

2020 May

# AI Final Walkthrough

Ling 

## AI in general

### 1. What did John McCarthy NOT do?

Some examples invented for AI and Lisp (many by John McCarthy and his lab at MIT)

★ if/then/else constructs

★ garbage collection  
dynamic typing

★ recursive function calls  
IDEs

first class functions

lexical closures

★ time sharing (servers, cloud)

Dartmouth Summer Research Project on Artificial Intelligence Organized by John McCarthy

### 2. Which of these statements is true about the Turing test and the Chinese room argument?

1950: Computing Machinery and Intelligence - Turing test

*For we can easily understand a machine's being constituted so that it can utter words, and even emit some responses to action on it of a corporeal kind, which brings about a change in its organs; for instance, if touched in a particular part it may ask what we wish to say to it; if in another part it may exclaim that it is being hurt, and so on. But it never happens that it arranges its speech in various ways, in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do.*

The Chinese room argument holds that a digital computer executing a program cannot be shown to have a "mind", "understanding" or "consciousness", regardless of how intelligently or human-like the program may make the computer behave.

*Searle imagines himself alone in a room following a computer program for responding to Chinese characters slipped under the door. Searle understands nothing of Chinese, and yet, by following the program for manipulating symbols and numerals just as a computer does, he sends appropriate strings of Chinese characters back out under the door, and this leads those outside to mistakenly suppose there is a Chinese speaker in the room.*

The narrow conclusion of the argument is that programming a digital computer may make it appear to understand language but could not produce real understanding. Hence the "Turing Test" is inadequate.

3.	<p>Which of these tasks was <b>NOT</b> solved much better by Deep Learning than previous algorithms?</p> <p>Tasks that were solved by deep learning much better than previous algorithms:</p> <ul style="list-style-type: none"> <li>Speech recognition</li> <li>Automatic text generation</li> <li>Valid ethical concerns</li> <li>Self-driving cars</li> </ul>
4.	<p>What is <b>NOT</b> crucial for deep learning algorithms?</p> <p>Things crucial to deep learning:</p> <ul style="list-style-type: none"> <li>Lots of data</li> <li>Lots of computational power (GPUs, now, TPUs)</li> </ul>
5.	<p>Which of these advances in AI that are used extensively in software technology today were <b>NOT</b> invented by John McCarthy's lab?</p>
6.	<p>Finish the sentence: Nobody supposes that the computational model of rainstorms in London</p> <p>Nobody supposes that the computational model of rainstorms in London will leave us all wet.</p>
7.	<p>Which theory says that our minds are in fact computer programs?</p> <p>Computational theory of mind</p>
8.	<p>Who was <b>NOT</b> present at the Dartmouth Summer Research Project on Artificial Intelligence?</p> <p>Some attendees:</p> <ul style="list-style-type: none"> <li>John McCarthy</li> <li>Ray Solomonoff</li> <li>Marvin Minsky</li> <li>Claude Shannon</li> <li>John Nash</li> <li>W. S. McCulloch</li> <li>Arthur Samuel</li> <li>Nat Rochester</li> <li>David Sayre</li> </ul> <p>Herbert Simon</p>





9.	<p>What was the name of the world's first chatterbot?</p> <p>ELIZA</p>
10.	<p>What was NOT one of the problems with AI identified in the Lighthill report?</p> <p>Problems</p> <ul style="list-style-type: none"> <li>Not enough computational power: in some NLP applications, 20 words would fit into the memory</li> <li>Commonsense knowledge and reasoning, the knowledge acquisition bottleneck</li> <li>Moravec's paradox: "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"</li> </ul>
11.	<p>What is Moravec's paradox?</p> <p>It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility.</p>
12.	<p>Who was not awarded the Turing prize despite being a significant contributor to deep learning?</p> <p>Yoshua Bengio, Geoffrey Hinton, and Yann LeCun are awarded the Turing prize</p>

## Search

1.	<p>Which of these is not a search algoirhm?</p> <p>Search algorithms:</p> <ul style="list-style-type: none"> <li>Backtrack</li> <li>Local search</li> <li>Graph search: BFS, DFS, A*</li> <li>Adversarial search: min-max, alpha-beta</li> <li>Evolutionary algorithms</li> </ul>
----	---

2.	<p>How can we NOT reduce the complexity of a state space?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> time complexity <ul style="list-style-type: none"> <li>• number of the iterations and running time of one iteration</li> </ul> </li> <li><input type="checkbox"/> space complexity <ul style="list-style-type: none"> <li>• size of the workspace</li> </ul> </li> <li><input type="checkbox"/> The computational complexity of an operator can be reduced if the states are completed with extra information that are maintained by the operator itself.</li> </ul>																
3.	<p>What does the complexity of a representation graph NOT depend on?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> number of paths driving from the start depends on <ul style="list-style-type: none"> <li>■ number of nodes and arcs</li> <li>■ branching factor: average number of outgoing arcs</li> <li>■ frequency of the cycles and diversity of their length</li> </ul> </li> </ul>																
4.	<p>Which of these is NOT true of a state space graph?</p> <table border="0"> <tr> <td><input type="checkbox"/> State-space model</td><td>Sate-graph</td></tr> <tr> <td>■ state</td><td>node</td></tr> <tr> <td>■ effect of an operator on a state</td><td>directed arc</td></tr> <tr> <td>■ cost of an operator</td><td>cost of arc</td></tr> <tr> <td>■ initial state</td><td>start node</td></tr> <tr> <td>■ final state</td><td>goal node</td></tr> </table> <p><input type="checkbox"/> Graph-representation: state-graph, start node, goal nodes</p> <table border="0"> <tr> <td>■ sequence of operators</td><td>directed path</td></tr> <tr> <td>■ solution</td><td>directed path from start to goal</td></tr> </table>	<input type="checkbox"/> State-space model	Sate-graph	■ state	node	■ effect of an operator on a state	directed arc	■ cost of an operator	cost of arc	■ initial state	start node	■ final state	goal node	■ sequence of operators	directed path	■ solution	directed path from start to goal
<input type="checkbox"/> State-space model	Sate-graph																
■ state	node																
■ effect of an operator on a state	directed arc																
■ cost of an operator	cost of arc																
■ initial state	start node																
■ final state	goal node																
■ sequence of operators	directed path																
■ solution	directed path from start to goal																
5.	<p>In which of these problems is the problem space NOT the same as paths of the representation graph starting from the start node?</p> <p>n-queens problem</p>																
6.	<p>Which of these is NOT true of a delta-graph?</p> <p><math>\delta</math>- graph: directed, arc-weighted, <math>\delta</math>-property, finite outgoing arcs from a node</p> <p>All path-finding problems can be described with a graph-representation. It is a triple <math>(R, s, T)</math> where <math>R=(N, A, c)</math> is a <math>\delta</math>- graph (representation graph)</p>																

7.	<p>Which of these algorithms use a tentative control strategy?</p> <ul style="list-style-type: none"> <li>• backtracking</li> <li>• graph-search</li> <li>• rule-based reasoning</li> </ul>
8.	<p>Which of these algorithms use an irrevocable control strategy?</p> <ul style="list-style-type: none"> <li>• local search</li> <li>• evolutionary alg.</li> <li>• resolution</li> </ul>
9.	<p>Which of these is a general control strategy?</p> <ul style="list-style-type: none"> <li>• irrevocable control strategy</li> <li>• tentative control strategy</li> </ul>
10.	<p>Can we think of the hill climbing method as a special case of tabu search?</p> <p>Yes</p>
11.	<p>In how many places does simulated annealing use randomness?</p> <p>2</p> <p>1 Instead of selecting the best child of the current node, the new node is picked randomly from among the children of the current node.</p> <p>2 The changing of the coefficient is based on an annealing schedule <math>(T_k, L_k)</math> <math>k=1,2,\dots</math> that rules that the coefficient be <math>T_1</math> during <math>L_1</math> steps, then be <math>T_2</math> at the next <math>L_2</math> steps, etc.</p>
12.	<p>Which of these is a drawback of the tabu search?</p> <ul style="list-style-type: none"> <li>– The size of the tabu set can be set only a posteriori.</li> <li>– Without a strong heuristics it can rarely find the goal, after wrong decisions it can lose itself or even stick in a dead end.</li> </ul>
13.	<p>Which of these is FALSE for local search algorithms?</p> <p>Local search algorithm</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> The global workspace of a local search contains only one (current) node of the representation graph with its small environment. Initially this current node is the start node. The search stops if the current node is a goal node or the search could not take the next step.</li> <li><input type="checkbox"/> In each step the current node is exchanged for its better child by a searching rule.</li> <li><input type="checkbox"/> The control strategy uses an evaluation (objective, fitness, heuristic) function to select a better child node. This function tries to estimate to what extent a node promises the achievement of the goal. This function involves some heuristics.</li> </ul>

14. Which of these is NOT a drawback of the hill climbing algorithm?
- Disadvantages:
- ☐ It can rarely find the goal without a strong heuristics because after a wrong decision it can lose itself or even stick in a dead end
    - several current nodes  local beam search
    - several attempts  random-restart search
    - give up the greedy strategy  simulated annealing
  - ☐ It can lose track around a local optimum or on an equidistant surface of the evaluation function (where neighboring nodes have identical values) if there are cycles in the representation graph (that cannot be recognized).
    - recognize smaller cycles  tabu search
15. Which of these algorithms was NOT invented to avoid hill climbing getting stuck in a dead end?
- Tabu search
16. What does the global workspace of backtracking search contain?
- contains one path from the start node to the current node with all untested outgoing arcs from the nodes of this path
- initially this path contains only the start node
  - the search terminates: either the current node is the goal, or the outgoing arcs of the start node are completely tested
17. What are the search rules of backtracking search?
- append a new untested outgoing arc driving from the current node to the end of the current path
  - remove the last arc of the current path (backtrack)
18. What is the control strategy of backtracking search?
- control strategy: applying the backtracking in last case
19. Which of these is NOT true about the first version of the backtracking search (BT1)?
- ☐ The first version of the backtracking algorithm (BT1) observes only the first two conditions of the backtracking: “dead end” and “checked crossroads”.
  - ☐ In a finite acyclic directed graph the BT1 always terminates, and if there exists a solution path, then it finds one.
  - ☐ It can be implemented with a recursive procedure
    - Starting: solution := BT1(start)

20.	<p>Which of these statements is NOT true about the second version of the backtracking search (BT2)?</p> <p><input type="checkbox"/> The second version of backtracking (BT2) implements all conditions of the backtracking step.</p> <p><input type="checkbox"/> In <math>\delta</math>- graphs the BT2 always terminates, and if there exists a solution path shorter than the depth bound, then it finds a solution path.</p> <p><input type="checkbox"/> It can be implemented with a recursive procedure</p> <ul style="list-style-type: none"> <li>– Starting: solution := BT2(&lt;start&gt;)</li> </ul>
21.	<p>Which of these statements is NOT true about the second version of the backtracking (BT2)?</p> <p>Same as above</p>
22.	<p>Which of these is an advantage of backtracking search?</p> <ul style="list-style-type: none"> <li>– always terminates, and finds solution (inside the depth bound)</li> <li>– implementation is simple</li> <li>– small memory</li> </ul>
23.	<p>What does the global workspace of graph search contain?</p> <p>global workspace: stores the discovered paths (the beginning part of all paths driving from the start node: this is the search graph) and separately records the last nodes of all discovered paths (they are called open nodes)</p> <ul style="list-style-type: none"> <li>• initial value: start node</li> <li>• termination condition: a goal node must be expanded or there is no open node</li> </ul>
24.	<p>What is the search rule of graph search?</p> <p>searching rules: expand open nodes</p>
25.	<p>What is the control strategy of graph search?</p> <p>control strategy: selects an open node to be expanded based on an evaluation function</p>
26.	<p>What kind of nodes are the open nodes?</p> <p>set of open nodes (OPEN) :</p> <p>the nodes that are waiting for their expansions because their successors are not known or not well-known</p>



27. How do we call the subgraph we store in the global workspace of graph search?  
search graph (G)
28. What kind of nodes are the closed nodes?
29. What does the parent pointer function ( $\pi$ ) point to?  
 $\square \pi : N \rightarrow N$  parent pointer function  
 –  $\pi(m)$  = one parent of  $m$  in  $G$ ,  $\pi(\text{start}) = \text{nil}$ 
  - $\pi$  determines a spanning tree in  $G$  and helps to take the solution path out from  $G$  after successful termination
  - If only the  $\pi(m)$  always showed an optimal path  $\text{start} \rightarrow m$  in  $G$  when the node  $m$  is generated
30. When is an evaluation function decreasing?  
 An evaluation function  $f: \text{OPEN} \rightarrow R$  is decreasing if for all nodes  $n$  ( $n \in N$ ) the  $f(n)$  never increases but always decreases when a cheaper path has been found to the node  $n$ .  
 For example the function  $g$  has got this property.
31. When is a node of a search graph correct?  
 The node  $m$  is correct if  $g(m)$  and  $\pi(m)$  are consistent and optimal  
 i.e.  $g(m) = c^\pi(\text{start}, m)$  and  $c^\pi(\text{start}, m) = \min_{\alpha \in \{\text{start} \rightarrow m\} \cap G} c^\alpha(\text{start}, m)$ .
32. Which of these statements is NOT true about the general graph search algorithm?
1.  $G := (\{\text{start}\}, \emptyset)$  ;  $\text{OPEN} := \{\text{start}\}$  ;  $\pi(\text{start}) := \text{nil}$  ;  $g(\text{start}) := 0$
  2. **loop**
  3. **if**  $\text{empty}(\text{OPEN})$  **then return** no solution
  4.  $n := \min_f(\text{OPEN})$
  5. **if**  $\text{goal}(n)$  **then return** solution  $(n, \pi)$
  6.  $\text{OPEN} := \text{OPEN} - \{n\}$
  7. **for**  $\forall m \in \Gamma(n) - \pi(n)$  **loop**
  8. **if**  $m \notin G$  or  $g(n) + c(n, m) < g(m)$  **then**
  9.  $\pi(m) := n$  ;  $g(m) := g(n) + c(n, m)$  ;  $\text{OPEN} := \text{OPEN} \cup \{m\}$
  10. **endloop**
  11.  $G := G \cup \{(n, m) \in A \mid m \in \Gamma(n)\}$
  12. **endloop**

33. Which of these statements is true about the general graph search?
- ☐ Each node is expanded only finite times in a  $\delta$ -graph.
  - ☐ The general graph-search always terminates in a finite  $\delta$ -graph.
  - ☐ The general graph-search finds a solution in a finite  $\delta$ -graph if there exists a solution.
34. Can we use order heuristic as a secondary control strategy in an uninformed graph search?
- The tie-breaking rules (secondary evaluation functions) may contain heuristics even in non-informed graph-search.
35. Which of these is depth first search (f is the evaluation function, g is the cost function, c is the cost of an edge)?
- $f = -g, c(n,m) = 1$
36. Which of these is breadth first search (f is the evaluation function, g is the cost function, c is the cost of an edge)?
- $f = g, c(n,m) = 1$
37. Which of these is uniform cost search (f is the evaluation function, g is the cost function, c is the cost of an edge)?
- $f=g$
38. What does admissibility mean for a graph search?
- $h(n) \leq h^*(n) \quad \forall n \in N$
- The heuristic function  $h:N \rightarrow R$  estimates the cost of the cheapest path from a node to the goal.
- remaining optimal cost from n to any goal node of T:  $h^*(n) = c^*(n,T)$
39. Which statement is **NOT** true about the constant 0 function?
- 0 (zero function) ~ fake heuristic function
- Zero function is non-negative, admissible and monotone.
40. Which of these is the look-forward graph search (f is the evaluation function, g is the cost function, h is the heuristic, h-star is the optimal cost, c is the cost of an edge)?
- $f=h$

41.	Which of these is the A algorithm (f is the evaluation function, g is the cost function, h is the heuristic, h-star is the optimal cost, c is the cost of an edge)? $f=g+h, h \geq 0$
42.	Which of these is the A-star algorithm (f is the evaluation function, g is the cost function, h is the heuristic, h-star is the optimal cost, c is the cost of an edge)? $f=g+h, h \geq 0, h \leq h^*$
43.	Which of these is the A-c (consistent) algorithm (f is the evaluation function, g is the cost function, h is the heuristic, h-star is the optimal cost, c is the cost of an edge)? $f=g+h, h \geq 0, h \leq h^*, h(n)-h(m) \leq c(n,m)$
44.	Which of these is a property of the A algorithm? finds solution if there exists one (even in infinite $\delta$ -graph)
45.	Which of these is <b>NOT</b> true about the A-c (consistent) algorithm? Algorithm A-c properties: <ul style="list-style-type: none"> <li>• finds optimal solution if there exists one (even in infinite <math>\delta</math>-graph)</li> <li>• expands a node at most once</li> </ul>
46.	When do we say that a heuristic function is monotone? $h(n)-h(m) \leq c(n,m) \quad \forall (n,m) \in A$
47.	Which of these statements is <b>NOT</b> true about breadth-first search? Breadth-first graph-search properties: <ul style="list-style-type: none"> <li>• finds the shortest (not the cheapest) solution if there exists one even in infinite <math>\delta</math>-graph</li> <li>• each node is expanded at most once</li> </ul>
48.	Which of these is true about uniform cost search? Uniform-cost graph-search properties: <ul style="list-style-type: none"> <li>• finds optimal (the cheapest) solution if there exists one even in infinite <math>\delta</math>-graph</li> <li>• each node is expanded at most once</li> </ul>

49. Which of these was **NOT** true about the two-player games we have been examining in the course?

- ☐ Two players take turns according to given rules until the game is over.
- ☐ The game is in a fully observable environment, i.e., the players know completely what both players have done and can do.
- ☐ Either the number of the possible steps in a current state or the length of the plays of the game are finite.
- ☐ Each step is unequivocal, its effect is predictable. The plays of the game do not depend on chance at all.
- ☐ The sum of the payoff values of the players at the end of the game is always zero. (In special case players can only win or lose. Sometimes a draw is also possible.)

two payoff functions:  $p_A, p_B: \text{final states} \rightarrow \mathbb{R}$  (players: A, B)

● In a zero-sum two-player game:  $p_A(t) + p_B(t) = 0$  for all final state  $t$

● In special case the range of these functions:  $+1, 0, -1$

- $+1$  if the player wins (winning final state for the very player)
- $-1$  if the player loses (losing final state for the very player)
- $0$  if the final state is a draw

50. What does the state of a two-player games represent?  
configuration + player next to move

51. What is the winning strategy in a two-player game?

The winning strategy shows how a player could win no matter what the opposite player does.

52. When do we cut in the alpha-beta algorithm?

Cutting rule: if there are an  $\alpha$  and  $\beta$  value on the current path so that  $\alpha \geq \beta$ .

53. What is the stationary test for minimax search?

The evaluation value of a node may be misleading if it significantly differs from the value of its parent node:

$$|f(\text{parent}) - f(\text{node})| > K$$

54.	<p>Which of these statements is <b>NOT</b> true about the game tree?</p> <p><input type="checkbox"/> Game tree can be interpreted by the players in different ways and these interpretations can be drawn with AND/OR trees.</p> <ul style="list-style-type: none"> <li>• there is OR connection between the arcs going from the nodes on the level of the current player</li> <li>• there is AND connection between the arcs going from the nodes on the level of the opponent player</li> </ul> <p><input type="checkbox"/> Both players have got their own AND/OR tree.</p> <p><input type="checkbox"/> The winning (or non-losing) strategy of one player is a hyper- path of his/her AND/OR tree that is driving from the root to winning goal nodes.</p> <p><input type="checkbox"/> The search of a winning strategy is a hyper-path-finding problem in an AND/OR tree.</p>												
55.	<p>Which of these is a step in the minimax algorithm?</p> <ol style="list-style-type: none"> <li>1. Several levels of the game tree are built up starting from the current state (depending on the time or the storage limit).</li> <li>2. The leaves of this subtree must be evaluated based on the evaluation function.</li> <li>3. A value can be computed for each inner node <ul style="list-style-type: none"> <li>• this is the maximum of the successors' values if the node is on our level,</li> <li>• this is the minimum of the successors' values if the node is on the opponent's level.</li> </ul> </li> <li>4. The next step will be towards the successor of the current state which has the largest value.</li> </ol>												
56.	<p>What is the game tree?</p> <p>Same as question 54 plus</p> <table> <tr> <td><input type="checkbox"/> node</td><td>- configuration (the same configuration may occur in several nodes)</td></tr> <tr> <td><input type="checkbox"/> level</td><td>- player (the levels of A and B alternate)</td></tr> <tr> <td><input type="checkbox"/> arc</td><td>- step (level by level)</td></tr> <tr> <td><input type="checkbox"/> root</td><td>- initial configuration</td></tr> <tr> <td><input type="checkbox"/> leaf</td><td>- terminal configuration</td></tr> <tr> <td><input type="checkbox"/> branch</td><td>- play of the game</td></tr> </table>	<input type="checkbox"/> node	- configuration (the same configuration may occur in several nodes)	<input type="checkbox"/> level	- player (the levels of A and B alternate)	<input type="checkbox"/> arc	- step (level by level)	<input type="checkbox"/> root	- initial configuration	<input type="checkbox"/> leaf	- terminal configuration	<input type="checkbox"/> branch	- play of the game
<input type="checkbox"/> node	- configuration (the same configuration may occur in several nodes)												
<input type="checkbox"/> level	- player (the levels of A and B alternate)												
<input type="checkbox"/> arc	- step (level by level)												
<input type="checkbox"/> root	- initial configuration												
<input type="checkbox"/> leaf	- terminal configuration												
<input type="checkbox"/> branch	- play of the game												

57.	What is the general control strategy of evolutionary algorithms? Irrevocable strategy.
58.	What does the evolutionary algorithm store in its global workspace? It handles several elements ( <b>individuals</b> ) of the problem space at each iteration and permanently modifies this population that becomes better and better until the solution (the goal, the best individual) appears.
59.	Which of these is <b>NOT</b> an evolutionary operator? Evolutionary operators – selection, recombination, mutation, replacement
60.	How do we code an individual? An individual is represented by a code (chromosome) that is most commonly a sequence of signals.
61.	How many steps does the evolutionary cycle consist of? 4: selection, recombination, mutation, replacement
62.	Where can we incorporate randomness into the evolutionary algorithm? First an initial population is selected mostly at <b>random</b> . Selection – Tournament: the selected individuals are the best individuals of <b>randomly</b> selected groups of the population – Culling: all individuals below a given threshold are discarded and then the individuals are selected <b>randomly</b> from the remaining individuals Recombination – Crossover: signals of the parent codes are exchanged at the positions chosen <b>randomly</b> Mutation – Each position of the code is subject to <b>random</b> change with a small independent probability (p).
63.	Where do we use selection in the evolutionary algorithm? Selection: better individuals are selected for reproduction

- |     |   |
|-----|---|
| 64. | <p>What is a good selection algorithm in evolutionary algorithms?</p> <p>The better individuals must be selected but the worse ones must be given a chance to be chosen. (stochastic method)</p>  |
| 65. | <p>What is the connection between crossover and recombination?</p> <p>pairs of the selected individuals (parents) are bred in order to create their offspring</p>   |
| 66. | <p>When does the evolutionary algorithm terminate?</p> <ul style="list-style-type: none"> <li>• either a goal individual appears in the population</li> <li>• or the overall fitness value of the population is not being changed</li> </ul>  |
| 67. | <p>Which of these is <b>not</b> a strategy parameter of evolutionary algorithms?</p> <p>Settings of the strategy parameters</p> <ul style="list-style-type: none"> <li>– size of the population, probability of mutation, rate of the offspring, rate of the replacement</li> </ul> |

## Machine Learning

- |    |   |
|----|---|
| 1. | <p>What does it mean for learning to be supervised?</p> <p>Supervised learning learns a function from labeled data</p> <p>Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples.</p>                |
| 2. | <p>What does it mean for learning to be unsupervised?</p> <p>Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.</p> |
| 3. | <p>What is an epoch?</p> <p>The number of passes through the entire training dataset the machine learning algorithm has completed. If the batch size is the whole training dataset (batch mode) then number of batches and epoch are both 1.</p>  |

4.	<p><b>What is a minibatch?</b></p> <p>"Minibatch" means that the gradient is calculated across the entire batch before updating weights. If you are not using a "minibatch", every training example in a "batch" updates the learning algorithm's parameters independently.</p>
5.	<p><b>Why do we use separate training and test sets?</b></p> <p>Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.</p>
6.	<p><b>Why do we use a validation set in addition to the training and test sets?</b></p> <p>In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation dataset in addition to the training and test datasets.</p>
7.	<p><b>What is a classification problem?</b></p> <p>A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease". A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes.</p>
8.	<p><b>What are the hyperparameters of a learning algorithm?</b></p> <p>In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins. By contrast, the values of other parameters are derived via training. Given these hyperparameters, the training algorithm learns the parameters from the data.</p>
9.	<p><b>When do we use the sigmoid activation function?</b></p> <p>It is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.</p>
10.	<p><b>When do we use the softmax activation function?</b></p> <p>The softmax activation function is used in neural networks when we want to build a multi-class classifier which solves the problem of assigning an instance to one class when the number of possible classes is larger than two.</p>



11.	<p>What is the definition of the ReLU activation function?</p> <p>ReLU stands for rectified linear unit, and is a type of activation function. Mathematically, it is defined as <math>y = \max(0, x)</math></p> <p>ReLU is the most commonly used activation function in neural networks, especially in CNNs.</p>
12.	<p>When do we use the ReLU activation function?</p> <p>It is used in almost all the convolutional neural networks or deep learning.</p>
13.	<p>When do we use the binary cross-entropy loss function?</p> <p>You use binary crossentropy on multi-label problems.</p> <p>Example: You want to determine the mood of a piece of music. Every piece can have more than one mood, for instance, it can be both "Happy" and "Energetic" at the same time. To solve this problem you use binary crossentropy.</p>
14.	<p>When do we use the categorical cross-entropy loss function?</p> <p>Use categorical crossentropy in classification problems where only one result can be correct.</p> <p>Example: In the MNIST problem where you have images of the numbers 0,1, 2, 3, 4, 5, 6, 7, 8, and 9. Categorical crossentropy gives the probability that an image of a number is, for example, a 4 or a 9.</p>
15.	<p>What would be the activation function and loss for a binary classification problem?</p>
16.	<p>What would be the activation function and loss for a multiclass classification problem?</p>
17.	<p>Which of these are stopwords?</p> <p>Stop Words: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.</p>

18. Which of these words were stemmed?

Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.

19. What does a language model do?

Language models determine word probability by analyzing text data. They interpret this data by feeding it through an algorithm that establishes rules for context in natural language. Then, the model applies these rules in language tasks to accurately predict or produce new sentences.

20. What is the bag of words model?

Bag of Words (BoW) is an algorithm that counts how many times a word appears in a document. It's a tally. Those word counts allow us to compare documents and gauge their similarities for applications like search, document classification and topic modeling. BoW is a also method for preparing text for input in a deep-learning net.

21. What is the difference between bag of words and TFIDF?

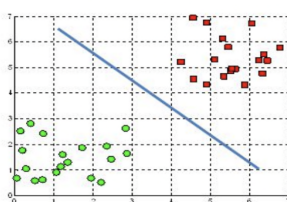
Bag of Words just creates a set of vectors containing the count of word occurrences in the document (reviews), while the TF-IDF model contains information on the more important words and the less important ones as well.

22. What kind of hyperplane is the Support Vector Machine (SVM) learning?

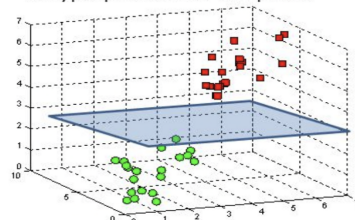
The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

A hyperplane in  $\mathbb{R}^2$  is a line

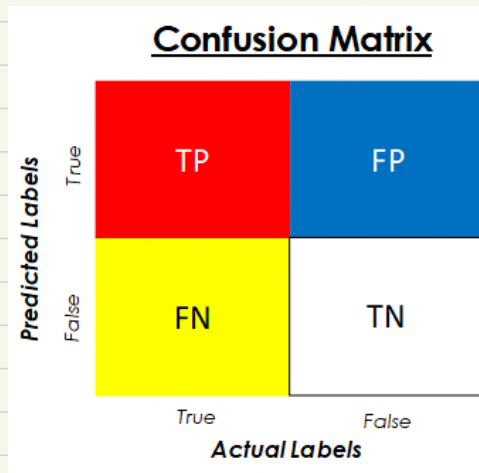


A hyperplane in  $\mathbb{R}^3$  is a plane



23. What do we use the confusion matrix for?

A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. It is used to measure the performance of a classification model. Confusion matrices are widely used because they give a better idea of a model's performance than classification accuracy does.



24. What is grid search? Why do we use it?

Grid search is a tuning technique that attempts to compute the optimum values of hyperparameters. It is an exhaustive search that is performed on a the specific parameter values of a model. The model is also known as an estimator.

25. When would you use random search instead of grid search?

Random Search sets up a grid of hyperparameter values and selects random combinations to train the model and score. This allows you to explicitly control the number of parameter combinations that are attempted. The number of search iterations is set based on time or resources. Given the same resources, Randomized Search can even outperform Grid Search.

26. What would be the one-hot encoding of [1, 3, 0]?

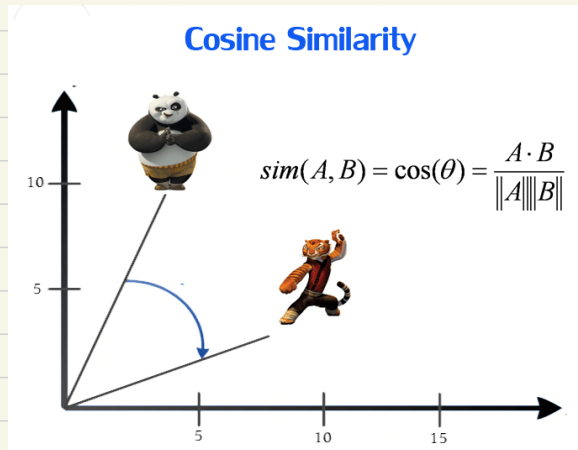
[1, 0, 0]

[0, 1, 0]

[0, 0, 1]

27. What does a word embedding do?

A word embedding is a learned representation for text where words that have the same meaning have a similar representation.



28. What is an example of clustering?

- Soft clustering example: topic models
- distribution-based clustering
- generating odd one out puzzles
- k-means

29. What is the difference between hard and soft clustering?

Hard clustering: an item can belong only to a single cluster

Soft clustering: weights describe the degree to which an item belongs to the clusters

30. What is NOT true of the k-means problem?

- Given: k, the number of clusters
- Each cluster is represented by its centroid
  - The mean of the points in the cluster
- Find the k centroids and assign the points to these in a way that minimizes the squared distances of the points from the centroid of their cluster
  - Equivalent to minimizing pairwise distances within clusters
  - NP-hard, so we approximate
  - We can only find a local optimum
  - We can run it multiple times with different random initializations

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k \frac{1}{2|S_i|} \sum_{\mathbf{x}, \mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2$$

31. What are the two steps of the k-means algorithm?

1 Assign each item to the nearest centroid:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\}$$

2 Compute the new centroids:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

32. What is NOT an issue with the k-means algorithm?

Issue

- k is too small
- k is too large
- bad initialization
- the real clusters are not centroid based

33. What does Latent Semantic Analysis do?

Latent semantic analysis (LSA) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.

34. What is NOT a reason to use dimensionality reduction?

- The data are low dimensional in a higher dimensional space
- Data visualization
- Noise reduction
- Decrease the complexity of the learning problem (better results, smaller runtimes, . . .)
- We can conjecture new relationships on the visualized lower dimensional data
- We need a lower dimensional and/or dense representation to solve a problem

35. What is a principal component in Principal Components Analysis (PCA)?

The projection of the dataset with the greatest variance is on the first axis

36. What is the relationship between Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)?

SVD

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$$

Computing PCA with SVD

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\hat{\mathbf{\Sigma}}^2\mathbf{W}^T\end{aligned}$$

$\mathbf{W}$  contains the eigenvectors of  $\mathbf{X}^T\mathbf{X}$ . The singular values are the square roots of the eigenvalues.

37. What does an autoencoder do?

An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal “noise”.

38. Why doesn't the autoencoder just do an identity transformation?

39. How many matrices does Latent Semantic Analysis produce from its input matrix?

3

