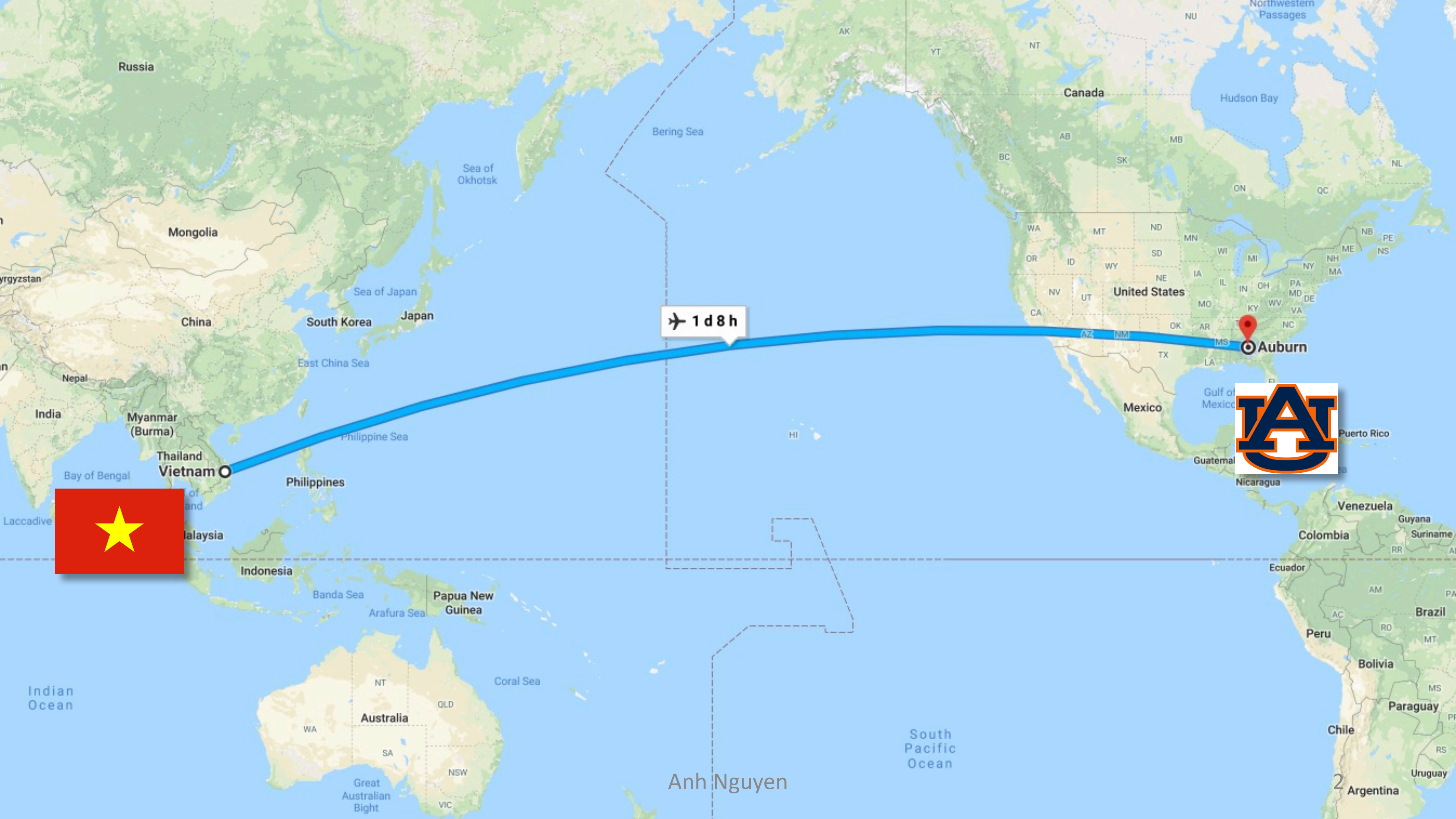
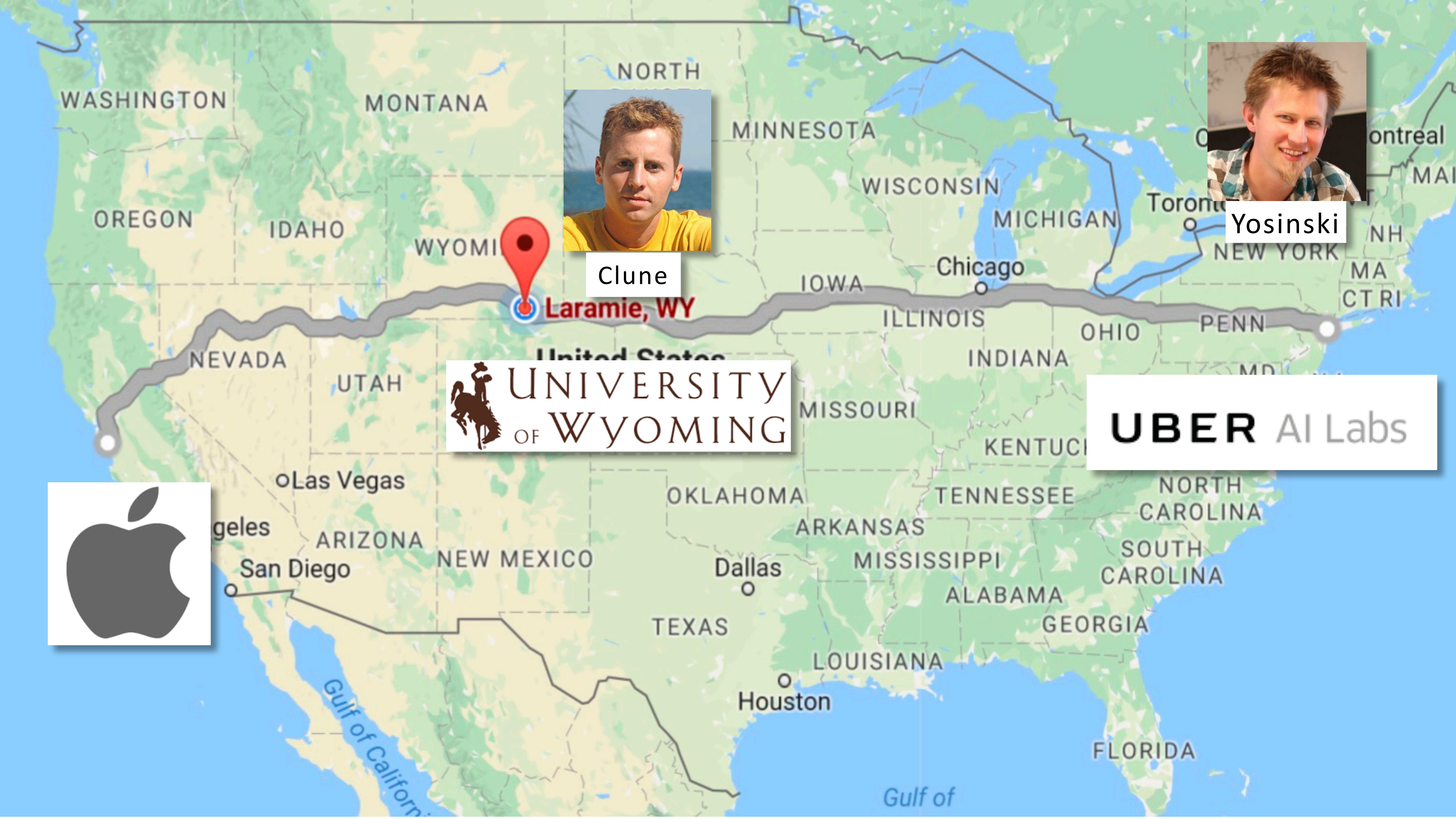
Lecture 1: Introduction

COMP 4970 / 7970

Deep Learning

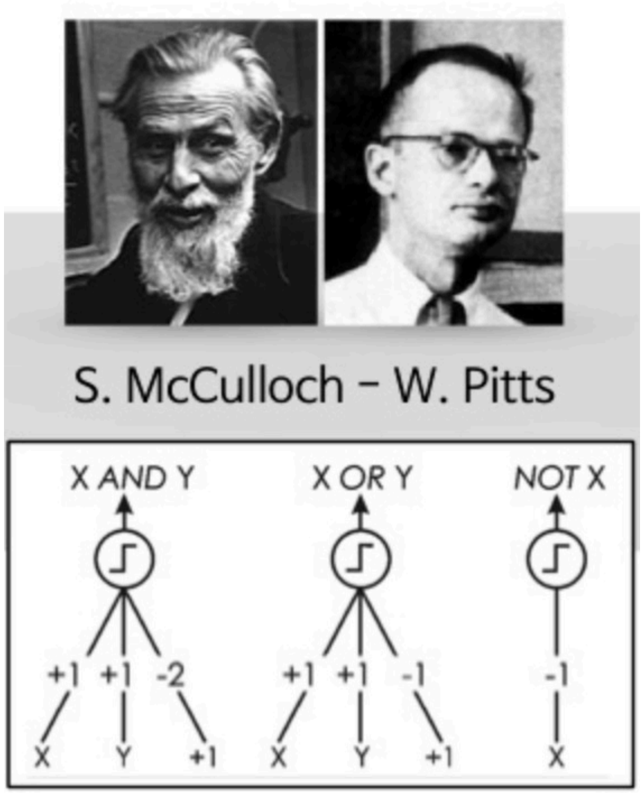

1 d 8 h
Vietnam
Auburn
Russia
Mongolia
China
South Korea
Japan
Sea of Japan
East China Sea
Sea of Okhotsk
Bering Sea
Nepal
India
Myanmar (Burma)
Thailand
Bay of Bengal
Laccadive
Malaysia
Philippines
Philippine Sea
Indonesia
Banda Sea
Arafura Sea
Papua New Guinea
Coral Sea
Australia
Great Australian Bight
Indian Ocean
South Pacific Ocean
Canada
Hudson Bay
Northwestern Passages
United States
Mexico
Gulf of Mexico
Guatemala
Nicaragua
Puerto Rico
Venezuela
Guyana
Suriname
Colombia
Ecuador
Peru
Brazil
Bolivia
Paraguay
Chile
Uruguay
Argentina


Clune
Yosinski
Laramie, WY
UNIVERSITY OF WYOMING
UBER AI Labs
WASHINGTON
MONTANA
NORTH
MINNESOTA
OREGON
IDAHO
WYOMI
WISCONSIN
MICHIGAN
Toront
ontreal
MAI
NH
NEW YORK
MA
CT RI
Chicago
IOWA
ILLINOIS
OHIO
PENN
NEVADA
UTAH
INDIANA
MD
MISSOURI
KENTUC
Las Vegas
geles
ARIZONA
NEW MEXICO
OKLAHOMA
TENNESSEE
NORTH CAROLINA
ARKANSAS
SOUTH CAROLINA
San Diego
Dallas
MISSISSIPPI
ALABAMA
GEORGIA
TEXAS
LOUISIANA
Houston
FLORIDA
Gulf of Californi
Gulf of

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

1943: **MCP Neuron**

History of Deep Learning



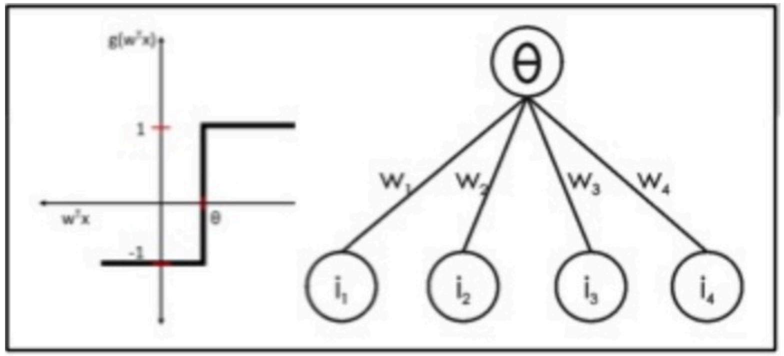
- Weights are not learned

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

1943: MCP Neuron

1958: **Perceptron**

F. Rosenblatt



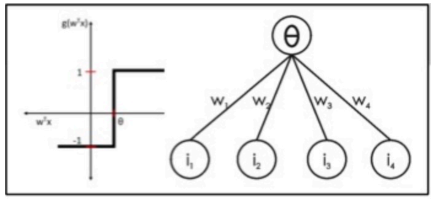
- Learnable Weights and Threshold

1940 1950 1960 1970 1980 1990 2000 2010 202

1943: MCP Neuron

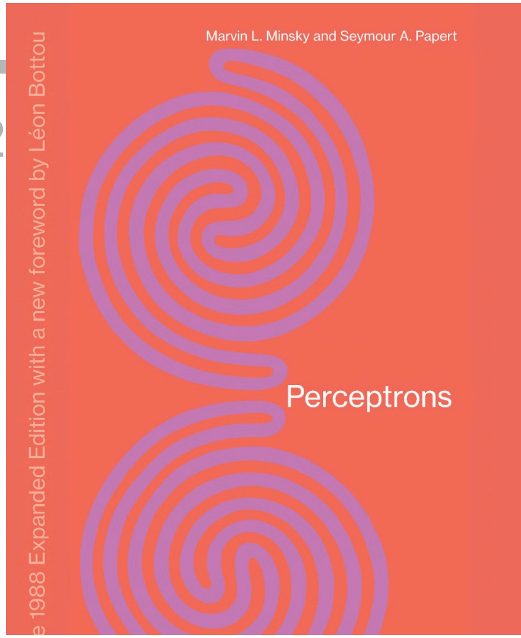
1969: **XOR problem**

1958: Perceptron



- Learnable Weights and Threshold

M. Minsky – S. Papert

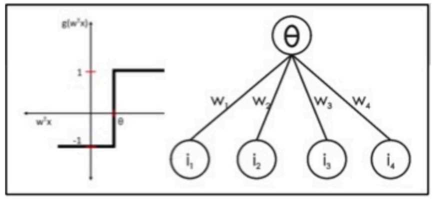


1940 1950 1960 1970 1980 1990 2000 2010 202

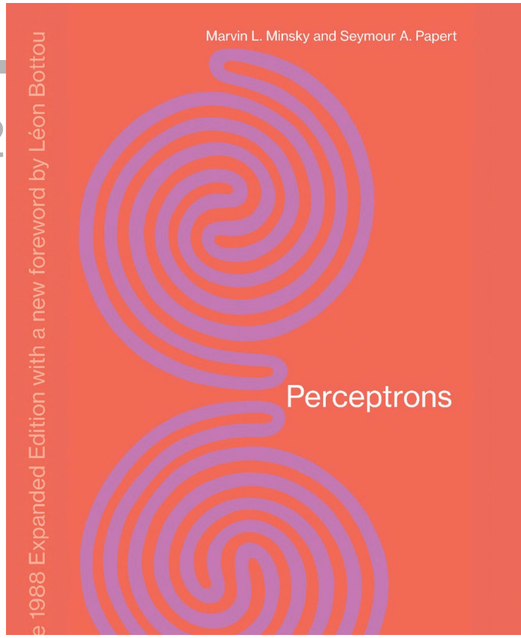
1943: MCP Neuron

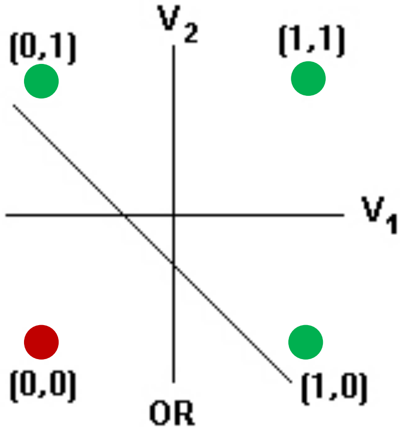
1969: **XOR problem**

1958: Perceptron



- Learnable Weights and Threshold



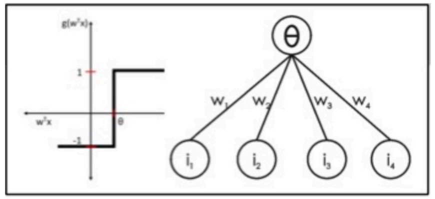


1940 1950 1960 1970 1980 1990 2000 2010 202

1943: MCP Neuron

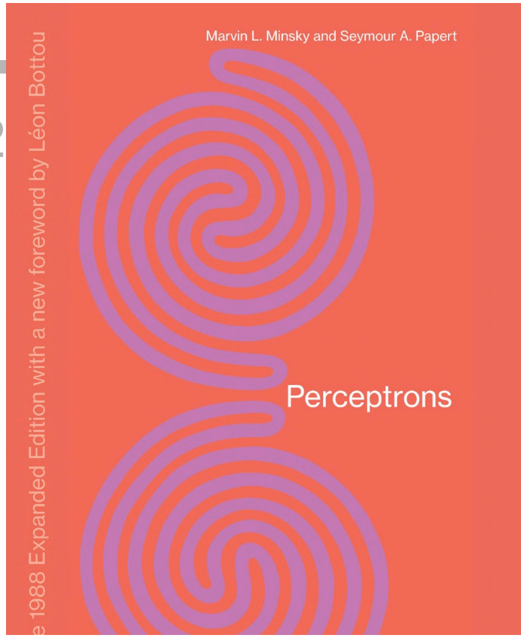
1969: **XOR problem**

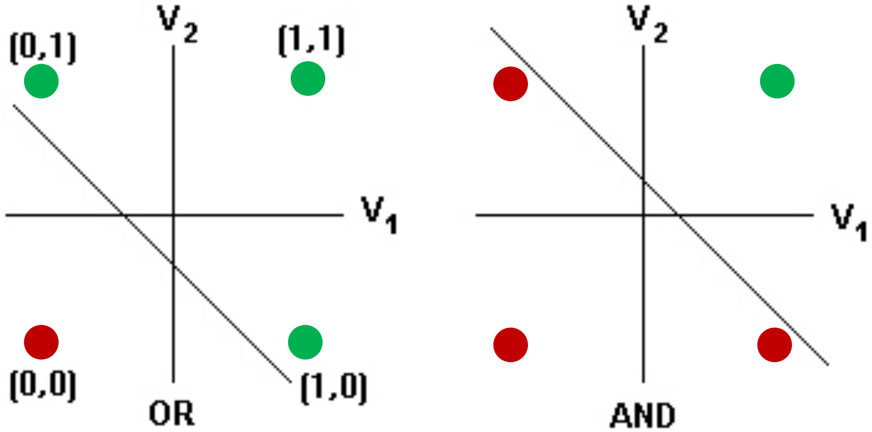
1958: Perceptron



• Learnable Weights and Threshold

M. Minsky – S. Papert

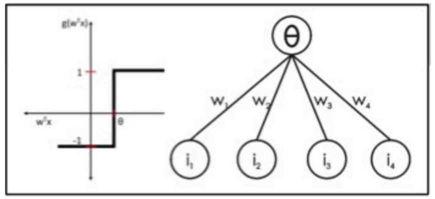




1940 1950 1960 1970 1980 1990 2000 2010 202

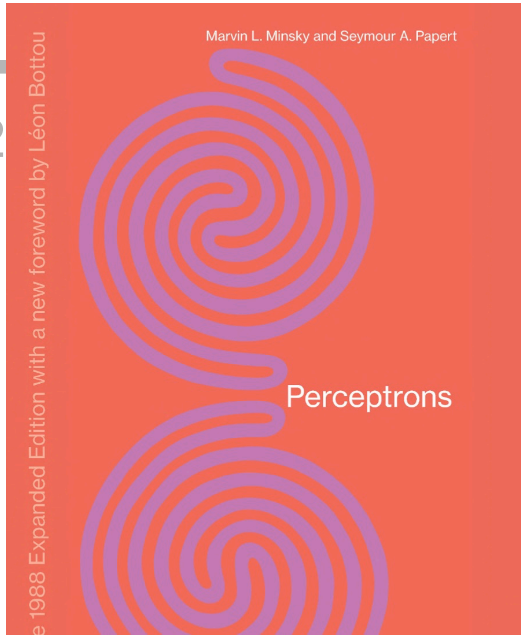
1943: MCP Neuron

1958: Perceptron

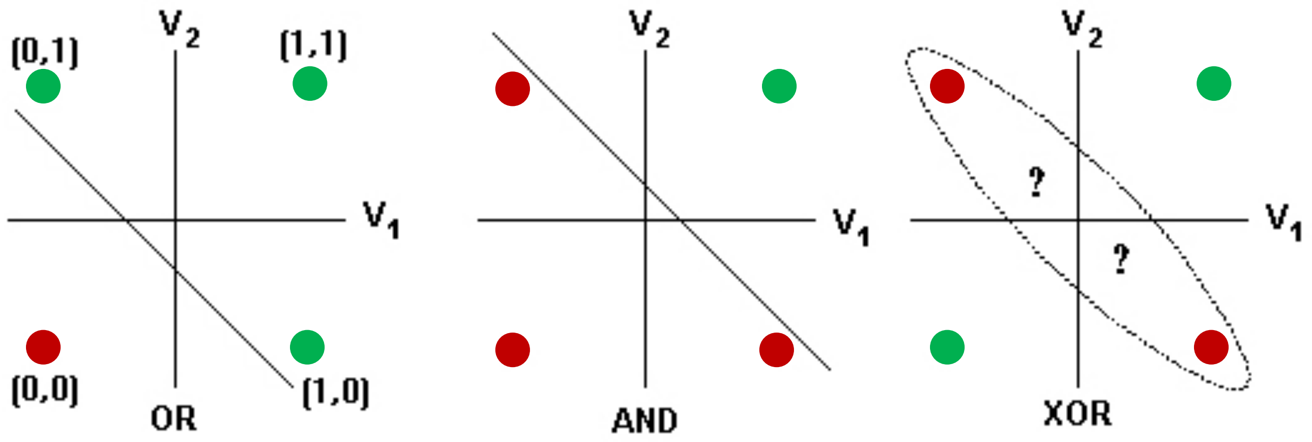


- Learnable Weights and Threshold

1969: **XOR problem**



- ~~Perceptron~~
- Need multi-layer perceptrons
- How to train MLPs?



Winter 1

1940 | 1950 | 1960 | 1970 | 1980 | 1990 | 2000 | 2010 | 2020 | ...

1943: MCP Neuron

1958: Perceptron

1969: **XOR problem**

- ~~Perceptron~~
- Need multi-layer perceptrons
- How to train MLPs?

- Severe funding cuts
- Reduced research interests

Winter 1

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

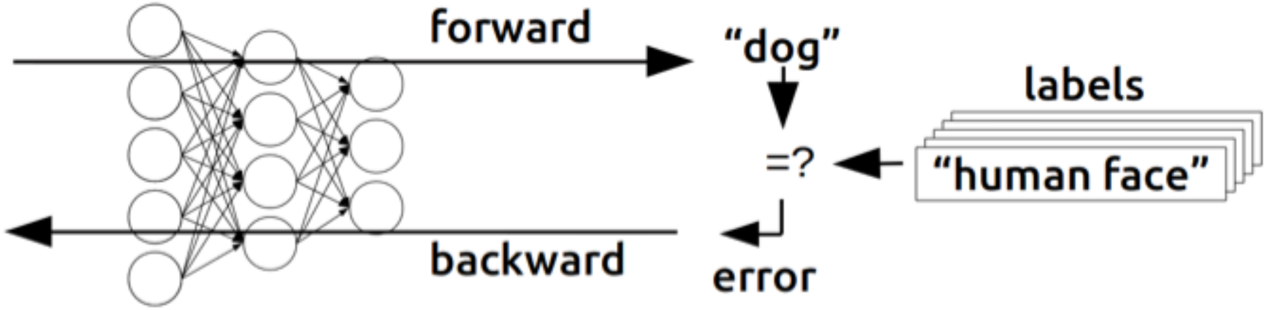
1943: MCP Neuron

1958: Perceptron

1969: XOR problem

1986: **Backpropagation**

- Key: Chain rule
- Solution to nonlinearly separable problems



Winter 1

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

1943: MCP Neuron

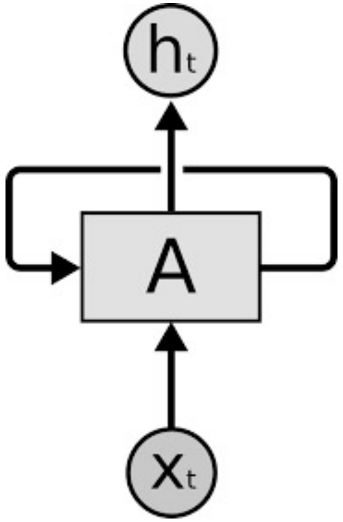
1969: XOR problem

1958: Perceptron

1986: Backpropagation

1982: **Hopfield network**

1986: **Jordan network**



Recurrent Neural Networks

Winter 1

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

1943: MCP Neuron

1969: XOR problem

1958: Perceptron

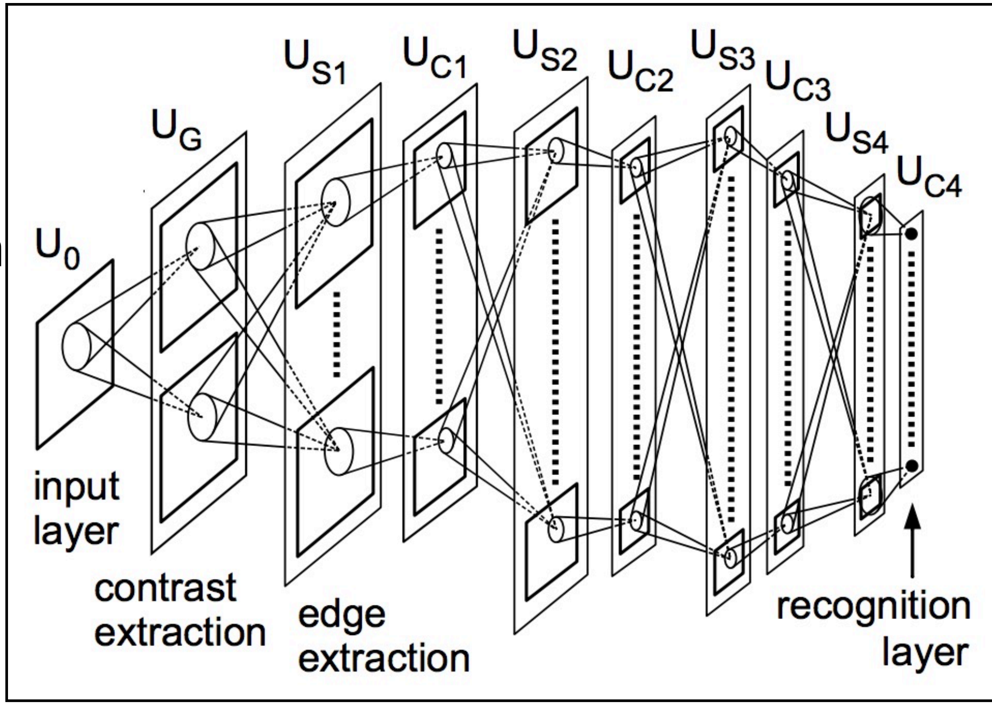
1986: Backpropagation

1982: Hopfield network

1986: Jordan network

1983: **Neocognitron**

K. Fukushima



- Biologically inspired
- Hierarchical, multi-layered
- Translation-invariant

Winter 1

1940 1950 1960 1970 1980 1990 2000 2010 2020 ...

1943: MCP Neuron

1969: XOR problem

1958: Perceptron

1986: Backpropagation

1982: Hopfield network

1986: Jordan network

1983: Neocognitron

Problems:

- Large computation
- Overfitting
- Vanishing gradients

Winter 1 Winter 2

1940 1950 1960 1970 1980 1990 2000 2010 2020 …

1943: MCP Neuron

1969: XOR problem

1958: Perceptron

1986: Backpropagation

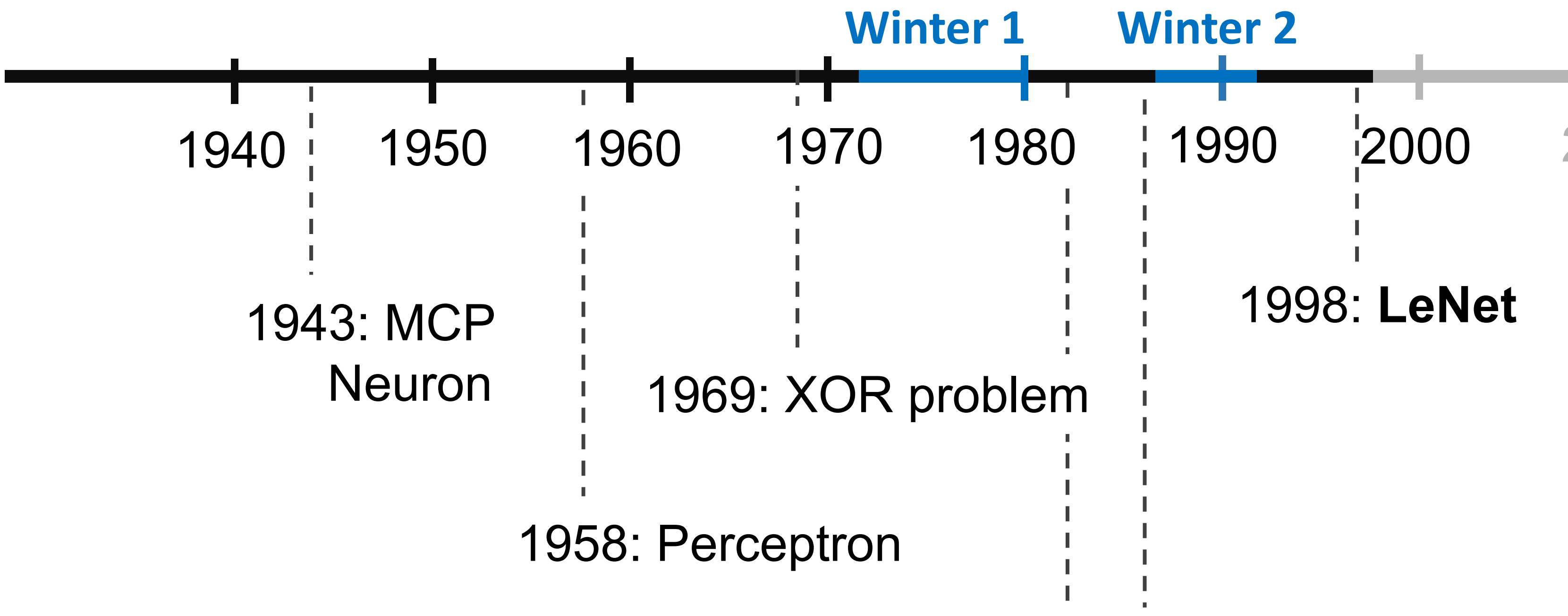
1982: Hopfield network

1986: Jordan network

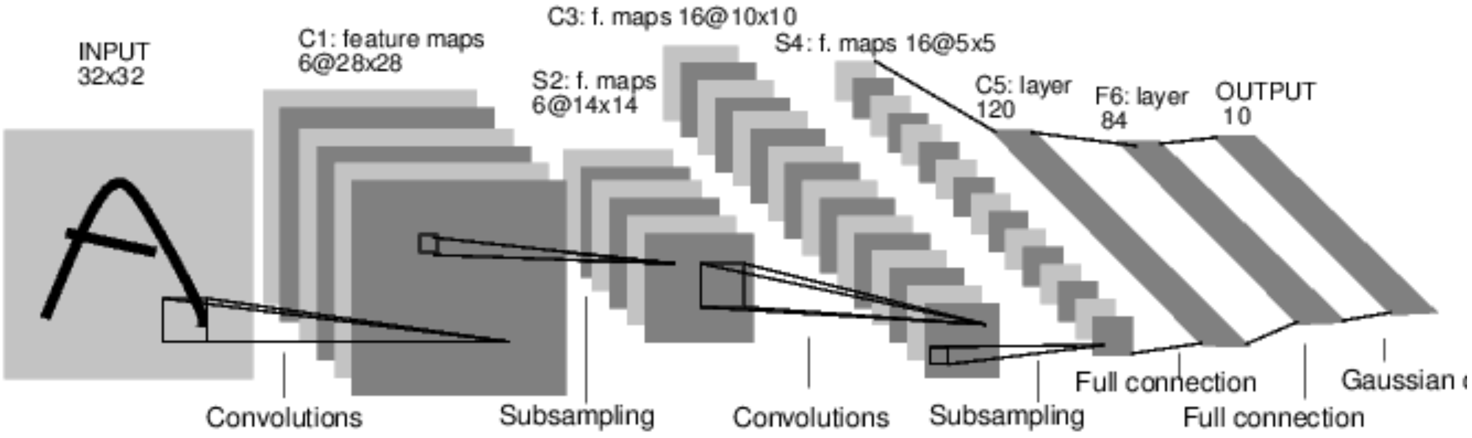
1983: Neocognitron

Problems:

- Large computation
- Overfitting
- Vanishing gradients


Winter 1
Winter 2
1940
1950
1960
1970
1980
1990
2000
2010
2020
...
1943: MCP Neuron
1958: Perceptron
1969: XOR problem
1998: **LeNet**

Yann Lecun


INPUT 32x32
C1: feature maps 6@28x28
S2: f. maps 6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer 120
F6: layer 84
OUTPUT 10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian


AT&T
LeNet 5
RESEARCH
answer: 384

Winter 1 Winter 2

1940 1950 1960 1970 1980 1990 2000 2010 2020 ...

1943: MCP Neuron

1958: Perceptron

1969: XOR problem

1982: Hopfield network

1983: Neocognitron

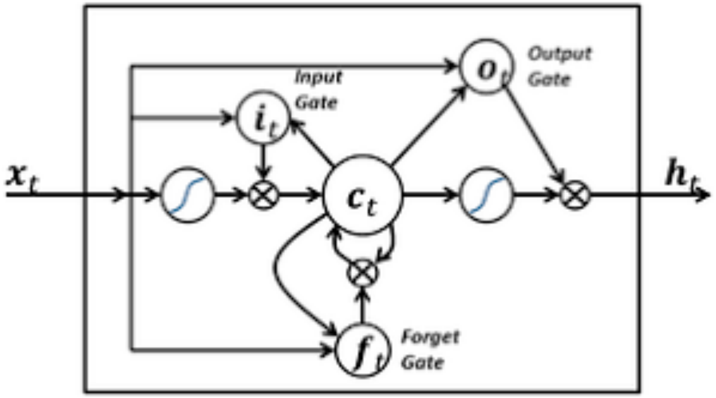
1986: Backpropagation

1986: Jordan network

1997: **LSTM**

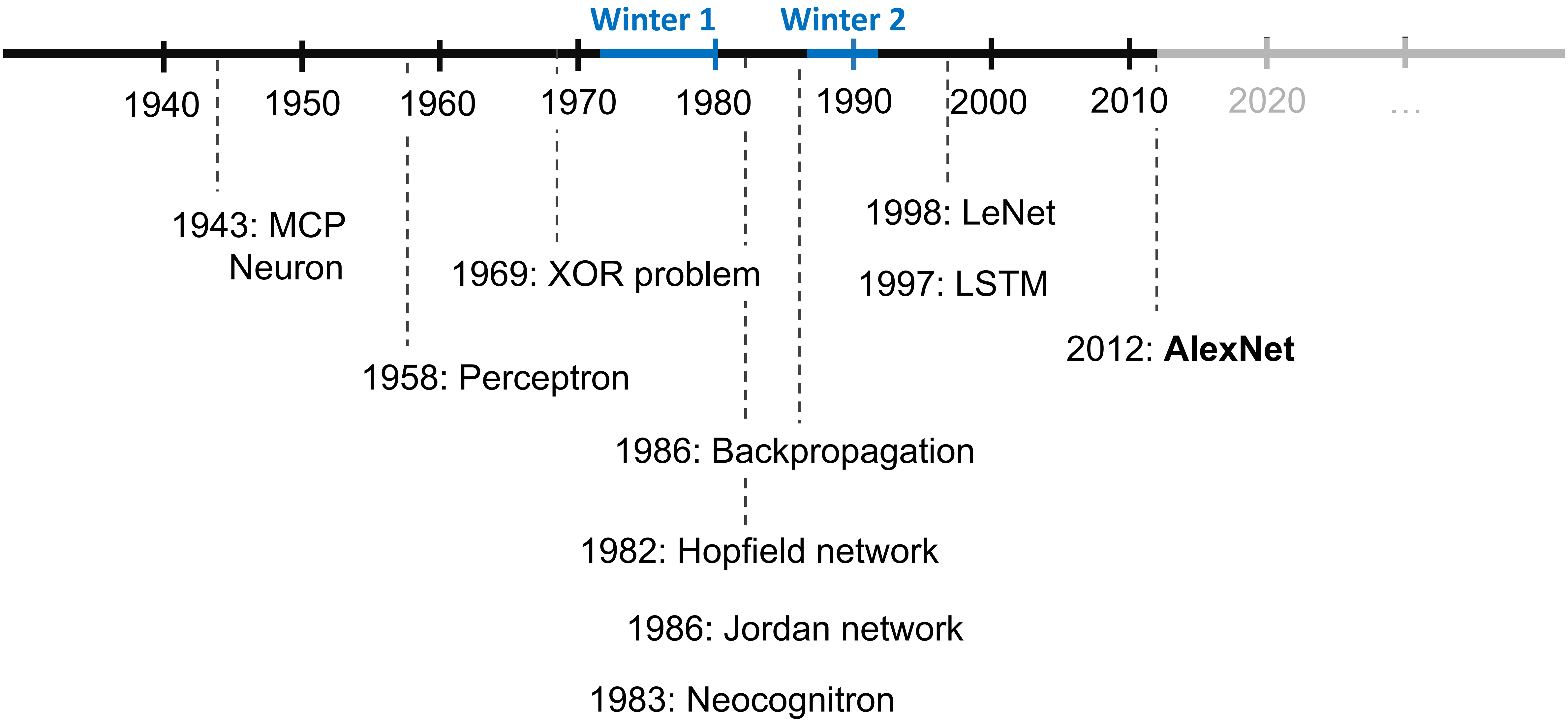
1998: LeNet

Jurgen Schmidhuber

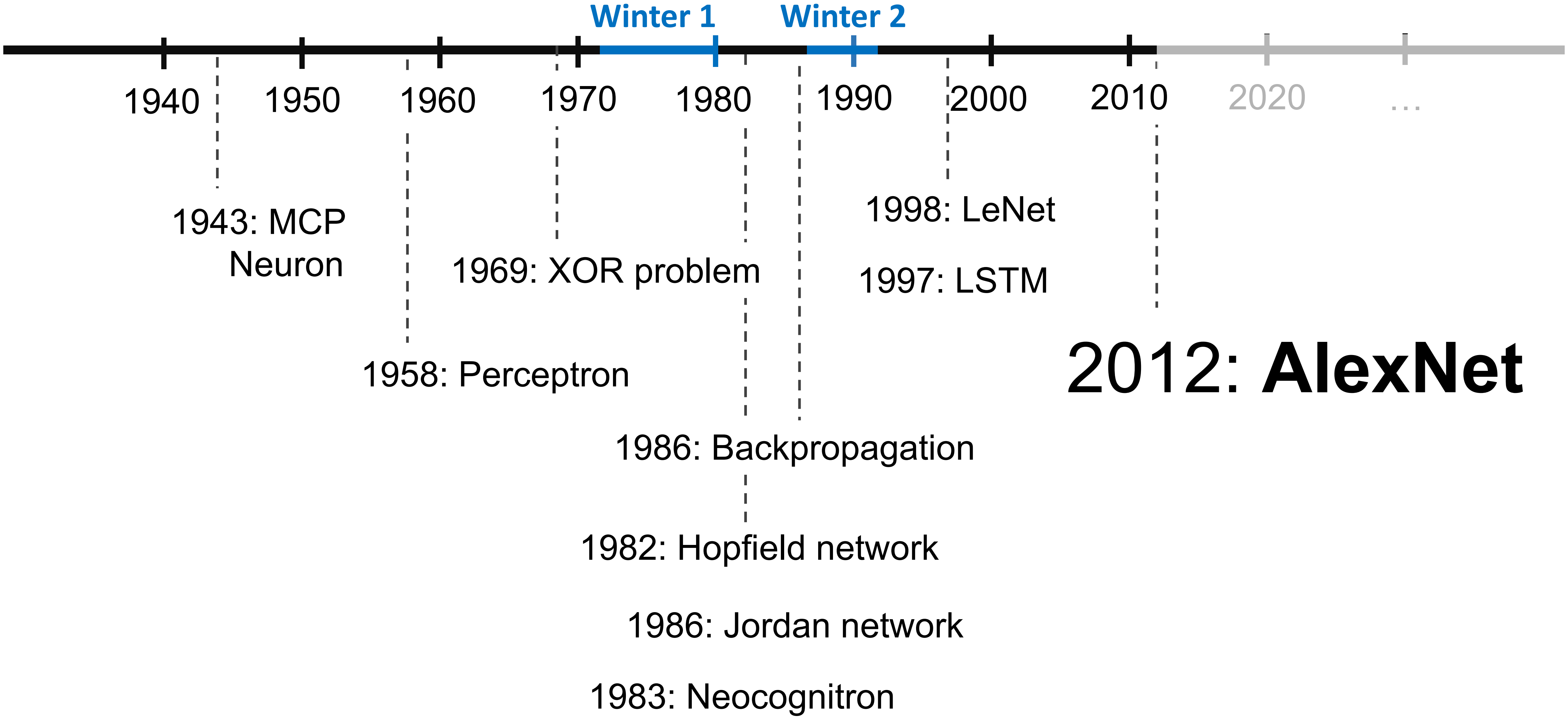


Long short-term memory (LSTM)

- Gates & identity functions
- Ameliorates vanishing gradients
- Learns long-term dependencies



Winter 1
Winter 2
1940
1950
1960
1970
1980
1990
2000
2010
2020
...
1943: MCP Neuron
1958: Perceptron
1969: XOR problem
1982: Hopfield network
1983: Neocognitron
1986: Backpropagation
1986: Jordan network
1997: LSTM
1998: LeNet
2012: **AlexNet**


Winter 1
Winter 2
1940
1950
1960
1970
1980
1990
2000
2010
2020
...
1943: MCP Neuron
1958: Perceptron
1969: XOR problem
1982: Hopfield network
1983: Neocognitron
1986: Backpropagation
1986: Jordan network
1997: LSTM
1998: LeNet
2012: **AlexNet**

2

2000 2010 2020 …

998: LeNet

997: LSTM

2012: **AlexNet**

gation

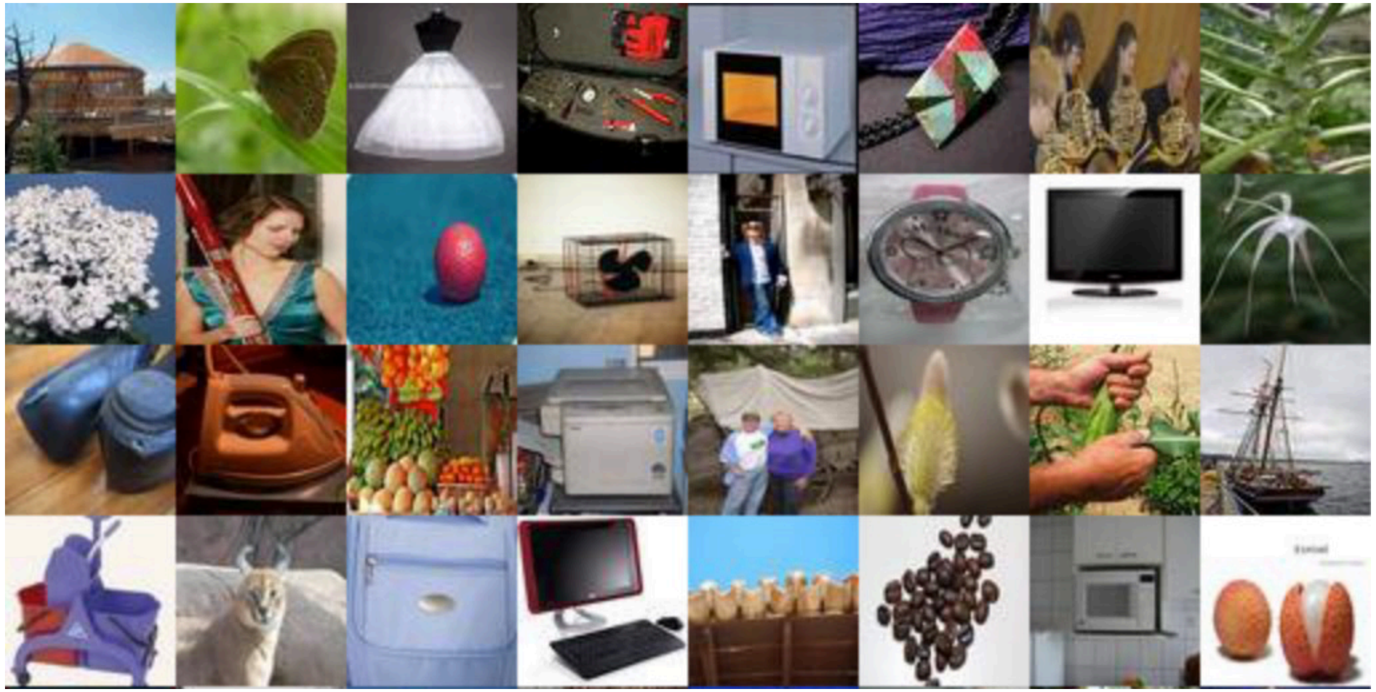
ork

twork

n

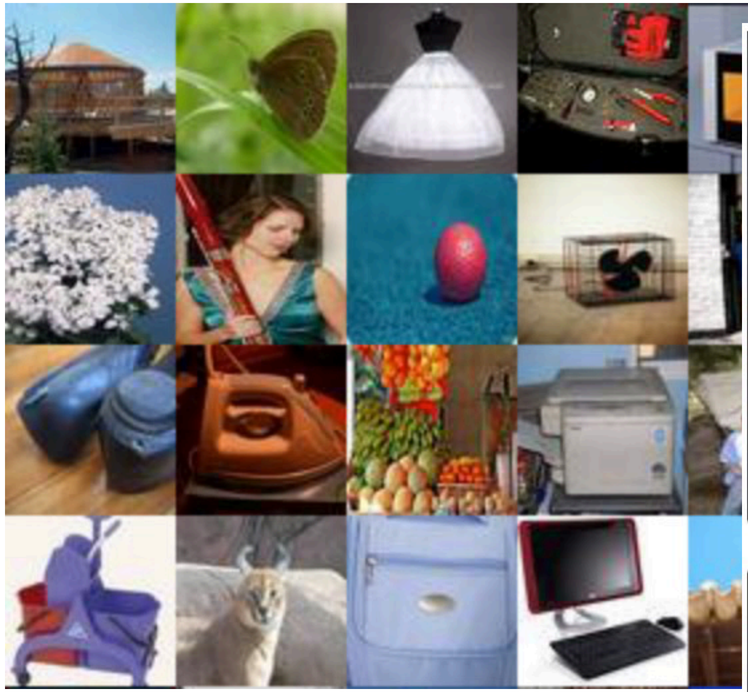
IMAGENET **Challenge**

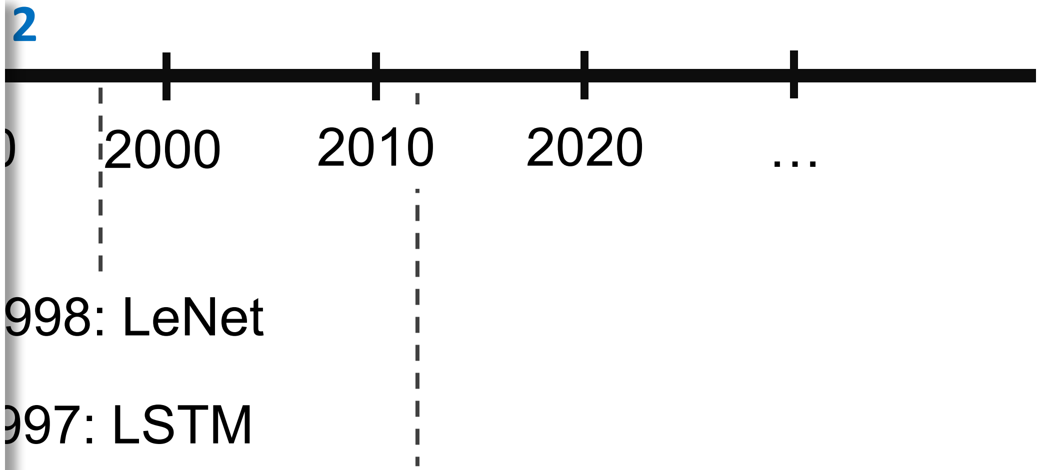
- 1.3M images
- 1000 categories



IMAGENET Challenge

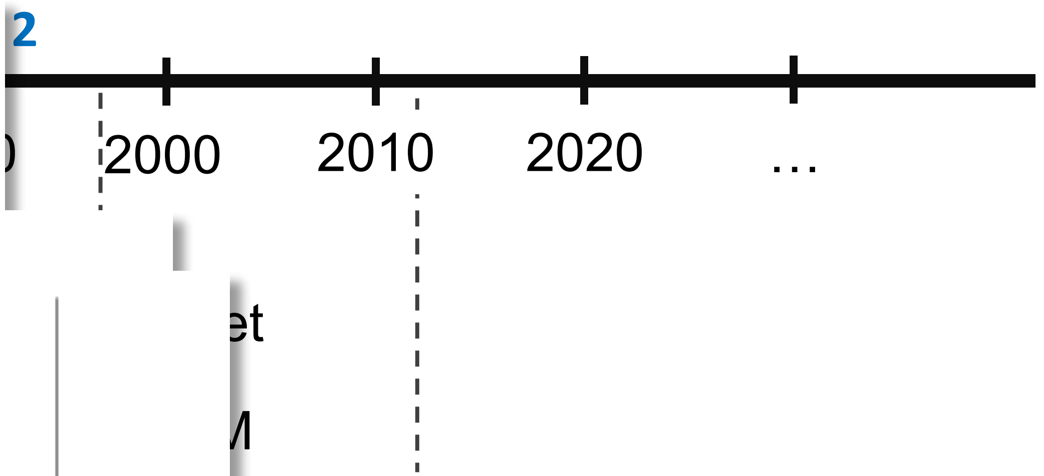
- 1.3M images
- 1000 categories

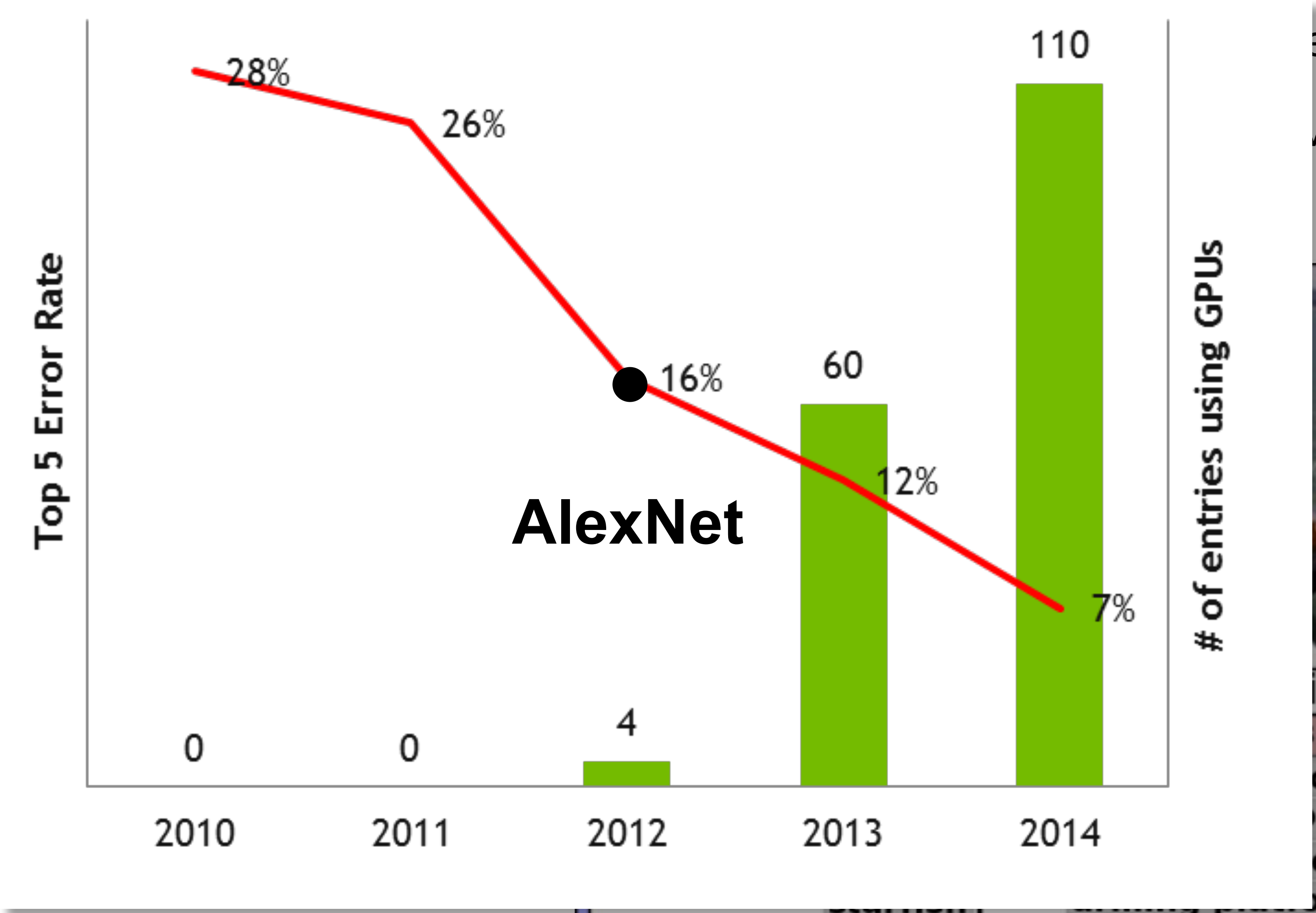




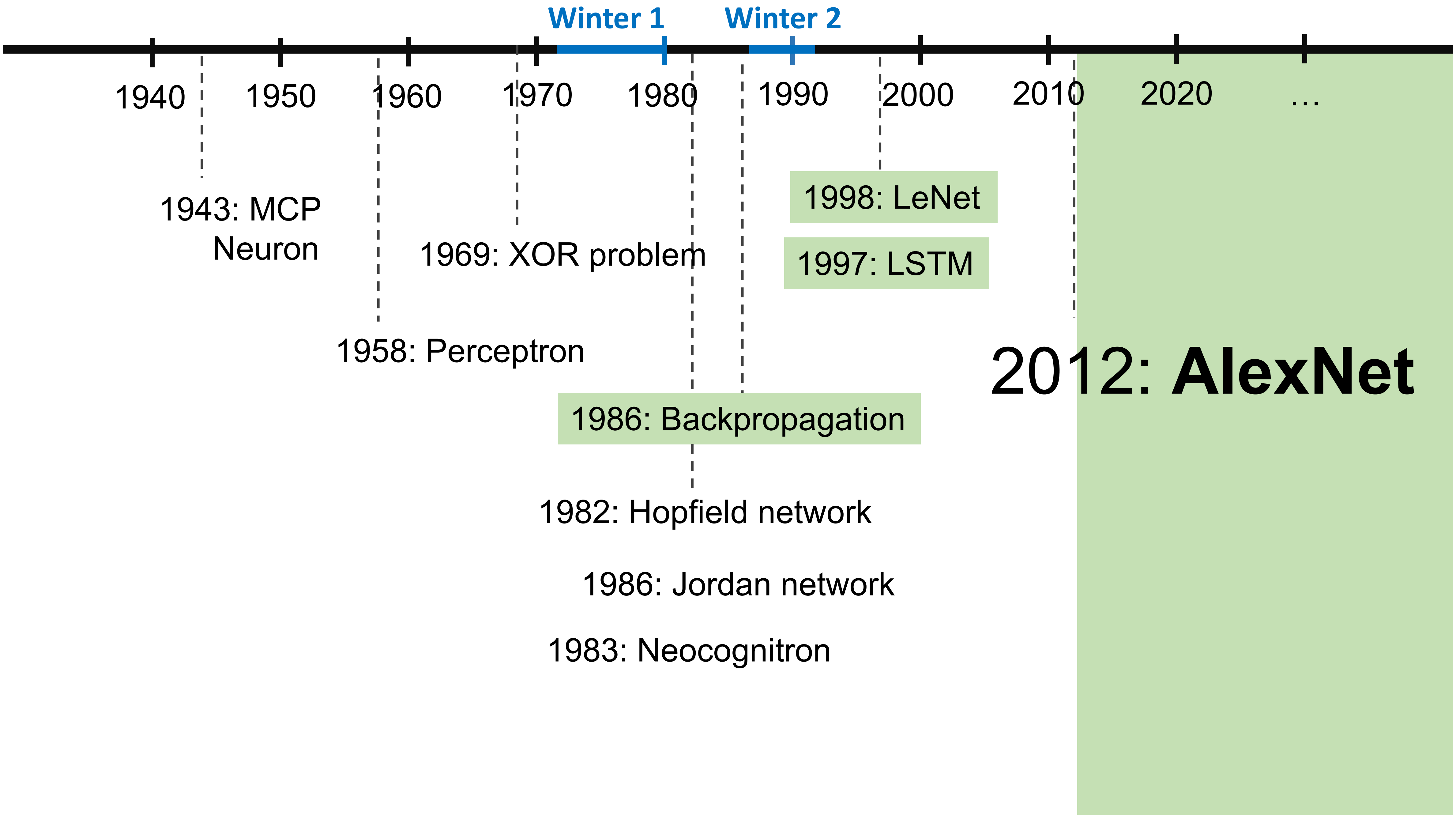
2012: **AlexNet**

IMAGENET **Challenge**



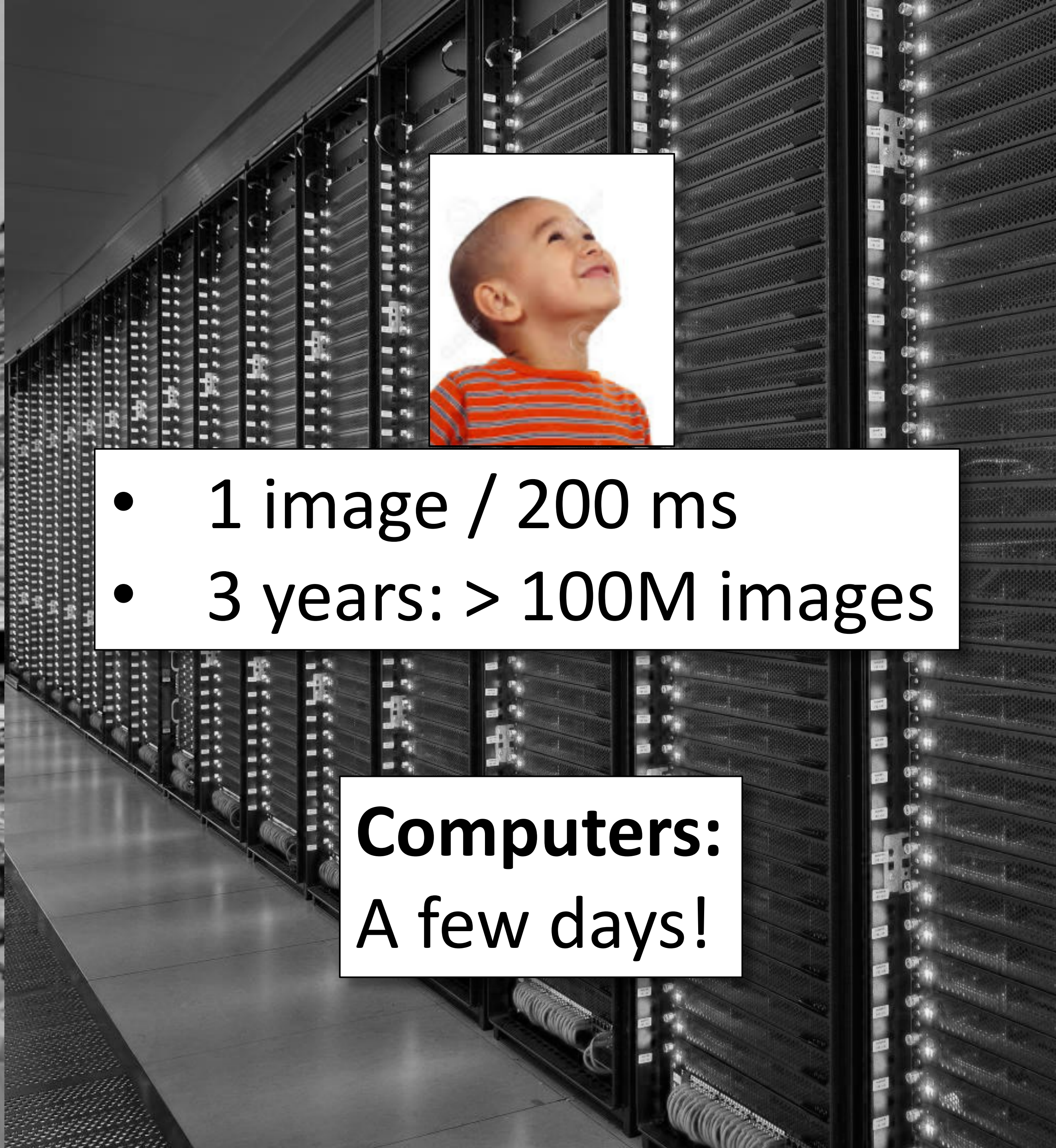


2012: **AlexNet**


Winter 1
Winter 2
1940
1950
1960
1970
1980
1990
2000
2010
2020
...
1943: MCP Neuron
1969: XOR problem
1958: Perceptron
1998: LeNet
1997: LSTM
1986: Backpropagation
1982: Hopfield network
1986: Jordan network
1983: Neocognitron
2012: **AlexNet**

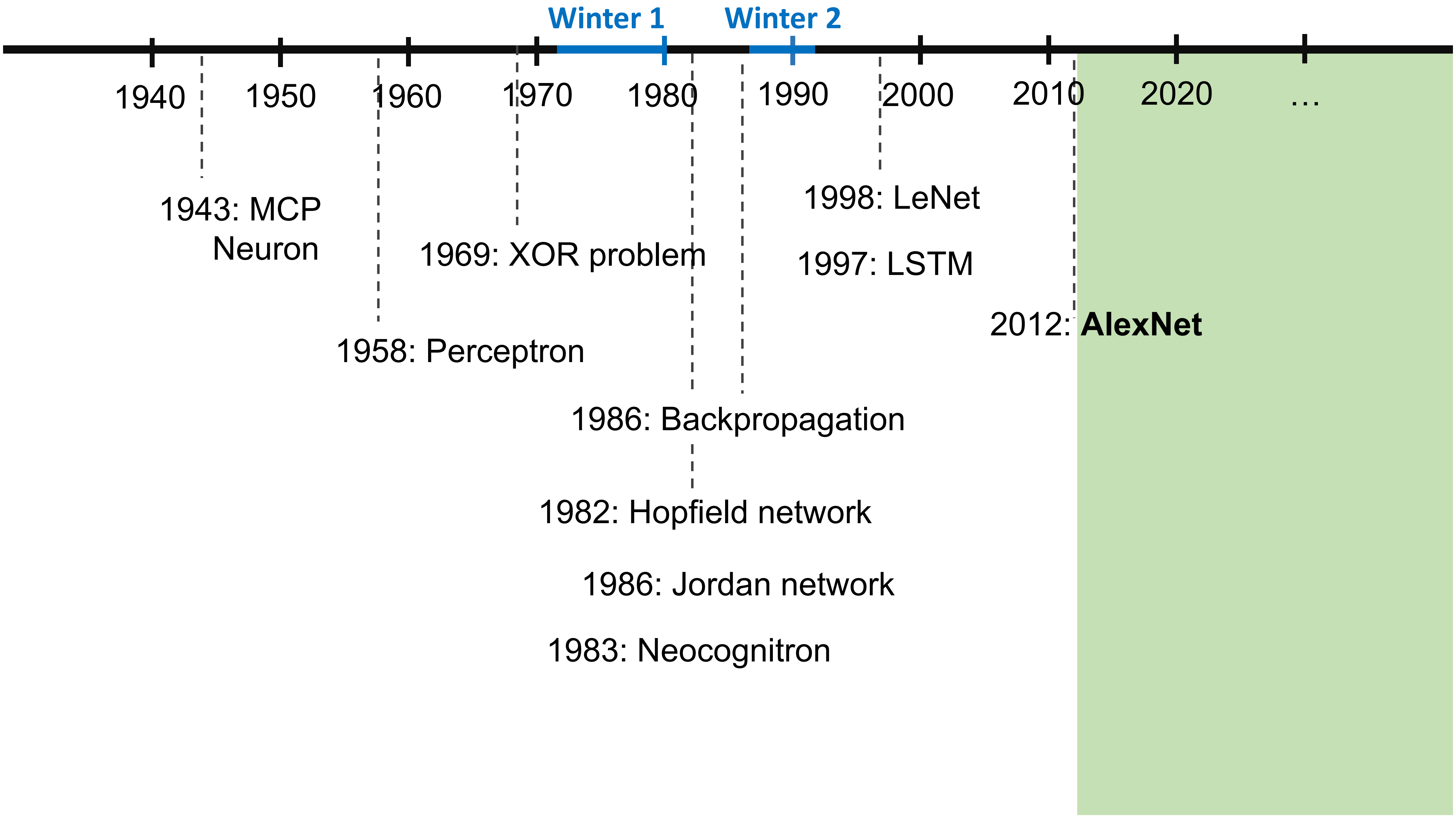

5 hrs YouTube video
per
second

5 hrs YouTube video per second



- 1 image / 200 ms
- 3 years: > 100M images

Computers:
A few days!


Winter 1
Winter 2
1940
1950
1960
1970
1980
1990
2000
2010
2020
...
1943: MCP Neuron
1958: Perceptron
1969: XOR problem
1982: Hopfield network
1983: Neocognitron
1986: Backpropagation
1986: Jordan network
1997: LSTM
1998: LeNet
2012: **AlexNet**

credit: Nvidia

Q: Which problems may NOT be solved by Deep Learning?

credit: Nvidia

Everyone is getting into Deep Learning!

Everyone is getting into Deep Learning!

LEARNING GRAPHICAL STATE TRANSITIONS

Daniel D. Johnson
Department of Computer Science
Harvey Mudd College
301 Platt Boulevard
ddjohnson@hmc.edu

Undergraduate student @

- Single author
- Oral at ICLR 2017 (~10%)

ABSTRACT

Graph-structured data is important in modeling relationships between multiple entities, and can be used to represent states of the world as well as many data structures. Li et al. (2016) describe a model known as a Gated Graph Sequence Neural Network (GGS-NN) that produces sequences from graph-structured input. In this work I introduce the Gated Graph Transformer Neural Network (GGT-NN), an extension of GGS-NNs that uses graph-structured data as an intermediate representation. The model can learn to construct and modify graphs in sophisticated ways based on textual input, and also to use the graphs to produce a variety

HARVEY MUDD COLLEGE

Everyone is getting into Deep Learning!

META LEARNING SHARED HIERARCHIES

Kevin Frans
Henry M. Gunn High School
Work done as an intern at OpenAI
kevinfrans2@gmail.com

John Schulman
OpenAI

bbeel
ctrical
ce

We develop a metalearning ap
cies, improving sample effici
primitives—policies that are executed for large numbers of timesteps. Specifi-

Everyone is getting into Deep Learning!

You can too! :-)

About the course

Class staff

Instructor: Dr. Anh Nguyen

- Email: anhnguyen@auburn.edu
- Office: Shelby 3101R
- MW: 10-11am

TA: Chengfei Wang

- Email: czw0078@auburn.edu
- Office: Shelby 2105
- TR: 10-11:30am

Pre-requisites:

- Calculus, Linear Algebra, Probability & Statistics
- Proficiency with Python (and Linux scripting)
  - All class assignments will be in Python (based on Stanford cs231n assignments)
- Introduction to Machine Learning
  - Forming loss functions, taking derivatives and stochastic gradient descent
  - Linear regression, SVM, PCA, etc.

- Q: How hard would the assignments be?
  - Self-test: HW 0 https://goo.gl/hgtDyk

Expectation

- Demanding in programming (Python)
- Reviewing / presenting (DL) research papers
- Project-based (deliverables: technical report + video)

Basics of NN / DL	Latest research papers (and your own project!)

[Stanford CS231n]

Module	Week	Event	Date	Description	Presenter	Notes
	1	L1	1/10	Course introduction & logistics		
		L2	1/12	Image Classification		A1 out
Training CNNs	2	*No class*	1/15	*M.L. King Day*		
		L3	1/17	Loss Functions and Optimization		
		L4	1/19	Introduction to Neural Networks		
	3	L5	1/22	Convolutional Neural Networks		
		L6	1/24	Training Neural Networks, part I		
		L7	1/26	Training Neural Networks, part II		**A1 due**
	4	L8	1/29	Training Neural Networks, part III		A2 out
		L9	1/31	CNN Architectures		**Proposal due**
		Presentation	2/2	Stochastic Depth Network	Group 1	
Understanding CNNs	5	L10	2/5	Introduction to Caffe	Chengfei	
		L11	2/7	Visualizing and Understanding CNNs, part I		
		Presentation	2/9	Network Dissection (Zhou et al. 2017)	Group 2	
	6	L12	2/12	Visualizing and Understanding CNNs, part II		
		L13	2/14	Fooling CNNs		**A2 due**
		Presentation	2/16	Celebrity glasses CMU (Sharif et al. 2017)	Group 3	
AI for Creativity	7	*No class*	2/19	*Presidents' Day*		
		L14	2/21	DeepDream / Style Transfer / Smile Vector / Creative AI		A3 out
		L15	2/23	Introduction to Tensorflow & Keras	Chengfei	
Detection & Segmentation	8	L16	2/26	Detection CNNs		
		L17	2/28	Segmentation CNNs		
		Presentation	3/2	DeepMask	Group 4	
Recurrent Neural Networks	9	L18	3/5	Recurrent Neural Networks part I		
		L19	3/7	Recurrent Neural Networks part II		
		Presentation	3/9	Visualizing & Understanding RNNs (Karpathy et al. 2015)	Group 5	
		No class	3/12 3/14 3/16	*Spring Break*		

Sequence to sequence	10	L20	3/19	Machine Translation		
		L21	3/21	Speech		**A3 due**
		Presentation	3/23	Neural Programmer-Interpreter (Reed & Freitas, 2015)	Group 6	
Theory of Deep Learning	11	L22	3/26	Theory of Deep Learning I		
		L23	3/28	Theory of Deep Learning II		
		Presentation	3/30	Information Bottleneck (Tishby 2017)	Group 7	
Generative Models	12	L24	4/2	Intro to Generative Models / Unsupervised Learning		
		L25	4/4	Autoregressive / PixelCNN		
		Presentation	4/6	SketchRNN (Ha & Douglas. 2017)	Group 8	
	13	L26	4/9	Variational methods / VAE		
		L27	4/11	Generative Adversarial Networks (GAN)		
		Presentation	4/13	VQ-VAE (Oord et al. 2017)	Group 9	
Deep RL	14	L28	4/16	Deep Reinforcement Learning I		
		L29	4/18	Deep Reinforcement Learning II		
		Presentation	4/20	AlphaGo / AlphaZero (DeepMind 2017)	Group 10	
Project	15	Presentation	4/23	Project Day 1		**Project due**
		Presentation	4/25	Project Day 2		
		Presentation	4/27	Project Day 3		
	16			Final exam		

Grading policy & Expectation

Group of 2

1. **Paper presentation:** 30%
    - Study one paper of *your* choice (with instructor approval)
        - Understand paper (including math)
        - Show a demo
        - Understand code (bonus for your own implementation)

    ❑ Present paper/demo/code [7970 - group]
    ❑ Ask questions & provide feedback on presentations [4970 - individual]
    ❑ Write a formal paper review [7970 - group]

Group of 2-3

2. **Project:** 40%
    - Application: Applying DL to your own project?
    - Research: Doing something cool / new / exciting? Extending a paper?

    ❑ Present project [7970, 4970]
    ❑ Write a paper [7970, 4970]

3. **Assignments:** 30%
    - 3 programming assignments x 10% (based on cs231n assignments) [7970, 4970]

Action items:

1. **Each graduate student:** presentation topic (deadline: Jan 22)
   - Send me your preference: (1) topic name; (2) paper cite (name & author)
   - Topics are first come first served!
2. **Everyone:** propose your project topic (deadline: Jan 31)
   - Encouraged to talk to classmates / TA / instructor
   - Form a group of 2-3 (at least 2; at most 3!)
   - Deliverables: Proposal (max: 1-page pdf)
     1. What is the problem that you will be investigating? Why is it interesting?
     2. What data will you use?
     3. What method or algorithm are you proposing/modifying from?
     4. What reading will you examine to provide context and background?
     5. How will you evaluate your results?

References

- https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html
- https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/
- https://medium.com/@Jaconda/a-concise-history-of-neural-networks-2070655d3fec
- http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning/
- Deep Learning Book (Goodfellow et al 2017)
- ImageNet dataset