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.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 1	98 minutes	7 out of 10
LATEST	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: 5.5 out of 10
Submitted Jan 17 at 11:13pm
This attempt took 65 minutes.

Question 1
1 / 1 pts
Which of your quiz scores will be dropped?

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

Question 2

1 / 1 pts

Match the corresponding terms and definitions introduced in Lecture 1.

Hint: Lecture 1: Slides on 31-81

The McCulloch and Pitts model

is a Logical Calculus of the Id

Alexander Bain

is known for his Connectionis

Lawrence Kubie

modeled the memory as a cir

Hebbian Learning

is a proposed mechanism to

Marvin Minsky and Seymour Papert

Their mechanism known as tl

One of David Hartley’s Observations

Our brain represents compou

Frank Rosenblatt

made the first algorithmically

Associationism Theory by Aristotle

These are his four laws: The

Question 3

1 / 1 pts

Which of the following are **NOT** part of Aristotle's laws of association? (select all that apply)

Hint: See lec 1: Slides on "Associationism" 31-35

- [ ] Events or things near the same place and time are associated.
- [ ] Thoughts about one event or thing triggers thoughts about related events or things.
- [x] Mental willpower is used to associate unrelated events or things
- [ ] Thoughts about one event or thing triggers thoughts about events or things that have opposite qualities
- [x] Associations are formed from logical deduction.

IncorrectQuestion 4

0 / 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?

Hint: Review Lec 2: Slides on "Caveat 2" (Slide 75)

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

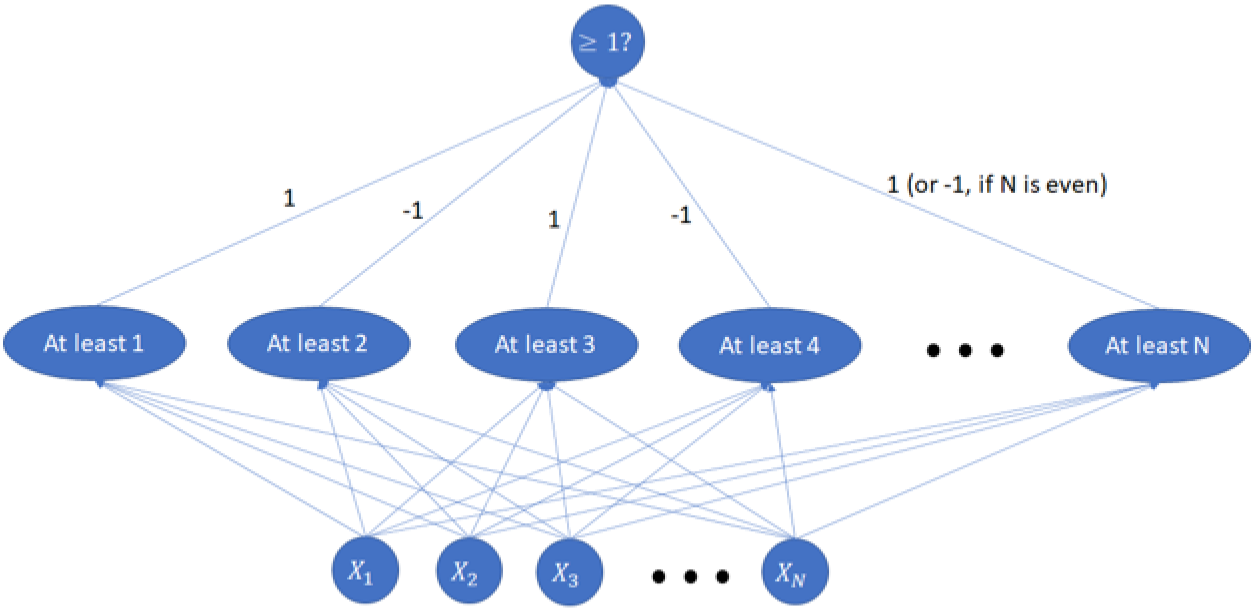
For *Boolean* gate networks, the number of neurons is exp(N), the number of *weights* is N*exp(N), which is faster than exponential.

If we consider *perceptron* networks, using threshold logic, it is possible to do so with polynomial weights using only one hidden layer.

Explanation: With a perceptron you can have a gate computing "at least K of the inputs are 1". So in the hidden layer you create neurons for "at least 1", "at least 2", "at least 3" … until N.

XOR is simply checking if the total number of 1s is odd.

So you can build the following circuit:



You can see how this works. For odd-number-of-bit inputs 1,3,5,7 etc, the weighted sum will be 1. For even inputs the sum will be 0.

So a threshold of 1 gives you the xorSince there are N inputs and N neurons in the first layer, no. of weights is N^2

IncorrectQuestion 5

0 / 1 pts

How does the number of weights in an XOR network with **arbitrarily many hidden layers and threshold logic gates** grow with the number of inputs to the network?

Hint: See lecture 2, *Slides on “Optimal depth” and “Network size” 113-123*

- Linear
- Between polynomial and exponential
- Faster than exponential
- Polynomial but faster than linear

In the XOR circuit shown in the slides, the number of perceptrons grows linearly with input, and each perceptron has only two intpus and hence a fixed number of weights (2), thus the number of weights too grows linearly with input.

PartialQuestion 6

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

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.

IncorrectQuestion 7

0 / 1 pts

Although we haven't yet covered training of neural networks in the lectures, we can give you this advance bit of information: The number of required training inputs to train a network properly is monotonically related to the number of parameters.

In general, as the depth of a NN increases, at what rate does the number of training observations required to adequately train the network change? (Choose the most appropriate answer)

Slide: lec 2, “The challenge of depth”

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

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). The number of required training inputs is monotonically related to the number of parameters.

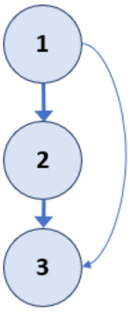
Question 8

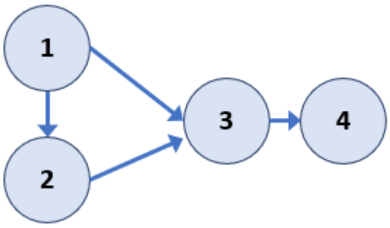
1 / 1 pts

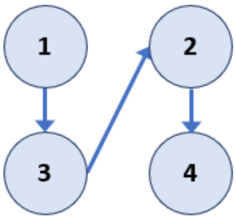
Which of the following NNs have a depth of 3? (Choose all that apply)

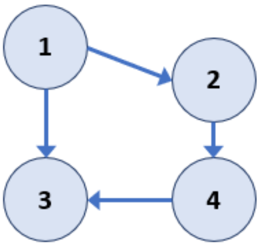
(Note: for the definition of network depth, see the lecture 2 recording)

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

- [ ] 

- [x] 

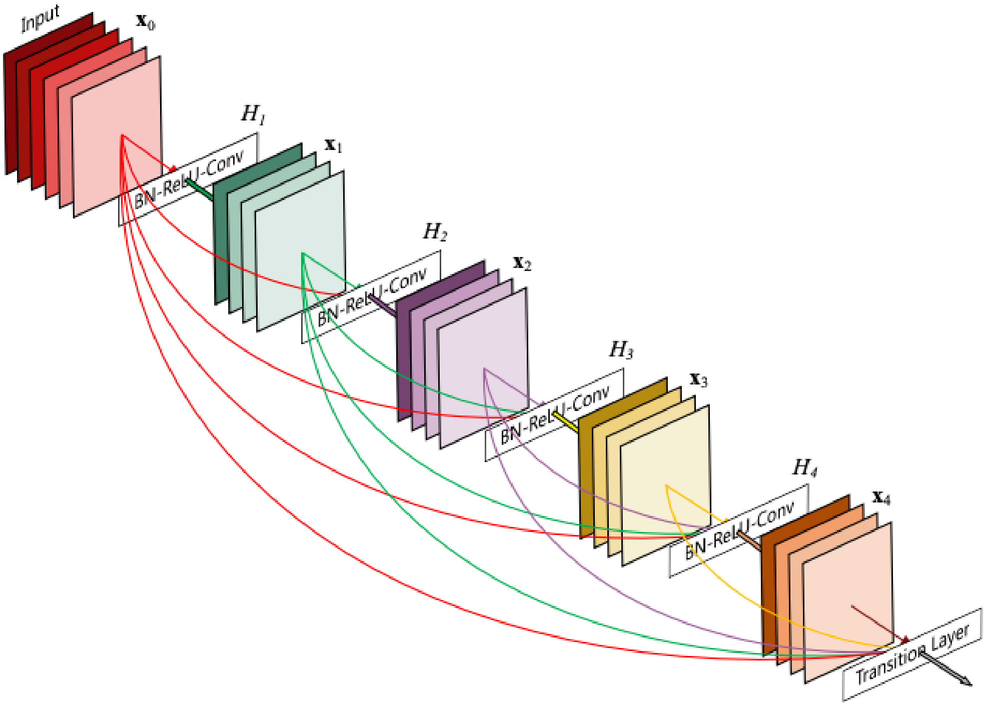
- [x] 

- [x] 

IncorrectQuestion 9

0 / 1 pts

Densenet (2017) is a CNN network architecture that achieved SoTA results with fewer params than many of its contemporaries. Its main idea was the ‘dense block’: a block where each layer output is concatenated into the input of the downstream layers.



Above is a diagram of a dense-block. Each “H_i” consists of Batchnorm->ReLU->Convolution. Each H_i outputs a “x_i” (one stack of squares in the image) .

What is the depth of the dense block above? (numeric answer, int and float are both fine)

Assume that "BN+RELU+CONV" is a single layer. The square planes simply represent the data flowing down between layers and are not layers themselves.

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

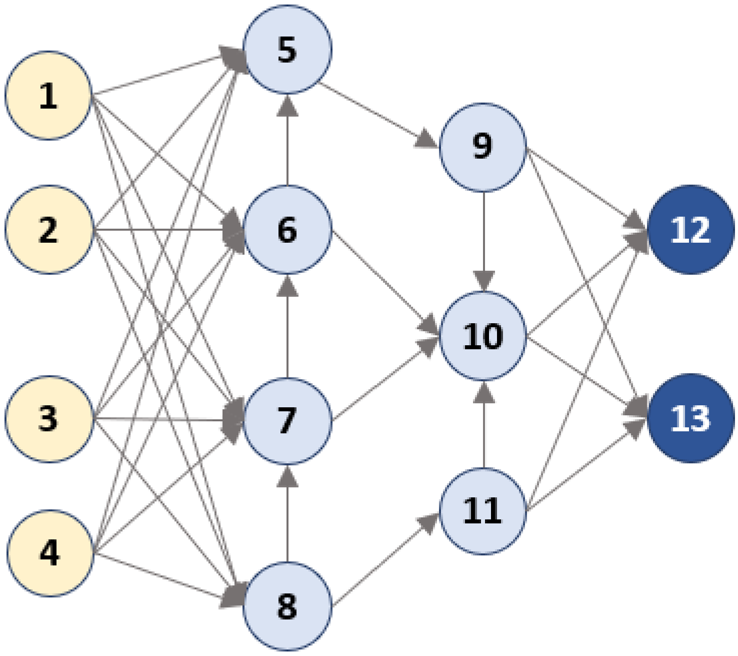
Slide: lec 2, "Deep Structures"

4

The longest path is Input->H_1->H_2->H_3->H_4->Transition: 5 edges.

Question 10

1 / 1 pts



If the yellow nodes are inputs (not neurons) and the dark blue nodes are outputs, which neurons are in layer 2?

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

Hint: lec 2, slides 19-20 on "What is a layer"

- 8
- 7, 11
- 5,6,7,8
- 9, 10, 11

Pay attention to the definitions. Definition of layer #: the neurons reachable (with the same number of edges) along the longest path from source to sink. If a node is reachable multiple times along the longest path, its layer # is the max of those reachable cases (i.e. if some node D was reachable along the longest path 3 times [1,2,3] along the longest path, it's layer number is 3).

The longest path would be from any input -> 8 -> 7 -> 6 -> 5 -> 9 -> 10 -> any output. Node 8 would count as layer 1, as you traverse 1 edge along the longest path to get there. While it is possible to visit 5,6,7,8 via one edge, they are visited later along the longest path. Layer 2 would be 7 AND 11, as 11 is not visited later along the longest path, but is still reachable with the same number of edges.

Quiz Score: 5.5 out of 10