Transformers: Understanding the Pipeline

Input $\rightarrow$ Computation $\rightarrow$ Output $\rightarrow$ WHY

Week 5: Transformers

Complete Example: Predicting the Next Word

INPUT: "The cat sat on the ___"

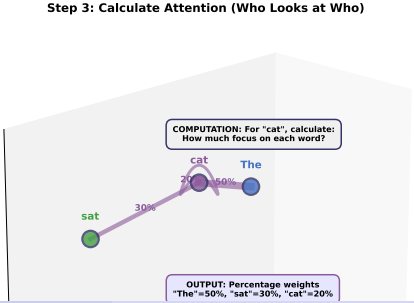
COMPUTATION (6 Steps):

1. Words → Numbers [0.2, 0.5, ...]
2. Add position: word 1, 2, 3, 4, 5
3. **Attention:** Each word looks at context
   - "sat" needs "cat" (WHAT sat?)
   - "on" needs "sat" (sitting ON something)
4. **Multi-Head Attention:**
   - Head 1: Grammar ("on the" → needs noun)
   - Head 2: Meaning (cat + sat → furniture)
   - Head 3: Position (final word prediction)
   - Head 4: Relations (cat sits ON things)
5. Combine all 4 perspectives
6. Predict next word

OUTPUT: Top predictions

- "mat": 85%
- "floor": 10%
- "table": 3%
- "rug": 2%

Result: "The cat sat on the **mat**"



The Simple Goal

INPUT:

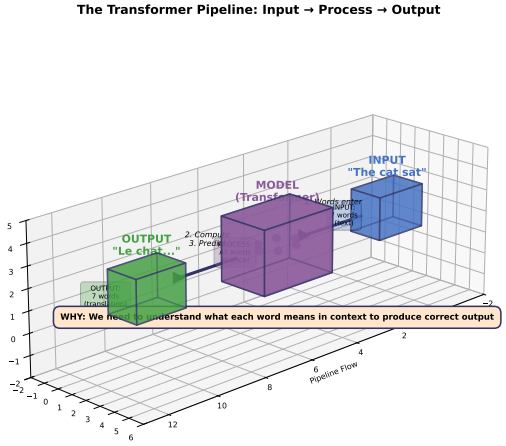
- Text: “The cat sat on the mat”
- 7 words (English)

OUTPUT:

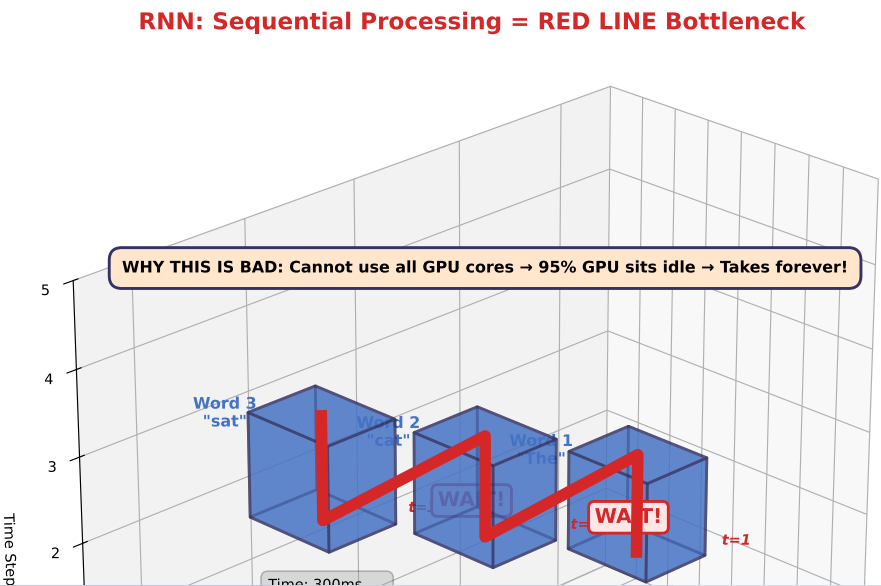
- Text: “Le chat était assis sur le tapis”
- 7 words (French)

THE TASK:

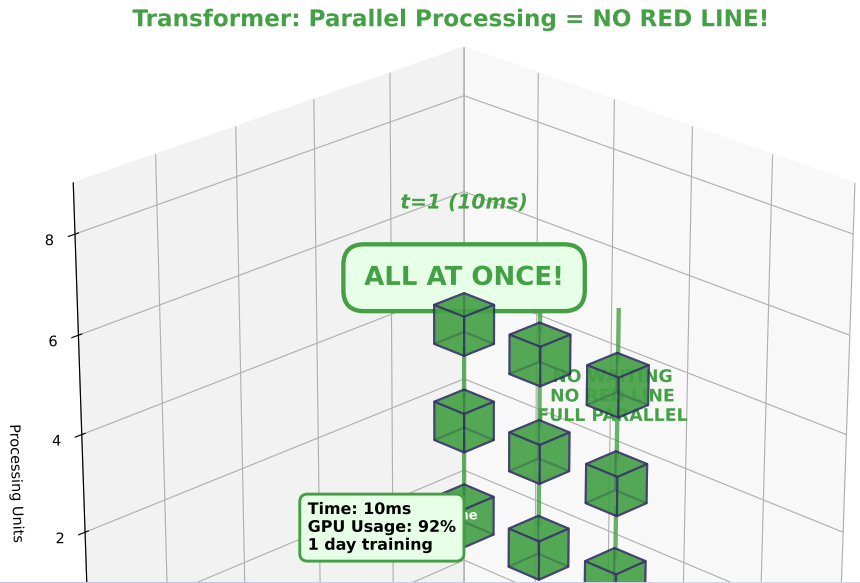
- Translate
- Predict next word
- Answer questions

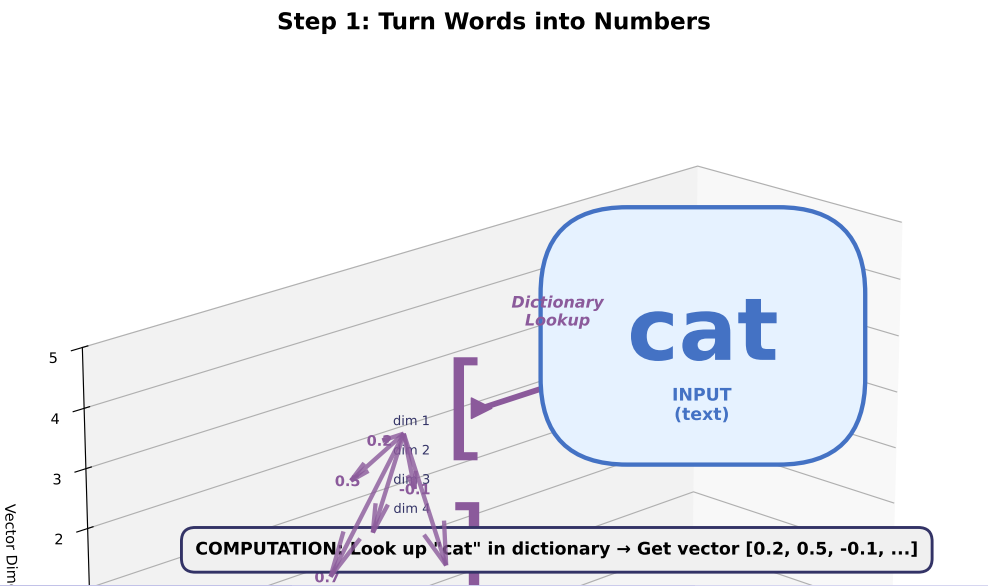


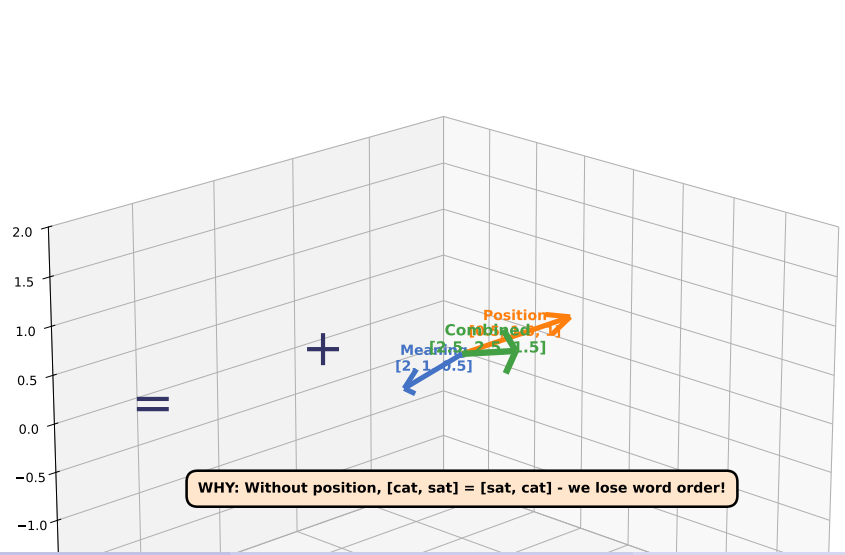
The Old Way: RNN (Sequential Processing)

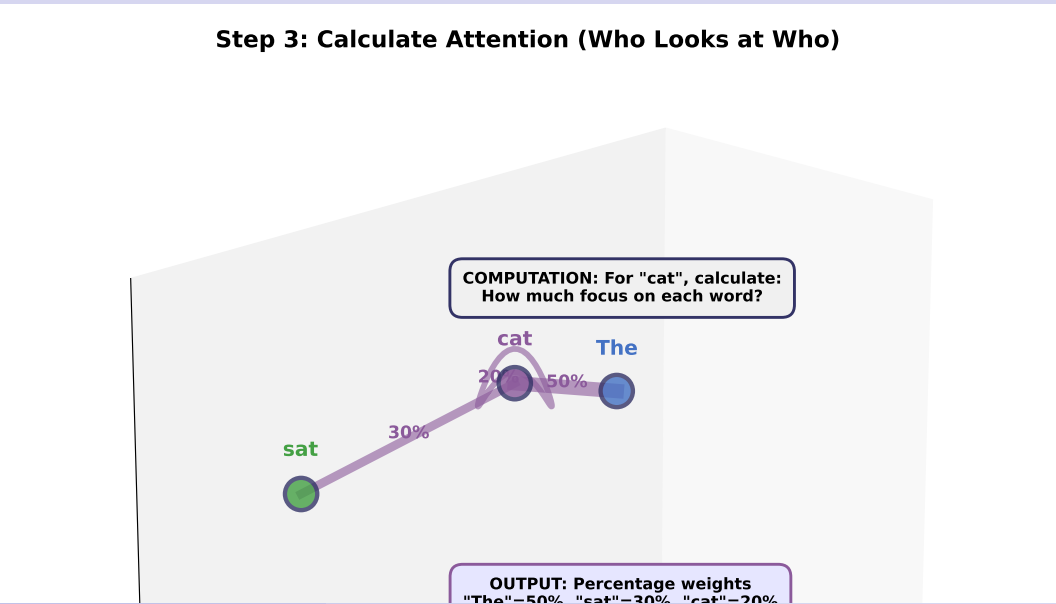


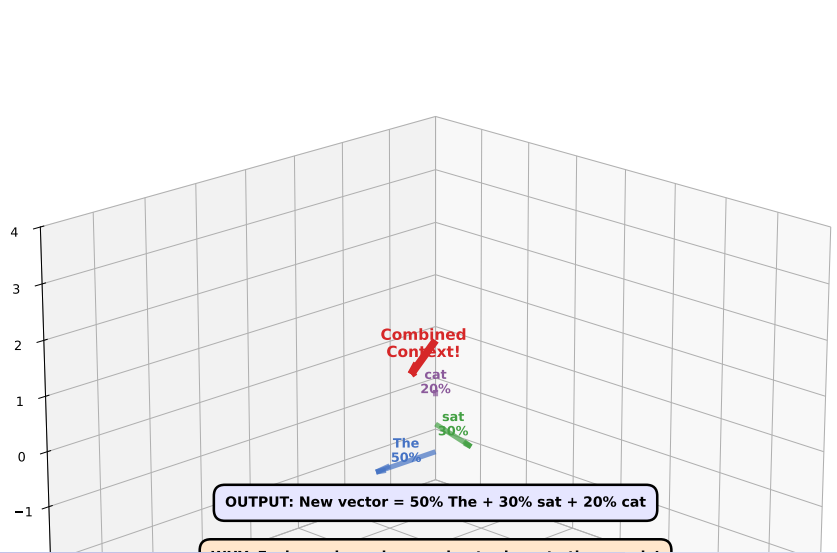
The New Way: Transformer (Parallel Processing)



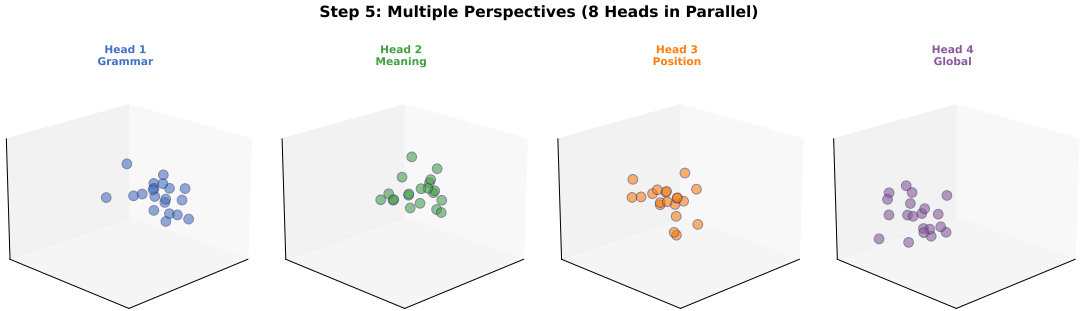

Step 1: Turn Words into Numbers
Dictionary Lookup
cat
INPUT (text)
dim 1
dim 2
dim 3
dim 4
0.2
0.5
-0.1
0.7
5
4
3
2
Vector Dim
COMPUTATION: Look up "cat" in dictionary → Get vector [0.2, 0.5, -0.1, ...]


Step 2: Add Position Information
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
Position
Combined
[2.5, 2.5, 1.5]
Meaning
[2, 1, 0.5]
WHY: Without position, [cat, sat] = [sat, cat] - we lose word order!


Step 3: Calculate Attention (Who Looks at Who)
COMPUTATION: For "cat", calculate:
How much focus on each word?
cat
The
20%
50%
30%
sat
OUTPUT: Percentage weights
"The"=50%, "sat"=30%, "cat"=20%


Step 4: Combine Information (Weighted Average)
4
3
2
1
0
−1
Combined Context!
cat 20%
sat 30%
The 50%
OUTPUT: New vector = 50% The + 30% sat + 20% cat

Step 5: Multiple Perspectives (Multi-Head Attention)



Step 6: Final Prediction

INPUT: Context-enriched vectors

- Each word knows about:
  - Its meaning
  - Its position
  - Related words (8 perspectives)

COMPUTATION:

- Feed through prediction layer
- Calculate probabilities for each possible next word

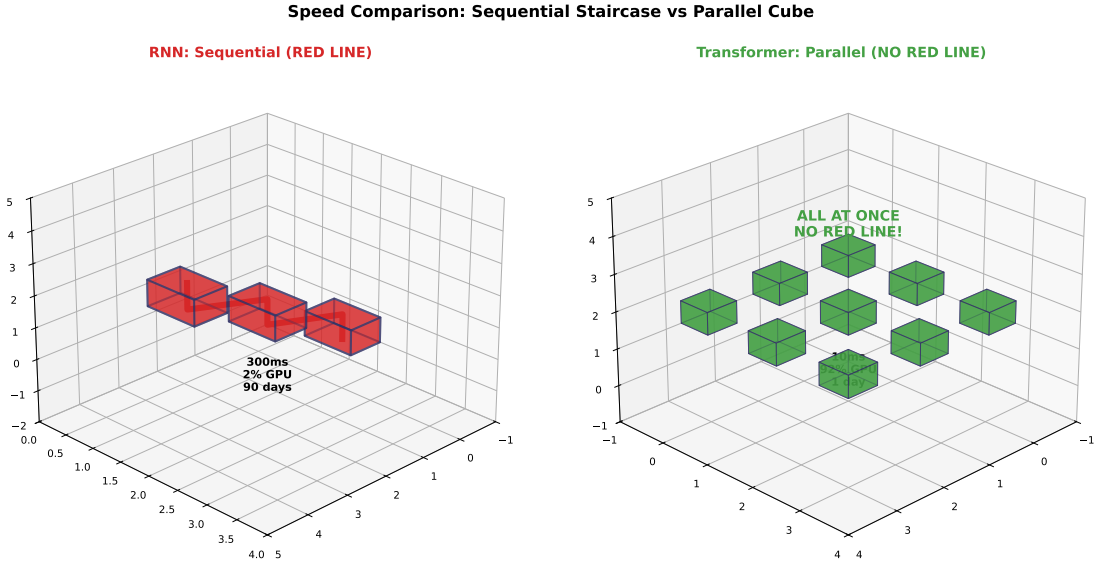
OUTPUT:

- Next word probabilities:
  - "Le": 85%
  - "The": 10%
  - Other: 5%
- Pick highest: "Le"

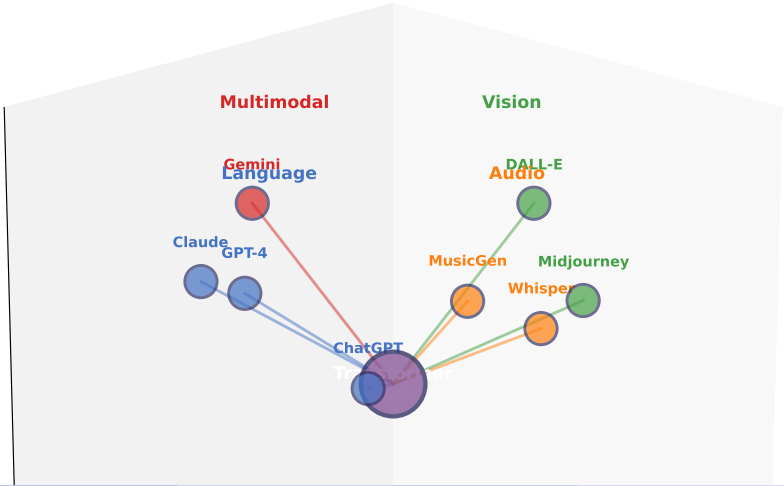
Result: Translation complete!

WHY: This is what we wanted all along - accurate prediction from context!

The Speed Secret: Parallel Processing



2024 Landscape: Transformers Power Everything



The Tradeoff: What We Gave Up

Advantages (PRO):

- 100x faster training
- Parallel processing
- 92% GPU utilization
- Works on any data type
- Enabled modern AI

Disadvantages (CON):

- More memory (quadratic)
- Needs more data
- Limited sequence length
- More complex to tune

THE DECISION: Speed + quality ¿ memory cost for modern AI

WHY ACCEPT TRADEOFF: Memory is cheap, time is expensive. Better to train fast even if uses more RAM.

Summary: The Pipeline Recap

The 6-Step Pipeline:

1. **Words → Numbers:** Dictionary lookup (embeddings)
2. **Add Position:** Vector addition (meaning + position)
3. **Calculate Attention:** Who looks at who? (percentage weights)
4. **Combine Information:** Weighted average (context-enriched)
5. **Multiple Perspectives:** 8 heads in parallel (grammar, meaning, position, ...)
6. **Predict Output:** Final layer (translation/next word)

KEY INSIGHT: All words processed in parallel!

- Result: 90 days → 1 day (90x speedup)
- Enabled: ChatGPT, GPT-4, DALL-E, Whisper, ...

Next Week: Pre-training & Fine-tuning - Now that training is fast, we can train HUGE models!

Transformers

Understanding the Pipeline

Input $\rightarrow$ Computation $\rightarrow$ Output $\rightarrow$ WHY

Questions?