UNDERSTANDING TRANSFORMER

The Engine Behind Modern AI

AI ကို ဘယ်တုန်းကစကြားဖူးပါသလဲ

1. chatgpt ကြောင့်
2. အလန်တူးရင်း
3. အီလွန်မက်စ်
4. အခြား (ဥပမာ Simbolo)

SPEAKER

Agga Min @ Rei

Data Scientist / NLP Researcher

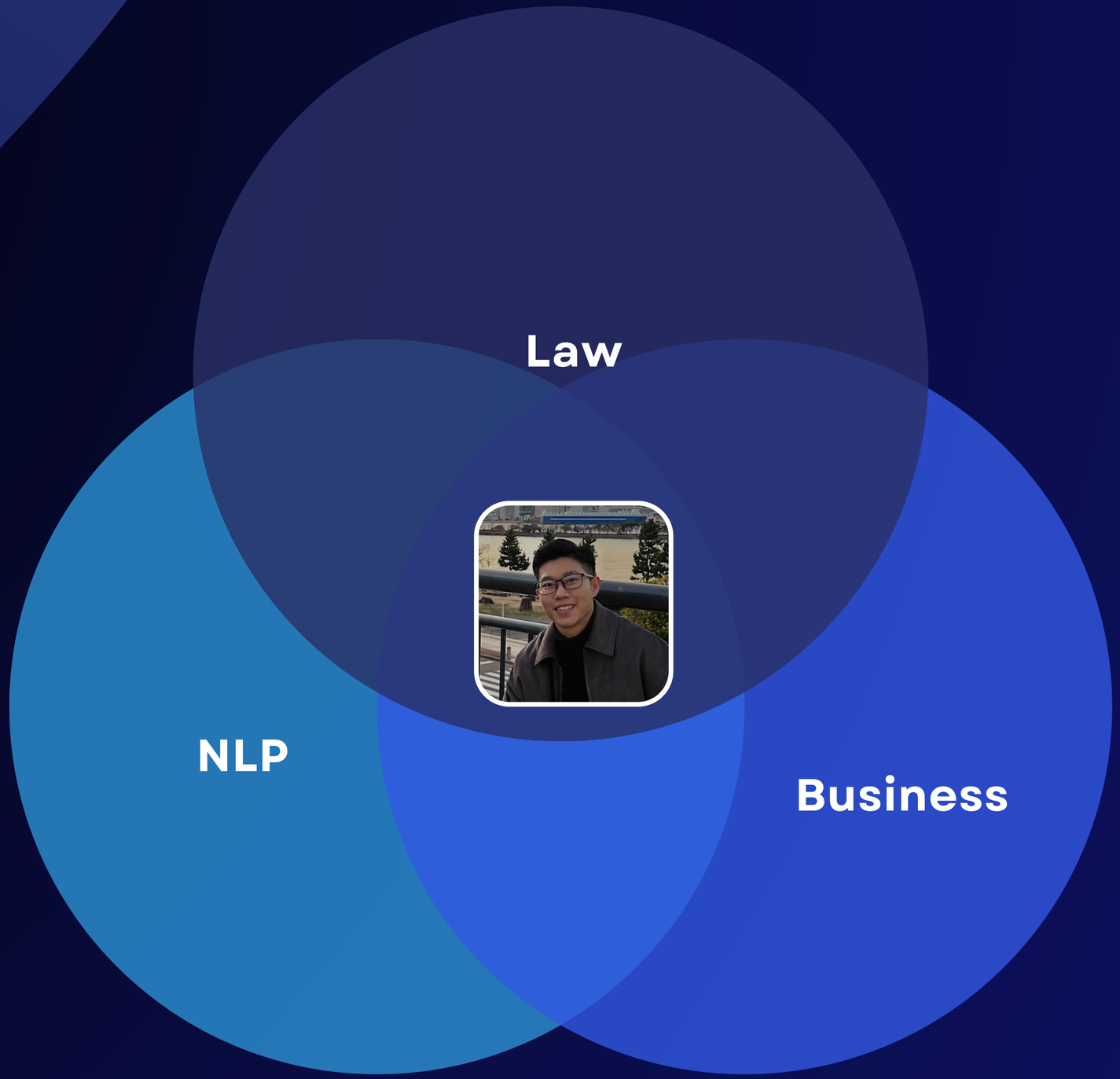
1-on-1 class instructor @ SimboloAI

MemoryLab Inc., Tokyo, Japan

Tokyo International University

Linkedin

Youtube



The Rise of Machine Learning in AI

ML has become a cornerstone of AI. Its ability to analyze vast amounts of data has fueled advancements in diverse fields, from healthcare and finance to transportation and entertainment.

1

Early AI

Rule-based systems, limited data, narrow applications.

2

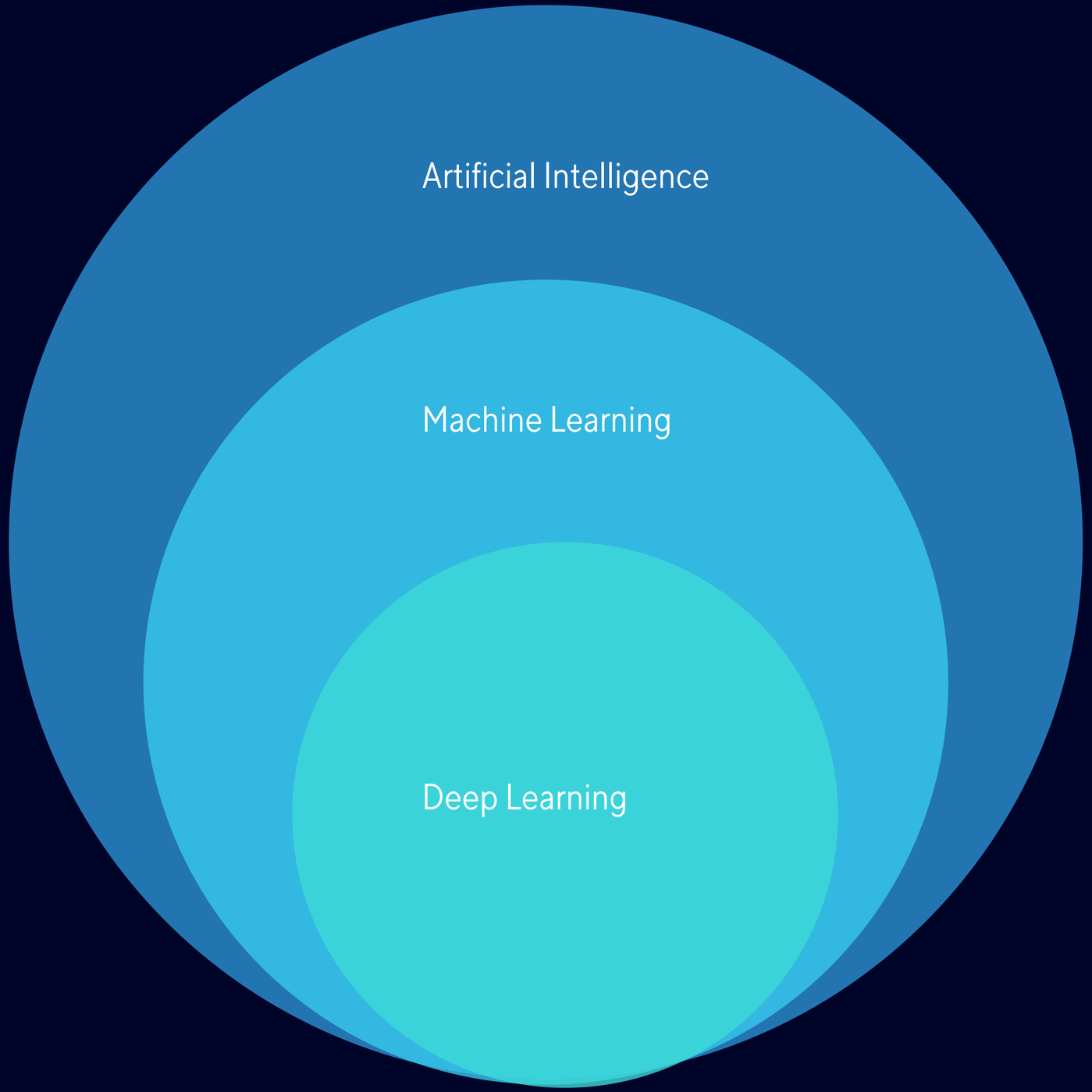
Machine Learning Era

Statistical methods, data-driven models, wider application domains.

3

Deep Learning Revolution

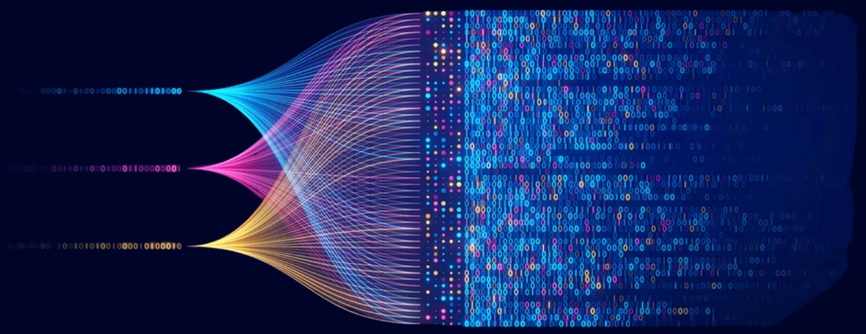
Neural networks, large datasets, complex AI systems.

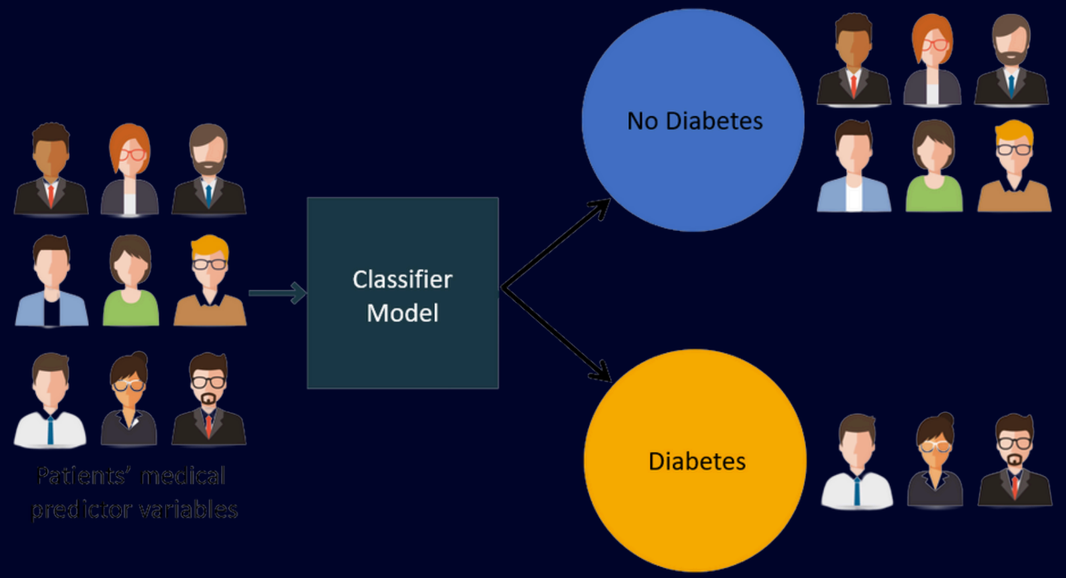


1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Supervised Learning: Predictive Modeling and Classification

Supervised learning algorithms learn from labeled data. They are trained to predict outcomes based on past patterns and can be used for classification or regression tasks.





1 Predictive Modeling

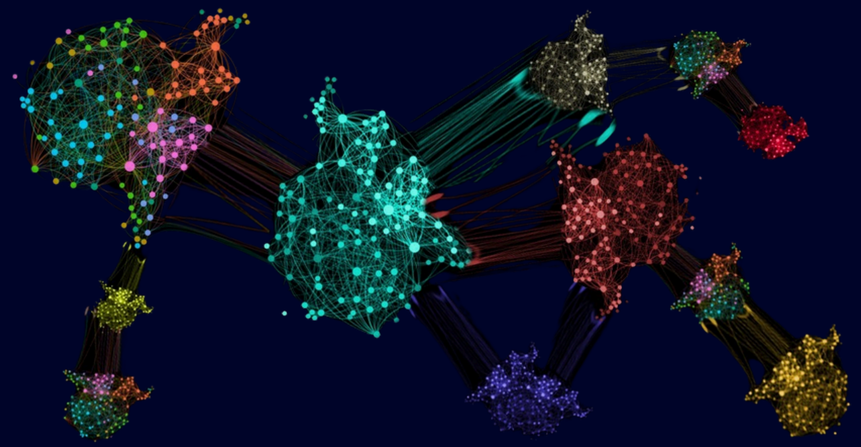
Predicting future values based on historical data.

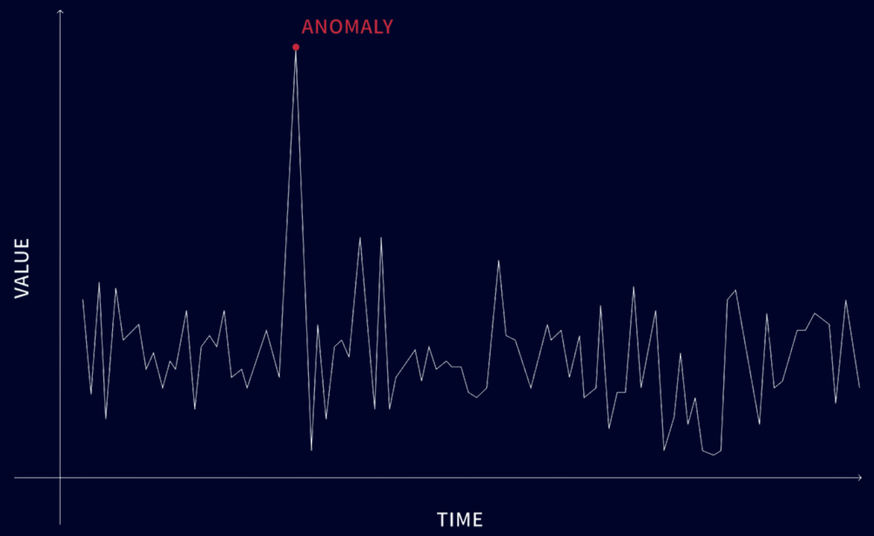
2 Classification

Categorizing data into predefined classes.

Unsupervised Learning: Clustering and Anomaly Detection

Unsupervised learning algorithms discover hidden patterns in unlabeled data. They are used to group data points into clusters or identify outliers.





1 Clustering

Grouping similar data points together. 1.K-Means Clustering, 2 Hierarchical Clustering

2 Anomaly Detection

Identifying unusual data points that deviate from the norm. 1.One-Class SVM, 2.Isolation Forest

Reinforcement Learning: Autonomous Decision-Making

Reinforcement learning enables agents to learn through trial and error. They interact with an environment, receiving rewards for desirable actions and penalties for undesirable ones.

1 **Agent**

The entity that interacts with the environment.

2 **Environment**

The context in which the agent operates.

3 **Actions**

The agent's choices within the environment.

4 **Rewards**

Feedback received by the agent for its actions.

Deep Learning: Neural Networks and Advanced AI Architectures

Deep learning employs artificial neural networks with multiple layers. These networks can learn complex patterns and represent high- dimensional data, enabling powerful AI capabilities.

1 **Convolutional Neural Networks (CNNs)**

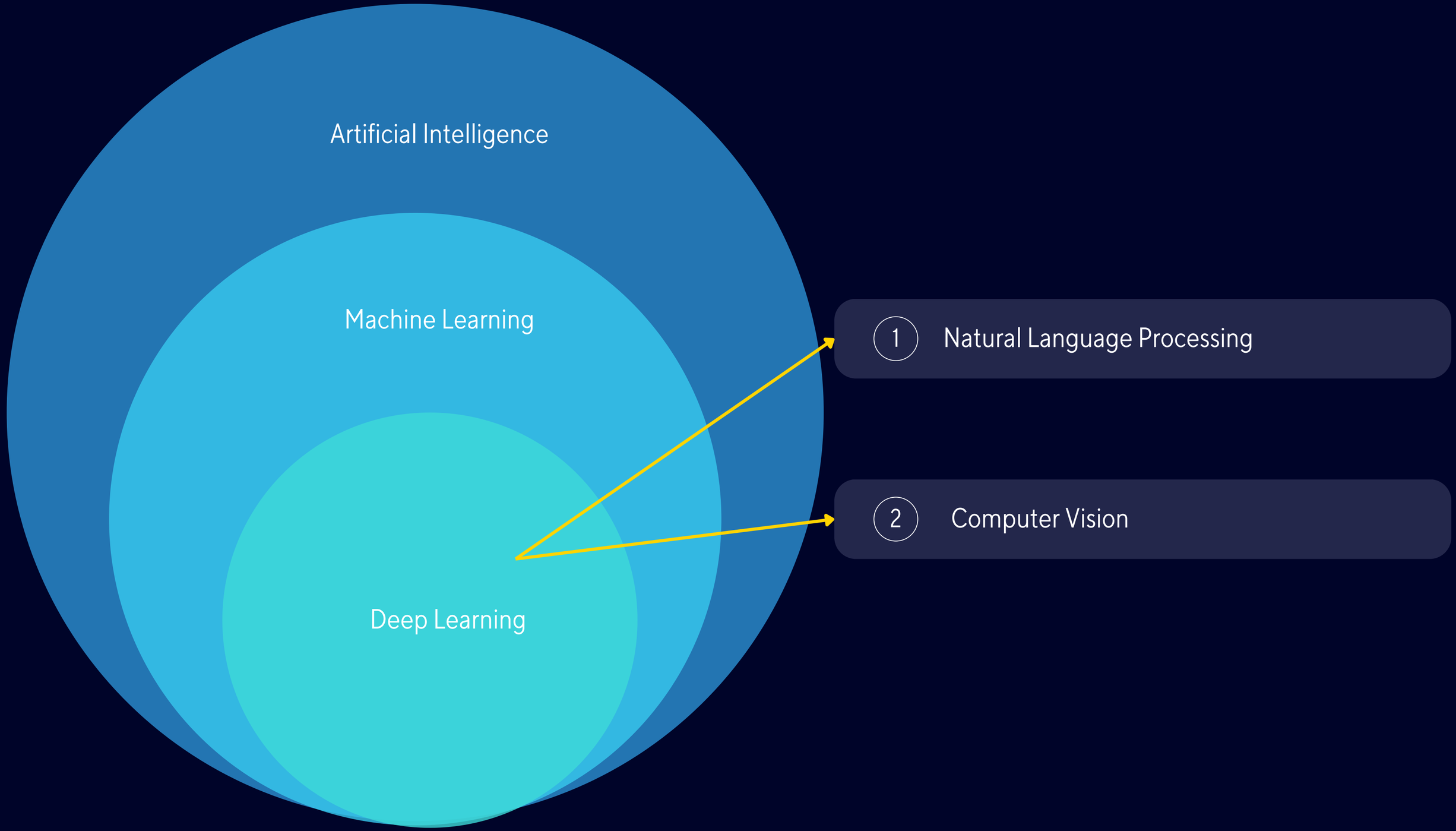
Process image data, ideal for computer vision tasks.

2 **Recurrent Neural Networks (RNNs)**

Handle sequential data, suitable for natural language processing.

3 **Generative Adversarial Networks (GANs)**

Generate realistic data, used for image synthesis and text generation.


Artificial Intelligence
Machine Learning
Deep Learning
1
Natural Language Processing
2
Computer Vision

Machine Learning for Natural Language Processing

Natural language processing (NLP) uses ML to understand and process human language. It allows computers to extract meaning from text, translate languages, and generate human-like text.

	TASK	EXAMPLE
1	Text Classification	Sentiment analysis, spam detection
2	Machine Translation	Translating between languages.
3	Text Summarization	Generating concise summaries of long texts.

Computer Vision and Image Recognition with Machine Learning

Computer vision utilizes ML to interpret and understand images. It allows computers to identify objects, recognize faces, and analyze scenes, enabling applications like autonomous vehicles and medical image analysis.

1. **Object Detection**

   Identifying objects in images or videos.

2. **Image Segmentation**

   Dividing images into meaningful regions.

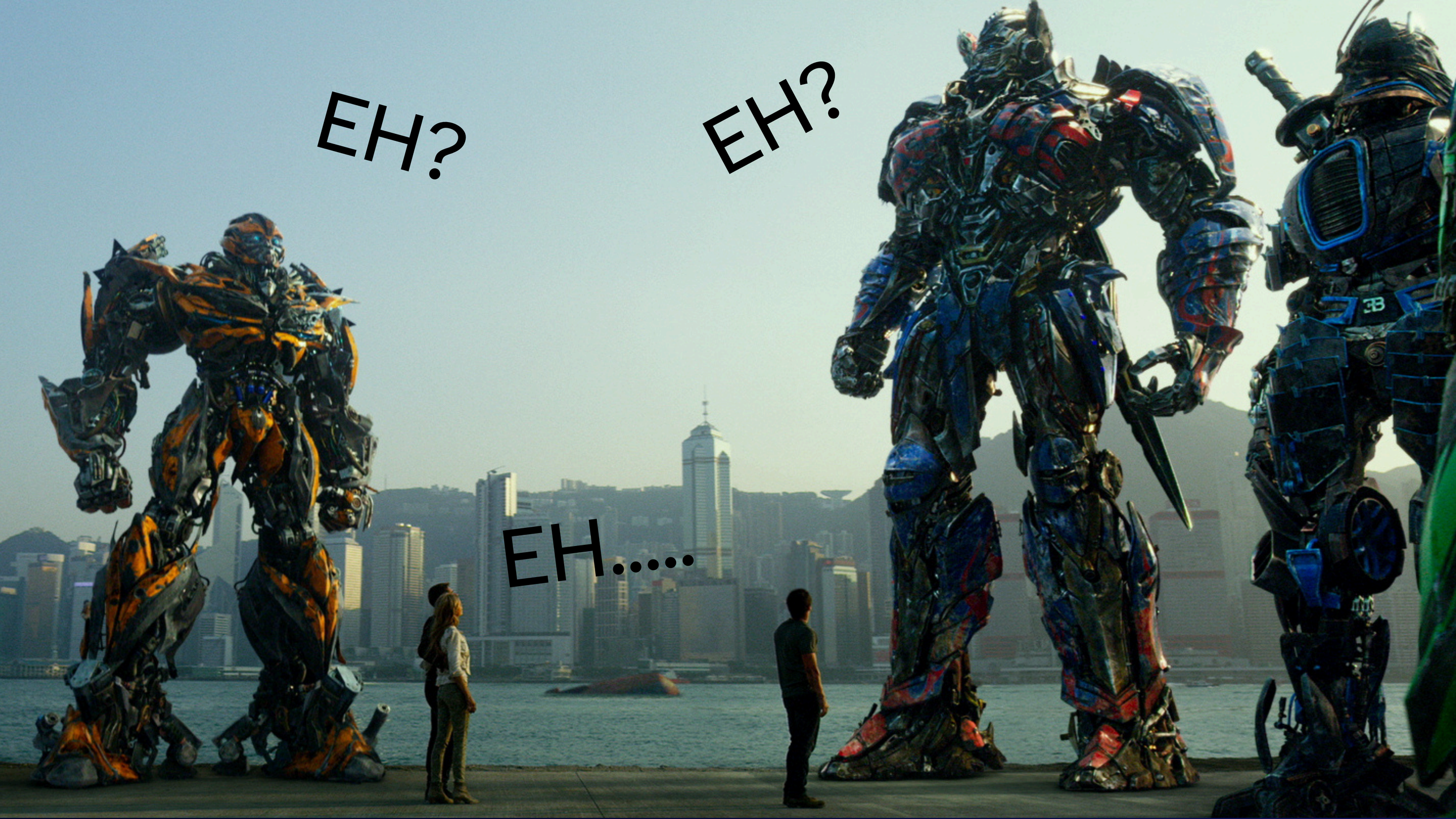
3. **Facial Recognition**

   Recognizing and identifying individuals based on their faces.

4. **Optical Character Recognition (OCR)**

   Extracting text from images.

TRANSFORMER


EH?
EH?
EH.....

What is Transformer?

A deep learning model introduced in the paper "Attention is All You Need" by *Vaswani et al.* in 2017.

WHAT IS DIFFERENT

1. **Self-Attention Mechanism** — to capture long-range dependencies in a single step
2. **Parallel Processing** — efficient utilization of computational resources.
3. **Multi-Head Attention** — capture various types of relationships and nuances

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

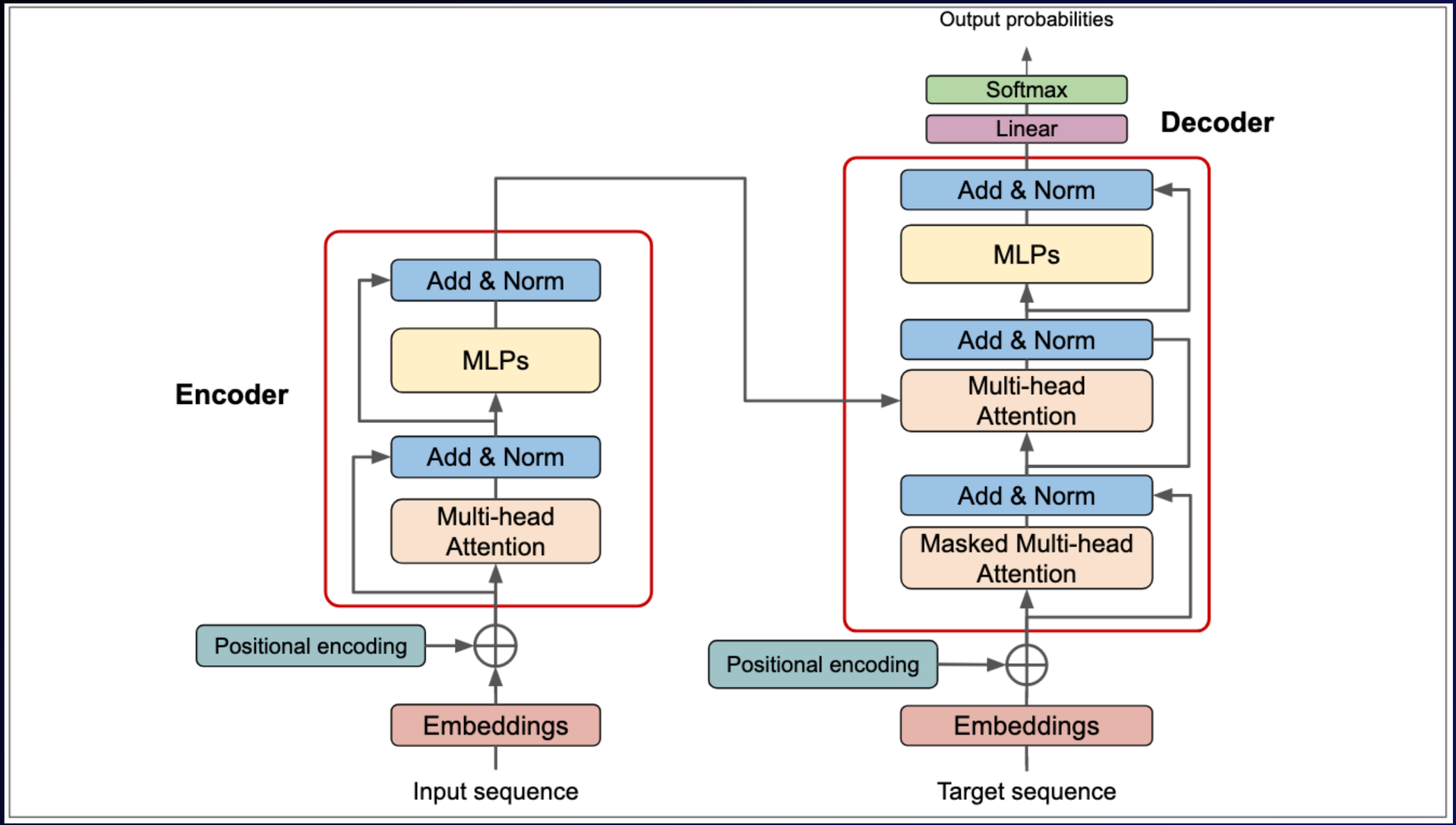
*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

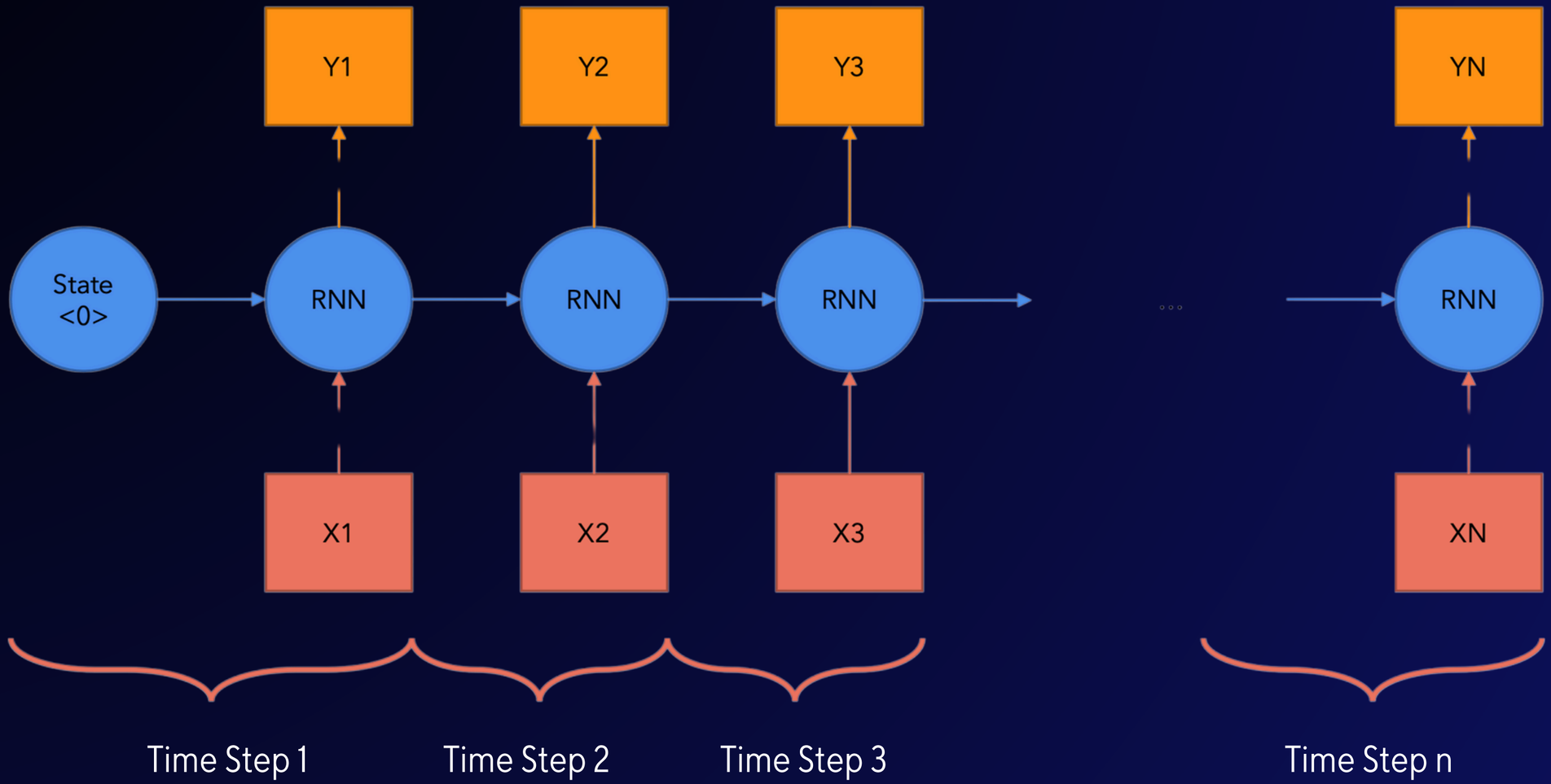
‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Transformer Architecture



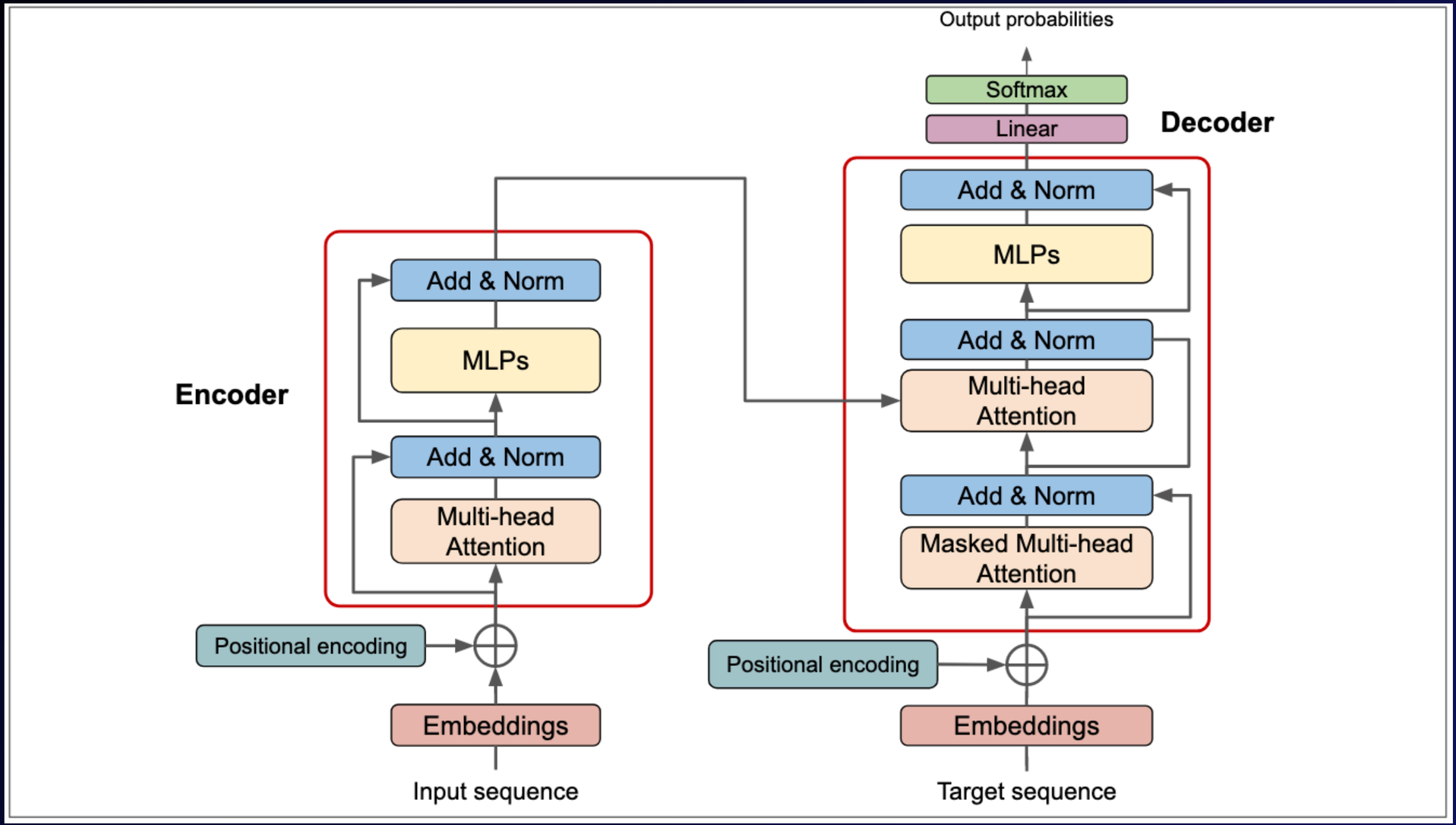
Recurrent Neural Network



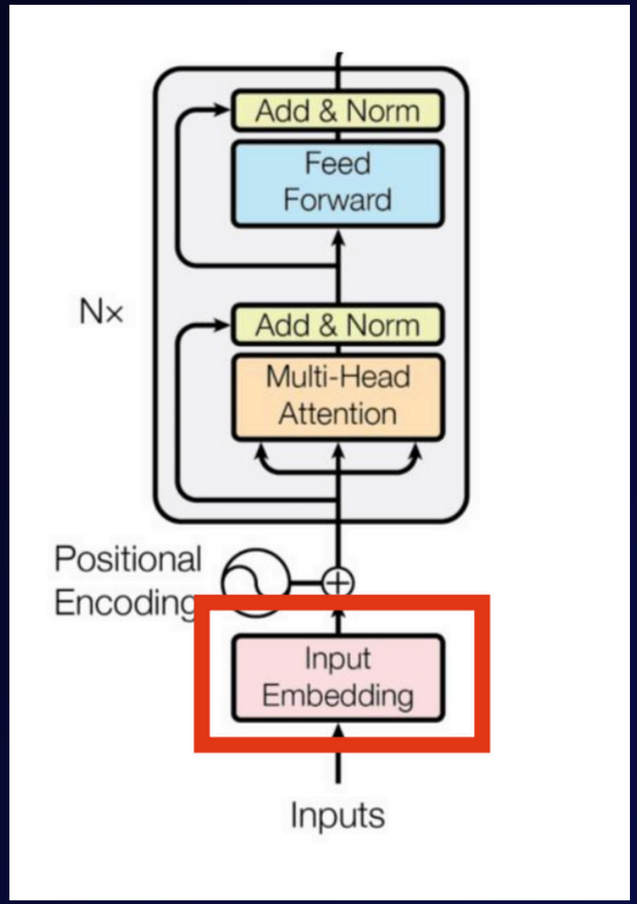
Problems with RNN

1. Slow computation for long sequence
2. Vanishing gradients
3. Difficulty in accessing the long time ago information

Transformer Architecture

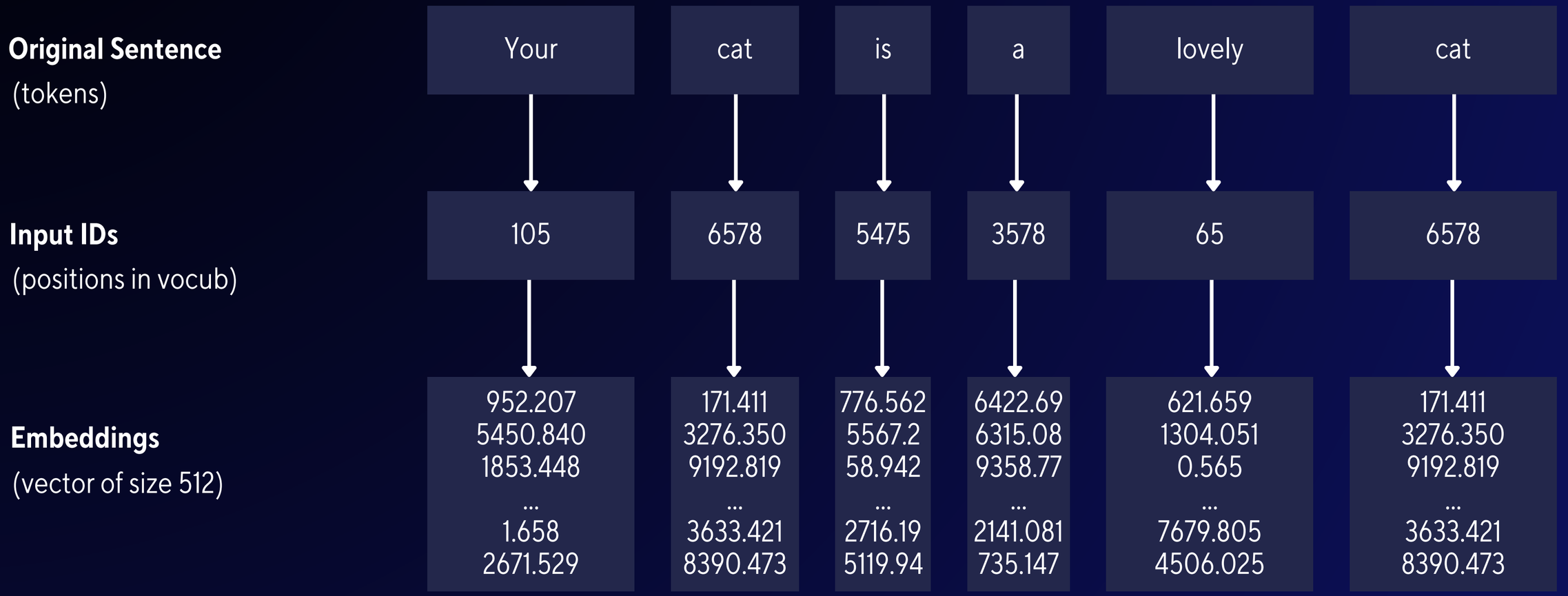


Embeddings

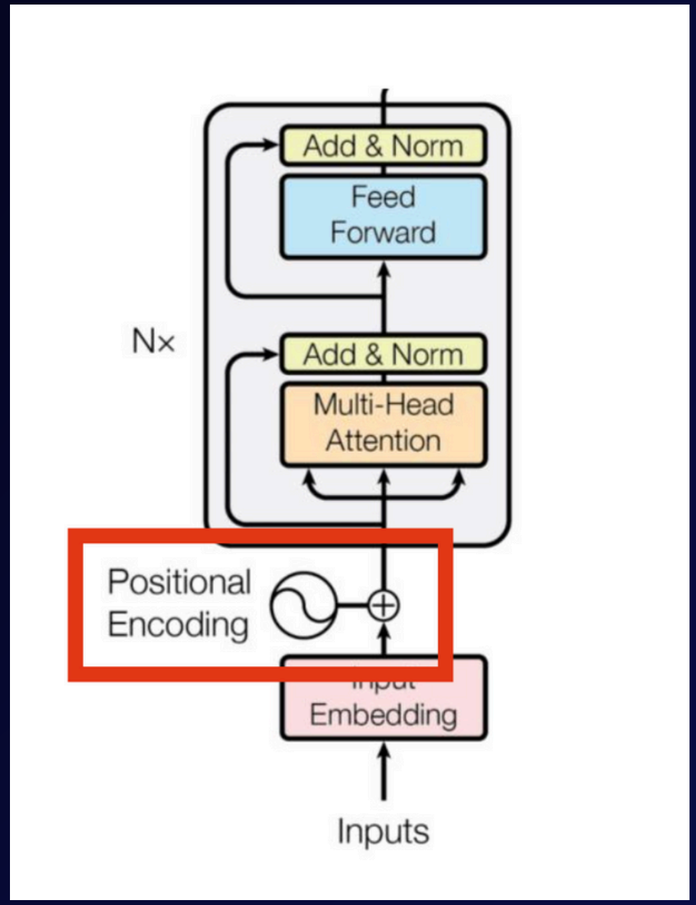

Add & Norm
Feed Forward
Nx
Add & Norm
Multi-Head Attention
Positional Encoding
Input Embedding
Inputs

What is Embedding?

Embeddings convert words into dense vectors of fixed size. They capture semantic relationships between words.

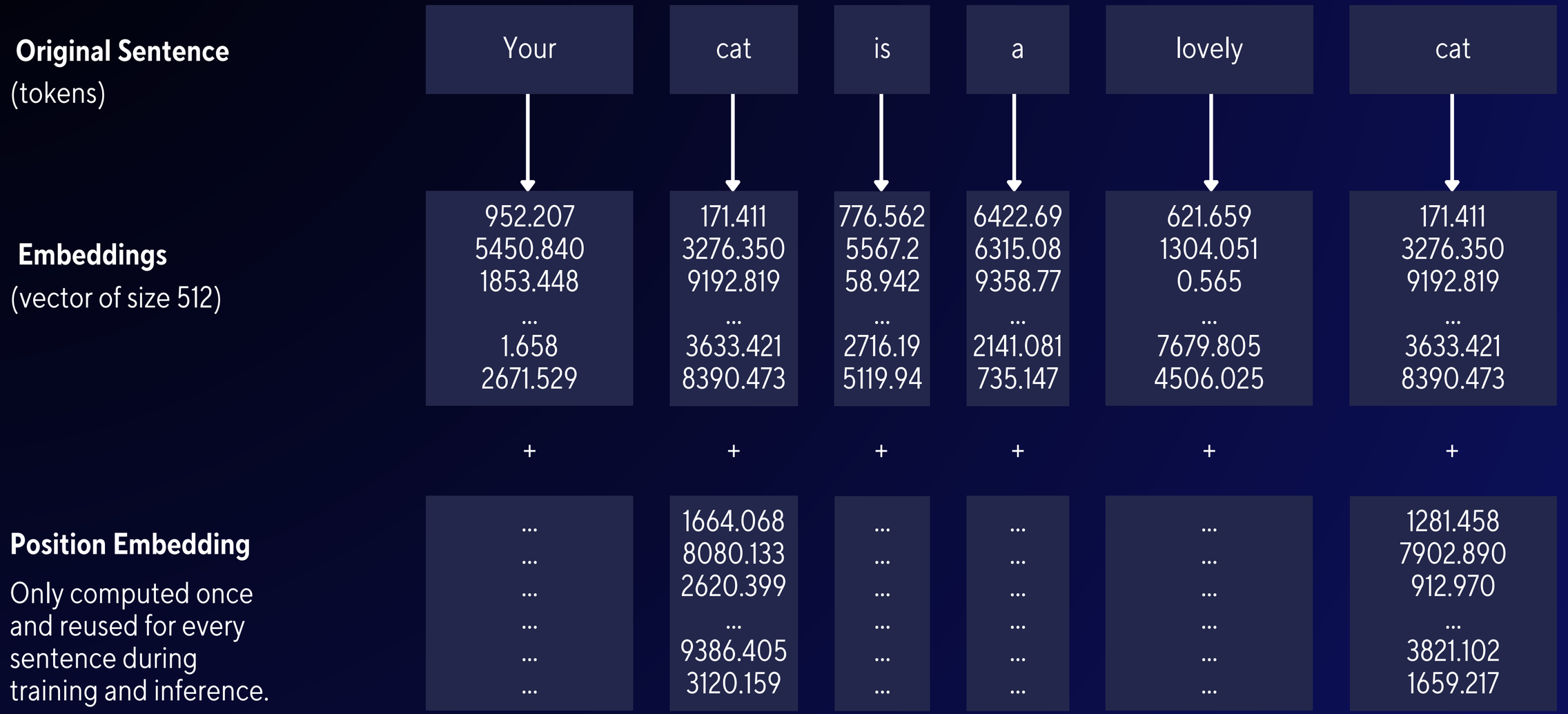


Positional Encoding



Positional Encoding

Since Transformers do not process data sequentially, positional encoding adds information about the position of each word in the sequence.

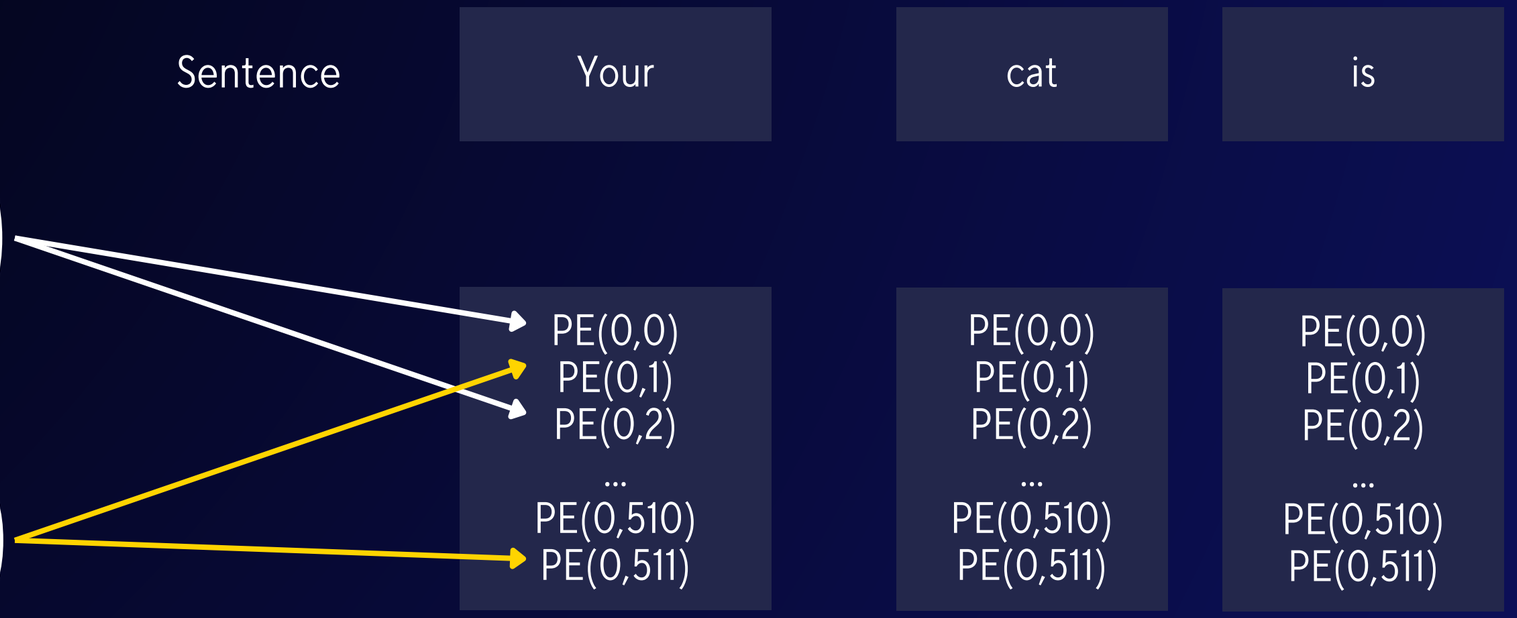


Positional Encoding

Since Transformers do not process data sequentially, positional encoding adds information about the position of each word in the sequence.

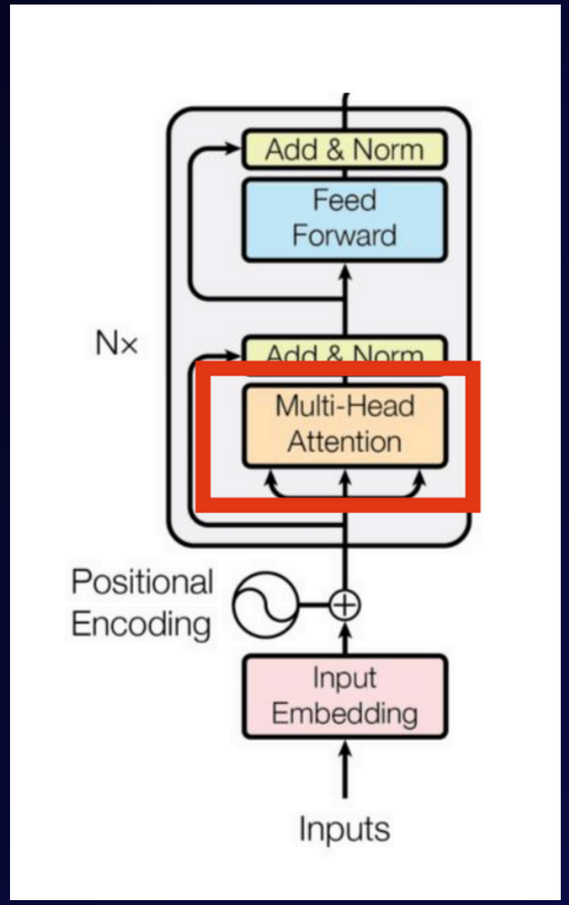
$$PE\left(POS, 2i\right) = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE\left(POS, 2i+1\right) = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$



We only need to compute the positional encodings once andthen reuse them for every sentence, no matter if it is training or inference.

MultiHead Attension



Self Attension

Self-attention allows the model to weigh the importance of different words in a sequence relative to each other. Multi-head attention runs several attention mechanisms in parallel.

$$Attention\left(Q,K,V\right)=softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V$$

$$\text{softmax}\left\{\frac{Q\ (6,\ 512)\ \times\ K^T\ (512,6)}{\sqrt{512}}\right\}\ =$$

	Your	Cat	is	a	lovely	cat
Your	0.268	0.210	0.132	0.124	0.157	0.132
Cat	0.124	0.278	0.210	0.210	0.132	0.157
is	0.147	0.132	0.262	0.146	0.210	0.132
a	0.210	0.195	0.132	0.212	0.227	0.132
lovely	0.146	0.132	0.210	0.157	0.227	0.157
cat	0.195	0.210	0.146	0.157	0.146	0.157

(6,6)

Self Attension

Self-attention allows the model to weigh the importance of different words in a sequence relative to each other. Multi-head attention runs several attention mechanisms in parallel.

	Your	Cat	is	a	lovely	cat
Your	0.268	0.210	0.132	0.124	0.157	0.132
Cat	0.124	0.278	0.210	0.210	0.132	0.157
is	0.147	0.132	0.262	0.146	0.210	0.132
a	0.210	0.195	0.132	0.212	0.227	0.132
lovely	0.146	0.132	0.210	0.157	0.227	0.157
cat	0.195	0.210	0.146	0.157	0.146	0.157

(6,6)

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V$$

$\times$ V (6,512) = Attention (6,512)

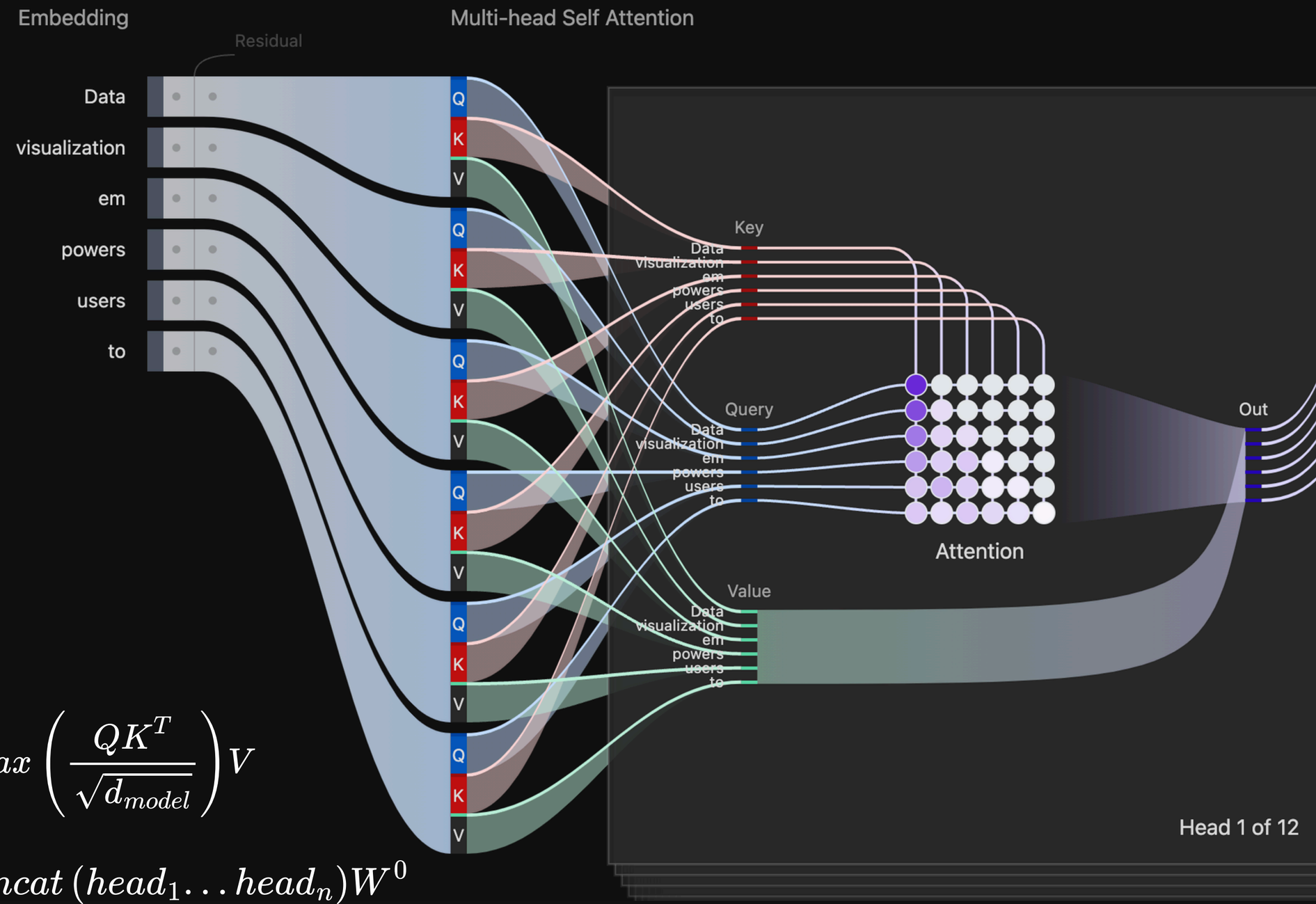
Multihead Attention

Self-attention allows the model to weigh the importance of different words in a sequence relative to each other. **Multi-head attention runs several attention mechanisms in parallel.**

$$Attention\left(Q,K,V\right)=softmax\left(\frac{QK^{T}}{\sqrt{d_{model}}}\right)V$$

$$Multihead\left(Q,K,V\right)=Concat\left(head_{1}...head_{n}\right)W^{0}$$

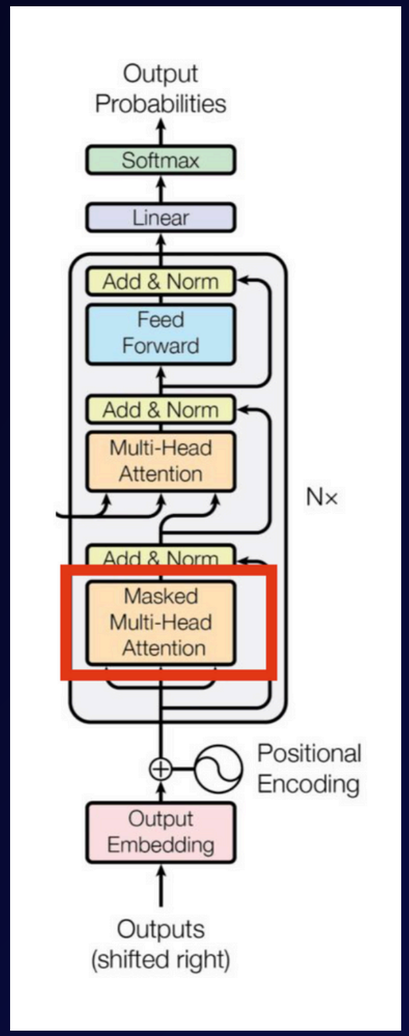
$$head_{i}=Attention\left(QW_{i}^{Q},KW_{i}^{K},VW_{i}^{V}\right)$$



$$Attention\,(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V$$

$$Multihead\,(Q,K,V) = Concat\,(head_1 \ldots head_n)W^0$$

Masked MultiHead Attention



Masked MultiHead Attention (Decoder)

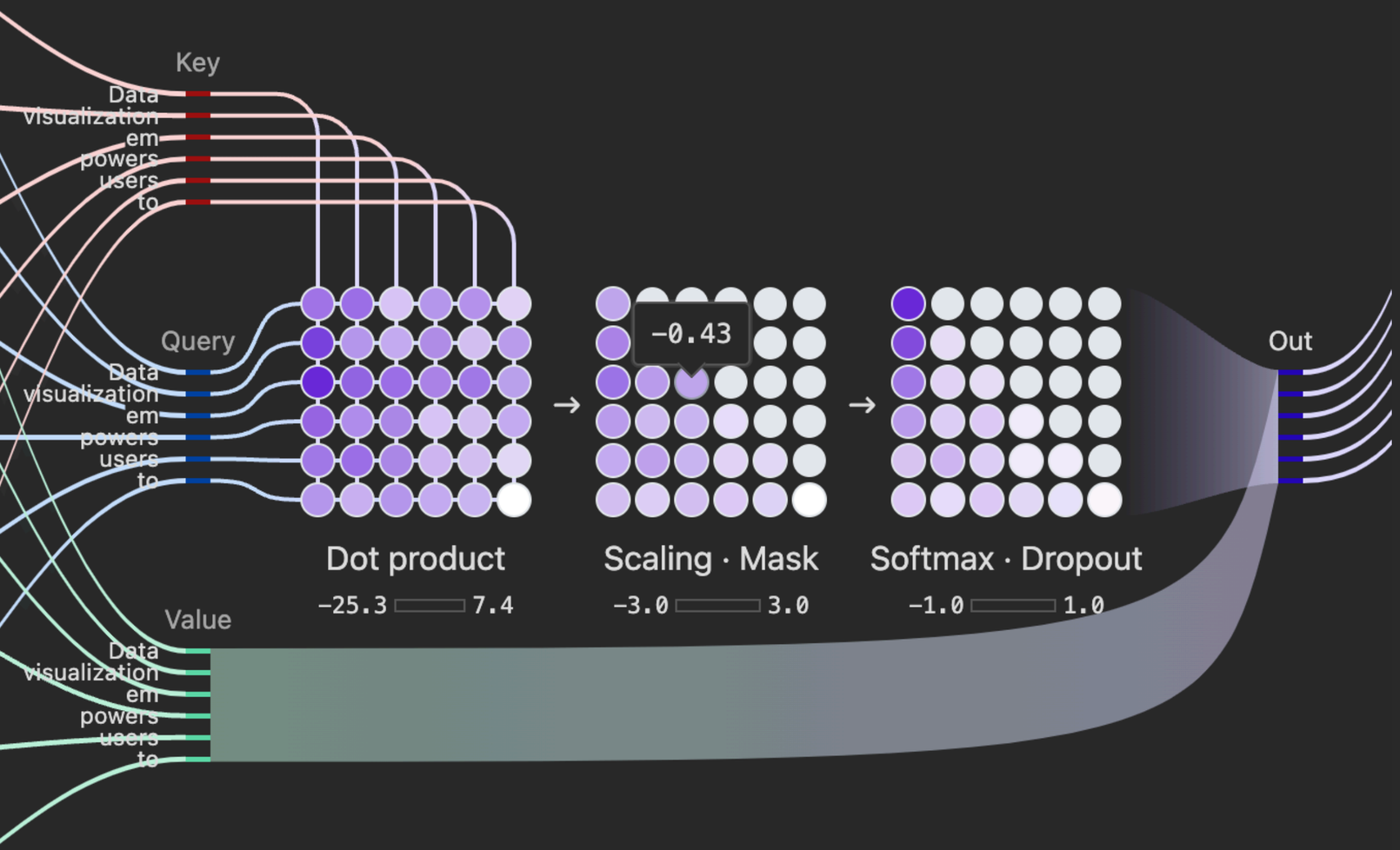
In tasks like language modeling or machine translation, when generating text, the future words should not be seen. This is where masked multi-head attention comes in, *primarily used in the decoder.*

	Your	Cat	is	a	lovely	cat
Your	0.268	~~0.210~~	~~0.132~~	~~0.124~~	~~0.157~~	~~0.132~~
Cat	0.124	0.278	~~0.210~~	~~0.210~~	~~0.132~~	~~0.157~~
is	0.147	0.132	0.262	~~0.146~~	~~0.210~~	~~0.132~~
a	0.210	0.195	0.132	0.212	~~0.227~~	~~0.132~~
lovely	0.146	0.132	0.210	0.157	0.227	~~0.157~~
cat	0.195	0.210	0.146	0.157	0.146	0.157

(6,6)

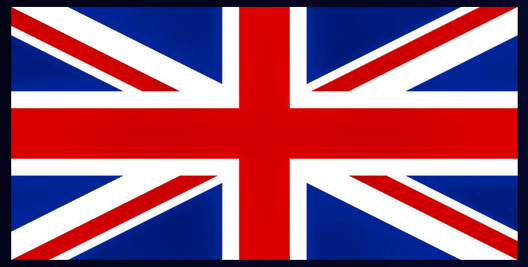
Masked MultiHead Attention

In tasks like language modeling or machine translation, when generating text, the future words should not be seen. This is where masked multi-head attention comes in, *primarily used in the decoder.*



All diagonal values are replace with -∞ before applying the softmax, which will replace them with zero.

TRAINING & INFERENCING

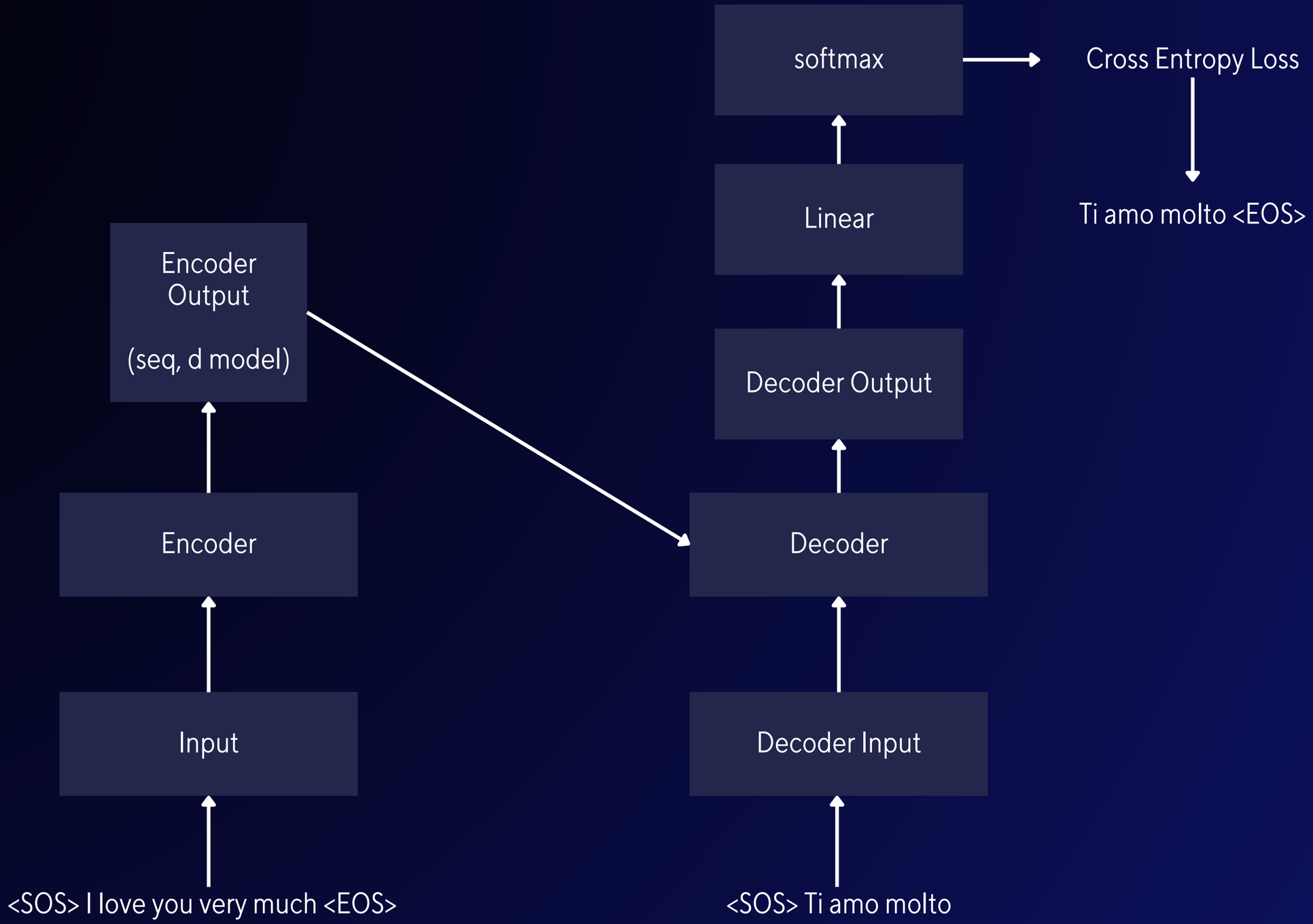


I love you very much

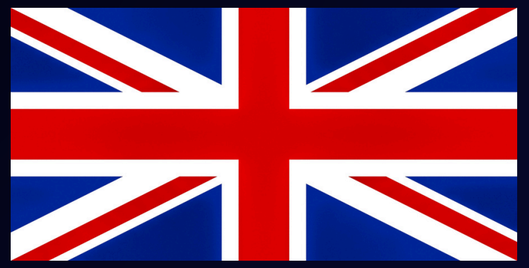
Ti amo molto

Training

Time Step = 1



Inferencing

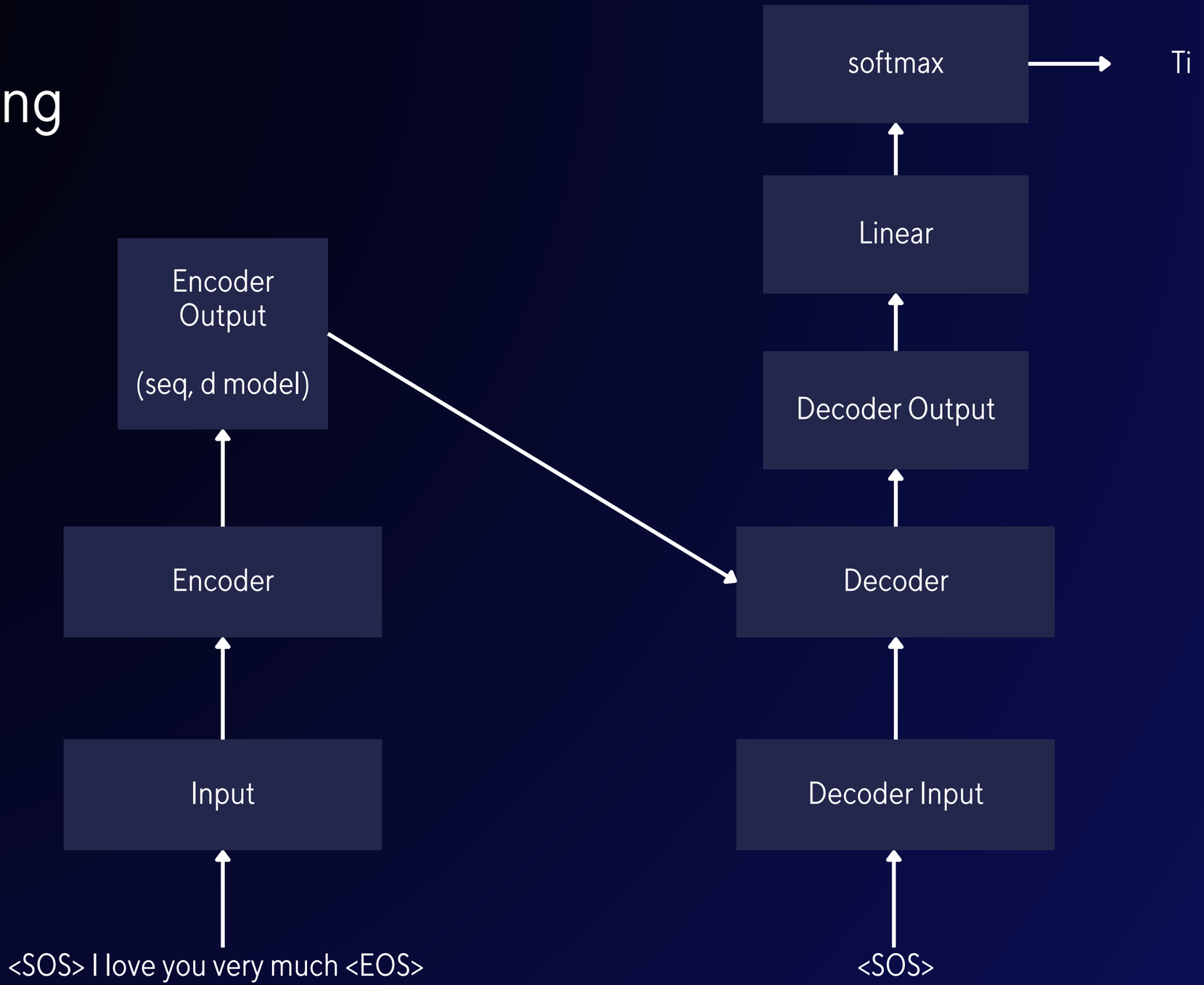


I love you very much

Ti amo molto

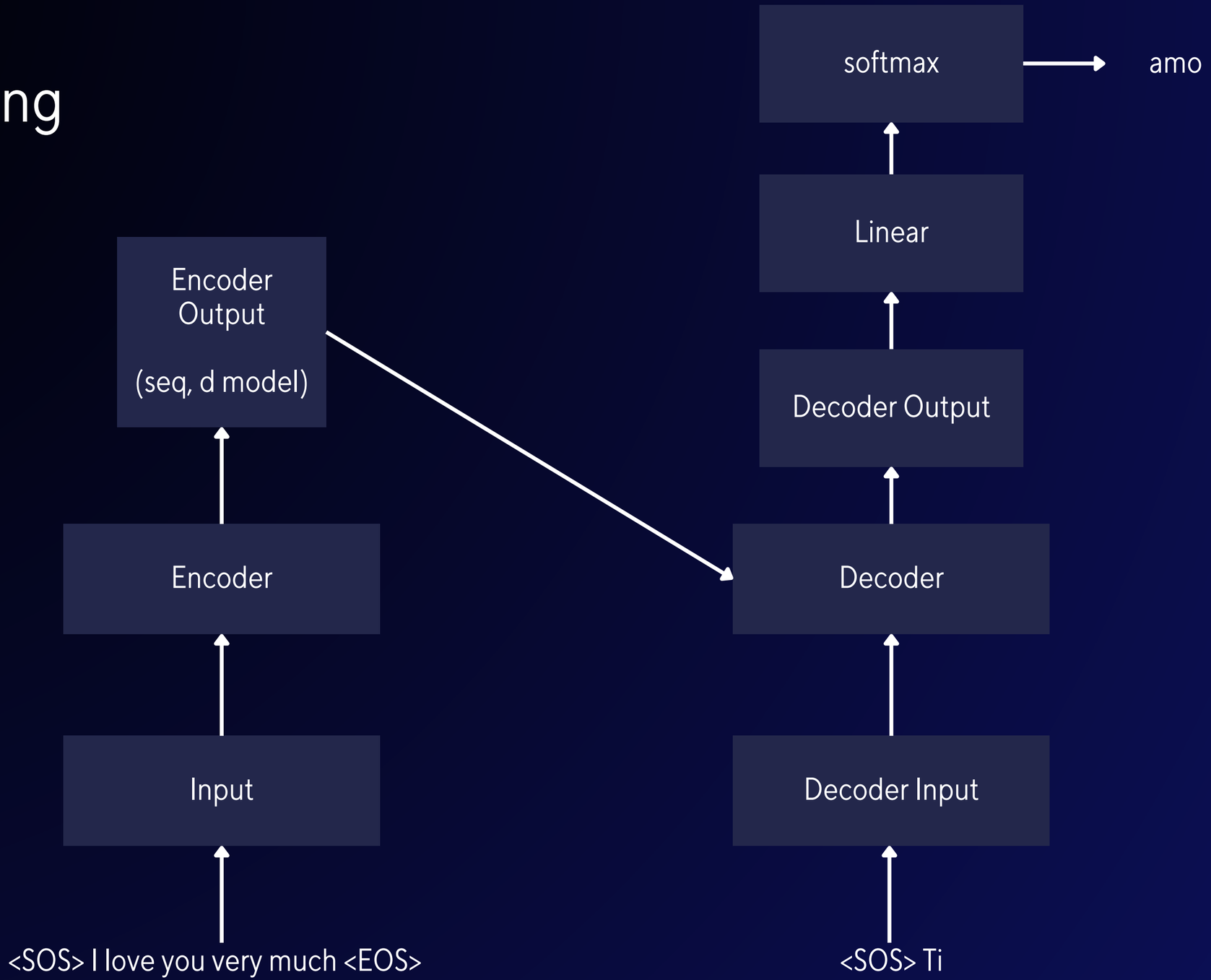
Inferencing

Time Step = 1



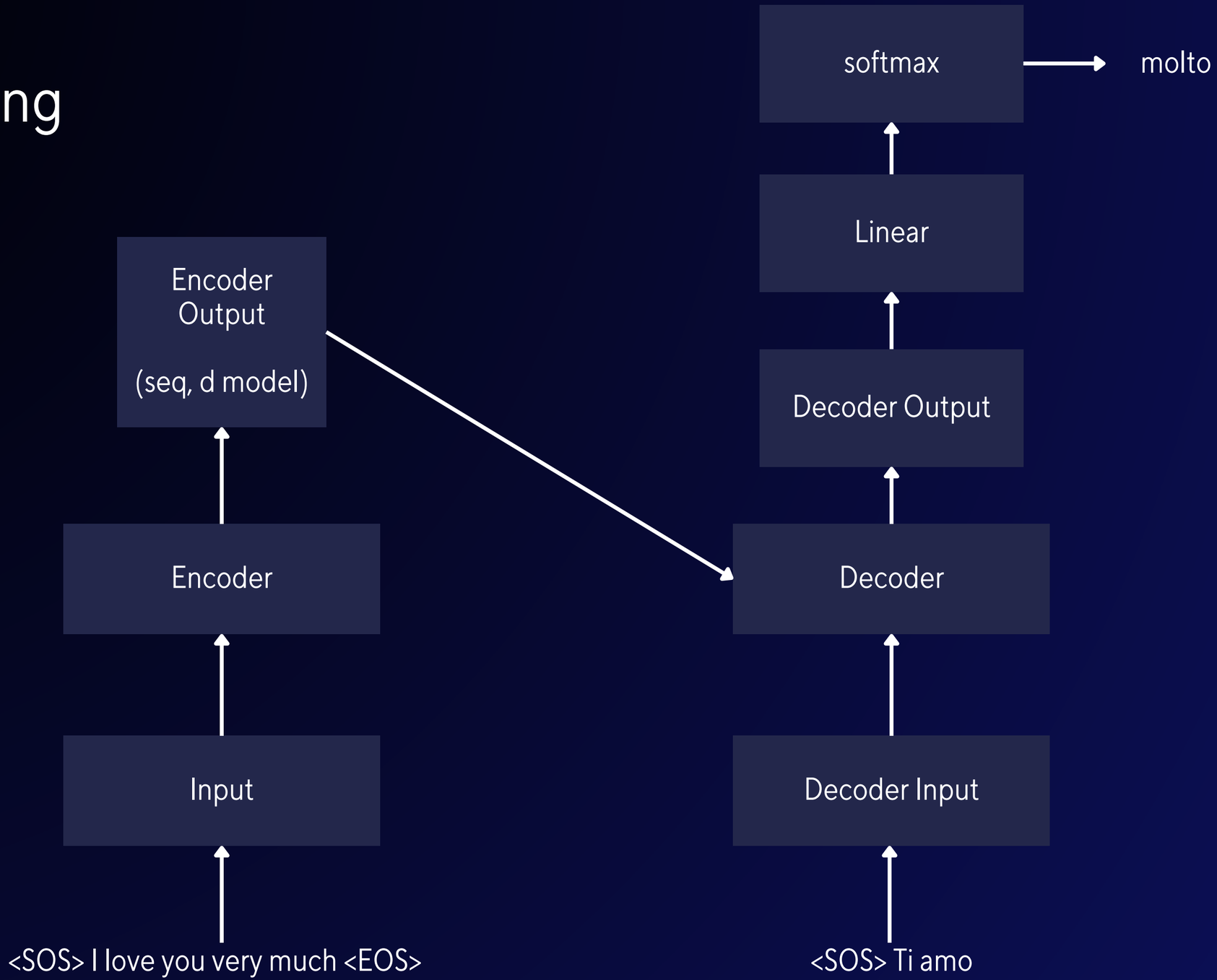
Inferencing

Time Step = 2



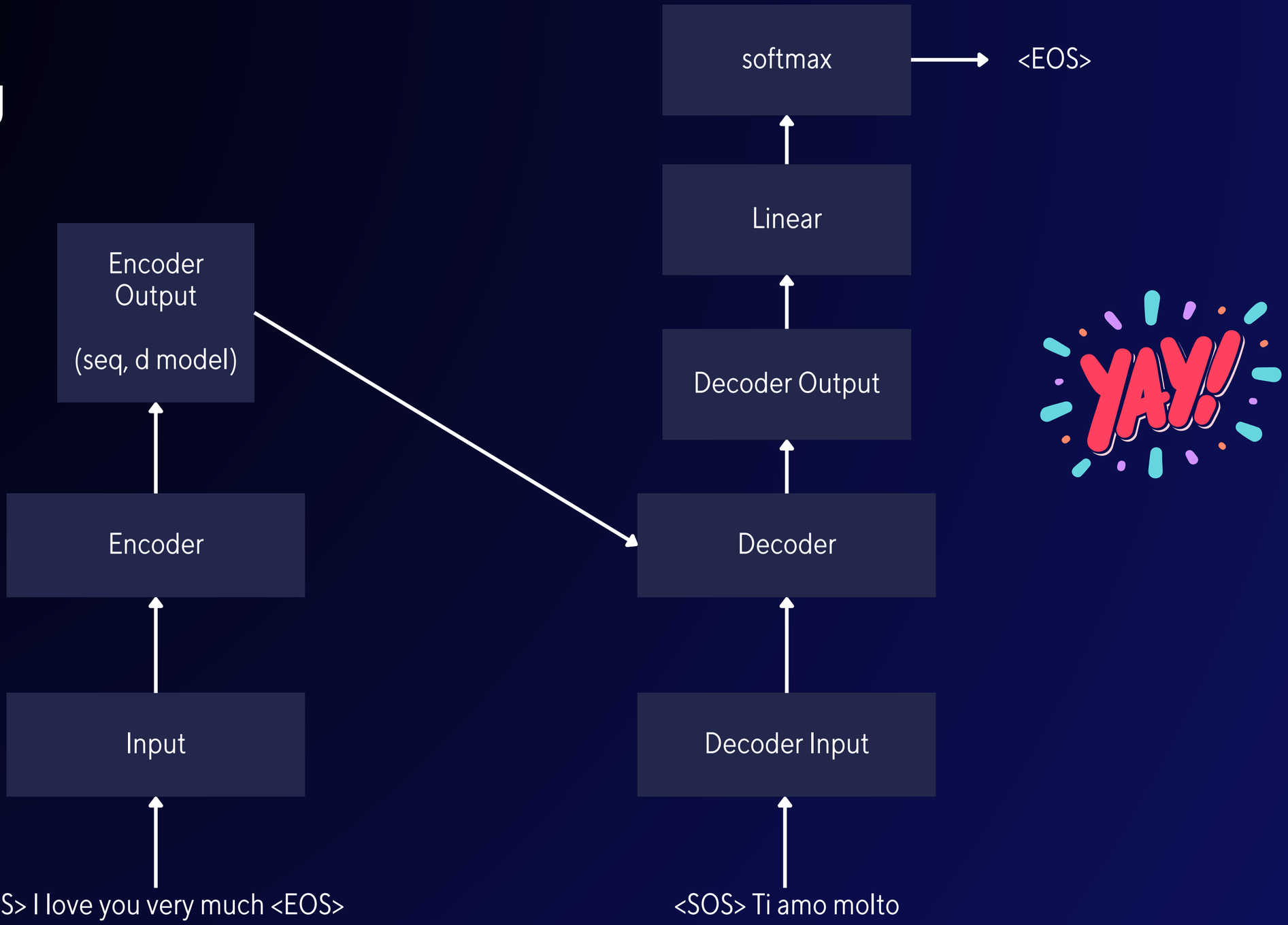
Inferencing

Time Step = 3



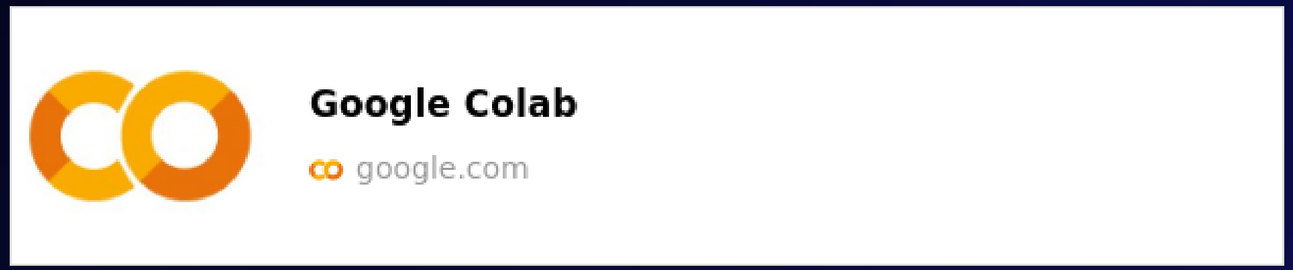
Inferencing

Time Step = 4



Coding Time!!!

Please Open Google Colab

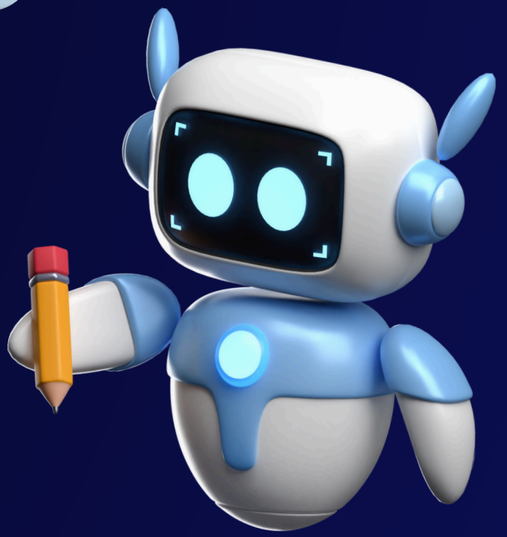


Question Time!

References

- https://arxiv.org/abs/1706.03762
- https://poloclub.github.io/transformer-explainer/
- http://nlp.seas.harvard.edu/2018/04/03/attention.html
- https://huggingface.co/docs/transformers/en/index
- https://pytorch.org

THANK YOU!



Agga Min @ Rei

Data Scientist / NLP Researcher

MemoryLab Inc., Tokyo, Japan

Tokyo International University

Linkedin

Youtube