2019 embedded VISION SUMMIT

An Introduction to Machine Learning and How to Teach Machines to See

Facundo Parodi
Tryolabs
May 2019

tryo·labs

Who are we?

© 2019 Tryolabs

Outline

Machine Learning

- Types of Machine Learning Problems
- Steps to solve a Machine Learning Problem

Deep Learning

- Artificial Neural Networks

Image Classification

- Convolutional Neural Networks

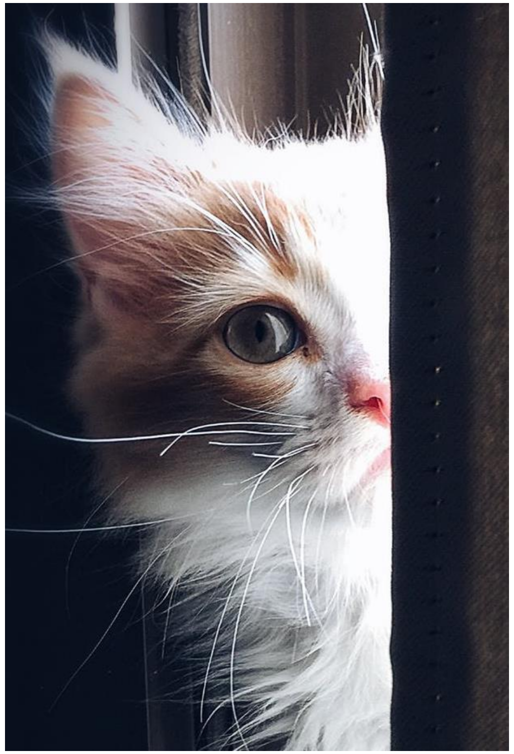
What is a Cat?

2019 embedded VISION SUMMIT

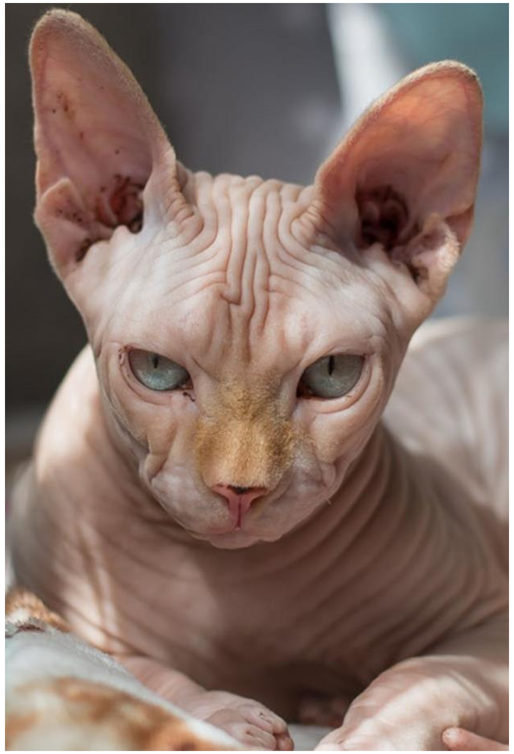
tryo·labs

What is a Cat?

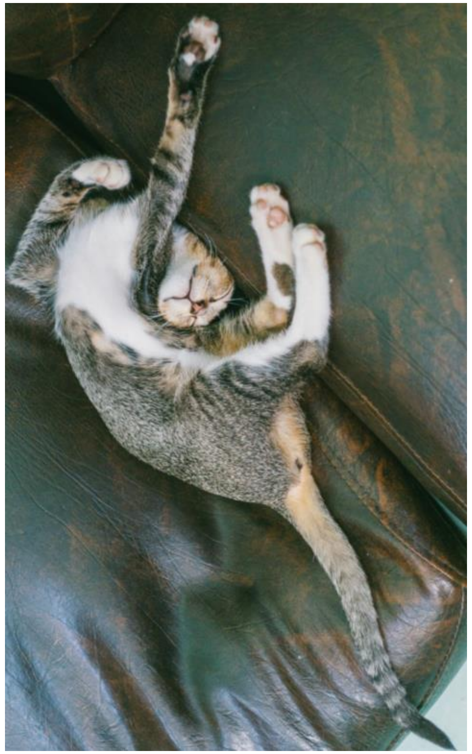
What is a Cat?



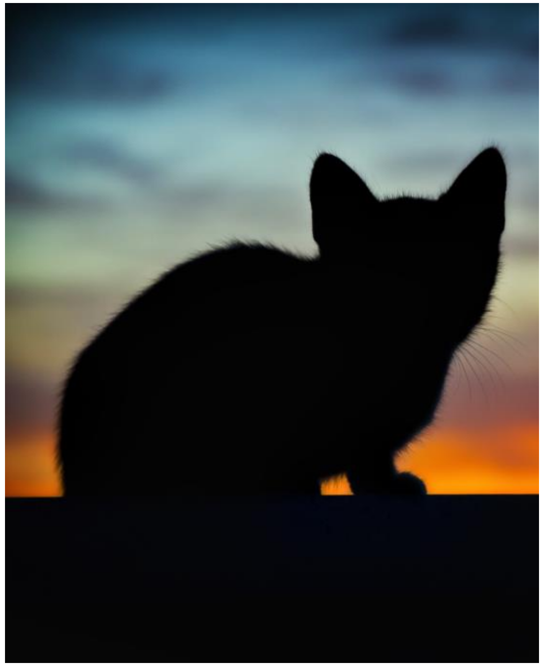
Occlusion



Diversity



Deformation



Lighting variations

Introduction to Machine Learning

2019 embedded VISION SUMMIT

tryo·labs

What is Machine Learning?

The subfield of computer science that "gives computers the ability to learn without being explicitly programmed".
(Arthur Samuel, 1959)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."
(Tom Mitchell, 1997)

$\underbrace{\textbf{Using data}}_{\text{Training}}$ for $\underbrace{\textbf{answering questions}}_{\text{Predicting}}$

The Big Data Era

Data

Data already available everywhere

Low storage costs: everyone has several GBs for “free”

Hardware more **powerful** and **cheaper** than ever before

Devices

Everyone has a computer fully packed with sensors:

- GPS
- Cameras
- Microphones

Permanently connected to Internet

Services

Cloud Computing:

- Online storage
- Infrastructure as a Service

User applications:

- YouTube
- Gmail
- Facebook
- Twitter

Types of Machine Learning Problems

- Supervised
- Unsupervised
- Reinforcement

Types of Machine Learning Problems

Supervised

Unsupervised

Reinforcement

Learn through **examples** of which we know the desired output (what we want to predict).

Is this a cat or a dog?

Are these emails spam or not?

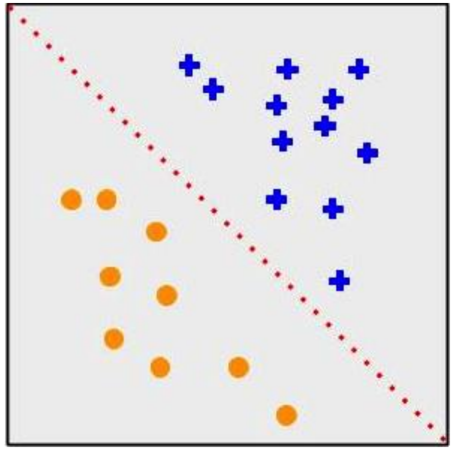
Predict the market value of houses, given the square meters, number of rooms, neighborhood, etc.

Types of Machine Learning Problems

- Supervised
  - Classification
  - Regression
- Unsupervised
- Reinforcement

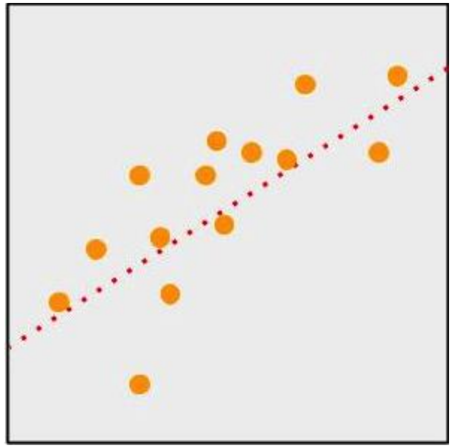
Classification

Output is a **discrete** variable (e.g., cat/dog)



Regression

Output is **continuous** (e.g., price, temperature)



Types of Machine Learning Problems

Supervised

Unsupervised

Reinforcement

There is **no *desired* output**. Learn something about the data. *Latent* relationships.

I have photos and want to put them in 20 groups.

I want to find anomalies in the credit card usage patterns of my customers.

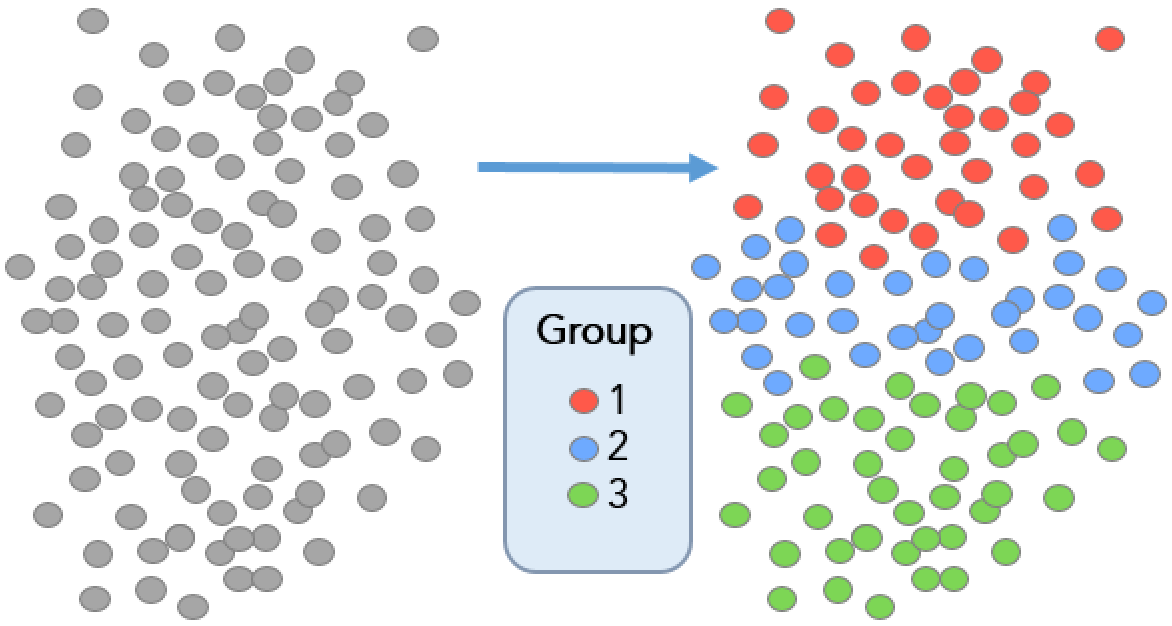
Types of Machine Learning Problems

Supervised

Unsupervised

Reinforcement

Useful for learning structure in the data (**clustering**), hidden correlations, reduce dimensionality, etc.



Types of Machine Learning Problems

Supervised

Unsupervised

Reinforcement

An agent **interacts** with an **environment** and watches the result of the interaction.

Environment gives feedback via a positive or negative **reward signal**.

Steps to Solve a Machine Learning Problem

Data Gathering
Collect data from various sources

Data Preprocessing
Clean data to have homogeneity

Feature Engineering
Making your data more useful

Algorithm Selection & Training
Selecting the right machine learning model

Making Predictions
Evaluate the model

Data Gathering

Might depend on human work

- Manual labeling for supervised learning.
- Domain knowledge. Maybe even experts.

May come for free, or “sort of”

- E.g., Machine Translation.

The more the better: Some algorithms need large amounts of data to be useful (e.g., neural networks).

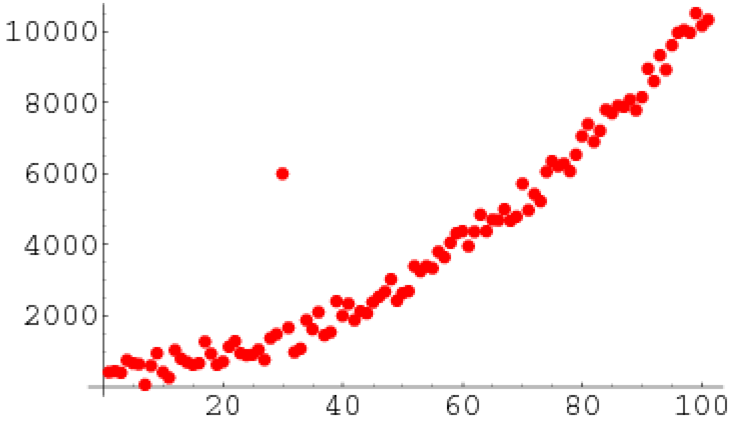
The **quantity** and **quality** of data dictate the model **accuracy**

Data Preprocessing

Is there anything **wrong** with the data?

- Missing values
- Outliers
- Bad encoding (for text)
- Wrongly-labeled examples
- Biased data
  - Do I have many more samples of one class than the rest?

Need to fix/remove data?



Feature Engineering

What is a feature?

> *A feature is an individual measurable property of a phenomenon being observed*

Our inputs are represented by a **set of features.**

To classify spam email, features could be:

- Number of words that have been *ch4ng3d* like this.
- Language of the email (0=English, 1=Spanish)
- Number of emojis

Buy ch34p drugs from the ph4rm4cy now :) :) :)

↓ Feature engineering

(2, 0, 3)

Feature Engineering

Extract **more** information from **existing** data, not adding “new” data per-se

- Making it more **useful**
- With good features, most algorithms can learn **faster**

It can be an art

- Requires thought and knowledge of the data

Two steps:

- Variable transformation (e.g., dates into weekdays, normalizing)
- Feature creation (e.g., n-grams for texts, if word is capitalized to detect names, etc.)

Algorithm Selection & Training

Supervised

- Linear classifier
- Naive Bayes
- Support Vector Machines (SVM)
- Decision Tree
- Random Forests
- k-Nearest Neighbors
- **Neural Networks (Deep learning)**

Unsupervised

- PCA
- t-SNE
- k-means
- DBSCAN

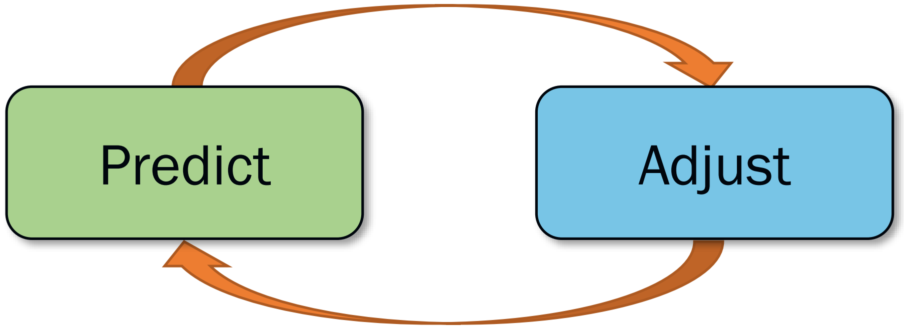
Reinforcement

- SARSA–λ
- Q-Learning

Algorithm Selection & Training

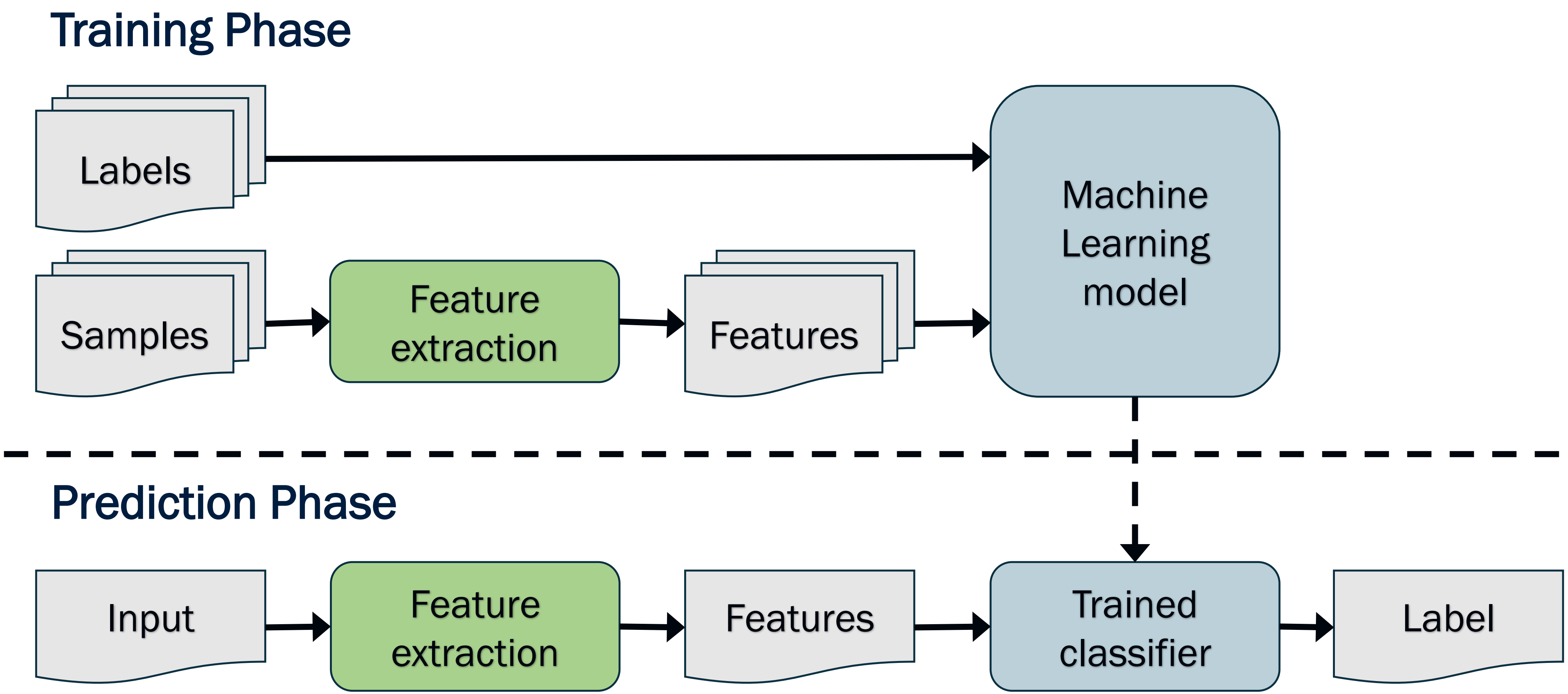
Goal of training: making the correct prediction as often as possible

- Incremental improvement:



- Use of metrics for **evaluating** performance and comparing solutions
- **Hyperparameter tuning:** more an art than a science

Making Predictions



Summary

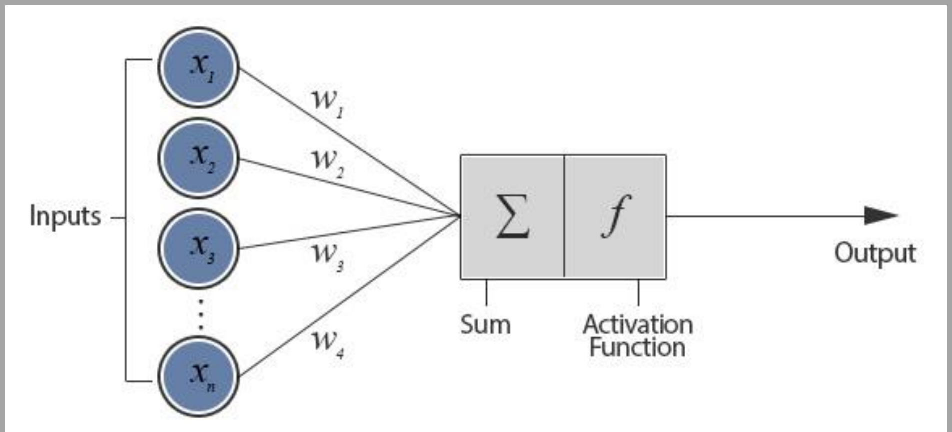
- Machine Learning is intelligent use of data to answer questions
- Enabled by an **exponential** increase in computing power and data availability
- Three big types of problems: **supervised**, **unsupervised**, **reinforcement**
- 5 steps to every machine learning solution:
    1. Data Gathering
    2. Data Preprocessing
    3. Feature Engineering
    4. Algorithm Selection & Training
    5. Making Predictions

Deep Learning

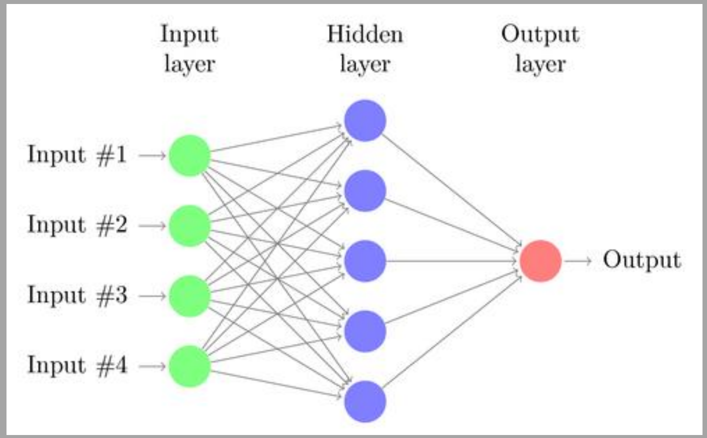
“Any sufficiently advanced technology is indistinguishable from magic.” (Arthur C. Clarke)

Artificial Neural Networks

- First model of artificial neural networks proposed in **1943**
- Analogy to the *human brain* greatly exaggerated
- Given some inputs (x), the network calculates some outputs (y), using a set of weights (w)



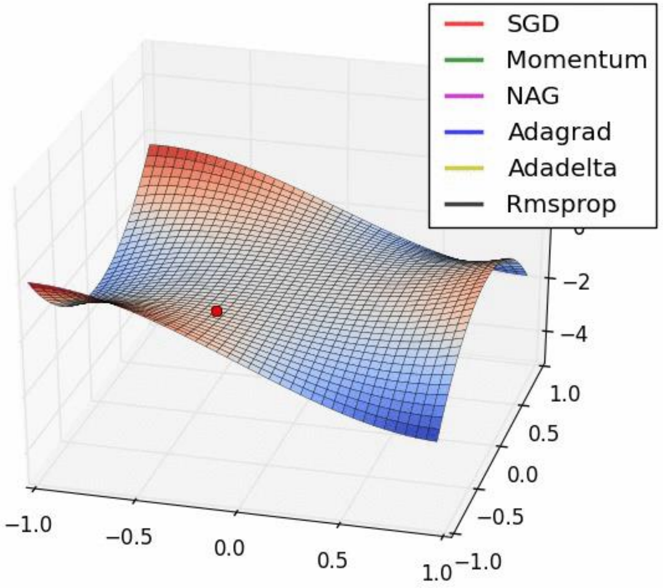
Perceptron *(Rosenblatt, 1957)*



Two-layer Fully Connected Neural Network

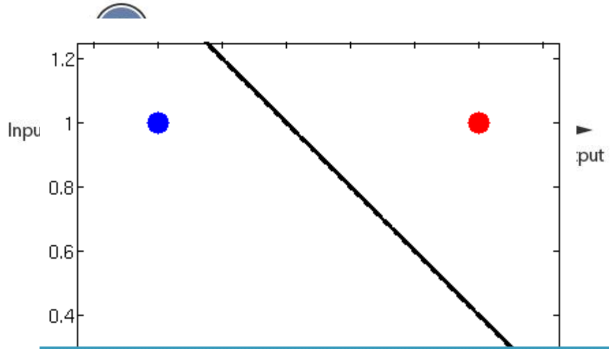
Loss function

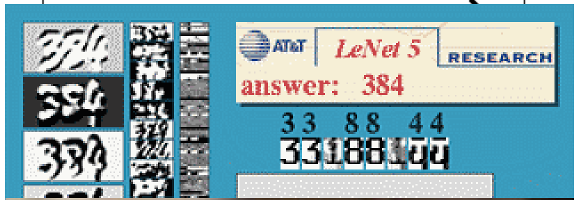
- Weights must be adjusted (**learned** from the data)
- Idea: define a function that tells us **how “close”** the network is to generating the desired output
- **Minimize** the loss ➔ **optimization** problem
- With a **continuous** and **differentiable** loss function, we can apply **gradient descent**



The Rise, Fall, Rise, Fall and Rise of Neural Networks

- Perceptron gained popularity in the 60s
    - Belief that would lead to true AI
- XOR problem and AI Winter (1969 – 1986)
- Backpropagation to the rescue! (1986)
    - Training of multilayer neural nets
    - LeNet-5 (Yann LeCun et al., 1998)
- Unable to scale. Lack of good data and processing power





The Rise, Fall, Rise, Fall and Rise of Neural Networks

- Regained popularity since ~2006.
    - Train each layer at a time
    - Rebranded field as **Deep Learning**
    - Old ideas rediscovered (e.g., Convolution)
- Breakthrough in 2012 with AlexNet (Krizhevsky et al.)
    - Use of GPUs
    - Convolution

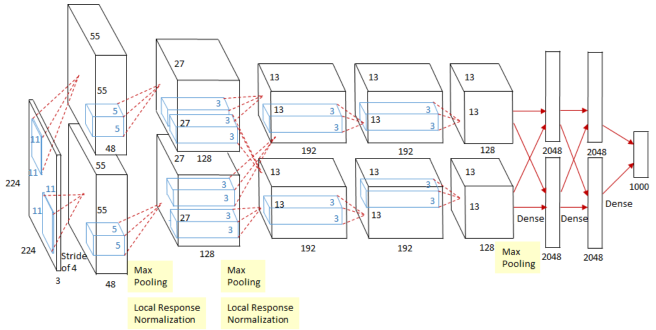
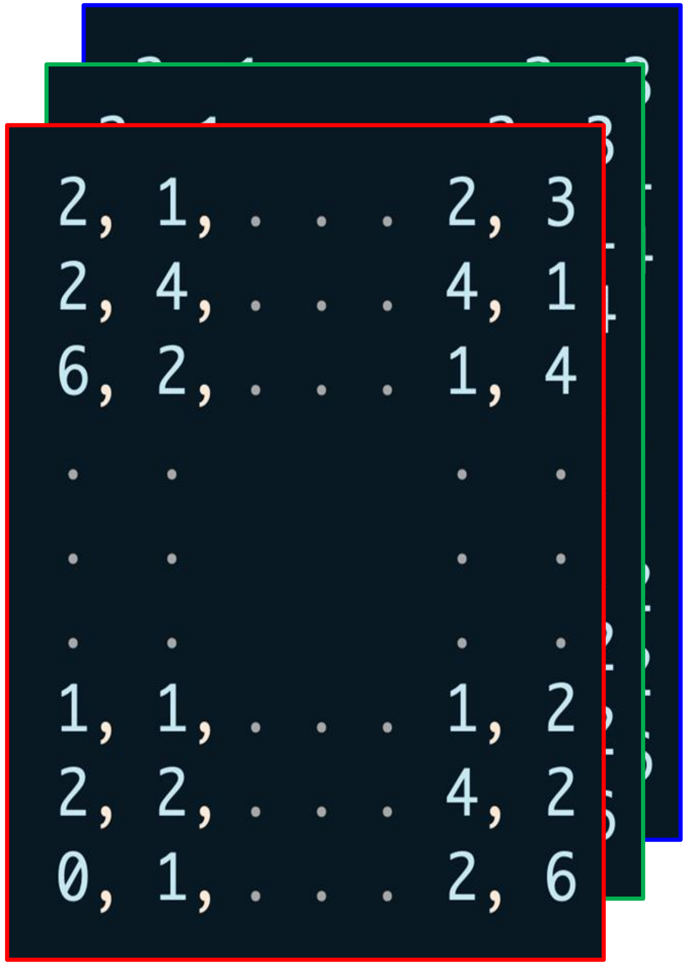


Image Classification with Deep Neural Networks

Digital Representation of Images

=



The Convolution Operator

$$\begin{bmatrix} 2 & 1 & 6 \\ 2 & 4 & 0 \\ 6 & 2 & 5 \end{bmatrix} \odot \underbrace{\begin{bmatrix} -2 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}}_{\text{Kernel}} = \begin{bmatrix} -4 & 0 & 6 \\ 2 & 4 & 0 \\ 0 & 0 & -5 \end{bmatrix} \xrightarrow{\Sigma} \boxed{3}$$

$$\begin{bmatrix} 2 & 1 & 6 & 3 & 4 & 1 & 2 \\ 2 & 4 & 0 & 1 & 1 & 1 & 4 \\ 6 & 2 & 5 & 3 & 2 & 5 & 1 \\ 1 & 1 & 5 & 7 & 3 & 3 & 1 \\ 2 & 2 & 4 & 3 & 1 & 2 & 4 \\ 0 & 1 & 1 & 3 & 4 & 1 & 2 \\ 3 & 2 & 3 & 2 & 6 & 4 & 4 \end{bmatrix}$$

Input

$$\boxed{3}$$

Output

The Convolution Operator

$$\begin{bmatrix}1 & 6 & 3\\ 4 & 0 & 1\\ 2 & 5 & 3\end{bmatrix} \odot \underbrace{\begin{bmatrix}-2 & 0 & 1\\ 1 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}}_{\text{Kernel}} = \begin{bmatrix}-2 & 0 & 3\\ 4 & 0 & 0\\ 0 & 0 & -3\end{bmatrix} \xrightarrow{\Sigma} \boxed{2}$$

$$\begin{bmatrix}2 & 1 & 6 & 3 & 4 & 1 & 2\\ 2 & 4 & 0 & 1 & 1 & 1 & 4\\ 6 & 2 & 5 & 3 & 2 & 5 & 1\\ 1 & 1 & 5 & 7 & 3 & 3 & 1\\ 2 & 2 & 4 & 3 & 1 & 2 & 4\\ 0 & 1 & 1 & 3 & 4 & 1 & 2\\ 3 & 2 & 3 & 2 & 6 & 4 & 4\end{bmatrix}$$

Input

$$3,\ \boxed{2}$$

Output

2019 embedded VISION SUMMIT

The Convolution Operator

$$\begin{bmatrix}4 & 1 & 2\\ 1 & 1 & 4\\ 2 & 5 & 1\end{bmatrix} \odot \underbrace{\begin{bmatrix}-2 & 0 & 1\\ 1 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}}_{\text{Kernel}} = \begin{bmatrix}-8 & 0 & 2\\ 1 & 1 & 0\\ 0 & 0 & -1\end{bmatrix} \xrightarrow{\Sigma} \boxed{-5}$$

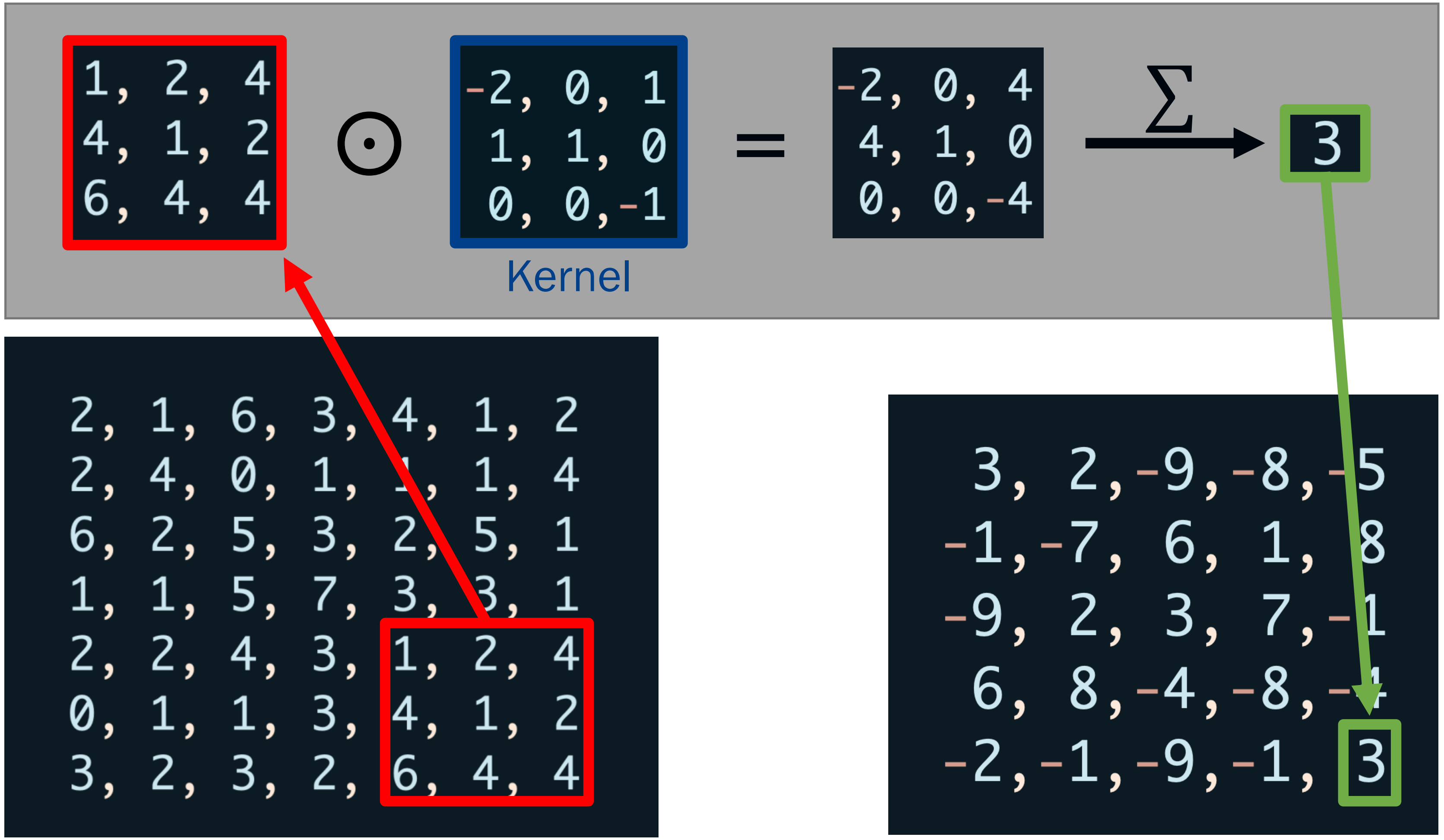
```
2, 1, 6, 3, 4, 1, 2
2, 4, 0, 1, 1, 1, 4
6, 2, 5, 3, 2, 5, 1
1, 1, 5, 7, 3, 3, 1
2, 2, 4, 3, 1, 2, 4
0, 1, 1, 3, 4, 1, 2
3, 2, 3, 2, 6, 4, 4
```

Input

```
3, 2,-9,-8,-5
```

Output

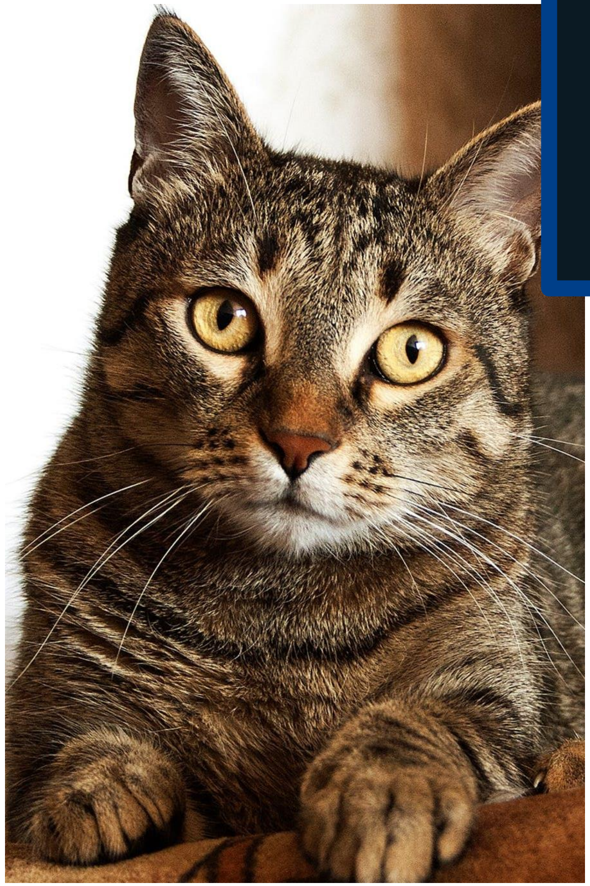
The Convolution Operator



Input

Output

The Convolution Operation



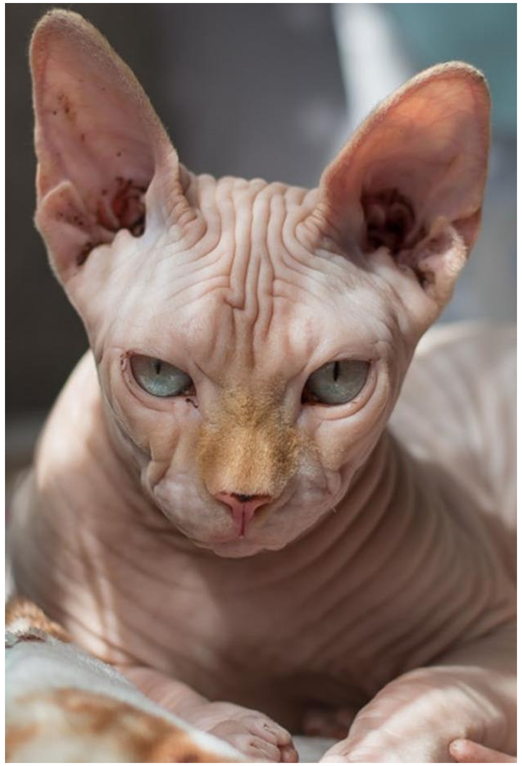
Input

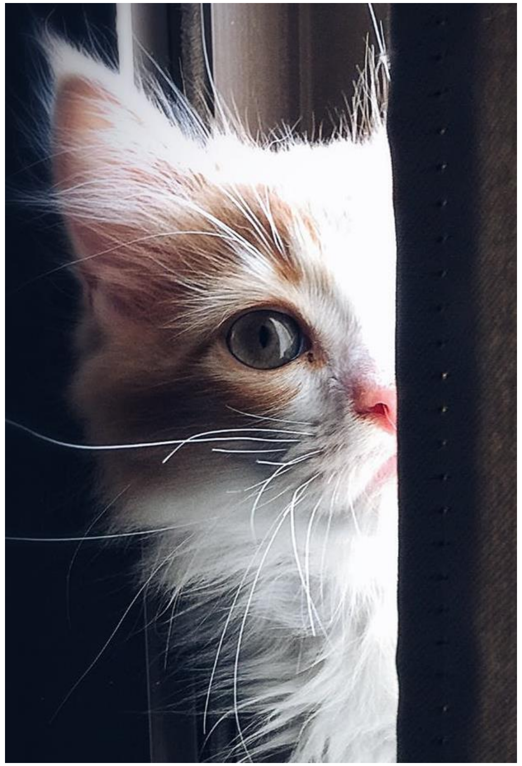
```
-1, 0, 1
-1, 0, 1
-1, 0, 1
```

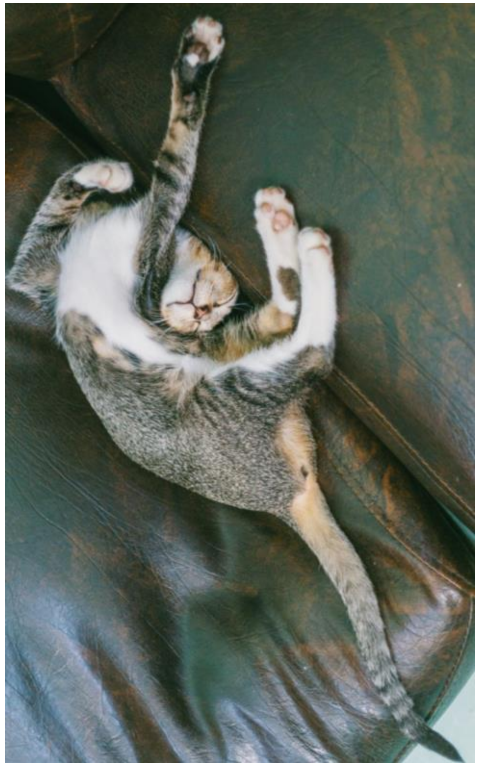
Kernel

Feature Map

The Convolution Operation

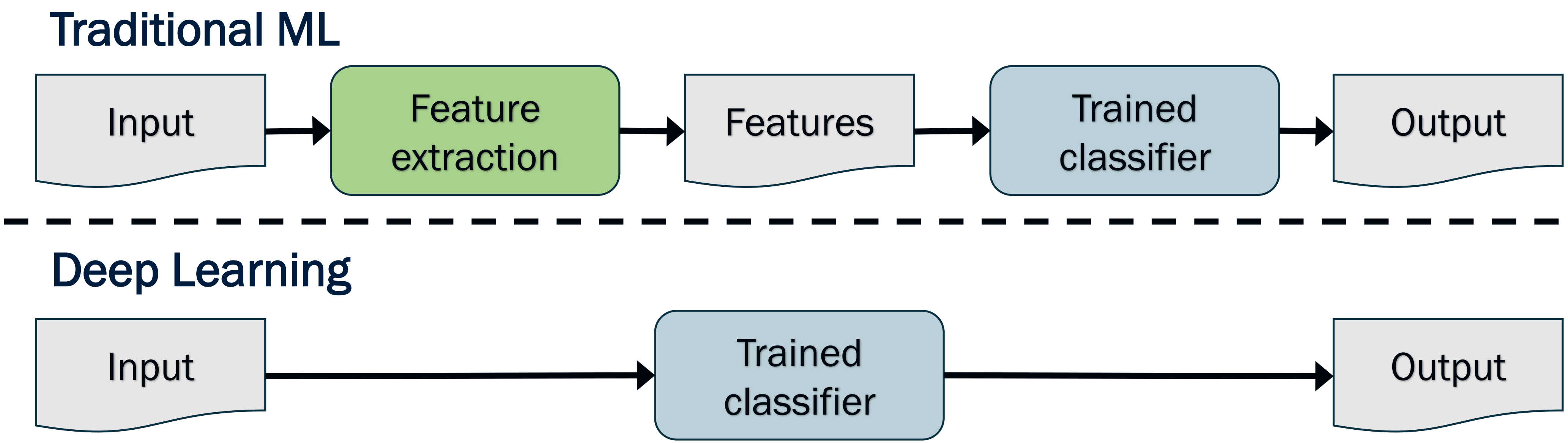






The Convolution Operation

- Takes spatial dependencies into account
- Used as a **feature extraction** tool
- **Differentiable** operation ➔ the kernels can be **learned**



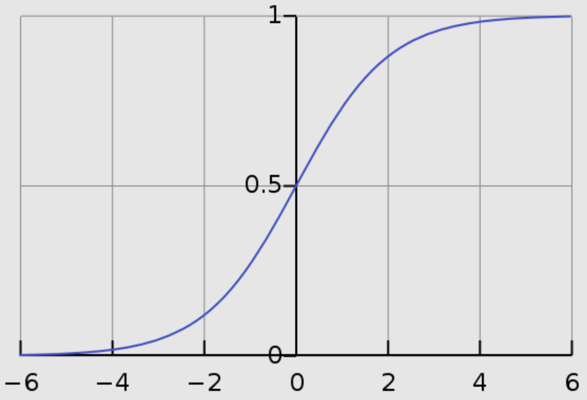
Non-linear Activation Functions

Increment the network's **capacity**

- Convolution, matrix multiplication and summation are linear

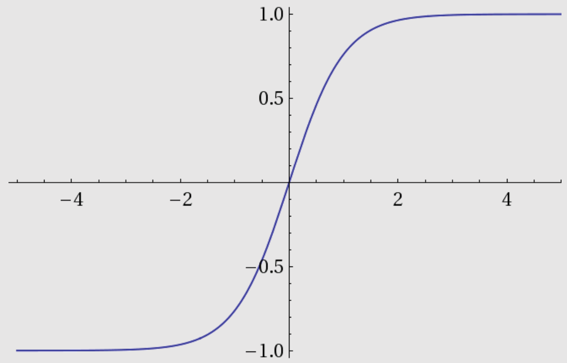
Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$



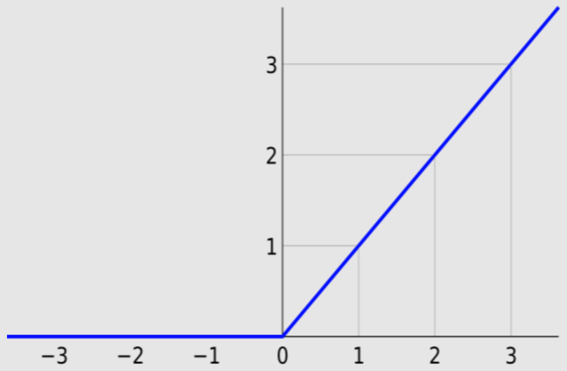
Hyperbolic tangent

$$tanh(x) = \frac{e^{2x-1}}{e^{2x+1}}$$

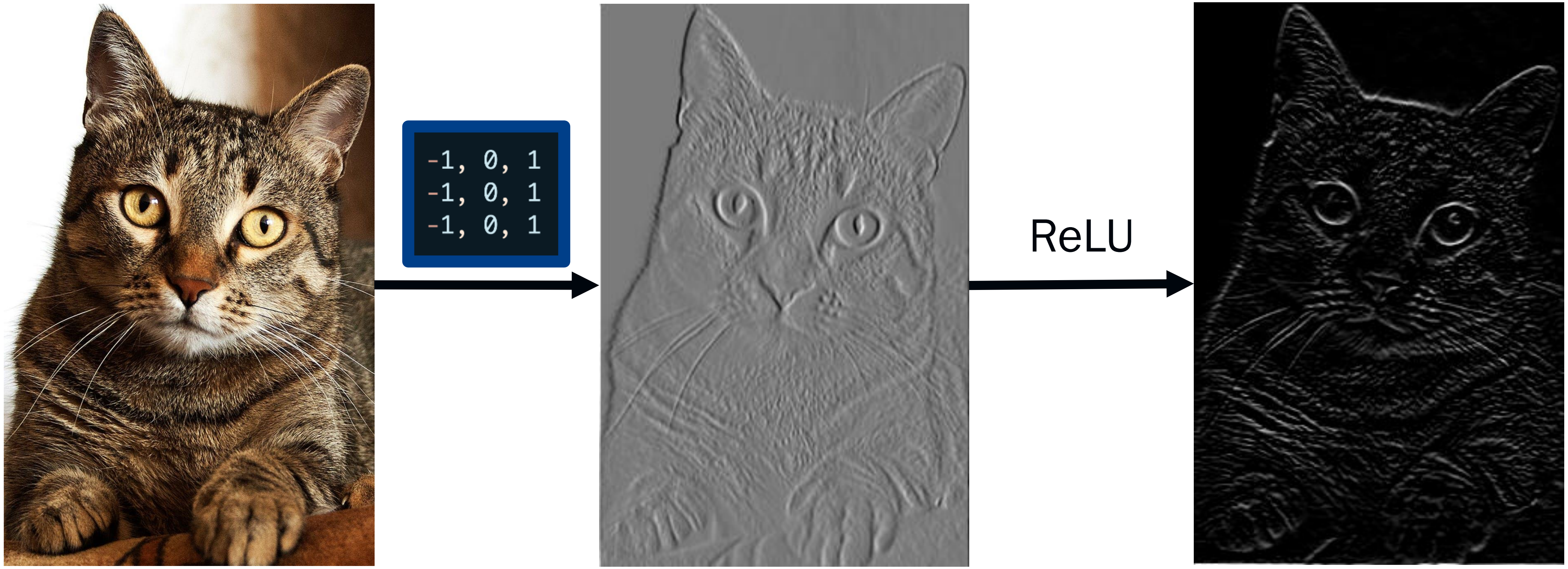


ReLU

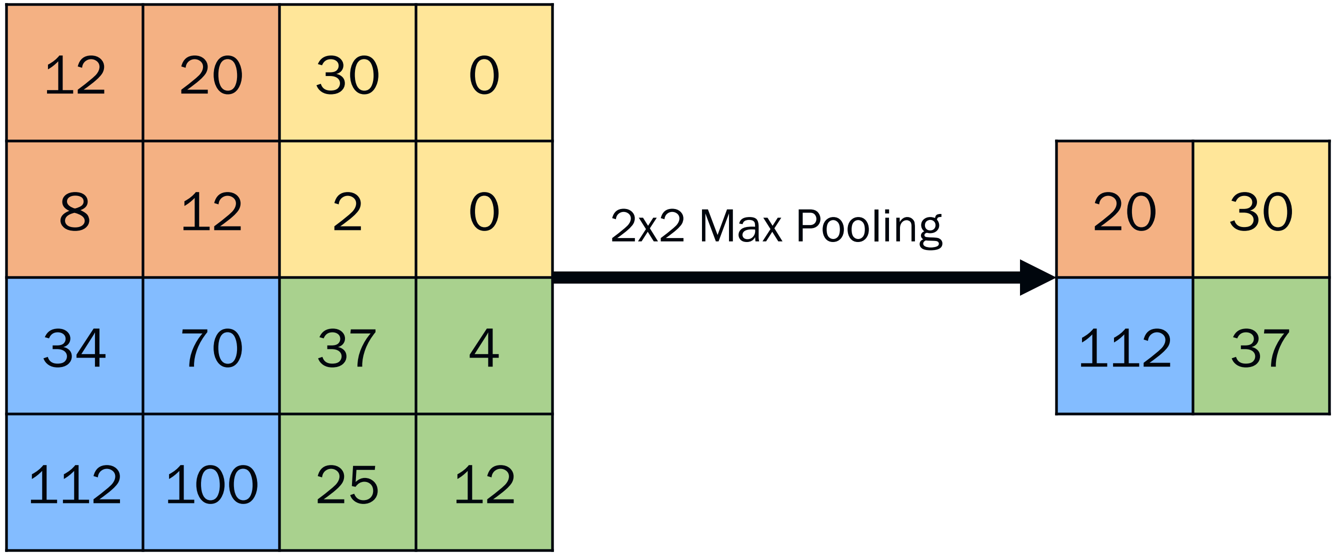
$$f(x) = \max(0, x)$$



Non-linear Activation Functions

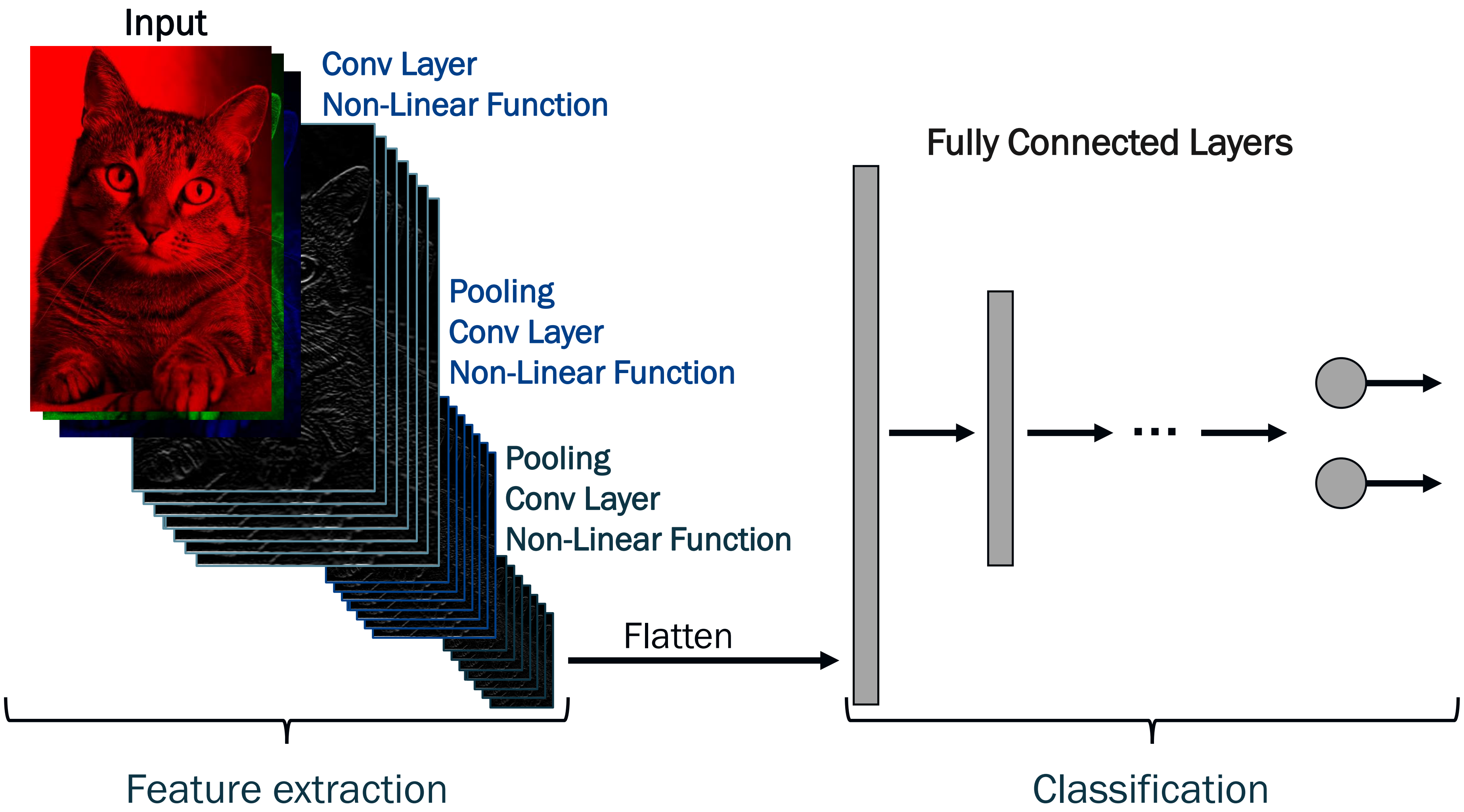


The Pooling Operation



- Used to reduce dimensionality
- Most common: **Max pooling**
- Makes the network invariant to small **transformations, distortions** and **translations.**

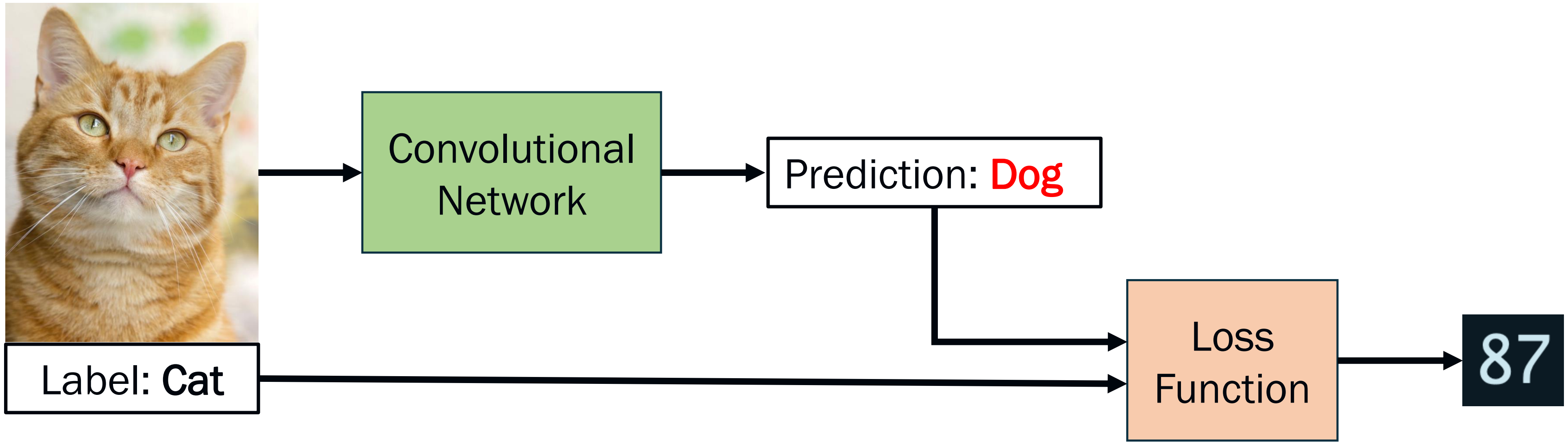
Putting all together



Training Convolutional Neural Networks

Image classification is a **supervised problem**

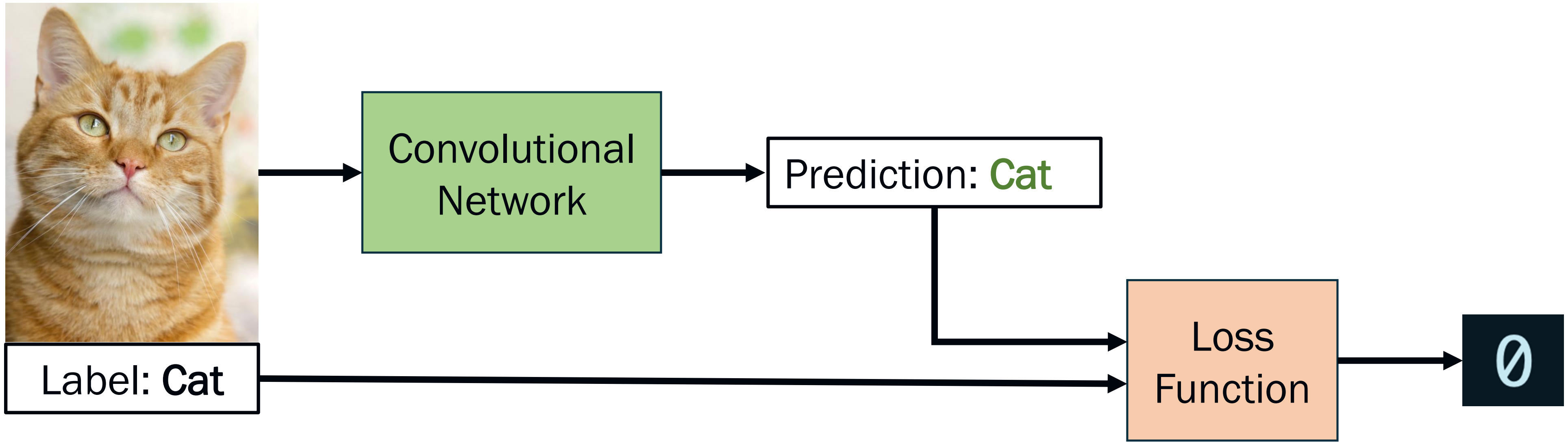
- Gather images and **label** them with desired output
- Train the network with **backpropagation**!



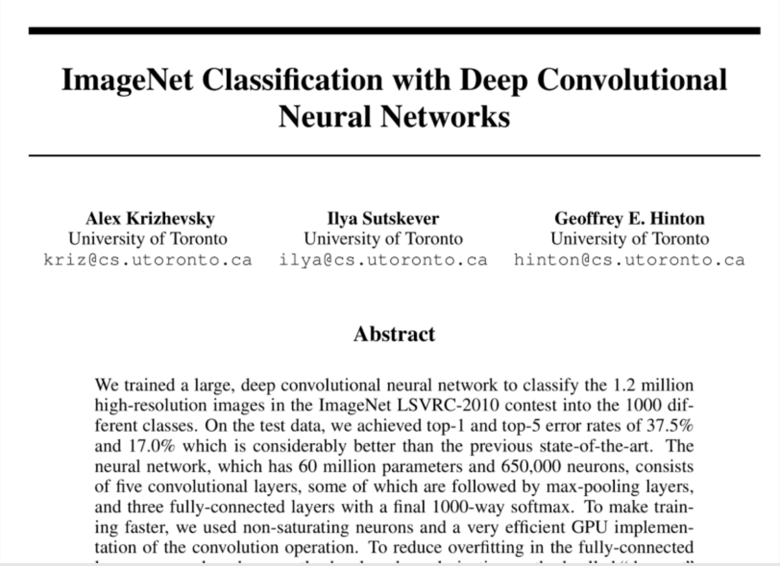
Training Convolutional Neural Networks

Image classification is a **supervised problem**

- Gather images and **label** them with desired output
- Train the network with **backpropagation**!



Surpassing Human Performance



ImageNet Classification with Deep Convolutional Neural Networks

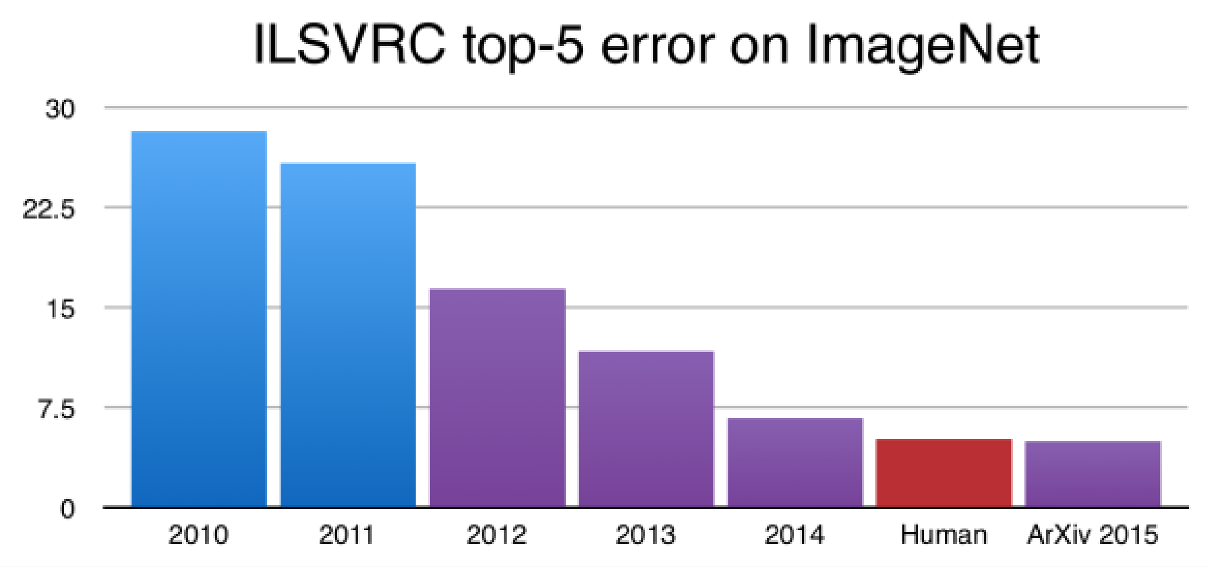
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

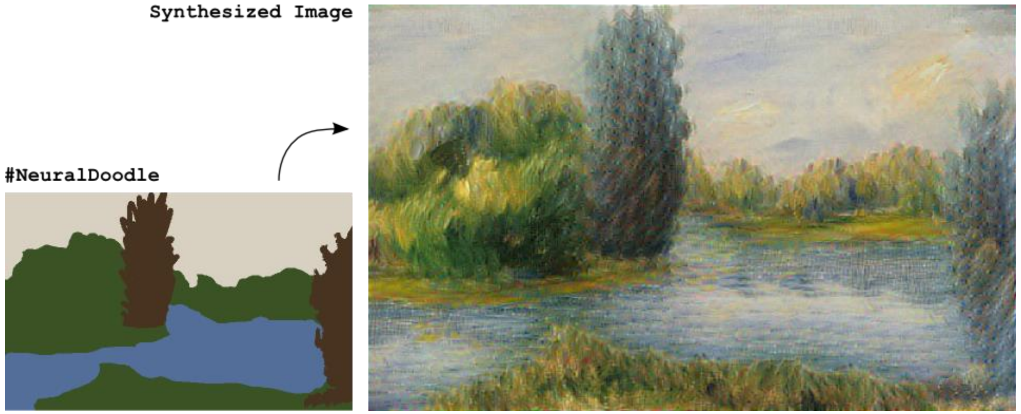
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected

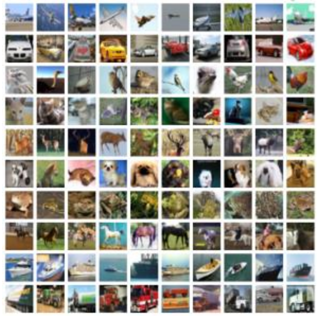


Deep Learning in the Wild

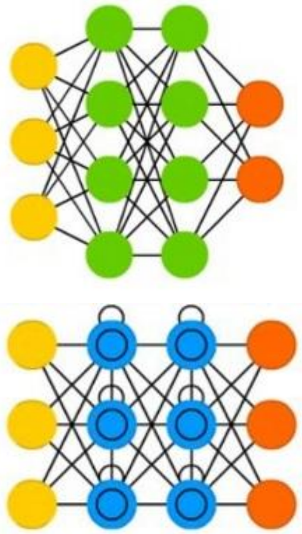


Deep Learning is Here to Stay

Data



Architectures



Frameworks

TensorFlow

K

torch

theano

Microsoft CNTK

Power

amazon web services

Google Cloud Platform

NVIDIA

Players

Microsoft

Google

Baidu 百度

DeepMind

Conclusions

Machine learning algorithms learn from data to find hidden relations, to make predictions, to interact with the world, ...

A machine learning algorithm is as good as its input data

- *Good model + Bad data = Bad Results*

Deep learning is making significant breakthroughs in: speech recognition, language processing, computer vision, control systems, ...

If you are not using or considering using Deep Learning to understand or solve vision problems, **you almost certainly should be**

Resource

Our work

Tryolabs Blog
https://www.tryolabs.com/blog

Luminoth (Computer Vision Toolkit)
https://www.luminoth.ai

To Learn More...

Google Machine Learning Crash Course
https://developers.google.com/machine-learning/crash-course/

Stanford course **CS229**: Machine Learning
https://developers.google.com/machine-learning/crash-course/

Stanford course **CS231n**: Convolutional Neural Networks for Visual Recognition
http://cs231n.stanford.edu/

2019 embedded VISION SUMMIT

Thank you!

tryo·labs