

# Assignment 2

CS5370: Deep Learning for Vision/AI5100: Deep Learning/AI2100: Deep Learning  
IIT-Hyderabad  
Jan-Apr 2021

**Max Marks:** 50  
**Due:** 24th Feb 2021 11:59 pm

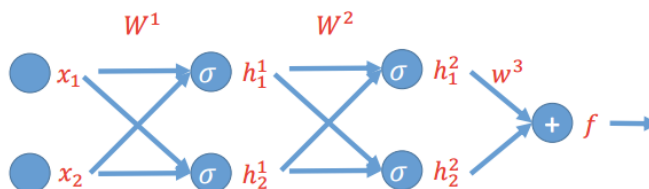
## Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single ZIP file, named `<Your_Roll_No>_Assign2`, with all your solutions, including code.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 10 grace days for late submission of assignments, of which upto 4 grace days can be used for a single assignment. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the Marks and Grace Days document, soon to be shared under the course Google drive.
- You have to use PYTHON for the programming questions.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

## 1 Theory (15 marks)

*You can submit your response as a PDF document, which can be typed out in LaTeX/Word, or handwritten and scanned. If handwritten, please ensure legibility of answers.*

1. (2 marks) Consider the problem of computing a homography from point matches that include outliers. If 50% of the initial matches are correct, how many iterations of the RANSAC algorithm would we expect to have to run in order to have a 95% chance of computing the correct homography?
2. (1 mark) Consider a 3-layer neural network (as shown in the diagram below):



$$h^1 = \sigma(W^1 x), h^2 = \sigma(W^2 h^1), f(x) = \langle w^3, h^2 \rangle$$

where  $W$  denotes a matrix,  $w$  denotes a vector and  $\sigma(\cdot)$  is applied elementwise. Compute  $\frac{\partial f}{\partial w_{i,j}^1}$ .

3. (1 mark) You are training a 3-layer neural network and would like to use backprop to compute the gradient of the cost function. In the backprop algorithm, one of the steps is to update  $\Delta_{ij}^{(2)} := \Delta_{ij}^{(2)} + \delta_i^{(3)} * (a^{(2)})_j$  for every  $i, j$ . Can you rewrite the above equation for all the weights in layer 2 in vector form? (HINT:  $\Delta^{(2)} := \Delta^{(2)} + \dots??$ )
4. (2 marks) Consider a neural network with 1 hidden layer,  $d$  inputs,  $M$  hidden units and  $c$  output units. Write down an expression for the total number of weights and biases in the network. Consider the derivatives of the error function with respect to the weights for one input example only. Using the fact that these derivatives are given by equations of the form  $\partial E^n / \partial w_{kj} = \delta_k z_j$ , write down an expression for the number of independent derivatives.
5. (4 marks) It is possible to mathematically show that minimizing the sum-of-squared error in a neural network is equivalent to/derived from the principle of Maximum Likelihood Estimation. (Read the first few chapters of <http://neuralnetworksanddeeplearning.com/> if you need help understanding this.) Now, consider a model in which the target data has the form:

$$\mathbf{y}_n = f(\mathbf{x}_n; \mathbf{w}) + \epsilon_n$$

where  $\epsilon_n$  is drawn from a zero mean Gaussian distribution having a fixed covariance matrix  $\Sigma$ . Derive the likelihood function for this dataset drawn from this distribution, and write down the corresponding error function. Such an error function is called *generalized least squares*. The usual sum-of-squares error function corresponds to the special case  $\Sigma = \sigma^2 \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix.

6. (2.5+2.5=5 marks) One of the major concerns while training a large neural network is the presence of symmetries in the weight space. For a given set of model parameters  $\{\theta\}$ , assuming a simple scaling i.e.  $\{\gamma\theta_l\}, \gamma \in R$  for weights in layer  $l$ , and multiplying the next layer's weights by  $1/\gamma$ , the loss function value remains the same for both weight configurations. This is called *scale symmetry*.
  - (a) Can you give any one problem caused by scale symmetry while training deep networks? Substantiate your answer with a proper reason or example.
  - (b) Provide any one other symmetry that a deep neural network suffers from (other than scale symmetry) in its weight space. Substantiate your answer with a proper reason or example.

## 2 Programming (35 marks)

- The programming questions are shared in “Assignment\_2.zip”. Please follow the instructions in the notebook. Turn-in the notebook via Google Classroom once you finish your work.
- Marks breakdown is as follows:

### Part 1

- Question 1: 2 marks
- Question 2: 9 marks
- Question 3: 4 marks

### Part 2

- Implementation of the network: 8 marks
- Question 4: 2 marks
- Question 5: 2 marks

- Question 6: 2 marks
- Question 7: 2 marks
- Question 8: 2 marks
- Question 9: 2 marks

**Part 3:** Ungraded