5. Introduction to Deep Neural Networks

Bookmark this page

Exercises due Oct 28, 2020 05:29 IST *Completed*

Introduction and Motivation to Deep Neural Networks

0:00 / 0:00 1.25x

Start of transcript. Skip to the end.

So we will connect these abstract little neurons

into highly parallel architectures.

And in deep forward neuron networks,

these units are arranged in layers,

from the input layer, where each unit holds

Video

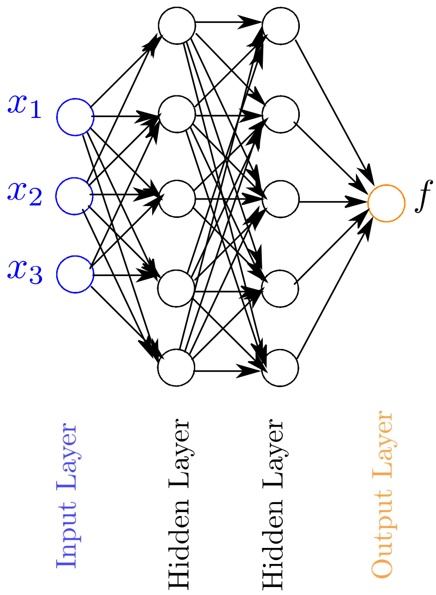
Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

A **deep (feedforward) neural network** refers to a neural network that contains not only the input and output layers, but also hidden layers in between. For example, below is a deep feedfoward neural network of 2 hidden layers, with each hidden layer consisting of 5 units:



One of the main advantages of deep neural networks is that in many cases, they can learn to extract very complex and sophisticated features from just the raw features presented to them as their input. For instance, in the context of image recognition, neural networks can extract the features that differentiate a cat from a dog based only on the raw pixel data presented to them from images.

The initial few layers of a neural networks typically capture the simpler and smaller features whereas the later layers use information from these low-level features to identify more complex and sophisticated features.

Representation Power of Neural Networks: 1

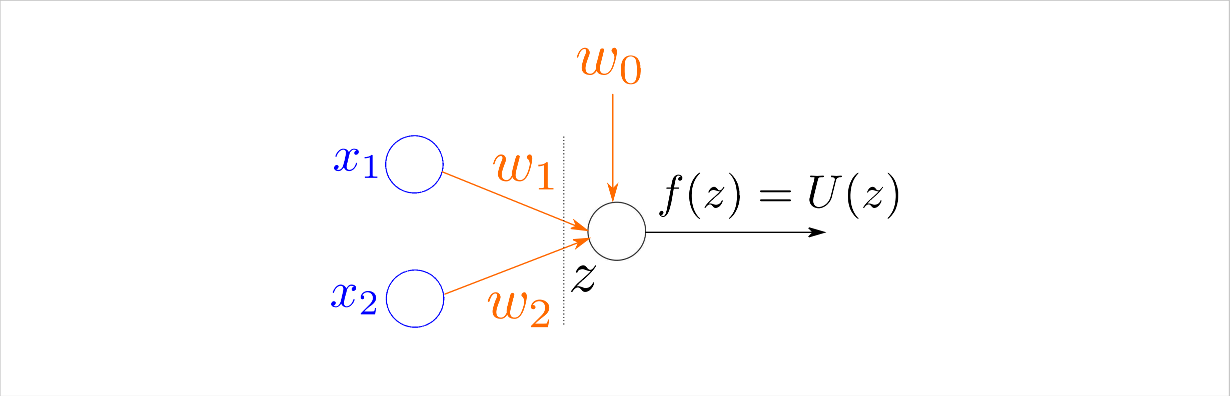
3/3 points (graded)

In these two problems, we are going to explore how a neural network can represent any given binary functions. We will start in this problem by building the logic NAND function using a simple neural network.

The logic NAND function is defined as

$$y \;=\; \mathrm{NOT}\,(x_1\;\;\mathrm{AND}\;\;x_2)$$

where x_1 and $x_2 \in \{0,1\}$ are binary inputs (and 1 denotes True and 0 denotes False).



We will use the above simple neural network with $z = w_1x_1 + w_2x_2 + w_0$ and the activation function $f$chosen to be the unit step function $U\,(z)$:

$$U\,(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0 \end{cases}.$$

Find w_0, w_1, and w_2 such that the output of the neural network $y = U\,(z)$ gives the NAND function as a function of x_1 and x_2.
(Different correct answers will be accepted.)

$w_0 =$ 2 ✔ **Answer:** See solution

$w_1 =$ -1 ✔ **Answer:** See solution

$w_2 =$ -1 ✔ **Answer:** See solution

Solution:

The NAND function outputs the following:

$$\mathrm{NAND}\,(x_1, x_2) \;=\; \begin{cases} 0 & \text{if } (x_1,\, x_2) = (1,1) \\ 1 & \text{otherwise} \end{cases}.$$

Since the activation function is the step function $U\,(z)$, we need $z = \leq 0$ when $(x_1,\, x_2) = (1,1)$, and $z > 0$ for $(x_1,\, x_2) = (0,0)$, $(0,1)$, or $(1,0)$. Since $z = w_0 + w_1x_1 + w_2x_2$, the above conditions translate to the following inequalities

$$\begin{aligned} w_0 + w_1 + w_2 &\leq 0 \\ w_0 &> 0 \\ w_0 + w_1 &> 0 \\ w_0 + w_2 &> 0. \end{aligned}$$

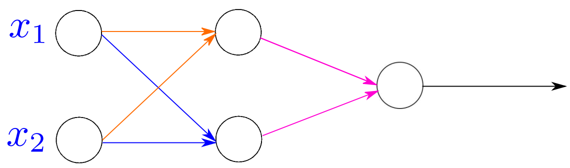
A valid example is $w_0 = 3, w_1 = -2, w_2 = -2.$

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Representation Power of Neural Networks: 2

2/2 points (graded)
Using the NAND function only as the basic neural network unit, we can build larger neural networks to implement other logic functions. For example, the follow neural network implements the logic AND function:

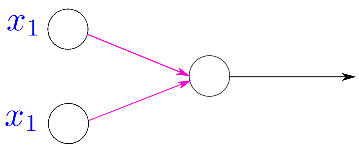


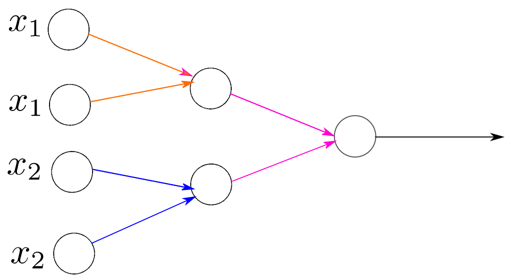
Note: Here, each pair of edges of the same color along with the nodes they are connected to form a neural network unit that represents the NAND function. (They do not represent values of inputs or outputs). In the example above, x_1 and x_2 are inputs to two NAND units, and are connected to output of respective units by the blue and orange arrows.

(Check that these output the correct values.)

Which logic function does each of the following neural networks implement?

(Choose one for each column.)





○ AND function	○ AND function
◉ NOT function	○ NOT function
○ OR function	◉ OR function
○ XOR function	○ XOR function

Solution:

NAND function is known as a universal logic function, which can be used to implement any boolean functions, including also XOR, without the use of any other type of function (except for the identity and zero function). Use De Morgan's law in boolean algebra.

$$\text{NOT}\,(x_1 \text{ AND } x_1) = \text{NOT } x_1,$$

and

$$\text{NOT}\,(\,\text{NOT}\;(x_1)\;\text{AND}\;\text{NOT}\,(x_2)) = \text{NOT}\,(\,\text{NOT}\,(x_1\;\text{OR}\;x_2)) = x_1\;\text{OR}\;x_2.$$

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Unit 3 Neural networks (2.5 weeks):Lecture 8. Introduction to Feedforward Neural Networks / 5. Introduction to Deep Neural Networks

Add a Post

Show all posts by recent activity

- Some drawing that helped me. — 3
  Pinned Community TA
- Representation Power of Neural Networks: 2 — 15
  The options given are not exhaustive. Or the picture chosen to display is wrong ?
- De Morgan's Law refresher — 1
  This is a helpful read if you need a quick refresher on De Morgan's Laws https://brilliant.org/wiki/de-morgans-laws/
- This website helped me... — 1
  http://www.ee.surrey.ac.uk/Projects/Labview/gatesfunc/#introduction Plus understanding two key points: 1. Each node is an NAND nod...
- Representation Power of Neural Networks: 1 — 6
  Grader says me wrong with my answer: w1=-3, w2=-3,w3=4. However I disagree. It says "(Different correct answers will be accepted.)". M...
- Multiple Answers to part 2 ? — 3
  Can it be that there are multiple solutions to part 2? I found that the second representation is valid for either AND, XOR & OR when you ...

Previous Next

Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate