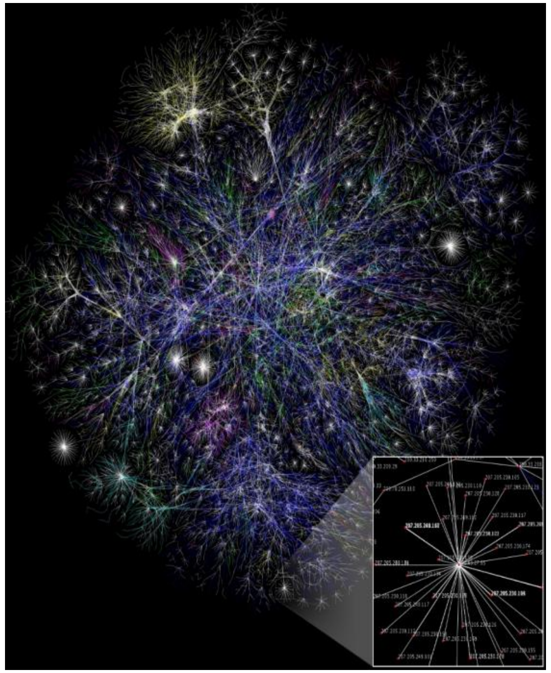
Introductory Course on Transformer Models and Generative AI Tools

- Introduction to Transformer Models
- Generative AI
- Applications of Transformer Models
- Training and Evaluating Transformer Models
- Advanced Topics in Transformers & Generative AI

Speaker : *Pranav Raikote, Senior AI Researcher, Plexflo* plexflo

Introduction

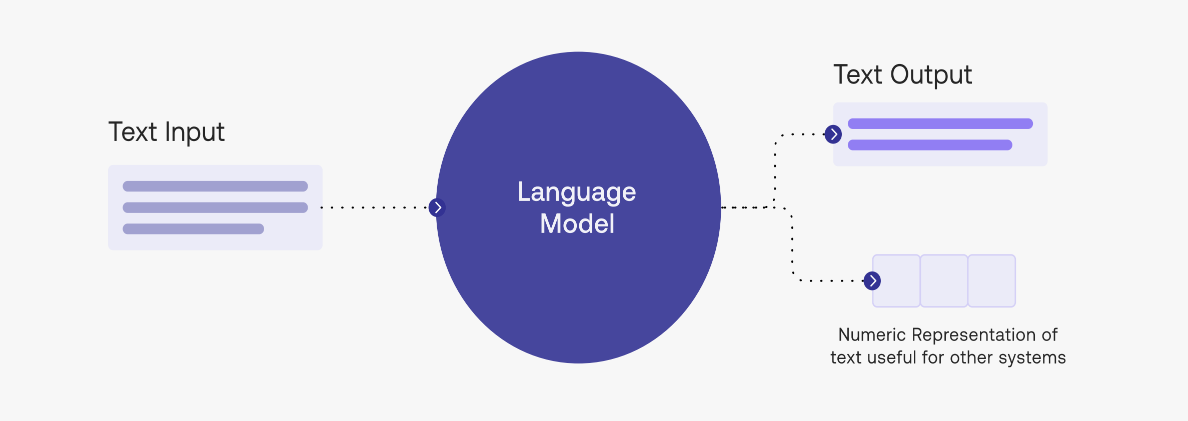
- Understanding Large Language Models
  - Language
  - LLM
  - Some of the Large Language Models
- Background About Transformer Models
- How are transformers different from previous Models
- What is Generative Artificial Intelligence (AI)?



Understanding Large Language Models

Language?

- It's how we learn about the world. Language is also how we connect and communicate — as people, and as groups and companies
- Despite the rapid evolution of software, computers remain limited in their ability to deal with language. Software is great at searching for exact matches in text, but often fails at more advanced uses of language — ones that humans employ daily
- There's a clear need for more intelligent tools that better understand language



LLM

- A large language model, or LLM, is a deep learning algorithm that can recognize, summarize, translate, predict and generate text and other content based on knowledge gained from massive datasets

Transformers

- ## Some of the Large Language Models
  - Pathways Language Model (PaLM) 540 billion parameter model, from Google Research
  - Generalist Language Model (GLaM) 1 trillion parameter model, from Google Research
  - Language Models for Dialog Applications (LaMDA) 137 billion parameter model from Google Research
  - Megatron-Turing NLG 530 billion parameter model, from Microsoft/Nvidia
  - DreamFusion/Imagen 3D image generation from Google Research
  - Get3D from Nvidia
  - MineClip from Nvidia
  - BLOOM: BigScience Large Open-science Open-access Multilingual Language Model with 176 billion parameters.

Why Transformers

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.

- ## Need for Transformers

  The dominant sequence transduction models are based on

  - Complex recurrent or convolutional neural networks
  - The best-performing models also connect the encoder and decoder through an attention mechanism.

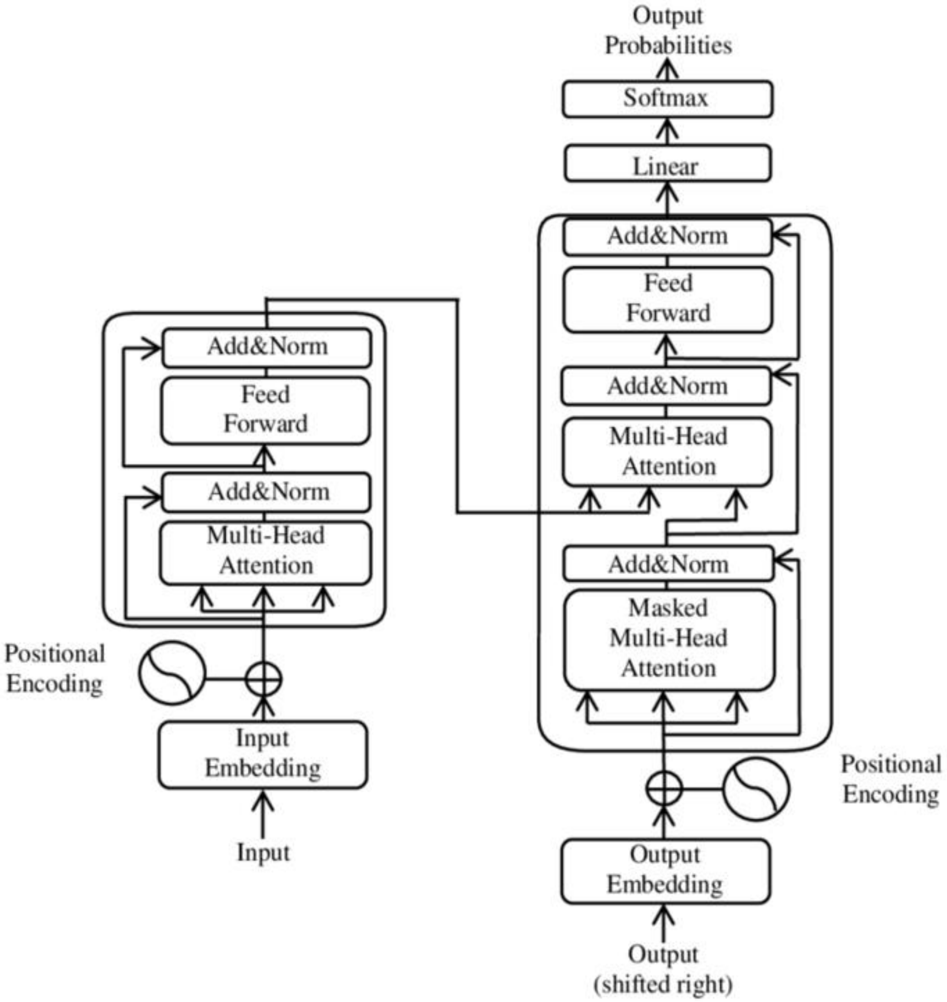
- ## Transformers

  - Developed in 2017 by Researchers at Google and the University of Toronto
  - Unlike recurrent neural networks, transformers could be very efficiently parallelized
  - With the right hardware, you could train some really big models.
  - Transformer is a deep learning model that adopts the mechanism of self-attention
  - It differentially weighs the significance of each part of the input data.
  - Like recurrent neural networks (RNNs), transformers are designed to process sequential input data, such as natural language with applications for tasks such as translation and text summarization.
  - However, unlike RNNs, transformers process the entire input all at once.

Transformers are a type of artificial intelligence (AI) model that uses a self-attention mechanism to process sequential data (When the points in the dataset are dependent on the other points in the dataset), such as natural language (a language that has developed naturally in use), by encoding and decoding input data into a continuous space. This allows for better generalization and understanding of the data, which can be used for various tasks such as language modeling, question answering, machine translation, and summarization.

Transformers are in many cases replacing convolutional and recurrent neural networks (CNNs and RNNs), the most popular types of deep learning models were developed just five years ago.

- **Model Architecture**
  - **Encoder and Decoder Stacks**
  - **Attention**
    - ✓ Scaled Dot-Product Attention
    - ✓ Multi-Head Attention
  - **Position-wise Feed-Forward Networks**
  - **Embeddings and Softmax**
  - **Positional Encoding**
- **Functioning**
  - Transduction models have an encoder-decoder structure
  - the encoder maps an input sequence of symbol representations (x1, ..., xn) to a sequence of continuous representations z = (z1, ..., zn)
  - The decoder then generates an output sequence (y1, ..., ym) of symbols one element at a time
  - At each step the model is auto-regressive consuming the previously generated symbols as additional input when generating the next.



Some of the Transformer Models

- Generative pre-trained transformer (GPT)
- GPT-2: Generative Pre-trained Transformer 2 with 1.5 billion parameters.
- GPT-3: Generative Pre-trained Transformer 3, with the unprecedented size of 2048-token-long context and 175 billion parameters (requiring 800 GB of storage).
- GPT-3.5/ChatGPT/InstructGPT from OpenAI
- GPT-NeoX-20B: An Open-Source Autoregressive Language Model with 20 billion parameters.
- BERT: Bidirectional Encoder Representations from Transformers (BERT)
- OPT-175B by Meta AI: another 175-billion-parameter language model. It is available to the broader AI research community.
- Point-E by OpenAI: a 3D model generator.
- RT-1 by Google: a model for operating robots
- VALL-E text to speech synthesis based on 3-second speech sample It was pre-trained on 60,000 hours of English speech from 7,000 unique speakers (dataset: LibriLight)

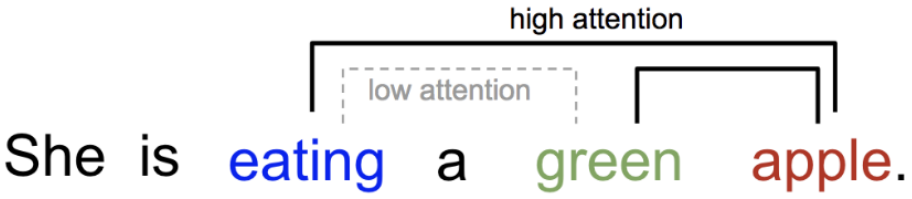
Problems in Training LLMs

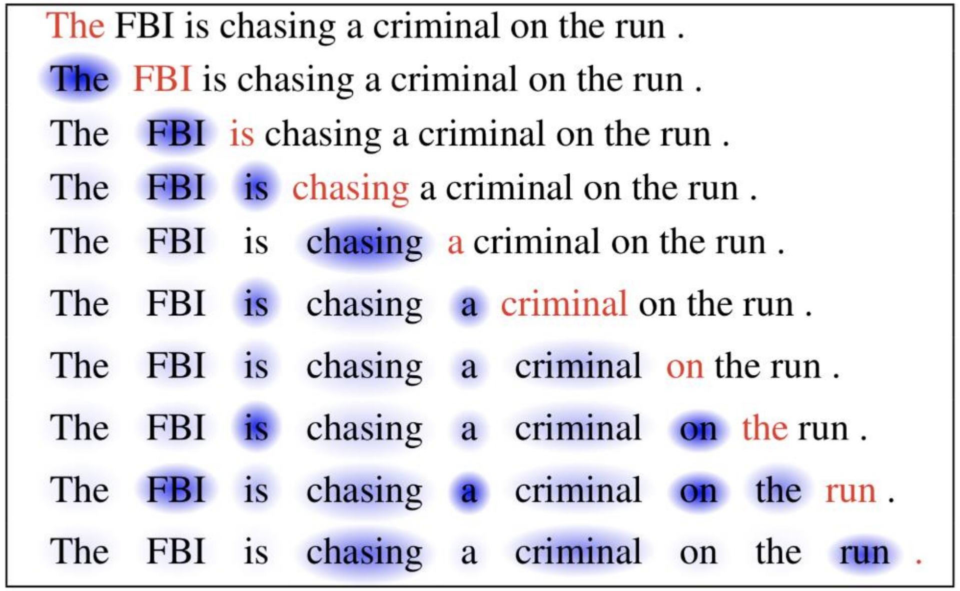
- Distributed/Parallel training
- Representation bottleneck
- Vanishing gradients
- Fixed context (sequence length)

Transformers

Attention Mechanism

- What should I pay attention to?





The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Image Credit

Transformer

- Highly capable Encoder-Decoder model with Self-Attention mechanism
- Train once and fine-tune on a variety of downstream tasks

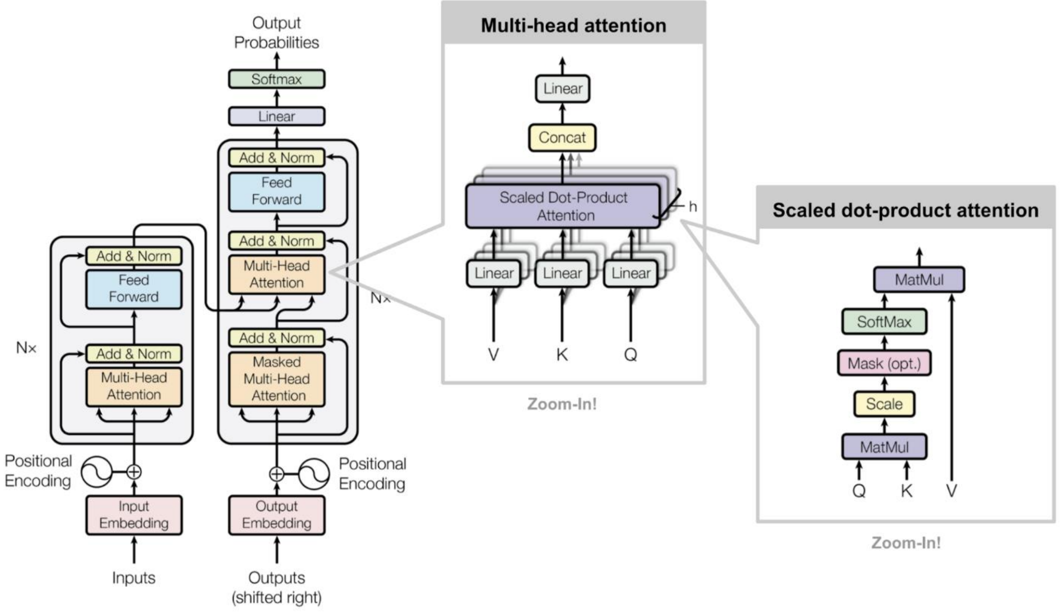
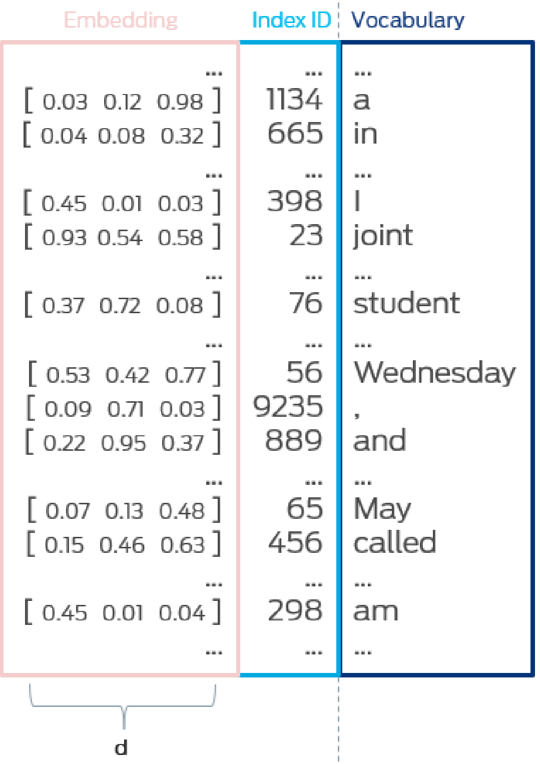


Image Credit

Embeddings

Input Embeddings



Embedding	Index ID	Vocabulary
...	...	...
[0.03 0.12 0.98]	1134	a
[0.04 0.08 0.32]	665	in
...	...	...
[0.45 0.01 0.03]	398	I
[0.93 0.54 0.58]	23	joint
...	...	...
[0.37 0.72 0.08]	76	student
...	...	...
[0.53 0.42 0.77]	56	Wednesday
[0.09 0.71 0.03]	9235	,
[0.22 0.95 0.37]	889	and
...	...	...
[0.07 0.13 0.48]	65	May
[0.15 0.46 0.63]	456	called
...	...	...
[0.45 0.01 0.04]	298	am
...	...	...

d

Positional Embeddings

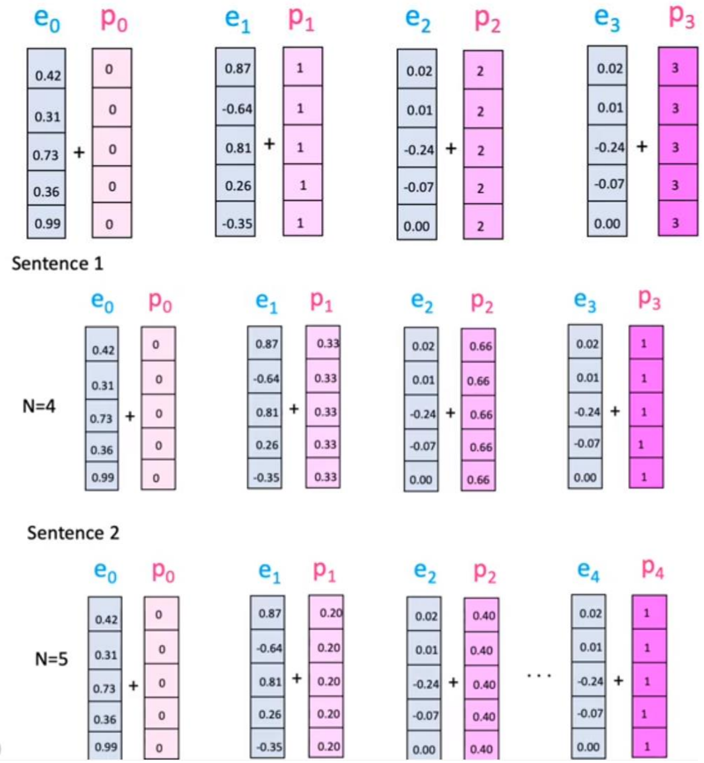
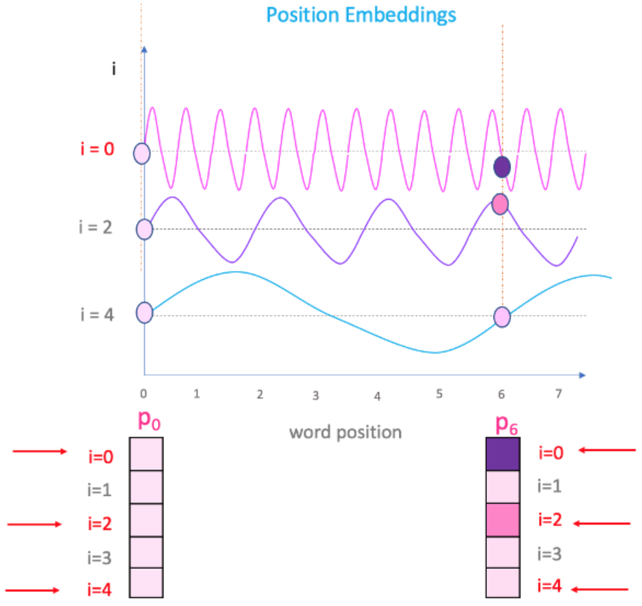


Image Credit

Positional Encoding

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

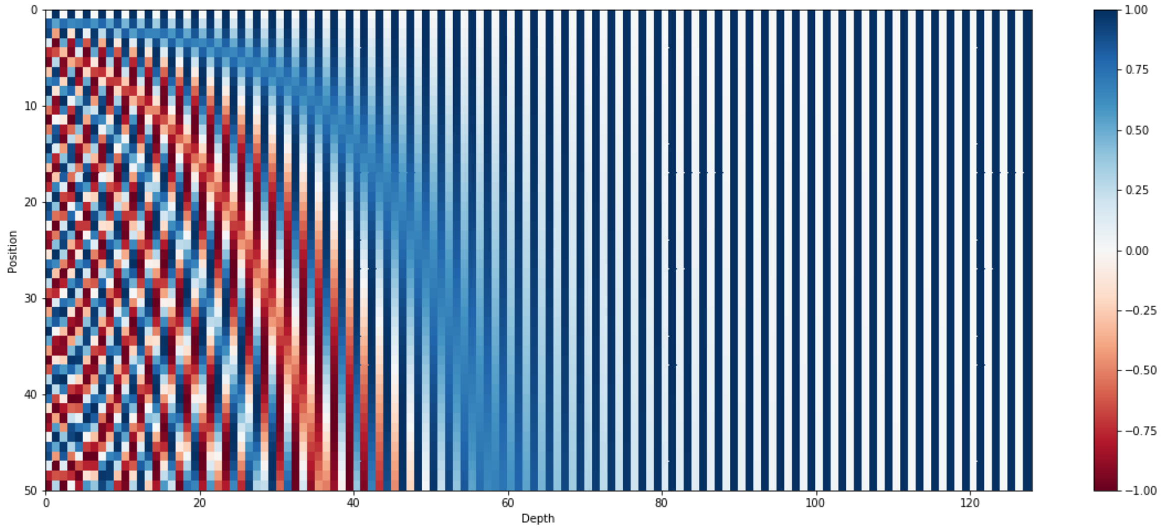
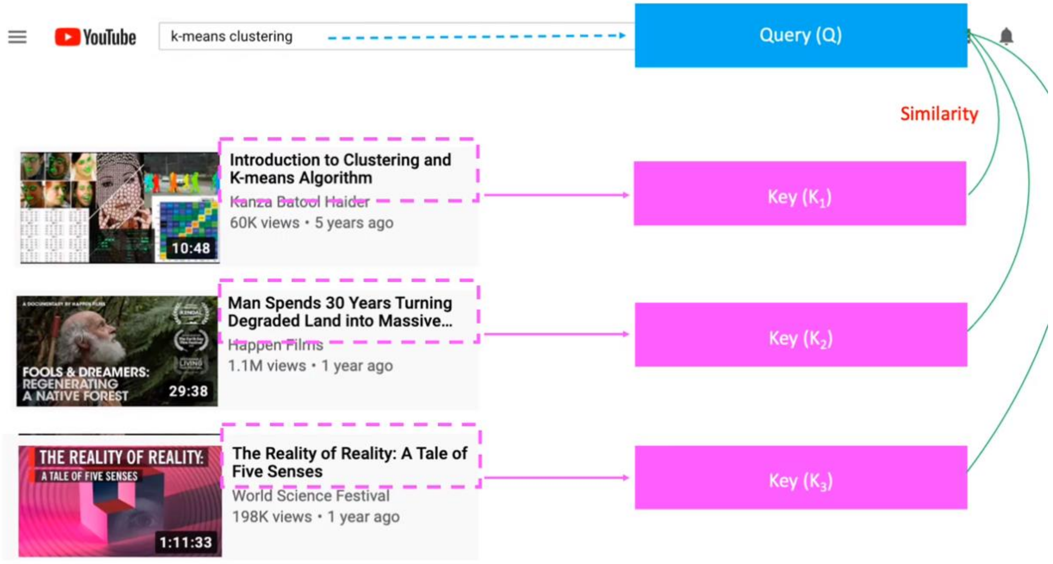


Image Credit

Self-Attention

He went to the **bank** and learned of his empty account, after which he went to a river **bank** and cried



Introduction to Clustering and K-means Algorithm
Kanza Batool Haider
60K views • 5 years ago

Man Spends 30 Years Turning Degraded Land into Massive...
Happen Films
1.1M views • 1 year ago

The Reality of Reality: A Tale of Five Senses
World Science Festival
198K views • 1 year ago

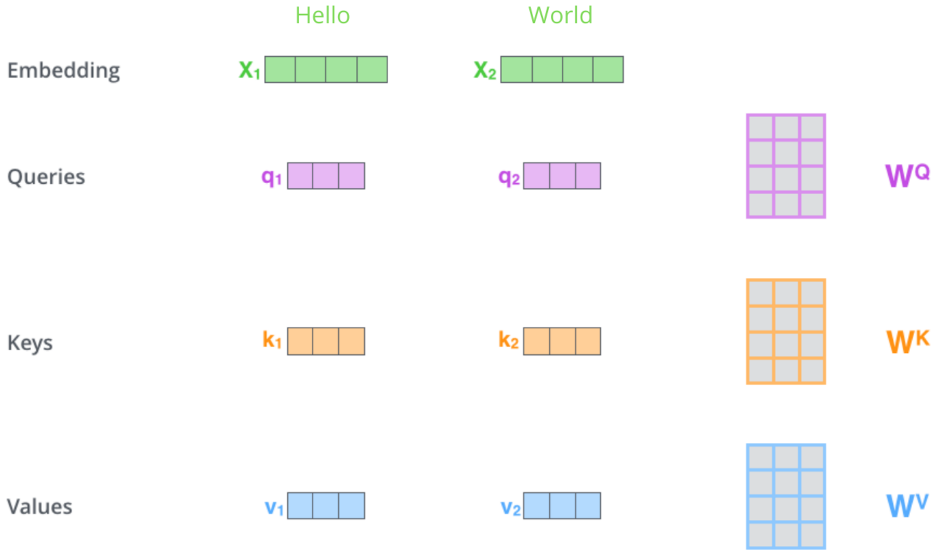
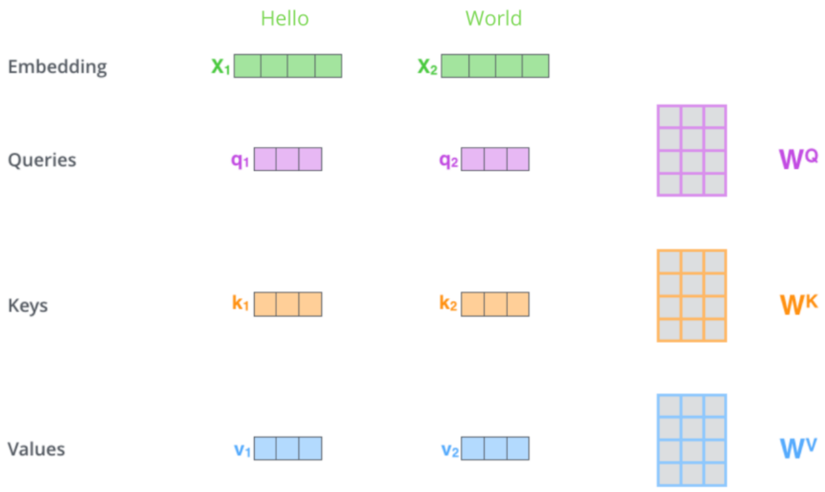
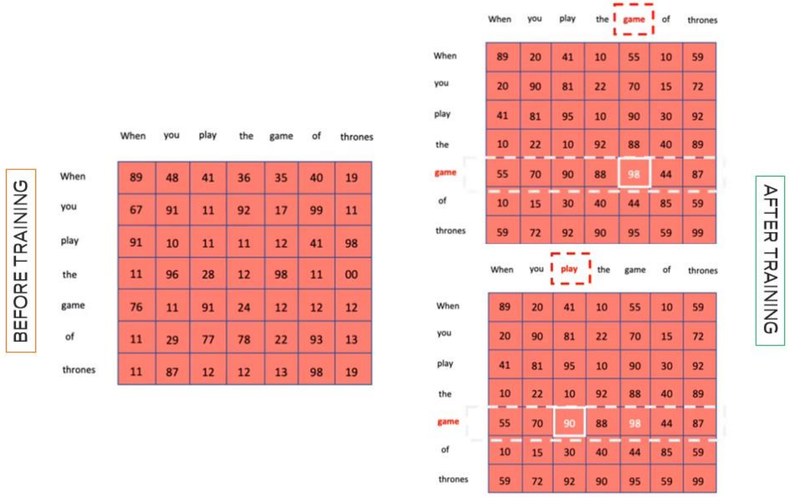


Image Credit

Self-Attention



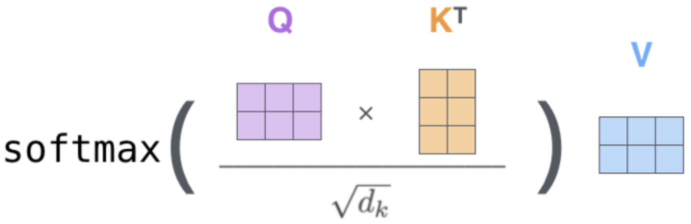

BEFORE TRAINING

	When	you	play	the	game	of	thrones
When	89	48	41	36	35	40	19
you	67	91	11	92	17	99	11
play	91	10	11	11	12	41	98
the	11	96	28	12	98	11	00
game	76	11	91	24	12	12	12
of	11	29	77	78	22	93	13
thrones	11	87	12	12	13	98	19

AFTER TRAINING

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	10	59
you	20	90	81	22	70	15	72
play	41	81	95	10	90	30	92
the	10	22	10	92	88	40	89
game	55	70	90	88	98	44	87
of	10	15	30	40	44	85	59
thrones	59	72	92	90	95	59	99

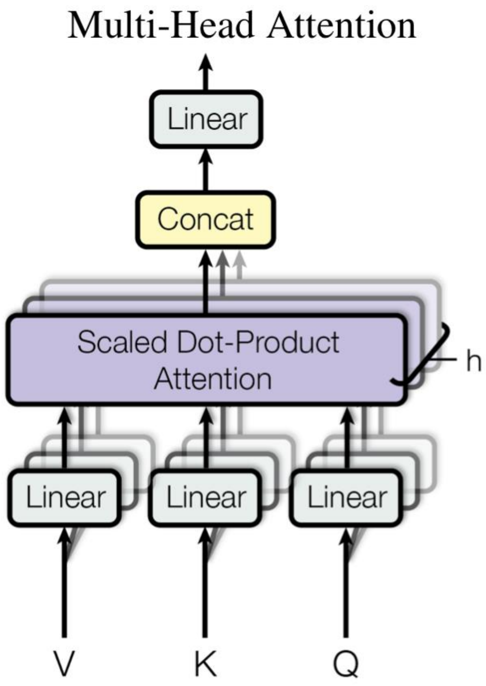
	When	you	play	the	game	of	thrones
When	89	20	41	10	55	10	59
you	20	90	81	22	70	15	72
play	41	81	95	10	90	30	92
the	10	22	10	92	88	40	89
game	55	70	90	88	98	44	87
of	10	15	30	40	44	85	59
thrones	59	72	92	90	95	59	99

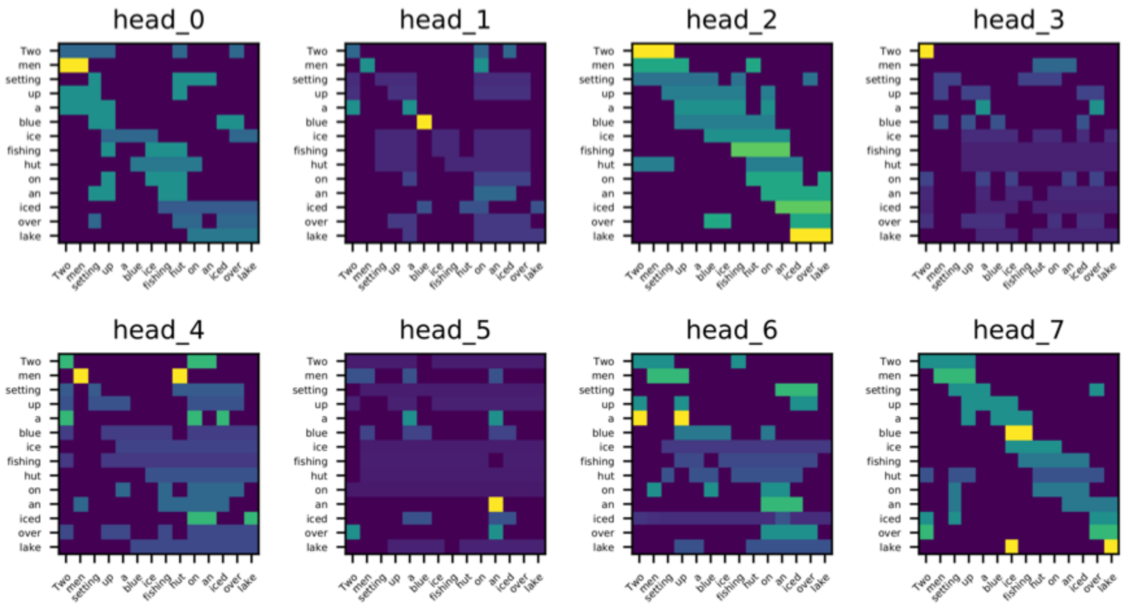


$$\mathrm{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V$$

Image Credit

Multi-Head Attention

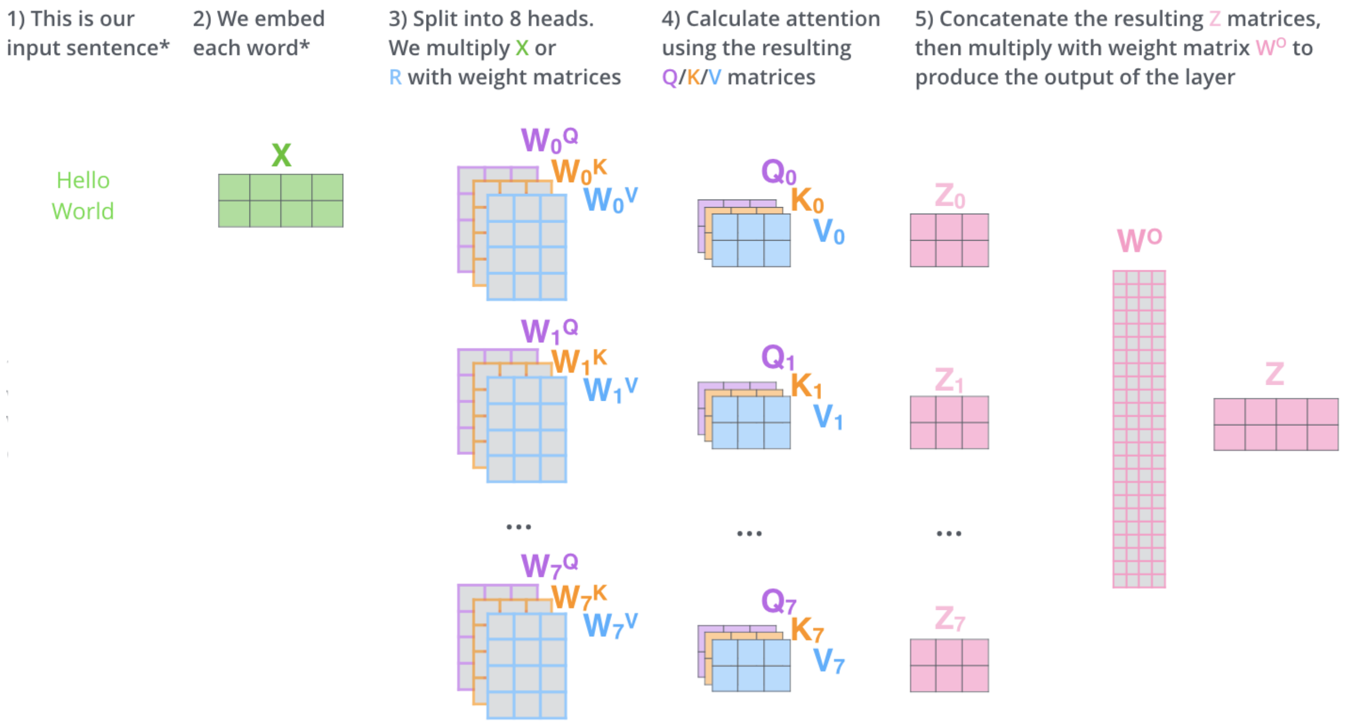




- 8 heads
- Representation sub-space
- Each head will have a 64-dim vector

Image Credit

Multi-Head Attention (Recap)



1) This is our input sentence*

2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Image Credit

Residual Connections & Feed Forward Network

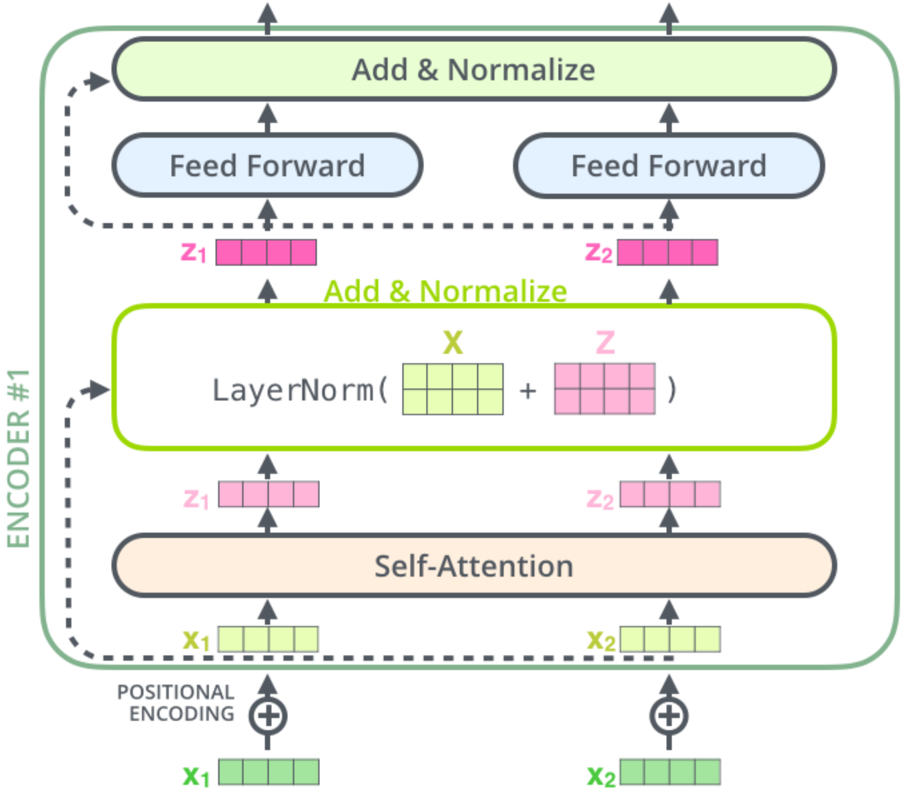
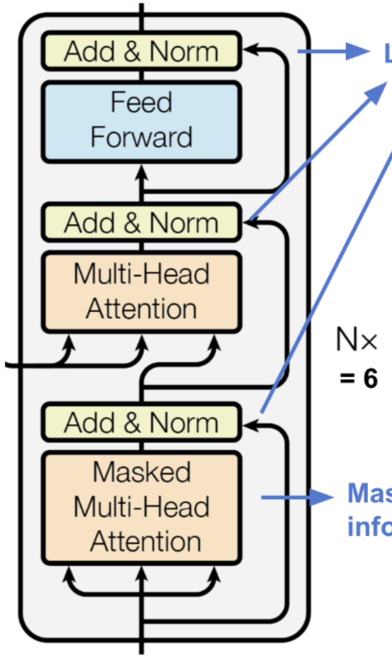


Image Credit

Decoder

- Autoregressive in nature
- 2 special Multi-Head Attention layers

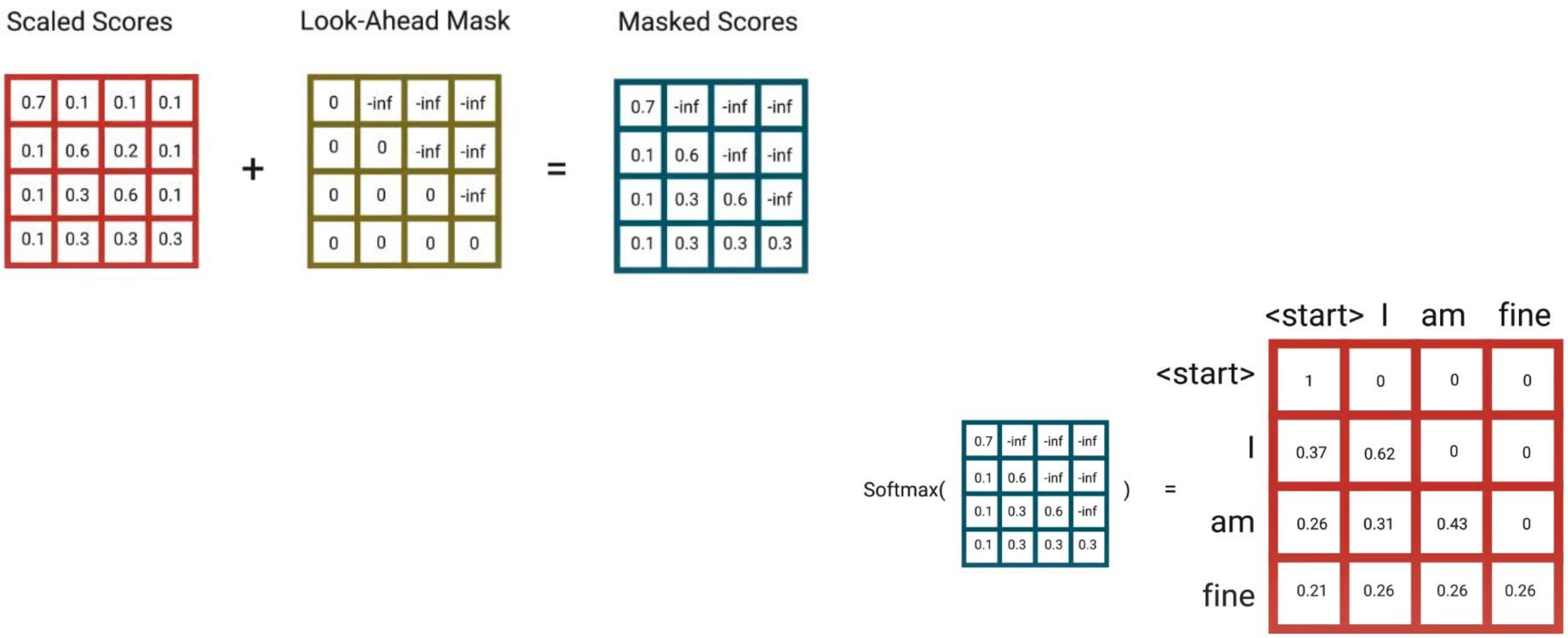


LayerNorm(x + SubLayer(x))

Masked: not to use the information in the future.

Image Credit

Masked Multi-Head Attention



Scaled Scores

Look-Ahead Mask

Masked Scores

$$\begin{bmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.6 & 0.1 \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix} + \begin{bmatrix} 0 & -inf & -inf & -inf \\ 0 & 0 & -inf & -inf \\ 0 & 0 & 0 & -inf \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.7 & -inf & -inf & -inf \\ 0.1 & 0.6 & -inf & -inf \\ 0.1 & 0.3 & 0.6 & -inf \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix}$$

$$\text{Softmax(}\begin{bmatrix} 0.7 & -inf & -inf & -inf \\ 0.1 & 0.6 & -inf & -inf \\ 0.1 & 0.3 & 0.6 & -inf \\ 0.1 & 0.3 & 0.3 & 0.3 \end{bmatrix}\text{)} = \begin{array}{r|cccc} & \text{<start>} & \text{I} & \text{am} & \text{fine} \\ \hline \text{<start>} & 1 & 0 & 0 & 0 \\ \text{I} & 0.37 & 0.62 & 0 & 0 \\ \text{am} & 0.26 & 0.31 & 0.43 & 0 \\ \text{fine} & 0.21 & 0.26 & 0.26 & 0.26 \end{array}$$

Image Credit

Encoder-Decoder Attention

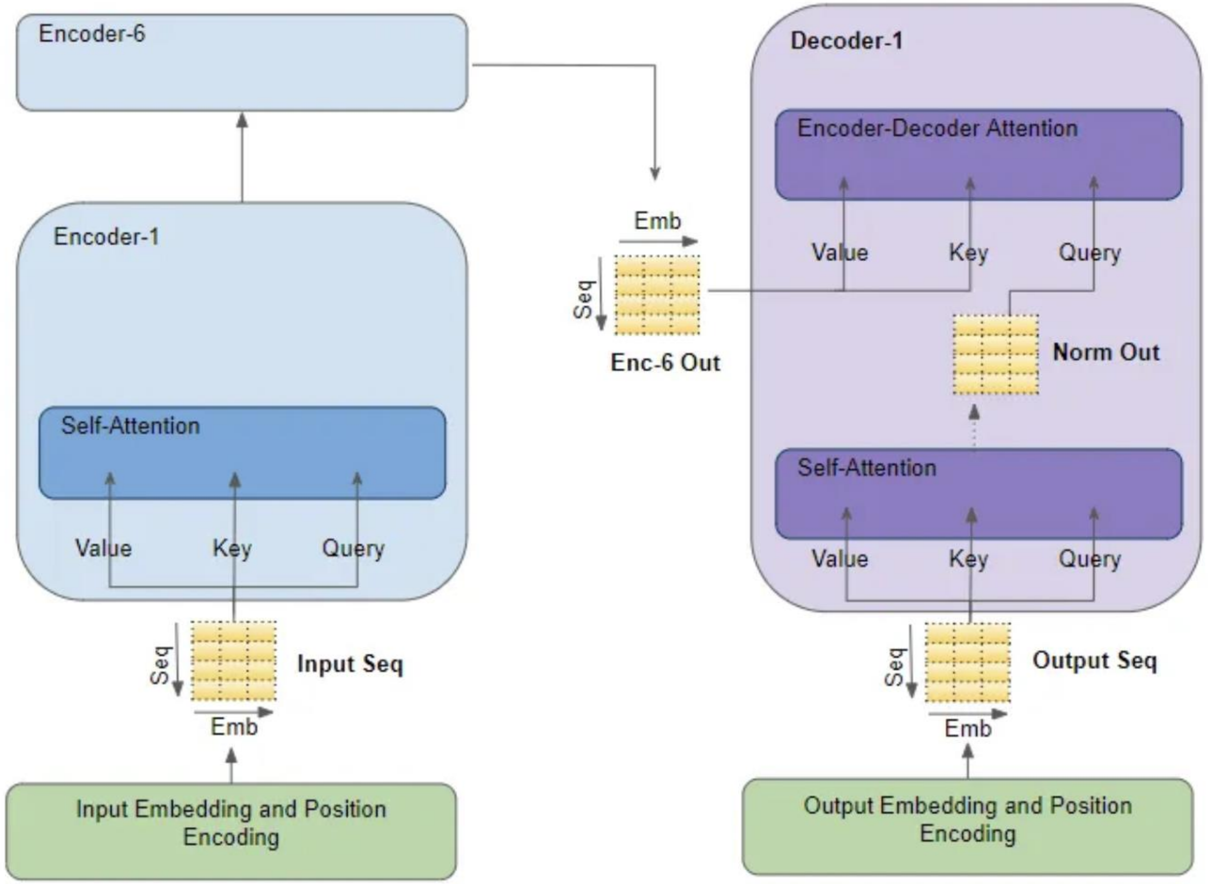


Image Credit

Final Layers & Loss Function

- Linear Layer with dimension equivalent to number of unique words
- Softmax Layer for class probabilities

- Adam optimizer with Categorical Cross-entropy (Log) Loss

$$Logloss_i = -[y_i \, ln \, p_i + (1 - y_i) \, ln(1 - p_i)]$$

Impact of Transformers

- New avenue and direction in Deep Learning w.r.t NLP & LLMs
- Initially it started off with an idea/approach to understand how languages function and how to represent them
- 66000+ citations and counting
- Base architecture to a plethora of new-age architectures like GPT and BERT

Generative AI

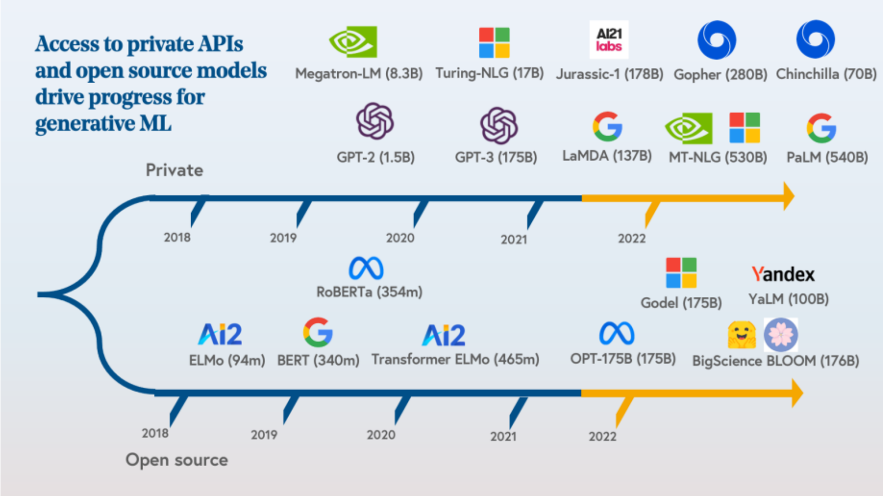
Generative AI

- Creation of data in various modalities like Images, Text, Audio, Video, and so on
- Solves various data problems
  - Data scarcity
  - Edge cases
  - Dangerous situations

AI can not only boost our analytic and decision-making abilities but also heighten creativity.

History of Generative AI

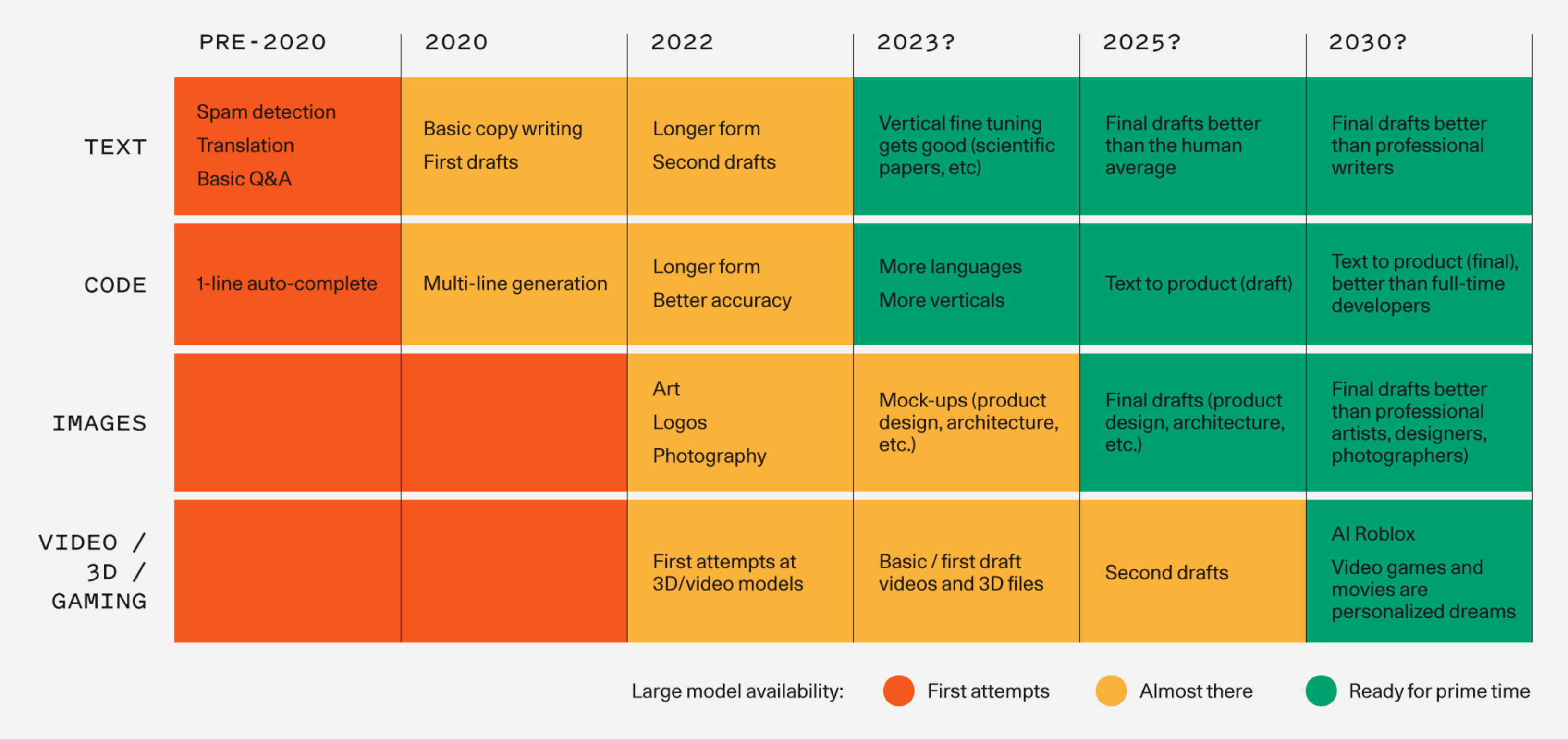
- First explored in 1950-60s, gained traction in 2010s
- Breakthrough in 2014 by Ian Goodfellow’s GANs paper
- Going really strong and dominating the Deep Learning space
  - ChatGPT, ImageGPT, CLIP, DALL-E, Stable Diffusion, etc.



Coolest idea in Deep Learning in the last 20 years – Yann LeCun

Image Credit

Generative AI Timeline



	PRE-2020	2020	2022	2023?	2025?	2030?
TEXT	Spam detection Translation Basic Q&A	Basic copy writing First drafts	Longer form Second drafts	Vertical fine tuning gets good (scientific papers, etc)	Final drafts better than the human average	Final drafts better than professional writers
CODE	1-line auto-complete	Multi-line generation	Longer form Better accuracy	More languages More verticals	Text to product (draft)	Text to product (final), better than full-time developers
IMAGES			Art Logos Photography	Mock-ups (product design, architecture, etc.)	Final drafts (product design, architecture, etc.)	Final drafts better than professional artists, designers, photographers)
VIDEO / 3D / GAMING			First attempts at 3D/video models	Basic / first draft videos and 3D files	Second drafts	AI Roblox Video games and movies are personalized dreams

Large model availability: 🔴 First attempts 🟡 Almost there 🟢 Ready for prime time

Global AI investment surged from $12.75 million in 2015 to $93.5 billion in 2021, and the market is projected to reach $422.37 billion by 2028. Already over $2B has been invested in Generative AI, up 425% since 2020

Image Credit

Types of Generative Models

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Autoregressive Models (Transformers-XL, PixelCNN, WaveNet)
- Markov Chain Monte Carlo Models

Generative Adversarial Networks

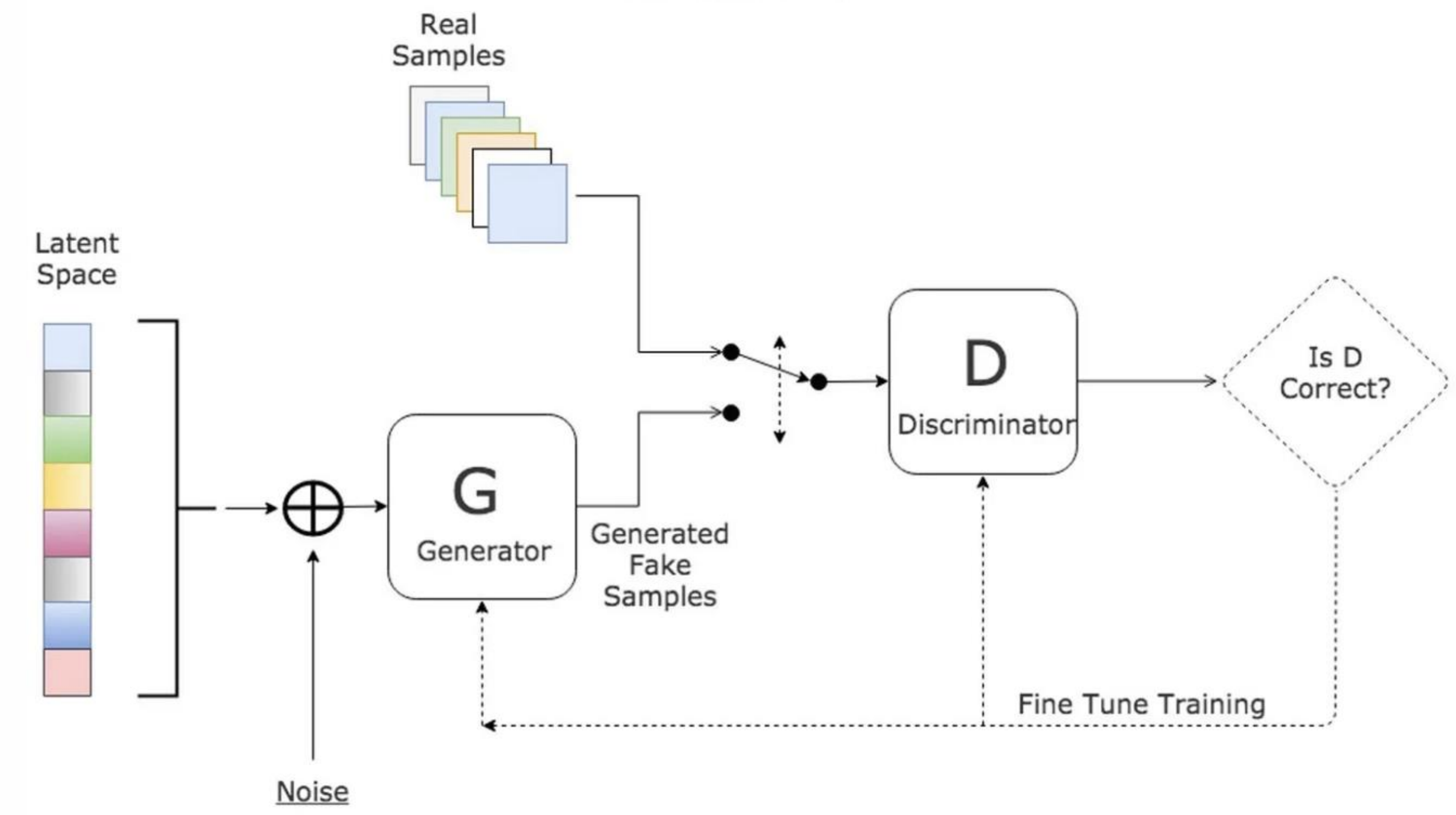
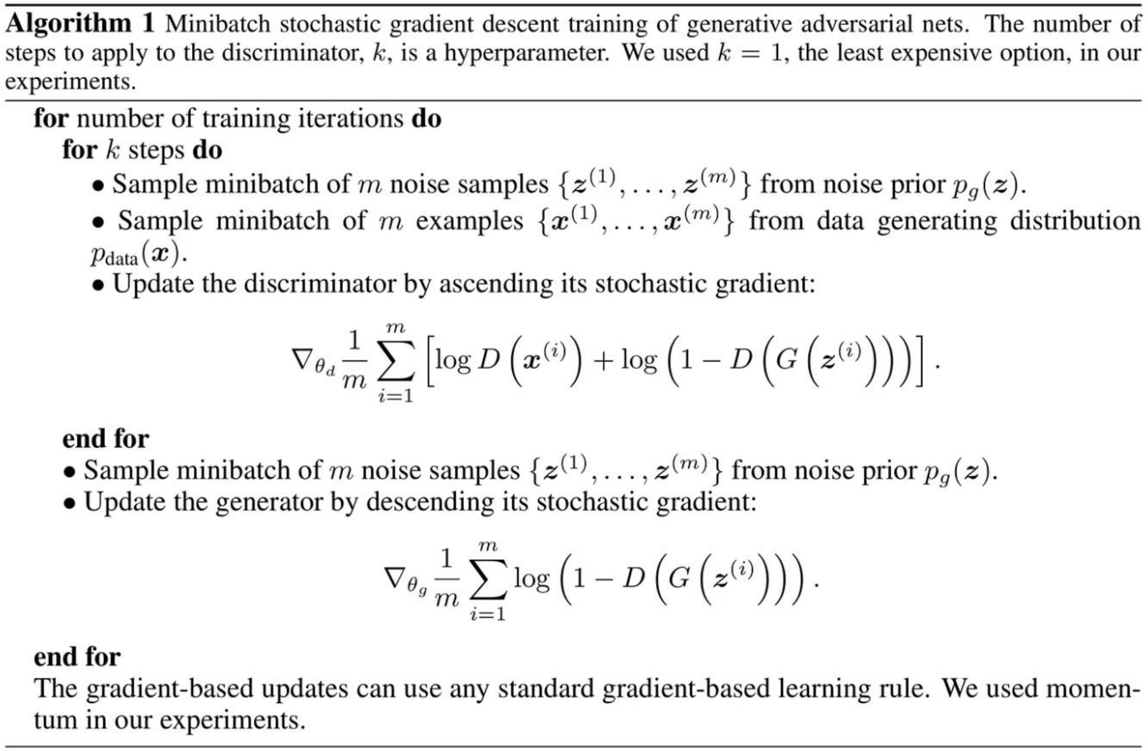


Image Credit

Generative Adversarial Networks



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

```
for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
        • Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
        • Update the discriminator by ascending its stochastic gradient:
            $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$
    end for
    • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.
```

Image Credit

Generative Adversarial Networks

- Problems/Nuances with GANs
  - Mode Collapse
  - Nash Equilibrium
  - Very Very Very difficult to train and get good outputs

- Popular architectures
  - Pix2Pix
  - StyleGAN
  - CycleGAN
  - VideoGAN

Auto-Encoders

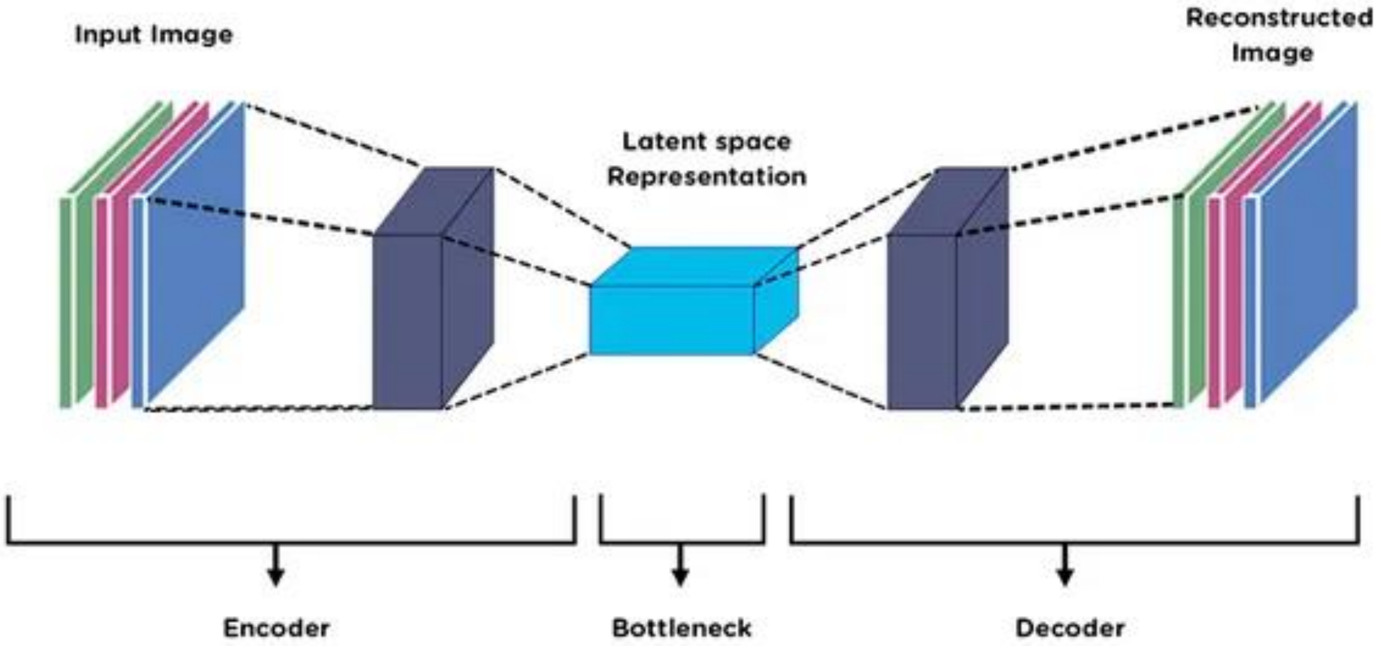


Image Credit

Variational Auto-Encoders

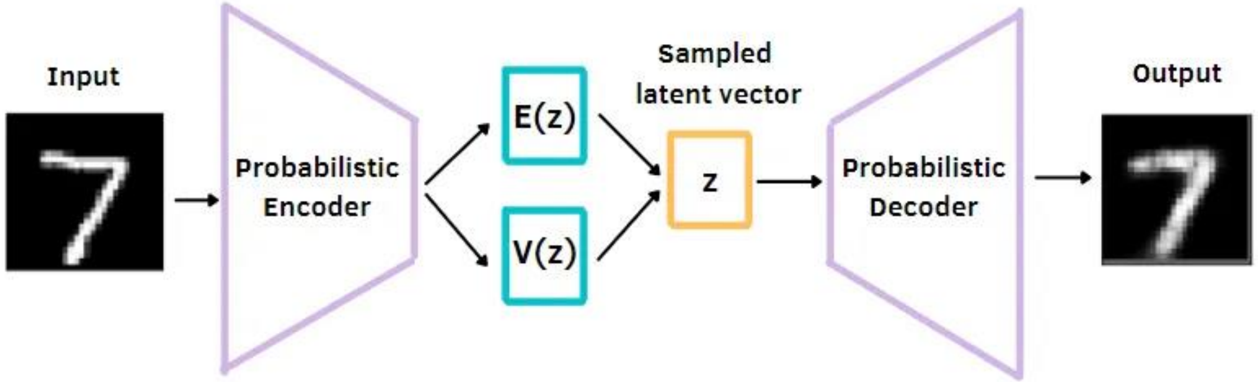
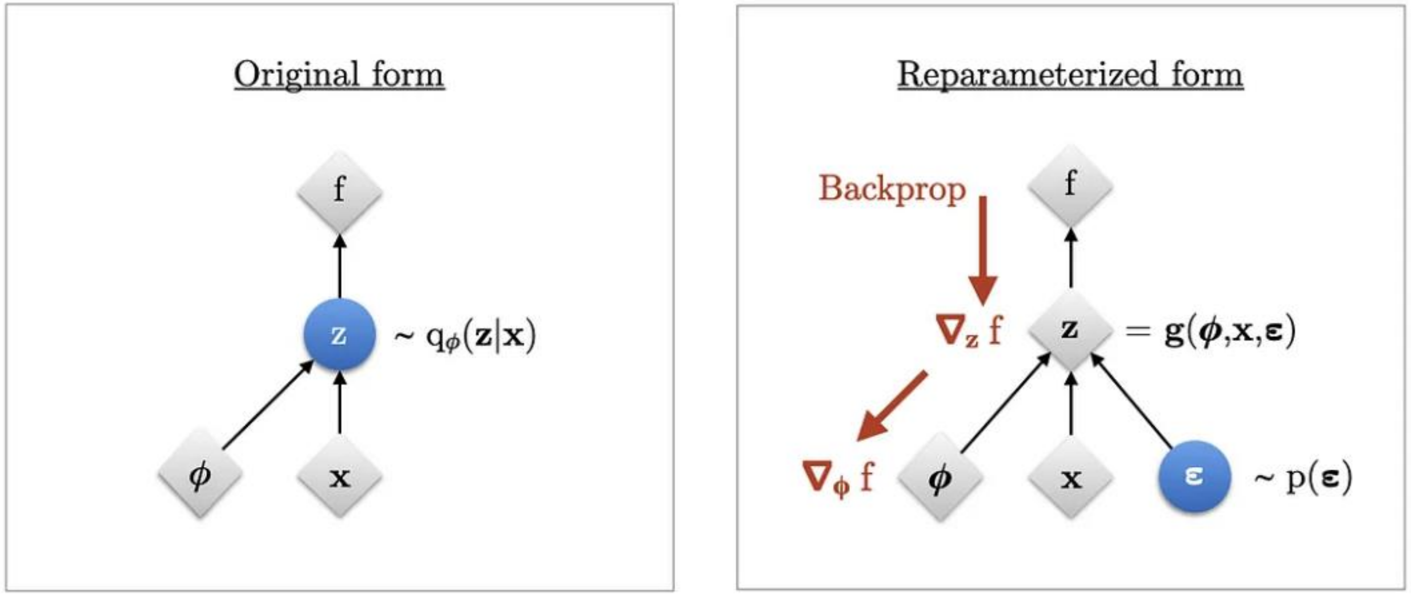


Image Credit

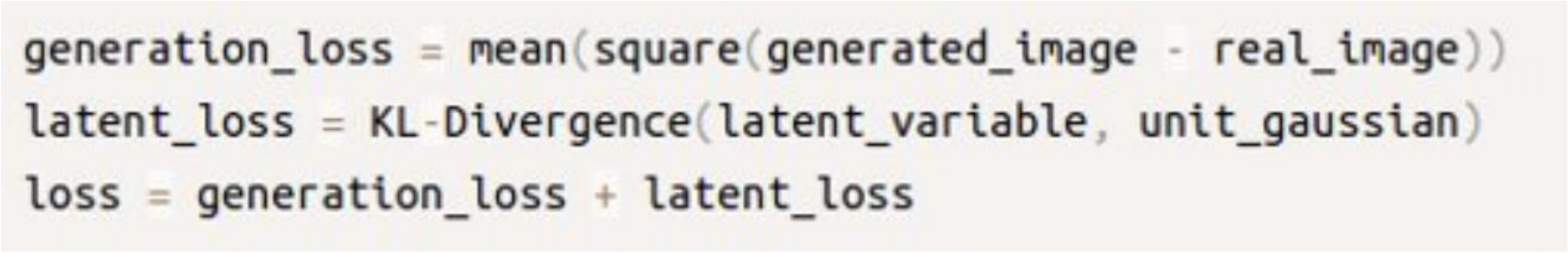
Variational Auto-Encoders



Original form

Reparameterized form

f
z
$\sim q_\phi(\mathbf{z}|\mathbf{x})$
ϕ
x
Backprop
f
$\nabla_z f$
z
$= g(\phi,\mathbf{x},\boldsymbol{\varepsilon})$
$\nabla_\phi f$
ϕ
x
ε
$\sim p(\boldsymbol{\varepsilon})$



```
generation_loss = mean(square(generated_image - real_image))
latent_loss = KL-Divergence(latent_variable, unit_gaussian)
loss = generation_loss + latent_loss
```

Image Credit

Auto-regressive Models

- Generate a token, given the previous token in an iterative manner
- Tend to forecast future behaviour
  - High impact in NLP & Time-Series domains
- Examples: Transformers-XL, GPT

Applications of Generative AI

- Realistic and diverse images. Applications in Graphics, Video games, Super resolution, Art, In-painting
- Language Translation, Summarization, Conversational AI
- Simulate complex systems
- Generate data for scientific research
  - Climate modelling
  - Drug discovery
  - Protein folding
- Power & Utility
  - Synthetic data
  - Identify and replicate energy saving patterns
  - Text2Sim, Text2Image, Text2Code, etc.

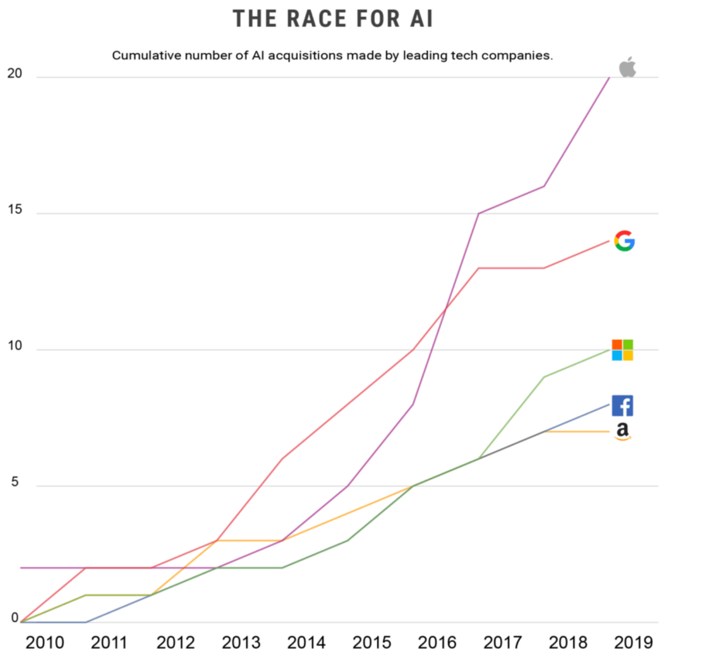
Popular Generative AI Models

- Co-Pilot/Codex, Stable Diffusion, PaLM, LaMDA, ChatGPT
- CLIP, DALL-E
- ImageGPT, GPT-3
- StyleGAN2
- BigGAN
- WaveNet
- DeepDream

Applications of Transformers

Applications of Transformers

- Regarded as AI 2.0
- Accelerated R&D in various industries
- Cross domain problem statements
    - Language, Images, Videos, Speech, Tabular
- Advances in Drug Discovery, Protein Folding
- Carbon Footprint



CBINSIGHTS

Image Credit

Applications of Transformers

- Vision
  - Image Classification: ViT, DeiT
  - Object Detection, Segmentation: DETR, SWIN Transformer
- Music
  - Music Transformer
  - MusicLM: Synthetic Music generation from text desciptions
- NLG
  - Image Captions, Generating Reports, Chatbots, Transcriptions
  - E-Commerce, Banking, Media, Art & Poetry, Assistants

Applications of Transformers

- Power & Utility Industry
  - Disaggregation
  - Load Forecasting
  - Pricing optimization
  - Solar & EV Propensity
  - Demand and Response
  - Material & Design Innovations
  - Outage Predictions, Grid Stability & Resilience, Restoration

Training & Evaluating Transformers

Training & Evaluating Transformers

- Data
  - Massive data required ~A few TBs
  - Little to no pre-processing
  - Data representation
- Compute
  - Massive GPU servers, mostly one-time investment
- Hyperparameters
  - 6 blocks of Encoders and Decoders
  - 8 Nvidia P100 GPUs, 300,000 steps optimized by Adam (varied LR)

Training & Evaluating Transformers

- Outline
  - Dataset -> Character level or Word level tokenization -> Indexing
  - Generate sequences from data
    - If Seq2Seq, then additional masking operation to be done
  - Embedding Representation
  - Define the functional blocks
  - Define the hyperparameters
  - Training & Testing loop
  - Stand-alone function for saving, loading and generation of text
- Evaluation Metrics
  - BLEU
  - Perplexity

Transformers in Action!

Advanced Topics in Transformers & Generative AI

Advanced Topics in Generative AI

- Transformers in Reinforcement Learning
  - Modified into a “Big Sequence Modelling”
  - Decision Transformers
- Policy Based Learning
  - ChatGPT
- DIET (Dual Intent Entity Transformer)

Decision Transformer

Image Credit

DIET

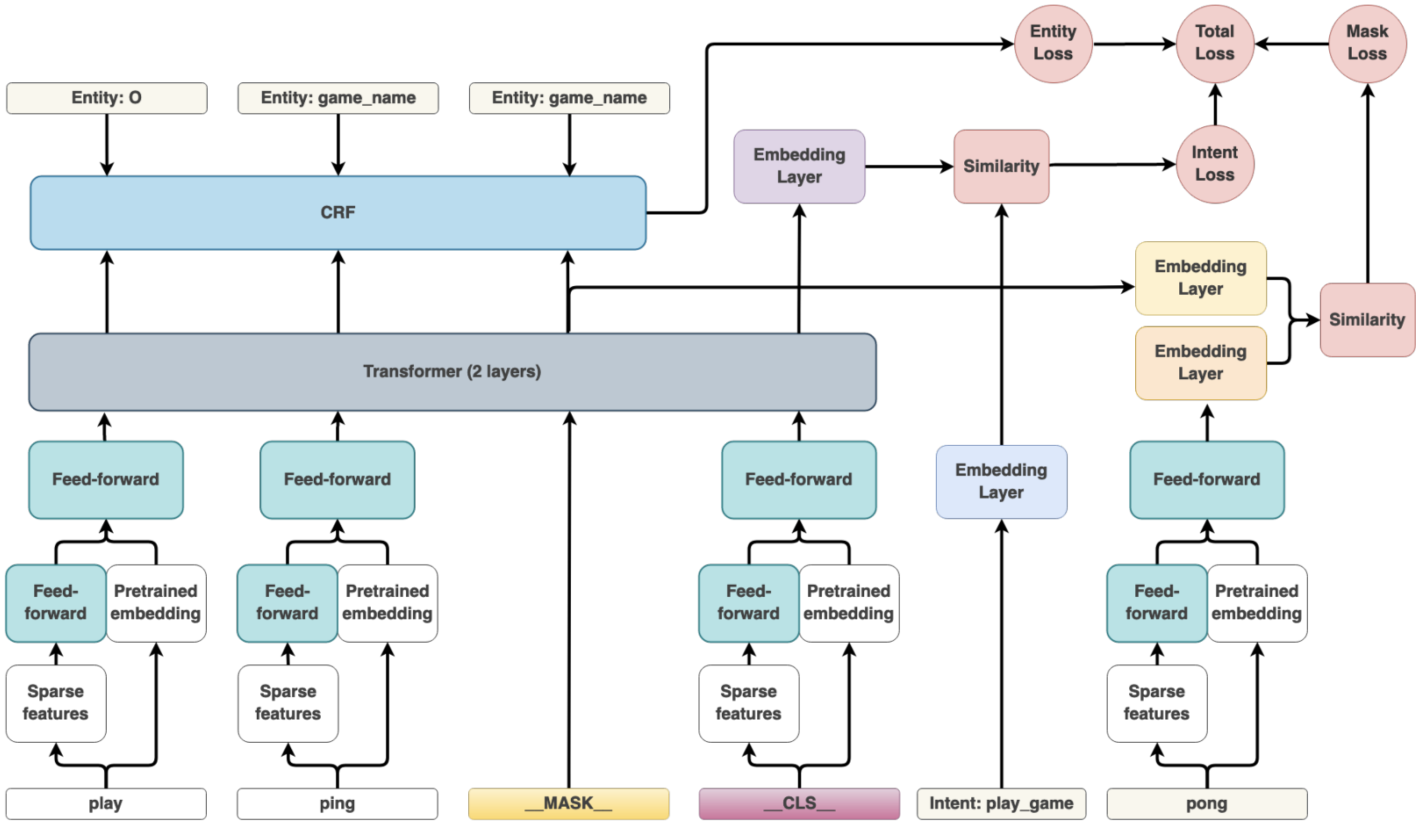


Image Credit

Key Takeaways/ Recommendations

- Large Language Models & Transformers
  - Understanding of How? Why? And Impact!
- Generative AI
  - Concept and Fundamental Architectures
  - Impact, Applications & Opportunities
- Train your own Generative AI Model
  - Generate novel language
  - How to use pre-trained models for downstream tasks
- Future of Power & Utility Industry with Generative AI

Thank You

For discussions/suggestions/queries email: ***isuw@isuw.in***
www.isuw.in
Links/References (If any)