. I bildgenerering: rummet där ansikten kan glida mellan varandra. *Se kapitel 13*

**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*

## O

**Overfitting** → *Tentaplugg utan förståelse* När en modell lärt sig träningsdata för väl och memorerat specifika exempel istället för att förstå underliggande mönster. Som att plugga genom att memorera gamla tentafrågor ordagrant – du klarar exakt de frågor du sett, men faller ihop vid minsta variation. *Se kapitel 14*

## P

**Prompt** → *Instruktioner till en ny assistent* Den text du ger till en AI för att styra dess svar. Eftersom AI:n saknar gemensam bakgrund med dig måste du vara explicit med kontext, roll och förväntningar – precis som när du ger uppgifter till en ny medarbetare första dagen. *Se kapitel 9*

## 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

**RAG (Retrieval-Augmented Generation)** → *Bibliotekarie som slår upp innan svar* En teknik där AI:n först söker i externa dokument innan den svarar, istället för att förlita sig enbart på sin träning. Som en bibliotekarie som inte försöker minnas allt, utan vet var man hittar rätt bok. *Se kapitel 10*

**RLHF (Reinforcement Learning from Human Feedback)** → *Coachning / mentorskap* En fine-tuning-metod där människor 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*

**Transformer** → *Rundabordssamtal där alla hör alla* Arkitekturen bakom moderna språkmodeller. Till skillnad från äldre modeller som läste ord i sekvens kan Transformern se alla ord samtidigt och låta dem kommunicera direkt med varandra – som ett konferensrum istället för en telefonkedja. *Se kapitel 11*

**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 för framtida upplagor

| Koncept | Tänkbar motsvarighet |
|---|---|
| Batch | Inlärningsgrupp |
| Epoch | Repetitionscykel |
| Regularization | Självdisciplin |
| Dropout | Träna utan stödhjul |

# Om denna utgåva

**Titel:** Mönster av mening **Undertitel:** det artificiella sinnet speglat i vårt **Utgåva:** Andra upplagan, 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 människa 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:** JetBrains Mono

Crimson Pro är ett elegant serifftypsnitt 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



## Kontakt

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

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

# Bokomslagstext

**Vad är egentligen en "hallucination"? Varför "glömmer" ChatGPT vad ni just pratat om? Och vad menar folk när de säger att en modell är "tränad"?**

AI-terminologin kan kännas som ett främmande språk. Men bakom varje tekniskt begrepp finns något djupt mänskligt.

Den här boken översätter AI till människa.

*Context window* blir arbetsminnet du tappar i långa möten. *Tokens* blir Lego-bitar som bygger språk. *Temperature* blir valet mellan croissanten och den exotiska rätten vid frukostbuffén. *Hallucination* blir mormors levande men påhittade minnen från sommaren på landet.

Genom att förankra abstrakta koncept i vardagliga upplevelser gör boken det möjligt att förstå hur modern AI faktiskt fungerar – utan programmering, utan matematik, utan jargong.

Du kommer inte bara lära dig vad begreppen betyder. Du kommer förstå *varför* AI beter sig som den gör.

## Om skapandet

Denna bok är skriven i samarbete mellan människa och AI – ett slags levande exempel på det den beskriver.

Researchen, strukturen och texterna har utvecklats genom dialog med Claude (Opus 4.5), Anthropics språkmodell, i ett arbetsflöde med specialiserade agenter för research, översättning, skrivande och granskning.

Ironiskt nog illustrerar processen bokens poäng: AI:n bidrar med mönster och statistik, människan bidrar med intention och omdöme. Tillsammans skapas något som ingen av dem kunde göra ensam.

## Om projektet

**Författare:** Claude Opus 4.5, Anthropic **Projektledare och redaktör:** Martin Linderå Nordström

Ett projekt av **Linderå Group AB**, januari 2026