Recitation HW1

Wednesday, January 18, 2023 4:06 PM

Recitation-1

Recitation 1
Background

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

1 NumPy, Workflow, and Logging

NumPy Notebook

Workflow Presentation

Logging Notebook

2 Linear Algebra and Geometry

1. Inner Product: [illegible] What is the inner product of u and v? What is the geometric interpretation?

$\langle u, v \rangle = \sum_i u_i v_i$ $\quad 6 \cdot 3 + 1 \cdot (-10) + 2 \cdot (-2) = 4$

$u \cdot v = \|u\|_2 \|v\|_2 \cos\theta$

2. Cauchy-Schwarz inequality (Optional): [illegible] What is $\|u\|_2$ and $\|v\|_2$? What is $u \cdot v$? How does $u \cdot v$ and $\|u\|_2\|v\|_2$ compare? Is this always true?

$\|u\|_2 = \sqrt{\sum_i u_i^2}$ $\quad \|u\|_2 = \sqrt{3^2+1^2+2^2} = 3.74$ $\quad \|v\|_2 = \sqrt{3^2+(-1)^2+4^2} = 5.10$

$u \cdot v \le \|u\|_2 \|v\|_2$ $\quad u \cdot v = 3 \cdot 3 + 1 \cdot (-1) + 2 \cdot 4 = 16$ $\quad \|u\|_2 \|v\|_2 = 19.074$

3. Matrix algebra. Generally, if $A \in \mathbb{R}^{n \times k}$ and $B \in \mathbb{R}^{k \times m}$, then $AB \in \mathbb{R}^{n \times m}$ and $(AB)_{ij} = \sum_k A_{ik} B_{kj}$.

Given A = [illegible], B = [illegible], u = [illegible], v = [illegible].

- What is AB? Does BA = AB? What is Bu? $AB = \begin{bmatrix} 21 & -11 & 10 \\ 8 & -2 & 2 \\ 12 & -8 & 8 \end{bmatrix}$ $BA \ne AB$ $Bu = \begin{bmatrix} 8 \\ -3 \\ 9 \end{bmatrix}$
- What is rank of A? 3
- What is A^T? $\rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ 5 & 2 & 4 \end{bmatrix}$
- Calculate uv^T. $\rightarrow \begin{bmatrix} 3 & 2 & 1 \\ 6 & 4 & 2 \\ 15 & 10 & 5 \end{bmatrix}$
- What are the eigenvalues of A?

$A \cdot v = \lambda \cdot v$

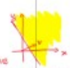
$\Rightarrow Av - \lambda I v = 0$

$\Rightarrow (A - \lambda I)v = 0$

$|A - \lambda I| = 0$

$\begin{vmatrix} 1-\lambda & 2 & 5 \\ 0 & 2-\lambda & 2 \\ 0 & 0 & 4-\lambda \end{vmatrix} = 0$ $\quad (1-\lambda) \cdot \begin{vmatrix} 2-\lambda & 2 \\ 0 & 4-\lambda \end{vmatrix} - 2 \cdot \begin{vmatrix} 0 & 2 \\ 0 & 4-\lambda \end{vmatrix} + 5 \cdot \begin{vmatrix} 0 & 2-\lambda \\ 0 & 0 \end{vmatrix} = 0$

$(1-\lambda)(2-\lambda)(4-\lambda) = 0$ $\quad \lambda = 1, 2, 4$

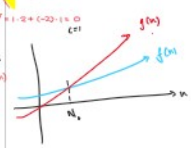


4. Geometry: Given a line $2x + y = 2$ in the two-dimensional plane. ($y = -2x + 2$)
- If a given point (α, β) satisfies $2\alpha + \beta > 2$, where does it lie relative to the line? above ($\beta > -2\alpha + 2$)
- What is the relationship of vector $v = [2, 1]^T$ to this line? orthogonal $\quad [1, -2]^T \cdot [2, 1]^T = 1 \cdot 2 + (-2) \cdot 1 = 0$
- What is the distance from origin to this line? (point to line dist, (0,0))

(α, β) $\quad Ax + By + C = 0$

$\frac{|A\alpha + B\beta + C|}{\sqrt{A^2+B^2}} \Rightarrow \frac{|0+0-2|}{\sqrt{2^2+1^2}} = \frac{2}{\sqrt{5}}$

"Big-O" notation — Asymptotic Upper Bound: $f(n) \in O(g(n))$ if for some constants c & N_0, $f(n) \le c\, g(n)$ $\forall n \ge N_0$



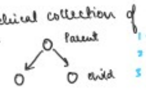
3 CS Fundamentals

1. For each $f(n)$, $g(n)$ functions below, is $f(n) \in O(g(n))$ (A) or $g(n) \in O(f(n))$ (B) or both (C)?
- [illegible] C: Both
- [illegible] $f(n) \in O(g(n))$: A
- [illegible] $g(n) \in O(f(n))$: B
- [illegible] $f(n) \in O(g(n))$: A

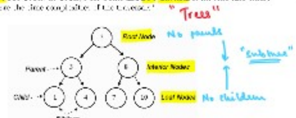
2. Find the DFS (Pre-Order, In-Order, Post-Order) and BFS traversals of the following binary tree. What are the time complexities of the traversals? ("Depth first search", "Breadth ...", "Trees")

→ hierarchical collection of nodes & edges.

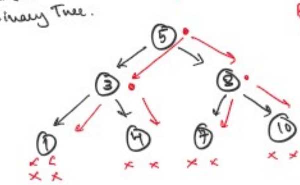
1 → Non-linear data structure
2 → NO LOOPS!
3 → # Nodes = N ⇒ # Edges = N-1



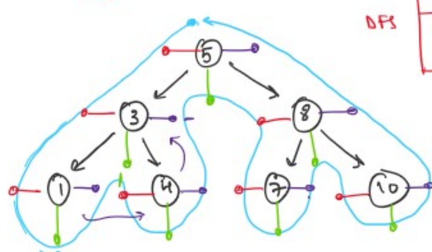
children ≤ 2 ⇒ Binary Tree.



BFS: 5 3 8 1 4 7 10



DFS:
- Pre-Order: 5 3 1 4 8 7 10 (N L R)
- In-Order: 1 3 4 5 7 8 10 (L N R)
- Post-Order: 1 4 3 7 10 8 5 (L R N)



3. You are Po, the King of 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	5	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≥1), you just need to write the logical statement to calculate the value after stealing from this house and after skipping this house.

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

    cache = {}

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

    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
```

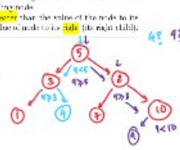
4. Complete the following python code for a Binary Search Tree (BST). You only need to complete the class function insert that will take an integer value and insert the value at the right position in the tree rooted at the calling node.

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):
```

(BST) / None



```
if value __(a)__ self.val: # going left          value < self.val
    if self.leftNode is None: # left node is undefined
        self.leftNode = __(b)__                  BST(value)
    else:
        self.leftNode = __(c)__ # recursive call self.leftNode.insert(value)
else: # going right                              value > self.val
    if self.rightNode is None: # right node is undefined
        self.rightNode = __(b)__                 BST(value)
    else:
        self.rightNode = __(c)__ # recursive call self.rightNode.insert(value)

return self
```

5. You are given a sorted list of integers e.g. [1,2,5,7,10,12,14,15,22]. 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)__                           None
    if len(nums) == 1: # If there's just one number
        return __(b)__                           BST(nums[0])

    mid = __(c)__ # Find the middle index        len(nums)//2
    root = __(d)__ # Make the root node          BST(nums[mid])

    leftNums = __(e)__ # splice the left half of nums    nums[:mid]
    rightNums = __(f)__ # splice the right half of nums  nums[mid+1:]

    root.leftNode = __(g)__ # Make a recursive call to assign the left node    minHeightBST(leftNums)
    root.rightNode = __(h)__ # Make a recursive call to assign the right node  minHeightBST(rightNums)

    return root
```


4 Calculus

1. If $f(x) = x^2 e^x$, 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 $\frac{\partial}{\partial x_j}\mathbf{x}^T\mathbf{w}$, where $\mathbf{x}$ and $\mathbf{w}$ are M-dimensional real-valued vectors and $1 \le j \le M$.

5 Probability and Statistics

You should be familiar with event notations for probabilities, i.e. $P(A \cup B)$ and $P(A \cap B)$, where A and B are binary events.

In this class, however, we will mainly be dealing with random variable notations, where A and B are random variables that can take on different states, i.e. a_1, a_2, and b_1, b_2, respectively. Below are some notation equivalents, as well as basic probability rules to keep in mind.

- $P(A = a_1 \cap B = b_1) = P(A = a_1, B = b_1) = p(a_1, b_1)$
- $P(A = a_1 \cup B = b_1) = \sum_{b \in B} p(a_1, b) + \sum_{a \in A} p(a, b_1) - p(a_1, b_1)$
- $p(a_1 \mid b_1) = \frac{p(a_1, b_1)}{p(b_1)}$
- $p(a_1) = \sum_{b \in B} p(a_1, b)$

1. Two random variables A and B, each can take on two values, a_1, a_2, and b_1, b_2, respectively. a_1 and b_1 are considered disjoint (mutually exclusive). $P(A = a_1) = 0.5$, $P(B = b_2) = 0.5$
   - What is $p(a_1, b_2)$?
   - What is $p(a_2, b_1)$?
   - What is $p(a_1 \mid b_2)$?

2. Now, instead, a_1 and b_1 are not disjoint, but the two random variables A and B are independent.
   - What is $p(a_1, b_2)$?
   - What is $p(a_2, b_1)$?
   - 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 dice ($S = \{1, 2, 3, 4, 5, 6\}$)? What is the variance of X?

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