Quiz-01

- Due Sep 1 at 11:59pm
- Points 10
- Questions 10
- Available Aug 30 at 12:01am - Sep 1 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	30 minutes	10 out of 10
LATEST	Attempt 3	30 minutes	10 out of 10
	Attempt 2	29 minutes	6.5 out of 10
	Attempt 1	78 minutes	7.5 out of 10

Correct answers are hidden.

Score for this attempt: 10 out of 10
Submitted Aug 31 at 4:52pm
This attempt took 30 minutes.

Question 1
1 / 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.

- [ ] Asking a TA for help debugging your code
- [x] Posting code in a public post on piazza

- [x] Discuss quiz solutions with another student before the deadline
- [ ] Discussing concepts from class with another student
- [ ] Helping another student debug their code

Question 2

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

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

Question 3

1 / 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”.

- [ ] 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.
- [x] 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).
- [x] 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.

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 function"

- 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..".

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

Question 6

1 / 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 exponentially
- Increases quadratically
- 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 7

1 / 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)

- [x] A single perceptron can compute a majority function.
- [ ] We will require a multilayer perceptron with $\Omega(\exp(N))$ perceptrons to compute a majority function
- [ ] The number of gates in the smallest Boolean circuit of AND, OR and NOT gates that computes the majority function is polynomial in N.
- [x] A fixed-depth Boolean circuit, comprising only AND, OR and NOT gates, will require $\Omega(\exp(N^\alpha))$ gates to compute the majority function ($\alpha > 0$)

A single perceptron can receive all inputs, compute the sum of their values, and use a simple threshold activation of >= N/2. Thus, an MLP is not required.

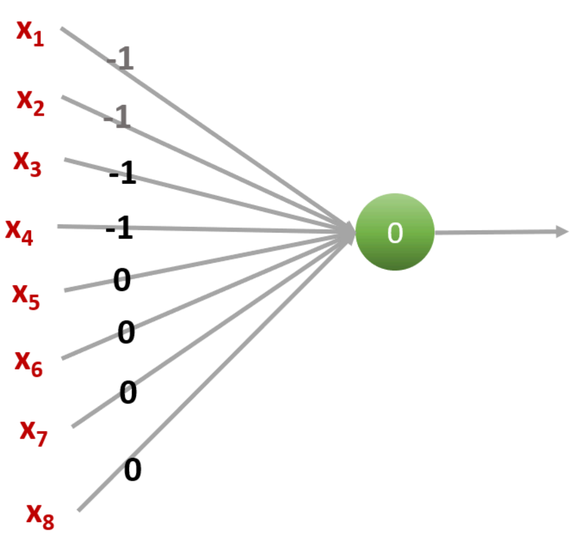
For Boolean circuits, though, the specific lower bound is $\Omega(\exp(N^{1/(d-1)}))$, where d is the depth of the circuit (Smolensky 1993).

Decent explanation at https://eccc.weizmann.ac.il/report/2019/133/download/

Question 8

1 / 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"

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

Question 9

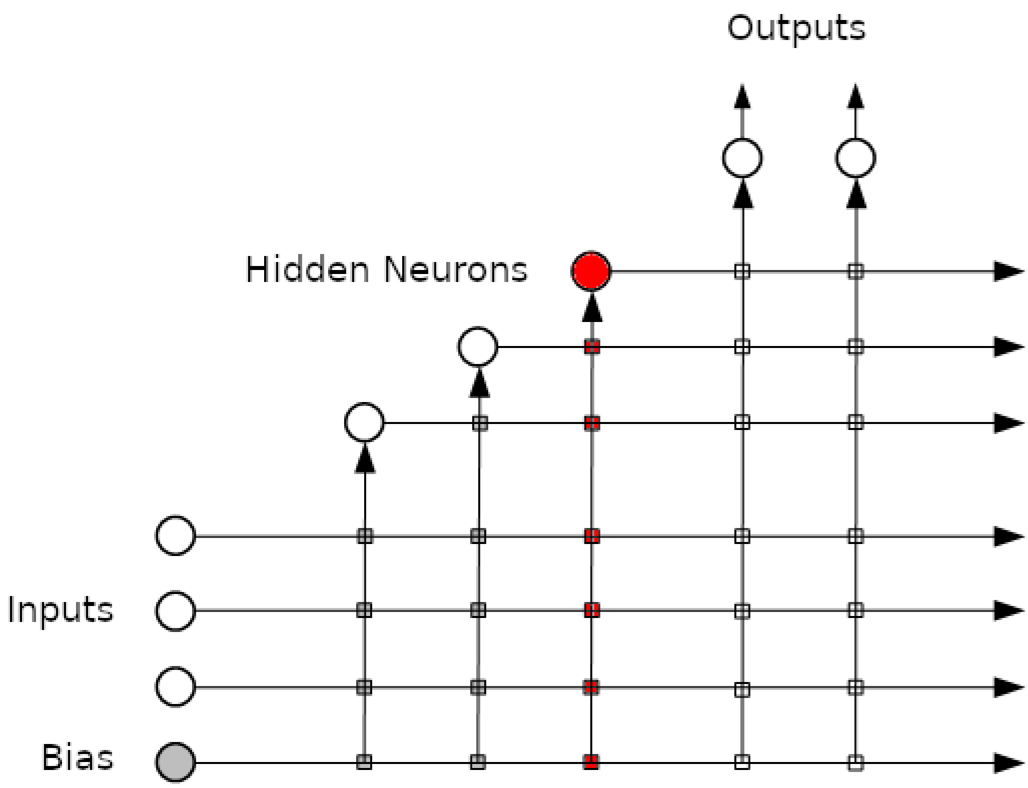
1 / 1 pts

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))

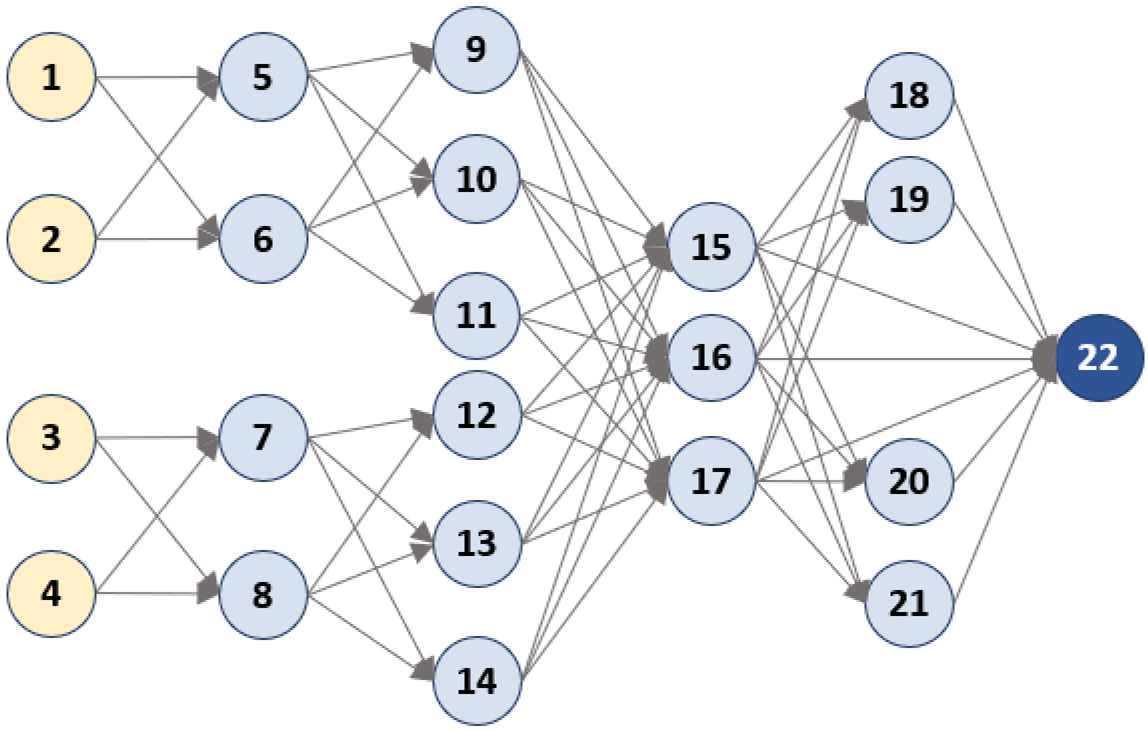
For example, in the diagram above there are 3 original input channels. Each new hidden unit has 3+n input channels, where n is the layer number from 1~N.

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, which neurons are in layer 2?

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

Slide: lec 2, "What is a layer"

- 9, 10, 11
- 15, 16, 17
- 3, 7, 12, 16, 22
- 9, 10, 11, 12, 13, 14

Don’t let the split inputs confuse you. Also remember that the inputs do not count as a layer. The answer must be 9, 10, 11, 12, 13, 14.

Quiz Score: 10 out of 10