Quiz 1

Started: Jan 18 at 10:23pm

Quiz Instructions

Intro and Universal Approximators

This quiz covers lectures 1 and 2. Several of the questions invoke concepts from the hidden slides in the slide deck, which were not covered in class. So please go over the slides before answering the questions.

You will have three attempts for the quiz. Questions will be shuffled and you will not be informed of the correct answers until after the deadline. While you may discuss the concepts underlying the questions with others, you must solve all questions on your own - see course policy.

Question 1 1 pts

Which of your quiz scores will be dropped?

Hint: watch Lecture 0

- ○ No scores will be dropped
- ○ Lowest 2 quiz scores
- ○ Lowest 1 quiz scores
- ○ Lowest 3 quiz scores

Question 2 1 pts

We sometimes say that neural networks are connectionist machines as opposed to von Neumann machines. Which of the following describe why we make this distinction? (select all that apply)

Slide: lec 1, “Connectionist Machines”. slide 47-48

- ☐ A von Neumann machine has a general purpose architecture with a processing unit that is distinct from the memory that holds the programs and data. A connectionist machine makes no distinction between processing unit and the program.

☐ It is possible to create hardware implementations of von Neumann machines (e.g. CPU's) as well as software implementations (e.g. virtual machines). However, connectionist machines can only be implemented in software (e.g. neural networks in Python).

☐ A von Neumann machine can be used for general-purpose computing by simply providing a different program, without changing the machine itself. A connectionist machine implements a specific program, and changing the program requires changing the machine.

☐ Because of its flexibility, a von Neumann machine is capable of computing any Boolean function of a given number of Boolean inputs, whereas connectionist machines, no matter how complex, are fundamentally unable to model certain types of Boolean functions.

Question 3 1 pts

What computational systems can compose arbitrary Boolean functions? (select all that apply)

Hint: Lecture 1, slides 52, 58–63 and 76–81. Also, “compose” is read as arbitrary Boolean functions, not some Boolean functions.

☐ Digital circuits containing only OR and NOT logic gates.

☐ McCulloch and Pitts's connectionist networks.

☐ One of Rosenblatt's perceptrons.

☐ Turing’s B-type machines.

Question 4 1 pts

Is the following statement true or false: for any Boolean function on the real plane which outputs a 1 within a single connected but NON-CONVEX region (*i.e.* not a convex polygon), you need at least three layers (including the output layer) to model the function EXACTLY.

Hint: Lecture 2 slides "Complex decision boundaries"

○ True

○
False

Question 5 1 pts

How does the number of weights (note: not neurons) in an XOR network with *threshold logic* perceptrons with *1 hidden layer* grow with the number of inputs to the network?

See lec 2: Slides on "Optimal depth" and "Network size" 113-123

○
Exponential or faster

○
Linear

○
Polynomial but faster than linear

○
Between polynomial and exponential

Question 6 1 pts

In general, as the depth of a NN increases, at what rate does the number of neurons/params required to represent a function change?

(Note: for the definition of network depth, see the lecture 2 recording)

Hint: Review Lec 2, slides on "The challenge of depth" (Slides 67-68)

○
Decreases linearly

○
Increases linearly

○
Increases quadratically

○
Decreases exponentially

Question 7 1 pts

A majority function is a Boolean function of N variables that produces a 1 if at least N/2 of the inputs are 1. Which of the following are true? (select all that apply)

Hint: Relevant Lecture 2 Slides: slides 29-30, 70, 75

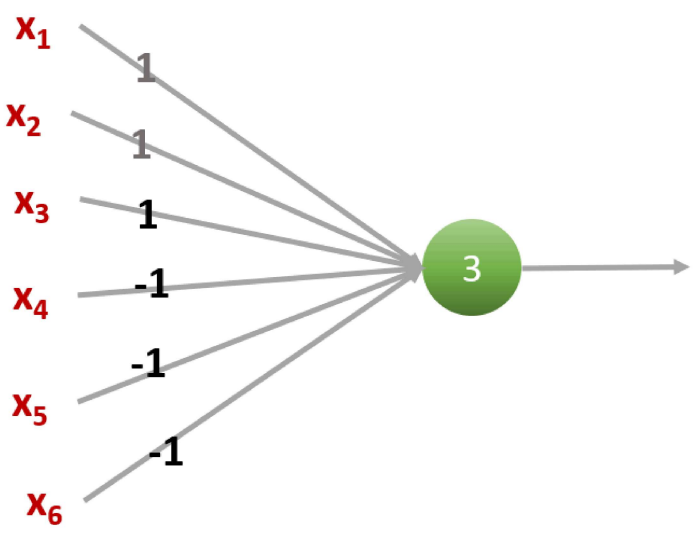
☐ The number of gates in the smallest Boolean circuit of AND, OR and NOT gates that computes the majority function is polynomial in N.

☐ A single perceptron can compute a majority function.

☐ A fixed-depth Boolean circuit, comprising only AND, OR and NOT gates, will require $\Omega(\exp(N^{\alpha}))$ gates to compute the majority function ($\alpha > 0$)

☐ We will require a multilayer perceptron with $\Omega(\exp(N))$ perceptrons to compute a majority function

Question 8 1 pts



Under which condition(s) is the perceptron graph above guaranteed to fire? Note that ~ is NOT. (select all that apply)

Slide: lec 2, "Perceptron as a Boolean gate" slides 26-30

☐ ~x1 & ~x2 & ~x3 & x4 & x5 & x6

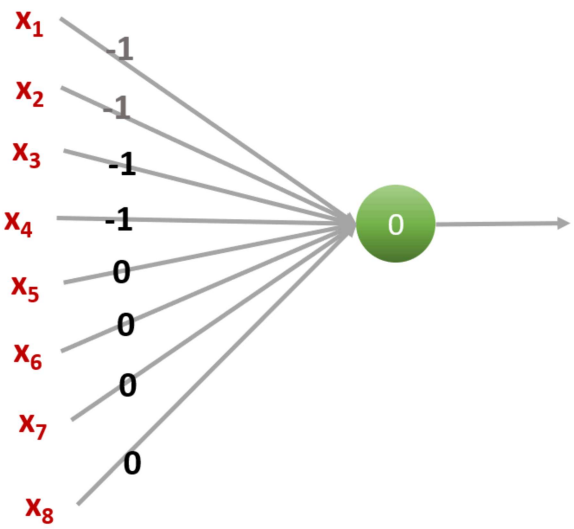
☐ Never fires

☐ x1 &~x2 & x3 & ~x4 & x5 & ~x6

☐ x1 & x2 & x3 & ~x4 & ~x5 &~x6

Question 9 1 pts

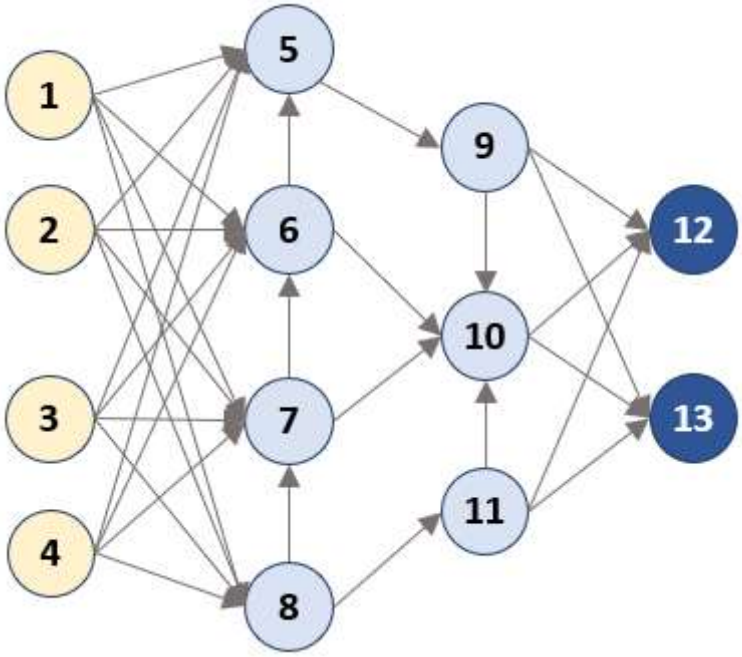
Under which conditions will the perceptron graph below fire? Note that ~ is NOT. (select all that apply)



Slide: lec 2, "Perceptron as a Boolean gate", slides 26-30

- [ ] x1 & x2 & x3 &x4
- [ ] Never fires
- [ ] fires only if x1, x2, x3, x4 are all 0, regardless of x5 .. x8
- [ ] ~x1 & ~x2 & ~x3 & ~x4

Question 10 1 pts



If the yellow nodes are inputs (not neurons) and the dark blue nodes are outputs, what is the depth of this NN?

(Note: for the definition of network depth and layer number, see the lecture 2 recording)

Hint: lec 2, "Deep Structures", *Slides:* 17-18

Quiz saved at 10:23pm Submit Quiz