CENG 506 Deep Learning

Lecture 1 - Introduction

Objectives

Gain an understanding of how convolutional neural networks operate

What type of neural networks are implemented for different learning tasks (especially computer vision tasks)

Learn to implement and train your own neural networks

Course Outline

Introduction

Image Classification Problem

Backpropagation

Neural Network Training

Convolutional Neural Networks

ConvNets for Spatial Localization and Object Detection

Understanding and Visualizing Convolutional Neural Networks

Software Packages and Frameworks for Deep Learning

Recurrent Neural Networks

Adversarial Networks, DeepReinforcementLearning

Resources

From Stanford

http://cs231n.stanford.edu/

http://deeplearning.stanford.edu/tutorial/

…

Among many other resources

http://neuralnetworksanddeeplearning.com/
by Michael Nielsen

http://www.deeplearningbook.org/
by Ian Goodfellow and Yoshua Bengio and Aaron Courville

Prerequisites

<u>High-level familiarity in Python</u>

For assignments. We may provide a tutorial lecture. Those who have a lot of programming experience in a different language (e.g. C/C++/Matlab/Java) will probably be fine with some extra work.

<u>College Calculus, some Linear Algebra</u>

You should be comfortable taking derivatives and understanding matrix vector operations and notation.

<u>Basic Probability and Statistics</u>

You should know basics of probabilities, gaussian distributions, mean, standard deviation, etc.

<u>Machine Learning</u> (Equivalence of CENG463, or Stanford's CS229)

We will be formulating cost functions, taking derivatives and performing optimization with gradient descent.

Grading

Midterm: ~35%

Final: ~45%

Homeworks: ~20%

- Group work is not allowed!

Other rules

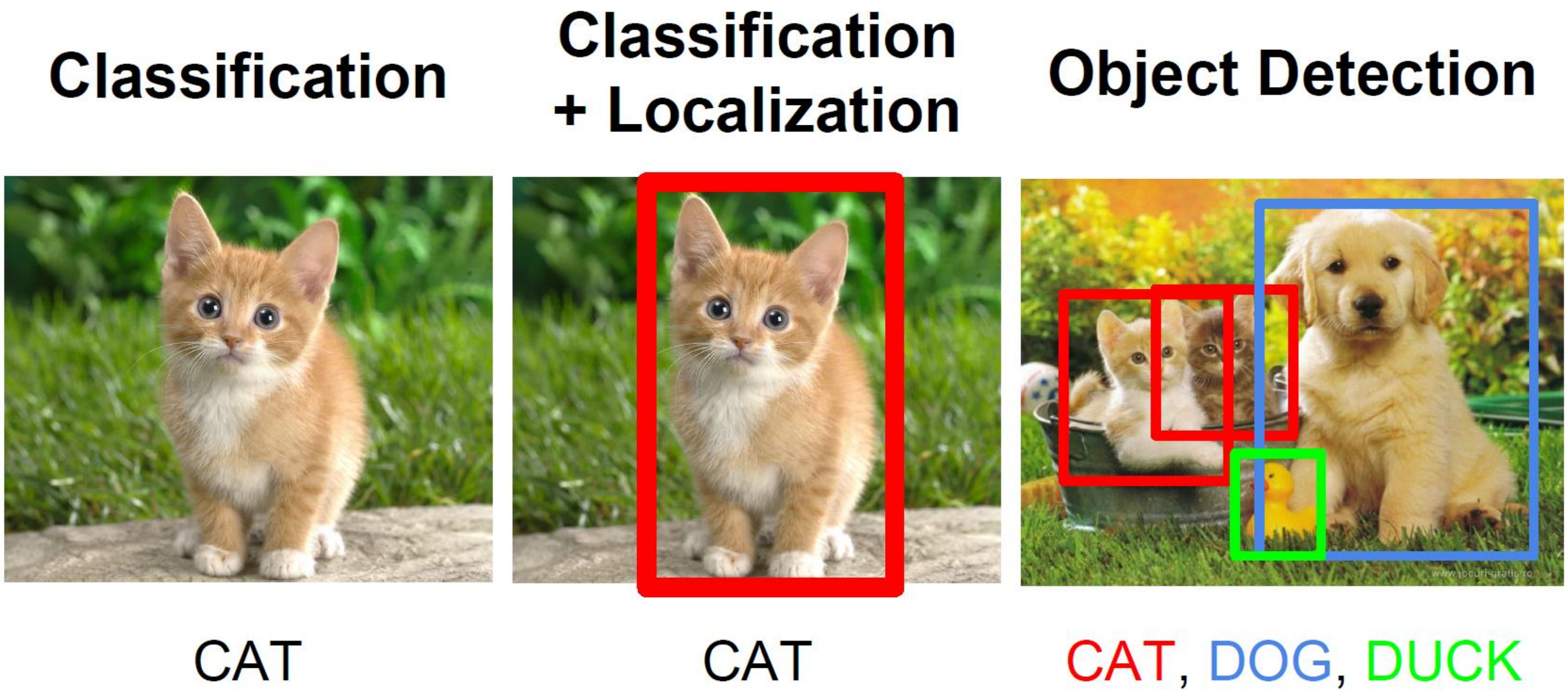
Every Thursday 10:00 A.M. A lecture of 2+ hours

All kinds of cheating, copying and plagiarism will result in penalty.

Slides, assignments and grades will be posted on cms.iyte.edu.tr (Course Management System). Please enroll yourselves.

Why Convolutional Neural Networks (CNNs) are Important?
The Case of Computer Vision

Some Computer Vision Tasks



Single object

Multiple objects

Image Classification

IMAGENET is dataset of 14M images from 22K categories.

Using a part of this dataset (1.4 M images from 1000 classes) a competition is organized. ImageNet Large-Scale Visual Recognition Challenge (ILSVRC).

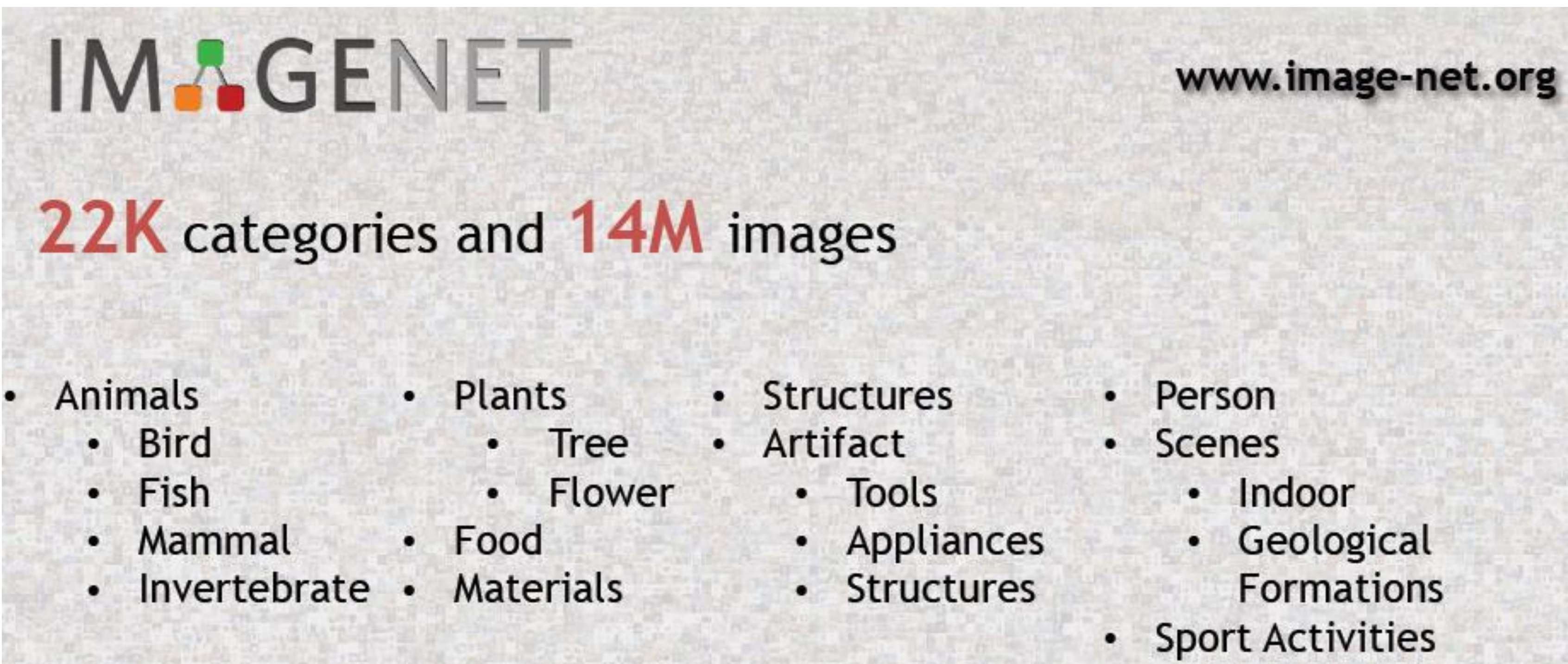


Image Classification

In ILSVRC, algorithms produce 5 top guesses for each image.
If actual image category is one of these 5 labels than guess is ‘correct’.

A test image

Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Accuracy: 1

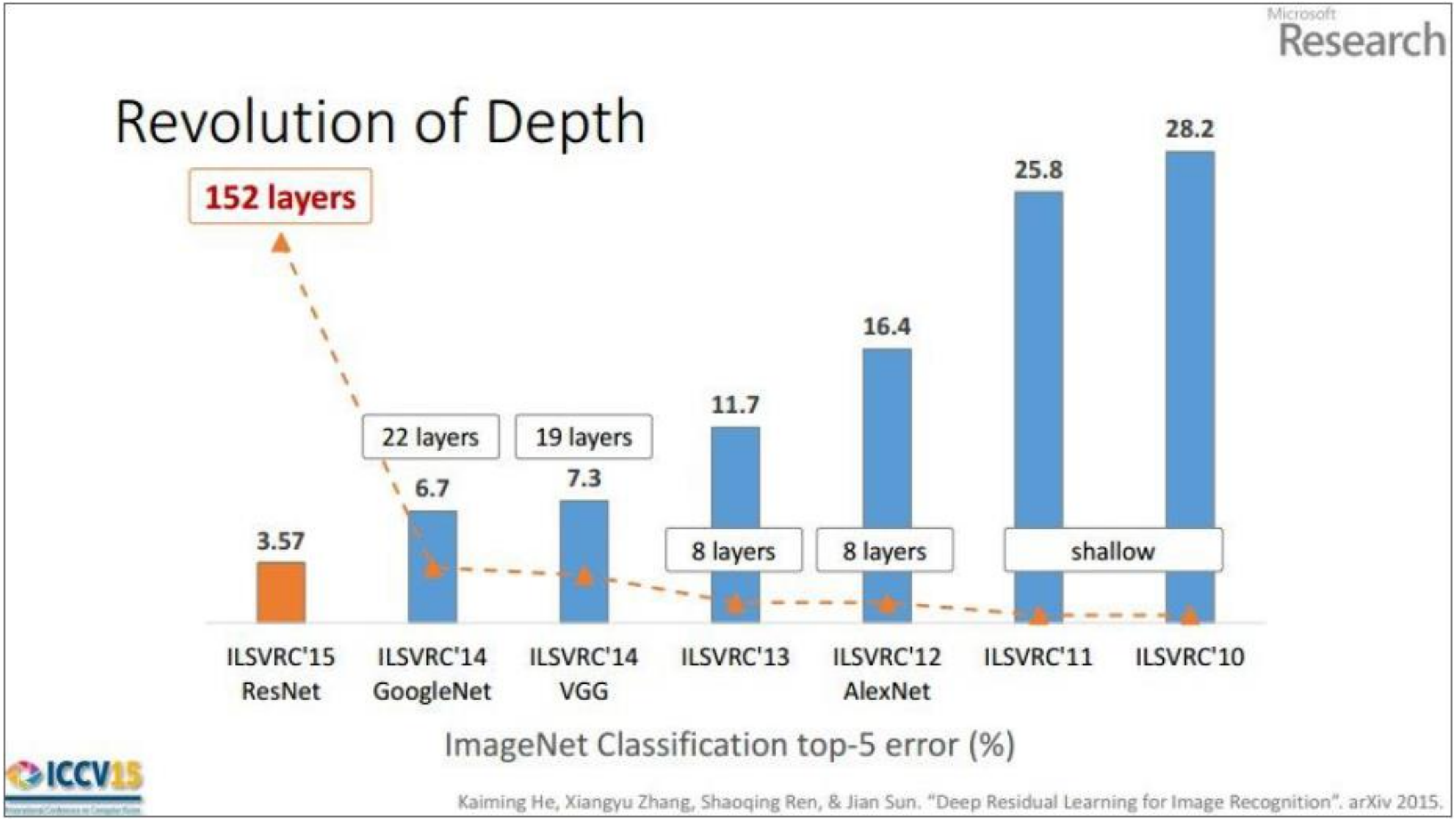
Scale
T-shirt
Giant panda
Drumstick
Mud turtle

Accuracy: 0

Images from Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”, IJCV 2015

Image Classification

ILSVRC image classification task top-5 error chart



(slide credit Kaiming He)

Other Example Tasks

CNNs became an important tool for tasks like image captioning and segmentation.

Figure from He et al, Mask R-CNN

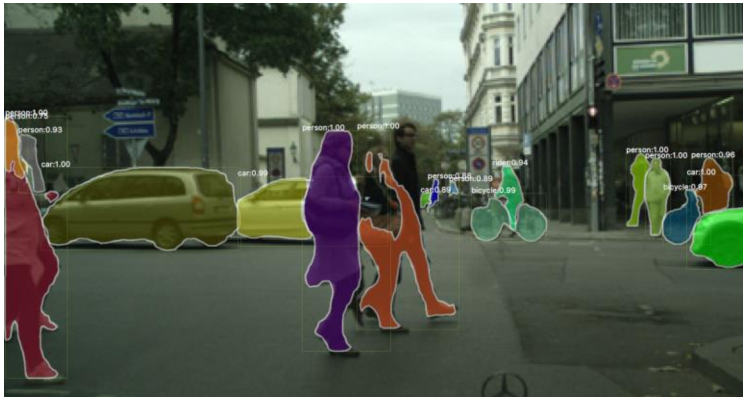


Figure by Andrej Karpathy, Fei-Fei Li

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

A Bit of History

CNNs were not invented overnight

The Fluctuating History of ANNs

The birth of neural nets with the Perceptron in 1957

The AI Winter of the 70s

Returning to popularity with backpropagation in 1986

Neural nets again losing favor in late 90s
(sometimes also called as 2nd AI Winter)

Revival of interest in neural nets in 2006

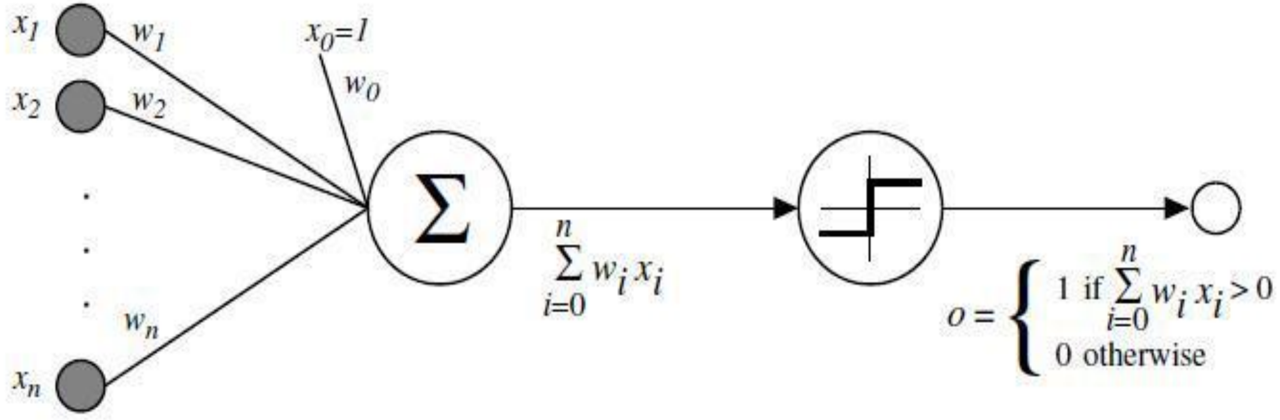
In 2012, Krizhevsky's CNN wins that year's ImageNet competition
(basically, the annual Olympics of computer vision).

Based on the post by Andrey Kurenkov:
http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning

Perceptrons - Structure

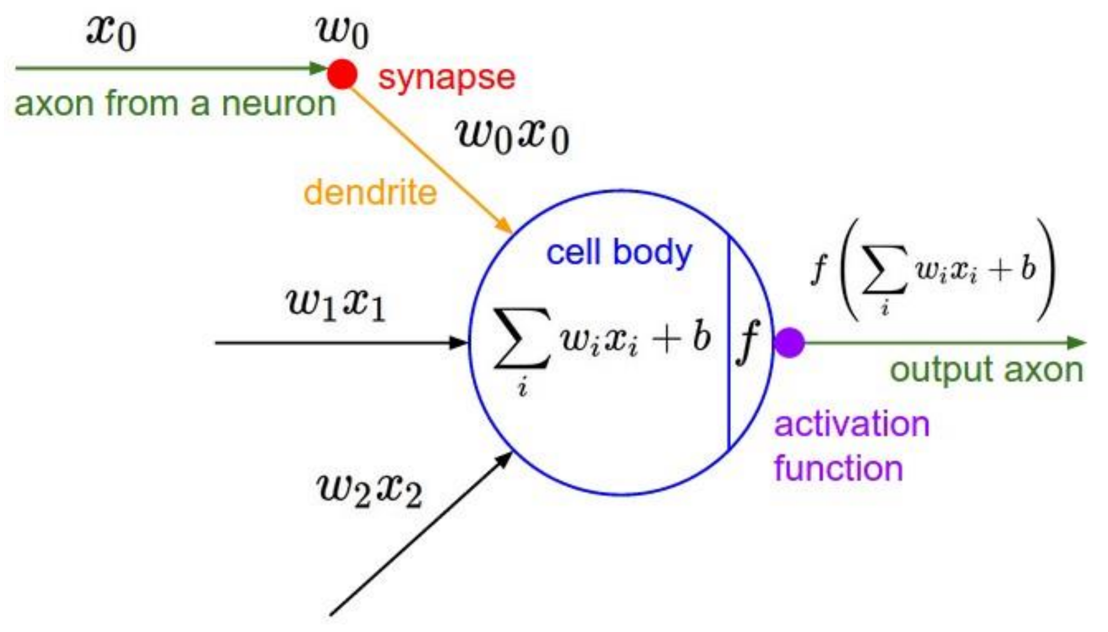
1957: Frank Rosenblatt's Perceptron is a simplified mathematical model of how the neurons in our brains operate:

- Takes a set of binary inputs (nearby neurons)
- Multiplies each input by a continuous valued weight (the synapse strength to each nearby neuron)
- Thresholds the sum of these weighted inputs to output a 1 if the sum is big enough and otherwise a 0 (in the same way neurons either fire or do not).



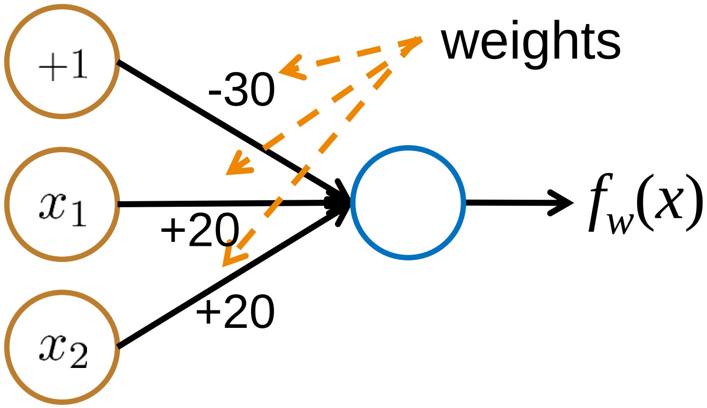
Perceptrons - Structure

Another diagram, showing the biological inspiration



This model was built on the work of McCulloch and Pitts (1943):
A neuron sums binary inputs and outputs a 1 if the sum exceeds a certain threshold value, and otherwise outputs a 0.

A Perceptron for logical AND operator



x_1	x_2	$f_w(x)$
0	0	0
0	1	0
1	0	0
1	1	1

$$f_w(x) = -30+20x_1+20x_2$$

$$f_w(x) = x_1 \text{ AND } x_2$$

But, how to train them? How they learn?

Rosenblatt came up with a way to make such artificial neurons learn.

Perceptrons - Training

Given a training set of input-output examples the Perceptron should 'learn' a function from, the algorithm is as follows:

1. Start off with a perceptron having random weights
2. For each example in training set, compute the Perceptron's output
3. If the output of the perceptron does not match the output:

   If the output should have been 0 but was 1, decrease the weights that had an input of 1.

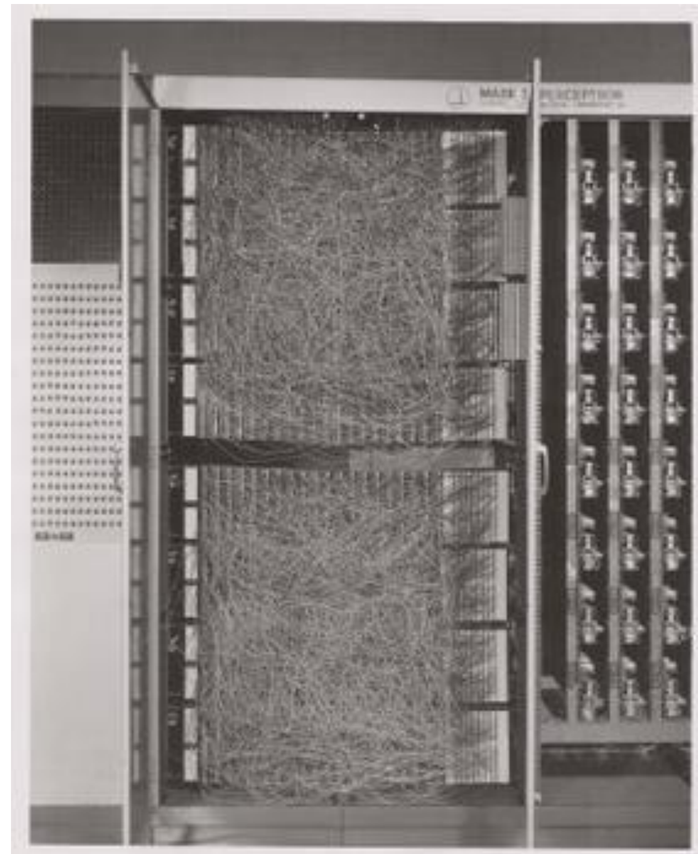
   If the output should have been 1 but was 0, increase the weights that had an input of 1.

Repeat steps 2-3 until the perceptron makes no more mistakes.

Perceptrons - Training

Rosenblatt implemented the idea of the Perceptron in custom hardware and showed it could be used to learn to classify simple shapes correctly with 20x20 pixel-like inputs.

In this case it learned a little toy function, but it was not difficult to envision more useful applications, right?



Mark I Perceptron at the Cornell Aeronautical Laboratory

False Promises

If we think about it, finding a set of weights for a number of inputs would not be enough to solve complex AI problems of Speech Recognition or Computer Vision.

However excitement was huge!

New York Times, July 8, 1958, Page 25

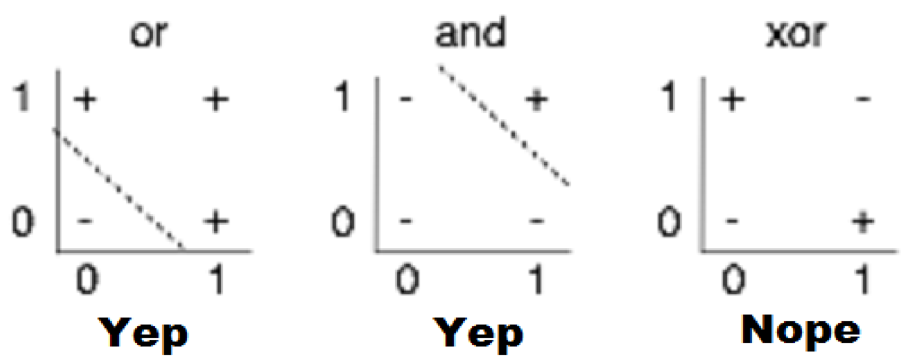
““The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence ... Dr. Frank Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers””

The First Winter

Marvin Minsky, founder of the MIT AI Lab, and Seymour Papert, director of the lab, were some of the researchers who were skeptical.

In 1969, they published their skepticism in the form of analysis on the limitations of Perceptrons in a seminal book aptly named Perceptrons.

In this book, this approach of AI was thought to have a dead-end. For instance, perceptrons could not learn the simple boolean function XOR because it is not linearly separable.

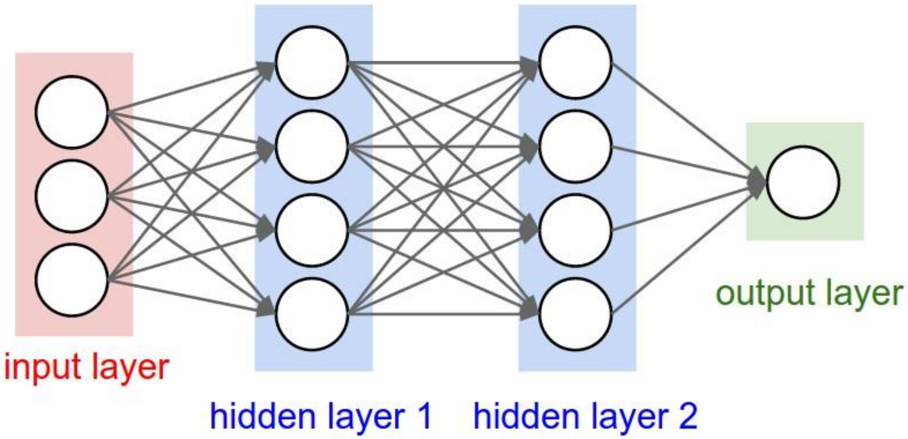


The First Winter

This publication is widely believed to cause the first AI Winter (freeze in funding and publications).

What was the problem?

A single layer of neurons is not enough to solve complicated problems. Rosenblatt's learning algorithm did not work for multiple layers.

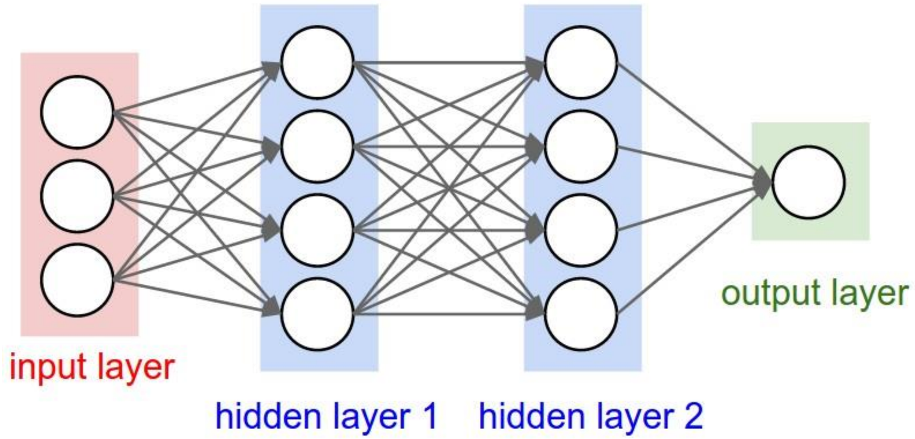


Backpropagation

We can use calculus to assign some of the blame for the mistakes in the output layer to each neuron in the previous hidden layer.

Then, we can further split up blame if there is another hidden layer, and so on - we backpropagate the error.

So, we can find how much the error changes if we change any weight in the neural net and use an optimization technique to find the optimal weights to minimize the error.



Backpropagation

Paul Werbos was first in the US to propose that backpropagation could be used for neural nets in his 1974 PhD Thesis.

Despite solving how multilayer neural nets could be trained, Werbos did not publish his findings until 1982 due to the AI Winter.

In 1986, this approach was re-popularized in “Learning representations by back-propagating errors” by David Rumelhart, Geoffrey Hinton, and Ronald Williams.

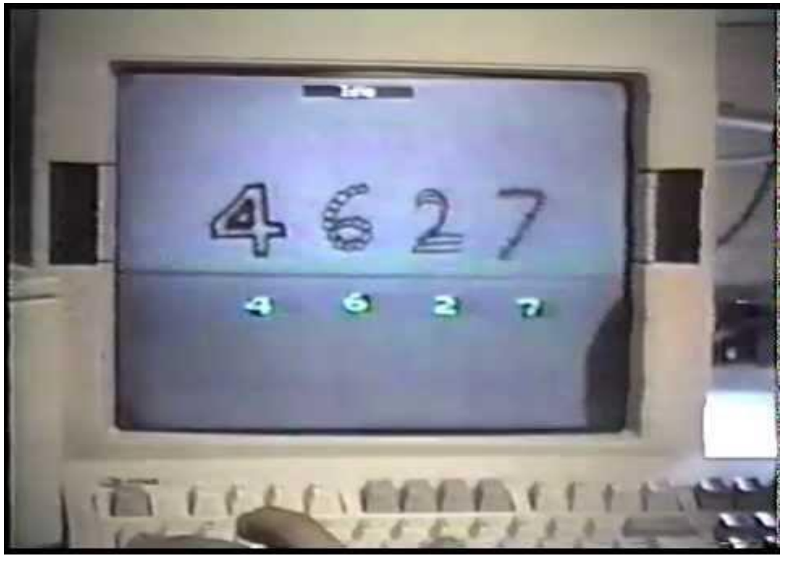
The idea was not new but it was this publication that made it widely understood how multilayer neural nets could be trained.

And… Return of the Neural Networks!

Convolutional Neural Networks

In 1989, Yann LeCun et al. at the AT&T Bell Labs demonstrated a very significant real-world application of backpropagation in ”Backpropagation Applied to Handwritten Zip Code Recognition”.

It was first to highlight key modifications of NNs toward modern deep learning: the first hidden layer of the neural net was convolutional.



https://www.youtube.com/watch?time_continue=2&v=FwFduRA_L6Q

The Second Winter

Backpropagation which was so recently a huge advance, soon became a problem.

Deep neural nets trained with backpropagation just did not work very well, and particularly did not work as well as nets with fewer layers.

The reason is that backpropagation relies on finding the error at the output layer and successively splitting it for prior layers.

It turns out that backpropagated error signals either shrink rapidly, or grow out of bounds, called ‘vanishing or exploding gradient problem’.

The Second Winter

Neural nets were unreliable and did not work very well. The computers were not fast enough, the algorithms were not smart enough, and people were not happy.

A new method called Support Vector Machines, which in the simplest terms could be described as a mathematically optimal way of training an equivalent to a two layer neural net, started to be seen superior to neural nets.

Around the mid 90s, a new AI Winter for neural nets began to emerge - the community once again lost faith in them.

The Second Comeback

Funding from the Canadian Institute for Advanced Research (CIFAR), which encourages basic research without direct application, motivated Geoffrey Hinton to move to Canada in 1987.

The funding was modest, but sufficient to enable a small group of researchers to keep working on the topic.

As Hinton tells it, they “rebranded” the frowned-upon field of neural nets with a new nickname: “Deep Learning”

In 2006, they published a paper that was significant enough to rekindle interest in neural nets: A fast learning algorithm for deep belief nets.

The Second Comeback

They argued that neural networks with many layers really could be trained well, if the weights are initialized in a clever way rather than randomly.

Hinton and two of his graduate students demonstrated their effectiveness at a challenging AI task: Speech Recognition.

They managed to improve on a decade-old performance record on a standard speech recognition dataset. This was an impressive achievement, but in retrospect it was only a hint at what was coming - many more broken records.

Analysis on Activation Functions

Xavier Glort and Yoshua Bengio published “Understanding the difficulty of training deep feedforward neural networks” in 2010, arguing the choice of non-linear activation function in a neural net makes a big impact on performance.

What, then, is the best activation function’?

Three different groups explored the question:

- A group with LeCun, “What is the best multi-stage architecture for object recognition?”
- A group with Hinton, “Rectified linear units improve restricted boltzmann machines”
- A group with Bengio, “Deep Sparse Rectifier Neural Networks”

They all found the same surprising answer: the very non-differentiable and very simple function $f(x)=max(0,x)$ tends to be the best.

Today

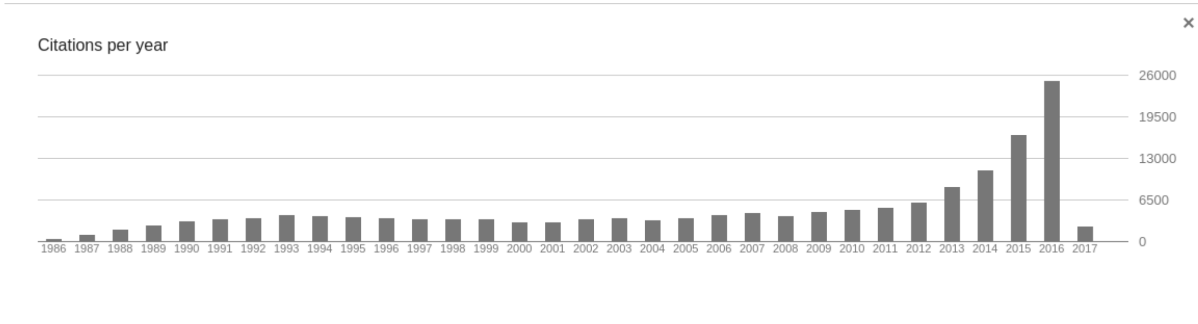
With all the discoveries since 2006, the reasons for the failure of the supervised learning in the past are summarized by Hinton as follows:

- Our labeled datasets were thousands of times too small.
- Our computers were millions of times too slow.
- We initialized the weights in a stupid way.
- We used the wrong type of nonlinearity.

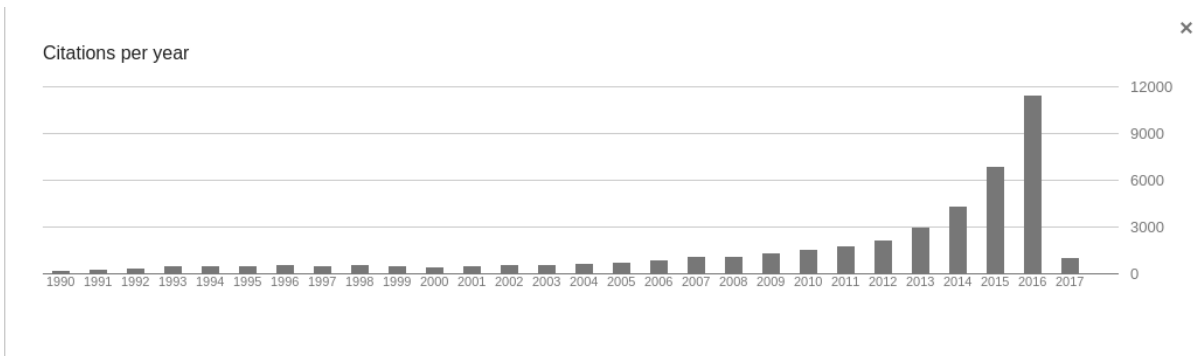
In 2012, a CNN (Hinton and students) was the winner of ILSVRC image classification task for the first time.

Today

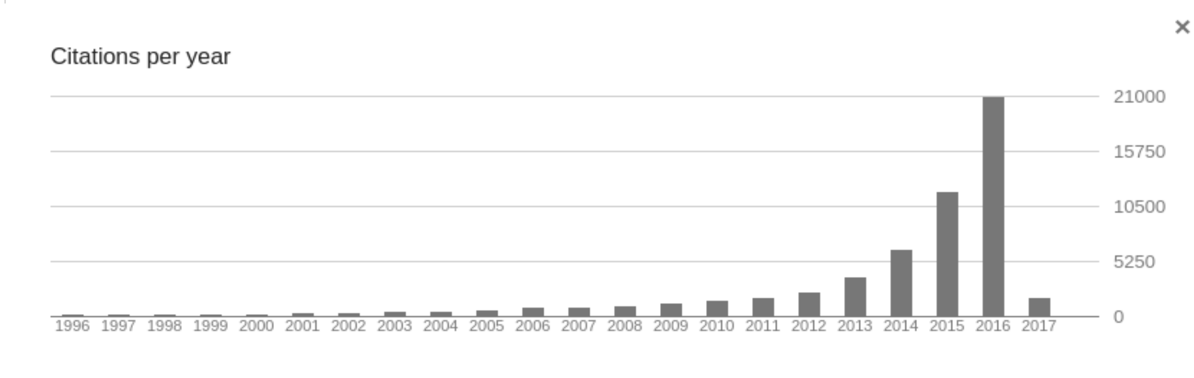
Hinton:



LeCun:



Bengio:



Today

There is a tsunami of Neural Networks.

> "Deep learning, in that vision, could transform almost any industry. "There are fundamental changes that will happen now that computer vision really works," says Jeff Dean, who leads the Google Brain project. Or, as he unsettlingly rephrases his own sentence, "now that computers have opened their eyes."

> "During a talk at WIRED 2016, Jürgen Schmidhuber presented the future of AI as something beyond just taking over jobs. "In 2050 there will be trillions of self-replicating robot factories on the asteroid belt," he told the audience. "A few million years later, AI will colonize the galaxy."

Today

But, by becoming proficient in a single task, it's very easy for a machine to seem intelligent.

> “It is impressive and surprising that these general-purpose, statistical models can learn meaningful relations from text alone, without any richer perception of the world, but this may speak much more about the unexpected ease of the task itself than it does about the capacity of the models. Just as checkers can be won through tree-search, so too can many semantic relations be learned from text statistics. Both produce impressive intelligent-seeming behaviour, but neither necessarily pave the way towards true machine intelligence. ……… If they have succeeded in anything superficially similar, it has been because they saw many hundreds of times more examples than any human ever needed to”.
>
> Luke Hewitt, a PhD student at the MIT Department of Brain and Cognitive Sciences
> http://thinkingmachines.mit.edu/blog/unreasonable-reputation-neural-networks

Today

So, is the winter coming?