بسم الله الرحمن الرحيم

In the Name of Allah, the Most Gracious, the Most Merciful

قَالَ رَبِّ اشْرَحْ لِي صَدْرِي ﴿٢٥﴾

٢٥۔ (موسیٰ علیہ السلام نے) عرض کیا: اے میرے رب! میرے لئے میرا سینہ کشادہ فرما دے۔

وَيَسِّرْ لِي أَمْرِي ﴿٢٦﴾

٢٦۔ اور میرا کار (رسالت) میرے لئے آسان فرما دے۔

وَاحْلُلْ عُقْدَةً مِّن لِّسَانِي ﴿٢٧﴾

٢٧۔ اور میری زبان کی گرہ کھول دے۔

يَفْقَهُوا قَوْلِي ﴿٢٨﴾

٢٨۔ کہ لوگ میری بات (آسانی سے) سمجھ سکیں۔

Surah Taha with Urdu Translation

رَبِّ إِنِّي لِمَا أَنزَلْتَ إِلَيَّ مِنْ خَيْرٍ فَقِيرٌ

اے میرے رب! بے شک میں، جو بھلائی
بھی تو میری طرف نازل فرمائے، اس کا محتاج ہوں۔

سورۃ القصص: آیت نمبر 24

CS4152: Deep Learning and Neural Networks

Lecture 1 (Introduction to Deep Learning, ANN & Applications)

Dr. Jameel Ahmad

Associate Professor

Jameel.ahmad@umt.edu.pk

Department of Computer Science

School of Systems and Technology

University of Management and Technology, Lahore

Today's class

- A little about me
- A little about you
- Intro to Deep Learning and Neural Networks
- Course logistics
- Questions

About me—my Journey

B.Sc. in EE UET(1989-1992)

MSc in SE PIEAS(1994-1996)

MSc in EE USC(2002-2005)

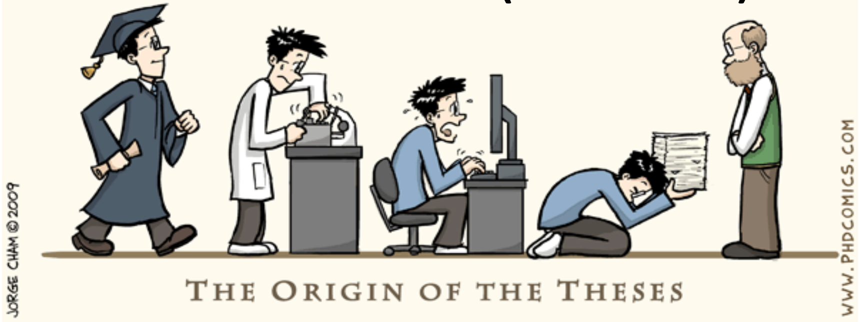
OSU
Oregon State
UNIVERSITY

RA in EE OSU(2005-2007)

Worked in Qualcomm(2007-2008)

Worked in Broadcom 2009-2010

PhD EE UET Lahore(2016-2019)



SEN (2010-2020)

SST(2020-2025)

About Instructor

- **PhD-Electrical Engineering**
  UET Lahore
- **MS-Electrical Engineering**
  University of Southern California (USC) Los Angeles USA
- **MS-Systems Engineering**
  Quadi Azam University Islamabad(PIEAS)
- **BS-Electrical Engineering**
  UET Peshawar
- **Total Publications (30) – Journals (21) & Conferences (9)**
- **Experience: Associate Professor at UMT**
- **Worked in QUALCOMM/ Broadcom USA/ PAEC/NESCOM**
- **Director Final Year Projects - EE.CS Department Since 2014-2018 and 2022-2025**
- **Focal Person NGIRI- FYP Funding for Students**
- **Research Interests**
  Machine Learning
  Image Processing and Computer Vision with Deep Learning
  Quantum Computing, Parallel and Distributed Computing, Cybersecurity
- Google Scholar
  (https://scholar.google.com/citations?hl=en&user=PbLAhoAAAAJ&view_op=list_works&sortby=pubdate)&user=PbL-AhoAAAAJ)

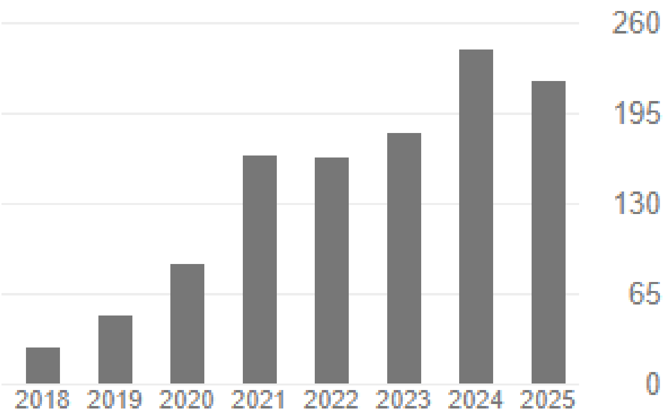
Dr. Jameel Ahmad

University of Management and Technology
(www.umt.edu.pk)

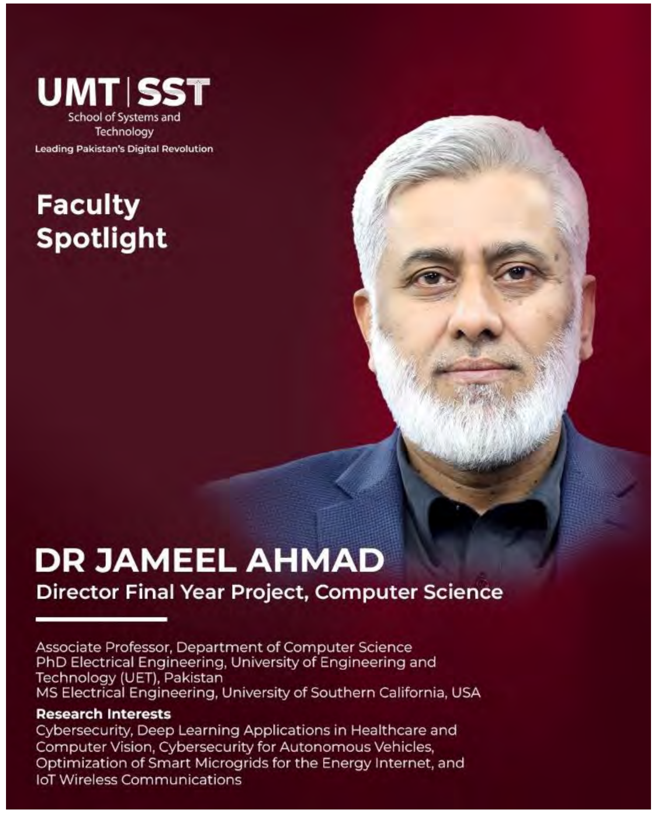
Deep Learning
Control and Optimization
Wireless Communication
Smart Grid

Cited by

	All	Since 2020
Citations	1152	1058
h-index	12	12
i10-index	14	14



Different Phases of My Life—once upon a time



Courses Taught at SST

- Machine Learning
- Computer Vision
- Deep Learning and Neural Networks
- Digital Image Processing
- Parallel and Distributed Computing
- Cloud Computing
- Programming Fundamentals
- Lot of other courses at UMT……..

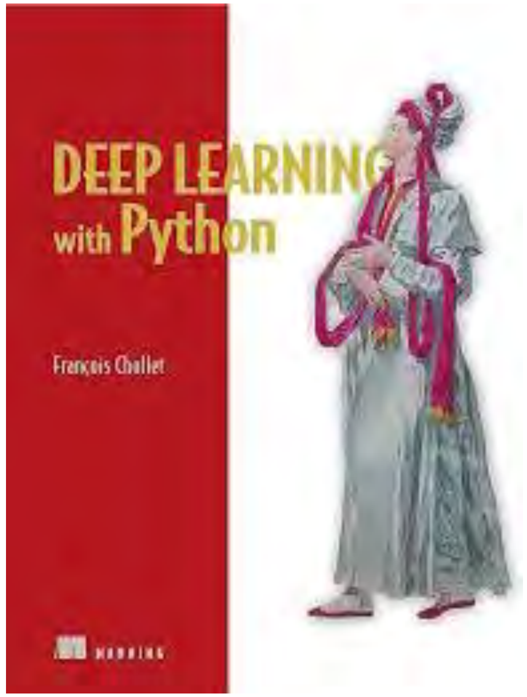
Welcome!

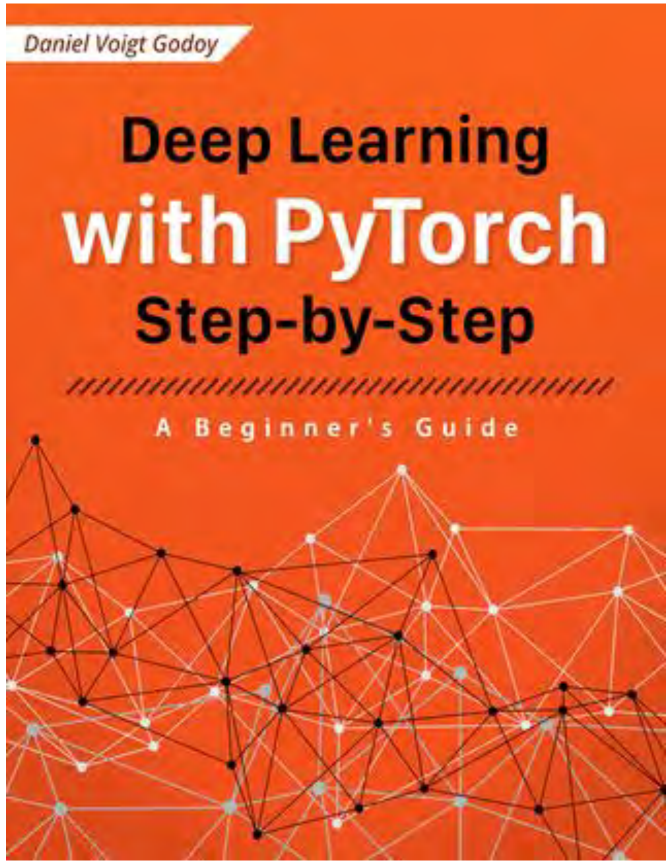
- Introductions to Course
- Administrative Matters
- Course Outline
- Applications of Deep Learning/ANN

About You …

What do you know already?

- Python Programming
- Have you taken Machine Learning Course
- C/C++ (Visual C++)
- Digital Image Processing, CV , OpenCV
    http://sourceforge.net/projects/opencvlibrary/
- Google CoLab
- GPU
- LLM
- How ChatGPT Works
- How DeepSeek works
- GAN





Administrative Information

- **Office**

  Level – 402, PhD Faculty Hall (SDT Building 1) , Cubicle-3 (On the right side)

- **Email**

  Jameel.ahmad@umt.edu.pk

- **Counselling Hours (BS-Students)**

  Every Tuesday, Wednesday, Friday (11:00 – 12:30)

- **Prior appointment is preferred before coming to the OFFICE**
- **Appointment via Email**
- **Moodle Website:** ***https://lms.umt.edu.pk/***
- **Class Schedule:** V3 – Tuesday and Friday (2pm-3:15pm)

Grading

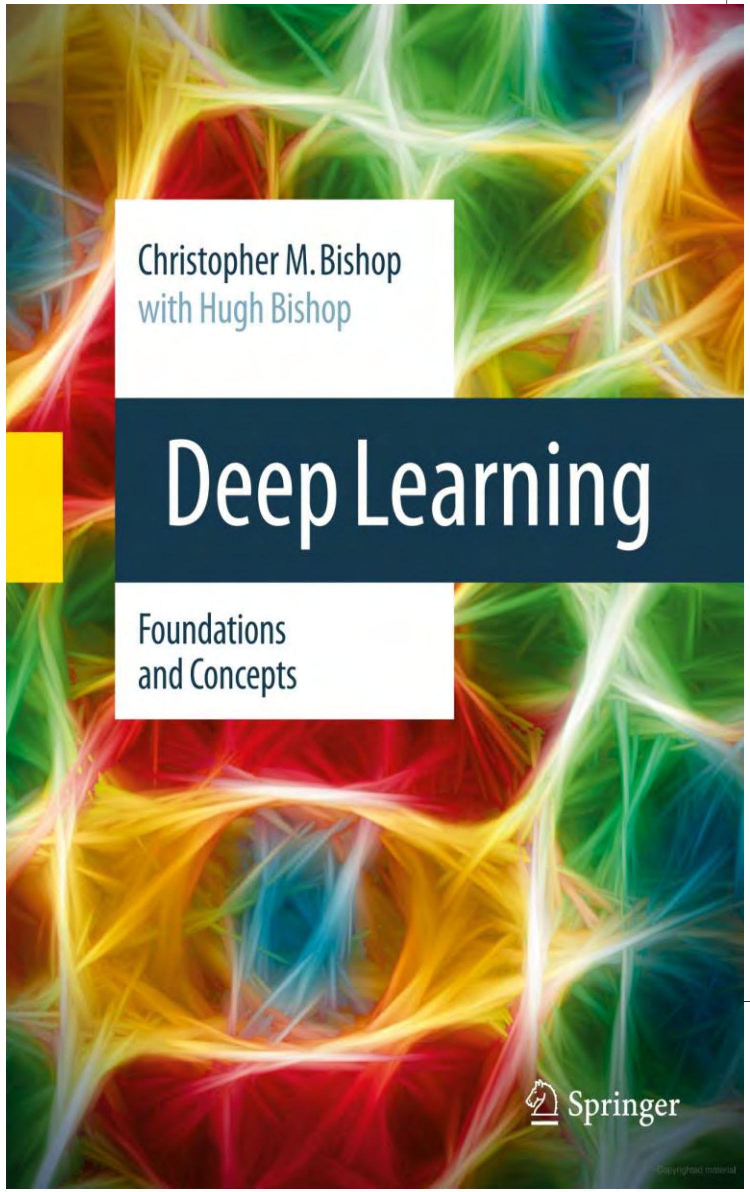
- The final grade depends on:
  - Homework, Quizzes and programming assignments (class activities): 25%
  - Exams: 25%(Midterm)+35(Final)=60%
  - Final project (may include class presentation): 15% (CCP+ Case Study)
  - Extra: 1~5% (for creative ideas, paper submission based on this course, etc.)

Outline

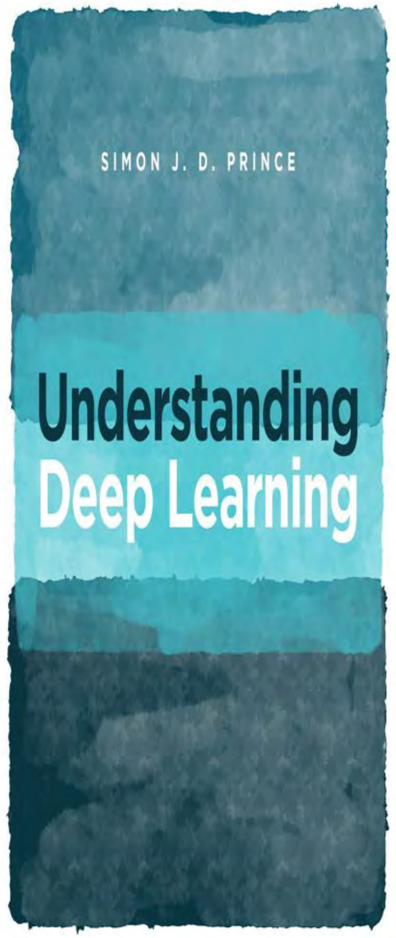
- Introductions
- Administrative Matters
- Course Outline

Books

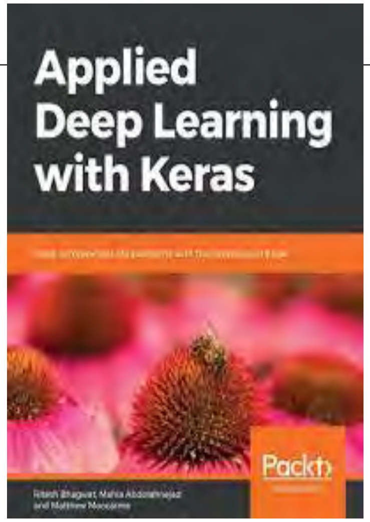
Deep Learning
Microsoft



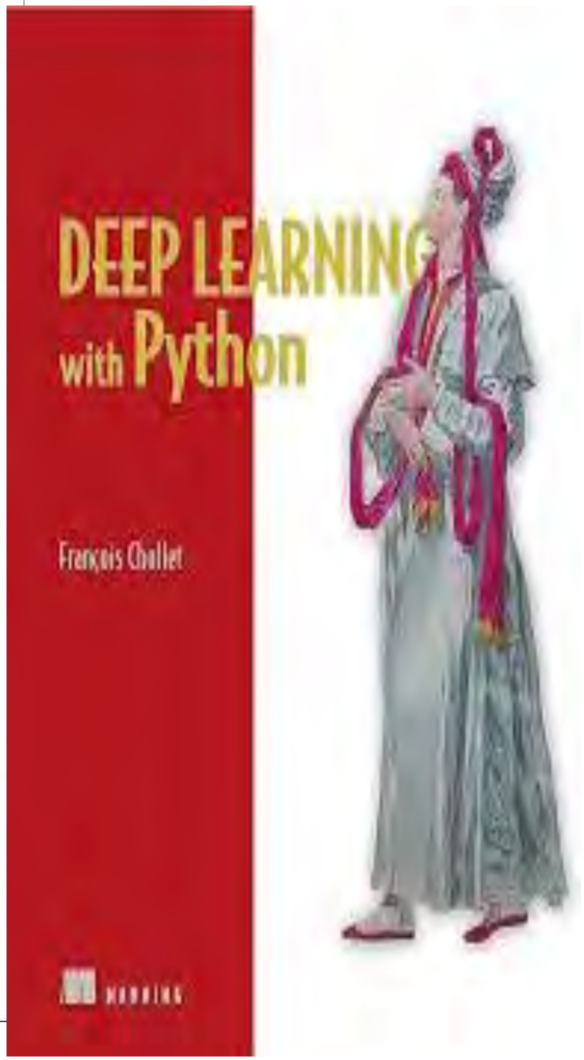
Christopher M. Bishop
with Hugh Bishop
Deep Learning
Foundations
and Concepts
Springer



SIMON J. D. PRINCE
Understanding
Deep Learning

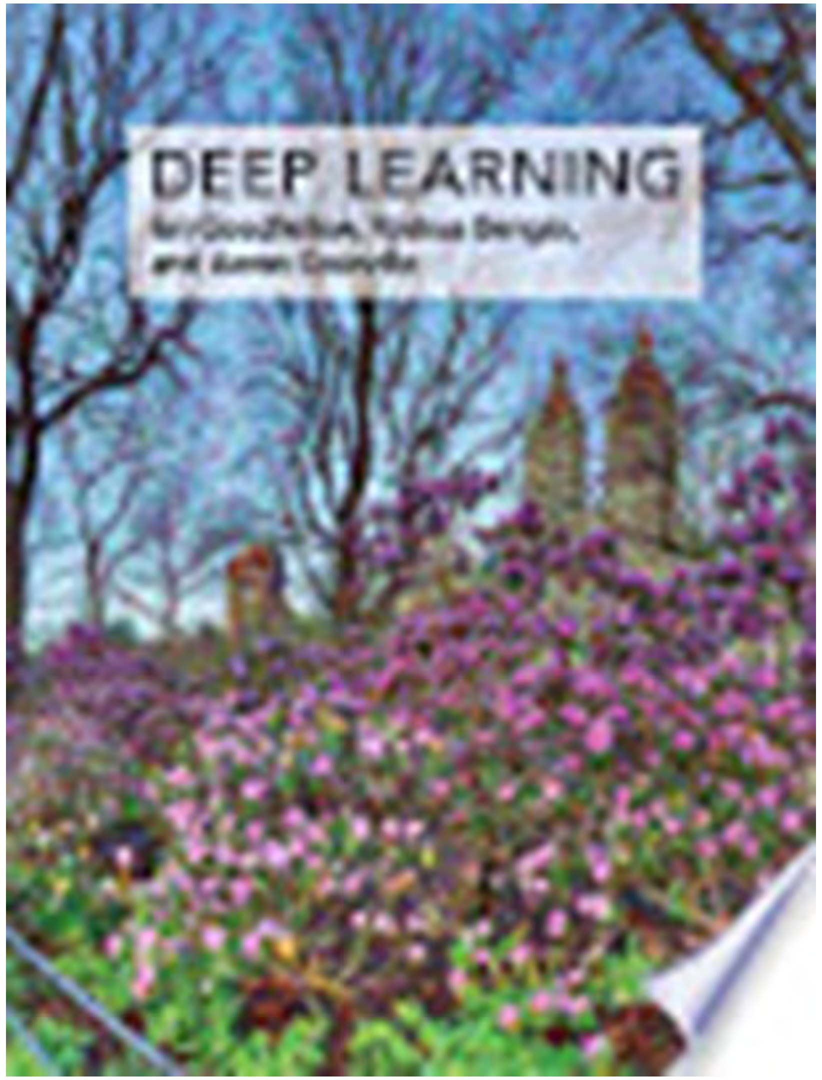


Applied
Deep Learning
with Keras
Packt

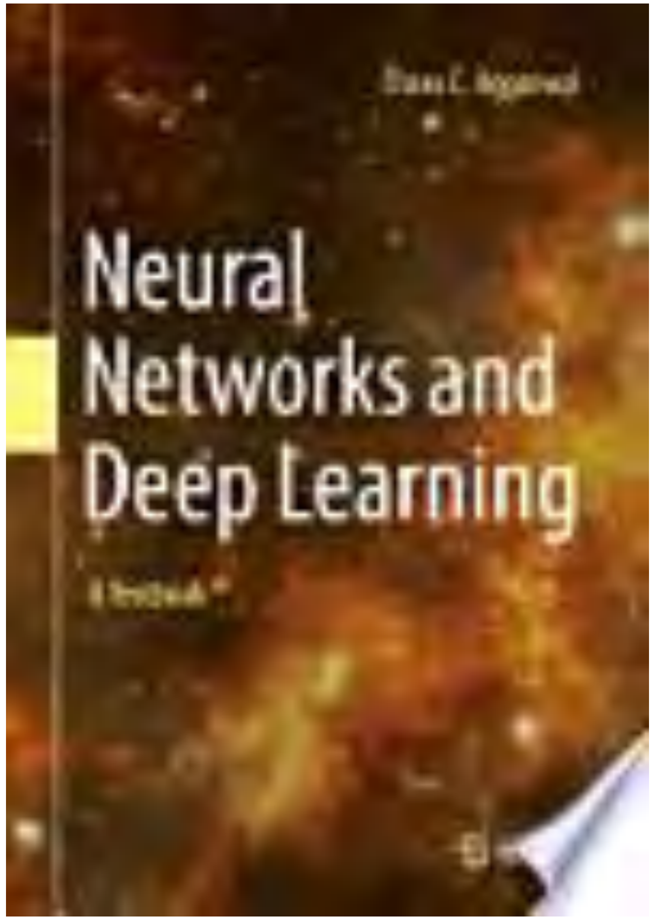


DEEP LEARNING
with Python
François Chollet
MANNING

Classic Textbooks

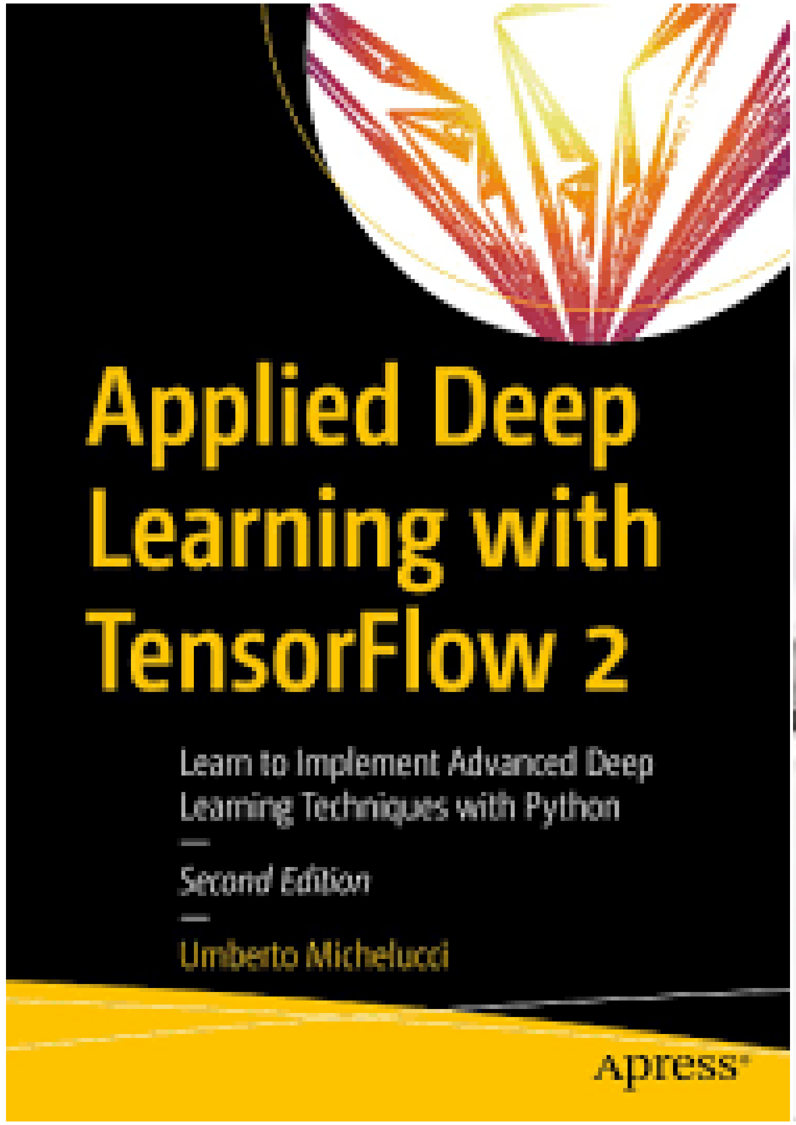


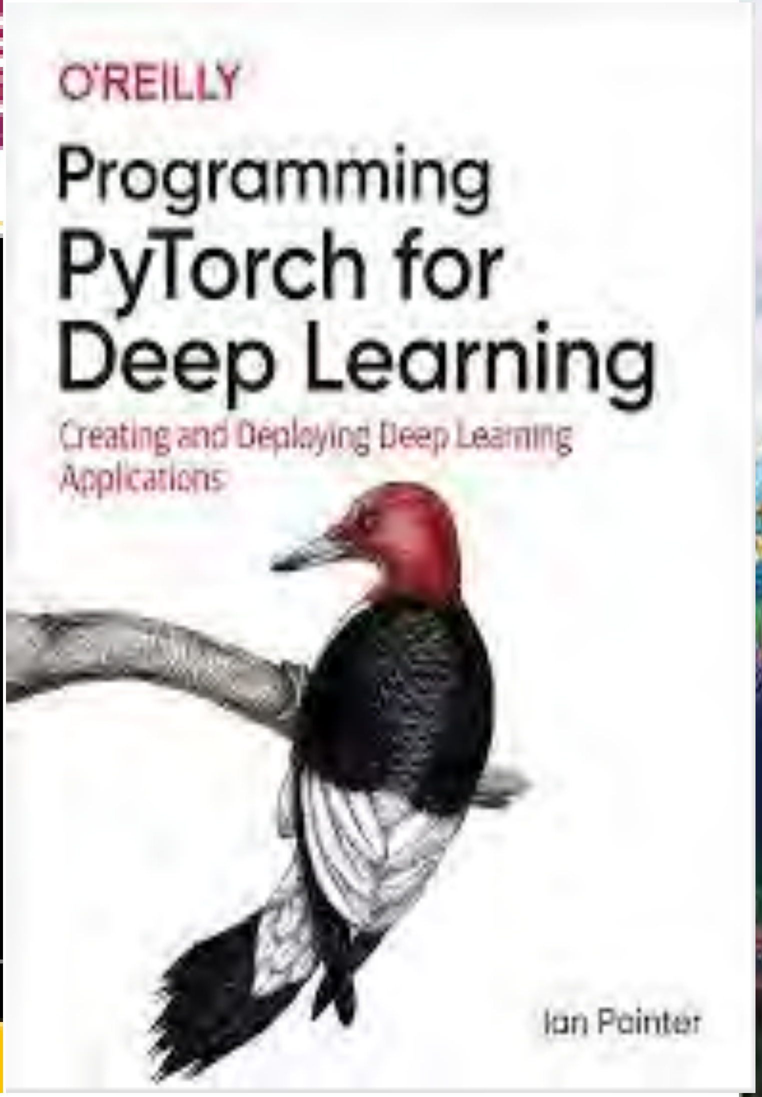
- Deep Learning
- By Ian Goodfellow, Yoshua Bengio, Aaron Courville · 2016

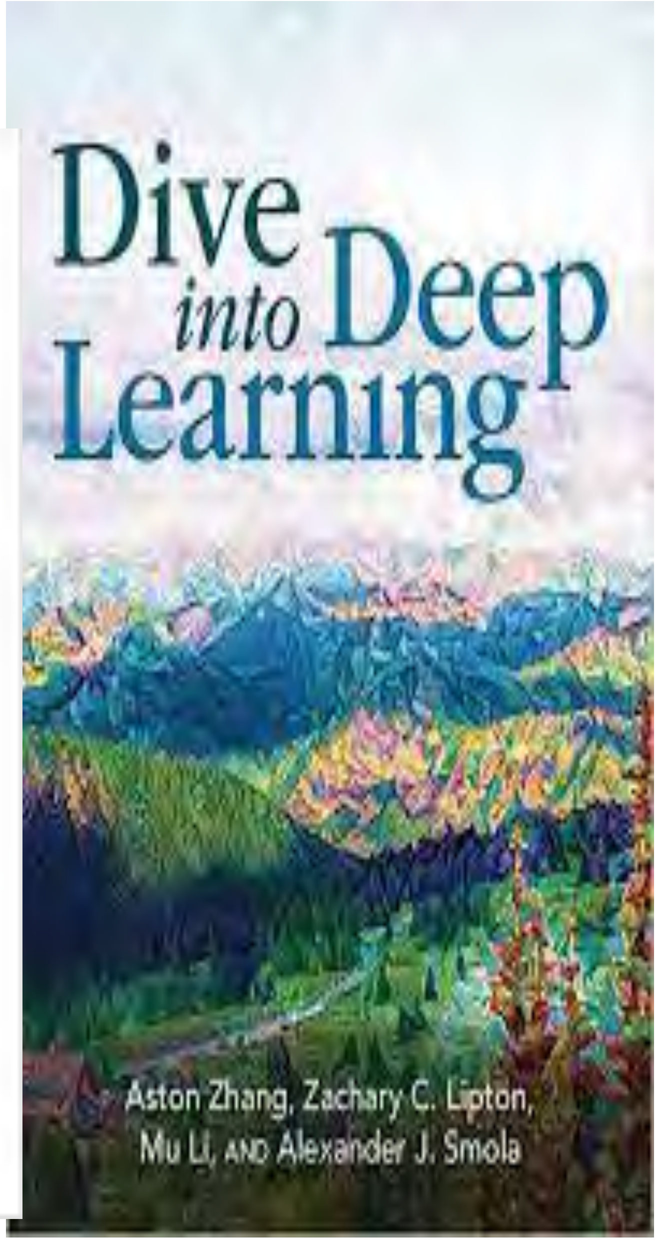


Neural Networks and Deep Learning: A Textbook - Charu C. Aggarwal · 2018

Reference Books







Classroom Etiquettes

- Bring your own Notebook/Register to make notes for yourself.
- Slides will be shared on **LMS** after **LECTURE**. You can download it from LMS.
- I encourage class participation
- Mobile phones – Silent or switch off
- **80% Attendance** (No Relaxation)
- Arrive on time in class (If you come late, missed topics will not be repeated for you).
- If you do not understand a point, raise your hand and ask me to explain or contact during office hours.
- Utilize **counselling hours**.
- Saturday & Sunday if **OFF**, so no one will be entertained on **OFF days**.
- No disturbance!!!! No Misconduct!!!! Professional conduct
- **REMEMBER:** Your first priority must be your studies

Look at the Syllabus

- Course Objectives
- Expected learning outcomes
- Detailed list of topics (maybe updated)

This course –Understanding Deep Learning (UDL)

- Chapter 1 - Introduction
- Chapter 2 - Supervised learning
- Chapter 3 - Shallow neural networks
- Chapter 4 - Deep neural networks
- Chapter 5 - Loss functions
- Chapter 6 - Training models
- Chapter 7 - Gradients and initialization
- Chapter 8 - Measuring performance
- Chapter 9 - Regularization
- Chapter 10 - Convolutional networks
- Chapter 11 - Residual networks
- Chapter 12 - Transformers
- Chapter 13 - Graph neural networks
- Chapter 14 - Unsupervised learning
- Chapter 15 - Generative adversarial networks
- Chapter 16 - Normalizing flows
- Chapter 17 - Variational autoencoders
- Chapter 18 - Diffusion models
- Chapter 19 - Deep reinforcement learning
- Chapter 20 - Why does deep learning work?
- Chapter 21 - Deep learning and ethics

Deep neural networks
How to train them
How to measure their performance
How to make that performance better

Academic Honesty, Integrity and Using AI tools

- The aim of this course is not merely to write Python code for deep learning, but to develop a deep understanding of the underlying mathematics, logic, algorithms, techniques, and architectures that drive it. When you rely on AI tools to generate your code or complete assignments, you bypass the fundamental learning objective—mastering the process through independent reasoning and problem-solving. True expertise in Deep Learning emerges from the effort to confront and overcome implementation challenges yourself.
- You may use AI tools for clarification, such as exploring definitions or gaining insight into concepts, but not for producing code or completing assigned tasks. Think of AI as a power tool: valuable for learning support, but counterproductive if it performs the core work on your behalf. Overdependence on it deprives you of the critical understanding of how and why algorithms and architectures function, ultimately hindering your analytical growth and problem-solving abilities. Always acknowledge any AI assistance used, and remember that the authenticity and integrity of your work remain your personal responsibility.

Lecture 1-2

Deep Learning and Neural Networks
An Overview
Jameel Ahmad, PhD

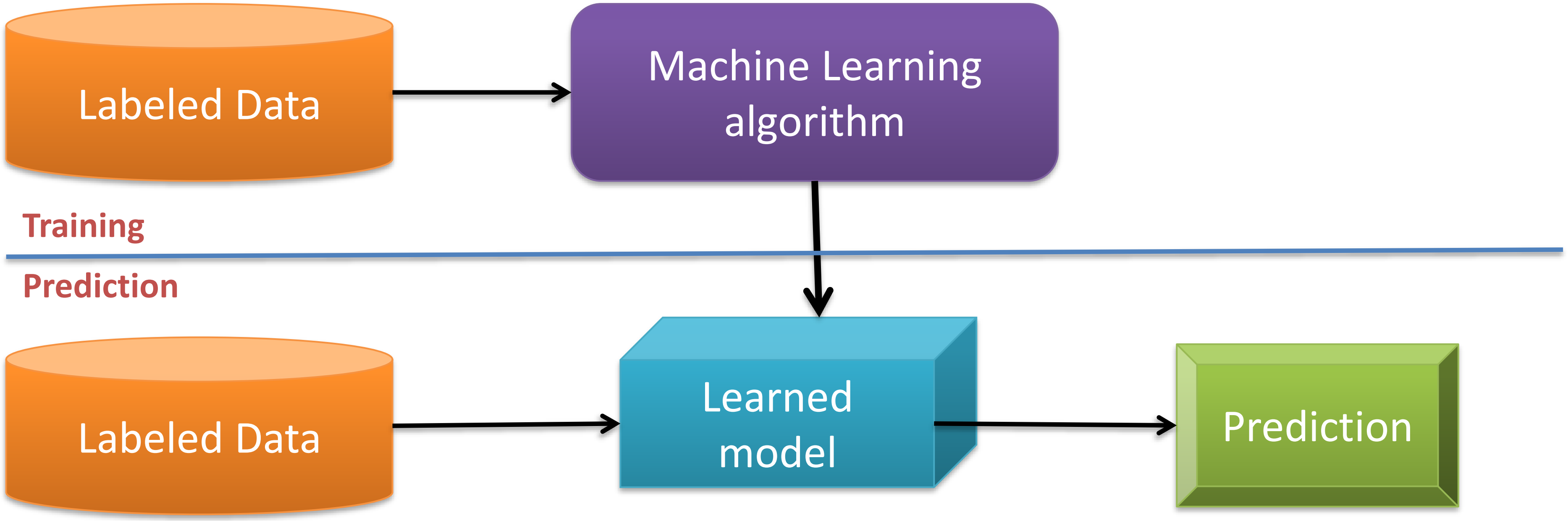
Lecture Outline

- Machine learning basics
  - Supervised and unsupervised learning
  - Linear and non-linear classification methods
- Introduction to deep learning
- Elements of neural networks (NNs)
  - Activation functions
- Training NNs
  - Gradient descent
  - Regularization methods
- NN architectures
  - Convolutional NNs
  - Recurrent NNs

Machine Learning Basics

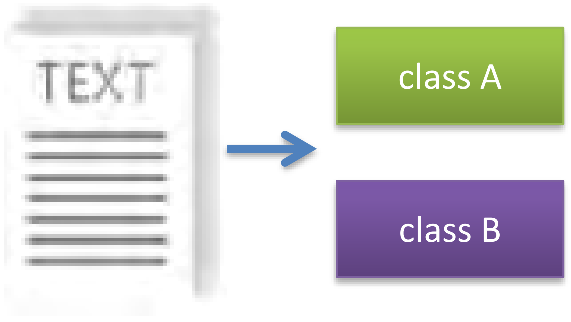
Machine Learning Basics

- ***Artificial Intelligence*** is a scientific field concerned with the development of algorithms that allow computers to learn without being explicitly programmed
- ***Machine Learning*** is a branch of Artificial Intelligence, which focuses on methods that learn from data and make predictions on unseen data

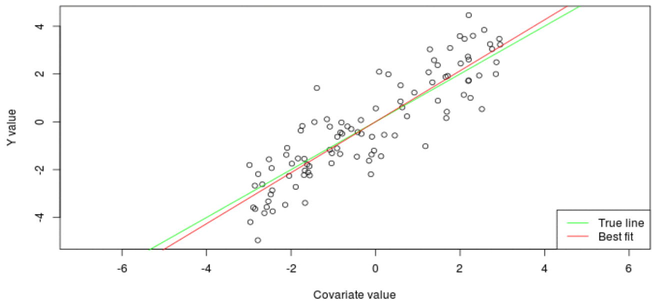


Machine Learning Types

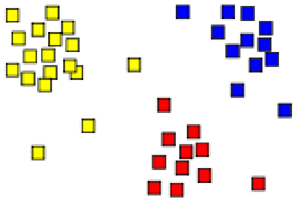
- ***Supervised***: learning with labeled data
  - Example: email classification, image classification
  - Example: regression for predicting real-valued outputs
- ***Unsupervised***: discover patterns in unlabeled data
  - Example: cluster similar data points
- ***Reinforcement learning***: learn to act based on feedback/reward
  - Example: learn to play Go



Classification



Regression



Clustering

Supervised Learning

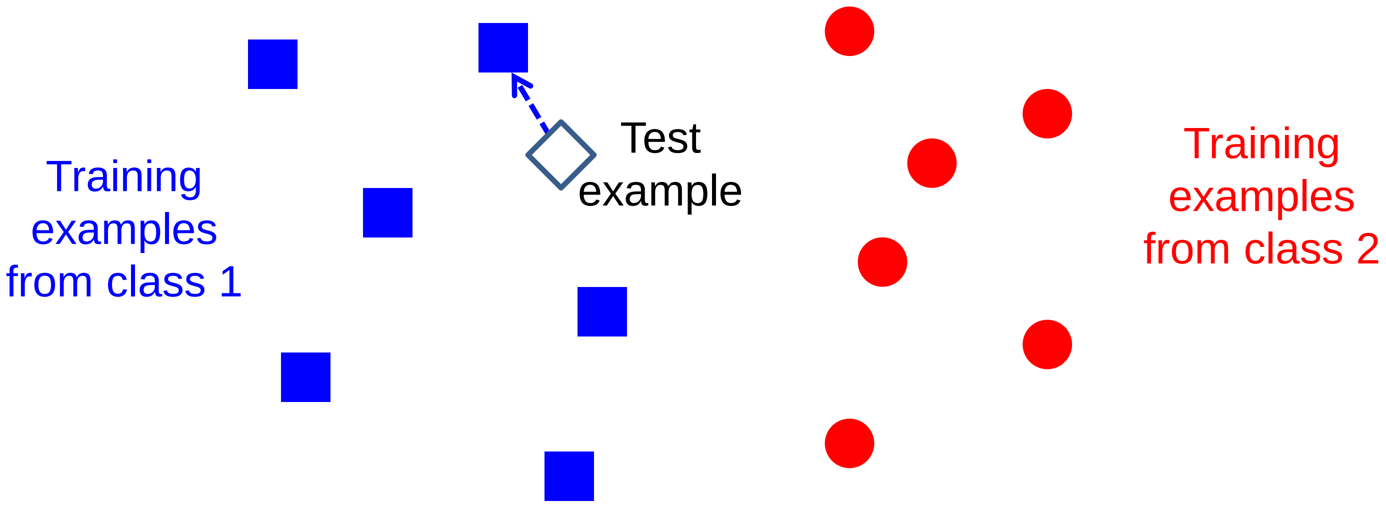
- *Supervised learning* categories and techniques
  - **Numerical classifier functions**
    - Linear classifier, perceptron, logistic regression, support vector machines (SVM), neural networks
  - **Parametric (probabilistic) functions**
    - Naïve Bayes, Gaussian discriminant analysis (GDA), hidden Markov models (HMM), probabilistic graphical models
  - **Non-parametric (instance-based) functions**
    - *k*-nearest neighbors, kernel regression, kernel density estimation, local regression
  - **Symbolic functions**
    - Decision trees, classification and regression trees (CART)
  - **Aggregation (ensemble) learning**
    - Bagging, boosting (Adaboost), random forest

Unsupervised Learning

- ***Unsupervised learning*** categories and techniques
  - **Clustering**
    - k-means clustering
    - Mean-shift clustering
    - Spectral clustering
  - **Density estimation**
    - Gaussian mixture model (GMM)
    - Graphical models
  - **Dimensionality reduction**
    - Principal component analysis (PCA)
    - Factor analysis

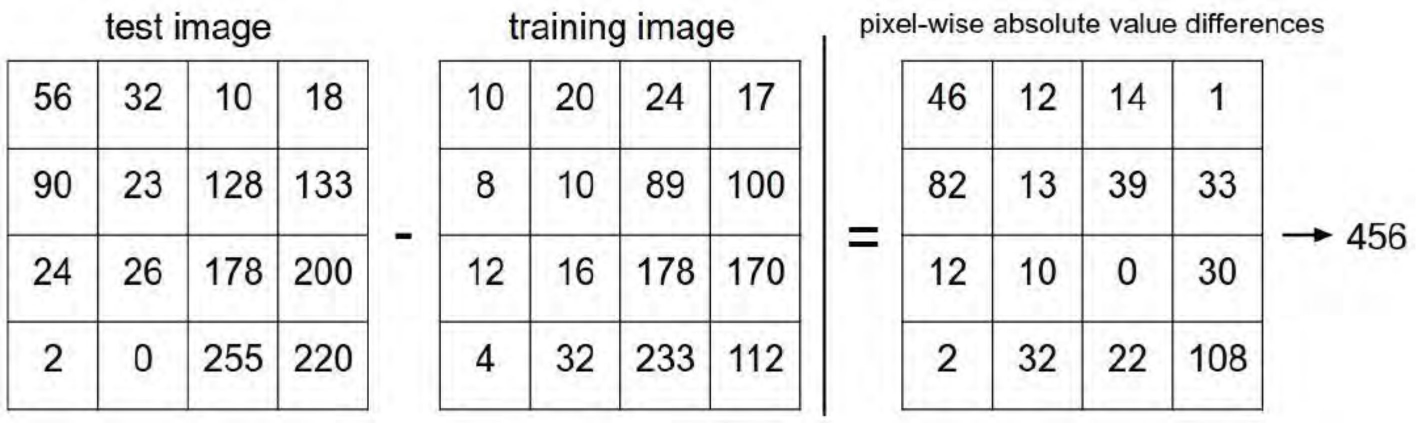
Nearest Neighbor Classifier

- ***Nearest Neighbor*** – for each test data point, assign the class label of the nearest training data point
  - Adopt a distance function to find the nearest neighbor
    - Calculate the distance to each data point in the training set, and assign the class of the nearest data point (minimum distance)
  - It does not require learning a set of weights



Nearest Neighbor Classifier

- For image classification, the distance between all pixels is calculated (e.g., using ℓ_1 norm, or ℓ_2 norm)
  - Accuracy on CIFAR-10: 38.6%
- Disadvantages:
  - The classifier must remember all training data and store it for future comparisons with the test data
  - Classifying a test image is expensive since it requires a comparison to all training images

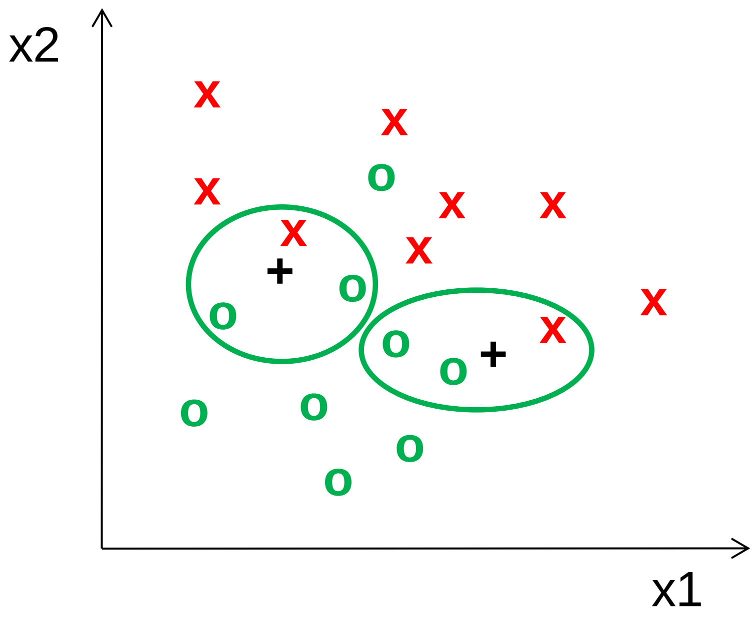


ℓ_1 norm
(Manhattan distance)

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

k-Nearest Neighbors Classifier

- ***k-Nearest Neighbors*** approach considers multiple neighboring data points to classify a test data point
  - E.g., 3-nearest neighbors
    - The test example in the figure is the + mark
    - The class of the test example is obtained by voting (based on the distance to the 3 closest points)



Linear Classifier

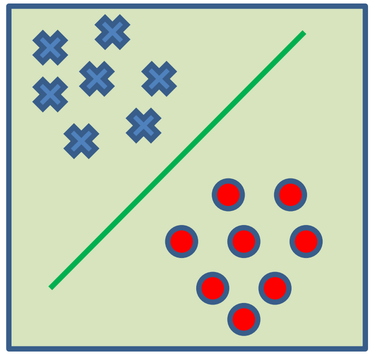
- ***Linear classifier***
  - Find a linear function f of the inputs x_i that separates the classes

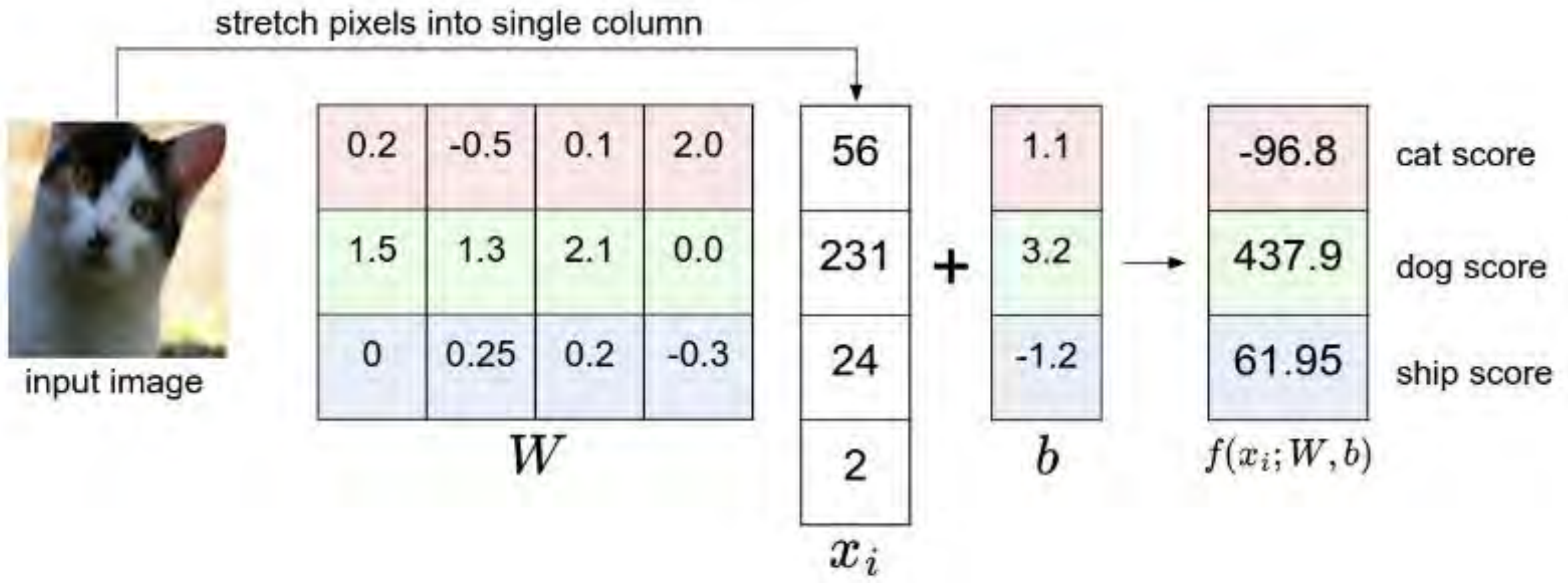
$$f(x_i, W, b) = Wx_i + b$$

  - Use pairs of inputs and labels to find the weights matrix W and the bias vector b
    - The weights and biases are the parameters of the function f
  - Several methods have been used to find the optimal set of parameters of a linear classifier
    - A common method of choice is the ***Perceptron*** algorithm, where the parameters are updated until a minimal error is reached (single layer, does not use backpropagation)
  - Linear classifier is a simple approach, but it is a building block of advanced classification algorithms, such as SVM and neural networks
    - Earlier multi-layer neural networks were referred to as multi-layer perceptrons (MLPs)

Linear Classifier

- The decision boundary is linear
  - A straight line in 2D, a flat plane in 3D, a hyperplane in 3D and higher dimensional space
- Example: classify an input image
  - The selected parameters in this example are not good, because the predicted cat score is low

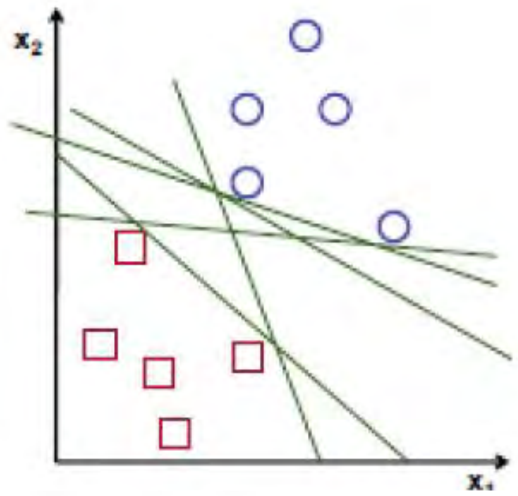


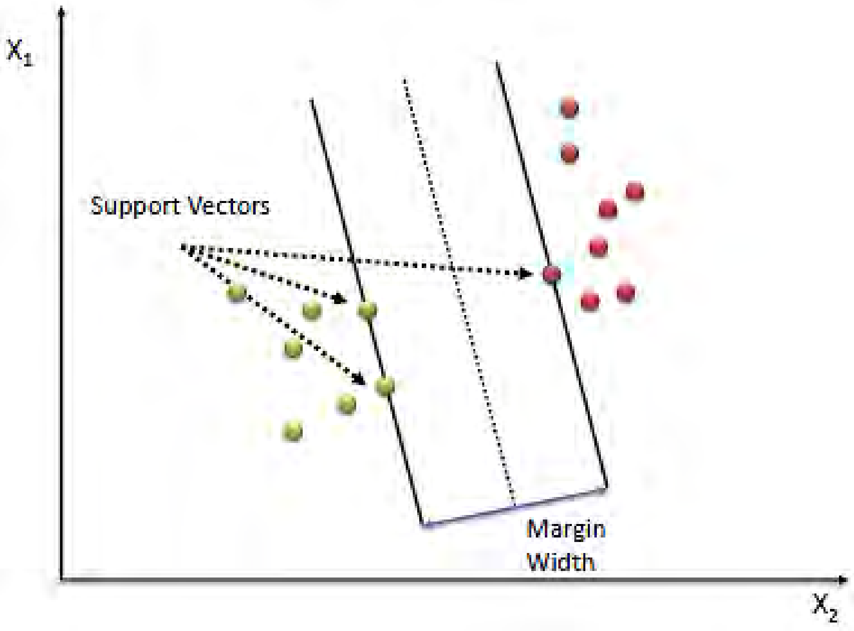


Support Vector Machines

- ***Support vector machines (SVM)***
  - How to find the best decision boundary?
    - All lines in the figure correctly separate the 2 classes
    - The line that is farthest from all training examples will have better generalization capabilities
  - SVM solves an optimization problem:
    - First, identify a decision boundary that correctly classifies the examples
    - Next, increase the geometric margin between the boundary and all examples
  - The data points that define the maximum margin width are called support vectors
  - Find W and b by solving:

$$\min \frac{1}{2}\|w\|^2$$
$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i$$





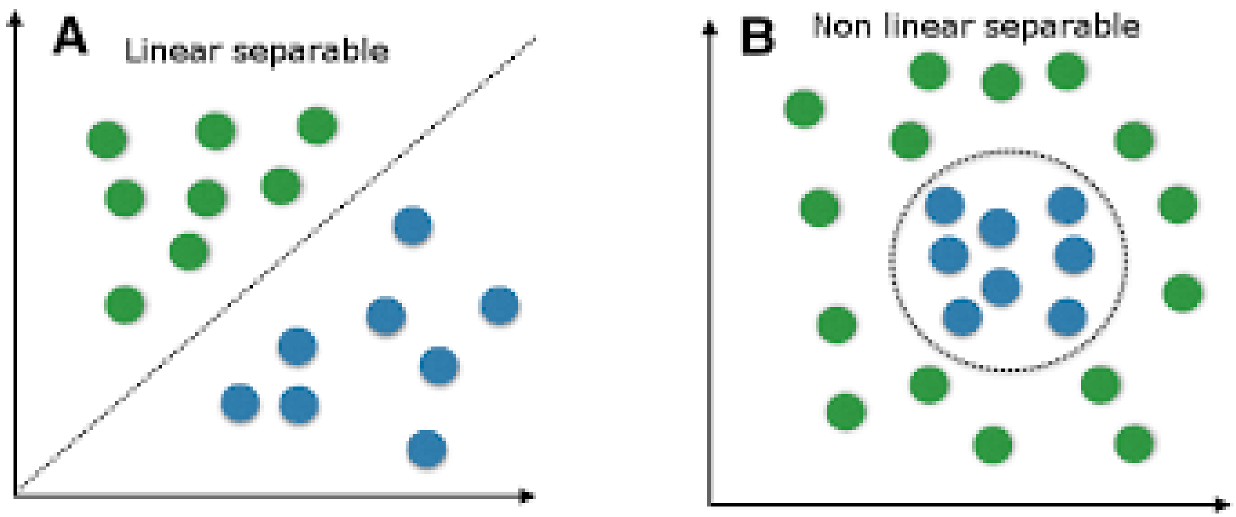
Linear vs Non-linear Techniques

- Linear classification techniques
  - Linear classifier
  - Perceptron
  - Logistic regression
  - Linear SVM
  - Naïve Bayes
- Non-linear classification techniques
  - *k*-nearest neighbors
  - Non-linear SVM
  - Neural networks
  - Decision trees
  - Random forest

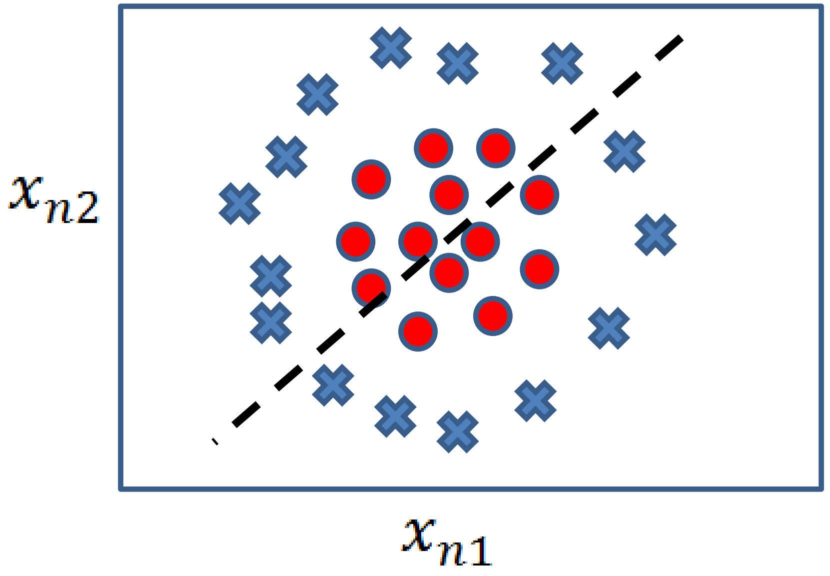
Linear vs Non-linear Techniques

Linear vs Non-linear Techniques

- For some tasks, input data can be linearly separable, and linear classifiers can be suitably applied



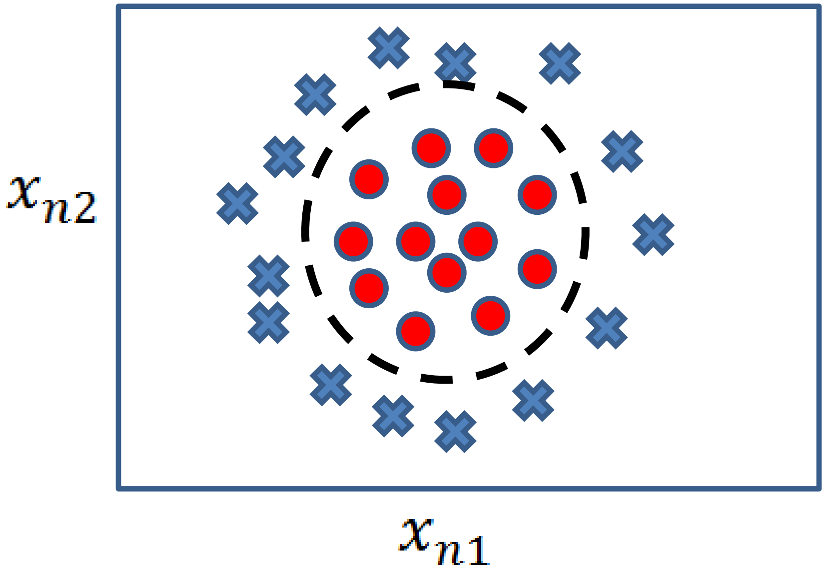
- For other tasks, linear classifiers may have difficulties to produce adequate decision boundaries



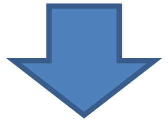
Non-linear Techniques

Linear vs Non-linear Techniques

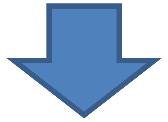
- Non-linear classification
  - Features z_i are obtained as non-linear functions of the inputs x_i
  - It results in non-linear decision boundaries
  - Can deal with non-linearly separable data



Inputs: $x_i = [x_{n1} \quad x_{n2}]$



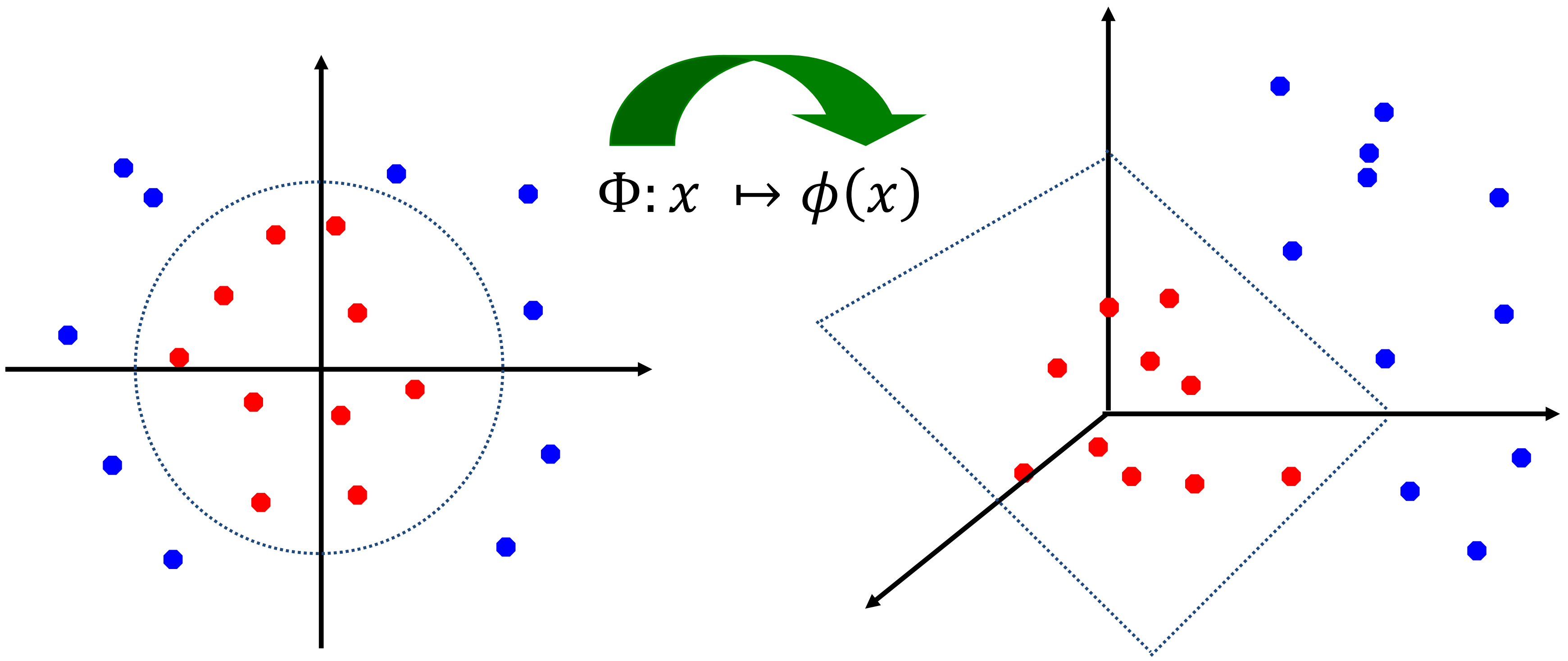
Features: $z_i = [x_{n1} \quad x_{n2} \quad x_{n1} \cdot x_{n2} \quad x_{n1}^2 \quad x_{n2}^2]$



Non-linear Support Vector Machines

Linear vs Non-linear Techniques

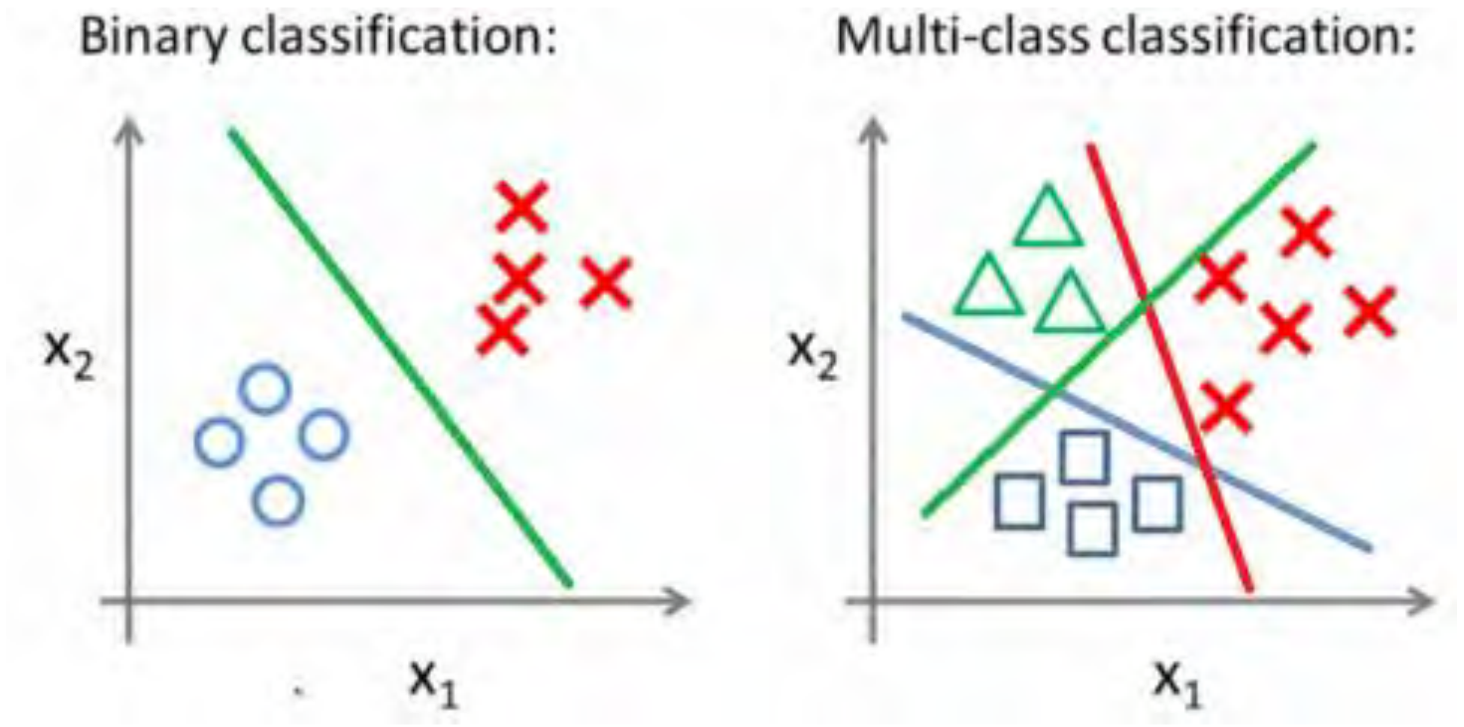
- ***Non-linear SVM***
  - The original input space is mapped to a higher-dimensional feature space where the training set is linearly separable
  - Define a non-linear kernel function to calculate a non-linear decision boundary in the original feature space



Binary vs Multi-class Classification

Binary vs Multi-class Classification

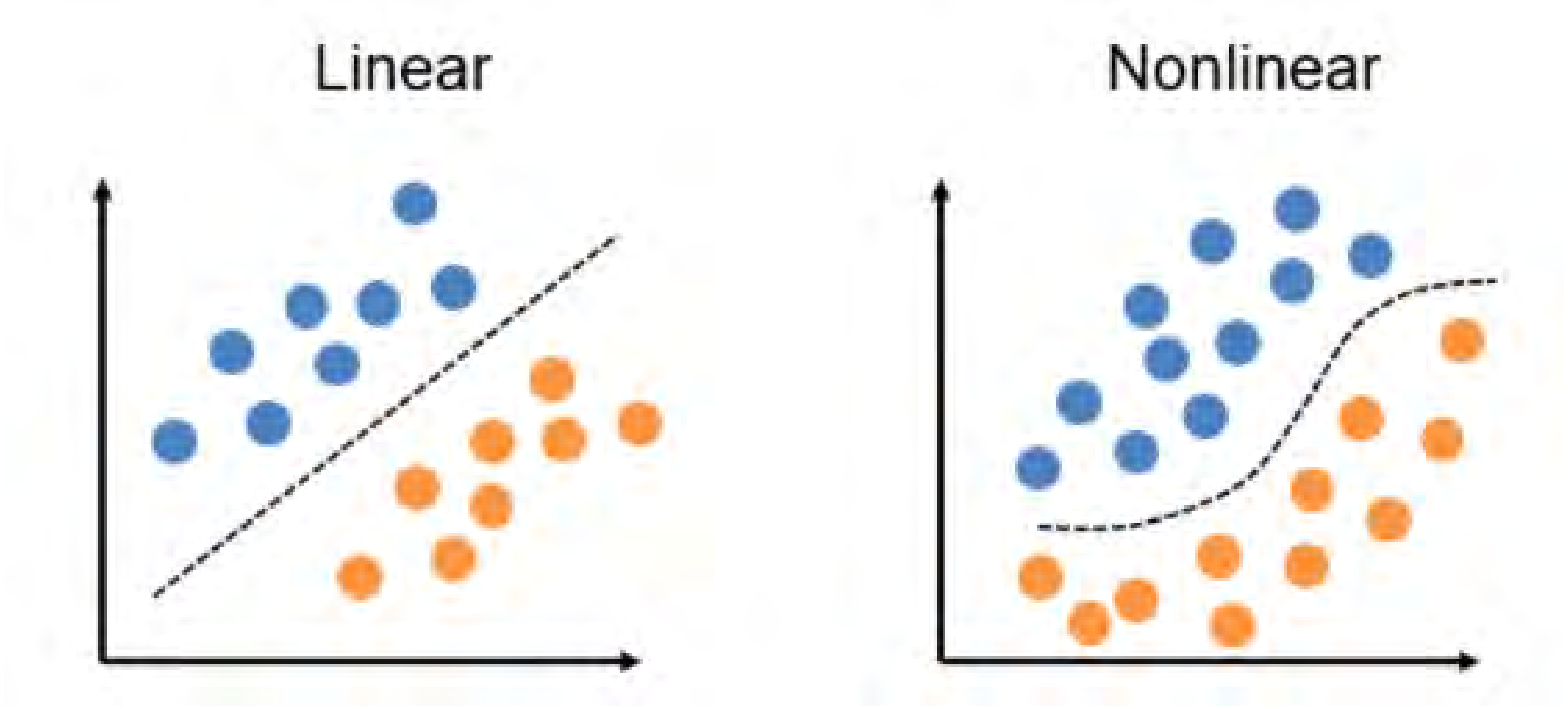
- A classification problem with only 2 classes is referred to as ***binary classification***
  - The output labels are 0 or 1
  - E.g., benign or malignant tumor, spam or no-spam email
- A problem with 3 or more classes is referred to as ***multi-class classification***



Binary vs Multi-class Classification

Binary vs Multi-class Classification

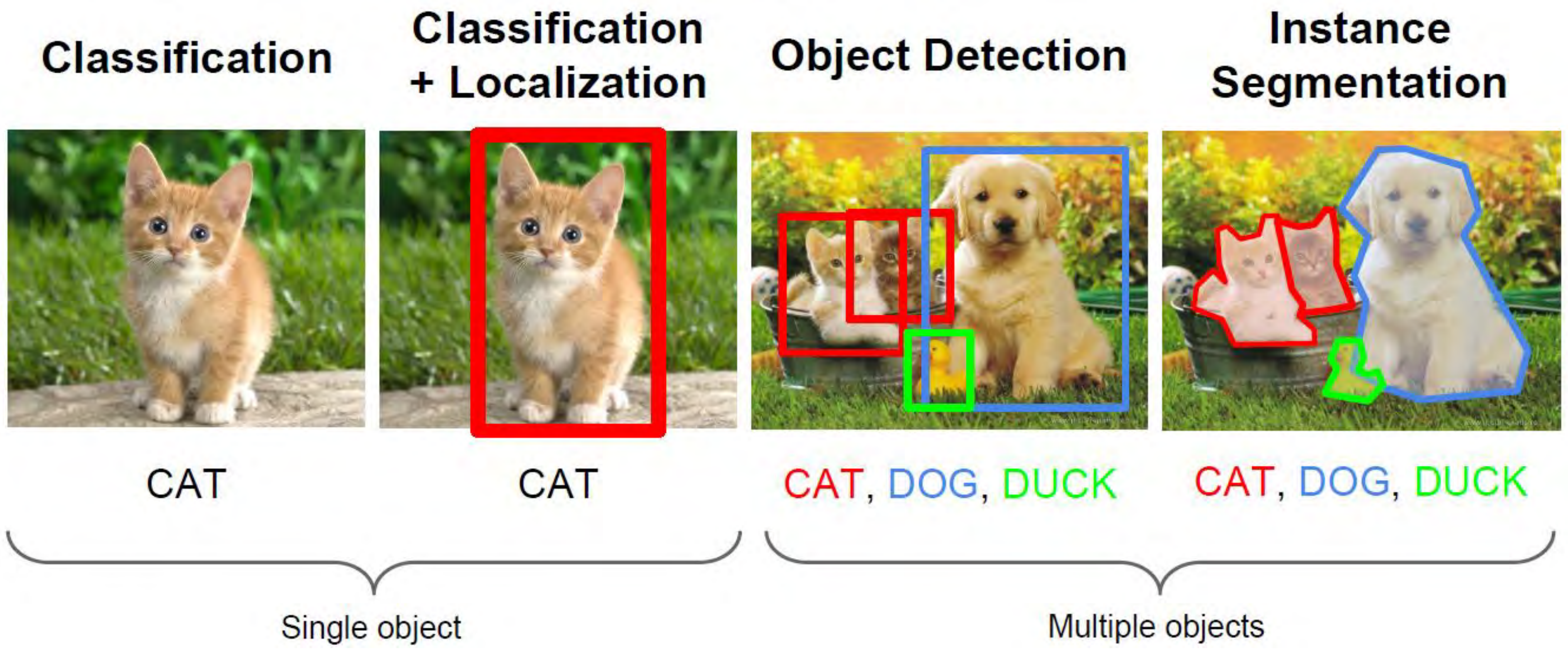
- Both the binary and multi-class classification problems can be linearly or non-linearly separated
  - Figure: linearly and non-linearly separated data for binary classification problem



Computer Vision Tasks

Machine Learning Basics

- Computer vision has been the primary area of interest for ML
- The tasks include: classification, localization, object detection, instance segmentation

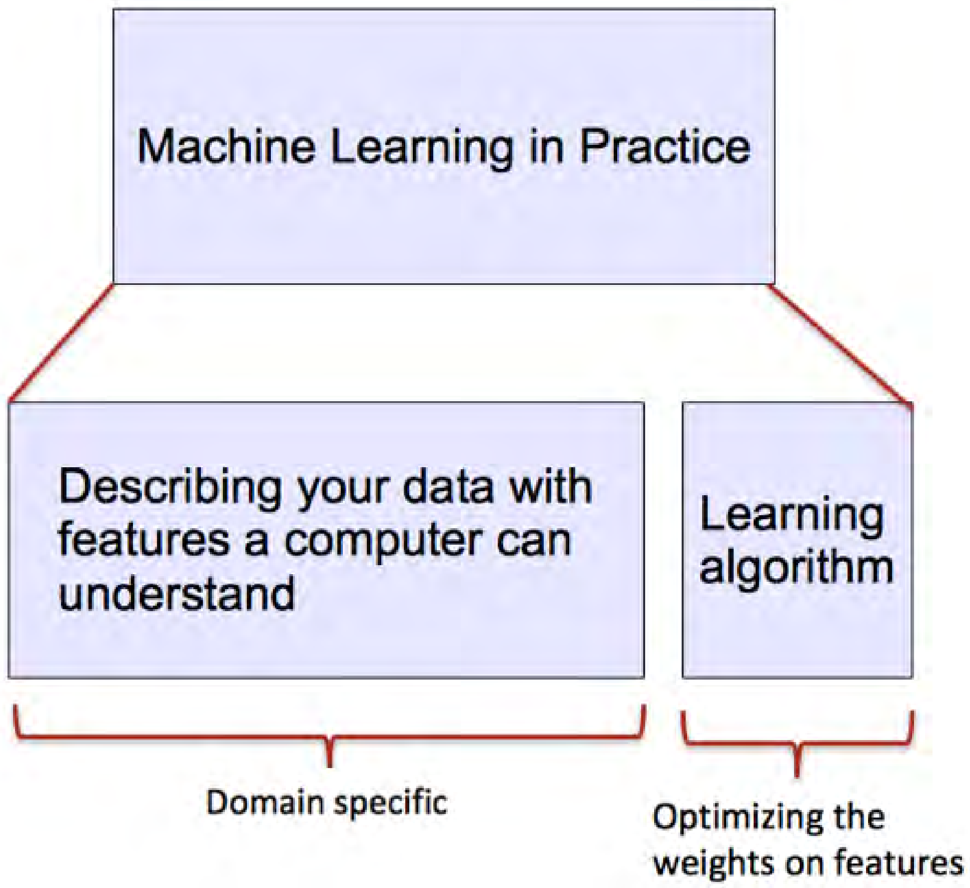


No-Free-Lunch Theorem

- Wolpert (2002) - The Supervised Learning No-Free-Lunch Theorems
- The derived classification models for supervised learning are simplifications of the reality
  - The simplifications are based on certain assumptions
  - The assumptions fail in some situations
    - E.g., due to inability to perfectly estimate ML model parameters from limited data
- In summary, ***No-Free-Lunch Theorem*** states:
  - **No single classifier works the best for all possible problems**
  - Since we need to make assumptions to generalize

ML vs. Deep Learning

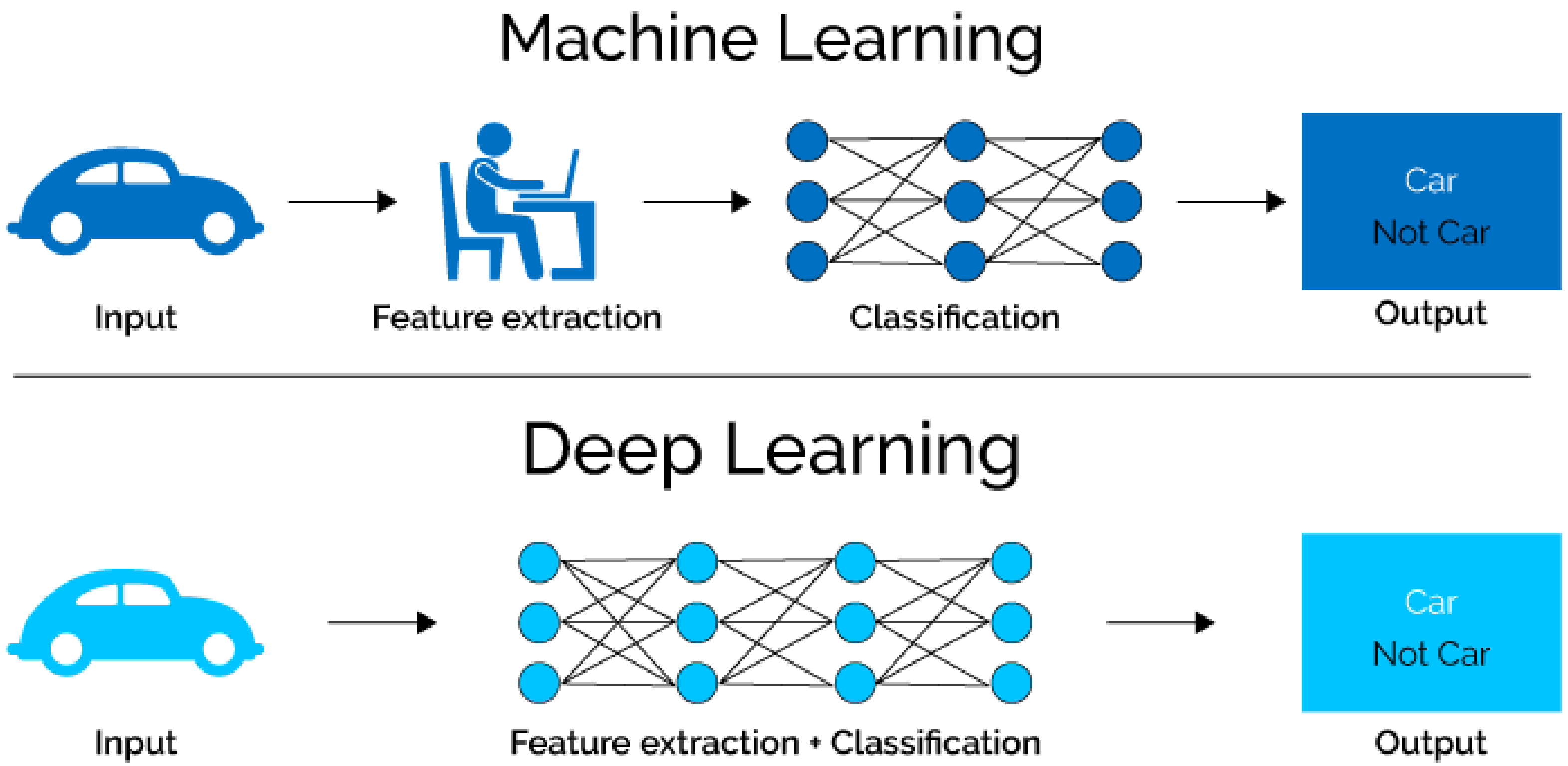
- Conventional machine learning methods rely on human-designed feature representations
  - ML becomes just optimizing weights to best make a final prediction



ML vs. Deep Learning

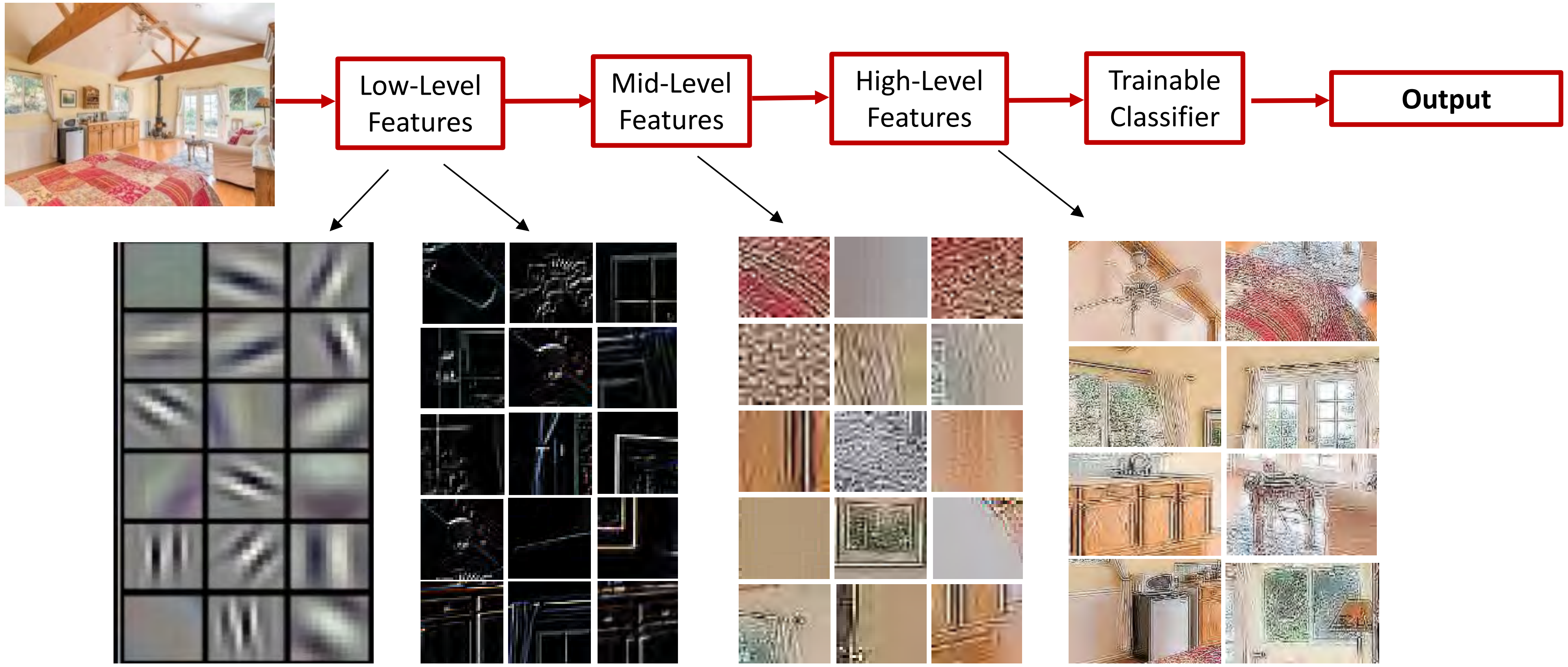
Introduction to Deep Learning

- ***Deep learning*** (DL) is a machine learning subfield that uses multiple layers for learning data representations
  - DL is exceptionally effective at learning patterns



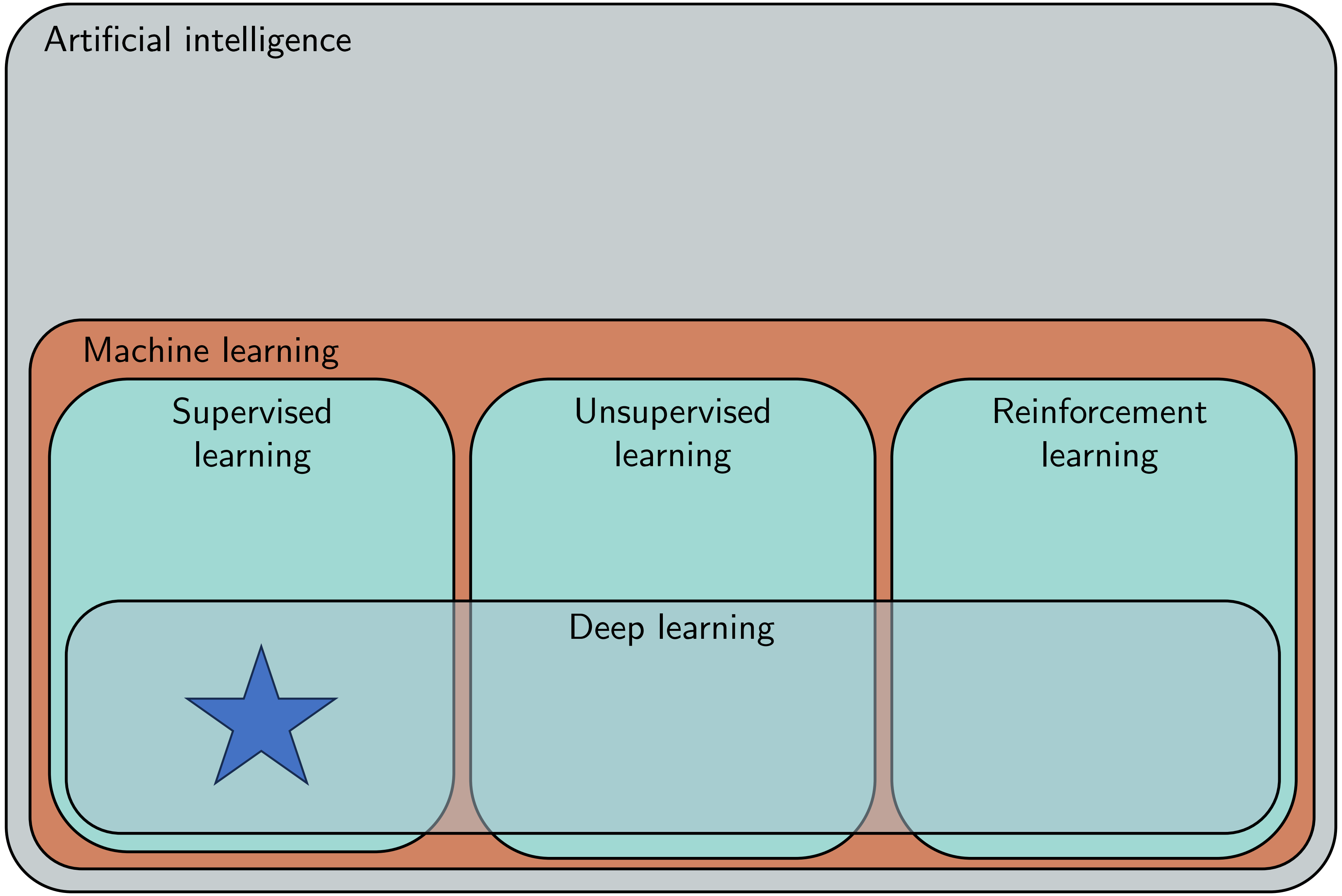
ML vs. Deep Learning

- DL applies a multi-layer process for learning rich hierarchical features (i.e., data representations)
  - Input image pixels → Edges → Textures → Parts → Objects



Why is DL Useful?

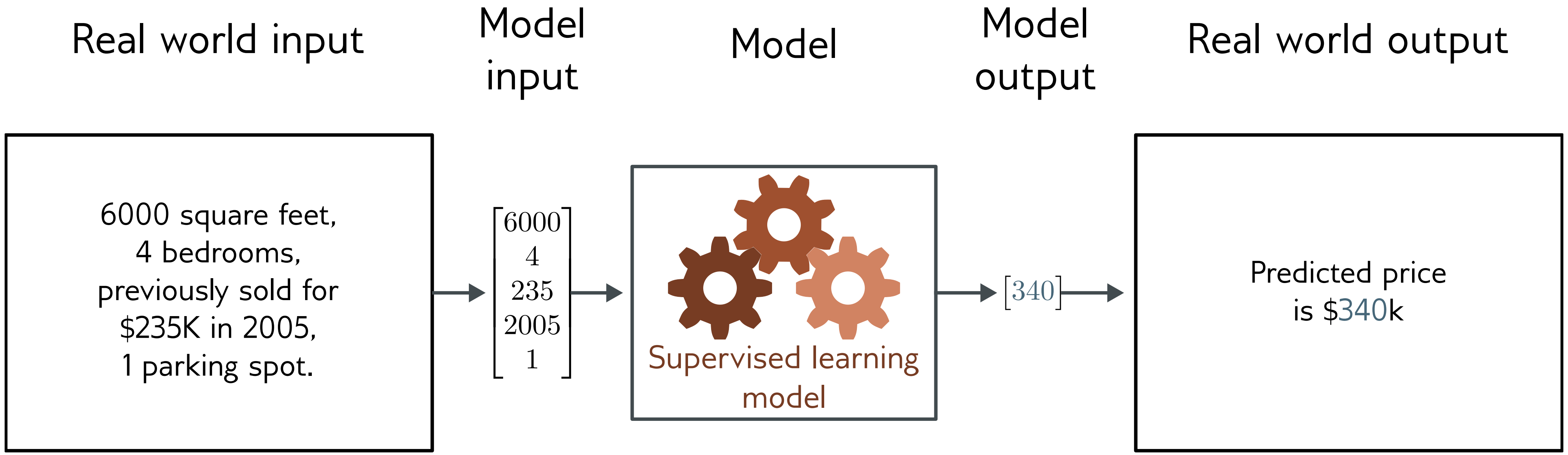
- DL provides a flexible, learnable framework for representing visual, text, linguistic information
  - Can learn in supervised and unsupervised manner
- DL represents an effective end-to-end learning system
- Requires large amounts of training data
- Since about 2010, DL has outperformed other ML techniques
  - First in vision and speech, then NLP, and other applications


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Supervised learning

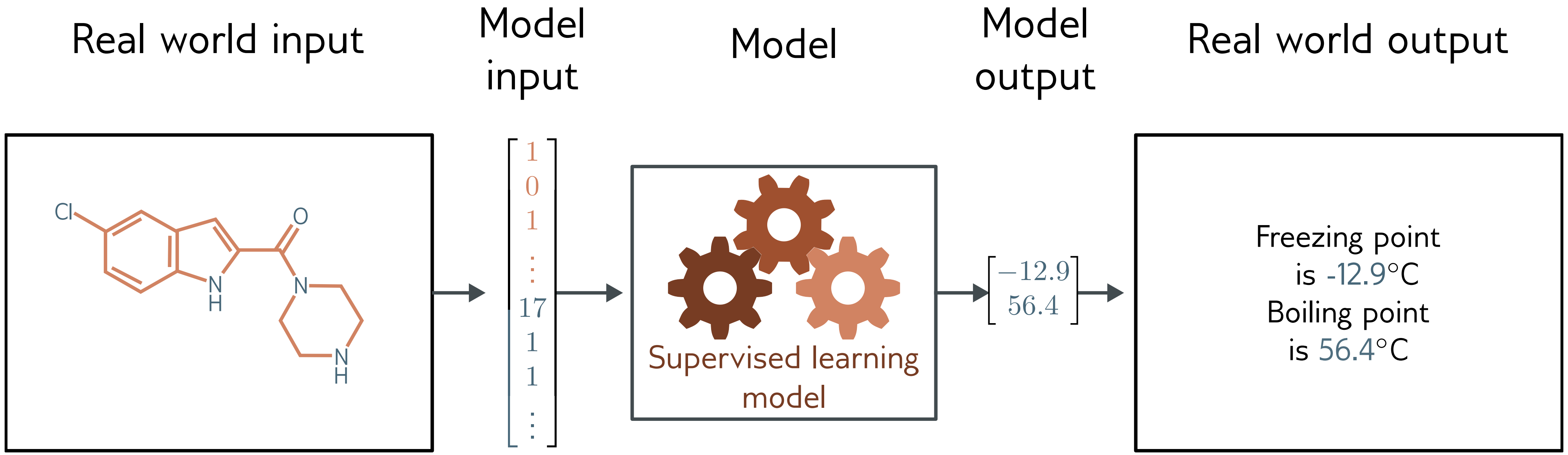
- Define a mapping from input to output
- Learn this mapping from paired input/output data examples

Regression



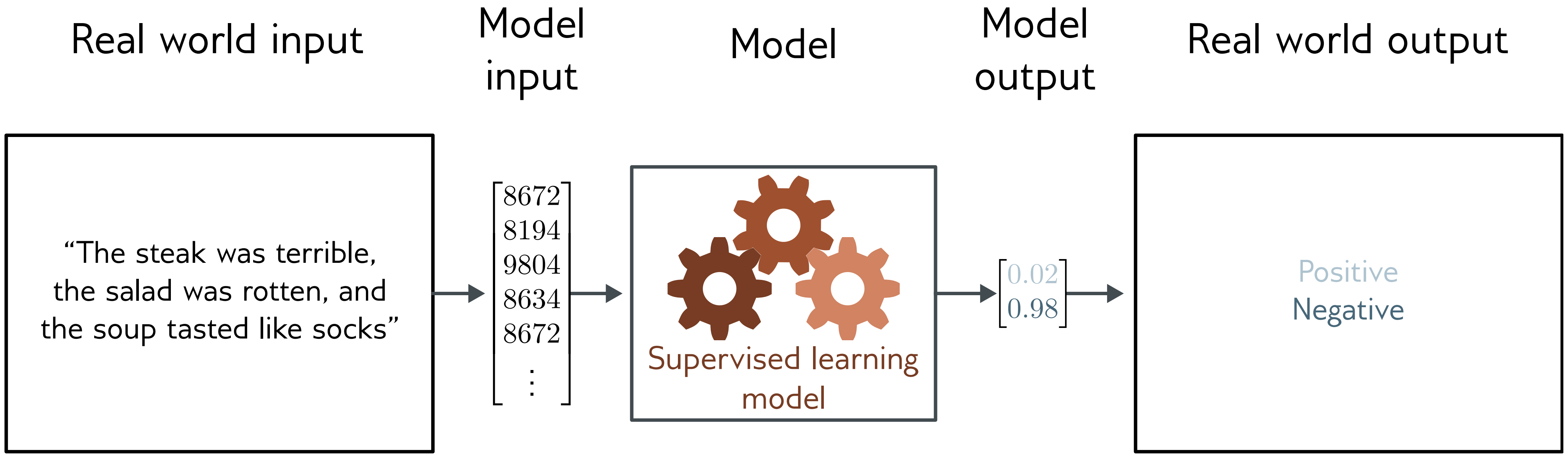
- Univariate regression problem (one output, real value)
- Fully connected network

Graph regression



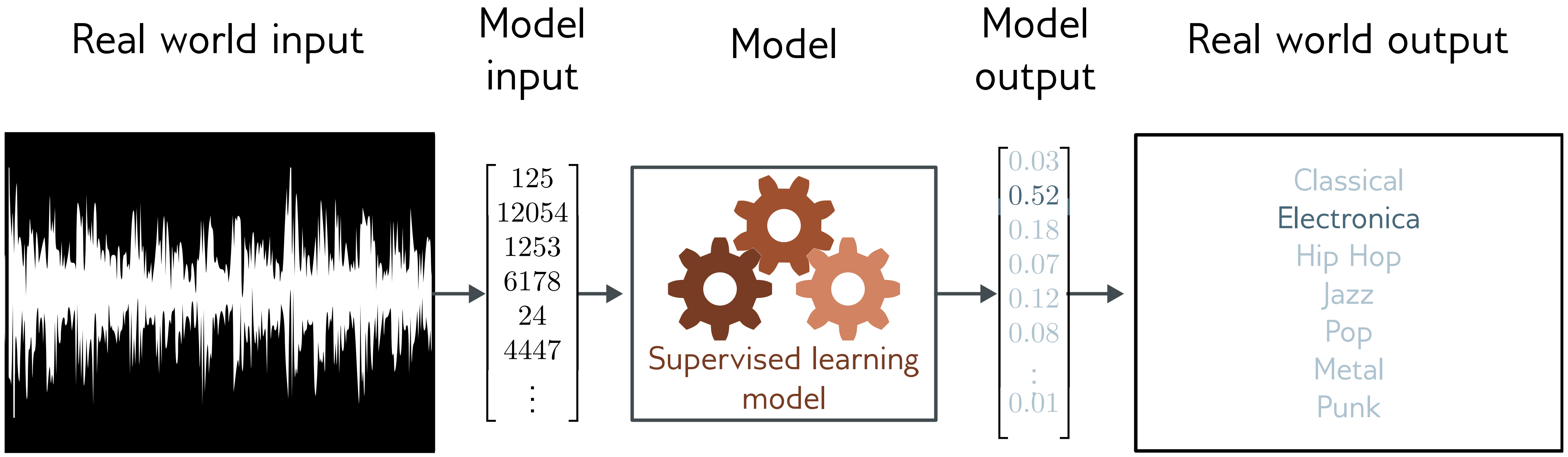
- Multivariate regression problem (>1 output, real value)
- Graph neural network

Text classification



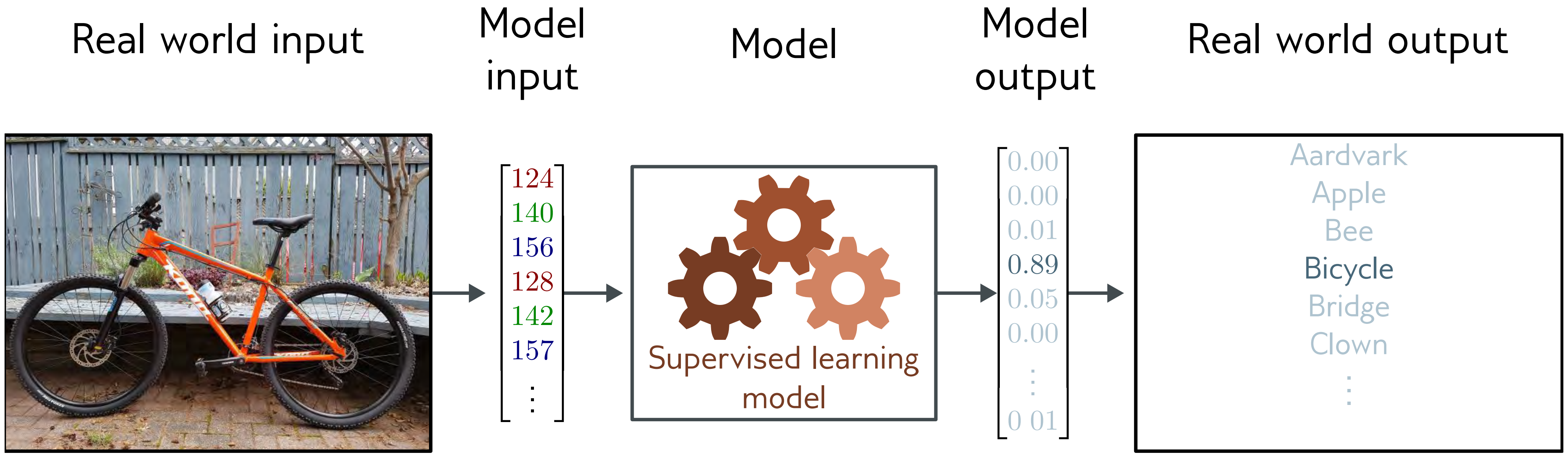
- Binary classification problem (two discrete classes)
- Transformer network

Music genre classification



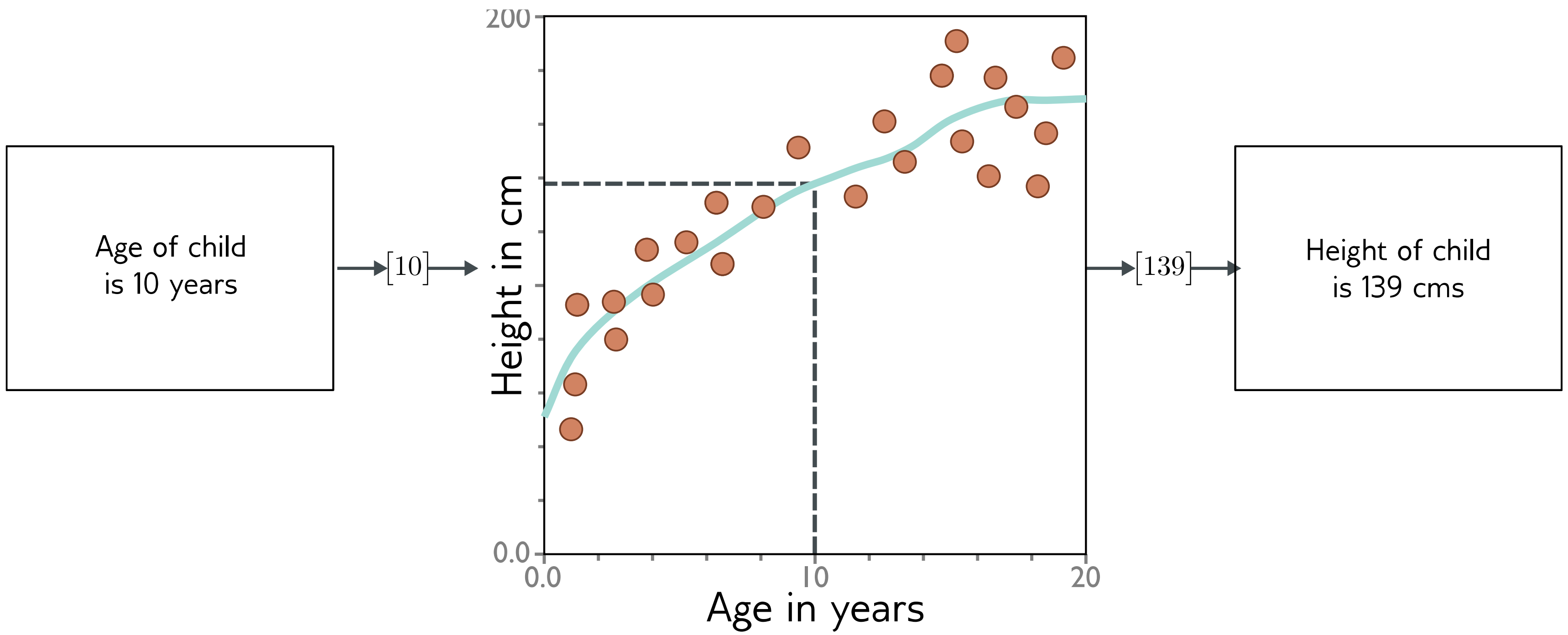
- Multiclass classification problem (discrete classes, >2 possible values)
- Recurrent neural network (RNN)

Image classification



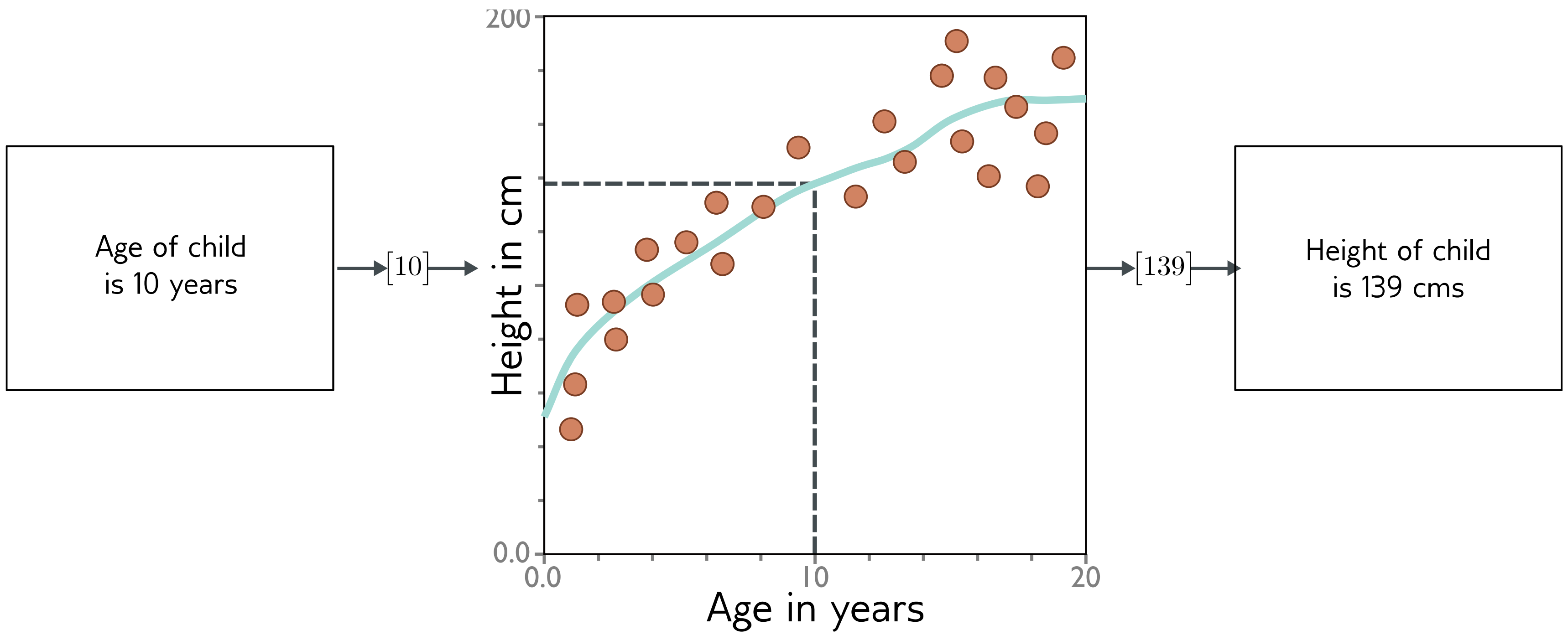
- Multiclass classification problem (discrete classes, >2 possible classes)
- Convolutional network

What is a supervised learning model?



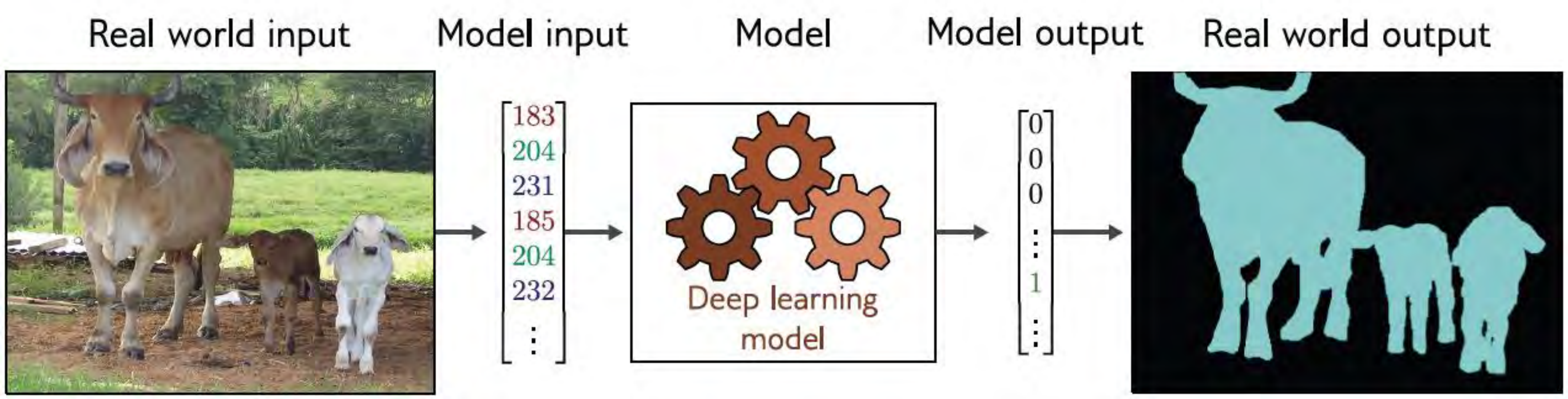
- An equation relating input (age) to output (height)
- Search through family of possible equations to find one that fits training data well

What is a supervised learning model?



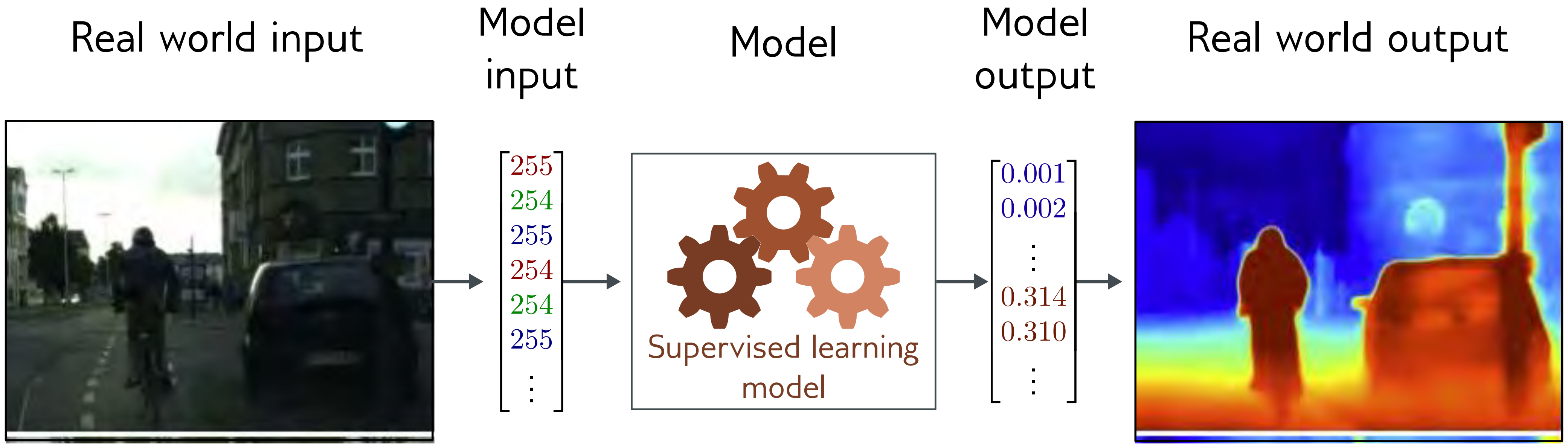
- Deep neural networks are just a very flexible family of equations
- Fitting deep neural networks = “Deep Learning”

Image segmentation



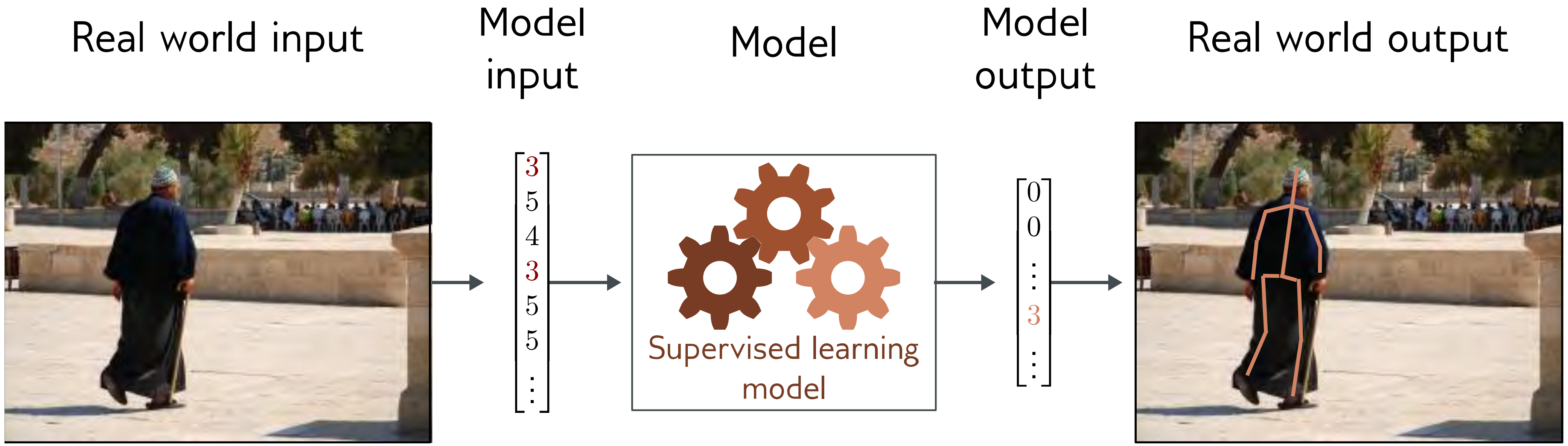
- Multivariate binary classification problem (many outputs, two discrete classes)
- Convolutional encoder-decoder network

Depth estimation



- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Pose estimation



- Multivariate regression problem (many outputs, continuous)
- Convolutional encoder-decoder network

Terms

- Regression = continuous numbers as output
- Classification = discrete classes as output
- Two class and multiclass classification treated differently
- Univariate = one output
- Multivariate = more than one output

Translation

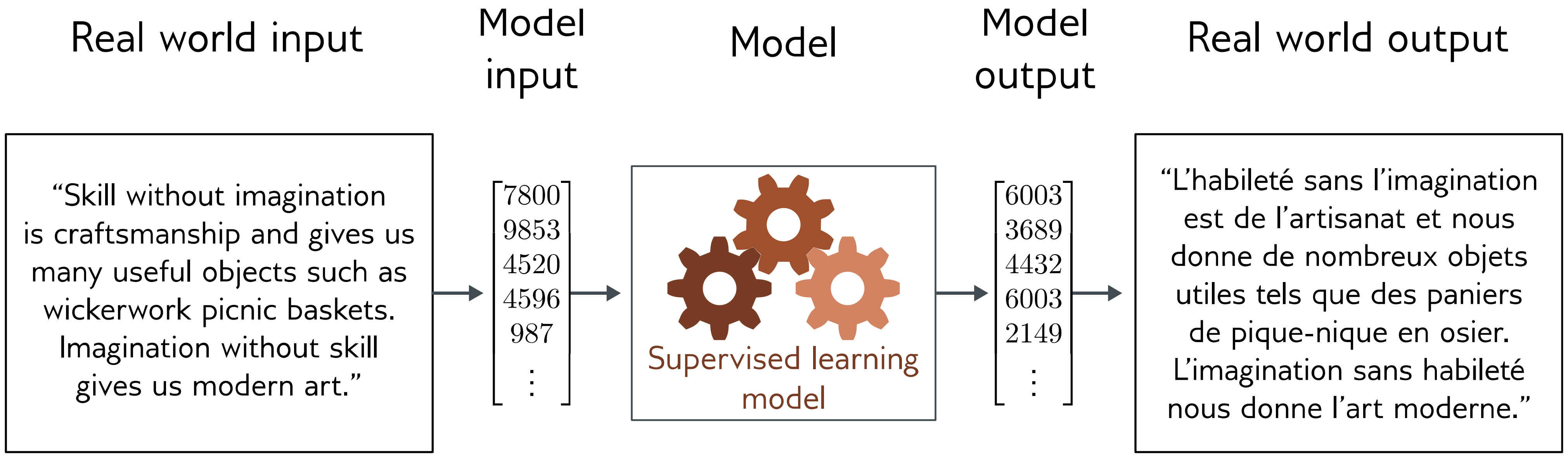


Image captioning

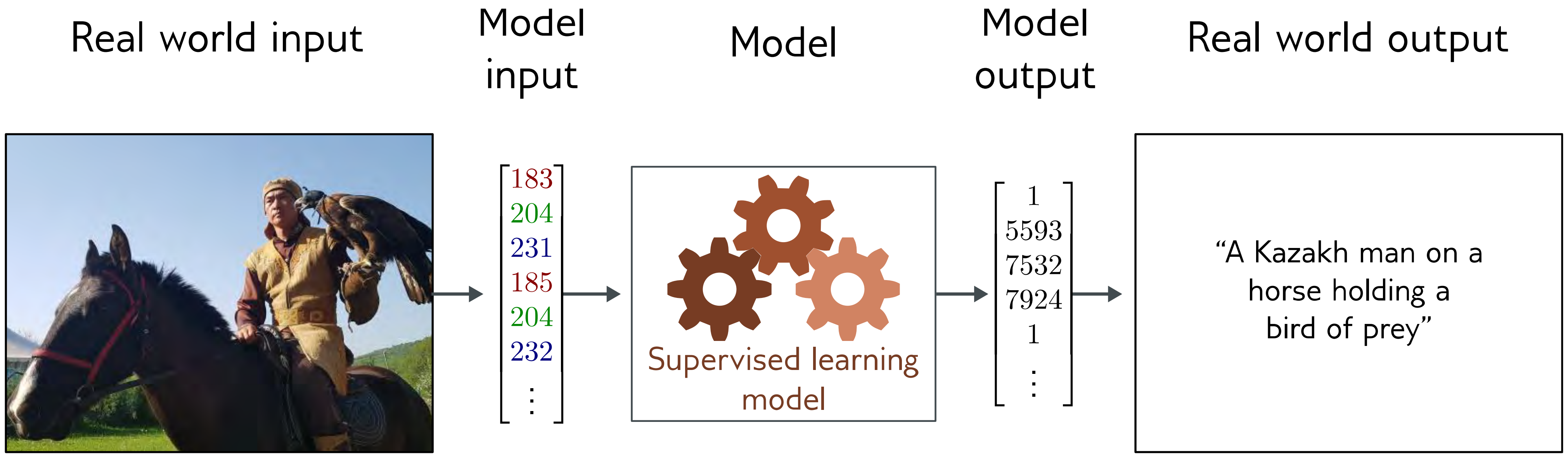
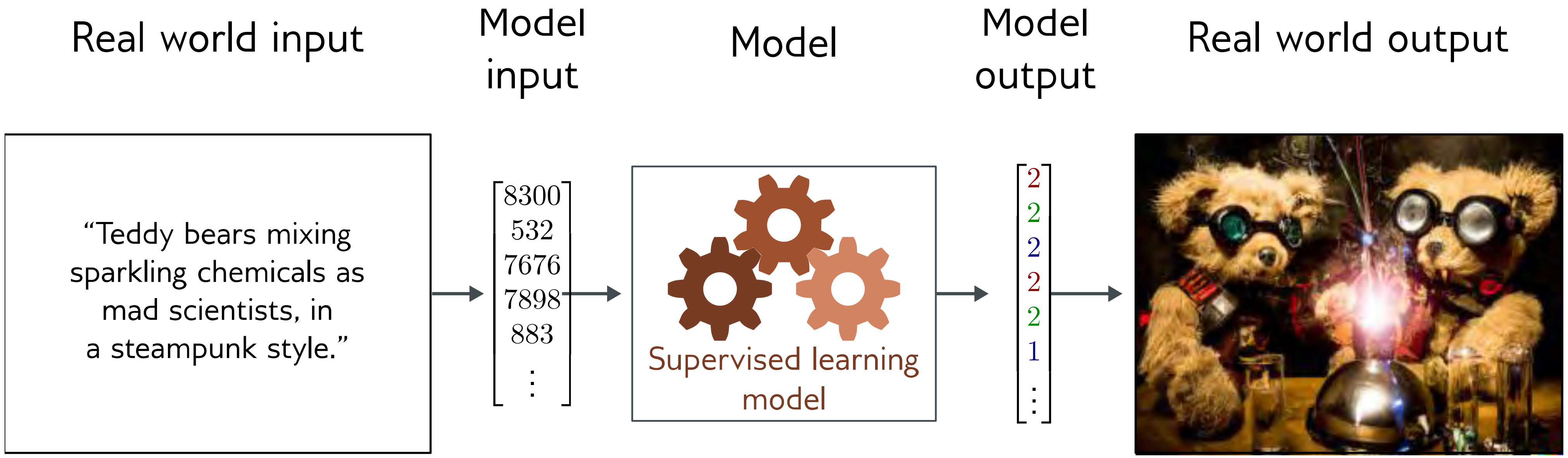


Image generation from text



What do these examples have in common?

- Very complex relationship between input and output
- Sometimes may be many possible valid answers
- But outputs (and sometime ules

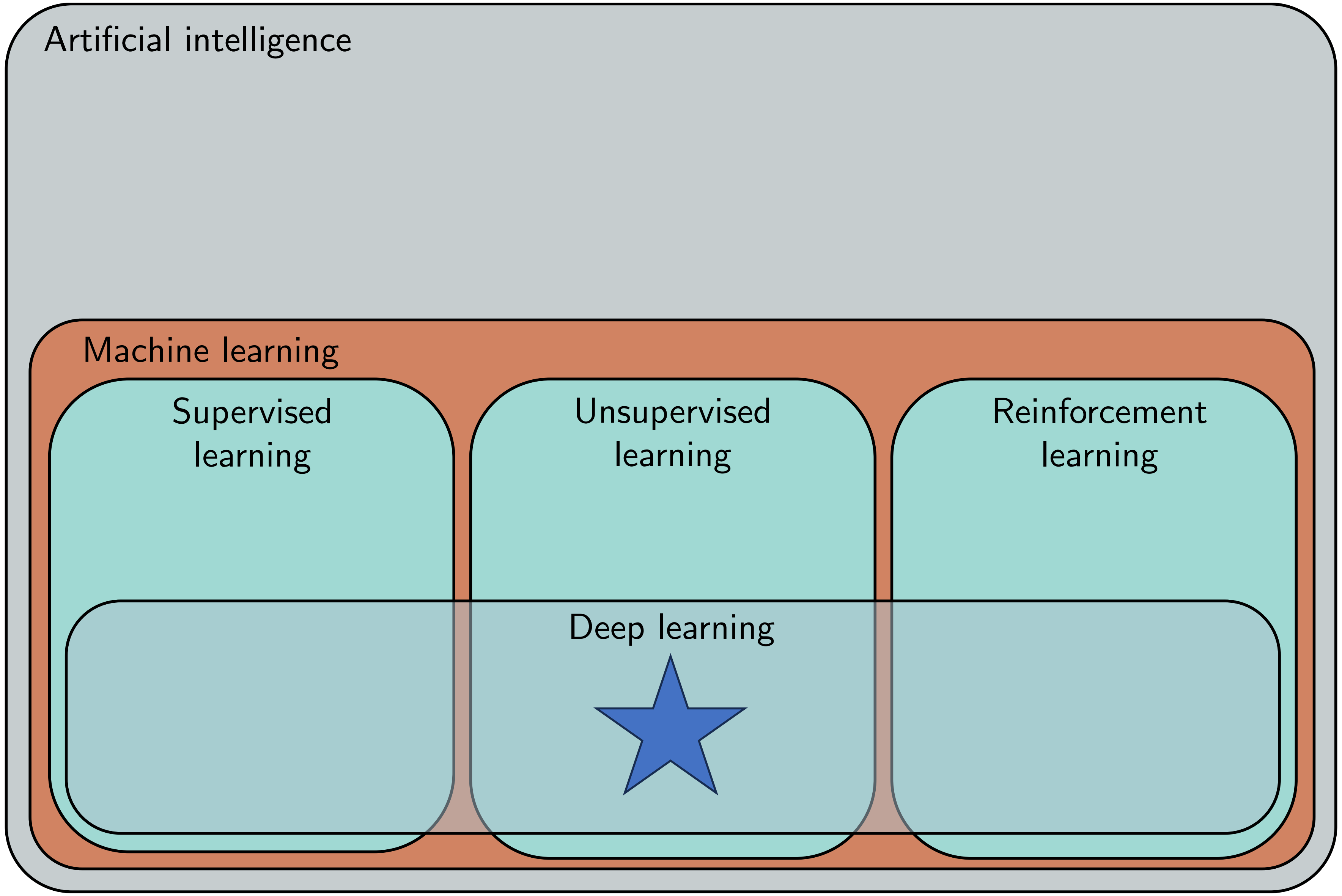
"A Kazakh man on a horse holding a bird of prey"

Language obeys grammatical rules

Natural images also have "rules"

Idea

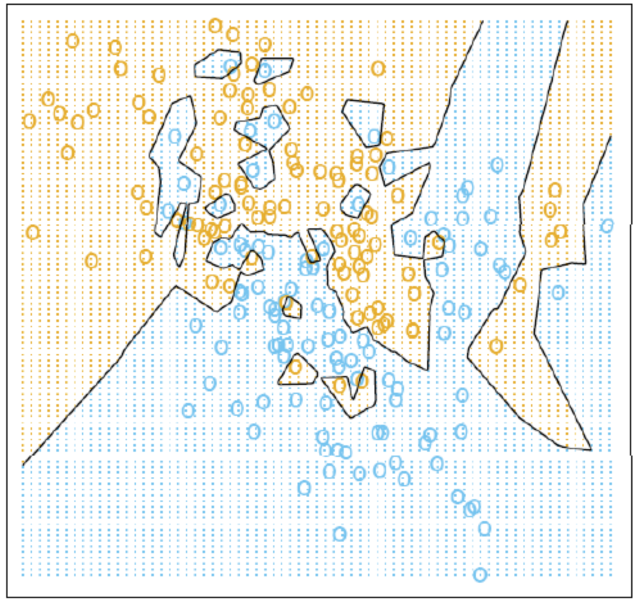
- Learn the “grammar” of the data from unlabeled examples
- Can use a gargantuan amount of data to do this (as unlabeled)
- Make the supervised learning task earlier by having a lot of knowledge of possible outputs


Artificial intelligence
Machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Deep learning

Representational Power

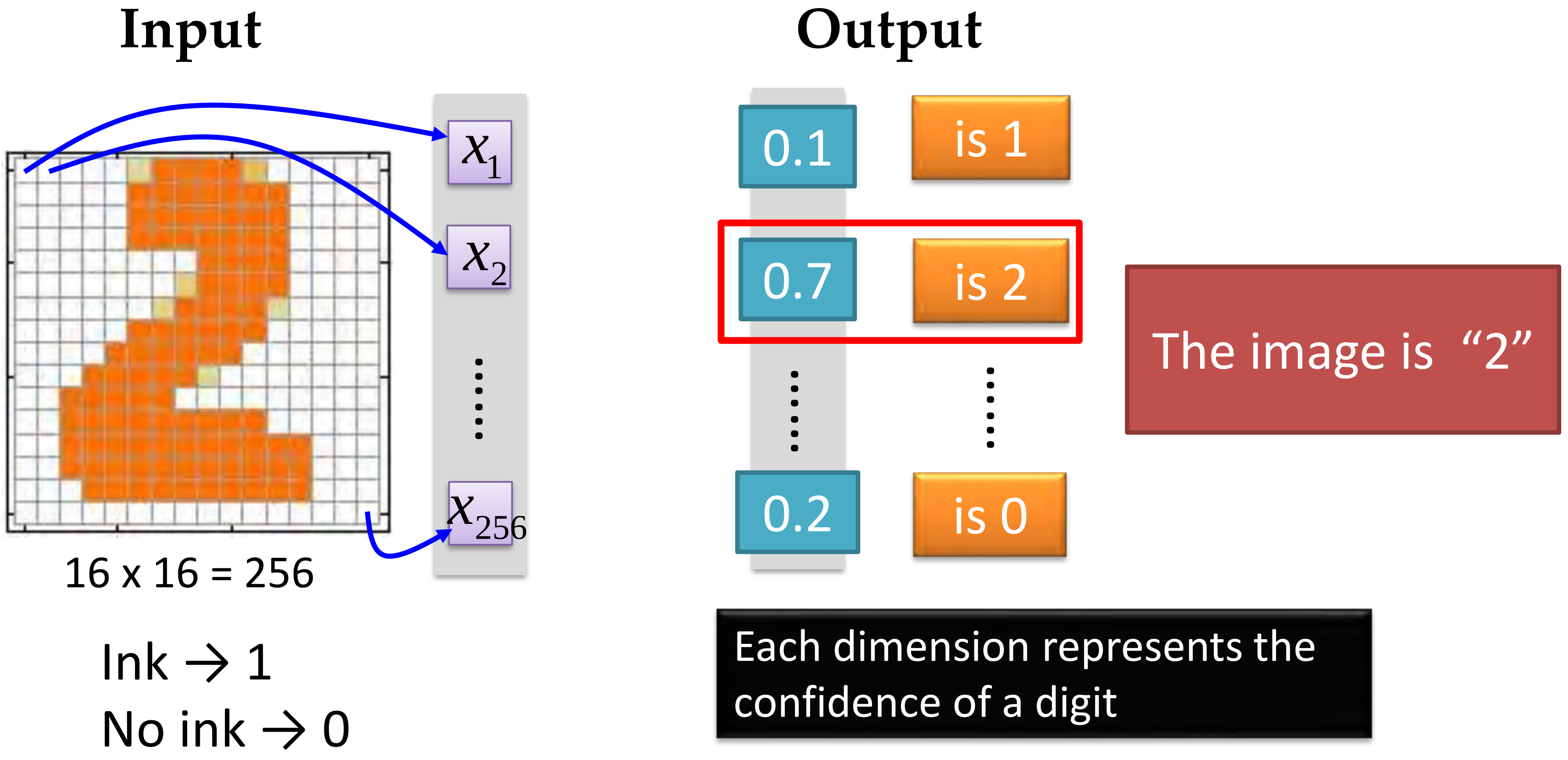
- NNs with at least one hidden layer are universal approximators
  - Given any continuous function $h(x)$ and some $\epsilon > 0$, there exists a NN with one hidden layer (and with a reasonable choice of non-linearity) described with the function $f(x)$, such that $\forall x, |h(x) - f(x)| < \epsilon$
  - I.e., NN can approximate any arbitrary complex continuous function

- NNs use nonlinear mapping of the inputs x to the outputs $f(x)$ to compute complex decision boundaries
- But then, why use deeper NNs?
  - The fact that deep NNs work better is an empirical observation
  - Mathematically, deep NNs have the same representational power as a one-layer NN



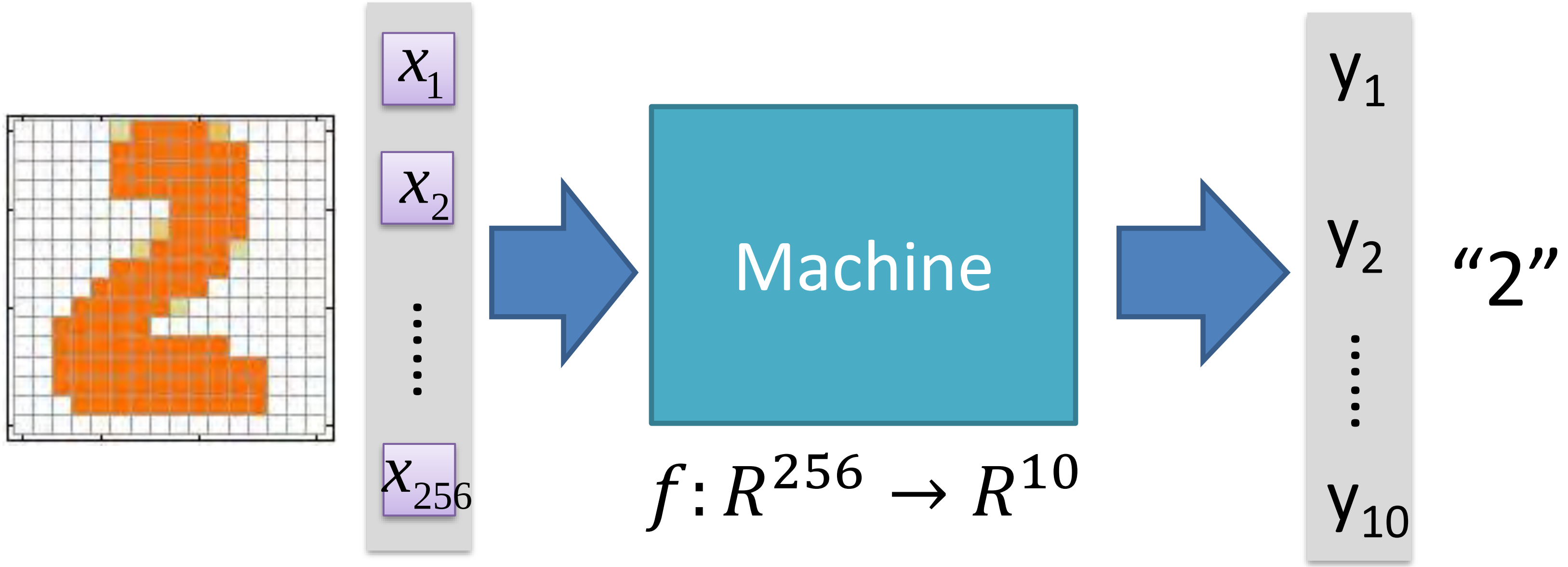
Introduction to Neural Networks

- Handwritten digit recognition (MNIST dataset)
  - The intensity of each pixel is considered an input element
  - Output is the class of the digit



Introduction to Neural Networks

- Handwritten digit recognition



The function f is represented by a neural network

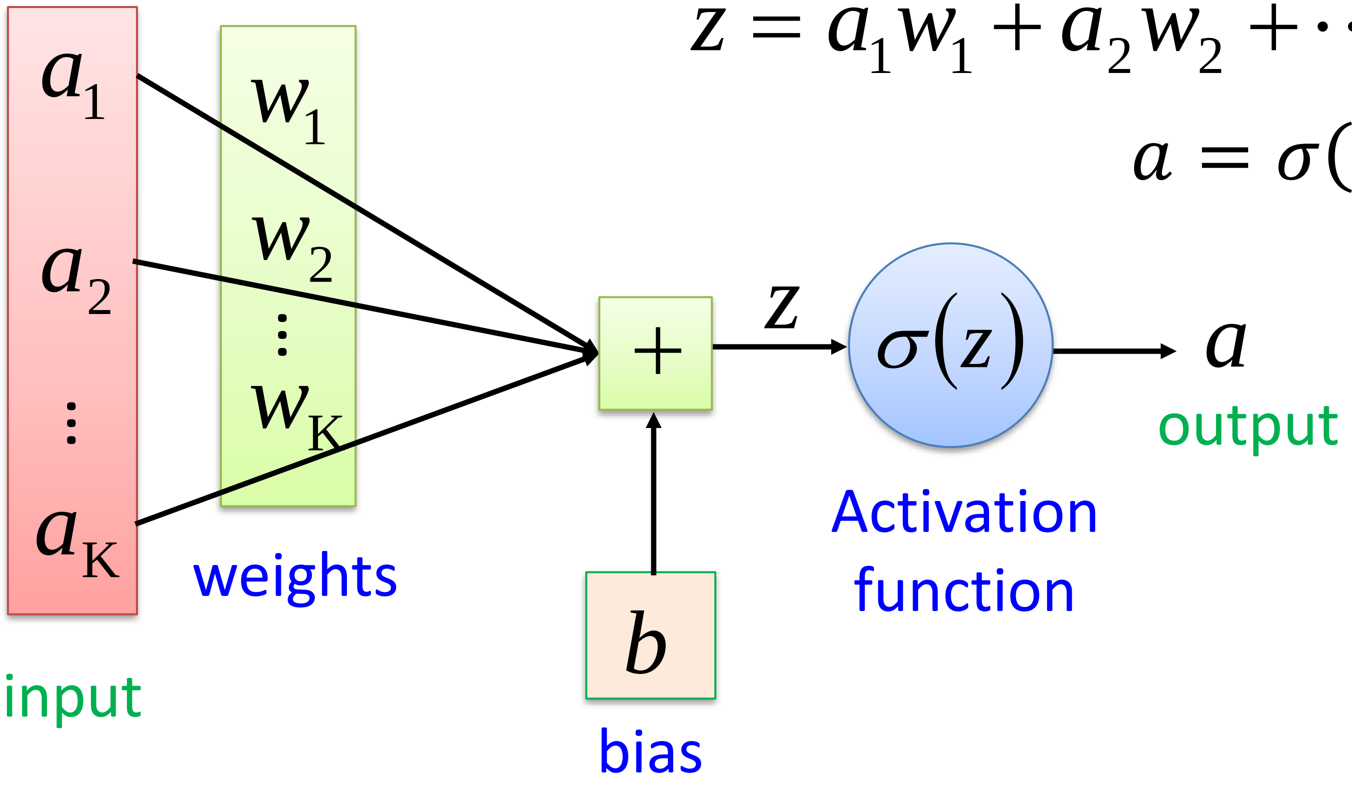
Elements of Neural Networks

Introduction to Neural Networks

- NNs consist of hidden layers with neurons (i.e., computational units)
- A single neuron maps a set of inputs into an output number, or $f: R^K \to R$

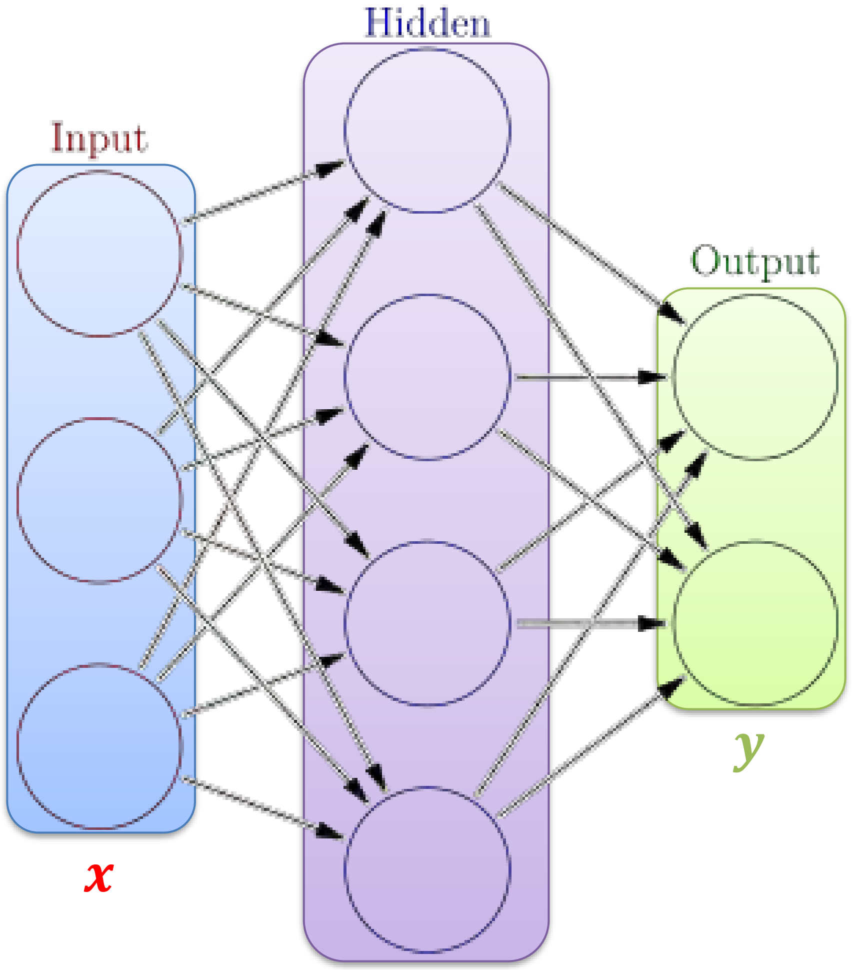
$$z = a_1 w_1 + a_2 w_2 + \cdots + a_K w_K + b$$

$$a = \sigma(z)$$



Elements of Neural Networks

- A NN with one hidden layer and one output layer



Weights Biases

$$\text{hidden layer } \boldsymbol{h} = \boldsymbol{\sigma}(\mathbf{W}_1 \boldsymbol{x} + \boldsymbol{b}_1)$$

$$\text{output layer } \boldsymbol{y} = \boldsymbol{\sigma}(\boldsymbol{W}_2 \boldsymbol{h} + \boldsymbol{b}_2)$$

Activation functions

4 + 2 = 6 neurons (not counting inputs)
[3 × 4] + [4 × 2] = 20 weights
4 + 2 = 6 biases

26 learnable parameters

Elements of Neural Networks

Introduction to Neural Networks

- A neural network playground [link]