Week 7 Exercises

The questions below are due on Sunday October 22, 2017; 11:00:00 PM.

You are not logged in.

If you are a current student, please Log In (https://introml.mit.edu/fall17/exercises/ex07?loginaction=login) for full access to this page.

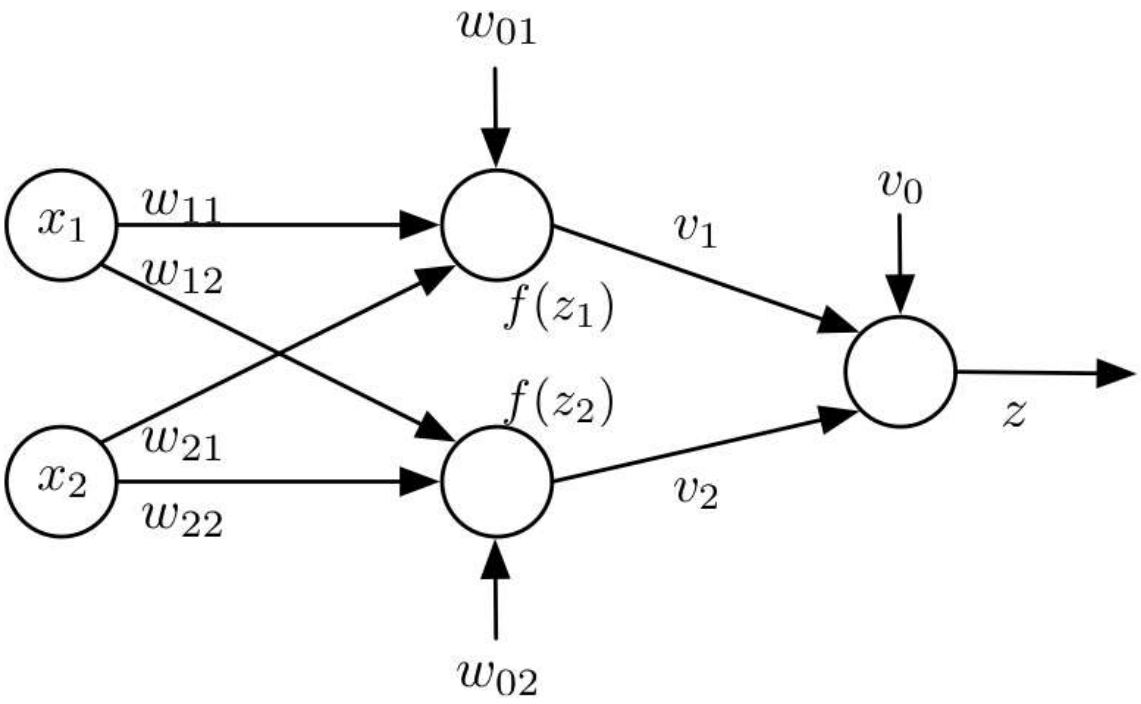
- Videos
  - Week 7, Lecture 1 (https:///introml.mit.edu/lecture_videos/lec_week7_part1.mp4)
  - Week 7, Lecture 2 (https:///introml.mit.edu/lecture_videos/lec_week7_part2.mp4)
- Class Notes for Week 7 (https://introml.mit.edu/__STATIC__/fall17/exercises/ex07/Wk7_notes.pdf)
- Required Exercises

1) PREDICT WITH STEPS

Consider the following data set:

```
X = np.array([[0, 1, 2],
              [0, 1, 2]])
Y = np.array([[0, 1, 0]])
```

We will be looking at the behavior of the following simple network:



Assume each unit has the step activation function

$$f(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$$

Let the weights in the first layer be be: $w_{0,1} = -0.5$, $w_{1,1} = 1$, $w_{2,1} = 0$ $w_{0,2} = 1.5$, $w_{1,2} = -1$, $w_{2,2} = 0$

1. Enter a matrix where each column represents the outputs of the hidden units ($f(z_1)$ and $f(z_2)$) for each of the input vectors in `X`.

Enter a Python list of lists [[a,b,c],[d,e,f]], each list is a row of the matrix.

```
[[0., 1., 1.], [1., 1., 0.]]
```

2. Pick weights for the second layer v_0, v_1, v_2 so that the desired outputs are predicted correctly.

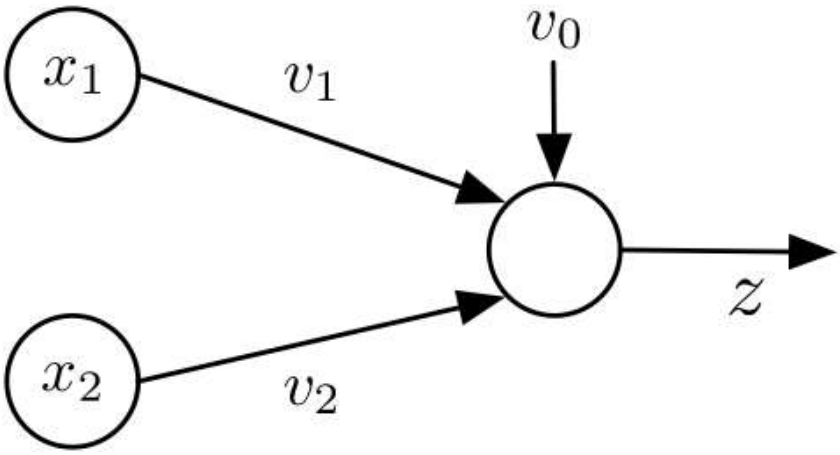
Enter a Python list of 3 numbers [v_0, v_1, v_2]

```
[-1., 1., 1.]
```

2) LEARN

Assume a single linear unit with two inputs, x_1 and x_2. The output of the unit is z where

$$z = v_1 x_1 + v_2 x_2 + v_0$$



Assume the initial weights are $v_0 = 1, v_1 = 1, v_2 = 1$, and the learning rate is 0.5 (not usually a good idea, but okay for now).

The current training example is $x^{(i)} = [1, 2]^T, y^{(i)} = -1$.

1. What is the output value z, given this input and the current weights?

Enter a number

```
4
```

2. What will the values of weights v_0, v_1, v_2 be after one step of stochastic gradient descent using SVM loss with $\lambda = 0$?

Enter a Python list of 3 numbers $[v_0, v_1, v_2]$

```
[0.5, 0.5, 0. ]
```

3. What would the output value z be, for this same input x, with these new weights?

Enter a number

```
1
```

4. What would happen to the v_i if we did another SGD update, for that same point, with learning rate 0.5, as before?

Enter a Python list of 3 numbers $[v_0, v_1, v_2]$

```
[ 0., 0., -1.]
```

5. Now what would the output be?

Enter a number

```
-2
```

6. What if we do one more update, for that same point?

Enter a Python list of 3 numbers $[v_0, v_1, v_2]$

```
[-2.5, -2.5, -3. ]
```