



賴冠廷教授  
Prof. Kuan-Ting Lai

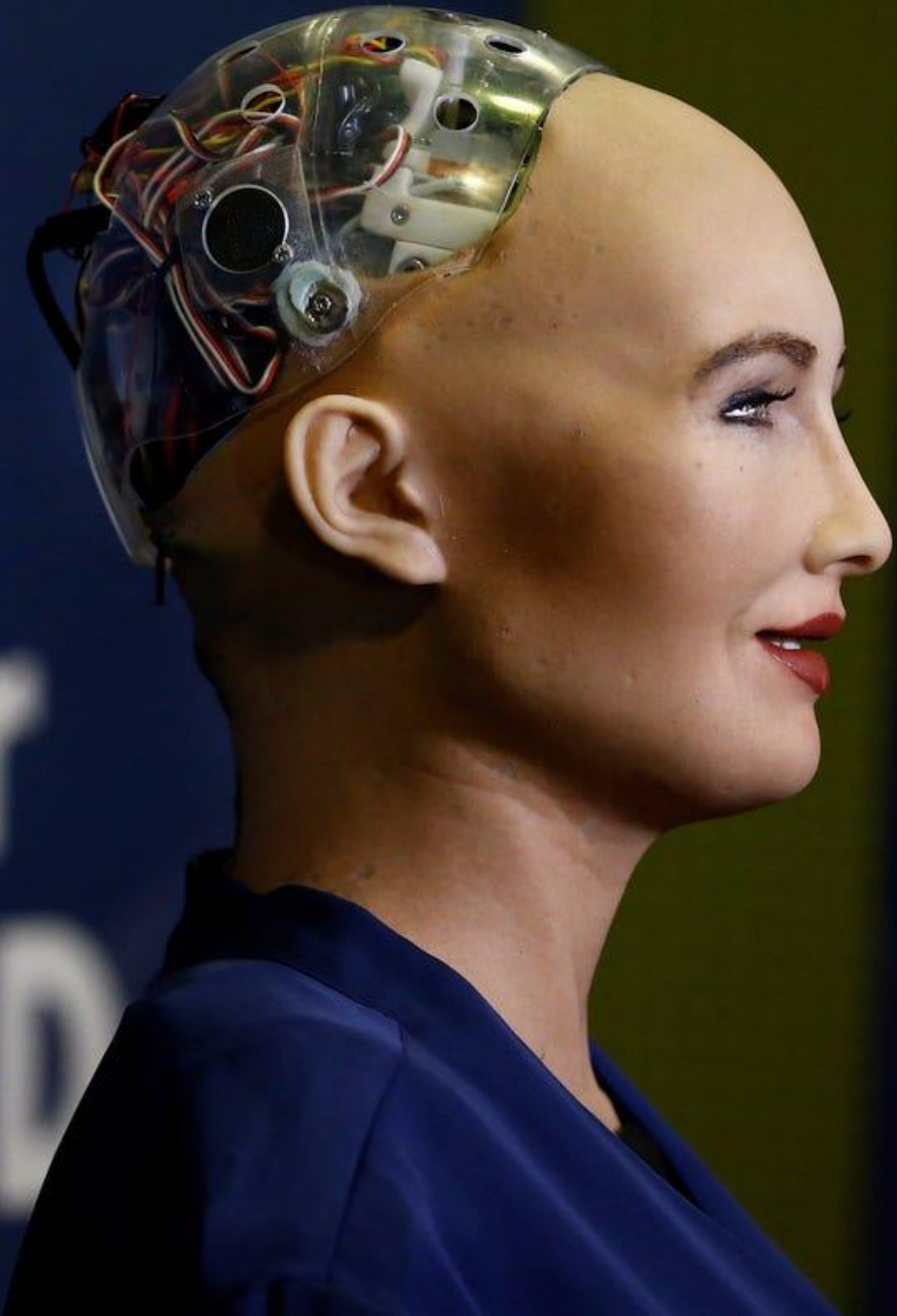
台北科技大學電子工程系

2020/3/20

# Artificial Intelligence

# Artificial Intelligence

AI for  
GOOD



# 1 The accelerating pace of change ...



# 2 ...and exponential growth in computing power ...

Computer technology, shown here climbing dramatically by powers of 10, is now progressing more each hour than it did in its entire first 90 years

## COMPUTER RANKINGS

By calculations per second per \$1,000



**Analytical engine**  
Never fully built, Charles Babbage's invention was designed to solve computational and logical problems



**Colossus**  
The electronic computer, with 1,500 vacuum tubes, helped the British crack German codes during WW II



**UNIVAC I**  
The first commercially marketed computer, used to tabulate the U.S. Census, occupied 943 cu. ft.

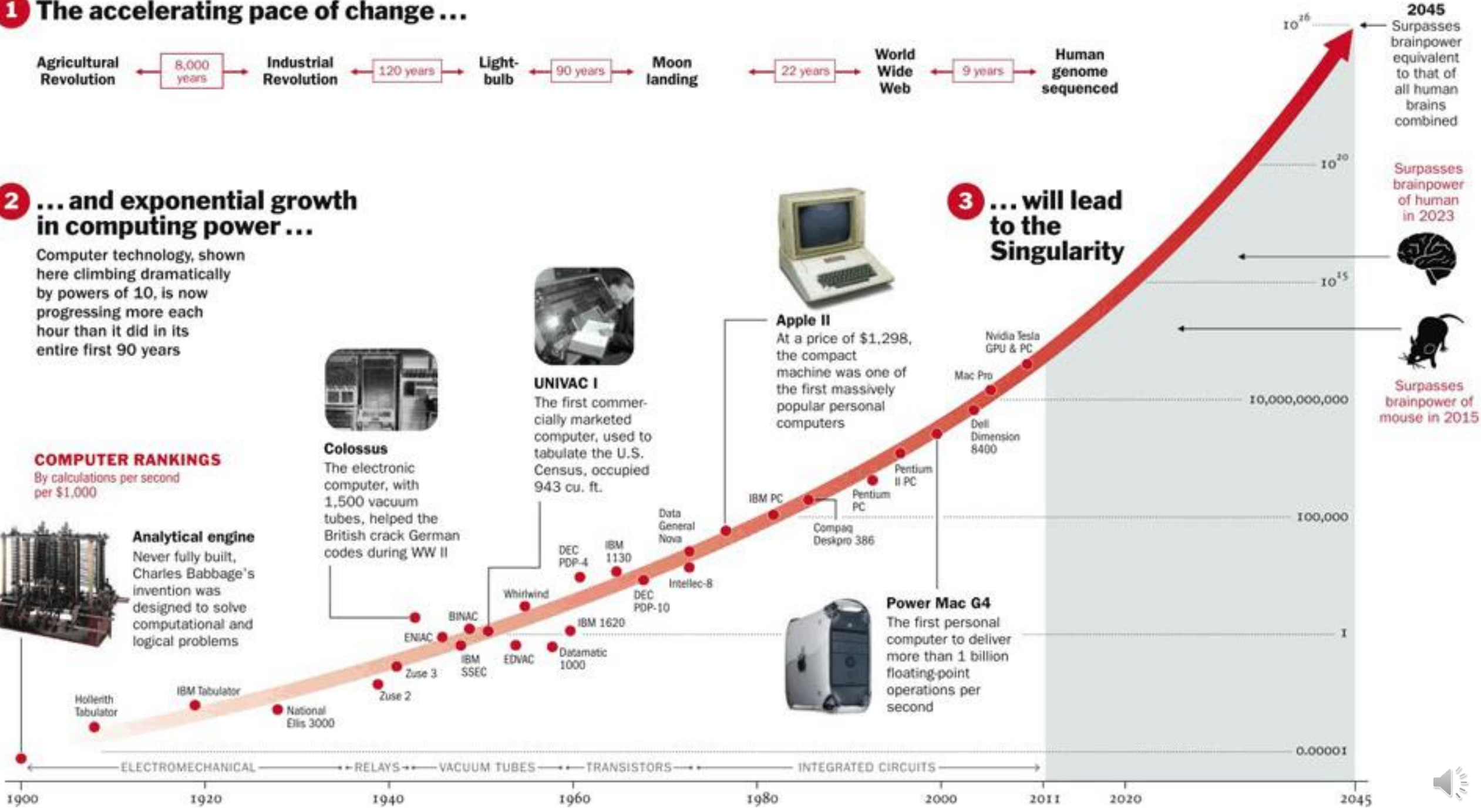


**Apple II**  
At a price of \$1,298, the compact machine was one of the first massively popular personal computers

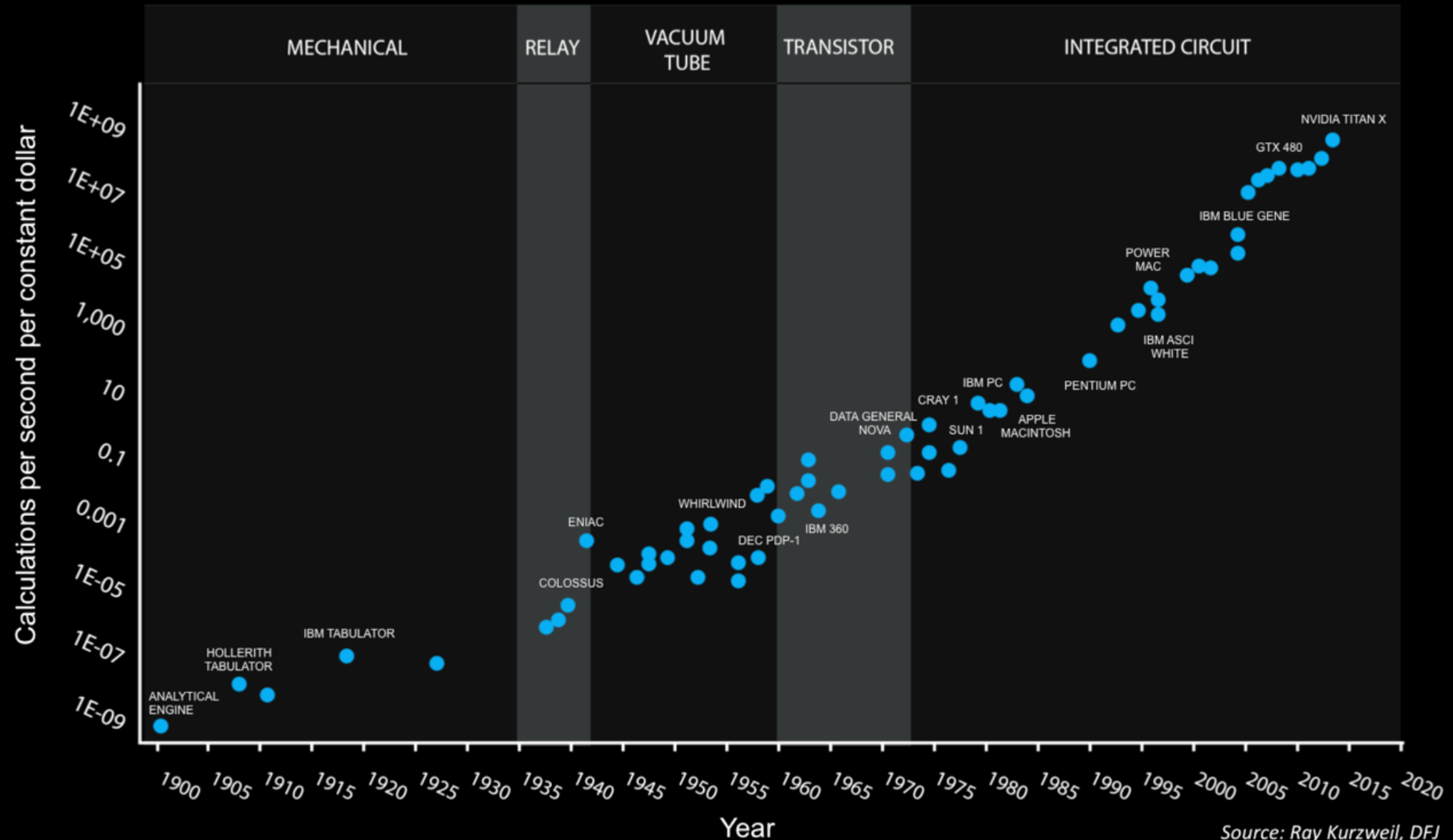


**Power Mac G4**  
The first personal computer to deliver more than 1 billion floating-point operations per second

# 3 ... will lead to the Singularity



# 120 Years of Moore's Law

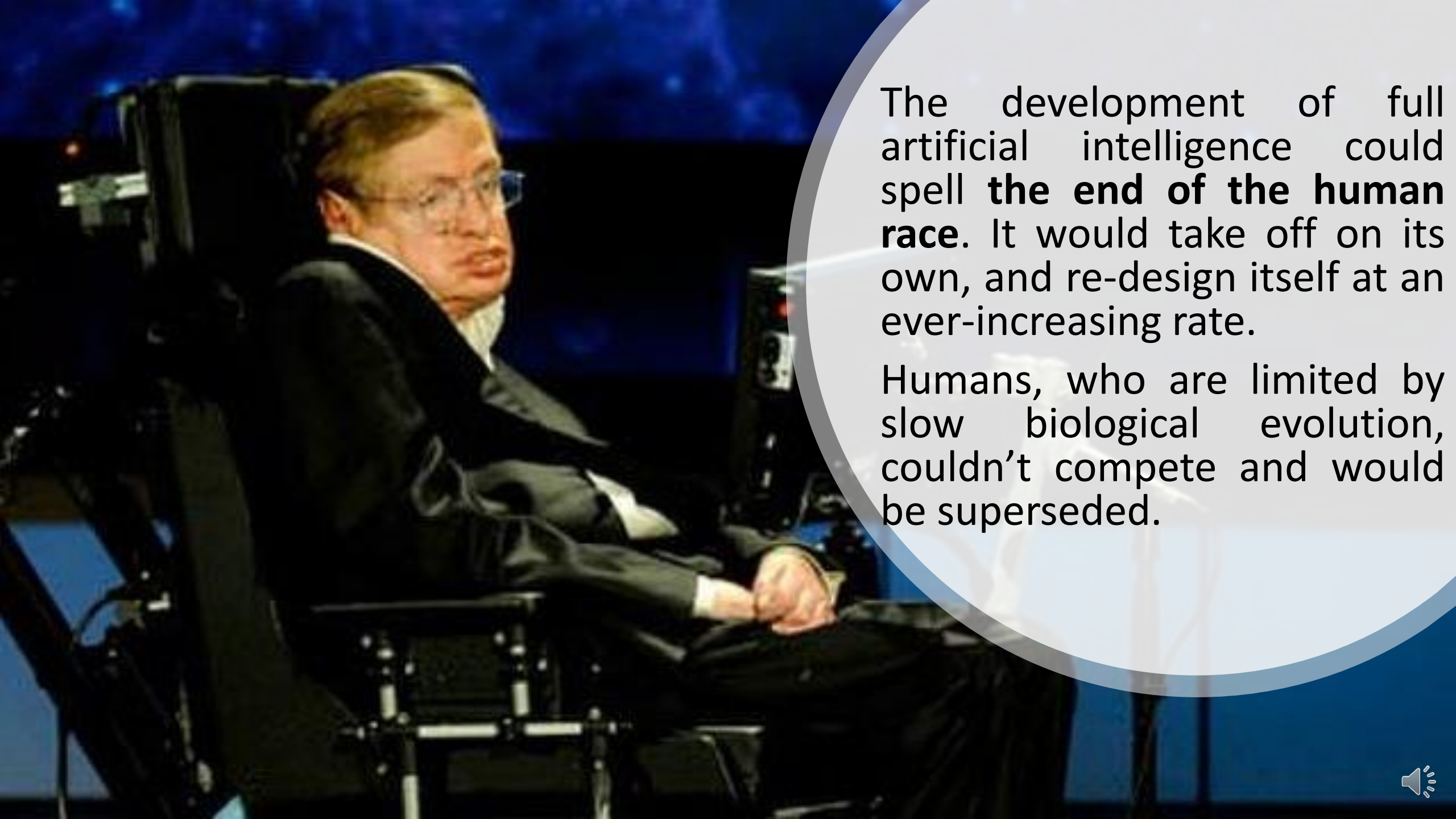




# The Singularity

I. J. Good





The development of full artificial intelligence could spell **the end of the human race**. It would take off on its own, and re-design itself at an ever-increasing rate.

Humans, who are limited by slow biological evolution, couldn't compete and would be superseded.



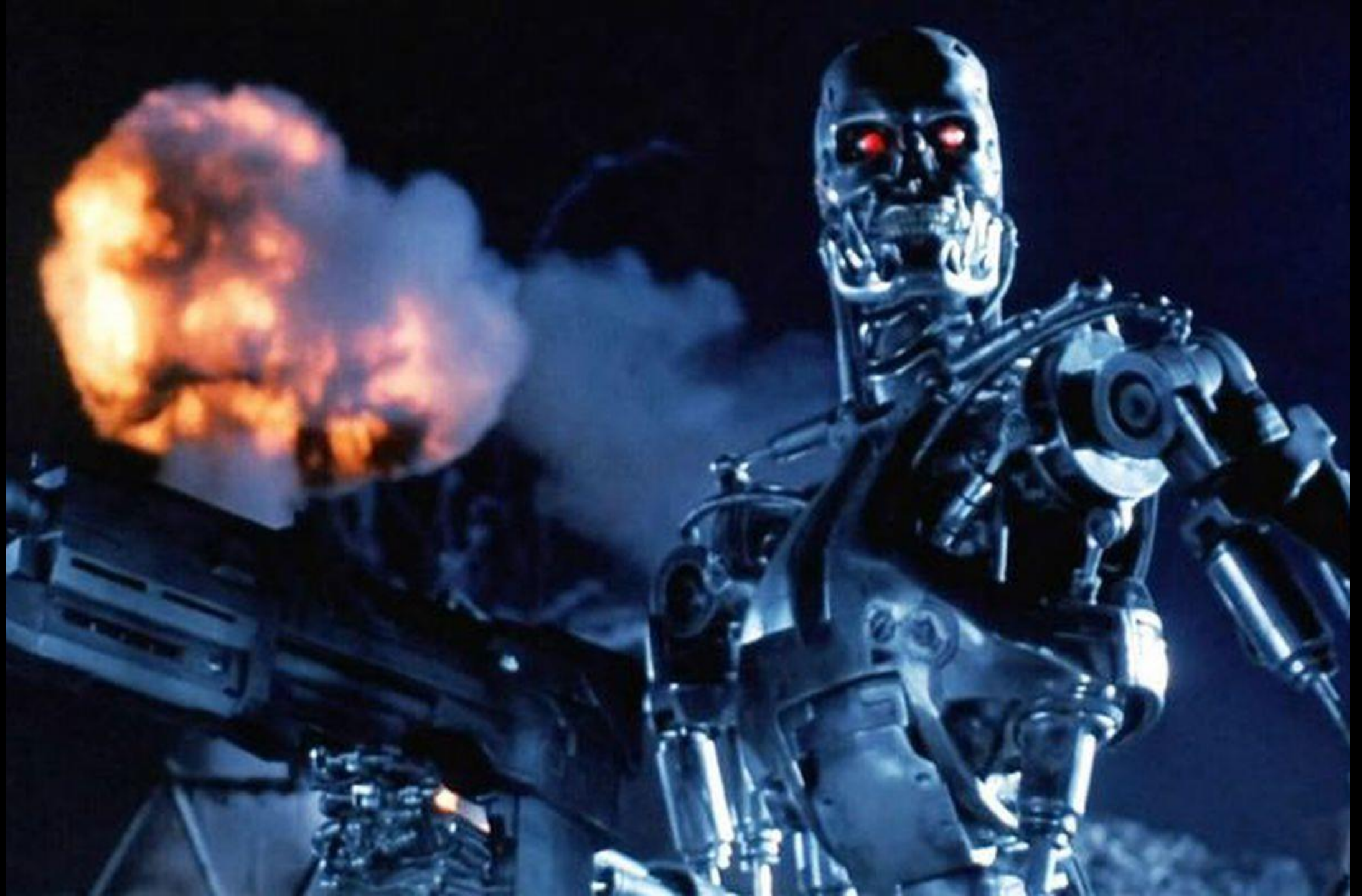


- Robots will do everything better than us
- AI is a greater risk than North Korea
- AI is a fundamental risk to the existence of human civilization



AI is —  
the last invention we'll  
ever make,  
the last challenge we'll  
ever face!



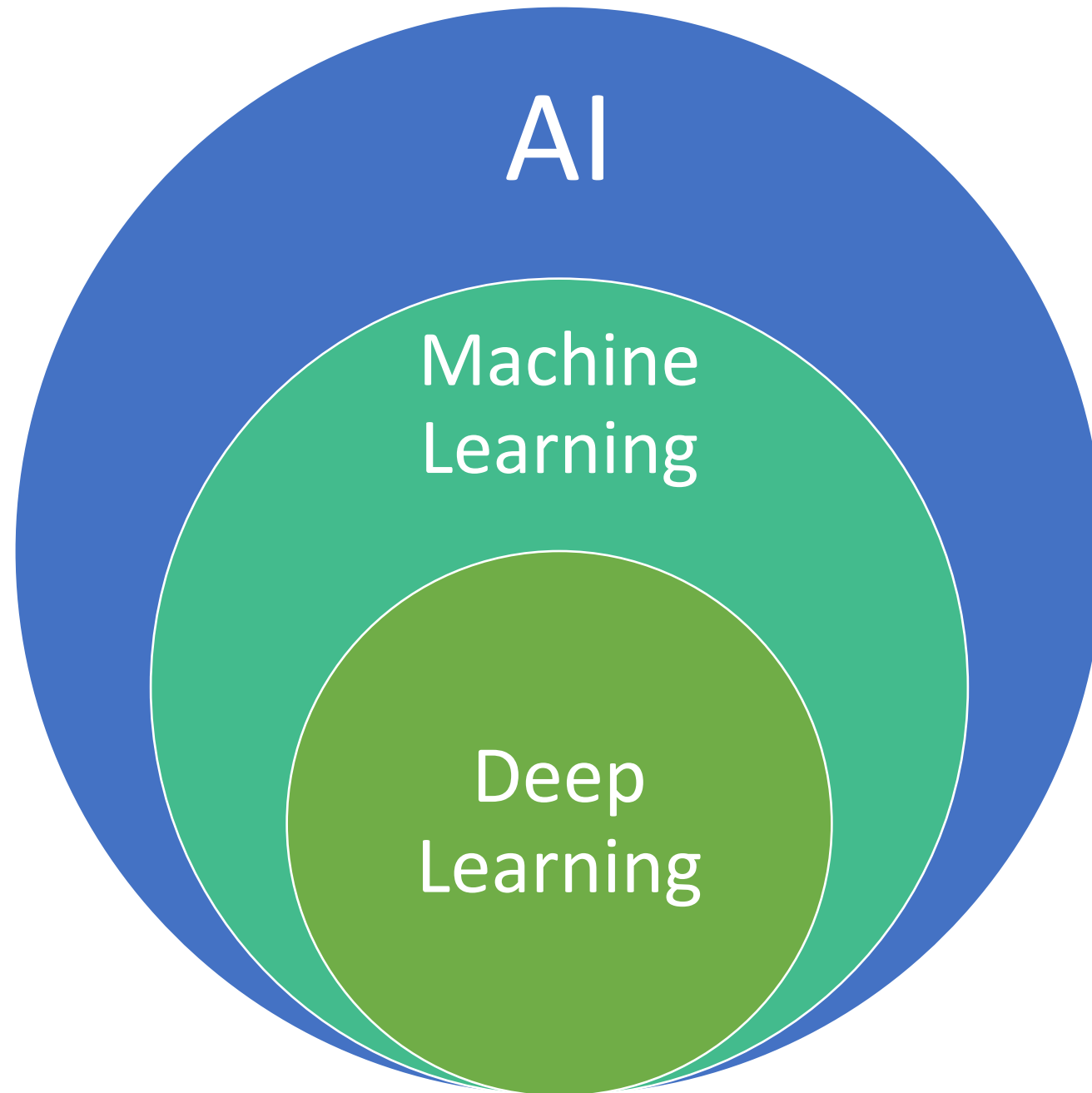


So, what is AI?

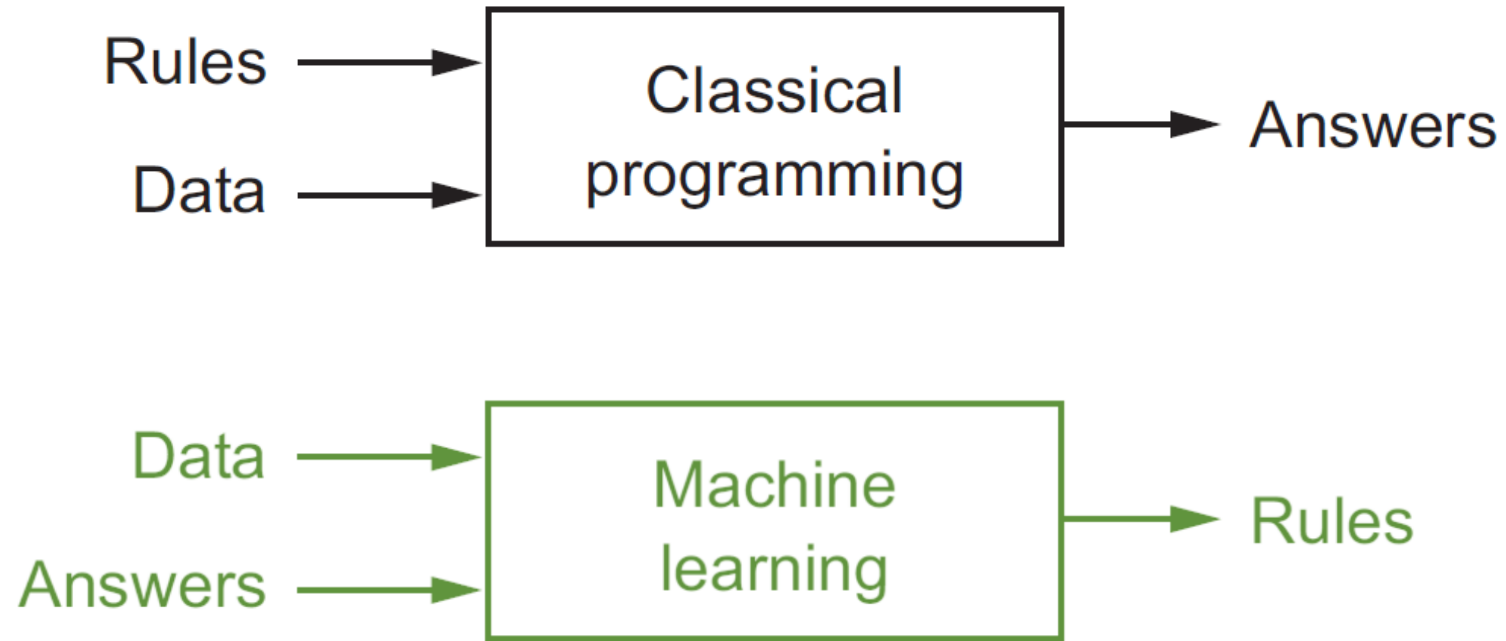


# Machine Learning



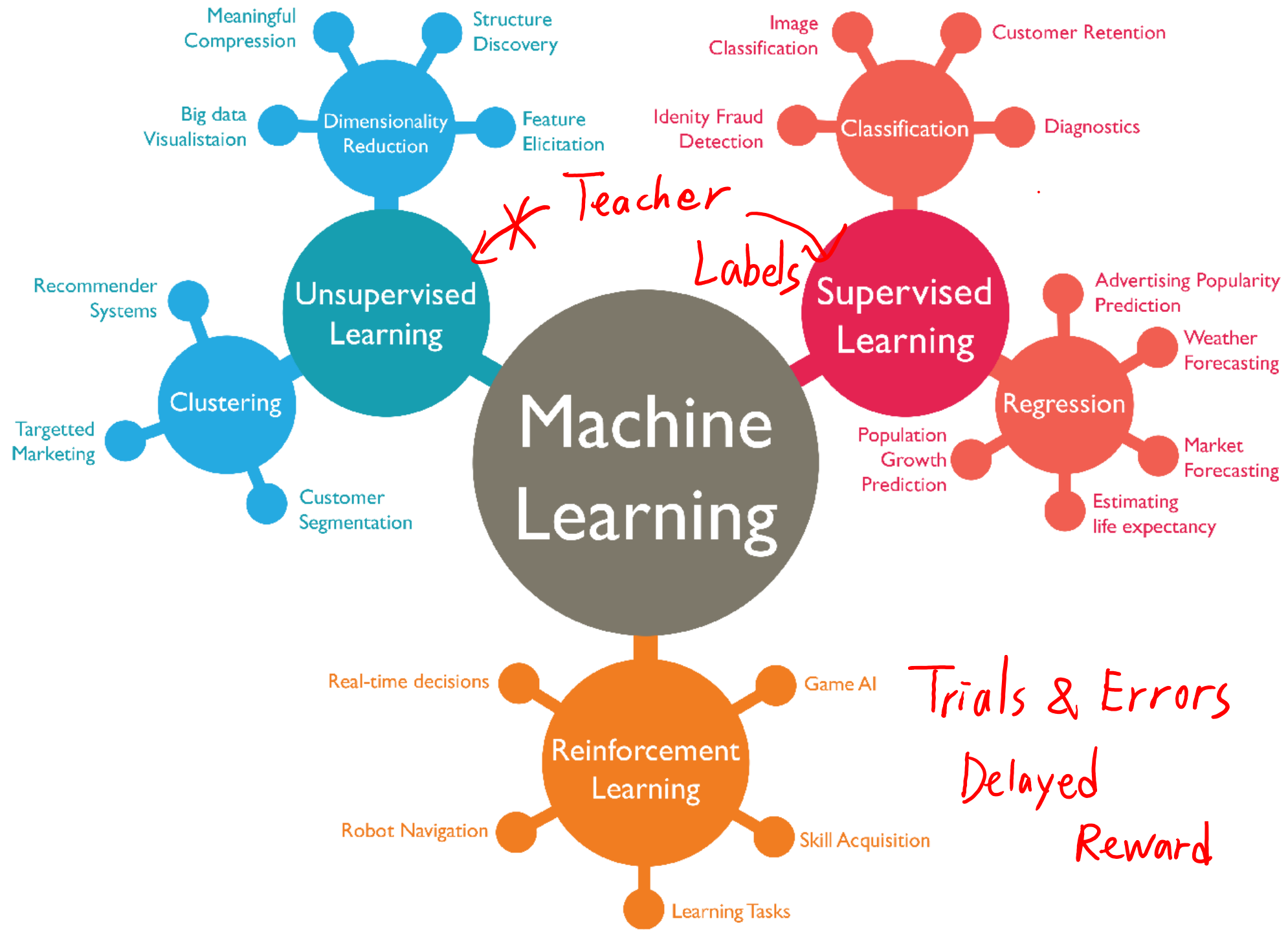


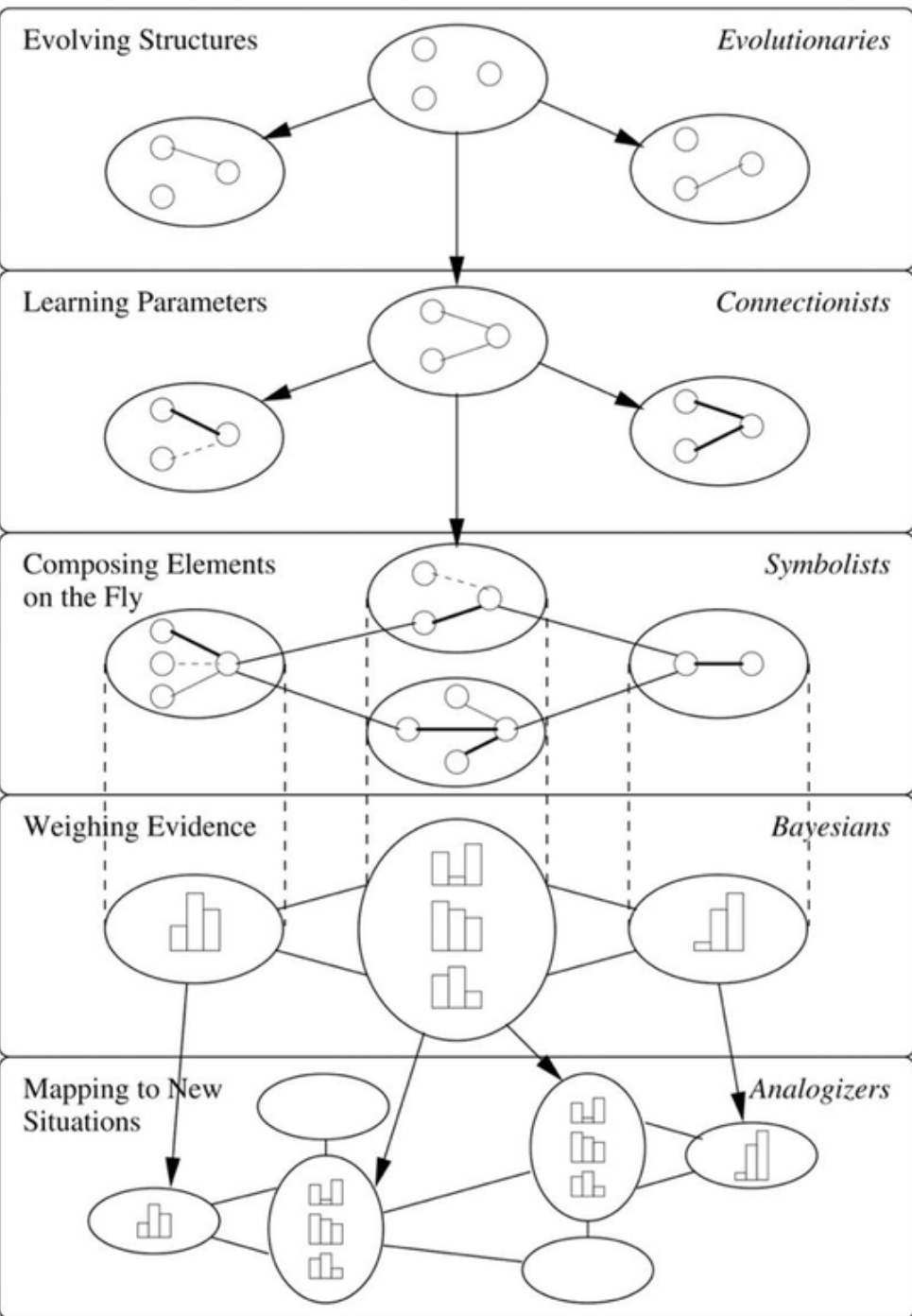
# Machine Learning (Statistical Learning)



Francois Chollet, "Deep Learning with Python," Manning, 2017





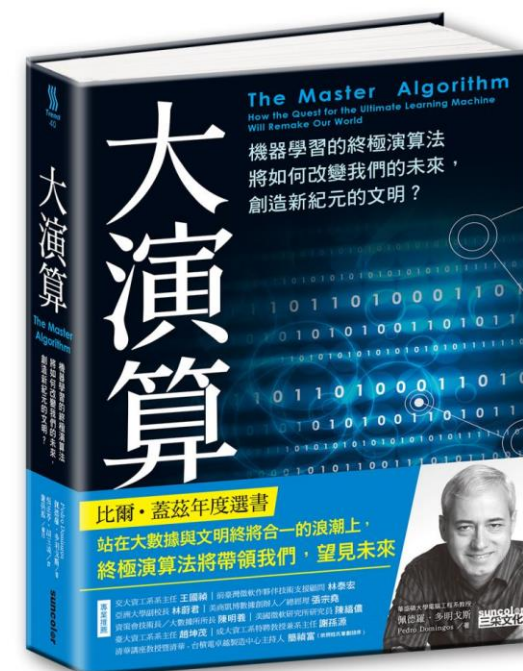
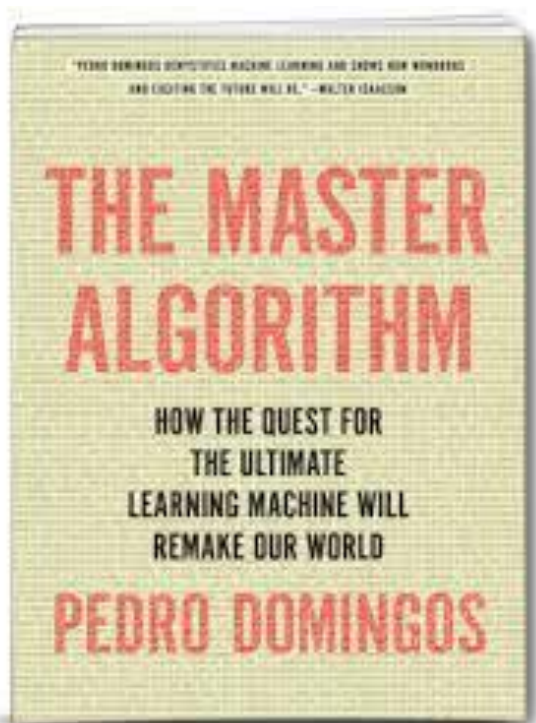


# Five Tribes of Machine Learning

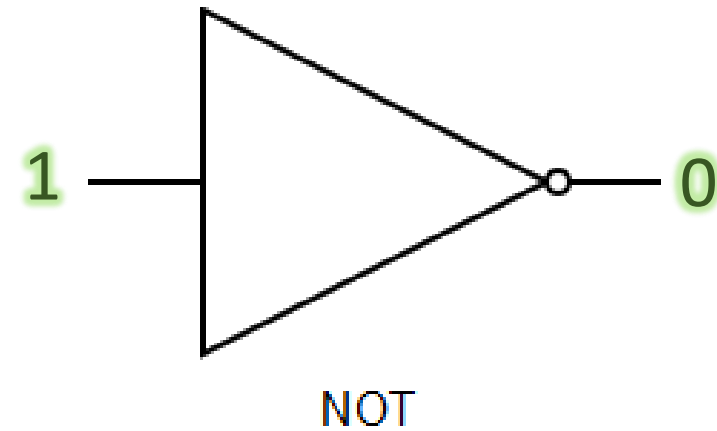
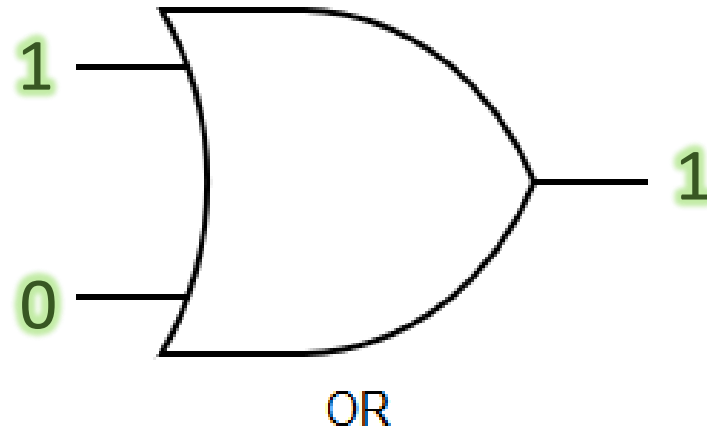
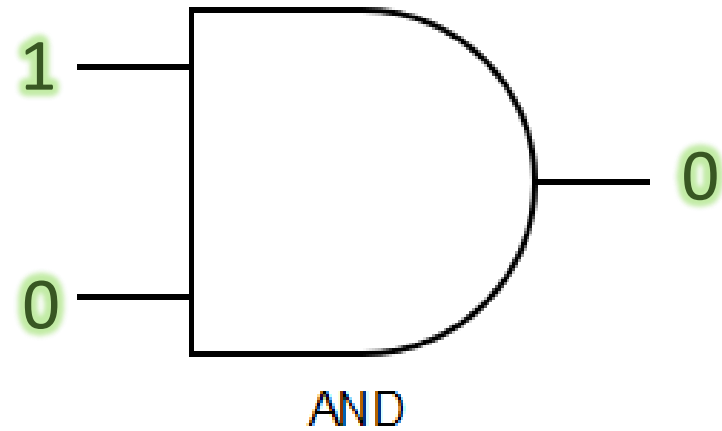
- **Evolutionaries** (演化法)
- **Connectionists** (類神經網路)
- **Symbolists** (邏輯歸納法)
- **Bayesians** (貝氏機率)
- **Analogizers** (類比近似)



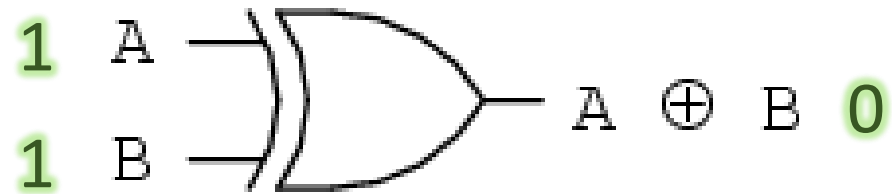
# The Master Algorithm – Pedro Domingos



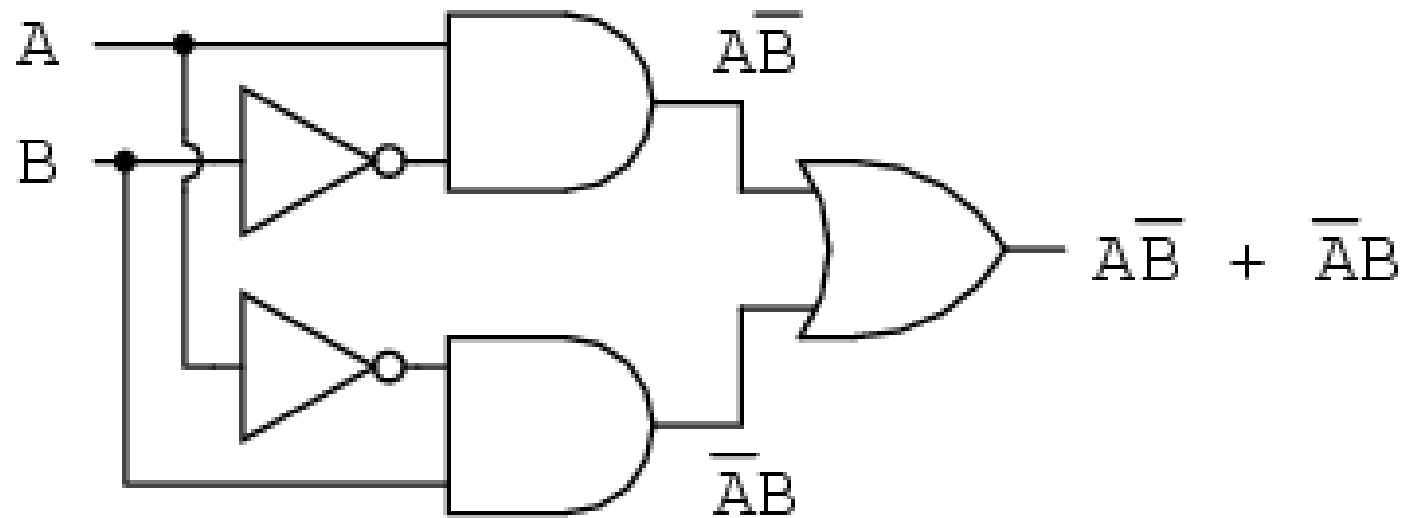
# All Algorithms can be Reduced to 3 Operations



# XOR



... is equivalent to ...

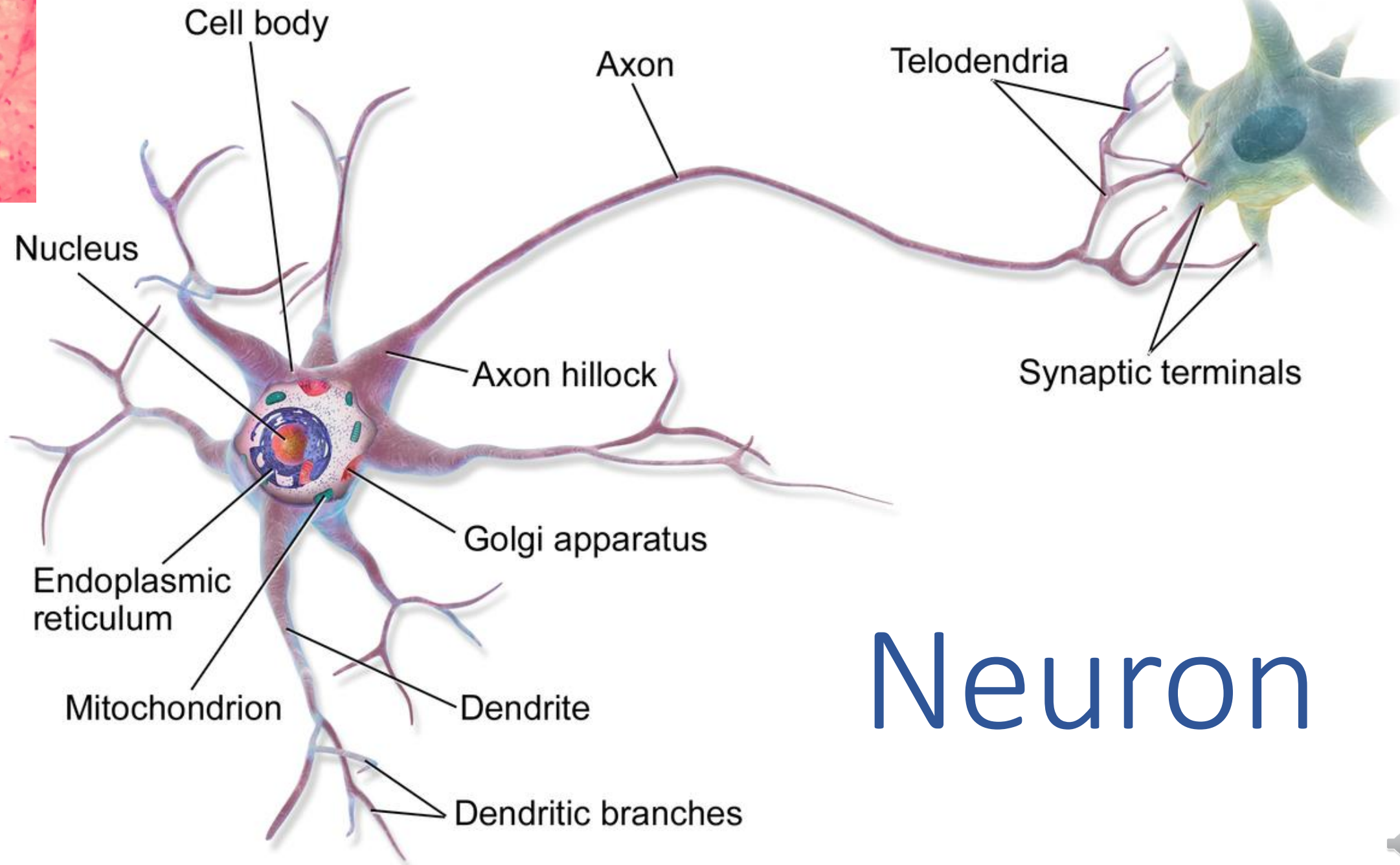
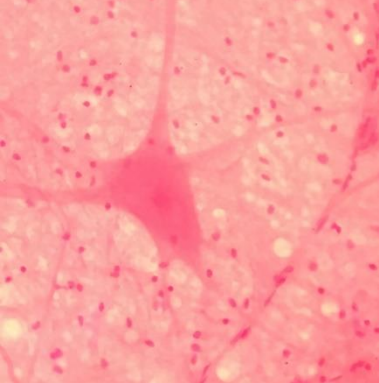


$$A \oplus B = A \bar{B} + \bar{A} B$$



# Neural Networks

The background of the slide features a complex network of blue, branching structures resembling neurons. These structures are interconnected by a web of fine lines. Scattered throughout this network are numerous small, glowing orange-yellow spheres, which represent nodes or data points within the network. The overall color scheme is a deep blue with vibrant orange highlights, creating a high-tech, digital aesthetic.



# Neuron



# Number of Connections in the Brain

**Neurons (for adults):**

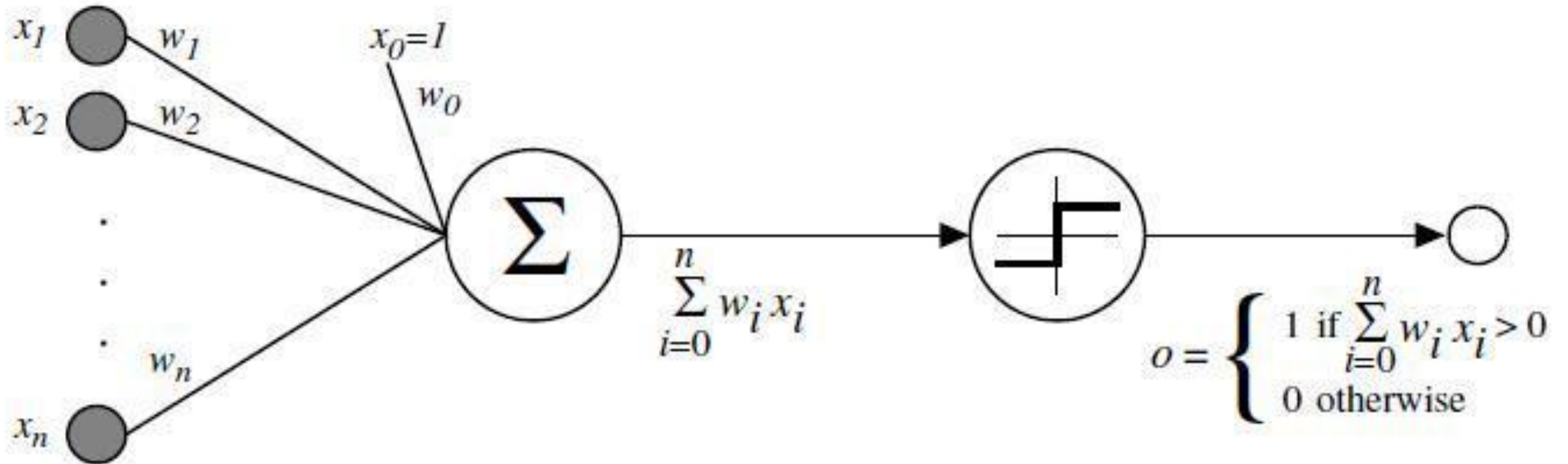
**$10^{11}$ , or 100 billion, 100000000000**

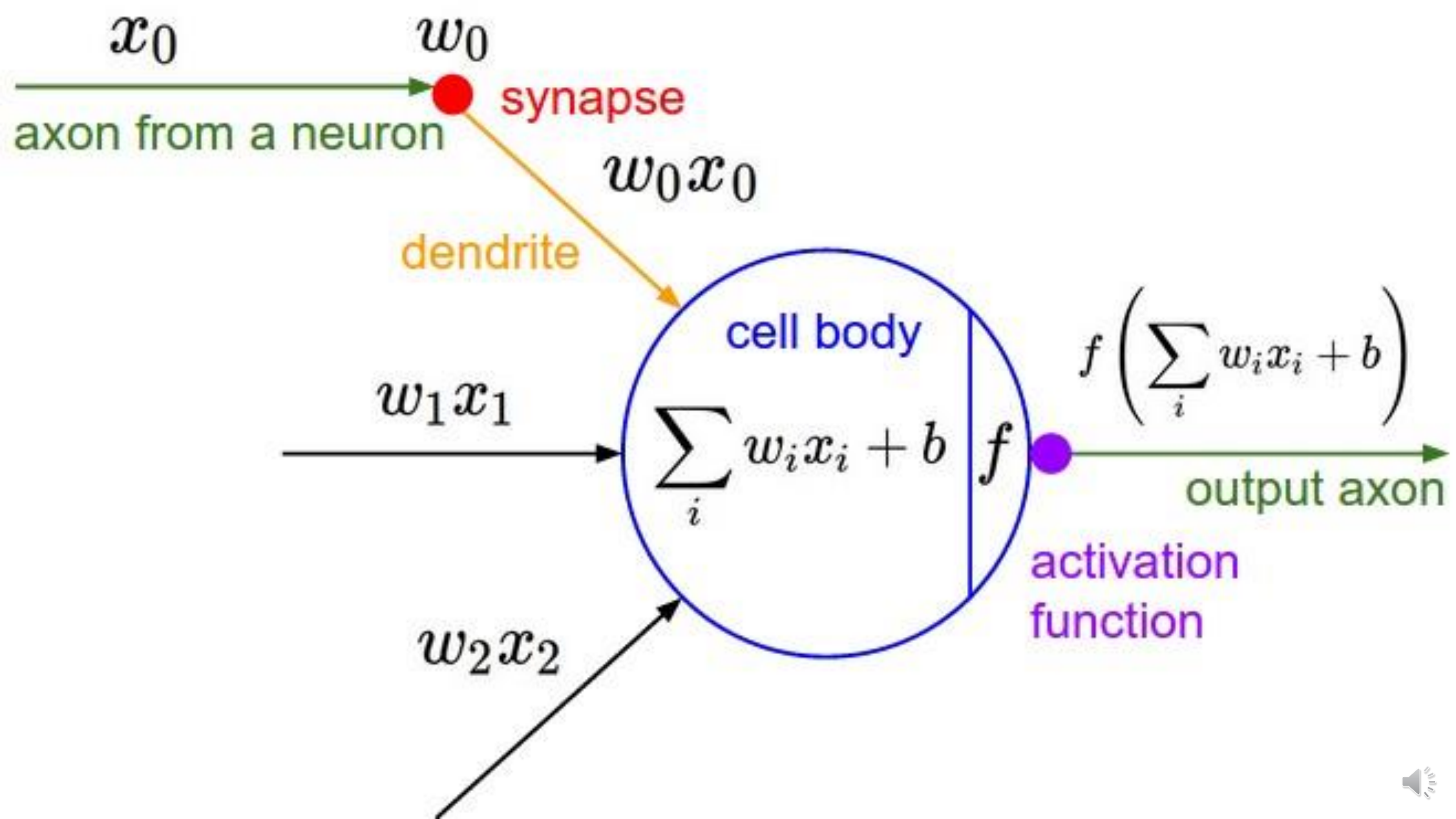
**Synapses (based on 1000 per neuron):**

**$10^{14}$ , or 100 trillion, 100000000000000**

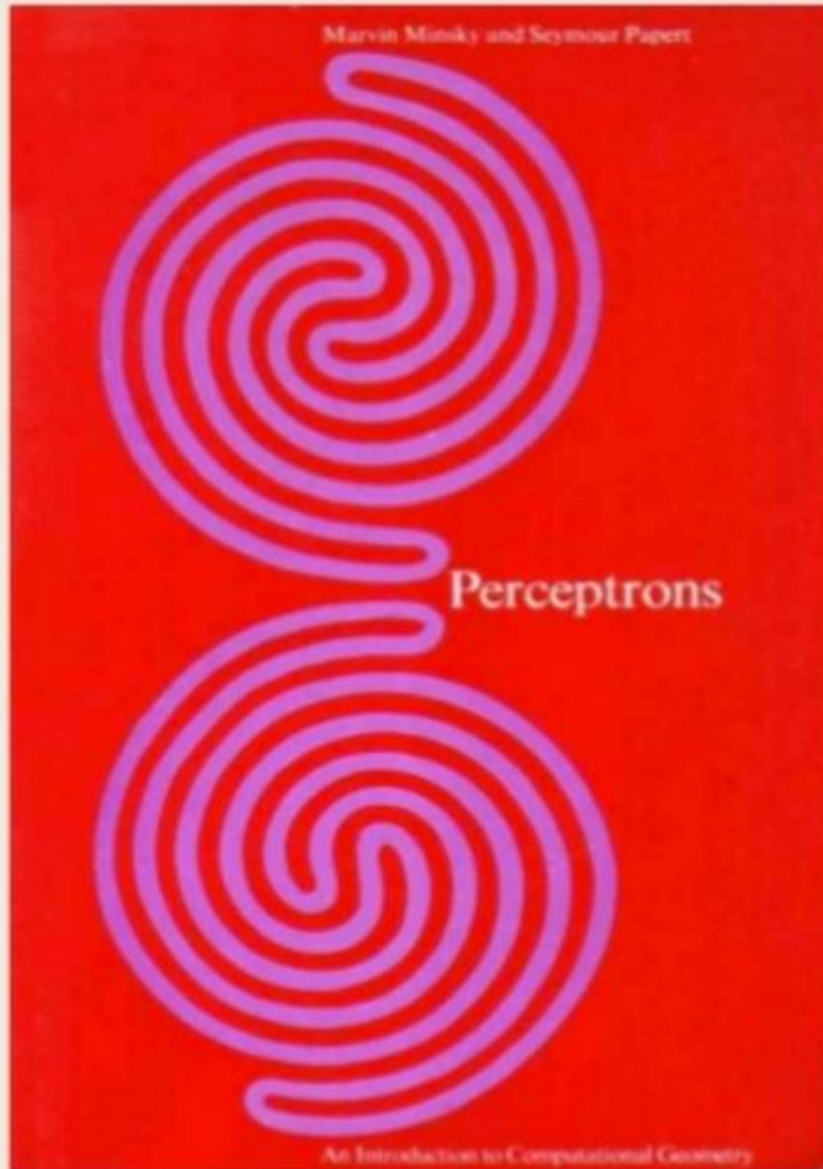


# Frank Rosenblatt's Perceptron (1957)

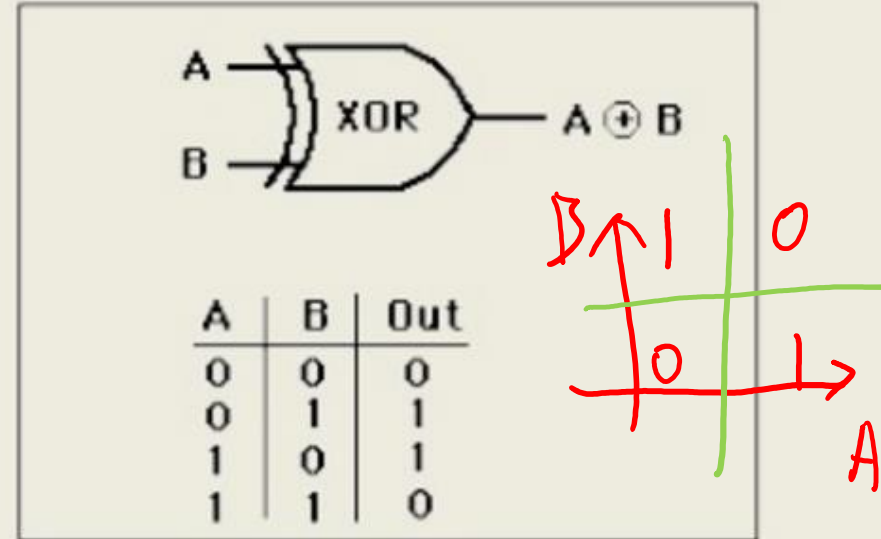




# 1969: Perceptrons can't do XOR!



<http://www.i-programmer.info/images/stories/BabBag/AI/book.jpg>



<http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/ietron/xor.gif>



Minsky & Papert

<https://constructingkids.files.wordpress.com/2013/05/minsky-papert-71-csolomon-x640.jpg>





AI Winter  
1969 - 1990



# Deep Learning



Geoffrey Hinton  
(Toronto, Google)

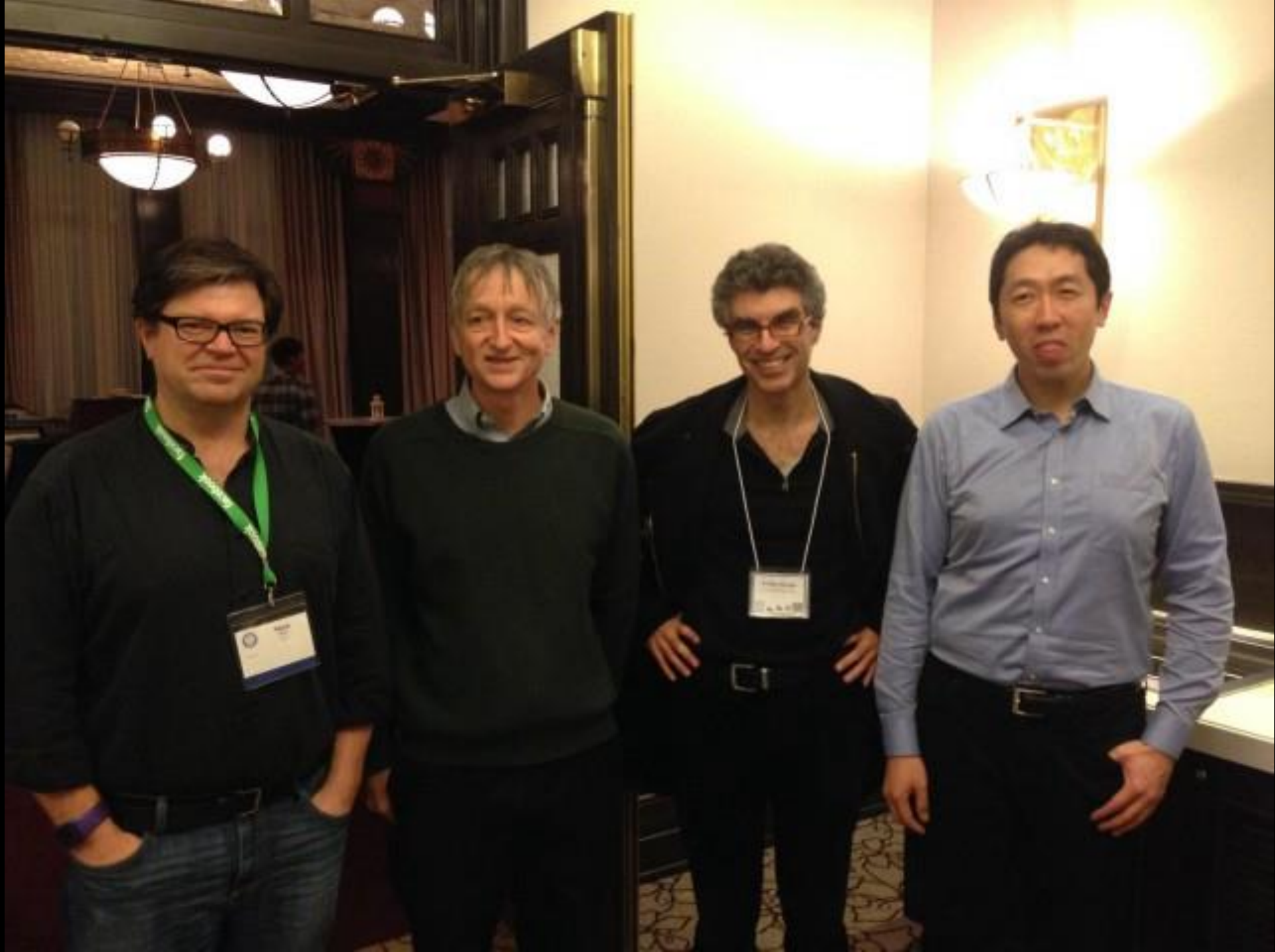


Yann LeCun  
(New York, Facebook)



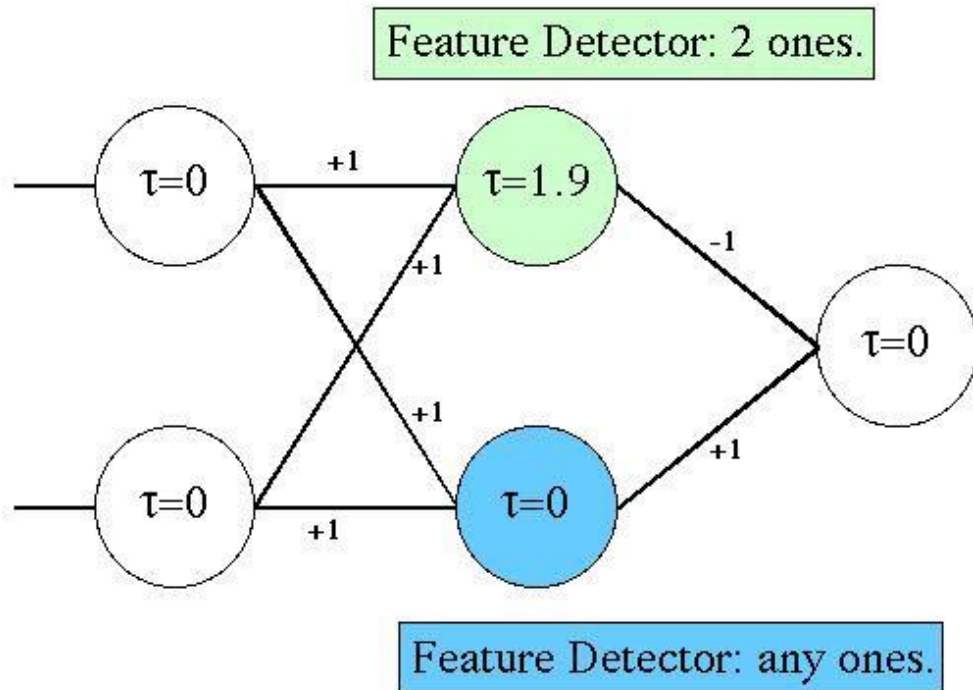
Yoshua Bengio  
(Montreal)





# Learning XOR (1986)

## XOR Network

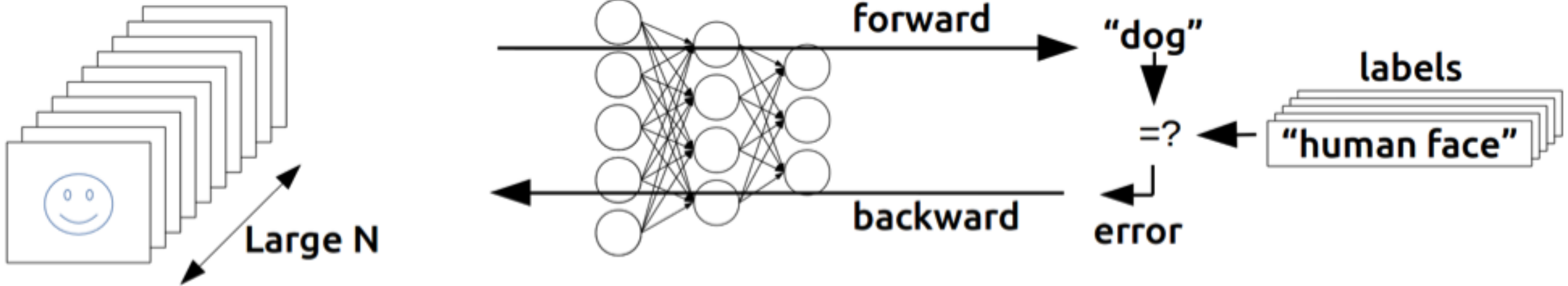


## Geoffrey Hinton



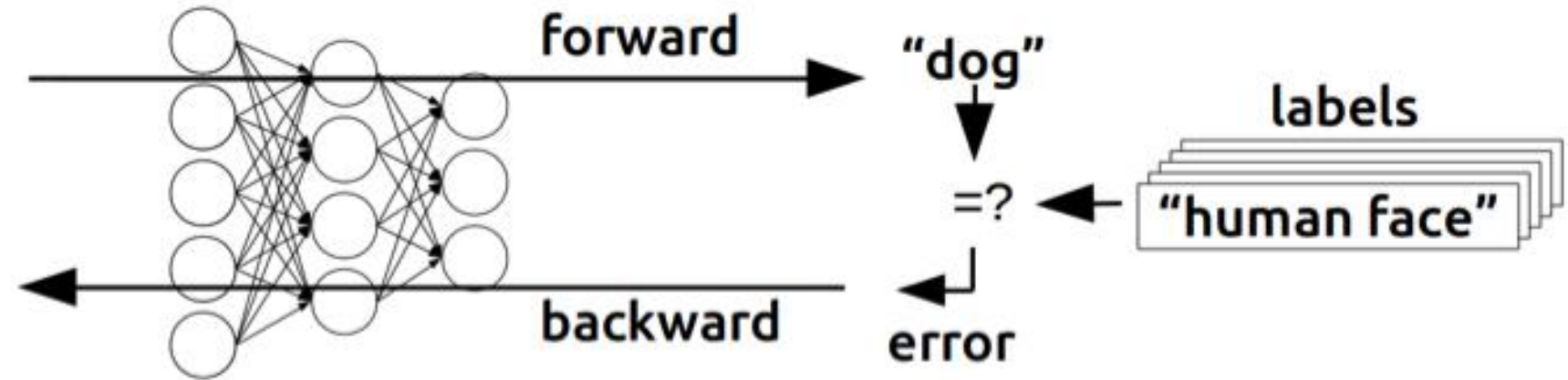
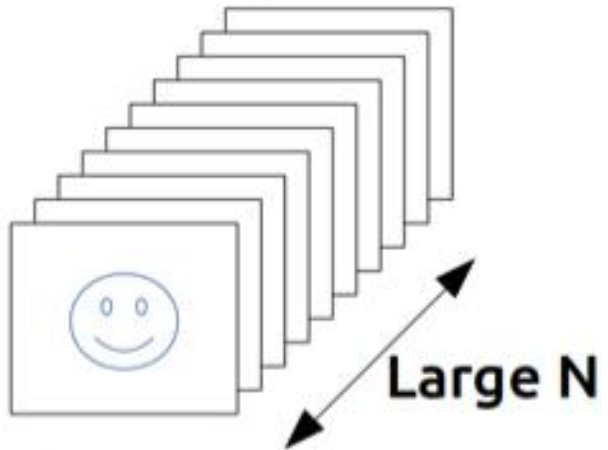
# Backpropagation

## Training

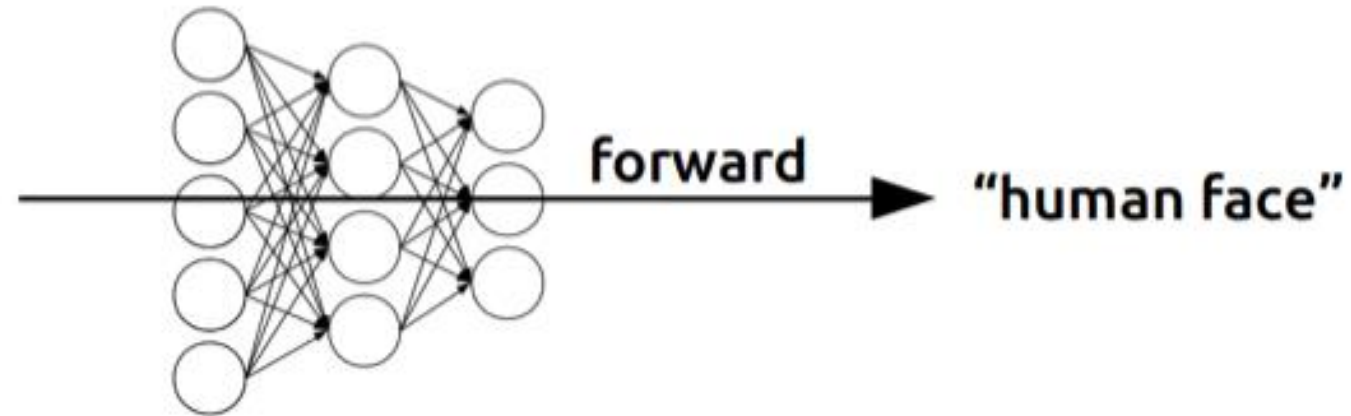
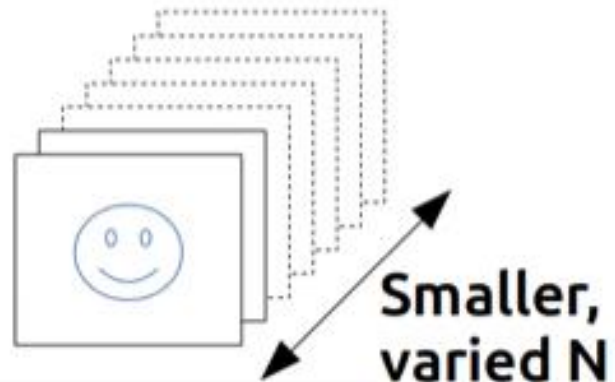


# Inference

## Training



## Inference



# Chain Rule

$$\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$$

$$\frac{d^2 y}{dx^2} = \frac{d^2 y}{du^2} \left( \frac{du}{dx} \right)^2 + \frac{dy}{du} \frac{d^2 u}{dx^2}$$

$$\frac{d^3 y}{dx^3} = \frac{d^3 y}{du^3} \left( \frac{du}{dx} \right)^3 + 3 \frac{d^2 y}{du^2} \frac{du}{dx} \frac{d^2 u}{dx^2} + \frac{dy}{du} \frac{d^3 u}{dx^3}$$

$$\frac{d^4 y}{dx^4} = \frac{d^4 y}{du^4} \left( \frac{du}{dx} \right)^4 + 6 \frac{d^3 y}{du^3} \left( \frac{du}{dx} \right)^2 \frac{d^2 u}{dx^2} + \frac{d^2 y}{du^2} \left( 4 \frac{du}{dx} \frac{d^3 u}{dx^3} + 3 \left( \frac{d^2 u}{dx^2} \right)^2 \right) + \frac{dy}{du} \frac{d^4 u}{dx^4}.$$



# Computation Graph

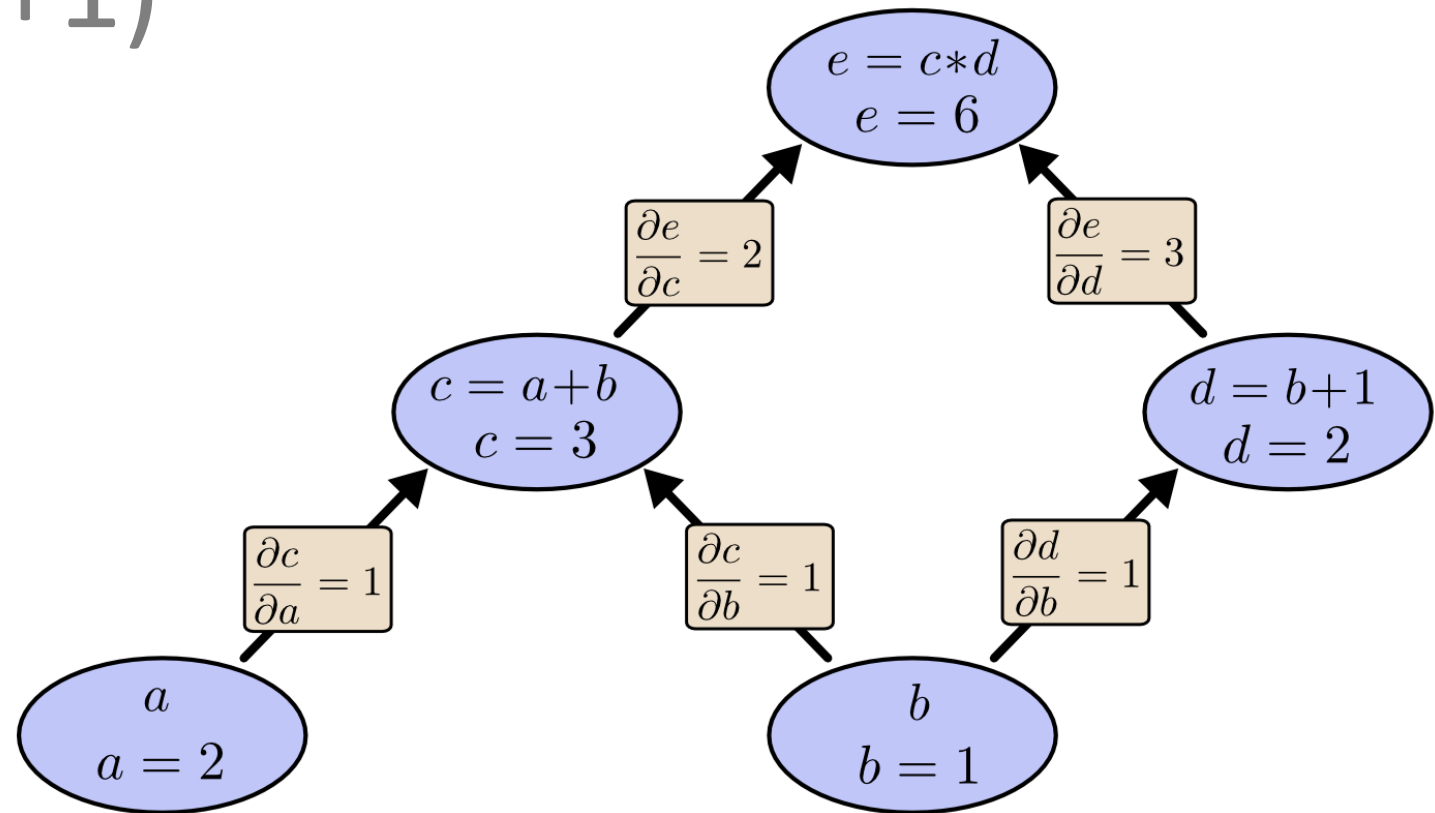
$$e = (a+b) * (b+1)$$

=>

$$c = a + b$$

$$d = b + 1$$

$$e = c * d$$



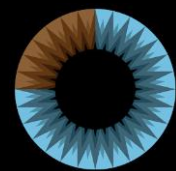
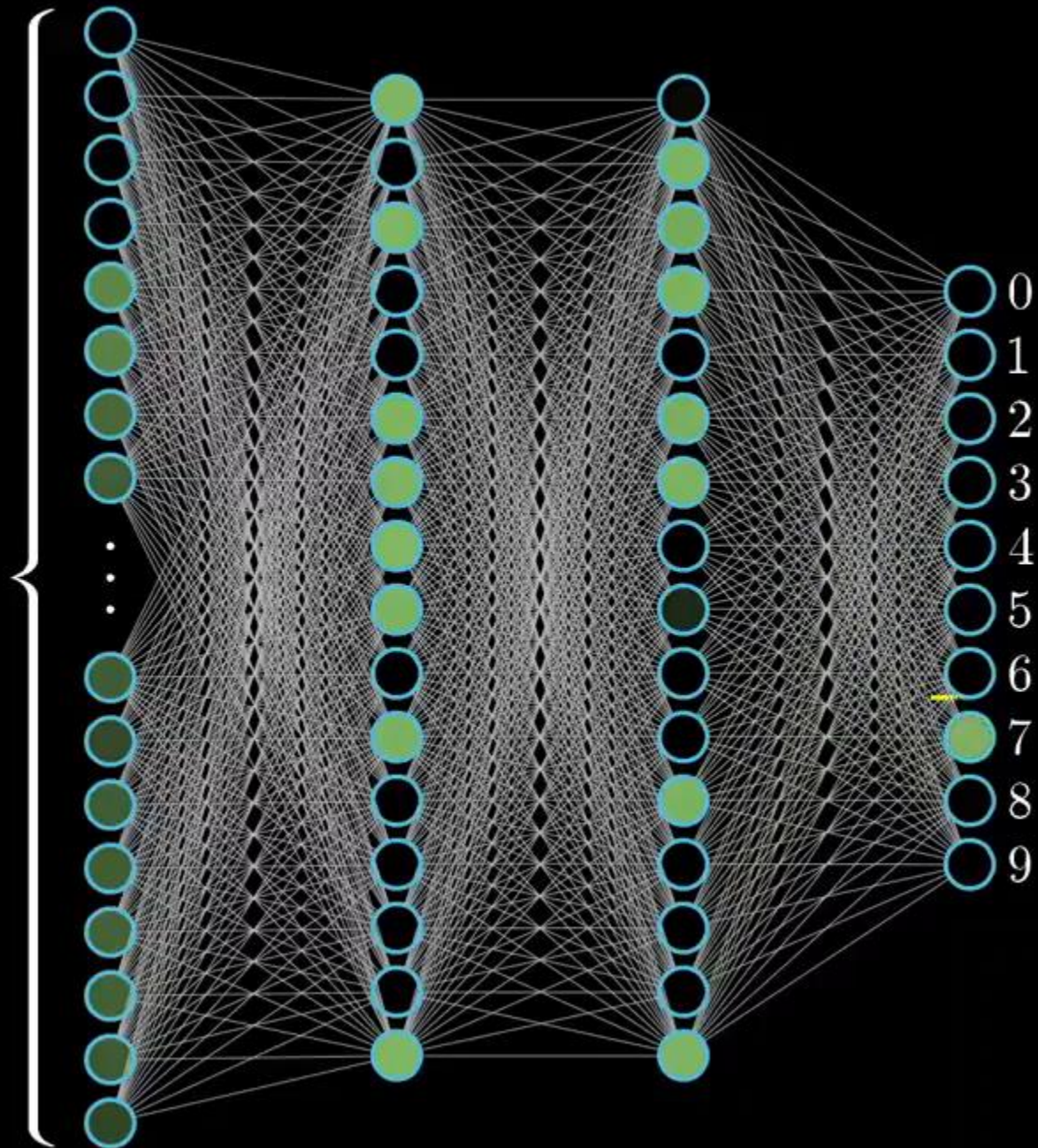
# Example: Recognizing Handwritten Digits

- MNIST dataset



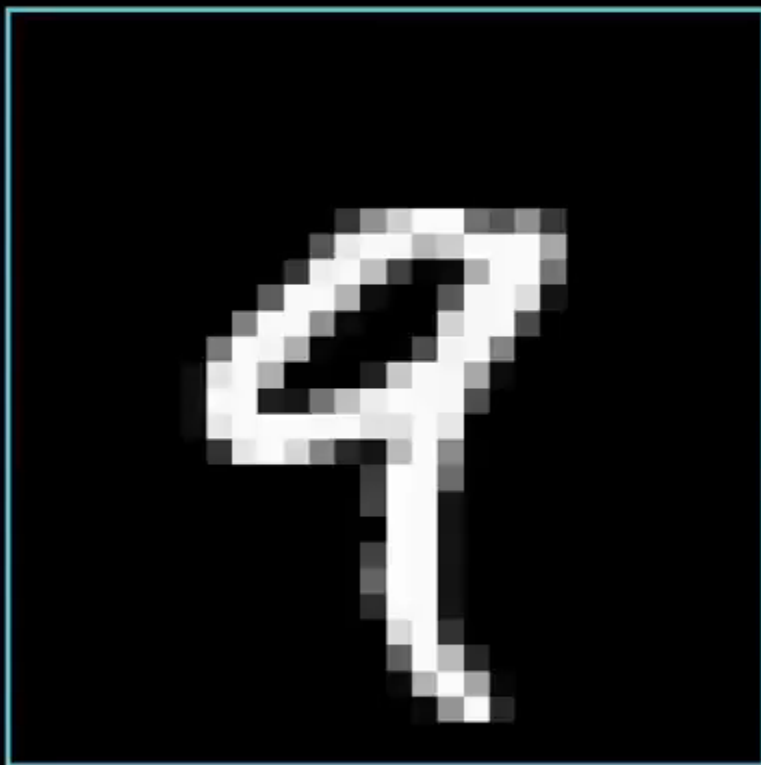


784

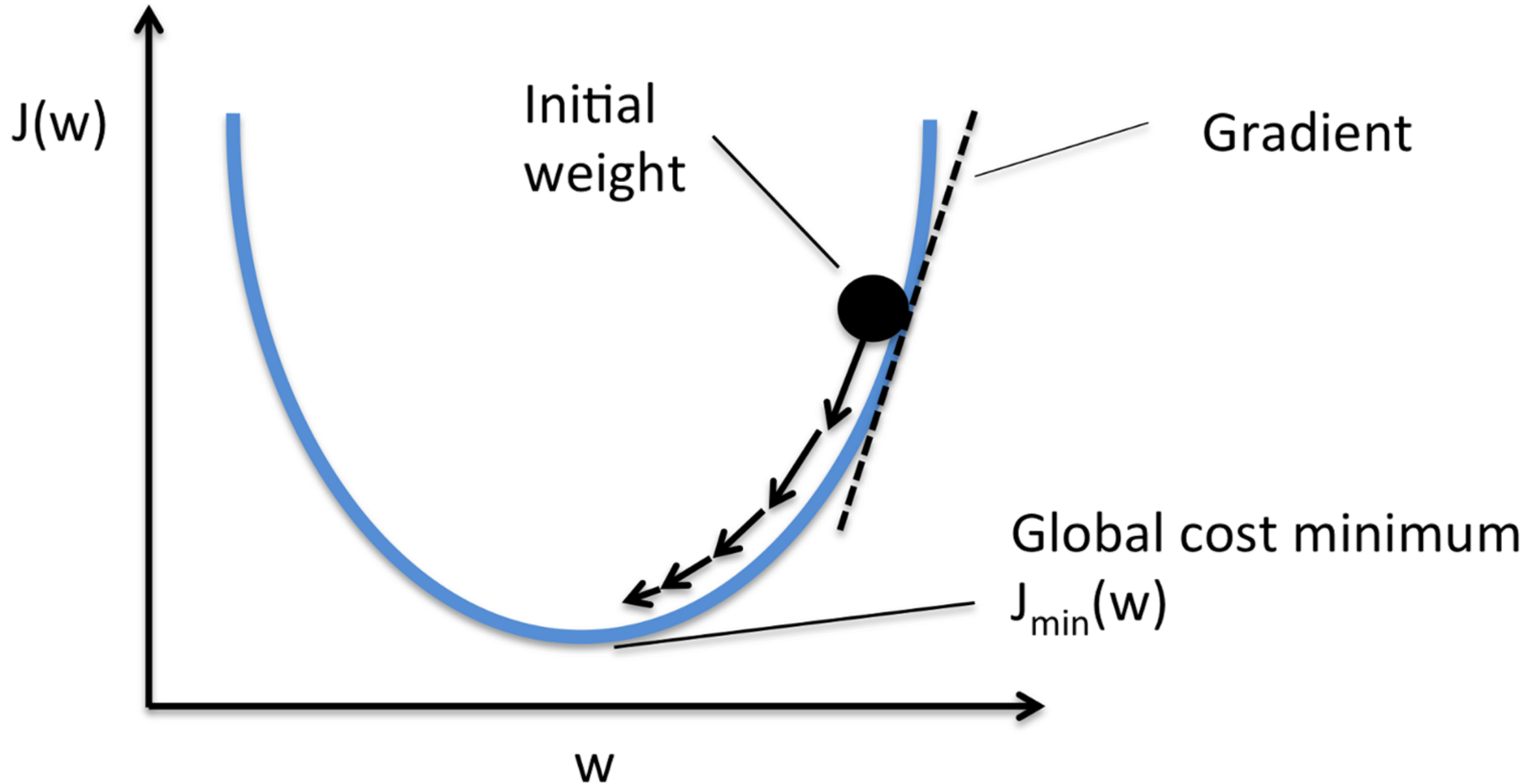


3Blue1Brown





# Gradient Descent



## Cost function

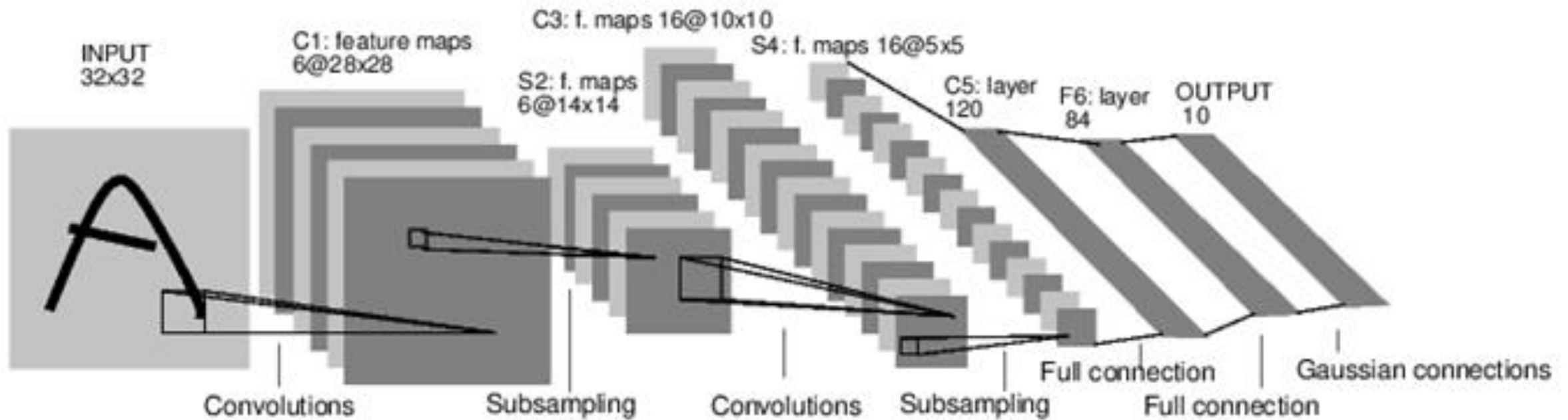
$$C(\underbrace{w_1, w_2, \dots, w_{13,002}}_{\text{Weights and biases}})$$

Weights and biases



# Convolutional Neural Network (LeNet-5)

- <https://medium.com/@sh.tsang/paper-brief-review-of-lenet-1-lenet-4-lenet-5-boosted-lenet-4-image-classification-1f5f809dbf17>



A Full Convolutional Neural Network (LeNet)



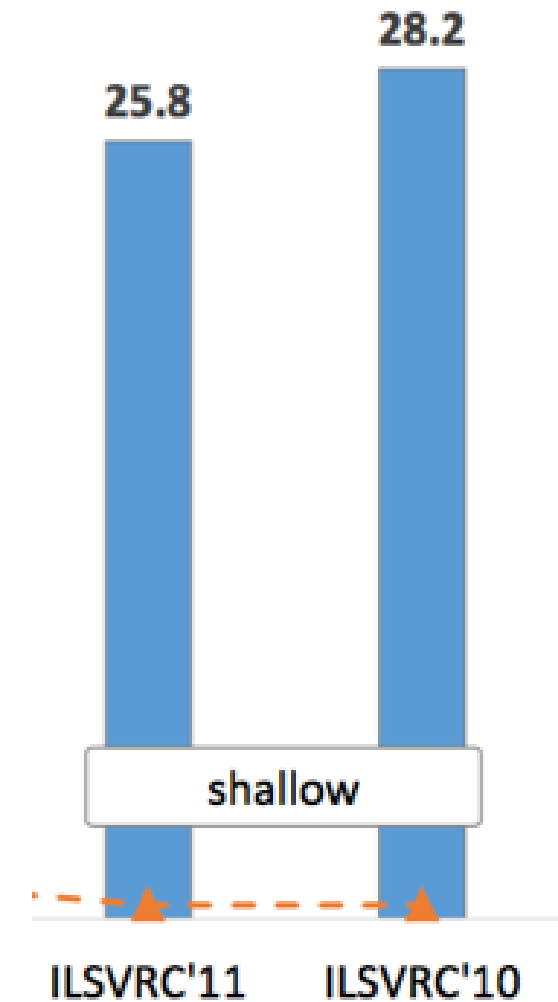


# ImageNet Large Scale Visual Object Recognition Challenge (ILSVRC)

- 1000 categories
- For ILSVRC 2017
  - **Training images** for each category ranges from 732 to 1300
  - 50,000 validation **images** and 100,000 test **images**.
- Total number of images in ILSVRC 2017 is around 1,150,000

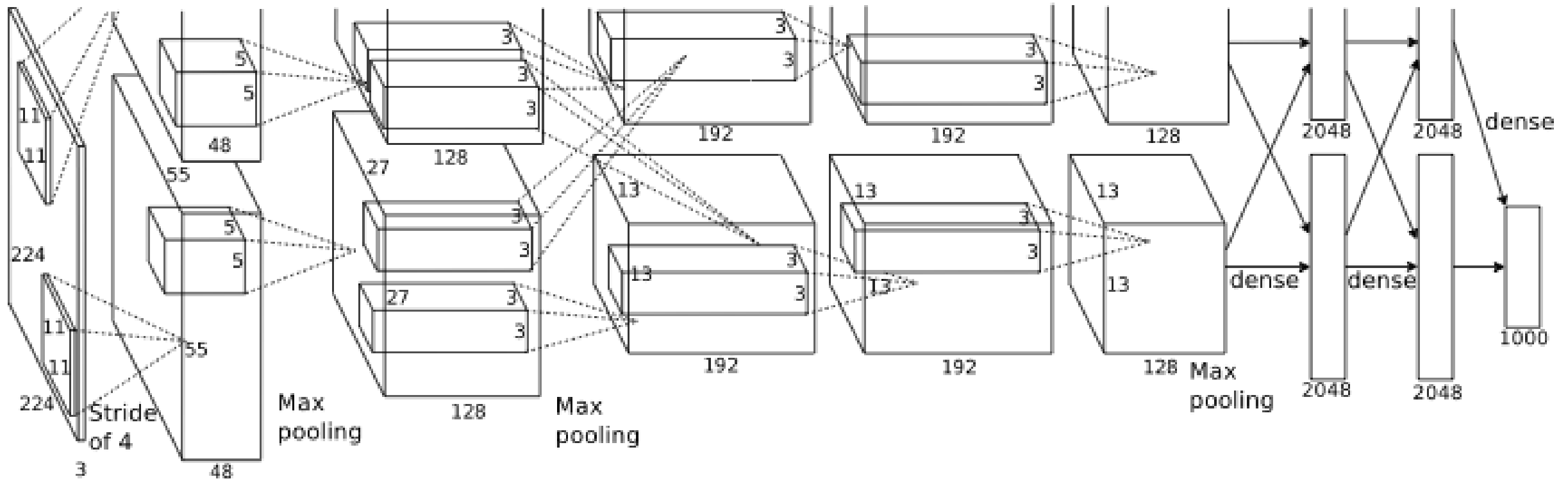


# Error Rate on ImageNet Challenge

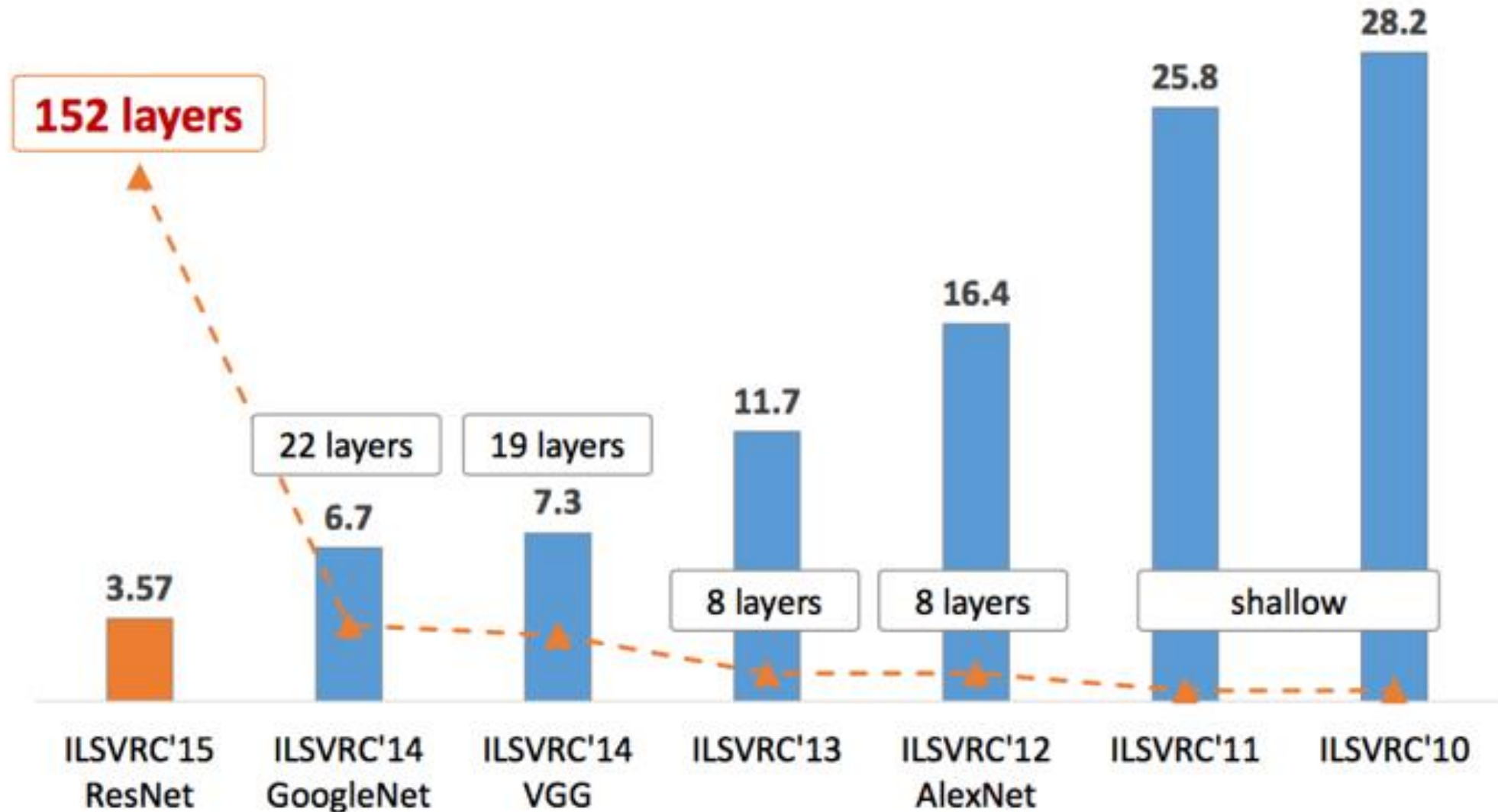


# Convolutional Neural Network (AlexNet)

- Alex Krizhevsky, Geoffery Hinton et al., 2012



# Error Rate on ImageNet Challenge



```

graph BT
    A[Convolution 5x5, 30] -- "30x144x144" --> B[Activation tanh]
    B -- "30x144x144" --> C[Pooling max, 2x2]
    C -- "30x144x144" --> D[Convolution 5x5, 30]
    D -- "30x144x144" --> E[Activation tanh]
    E -- "30x144x144" --> F[Pooling max, 2x2]
    F -- "29x200" --> G[Flatten]
    G -- "300" --> H[Fully Connected 500]
    H -- "300" --> I[Activation tanh]
    I -- "300" --> J[Fully Connected 10]
    J -- "10" --> K[Softmax Output]
  
```

The diagram illustrates the SoftNet architecture, showing the flow of data through various layers. The layers are represented by colored boxes, and the connections between them are labeled with their respective dimensions or operations.

- Input:** A red box labeled "Convolution 3x5/7, 256".
- Activation:** A yellow box labeled "Activation relu".
- Pooling:** A blue box labeled "Pooling max, 3x3/2".
- Layer Normalization:** A pink box labeled "L2RN".
- Convolution:** A red box labeled "Convolution 3x3/7, 256".
- Activation:** A yellow box labeled "Activation relu".
- Convolution:** A red box labeled "Convolution 3x3/7, 256".
- Activation:** A yellow box labeled "Activation relu".
- Convolution:** A red box labeled "Convolution 3x3/7, 256".
- Activation:** A yellow box labeled "Activation relu".
- Flatten:** An orange box labeled "Flatten".
- Dropout:** A pink box labeled "Dropout".
- Fully Connected:** A red box labeled "Fully Connected 4096".
- Activation:** A yellow box labeled "Activation relu".
- Dropout:** A pink box labeled "Dropout".
- Fully Connected:** A red box labeled "Fully Connected 2".
- Output:** A pink box labeled "SoftNet Output".

The connections between the layers are labeled with their respective dimensions or operations:

- From Input to Activation: 256x36x36
- From Activation to Pooling: 256x36x36
- From Pooling to L2RN: 256x36x36
- From L2RN to Convolution: 256x36x36
- From Convolution to Activation: 256x36x36
- From Convolution to Convolution: 256x36x36
- From Convolution to Activation: 256x36x36
- From Convolution to Convolution: 256x36x36
- From Convolution to Activation: 256x36x36
- From Convolution to Flatten: 256x36x36
- From Flatten to Dropout: 2048
- From Dropout to Fully Connected: 4096
- From Fully Connected to Activation: 4096
- From Activation to Dropout: 4096
- From Dropout to Fully Connected: 4096
- From Fully Connected to Output: 2

```

graph TD
    A[Soften/Output] -- 2 --> B[FullyConnected 2]
    B -- 4096 --> C[Dropout]
    C -- 4096 --> D[Activation sftu]
    D -- 4096 --> E[FullyConnected 4096]
    E -- 4096 --> F[Dropout]
    F -- 4096 --> G[Activation sftu]
    G -- 4096 --> H[FullyConnected 4096]
    H -- 12120 --> I[Flatten]
    I -- 312x30x39 --> J[Pooling max, 2x2x3]
    J -- 312x30x39 --> K[Activation sftu]
    K -- 312x30x39 --> L[Convolution 3x3x3]
    L -- 312x30x39 --> M[Activation sftu]
    M -- 312x30x39 --> N[Convolution 3x3x3]
    N -- 312x30x39 --> O[Pooling max, 2x2x3]
    O -- 312x30x39 --> P[Activation sftu]
    P -- 312x30x39 --> Q[Convolution 3x3x3]
    Q -- 312x30x39 --> R[Activation sftu]
    R -- 312x30x39 --> S[Convolution 3x3x3]
    S -- 256x30x39 --> T[Pooling max, 2x2x3]
    T -- 256x30x39 --> U[Activation sftu]
    U -- 256x30x39 --> V[Convolution 3x3x3]
    V -- 256x30x39 --> W[Activation sftu]
    W -- 256x30x39 --> X[Convolution 3x3x3]
    X -- 256x30x39 --> Y[Pooling max, 2x2x3]
    Y -- 256x30x39 --> Z[Activation sftu]
    Z -- 256x30x39 --> AA[Convolution 3x3x3]
    AA -- 256x30x39 --> AB[Activation sftu]
    AB -- 256x30x39 --> AC[Convolution 3x3x3]
    AC -- 256x30x39 --> AD[Pooling max, 2x2x3]
    AD -- 256x30x39 --> AE[Activation sftu]
    AE -- 256x30x39 --> AF[Convolution 3x3x3]
    AF -- 256x30x39 --> AG[Activation sftu]
    AG -- 256x30x39 --> AH[Convolution 3x3x3]
    AH -- 256x30x39 --> AI[Pooling max, 2x2x3]
    AI -- 256x30x39 --> AJ[Activation sftu]
    AJ -- 256x30x39 --> AK[Convolution 3x3x3]
    AK -- 256x30x39 --> AL[Activation sftu]
    AL -- 256x30x39 --> AM[Convolution 3x3x3]
    AM -- 256x30x39 --> AN[Pooling max, 2x2x3]
    AN -- 256x30x39 --> AO[Activation sftu]
    AO -- 256x30x39 --> AP[Convolution 3x3x3]
    AP -- 256x30x39 --> AQ[Activation sftu]
    AQ -- 256x30x39 --> AR[Convolution 3x3x3]
    AR -- 256x30x39 --> AS[Pooling max, 2x2x3]
    AS -- 256x30x39 --> AT[Activation sftu]
    AT -- 256x30x39 --> AU[Convolution 3x3x3]
    AU -- 256x30x39 --> AV[Activation sftu]
    AV -- 256x30x39 --> AW[Convolution 3x3x3]
    AW -- 256x30x39 --> AX[Pooling max, 2x2x3]
    AX -- 256x30x39 --> AY[Activation sftu]
    AY -- 256x30x39 --> AZ[Convolution 3x3x3]
    AZ -- 256x30x39 --> BA[Activation sftu]
    BA -- 256x30x39 --> BB[Convolution 3x3x3]
    BB -- 256x30x39 --> BC[Pooling max, 2x2x3]
    BC -- 256x30x39 --> BD[Activation sftu]
    BD -- 256x30x39 --> BE[Convolution 3x3x3]
    BE -- 256x30x39 --> BF[Activation sftu]
    BF -- 256x30x39 --> BG[Convolution 3x3x3]
    BG -- 256x30x39 --> BH[Pooling max, 2x2x3]
    BH -- 256x30x39 --> BI[Activation sftu]
    BI -- 256x30x39 --> BJ[Convolution 3x3x3]
    BJ -- 256x30x39 --> BK[Activation sftu]
    BK -- 256x30x39 --> BL[Convolution 3x3x3]
    BL -- 256x30x39 --> BM[Pooling max, 2x2x3]
    BM -- 256x30x39 --> BN[Activation sftu]
    BN -- 256x30x39 --> BO[Convolution 3x3x3]
    BO -- 256x30x39 --> BP[Activation sftu]
    BP -- 256x30x39 --> BQ[Convolution 3x3x3]
    BQ -- 256x30x39 --> BR[Pooling max, 2x2x3]
    BR -- 256x30x39 --> BS[Activation sftu]
    BS -- 256x30x39 --> BT[Convolution 3x3x3]
    BT -- 256x30x39 --> BU[Activation sftu]
    BU -- 256x30x39 --> BV[Convolution 3x3x3]
    BV -- 256x30x39 --> BW[Pooling max, 2x2x3]
    BW -- 256x30x39 --> BX[Activation sftu]
    BX -- 256x30x39 --> BY[Convolution 3x3x3]
    BY -- 256x30x39 --> BZ[Activation sftu]
    BZ -- 256x30x39 --> C0[Convolution 3x3x3]
    C0 -- 256x30x39 --> C1[Activation sftu]
    C1 -- 256x30x39 --> C2[Convolution 3x3x3]
    C2 -- 256x30x39 --> C3[Pooling max, 2x2x3]
    C3 -- 256x30x39 --> C4[Activation sftu]
    C4 -- 256x30x39 --> C5[Convolution 3x3x3]
    C5 -- 256x30x39 --> C6[Activation sftu]
    C6 -- 256x30x39 --> C7[Convolution 3x3x3]
    C7 -- 256x30x39 --> C8[Pooling max, 2x2x3]
    C8 -- 256x30x39 --> C9[Activation sftu]
    C9 -- 256x30x39 --> CA[Convolution 3x3x3]
    CA -- 256x30x39 --> CB[Activation sftu]
    CB -- 256x30x39 --> CC[Convolution 3x3x3]
    CC -- 256x30x39 --> CD[Pooling max, 2x2x3]
    CD -- 256x30x39 --> CE[Activation sftu]
    CE -- 256x30x39 --> CF[Convolution 3x3x3]
    CF -- 256x30x39 --> CG[Activation sftu]
    CG -- 256x30x39 --> CH[Convolution 3x3x3]
    CH -- 256x30x39 --> CI[Pooling max, 2x2x3]
    CI -- 256x30x39 --> CJ[Activation sftu]
    CJ -- 256x30x39 --> CK[Convolution 3x3x3]
    CK -- 256x30x39 --> CL[Activation sftu]
    CL -- 256x30x39 --> CM[Convolution 3x3x3]
    CM -- 256x30x39 --> CN[Pooling max, 2x2x3]
    CN -- 256x30x39 --> CO[Activation sftu]
    CO -- 256x30x39 --> CP[Convolution 3x3x3]
    CP -- 256x30x39 --> CQ[Activation sftu]
    CQ -- 256x30x39 --> CR[Convolution 3x3x3]
    CR -- 256x30x39 --> CS[Pooling max, 2x2x3]
    CS -- 256x30x39 --> CT[Activation sftu]
    CT -- 256x30x39 --> CU[Convolution 3x3x3]
    CU -- 256x30x39 --> CV[Activation sftu]
    CV -- 256x30x39 --> CW[Convolution 3x3x3]
    CW -- 256x30x39 --> CX[Pooling max, 2x2x3]
    CX -- 256x30x39 --> CY[Activation sftu]
    CY -- 256x30x39 --> CZ[Convolution 3x3x3]
    CZ -- 256x30x39 --> D0[Activation sftu]
    D0 -- 256x30x39 --> D1[Convolution 3x3x3]
    D1 -- 256x30x39 --> D2[Pooling max, 2x2x3]
    D2 -- 256x30x39 --> D3[Activation sftu]
    D3 -- 256x30x39 --> D4[Convolution 3x3x3]
    D4 -- 256x30x39 --> D5[Activation sftu]
    D5 -- 256x30x39 --> D6[Convolution 3x3x3]
    D6 -- 256x30x39 --> D7[Pooling max, 2x2x3]
    D7 -- 256x30x39 --> D8[Activation sftu]
    D8 -- 256x30x39 --> D9[Convolution 3x3x3]
    D9 -- 256x30x39 --> DA[Activation sftu]
    DA -- 256x30x39 --> DB[Convolution 3x3x3]
    DB -- 256x30x39 --> DC[Pooling max, 2x2x3]
    DC -- 256x30x39 --> DD[Activation sftu]
    DD -- 256x30x39 --> DE[Convolution 3x3x3]
    DE -- 256x30x39 --> DF[Activation sftu]
    DF -- 256x30x39 --> DG[Convolution 3x3x3]
    DG -- 256x30x39 --> DH[Pooling max, 2x2x3]
    DH -- 256x30x39 --> DI[Activation sftu]
    DI -- 256x30x39 --> DJ[Convolution 3x3x3]
    DJ -- 256x30x39 --> DK[Activation sftu]
    DK -- 256x30x39 --> DL[Convolution 3x3x3]
    DL -- 256x30x39 --> DM[Pooling max, 2x2x3]
    DM -- 256x30x39 --> DN[Activation sftu]
    DN -- 256x30x39 --> DO[Convolution 3x3x3]
    DO -- 256x30x39 --> DP[Activation sftu]
    DP -- 256x30x39 --> DQ[Convolution 3x3x3]
    DQ -- 256x30x39 --> DR[Pooling max, 2x2x3]
    DR -- 256x30x39 --> DS[Activation sftu]
    DS -- 256x30x39 --> DT[Convolution 3x3x3
```

[illegible][illegible]

A close-up shot of Leonardo DiCaprio from the movie Inception. He is wearing a dark suit and tie, looking slightly to his right with a serious expression. The lighting is warm and dramatic, typical of the film's aesthetic. Another person's head and shoulder are visible in the foreground on the right, partially obscuring the view.

**WE NEED TO GO**

**DEEPER**

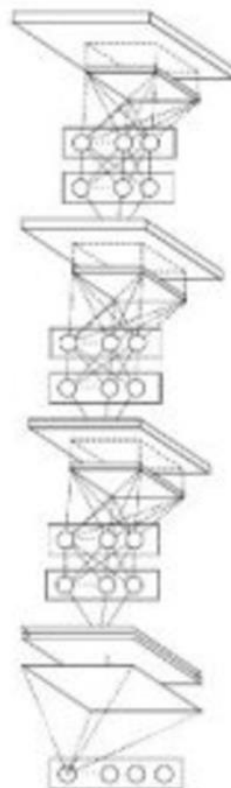
# AlexNet



# VGG



# Network in Network



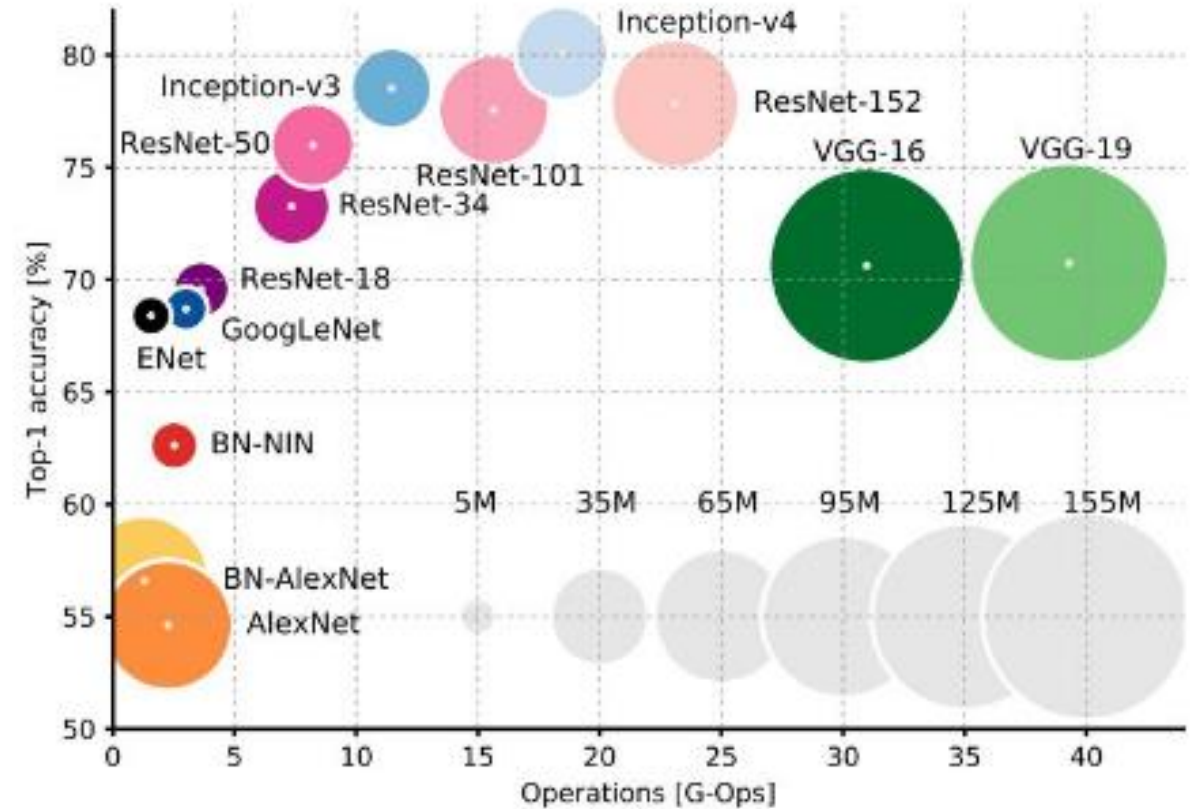
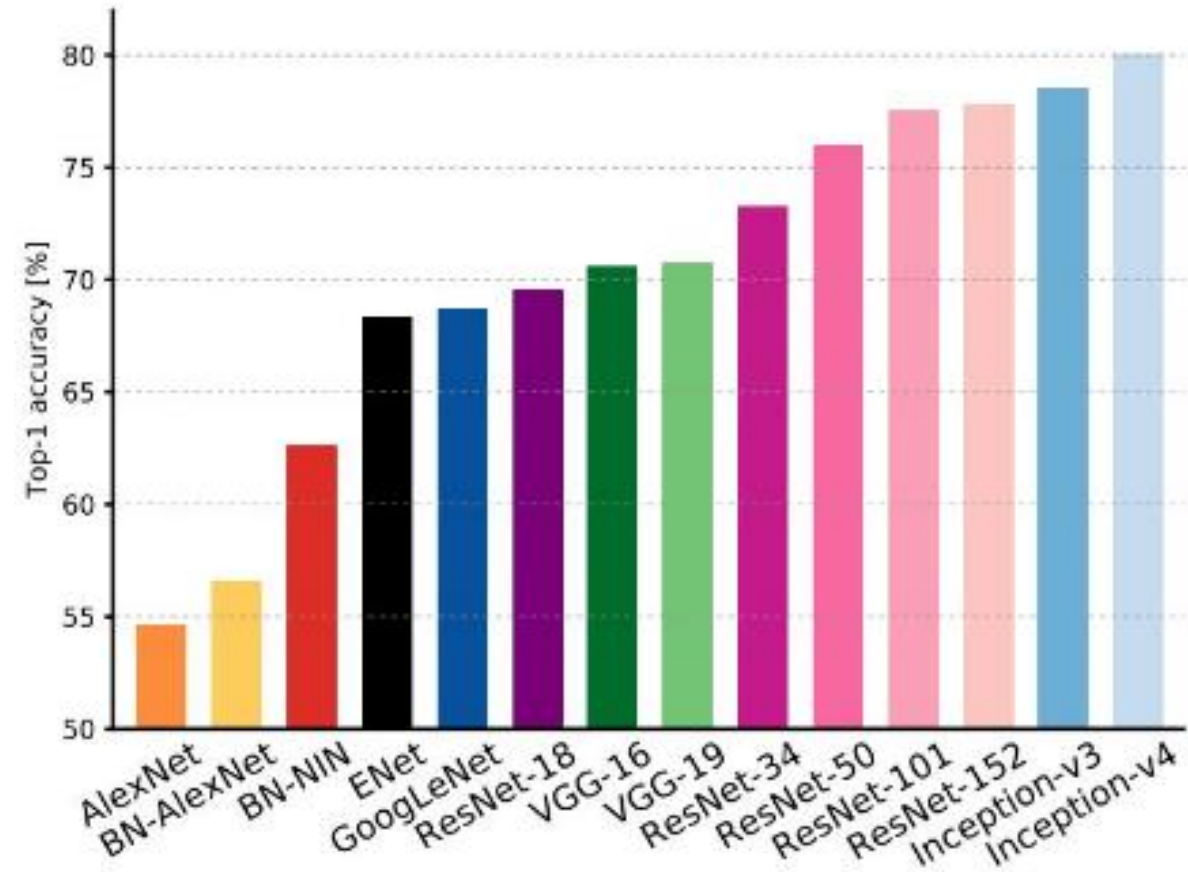
# GoogLeNet



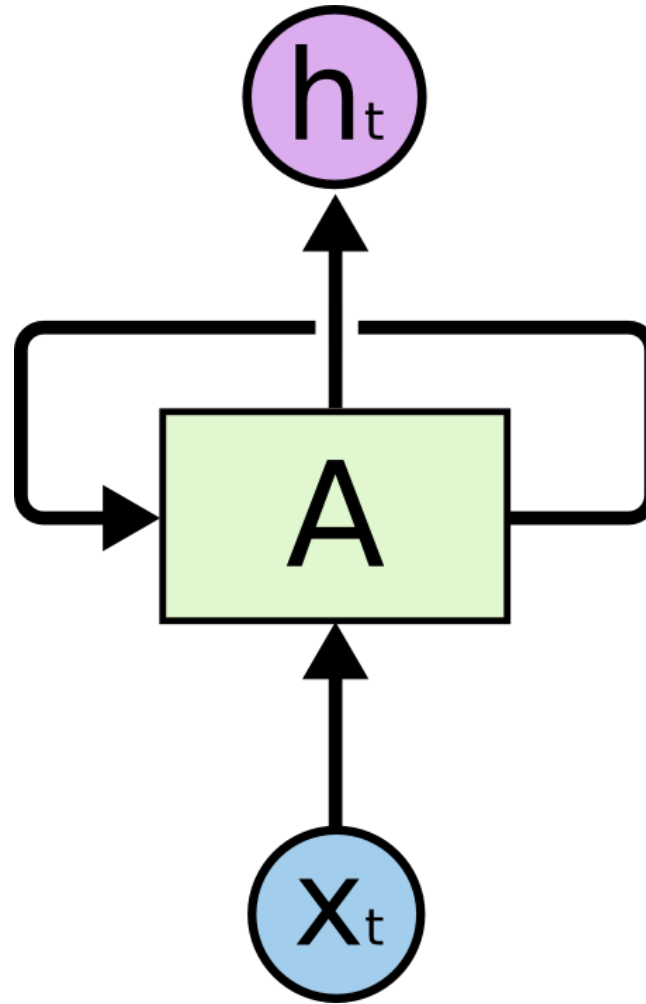
目擊狗案

# ResNet

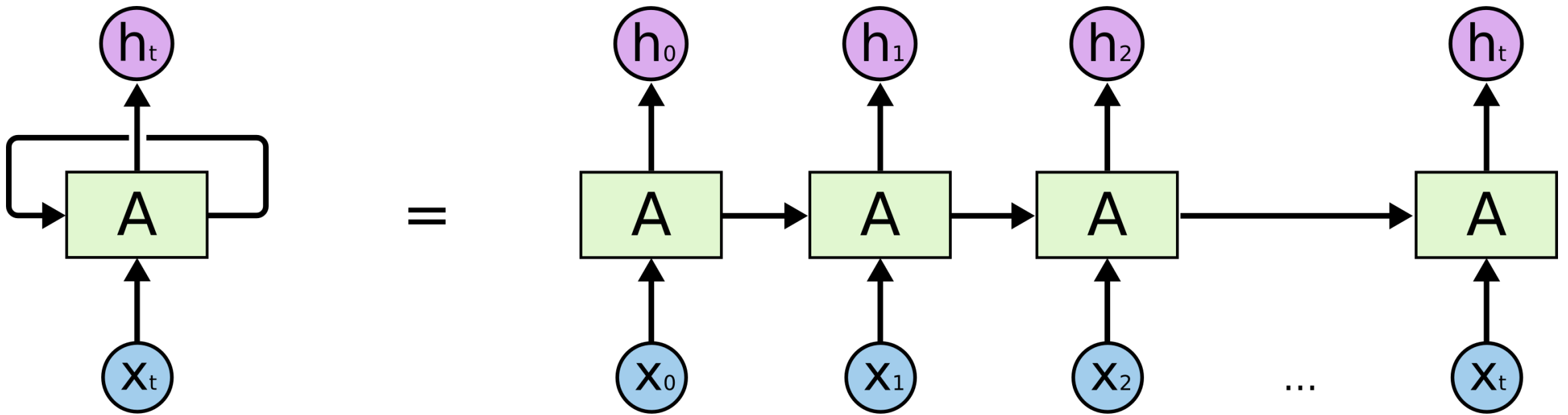




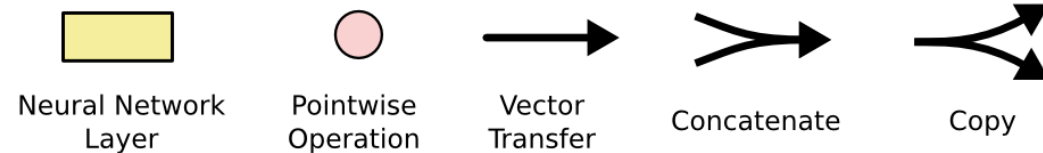
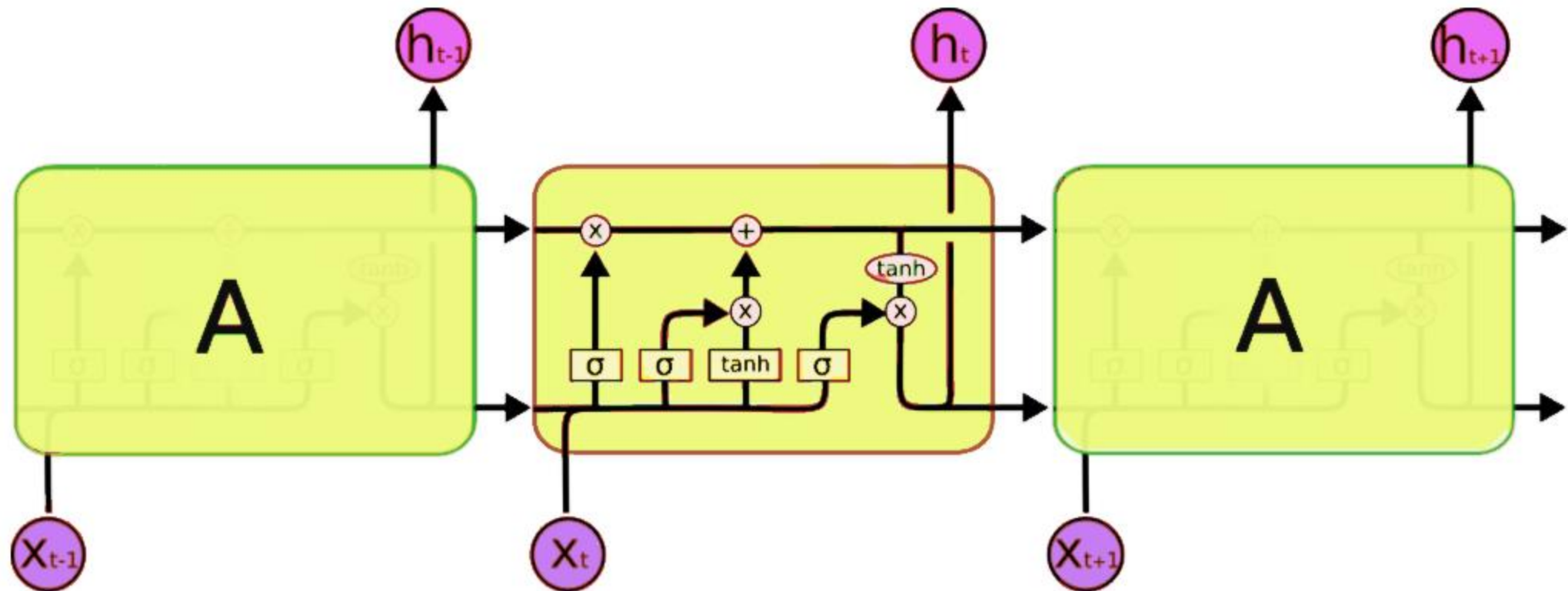
# Recurrent Neural Networks (RNN)



# Unroll the RNN



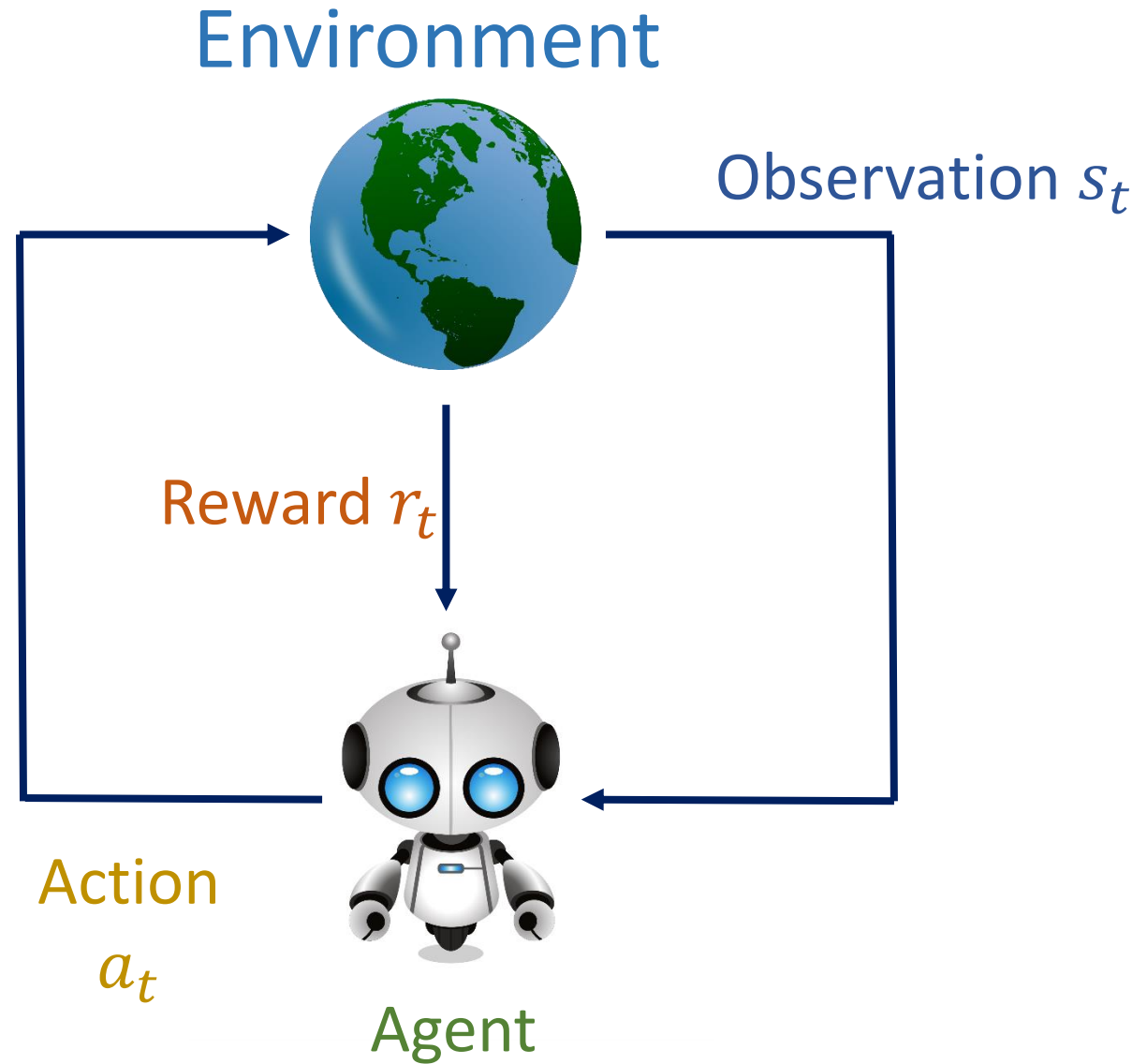
# Long Short-term Memory (LSTM)



# Deep Reinforcement Learning



# Reinforcement Learning

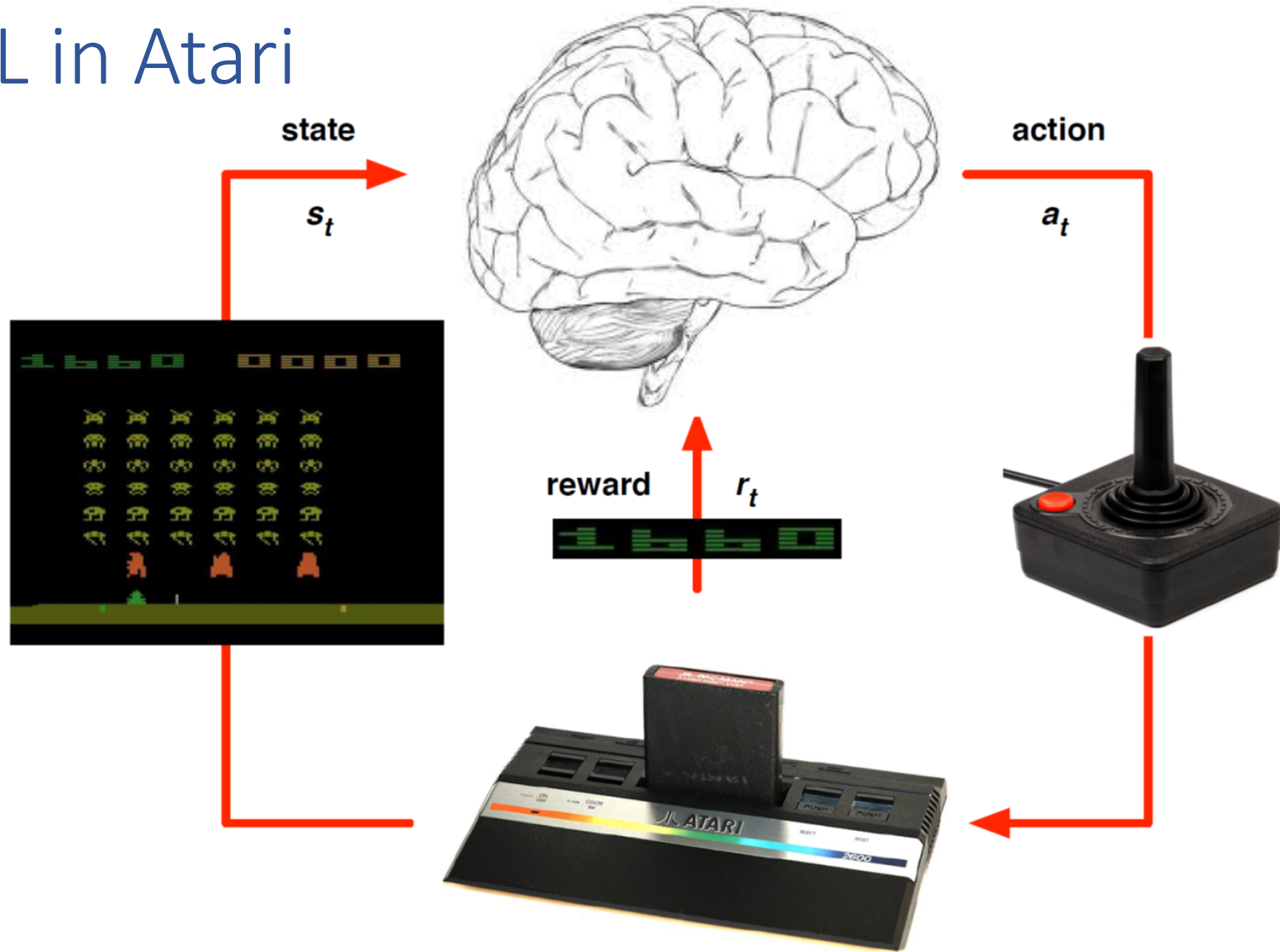




Google DeepMind



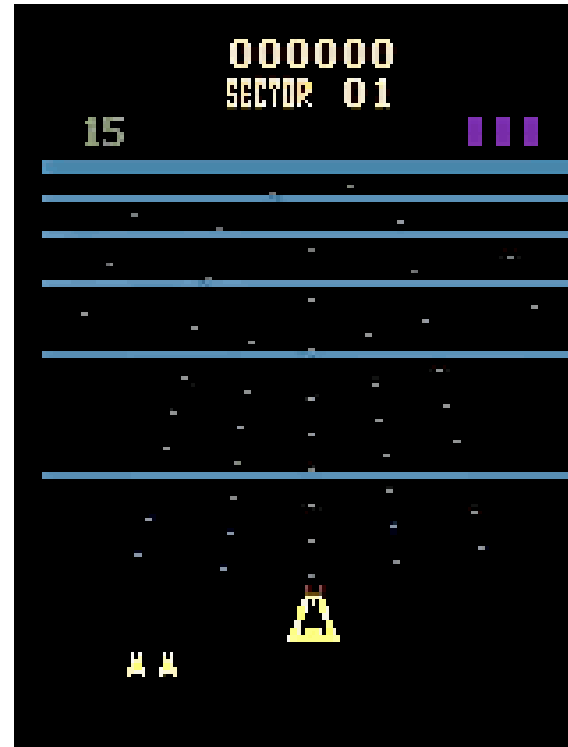
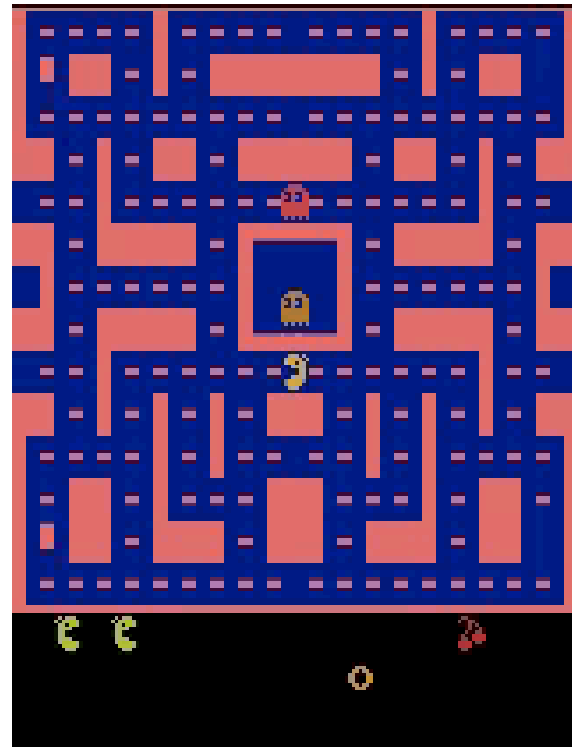
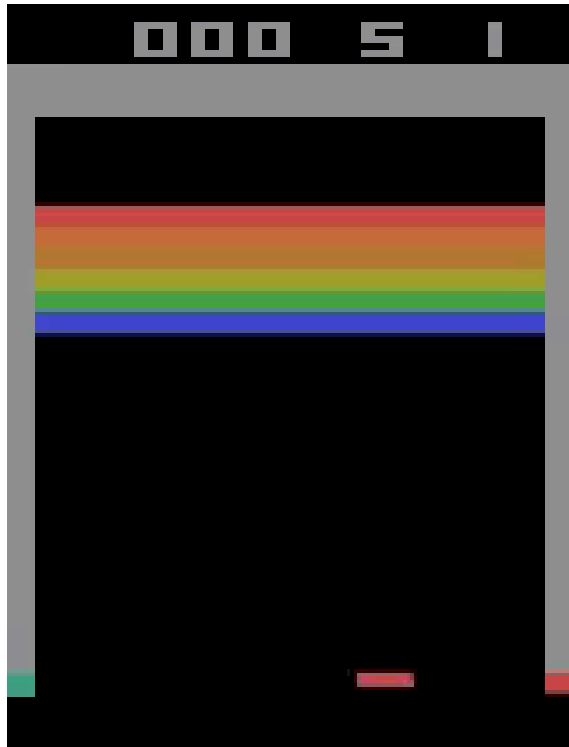
# DRL in Atari



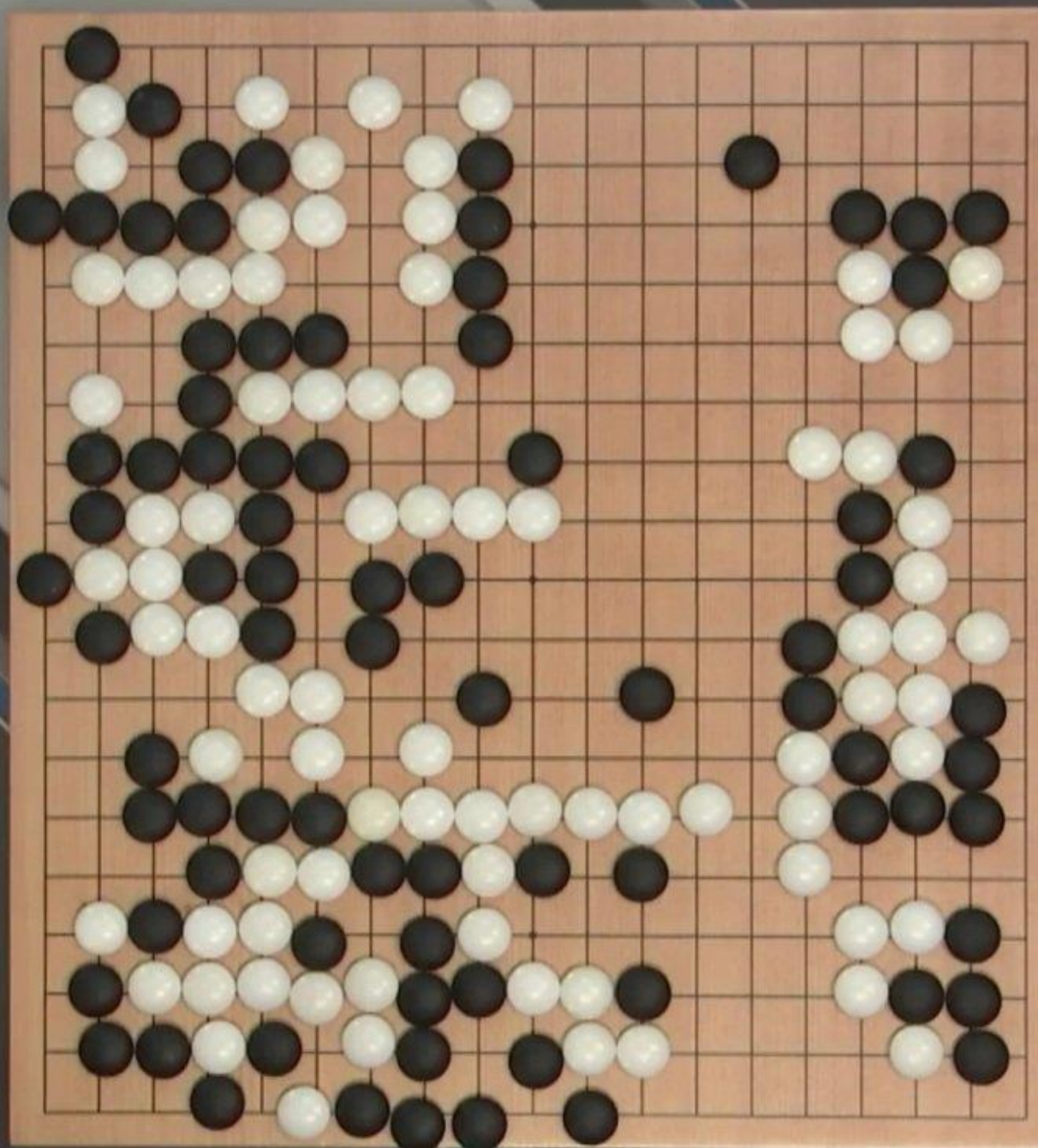
Mnih et al., "Human Level Control through Deep Reinforcement Learning," *Nature*, 2015



# Learning to Play Atari Games



● ALPHAGO  
00:10:29



● LEE SEDOL  
00:01:00



# Dr. Aja Huang (黃士杰)



# The Complexity of Go vs Chess

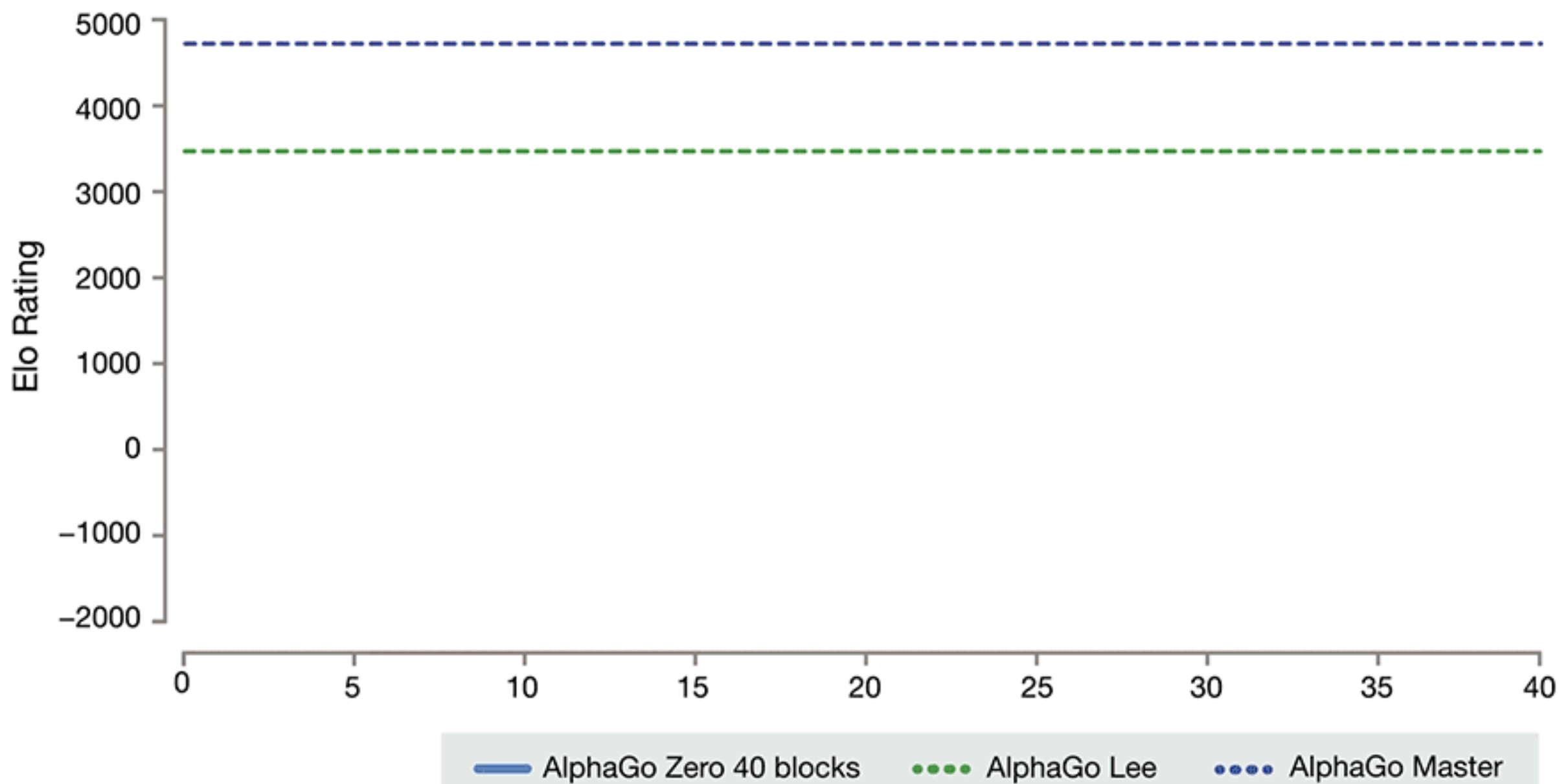
Game	Board size	State space	Game tree size
Go	19 x 19	$10^{172}$	$10^{360}$
Chess	8 x 8	$10^{50}$	$10^{123}$
Checkers	8 x 8	$10^{18}$	$10^{54}$

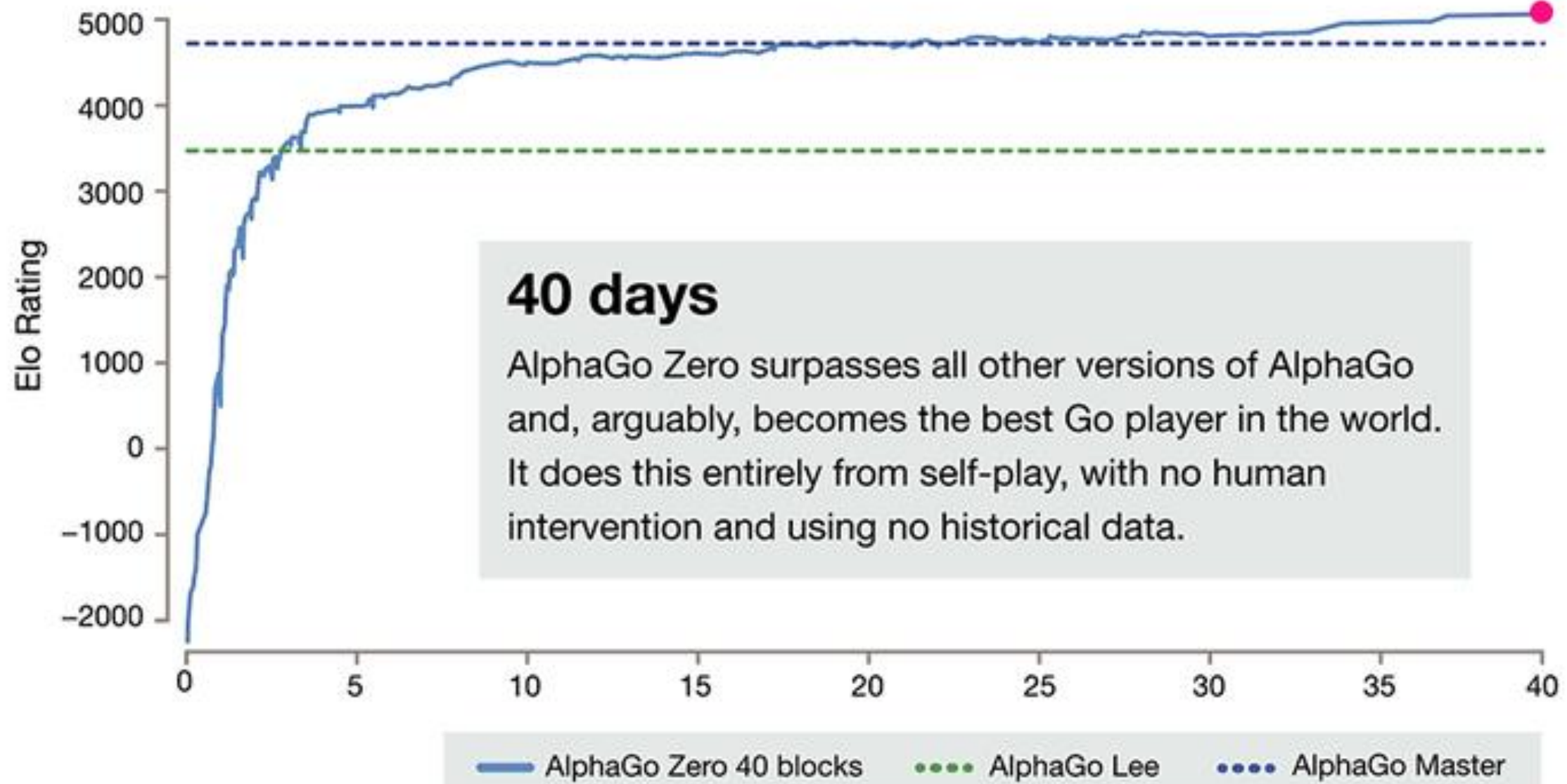


# AlphaGo Zero

Starting from scratch







# Human Extinction ?







# Virtual-to-real Learning

- Inspired by DeepMind (Mnih et al., *Nature*, 2015)
  - “Human Level Control through Deep Reinforcement Learning”
- Applied to computer vision applications
  - **Image segmentation:** Armeni et al. (2016), Qiu et al., (2017)
  - **Indoor navigation:** Brodeur et al. (2017), Gupta et al. (2017), Savva et al. (2017), Wu et al. (2018)
  - **Autonomous vehicles:** Marinez et al. (2017), Muller et al. (2018), Pan et al. (2017), Shah et al. (2018)

UnrealCV



CAD<sup>2</sup>Real

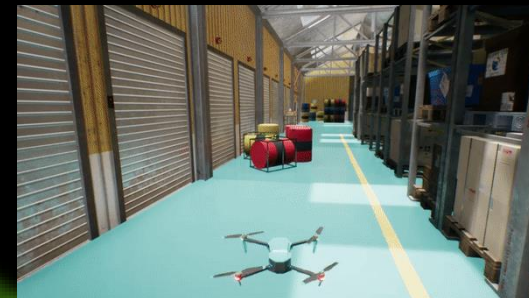


Semantic Segmentation

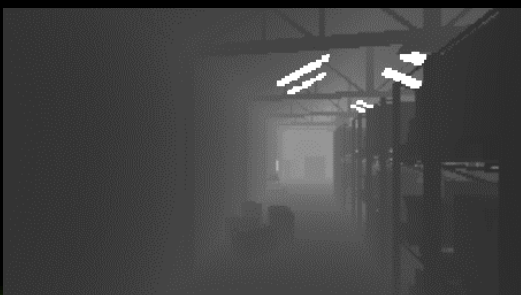


VIVID

Autonomous Navigation



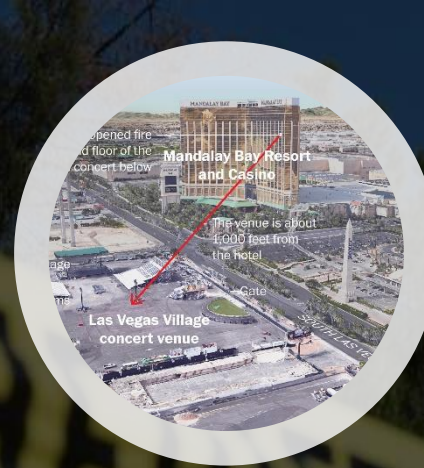
Depth Prediction



Action Recognition



# Simulate Real-life Events



# Searching for the Shooter



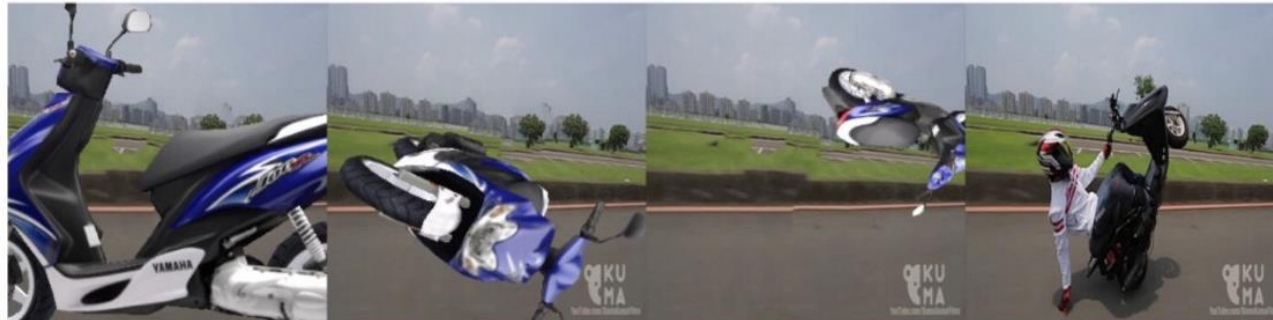
# Limits of Deep Learning



# No Idea of Real World



**school bus** 1.0 **garbage truck** 0.99 **punching bag** 1.0 **snowplow** 0.92



**motor scooter** 0.99 **parachute** 1.0 **bobsled** 1.0 **parachute** 0.54



**fire truck** 0.99 **school bus** 0.98 **fireboat** 0.98 **bobsled** 0.79



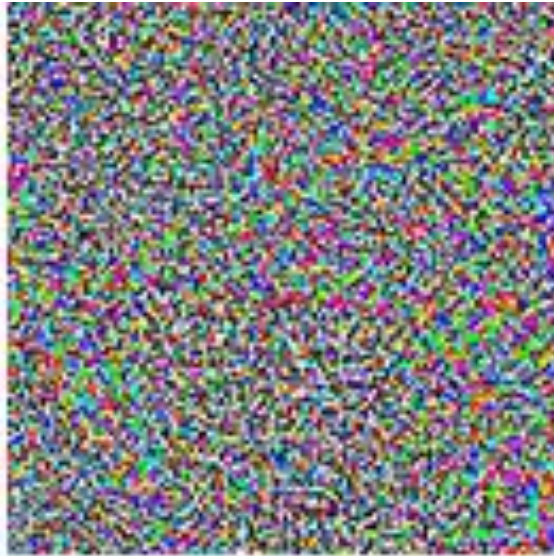
# Adversarial Attack



"panda"

57.7% confidence

+  $\epsilon$



=



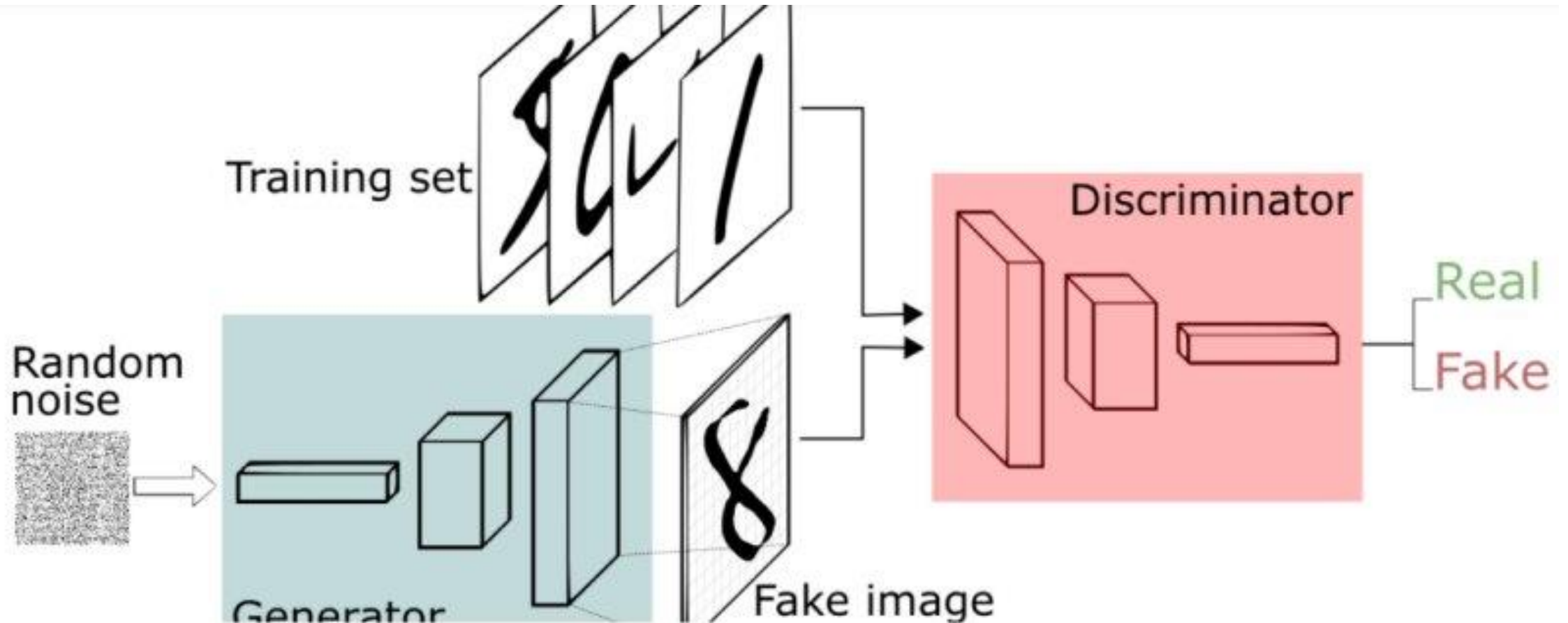
"gibbon" ●

99.3% confidence



# Generative Adversarial Networks (GAN)

- Ian Goodfellow



# Painting like Van Gogh



# Super Resolution

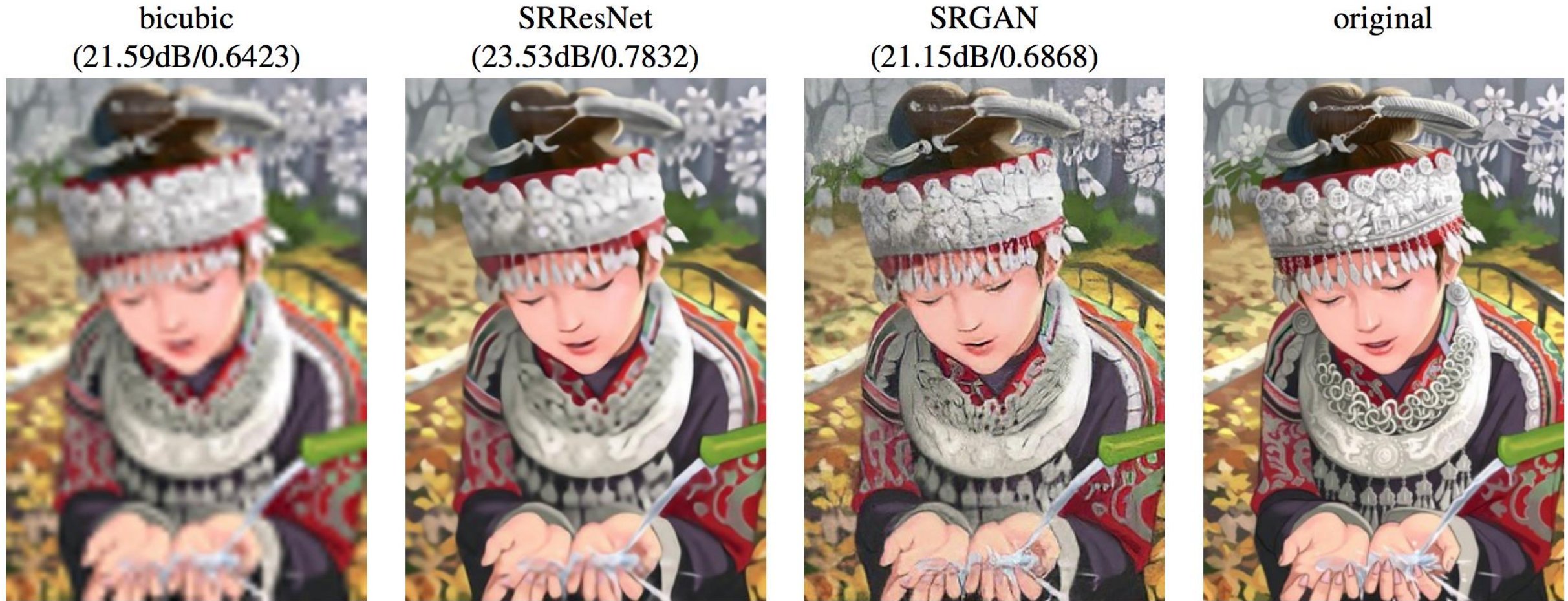
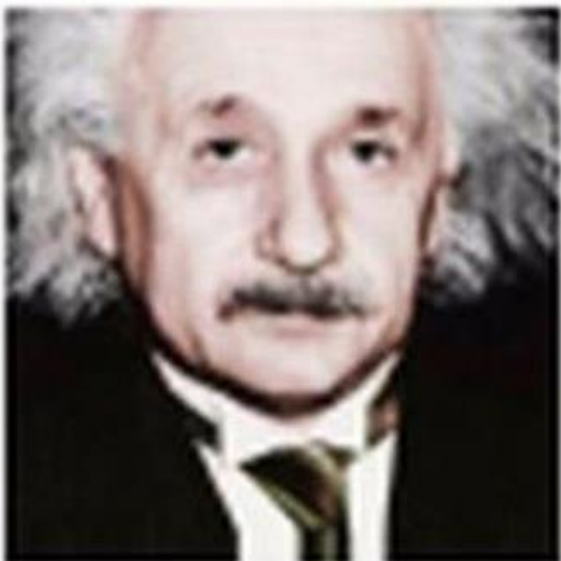


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]



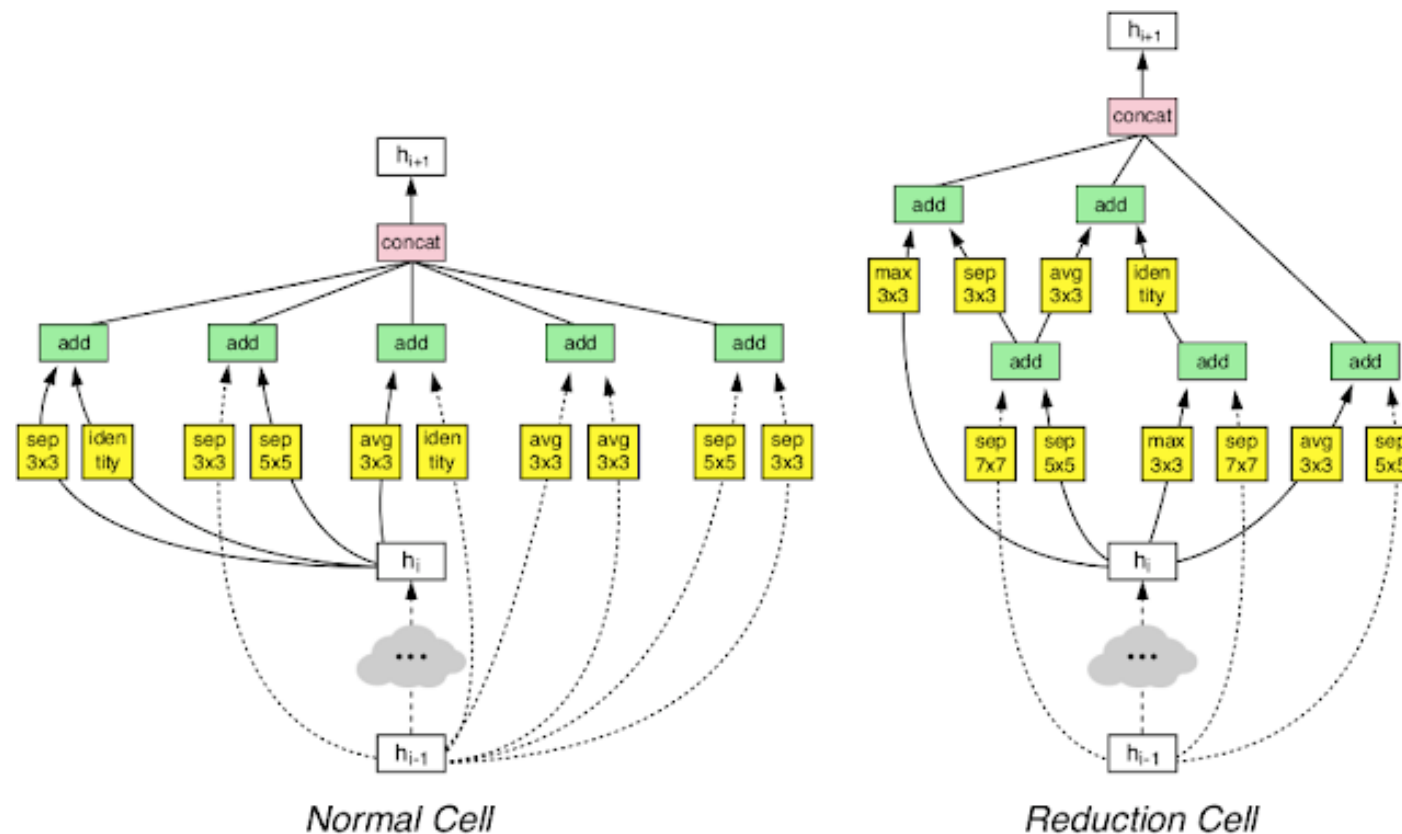


DeepFake: Is this you?

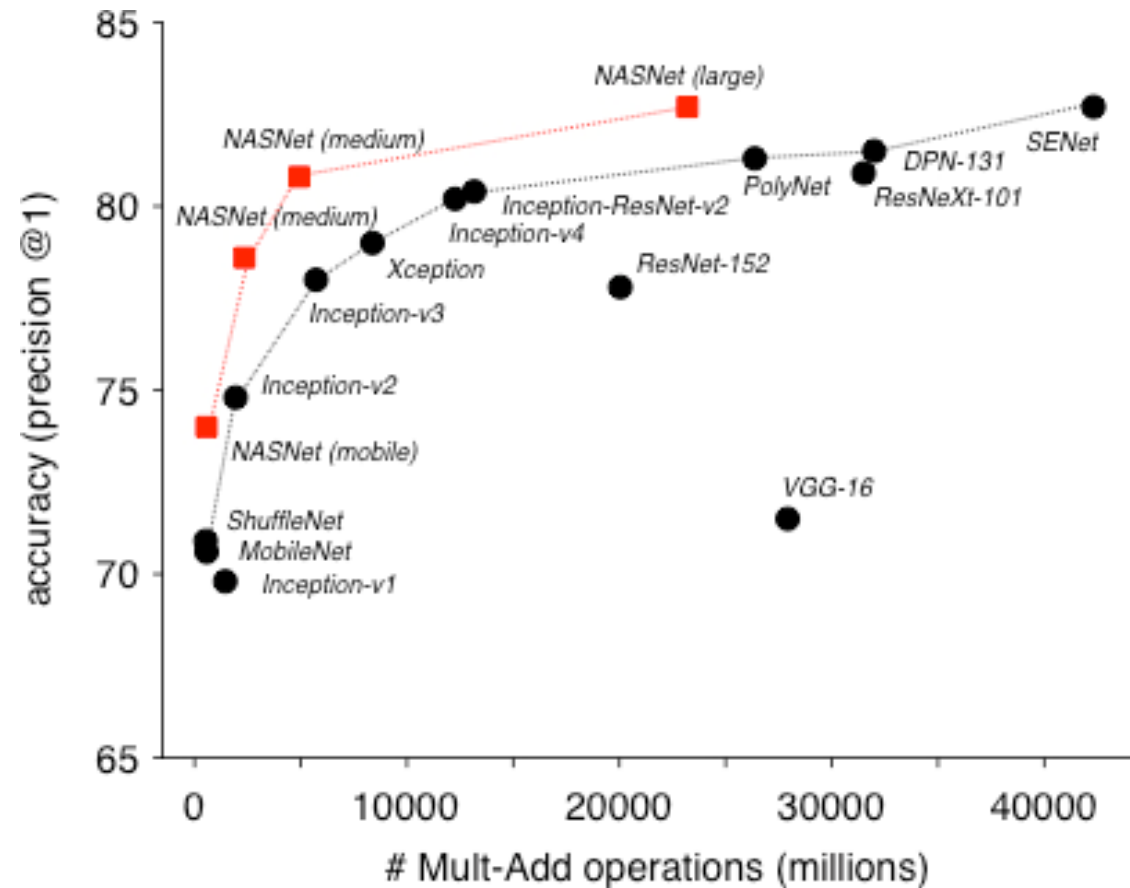


# Google's AutoML

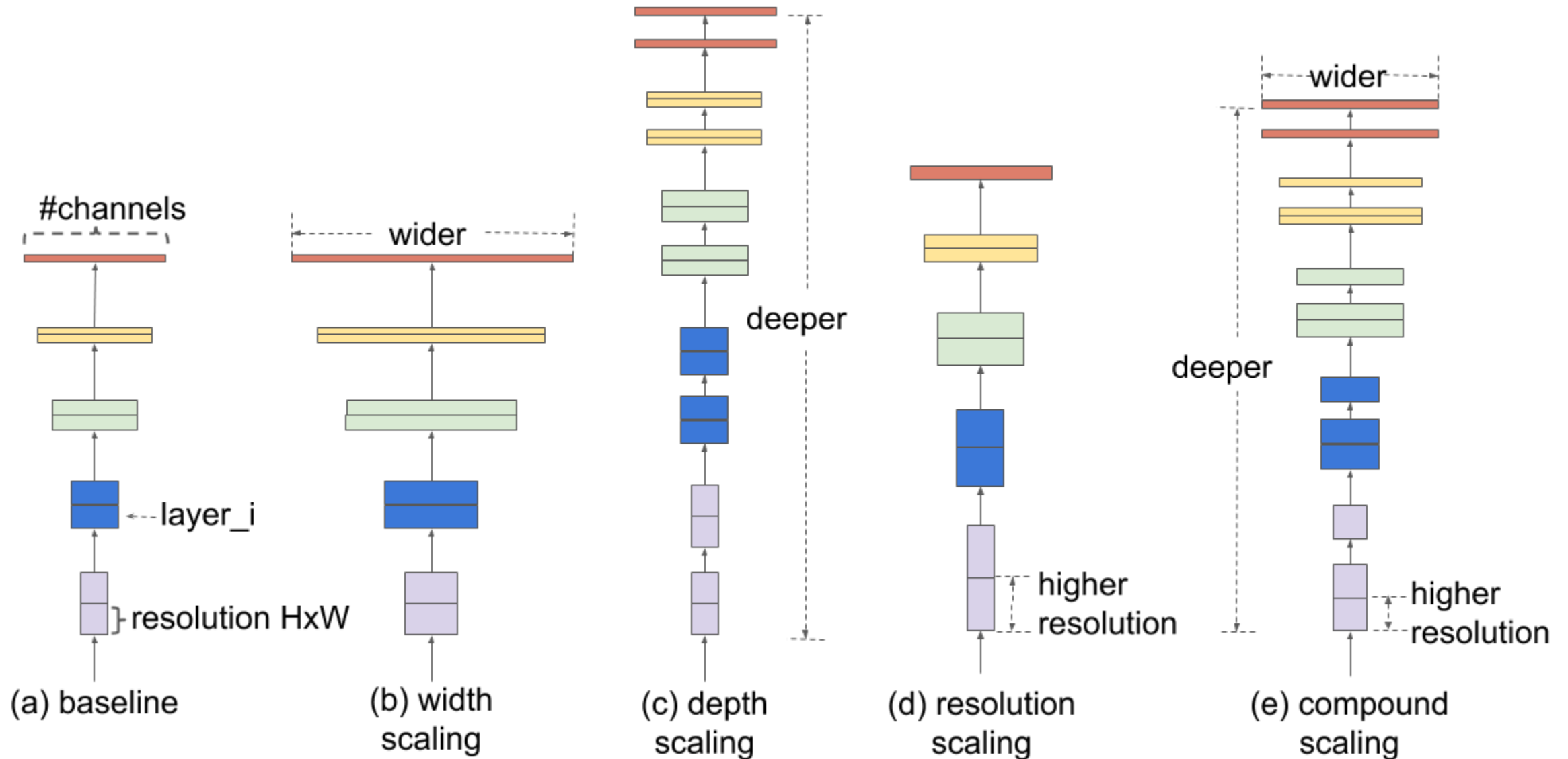
- Learning neural network cells automatically

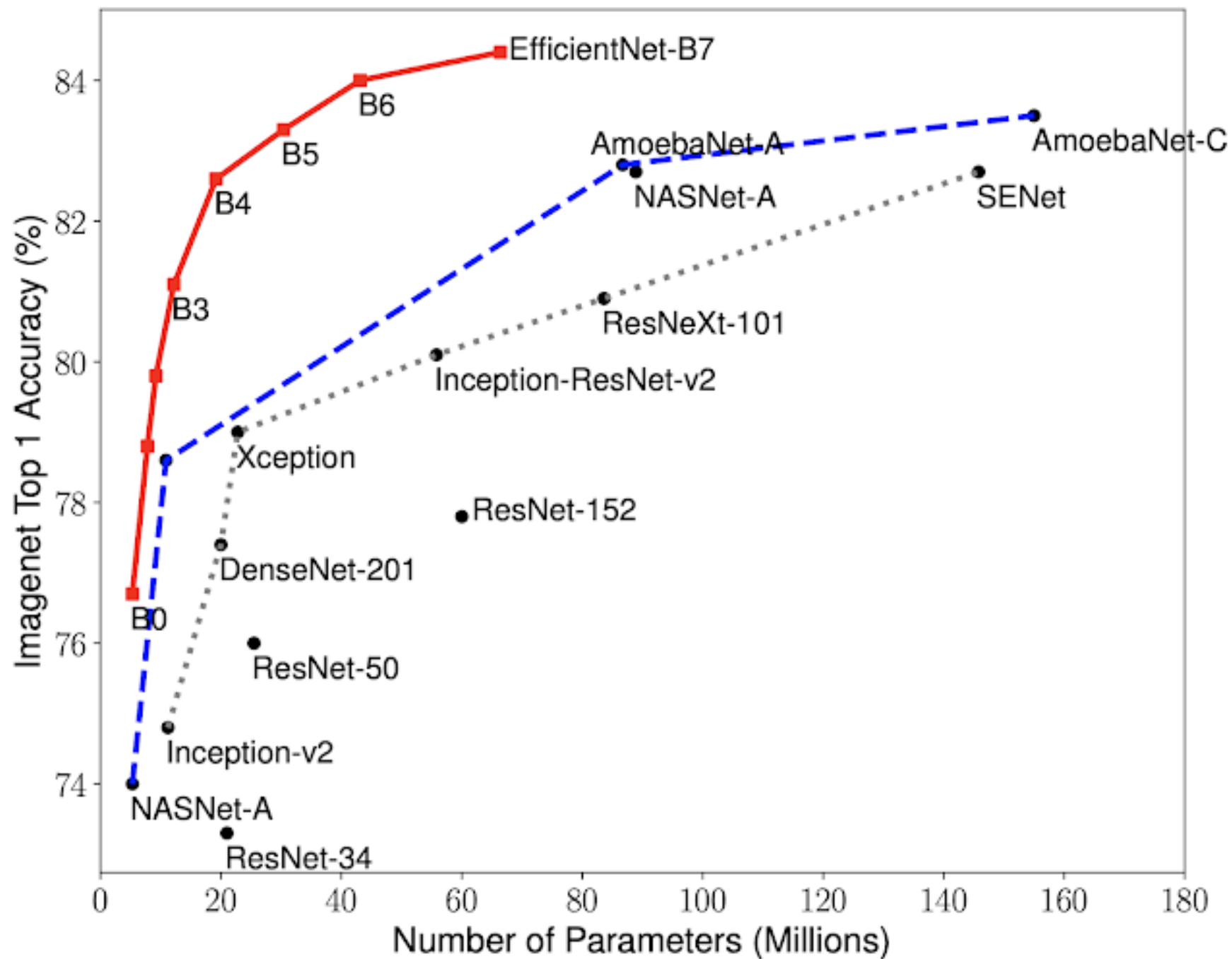


# AutoML on ImageNet



# EfficientNet (May, 2019)





# References

1. <https://www.buzzfeed.com/kasiagalazka/science-fiction-things-that-actually-exist-now>
2. <https://www.geek.com/movies/10-movies-that-helped-create-real-technology-1740036/>
3. <https://www.gadgetsnow.com/slideshows/8-sci-fi-movie-technologies-that-are-real-now/Video-calling/photolist/52869590.cms>
4. What is backpropagation really doing?  
<https://www.youtube.com/watch?v=llg3gGewQ5U>
5. <http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>
6. <https://pmirla.github.io/2016/08/16/AI-Winter.html>
7. <https://tw.saowen.com/a/6cdc2f1279016e566832bb1234e06d321992dd1fabcd4a2e0a3e16fc0dc09dc>