Introduction to Deep Learning

Lecture 17

15.076 | Analytics for a Better World

"Deep Learning, oil on canvas, masterpiece by Salvador Dali", a creation by Léonard Boussioux with 30+ prompts in DALL-E

Plan

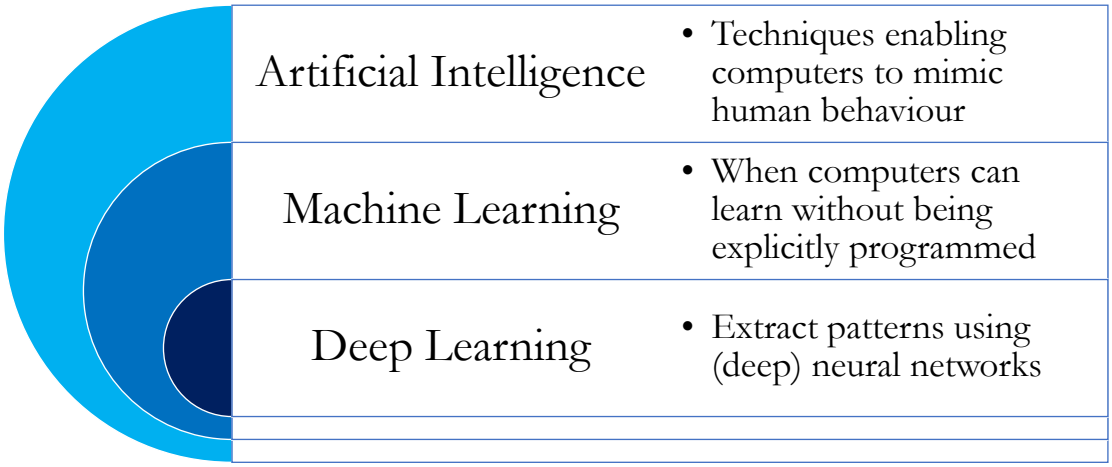
Broad Introduction of AI

Part I: Fundamental Concepts of Neural Networks

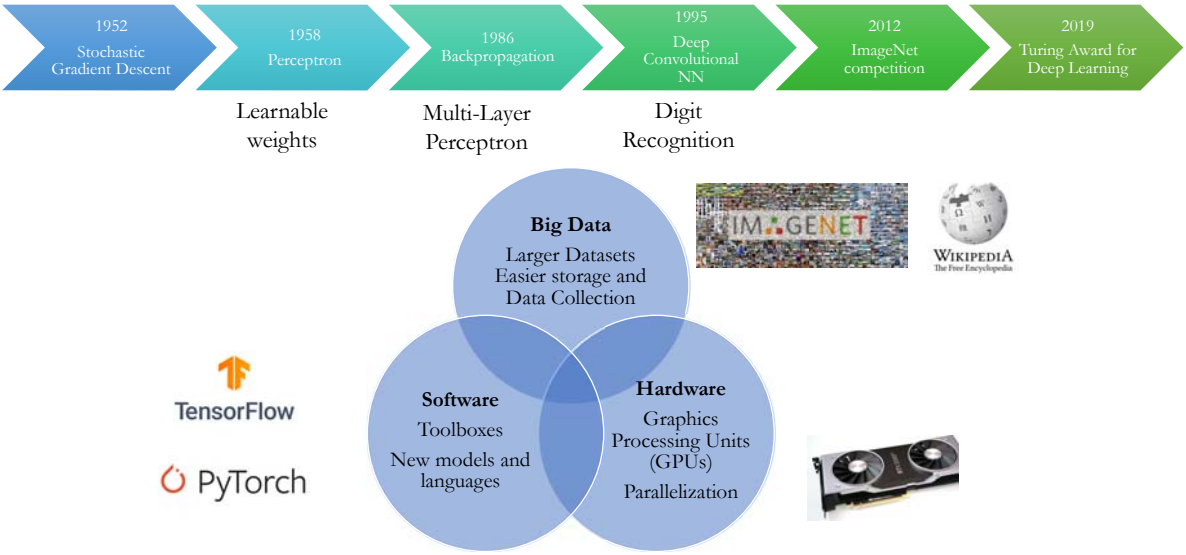
- The perceptron model
- Building neural networks
- Training neural networks
- Handle Overfitting

Part II: Convolutional Neural Networks

Deep Learning in the landscape

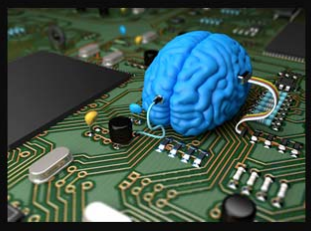


What is the history behind the current AI boom?



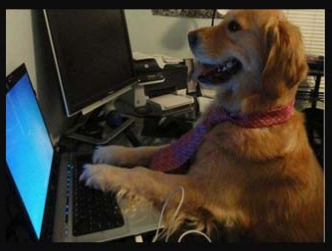
Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do



What mathematicians think I do

What I think I do

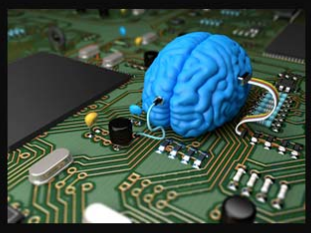
```
from keras import *
```

What I actually do

5

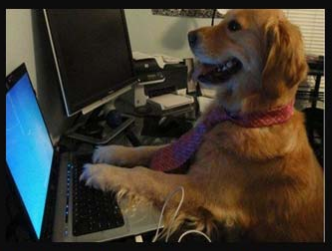
Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do



What mathematicians think I do

What I think I do

~~from keras import *~~

So 2022...

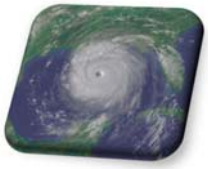
What I actually do

6

Deep Learning

7

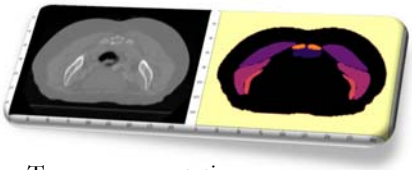
During my PhD, I have used Deep Learning for:



Tropical Cyclone Forecasting

Healthcare Operations

Wildlife Identification



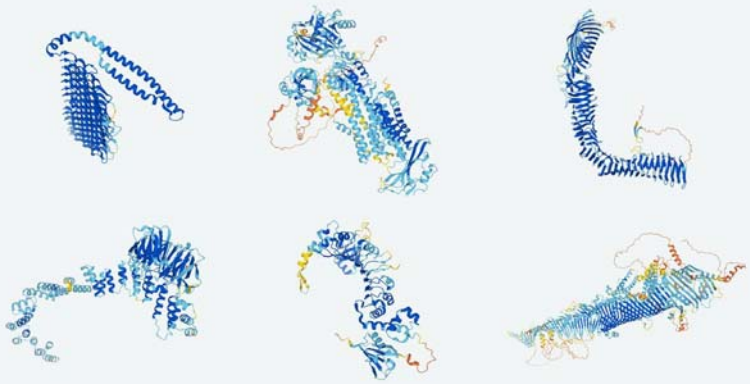
Tumor segmentation

Stamp analytics

8

AlphaFold

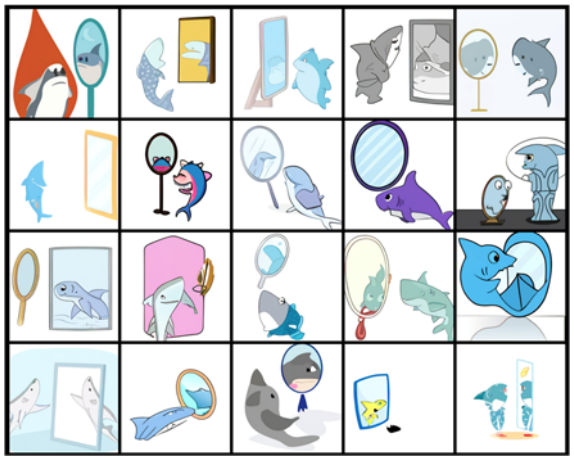
An AI system developed by DeepMind that predicts a protein's 3D structure from its amino acid sequence.



DALL-E in 2022

https://openai.com/blog/dall-e/

BabyShark in an existential crisis after its overwhelming success on YouTube



DALL-E Prompt: an illustration of a baby shark in pajamas staring at its reflection in the mirror

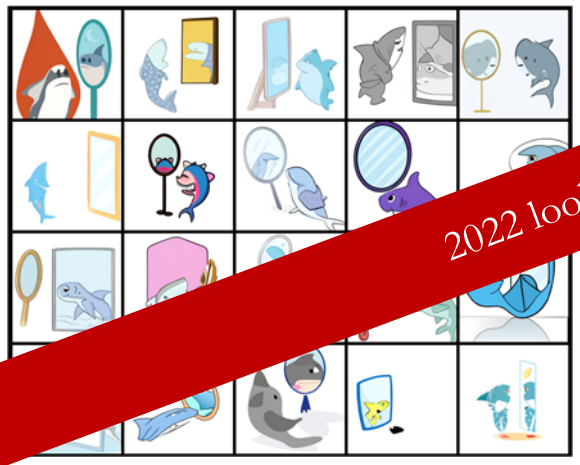
Pikachu very proud for its first day at MIT

DALL-E Prompt: an illustration of a pikachu with a cape holding a calculator

DALL-E in 2022

https://openai.com/blog/dall-e/

BabyShark in an existential crisis after its overwhelming success on YouTube



DALL-E Prompt: an illustration of a baby shark in pajamas staring at its reflection in the mirror

Pikachu very proud for its first

DALL-E Prompt: an illustration of a pikachu with a cape holding a calculator

2022 looks so retro…

Same prompts in 2023…

DALL-E 2

Midjourney

DALL-E 2

I launched my digital artist career! Exhibition coming soon!

With great power comes great responsibility

AI that recognizes faces from the whole world

Real-time satellite analysis to get trading side information

Bias, racism

Killer drones, AI-powered weapons

15

With great power comes great responsibility

AI

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

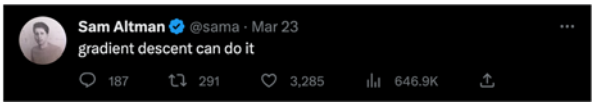
tion

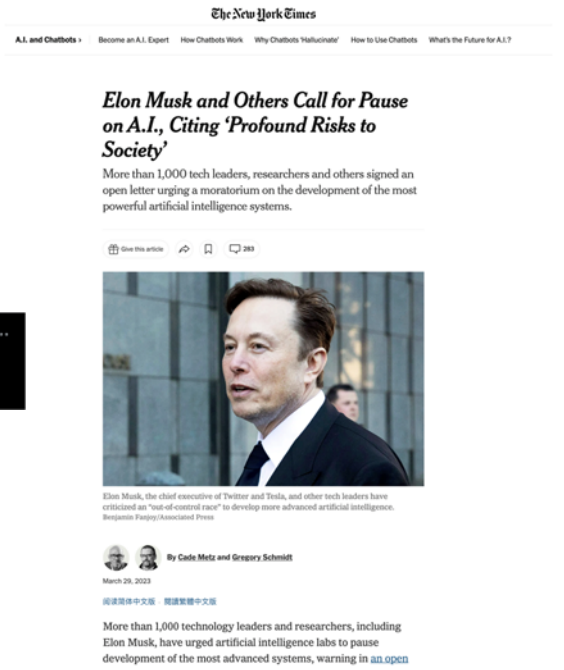
Bias, racism

Killer drones, AI-powered weapons

16

AGI doomers,
AI moratorium,
AI alignment,
Scale is all you need.





The New York Times

A.I. and Chatbots › Become an A.I. Expert How Chatbots Work Why Chatbots 'Hallucinate' How to Use Chatbots What's the Future for A.I.?

Elon Musk and Others Call for Pause on A.I., Citing 'Profound Risks to Society'

More than 1,000 tech leaders, researchers and others signed an open letter urging a moratorium on the development of the most powerful artificial intelligence systems.

Give this article 283

Elon Musk, the chief executive of Twitter and Tesla, and other tech leaders have criticized an "out-of-control race" to develop more advanced artificial intelligence.
Benjamin Fanjoy/Associated Press

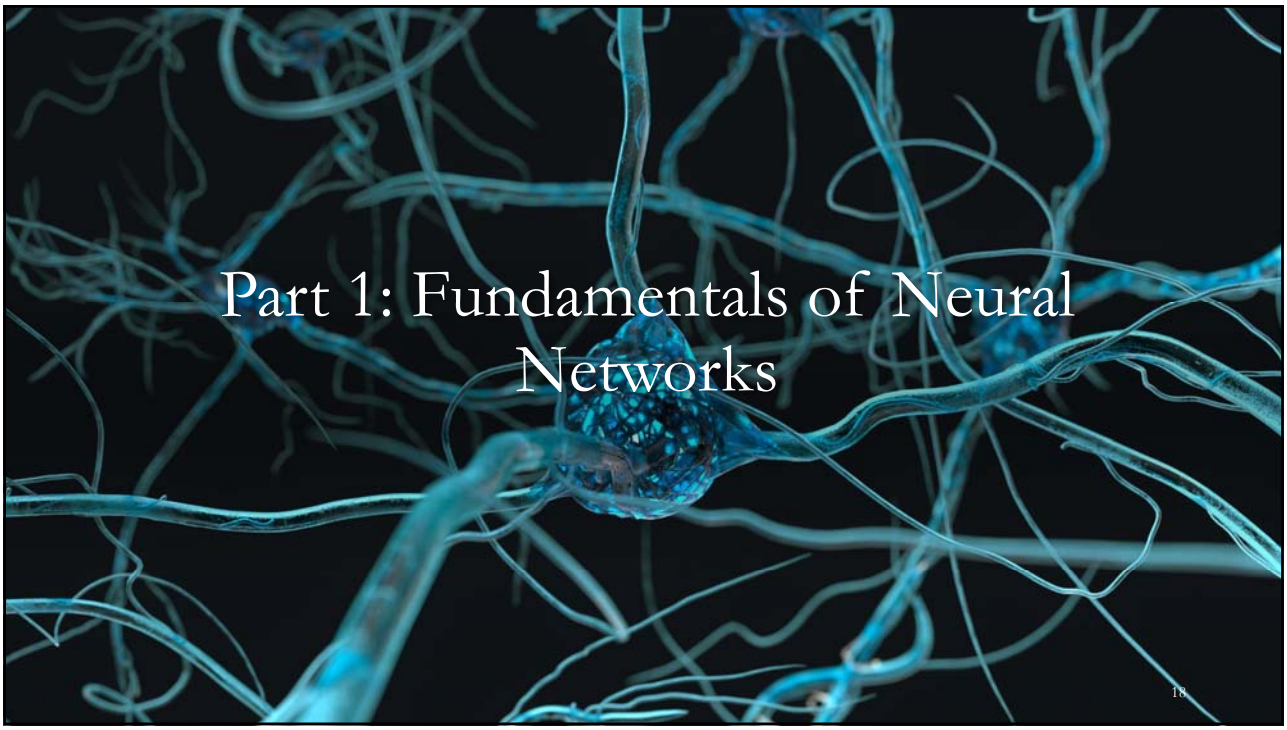
By Cade Metz and Gregory Schmidt

March 29, 2023

阅读简体中文版 · 閱讀繁體中文版

More than 1,000 technology leaders and researchers, including Elon Musk, have urged artificial intelligence labs to pause development of the most advanced systems, warning in an open

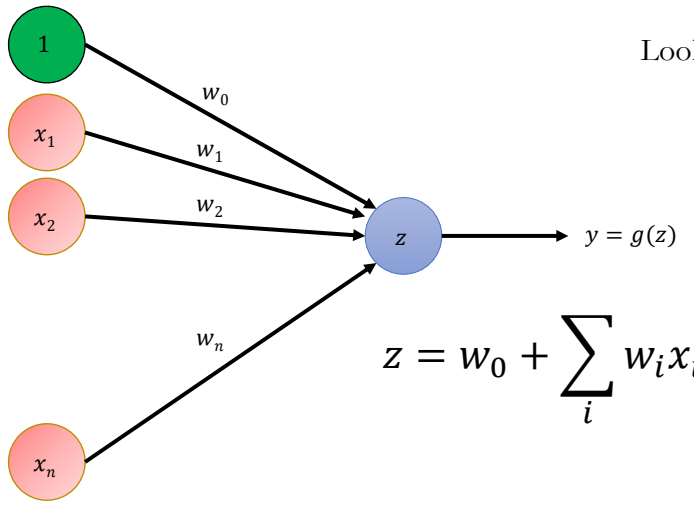
17



Part 1: Fundamentals of Neural Networks

18

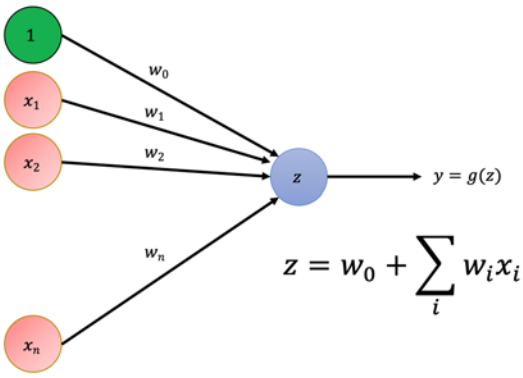
A simplified perceptron



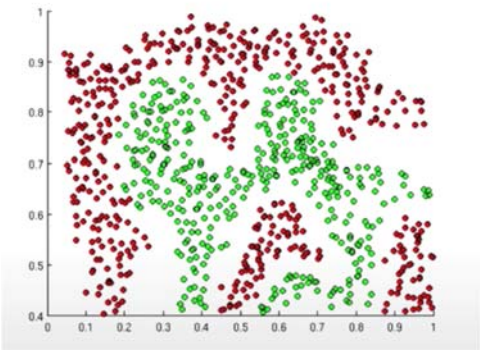
Looks like a regression!

$$z = w_0 + \sum_i w_i x_i$$

How can we separate the red and green points?

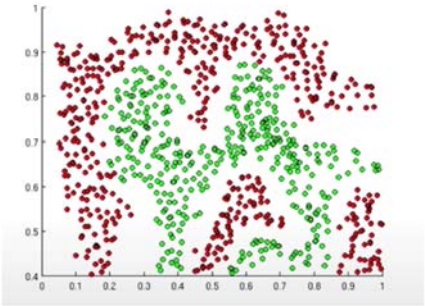


$$z = w_0 + \sum_i w_i x_i$$

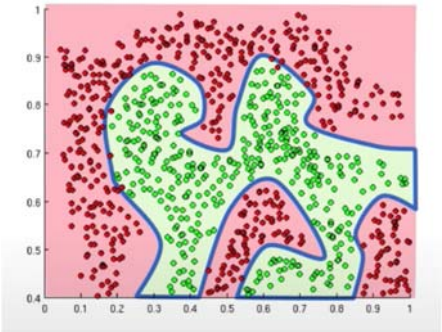


Using activation functions!

$z = w_0 + \sum_i w_i x_i \longrightarrow y = g(z)$, where g is a non-linear activation function.

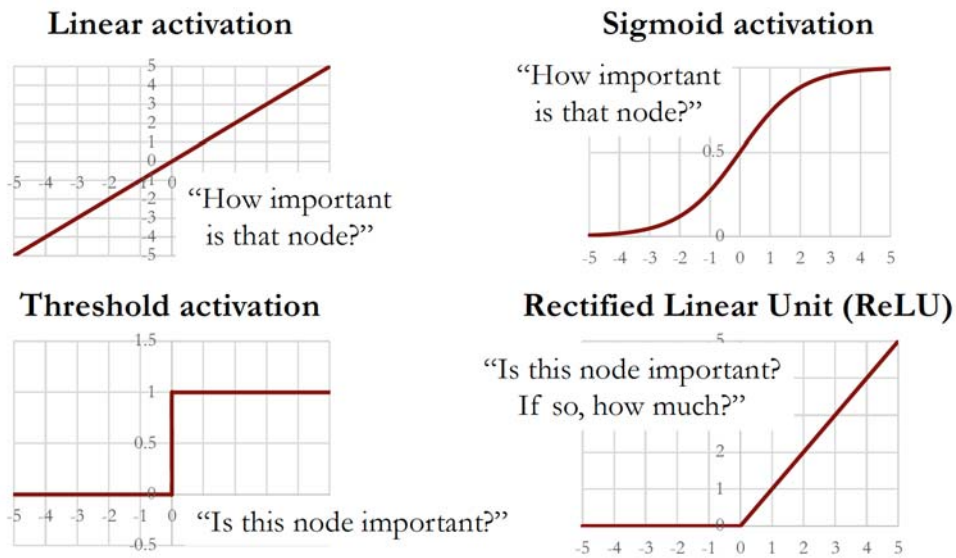


Linear activation functions produce linear decisions no matter the network size.

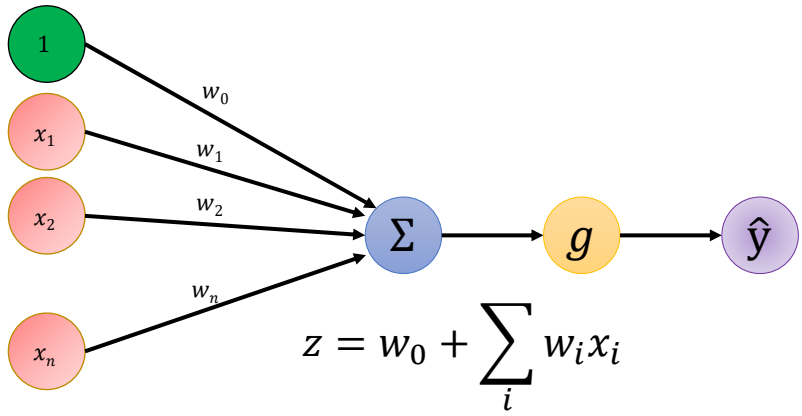


Non-linear activations can help approximate arbitrarily complex functions.

Classic Activation Functions



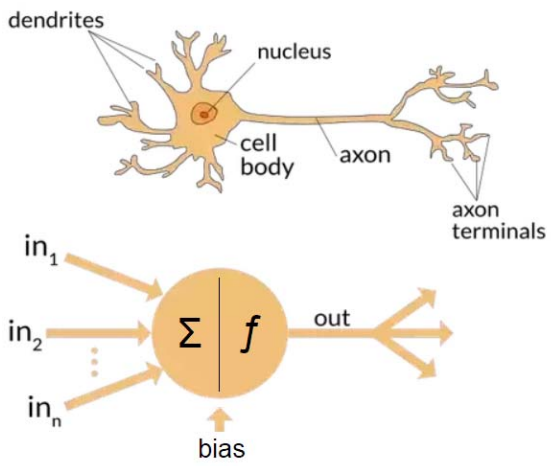
The full perceptron: forward propagation



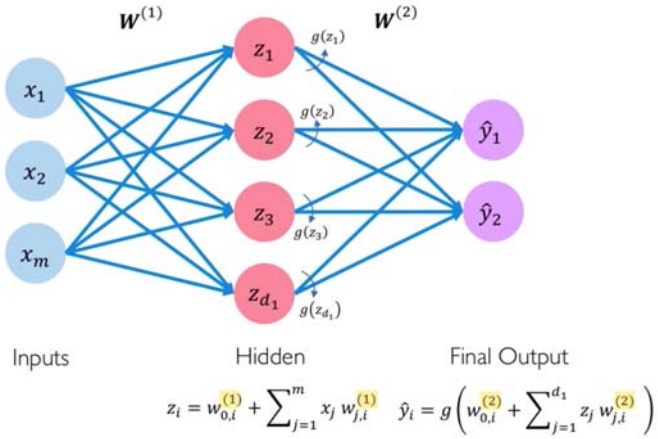
$$z = w_0 + \sum_i w_i x_i$$

$$\hat{y} = g(\sum_i w_i x_i)$$

Artificial Neuron vs Brain Neuron



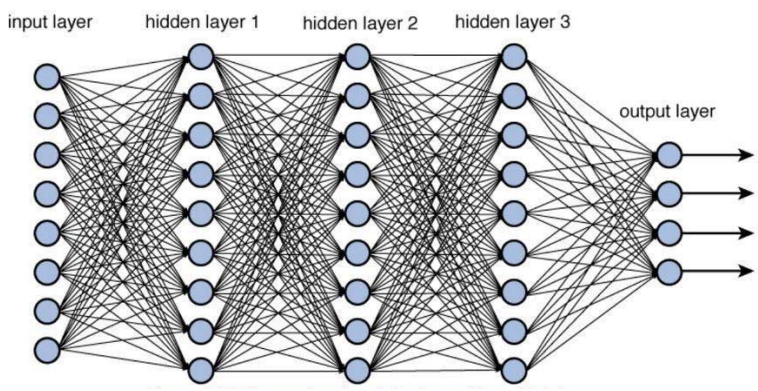
One-layer Neural Network



$$z_i = w_{0,i}^{(1)} + \sum_{j=1}^{m} x_j \, w_{j,i}^{(1)} \qquad \hat{y}_i = g\left(w_{0,i}^{(2)} + \sum_{j=1}^{d_1} z_j \, w_{j,i}^{(2)}\right)$$

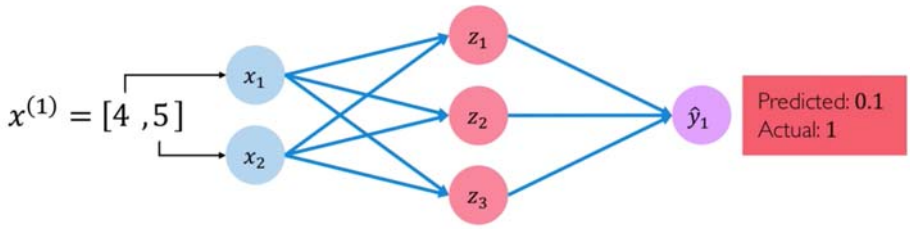
Picture source: http://introtodeeplearning.com/

Deep Neural Network



Loss function

The loss of our network measures the cost incurred from incorrect predictions.

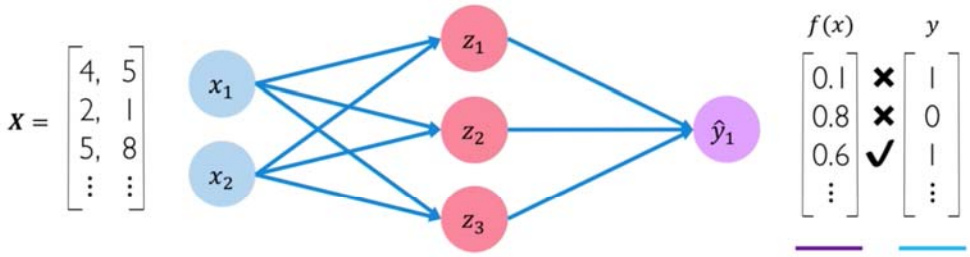


$$\mathcal{L}\big(\underbrace{f\left(x^{(i)}; \boldsymbol{W}\right)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}}\big)$$

Picture source: http://introtodeeplearning.com/

Empirical Loss

The empirical loss measures the total loss over our entire dataset.

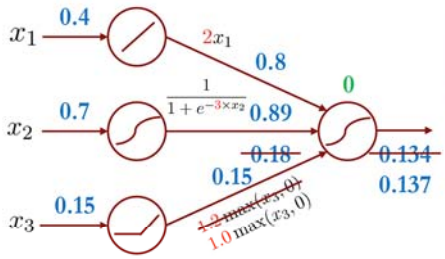


Also known as:
- Objective function
- Cost function
- Empirical Risk

$$\boldsymbol{J}(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\big(\underbrace{f\left(x^{(i)}; \boldsymbol{W}\right)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}}\big)$$

Picture source: http://introtodeeplearning.com/

Finding the Best Model Fit (explanation on the board)



$$y = \frac{1}{1+e^{-\left(2x_1+\frac{1}{1+e^{-3\times x_2}}+1.2\max(x_3,0)\right)}}$$

$$y = \frac{1}{1+e^{-\left(2x_1+\frac{1}{1+e^{-3\times x_2}}+1.0\max(x_3,0)\right)}}$$

- The parameters ("weights") are chosen to maximize the model's quality of fit—so the predictions $\hat{y}_i$ match the observations y_i

regression

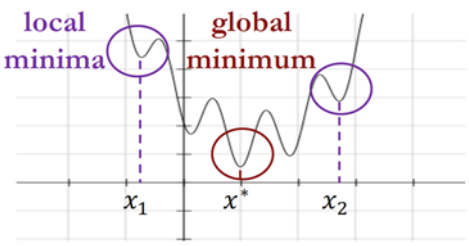
$$\min \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

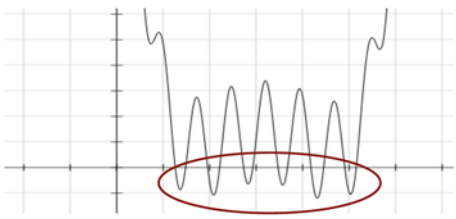
classification

$$\min -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

29

Nonlinear Optimization





Deep learning loss functions often have many local minima.

→ Deep learning outputs may not be the best possible ones.

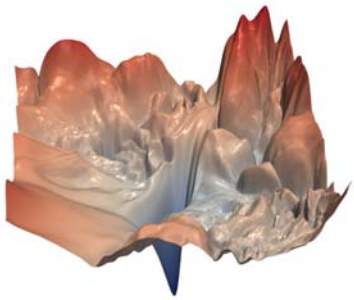
However, many local minima often lead to similar solutions.

→ Deep learning outputs are often surprisingly "good enough"!

30

Training Neural Networks looks Hard!

Problem 1: Need to optimize a huge number of weights $\boldsymbol{W}$.
Problem 2: Loss landscape not smooth at all.

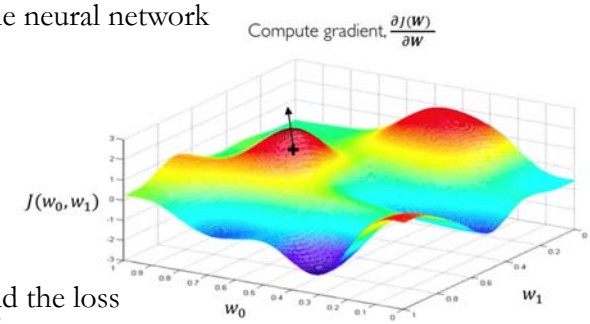


31

Fortunately there is Gradient Descent!

Objective: Minimizing average loss function of the neural network

$$\boxed{\mathcal{J}(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(f(\boldsymbol{x}_i, \boldsymbol{W}), y_i\right)}$$

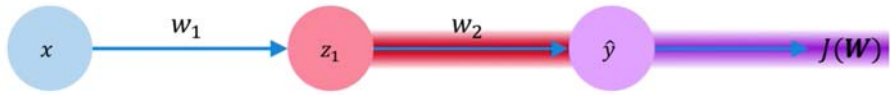


Algorithm:

1. Initialize weight matrix $\boldsymbol{W}$
2. **Backpropagation**: Loop until convergence…
   a. **Forward pass**: Compute the predictions and the loss
   b. **Backward pass**: Compute the gradient $\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$
   (i.e., what happens if we change the weights just a tiny bit)
   c. **Update**: $\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$

   α: "learning rate"
3. Return the weight matrix $\boldsymbol{W}$ and the predictions.

And backpropagation with chain rule



$$\frac{\partial J(\boldsymbol{W})}{\partial w_2} = \frac{\partial J(\boldsymbol{W})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w_2}$$

Picture source: http://introtodeeplearning.com/

But how do you choose the learning rate?

Remember:
Optimization through gradient descent

$$\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

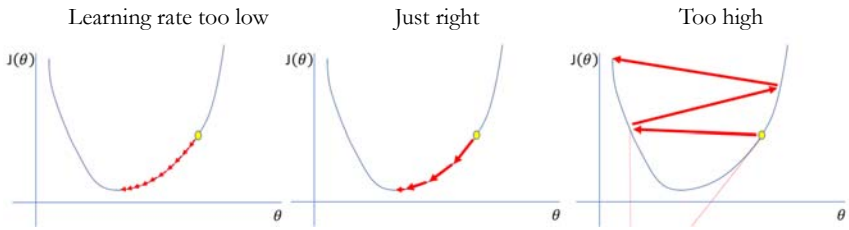
How can we set
the learning rate?

Ideas?

Adaptive Learning Rate

Learning rate is not fixed anymore, instead we can adapt it depending on:
- How large gradients are,
- How fast the learning is happening (momentum),

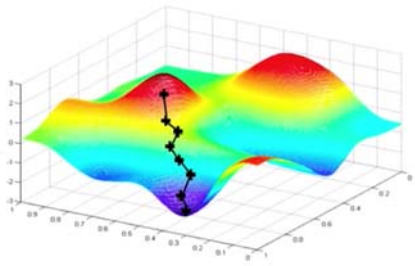
In Deep Learning, Adam is an extremely popular method and does all this automatically. You still need to choose a base value (typically between $10^{-3}/5.10^{-5}$ that depends on the task and architecture.)



Another problem: computing gradients is expensive!

Objective: Minimizing average loss function of the neural network

$$\mathcal{J}(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(f(\boldsymbol{x}_i, \boldsymbol{W}), y_i\right)$$



Algorithm:

1. Initialize weight matrix $\boldsymbol{W}$
2. **Backpropagation**: Loop until convergence…
   a. **Forward pass**: Compute the predictions and the loss
   b. **Backward pass**: Compute the gradient $\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$ → Expensive to compute
   c. **Update**:
   $$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$$
3. Return the weight matrix $\boldsymbol{W}$ and the predictions.

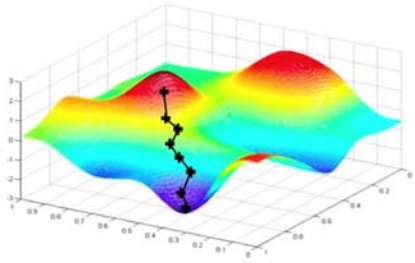
Solution: Stochastic Gradient Descent

Objective: Minimizing average loss function of the neural network

$$\mathcal{J}(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(f(\boldsymbol{x}_i, \boldsymbol{W}), y_i\right)$$

Algorithm:

1. Initialize weight matrix $\boldsymbol{W}$
2. **Backpropagation**: Loop until convergence…
   a. **Forward pass**: Compute the predictions and the loss
   b. **Backward pass**: Compute the gradient $\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$
   c. **Update**:
   $$\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{W}}$$
3. Return the weight matrix $\boldsymbol{W}$ and the predictions.



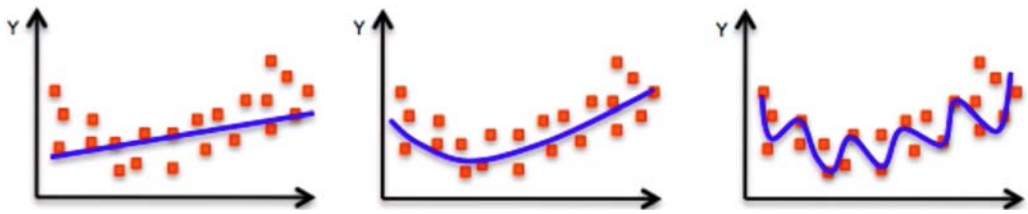
Expensive to compute

Batch Gradient Descent

$$\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{B}\sum_{k=1}^{B}\frac{\partial J_k(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

We use mini-batches while training allowing for a smoother convergence and larger learning rates.

Another classic ML problem: overfitting



Underfitting
Model does not have capacity to fully learn the data

Ideal Fit

Overfitting
Too complex, does not generalize well

Overfitting in Neural Networks 1/3

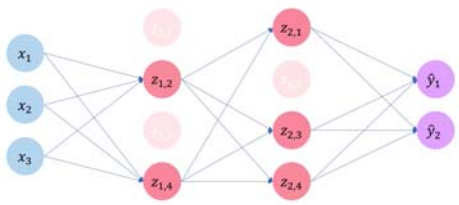
Overfitting #1: **The neural network is "too complex"**

- Too many hidden layers,
- Too many nodes in each layer.

→ Some neurons learn all the signal and others become useless, which is bad for generalization.

Regularization idea #1: **Drop-out**

Randomly dropping (i.e., setting to 0) some nodes in the network during training to force the network to not rely on just a few nodes.



39

Overfitting in Neural Networks 2/3

Overfitting #2: The neural network is "**too well trained**".

Regularization idea #2: **Early Stopping**

Stop the training early—after a moderate number of iterations—before the in-sample loss is completely minimized.

40

Overfitting in Neural Networks 3/3

Regularization idea #3: **Embed regularization** ideas in the training of neural networks, in the objective of the non-linear optimization.

- Recall the objective of minimizing the empirical loss:

$$\boxed{\mathcal{J}(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(f(\boldsymbol{x}_i, \boldsymbol{W}), y_i\right)}$$

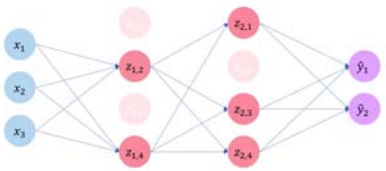
- We can penalize this loss-minimization objective as follows (analog of ridge regression in the context of neural networks)

$$\boxed{\min\left(\mathcal{J}(\boldsymbol{W}) + \lambda \times \sum_{i \in \text{layers}} \sum_{k \in \text{nodes of } i} w_{ik}^2\right)}$$

41

Summary: The 3 Regularization Techniques

Regularization 1: Dropout

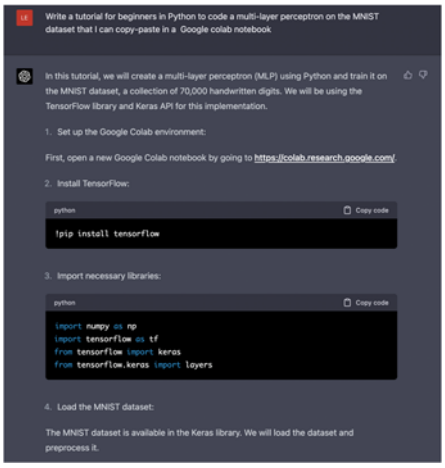


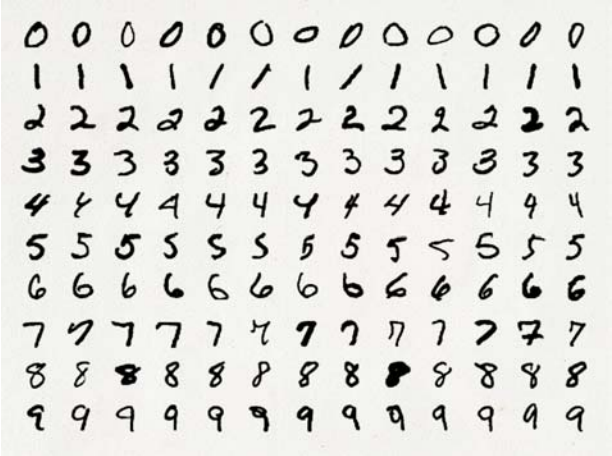
Regularization 2: Early Stopping

Regularization 3: L1 or L2 penalty on the NN weights.

Picture sources: http://introtodeeplearning.com/

Quick ChatGPT demo: Classifying MNIST digits





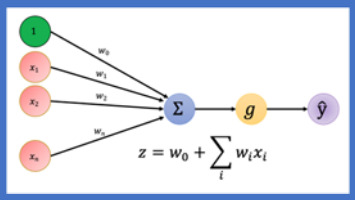
MNIST dataset

43

Summary of the neural networks fundamentals

The Perceptron

- Structure
- Nonlinear activation functions

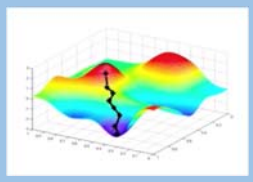


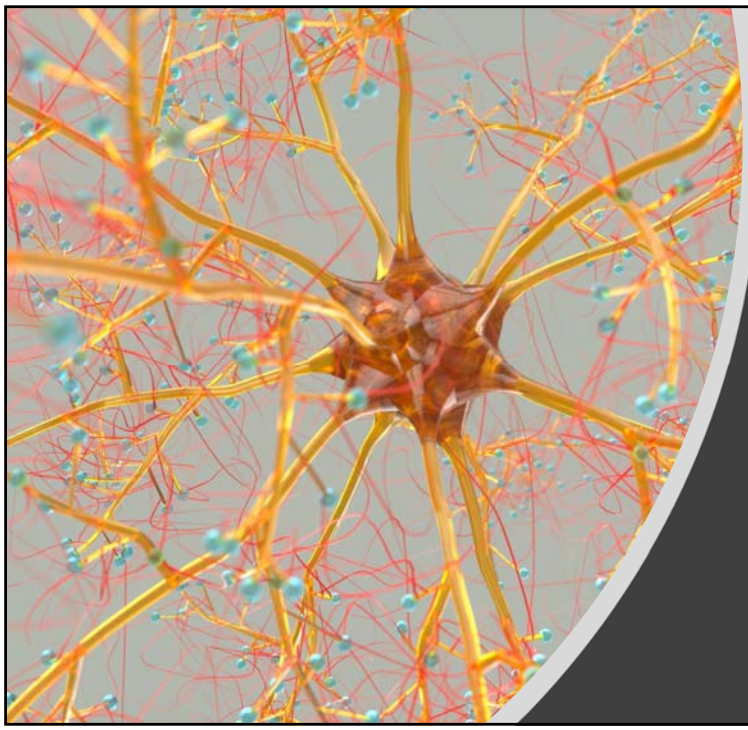
Neural Networks

- Stacking layers of perceptrons
- Backpropagation and gradient descent

Training in Practice

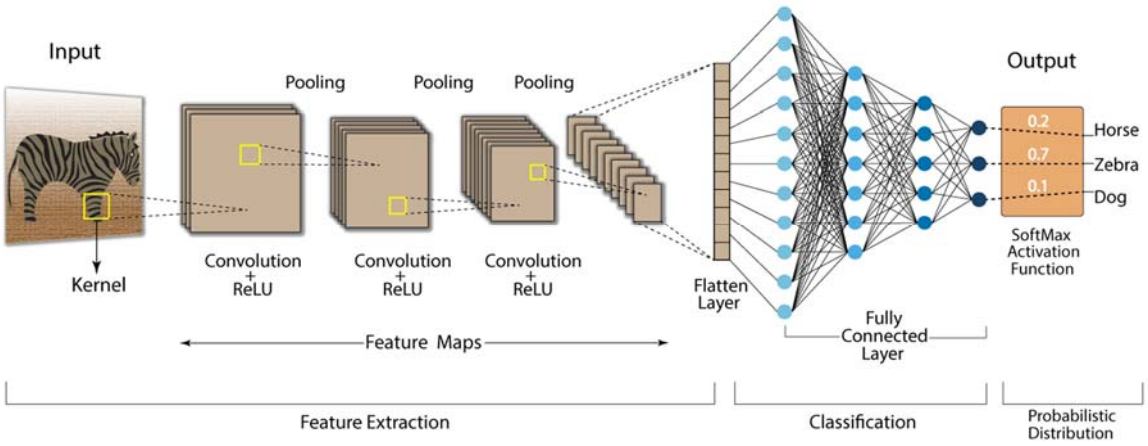
- Adaptive learning
- Batching
- Regularization





Convolutional Neural Networks

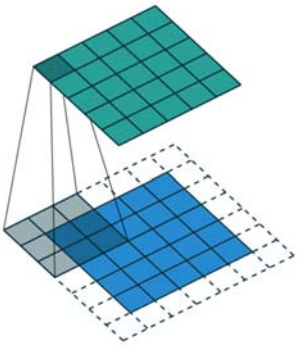
Overall Architecture of a CNN



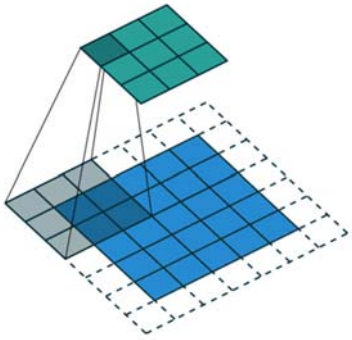
Images source: https://www.analyticsvidhya.com/blog/2021/05/20-questions-to-test-your-skills-on-cnn-convolutional-neural-networks/

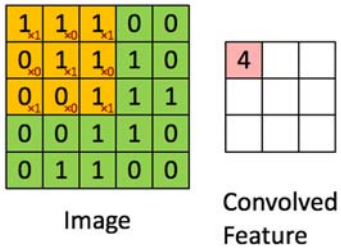
Convolution Operation

Padding: 1
Stride: 1



Padding: 1
Stride: 2

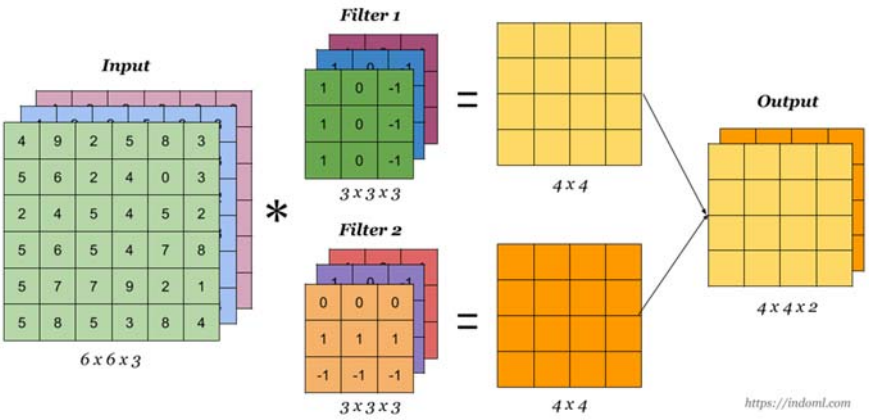




47

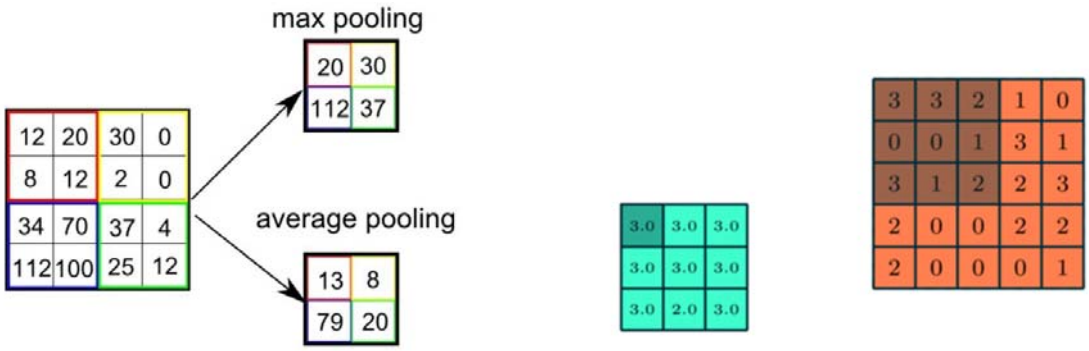
Images source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3-dimensional data and Conv2D (optional)



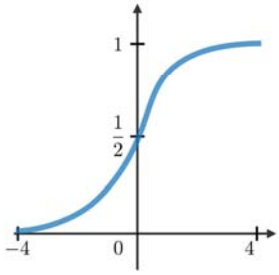
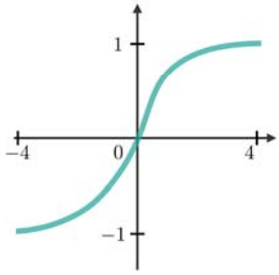
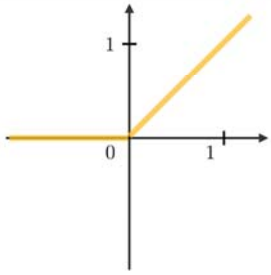
48

Pooling

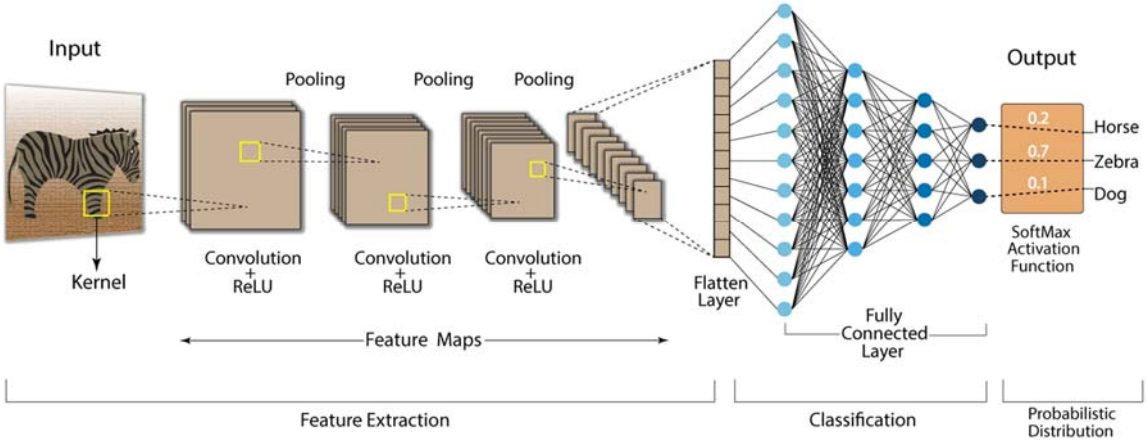


Images source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Activation functions

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

Now we can deal with images!

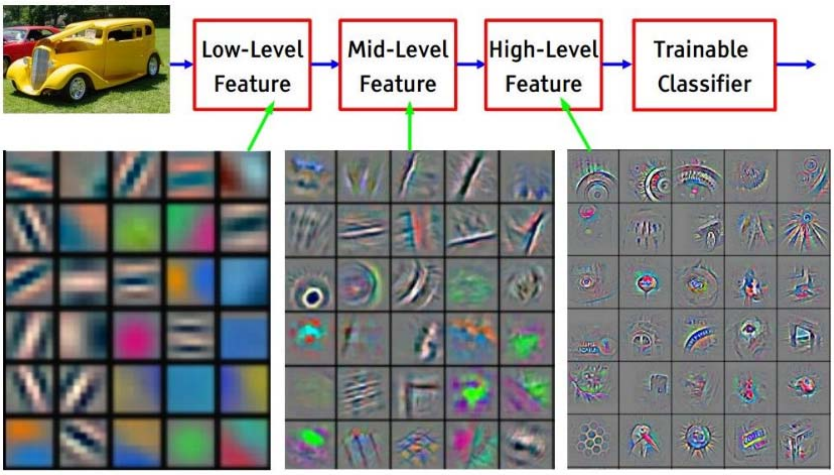


Check the fancy visualization of features learned with ConvNets: https://microscope.openai.com/models

Image source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

51

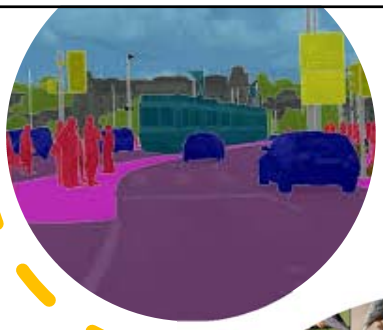
Visualizing CNN filters



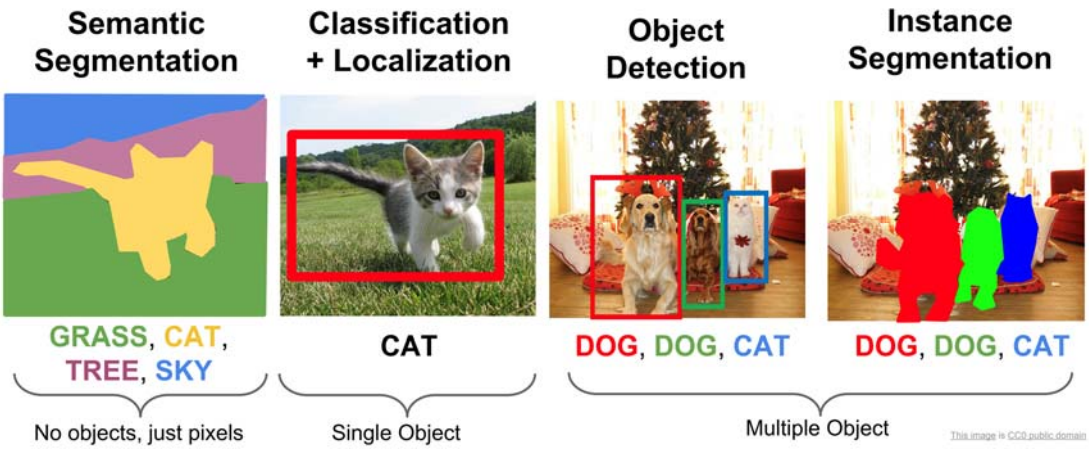
52

What kind of tasks can we perform?

- Classification, Regression
- Segmentation
- Prediction of next image
- Content generation
- Feature extraction
- Descriptions
- …

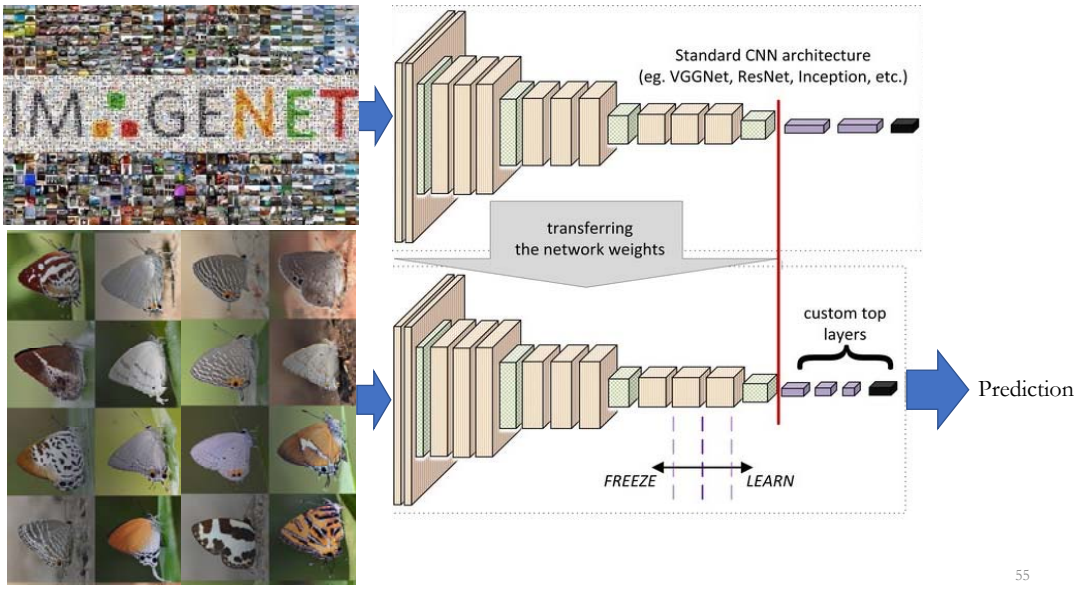


Other Computer Vision Tasks



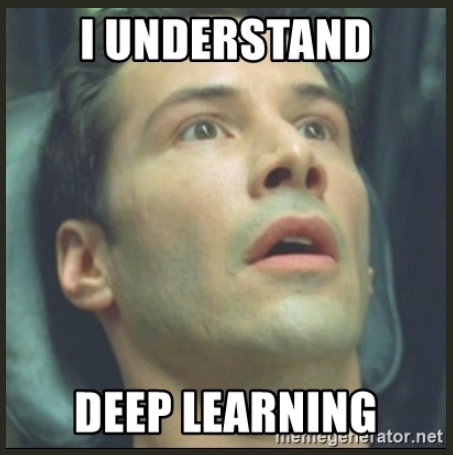
54

Transfer Learning (optional)



55

Please ask any questions!



56

Resources

MIT Spring Courses that may be offered again in the future:

6.S985 Artificial Intelligence for Business

6.S986 Large Language Models and Beyond

15.S04 Special Seminar in Management (Intro to Deep Learning)

The famous Introduction to Deep Learning class of MIT covers more because this is a full week of content. I highly recommend checking their slides in the topics of interest! http://introtodeeplearning.com/

The Computer Vision class of MIT (you have a quick glimpse at the subjects of interest): http://6.869.csail.mit.edu/sp22/

The NLP course of MIT: https://www.mit.edu/~jda/teaching/6.864/

Berkeley course about DL. Excellent too! I have looked at the slides about Transformers there several times in the past! https://cs182sp21.github.io/

Stanford course about CNNs, excellent as well: http://cs231n.stanford.edu/

58