Deep Learning	CS60010
Spring Semester, 2017	Mid-Sem
Maximum Marks: 50	Time Limit: 2 Hours

This exam contains 4 pages (including this cover page) and 7 problems.

You may *not* use your books and notes for this exam. Be *precise* in your answers. All the *sub-parts* of a problem should be answered at *one place* only. On multiple attempts, *cross* any attempt that you do not want to be graded for.

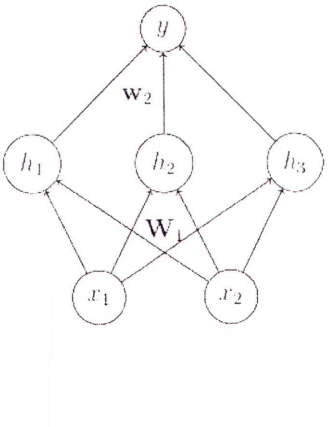
There are no clarifications. In case of doubt, you can take a valid assumption, state that properly and continue.

Answer Part A in the first FIVE pages of the answer book, and Part B starting from Page 6 of the Answer Book.

PART A

1. (9 points)
   (a) (2 points) Define a loss function for classification. Assume that the training examples are $\{(x_i, y_i)\}$, the function learned is $f(x_i; \theta)$ and there are k classes.
   (b) (1 points) What are the hyperparameters in a neural network which you can adjust to deal with overfitting?
   (c) (2 points) How do you choose those hyperparameters to deal with overfitting?
   (d) (2 points) You design a neural network and after training you find that the training error and cost are high and your validation cost and error are almost equal to it. Answer the following questions:
      i. What does this mean for your model?
      ii. What actions would you take?
   (e) (2 points) Why is the Rectified Linear Unit (ReLU) often preferred over the sigmoid function as an activation function in deep networks? [1-2 sentence]

2. (3 points) Linear regression can be made more powerful using a basis function expansion, i.e. by using a function Φ which maps each data point x to a feature vector $\Phi(x)$. We may analogously represent this using a feed-forward neural net with one hidden layer, where one set of weights is held fixed. (Such a network is shown in the following figure.)

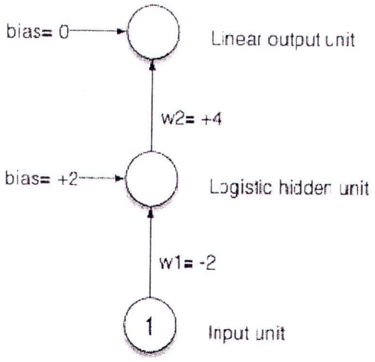


(a) Which set of weights is held fixed?

(b) Briefly state what the hidden activations (h) and both sets of weights (w_1 and w_2) correspond to.

3. (4 points) Consider the small neural network given below with one input unit, one hidden unit (logistic), and one output unit (linear). Consider one training case for which input value is 1 and the output is 1. Suppose you use the loss function

$$E = (t - y)^2/2$$



(a) What is the output of the hidden unit and the output unit for this training case?

(b) What is the loss for this training case?

(c) What is the derivative of the loss w.r.t. w2 for this training case?

(d) What is the derivative of the loss w.r.t. w1 for this training case?

4. (9 points)

(a) (4 points) Suppose we have a 3-dimensional input $\vec{x} = (x_1, x_2, x_3)$ connected to 4 neurons with the exact same weights $\vec{w} = (w_1, w_2, w_3)$ where: $x_1 = 2$, $w_1 = 1$, $x_2 = 1$, $w_2 = 0.5$, $x_3 = 1$, $w_3 = 0$, and the bias $b = 0.5$. We calculate the output of each of the four neurons using the input $\vec{x}$, weights $\vec{w}$ and bias b.

If $y_1 = 0.95$, $y_2 = 3$, $y_3 = 1$, $y_4 = 3$, make valid guesses for the neuron types of y_1, y_2, y_3 and y_4. The possible types are Linear, Binary Threshold, Logistic Sigmoid and Rectified Linear.

i. Type of y_1

ii. Type of y_2

iii. Type of y_3

iv. Type of y_4

(b) (3 points) Explain what effect the following operations will have on the (i) bias and (ii) variance of your model. The possible answers are 'increases', 'decreases' or 'no change'.

i. Regularizing the weights in a linear/ logistic regression model

ii. Increasing the number of hidden units in an artificial neural network.

iii. Using dropout to train a deep neural network.

(c) (2 points) Fill in the value for w in this example of gradient descent in $E(w)$. Calculate the weight for iteration 2 of gradient descent where the step-size is $\eta = 1.0$ and the *momentum coefficient* is 0.5. Assume the *momentum* is initialized to 0.0.

Iteration	w	$-\nabla_w E$
0	1.0	1.0
1	2.0	0.5
2		0.25

PART B

5. (10 points) Suppose you are using CBOW architecture for learning the word vectors. During the training phase, suppose the input (context word) vector for each word c is given by u_c, and the output (center word) vector for each word o is given by v_o. Also suppose that you are using a $d-$dimensional representation and a window of size k (i.e., k words preceding and following the center word). Also assume the vocabulary size of W. Suppose you are at a particular window during your training [with words $t-k, \ldots, t, \ldots, t+k$].

   (a) (3 points) What are the dimensions of the input, hidden and output layers, and what will be the scores at each of these in terms of the current window and vectors defined above?

   (b) (3 points) What will be the cross-entropy error?

   (c) (4 points) Suppose you want to use this single window to update your parameters. Derive the parameter update formulation for the output word vector v_t for the current window.

6. (8 points) Answer the following questions (very briefly, to the point, with required equations. *Essays will not be graded*):

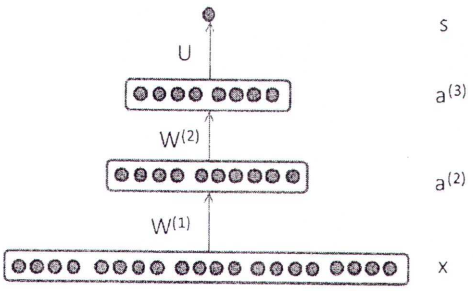
   (a) (3 points) You can use one of the architectures (CBOW, Skip-gram etc.) to learn the representations of thw words. How would you learn the representation of phrases instead (e.g., 'New York Times' etc.)?

   (b) (3 points) Suppose you want to obtain multilingual word embeddings (words from 2 different languages mapped to the same space). However, you do not have a parallel corpus. How would you achieve this? You may assume that you have access to a dictionary between these two languages.

   (c) (2 points) Suppose the word vectors for a given word w is given by v_w. How would you use the word vectors thus obtained to solve an analogy problem, i.e., you need to predict the missing word here: $a : b :: ? : d$.

7. (7 points) Consider the following neural network. Also, suppose that you also want to update the weights of your word vectors (i.e., the input layer). Derive the derivative of the score s with respect to the input vectors x as well as the biases $b^{(1)}$ and $b^{(2)}$. [Hint: Suppose the local error signals are given by $\delta^{(3)}$ and $\delta^{(2)}$ corresponding to weights $W^{(2)}$ and $W^{(1)}$, respectively. You may use the error signals to denote the above weight updates.]

$$\begin{aligned} x &= z^{(1)} = a^{(1)} \\ z^{(2)} &= W^{(1)}x + b^{(1)} \\ a^{(2)} &= f\left(z^{(2)}\right) \\ z^{(3)} &= W^{(2)}a^{(2)} + b^{(2)} \\ a^{(3)} &= f\left(z^{(3)}\right) \\ s &= U^T a^{(3)} \end{aligned}$$


s
U
a(3)
W(2)
a(2)
W(1)
x