

Homework 04

Submission Notices:

- Conduct your homework by filling answers into the placeholders in this file (in Microsoft Word format). Questions are shown in black color, *instructions/hints are shown in italics and blue color*, and *your content should use any color that is different from those*.
- After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,
 <StudentID-1>_<StudentID-2>_HW04.pdf (Student IDs are sorted in ascending order)
 E.g., **2112001_2112002_HW04.pdf**
and then submit the file to Moodle directly *WITHOUT* any kinds of compression (.zip, .rar, .tar, etc.).
- Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.
 1. Wrong file/filename format, e.g., not a pdf file, use "-" instead of "_" for separators, etc.
 2. Disorder format of problems and answers
 3. **Conducted not in English**
 4. Cheating, i.e., copying other students' works or letting other students copy your work.

Problem 1. (2pts) Aside from the types of learning introduced in the lectures (i.e., supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning), introduce two types of learning. For each type of learning, describe the **learning process** and **two applications**.

Please fill your answer in the table below

Types of learning	Description
1. Bayesian Learning	<p><i>Learning process: step-by-step overview of the learning process in Bayesian learning:</i></p> <ol style="list-style-type: none">1. Define the Prior Distribution: the prior distribution represents the initial beliefs or knowledge about the parameters of the model. The prior distribution is chosen based on existing information or assumptions. It can be a probability distribution over the model parameters, such as a Gaussian distribution, a beta distribution, or a Dirichlet distribution.2. Collect Training Data: which consists of input features and corresponding target values (if available). The training data is used to update the prior distribution and obtain the posterior distribution.3. Likelihood Function: it reflects the probability of observing the training data given the model parameters. It quantifies how likely the observed data is under different parameter settings.4. Compute the Posterior Distribution: it represents the updated beliefs about the model parameters after observing the training data. It is computed using Bayes' theorem, which

	<p>involves multiplying the prior distribution by the likelihood function and normalizing the result.</p> <p>5. Predictions and Inference: it can be used for making predictions and performing inference. Predictions are typically made by taking the expected value or a sample from the posterior distribution.</p> <p>6. Iterative Process: This iterative process allows the model to refine its beliefs and predictions as more data becomes available. Bayesian learning is often an iterative process. After making predictions, new data can be collected, and the posterior distribution can be updated again based on the combination of the prior distribution and the new data.</p> <p><i>Application:</i> some notable applications of Bayesian learning:</p> <p>1.Natural Language Processing: Bayesian learning has been utilized in natural language processing (NLP) tasks such as text classification, sentiment analysis, and language modeling. Bayesian methods can be used to model the distribution of words or phrases given specific classes or topics, allowing for probabilistic classification or generation of text.</p> <p>2. Bayesian Networks and Causal Inference: Bayesian networks are graphical models that represent probabilistic relationships between variables. They have been used for tasks such as causal inference, decision making, and risk analysis. Bayesian networks allow for the modeling of complex dependencies and the propagation of uncertainty through the network. They are used in diverse domains, including healthcare, finance, and environmental modeling.</p> <p>3. Regression Analysis: Bayesian learning can be used for regression analysis, where the goal is to predict a continuous target variable based on input features. By modeling the posterior distribution over the regression parameters, Bayesian learning provides a distribution of likely values for the target variable. This allows for more robust predictions and enables the quantification of uncertainty in the predictions. Bayesian regression is widely used in fields such as finance, economics, and environmental modeling.</p>
<p>2. Unsupervised Representation Learning</p>	<p><i>Learning process:</i> step-by-step overview of the learning process in unsupervised representation learning:</p> <p>1.Define the Architecture or Model: The first step is to choose or design an architecture or model that will be used for unsupervised representation learning. Common models include autoencoders, generative adversarial networks (GANs), and self-organizing maps (SOMs).</p>

	<p>2. Preprocessing and Data Preparation: Depending on the specific model and requirements, preprocessing steps such as normalization, dimensionality reduction, or feature extraction may be applied to the input data. The goal is to prepare the data in a suitable format for the chosen unsupervised learning model.</p> <p>3. Learn the Latent Representations: The next step is to train the chosen model on the unlabeled data to learn the latent representations or features. The model is presented with the input data, and its parameters are adjusted iteratively to minimize a chosen objective function.</p> <p>4. Reconstruction or Generation: The model is trained to reconstruct the input data from the learned latent representations. This reconstruction process helps to ensure that the learned representations capture the essential information in the data.</p> <p>5. Evaluation and Fine-tuning: After the initial training, the learned representations are evaluated based on specific criteria or downstream tasks. Evaluation can involve quantitative metrics, such as clustering performance or reconstruction error, as well as qualitative analysis to assess the quality and usefulness of the learned representations.</p> <p>6. Transfer Learning or Downstream Tasks: These representations can be used as input features for downstream tasks. The learned representations can be transferred and utilized in tasks such as classification, clustering, anomaly detection, or visualization, where labeled data may be limited or unavailable.</p> <p><i>Application:</i> some notable applications:</p> <p>1.Dimensionality Reduction and Visualization: Unsupervised representation learning methods, such as autoencoders and t-SNE (t-distributed Stochastic Neighbor Embedding), can be employed for dimensionality reduction and visualization of high-dimensional data. By learning lower-dimensional representations that preserve the important characteristics of the data, these methods enable visual exploration and analysis of complex datasets. They are particularly valuable in fields like genomics, image processing, and natural language processing.</p> <p>2. Pretraining for Supervised Learning: Unsupervised representation learning can serve as a pretraining step for supervised learning tasks. By learning meaningful representations from unlabeled data, such as raw text or images, unsupervised models can capture useful features that can be transferred to supervised models. Pretraining with unsupervised</p>
--	---

	<p>learning has shown significant improvements in tasks like sentiment analysis, object recognition, and document classification, especially when labeled data is limited.</p> <p>3. Generative Modeling and Data Synthesis: Unsupervised representation learning techniques, like generative adversarial networks (GANs) and variational autoencoders (VAEs), can generate new data samples that resemble the training data distribution. This capability is useful in generating synthetic data for data augmentation, data imputation, and data privacy protection. Generative modeling is applied in areas such as image synthesis, text generation, and drug discovery.</p>
--	---

Problem 2. (3pts) Consider the following training dataset, in which **Transportation** is the target attribute. Show calculations to choose an attribute for the **root node** of the ID3 decision tree.

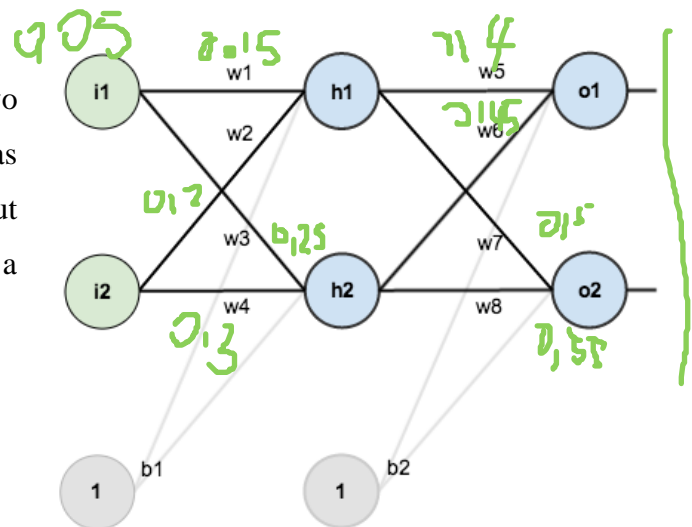
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Please fill your answer in the white cells of the following table

	Attribute values	Counts			Metric values		
		Bus	Car	Train	H	AE	IG
Whole	10	4	3	3	1.571		
Gender (0.5pt)	Female	1	2	2	1.522	1.446	0.125
	Male	3	1	1	1.371		
Car Ownership (0.5pt)	0	2	0	1	0.918	1.036	0.535
	1	2	1	2	1.522		
	2	0	2	0	0		
Travel Cost (0.5pt)	Cheap	4	0	1	0.722	0.361	1.210
	Expensive	0	3	0	0		
	Standard	0	0	2	0		
Income Level (0.5pt)	Low	2	0	0	0	0.875	0.696
	Medium	2	1	3	1.459		
	High	0	2	0	0		

Problem 3. (3pts) Given a neural network with two inputs, two hidden neurons, two output neurons, as shown below. Additionally, in the hidden and output layers, each of which will include a bias that has a constant output value of 1

- Learning rate 0.5
- Input values: $i1=0.05$ $i2=0.10$
- Target values: $t1=0.01$ $t2=0.99$
- Bias values: $b1=0.35$ $b2=0.60$
- Initial weight: $w1=0.15$ $w2=0.20$ $w3=0.25$ $w4=0.30$
 $w5=0.40$ $w6=0.45$ $w7=0.50$ $w8=0.55$



Present all calculations required to perform the backpropagation once (i.e. one forward pass and one backward pass) on the given neural network in the following cases

a) Ignore all biases (1.5pts)

Forward pass

	h1	h2	o1	o2
Sum	0.0275	0.0425	0.4326	0.5343
Sigmoid	0.507	0.5106	0.6065	0.6305

Backward pass

w1	w2	w3	w4	w5	w6	w7	w8
0.364	0.4136	0.5213	0.5714	0.15	0.2	0.25	0.3

b) Take into account all biases (1.5pts)

Forward pass

	h1	h2	o1	o2
Sum	0.3775	0.3925	1.105	1.224
Sigmoid	0.5932	0.5968	0.751	0.772

Backward pass

w1	w2	w3	w4	w5	w6	w7	w8	b1	b2
0.149	0.0199	0.249	0.299	0.358	0.408	0.511	0.561	0.35	0.6

Problem 4. (2pts) You are given the following tables, which represent the outcomes of some functions. The functions take two values x and y and output the outcomes of the operations. Please identify **at least two models** for each of the functions that are perfectly represent the functions for some choice of parameters. Justify your answer. Note: there are no constraints on the architecture (e.g, the number of neurons, activation function, or the best splitting criterion), and the depth of decision tree is 0-index.

a) (1pt) $f(x, y) = x \oplus y$.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

- ☐ A neural network with no hidden layer
- ☒ A neural network with a single hidden layer
- ☐ A decision tree of depth one
- ☒ A decision tree of depth two

Explanation:

A perceptron represented as $g(x, y) = \alpha(\beta(x, y))$, where $\beta(x, y) = ax + by + c$, and α is a monotonic activation function, cannot correctly classify the XOR function. This is proven by showing that if both (0, 0) and (1, 1) are correctly classified, it leads to a contradiction in the activation function α . Hence, a perceptron is incapable of representing XOR.

On the other hand, a neural network with a hidden layer has the capacity to represent XOR. The universal approximation theorem states that a single hidden layer network can represent any function given a sufficient number of neurons. For XOR, a small network with the representation $g(x, y) = \text{sgn}(B \cdot \text{sgn}(A \cdot (yx)))$ is sufficient, where $\text{sgn}(x)$ denotes the sign function and A and B are specific weight matrices. A decision tree of depth one can only split on a single variable, and since XOR depends on the values of both variables, it cannot be represented by a decision tree of depth one. However, a decision tree of depth two can represent any two-variable boolean function, including XOR.

b) (1pt) $f(x, y) = \neg(x \vee y)$

x	y	$\neg(x \vee y)$
0	0	1
0	1	0
1	0	0
1	1	0

- ☐ A neural network with no hidden layer
- ☐ A neural network with a single hidden layer
- ☐ A decision tree of depth one
- ☐ A decision tree of depth two

Explanation:

A single layer neural network $g(x, y)$ accurately classifies the function $f(x, y)$ using a specific formula. A two-layer neural network can also classify $f(x, y)$ correctly, with the second layer serving as an identity function. A decision tree with a depth of one is unable to represent $f(x, y)$ since it relies on two variables. However, a decision tree with a depth of two can represent $f(x, y)$ and other two-variable boolean functions.