

COMP47650 – DEEP LEARNING

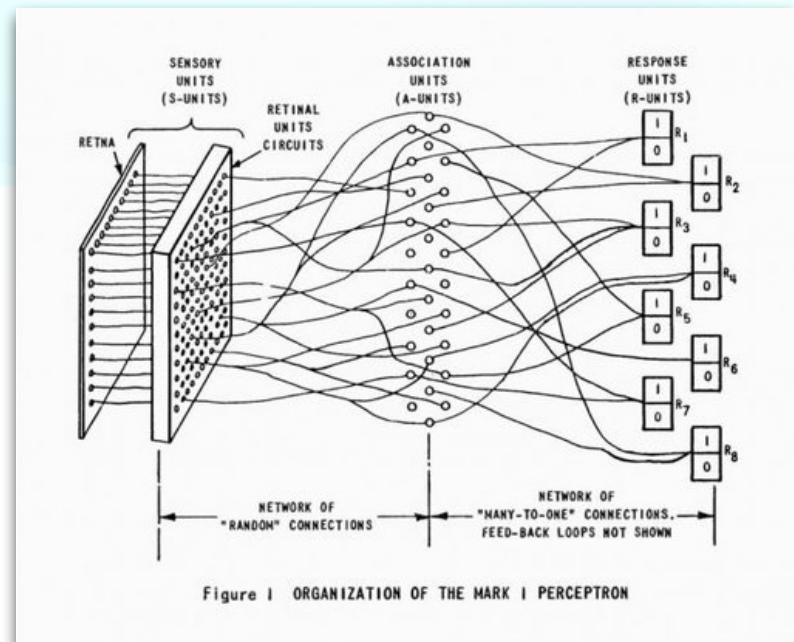
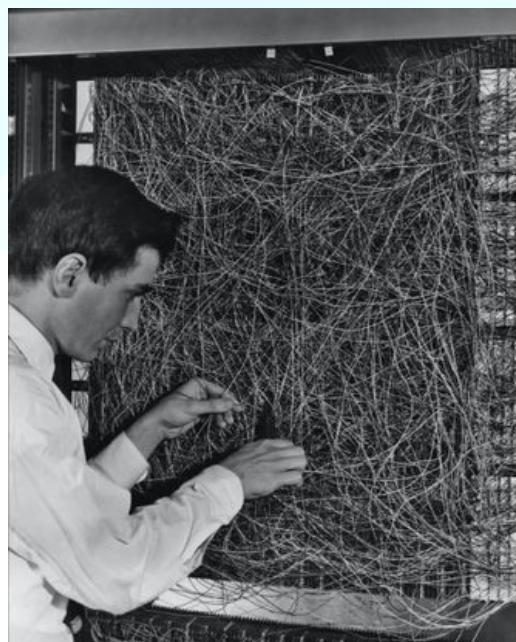
Quick Recap

SLIDE DECK 0

guenole.silvestre@ucd.ie
School of Computer Science
University College Dublin

January 2025

1957 – Perceptron (Rosenblatt)



1966 – ELIZA (Weizenbaum)

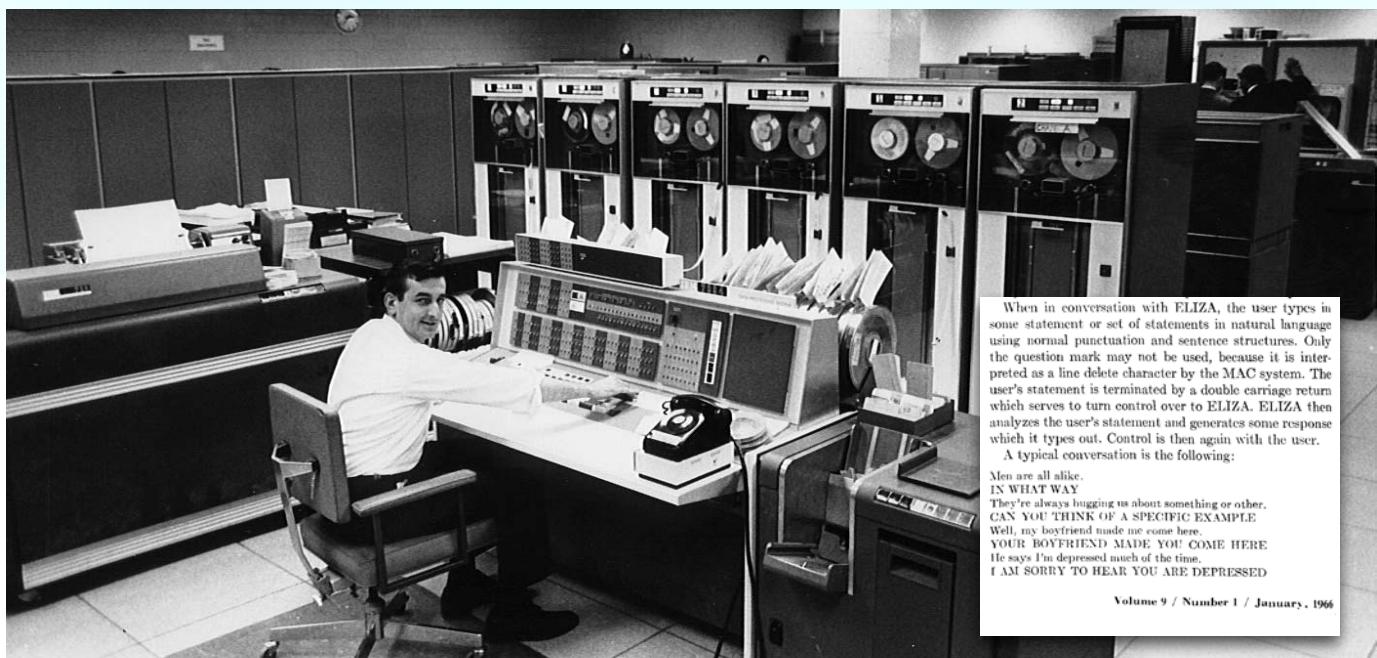


ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.*

ELIZA is a program operating within the MAC time-sharing system at MIT which makes certain kinds of natural language conversation between man and computer possible. Input sentences are analyzed on the basis of decomposition rules which are triggered by key words appearing in the input text. Responses are generated by reassembly rules associated with selected decomposition rules. The fundamental technical problems with which ELIZA is concerned are: (1) the identification of key words, (2) the discovery of minimal context, (3) the choice of appropriate transformations, (4) generation of responses in the absence of key words, and (5) the provision of an editing capability for ELIZA "scripts". A discussion of some psychological issues relevant to the ELIZA approach as well as of future developments concludes the paper.

IBM 7604 – Early Sixties



When in conversation with ELIZA, the user types in some statement or set of statements in natural language using normal punctuation and sentence structures. Only the question mark may not be used, because it is interpreted as a line delete character by the MAC system. The user's statement is terminated by a double carriage return which serves to turn control over to ELIZA. ELIZA then analyzes the user's statement and generates some response which it types out. Control is then again with the user.

A typical conversation is the following:

Men are all alike.
IN WHAT WAY
They're always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well, my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says I'm depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED

Volume 9 / Number 1 / January, 1966

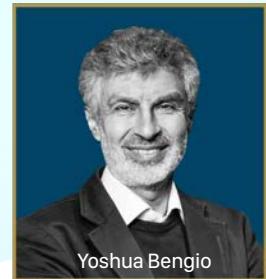
THE ACM TURING AWARD 2018



Yann LeCun



Geoffrey Hinton



Yoshua Bengio

THE NOBEL PRIZE IN PHYSICS 2024



John J. Hopfield

Geoffrey E. Hinton

THE NOBEL PRIZE IN CHEMISTRY 2024



David Baker

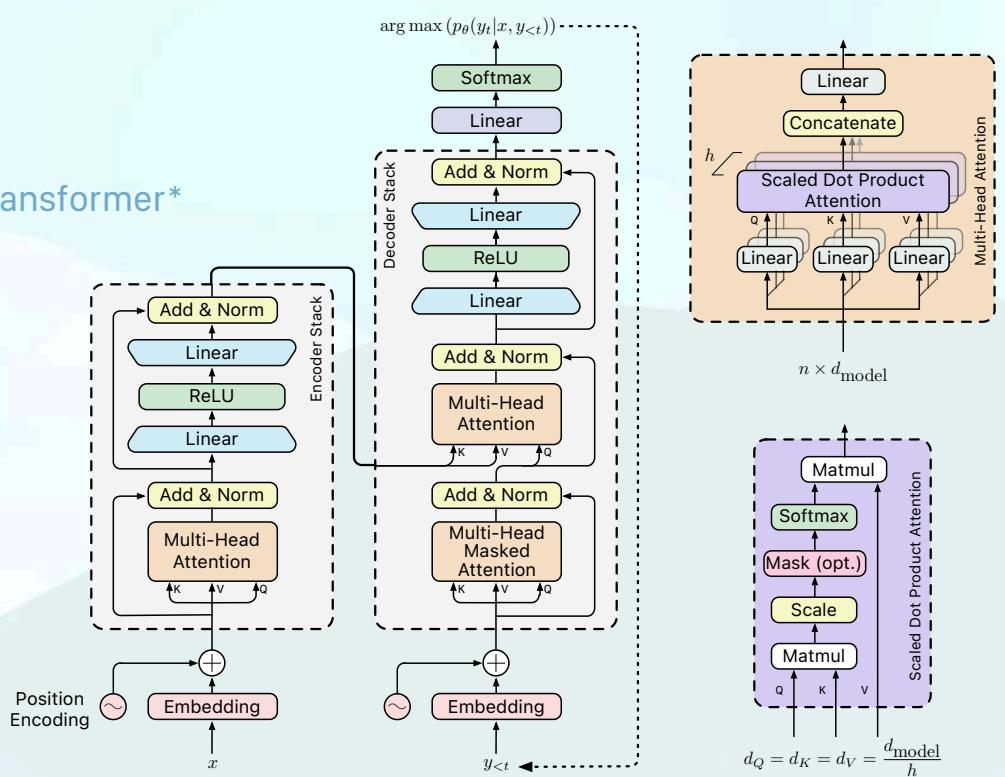
Demis Hassabis

John M. Jumper

Illustrations: Niklas Elmehed

2017

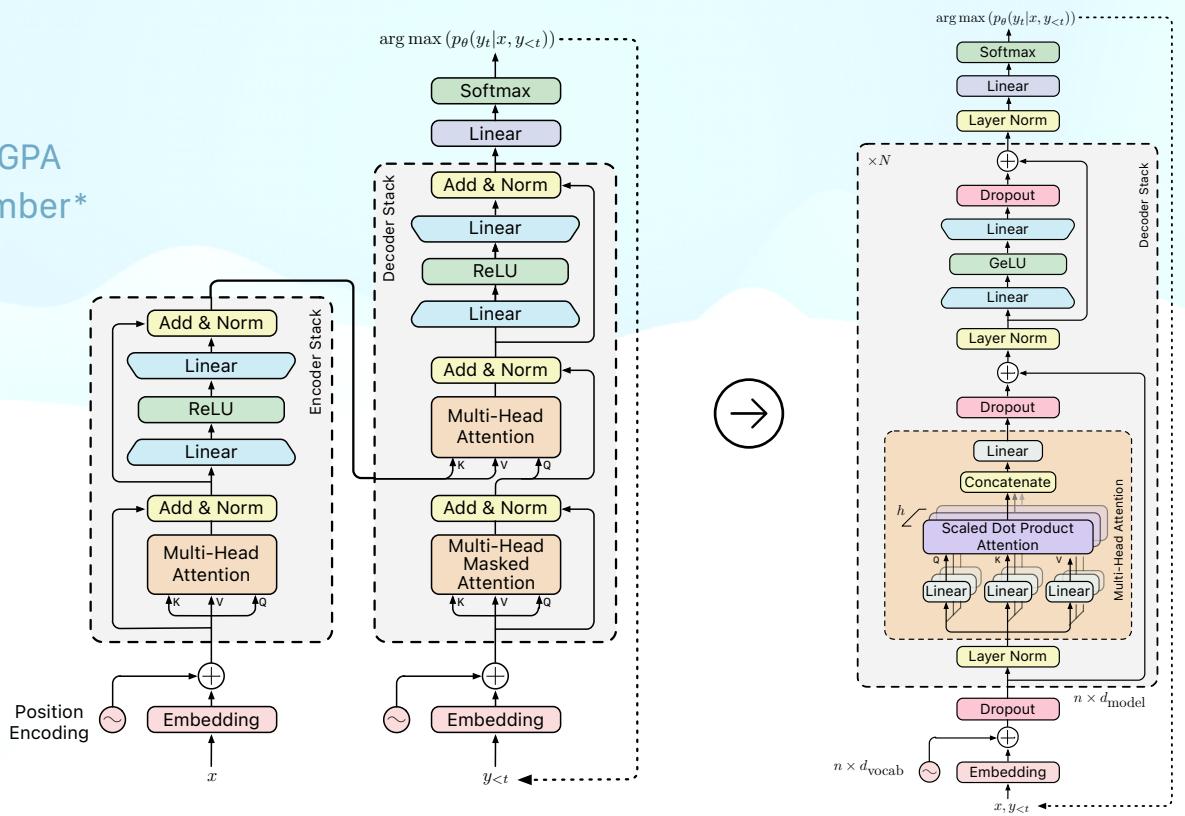
A year to remember
Google proposed The Transformer*



*Attention is All You Need, Vaswani et al (NeurIPS 2017)

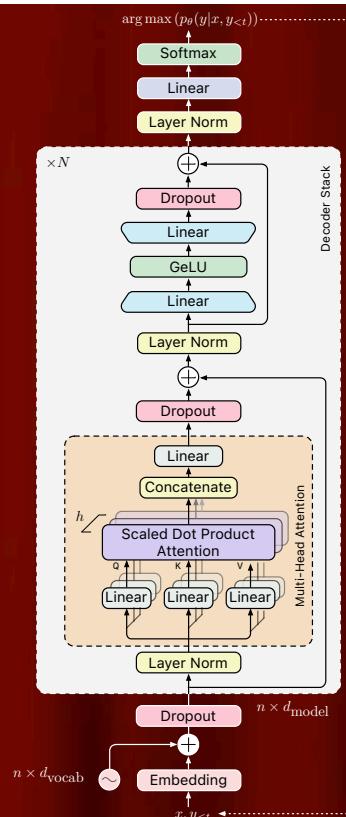
2023

A year your GPA
must remember*



*Cheat GPT (openAI, 2022-23)

Let's have a chat
with GPT



LET'S CODE THIS IN 5...

What just happened?

```
# STEP 1: Assume PyTorch as the deep learning framework. No need for explanation nor
justification. Implement the scaled dot product attention.

import torch
import torch.nn as nn
import torch.nn.functional as F

def scaled_dot_product_attention(query, key, value, mask=None, dropout=None):
    d_k = query.size(-1)
    scores = torch.matmul(query, key.transpose(-2, -1)) / torch.sqrt(torch.tensor(d_k, dtype=torch.float))

    if mask is not None:
        scores = scores.masked_fill(mask == 0, float('-inf'))

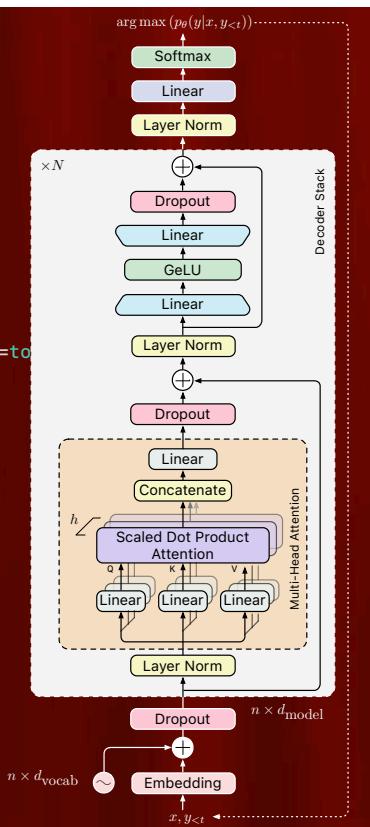
    attention_weights = F.softmax(scores, dim=-1)

    if dropout is not None:
        attention_weights = dropout(attention_weights)

    output = torch.matmul(attention_weights, value)
    return output, attention_weights

# STEP 2: Implement now the multi-head attention using this function with d_model
and h as input. Split input over the h heads equally.

class MultiHeadAttention(nn.Module):
    def __init__(self, d_model, h):
        super(MultiHeadAttention, self).__init__()
        self.d_model = d_model
        self.h = h
```





That's all Folks!