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
LATEST	Attempt 1	98 minutes	7 out of 10

Correct answers are hidden.

Score for this attempt: 7 out of 10
Submitted Jan 17 at 10:05pm
This attempt took 98 minutes.

IncorrectQuestion 1
0 / 1 pts

According to the university's academic integrity policies and 11785 course policies, which of the following practices are **NOT** allowed in this course? Select all that apply.

- [x] Helping another student debug their code
- [ ] Discussing concepts from class with another student
- [x] Posting code in a public post on piazza
- [x] Asking a TA for help debugging your code
- [ ] Discuss quiz solutions with another student before the deadline

- [x] Creating a piazza post asking for help debugging code without prior attempts to debugging it yourself.

Refer to Recitation 0N.

Also See https://www.cmu.edu/policies/student-and-student-life/academic-integrity.html (https://www.cmu.edu/policies/student-and-student-life/academic-integrity.html) . Other than these restrictions, we highly encourage you to discuss course concepts with other students! Post (reasonable) questions on Piazza, and answer others. Learning is most fun with friends :)

Question 2
1 / 1 pts

Is the following statement true or false? Hebbian learning allows reduction in weights and learning is bounded.

Slide: lec 1, "Hebbian Learning"

- ( ) True
- (•) False

Slides 65: If neuron x repeatedly triggers neuron y, the synaptic knob (Weight) connecting x to y gets larger. Hence the weight only increases and a mechanism for weight reduction is not given. Also, the upper bound to which the weight increases to is not define in the learning, making it unbounded.

Question 3
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 u]

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 4
1 / 1 pts

A neural network can compose any function with real-valued inputs (with either Boolean or real-valued outputs) perfectly, given a sufficient number of neurons

Hint: Lec 2, Slide : "MLPs as a continuous valued regression" 127-130

- ( ) True
- (•) False

Question 5
1 / 1 pts

What is the implication of Shannon's theorem on network size (as a function of input size) for Boolean functions?

Slide: lec 2, "Caveat 1: Not all Boolean functions..".

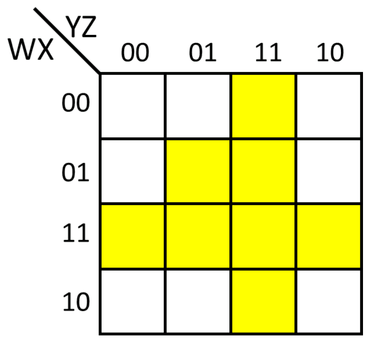
- (•) Nearly all functions require exponential-sized networks
- ( ) Only a relatively small proportion of functions require exponentially sized inputs
- ( ) Most functions require polynomial sized networks.
- ( ) All functions require exponential-sized networks

For those interested in a deep dive, here is Shannon's paper, around page 76 (access using CMU account):

https://ieeexplore.ieee.org/document/6771698 (https://ieeexplore.ieee.org/document/6771698)

IncorrectQuestion 6
0 / 1 pts

What is the fewest neurons needed (including any output layer neurons) for a network to implement the truth table shown by the following Karnaugh map? (numeric answer, int and float are both fine)



Hint: lec 2, "Reducing a Boolean Function". Slide 47 - 50

[3]

O = WX + YZ + XZ, one neuron for each summand and one output neuron (total 4).

IncorrectQuestion 7
0 / 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.
- [ ] Using a threshold network, as deep as you need, to precisely calculate the L1 distance from a point to the origin.
- [x] Using a threshold network with one hidden layer to perfectly classify all digits in the MNIST dataset.

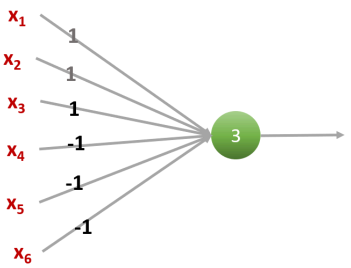
1. There is a finite number of MNIST digits and a single hidden layer network is a universal approximator, so a finite network can classify the data.

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

- [ ] 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

- [ ] ~x1 & ~x2 & ~x3 & x4 & x5 & x6

Question 9

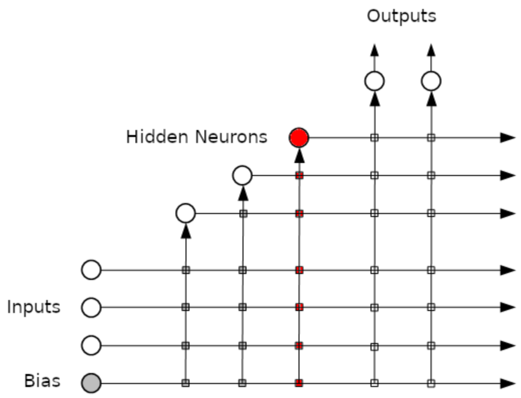
1 / 1 pts

The Cascade Correlation architecture (1990, by CMU's Prof. Fahlman!) is relatively unique in that it iteratively modifies its own architecture during training.

It is initialized with no hidden units; to begin, it only has a number of input channels (determined by the dataset) and a number of output units (which may/may not have non-linear activations). This is akin to a single-layer NN.

We then run this training routine:

1. Train output neurons until performance plateaus
2. If error is below some threshold, break
3. Else, freeze ALL network weights. Add a new hidden unit that receives the ORIGINAL input signals AND the outputs of other hidden neurons as inputs.
4. Train this new unit to correlate with the residual errors from previous runs
5. Once adequately trained, attach its outputs to the inputs of the output units. Freeze this unit and unfreeze the output units.
6. Repeat



(img source **https://towardsdatascience.com/cascade-correlation-a-forgotten-learning-architecture-a2354a0bec92** (https://towardsdatascience.com/cascade-correlation-a-forgotten-learning-architecture-a2354a0bec92))

For example, in the diagram above there are 3 original input channels. Each new hidden unit has 3+n input channels, where n is the layer number from 1~N.

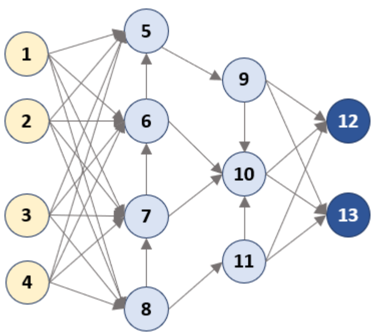
What is the depth of the network above?

(numeric answer, int and float are both fine, also for the definition of network depth, see the lecture 2 recording)

4

Question 10

1 / 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

7

Quiz Score: 7 out of 10