FAANG最新AI相关面试题

Apple ML/AI Engineer Internship

1. 面试经历

> 记苹果 ML/AI Engineer Internship 面试。苹果的面试安排比较神奇，每个项目组招人过程是独立的，所以每个组都得从头面起。
>
> 每个组都是**一共三轮：**
>
> - 前两轮是 technical：2 coding 或者 1 coding + 1 design
> - 最后一轮是 team match：从 **behavior** 到 **system design**，问啥都有可能。
>
> **我不能说具体考了什么题**，但是苹果的 coding 题总体十分简单，侧重点更在于**你对在算法的理解和实际应用里的tradeoff**。
>
> 比如我被问了好几次，**如果这个算法在手机上运行，内存有限的情况下，该怎么办?**

针对上述问题，提供一些可能的解决方案：

a. **优化算法**：尽可能减少算法需要的内存。这可以通过减少数据结构的大小、简化算法逻辑、使用更高效的算法等方式实现。

b. **减少输入数据**：如果算法需要处理大量数据，可以尝试减少输入数据的规模，例如通过采样、分块等方式来处理数据。

c. **使用流式处理**：如果算法需要处理的数据太大，无法一次性加载到内存中，可以考虑使用流式处理。这种方式允许你逐步处理数据，而不是一次性处理所有数据。

d. **利用硬件加速**：一些手机可以使用硬件加速，例如GPU等，可以加速某些算法的处理速度。如果你的算法可以利用硬件加速，可以考虑使用这种方式。

e. **分布式计算**：如果你的算法需要处理的数据量太大，无法在单个手机上处理，可以考虑使用分布式计算。这种方式允许你将任务分配给多个手机或服务器，并将结果合并到一起。

Amazon（AS、MLE）面试题

1. 找出一个string中的第一个重复字母

> OA：
>
> 因为面的是AS，所以coding不是很难，其中一道是经典5题中的一个（在下面），另外一个也不难，大概记得问的是找出一个string中的第一个重复字母。

> VO：
>
> 电面是两个面试官，问的问题还是挺难的。首先是根据简历让我介绍一个project然后问了一些细节，这部分还好。之后就开始brainstorm问了许多design的问题，然后逐渐深入，同时问了涉及到的DL和ML的基础知识（感觉是同时照顾到了research的广度和深度）。
>
> BQ同样也是两个问题，让我说一些个人经历（这部分我个人准备的不是很好，答的时候interviewer引导了我好多次最终才满意），这部分大家还是要好好准备。
>
> 主要的takeaway是VO还是要尽量反应快一些，不要有太多0交流的思考时间，而且要自信不要怕出错。如果实在不会可以让interviewer给一些提示，观察对方的状态避免自说自话的情况。最后leadership还是比较重要的，可以面试之前多准备一些experience然后灵活运用。

2. 经典5题（一）

> - Anagrams
>
> A list of unique product IDs needs to be extracted from a database of products available on Amazon's website.In the database, multiple versions of the products IDs exist such that the order of the characters varies (e.g., code and ecod are the same product). To create the unique list, all anagrams must be removed. Two strings are anagrams if they are permutations of each other. In other words, both strings have the same size and the same characters. For example, ""aaagmnrs"" is an anagram of ""anagrams"". Given a list of strings, remove each string that is an anagram of an earlier string,then return the remaining list in sorted order. Write an algorithm that returns a list of strings after removing anagram string in sorted order.
>
> Write an algorithm that returns a list of strings after removing anagram string in sorted order.
>
> **Function Description**
>
> Complete the function removeAnagram in the editor below.
>
> removeAnagram has the following parameter:
>
> textList, a list of strings representing product IDs.
>
> Returns
>
> String Array
>
> **Constraints**
>
> - 0 <= num <= 1000
> - 0 <= m <= num; where m represents the size of output list
> - 1 <= length of text[i] <= 1000
> - 0 <= i < num
>
> **Note:**Each string text[i] is made up of characters in the range ASCII[a-z]

Input Format For Custom Testing

The first line contains an integer, n, denoting the number of elements in textList.

Each line i of the n subsequent lines (where 0 i < n) contains an STRING.

Sample Case 0

Sample Input For Custom Testing

```
5 -> textList[] size n = 5
1 -> textList = ["code", "doce", "ecod", "framer", "frame"]
1
1
1
2
```

Sample Output

```
code
frame
framer
```

Explanation

"code" and "doce" are anagrams. Remove "doce" from the list and keep the first occurrence "'code" in the list.

"code" and "ecod" are anagrams. Remove "ecod" from the list and keep the first occurrence "code" in the list

"code" and "framer" are not anagrams. Keep both strings in the list.

"framer" and "frame" are not anagrams due to the extra 'r' in 'framer' Keep both strings in the list

Order the remaining strings in ascending order: ["code","frame","framer"].

So the output is ["code","frame","framer"].

3. 经典5题（二）

- Overlap array 相似题 https://www.lintcode.com/problem/156/

A Supply Chain Manager at an Amazon warehouse is reviewing the logs of when trucks arrived at and departed from their warehouse. Please help them with their review by completing the

> following challenge: Given a collection of time intervals, [start, end], merge and return the overlapping intervals sorted in ascending order of their start times.
>
> **Example**
>
> Intervals = [[7,7],[2,3] [6, 11][1,2]]
>
> The interval [1, 2] merges with [2, 3]while [7, 7] merges with [6, 11]. There are no more overlapping intervals. The answer is [[1, 3] [6, 11]]
>
> **Function Description**
>
> Complete the function getMergedintervals in the
>
> editor below.
>
> getMergedIntervals has the following parameter(s):
>
> int intervals[[n][2]]: the time intervals
>
> **Returns**
>
> int[][2]: the merged intervals in sorted order
>
> **Constraints**
>
> - 1 <= n <= 10^5
> - 1 <= intervals[i][2] <= 10^9
> - intervals[i][0] <= intervals[i][1] for all i.

4. 经典5题（三）

> - knapsack problem
>
> Your team at Amazon is building a quiz-style application to help students prepare for certification exams.Each quiz module tests one or more subjects and limits the number of answers students can provide.You have been asked to examine the impact of this limit on the ability of students to "pass" certain subjects within quiz modules. To do this, please review and solve the following: Imagine a student has already answered *answered[i]* questions in each of the the i^th subjects,and still has time to answer a total of q more questions overall. For each i^th subject,the number of questions answered has to be at least *needed[i]* in order to pass.Determine the maximum number of subjects the student can pass if the *q* additional answered questions are optimally distributed among the subjects.

For example, consider that there are n=2subjects and needed=[4,5] answered questions,respectively,to pass.lmagine the student has answered answered=[2,4] questions in the two subjects so far,and can answer another q=1 questions across all subjects combined.In that case, the best outcome is to answer an additional question in the second subject,in order to pass it,as 2 more answers are required to pass the first subject.The maximum number of subjects that can be passedis 1.

Function Description

Complete the function *maxSubiectsNumber* in the editor below.The function must return an integer that represents the maximum number of subjects that can be passed.

maxSubjectsNumber has the following parameter(s):

answered[answered[0],... answered[n-1]]:an array of integers

needed[needed[0],...needed[n-1]]: an array of integers

q: an integer

Constraints

1n105

0 <= answered[i], needed[i],q <= 10^9

Input Format For Custom Testing

The first line contains an integer, *n*, the number of elements *answered*,and the number of subjects.

Each of the next n lines contains one integer, answered[i], the number questions already answered for the i^th subject.

The next line contains the integer,*n*, the number of subjects.

Each of the next n lines contains one integer,*needed[i]*, the number of questions answered needed to pass the i^th subject

The next line contains an integer, *q*, the total number of additional questions the student can answer across all subjects.

Sample Case

Sample Input For Custom Testing

```
3
24
27
```

```
0
3
51
52
100
100
```

Sample Output

```
2
```

Explanation

Here answered=[24,27,0] and needed=[51,52,100]. The additional answers needed to pass are [27,25,100]. The best distribution is at least 27+25=52 questions among the first two subjects.It would take all q=100 questions to pass the third subject.

5. 经典5题（四）

- divide problem

You've been asked to help your teammate with the following assignment.Determine the factors of a number, (i.e.,all positive integer values that evenly divide into a number) and then return the p^th element of the list, sorted ascending. If there is no p^th element, return 0.

Example

n = 20

P = 3

The factors of 20 in ascending order are {1,2,4,5,10,20}.Using 1- based indexing,if p=3,then 4 is returned.If p>6, 0 would be returned.

Function Description

Complete the function *pthFactor* in the editor below.

pthFactor has the following parameter(s):

int n: the integer whose factors are to be found

int p: the index of the factor to be returned

Returns:

int: the long integer value of the p^th integer factor of n or, if there is no factor at that index, then 0 is returned

Constraints

1 <= n <= 10^15

1 <= p <= 10^9

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the number to factor.

The second line contains an integer *p*, the 1-based index of the factor to return.

Sample Case 0

Sample Input 0

```
STDIN    Function
-----    --------
10    ->     n=10
3     ->     p=3
```

Sample Output 0

```
5
```

Explanation 0

Factoring n = 10 results in {1.2,5,10}. Return the p=3^rd factor, 5, as the answer.

Sample Case 1

Sample Input 1

```
STDIN    Function
-----    --------
10    ->     n=10
5     ->     p=5
```

Sample Output 1

```
1 0
```

Explanation 1

Factoring n=10 results in {1,2,5,10}.There are only 4 factors and p=5,therefore 0 is returned as the answer.

6. 经典5题（五）

- throttling gateway

Non-critical requests for a transaction system are routed through a throttling gateway to ensure that the network is not choked by non-essential requests

The gateway has the following limits:

- The number of transactions in any given second cannot exceed 3.
- The number of transactions in any given 10 second period can not exceed 20. A ten-second period includes all requests arriving from any time max(1, T-9) to T (inclusive of both) for any valid time T
- The number of transactions in any given minute can not exceed 60. Similar to the above, 1 minute is from max(1,T-59) to T.

Any request that exceeds any of the above limits will be dropped by the gateway. Given the times at which different requests arrive sorted ascending, find how many requests will be dropped

Note: Even if a request is dropped, it is still considered for future calculations. Although, if a request is to be dropped due to multiple violations, it is still counted only once

Example

n=27

requestTime=[1,1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,7,11,11,11,11]

- Request 1-Not Dropped
- Request 1 -Not Dropped.
- Request 1-Not Dropped
- Request 1-Dropped.At most 3 requests are allowed in one second

- No request will be dropped till 6 as all comes at an allowed rate of 3 requests per second and the 10-second clause is also not violated.
- Request 7-Not Dropped.The total number of requests has reached 20 now
- Request 7-Dropped.At most 20 requests are allowed in ten seconds
- Request7-Dropped.At most 20 requests are allowed in ten seconds
- Request 7-Dropped.At most 20 requests are allowed in ten seconds.Note that the 1-second limit is also violated here
- Request 11-Not Dropped.The 10-second window has now become 2 to 11. Hence the total number of requests in this window is 20 now.
- Request 11-Dropped.At most 20 requests are allowed in ten seconds
- Request 11-Dropped.At most 20 requests are allowed in ten seconds
- Request 11-Dropped.At most 20 requests are allowed in ten seconds.Also,at most 3 requests are allowed per second.

Hence, a total of 7 requests are dropped

Function Description

Complete the droppedRequests function in the editor below

droppedRequests has the following parameter(s):

int requestTime[n]:an ordered array of integers that represent the times of various requests

Returns

Int: the total number of dropped requests.

Constraints

1 <= n <= 10^6

1 <= requestTime[i] <= 10^9

It is hard to understand the question, but after knowing the meaning, it an easy question.
Time Complexity: O(N)
Space Complexity: O(N)

```
def droppedRequests(requestTime):
# Write your code here
    dropped = 0
```

```
#       to keep track of already dropped we will use dictionary
    prevDrop = {}

    for i in range(len(requestTime)):
        if i > 2 and requestTime[i] == requestTime[i-3]:
            if requestTime[i] not in prevDrop or prevDrop[requestTime[i]] != i
                prevDrop[requestTime[i]] = i
                dropped += 1

        elif i > 19 and requestTime[i] - requestTime[i-20] < 10:
            if requestTime[i] not in prevDrop or prevDrop[requestTime[i]] != i
                prevDrop[requestTime[i]] = i
                dropped += 1

        elif i > 59 and requestTime[i] - requestTime[i-60] < 60:
            if requestTime[i] not in prevDrop or prevDrop[requestTime[i]] != i
                prevDrop[requestTime[i]] = i
                dropped += 1

    return dropped
```

Amazon（MLE）

1. Merge Intervals

原题：https://www.lintcode.com/problem/156/

2. Remove anagram

相似题：https://www.lintcode.com/problem/158/

Meta（MLE）

> **面试经历**
>
> 上周面了 Meta 的 SDE 实习，烙印两道题，第一题是给一个 linkedlist （升序），在其中插入一个元素。第二题是计算器。感觉烙印很冷漠，基本上就是直接 copy 题目然后让我自己看，不给解释，感觉没有什么沟通，很难。

1. 升序链表插入元素

给一个linkedlist （升序），在其中插入一个元素。

> 插入一个元素到升序的链表中，需要找到插入的位置，使得插入后的链表仍然保持升序。

> 具体的插入方法是，在找到合适的位置后，将新元素的前一个节点的 next 指向新元素，新元素的 next 指向原先的后一个节点。
>
> 以下是Python代码示例：
>
> ```
> class ListNode:
>     def __init__(self, val=0, next=None):
>         self.val = val
>         self.next = next
>
> def insert_node(head, insert_val):
>     new_node = ListNode(insert_val)
>     if not head: # 如果链表为空，直接返回新节点
>         return new_node
>     if insert_val < head.val: # 如果新节点应该插入到链表的头部
>         new_node.next = head
>         return new_node
>     cur_node = head
>     while cur_node.next and insert_val > cur_node.next.val: # 找到新节点应该
>         cur_node = cur_node.next
>     new_node.next = cur_node.next
>     cur_node.next = new_node
>     return head
>
> ```

2. 计算器实现

> 下面代码所实现的计算器**使用了栈来存储数字和操作符**，并使用了**逆波兰表达式**来进行计算。
>
> ```
> import re
>
> # 定义操作符的优先级
> precedence = {'+': 1, '-': 1, '*': 2, '/': 2}
>
> # 将中缀表达式转换为后缀表达式
> def infix_to_postfix(expression):
>     # 将中缀表达式分割为数字和操作符
>     tokens = re.findall(r'(\d+|\S)', expression)
>     stack = []
>     output = []
>     for token in tokens:
>         # 如果是数字，则将其添加到输出列表中
>         if token.isdigit():
> ```

```python
            output.append(token)
        # 如果是左括号，则将其添加到栈中
        elif token == '(':
            stack.append(token)
        # 如果是右括号，则将栈中的操作符依次弹出，直到遇到左括号为止，并将这些操作符
        elif token == ')':
            while stack and stack[-1] != '(':
                output.append(stack.pop())
            stack.pop()
        # 如果是操作符，则将栈中的操作符依次弹出，直到遇到优先级低于或等于当前操作符
        else:
            while stack and precedence.get(stack[-1], 0) >= precedence.get(
                output.append(stack.pop())
            stack.append(token)
    # 将栈中的剩余操作符添加到输出列表中
    while stack:
        output.append(stack.pop())
    return ' '.join(output)

# 计算后缀表达式的值
def evaluate_postfix(expression):
    stack = []
    tokens = expression.split()
    for token in tokens:
        # 如果是数字，则将其压入栈中
        if token.isdigit():
            stack.append(float(token))
        # 如果是操作符，则弹出栈顶的两个数字，并根据操作符进行计算，将结果压入栈中
        else:
            num2 = stack.pop()
            num1 = stack.pop()
            if token == '+':
                stack.append(num1 + num2)
            elif token == '-':
                stack.append(num1 - num2)
            elif token == '*':
                stack.append(num1 * num2)
            elif token == '/':
                if num2 == 0:
                    raise ValueError('除数不能为0')
                stack.append(num1 / num2)
    # 返回栈顶元素作为计算结果
    return stack[-1]

while True:
    try:
        expression = input('请输入表达式: ')
        postfix_expression = infix_to_postfix(expression)
        result = evaluate_postfix(postfix_expression)
        print('计算结果: ', result)
        break
    except Exception as e:
        print('计算错误: ', e)

```

> 这个计算器代码中，我们首先定义了一个**操作符的优先级字典**，用于在后面的转换和计算过程中进行比较。然后，我们定义了一个**将中缀表达式转换为后缀表达式的函数** `infix_to_postfix()` ，这个函数使用了栈来存储操作符，并按照操作符的优先级将操作符添加到输出列表中。转换过程中，我们使用了正则表达式来将中缀表达式分割成数字和操作符，使用了左右括号来处理优先级和括号的情况。
>
> 然后，我们定义了一个**计算后缀表达式的函数** `evaluate_postfix()` ，这个函数也使用了栈来存储数字，并按照操作符依次弹出栈顶的两个数字进行计算，并将结果压入栈中。计算过程中，我们使用了split函数将后缀表达式分割成数字和操作符，使用了 `if-elif` 语句来处理不同的操作符情况。
>
> 最后，我们在一个 `while` 循环中不断提示用户输入表达式，并将表达式转换为后缀表达式，然后计算出结果并输出。如果表达式有误，我们将输出错误信息。用户可以在循环中进行多次计算，直到输入 exit 结束程序。
>
> **如果需要在计算器中支持其他高级数学运算和函数，可以考虑在 `infix_to_postfix()` 函数中添加相应的操作符和函数。**

Netflix（MLE）

1. 面经（一）

> Netflix ML intern 面试一共三轮：
>
> a. 第一轮 OA 比较简单，平时刷题多的同学可能会遇到原题
>
> b. 第二轮 ML 快问快答和过 CV
>
> c. 第三轮群面准备 45min 的 presentation，最后 15min 是 manager 的 BQ

2. 面经（二）

> 刚接到 Netflix NG Reseach Scientist面试（OA + 一轮电面 + 三轮onsite + 两轮HM面）中的三轮onsite面。

> 第一轮面了很多machine learning 八股文，总结下就是**任何关于 machine learning 的都会问到**(overfitting, logistic regression, data imbalance...)，建议找本教科书看看。

Google Brain Research（MLE）

1. 面经

> 前期会有两轮技术电面，**问题很多而且语速非常快。考察很多知识点。需要有相当的知识储备。**
>
> 十二月一号联系我说有两个 team 对我比较感兴趣。一个是 NYC 的 NLP 组 一个是 MTV 的 Search 组。
>
> NLP 组主要是做 NLG 的 evaluation。Search 组主要是 safety 方面的。都约了视频面试。
>
> 一月初又被告知有两个组对我感兴趣，都是 MTV 的组。一个是 NLX 组一个是 Brain Team 的。
>
> NLX 组的是和产品组合作的 做类似搜索体验优化的。Brain Team 是做大模型的。都约了视频面试。
>
> 最后决定去 Brain Team。

Google（MLE）

1.What do you understand by Precision and Recall?

Recall is the number of relevant documents retrieved by a search divided by the total number of the existing relevant documents, while precision is the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search

2.What is a Confusion Matrix?

The confusion is a 26 by 26 matrix with the probability of each reaction to each stimulus. This explains the name and matches the use of machine learning today.

3.What is the difference between inductive and deductive learning?

The main difference between inductive and deductive reasoning is that inductive reasoning aims at developing a theory while deductive reasoning aims at testing an existing theory. Inductive reasoning moves from specific observations to broad generalizations, and deductive reasoning the other way around.

4.How is KNN different from K-means clustering?

K-means is an unsupervised learning algorithm used for the clusteringproblem whereas KNN is a supervised learning algorithm used for classification and regression problems. This is the basic difference between K-means and KNN algorithms. It makes predictions by learning from the past available data.

5.What is the ROC curve and what does it represent?

An ROC curve is a graph showing the performance of a classification model at all the classification thresholds. This curve plots two parameters: TruePositive Rate. False Positive Rate.

6.What'sthe difference between Type I and Type II error?

Type 1 error, in statistical hypothesis testing, is the error caused by rejecting a null hypothesis when it is true. Type II error is the error that occurs when the nullhypothesis is accepted when it is not true. Type I error is equivalent to a false positive. Type IIerror is equivalent to a false negative.

7.Is it better to have too many false positives or too many falsenegatives? Explain.

In medical testing, false negatives may provide a falsely reassuring message to patients and physicians that the disease is absent, when it
is actually present. This sometimes leads to inappropriate or inadequate treatment of both the patient and their disease. So, it is desired to have too many false positive.

8.Which is more important to you-model accuracy or model performance?

The accuracy is extremely critical, even if the models would take minutes or hours to make a prediction. Other applications require real time performance,even if this comes at a cost of accuracy.

9.What's the trade-off between bias and variance?

The bias-variance tradeoff refers to a decomposition of the prediction error in the machine learning as the sum of a bias and a variance term. An example of the bias-variance tradeoff in practice.

10.What is the difference between supervised and unsupervised machine learning?

Supervised learning algorithms are trained using labeled data. UInsupervised learning algorithms are trained using unlabeled data. supervised learning, input data is provided to the model along with the output. unsupervised learning, only input data is provided to the model.

11.How is KNN different from k-means clustering?

KNN represents a supervised classification algorithm that will give new data points according to the k number or the closest data points, whilek-means clustering is an unsupervised clustering algorithm that gathers and groups data into k number of clusters.

12.Explain how a ROC curve works.

A ROC curve is constructed by plotting the true positive rate against the false positive rate. A discrete classifier that returns only the predicted class gives a single point on the ROC space.

13.What is Bayes' Theorem? How is it useful in a machine learning context?

Bayes Theorem is a useful tool for applied machine learning. It provides a way of thinking about the relationship between data and a model. A machine learning algorithm or model is a specific way of thinking about the structured relationships in the data.

14.Why is "Naive" Bayes naive?

Naive Bayes is called naive because it assumes that each input variable is independent. This is a strong assumption and unrealistic for real data; however,that the technique is very effective on a large range of complex problems.

15.Explain the difference between L1 and L2 regularization

The main intuitive difference between the L1 and L2 regularization is that L1 regularization tries to estimate the median of the data while L2 regularization tries to estimate the meaning of the data to avoid overfitting. That value will also be the median of the data distribution mathematically.

16.What's your favorite algorithm, and can you explain it tome in less than a minute?

Hands down logistic regression (with many bells and whistles like stochastic gradient descent, feature hashing and penalties).

17.What's the difference between Type I and Type II errors?

Type 1 error, in statistical hypothesis testing, is the error caused by rejecting a null hypothesis when it is true. Type II error is the error that occurs when the null hypothesis is accepted when it is not true. Type I error is equivalent to false positive.Type II error is equivalent to a false negative.

1.What is the difference between Machine Learning and Deep Learning

Machine Learning forms a subset of Artificial Intelligence, where we use statistics and algorithms to train machines with data, thereby, helping them improve with experience. Deep Learning is a part of Machine Learning, which involves mimicking the human brain in terms of structures called neurons, thereby forming neural networks.

2.Whatisaperceptron?

A perceptron is similar to the actual neuron in the human brain.It receives inputs from various entities and applies functions to these inputs, which transform them into output.

A perceptron is mainly used to perform binary classification where it sees an input, computes functions based on the weights of the input, and outputs the required transformation.

3.How is Deep Learning better than Machine Learning?

Machine Learning is powerful in a way that it is sufficient to solve most of the problems. However, Deep Learning gets an upper hand when it comes to working with data that has a large number of dimensions. With data that is large in size, a Deep Learning model can easily work with it as it is built to handle this.

4.What are some of the most used applications of Deep Learning?

Deep Learning is used in a variety of fields today. The most used ones are as follows:

- Sentiment Analysis
- Computer Vision
- Automatic Text Generation
- Object Detection * Natural Language Processing
- Image Recognition

5.What is the meaning of overfitting?

Overfitting is a very common issue when working with Deep Learning. It is a scenario where the Deep Learning algorithm vigorously hunts through the data to obtain some valid information. This makes the Deep Learning model pick up noise rather than useful data causing very high variance and low bias. This makes the model less accurate, and this is an undesirable effect that can be prevented.

6.What are activation functions?

Activation functions are entities in Deep Learning that are used to translate inputs into a usable output parameter. It is a function that decides if a neuron needs activation or not by calculating the weighted sum on it with the bias. Using an activation function makes the model output non-linear.There are many types of activation functions:

- ReLU
- Softmax
- Sigmoid
- Linear
- Tanh

7.Why is Fourier transform used in Deep Learning?

Fourier transform is an effective package used for analyzing and managing large amounts of data present in a database. It can take in real-time array data and process it quickly. This ensures that high efficiency is maintained and also makes the model more open to processing a variety of signals.

8.What is a Boltzmann machine?

A Boltzmann machine is a type of recurrent neural network that uses binary decisions, alongside biases, to function. These neural networks can be hooked up together to create deep belief networks, which are very sophisticated and used to solve the most complex problems out there.

9.What are some of the advantages of using TensorFlow?

TensorFlow has numerous advantages, and some of them are as follows:

- High amount of flexibility and platform independence
- Trains using CPU and GPU
- Supports auto differentiation and its features
- Handles threads and asynchronous computation easily

- Open-source
- Has a large community

10.What is a computational graph in Deep Learning?

- A computation graph is a series of operations that are performed to take inputs and arrange them as nodes in a graph structure. It can be considered as a way of implementing mathematical calculations into a graph. This helps in parallel processing and provides high performance in terms of computational capability.

11.What is CNN?

CNNs are convolutional neural networks that are used to perform analysis of images and visuals. These classes of neural networks can input a multi-channel image and work on it easily.

These Deep Learning questions must be answered in a concise way. So make sure

to understand them and revisit them if necessary.

12.What are the various layers present in a CNN?

There are four main layers that form a convolutional neural network:

- Convolution: These are layers consisting of entities called filters that are used as parameters to train the network
- ReLu: It is used as the activation function and is always used with the convolution layer.
- Pooling: Pooling is the concept of shrinking the complex data entities that form after convolution and is primarily used to maintain the size of an image after shrinkage.
- Connectedness: This is used to ensure that all of the layers in the neural network are fully connected and activation can be computed using the bias easily.

13.What is an RNN in Deep Learning?

RNNs stand for recurrent neural networks, which form to be a popular type of artificial neural network. They are used to process sequences of data, text, genomes, handwriting, and more. RNNs make use of backpropagation for the training requirements.

14.What is a vanishing gradient when using RNNs?

Vanishing gradient is a scenario that occurs when we use RNNs. Since RNNs make use of backpropagation, gradients at every step of the way will tend to get smaller as the network traverses through backward iterations. This equates to model learning very slowly, thereby, causing efficiency problems in the network.

15.What is exploding gradient descent in Deep Learning?

Exploding gradients are an issue causing a scenario that clumps up the gradients.

This creates a large number of updates of the weights in the model when training.

The working of gradient descent is based on the condition that the updates are small and controlled. Controlling the updates will directly affect the efficiency of the model.

16.What is the use of LSTM?

LSTM stands for long short-term memory. It is a type of RNN that is used to sequence a string of data. It consists of feedback chains that give it the ability to perform like a general-purpose computational entity.

17.Where are autoencoders used?

Autoencoders have a wide variety of usages in the real world. The following are some of the popular ones:

- Adding color to black-white images
- Removing noise from images
- Dimensionality reduction
- Feature removal and variation

18.What are the types of autoencoders?

There are four main types of autoencoders:

- Deep autoencoders
- Convolutional autoencoders
- Sparse autoencoders
- Contractive autoencoders

19.What is a Restricted Boltzmann Machine?

A Restricted Boltzmann Machine, or RBM for short, is an undirected graphical model that is popularly used in Deep Learning today. It is an algorithm that is used to perform:

- Dimensionality reduction
- Regression
- Classification
- Collaborative filtering
- Topic modeling

20.What are some of the limitations of Deep Learning?

There are a few disadvantages of Deep Learning as mentioned below:

- Networks in Deep Learning require a huge amount of data to train well.
- Deep Learning concepts can be complex to implement sometimes.
- Achieving a high amount of model efficiency is difficult in many cases.

These are some of the vital advanced deep learning interview questions that you have to know about!

21.What are the variants of gradient descent?

There are three variants of gradient descent as shown below:

- Stochastic gradient descent: A single training example is used for calculation of gradient and for updating parameters.
- Batch gradient descent: Gradient is calculated for the entire dataset, and parameters are updated at every iteration.
- Mini-batch gradient descent: Samples are broken down into smaller-sized batches and then worked on as in the case of stochastic gradient descent.

22.Why is mini-batch gradient descent so popular?

Mini-batch gradient descent is popular as:

- It is more efficient when compared to stochastic gradient descent.
- Generalization is done by finding the flat minima.
- It helps avoid the local minima by allowing the approximation of the gradient for the entire dataset.

23.What are deep autoencoders?

Deep autoencoders are an extension of regular autoencoders. Here, the first layer is responsible for the first-order function execution of the input. The second layer will take care of the second-order functions, and it goes on. Usually, a deep autoencoder is a combination of two or more symmetrical deep belief networks where:

- The first five shallow layers consist of the encoding part
- The other layers take care of the decoding part

24.Why is the Leaky ReLU function used in Deep Learning?

Leaky ReLU, also called LReL, is used to manage a function to allow the passing of small-sized negative values if the input value to the network is less than zero.

25.What are some of the examples of unsupervised learning algorithms in Deep Learning?

There are three main unsupervised learning algorithms in Deep Learning:

- Autoencoders
- Boltzmann machines
- Self-organizing maps

26.Can we initialize the weights of a network to start from zero?

Yes, it is possible to begin with zero initialization. However, it is not recommended to use because setting up the weights to zero initially will cause all of the neurons to produce the same output and the same gradients when performing backpropagation. This means that the network will not have the ability to learn at all due to the absence of asymmetry between each of the neurons.

27.What are some of the examples of supervised learning algorithms in Deep Learning?

There are three main supervised learning algorithms in Deep Learning:

- Artificial neural networks
- Convolutional neural networks
- Recurrent neural networks