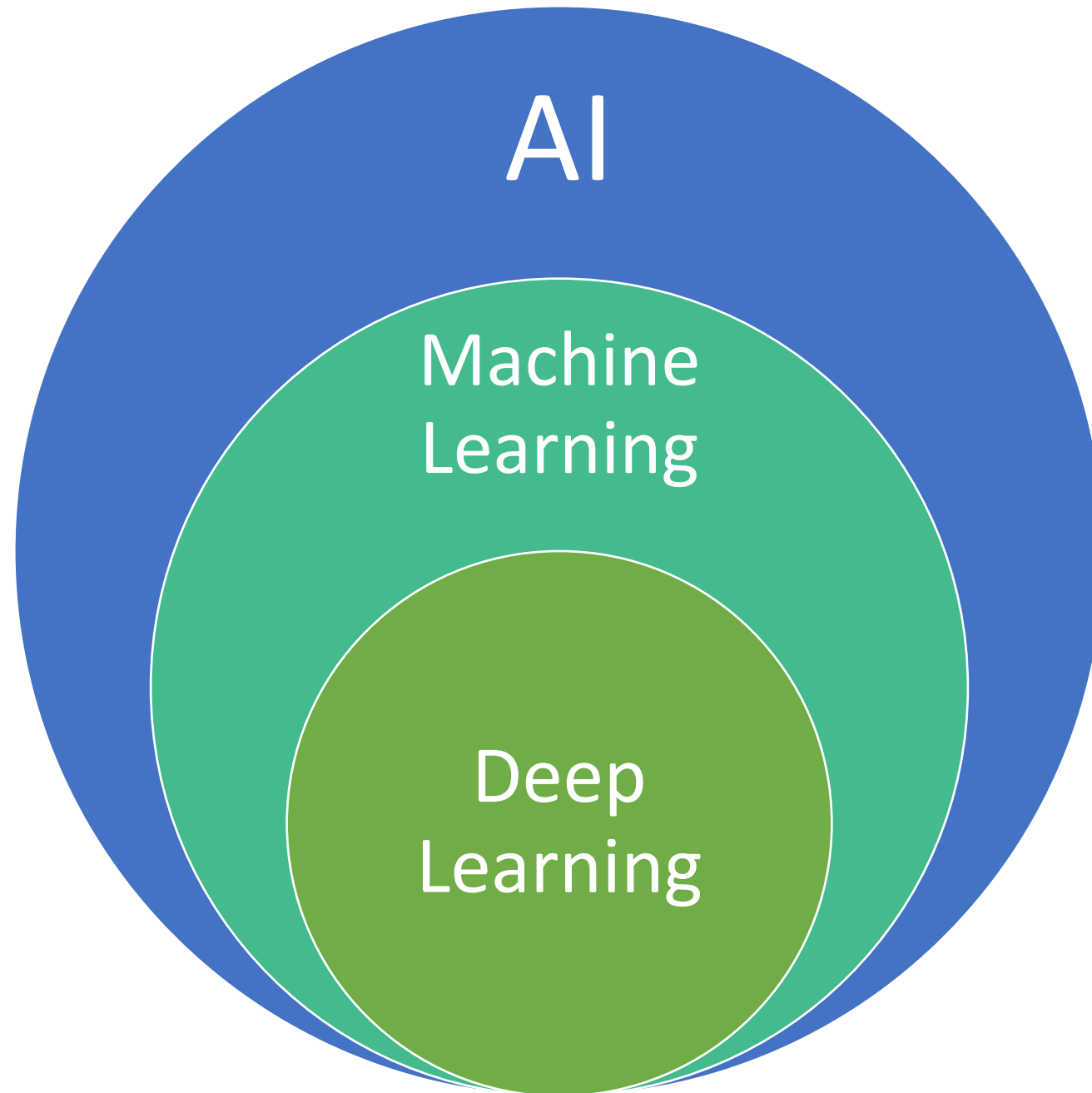


# Introduction to Deep Learning

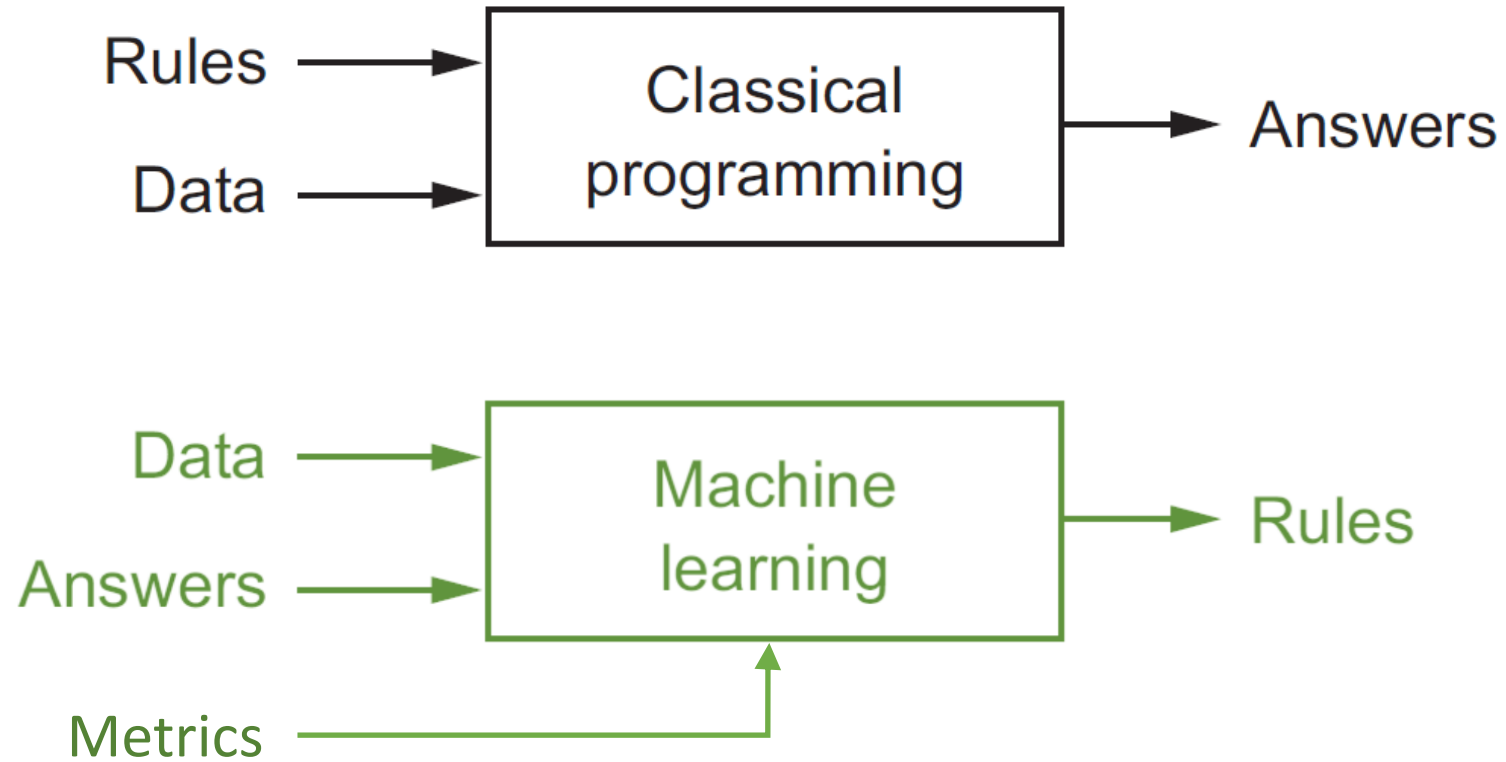
Prof. Kuan-Ting Lai

2021/9/28



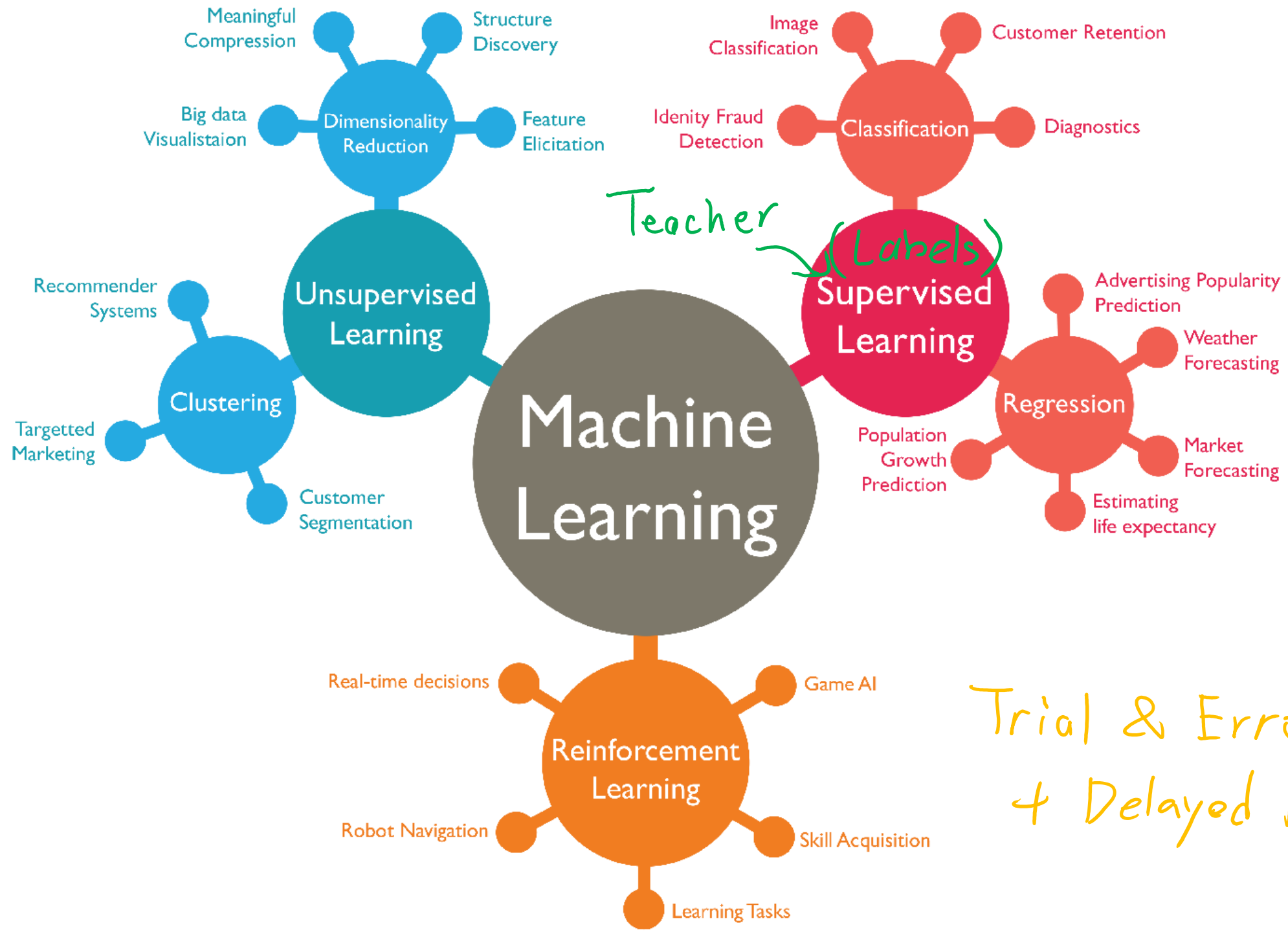


# Machine Learning (Statistical Learning)



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

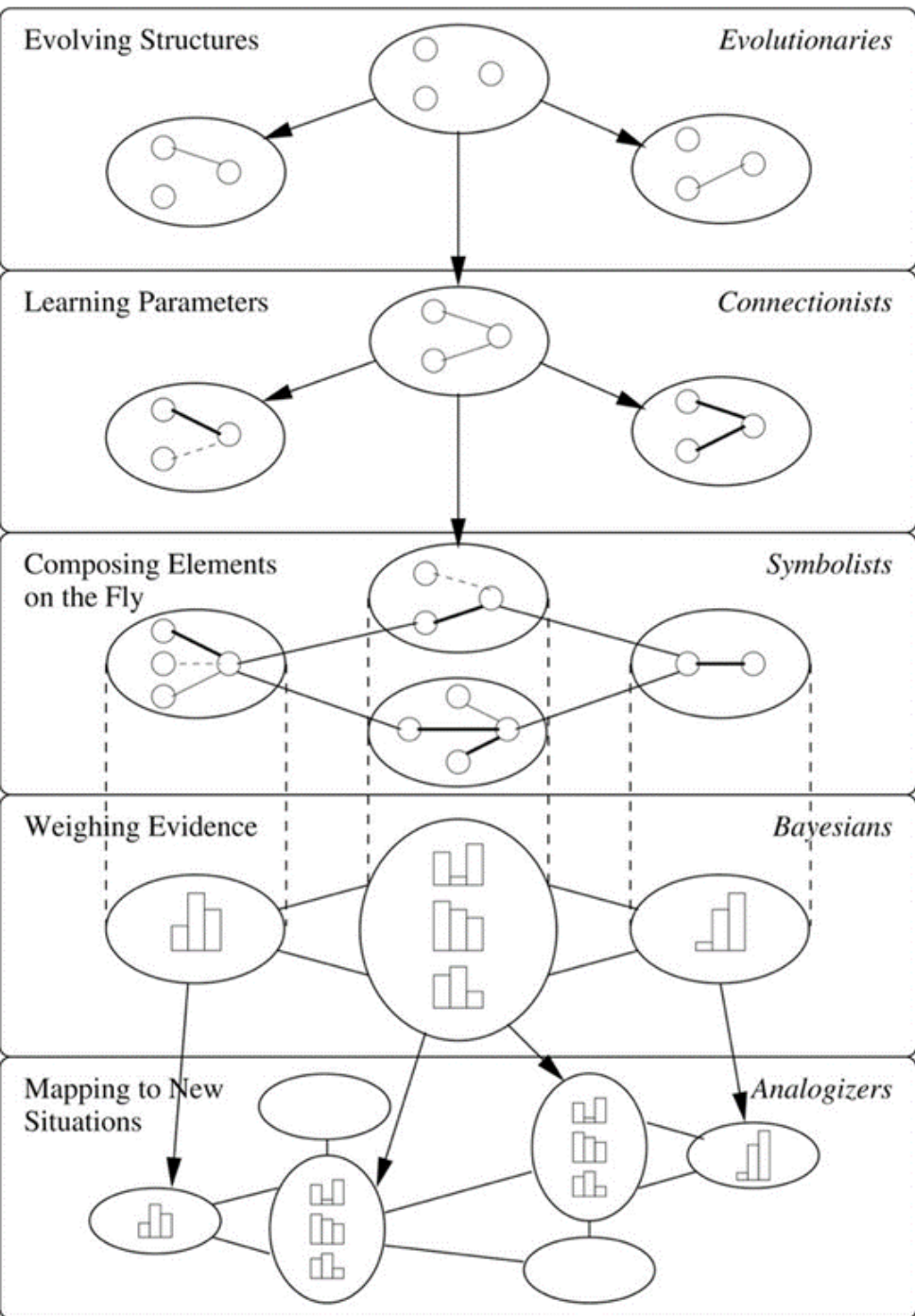




Trial & Error  
+ Delayed Reward





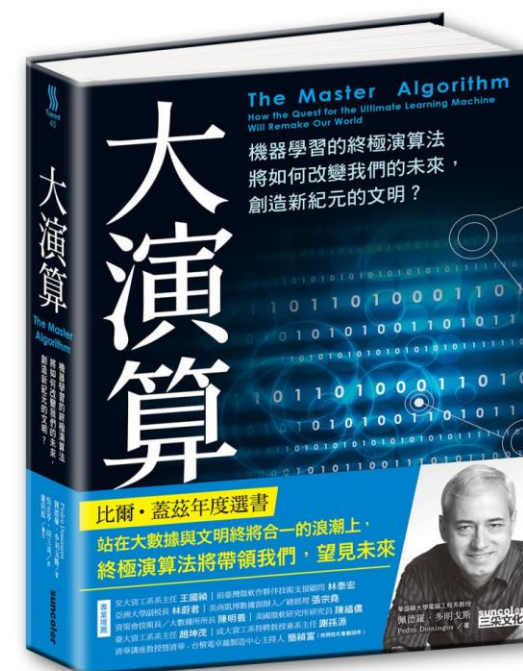
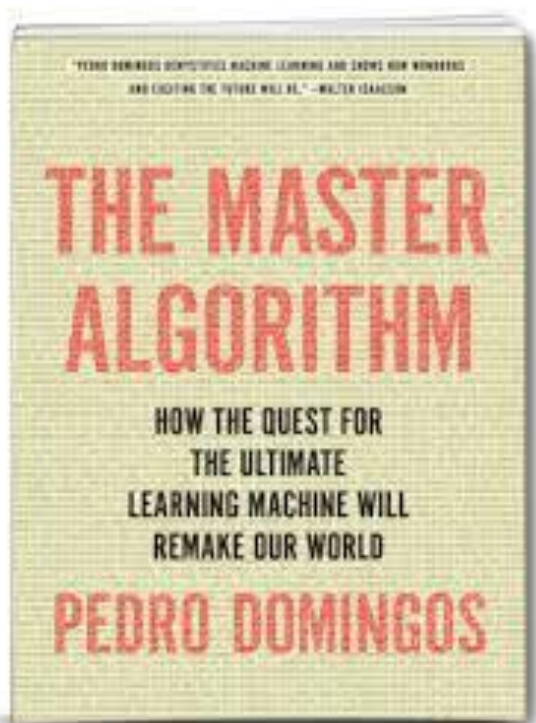


# Five Tribes of Machine Learning

- Evolutionaries
- Connectionists
- Symbolists
- Bayesians
- Analogizers

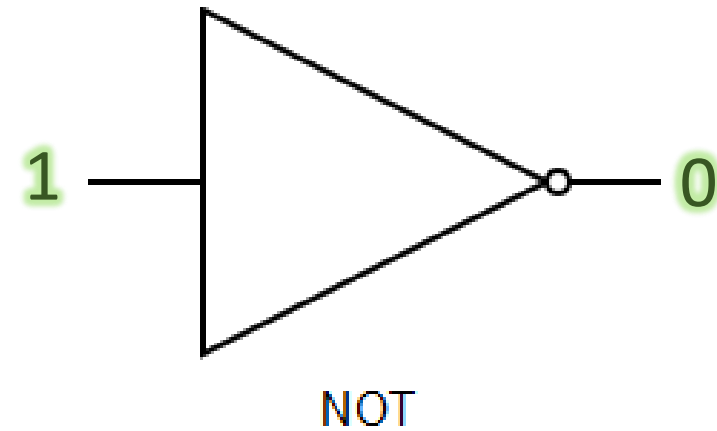
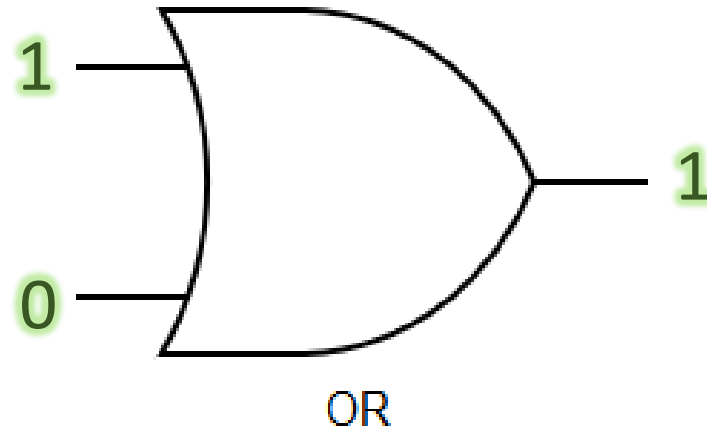
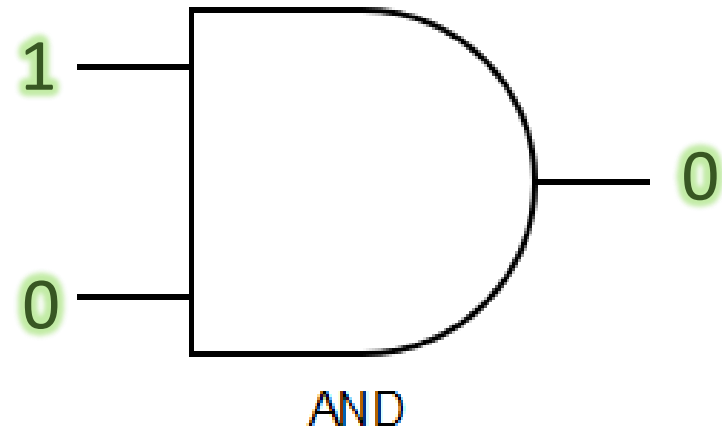


# The Master Algorithm – Pedro Domingos

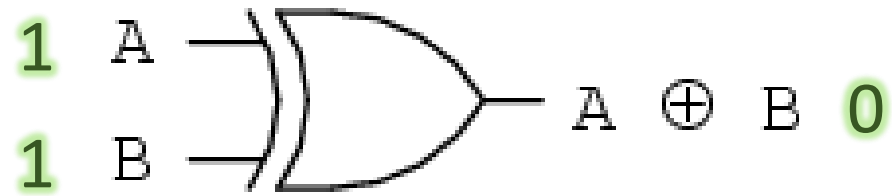


# 3 Basic Operations of Algorithms

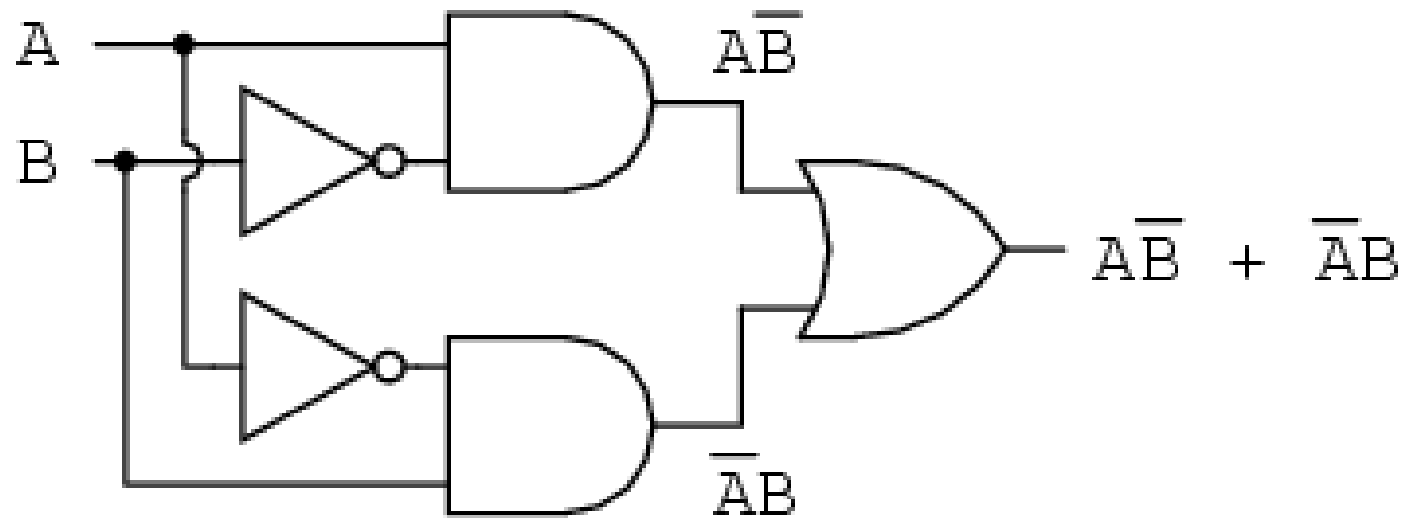
- All Algorithms can be Reduced to 3 Operations



# XOR



... is equivalent to ...



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

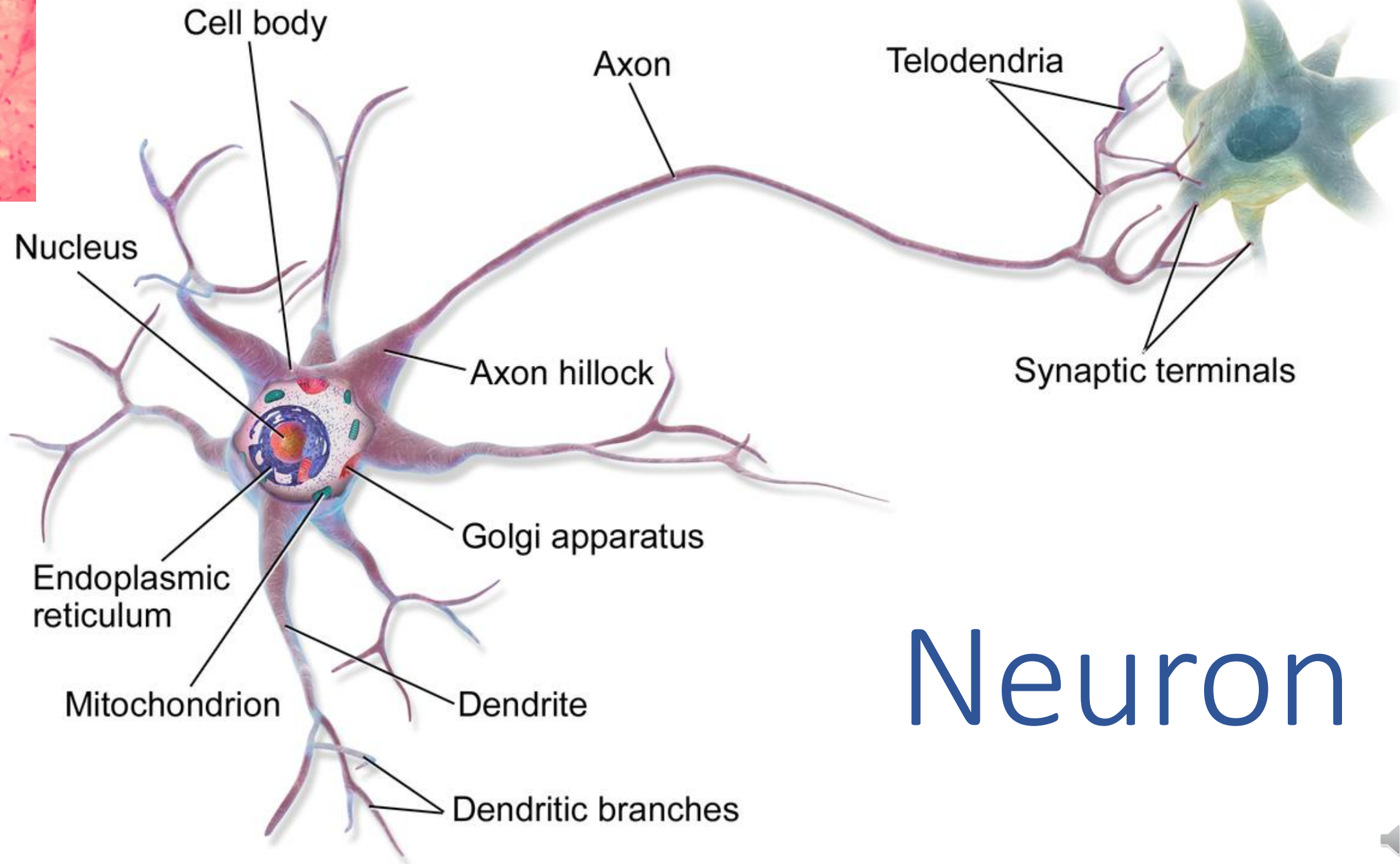
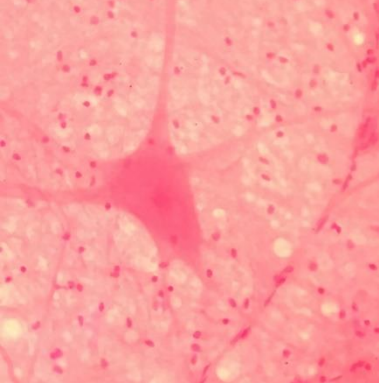




# Neural Networks

The background of the slide is a complex, artistic representation of a neural network. It features several large, blue, star-shaped neurons with multiple branching processes. These neurons are interconnected by a dense web of thin, light-blue lines. Scattered throughout this network are numerous small, glowing orange-yellow dots, which represent nodes or data points. The overall color palette is dominated by deep blues and vibrant oranges, creating a high-tech, futuristic aesthetic. The text 'Neural Networks' is centered over this background.





# Neuron



# Number of Connections in the Brain

**Neurons (for adults):**

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

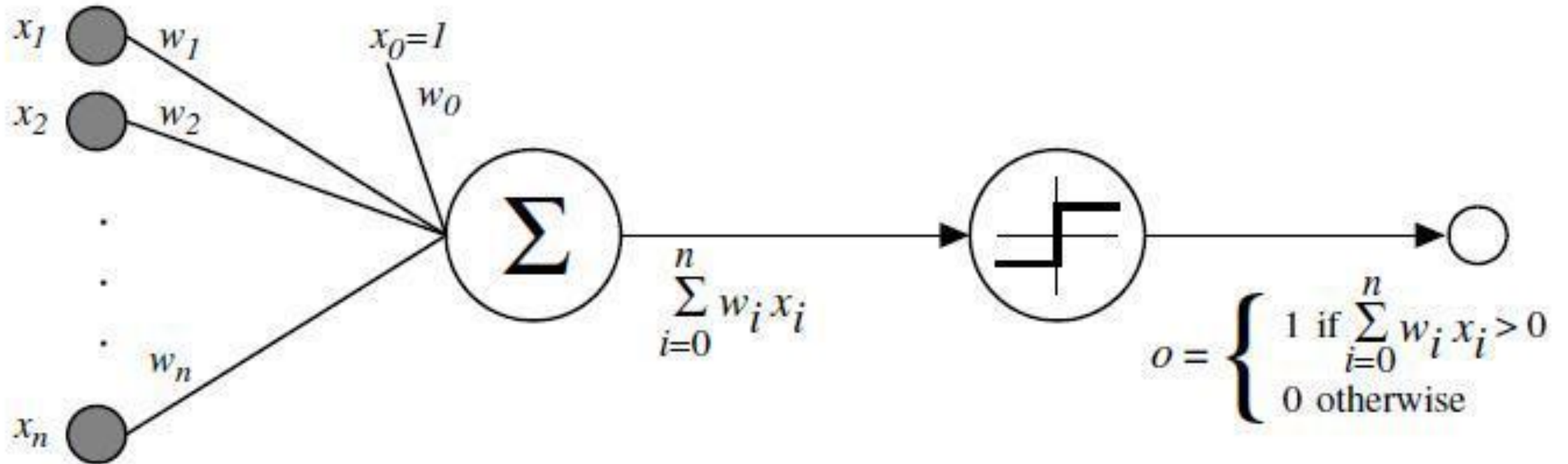
**Synapses (based on 1000 per neuron):**

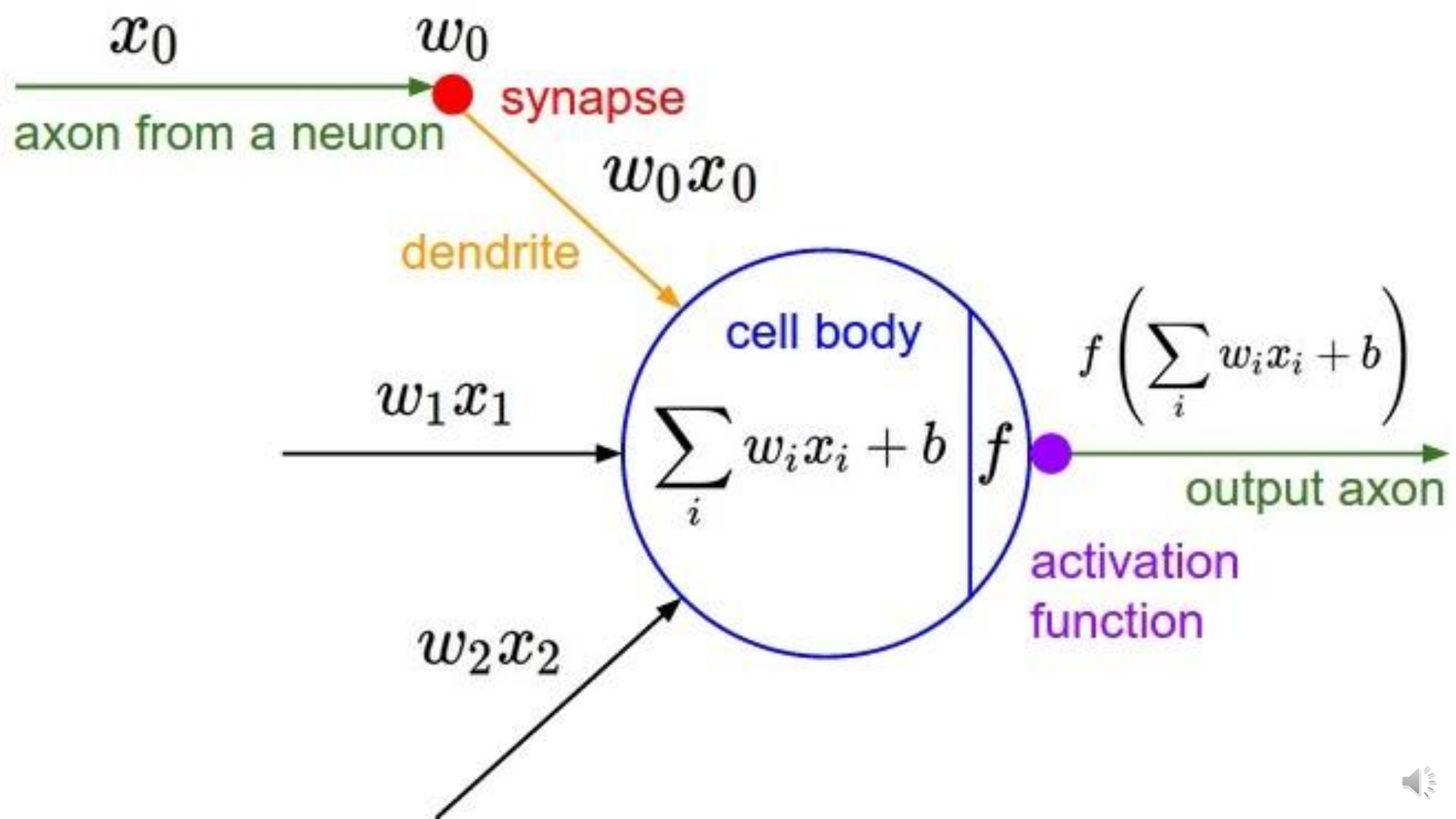
**$10^{14}$ , or 100 trillion, 100000000000000**





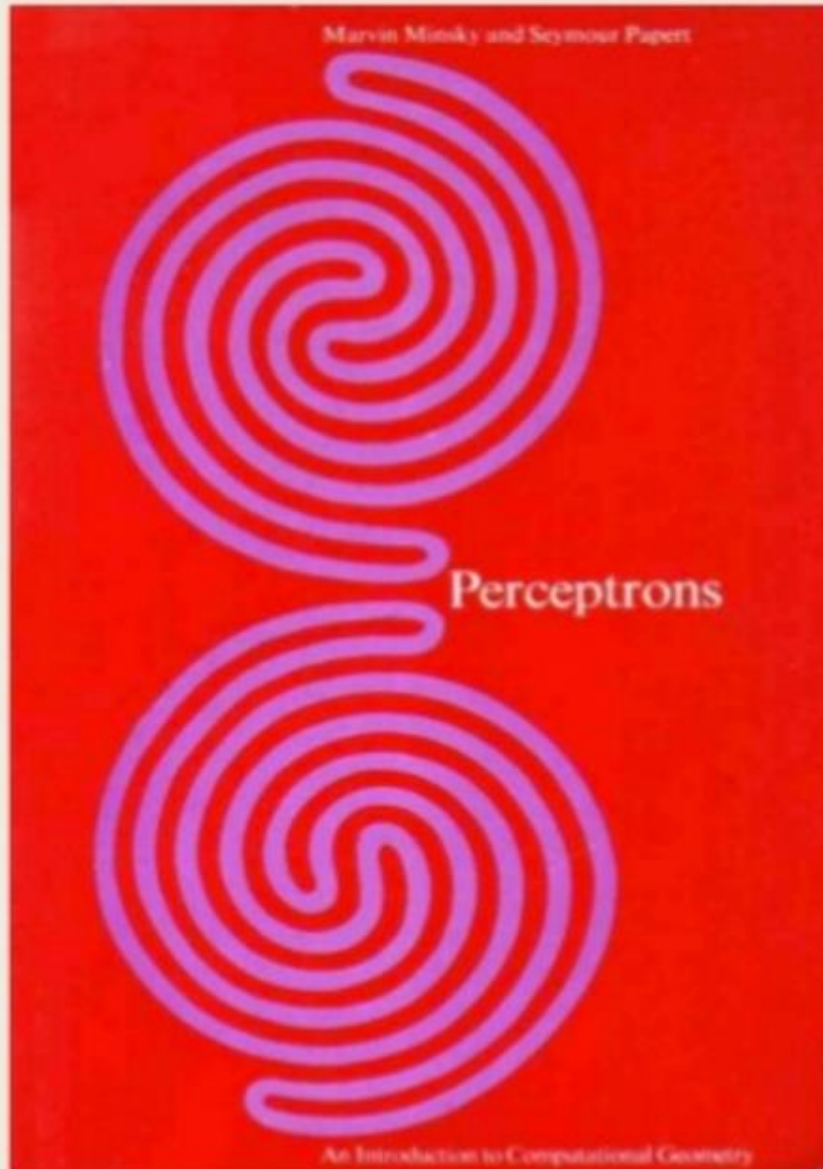
# Frank Rosenblatt's Perceptron (1957)



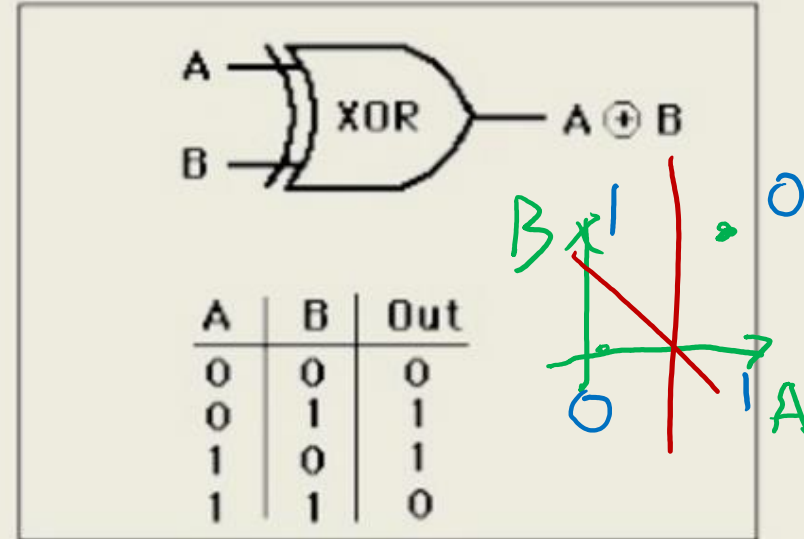




# 1969: Perceptrons can't do XOR!



<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)



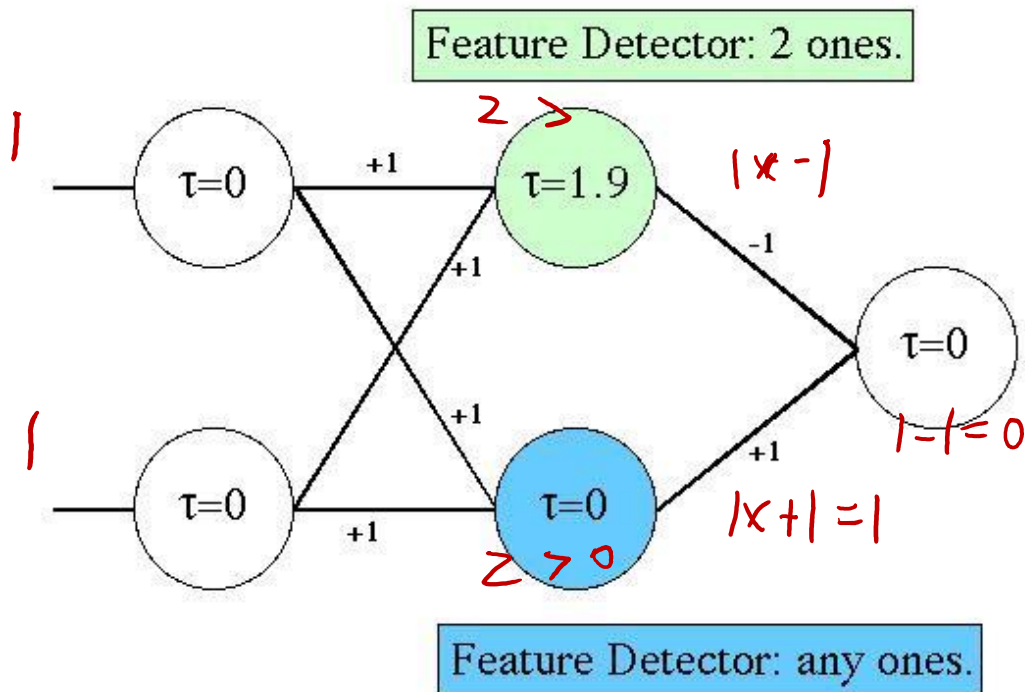




# Learning XOR (1986)

Geoffrey Hinton

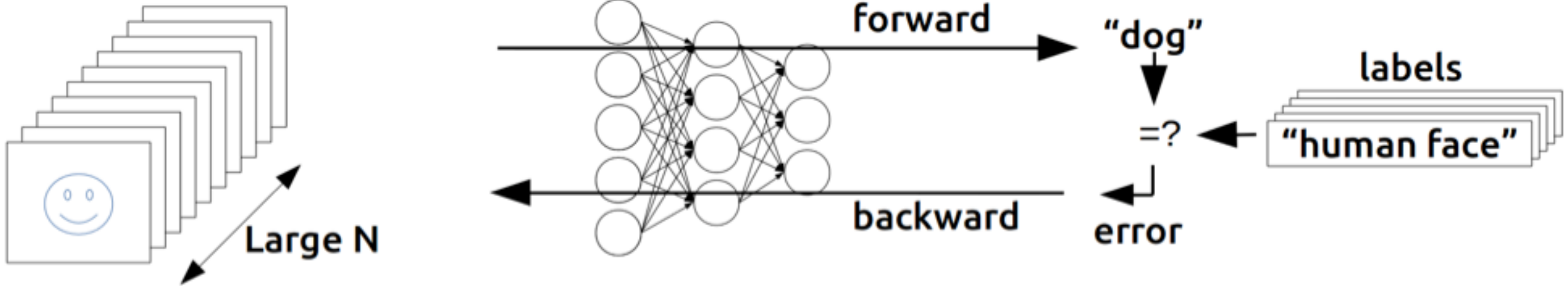
## XOR Network





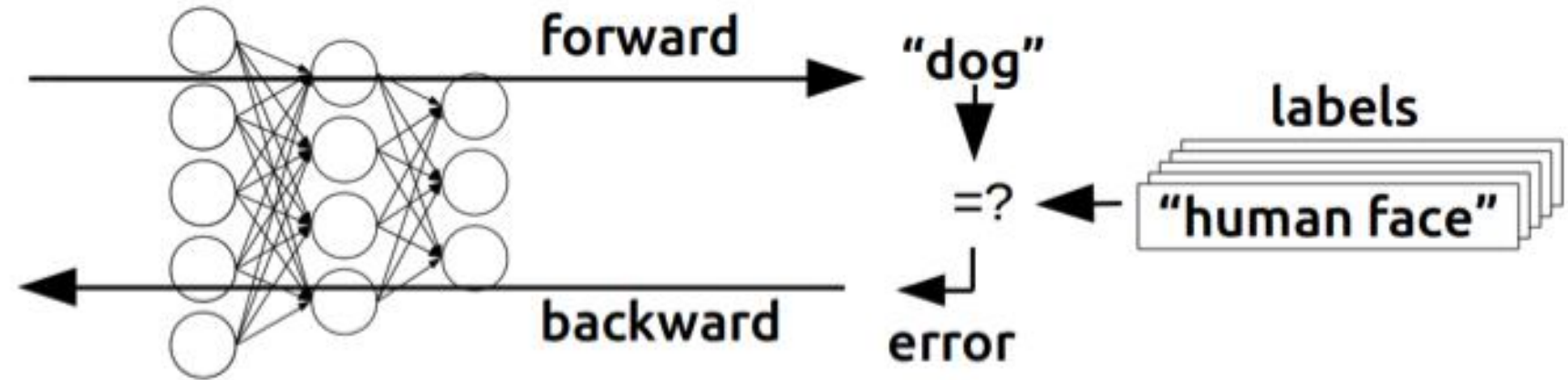
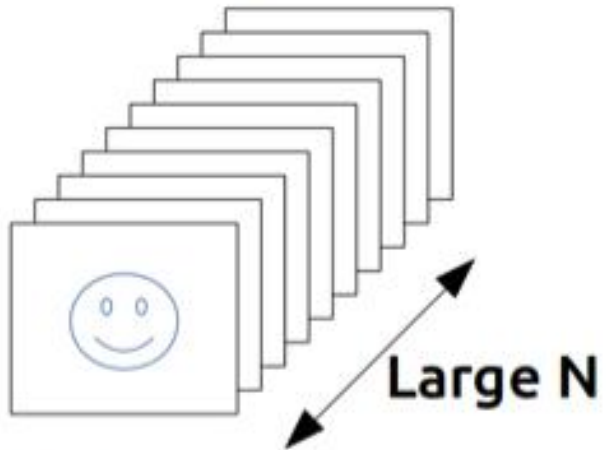
# Backpropagation

## Training

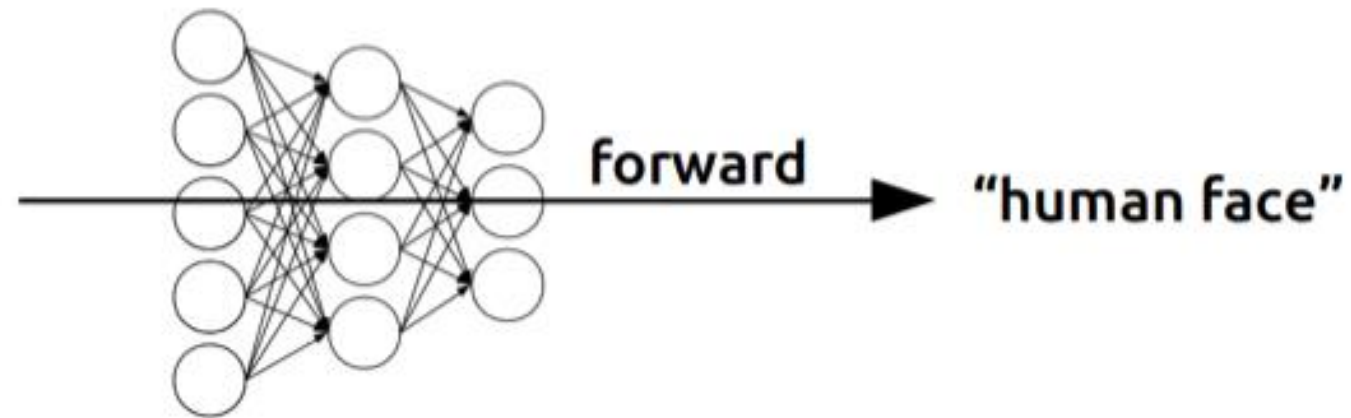
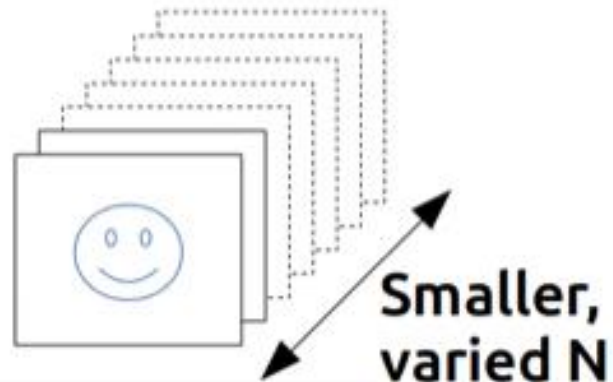


# Inference

## Training



## Inference



# Chain Rule

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

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

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

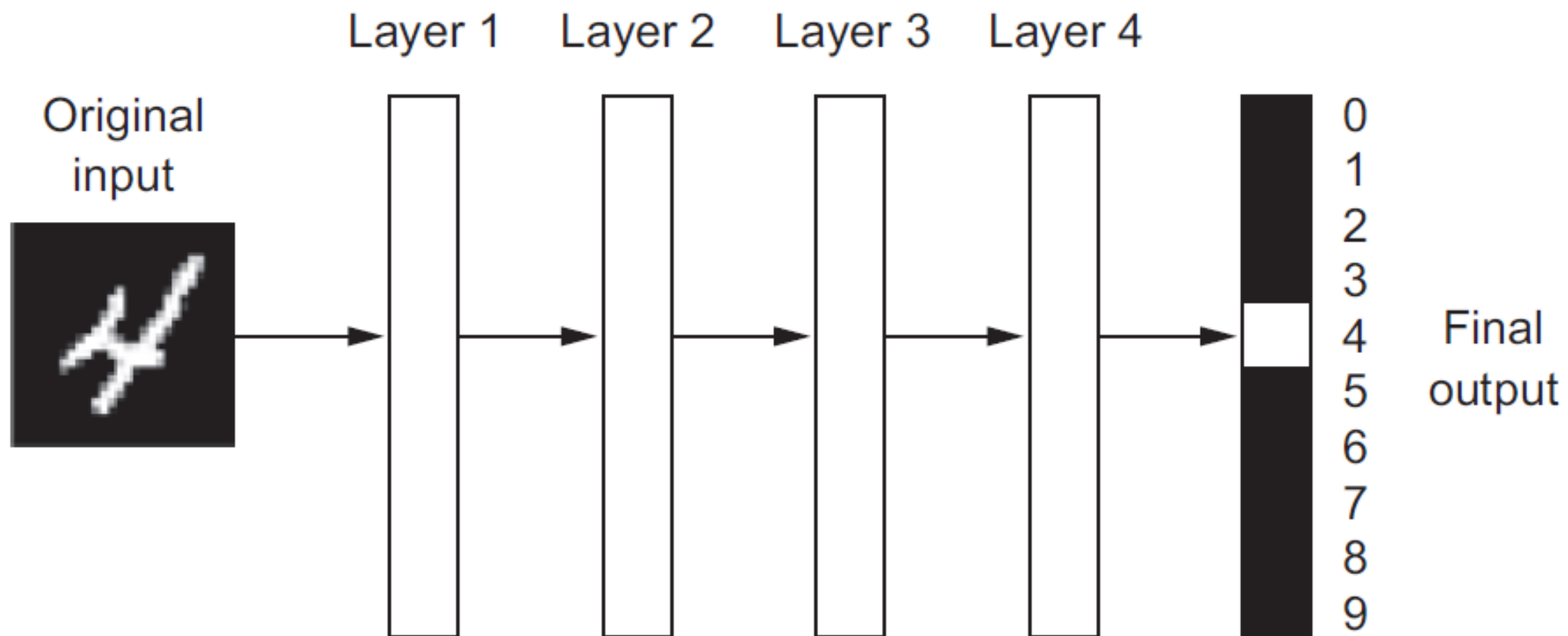
$$\frac{d^4 y}{dx^4} = \frac{d^4 y}{du^4} \left( \frac{du}{dx} \right)^4 + 6 \frac{d^3 y}{du^3} \left( \frac{du}{dx} \right)^2 \frac{d^2 u}{dx^2} + \frac{d^2 y}{du^2} \left( 4 \frac{du}{dx} \frac{d^3 u}{dx^3} + 3 \left( \frac{d^2 u}{dx^2} \right)^2 \right) + \frac{dy}{du} \frac{d^4 u}{dx^4}.$$



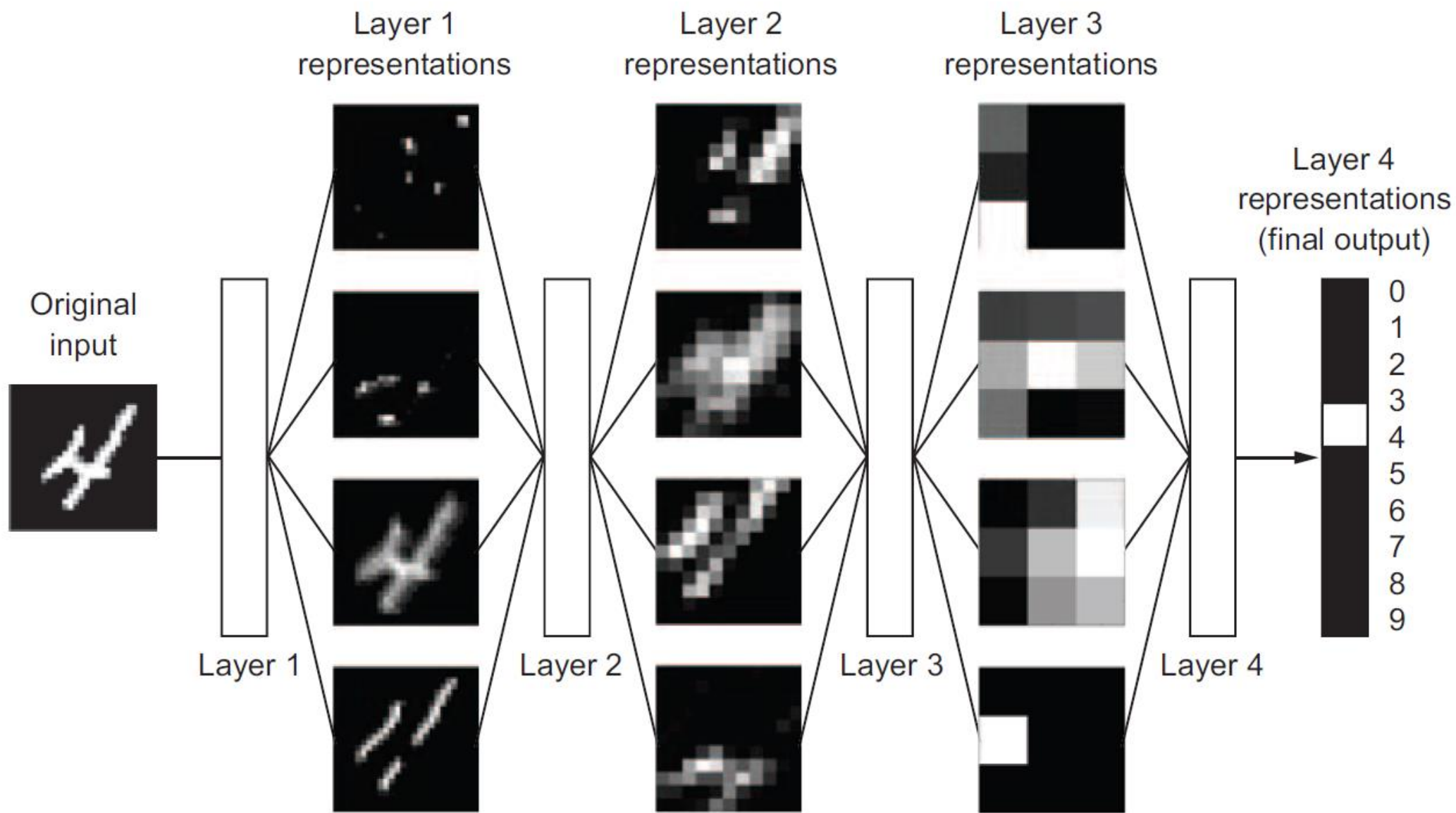
# Example: Recognizing Handwritten Digits

- MNIST dataset

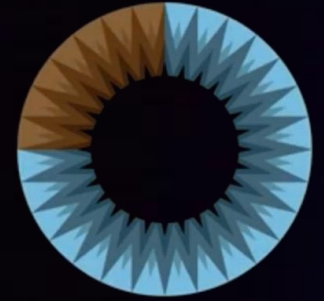








# Grant Sanderson

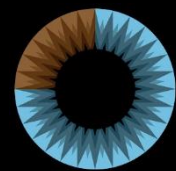
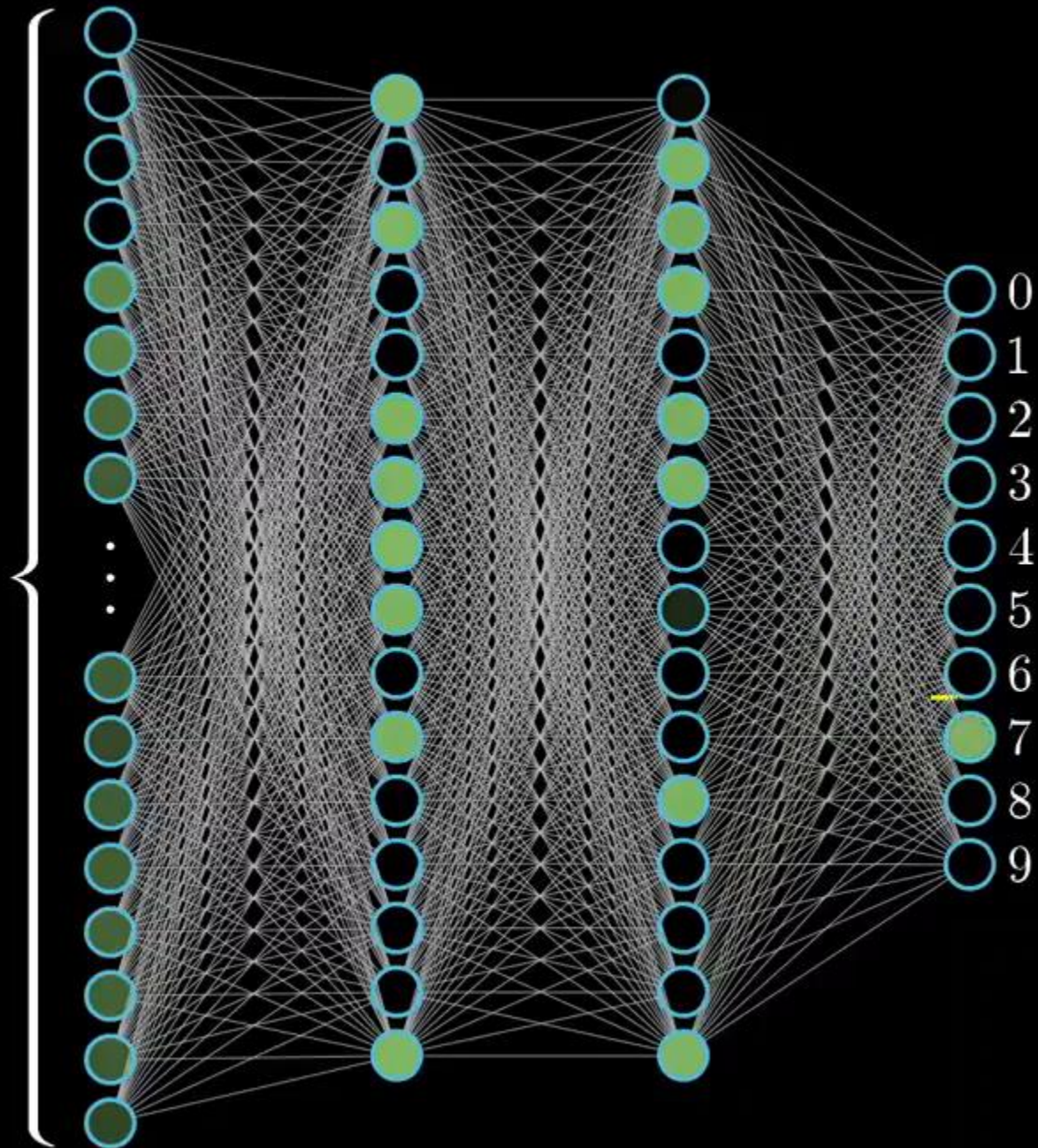


3Blue1Brown

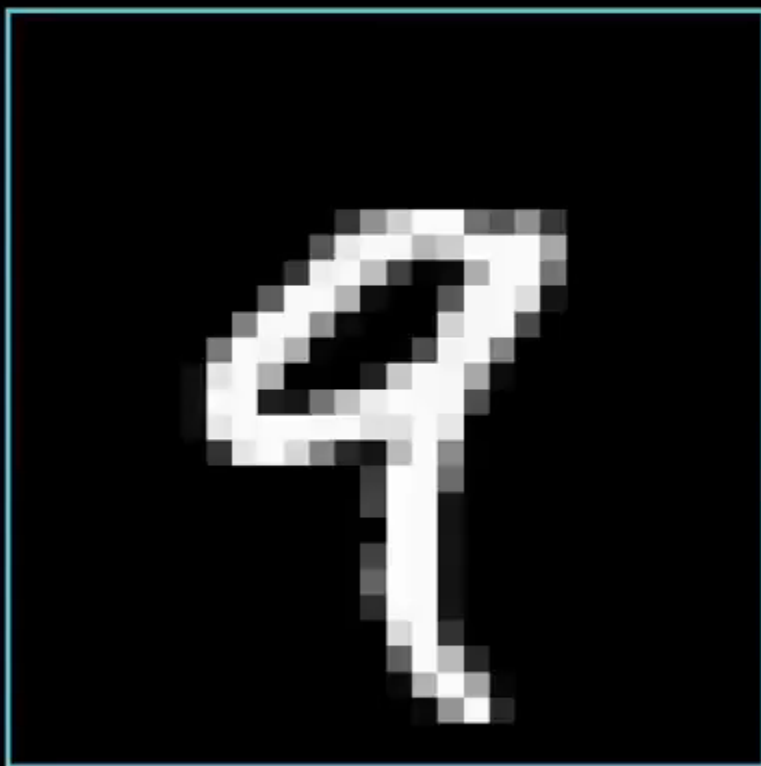




784

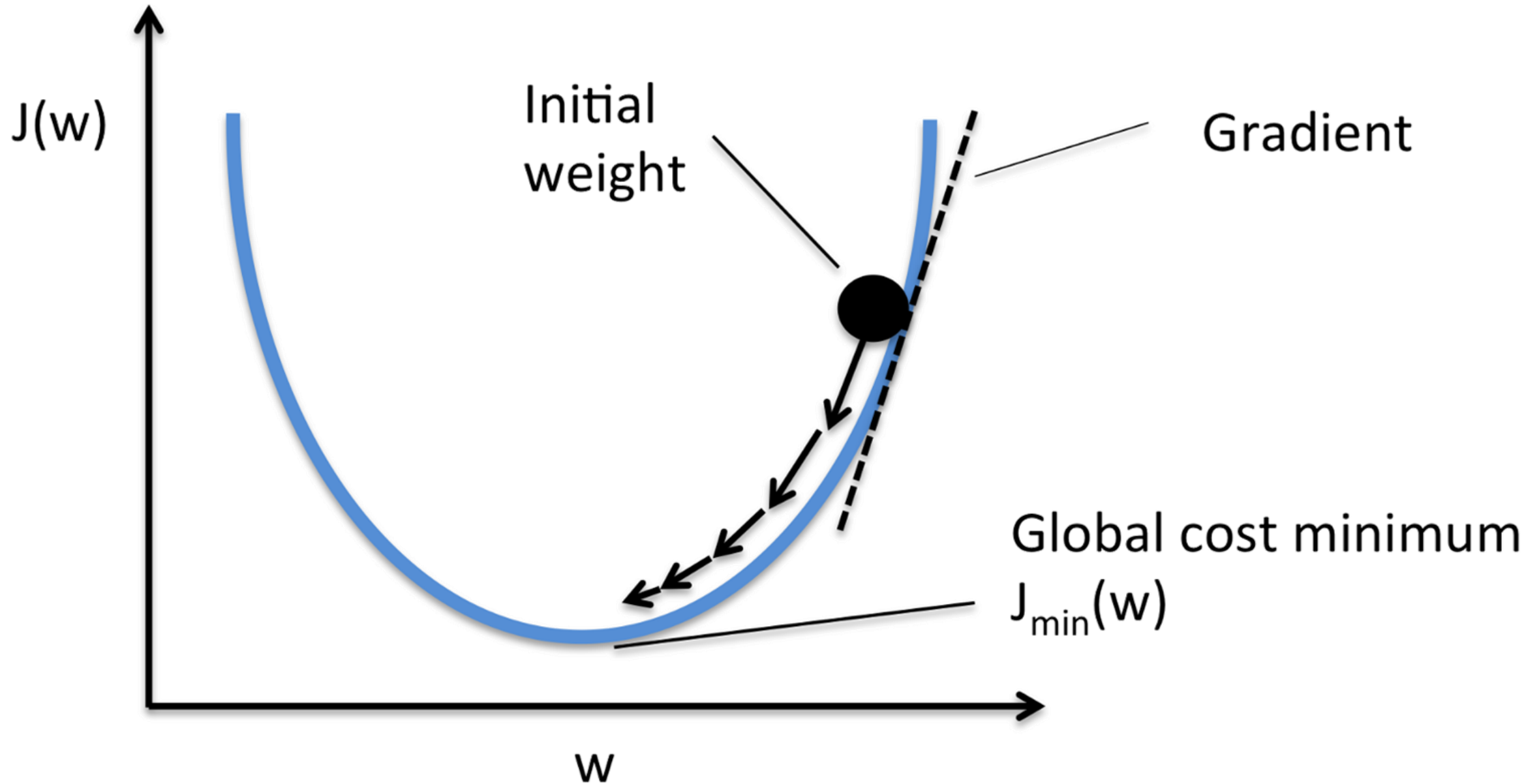


3Blue1Brown

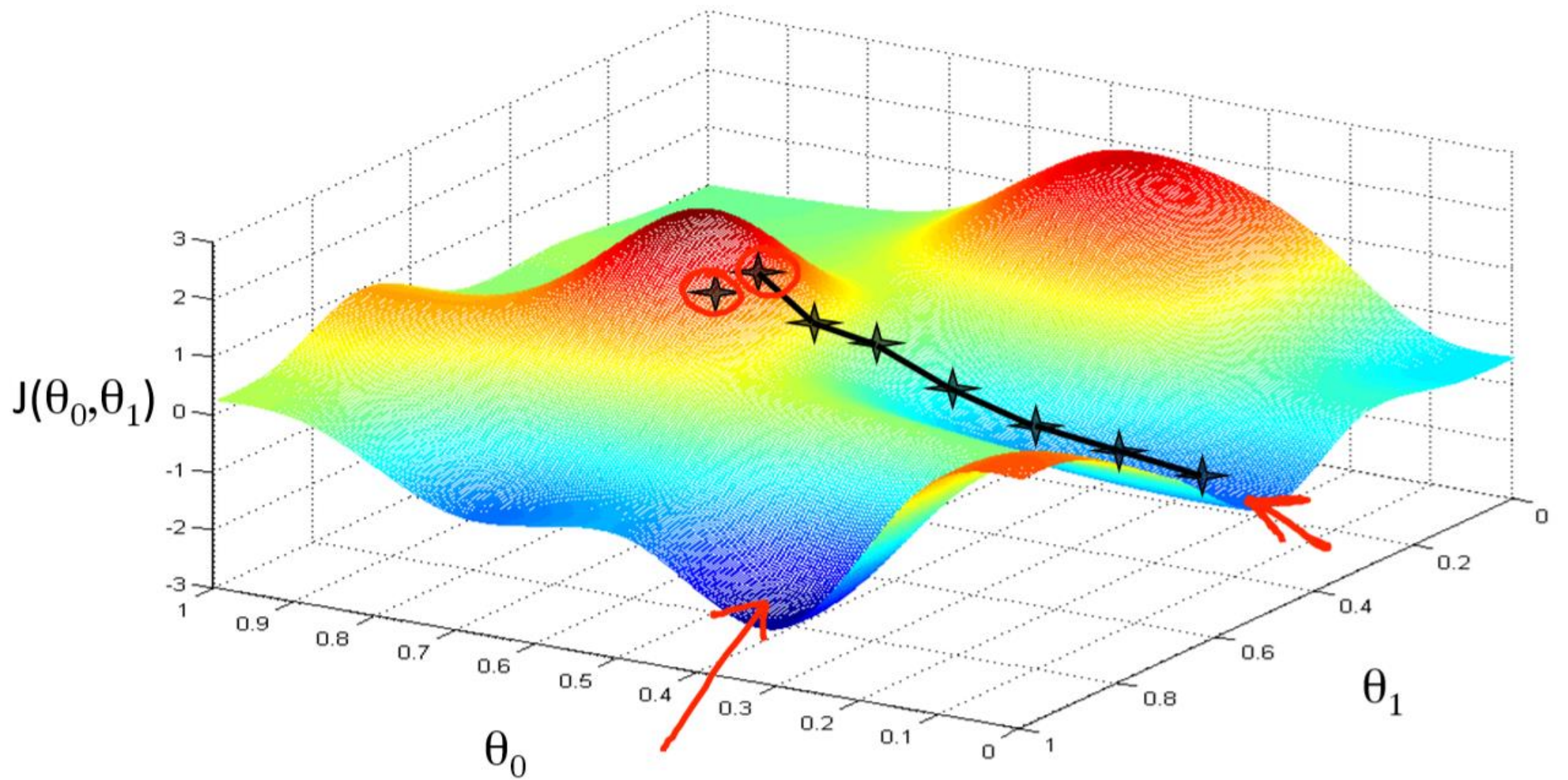




# Gradient Descent







<https://hackernoon.com/gradient-descent-aynk-7cbe95a778da>

# Cost Function

- Mean-Squared Error

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (f_{\theta}(x_i) - y_i)^2$$



# Gradient Descent of MSE

- Gradient of MSE

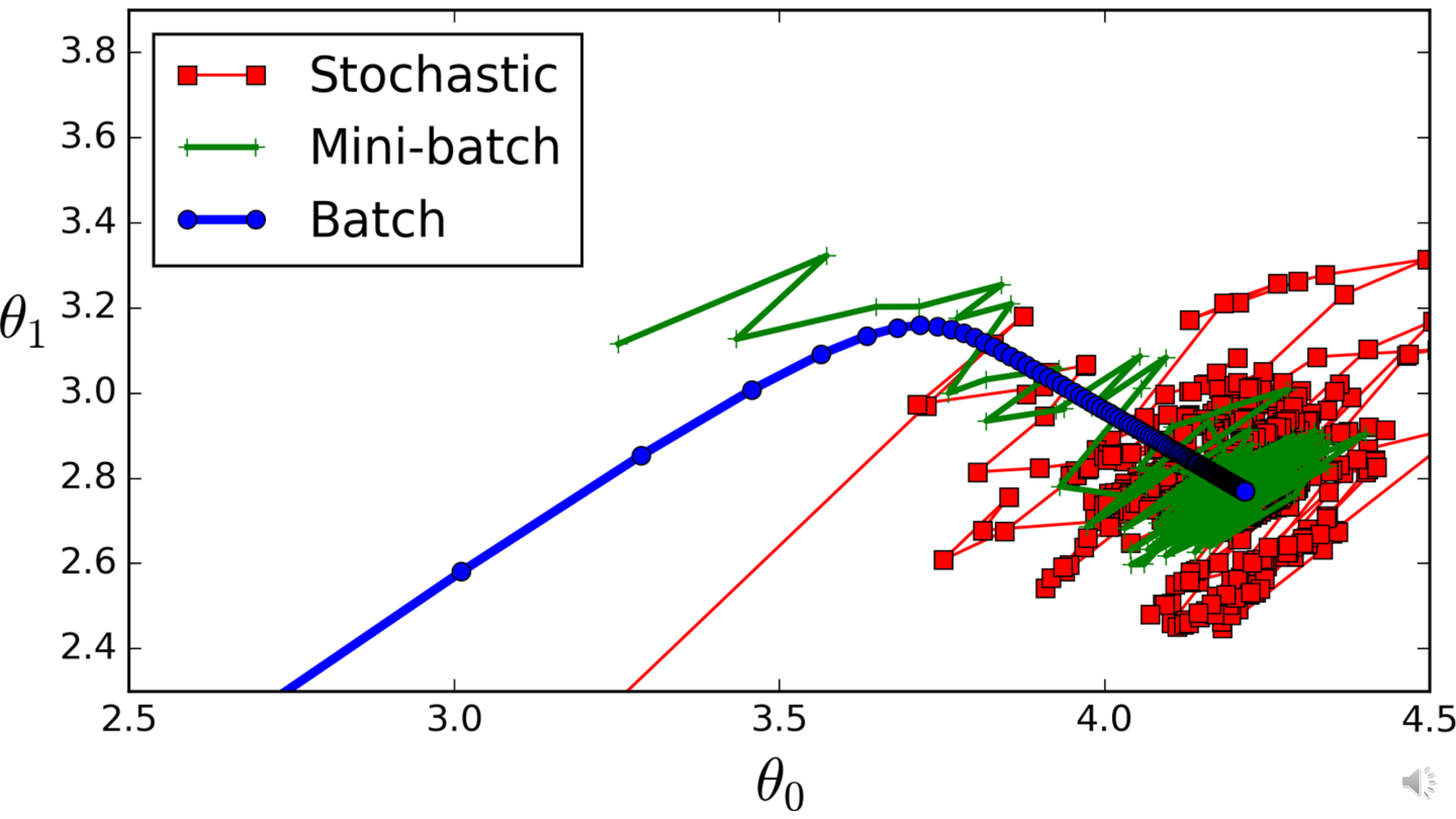
$$\frac{\partial J(\theta)}{\partial \theta} = \frac{2}{N} \sum_{i=1}^N (f_{\theta}(x_i) - y_i) f'_{\theta}(x_i)$$

- Update

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

- Repeat until Convergence







## Cost function

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

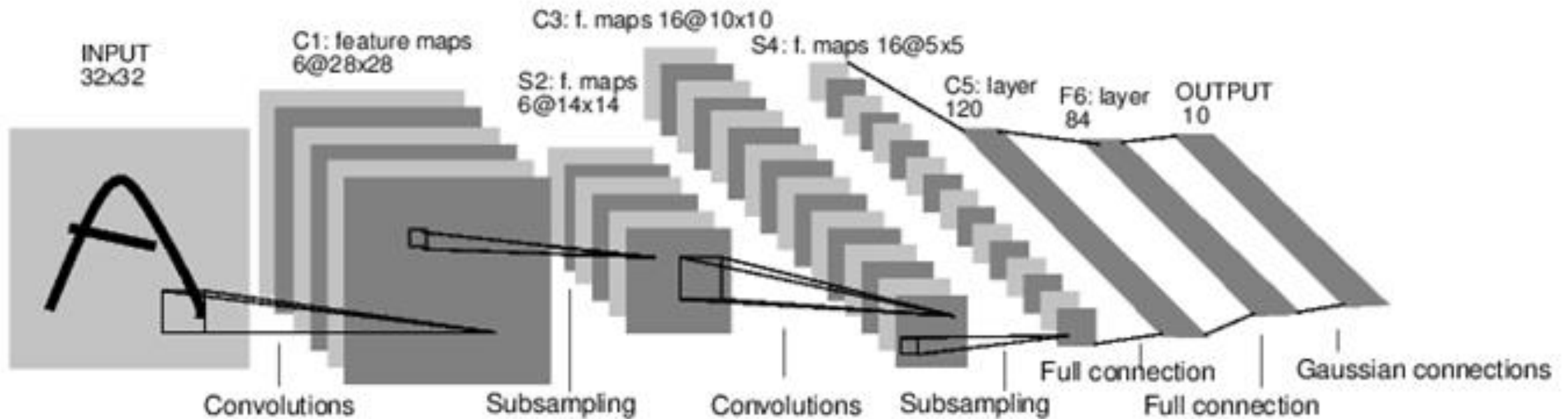
Weights and biases

# Convolutional Neural Networks (CNNs)



# Convolutional Neural Networks (CNNs)

- <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)





14,197,122 images, 21841 classes  
(2021/9/21)



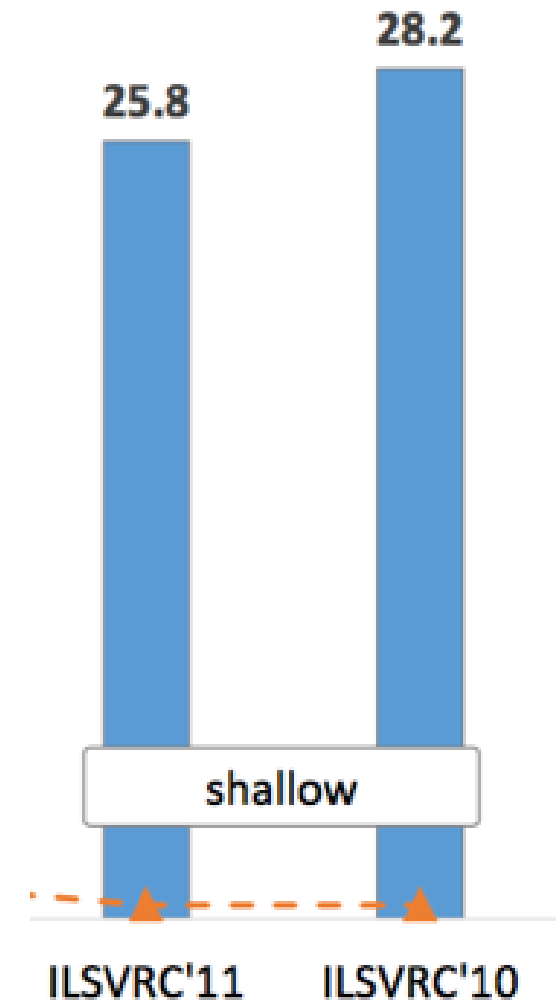


# ImageNet Large Scale Visual Object Recognition Challenge (ILSVRC)

- 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

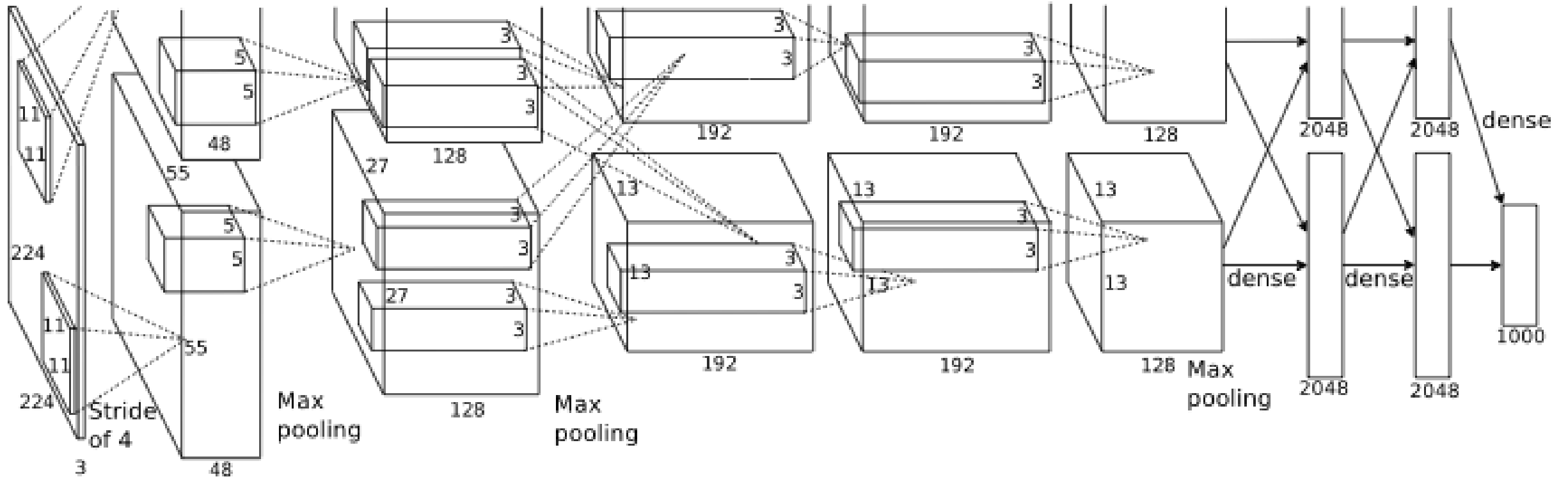


# Winners' Error Rates on ImageNet Challenge

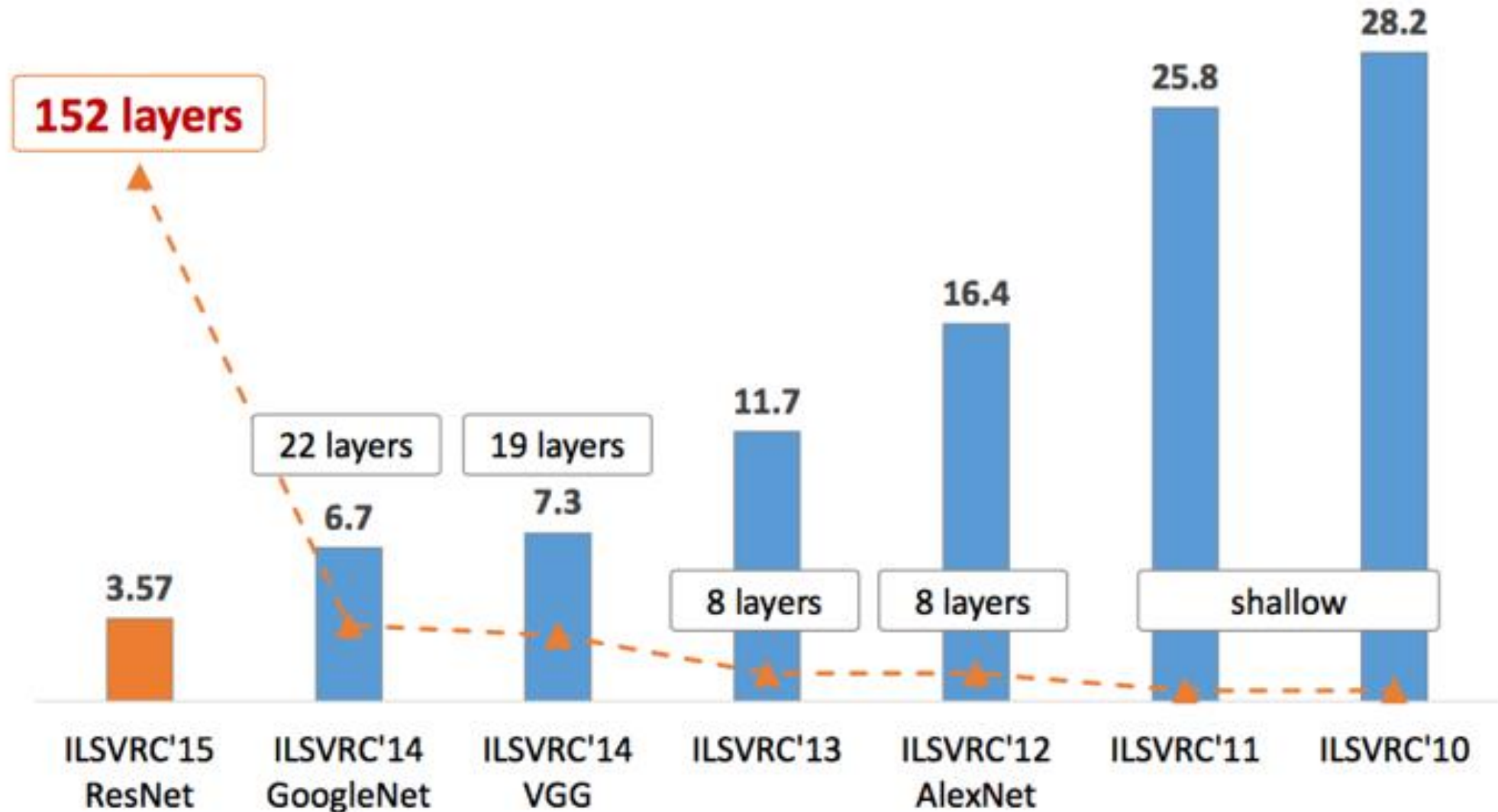


# Convolutional Neural Network (AlexNet)

- Alex Krizhevsky, Geoffery Hinton et al., 2012



# Winners' Error Rates on ImageNet Challenge





```

graph BT
    A[Convolution 5x5, 30] -- "30x140x140" --> B[Activation tanh]
    B -- "30x140x140" --> C[Pooling max, 2x2]
    C -- "30x140x140" --> D[Convolution 5x5, 30]
    D -- "30x140x140" --> E[Activation tanh]
    E -- "30x140x140" --> 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]
  
```

```

graph BT
    FC2[Fully Connected 20736] --> A2[Activation relu]
    A2 --> D2[Dropout]
    D2 --> FC3[Fully Connected 4096]
    FC3 --> A3[Activation relu]
    A3 --> D3[Dropout]
    D3 --> FC4[Fully Connected 4096]
    FC4 --> A4[Activation relu]
    A4 --> D4[Dropout]
    D4 --> S[Softmax Output]
    F[Flatten] --> FC1[Fully Connected 20736]
    FC1 --> A1[Activation relu]
    A1 --> D1[Dropout]
    D1 --> P1[Pooling max, 3x3/2]
    P1 --> A0[Activation relu]
    A0 --> C1[Convolution 3x3, 256]
    C1 --> A0_1[Activation relu]
    A0_1 --> C2[Convolution 3x3, 384]
    C2 --> A0_2[Activation relu]
    A0_2 --> C3[Convolution 3x3, 384]
    C3 --> A0_3[Activation relu]
    A0_3 --> C4[Convolution 3x3, 256]
    C4 --> A0_4[Activation relu]
    A0_4 --> P2[Pooling max, 3x3/2]
    P2 --> LRN[LRN]
    LRN --> A0_5[Activation relu]
    A0_5 --> C5[Convolution 3x3, 256]
    C5 --> A0_6[Activation relu]
    A0_6 --> P3[Pooling max, 3x3/2]
    P3 --> A0_7[Activation relu]
    A0_7 --> C6[Convolution 3x3, 256]
    C6 --> A0_8[Activation relu]
    A0_8 --> P4[Pooling max, 3x3/2]
    P4 --> A0_9[Activation relu]
    A0_9 --> C7[Convolution 3x3, 256]
    C7 --> A0_10[Activation relu]
    A0_10 --> P5[Pooling max, 3x3/2]
    P5 --> A0_11[Activation relu]
    A0_11 --> C8[Convolution 3x3, 256]
    C8 --> A0_12[Activation relu]
    A0_12 --> P6[Pooling max, 3x3/2]
    P6 --> A0_13[Activation relu]
    A0_13 --> C9[Convolution 3x3, 256]
    C9 --> A0_14[Activation relu]
    A0_14 --> P7[Pooling max, 3x3/2]
    P7 --> A0_15[Activation relu]
    A0_15 --> C10[Convolution 3x3, 256]
    C10 --> A0_16[Activation relu]
    A0_16 --> P8[Pooling max, 3x3/2]
    P8 --> A0_17[Activation relu]
    A0_17 --> C11[Convolution 3x3, 256]
    C11 --> A0_18[Activation relu]
    A0_18 --> P9[Pooling max, 3x3/2]
    P9 --> A0_19[Activation relu]
    A0_19 --> C12[Convolution 3x3, 256]
    C12 --> A0_20[Activation relu]
    A0_20 --> P10[Pooling max, 3x3/2]
    P10 --> A0_21[Activation relu]
    A0_21 --> C13[Convolution 3x3, 256]
    C13 --> A0_22[Activation relu]
    A0_22 --> P11[Pooling max, 3x3/2]
    P11 --> A0_23[Activation relu]
    A0_23 --> C14[Convolution 3x3, 256]
    C14 --> A0_24[Activation relu]
    A0_24 --> P12[Pooling max, 3x3/2]
    P12 --> A0_25[Activation relu]
    A0_25 --> C15[Convolution 3x3, 256]
    C15 --> A0_26[Activation relu]
    A0_26 --> P13[Pooling max, 3x3/2]
    P13 --> A0_27[Activation relu]
    A0_27 --> C16[Convolution 3x3, 256]
    C16 --> A0_28[Activation relu]
    A0_28 --> P14[Pooling max, 3x3/2]
    P14 --> A0_29[Activation relu]
    A0_29 --> C17[Convolution 3x3, 256]
    C17 --> A0_30[Activation relu]
    A0_30 --> P15[Pooling max, 3x3/2]
    P15 --> A0_31[Activation relu]
    A0_31 --> C18[Convolution 3x3, 256]
    C18 --> A0_32[Activation relu]
    A0_32 --> P16[Pooling max, 3x3/2]
    P16 --> A0_33[Activation relu]
    A0_33 --> C19[Convolution 3x3, 256]
    C19 --> A0_34[Activation relu]
    A0_34 --> P17[Pooling max, 3x3/2]
    P17 --> A0_35[Activation relu]
    A0_35 --> C20[Convolution 3x3, 256]
    C20 --> A0_36[Activation relu]
    A0_36 --> P18[Pooling max, 3x3/2]
    P18 --> A0_37[Activation relu]
    A0_37 --> C21[Convolution 3x3, 256]
    C21 --> A0_38[Activation relu]
    A0_38 --> P19[Pooling max, 3x3/2]
    P19 --> A0_39[Activation relu]
    A0_39 --> C22[Convolution 3x3, 256]
    C22 --> A0_40[Activation relu]
    A0_40 --> P20[Pooling max, 3x3/2]
    P20 --> A0_41[Activation relu]
    A0_41 --> C23[Convolution 3x3, 256]
    C23 --> A0_42[Activation relu]
    A0_42 --> P21[Pooling max, 3x3/2]
    P21 --> A0_43[Activation relu]
    A0_43 --> C24[Convolution 3x3, 256]
    C24 --> A0_44[Activation relu]
    A0_44 --> P22[Pooling max, 3x3/2]
    P22 --> A0_45[Activation relu]
    A0_45 --> C25[Convolution 3x3, 256]
    C25 --> A0_46[Activation relu]
    A0_46 --> P23[Pooling max, 3x3/2]
    P23 --> A0_47[Activation relu]
    A0_47 --> C26[Convolution 3x3, 256]
    C26 --> A0_48[Activation relu]
    A0_48 --> P24[Pooling max, 3x3/2]
    P24 --> A0_49[Activation relu]
    A0_49 --> C27[Convolution 3x3, 256]
    C27 --> A0_50[Activation relu]
    A0_50 --> P25[Pooling max, 3x3/2]
    P25 --> A0_51[Activation relu]
    A0_51 --> C28[Convolution 3x3, 256]
    C28 --> A0_52[Activation relu]
    A0_52 --> P26[Pooling max, 3x3/2]
    P26 --> A0_53[Activation relu]
    A0_53 --> C29[Convolution 3x3, 256]
    C29 --> A0_54[Activation relu]
    A0_54 --> P27[Pooling max, 3x3/2]
    P27 --> A0_55[Activation relu]
    A0_55 --> C30[Convolution 3x3, 256]
    C30 --> A0_56[Activation relu]
    A0_56 --> P28[Pooling max, 3x3/2]
    P28 --> A0_57[Activation relu]
    A0_57 --> C31[Convolution 3x3, 256]
    C31 --> A0_58[Activation relu]
    A0_58 --> P29[Pooling max, 3x3/2]
    P29 --> A0_59[Activation relu]
    A0_59 --> C32[Convolution 3x3, 256]
    C32 --> A0_60[Activation relu]
    A0_60 --> P30[Pooling max, 3x3/2]
    P30 --> A0_61[Activation relu]
    A0_61 --> C33[Convolution 3x3, 256]
    C33 --> A0_62[Activation relu]
    A0_62 --> P31[Pooling max, 3x3/2]
    P31 --> A0_63[Activation relu]
    A0_63 --> C34[Convolution 3x3, 256]
    C34 --> A0_64[Activation relu]
    A0_64 --> P32[Pooling max, 3x3/2]
    P32 --> A0_65[Activation relu]
    A0_65 --> C35[Convolution 3x3, 256]
    C35 --> A0_66[Activation relu]
    A0_66 --> P33[Pooling max, 3x3/2]
    P33 --> A0_67[Activation relu]
    A0_67 --> C36[Convolution 3x3, 256]
    C36 --> A0_68[Activation relu]
    A0_68 --> P34[Pooling max, 3x3/2]
    P34 --> A0_69[Activation relu]
    A0_69 --> C37[Convolution 3x3, 256]
    C37 --> A0_70[Activation relu]
    A0_70 --> P35[Pooling max, 3x3/2]
    P35 --> A0_71[Activation relu]
    A0_71 --> C38[Convolution 3x3, 256]
    C38 --> A0_72[Activation relu]
    A0_72 --> P36[Pooling max, 3x3/2]
    P36 --> A0_73[Activation relu]
    A0_73 --> C39[Convolution 3x3, 256]
    C39 --> A0_74[Activation relu]
    A0_74 --> P37[Pooling max, 3x3/2]
    P37 --> A0_75[Activation relu]
    A0_75 --> C40[Convolution 3x3, 256]
    C40 --> A0_76[Activation relu]
    A0_76 --> P38[Pooling max, 3x3/2]
    P38 --> A0_77[Activation relu]
    A0_77 --> C41[Convolution 3x3, 256]
    C41 --> A0_78[Activation relu]
    A0_78 --> P39[Pooling max, 3x3/2]
    P39 --> A0_79[Activation relu]
    A0_79 --> C42[Convolution 3x3, 256]
    C42 --> A0_80[Activation relu]
    A0_80 --> P40[Pooling max, 3x3/2]
    P40 --> A0_81[Activation relu]
    A0_81 --> C43[Convolution 3x
```

[illegible]

The diagram illustrates the VGG-16 architecture, a deep convolutional neural network. It starts with an input image of size 224x224x3. The architecture is divided into two main stages of convolutional layers, each followed by a max pooling layer. The first stage consists of two convolutional layers (3x3 and 3x3) and a max pooling layer. The second stage also consists of two convolutional layers (3x3 and 3x3) and a max pooling layer. The output of the second stage is flattened and passed through two fully connected layers (1000 units each) to produce the final softmax output. The diagram uses color-coding: red for convolutional layers, yellow for activation functions, blue for pooling, and orange for concatenation and flattening.

[illegible]

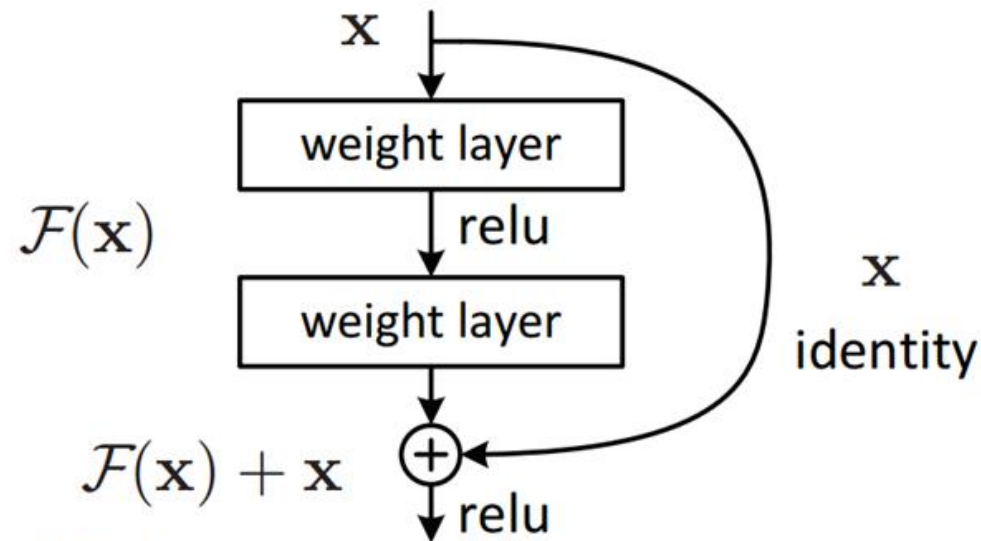
A close-up shot of Leonardo DiCaprio in a dark suit and tie, looking slightly to his right with a serious expression. Another man's head and shoulder are visible in the foreground on the right, partially obscuring the view. The background is blurred, suggesting an indoor setting with windows.

**WE NEED TO GO**

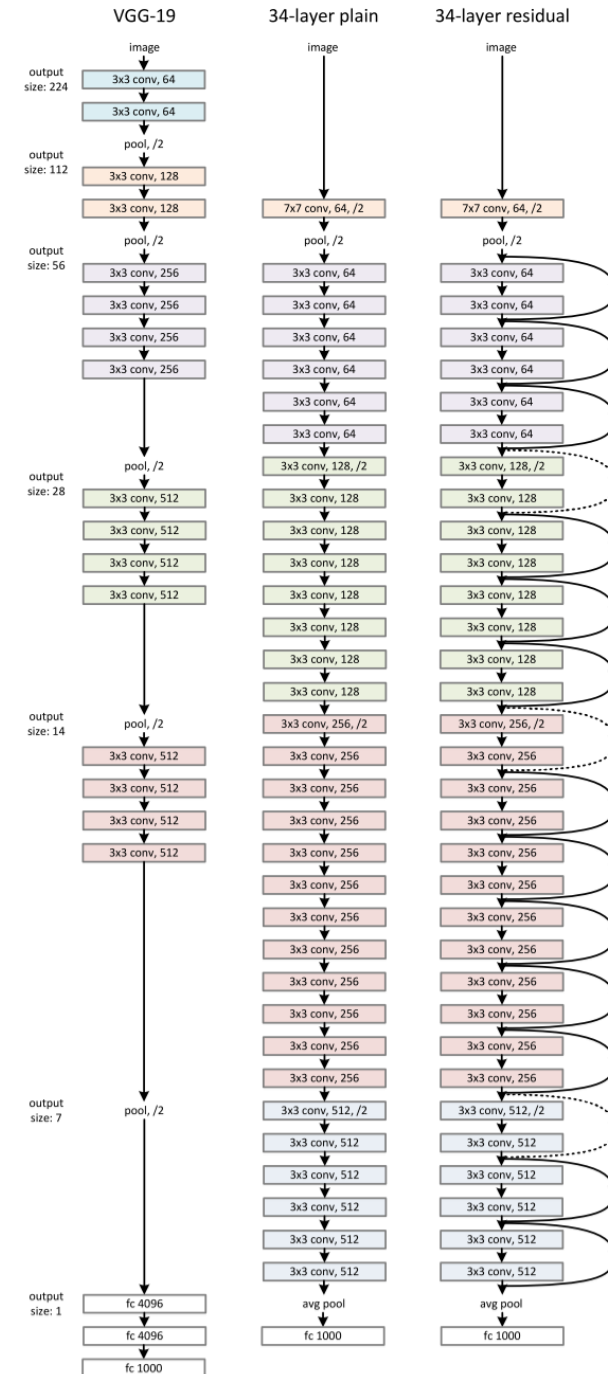
**DEEPER**

# ResNet (2015)

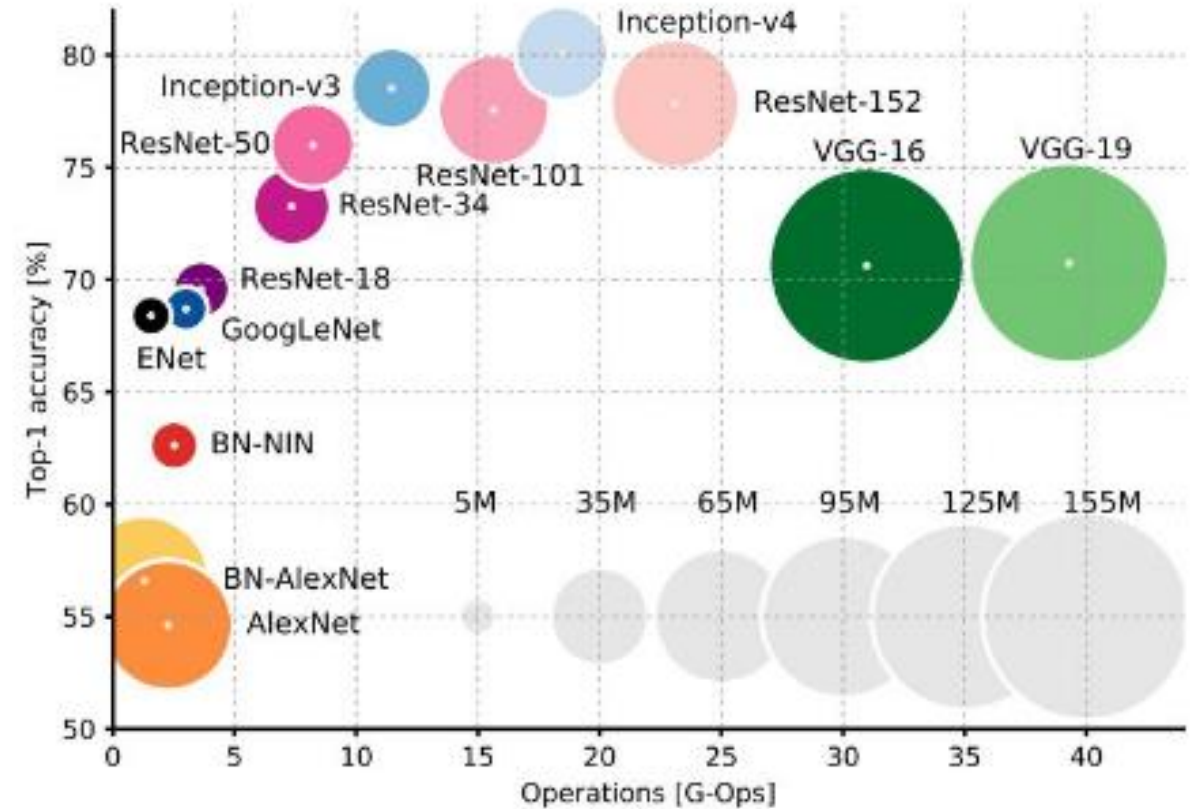
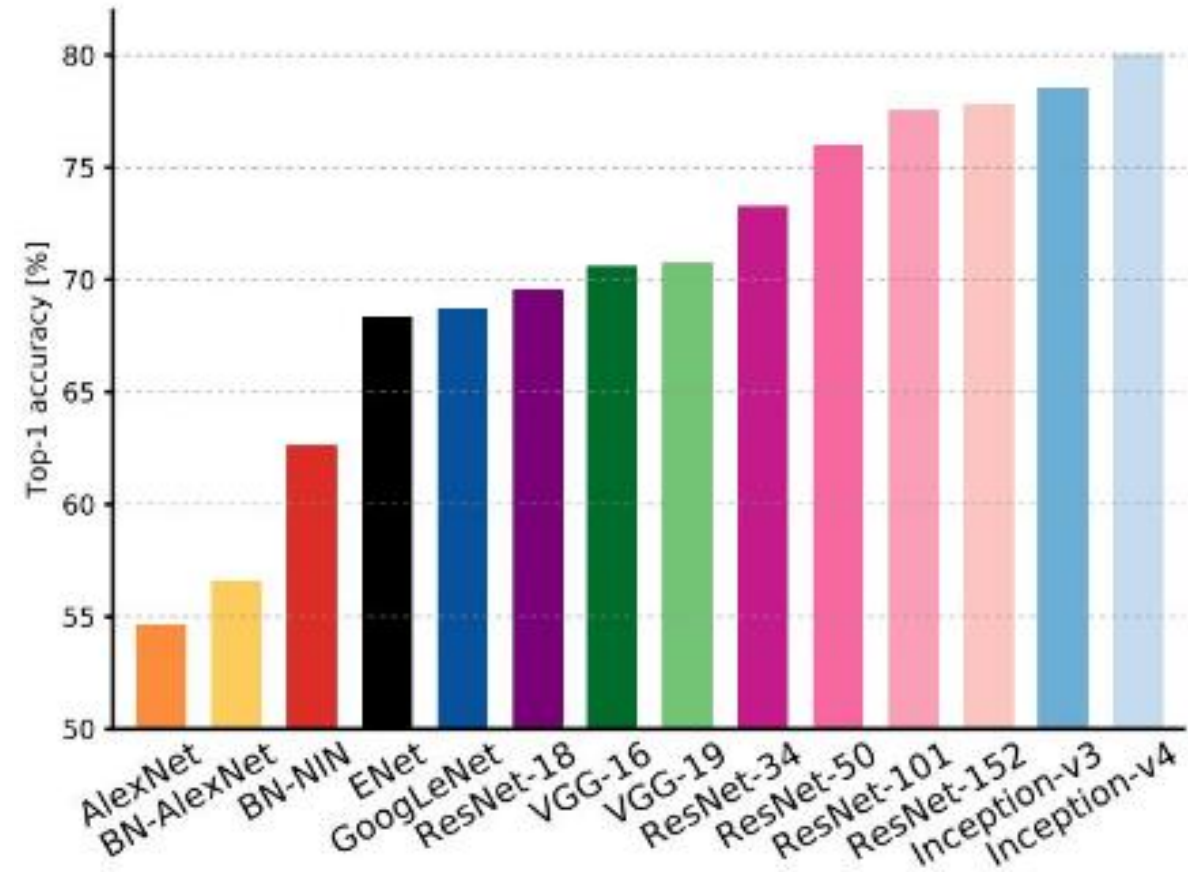
- Residual Neural Network
- Proposed “skip connection”
- 152-layer with 3.57% error rate



A residual block



# CNN Comparison

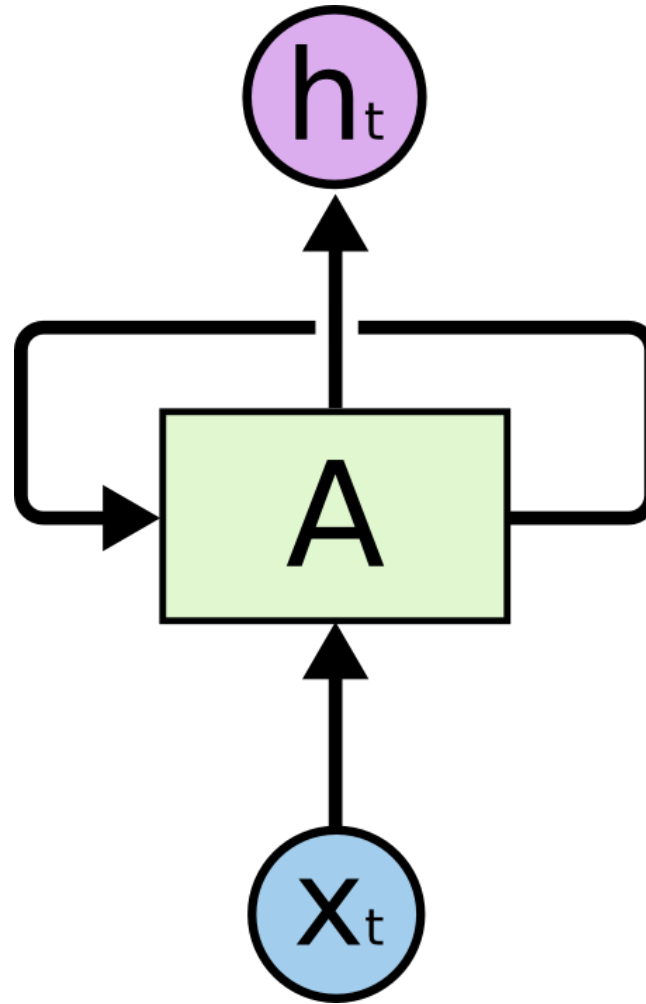




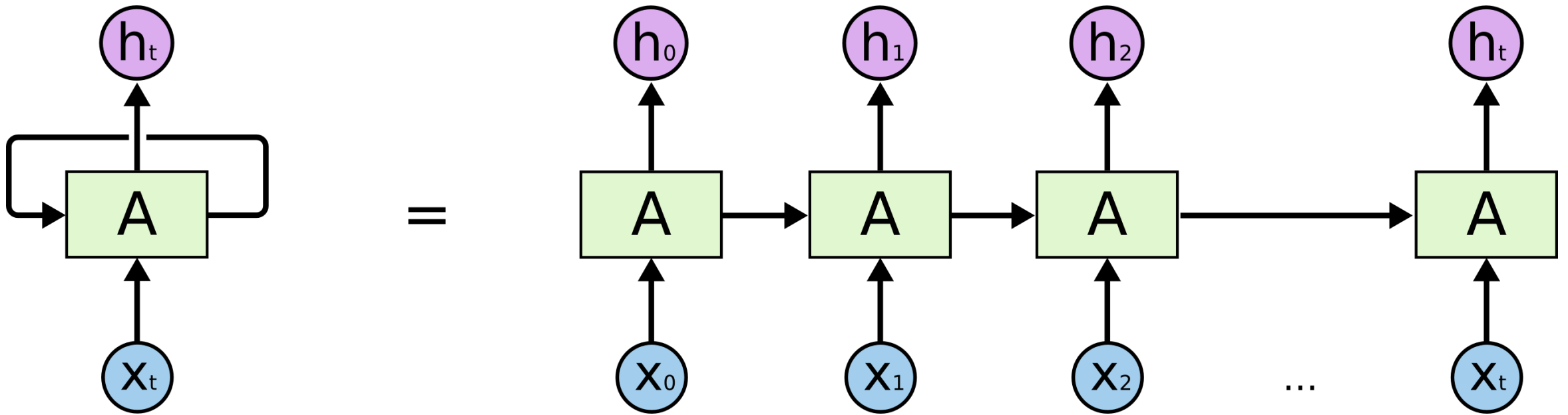
# Recurrent Neural Networks (RNNs) and LSTM



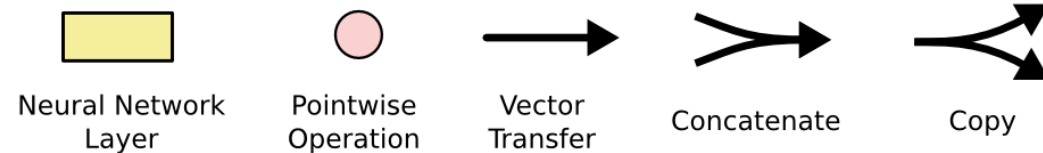
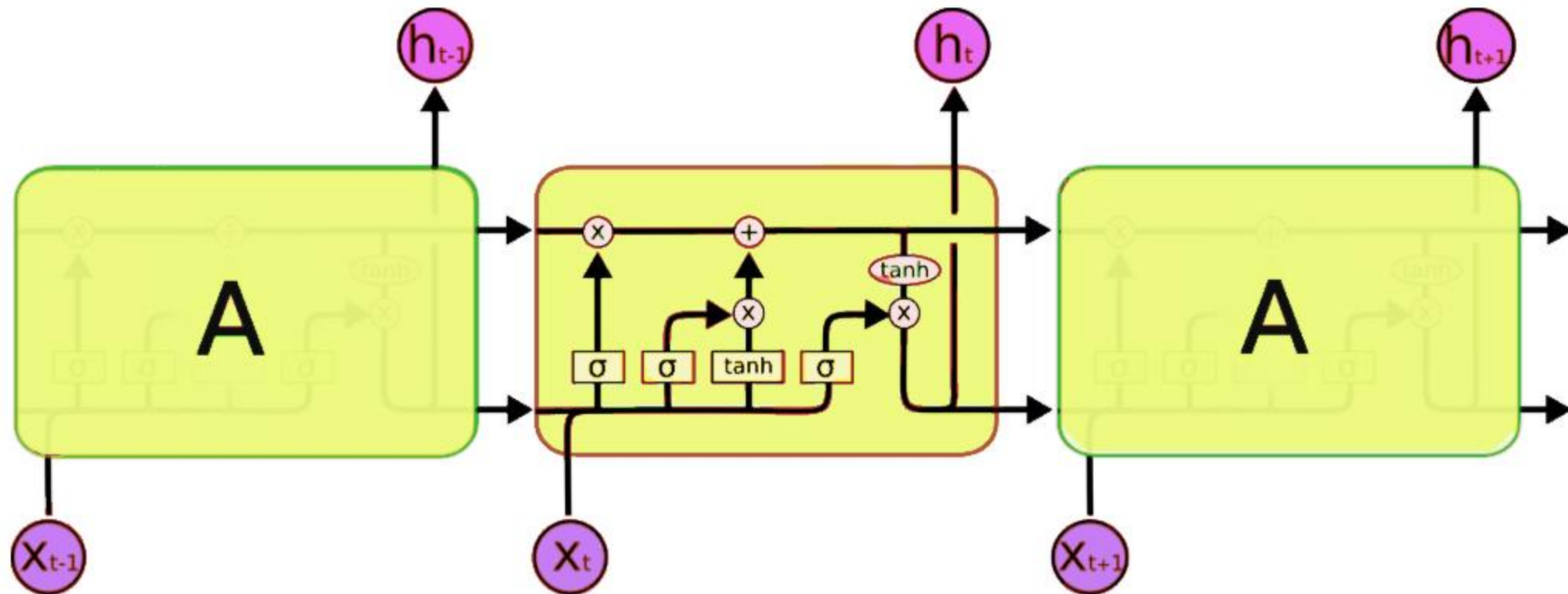
# Recurrent Neural Networks (RNNs)



# Unroll the RNN



# Long Short-term Memory (LSTM)





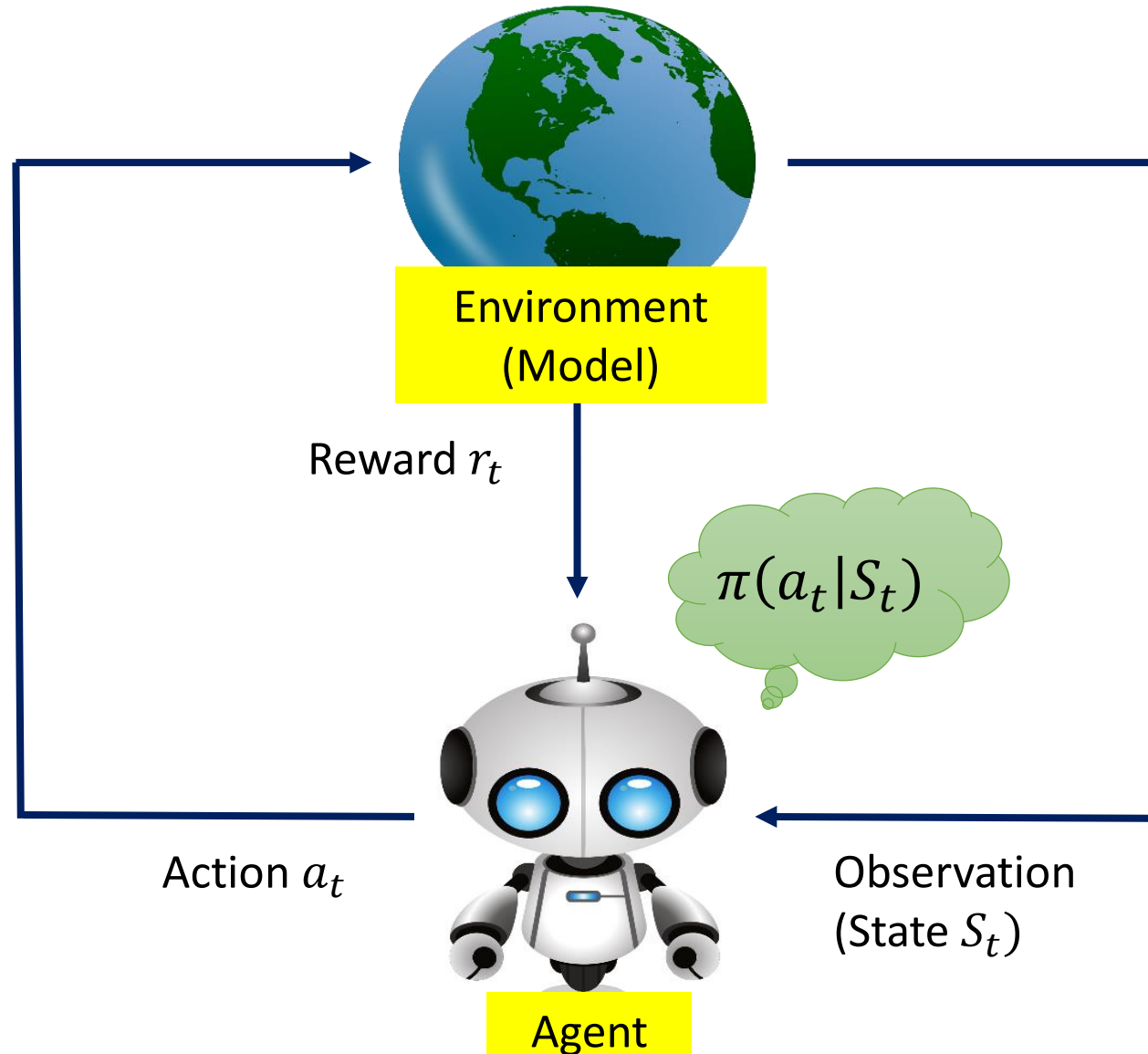
# Deep Reinforcement Learning



# What is Reinforcement Learning?



# Fundamentals of Reinforcement Learning





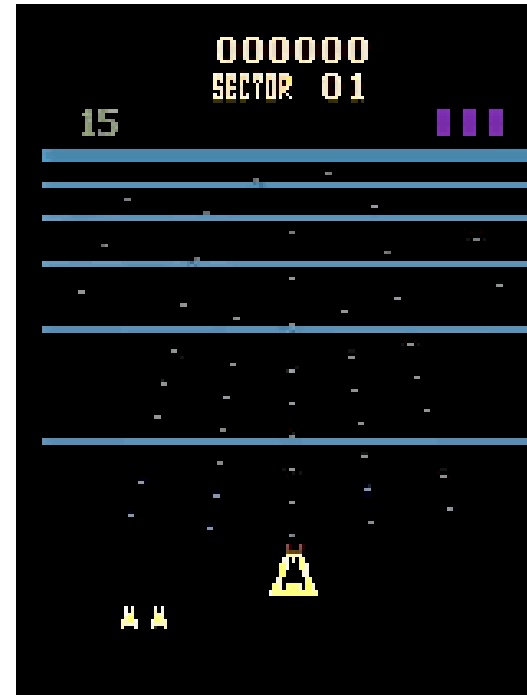
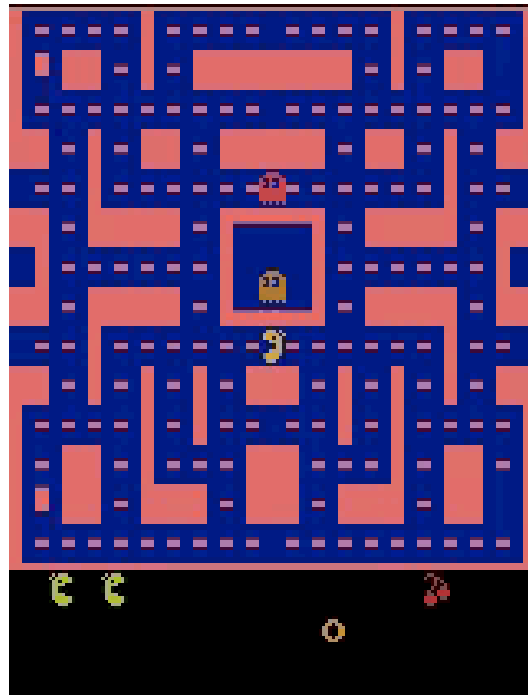
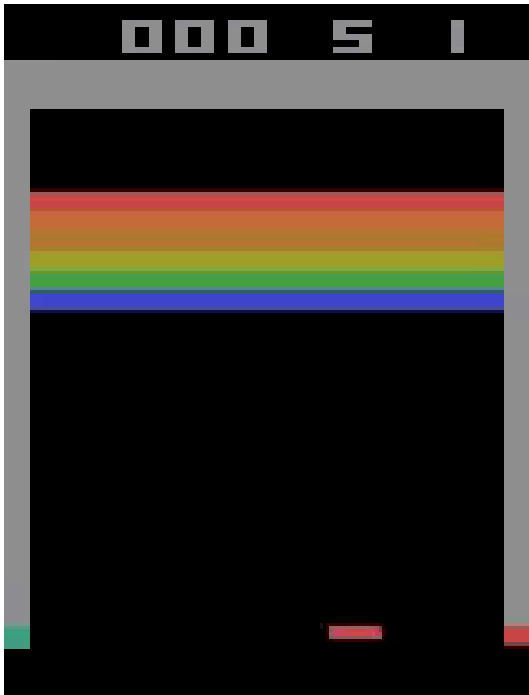
Google DeepMind



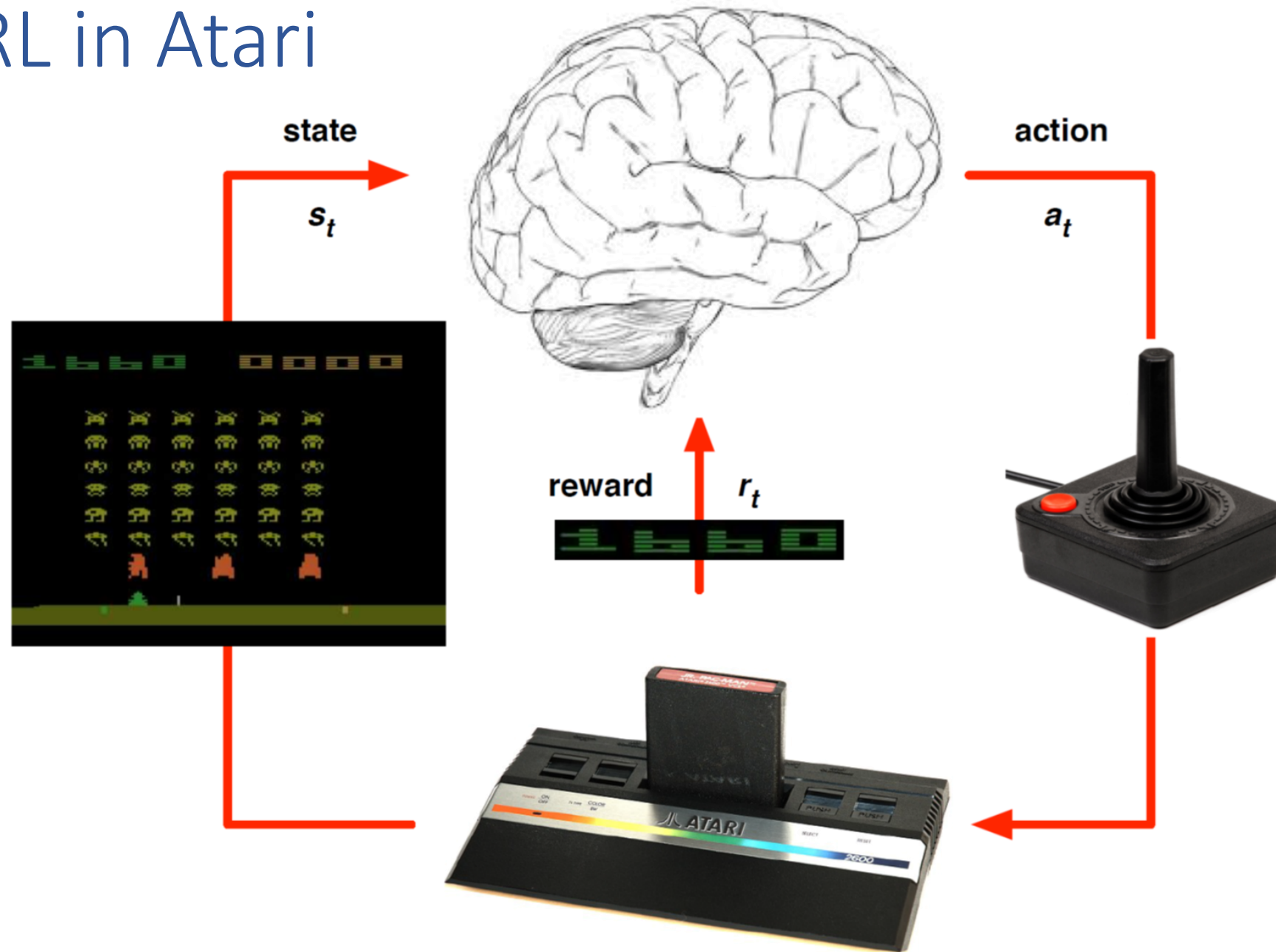


# Learn to Play Atari Games

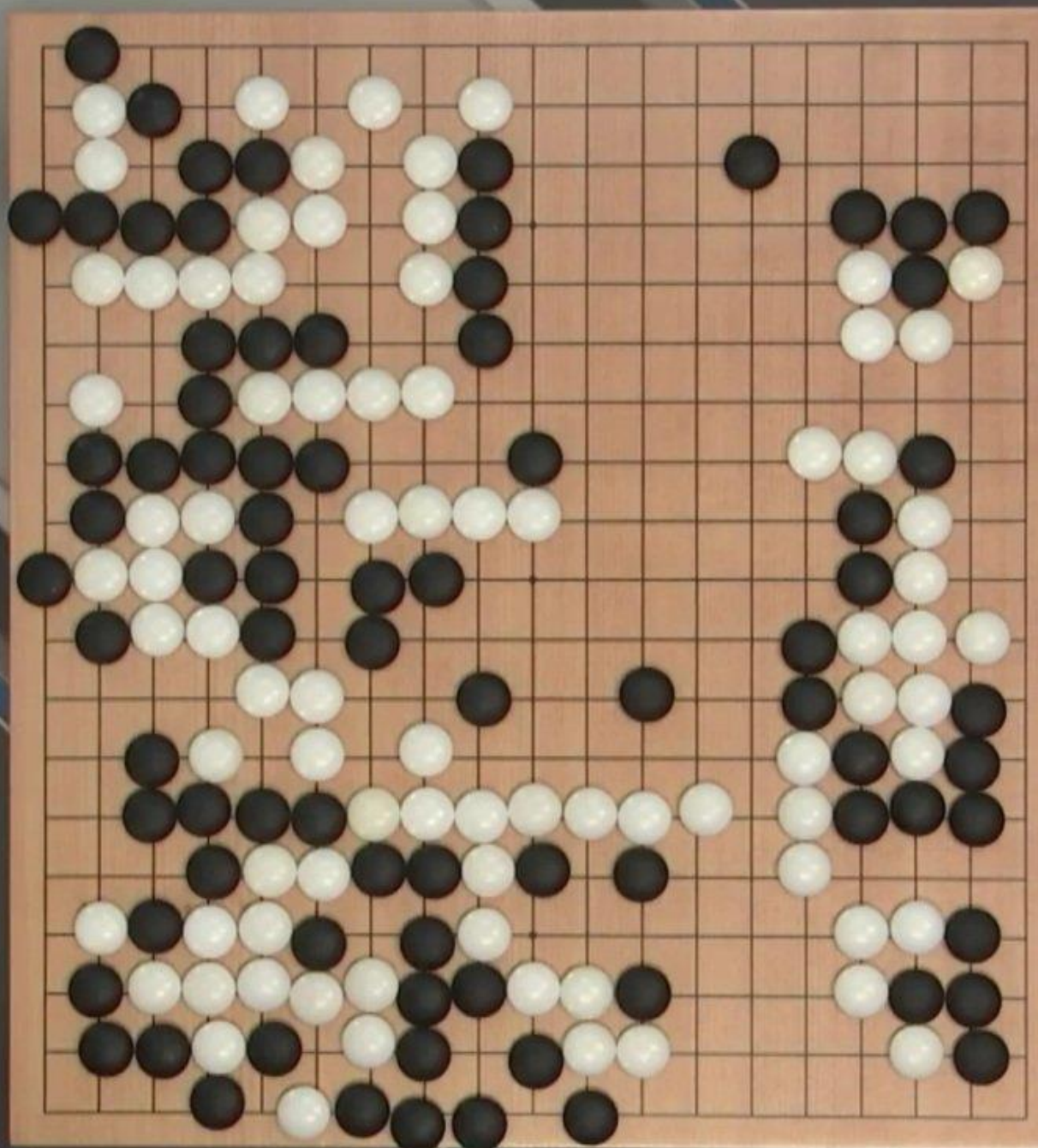
- Mnih et al., “Human Level Control through Deep Reinforcement Learning,” *Nature*, 2015



# DRL in Atari



● 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	$10^{18}$	$10^{54}$

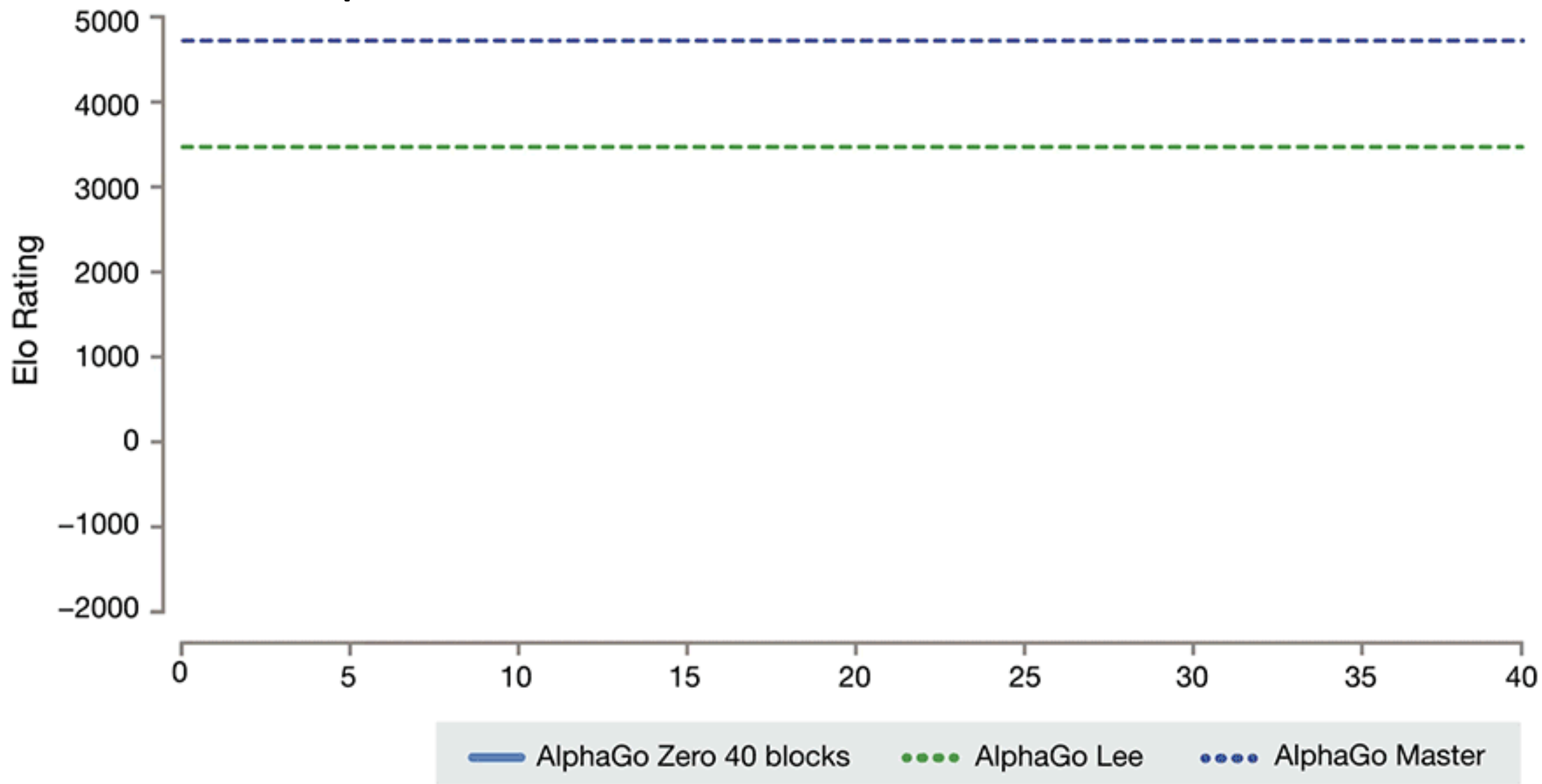


# AlphaGo Zero

Starting from scratch



# AlphaGo Zero



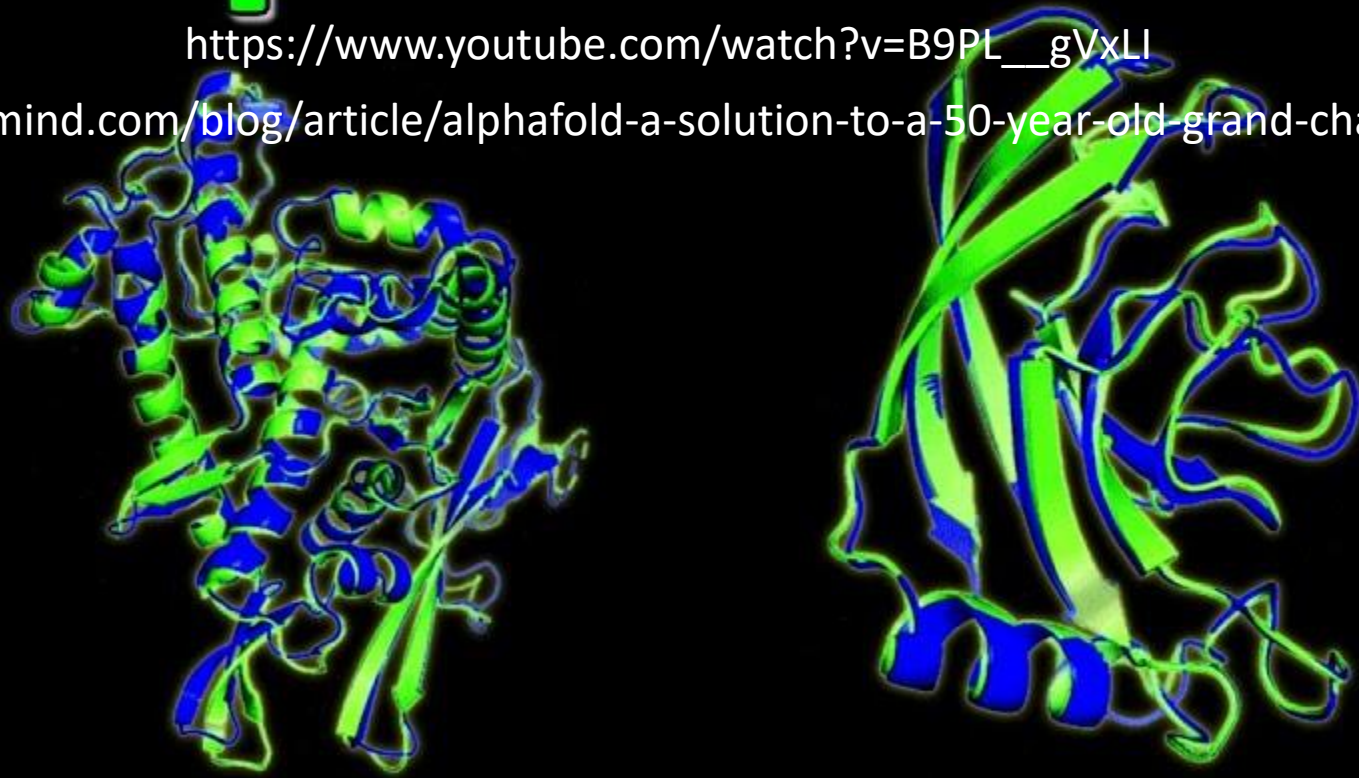




# Google DeepMind's **AlphaFold 2**

[https://www.youtube.com/watch?v=B9PL\\_gVxLI](https://www.youtube.com/watch?v=B9PL_gVxLI)

<https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>



# AI Breakthrough in Biology



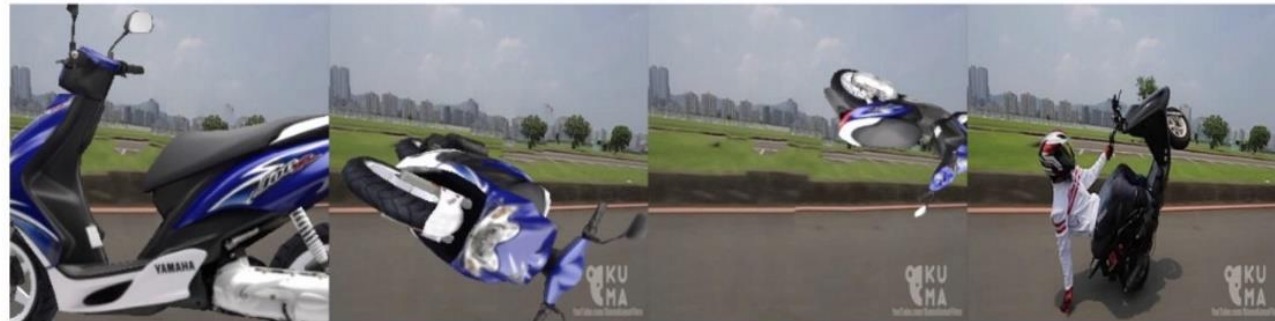
# Limits of Deep Learning



# 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





2020-06-01 06:44:03

# Tesla Autopilot Hit an Overturned Truck in Taiwan

民視新聞台 HD



嘉義



大貨車翻覆橫倒車道 特斯拉高速撞進車廂





2020-06-01 06:43:53

國1 北 267K+650 水上路段

民視新聞台 HD



大貨車翻覆橫倒車道 特斯拉高速撞進車廂

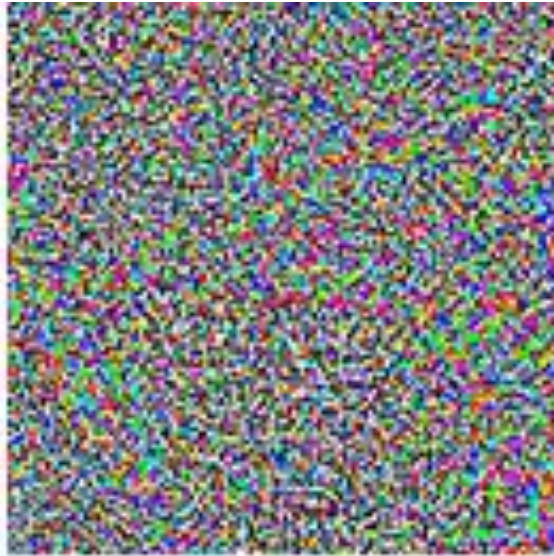
# Adversarial Attack



"panda"

57.7% confidence

+  $\epsilon$



=



"gibbon"

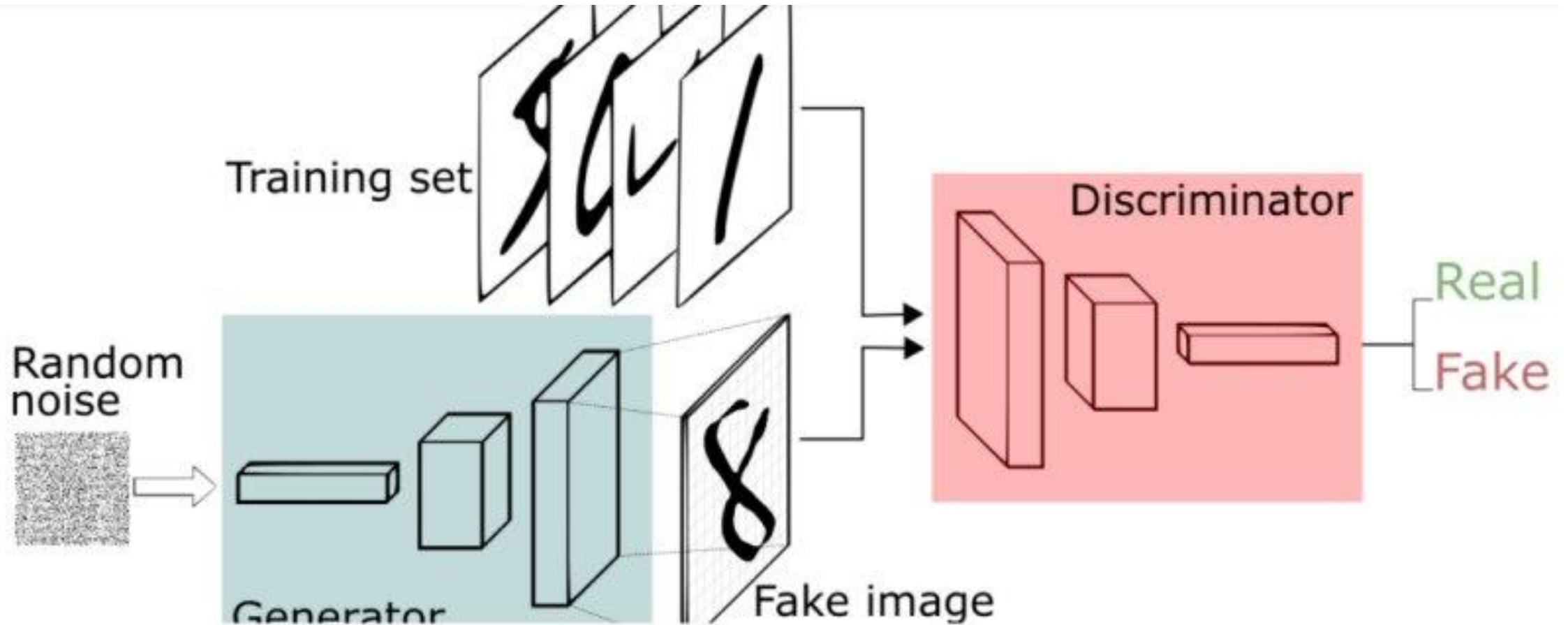
99.3% confidence





# Generative Adversarial Networks (GAN)

- Ian Goodfellow



# Super Resolution

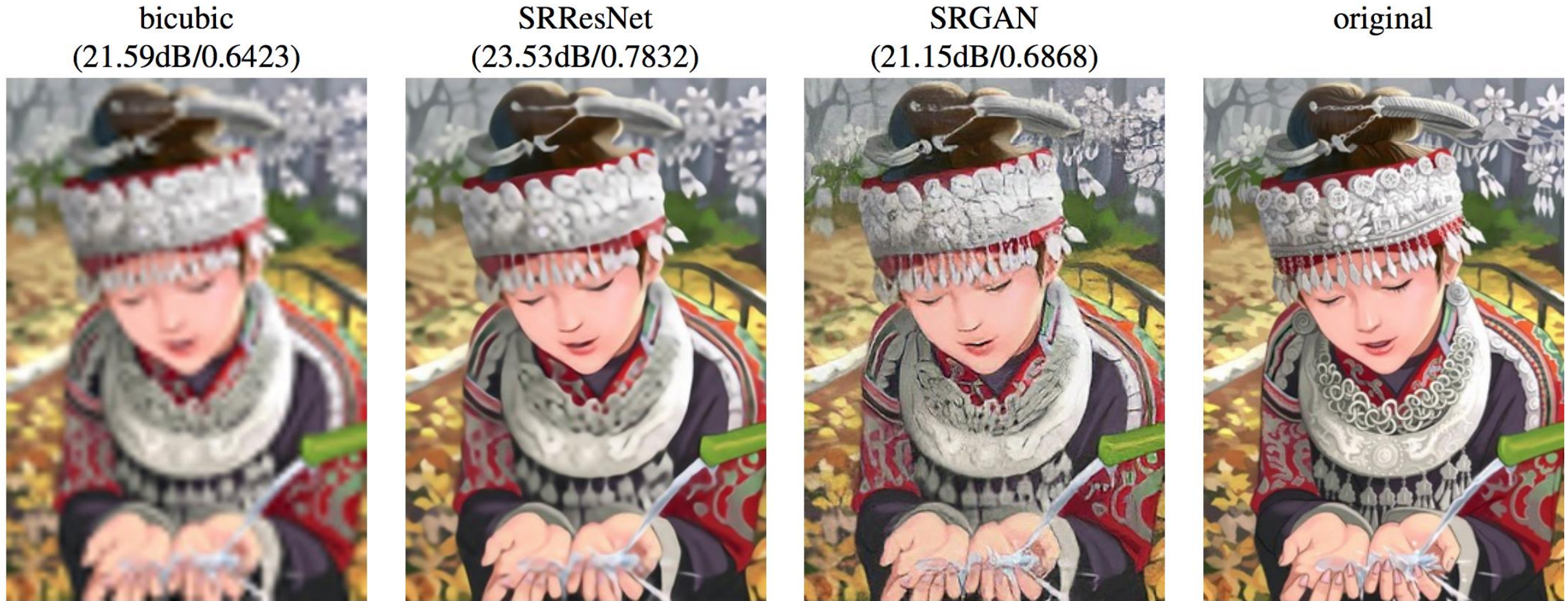
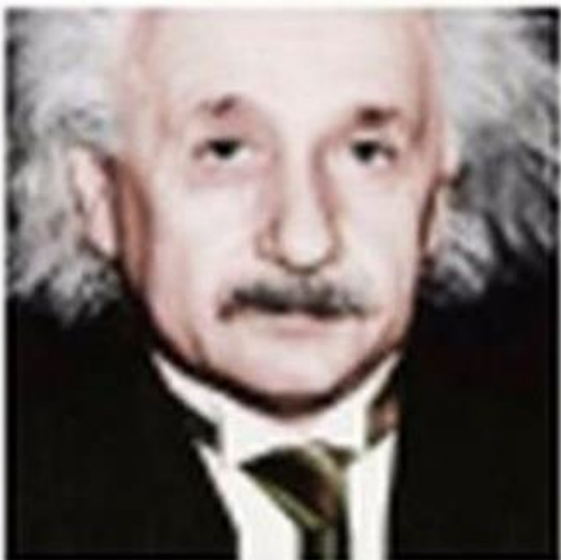


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]







DeepFake: Is this you?





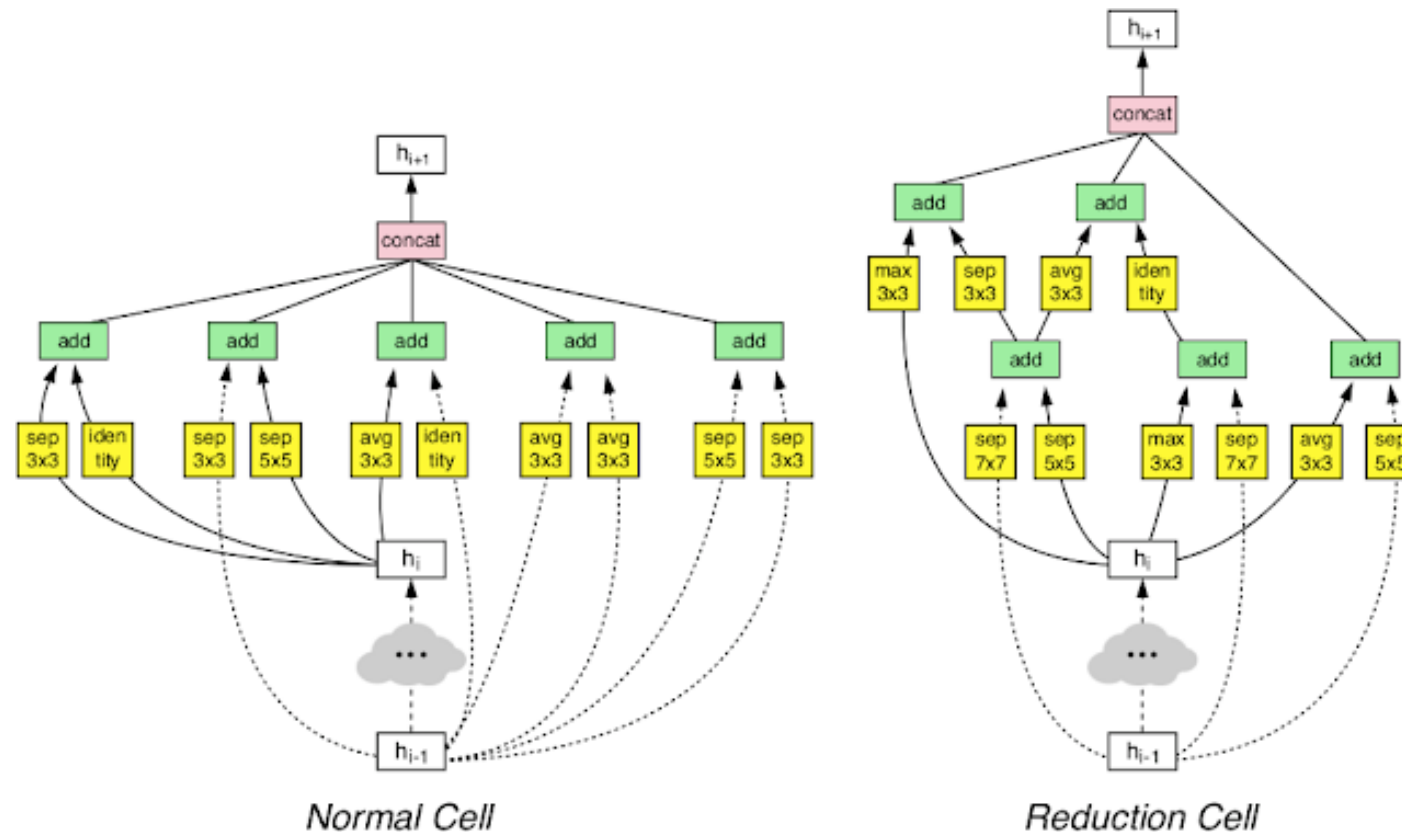
Buzzfeed



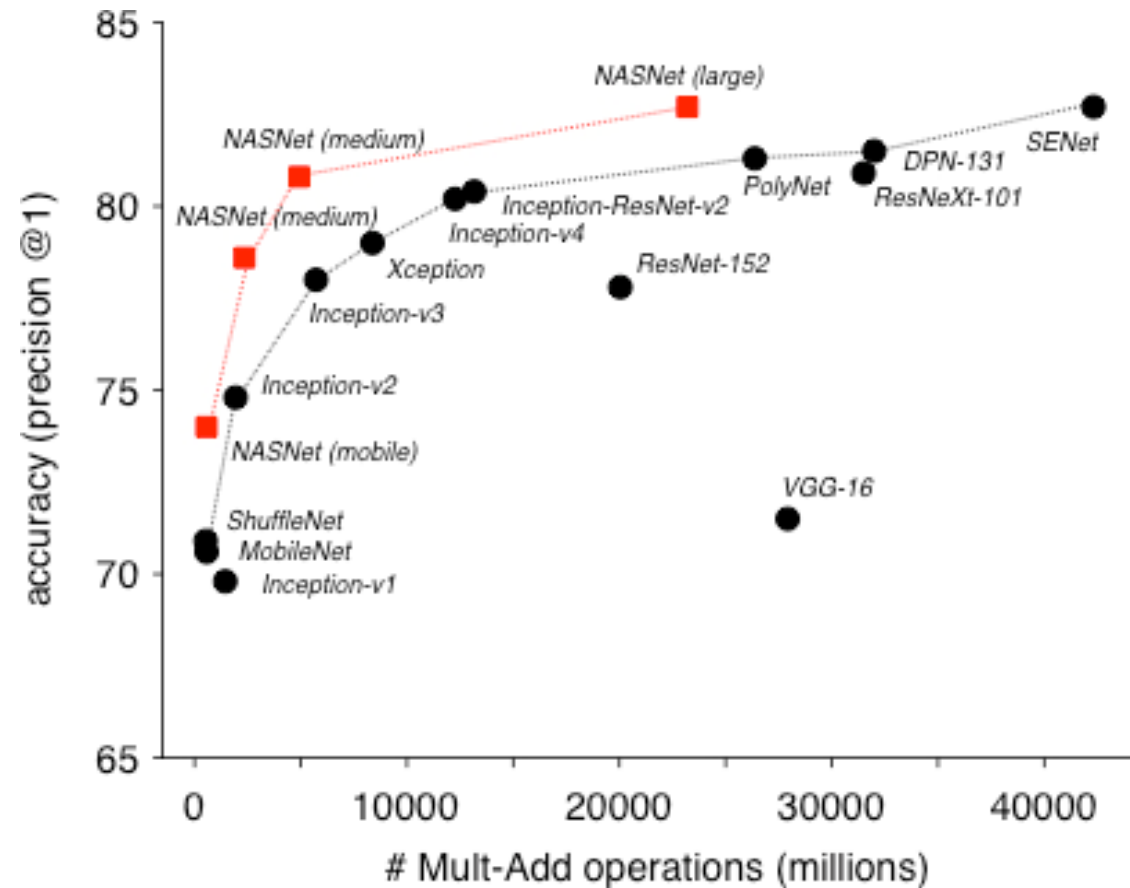
<https://www.youtube.com/watch?v=gLoI9hAX9dw>

# Google's AutoML

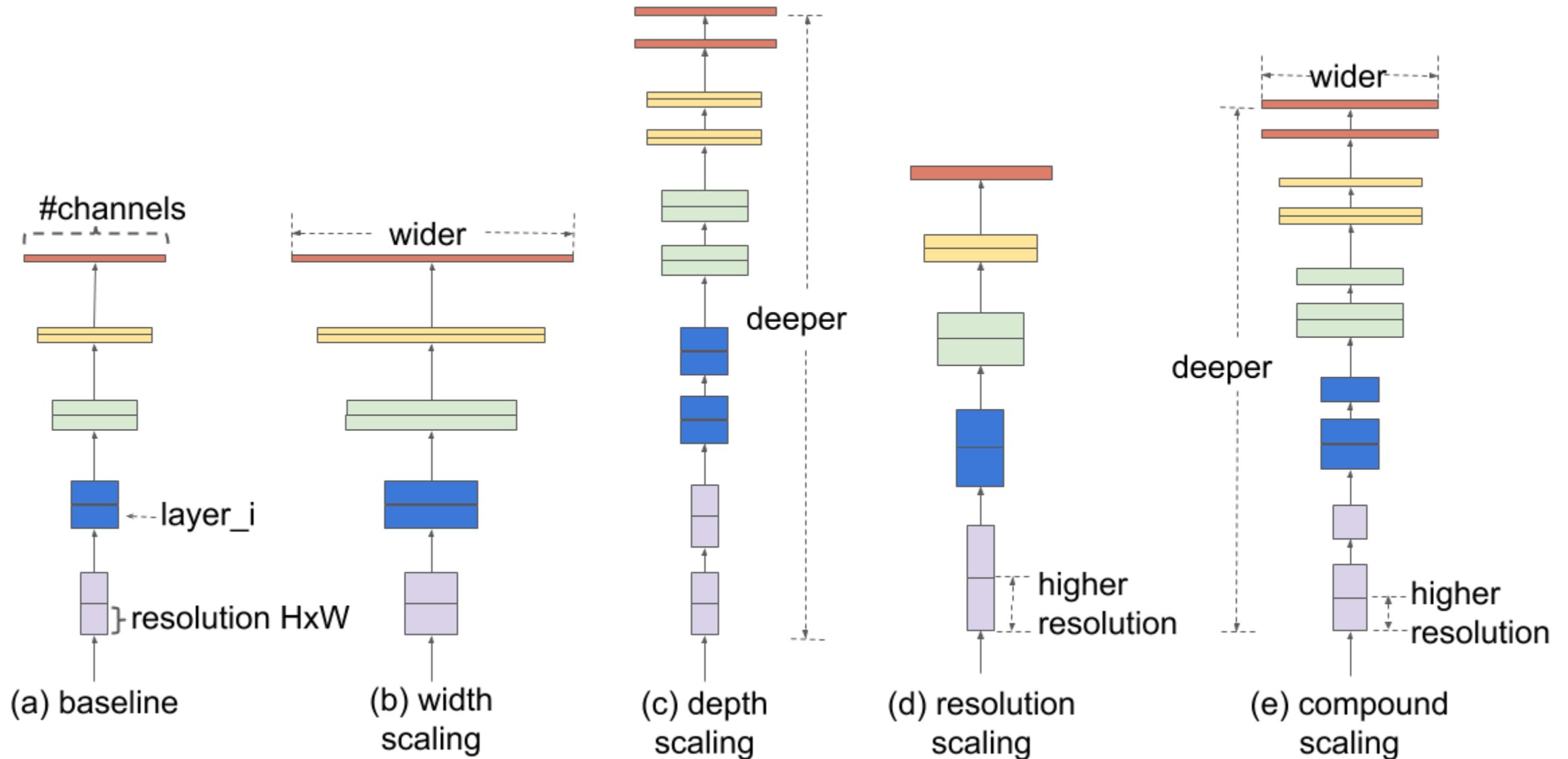
- Learning neural network cells automatically



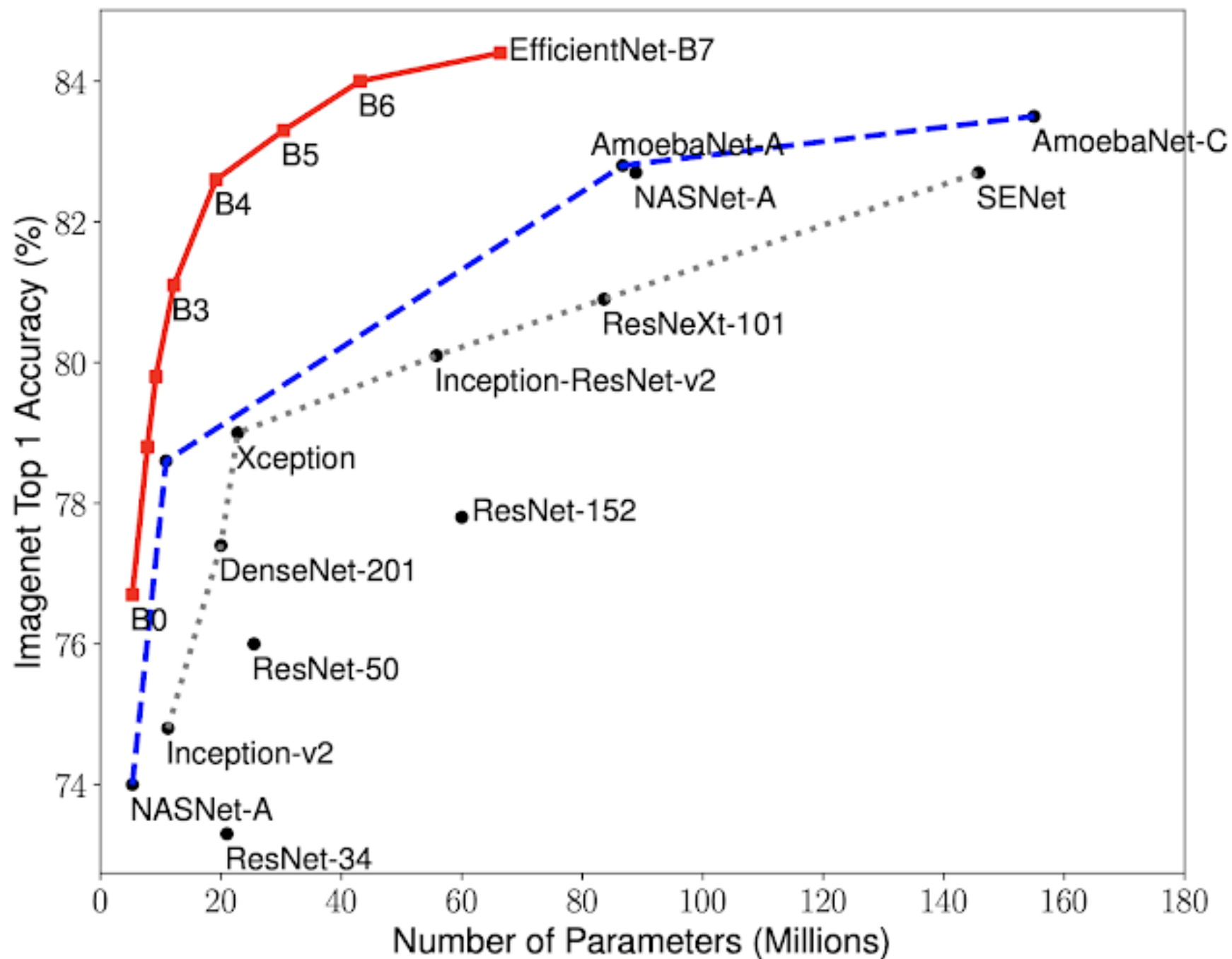
# AutoML on ImageNet



# EfficientNet (May, 2019)

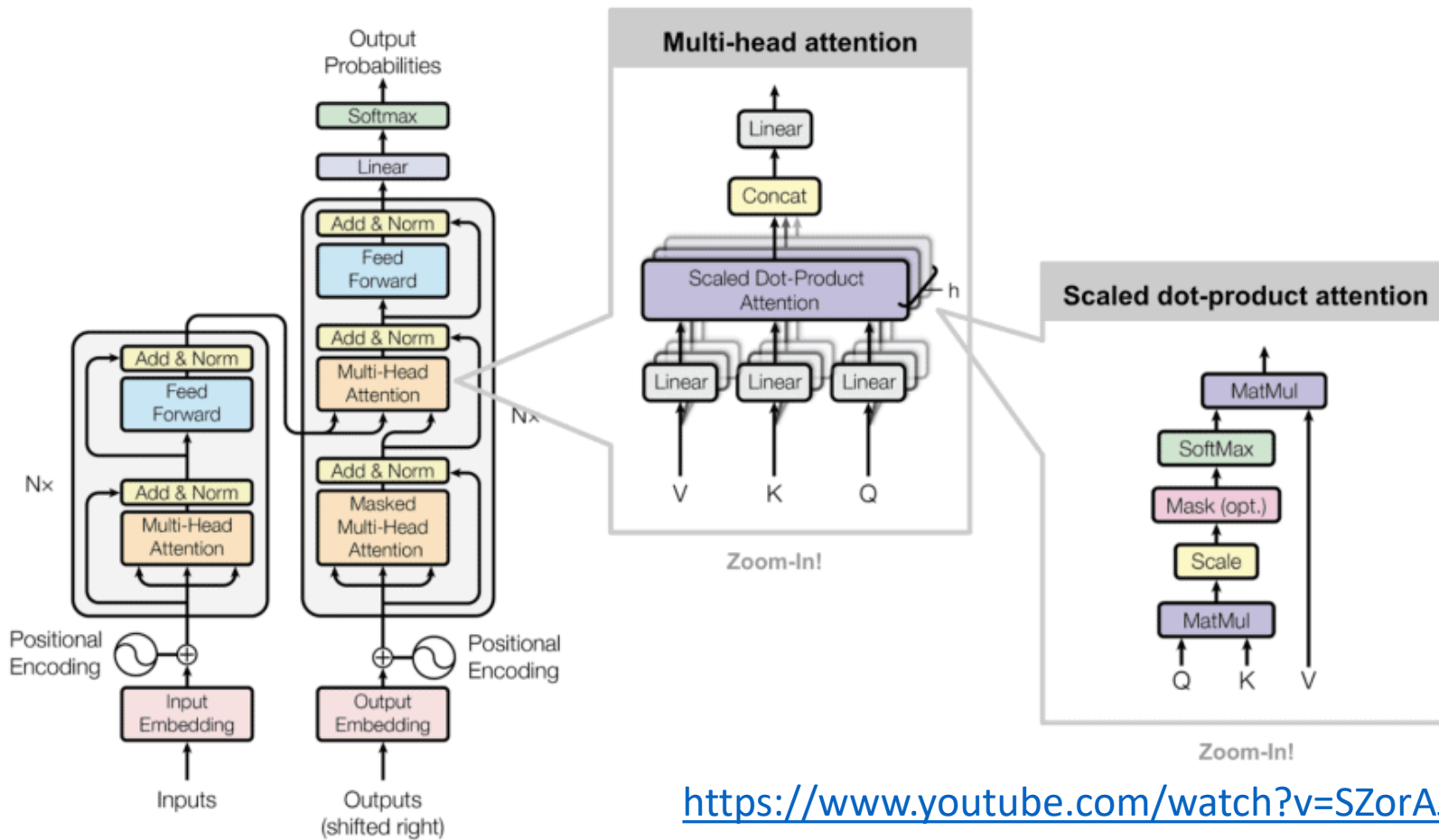






# Transformer & GPT3

Attention  
is All You  
Need!



<https://www.youtube.com/watch?v=SZorAJ4I-sA>



# huggingface.co

The screenshot shows the 'Write With Transformer' web application interface. The browser address bar displays 'transformer.huggingface.co/doc/distil-gpt2'. The page header includes a unicorn logo, the title 'Write With Transformer', and the selected model 'distil-gpt2'. Below the header, there are controls for 'Shuffle initial text', 'Trigger autocomplete' (with a 'tab' key indicator), 'Select suggestion' (with up/down arrow and 'enter' key indicators), and 'Cancel suggestion' (with an 'esc' key indicator). A 'Save & Publish' button with a share icon is on the right. The main text area contains the prompt 'Who is Kuan-Ting Lai?'. A dropdown menu shows two suggestions: 'It was a story about a Korean couple who were just starting a new life i...' and 'The world-famous and respected Chinese martial artist Kuan-Ting Lai', with the second option highlighted. On the left, the 'Model & decoder settings' panel shows sliders for 'Model size' (set to 'distilgpt2/small'), 'Top-p' (set to 0.9), 'Temperature' (set to 1), and 'Max time' (set to 1). The browser's bookmark bar at the top lists various categories like '應用程式', 'Downloads', 'Bookmarks', and 'Deep Learning'.

transformer.huggingface.co/doc/distil-gpt2

Write With Transformer **distil-gpt2**

Shuffle initial text Trigger autocomplete or **tab** Select suggestion **↑** **↓** and **enter** Cancel suggestion **esc** Save & Publish

Who is Kuan-Ting Lai?

It was a story about a Korean couple who were just starting a new life i...

The world-famous and respected Chinese martial artist Kuan-Ting Lai

Model & decoder settings

Model size **distilgpt2/small**

Top-p **0.9**

Temperature **1**

Max time **1**

# Key Takeaways

1. Deep learning is a branch of Machine Learning, which is a sub-field of Artificial Intelligence.
2. There are two stages in machine learning: training (learning) and testing (inference).
3. Gradient Descent is used to train NN models by updating weights to minimize the prediction errors.
4. Convolutional Neural Networks (CNN) are used to recognize images.
5. RNN and LSTM are used to recognize sequential data such as text or speech.
6. Generative Adversarial Networks (GANs) can be used to generate fake data.
7. Transformer told us that attention is all you need!
8. Deep Reinforcement Learning can not only play Go, but also study new drugs.





# 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=Ilg3gGewQ5U>
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/6cdc2f1279016e566832bb1234e06d321992dd1fabcdf4a2e0a3e16fc0dc09dc>
8. <https://tectales.com/ai/ai-used-to-identify-different-types-of-brain-injuries.html>
9. [Transformers, explained: Understand the model behind GPT, BERT, and T5](#)
10. <https://transformer.huggingface.co/>
11. <https://www.scienceabc.com/innovation/what-is-artificial-intelligence.html>