Quiz-01

- Due Jan 19 at 11:59pm
- Points 10
- Questions 10
- Available Jan 17 at 6pm - Jan 19 at 11:59pm
- Time Limit None
- Allowed Attempts 3

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.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 3	58 minutes	8.5 out of 10
LATEST	Attempt 3	58 minutes	8.5 out of 10
	Attempt 2	65 minutes	5.5 out of 10
	Attempt 1	98 minutes	7 out of 10

Correct answers are hidden.

Score for this attempt: 8.5 out of 10
Submitted Jan 18 at 12:35am
This attempt took 58 minutes.

Question 1

1 / 1 pts

Which of your quiz scores will be dropped?

- ( ) Lowest 3 quiz scores
- ( ) No scores will be dropped
- (•) Lowest 2 quiz scores
- ( ) Lowest 1 quiz scores

Question 2

1 / 1 pts

Which concept is most closely described by the following sentence? "The finding that neurons whose connections form cycles can model memory."

Hint: Lec 1, slide “other things MLPs can do”

- Sanger's Rule
- Parallel Distributed Processing
- Turing's B-type machines
- Lawrence Kubie's Loop Networks

Question 3

1 / 1 pts

What term does the following sentence describe? "A modification of Hebbian learning addressing its lack of weight reduction and its instability."

Hint: See lec 1, "Hebbian Learning"

- Sanger’s rule
- Turing’s A-type machines
- Parallel distributed processing
- Lawrence Kubie’s loop networks

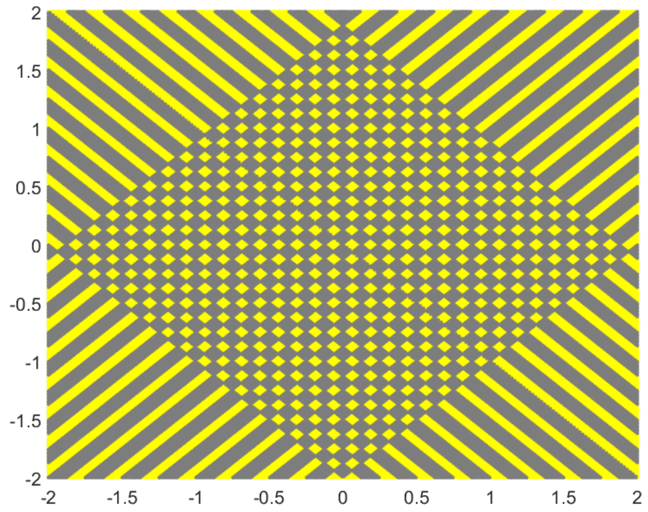
Question 4

1 / 1 pts

We would like to compose an MLP that can represent a *three-dimensional* version of the cross-hash function shown below (the illustration is 2-dimension because it is difficult to depict the 3-D version of the map). The input is now a three-dimensional vector. The output map comprises alternating cuboids of yellow and grey, where yellow represents a region where the function must output 1, and grey cuboids are regions where it must output 0.

Assume the network is restricted to only two hidden layers (fixed total depth of 3 including output layer). Which of the following statements is true of the number of neurons (not weights) required to model the function below?

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



Hint : Slide: lec 2, "Network size?"

- It is polynomial in the number of hyperplanes required to form the cross hash
- It is linear in the number of hyperplanes required to form the cross hash
- It is quadratic in the number of hyperplanes required to form the cross hash
- It is exponential in the number of hyperplanes required to form the cross hash

It is polynomial in the number of planes, but exponential in the dimensionality of the space.

Explanation: You can have at most N hyperplanes in each dimension. These are parallel (don't intersect). If you have N hyperplanes on each of 2 dimensions, they will intersect N^2 times in a crosshash. Each intersection corresponds to one square, so you get up to N^2 squares.

In D dimensions, you have N^D intersections, so you get up to N^D hypercube cells.

IncorrectQuestion 5
0 / 1 pts

How does the number of **weights** in an Boolean XOR network (composed of Boolean gates) with **2 hidden layers and fixed depth** grow with the number of inputs to the network?

Hint: Lecture 2 Slides: Slides 66-68

- Between polynomial and exponential
- Exponential or faster
- Polynomial but faster than linear
- Linear

See slide titled "The challenge of depth". If the network depth is fixed to any K, the number of neurons grows exponentially in N.

The number of neurons is O(2^{CN}) in Kth layer, if the number of neurons grows exponentially, how fast does the number of weights grow?

Question 6

1 / 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)

- ( ) Increases quadratically
- ( ) Decreases linearly
- ( ) Increases linearly
- (•) Decreases exponentially

Explanation: the worst case problem of XORs is what we’ll use for illustration. With one hidden layer you need O (exp(N)) neurons. WIth 3, you need O(exp(N/2)) neurons (the first 2 layers to compute XORs, which results in N/2 variables, and then one hidden layer to compute the XOR of N/2 variables. with 5 If you terminate the XOR net after K layers, the number of neurons required is O (exp (N/4)) etc.

So the reduction is exponential with increasing depth.

In general, for a given function, deeper networks will require exponentially fewer parameters than shallower ones to model the function accurately (exactly, or with arbitrary precision.

PartialQuestion 7

0.5 / 1 pts

Which of the following are impossible in theory? Assume all networks are finite in size, though they can be as large as needed. (select all that apply)

Hint: (1) The MNIST dataset is finite, (2) The neural network is a universal approximator.

- [ ] Using a threshold network, as deep as you need, to determine if an arbitrary 2D input lies within the square with vertices {(1, 0), (-1, 0), (0, 1), (0, -1)}.
- [x] Using a threshold network with one hidden layer to determine with certainty if an arbitrary 2D input lies within the unit circle.
- [x] Using a threshold network with one hidden layer to perfectly classify all digits in the MNIST dataset.
- [x] Using a threshold network, as deep as you need, to precisely calculate the L1 distance from a point to the origin.

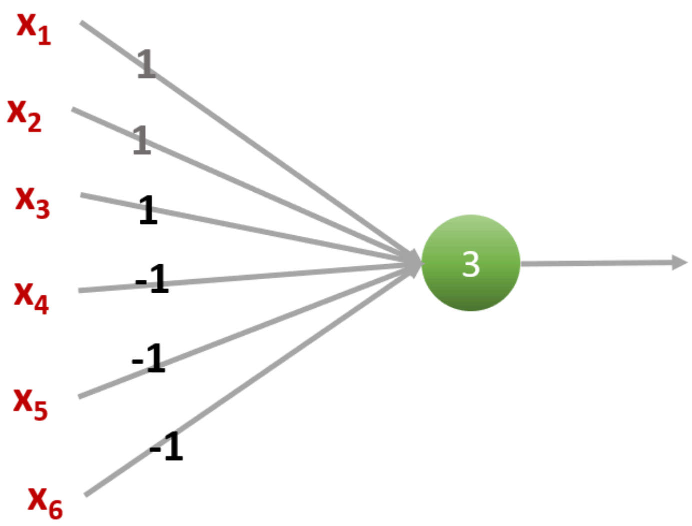
1. There is a finite number of MNIST digits and a single hidden layer network is a universal approximator, so a finite network can classify the data.

2. A finite network can only approximate a circular decision boundary, and some points will be misclassified.

3. The L1 distance is a continuous function of the input. You cannot model it perfectly with discontinuous functions like the threshold function.

Question 8

1 / 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
- [x] x1 & x2 & x3 & ~x4 & ~x5 &~x6

x1 & x2 & x3 & ~x4 & ~x5 &~x6 = 1(1) + 1(1) +1(1) + 0(-1)+0(-1) +0(-1) =3

x1 &~x2 & x3 & ~x4 & x5 & ~x6= 1(1) + 0(1) +1(1) + 0(-1)+1(-1) +0(-1) =1

~x1 & ~x2 & ~x3 & x4 & x5 & x6 = 0(1) + 0(1) +0(1) + 1(-1)+1(-1) +1(-1) =-3

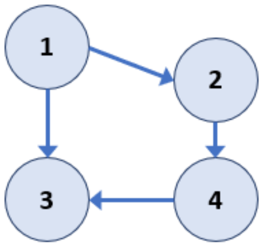
Question 9

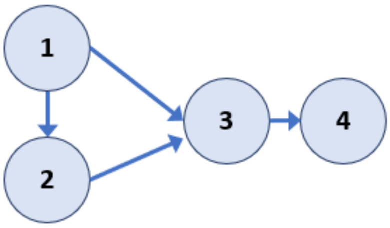
1 / 1 pts

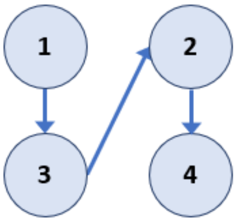
Which of the following NNs have a depth of 3? (Choose all that apply)

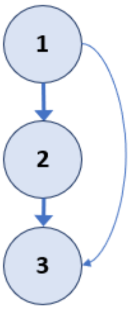
(Note: for the definition of network depth, see the lecture 2 recording)

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

- [x] 

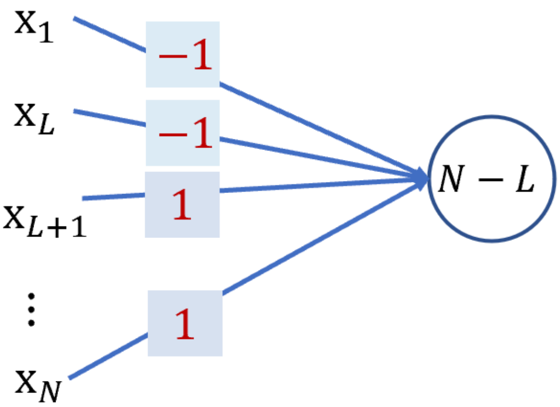
- [x] 

- [x] 

- [ ] 

Question 10

1 / 1 pts



What boolean function does the above perceptron represent?

(Hint: note that L=2 and note that a line above a term means “not”, i.e. $\overline{A}$ means “not A”)

Slide: lec 2, “Perceptron as a Boolean gate”

- $\left(\bigwedge_{i=L+1}^{N} \bar{X}_i\right) \vee \left(\bigwedge_{i=L+1}^{N} X_i\right)$

- $\left(\bigvee_{i=1}^{N} \bar{X}_i\right)$

- $\left(\bigwedge_{i=1}^{N} X_i\right)$

- $\overline{\left(\bigvee_{i=1}^{L} X_i\right)} \wedge \left(\bigwedge_{i=L+1}^{N} X_i\right)$

There are N weights in total. The maximum possible negative contribution is -2 (X_1 and L fire). The maximum possible positive contribution is N - L (X_(L+1) to X_N fire). Given that the threshold is N - L, we need to have all possible positive contributions, and not a single negative contribution. This is described in the answer.

Quiz Score: 8.5 out of 10