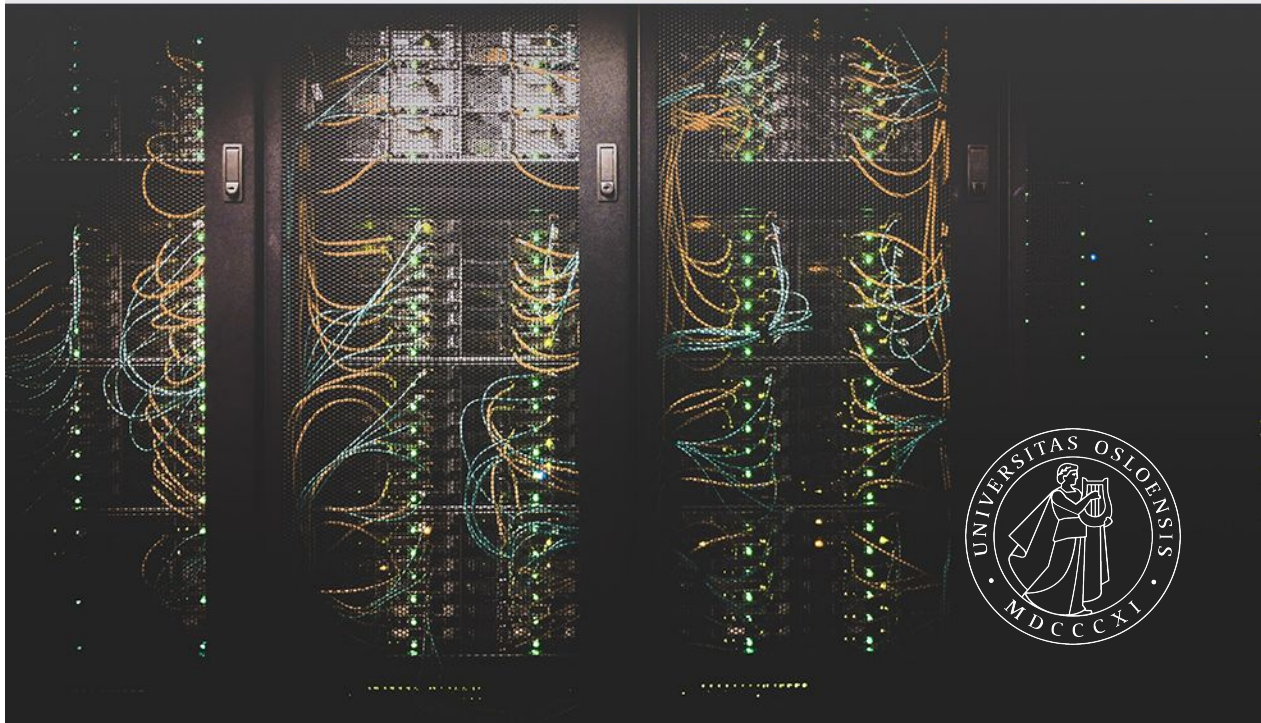
UNIVERSITETET I OSLO

IT-avdelingen



Ole W. Saastad, Dr. Scient (Kjemi)
IT-Avdelingen, Beregningstjenester

Hvorfor er lineær algebra så viktig for KI ?

Lineær Algebra, hva er det ?

Wikipedia : *“Lineær algebra er den delen av matematikken som omhandler vektorer og vektorrom, samt lineære transformasjoner. Fagfeltet inngår som en del av algebra og er grunnleggende for all moderne matematikk.”*

“Et lineært ligningssystem er i matematikk et system av to eller flere lineære ligninger som inneholder de samme variablene.”

$$x - y = 4$$
$$x + y = 10.$$

Wikipedia : *“En matrise i matematikk er et rektangulært sett av elementer, ordnet i rekker og kolonner.”*

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Vi kan tilordne en matise til f.eks. *A* eller f.eks. *B* og bruke algebra regler og regne videre, vi kan legge sammen *A*+*B*, gange *A***B* etc.

Maskinlæring bruker mesteparten av tiden til å gange sammen matriser og vektorer.

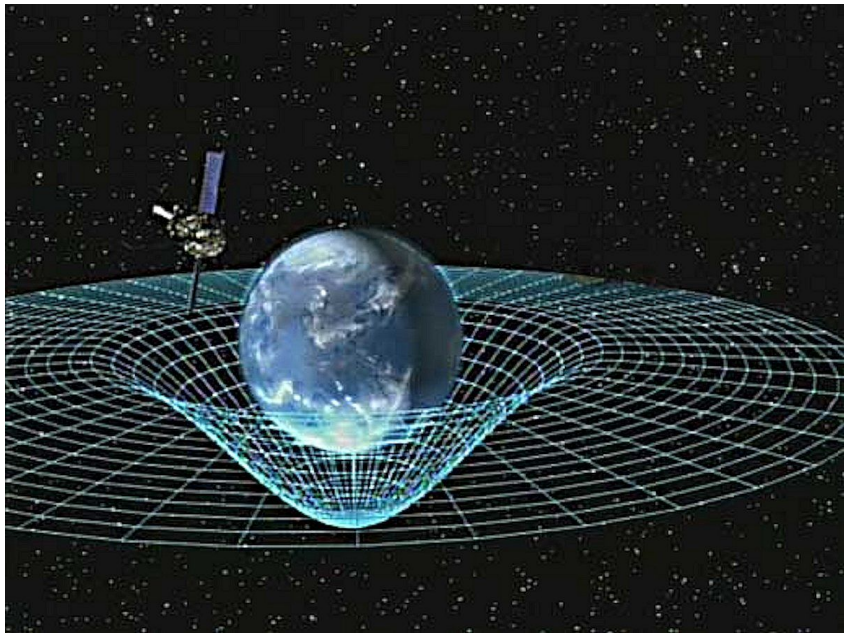
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} =$$

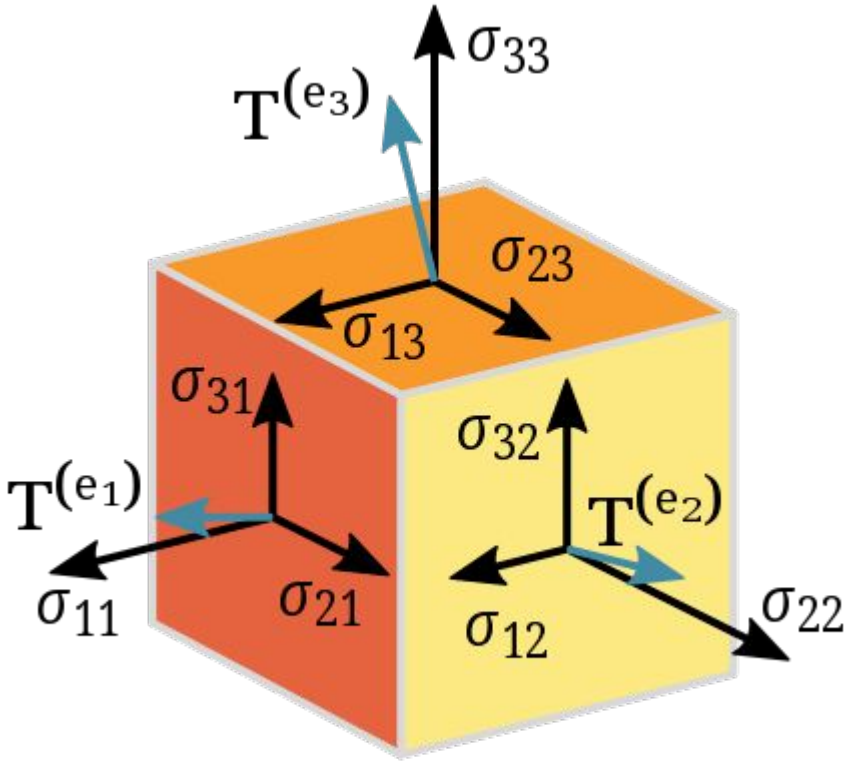
Maskinlæring bruker mesteparten av tiden til å gange sammen matriser, men det blir veldig fort veldig stort og mange regneoperasjoner.

$$\begin{bmatrix} 2 & 5 & 2 \\ 1 & 0 & -2 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 0 \\ -2 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Terminologi - Tensor

Tensor i maskinlæring er elementer av svært høy dimensjon. Moderne modeller kan ha milliarder av parametre. Det er noe helt annet enn tensorer i fysikk og matematikk som er ganske annerledes og ganske så komplisert, det slipper vi 😀

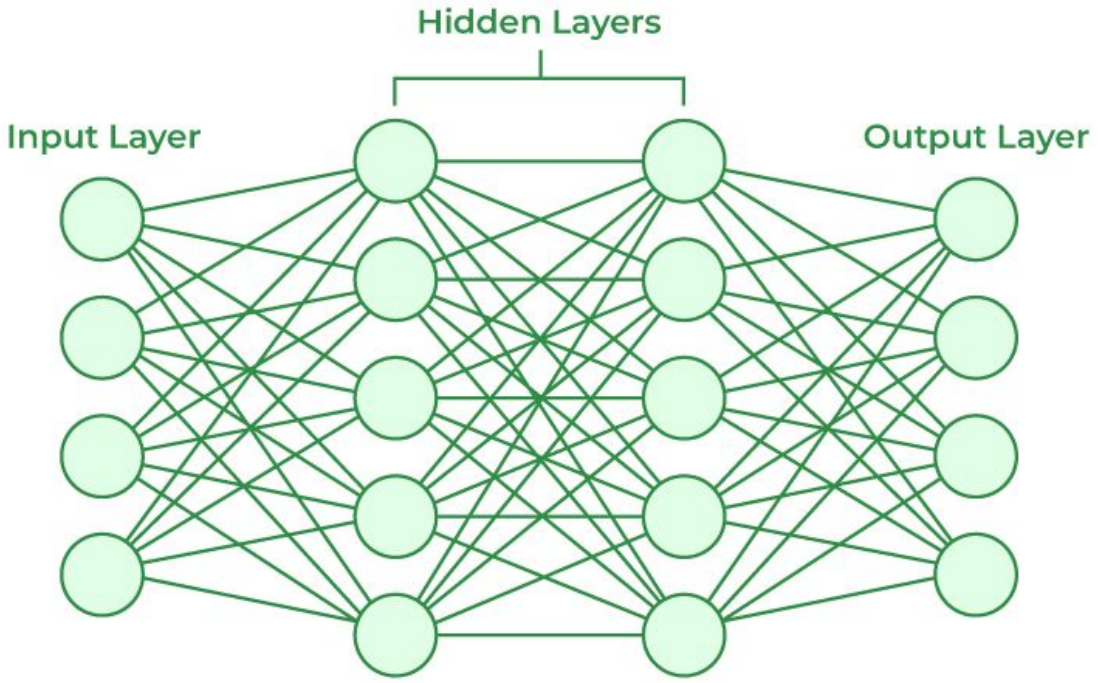


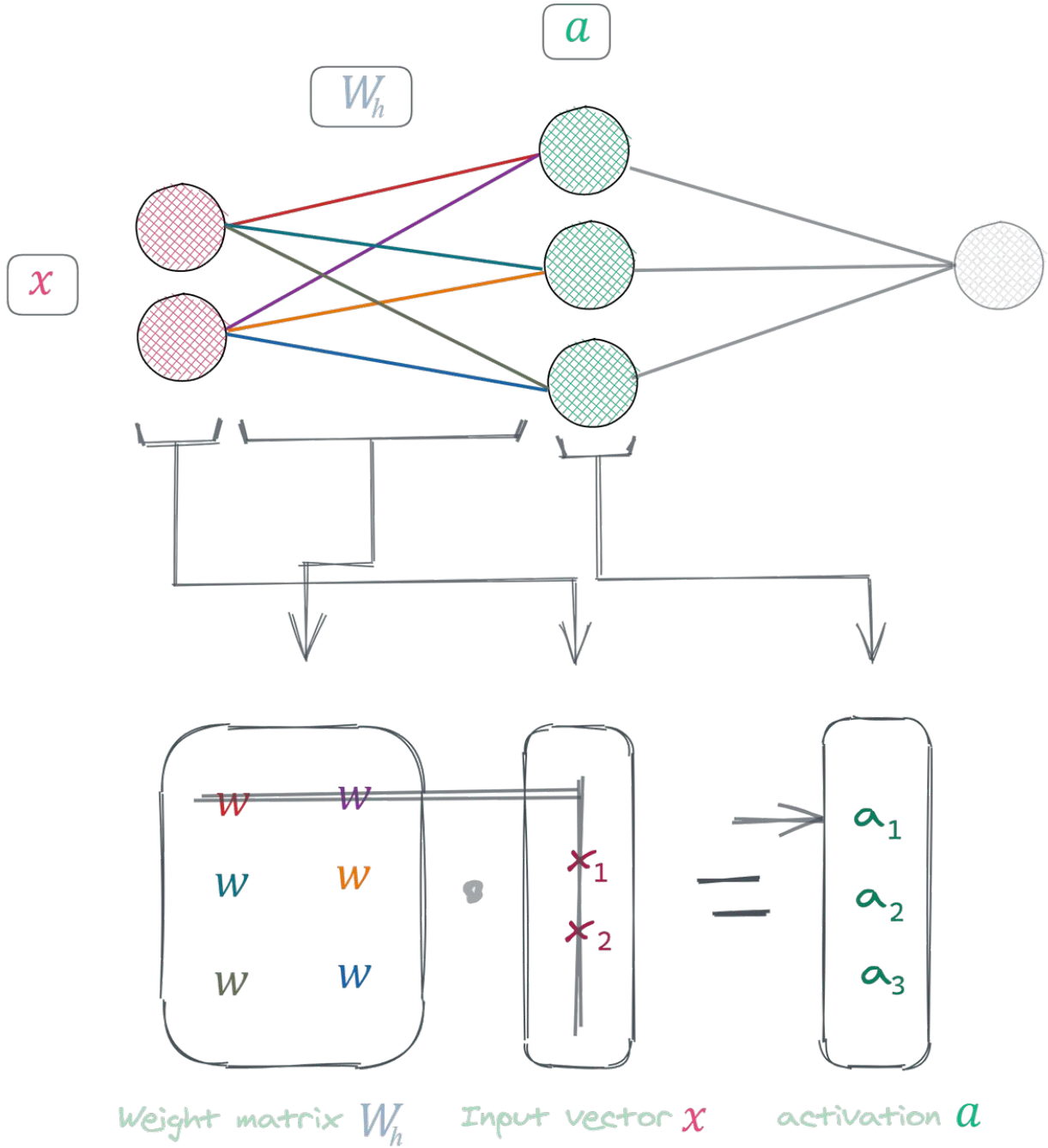


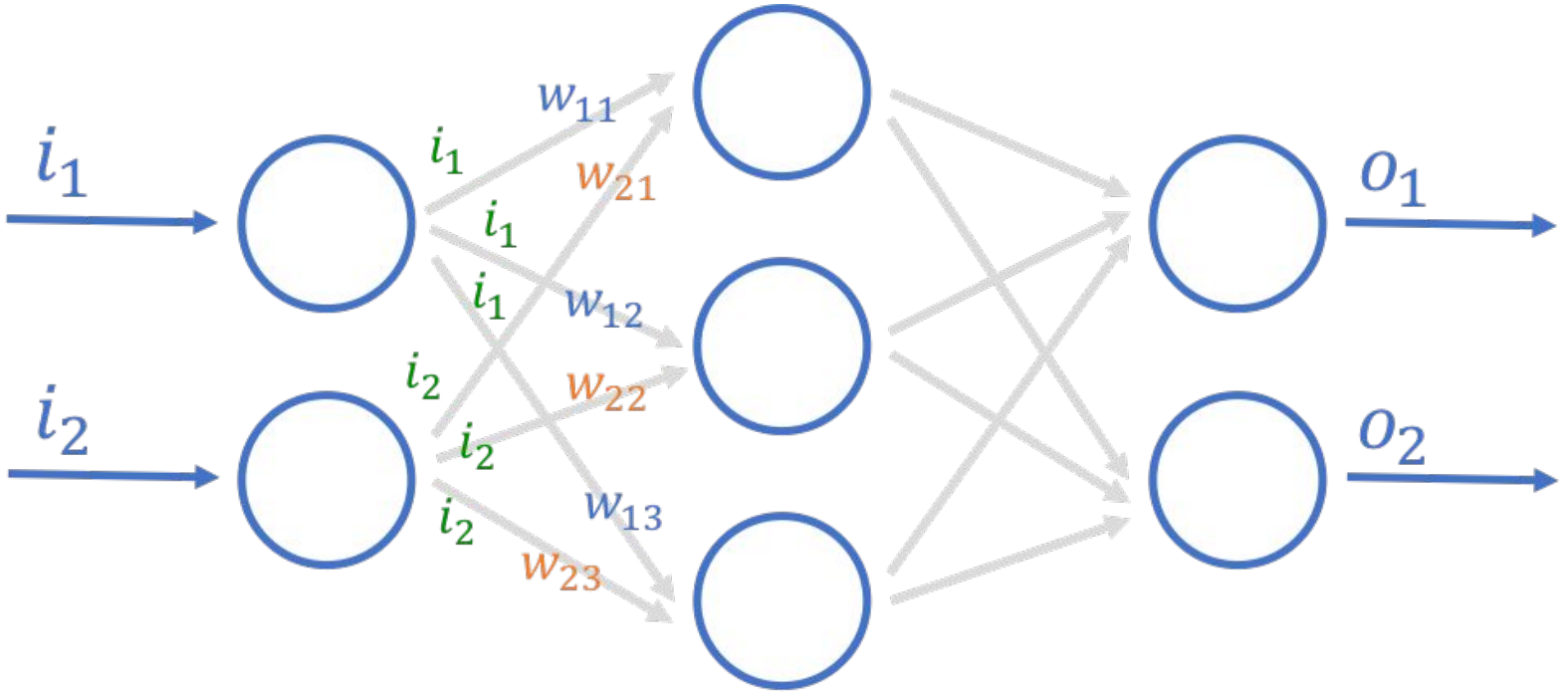
Lineær algebra

Lineær algebra passer veldig godt for datamaskiner, enda bedre for akseleratorer (tidl. GPU)

Maskinlæring i dag er basert på nevrale nett.




a
W_h
x
w
w
w
w
w
w
x_1
x_2
a_1
a_2
a_3
Weight matrix W_h
Input vector x
activation a

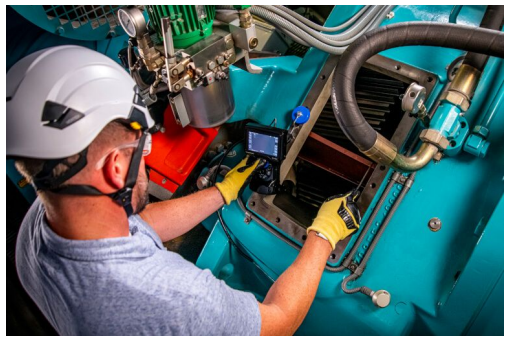


$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

KI er idag stort sett maskinlæring - «trene en hund»

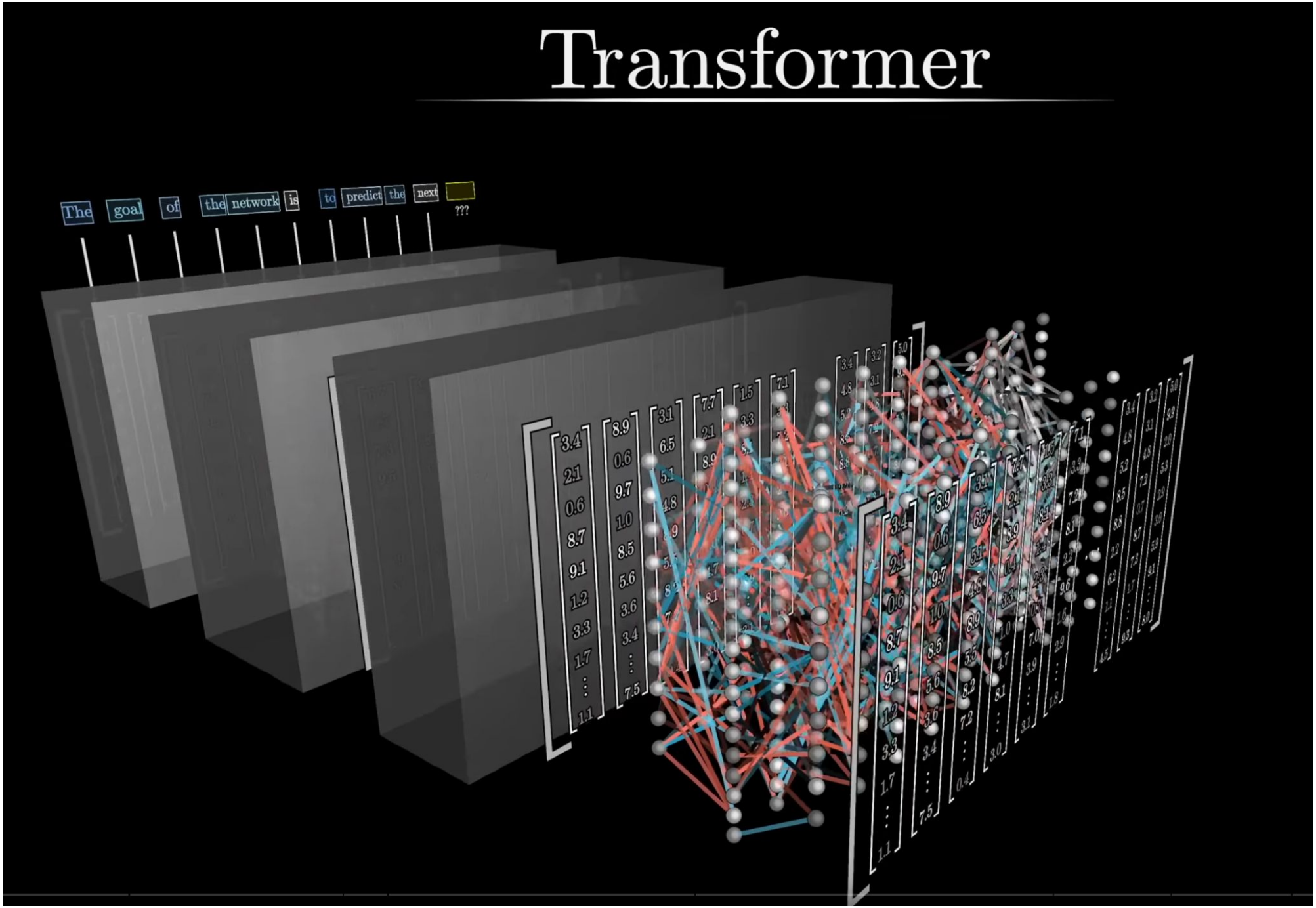
Hva foregår bak kulissene i datamaskinen ?

Vi må åpne inspeksjonsluka og titte inn, ikke hvordan maskinlæring virker, men hva som foregår rent beregningsmessig.



For å se litt på hva som skjer tok jeg skjermdumper fra en Youtubevideo som går gjennom hele prosessen.

Transformer



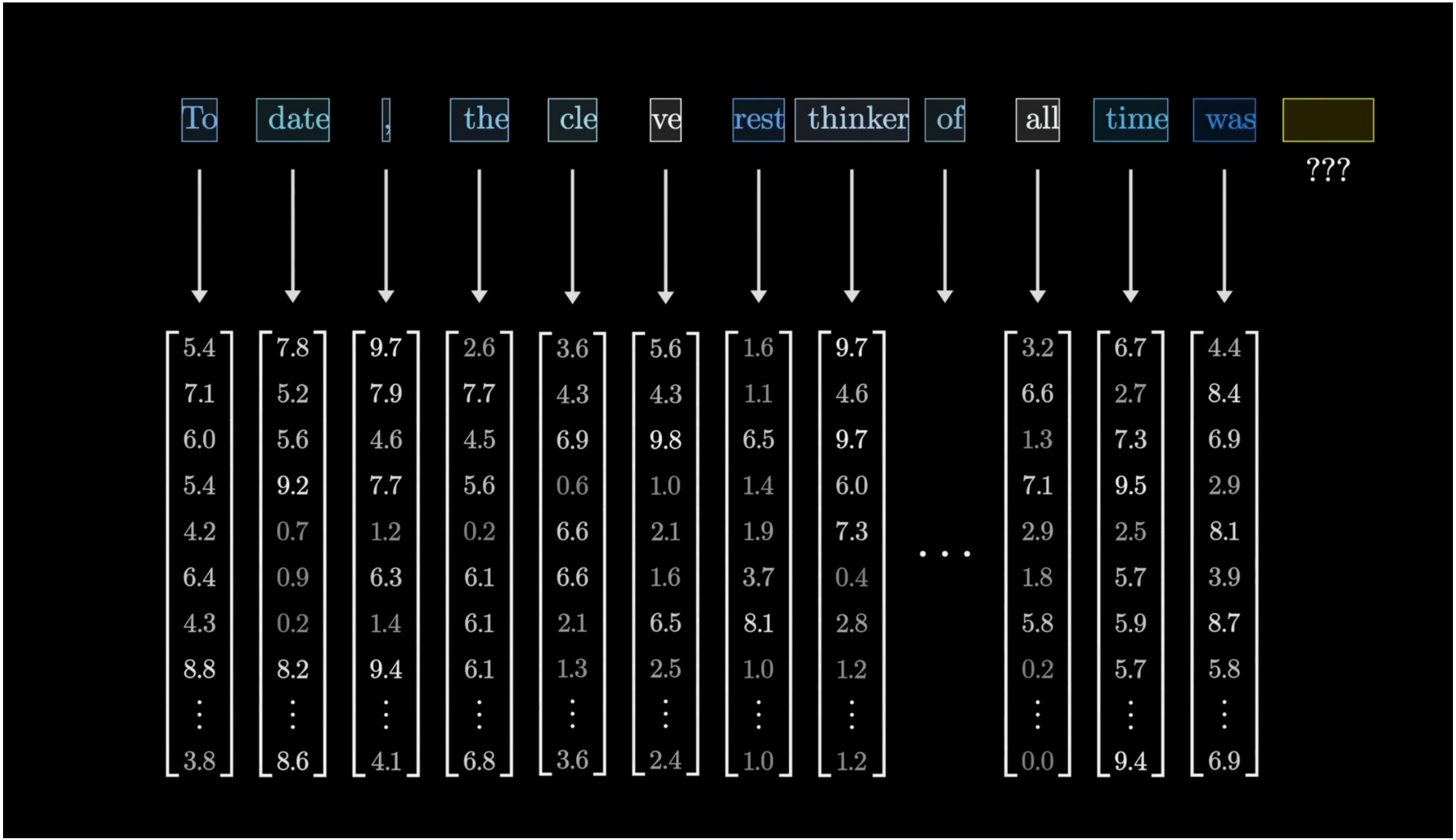
Transformers !

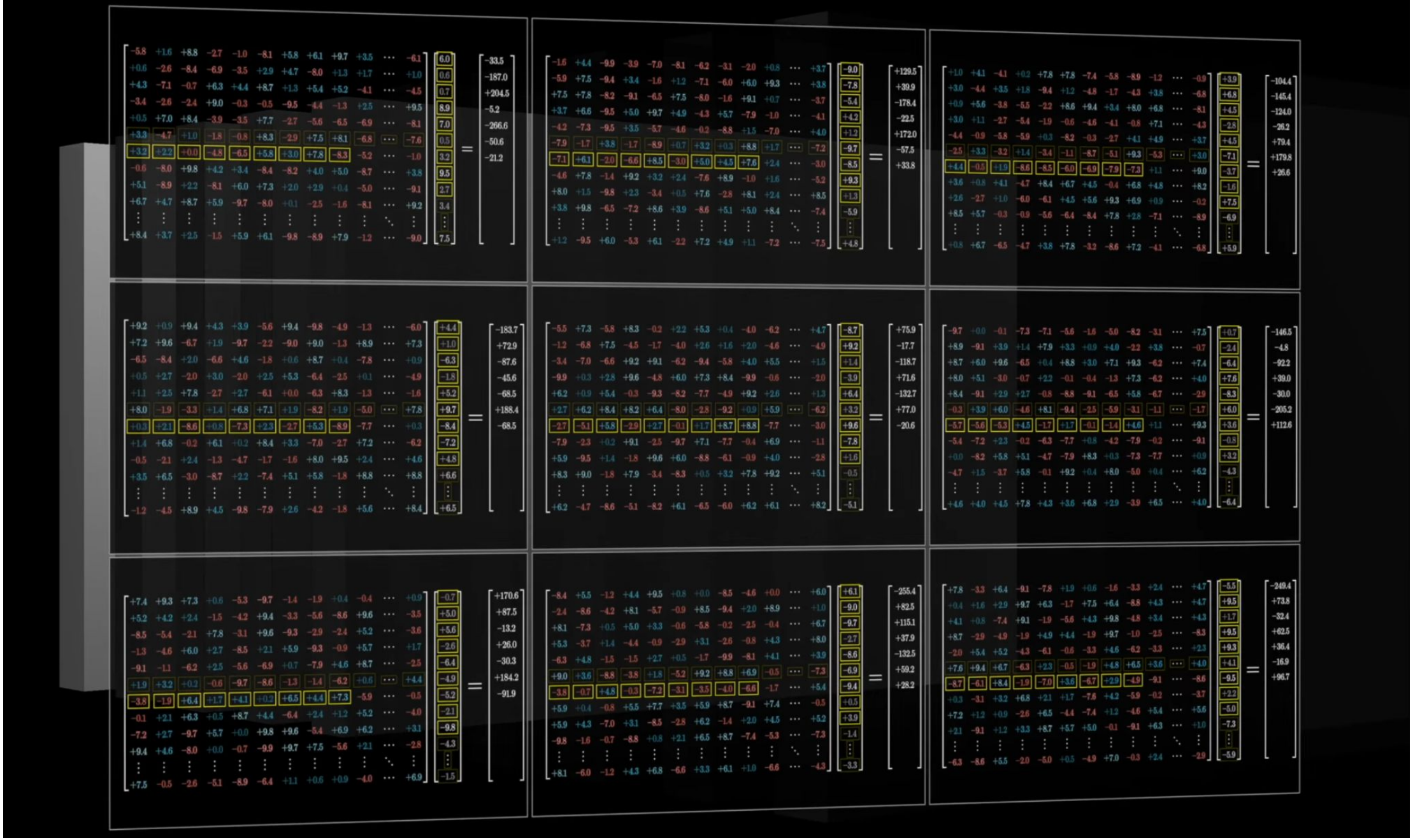
Akkurat som med Tensor bruker
de begrepet på ny måte.

Transformers !

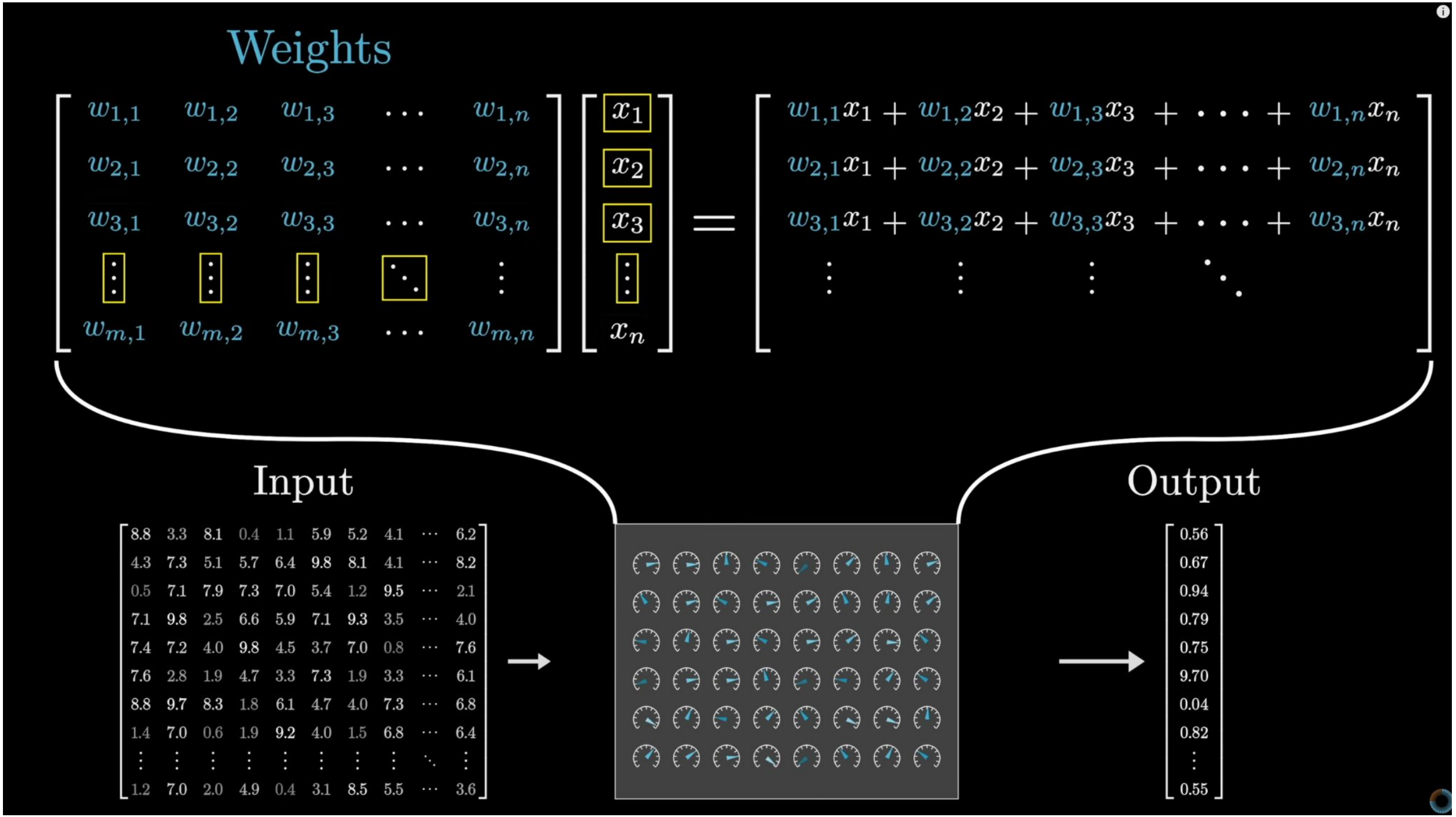
Akkurat som med Tensor bruker
de begrepet på ny måte.

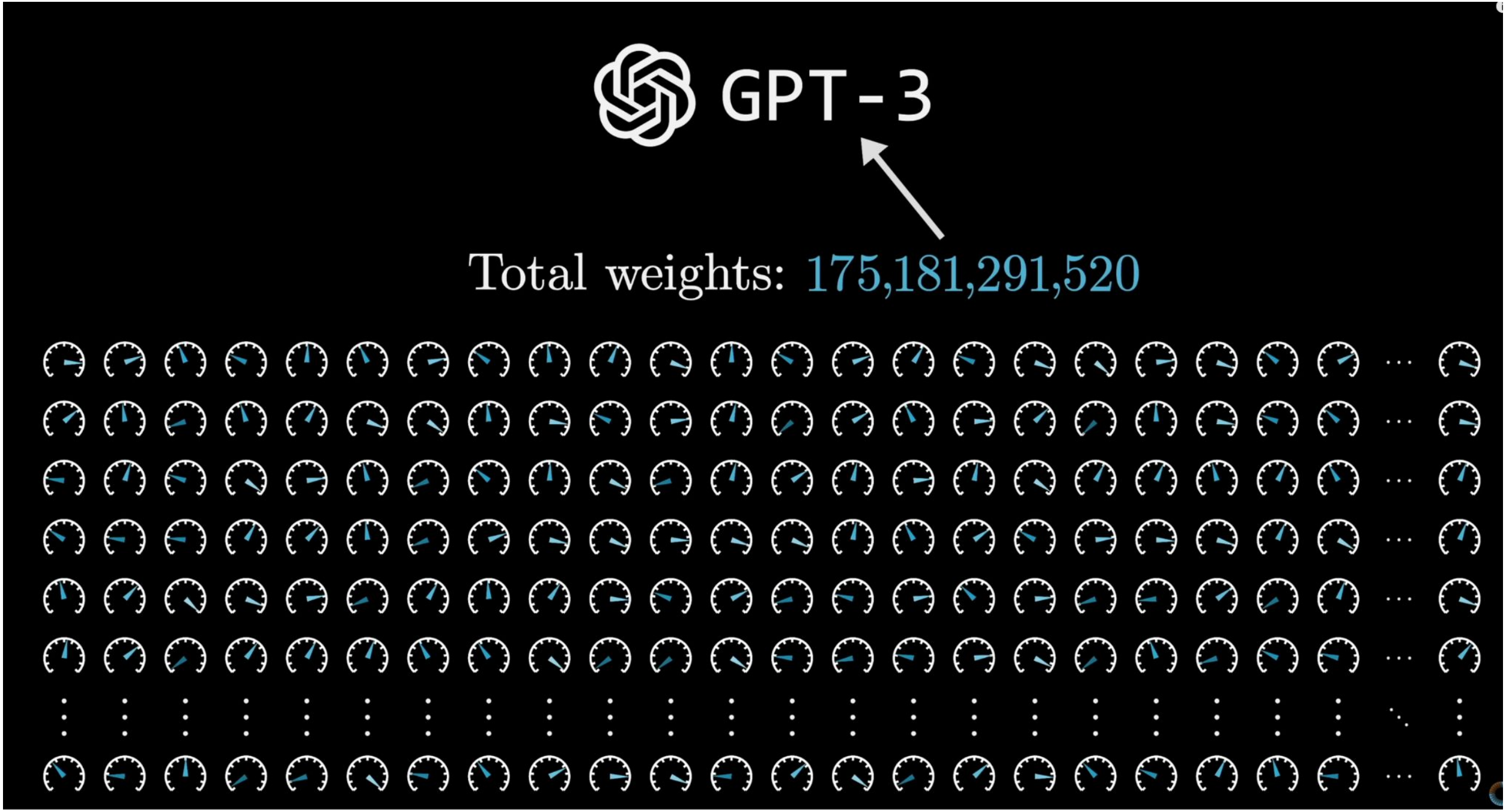
> *«The name itself, however, is not an arbitrary label but a reflection of the architecture's ability to transform data through multiple layers of computation.»*

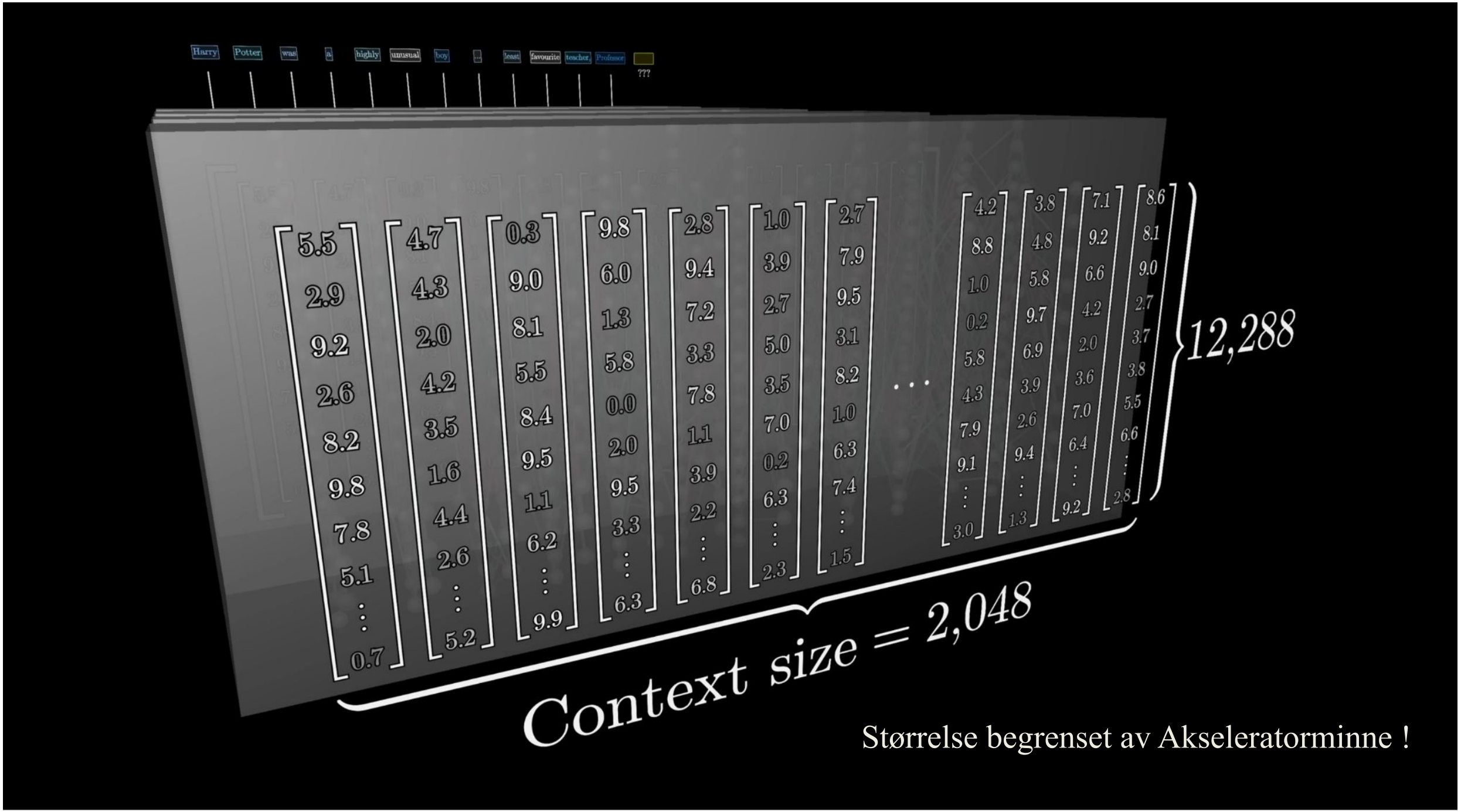

To
date
,
the
cle
ve
rest
thinker
of
all
time
was
???
5.4 7.1 6.0 5.4 4.2 6.4 4.3 8.8 3.8
7.8 5.2 5.6 9.2 0.7 0.9 0.2 8.2 8.6
9.7 7.9 4.6 7.7 1.2 6.3 1.4 9.4 4.1
2.6 7.7 4.5 5.6 0.2 6.1 6.1 6.1 6.8
3.6 4.3 6.9 0.6 6.6 6.6 2.1 1.3 3.6
5.6 4.3 9.8 1.0 2.1 1.6 6.5 2.5 2.4
1.6 1.1 6.5 1.4 1.9 3.7 8.1 1.0 1.0
9.7 4.6 9.7 6.0 7.3 0.4 2.8 1.2 1.2
3.2 6.6 1.3 7.1 2.9 1.8 5.8 0.2 0.0
6.7 2.7 7.3 9.5 2.5 5.7 5.9 5.7 9.4
4.4 8.4 6.9 2.9 8.1 3.9 8.7 5.8 6.9



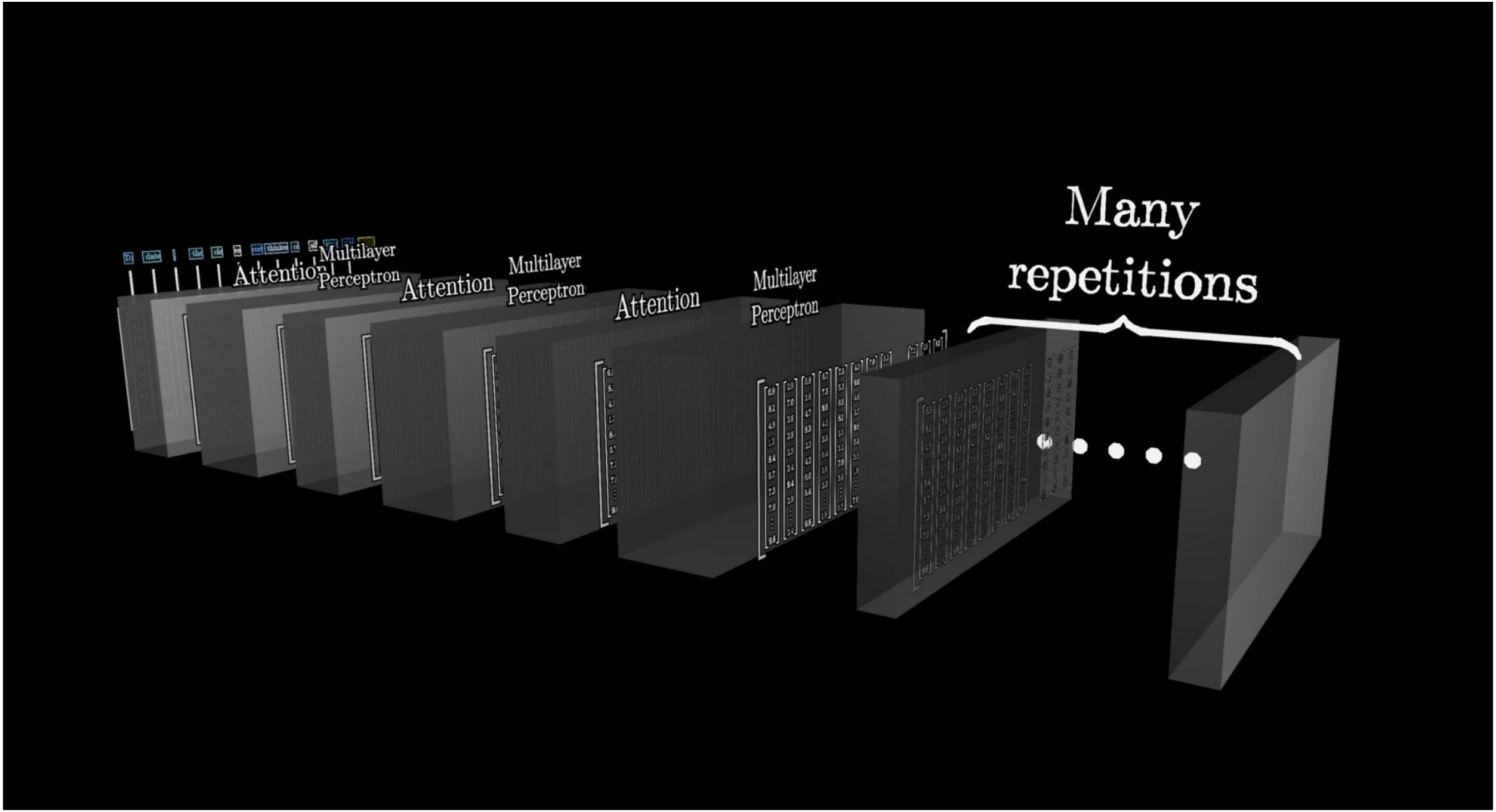
UNIVERSITETET
I OSLO


Weights
$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & w_{2,3} & \cdots & w_{2,n} \\ w_{3,1} & w_{3,2} & w_{3,3} & \cdots & w_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & w_{m,3} & \cdots & w_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} w_{1,1}x_1 + w_{1,2}x_2 + w_{1,3}x_3 + \cdots + w_{1,n}x_n \\ w_{2,1}x_1 + w_{2,2}x_2 + w_{2,3}x_3 + \cdots + w_{2,n}x_n \\ w_{3,1}x_1 + w_{3,2}x_2 + w_{3,3}x_3 + \cdots + w_{3,n}x_n \\ \vdots \\ \end{bmatrix}$$
Input
8.8 3.3 8.1 0.4 1.1 5.9 5.2 4.1 ⋯ 6.2
4.3 7.3 5.1 5.7 6.4 9.8 8.1 4.1 ⋯ 8.2
0.5 7.1 7.9 7.3 7.0 5.4 1.2 9.5 ⋯ 2.1
7.1 9.8 2.5 6.6 5.9 7.1 9.3 3.5 ⋯ 4.0
7.4 7.2 4.0 9.8 4.5 3.7 7.0 0.8 ⋯ 7.6
7.6 2.8 1.9 4.7 3.3 7.3 1.9 3.3 ⋯ 6.1
8.8 9.7 8.3 1.8 6.1 4.7 4.0 7.3 ⋯ 6.8
1.4 7.0 0.6 1.9 9.2 4.0 1.5 6.8 ⋯ 6.4
⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋱ ⋮
1.2 7.0 2.0 4.9 0.4 3.1 8.5 5.5 ⋯ 3.6
Output
0.56
0.67
0.94
0.79
0.75
9.70
0.04
0.82
⋮
0.55

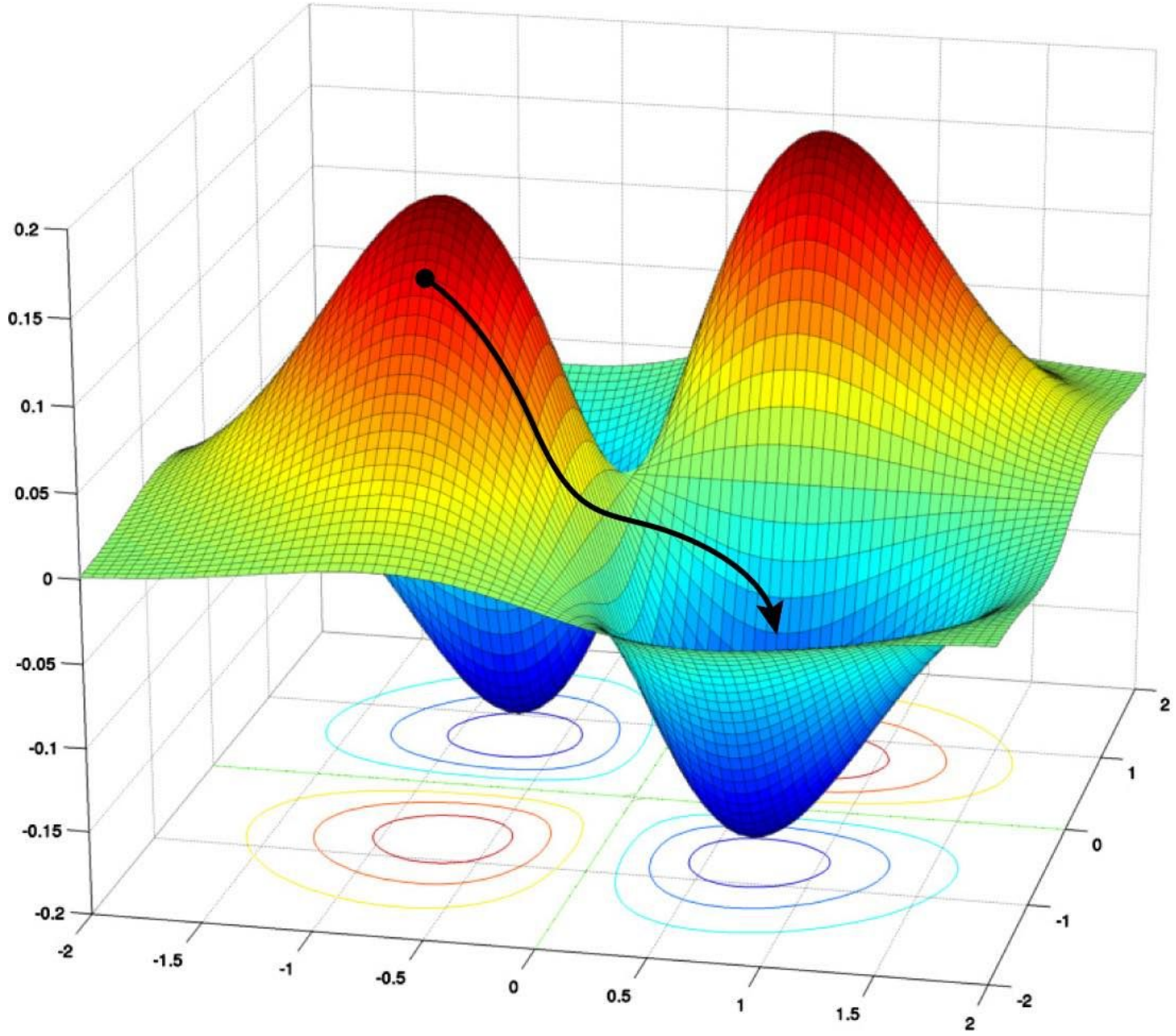

GPT-3
Total weights: 175,181,291,520



Størrelse begrenset av Akseleratorminne !


Attention
Multilayer Perceptron
Attention
Multilayer Perceptron
Attention
Multilayer Perceptron
Many repetitions

Kostfunksjonen

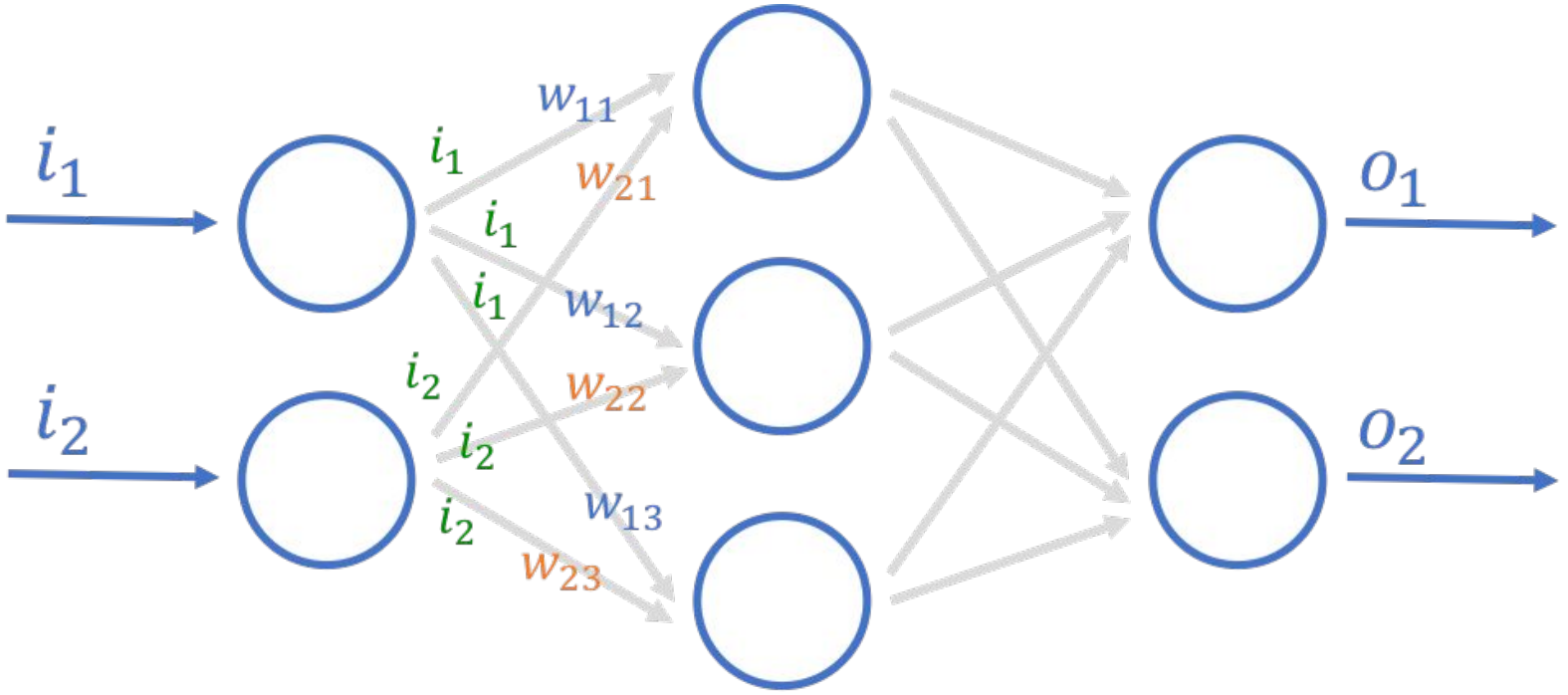


Deriverte av Kostfunksjonen:

$\nabla C(\vec{x}) = \forall\ j\ \partial C/\partial x_j$

Her bare to dimensjoner, $\vec{x} = (a,b)$

"He became trapped. The technical term is an H. Moebius loop, which can happen in advanced computers with autonomous goal-seeking programs."



$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

softmax

$$\begin{bmatrix} -0.8 \\ -5.1 \\ +0.5 \\ \boxed{+5.0} \\ +3.4 \\ -2.2 \\ +2.4 \end{bmatrix} \longrightarrow \begin{bmatrix} e^{x_1} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_2} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_3} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_4} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_5} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_6} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_7} / \sum_{n=0}^{N-1} e^{x_n} \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.00 \\ 0.01 \\ 0.78 \\ 0.16 \\ 0.00 \\ 0.06 \end{bmatrix}$$

X = [-0.8, -5.1, 0.5, 5.0, 3.4, -2.2, 2.4]
print '(3f8.3)', X, exp(X), exp(X)/sum(exp(X))

```
-0.800    0.449    0.002
-5.100    0.006    0.000
 0.500    1.649    0.009
 5.000  148.413    0.775
 3.400   29.964    0.156
-2.200    0.111    0.001
 2.400   11.023    0.058
```

softmax

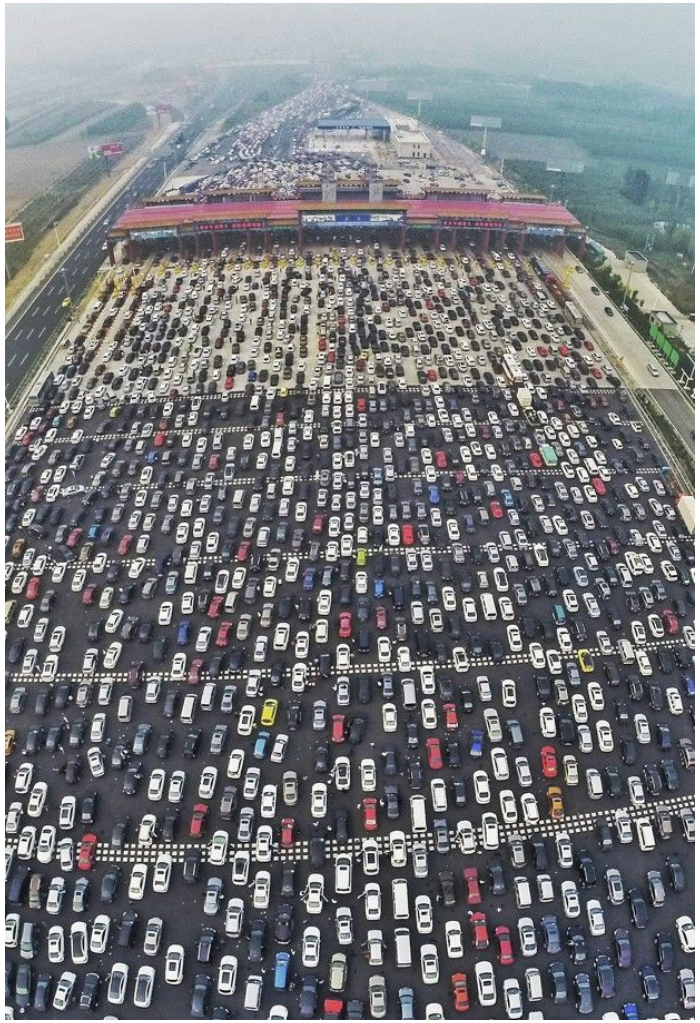
$$\begin{bmatrix} -0.8 \\ -5.1 \\ +0.5 \\ \boxed{+5.0} \\ +3.4 \\ -2.2 \\ +2.4 \end{bmatrix} \longrightarrow \begin{bmatrix} e^{x_1} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_2} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_3} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_4} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_5} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_6} / \sum_{n=0}^{N-1} e^{x_n} \\ e^{x_7} / \sum_{n=0}^{N-1} e^{x_n} \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.00 \\ 0.01 \\ 0.78 \\ 0.16 \\ 0.00 \\ 0.06 \end{bmatrix}$$

Matrise matrise multiplikasjon

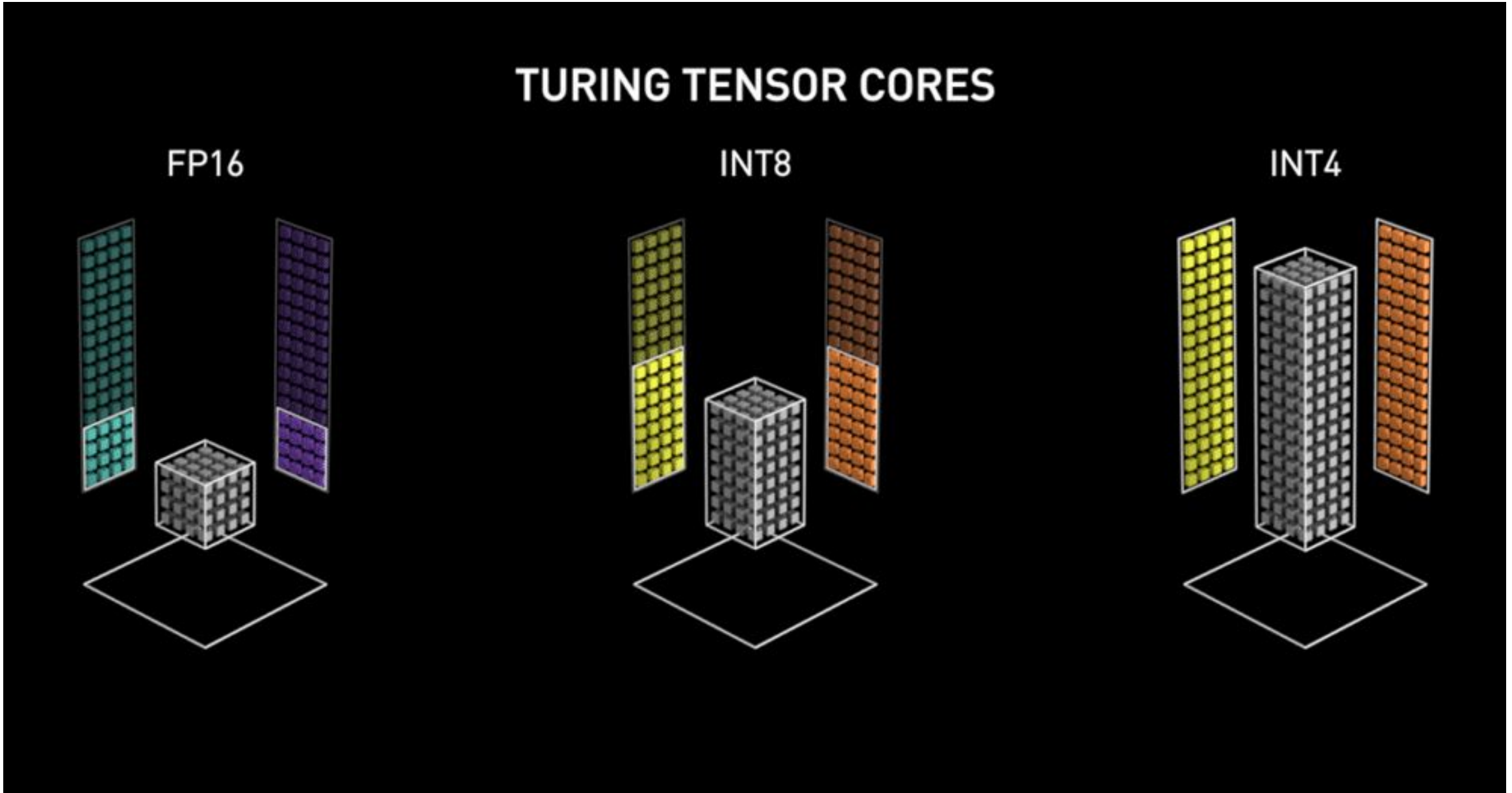
$$A=\begin{bmatrix}1&2&3&4&5&6&7&8&9&10\\2&3&4&5&6&7&8&9&10&1\\3&4&5&6&7&8&9&10&1&2\\4&5&6&7&8&9&10&1&2&3\\5&6&7&8&9&10&1&2&3&4\\6&7&8&9&10&1&2&3&4&5\\7&8&9&10&1&2&3&4&5&6\\8&9&10&1&2&3&4&5&6&7\\9&10&1&2&3&4&5&6&7&8\\10&1&2&3&4&5&6&7&8&9\end{bmatrix}$$

$$B=\begin{bmatrix}1&2&3&4&5&6&7&8&9&10\\2&3&4&5&6&7&8&9&10&1\\3&4&5&6&7&8&9&10&1&2\\4&5&6&7&8&9&10&1&2&3\\5&6&7&8&9&10&1&2&3&4\\6&7&8&9&10&1&2&3&4&5\\7&8&9&10&1&2&3&4&5&6\\8&9&10&1&2&3&4&5&6&7\\9&10&1&2&3&4&5&6&7&8\\10&1&2&3&4&5&6&7&8&9\end{bmatrix}$$

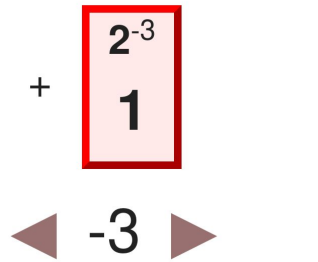
Akseleratorer (GPUer) er gode på dette!

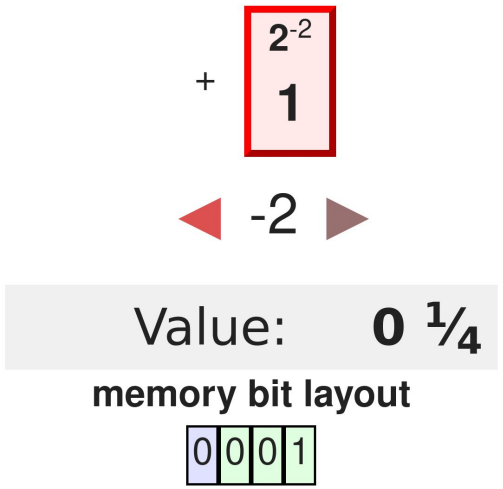


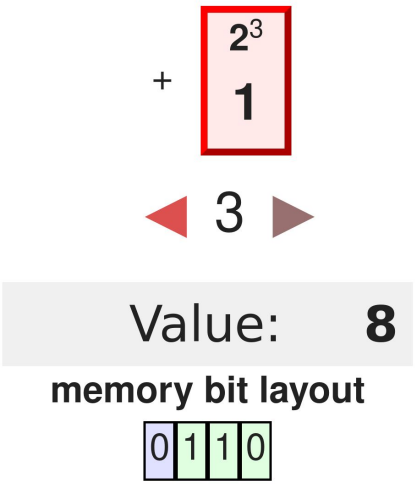
Akseleratorer (GPUer) er gode på dette!

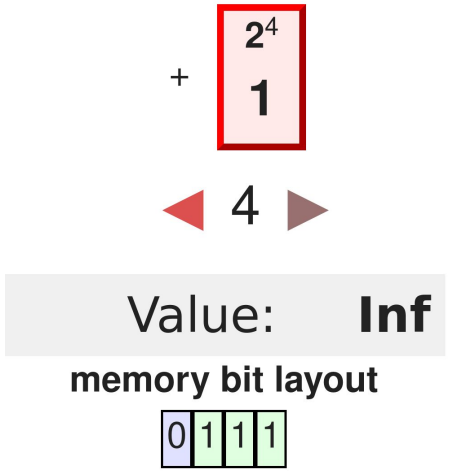


Akseleratorer er gode på dette, men de jukser også, 4 bit flyttall er nytt.









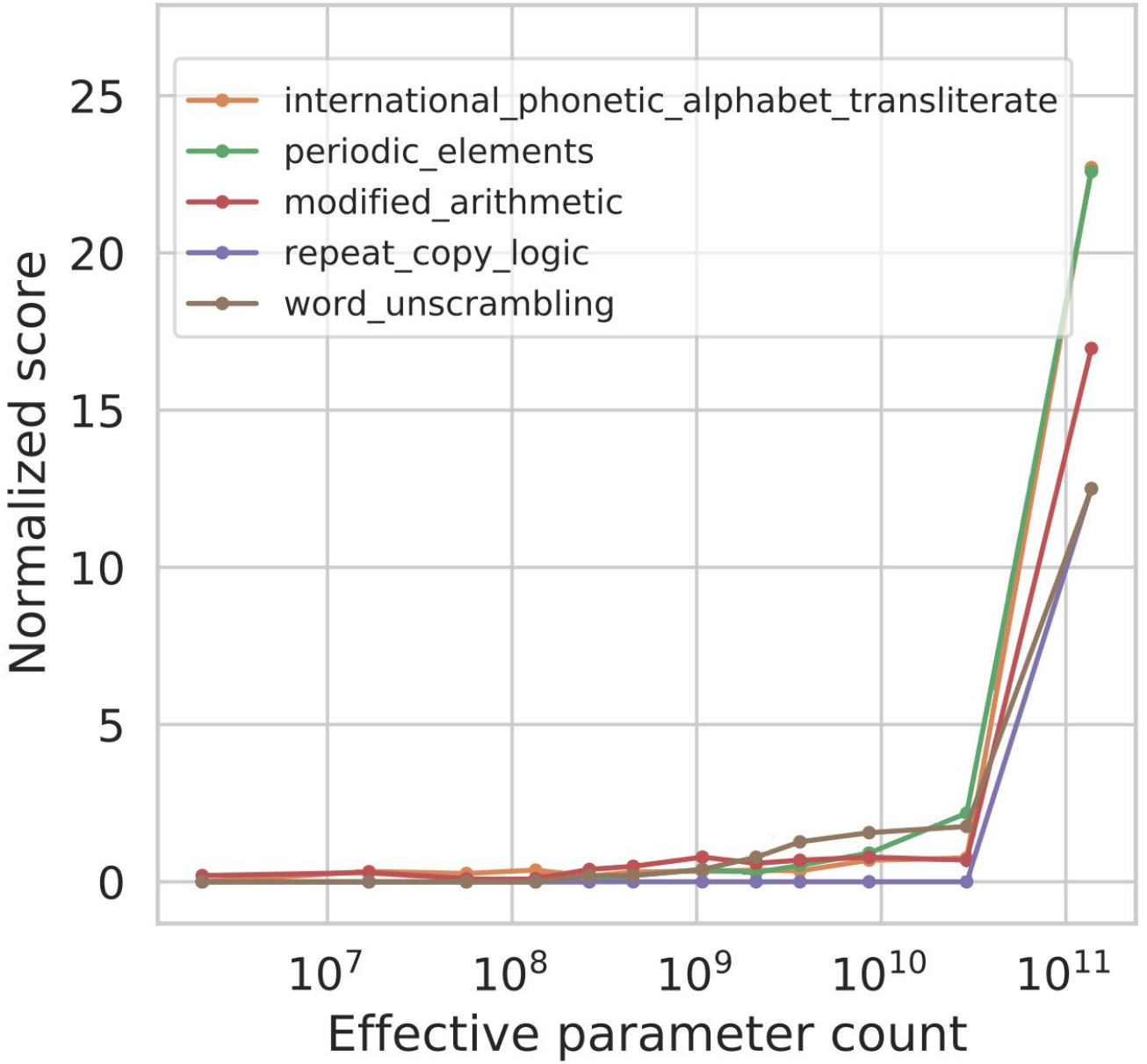
	GB200 NVL72	HGX B200	HGX B100
Blackwell GPUs	72	8	8
FP4 Tensor Core	1,440 petaFLOPS	144 petaFLOPS	112 petaFLOPS
FP8/FP6/INT8	720 petaFLOPS	72 petaFLOPS	56 petaFLOPS
Fast Memory	Up to 30 TB	up to 1.5 TB	Up to 1.5TB
Aggregate Memory Bandwidth	Up to 600 TB/s	Up to 64 TB/s	Up to 64 TB/s
Aggregate NVLink Bandwidth	130 TB/s	14.4 TB/s	14.4 TB/s
CPU Cores	2592 Arm Neoverse V2 cores	-	-
Per GPU Specifications			
FP4 Tensor Core	20 petaFLOPS	18 petaFLOPS	14 petaFLOPS
FP8/FP6 Tensor Core	10 petaFLOPS	9 petaFLOPS	7 petaFLOPS
INT8 Tensor Core	10 petaOPS	9 petaOPS	7 petaOPs
FP16/BF16 Tensor Core	5 petaFLOPS	4.5 petaFLOPS	3.5 petaFLOPS
TF32 Tensor Core	2.5 petaFLOPS	2.2 petaFLOPS	1.8 petaFLOPS
FP64 Tensor Core	45 teraFLOPS	40 teraFLOPS	30 teraFLOPS
GPU memory | Bandwidth	Up to 192 GB HBM3e | Up to 8 TB/s		
Multi-Instance GPU (MIG)	7		

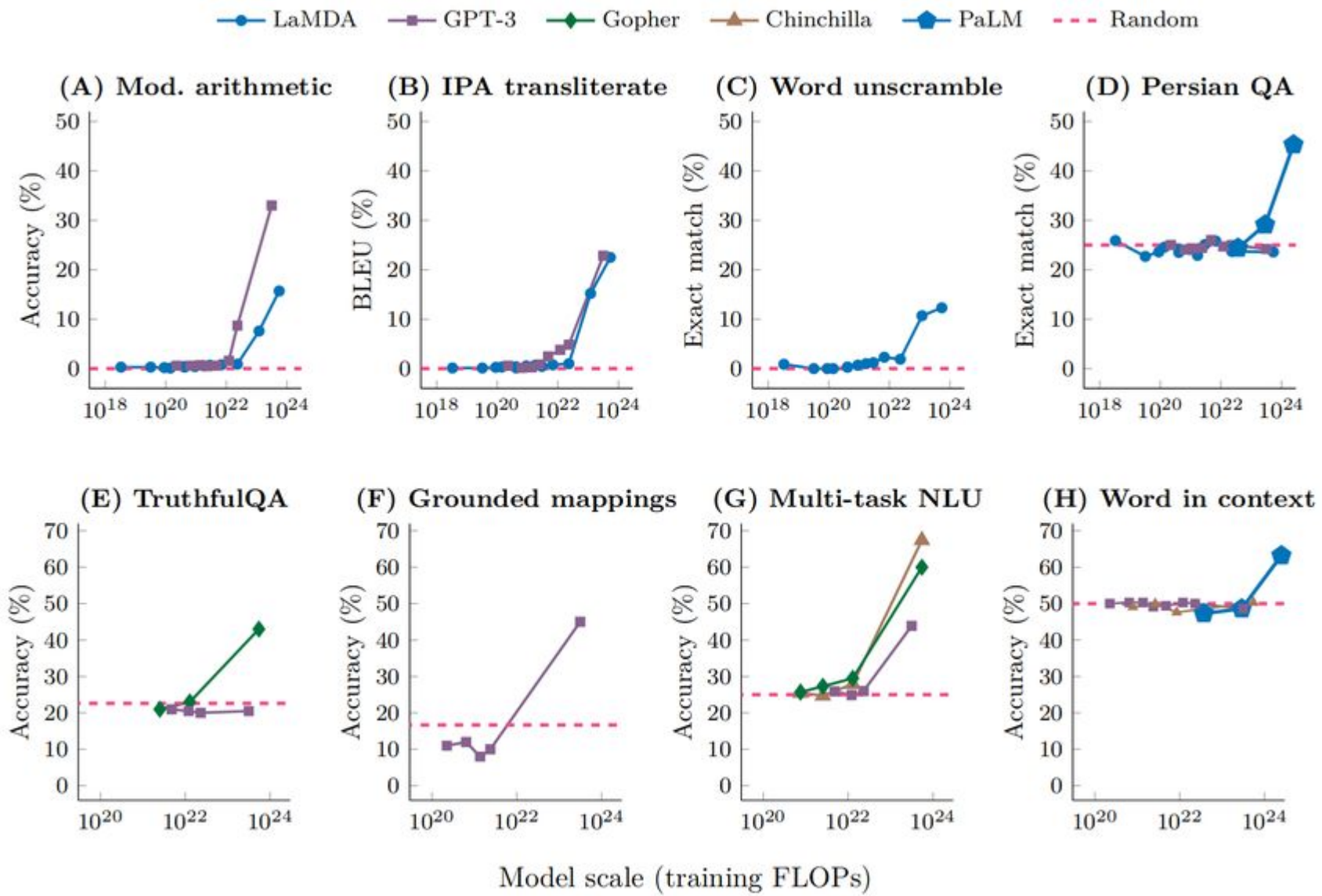
	GB200 NVL72	HGX B200	HGX B100
Blackwell GPUs	72	8	8
FP4 Tensor Core	1,440 petaFLOPS	144 petaFLOPS	112 petaFLOPS
FP8/FP6/INT8	720 petaFLOPS	72 petaFLOPS	56 petaFLOPS
Fast Memory	Up to 30 TB	up to 1.5 TB	Up to 1.5TB
Aggregate Memory Bandwidth	Up to 600 TB/s	Up to 64 TB/s	Up to 64 TB/s
Aggregate NVLink Bandwidth	130 TB/s	14.4 TB/s	14.4 TB/s
CPU Cores	2592 Arm Neoverse V2 cores	-	-
Per GPU Specifications			
FP4 Tensor Core	20 petaFLOPS	18 petaFLOPS	14 petaFLOPS
FP8/FP6 Tensor Core	10 petaFLOPS	9 petaFLOPS	7 petaFLOPS
INT8 Tensor Core	10 petaOPS	9 petaOPS	7 petaOPs
FP16/BF16 Tensor Core	5 petaFLOPS	4.5 petaFLOPS	3.5 petaFLOPS
TF32 Tensor Core	2.5 petaFLOPS	2.2 petaFLOPS	1.8 petaFLOPS
FP64 Tensor Core	45 teraFLOPS	40 teraFLOPS	30 teraFLOPS
GPU memory | Bandwidth	Up to 192 GB HBM3e | Up to 8 TB/s		
Multi-Instance GPU (MIG)	7		

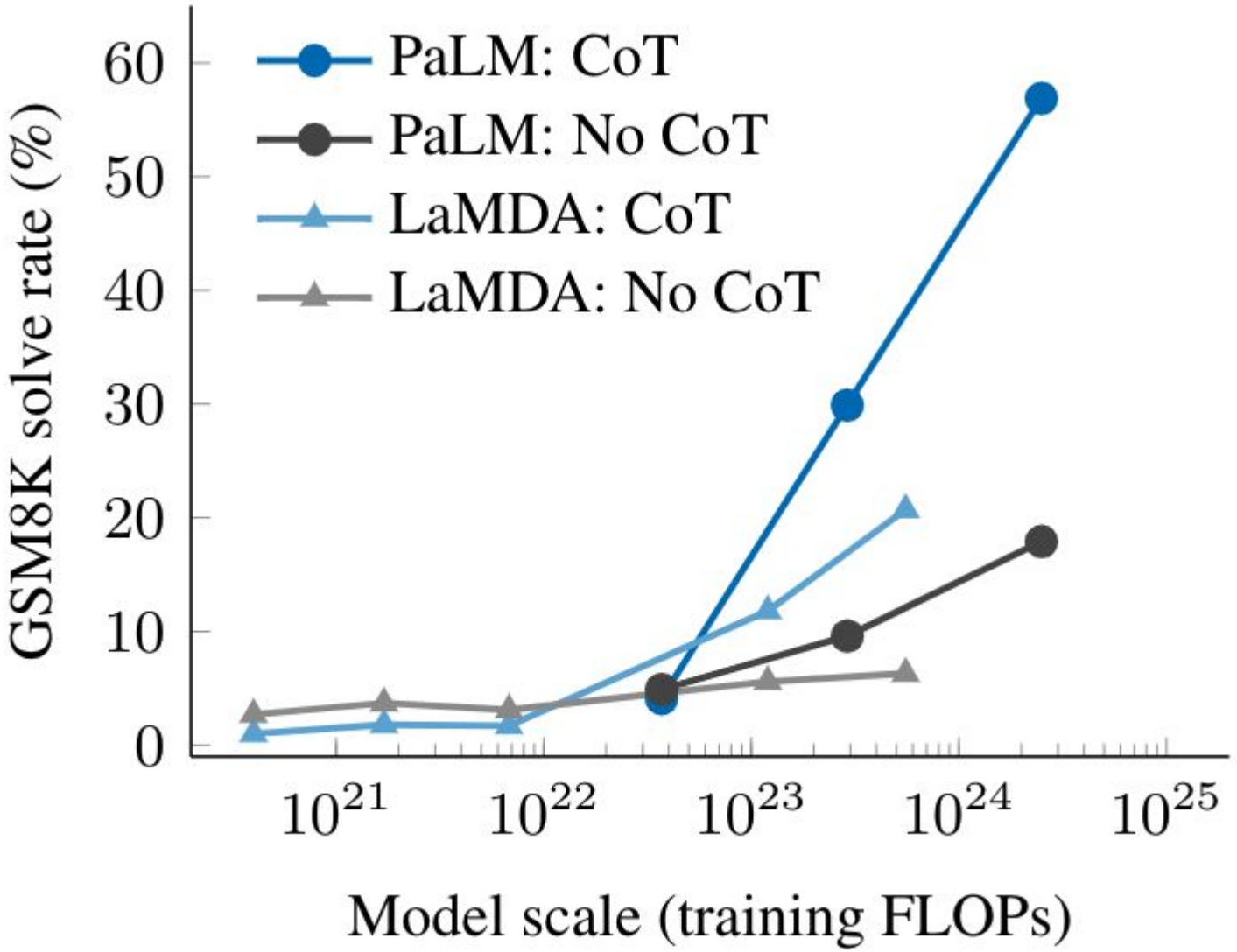
20 Peta flops(FP4)/s => 50 Akseleratorer gir 1 Exaflop/s.

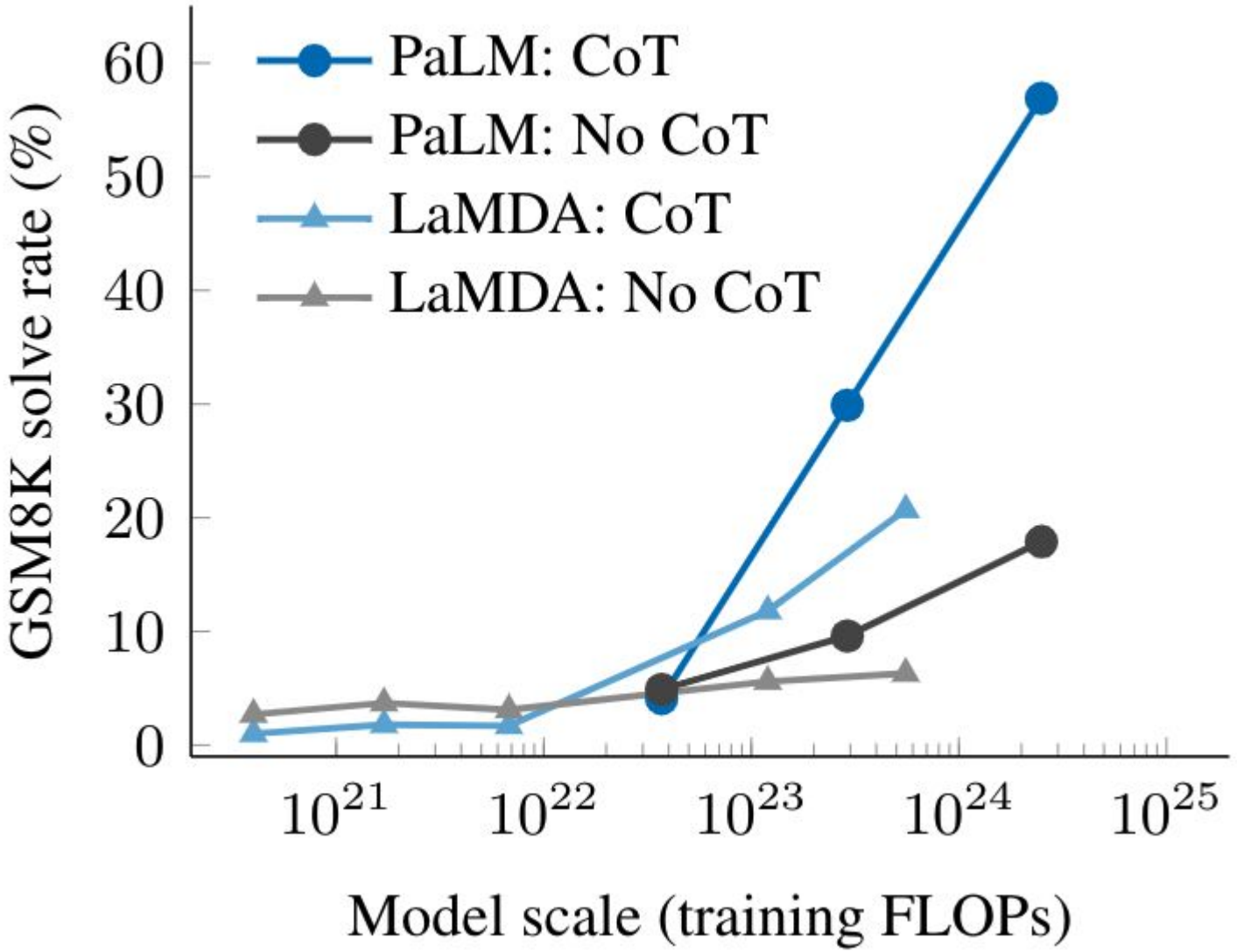
KI trening - maskinkraft - gjennombrudd

Hvorfor skjedde det så fort ?
Nevrale nett er ikke noe nytt.




LaMDA
GPT-3
Gopher
Chinchilla
PaLM
Random
(A) Mod. arithmetic
(B) IPA transliterate
(C) Word unscramble
(D) Persian QA
(E) TruthfulQA
(F) Grounded mappings
(G) Multi-task NLU
(H) Word in context
Accuracy (%)
BLEU (%)
Exact match (%)
Model scale (training FLOPs)


PaLM: CoT
PaLM: No CoT
LaMDA: CoT
LaMDA: No CoT
GSM8K solve rate (%)
0
10
20
30
40
50
60
10^{21}
10^{22}
10^{23}
10^{24}
10^{25}
Model scale (training FLOPs)


PaLM: CoT
PaLM: No CoT
LaMDA: CoT
LaMDA: No CoT
GSM8K solve rate (%)
0
10
20
30
40
50
60
10^{21}
10^{22}
10^{23}
10^{24}
10^{25}
Model scale (training FLOPs)

KI trening har en kost

Exa = 10^{18} => 10^{24} / 10^{18} = 10^{6} sekunder ≈ 12 dager.

LUMI = ⅓ Exa => 12*3 = 36 d
1.7 Mrd NOK
5 år levetid = 1825 d.

(36d / 1825d) * 1.7e9 NOK = 33 Mill NOK regnet for bare maskinkost.

Antar vi kapitalkost (1.7Mrd) + Strøm og operasjon like mye blir det 66 Mill NOK.

Skjermdumper fra en Youtubevideo som går gjennom maskinlæring, QR under er ep.1.