

# AI - Past, Present & Future

Jaideep Ganguly, ScD (MIT)

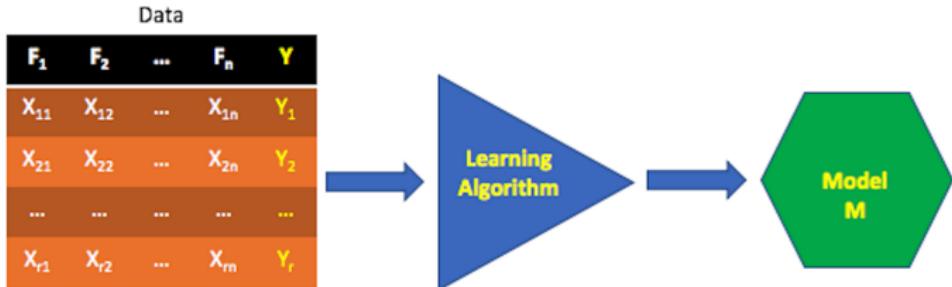
March 31, 2024

# About Jaideep

- ① VP of Software Development, Salesforce
- ② Director of Software Development, Amazon [2010 - 2019].
- ③ Director of Engineering, Microsoft [2006 - 2010].
- ④ Doctor of Science, Massachusetts Institute of Technology (MIT).
- ⑤ Master of Science, Massachusetts Institute of Technology (MIT).
- ⑥ Bachelor of Technology, Indian Institute of Technology, Kharagpur.

# What is Machine Learning (ML)? Classification!

- ① In 1959, Arthur Samuel (MIT), defined ML, a subset of AI, as a “*field of study that gives computers the ability to learn without being explicitly programmed*”.



- ② ML is effective for complex tasks where deterministic solution don't suffice, e.g., speech recognition, handwriting recognition, spam, fraud detection, etc. These cannot be solved manually in a large scale.

# Can a Computer Think?

- ① The human brain is a remarkable organ. It has enabled us understand science and advance mankind.
- ② The idea of mimicking the human brain or even improving the human cognitive functions is an alluring one and is an objective of Artificial Intelligence research.
- ③ But we are not even close in-spite of a century of research. However, it continues to have a major hold on our imagination given the potential of the rewards.
- ④ *The question of whether a computer can think is no more interesting than the question of whether a submarine can swim" - Dijkstra*
- ⑤ It is more interesting to understand the evolution of Machine Learning  
- *How did it start, here are we today and where do we go from here.*

# Knowledge Based (KB) Expert Systems - 1980-1995

- ① The field of AI was defined as computers performing tasks that were specifically thought of as something only humans can do.
- ② In the 1980s, the expert systems were of great interest and focused on knowledge and inference mechanisms. They did a good job in their domains but were narrow in specialization and were difficult to scale.
- ③ Once these systems worked, they were no longer considered to be AI! For example, today the best chess players are routinely defeated by computers but chess playing is no longer really considered as AI! [McCarthy](#) referred to as the "AI effect". IBM's Watson is one such program at a level such as that of a human expert.
- ④ Fifty years ago [Jim Slagle's \(MIT\)](#) symbolic integration program (MACSYMA) was a tremendous achievement.
- ⑤ It is very hard to build a program that has "common sense" and not just narrow domains of knowledge.

# From Perceptrons to Deep Learning (DL - 1995-2015)

- ① Rosenblatt (Cornell) is credited with the concept of Perceptrons, "a machine which responds like the human mind" as early as in 1957. In a critical book written in 1969, Marvin Minsky and Seymour Papert showed that Rosenblatt's original system was blind to simple XOR.
- ② In 2006, Hinton developed Deep Learning which extends earlier important work by Yann LeCun (New York Univ). In ImageNet 2012, Hinton achieved the best accuracy in image recognition over > 10%.
- ③ DL's innovation is to have models learn categories incrementally, attempting to nail down lower-level categories (like letters) before attempting to acquire higher-level categories (like words).
- ④ DL powers Amazon's Alexa, Google's search, Facebook's news feed and Netflix's recommendation engine.
- ⑤ Every decade focus has shifted - Neural Networks in the late '50s and '60s, symbolic approaches in the '70s, KB systems in the '80s, Bayesian Networks in the '90s, Support Vector machines in the '00s, and Neural Networks again in the '10s. Now GPT, LLM in the '20s.

# Linear Regression - Elementary Algebra/Calculus

- ① **Regression** is a statistical approach to find the relationship between variables between  $X_i$  and  $Y_i$ .
- ② A common function used to model training data is a **linear regression model**. The **model** or the **hypothesis** is given by:

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \dots + \theta_n x_{in} \quad (1)$$

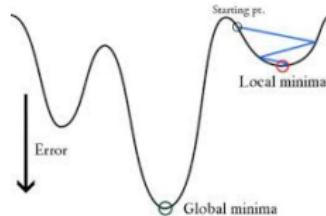
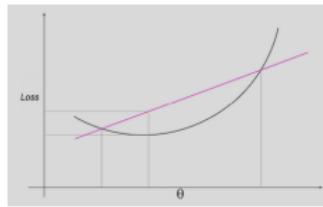
where  $x_{ij}$  is the observed value of the feature  $x_j$ ,  $y_i$  is the predicted value of the outcome and  $\theta_i$  are constants the values of which need to be determined. For now, we limit the the values of the features  $x_{ij}$  to numbers, positive or negative. Later on, we will study techniques on how to deal with the situation where the feature value is a string.

$$\text{Squared Error Loss (L)} = \frac{1}{2} \times \sum_{i=1}^r (\hat{y}_i - y_i)^2 \quad (2)$$

This loss function is always positive, there is a minimum, is monotonically increasing from that minimum value in both positive and negative directions. Such a function is called a **convex function**.

- ③ We need to minimize some loss function over the training data.

# Gradient Descent



- ① Minimize by setting the partial derivative of the loss wrt  $\theta_j$  to zero.

$$\frac{\partial L}{\partial \theta_0} = \sum_{i=1}^r (\hat{y}_i - y_i) = 0 \quad \frac{\partial L}{\partial \theta_j} = \sum_{i=1}^r (\hat{y}_i - y_i) x_{ij} = 0 \quad (3)$$

- ② Randomly assigning values to  $\theta_j$ . For a convex function it does not matter what the initial weights are as it will always converge.  $\hat{y}$  and  $x_{ij}$  are observed and  $y_i$  is computed. This means the slope of the loss function can be readily computed.

# Logistic Regression

In many cases the value of  $y_i$  need to be bounded between 0 and 1. For logistic regression, Least Squared Error will result in a *non-convex* graph with local minimums and hence is not feasible. In such cases, we use the **logistic regression model** as given below:

$$y_i = \frac{1}{1 + e^{-z}} \quad (4)$$

$$z_i = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (5)$$

- for  $z$  is large +ve number,  $\frac{1}{e^z} = 0$ ;  $y_i = 1$
- for  $z = 0$ ,  $y_i = 0.5$
- for  $z$  is large -ve number,  $y_i = 0$

Hence, the value of  $y_i$  is bounded between 0 and 1 for  $z$  between  $-\infty$  and  $\infty$ .

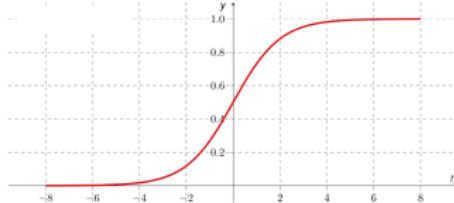
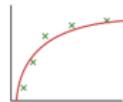
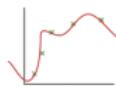


Figure: Sigmoid Curve

# Predictions & Errors



(a) Correct Fit



(b) Over fit



(c) Under fit

Predictions from the model will have differences or errors. **Overfitting** will occur when an excessive number of features are used than required.

**Underfitting** occurs when an insufficient number of features are used than required.

- ① **Bias** is the difference between average model prediction and the true target value and **variance** is the variation in predictions across different training data samples. Simple models with small number of features have high bias and low variance whereas complex models with large number of features have low bias and high variance.
- ② **Regularization** is a technique used to avoid problem of overfitting. It prevents overfitting in linear models by a penalty term that penalizes large weight values.

# Information & Uncertainty

- ① In Claude Shannon's (MIT) information theory, one bit of information reduces the uncertainty by 2. Similarly, if 3 bits of information are sent, then the reduction in uncertainty by  $2^3$ , i.e., 8. This is intuitive. With 3 bits, there could be 8 possible values and so if a particular set of bits are transmitted, 8 possibilities are eliminated with 1 certainty.
- ② **Information Content** When the information is probabilistic, the self-information  $I_x$ , or Information Content of *measuring a random variable X as outcome x is defined as:*

$$I_x = \log \left( \frac{1}{p(x)} \right) = -\log(p(x)) \quad (6)$$

where  $p(x)$  is probability mass function.

- ③ **Shannon Entropy of the random variable X is defined as:**

$$H(X) = \sum_x -p(x)\log(p(x)) = E(I_x) \quad (7)$$

It is the *expected information content* of the measurement of  $X$ .

# Cross Entropy - Kullback–Leibler (KL) divergence

- ① Cross-Entropy is defined as:

$$H(p, q) = - \sum_i p(i) \log_2 q(i) \quad (8)$$

where  $p$  is the true distribution and  $q$  is the predicted distribution. If the predictions are perfect, then the cross-entropy is same as the entropy. If the prediction differs, then there is a divergence which is known as *Kullback – Leibler (KL) divergence*. Hence,

$$\text{Cross Entropy} = \text{Entropy} + \text{KL Divergence}$$

$$\text{KL Divergence} = H(p, q) - H(p)$$

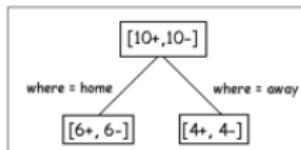
Hence, if the predicted distribution is closer to true distribution when KL divergence is low.

# Decision Tree

- ① Decision Tree is a supervised machine learning algorithm where You try to separate your data and group the samples together in the classes they belong to. You maximize the purity of the groups as much as possible each time you create a new node of the tree.
- ② At each step, each branching, you want to decrease the entropy, so this quantity is computed before the cut and after the cut. If it decreases, the split is validated and we can proceed to the next step, otherwise, we must try to split with another feature or stop this branch.
- ③ XGBoost is a machine learning algorithm that belongs to the ensemble learning category, specifically the gradient boosting framework. It utilizes decision trees as base learners and employs regularization techniques to enhance model generalization.

## Example

Following is a table for win/loss of soccer games at home and away.

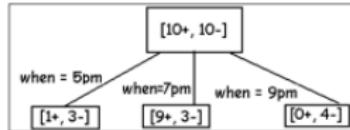


The entropy is:

$$H\left(\frac{6}{12}, \frac{6}{12}\right) = -\frac{6}{12} \log_2\left(\frac{6}{12}\right) - \frac{6}{12} \log_2\left(\frac{6}{12}\right) = 0.69$$

$$H\left(\frac{4}{8}, \frac{4}{8}\right) = -\frac{4}{8} \log_2\left(\frac{4}{8}\right) - \frac{4}{8} \log_2\left(\frac{4}{8}\right) = 0.69$$

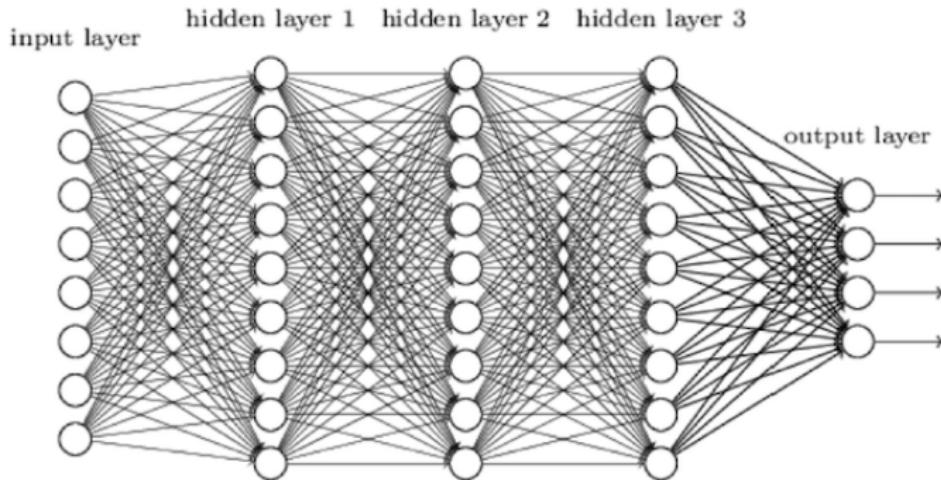
$$H = -\frac{12}{20} \times H\left(\frac{6}{12}, \frac{6}{12}\right) - \frac{8}{20} H\left(\frac{4}{8}, \frac{4}{8}\right) = 0.69$$



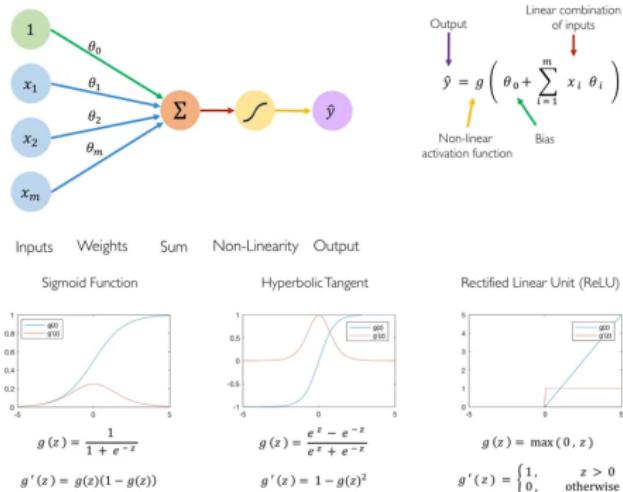
Information gain is  $0.69 - 0.45 = 0.24$ .

# Deep Neural Net

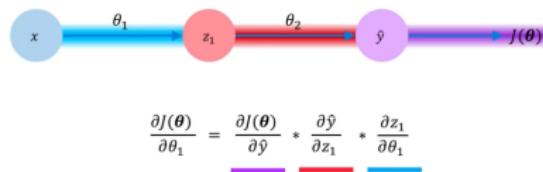
- ① Deep Learning Success Stories - Image Recognition, Speech Comprehension, Chatbot.
- ② DNNs are suitable where the raw underlying features are not individually interpretable. This success is attributed to their ability to learn hierarchical representations, unlike traditional methods that rely upon hand-engineered features.



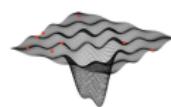
# Deep Neural Net



## Back Propagation



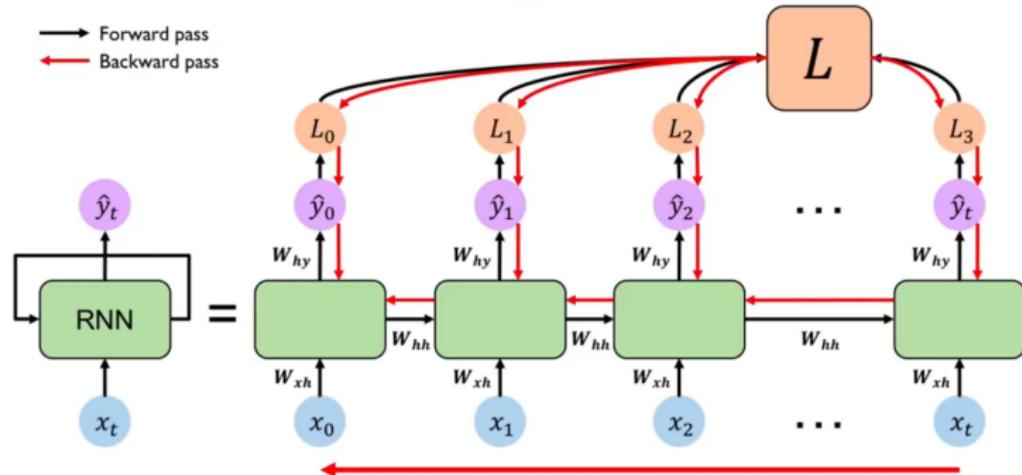
Repeat for every weight in the network using gradients from later layers.



(a) Global versus Local Minima

# RNN

RNNs are connectionist models with the ability to selectively pass information across sequence steps while processing sequential data one element at a time.

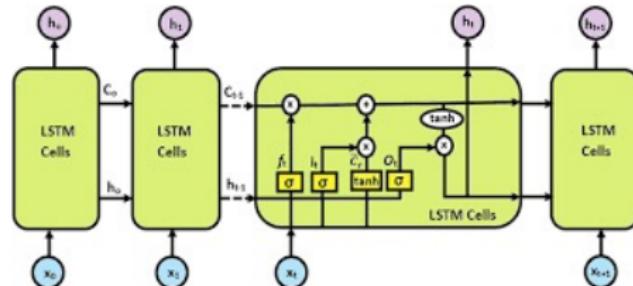


But problems of vanishing and exploding gradients occur when backpropagating errors across many time steps.

# Long Short Term Memory - Hochreiter & Schmidhuber

Capable of handling the vanishing gradient problem faced by RNN.

- ① Maintain a separate cell state. The cell state is the horizontal line running through the top of the LSTM unit. It acts as a conveyor belt, allowing information to flow unchanged across many time steps.
- ② 3 gates.  
**Forget Gate:** Determines which information from the previous cell state should be forgotten.  
**Input Gate:** Controls the update of the cell state by selectively adding new information to it.  
**Output Gate:** Decides what information should be output from the cell state based on the current input and the previous cell state.  
Backpropagation does not require matrix multiplication



# Reinforcement Learning

- ① A type of machine learning paradigm where an agent learns to make decisions by interacting with an environment to achieve a specific goal.
- ② Reinforcement learning relies on a system of rewards and punishments to guide the learning process.

# Generalised Pretrained Transformer (GPT) - 2014, 2017

---

## 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

**Lukasz Kaiser\***

Google Brain

lukaszkaiser@google.com

**Illia Polosukhin\* ‡**

illia.polosukhin@gmail.com

"Neural Machine Translation by Jointly Learning to Align and Translate"  
by Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 2014.  
Introduced concept of attention mechanism and laid the foundation for  
subsequent developments in NLP and DL, including the transformer  
architecture introduced in "Attention Is All You Need."

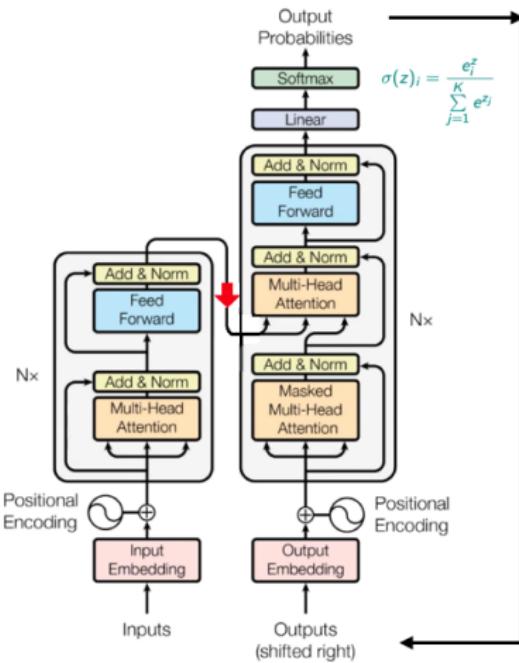
# Attention

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

- ① Attention mechanism has an infinite reference window
- ② In contrast, Recurring Neural Network (RNN) has a short reference window, Long Short Term Memory (LSTM) has a longer window. RNN does not work, LSTM has limited capability.

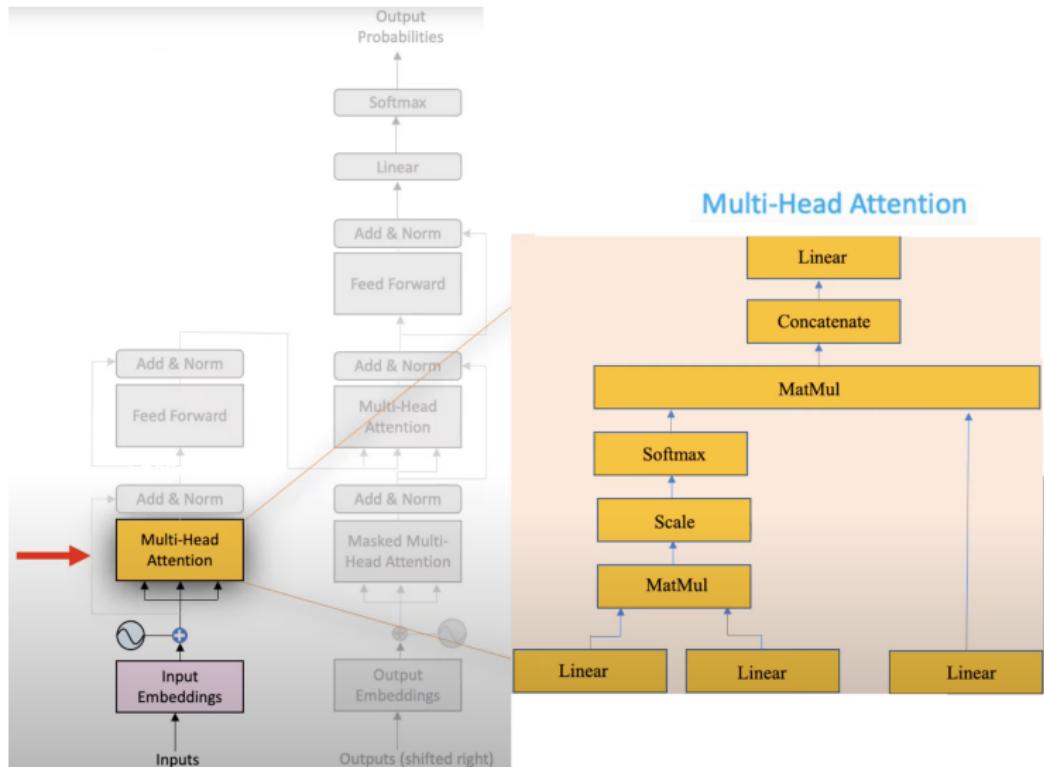
# Key Concepts

- ① Encode, Positional encoding
- ② Key, Query and Value → Attention Filter
- ③ Multi Head Attention
- ④ Information preservation, normalisation
- ⑤ Decode, Masked Attention
- ⑥ Potential Innovations
- ⑦ Conclusion

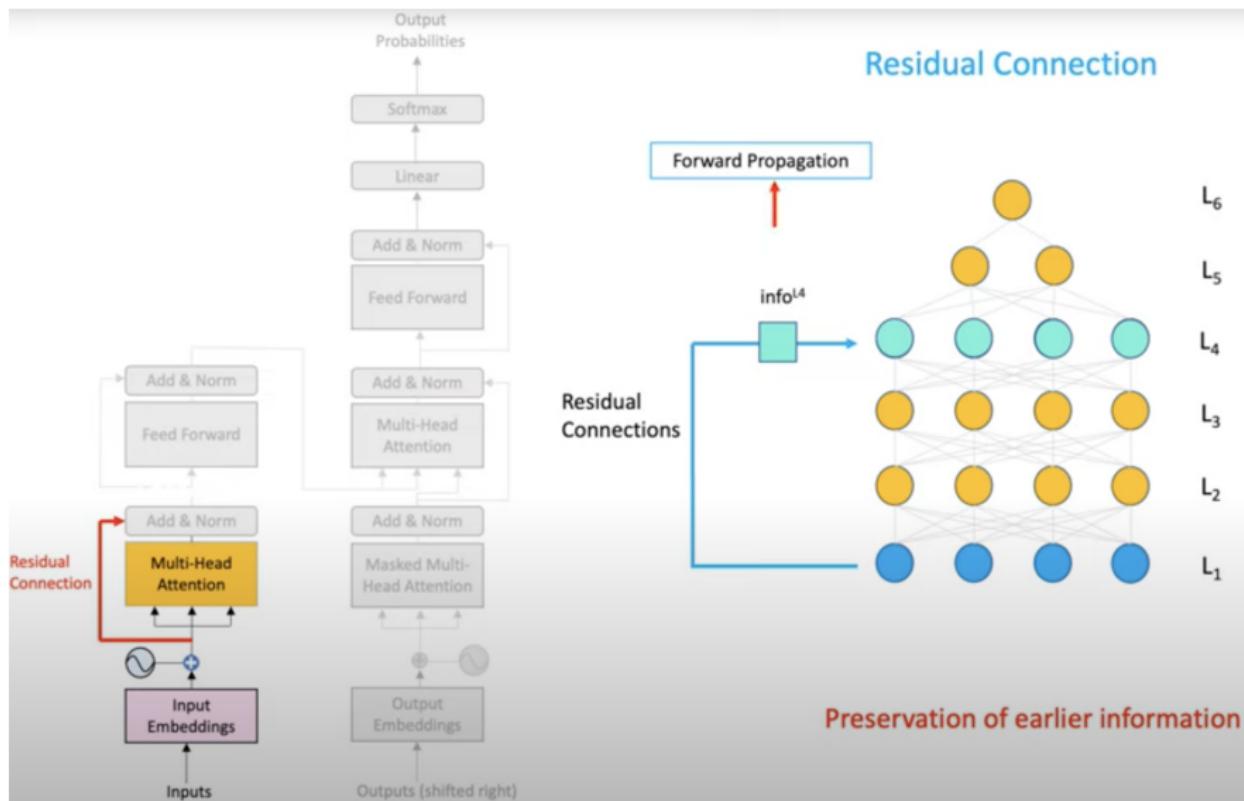


Decoders are autoregressive models;  
They are trained to predict the next token  
after reading the preceding ones

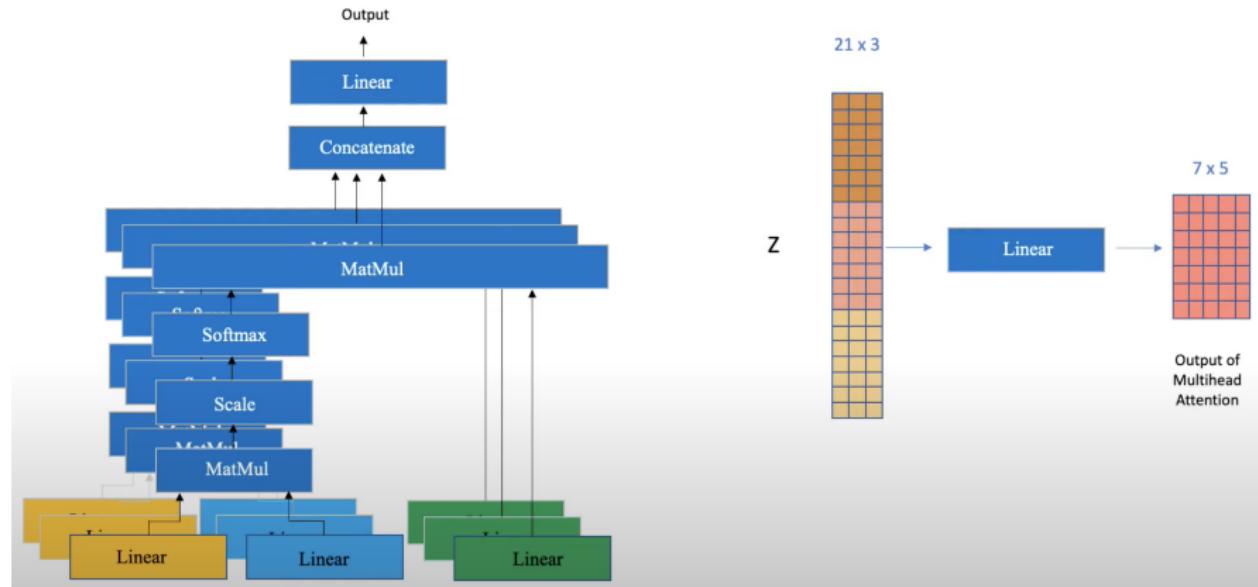
# Multi Head Attention



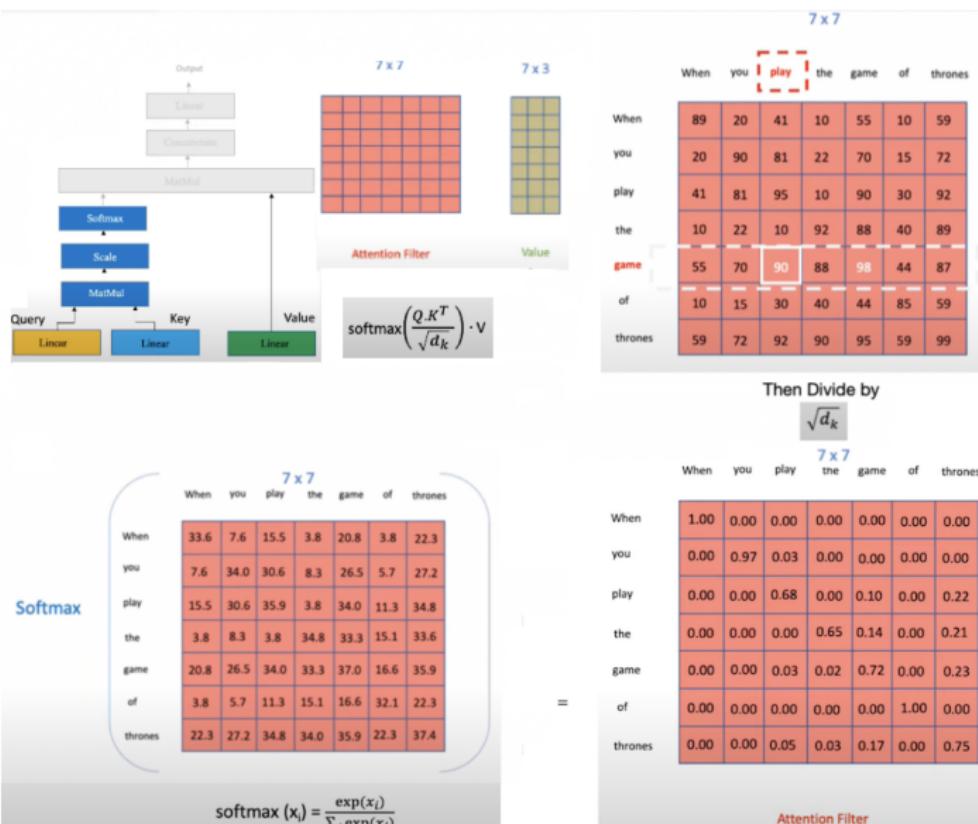
# Information Preservation



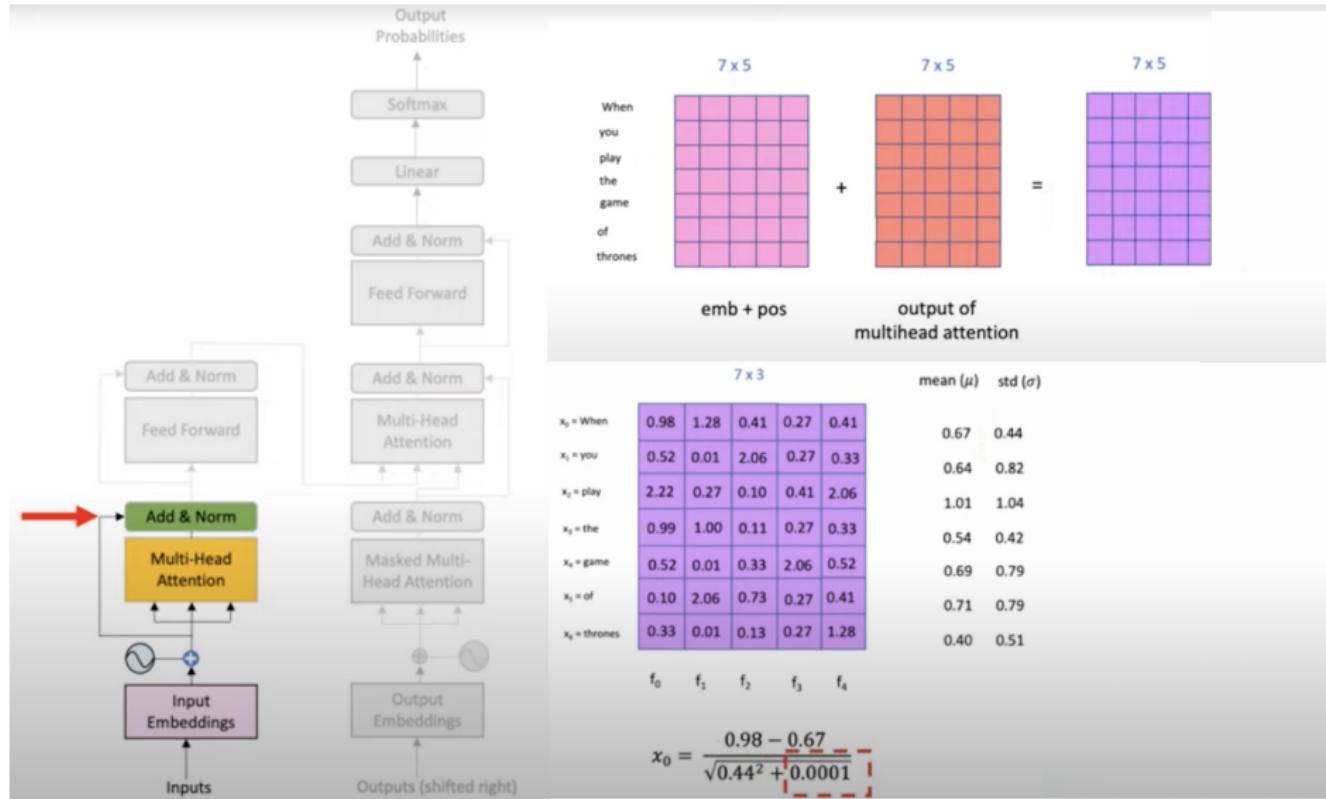
# Multi Head Attention - Concatenation



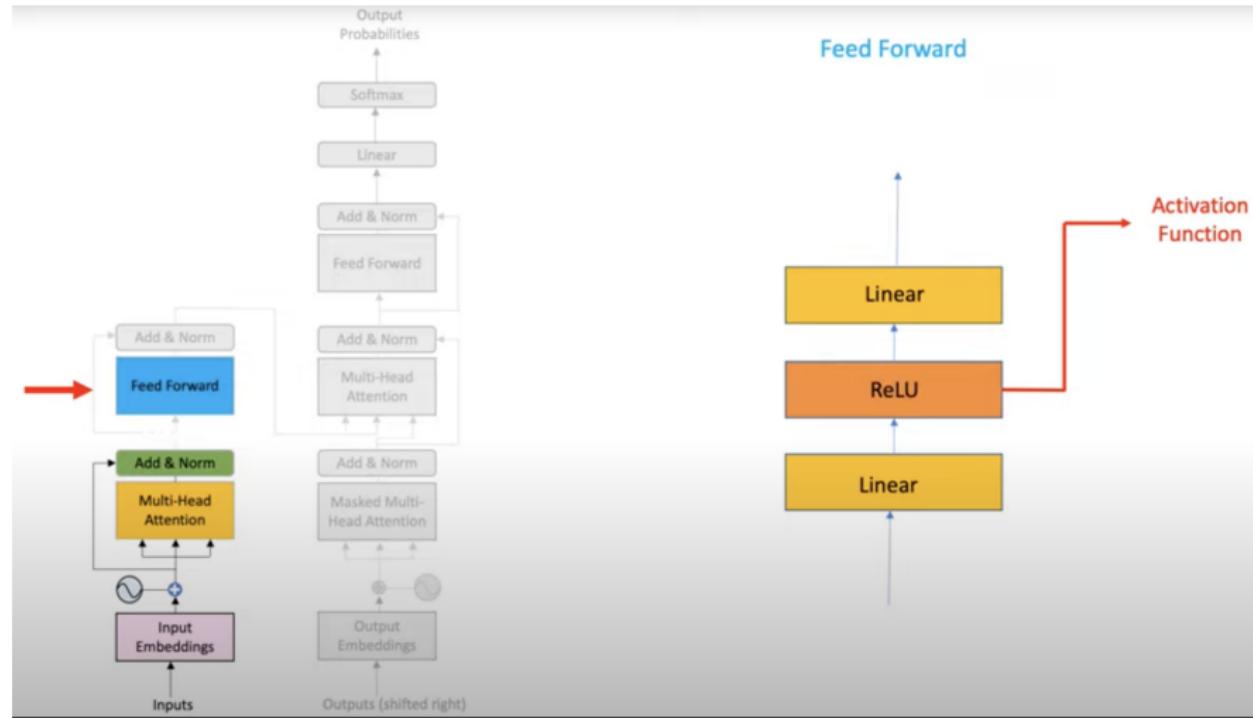
# Attention Filter - filter out unnecessary information (noise)



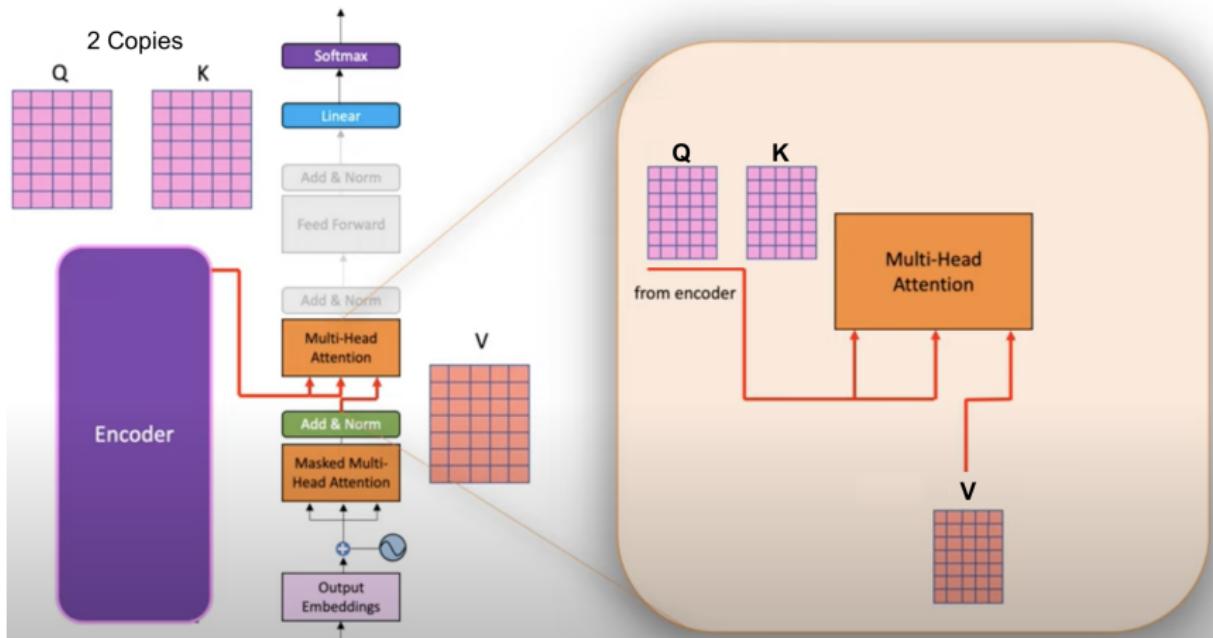
# Add (Preserve Information) & Normalisation



# Feed Forward



# Decoder Layer



# Masking

	<start>	I	am	no	man	<end>
<start>	33.6	7.6	15.5	3.8	20.8	22.3
I	7.6	34.0	30.6	8.3	26.5	27.2
am	15.5	30.6	35.9	3.8	34.0	34.8
no	3.8	8.3	3.8	34.8	33.3	33.6
man	20.8	26.5	34.0	33.3	37.0	35.9
<end>	3.8	5.7	11.3	15.1	16.6	37.4

Attention Filter

<start> I am no man <end>

+

	<start>	I	am	no	man	<end>
<start>	0	-inf	-inf	-inf	-inf	-inf
I	0	0	-inf	-inf	-inf	-inf
am	0	0	0	-inf	-inf	-inf
no	0	0	0	0	-inf	-inf
man	0	0	0	0	0	-inf
<end>	0	0	0	0	0	0

Mask Filter

<start> I am no man <end>

	<start>	I	am	no	man	<end>
<start>	1	0	0	0	0	0
I	0.01	0.99	0	0	0	0
am	0.001	0.004	0.995	0	0	0
no	0.003	0.004	0.003	0.99	0	0
man	0.003	0.003	0.04	0.02	0.93	0
<end>	0.001	0.001	0.001	0.001	0.001	0.995

Masked-Attention Filter

## LLM - Summary

- ① GPT does not replicate human writing or speaking processes. Although they imitate human writing, any apparent cleverness primarily arises from our inclination to attribute human characteristics to non-human entities (anthropomorphization).
- ② LLMs are essentially establishing statistical connections among vectors representing words and more extended grammatical structures. Each word within a sentence is linked to the subsequent word in the sequence with an associated probability.
- ③ This diverges significantly from human cognitive processes, where we employ word meanings to construct intricate and precise structures of "meanings" and definitions. For LLMs, definitions are confined to statistical interrelations among intricate vectors encompassing words, sentences, and more extensive grammatical constructs.
- ④ LLMs cannot innovate, not just yet. But the results are often stunning! LLMs do have massive use cases.

## LLM pitfalls?

- ① LLMs can do jaw-dropping things. But nobody knows exactly why.
- ② Figuring it out is one of the biggest scientific puzzles of our time and a crucial step towards controlling more powerful future models.
- ③ Burda and Edwards teamed up with colleagues at OpenAI to study the phenomenon. They found that in certain cases, models could seemingly fail to learn a task and then all of a sudden just get it, as if a lightbulb had switched on. This wasn't how deep learning was supposed to work. They called the behavior "grokking".
- ④ According to classical statistics, the bigger a model gets, the more prone it is to overfitting.
- ⑤ Double descent - performance of the model initially decreases, then increases, and finally decreases again as the model complexity or dataset size increases. Contradicts the traditional bias-variance tradeoff.
- ⑥ This behavior, dubbed benign overfitting, is not fully understood. Raises basic questions about how models should be trained.

# LangChain - Apps powered by LLMs

- ① The basic building block of LangChain is the LLM, which takes in text and generates more text. Lots of LLM providers ... OpenAI, Anthropic, Cohere, Hugging Face, Llama-cpp, etc.
- ② LLM applications do not pass user input directly into an LLM. Usually they will add the user input to a larger piece of text, called a prompt template, that provides additional context on the specific task at hand. LangChain facilitates prompt management and optimization.
- ③ Combine LLMs and Prompts in multi-step workflows.
- ④ Combine LLM with your own text data. Large number of document loaders available - JSON, AWS S3, Azure Blob Storage File, Confluence, CSV, Email, PDF, Excel, MS Word/Excel, G Drive, ...
- ⑤ Agents involve an LLM making decisions about which Actions to take, taking that action, seeing an Observation, and repeating that until done. Zero-shot means the agent functions on the current action only - it has no memory. It uses the ReAct framework to decide which tool to use, based solely on the tool's description.

# Questions?

Thank You!