Fundamentals of Deep Learning

Dr Jony Castagna – Hartree Centre

NVidia DLI Ambassador

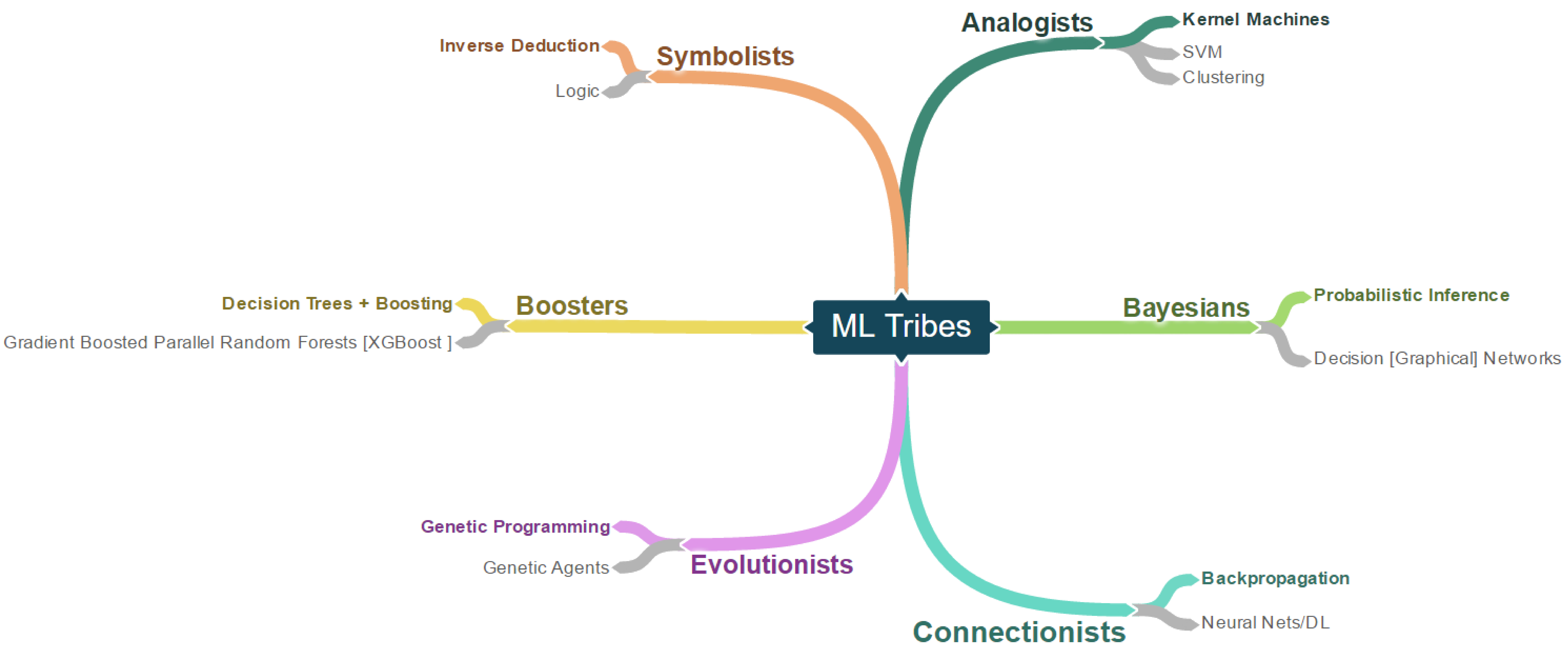
Agenda

- Part 1: An Introduction to Deep Learning
- Part 2: How a Neural Network Trains
- Part 3: Convolutional Neural Networks
- Part 4: Data Augmentation and Deployment
- Part 5: Pre-Trained Models
- Part 6: Advanced Architectures

AI …is ALREADY 70 years of RESEARCH!

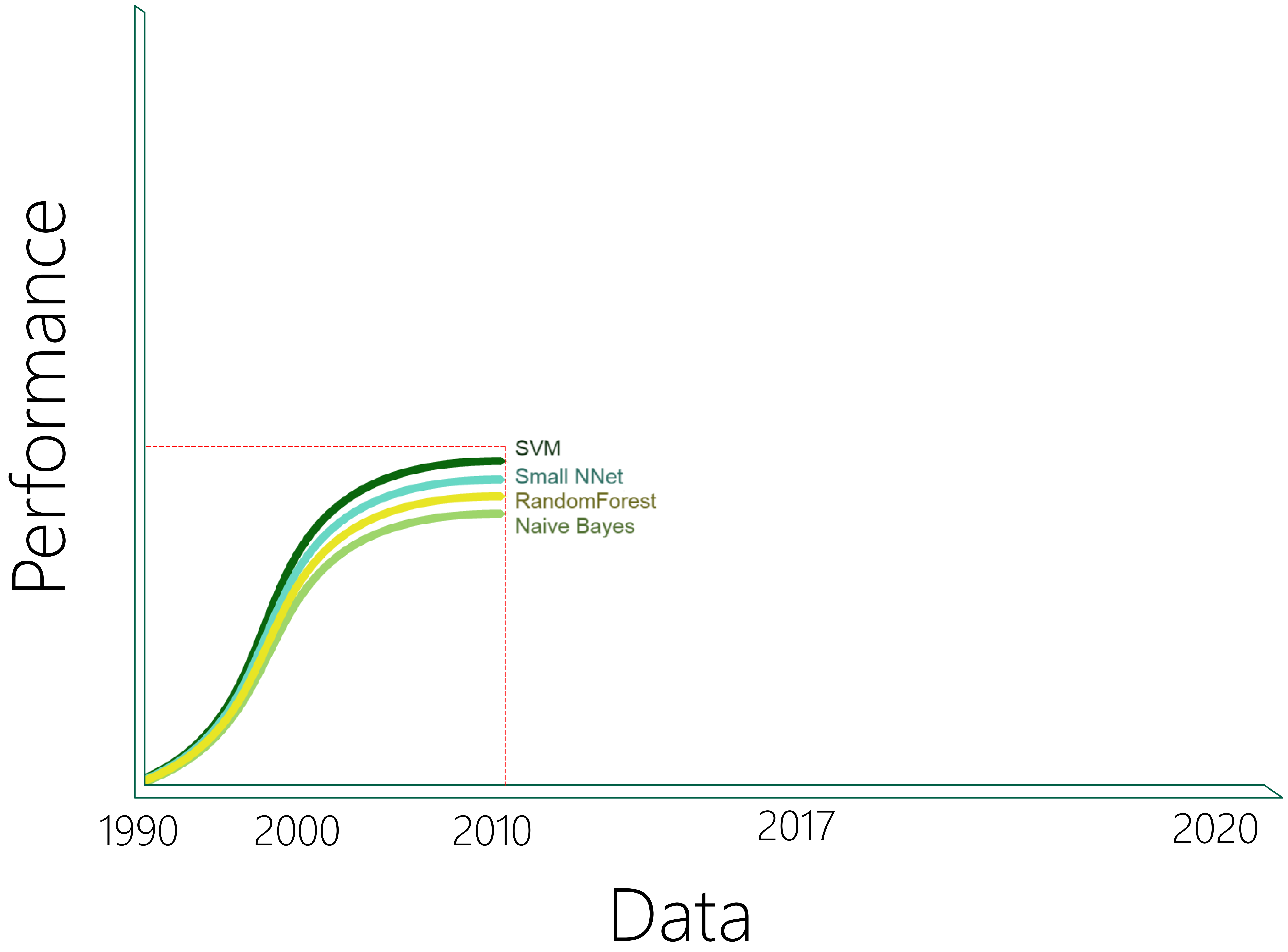
NVIDIA

ML Tribes

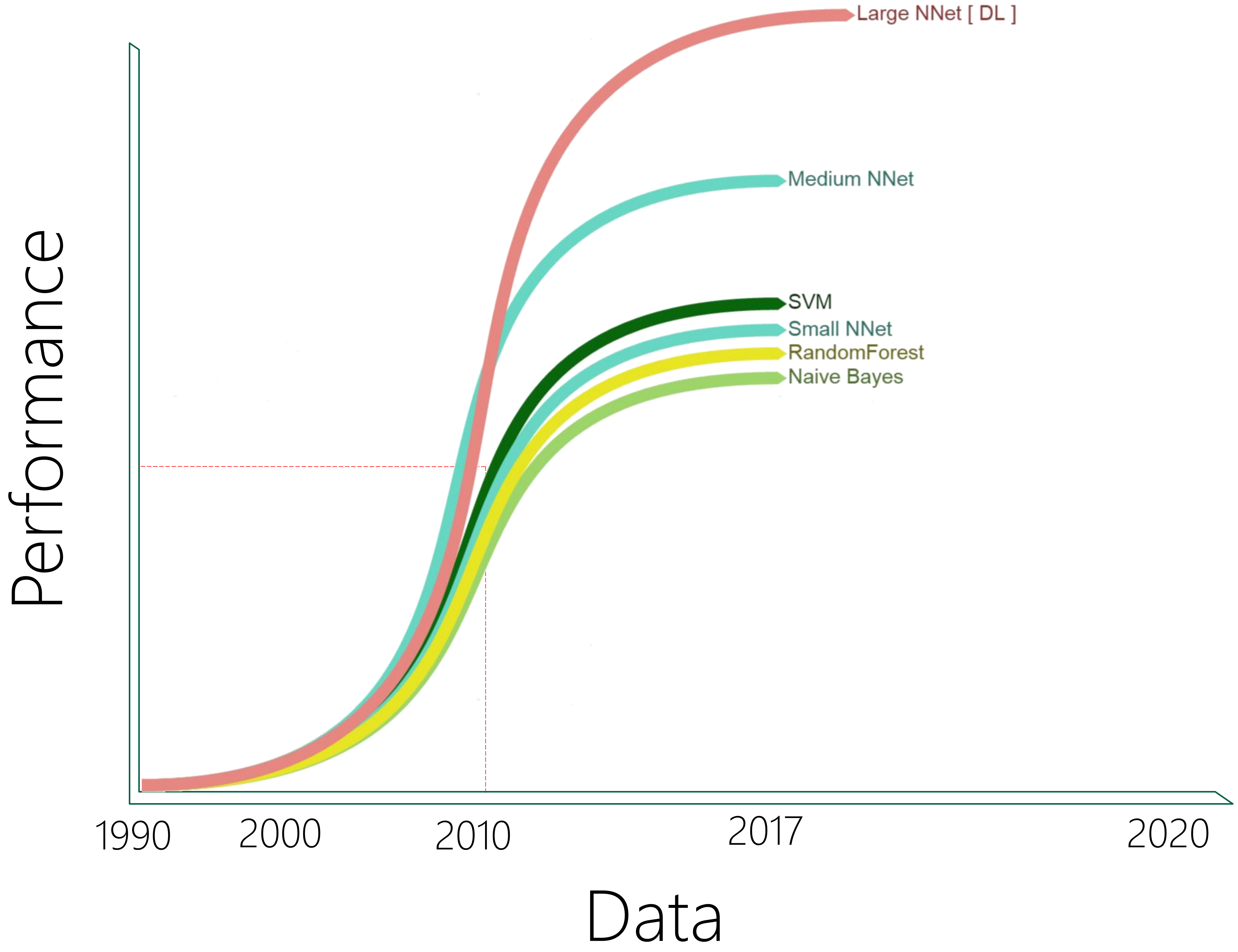


Tribe	Origins	Master Algorithm
Symbolists	Logic, philosophy	Inverse deduction
Connectionists	Neuroscience	Backpropagation
Evolutionaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

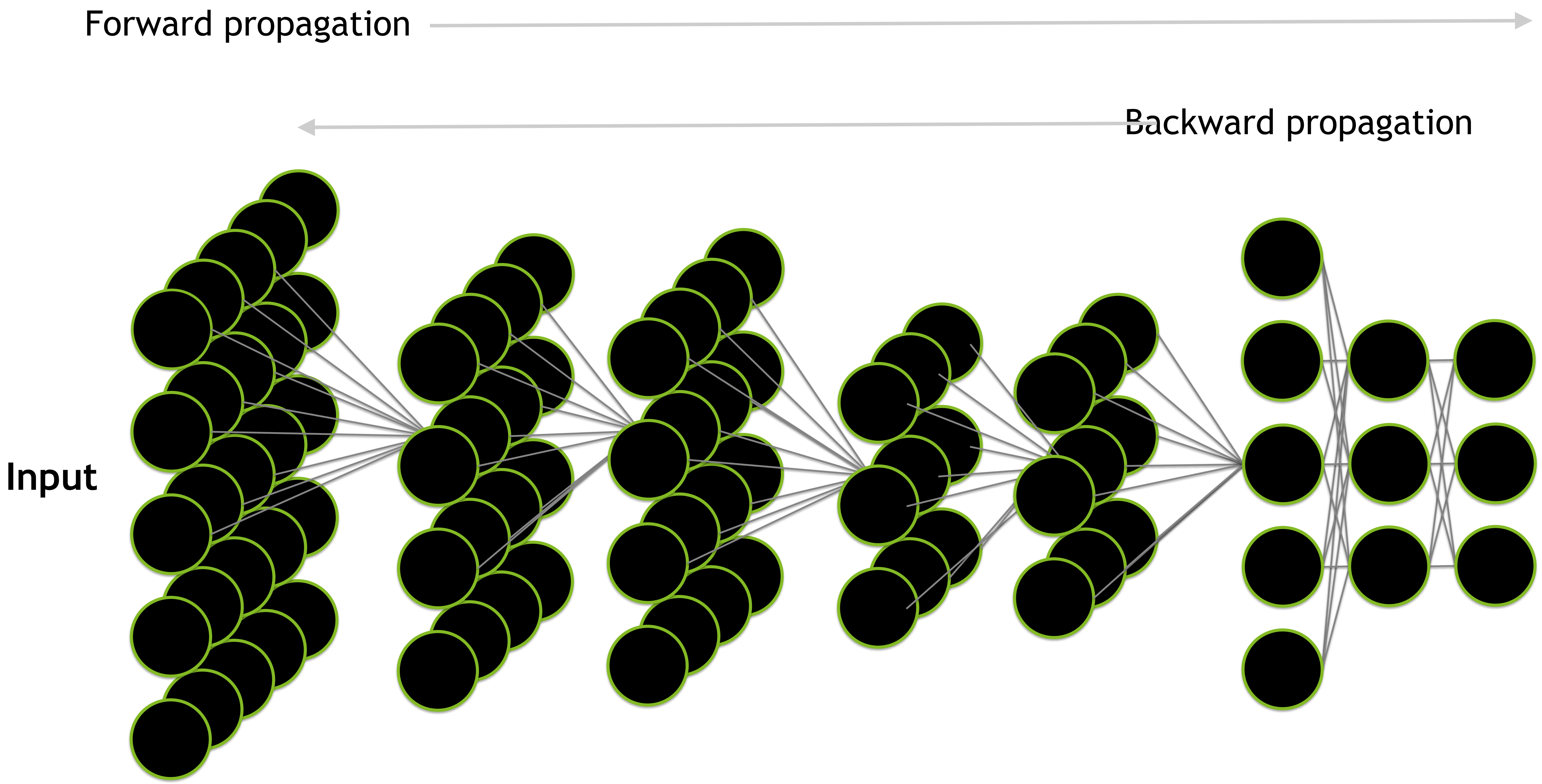
Trend #1 [Scale]



Trend #1 [Scale]



DEEP LEARNING APPROACH



Process

- Forward propagation yields an inferred label for each training image
- Loss function used to calculate difference between known label and predicted label for each image
- Weights are adjusted during backward propagation
- Repeat the process

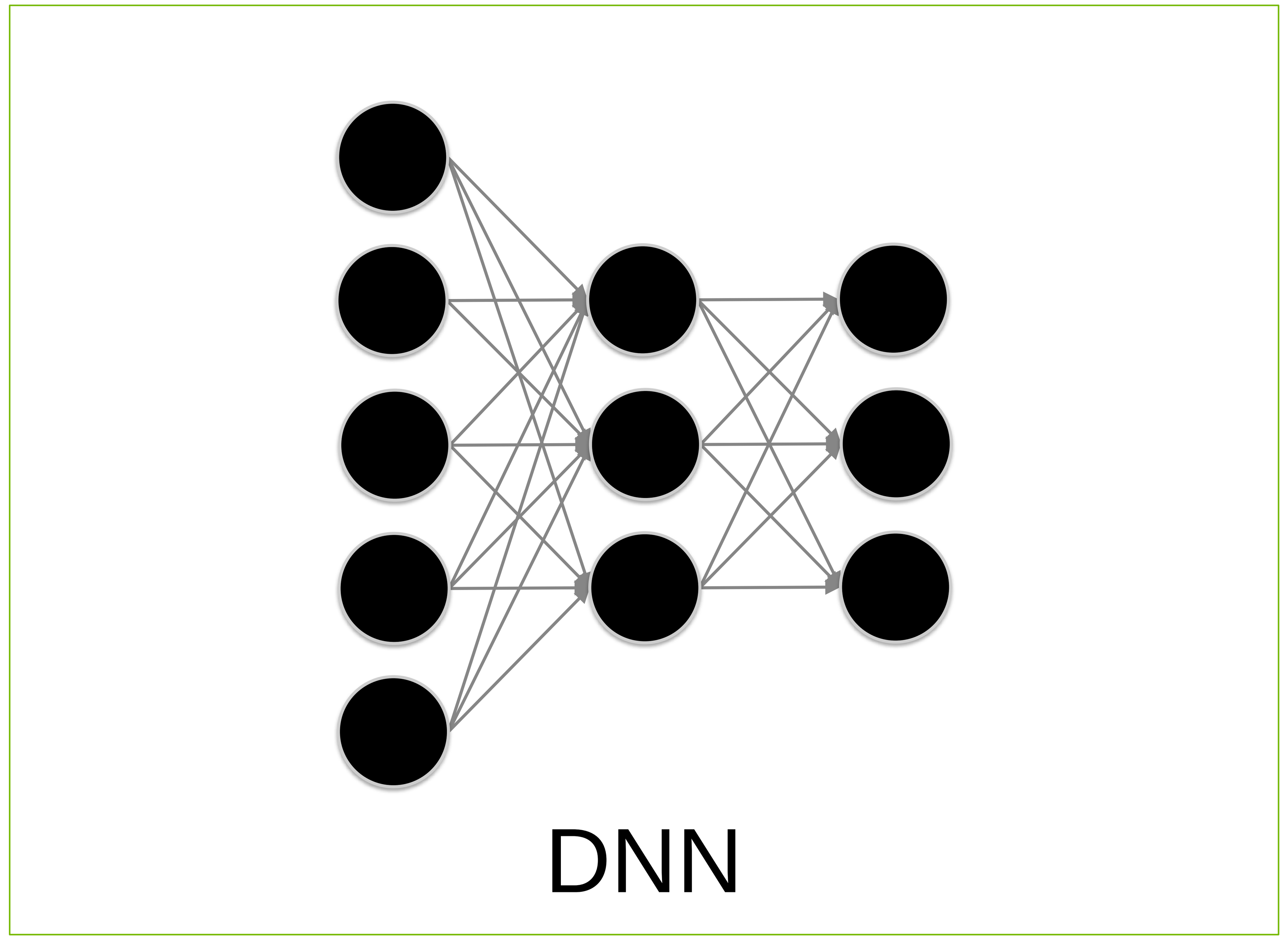
Deep Learning Compared to Other AI

- Depth and complexity of networks
- Up to billions of parameters (and growing)
- Many layers in a model
- Important for learning complex rules

Computing Power

Need a way for our artificial "brain" to observe lots of data within a practical amount of time.

THE BIG BANG IN MACHINE LEARNING



THE EXPANDING UNIVERSE OF MODERN AI

“ THE BIG BANG ”

Big Data
GPU
Algorithms

RESEARCH

Berkeley
Carnegie Mellon University
DEEPMIND
MIT
NYU
OpenAI
Université de Montréal
University of Oxford
University of Toronto

CORE TECHNOLOGY / FRAMEWORKS

Preferred Networks — Chainer
facebook. — torch
Université de Montréal — theano
Google — TensorFlow
Berkeley — Caffe
Microsoft — CNTK
University of Oxford — MatConvNet
NVIDIA. — cuDNN

AI-as-a-PLATFORM

amazon web services
IBM Watson
Google
Microsoft Azure

START-UPS

api.ai — Personal Assistants — conversational interface
BLUE RIVER TECHNOLOGY — Agriculture — crop-yield optimization
clarifai — Tech — visual recognition platform
deep genomics — Genomics — genetic interpretation
drive.ai — Automotive — computer vision
MetaMind — eCommerce & Medical — recommendation engines
Morpho — Tech — computer vision
Orbital Insight — Geospatial — predictions from images
nervana — Tech — AI-as-a-service
SADAKO — Waste Management — sorting robots
SocialEyes® — Medical — diabetic retinopathy
HOW ARE YOU — Education — teaching robots

1,000+ AI START-UPS

$5B IN FUNDING

Source: Venture Scanner

INDUSTRY LEADERS

Alibaba.com
AstraZeneca
Audi
Baidu 百度
Bloomberg
BMW
charles SCHWAB
cisco
ebay
FANUC ROBOTICS
Ford
GE
gsk
THE HOME DEPOT
MASSACHUSETTS GENERAL HOSPITAL
Mercedes-Benz
MERCK
Pinterest
Schlumberger
Shell
SIEMENS
TARGET
TESLA
TOYOTA
Twitter
UBER
VOLVO
Walmart
YAHOO
Yandex
yelp

DEEP LEARNiNG iN PRODUCTION

Speech Recognition

Recommender Systems

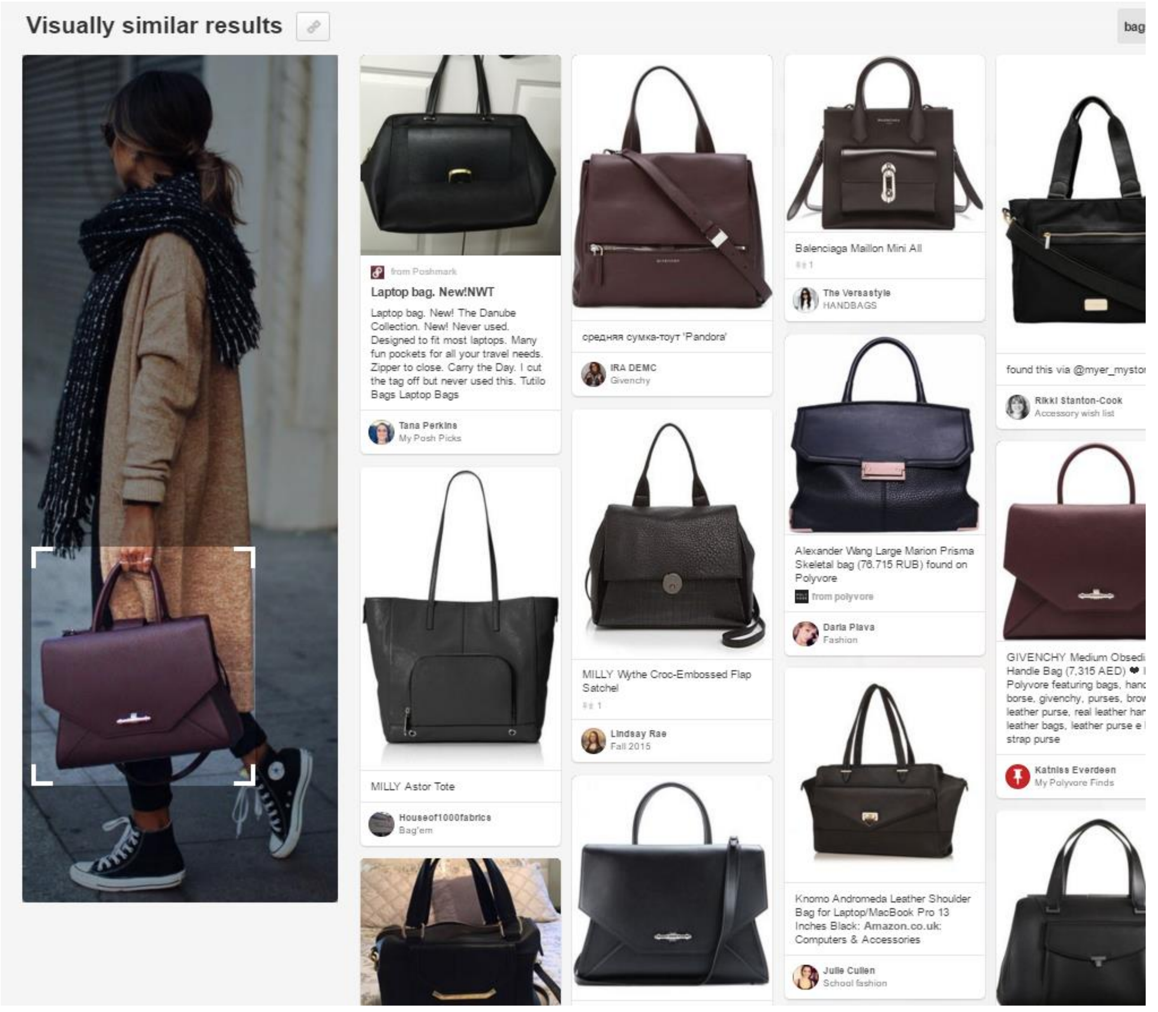
Autonomous Driving

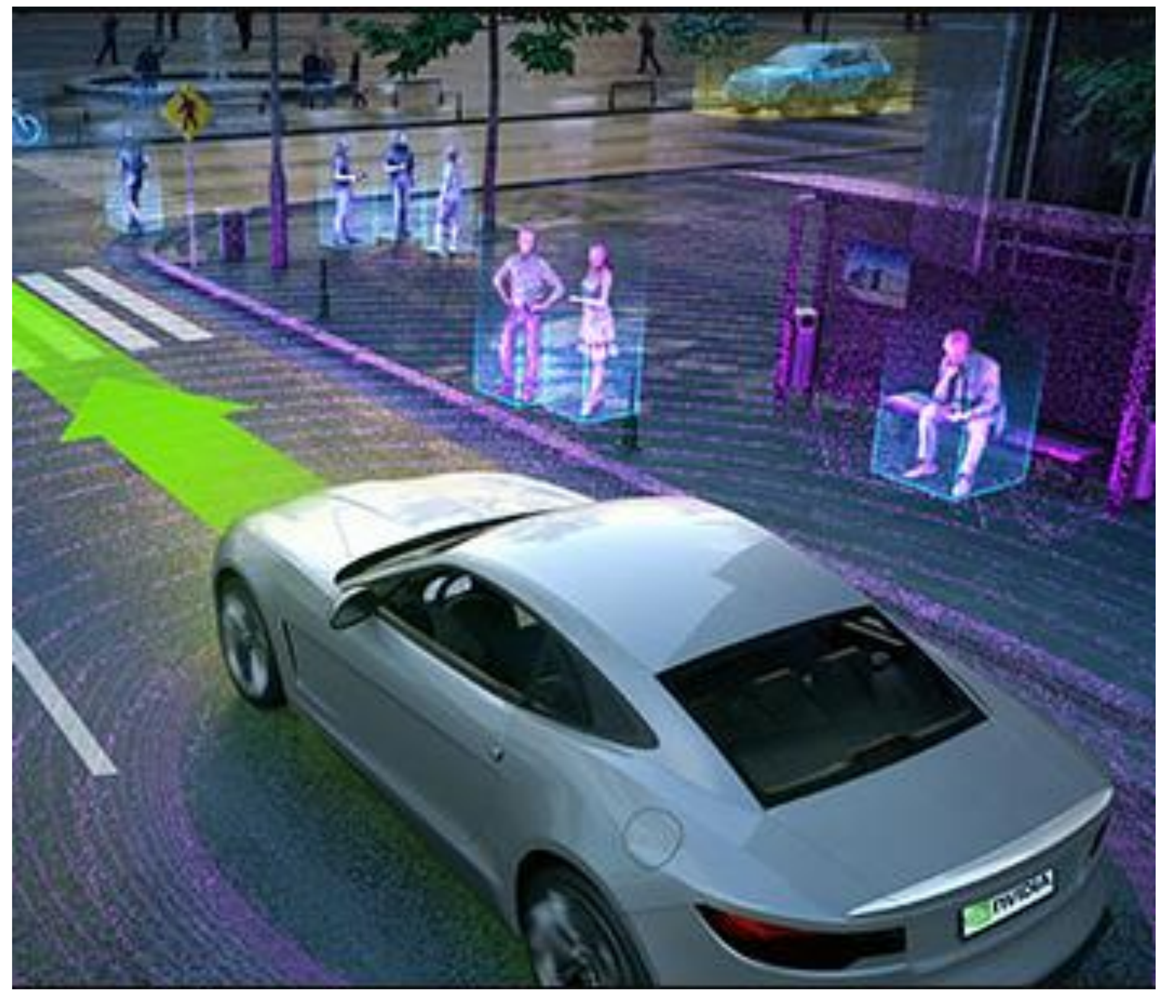
Real-time Object Recognition

Robotics

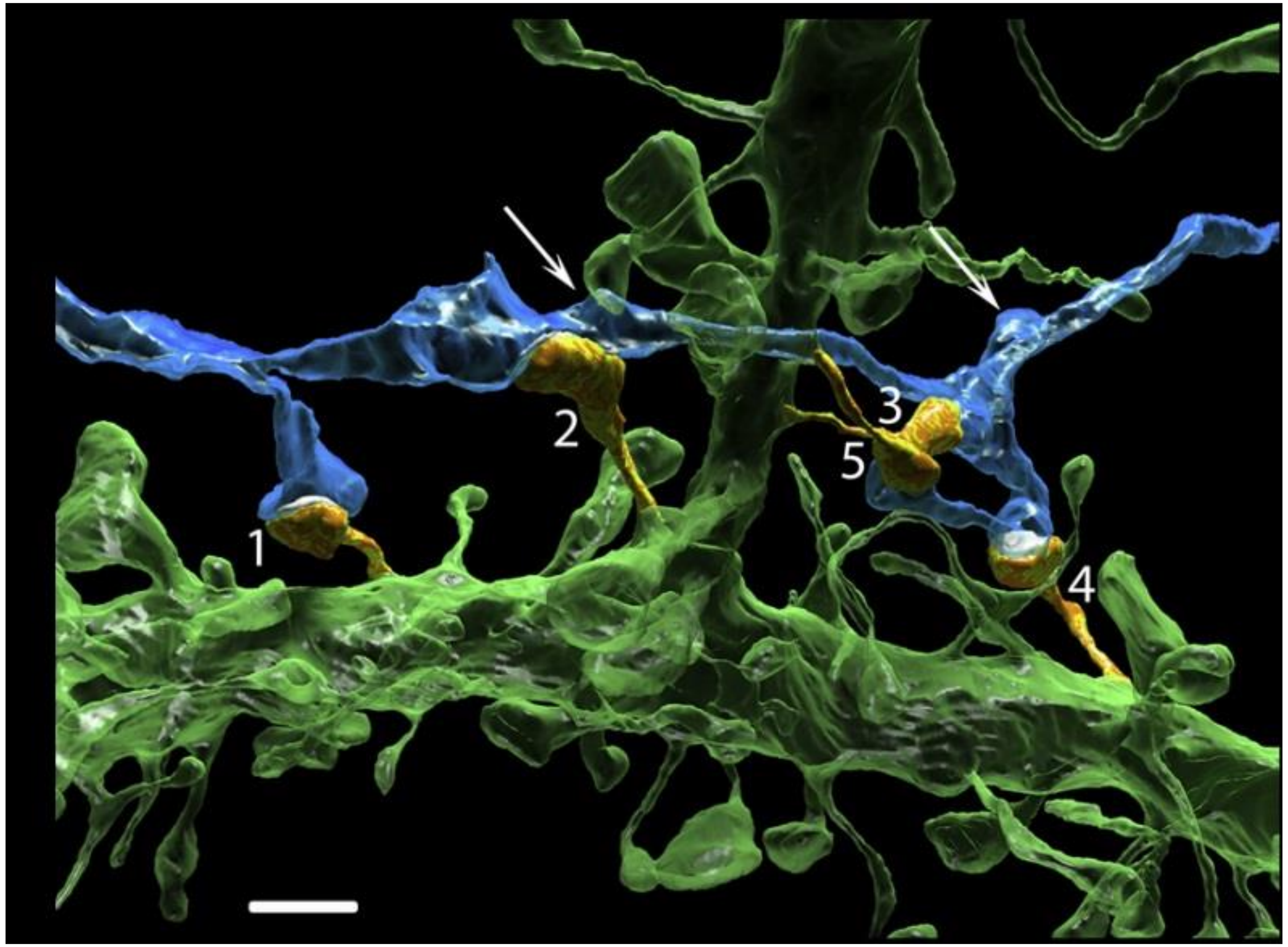
Real-time Language Translation

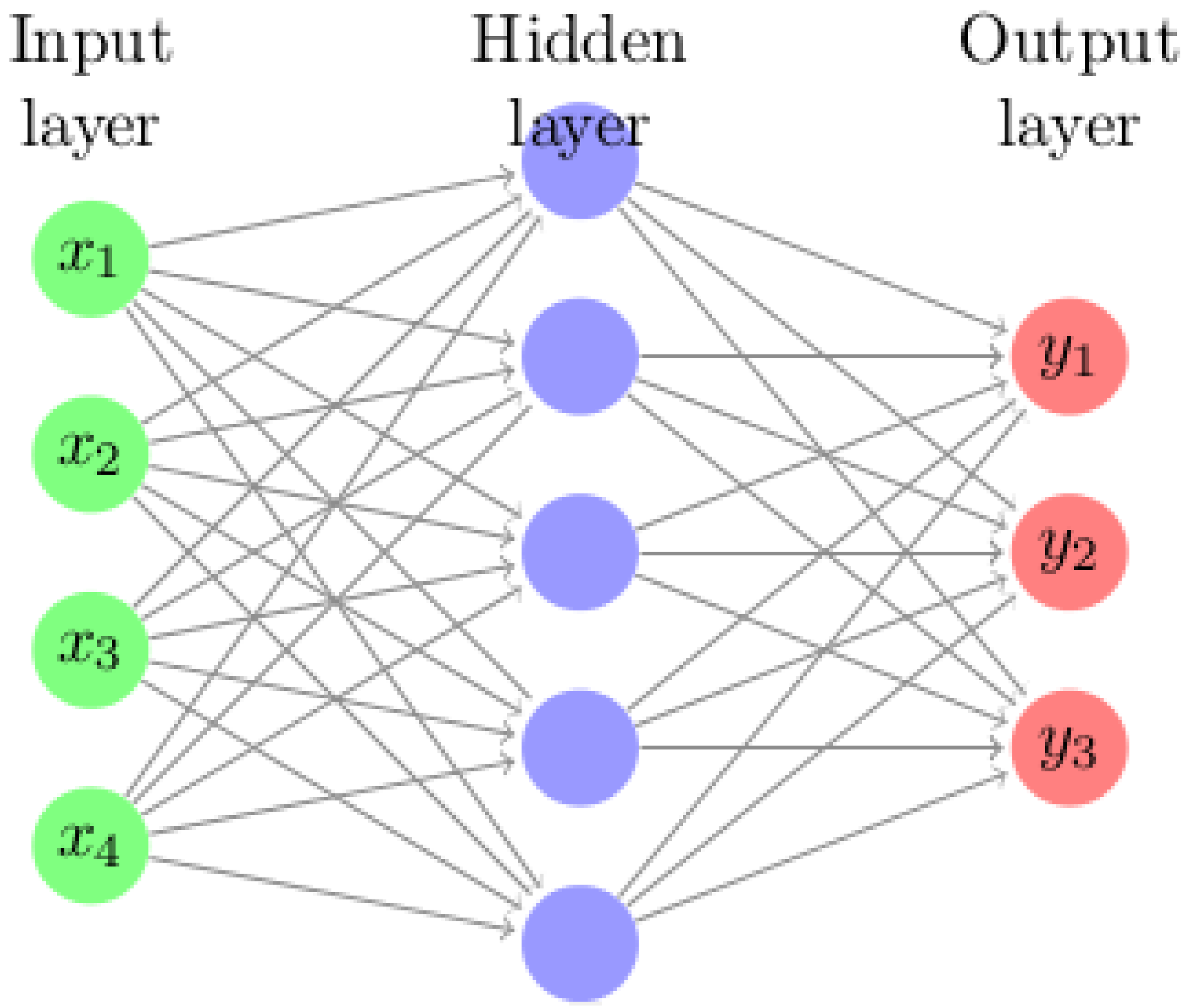
Many More…





What are NN: biological Inspiration





How Do Children Learn?

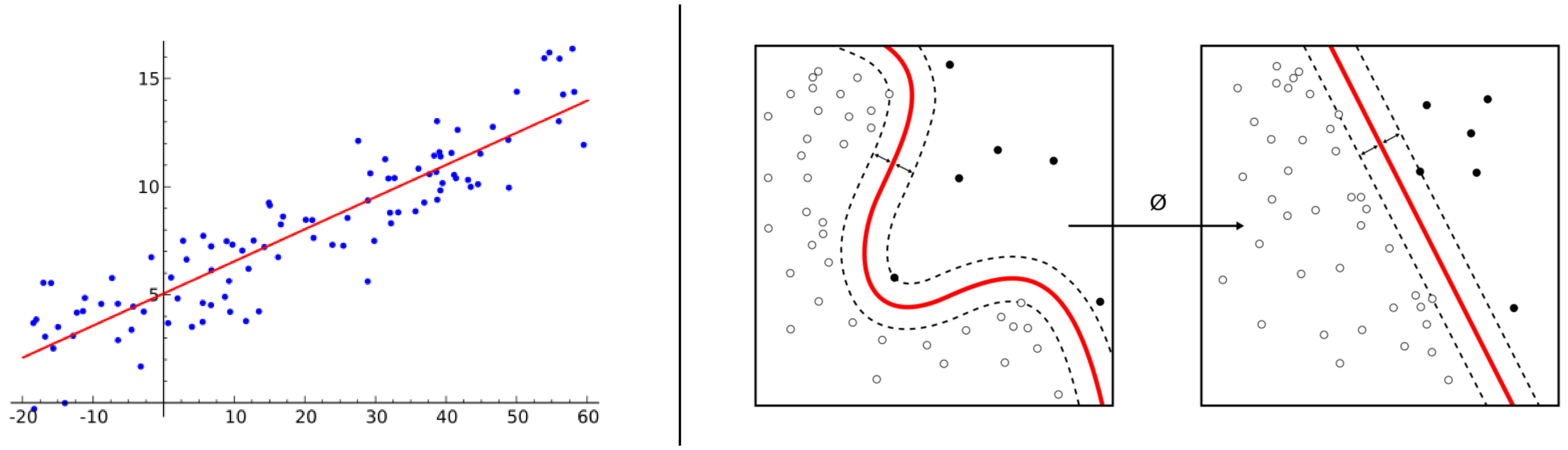
- Expose them to lots of data
- Give them the “correct answer”
- They will pick up the important patterns on their own

Difference in Workflow

Classic Machine Learning [1990 : now]

Hand Designed Features → Model / Mapping

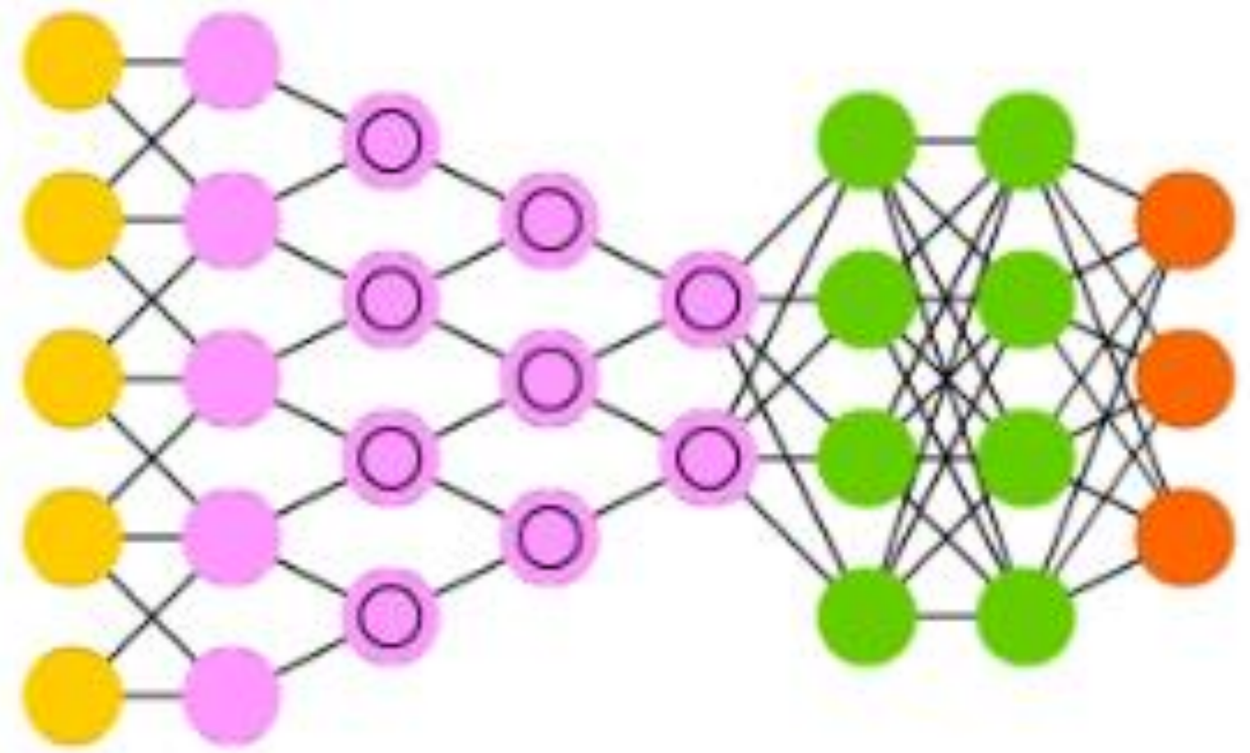
Examples [Regression and SVMs]



Deep/End-to-End Learning [2012 : now]

Simple Features → Complex Features → Model/ Mapping

Example [Conv Net]



Expert Systems - Limitations

What are these three images?

When to Choose Deep Learning

Classic Programming

If rules are clear and straightforward, often better to program it

Deep Learning

If rules are nuanced, complex, difficult to discern, use deep learning

Attention!!

1. Deep learning frameworks are libraries
2. Matrix orientation
3. Graph vs imperative
4. Continuous new architectures
5. Lack of solid reference

Careful!!

Matrix orientation:

$$\begin{pmatrix} 111 & 121 & 131 & 141 \\ 211 & 221 & 231 & 241 \\ 311 & 321 & 331 & 341 \end{pmatrix}$$

$$A(3x4x2) = \begin{pmatrix} 112 & 122 & 132 & 142 \\ 212 & 222 & 232 & 242 \\ 312 & 322 & 332 & 342 \end{pmatrix}$$

Fortran is matching the mathematical standard!

PyTorch, TensorFlow

$$\begin{pmatrix} 411 & 412 \\ 421 & 422 \\ 431 & 432 \end{pmatrix}$$

$$\begin{pmatrix} 311 & 312 \\ 321 & 322 \\ 331 & 332 \end{pmatrix}$$

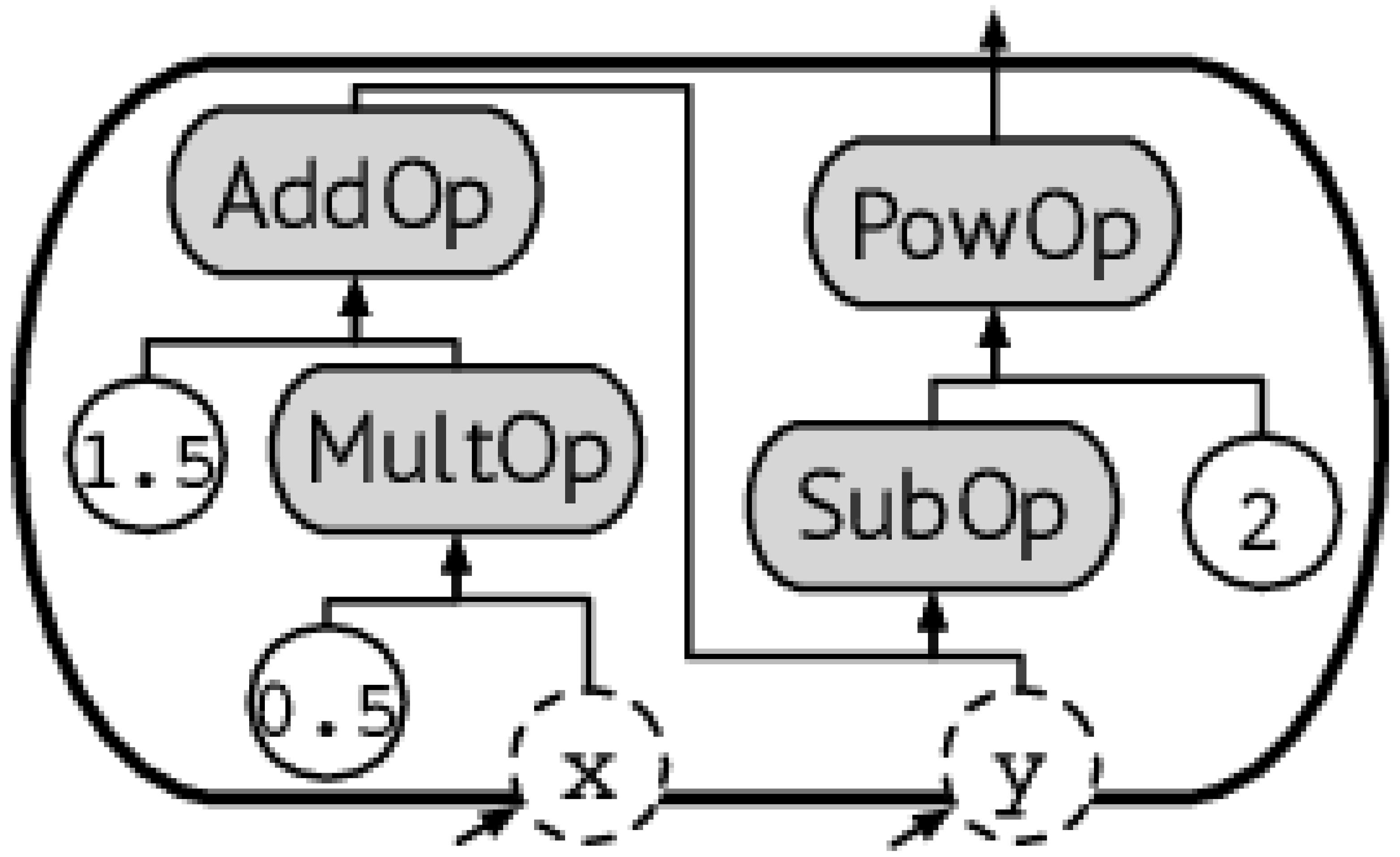
$$\begin{pmatrix} 211 & 212 \\ 221 & 222 \\ 231 & 232 \end{pmatrix}$$

$$A(4x3x2) = \begin{pmatrix} 111 & 112 \\ 121 & 122 \\ 131 & 132 \end{pmatrix}$$

Graph vs imperative program

```
def loss_fn(x, y):
    y_ = 0.5 * x + 1.5
    return (y_ - y) ** 2
```

(a) A source code snippet of an imperative DL program



(b) A symbolic DL graph generated from `loss_fn`

Easy to write, difficult to optimize (PyTorch)

Difficult to write, easy to optimize (TensorFlow)

Questions?