## **Question 1.** [10 MARKS]

**Part (a)** [1 mark]
For a convex optimization problem which of the following will never converge:

i. batch gradient descent
ii. mini-batch gradient descent
iii. stochastic gradient descent
iv. only (ii) and (iii) will not converge
v. none of the above

**Part (b)** [1 mark]
What is the computational complexity of sorting a list of n numbers?

i. O(n)
ii. O(nlogn)
iii. O($n^2$)
iv. O($n^2$logn)
v. none of the above

**Part (c)** [1 mark]
What is the computational complexity of the following sample code?

```
a = n
while a > 1:
    a=a*0.9
```

i. O(n)
ii. O(sqrt(n))
iii. O(logn)
iv. O(nlogn)
v. none of the above

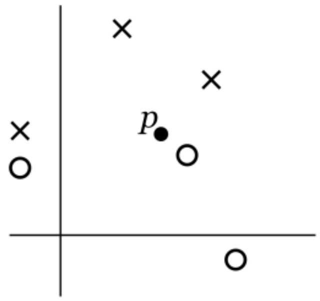
**Part (d)** [1 mark]
L2 regularization should be applied on which of the following:

i. learning rate
ii. bias
iii. weights
iv. both (ii) and (iii)
v. none of the above

**Part (e)** [1 mark]
Which of the following is true about optimizers?

i. We can speed up training by applying a different learning rates on each weight.
ii. Reducing the batch size will always speed up training time.
iii. You wouldn’t use SGD to train a linear regression model.
iv. You can only find the global minimum when the problem is convex.
v. none of the above

**Part (f)** [1 mark]
We would like to use 1-Nearest Neighbour to classify point *p* using the data to the right. What is our prediction if we use cosine similarity distance? Euclidean distance?



i. Cosine distance: O, Euclidean distance: O
ii. Cosine distance: X, Euclidean distance: O
iii. Cosine distance: O, Euclidean distance: X
iv. Cosine distance: X, Euclidean distance: X

**Part (g)** [1 mark]
Which of the following about a high variance model (in the context of bias-variance tradeoff) is true, compared to a high bias model?

i. A high variance model is more prone to underfitting.
ii. A high variance model requires more training data to train.
iii. A high variance model will have a higher training accuracy.
iv. A high variance model should be trained with a smaller batch size.
v. Both (ii) and (iii) are true.

**Part (h)** [1 mark]
For which of the following problems would you choose a machine learning technique?

i. Determining where a piece of Python code prints out the value “Hello, world”.
ii. Determining whether a photograph is in black and white or in colour.
iii. Determining whether a photograph is of a young person or an old person.
iv. All of the above.
v. Only (ii) and (iii).

**Part (i)** [1 mark]
Which of the following will most likely produce a more noisy training curve?

i. Decreasing the batch size.
ii. Decreasing the learning rate.
iii. Decreasing the size of the training set.
iv. Increasing the size of the training set.
v. Increasing the number of parameters of the neural network.

**Part (j)** [1 mark]
Which of the following helps prevent overfitting?

i. Increasing the number of layers in a neural network.
ii. Training for more epochs.
iii. Using a larger batch size.
iv. Using a larger training set.
v. Both (iii) and (iv).

## **Question 2.** [10 MARKS]

Circle either “True” or “False” for each of the below statements.

a True False Data augmentation techniques can be applied to both the training and test data to limit overfitting.

b. True False Gradient descent cannot get stuck in a local minimum when training a logistic regression model.

c. True False PCA can be applied on a correlation matrix

d. True False It is not necessary to have a target variable for applying dimensionality reduction algorithms.

e. True False In general, a mixture of gaussian model will perform better if you restrict the covariance matrix to be diagonal.

f. True False L2 regularization works better when the data is standardized.

g. True False You cannot use ROC to measure the performance on a multiclass classification problem.

h. True False You can use squared-error loss to solve classification problems.

i. True False Importance sampling allows us to sample from a known distribution when using the true distribution is intractable.

j True False Hyperparameters are tuned and updated during backpropagation.

## **Question 3.** [8 MARKS]

Provided below is sample code for k-means clustering. You may assume all the necessary libraries are includes and that there are no syntax errors.

```
def kmeans(x, k, n_iter):
    ind = np.random.randint(0, len(x)-1, k)
    centroids = x[ind, :]
    distances = compute_distances(x, centroids)
    labels = np.array([np.argmin(i) for i in distances])
    for _ in range(n_iter):
        centroids = []
        for ind in range(k):
            cent = x[points==ind].mean(axis=0)
            centroids.append(cent)
        centroids = np.vstack(centroids)
        distances = compute_distances(x, centroids)
        labels = np.array([np.argmin(i) for i in distances])
    return (labels, centroids)
def compute_distances(x, centroids):



    return distances
```

**Part (a)** [2 mark]
Fill in the `compute_distances` function to obtain the Euclidean distance between all points and *k* clusters.

**Part (b)** [4 mark]
What would you add and/or change in the `kmeans` function to ensure that we converge to the global minimum? Provide your answer(s) in/next to the code above. Assume *k* is fixed.

**Part (c)** [2 mark]
As the dimensionality of the input data increases what happens to the Euclidean distance measurements?

## **Question 4.** [8 MARKS]

Consider a dataset of physiological measurements of elite athletes.

| | Meaning of variables |
|---|---|
| X1 | Skinfold thickness |
| X2 | Grip strength |
| X3 | Maximal vertical jump capacity |
| X4 | Maximal lactate steady state (endurance) |
| X5 | Maximum oxygen uptake (aerobic fitness) |
| X6 | Mean corpuscalar hemoglobin count (resistance to and recovery from fatigue) |

| | Meaning of variables |
|---|---|
| X7 | Anaerobic power |
| X8 | Maximum heart rate |
| X9 | Muscle mass |
| X10 | Muscle fatigue onset time |
| X11 | Pulmonary ventilation rate |

A principal component analysis was performed yielding the following results.

Eigenvalues = $[7.49\ 3.23\ 1.84\ 0.26\ 0.20\ 0.13\ 0.08\ 0.05\ 0.04\ 0.02\ 0.01]^T$

The first five eigenvectors are:

$$\text{Eigenvectors} = \begin{vmatrix} 0.075 & -0.034 & -0.130 & 0.241 & 0.019 \\ 0.010 & -0.109 & 0.007 & -0.122 & -0.008 \\ -0.147 & -0.102 & -0.164 & 0.345 & 0.027 \\ 0.397 & -0.032 & -0.033 & -0.109 & -0.006 \\ 0.833 & -0.165 & -0.113 & -0.220 & -0.011 \\ -0.232 & -0.891 & -0.285 & -0.193 & -0.032 \\ 0.004 & -0.003 & 0.040 & 0.029 & 0.004 \\ 0.034 & -0.166 & 0.208 & -0.026 & 0.008 \\ -0.234 & 0.109 & 0.311 & -0.783 & -0.066 \\ -0.108 & 0.341 & -0.849 & -0.294 & -0.062 \\ 0.016 & -0.001 & 0.038 & 0.095 & -0.995 \end{vmatrix}$$

**Part (a)** [2 marks]

If we would like to capture 90% of the variance, what would you recommend as the dimensionality? Justify your answer.

**Part (b)** [2 marks]
Given a sample x = [0.1 -0.3 0.4 -0.1 0.2 0.6 -0.2 0.5 -0.1 0.9 -0.1]$^T$ what would be its new coordinates using the top two principal components?

**Part (c)** [2 marks]
How would you go about determining a name or label for the principal components? Try to assign a semantic label for the first two principal components.

**Part (d)** [2 marks]
Can SVD and PCA produce the same projection result? If yes, under what condition are they the same?

## Question 5. [7 MARKS]

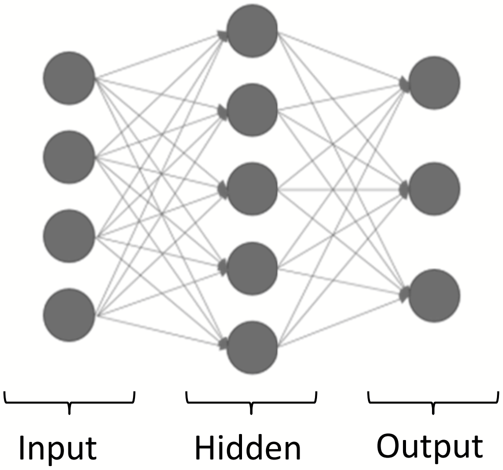
Answer the following questions given that V is of $\mathbb{R}^2$ and we have two sets of bases, U = {$[a\ b]^T$, $[c\ d]^T$}, W = {$[e\ f]^T$, $[g\ h]^T$} that span V. We are also provided sample points $v_1 = [1\ 1]^T$ and $v_2 = [2\ 1]^T$ and their coordinates in U and W, $[v_1]_W = [7\ 1]^T$, $[v_2]_W = [13\ 3]^T$, $[v_1]_U = [-1\ 25]^T$and $[v_2]_U = [-6\ 22]^T$.

**Part (a)** [5 marks]
Compute the transformation matrix $A_{U \to W}$.

**Part (b)** [2 marks]
If $[v_3]_W = [1\quad -4]^T$, calculate $[v_3]_U$.

## **Question 6.** [9 MARKS]

The following question pertains to a 2-layer artificial neural network used for multiclass classification. The network uses a tanh activation on the hidden layer and softmax activation on the output layer. The error is computed using cross-entropy loss.



Hint:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

**Part (a)** [2 mark]
Write out the equations to perform the forward pass for the proposed neural network using vectorized notation.

**Part (b)** [2 mark]
Can this network learn nonlinearly separable decision boundaries? If yes, which part(s) of the architecture are necessary for nonlinear modelling?

**Part (c)** [5 mark]
Determine the gradients with respect to the weights using vectorized notation. For this calculation you can ignore the bias terms.

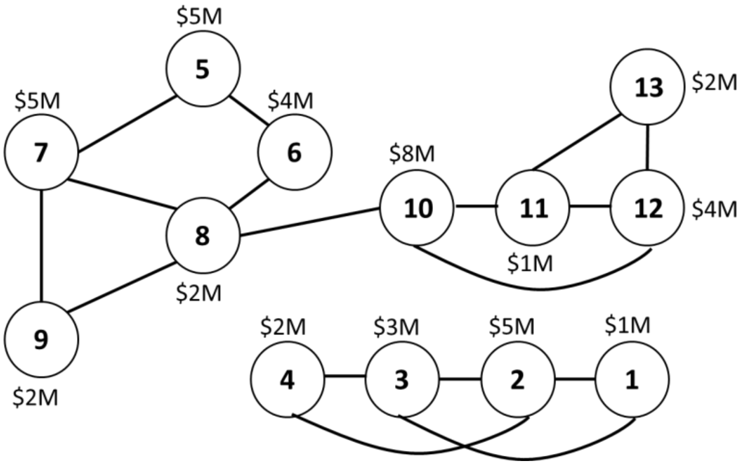
**Question 7.** [4 MARKS]

Show that minimizing the cross-entropy loss is equivalent to maximizing the log-likelihood of the training data under the assumption that data can be modeled by the provided distribution, where y $\epsilon$ {0, 1}:

$$P(y|x,\theta) = \hat{y}(x)^{y}\big(1 - \hat{y}(x)\big)^{(1-y)}$$

## Question 8. [4 MARKS]

The following diagram represents a system of buildings that are interconnected on the university campus. Your goal is to select the optimal locations (1 – 13) to construct a Tim Hortons so that students can obtain a beverage and/or snack without having to traverse more than one connection. For example, a Tim Hortons at location 1 can be accessed by students in buildings 1, 2 and 3.



Using a greedy algorithm find the optimized sets of locations to construct a Tim Hortons. The cost to construct the Tim Horton's is specified for each of the proposed locations. Show all your work.