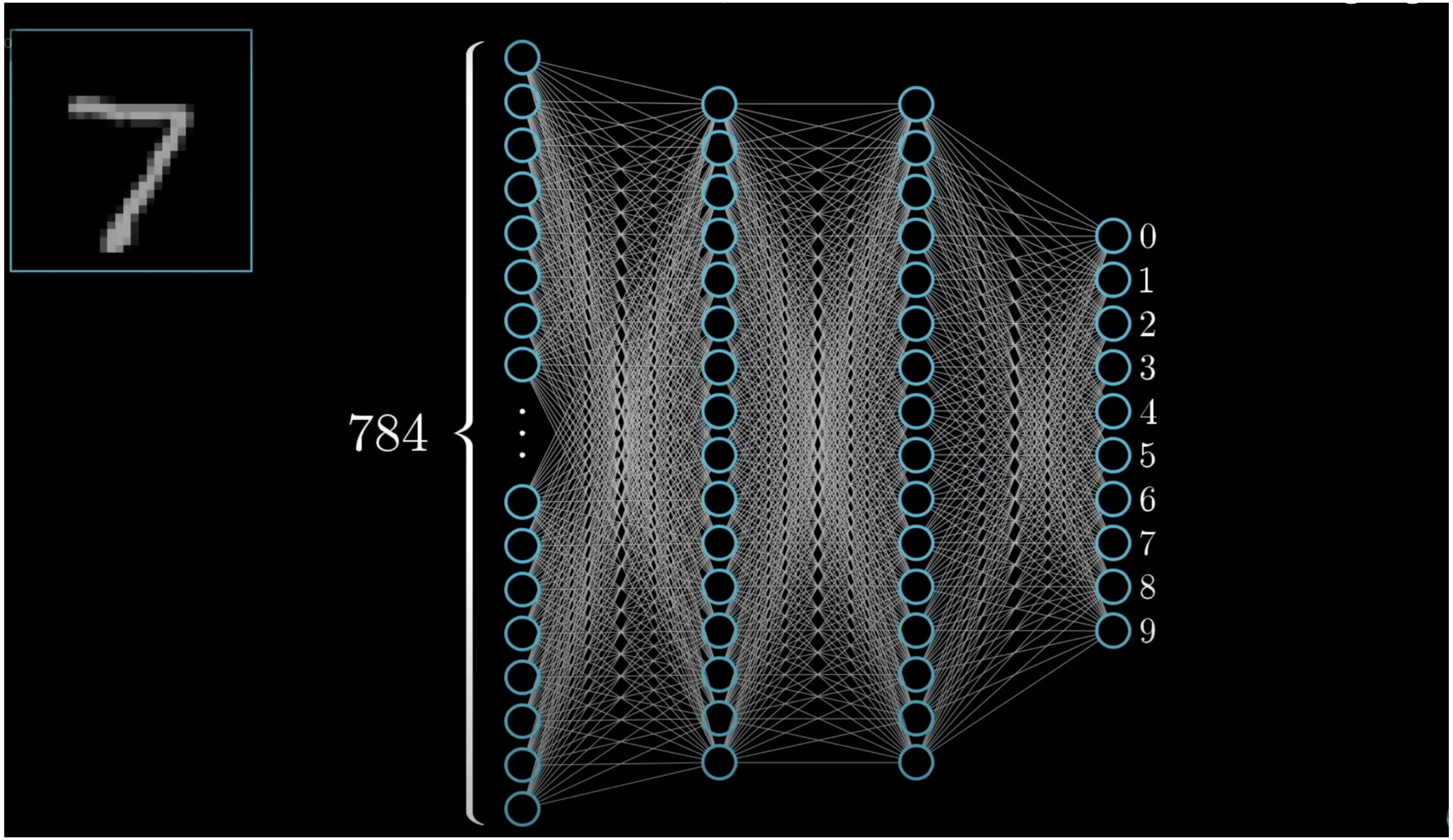
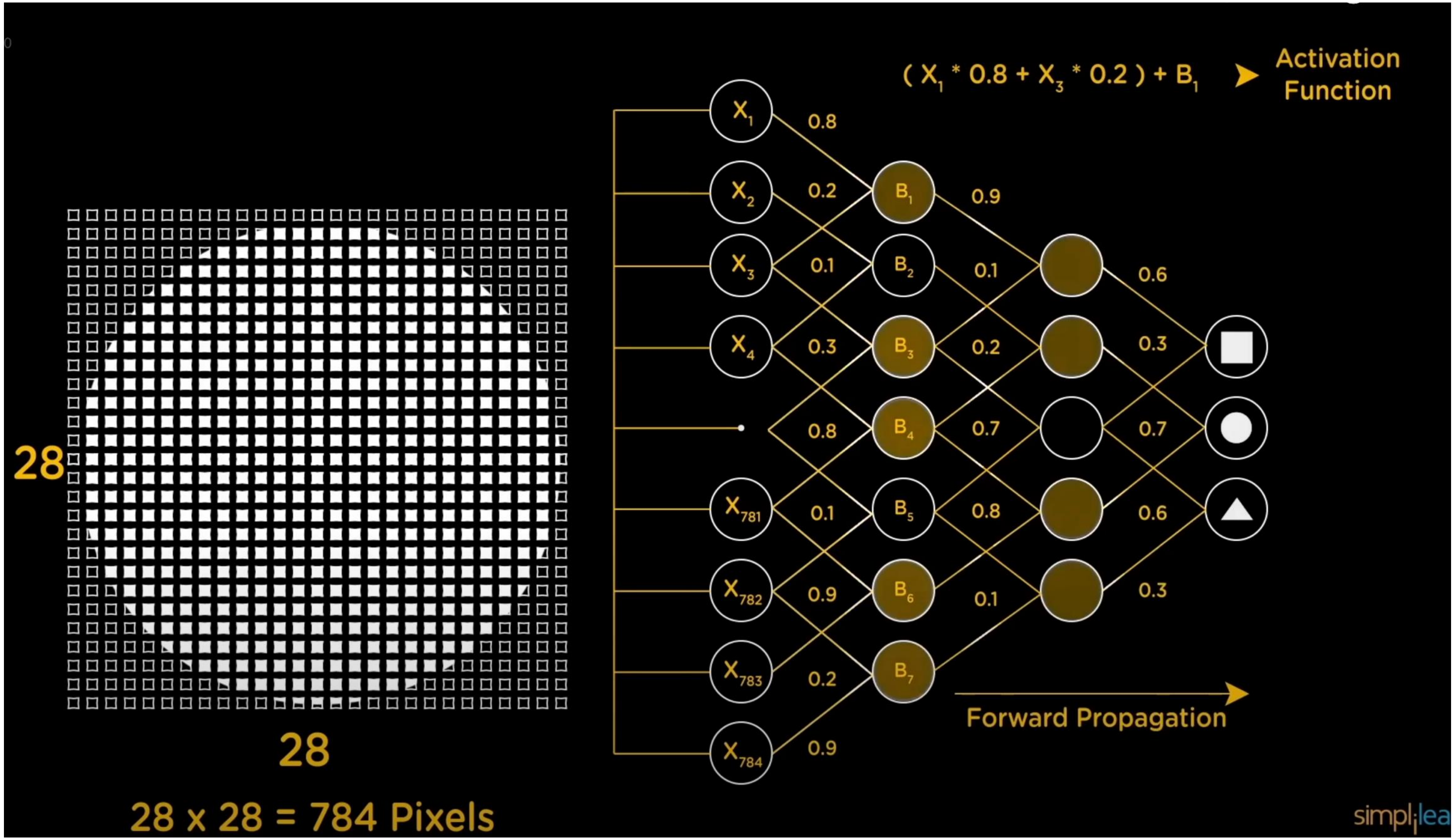
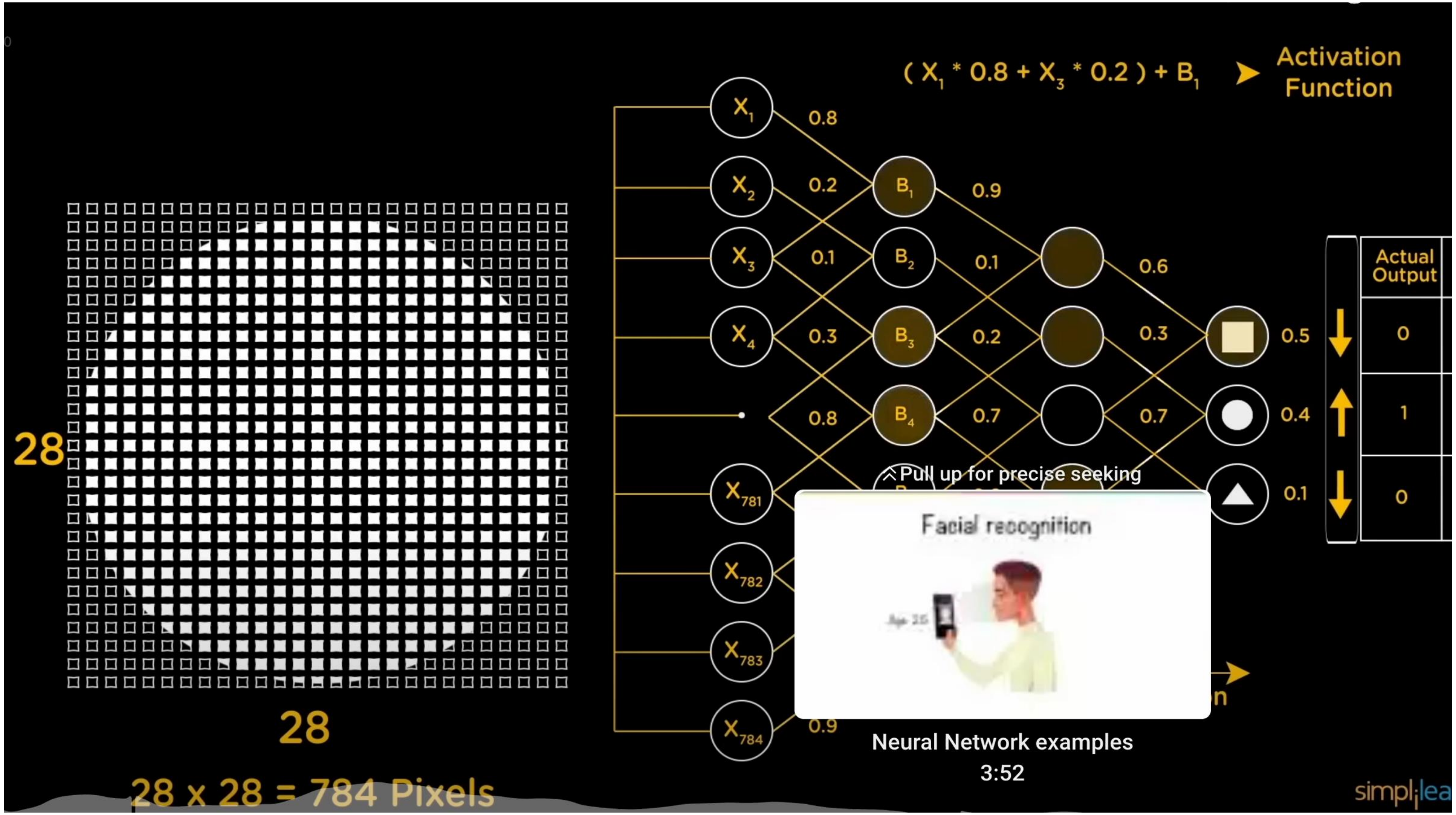
ALL YOU NEED TO KNOW ABOUT TRANSFORMERS


784
0
1
2
3
4
5
6
7
8
9


$(X_1 * 0.8 + X_3 * 0.2) + B_1$
Activation Function
28
28
X_1
X_2
X_3
X_4
X_{781}
X_{782}
X_{783}
X_{784}
0.8
0.2
0.1
0.3
0.8
0.1
0.9
0.2
0.9
B_1
B_2
B_3
B_4
B_5
B_6
B_7
0.9
0.1
0.2
0.7
0.8
0.1
0.6
0.3
0.7
0.6
0.3
Forward Propagation
28 x 28 = 784 Pixels

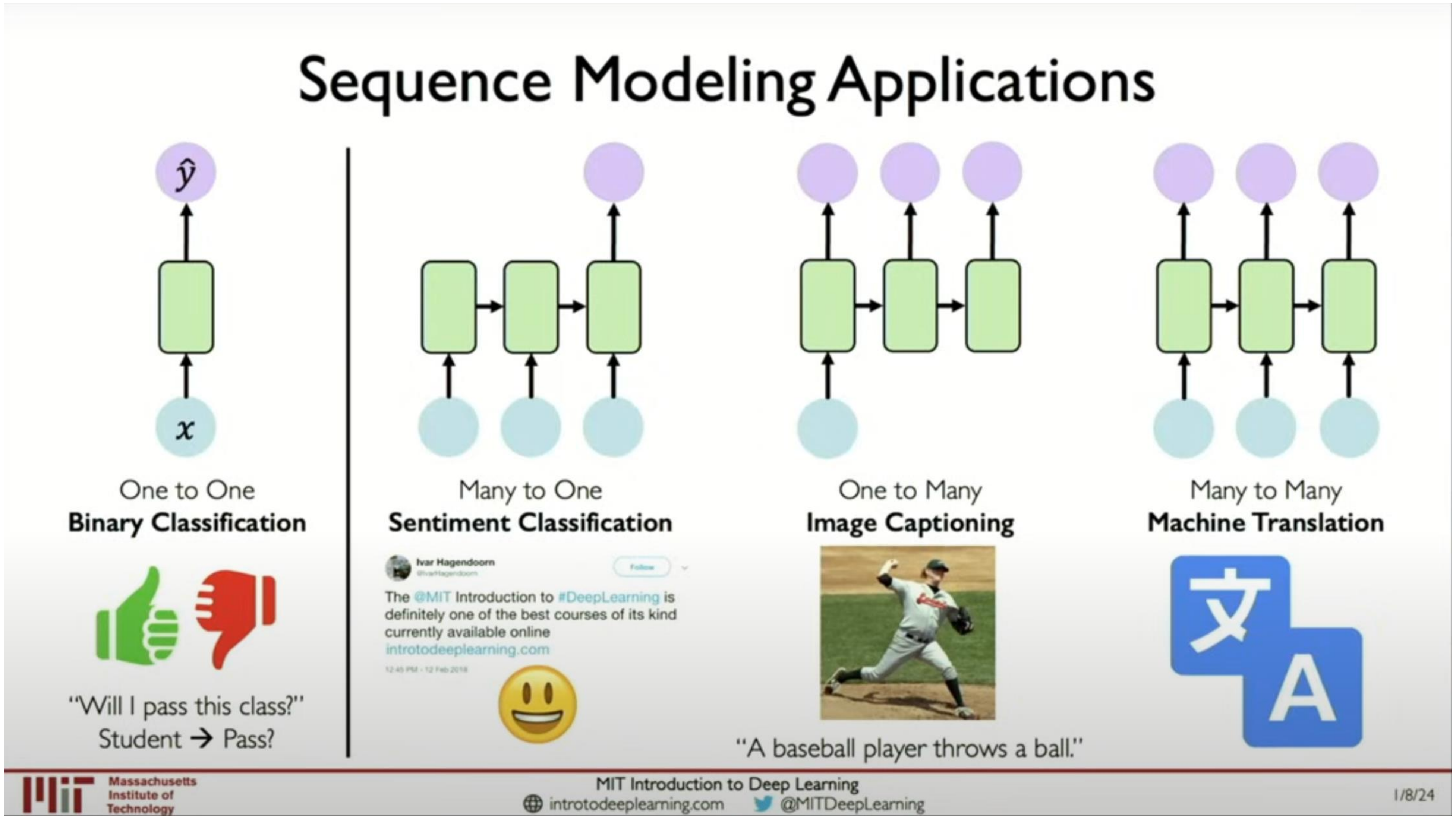

(X_1 * 0.8 + X_3 * 0.2) + B_1
Activation Function
X_1
X_2
X_3
X_4
X_{781}
X_{782}
X_{783}
X_{784}
B_1
B_2
B_3
B_4
0.8
0.2
0.1
0.3
0.8
0.9
0.9
0.1
0.2
0.7
0.6
0.3
0.7
0.5
0.4
0.1
Actual Output
0
1
0
28
28
28 x 28 = 784 Pixels
Pull up for precise seeking
Facial recognition
Neural Network examples
3:52
simplilea

WHY DO WE NEED SEQUENCE MODELS?

- Traditional machine learning models struggle with sequential data because they assume that each data point is independent of others. However, in tasks like predicting the next word in a sentence or recognizing speech, the order of data matters

- **Examples of sequence data**:
  - Text (sentences, paragraphs)
  - Time series data (stock prices, weather forecasts)
  - Speech/audio data (speech recognition)

- **Challenge**: We need models that can "remember" previous information while making predictions about the current step.



Sequence Modeling Applications

One to One
Binary Classification

"Will I pass this class?"
Student → Pass?

Many to One
Sentiment Classification

Ivar Hagendoorn
The @MIT Introduction to #DeepLearning is definitely one of the best courses of its kind currently available online
introtodeeplearning.com

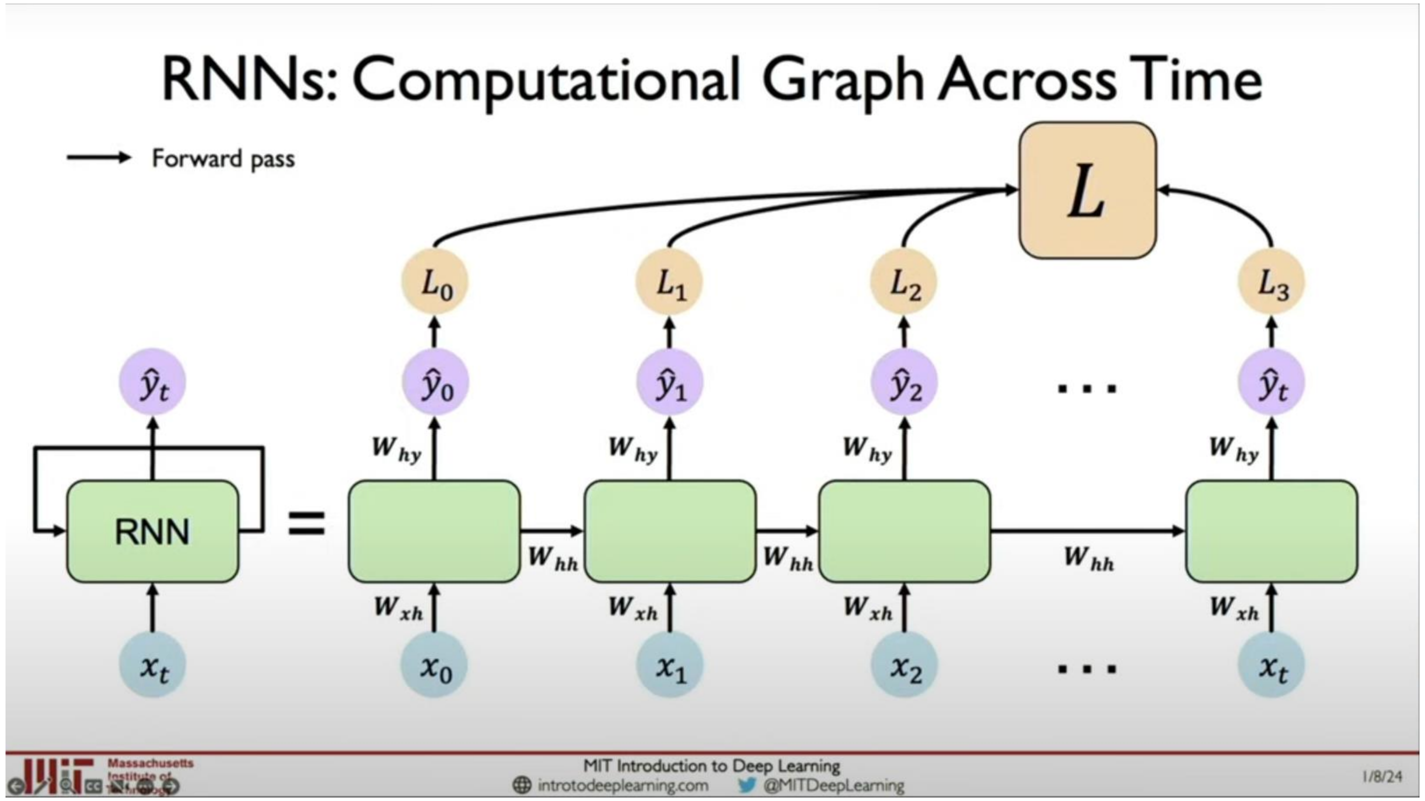
One to Many
Image Captioning

"A baseball player throws a ball."

Many to Many
Machine Translation

RECURRENT NEURAL NETWORKS (RNNS)

- RNNs are a type of neural network designed to work with sequential data. They "loop" through the data to retain information about previous time steps.



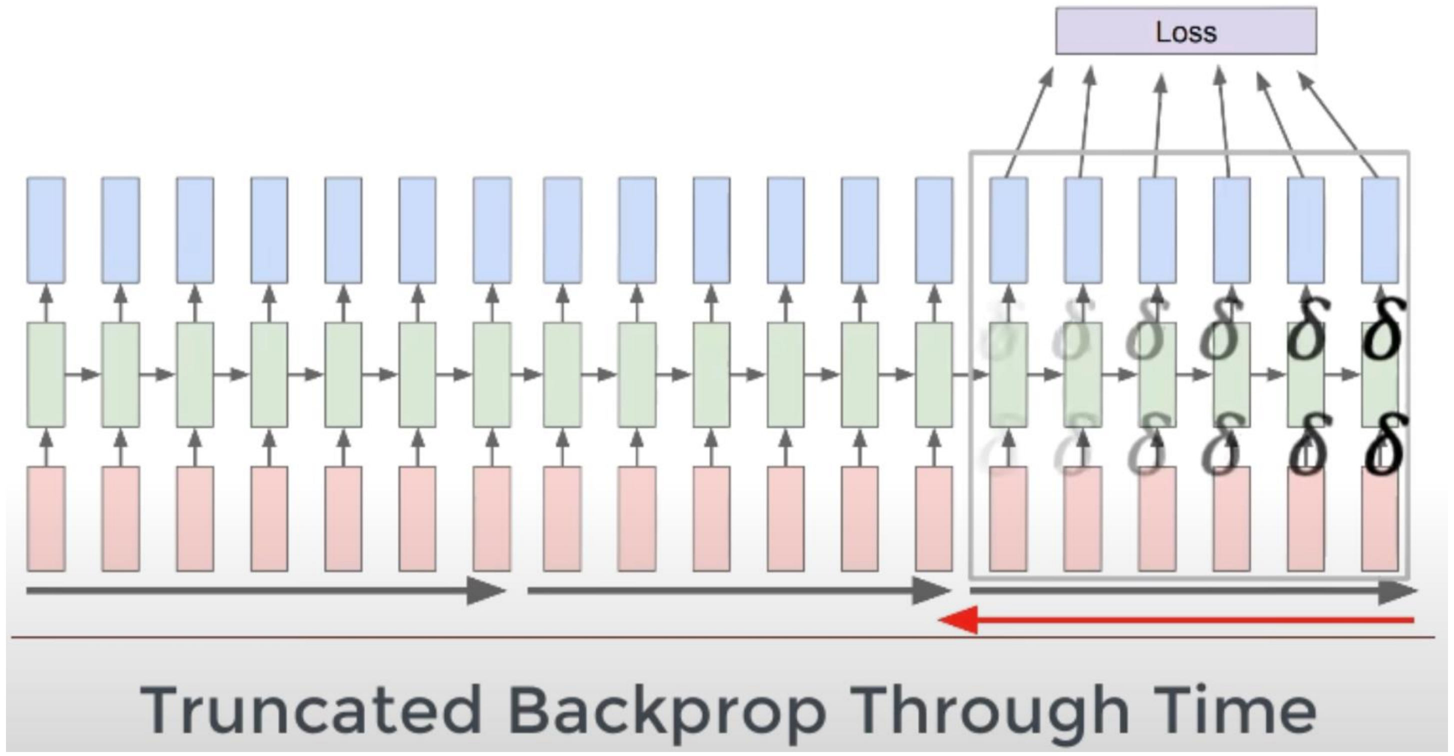
A BIT MORE ABOUT RNN

- **How it works**: RNNs have a hidden state that captures information from previous steps and passes it on to the next step. This way, each output depends on both the current input and the hidden state.
- **Architecture**: In each time step, the RNN receives the current input and the hidden state from the previous step. It updates the hidden state and generates an output.
- **Problem**:
  - Memory Loss
  - Training Issue
  - RNNs suffer from the **vanishing gradient problem**. As the network backpropagates through time (BPTT), the gradients can shrink to very small values, making it difficult for the model to learn long-range dependencies.

APPLICATIONS OF RNN

- **Language Modeling**: Predicting the next word in a sequence based on the previous words.
- **Sentiment Analysis**: Classifying a sentence as positive or negative by analyzing the sequence of words.
- **Sequence Classification**: Labeling entire sequences (e.g., a sequence of stock prices) into categories like "bullish" or "bearish.
- "**Example**: An RNN could predict the next word in "I love to drink ___" by analyzing the sequence of words leading up to the blank.

VANISHING / EXPLODING GRADIENT PROBLEM



LONG SHORT TERM MEMORY (LSTM)

- LSTMs are a type of RNN that solves the vanishing gradient problem by introducing **gates** that control the flow of information. This allows LSTMs to "remember" information over longer sequences.
- **Key components**:
  - **Cell state**: The memory of the network that flows through time steps, updated by the gates.
  - **Forget gate**: Decides which information to keep and which to discard from the previous cell state.
  - **Input gate**: Decides which information to add to the cell state from the current input.
  - **Output gate**: Decides which part of the cell state to output as the hidden state.

LONG SHORT TERM MEMORY (LSTM)

- **Advantage**: LSTMs can capture long-term dependencies in data, like remembering the beginning of a sentence even after processing many words.
- **Problem:** Too slow to train!

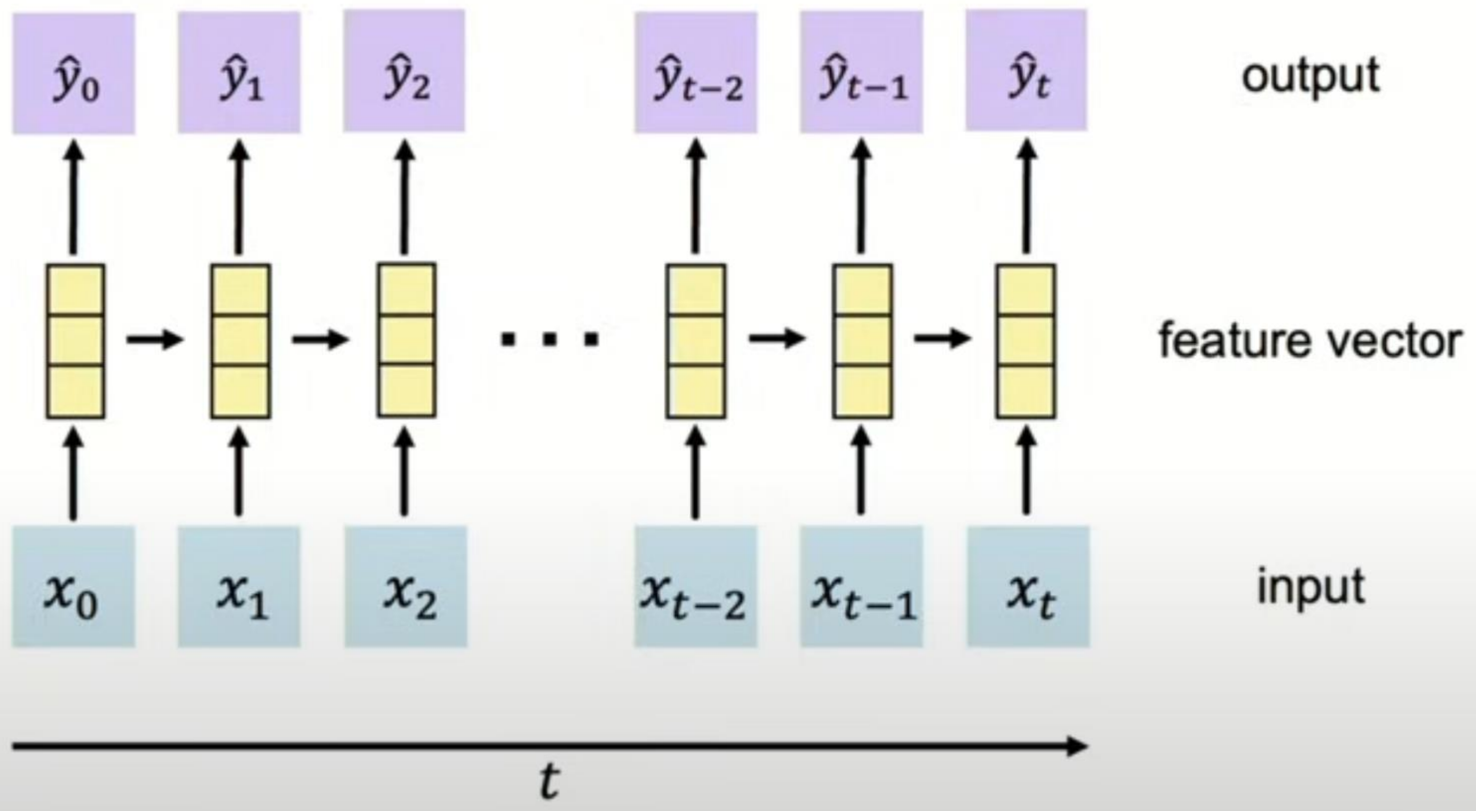
PROBLEMS WITH RNN & LSTM

- Encoding Bottleneck (Memory)
- No parallelization
- Too slow to train
- Information Loss

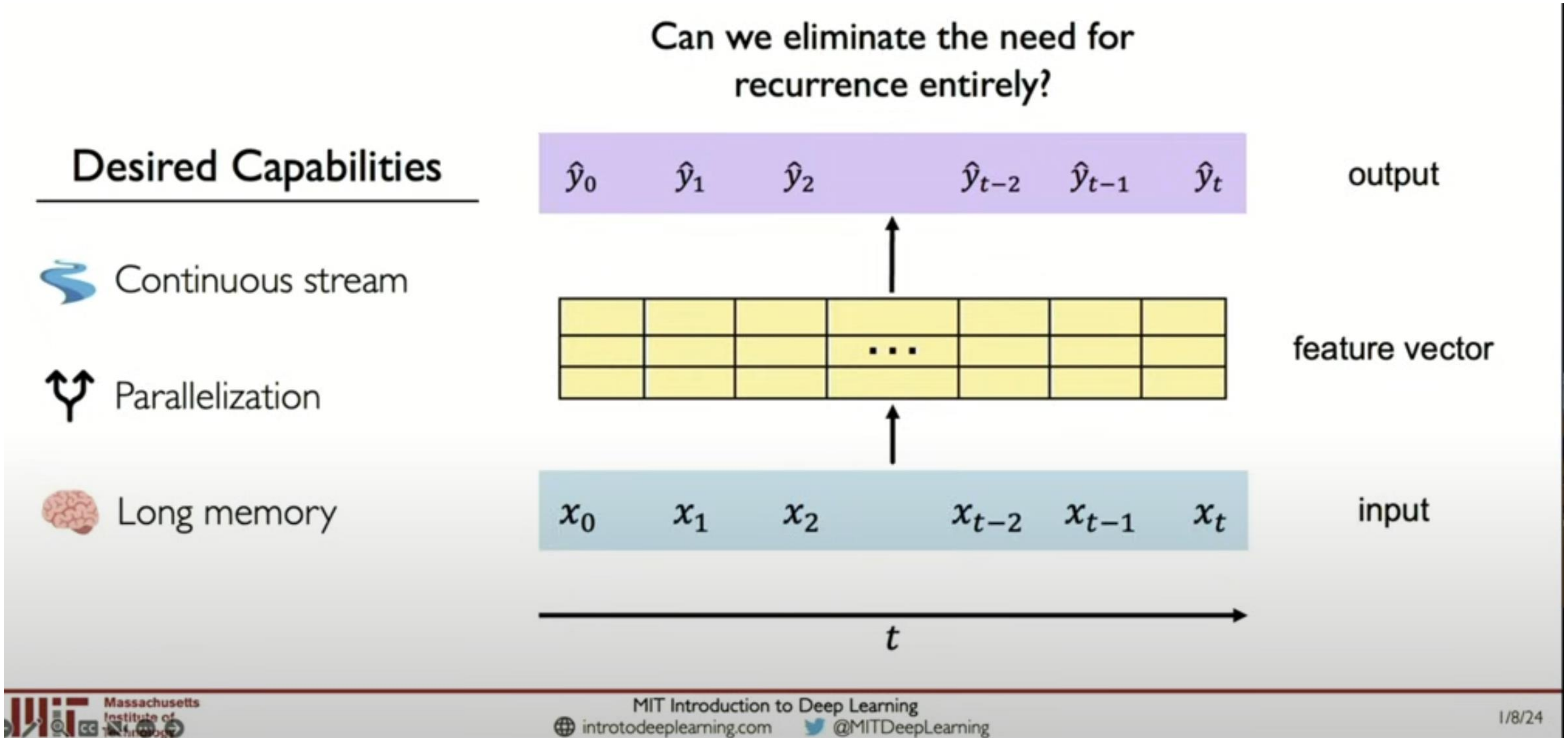
RNNs: recurrence to model sequence dependencies

Limitations of RNNs

- Encoding bottleneck
- Slow, no parallelization
- Not long memory



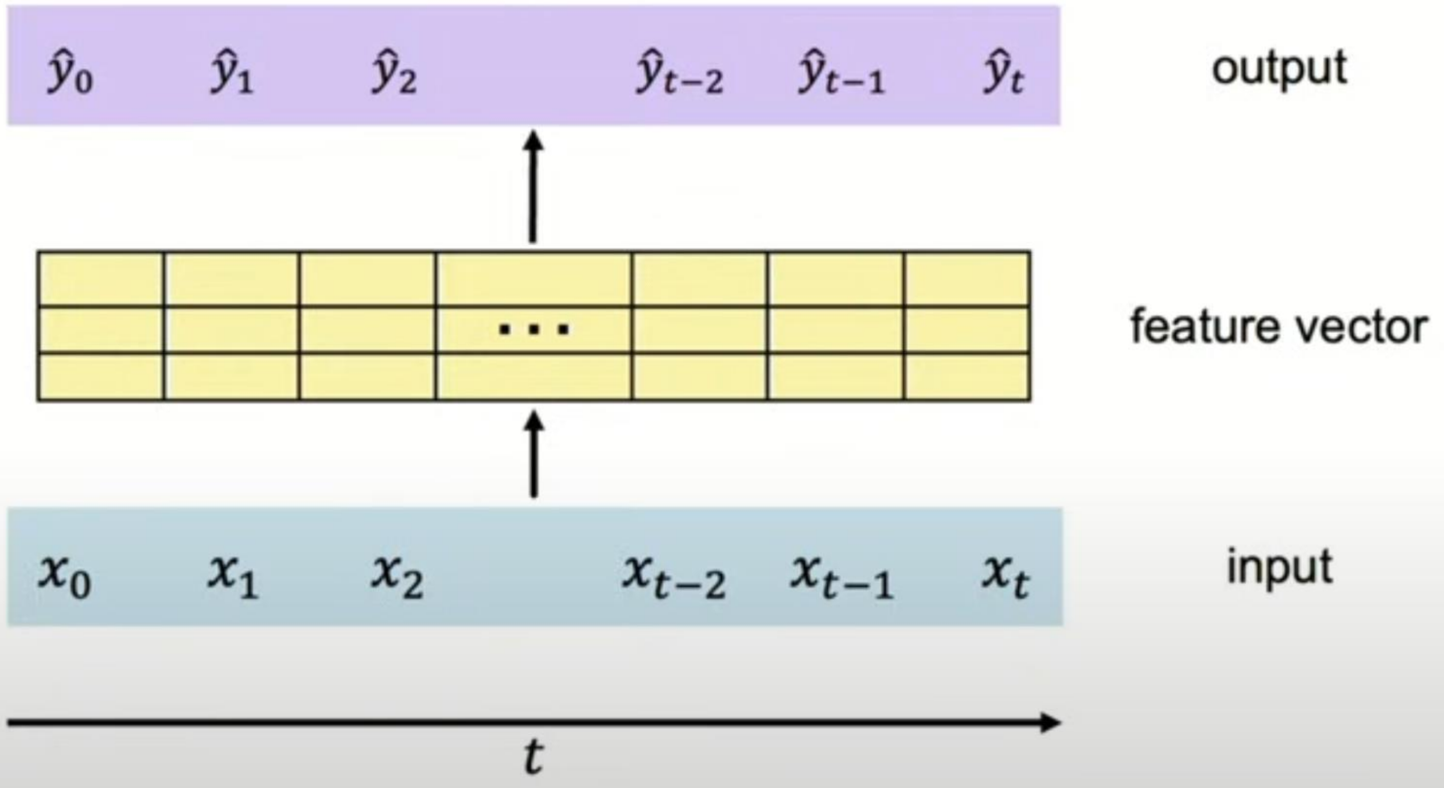
$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ $\hat{y}_{t-2}$ $\hat{y}_{t-1}$ $\hat{y}_t$
output
feature vector
x_0 x_1 x_2 x_{t-2} x_{t-1} x_t
input
t



Can we eliminate the need for recurrence entirely?

Desired Capabilities

- Continuous stream
- Parallelization
- Long memory


$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ $\hat{y}_{t-2}$ $\hat{y}_{t-1}$ $\hat{y}_t$
output
...
feature vector
x_0 x_1 x_2 x_{t-2} x_{t-1} x_t
input
t

Idea 1: Feed everything into dense network

✓ No recurrence

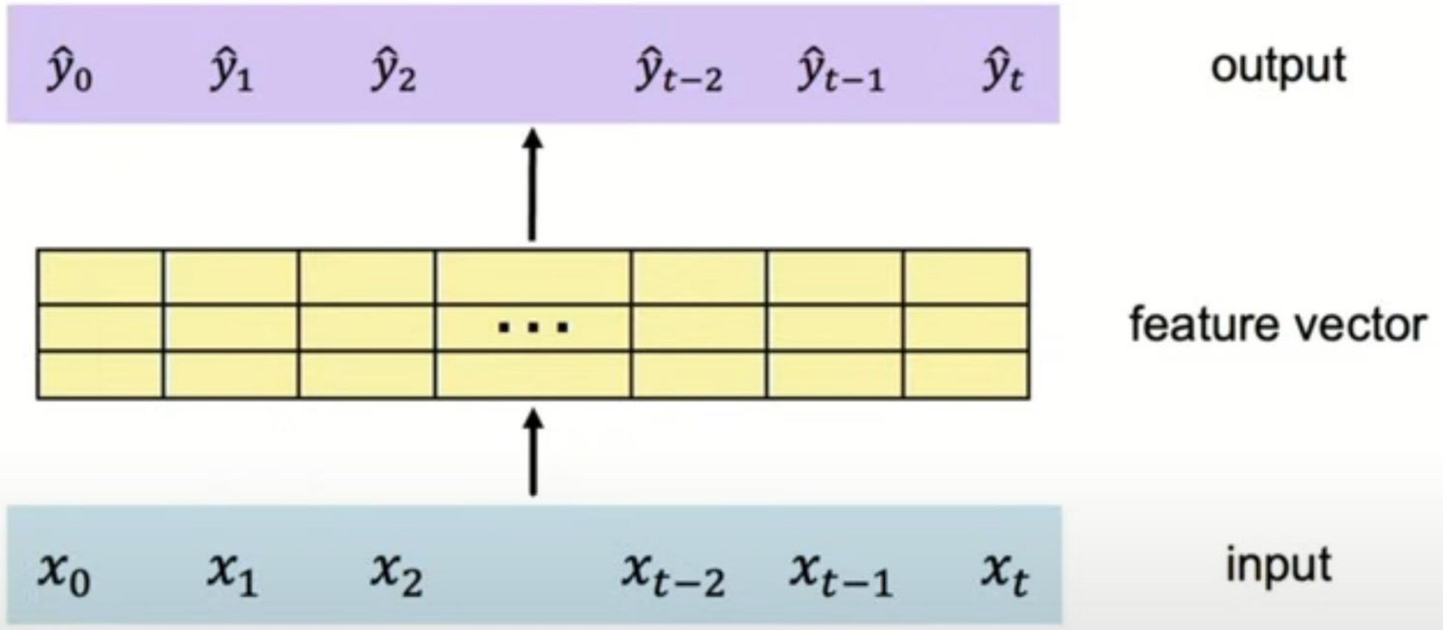
✗ Not scalable

✗ No order

✗ No long memory

Idea: Identify and attend to what's important

Can we eliminate the need for recurrence entirely?


$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ $\hat{y}_{t-2}$ $\hat{y}_{t-1}$ $\hat{y}_t$
output
...
feature vector
x_0 x_1 x_2 x_{t-2} x_{t-1} x_t
input

Intuition Behind Self-Attention

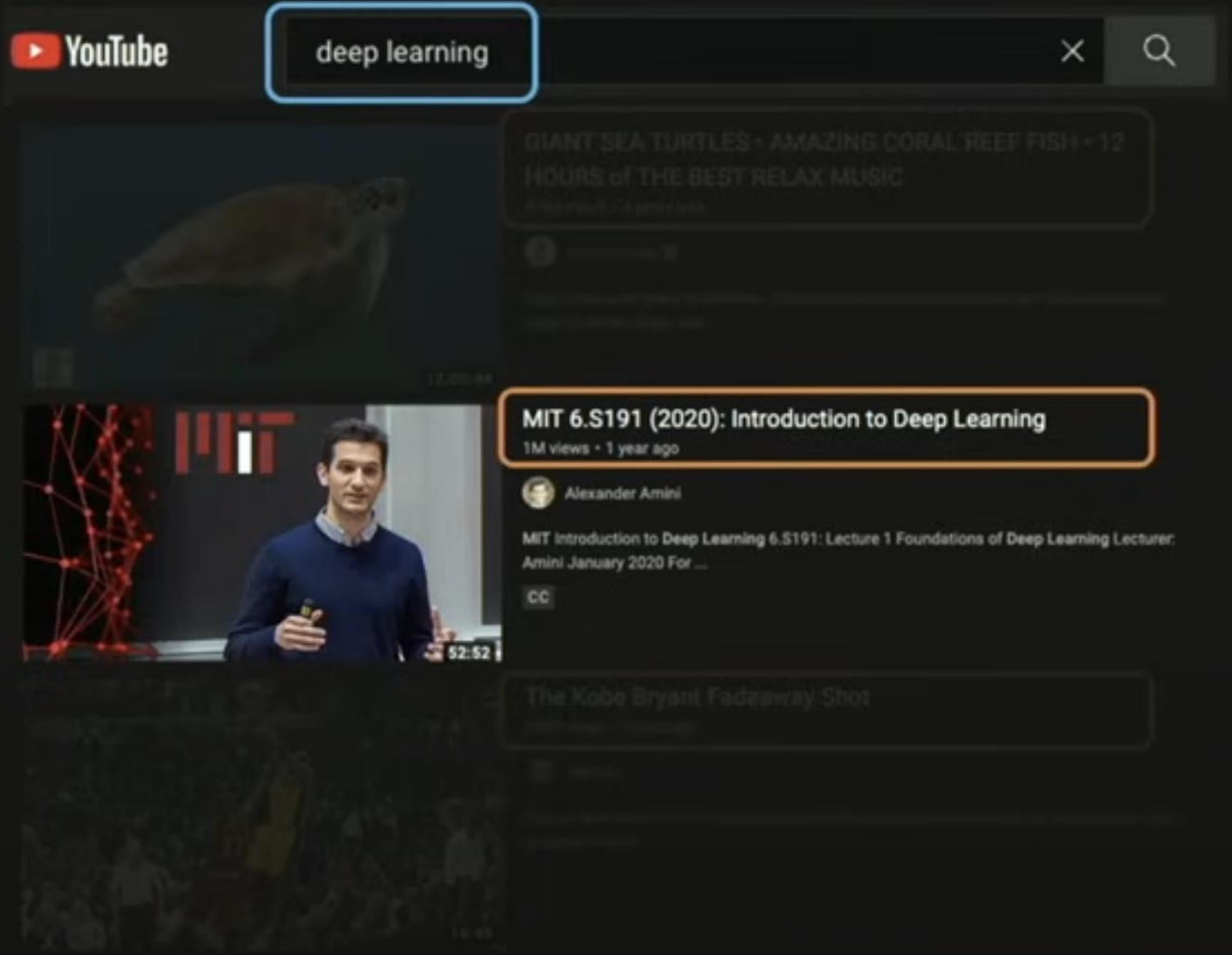
Attending to the most important parts of an input.

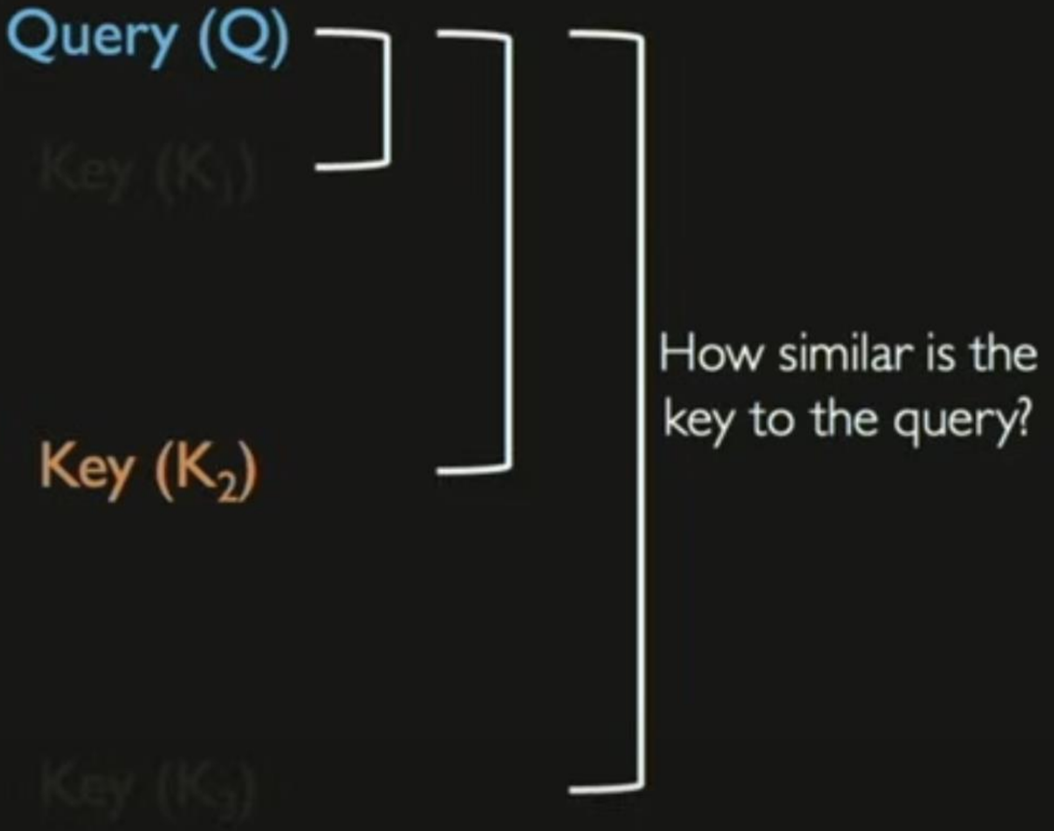
Intuition Behind Self-Attention

Attending to the most important parts of an input.

1. Identify which parts to attend to
2. Extract the features with high attention

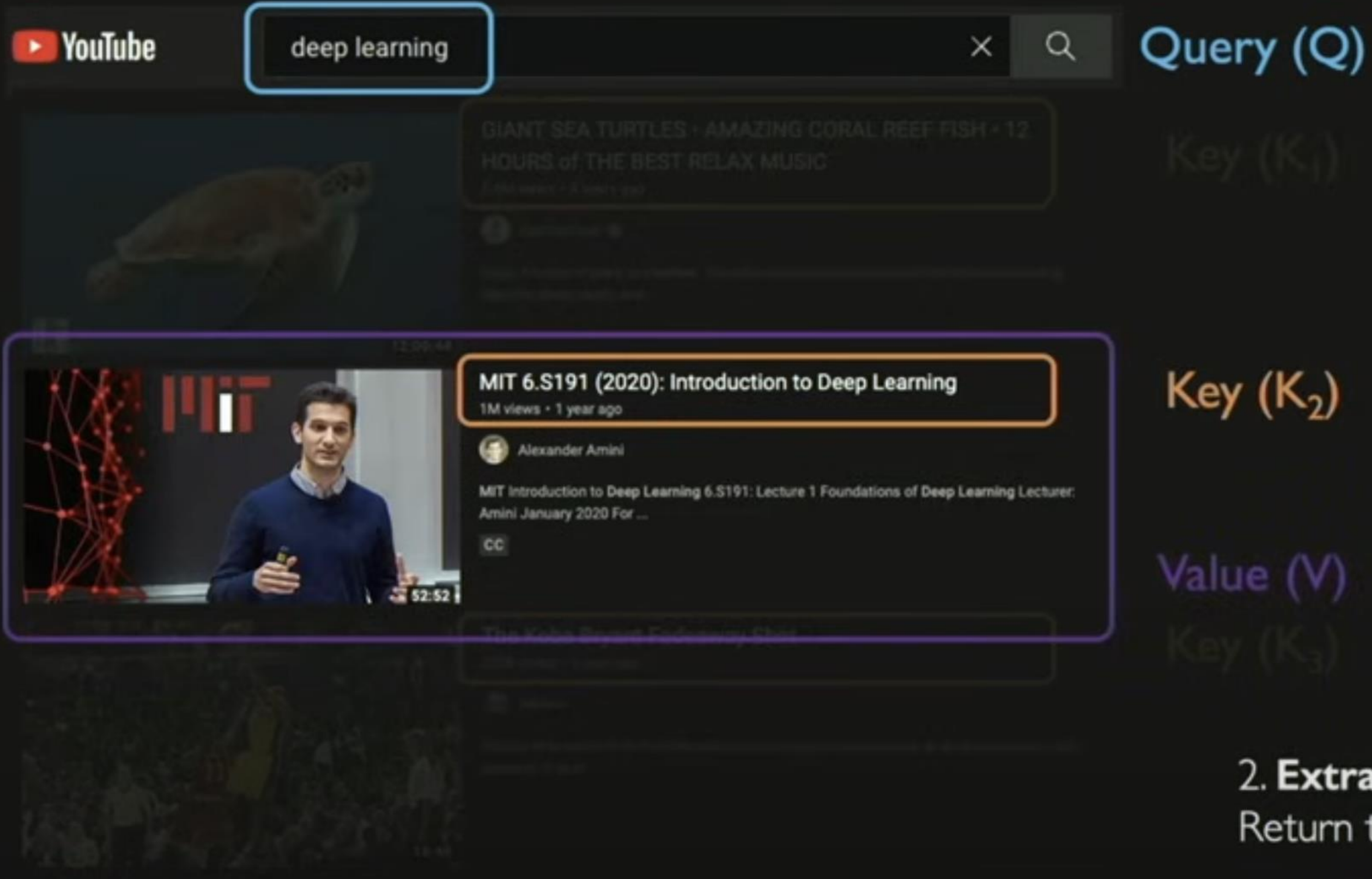
Understanding Attention with Search


YouTube
deep learning
GIANT SEA TURTLES • AMAZING CORAL REEF FISH • 12 HOURS of THE BEST RELAX MUSIC
MIT 6.S191 (2020): Introduction to Deep Learning
1M views • 1 year ago
Alexander Amini
MIT Introduction to Deep Learning 6.S191: Lecture 1 Foundations of Deep Learning Lecturer: Amini January 2020 For ...
CC
52:52
The Kobe Bryant Fadeaway Shot



1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search



GIANT SEA TURTLES • AMAZING CORAL REEF FISH • 12 HOURS of THE BEST RELAX MUSIC

MIT 6.S191 (2020): Introduction to Deep Learning
1M views • 1 year ago
Alexander Amini
MIT Introduction to Deep Learning 6.S191: Lecture 1 Foundations of Deep Learning Lecturer: Amini January 2020 For ...

The Kobe Bryant Fadeaway Shot

2. **Extract values based on attention:**
Return the values highest attention

MEET TRANSFORMERS

Transformer Components

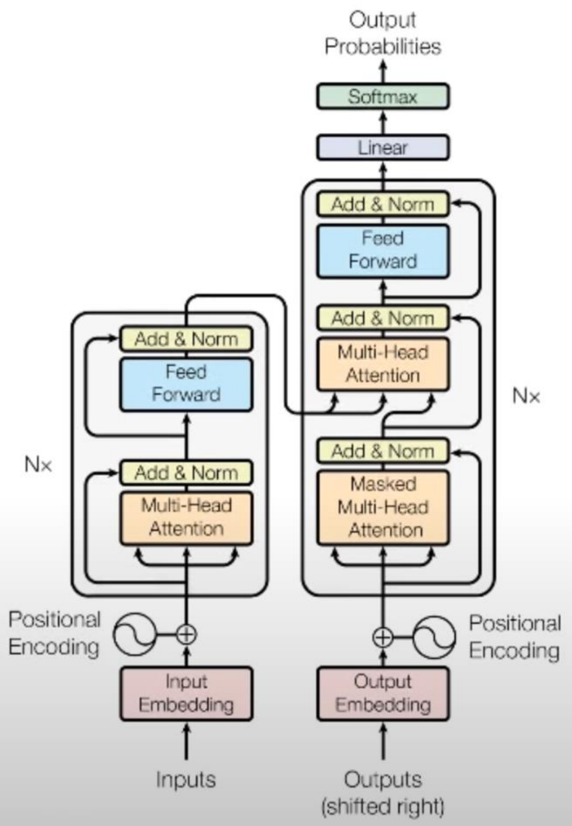


Figure 1: The Transformer - model architecture.

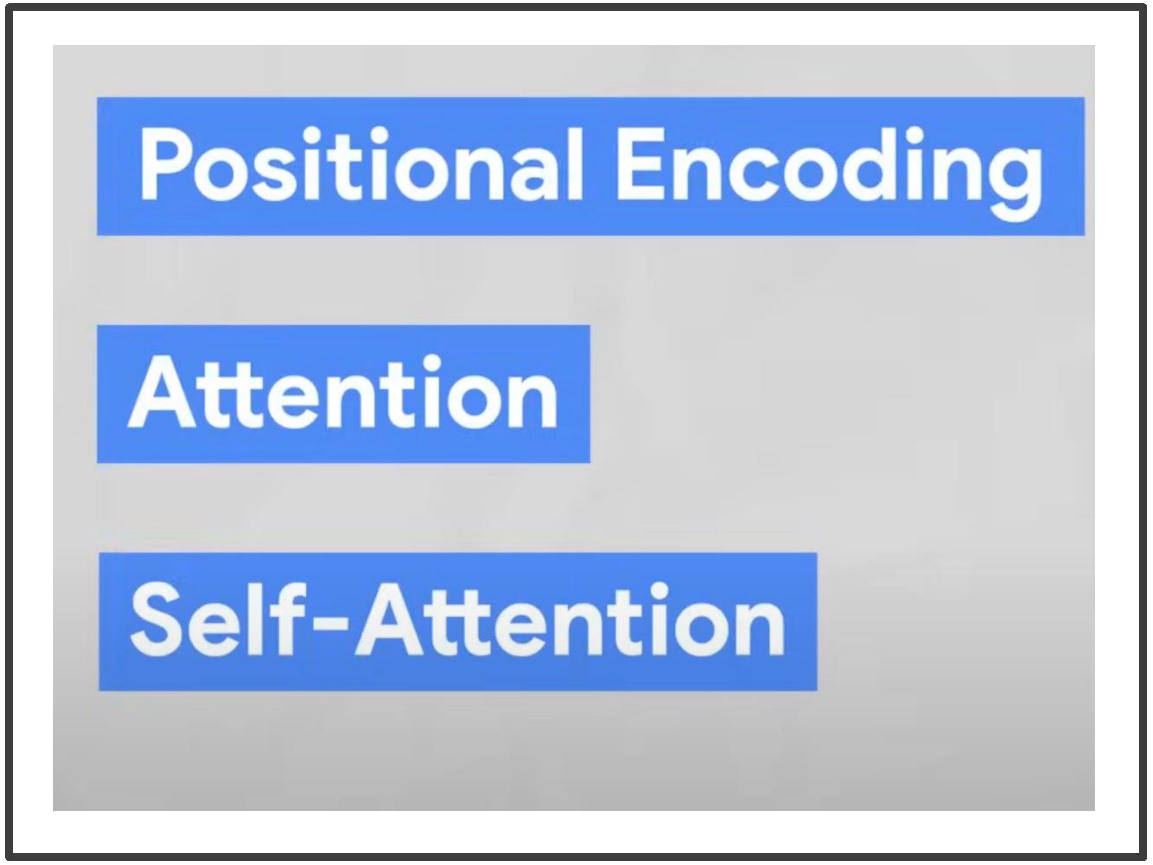
WHAT ARE TRANSFORMERS?

- **Transformer architecture**: Introduced by Vaswani et al. in 2017, transformers use the **self-attention mechanism**, which allows them to process the entire sequence at once rather than step-by-step.
- **Key idea**: Instead of relying on hidden states to remember information, transformers look at the entire sequence to decide how much attention to give to each part of the input. This makes them much faster and more efficient, especially for long sequences.

3 MAGICAL WORDS…

- **Transformers** are nothing but 3 magical words, can you guess?

3 MAGICAL WORDS…

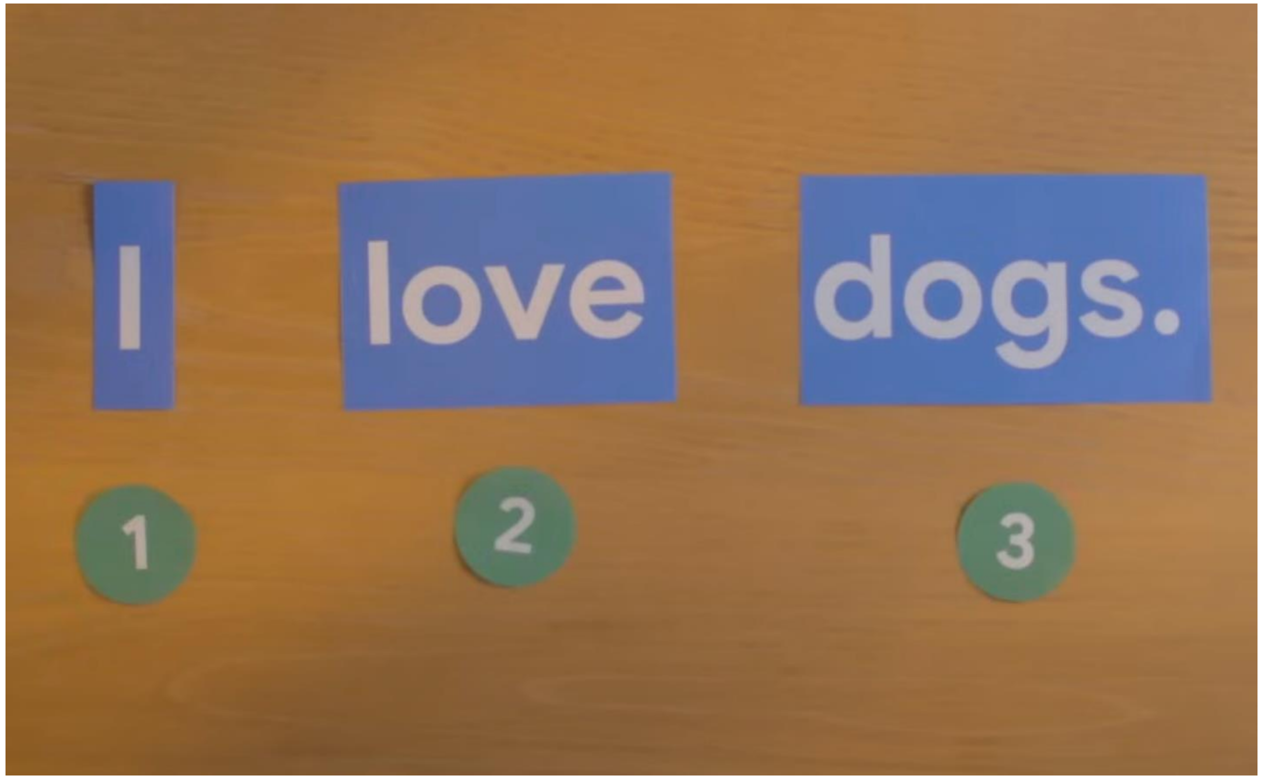

Positional Encoding

Attention

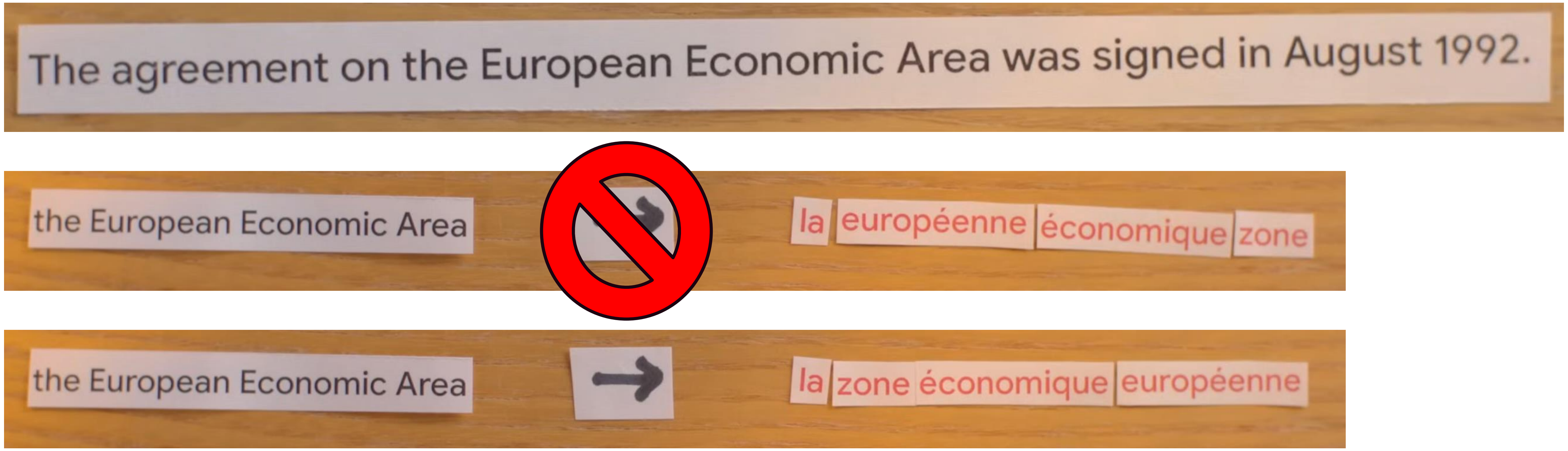
Self-Attention

- **Transformers** are nothing but 3 magical words, can you guess?

POSITIONAL ENCODINGS

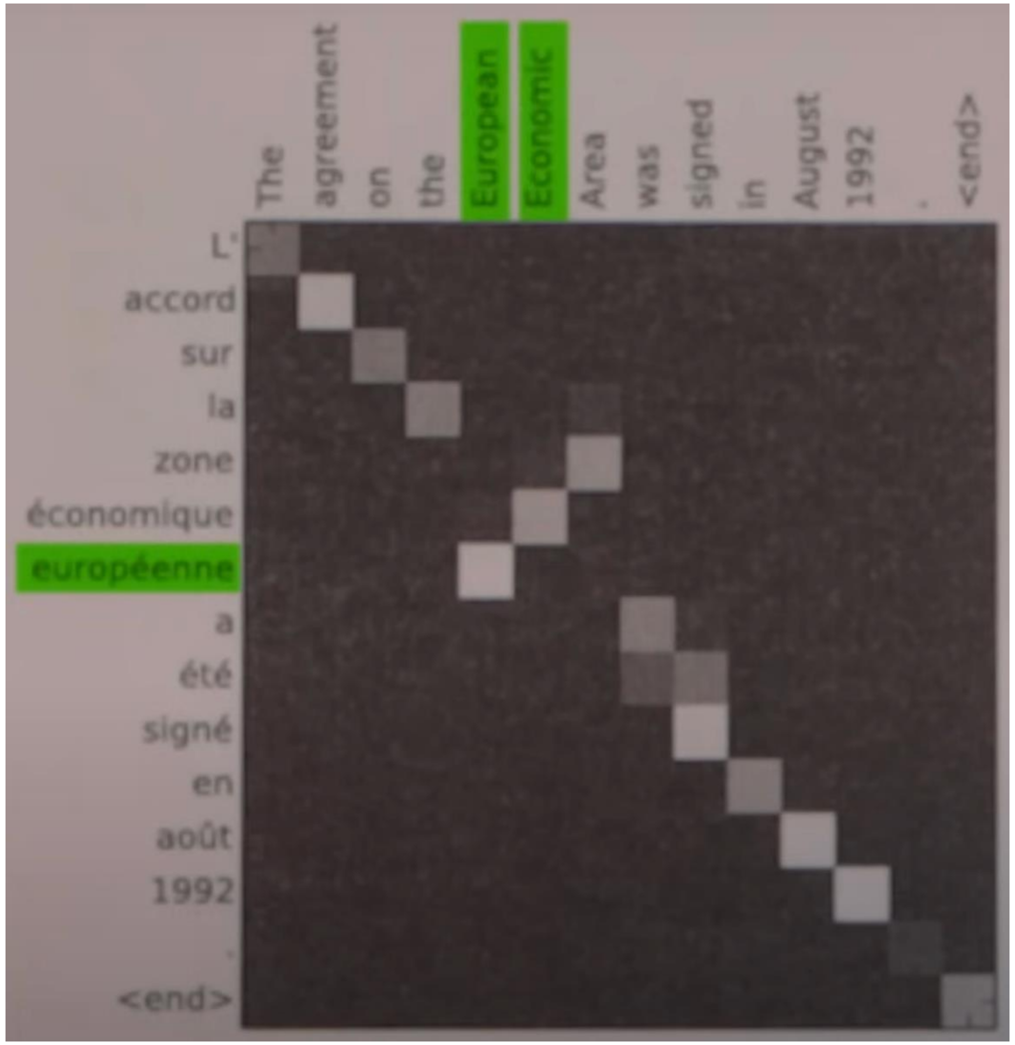


ATTENTION

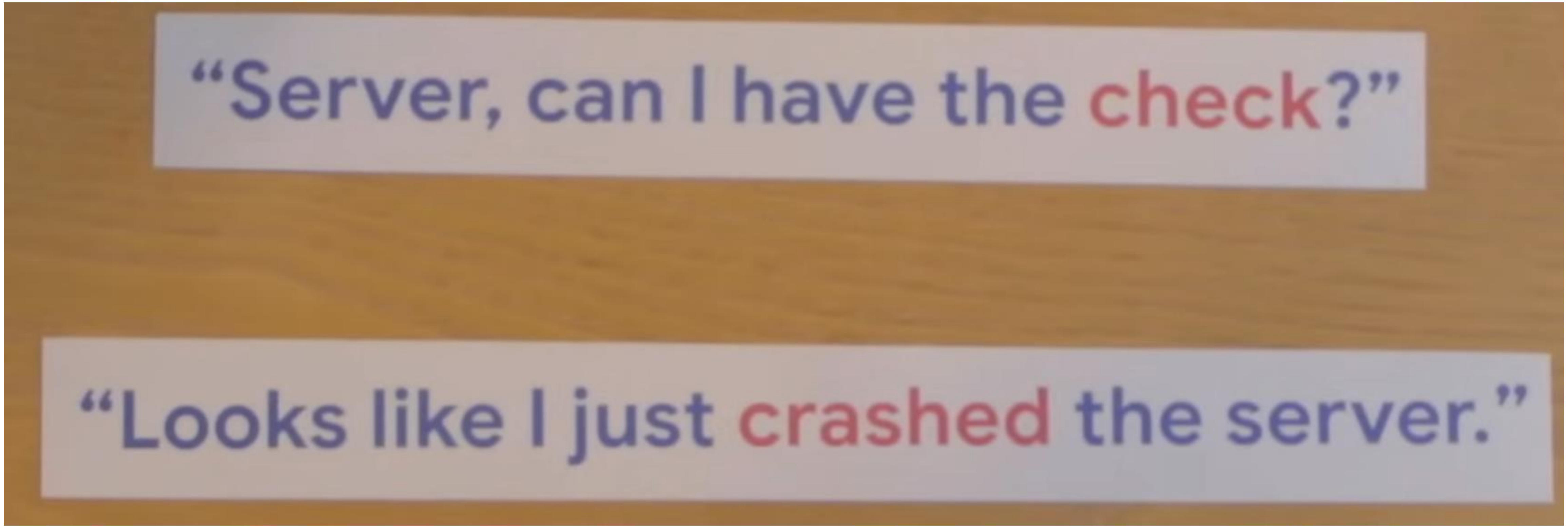

The agreement on the European Economic Area was signed in August 1992.
the European Economic Area
la européenne économique zone
the European Economic Area
la zone économique européenne

Allows the text model to look every single word from the input sentence to decide what should be the words in output sentence

ATTENTION


The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

SELF ATTENTION



"Server, can I have the check?"

"Looks like I just crashed the server."

In self-attention, every word in a sentence is compared with every other word to determine which words are most important for predicting the next word.

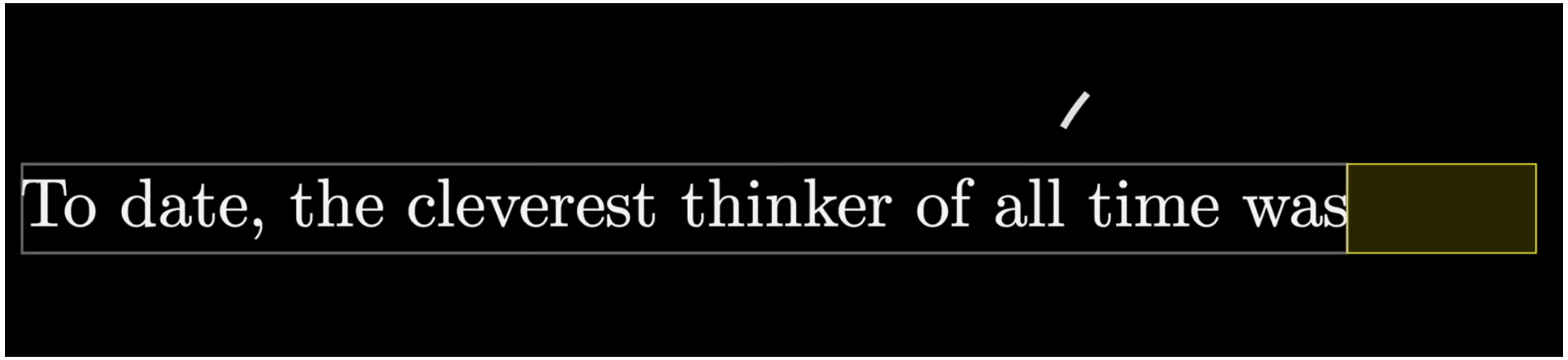
WHY TRANSFORMERS ARE POWERFUL

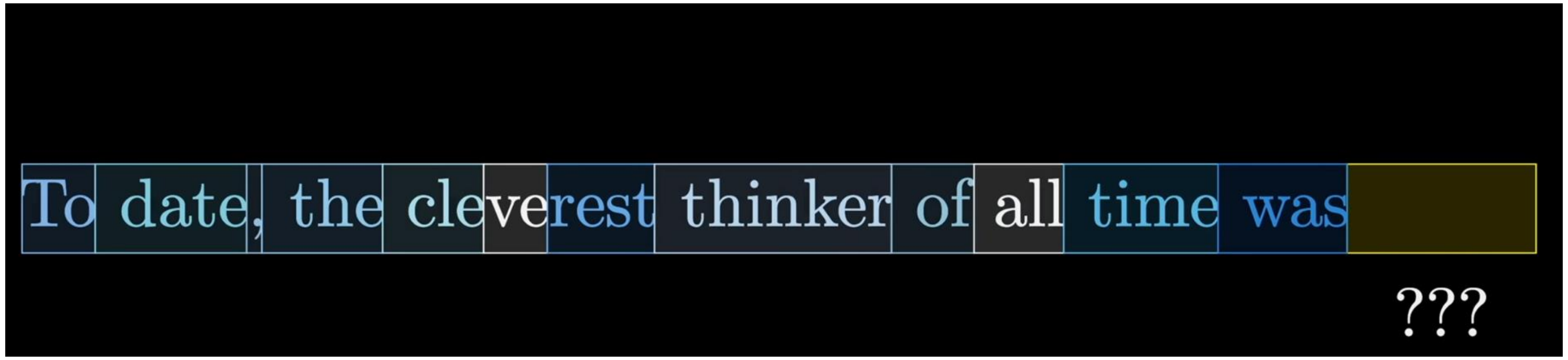
Why are transformers superior to RNNs/LSTMs?

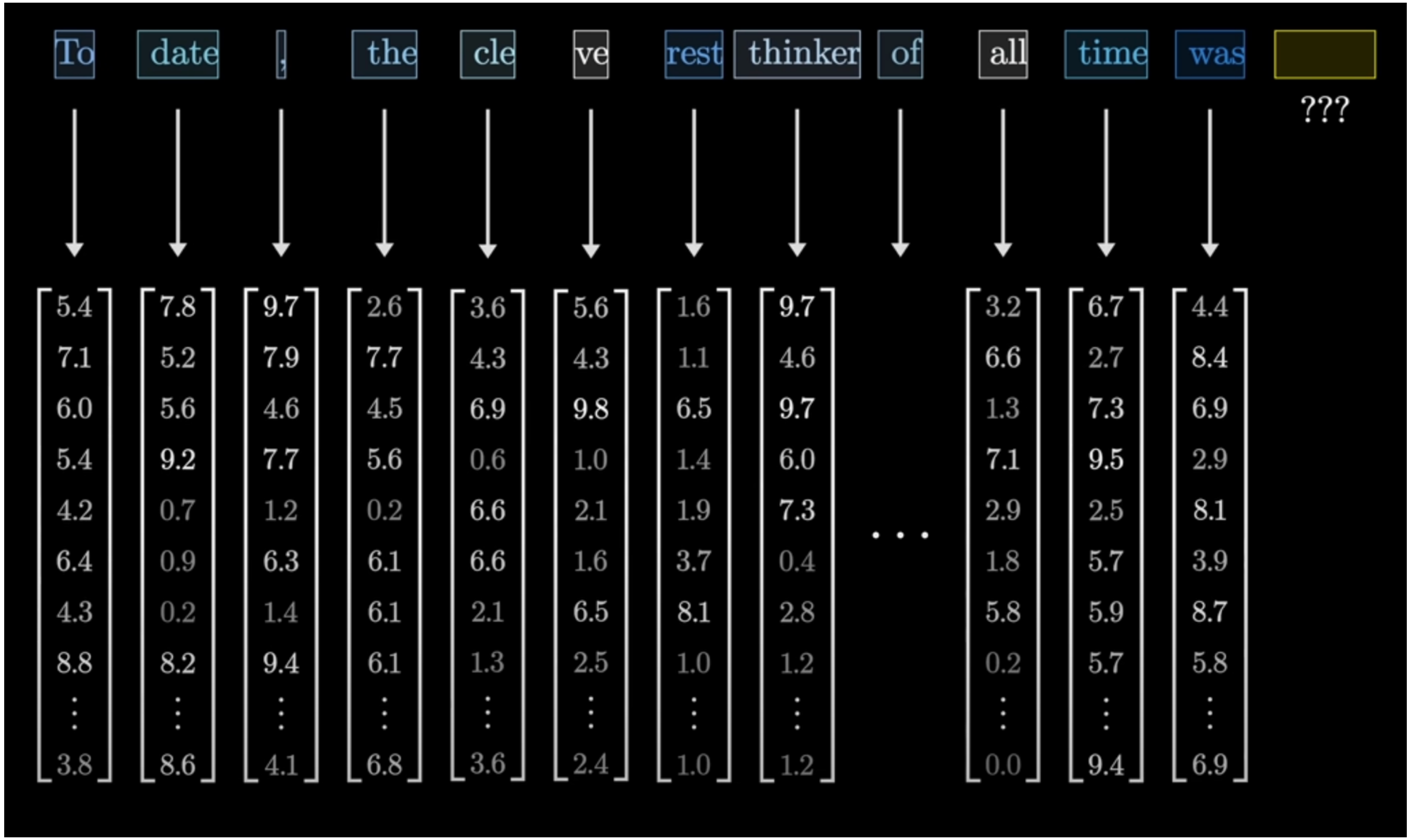
- **Parallelization**: Transformers can process the entire sequence at once, whereas RNNs process one step at a time. This makes transformers faster to train.
- **Long-range dependencies**: Transformers can easily attend to distant parts of a sequence, which is hard for RNNs and LSTMs.
- **State-of-the-art results**: Transformers consistently outperform RNNs and LSTMs on tasks like machine translation, text generation, and summarization.

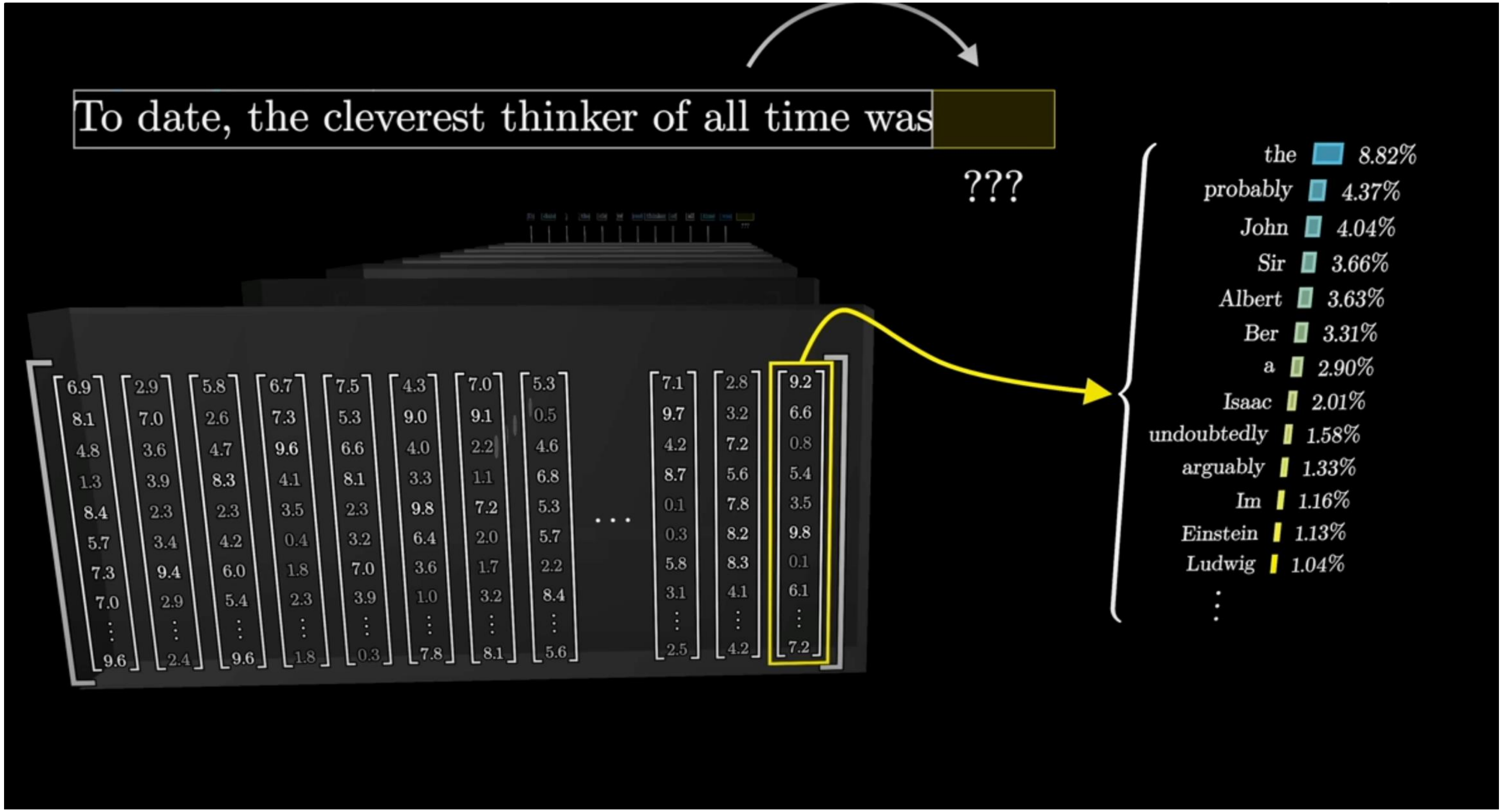
INTRODUCTION TO GPT (GENERATIVE PRETRAINED TRANSFORMER)

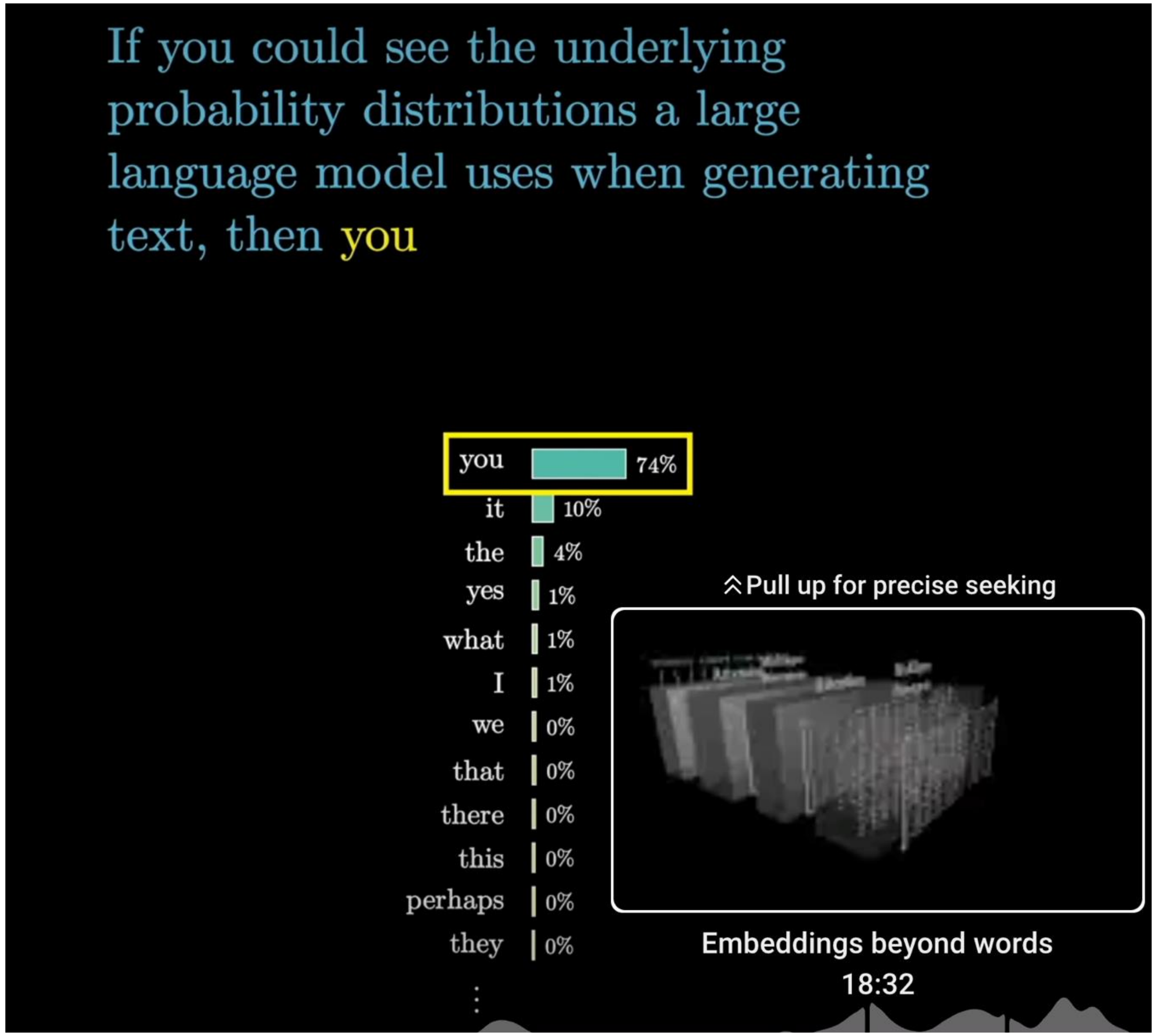
- GPT is a language model built on the transformer architecture, developed by OpenAI. It is pretrained on large corpora of text data and then fine-tuned for specific tasks.






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


To date, the cleverest thinker of all time was
???
the 8.82%
probably 4.37%
John 4.04%
Sir 3.66%
Albert 3.63%
Ber 3.31%
a 2.90%
Isaac 2.01%
undoubtedly 1.58%
arguably 1.33%
Im 1.16%
Einstein 1.13%
Ludwig 1.04%



If you could see the underlying probability distributions a large language model uses when generating text, then you

you	74%
it	10%
the	4%
yes	1%
what	1%
I	1%
we	0%
that	0%
there	0%
this	0%
perhaps	0%
they	0%