Introduction to Machine Learning

Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 4.3.4

Artificial Intelligence &
Information Analysis Lab

Introduction to Machine Learning

General definition

- ***Machine Learning*** (***ML***) is the process of improving learner's performance of a task through experience.
- This is typically achieved by defining an appropriate ***objective function*** for the given algorithm and the given task, then optimizing it.

Historic retrospective

- Statistics.
- 1950s: Perceptron.
- 1960s: Bayesian Learning.
- 1980s: Multilayer Perceptrons.
- 1990s: Kernel methods.
- 2010s: Deep Neural Networks.

Introduction to Machine Learning

General notations:

- $\mathbf{x} \in \mathbb{R}^n$: ML model input feature vector.
- $\mathbf{y} \in \mathbb{R}^m$: target label vector.
- $\hat{\mathbf{y}} \in \mathbb{R}^m$: predicted (estimated) ML model output vector.
- N: number of examples in the dataset $\mathcal{D}$.
- n: input vector dimensionality
- m: output dimensionality (e.g. number of classes).

- ***ML model***: a learnable function typically of the form $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$.
  - Its structure may be predefined.
  - Its parameter vector $\boldsymbol{\theta}$ is typically learned through training, by optimizing an objective function $J(\mathbf{y}, \hat{\mathbf{y}})$.

Introduction to Machine Learning

- **Supervised learning**
  - **Classification/recognition/identification, Identity verification**
  - **Regression, Object detection**
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- Semi-supervised learning
  - Label propagation
- Self-supervised learning
  - Autoencoders
- Reinforcement Learning
  - Curiosity driven Learning
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- Other topics

Supervised Learning

- A sufficient large training sample set $\mathcal{D}$ is required for Supervised Learning (regression, classification):

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}.$$

- $\mathbf{x}_i \in \mathbb{R}^n$: n –dimensional input (feature) vector of the i-th training sample.
- $\mathbf{y}_i$: its target label (output).
- Target vector $\mathbf{y}$ can be:
  - real-valued vector: $\mathbf{y} \in [0, 1]^m, \mathbf{y} \in \mathbb{R}^m$;
  - binary-valued vector $\mathbf{y} \in \{0,1\}^m$ or even categorical.

Classification/Recognition/ Identification

- Given a set of classes $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$ and a sample $\mathbf{x} \in \mathbb{R}^n$, the ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ predicts a class membership vector $\hat{\mathbf{y}} \in [0, 1]^m$ for input sample $\mathbf{x}$, where $\boldsymbol{\theta}$ are the learnable model parameters.

- Essentially, a probabilistic distribution is computed of the likelihood of the given sample $\mathbf{x}$ belonging to each class $\mathcal{C}_i$.

- Single-target classification:
  - classes $\mathcal{C}_i, i = 1, \dots, m$ are mutually exclusive: $||\mathbf{y}||_1 = 1$.
- Multi-target classification:
  - classes $\mathcal{C}_i, i = 1, \dots, m$ are not mutually exclusive: $||\mathbf{y}||_1 \geq 1$.

Classification/Recognition/ Identification

- ***Training***: Given N pairs of training samples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in [0,1]^m$, estimate $\boldsymbol{\theta}$ by minimizing a loss function: $\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\mathbf{y}, \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta}))$.

- ***Inference***: Given an input example $\mathbf{x} \in \mathbb{R}^n$, compute the prediction vector $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ and assign the example to the most probable class with label $\mathrm{L} = \underset{i}{\operatorname{argmax}}(\hat{\mathbf{y}} = [\hat{y}_i]^T, i = 1, \dots, m)$.

- ***Testing***: Given N_t pairs of testing examples $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N_t\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in [0,1]^m$, compute (***predict***) $\hat{\mathbf{y}}_i$ and calculate a performance metric, e.g., classification accuracy.

Classification/Recognition/ Identification

Optimal step between training and testing:

- ***Validation***: Given N_v pairs of testing examples (different from either training or testing examples) $\mathcal{D}_v = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N_v\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in [0,1]^m$, compute (predict) $\hat{\mathbf{y}}_i$ and validate using a performance metric.
- ***k-fold cross-validation*** (optional):
- Use only a percentage $(100 - \frac{100}{k})\%$, of the data for training and the rest for validation ($\frac{100}{k}\%$, e.g., 20%). Repeat it k times, until all data used for training and testing).
- Example: for 5-fold validation, 5 rounds each using:
  - 80% of the data for training and 20% for testing.

Classification

***Two-class classification*:**

- Two class ($m = 2$) and multiple class ($m > 2$) classification.
- Example: *Face detection* (two classes).

- Two class (binary) classification
  - One (binary) hypothesis to be tested:

$$\mathcal{H}_1: \quad \mathbf{x} \in \mathcal{C}_1, \qquad \mathcal{H}_2: \quad \mathbf{x} \in \mathcal{C}_2.$$

Classification

Multiclass Classification $(m > 2)$**:**

- Multiple $(m > 2)$ hypotheses testing: choose a winner class out of m classes.
- Binary hypothesis testing:
  - ***One class against all***: m binary hypotheses.
    - one must be proven true.
  - ***Pair-wise class comparisons***: $m(m-1)/2$ binary hypotheses.

Face Recognition/identification

Problem statement:

- To identify a face identity
- Input for training: several facial ROIs per person
- Input for inference: a facial ROI
- Inference output: the face id
- Training set consists of annotated images $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N\}$

- Supervised learning applications:

  Biometrics
  Surveillance applications
  Video analytics.

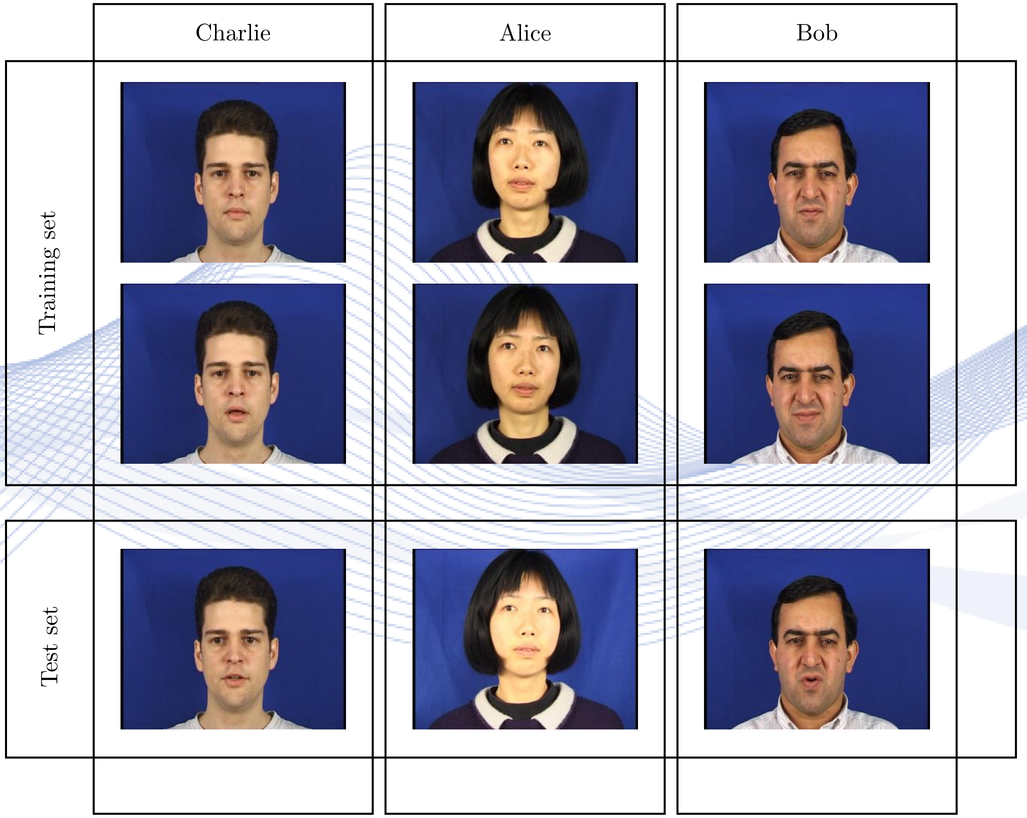
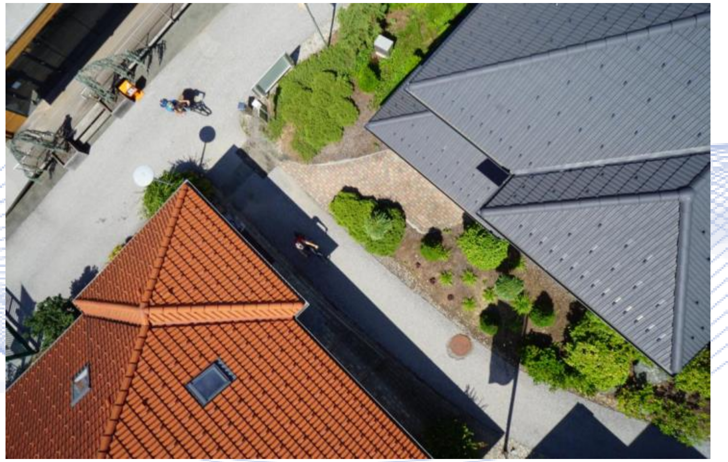
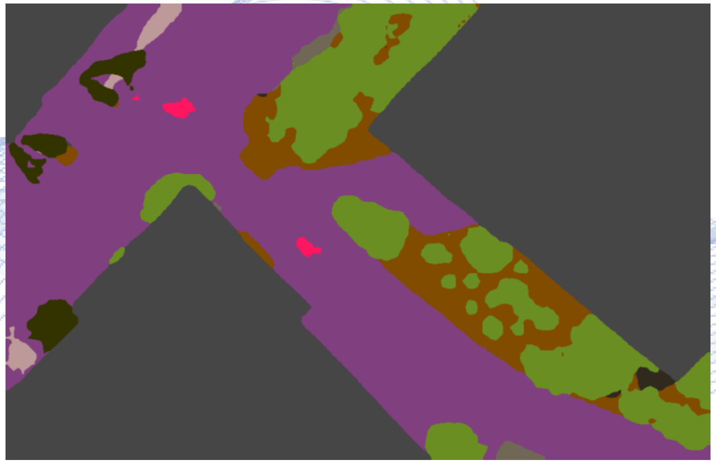


Image segmentation

Given a region class label set $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$, an image $\mathbf{x} \in \mathbb{R}^n$ must be segmented in m regions resulting in a segmentation map $\mathbf{Y} \in \mathbb{R}^{m \times n}$.

- The ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ and predicts a segmentation map $\widehat{\mathbf{Y}} \in \mathbb{R}^{m \times n}$, where a column $\hat{\mathbf{y}}_j \in \mathbb{R}^m, j = 1, \dots, n$ of $\widehat{\mathbf{Y}}$ provides a class label vector for each image pixel of the input image sample.
- The ML model must be trained on annotated images.





Regression

Given a sample $\mathbf{x} \in \mathbb{R}^n$ and a function $\mathbf{y} = \boldsymbol{g}(\mathbf{x}), \mathbf{y} \in \mathbb{R}^m$, the model predicts ***real-valued quantities*** for that sample: $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$, where $\hat{\mathbf{y}} \in \mathbb{R}^m$ and $\boldsymbol{\theta}$ are the learnable parameters of the model.

- **Training**: Given N pairs of training examples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^m$, estimate $\boldsymbol{\theta}$ by minimizing a loss function: $\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} J(\mathbf{y}, \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta}))$.

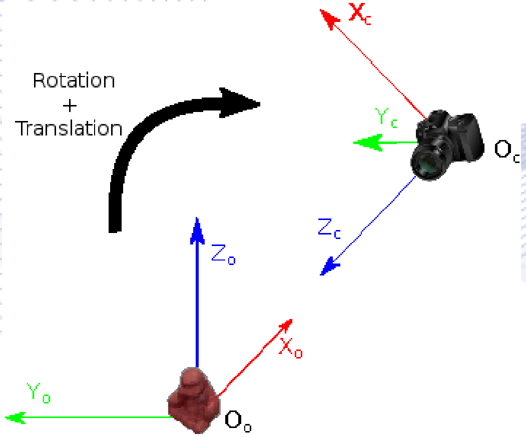
- **Testing**: Given N_t pairs of testing examples $\mathcal{D}_t = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N_t\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^m$, compute (predict) $\hat{\mathbf{y}}_i$ and calculate a performance metric, e.g., MSE.

Regression

- **Regression:**
  - Example: In object detection, regress object ROI parameters (width W, height H, offsets X, Y).
  - **Function approximation:** it is essentially regression, when the function $\mathbf{y} = \boldsymbol{f}(\mathbf{x})$ is known.

6D object pose regression

- ***Object pose***: 3 3D object translation and 3 3D rotation parameters vs camera coordinate system or ***Camera pose***: 3D camera translation and 3 3D rotation parameters vs a reference coordinate system (e.g., the scene).
- Can be formulated purely as a regression task, or a hybrid one with classification and regression approaches.
- Only a set of pose-annotated object pictures are needed for ML model training.



Multi-task Machine Learning

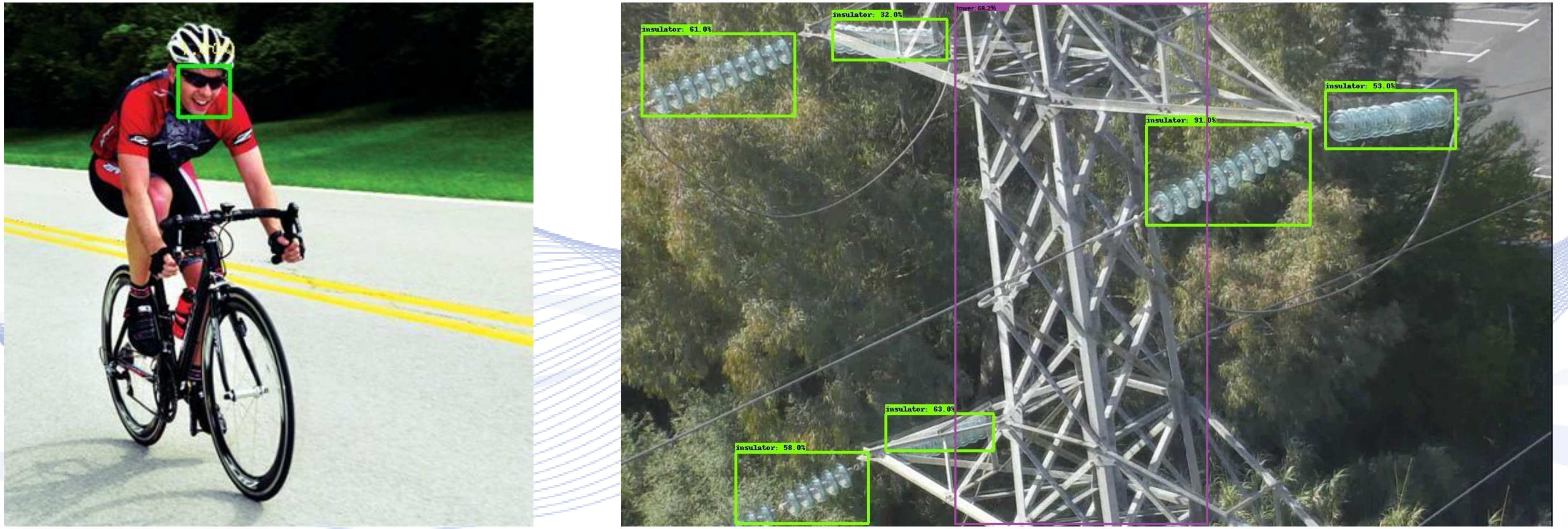
- The same ML model $\mathbf{y} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ is optimized to learn performing multiple tasks, e.g.:
  - Object recognition
  - Region-of-Interest (bounding box) regression
  - Region segmentation
  - Depth regression.
- Output: $\mathbf{y} = [\mathbf{y}_1^T | \dots | \mathbf{y}_M^T]^T$ for M different tasks.

- Optimization of a joint cost function:

$$\min_{\boldsymbol{\theta}} J(\mathbf{y}, \hat{\mathbf{y}}) = \alpha_1 J_1(\mathbf{y}, \hat{\mathbf{y}}) + \cdots + \alpha_M J_M(\mathbf{y}, \hat{\mathbf{y}}).$$

Object Detection

- Object detection = classification + localization:
- Find ***what*** is in a picture as well as ***where*** it is.



a) Face detection; b) Electrical Insulator detection.

Object Detection

Detection is a multi-objective machine learning:

- combination of classification and regression.
- Given a set of classes $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$ and an image sample $\mathbf{x} \in \mathbb{R}^n$, the model predicts (for one object instance only) an output vector $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_1^T | \hat{\mathbf{y}}_2^T]^T$ consisting of:
  - A class vector $\hat{\mathbf{y}}_1 \in [0, 1]^m$ and
  - A bounding box parameter vector $\hat{\mathbf{y}}_2 = [x, y, w, h]^T$ corresponding to object ROI.
- Optimization of a joint cost function:

$$\min_{\boldsymbol{\theta}} J(\mathbf{y}, \hat{\mathbf{y}}) = \alpha_1 J_1(\mathbf{y}_1, \hat{\mathbf{y}}_1) + \alpha_2 J_2(\mathbf{y}_2, \hat{\mathbf{y}}_2).$$

- The above vector pair will be computed for every possible target detected in the image sample $\mathbf{x}$.

Object tracking

- Given two video frames at times $t, t+1$ and a detected object at time t described by:
  - a vector $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_1^T | \mathbf{x}^T | \boldsymbol{\mathcal{I}} | \boldsymbol{\mathcal{M}} | \hat{\mathbf{y}}_2^T]^T(t)$ consisting of:
  - ROI parameter vector $\hat{\mathbf{y}}_1(t) = [x, y, w, h]^T$.
  - ROI image content (feature) vector $\mathbf{x}^T(t)$.
  - A unique object id $\boldsymbol{\mathcal{I}}$.
  - Object model $\boldsymbol{\mathcal{M}}$ (optional). It can be learnt:
    - a) a set of representative images, b) an ML model.
  - (optional) Object identification/recognition:
    - produce a class vector $\hat{\mathbf{y}}_2(t) \in [0, 1]^m$.
    - At times, $\boldsymbol{\mathcal{I}}$ may coincide with the winner class label $\hat{\mathbf{y}}_{1w}$.

Object tracking

- Track this object in video frame $t+1$:
  - (Optional) Predict object position $[x,y]^T(t+1)$;
  - Find ROI parameter vector $\hat{\mathbf{y}}_1(t+1)=[x,y,w,h]^T$ within a ***search region*** on video frame $t+1$;
  - Retain object id $\boldsymbol{\mathcal{I}}(t+1)=\boldsymbol{\mathcal{I}}(t)$;
  - update model vector $\boldsymbol{\mathcal{M}}$(optional);
  - Object identification/recognition (optional)
    - produce a class vector $\hat{\mathbf{y}}_2(t+1)\in[0,1]^m$.

Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- **Unsupervised learning**
  - **Clustering**
  - **Dimensionality reduction, data retrieval**
- Semi-supervised learning
  - Label propagation
- Self-supervised learning
  - Autoencoders
- Reinforcement Learning
  - Curiosity driven Learning
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- Other topics

Unsupervised Learning

- In ***unsupervised learning,*** the ML model is provided with samples containing exclusively input feature vectors, without neither labels nor any information about the specific desired output:

$$\mathcal{D} = \{\mathbf{x}_i, i = 1,2, \dots, N\}$$

- $\mathbf{x} \in \mathbb{R}^n$.
- Unsupervised learning-based models are used for discovering the underlying structure of the data.

Clustering

- **Input:** A predefined number of clusters $\mathcal{C} = \{\mathcal{C}_i, i = 1, 2, \ldots, m\}$ and a set of unlabeled samples $\mathcal{D} = \{\mathbf{x}_i, i = 1, 2, \ldots, N\}$ $\mathbf{x}_i \in \mathbb{R}^n$.
  - Number of clusters m may be unknown.
- **Output:** Sample set $\mathcal{D} = \{\mathbf{x}_i, i = 1, 2, \ldots, N\}$ partition to m clusters $\mathcal{C}_i, i = 1, \ldots, m$
  - Cluster samples are similar and dissimilar to the samples of other clusters based on similarity/distance metric $\| \cdot \|$.
- Basically, clustering involves unlabeled data according to feature similarities.

Face clustering

Problem statement:

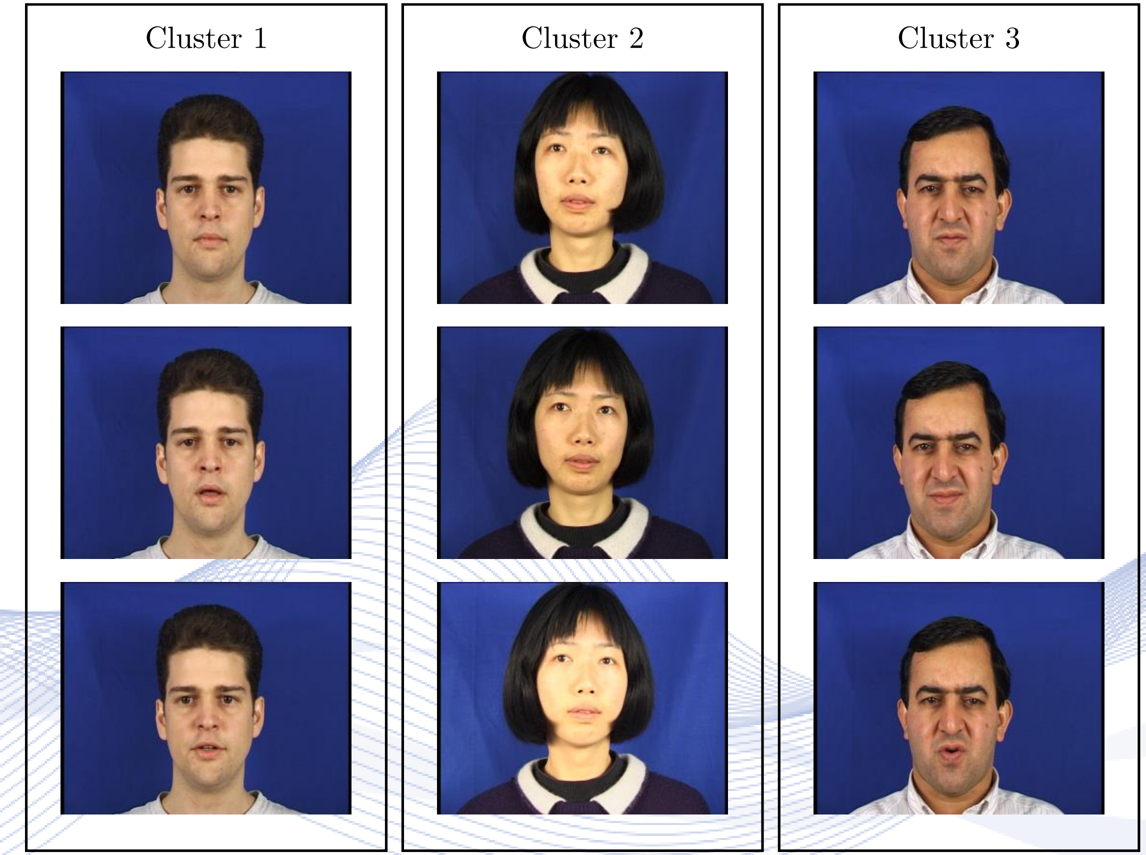
- To cluster facial images
- Input: many facial ROIs
- Output: facial image clusters

- Unsupervised learning
- Applications:

  Biometrics

  Surveillance applications

  Video analytics



Siamese Networks

Triplet loss function:

Let $\mathbf{x}_a$, $\mathbf{x}_p$, $\mathbf{x}_n$ be an ***anchor image sample***, a ***positive image sample*** similar to $\mathbf{x}_a$ in terms of a specific characteristic (e.g. class, 3D pose) and a ***negative image sample*** dissimilar to $\mathbf{x}_a$in terms of the same characteristic, respectively.

- Also, let $\mathbf{y}_a$, $\mathbf{y}_p$, $\mathbf{y}_n$ be the outputs of a ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ using $\mathbf{x}_a$, $\mathbf{x}_p$, $\mathbf{x}_n$ as inputs, respectively.
- The goal is to minimize a triplet loss function:

$$J_{triplet} = \max\left(||\mathbf{y}_a, \mathbf{y}_p|| - ||\mathbf{y}_a, \mathbf{y}_n|| + m, 0\right).$$

- $||\cdot||$: a distance metric.
- m: the selected margin.

Siamese Networks

- ***Siamese networks*** employ a triplet loss function.

- The objective of $J_{triplet}$ is to keep the distance between the anchor and positive samples smaller than the distance between the anchor and negative samples.

Artificial Intelligence & Information Analysis Lab

Dimensionality Reduction

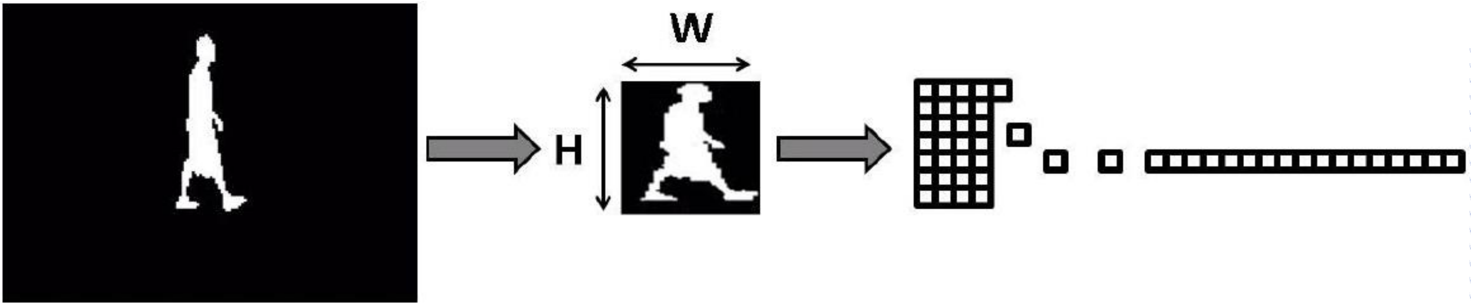
- Given a sample $\mathbf{x} \in \mathbb{R}^n$, the ML model computes a new sample representation:

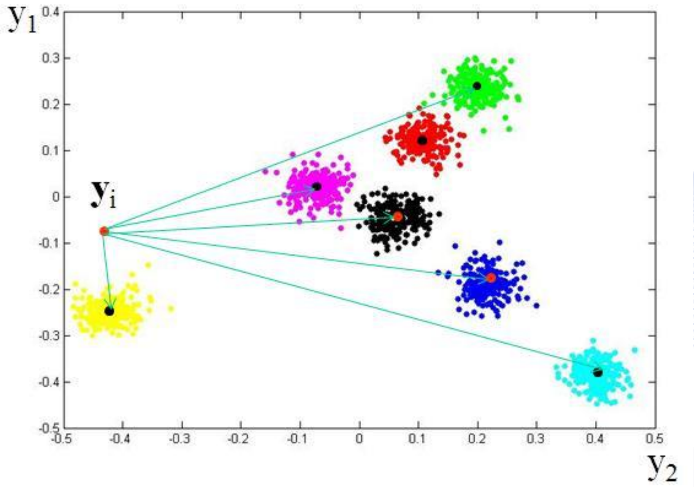
$$\hat{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}; \boldsymbol{\theta}),$$

- $\boldsymbol{\phi}: \mathbb{R}^n \rightarrow \mathbb{R}^d$ is a function, mapping $\mathbf{x}$ to a lower dimensionality space d, $d \ll n$,
- $\boldsymbol{\theta}$ is the learnable parameter vector of the model.
- The representation $\hat{\mathbf{x}}$ is meant:
  - to capture relevant high level information from the initial sample $\mathbf{x}$;
  - provide abstraction from detail;
  - increase robustness to noise.

Dimensionality Reduction

- Example: Human posture visualization.
- Dimensionality reduction from $\mathbf{p} \in \mathbb{R}^{HW}$ to $\mathbf{y} \in \mathbb{R}^2$





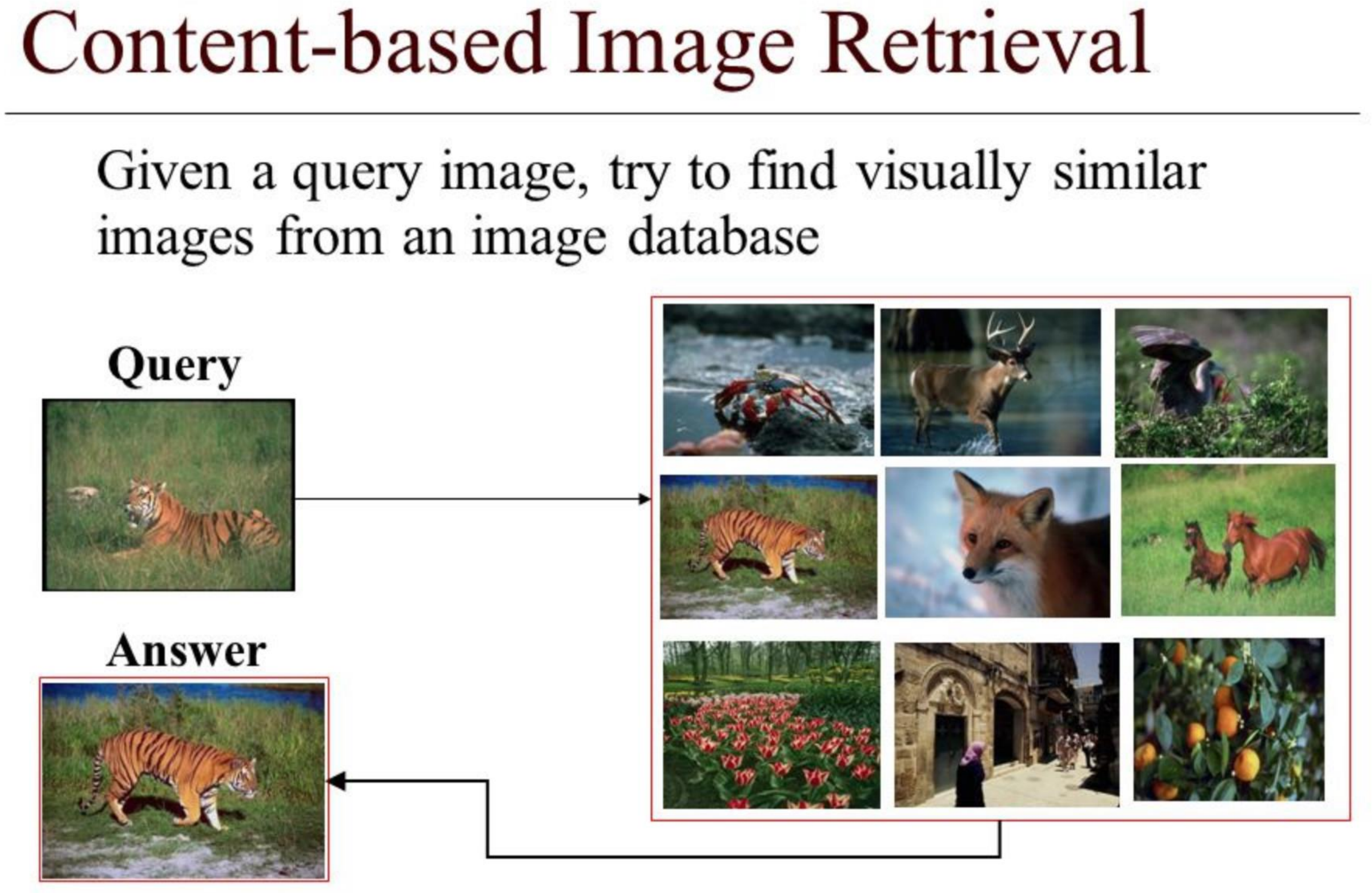
Binary human body image

Posture image of fixed size

Posture vector $\mathbf{p} \in \mathbb{R}^{HW}$

Posture visualization $\mathbf{y} \in \mathbb{R}^2$

Data Retrieval



Content-based Image Retrieval

Given a query image, try to find visually similar images from an image database

http://slideplayer.com/slide/3415344/12/images/6/Content-based+Image+Retrieval.jpg

Data Retrieval

Unsupervised data retrieval:

- Optional application of dimensionality reduction as a first step.

- Given a sample $\mathbf{x} \in \mathbb{R}^n$, the model initially computes a representation of that sample: $\hat{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}; \boldsymbol{\theta})$, where $\hat{\mathbf{x}} \in \mathbb{R}^d, d \ll n$ and $\boldsymbol{\theta}$ are learnable parameters of the model.
- Once this representation has been produced the goal of a ML retrieval model is to compare a ***query*** sample $\hat{\mathbf{x}}^q$ with every sample in a gallery set $\mathcal{G} = \{\hat{\mathbf{x}}_i^g, i = 1, 2, \dots, N_g\}$ based on a distance/similarity metric $\| \cdot \|$, and rank the gallery samples in a descending similarity order.

Identity Verification

- ***Unsupervised version***: In this version, the problem is reformulated to "are these two images of the same person?"
- Given a sample $\mathbf{x} \in \mathbb{R}^n$, the model initially computes a representation of that sample: $\hat{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}; \boldsymbol{\theta})$, where $\hat{\mathbf{x}} \in \mathbb{R}^d, d \ll n$ and $\boldsymbol{\theta}$ are the learnable parameters of the model. Once this representation has been produced the goal of a verification model is to determine whether two samples $\mathbf{x}_1$ (test sample) and $\mathbf{x}_2$ (identity sample) match:

$$\hat{y} = \begin{cases} 1, \|\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2\| < \varepsilon \\ 0, \|\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2\| \geq \varepsilon \end{cases}.$$

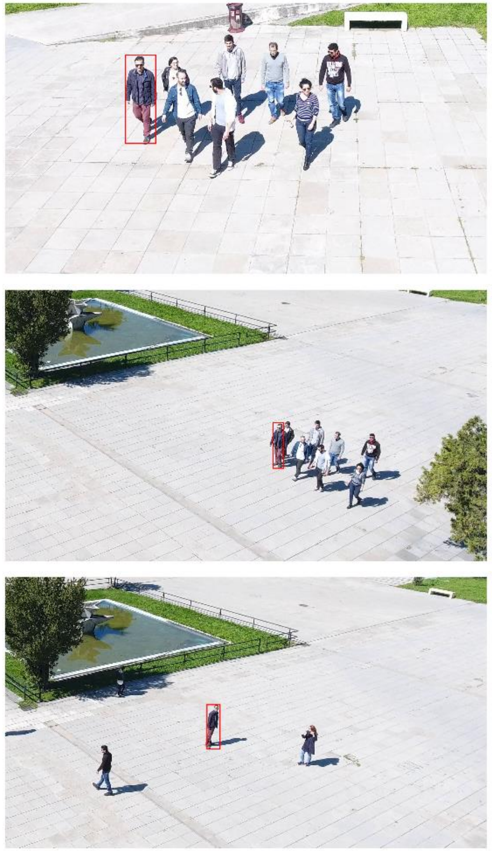
$\| \cdot \|$: a distance/similarity metric and ε is a threshold value.

Person re-identification

Re-identification

- Refers to the problem of associating/matching images of the same person taken:
  - from different cameras or
  - from the same camera in different occasions (e.g., night day).
- It can be solved as a data retrieval problem.

Example



Person re-identification

Re-identification as an image retrieval problem:

- Let $\hat{\boldsymbol{x}}^q$ be a query instance (person image) in one occasion/camera.
  - Typically, it results from: a) person detection or b) person tracking.
- Let $\mathcal{G}$= $\{\hat{\boldsymbol{x}}_i{}^G, i = 1, \dots, N_g\}$ be a gallery set, containing representations of various people in another condition.
  - They result from person detection/tracking and person identification
- Compare the query instance $\hat{\boldsymbol{x}}^q$ with every sample in $\mathcal{G}$, according to a similarity metric $||\cdot||$, and retrieve the most relevant item.

Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- **Semi-supervised learning**
  - **Label propagation**
- Self-supervised learning
- Reinforcement Learning
  - Curiosity driven Learning
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- Other topics

Semi-Supervised Learning

Semi-supervised learning:

- Combination of supervised and unsupervised learning.
- It relies on the existence of a large amount of training data, whose minority contains output information (data labels).
- Training dataset $\mathcal{D}$ consists of:
  - a set of N_1 labeled training examples, $\mathcal{D}_1 = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N_1\}$.
  - a set of N_2 unlabeled examples, $\mathcal{D}_2 = \{\mathbf{x}_i, i = 1, \ldots, N_2\}$, where $N_1 \ll N_2$:

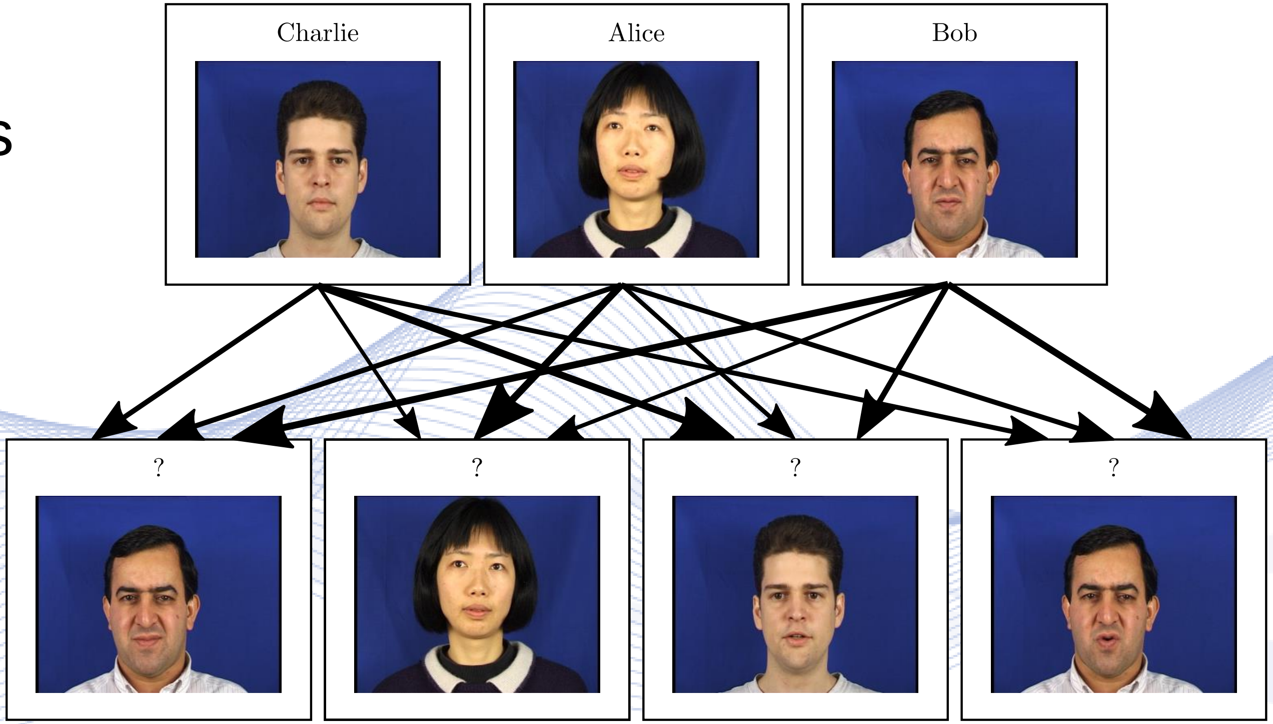
$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2.$$

- It is particularly useful for exploiting data structure (geometry) information.

Facial label propagation

Problem statement:

- To transfer labels from labeled to unlabeled facial images
- Input: a) labeled facial ROIs,
  b) unlabeled facial ROIs
- Output: facial image labels



- Semi-supervised learning
- Applications:

  Biometrics

  Surveillance applications

  Video analytics

Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- Semi-supervised learning
  - Label propagation
- **Self-supervised learning**
  - **Autoencoders**
- Reinforcement Learning
  - Curiosity driven Learning
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- Other topics

Self-Supervised Learning

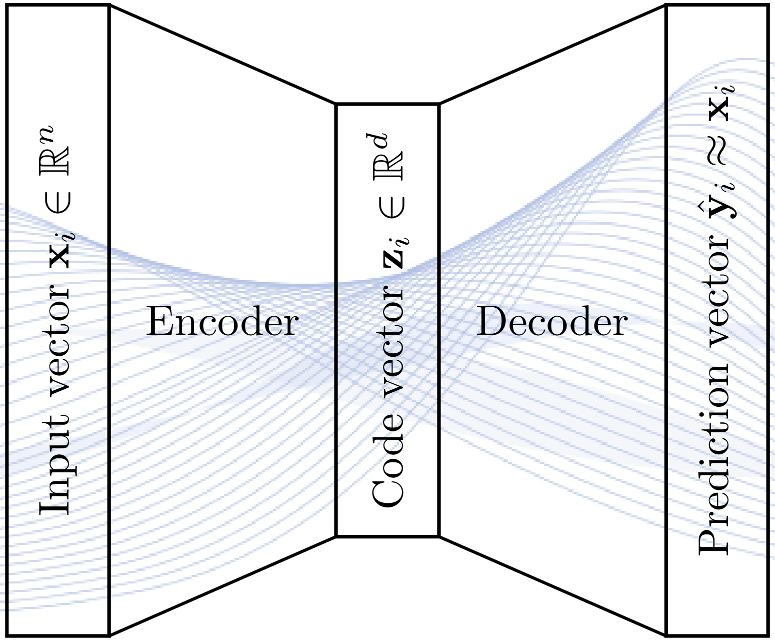
- ***Self-supervised learning*** resembles supervised learning.
- It relies on pairs of input-outputs, $(\mathbf{x}_i, \mathbf{y}_i)$ for ML model training.
- However, it does not require an explicit form of target labels $\mathbf{y}_i$.

- Instead, the necessary supervisory information is extracted from the input feature structure and correlations.

Autoencoders

Given a sample $\mathbf{x} \in \mathbb{R}^n$ and a function $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$, the model output $\hat{\mathbf{y}}$ should be equal to the model input $\mathbf{x}$:

- ***Training***: Given N pairs of training examples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N\}$, where $\hat{\mathbf{y}}_i = \mathbf{x}_i,\ \mathbf{x}_i \in \mathbb{R}^n$, estimate $\boldsymbol{\theta}$ by minimizing a loss function: $\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\mathbf{x}, \hat{\mathbf{y}})$.


Input vector $\mathbf{x}_i \in \mathbb{R}^n$
Encoder
Code vector $\mathbf{z}_i \in \mathbb{R}^d$
Decoder
Prediction vector $\hat{\mathbf{y}}_i \approx \mathbf{x}_i$

Autoencoder structure.

Video Frame Interpolation

Given a pair of consecutive video frames, video frame interpolation refers to the task of producing an in-between frame. This can be accomplished by learning two convolutional kernels, one for each frame.

- ***Training***: Given three consecutive frames at times $t, t+1, t+2$, then frames t and $t+2$ are concatenated into the input sample, while frame $t+1$ provides the ground truth.

- ***Testing***: The preceding frame is convolved with the preceding convolution kernel, while the subsequent frame is convolved with the subsequent convolution kernel. The sum of the convolution results provides the prediction for the in-between frame.

Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- Semi-supervised learning
  - Label propagation
- Self-supervised learning
  - Autoencoders
- **Reinforcement Learning**
  - **Curiosity driven Learning**
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- Other topics

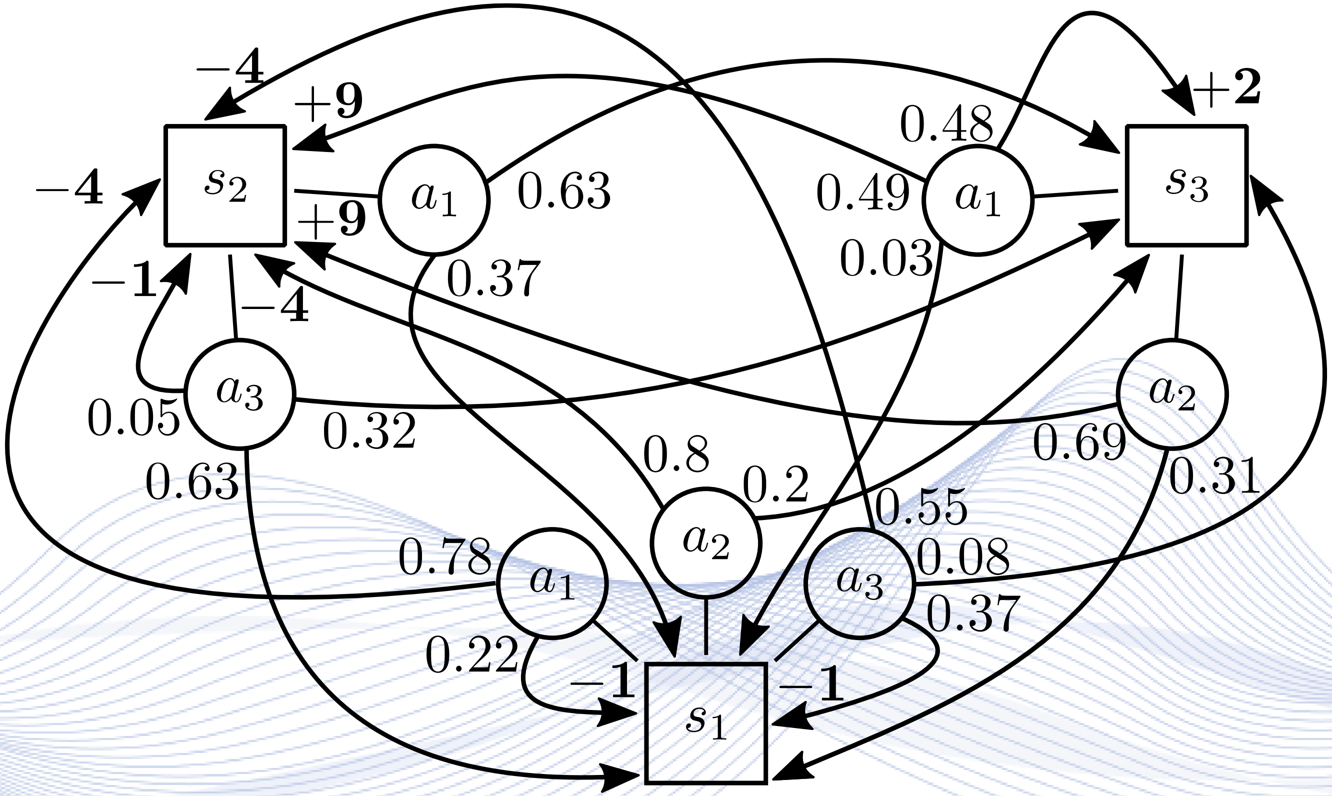
Reinforcement Learning

- ***Reinforcement Learning***: interaction scheme between an ML agent and its environment, in order to maximize some notion of cumulative rewards.

- Given a finite set of states $\mathcal{S} = \{s_i, i = 1,2, \dots, N_s\}$, a finite set of actions $\mathcal{A} = \{a_i, i = 1,2, \dots, N_a\}$, a reward function $R_a(s_i, s_j)$ and a probability function $P_a(s_j, r | s_i, a)$, where r is a reward, the goal of an RL model is to find a policy that maximizes a cumulative reward signal.

- ***Experience replay:*** Online reinforcement learning, based on remembering and reusing past experiences.

Reinforcement Learning



Markov process state transition diagram.

Artificial Intelligence &
Information Analysis Lab

Curiosity-Driven Learning

- One issue in Reinforcement Learning is that, if the reward function is too sparse, the learner may never receive rewards, thus have no way of improving.

- This can be addressed by manually designing a ***more dense reward function***, however this approach has drawbacks of its own (too complex, unintended consequences).

- In Curiosity-Driven Learning, the ML model attempts to predict its next state, then ***favors transitions that lead to wrong predictions***, as that indicates an unexplored region of its parameter vector space.

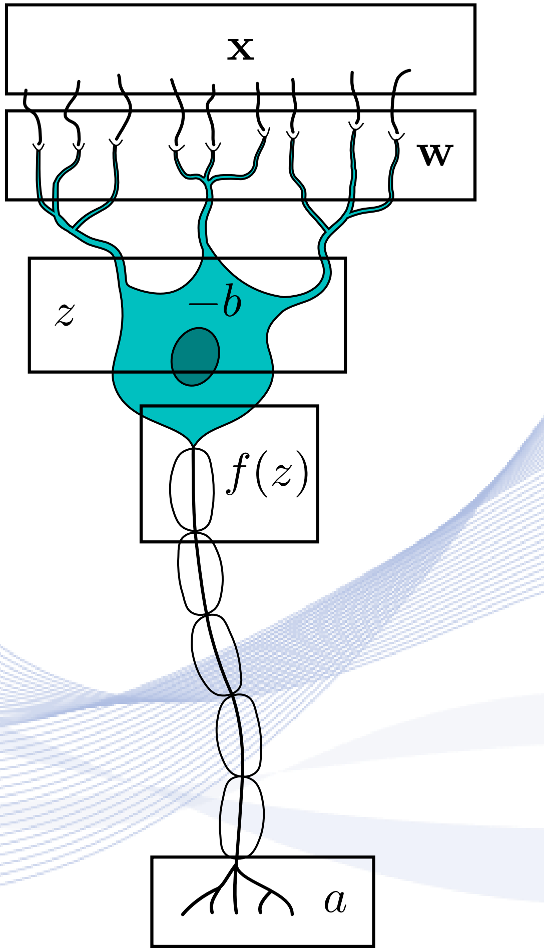
Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- Semi-supervised learning
  - Label propagation
- Self-supervised learning
  - Autoencoders
- Reinforcement Learning
  - Curiosity driven Learning
- **Neural Networks**
  - **Artificial Neural Networks, Deep Neural Networks**
  - **Adversarial Machine Learning**
  - **Generative Machine Learning**
  - **Temporal Machine learning (RNN)**
- Other topics

Artificial Neural Networks

Artificial neurons are mathematical models loosely inspired by their biological counterparts.

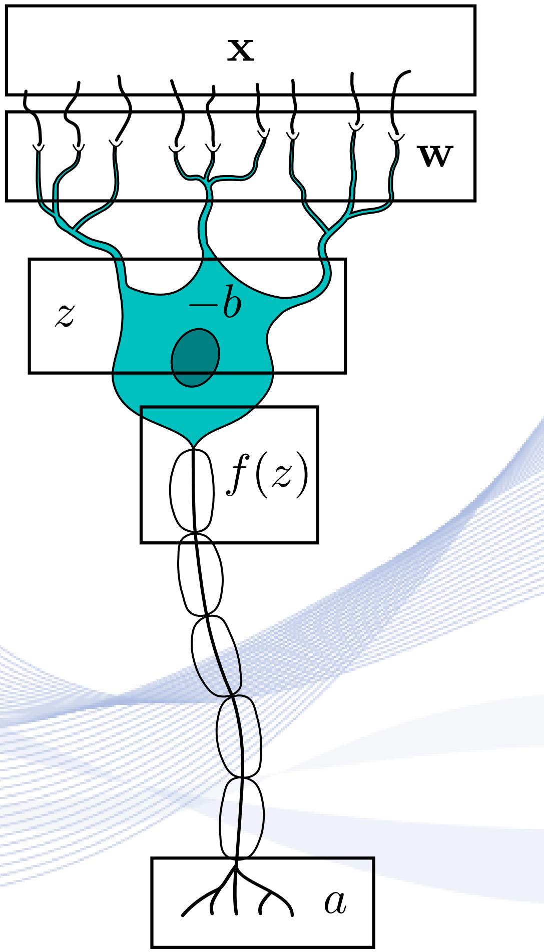
- Incoming signals: $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T,\ x_i \in \mathbb{R}$.
- Synaptic weights: $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T, \quad w_i \in \mathbb{R}$.
- Synaptic bias: $b \in \mathbb{R}$.
- Synaptic integration: $z = \sum_{i=1}^{n} w_i x_i + b = \mathbf{w}^T \mathbf{x} + \mathrm{b}$.
- Output nonlinearity.



Artificial Neural Networks

Artificial Neural Networks (***ANNs***) have a layered structure:

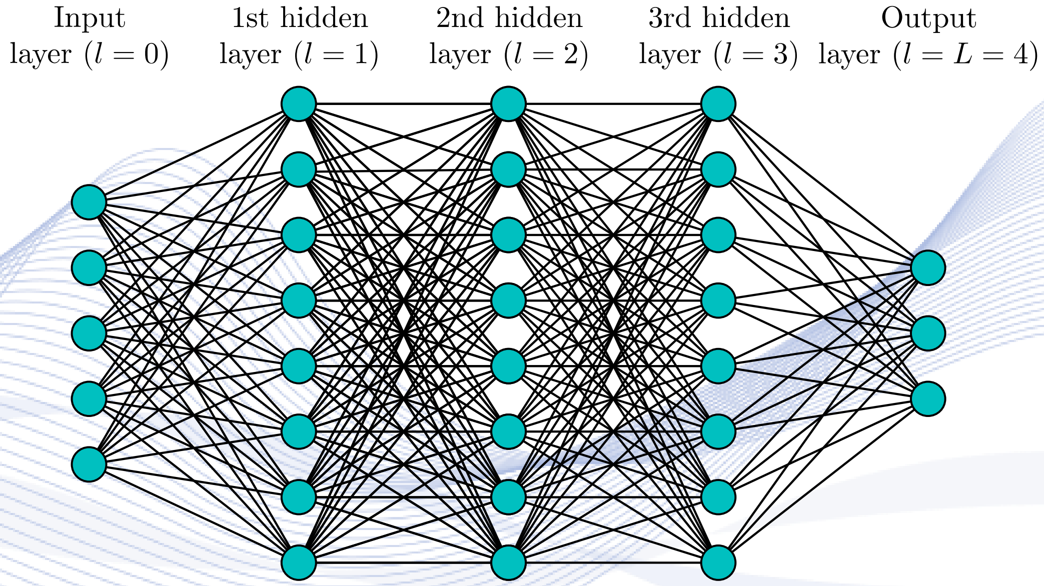
- Each layer consists of artificial neurons.
- They learn a function $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta})$ during training.



Deep Neural Networks

Deep Neural Networks (***DNNs***) have a count of layers (depth) $L \geq 3$.

- There are multiple hidden layers with regard to the MLP reference model.
- Typically, first layers are convolutional, latter ones are fully connected (CNNs).


Input layer $(l = 0)$
1st hidden layer $(l = 1)$
2nd hidden layer $(l = 2)$
3rd hidden layer $(l = 3)$
Output layer $(l = L = 4)$

Deep Neural Network with $L = 4$

Adversarial Machine Learning

Adversarial machine learning:

- Given a class label set $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$ and a trained ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}, \boldsymbol{\theta}), \hat{\mathbf{y}} \in [0,1]^m$
- find a perturbation $\mathbf{p}$, so that a perturbed test sample instance $\mathbf{x}_p = \mathbf{x} + \mathbf{p}$ (***adversarial sample***) is misclassified by the ML model as: $\hat{\mathbf{y}}_p = \boldsymbol{f}\left(\mathbf{x}_p; \boldsymbol{\theta}\right)$, where $\underset{i}{\operatorname{argmax}}(\hat{\mathbf{y}}_p = [\hat{y}_{p_i}]^T) \neq \underset{i}{\operatorname{argmax}}(\hat{\mathbf{y}} = [\hat{y}_i]^T) = \underset{i}{\operatorname{argmax}}(\mathbf{y} = [y_i]^T)$.

- ***ML training set augmentation***: during the training process apart from using real samples $\mathbf{x}_i, i = 1, \dots, N$ in the training set, we also include their perturbed instances $\mathbf{x}_{p_i}$, so that both $\mathbf{x}_i$ and $\mathbf{x}_{p_i}$ are correctly classified.
- Adversarial training works as a regularization technique, in order to derive a more robust ML model.

Object de-identification

Object de-identification:

- Given a class label set $\mathcal{C} = \{\mathcal{C}_i, i = 1, \ldots, m\}$ and a trained ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x}; \boldsymbol{\theta}), \hat{\mathbf{y}} \in [0,1]^m$,
- perturb a test sample instance $\mathbf{x}_p = \mathbf{x} + \mathbf{p}$ (*de-identified sample*), so that the ML model classifies this perturbed sample as: $\hat{\mathbf{y}}_p = \boldsymbol{f}(\mathbf{x}_p; \boldsymbol{\theta})$, where ideally $\hat{\mathbf{y}}_p$ is very different from $\hat{\mathbf{y}}$, i.e.,
- the object is not recognized anymore.

- Typically $\mathbf{x}_p$ is 'similar' to $\mathbf{x}$ (it has imperceivable differences):
  - $\mathbf{x}_p$ has the same probability distribution with $\mathbf{x}$ and/or
  - $\|\mathbf{x} - \mathbf{x}_p\|$ is small.

Deep Generative Learning

- Generate fake data samples.
- ***Generative Adversarial Network*** (***GAN***): Generator-Discriminator pair.
  - The ***Generator*** network generates images from the training set and a noise vector, in order to fool the Discriminator.
  - The ***Discriminator*** network must distinguish between genuine samples and fake samples from the Generator.
  - They take alternate turns training. The Generator learns to produce more convincing fake samples.
- ***Variational Autoencoder*** (***VAE***)
  - The encoder part maps inputs onto a distribution with mean vector $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$.
  - Random vectors from that distribution result are decoded into fake samples.
  - Re-parameterization trick: isolate stochasticity using $\boldsymbol{r}_i \sim N(0,1)$ when generating random codes as $\boldsymbol{z}_i = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{r}_i$, which allows backpropagation.

Recurrent Neural Networks

- An RNN typically processes temporal information:
  - signals/ time sequences.
- It consists of recurrent layers.
- A recurrent network takes into consideration the stored information from the past layer outputs (hidden state), modeled as $\mathbf{h}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{h}(t-1), \boldsymbol{\theta})$.

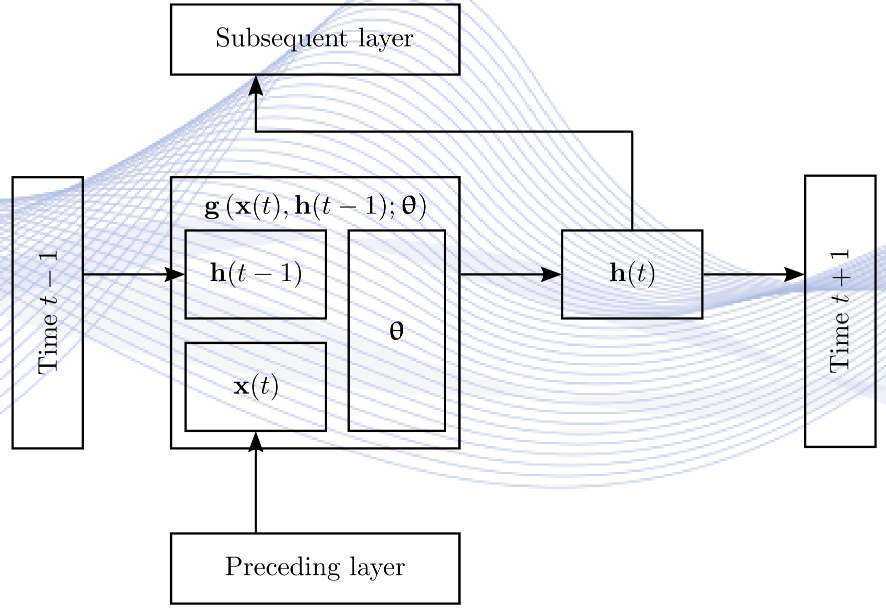
$\mathbf{x}(t)$: input instance.

$\mathbf{h}(t-1)$: hidden state.

φ: activation function.

$\hat{\mathbf{y}}(t) = \mathbf{h}(t)$: the output.

t: represents the time instance.



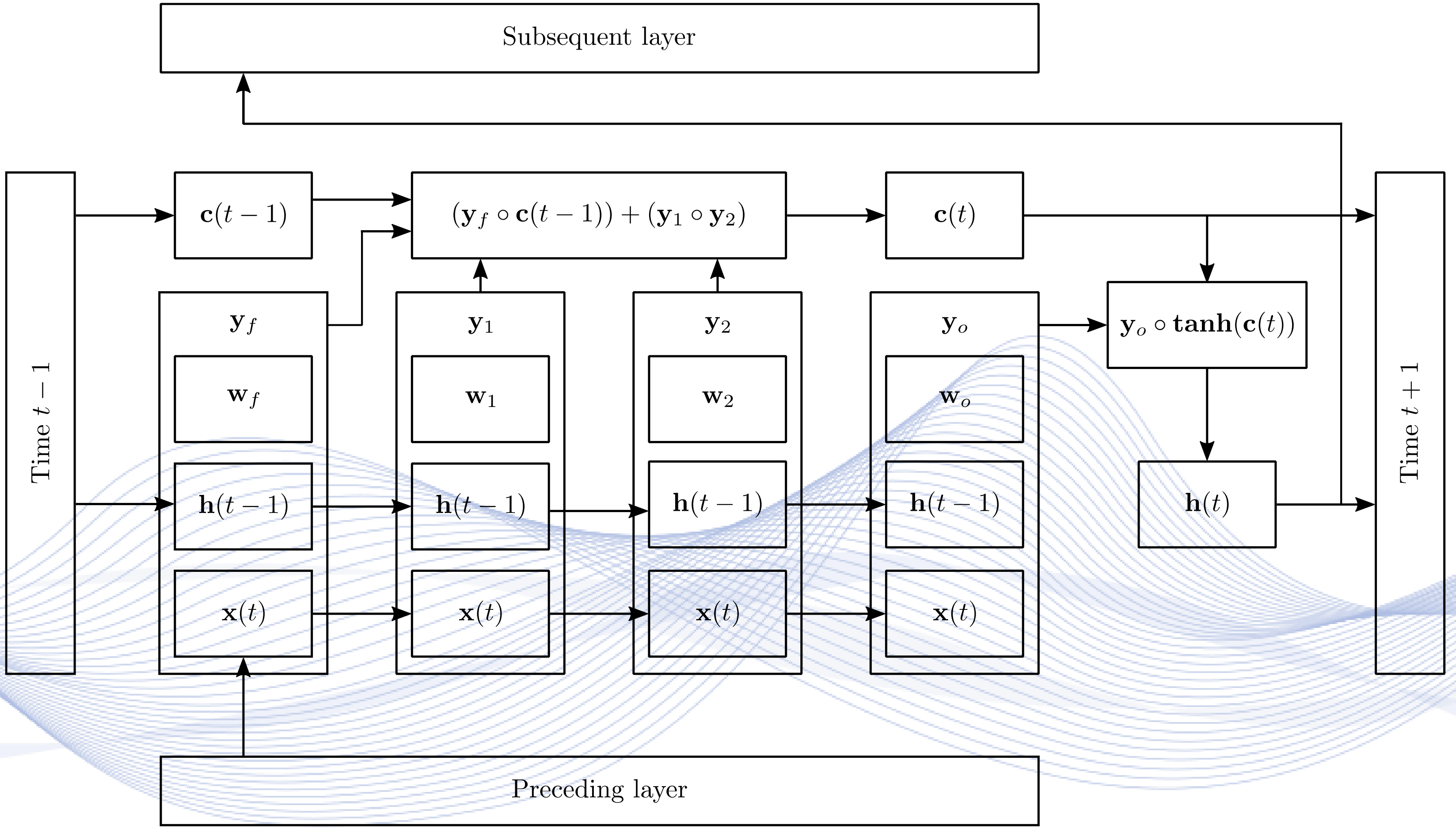
Recurrent Neural Networks

- Long Short Term Memory (LSTM) networks are a variation of RNNs.

- A LSTM layer is called a *cell* and contains a cell state vector in addition to the hidden state vector.

- The operation of a LSTM cell is dictated by three gates.
  - The **forget gate** removes information from the cell state through multiplication.
  - The **input gate** updates the cell state based on the input and hidden state.
  - The **output gate** combines the cell state and input into the next hidden state, which also serves as output.

Recurrent Neural Networks



Continual Learning

- The learning takes place after initial training, as new examples are encountered.

- An issue that emerges is that of catastrophic forgetting.

- Three categories of solutions to prevent catastrophic forgetting:

  - ***Periodically feed*** older data through the network.

  - Penalize larger deviations from the initial parameter vector.

  - ***Expand network architecture*** to learn new data without any changes to original.

Few-Shot Learning

Few-Shot Learning uses extremely few examples.

- Data-based approaches:
  - Include transformed samples in training set.
  - ***Use data generators***.
  - Add similar samples from larger dataset.
- Model-based approaches:
  - ***Use very simple model.***
  - Multi-task learning.
  - Dimensionality reduction.
- Fine tuning approaches:
  - ***Use pre-trained model***.
  - Very few iterations.
  - Only adjust small subset of parameters.

Transfer Learning. Domain Adaptation

Transfer Learning uses a pre-trained model for new ML task:

- Same data, different task, e.g., using an ImageNet classifier for image segmentation.

Domain Adaptation

- Different data, same task, e.g., using an ImageNet classifier to classify other image datasets.

- Different data, different task, e.g., using a pre-trained network as a feature extractor.

Knowledge Distillation

Knowledge Distillation uses a Larger network (teacher network) to train a smaller network (student network).

- Use the output of the teacher as target vectors for the student.

Introduction to Machine Learning

- Supervised learning
  - Classification/recognition/identification, Identity verification
  - Regression, Object detection
- Unsupervised learning
  - Clustering
  - Dimensionality reduction, data retrieval
- Semi-supervised learning
  - Label propagation
- Self-supervised learning
  - Autoencoders
- Reinforcement Learning
  - Curiosity driven Learning
- Neural Networks
  - Artificial Neural Networks, Deep Neural Networks
  - Adversarial Machine Learning
  - Generative Machine Learning
  - Temporal Machine learning (RNN)
- **Other topics**

Other topics

- ***Bio-inspired learning***: Several ML models and algorithms are inspired by biology (NNs, CNNs, genetic algorithms).
- ***Federated Learning***: Using a server to assign tasks to clients, which receive the model, train it on local data and send the results back to the server for consolidation.
- ***Ensemble Learning***: Combining weight outputs of different variations of a model, or different models, or models trained with different methods.

References

[BIS2006] C. M. Bishop. “Pattern Recognition and Machine Learning”, Springer, 2006.

[THE2010] S. Theodoridis, K. Koutroumbas, “Pattern Recognition”, Academic Press, 2010.

[GBCB16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[GPAM+14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’14, page 2672-2680, 2014. MIT Press.

[HS97] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735-1780, 1997.

References

[KMR15] J. Konecny, B. McMahan, and D. Ramage. Federated optimization: Distributed optimization beyond the datacenter. *CoRR*, 2015.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097-1105, 2012.

[KW19] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307-392, 2019.

[Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

References

[MJR15] S. Mohamed and D. Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 28*, pages 2125-2133. Curran Associates, Inc., 2015.

[NML17] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In Proceedings of the IEEE *Conference on Computer Vision and Pattern Recognition*, pages 670-679, 2017.

[Rok10] L. Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33(1-2):1-39, 2010.

[WYKN20] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1-34, 2020.

Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr

Artificial Intelligence &
Information Analysis Lab