Section 1: Deep learning

1. Gradient vanishing, gradient exploding
2. Attention
3. What is the difference between RNN and CRF
4. Object detection in an image/video : fast R-CNN
5. GAN vs. CNN
6. Why Kernel filter size=3, 5, 7? centra pixel, understanding, padding
7. Googlenet vs. ResNet: skip, inception
8. Batch normalization: scaling
9. How to apply deep learning to NLP? Image caption, LSTM
10. SIFT feature: invariant
11. Auto-encoder: learn the representation for each class; flood
12. questions:

    Assume that we have a feed forward neural net of N hidden layers and M inputs. Assume that all the weights of the network (W) are also known. The activation function for all the neurons in the hidden layers is $ReLU$ and the output neuron activation function is $Sigmoid$. The output of this network classifies the inputs as good or bad based on some known threshold T. If the result of the output layer is greater than T then the input is classified as good and if it is not, it is classified as bad.

    For an input vector X, x_i is the ith input of the network ($0 < i <$ M). If X is classified as bad, what should be the x_i (i.e. target value) to classify the X as a good assuming all other inputs $\left(x_j\ \forall j \le M, j \neq i\right)$ are constant. Ideally, we want to find x_i with a single pass. Propose your approach to find x_i and provide all the necessary equations for solving x_i idealy in one pass. You also need to provide an example that shows the accuracy of your solution using the provided equations.
13.

Section 2. Coding:

You are given a m x n 2D grid initialized with these three possible values.

-1 - A wall or an obstacle.

0 - A gate.

INF - Infinity means an empty room.

Fill each empty room with the distance to its nearest gate. If it is impossible to reach a gate, it should be filled with INF.

Example
Given the 2D grid:

```
INF  -1  0  INF

INF INF INF  -1

INF  -1 INF  -1

  0  -1 INF INF
```

return the result:

```
  3  -1   0   1

  2   2   1  -1

  1  -1   2  -1

  0  -1   3   4
```

```
class Solution:
        """
        @param: rooms: m x n 2D grid
        @return: nothing
        """
    def wallsAndGates(self, rooms):

        For i in xrange(len(rooms)):
                For j in xrange(len(rooms[0])):
                        if rooms[i][j] == INF:
                                Rooms[i][j] = 1

        For i in xrange(len(rooms)):
```

```
For j in xrange(len(rooms[0])):
        if rooms[i][j] == 1:
                while(rooms[i][j] != -1)
```