## Assignment 1

## **Re-submit Assignment**

**Due** Jan 30 by 10am **Points** 40 **Submitting** a file upload

This assignment is supposed to be submitted either through Canvas or on paper by January 30 10:00AM (start of the class) to the TA.

1. Consider N data points independent and uniformly distributed in a p-dimensional unit ball (for every  $x\in \mathbb{B}$ ,  $\|x\|^2\leq 1$ ) centered at the origin. The median distance from the origin to the closest data point is given by the expression:

$$d(p,N) = \left(1-rac{1}{2}^{rac{1}{N}}
ight)^{rac{1}{p}}$$

Prove this expression (8 points). Compute the median distance d(p,N) for N=10000, p=1000 (2 points).

Hint: The volume of a ball in p dimensions is  $V_p\left(R
ight)=rac{\pi^{rac{p}{2}}}{\Gamma\left(rac{p}{2}+1
ight)}R^p$  , where R is the radius of the ball, and  $\Gamma$ 

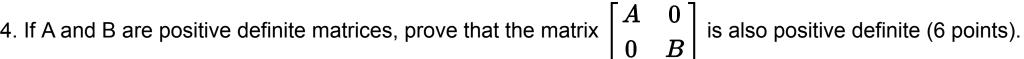
is the Gamma function. A point being the closest point to the origin means that there is no point in the N data points that has a smaller distance to the origin than itself. What is the probability for that to happen with a uniform distribution in a unit ball?

2. Compute the gradient  $abla f\left( x
ight)$  and Hessian  $abla^{2}f\left( x
ight)$  of the function

$$f(x)=(x_1+x_2)(x_1x_2+x_1x_2^2)$$

Find at least 3 stationary points of this function (3 points). Compute the Hessian at each stationary point you found (5 points). Show that  $x=\left[\frac{3}{8},-\frac{6}{8}\right]^{\top}$  is the only local maximum of this function (2 point).

3. Show that the function  $f(x)=8x_1+12x_2+x_1^2-2x_2^2$  has only one stationary point (2 points), and that it is neither a minimum nor a maximum, but is a saddle point (2 points).



5. Derive the forward (computing  $z_2$ ) (4 points) and backpropagation (computing the gradients) (6 points) functions for a 1-hidden layer neural network with k hidden nodes, a sigmoidal transformation function (for each node and the final layer generating  $z_2$ ):  $\mathbf{z}_1 = \frac{1}{1+\exp(-\mathbf{W}^\top\mathbf{x}-\mathbf{b}_1)}$ ,  $z_2 = \frac{1}{1+\exp(-\mathbf{w}_2^\top\mathbf{z}_1-b_2)}$  and a 2-class cross-entropy loss function  $y^*\log z_2 + (1-y^*)\log(1-z_2)$  at the final layer for the neural network (hint: you are allowed to combine the sigmoid and cross-entropy layers into a single one, it might be easier).  $\mathbf{W}$  is of dimensionality  $d \times k$ , x is  $d \times 1$ ,  $\mathbf{z}_1$  is  $k \times 1$ .