Recitation 1
Background

10-301/10-601: Introduction to Machine Learning

01/19/2024

1 NumPy, Workflow, and Logging

NumPy Notebook

Workflow Presentation

Logging Notebook

2 Linear Algebra and Geometry

1. **Inner Product:** $\mathbf{u} = \begin{bmatrix} 6 & 1 & 2 \end{bmatrix}^T, \mathbf{v} = \begin{bmatrix} 3 & -10 & -2 \end{bmatrix}^T$, what is the inner product of $\mathbf{u}$ and $\mathbf{v}$? What is the geometric interpretation?

2. **Cauchy-Schwarz inequality** (Optional): Given $\mathbf{u} = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix}^T, \mathbf{v} = \begin{bmatrix} 3 & -1 & 4 \end{bmatrix}^T$, what is $||\mathbf{u}||_2$ and $||\mathbf{v}||_2$? What is $\mathbf{u} \cdot \mathbf{v}$? How do $\mathbf{u} \cdot \mathbf{v}$ and $||\mathbf{u}||_2 ||\mathbf{v}||_2$ compare? Is this always true?

3. **Matrix algebra.** Generally, if $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{B} \in \mathbb{R}^{N \times P}$, then $\mathbf{AB} \in \mathbb{R}^{M \times P}$ and $(AB)_{ij} = \sum_k A_{ik} B_{kj}$.

   Given $\mathbf{A} = \begin{bmatrix} 1 & 2 & 5 \\ 0 & 2 & 2 \\ 0 & 0 & 4 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 4 & -3 & 2 \\ 1 & 1 & -1 \\ 3 & -2 & 2 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$.

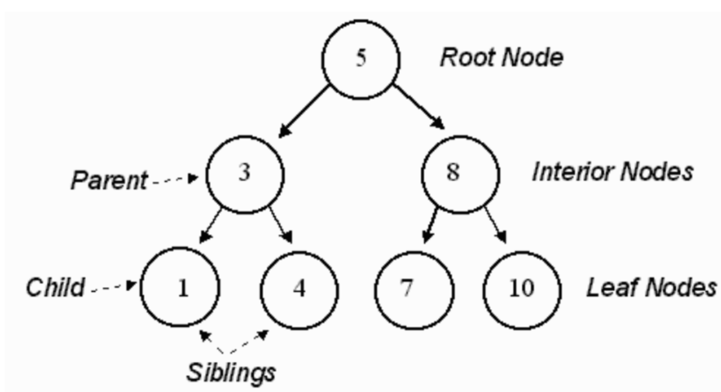
   - What is $\mathbf{AB}$? Does $\mathbf{BA} = \mathbf{AB}$? What is $\mathbf{Bu}$?
   - What is rank of $\mathbf{A}$?
   - What is $\mathbf{A}^T$?
   - Calculate $\mathbf{uv}^T$.
   - What are the eigenvalues of $\mathbf{A}$?

4. **Geometry:** Given a line $2x + y = 2$ in the two-dimensional plane,
   - If a given point (α, β) satisfies $2\alpha + \beta > 2$, where does it lie relative to the line?
   - What is the relationship of vector $\mathbf{v} = [2, 1]^T$ to this line?
   - What is the distance from origin to this line?

3 CS Fundamentals

1. For each (f, g) functions below, is $f(n) \in \mathcal{O}(g(n))$ or $g(n) \in \mathcal{O}(f(n))$ or both?
   - $f(n) = \log_2(n)$, $g(n) = \log_3(n)$
   - $f(n) = 2^n$, $g(n) = 3^n$
   - $f(n) = \frac{n}{50}$, $g(n) = \log_{10}(n)$
   - $f(n) = n^2$, $g(n) = 2^n$

2. Find the DFS Pre-Order, In-Order, Post-Order and BFS traversal of the following binary tree. What are the time complexities of the traversals?



3. You are Po, the Kung Fu Panda, whose favourite food is Steamed Pork Dumplings. So, he plans to steal them from village houses arranged in a straight line.

   There is also a special security system in place that alerts Master Shifu if there has been a theft in 2 consecutive houses.

dumplings	1	2	3	1	4	6	2	4

cache	0 (Steal)	1							
	1 (Skip)	0							

You need to help Po by employing the Bottom-Up Dynamic Programming approach to steal the maximum number of dumplings from the houses without alerting Master Shifu.

For any given house with index i ($i > 0$), you just need to write the logical statement to calculate the value after stealing from this house and after skipping this house.

```
def stealDumplings(dumplings: List[int]):
        n = len(dumplings)

        cache = {}

        cache[(0, 0)] = __[a]__ #value of cache stealing the first house
        cache[(1,0)] = __[b]__ #value of cache skipping the first house

        for i in range(1,n):
            #Stealing from this house
            cache[(0, i)] = __[c]__

            #Skipping this house
            cache[(1, i)] = __[d]__

        return __[e]__ # return final answer
```

4. Complete the following python code for a Binary Search Tree (BST). You only need to complete the class function `insert` that will take a an integer **value** and insert the value at the right position in the tree rooted at the calling node.
Note: In a BST, each node's value is strictly greater than the value of the node to its left (its left child) and less than or equal to the value of node to its right (its right child).

```
class BST:
    def __init__(self, val):
        self.val      = val
        self.leftNode = None
        self.rightNode = None

    def insert(self, value):
```

```
        if value __[a]__ self.val: # going left
            if self.leftNode is None: # left node is undefined
                self.leftNode = __[b]__
            else:
                self.leftNode = __[c]__ # recursive call

        else: # going right
            if self.rightNode is None: # right node is undefined
                self.rightNode = __[d]__
            else:
                self.rightNode = __[e]__ # recursive call

        return self
```

5. You are given a sorted list of integers *e.g.*, $[1, 2, 5, 7, 10, 13, 14, 15, 22]$. Complete the recursive python function below that will take this list, create a Binary Search Tree (BST) with the minimum height possible, and return the root node of that tree.

```
class BST:
    def __init__(self, val):
        self.val      = val
        self.leftNode = None
        self.rightNode = None

def minHeightBST(nums: List[int]):
    if len(nums)==0: # If nums is empty
        return __[a]__
    if len(nums) == 1: # If there's just one number
        return __[b]__

    mid  = __[c]__ # Find the middle element
    root = __[d]__ # Make the root node

    leftNums = __[e]__ # splice the left half of nums
    rightNums = __[f]__# splice the right half of nums

    root.leftNode = __[g]__ # Make a recursive call to assign the left node
    root.rightNode = __[h]__ # Make a recursive call to assign the right node

    return root
```

4 Calculus

1. If $f(x) = x^3 e^x$, find $f'(x)$.

2. If $f(x) = e^x$, $g(x) = 4x^2 + 2$, find $h'(x)$, where $h(x) = f(g(x))$.

3. If $f(x, y) = y \log(1 - x) + (1 - y) \log(x)$, $x \in (0, 1)$, evaluate $\frac{\partial f(x,y)}{\partial x}$ at the point $(\frac{1}{2}, \frac{1}{2})$.

4. Find $\frac{\partial}{\partial w_j} \mathbf{x}^T \mathbf{w}$, where $\mathbf{x}$ and $\mathbf{w}$ are M-dimensional real-valued vectors and $1 \leq j \leq M$.

5 Probability and Statistics

You should be familiar with some of the basic concepts of probability, such as events and outcomes in a sample space. In this class, however, we will mainly be dealing with random variables and their distributions.

Suppose we have two random variables, A and B, where A takes on values a_1, a_2 and B takes on values b_1, b_2. Let $P(A = a_1) = 0.5$ and $P(B = b_1) = 0.5$.

1. Suppose a_1 and b_2 are disjoint (mutually exclusive).
   - What is $p(a_1, b_2)$?
   - What is $p(a_1, b_1)$?
   - What is $p(a_1 \mid b_2)$?
2. Suppose instead that A, B are independent.
   - What is $p(a_1, b_2)$?
   - What is $p(a_1, b_1)$?

- What is $p(a_1 \mid b_2)$?

3. A student is looking at her activity tracker (Fitbit/Apple Watch) data and she notices that she seems to sleep better on days that she exercises. They observe the following:

Exercise	Good Sleep	Probability
yes	yes	0.3
yes	no	0.2
no	no	0.4
no	yes	0.1

- What is the $P(GoodSleep = yes \mid Exercise = yes)$?

- Why doesn't $P(GoodSleep = yes, Exercise = yes) = P(GoodSleep = yes) \cdot P(Exercise = yes)$?

- The student merges her activity tracker data with her food logs and finds that the $P(Eatwell = yes \mid Exercise = yes, GoodSleep = yes)$ is 0.25. What is the probability of all three happening on the same day?

4. What is the expectation of X where X is a single roll of a fair 6-sided die ($S = \{1, 2, 3, 4, 5, 6\}$)? What is the variance of X?

5. Imagine that we had a new die where the sides were $S = \{3, 4, 5, 6, 7, 8\}$. How do the expectation and the variance compare to our original dice?