National University of Computer and Emerging Sciences

School of Computing | Spring 2024 | Islamabad Campus

AI3003 ANN

Final Exam

Tuesday, May 21, 2024

Total Time: 3 Hours

Course Instructor

Total Marks: 145

Mr. Hassan Raza

Student Name	Roll No	Section	Signature

Instructions

- Return the question paper with the answer sheet after attempting.
- **Must fill the answers(A,B,C or D) of first 3 questions in the table given at the end of questions.**

Question I ..... (25 Marks)

Check the correct box (or boxes) for MCQs. Overwriting will result in a zero.

(a) **(1 Mark)** Which of the following is an application of NN (Neural Network)?

- ◯ Data validation
- ◯ Sales forecasting
- ◯ Risk management
- ◯ All of the mentioned

(b) **(1 Mark)** What are dendrites?

- ◯ fibers of nerves
- ◯ nuclear projections
- ◯ other name for nucleus
- ◯ none of the mentioned

(c) **(1 Mark)** The fundamental unit of network is

- ◯ brain
- ◯ nucleus
- ◯ neuron
- ◯ axon

(d) **(1 Mark)** What is shape of dendrites like

- ◯ oval
- ◯ round
- ◯ tree
- ◯ rectangular

(e) **(1 Mark)** Function of dendrites is?

- ◯ receptors
- ◯ transmitter
- ◯ both receptor & transmitter

- ◯ none of the mentioned

(f) **(1 Mark)** What is purpose of Axon?

- ◯ receptors
- ◯ transmitter
- ◯ transmission
- ◯ none of the mentioned

(g) **(1 Mark)** On what parameters can change in weight vector depend?

- ◯ learning parameters
- ◯ input vector
- ◯ learning signal
- ◯ all of the mentioned

(h) **(1 Mark)** What is generalization?

- ◯ the ability of a pattern recognition system to approximate the desired output values for pattern vectors which are not in the test set.
- ◯ the ability of a pattern recognition system to approximate the desired output values for pattern vectors which are not in the training set.
- ◯ can be either way
- ◯ none of the mentioned

(i) **(1 Mark)** Back propagation is a learning technique that adjusts weights in the neural network by propagating weight changes.

- ◯ Forward from source to sink
- ◯ Backward from sink to source
- ◯ Forward from source to hidden nodes
- ◯ Backward from sink to hidden nodes

(j) **(1 Mark)** Identify the following activation function :
f(V) = Z + (1/ 1 + exp (– x * V + Y)) Z, X, Y are parameters.

- ◯ Step function
- ◯ Ramp function
- ◯ Sigmoid function
- ◯ Gaussian function

(k) **(1 Mark)** A neuron with 3 inputs has the weight vector [0.2 -0.1 0.1]T and a bias $\theta = 0$. If the input vector is X = [0.2 0.4 0.2]T then the total input to the neuron is:

- ◯ 0.20
- ◯ 1.0
- ◯ 0.02
- ◯ -1.0

(l) **(1 Mark)** Why is the XOR problem exceptionally interesting to neural network researchers?

- ◯ Because it can be expressed in a way that allows you to use a neural network
- ◯ Because it is complex binary operation that cannot be solved using neural networks
- ◯ Because it can be solved by a single layer perceptron
- ◯ Because it is the simplest linearly inseparable problem that exists.

(m) **(1 Mark)** What is back propagation?

- ◯ It is another name given to the curvy function in the perceptron
- ◯ It is the transmission of error back through the network to adjust the inputs
- ◯ It is the transmission of error back through the network to allow weights to be adjusted so that the network can learn.

- ◯ None of the mentioned

(n) **(1 Mark)** A perceptron adds up all the weighted inputs it receives, and if it exceeds a certain value, it outputs a 1, otherwise it just outputs a 0.

- ◯ True
- ◯ False
- ◯ Sometimes – it can also output intermediate values as well
- ◯ Can't say

(o) **(1 Mark)** What is an Artificial Neural Network (ANN)?

- ◯ A computational model inspired by the human brain b) A machine learning algorithm used for image processing c) A statistical analysis technique for data clustering d) A programming language for neural network implementation

(p) **(1 Mark)** What is the basic building block of an Artificial Neural Network?

- ◯ Neuron b) Activation function c) Gradient descent d) Loss function

(q) **(1 Mark)** Which of the following activation functions is commonly used in ANNs?

- ◯ ReLU (Rectified Linear Unit)
- ◯ Sigmoid
- ◯ Tanh (Hyperbolic Tangent)
- ◯ All of the above

(r) **(1 Mark)** What is the purpose of the activation function in an ANN?

- ◯ It determines the output of a neuron
- ◯ It introduces non-linearity to the network
- ◯ It enables the network to learn complex patterns
- ◯ All of the above

(s) **(1 Mark)** What is the function of the input layer in an ANN?

- ◯ It receives input data and passes it to the hidden layers
- ◯ It performs mathematical computations on the input data
- ◯ It stores the trained weights and biases of the network
- ◯ None of the above

(t) **(1 Mark)** Which layer of an ANN is responsible for making predictions or producing the final output?

- ◯ Input layer
- ◯ Hidden layer
- ◯ Output layer
- ◯ All layers contribute equally

(u) **(1 Mark)** What is the purpose of the backpropagation algorithm in ANN training?

- ◯ To update the weights and biases based on the prediction error
- ◯ To initialize the weights and biases of the network
- ◯ To determine the number of hidden layers and neurons
- ◯ None of the above

(v) **(1 Mark)** Which of the following is a common loss function used in ANNs for binary classification?

- ◯ Mean Absolute Error (MAE)
- ◯ Mean Squared Error (MSE)
- ◯ Binary Cross-Entropy
- ◯ Categorical Cross-Entropy

(w) **(1 Mark)** What is the purpose of the forward pass in ANN training?

- ◯ To compute the predicted output based on the current weights and biases

- ◯ To adjust the weights and biases using gradient descent
- ◯ To identify misclassified samples and update the model
- ◯ None of the above

(x) **(1 Mark)** What is the primary goal of training an ANN?

- ◉ To minimize the prediction error on the training data
- ◯ To maximize the number of neurons in the hidden layers
- ◯ To achieve 100% accuracy on the test data
- ◯ None of the above

(y) **(1 Mark)** Which of the following is a common optimization algorithm used in ANN training?

- ◯ Gradient Descent
- ◯ Stochastic Gradient Descent (SGD)
- ◯ Adam
- ◉ All of the above

Question II ..... **(25 Marks)**

(a) **(1 Mark)** What is the purpose of regularization in ANN training?

- ◉ To prevent overfitting by adding a penalty term to the loss function
- ◯ To increase the model's capacity for learning complex patterns
- ◯ To speed up the training process by adjusting the learning rate
- ◯ None of the above

(b) **(1 Mark)** What is the vanishing gradient problem in ANNs?

- ◉ When the gradients become extremely small during backpropagation
- ◯ When the gradients become extremely large during backpropagation
- ◯ When the weights and biases are initialized randomly
- ◯ None of the above

(c) **(1 Mark)** Which type of ANN architecture is used for processing sequential data?

- ◉ Recurrent Neural Network (RNN)
- ◯ Convolutional Neural Network (CNN)
- ◯ Multilayer Perceptron (MLP)
- ◯ Radial Basis Function Network (RBFN)

(d) **(1 Mark)** What is the purpose of dropout regularization in ANN training?

- ◉ To randomly disable neurons during training to prevent overfitting
- ◯ To increase the learning rate for faster convergence
- ◯ To add additional layers to the network for increased capacity
- ◯ None of the above

(e) **(1 Mark)** Which of the following is an advantage of using ANNs for pattern recognition?

- ◯ Ability to learn from large amounts of data
- ◯ Robustness to noise and variations in input
- ◯ Scalability to handle complex tasks
- ◉ All of the above

(f) **(1 Mark)** What is the purpose of cross-validation in ANN training?

- ◉ To evaluate the generalization performance of the model
- ◯ To split the data into training and test sets
- ◯ To perform hyperparameter tuning
- ◯ None of the above

(g) **(1 Mark)** Which type of ANN architecture is commonly used for image classification tasks?

- ◯ Convolutional Neural Network (CNN)
- ◯ Recurrent Neural Network (RNN)
- ◯ Radial Basis Function Network (RBFN)
- ◯ Multilayer Perceptron (MLP)

(h) **(1 Mark)** What is the purpose of weight initialization in ANN training?

- ◯ To set the initial values of the weights and biases in the network
- ◯ To adjust the learning rate during training
- ◯ To compute the gradient of the loss function
- ◯ None of the above

(i) **(1 Mark)** Which activation function is commonly used in the output layer for binary classification in ANNs?

- ◯ Sigmoid
- ◯ ReLU (Rectified Linear Unit)
- ◯ Tanh (Hyperbolic Tangent)
- ◯ Softmax

(j) **(1 Mark)** What is the purpose of learning rate scheduling in ANN training?

- ◯ To adjust the learning rate during training for better convergence
- ◯ To increase the number of epochs for longer training
- ◯ To shuffle the training data between epochs
- ◯ None of the above

(k) **(1 Mark)** Which of the following techniques can be used to prevent overfitting in ANN training?

- ◯ Dropout regularization
- ◯ L1 and L2 regularization
- ◯ Early stopping
- ◯ All of the above

(l) **(1 Mark)** What is the purpose of the bias term in an ANN?

- ◯ To provide a threshold for neuron activation
- ◯ To add an additional feature to the input data
- ◯ To prevent overfitting by adjusting the learning rate
- ◯ None of the above

(m) **(1 Mark)** What is the purpose of momentum in the optimization algorithm used for ANN training?

- ◯ To accelerate the convergence of the algorithm
- ◯ To prevent overfitting by regularizing the model
- ◯ To adjust the learning rate during training
- ◯ None of the above

(n) **(1 Mark)** What is the purpose of a validation set in ANN training?

- ◯ To tune the hyperparameters of the model
- ◯ To evaluate the model's performance during training
- ◯ To update the model's weights and biases
- ◯ None of the above

(o) **(1 Mark)** Which type of ANN architecture is commonly used for natural language processing tasks?

- ◯ Recurrent Neural Network (RNN)
- ◯ Convolutional Neural Network (CNN)
- ◯ Multilayer Perceptron (MLP)
- ◯ Radial Basis Function Network (RBFN)

(p) **(1 Mark)** What is the purpose of mini-batch training in ANN training?

- ○ To update the model's weights and biases after processing a subset of the training data
- ○ To reduce the computational complexity of the training process
- ○ To increase the learning rate for faster convergence
- ○ None of the above

(q) **(1 Mark)** What is the main advantage of using deep neural networks compared to shallow neural networks?

- ○ Ability to learn hierarchical representations of data
- ○ Faster convergence during training
- ○ Lower computational complexity
- ○ None of the above

(r) **(1 Mark)** Which technique is used to initialize the weights of a deep neural network layer by layer?

- ○ Xavier/Glorot initialization
- ○ He initialization
- ○ Random initialization
- ○ None of the above

(s) **(1 Mark)** Which activation function is commonly used in the hidden layers of a neural network to introduce non-linearity?

- ○ Sigmoid
- ○ ReLU (Rectified Linear Unit)
- ○ Tanh (Hyperbolic Tangent)
- ○ Linear

(t) **(1 Mark)** What is the purpose of the activation function in a neural network?

- ○ It determines the number of layers in the network
- ○ It normalizes the input data
- ○ It introduces non-linearity in the network
- ○ It controls the learning rate of the network

(u) **(1 Mark)** Which neural network architecture is used for handling sequential data, such as natural language processing or time series analysis?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ○ Recurrent Neural Network (RNN)
- ○ Radial Basis Function Network (RBFN)

(v) **(1 Mark)** Which neural network architecture is commonly used for image classification tasks?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ○ Recurrent Neural Network (RNN)
- ○ Radial Basis Function Network (RBFN)

(w) **(1 Mark)** Which algorithm is used for updating the weights in a neural network during the training process?

- ○ Backpropagation
- ○ Gradient Descent
- ○ Stochastic Gradient Descent (SGD)
- ○ All of the above

(x) **(1 Mark)** What is the purpose of the bias term in a neural network?

- ○ It controls the learning rate of the network
- ○ It adds flexibility to the decision boundaries of the network

- ◯ It introduces non-linearity in the network
- ◯ It allows shifting the activation function

(y) **(1 Mark)** Which algorithm is used for updating the weights in a neural network with a single training example at a time?

- ◯ Backpropagation
- ◯ Gradient Descent
- ◯ Stochastic Gradient Descent (SGD)
- ◯ Mini-batch Gradient Descent

Question III ..... **(30 Marks)**

(a) **(1 Mark)** Which technique is used for preventing overfitting in a neural network by randomly dropping out neurons during training?

- ◯ Dropout
- ◯ Batch Normalization
- ◯ L1 Regularization
- ◯ L2 Regularization

(b) **(1 Mark)** What is the purpose of the loss function in a neural network?

- ◯ It measures the accuracy of predictions
- ◯ It measures the complexity of the model
- ◯ It quantifies the difference between predicted and actual values
- ◯ It controls the learning rate of the network

(c) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the previous weight update?

- ◯ Backpropagation through time (BPTT)
- ◯ Momentum Propagation (MProp)
- ◯ Levenberg-Marquardt Algorithm
- ◯ Quickprop

(d) **(1 Mark)** Which neural network architecture is used for handling both sequential and spatial data, such as video processing or 3D image analysis?

- ◯ Feedforward Neural Network (FNN)
- ◯ Convolutional Neural Network (CNN)
- ◯ Recurrent Neural Network (RNN)
- ◯ Long Short-Term Memory (LSTM) Network

(e) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of steepest descent?

- ◯ Backpropagation
- ◯ Gradient Descent
- ◯ Momentum Gradient Descent
- ◯ Newton's Method

(f) **(1 Mark)** What is the purpose of the learning rate in a neural network?

- ◯ It controls the speed of convergence during training
- ◯ It determines the number of hidden layers in the network
- ◯ It introduces non-linearity in the network
- ◯ It allows shifting the activation function

(g) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient?

- ○ Backpropagation
- ✓ Gradient Descent
- ○ Adam Optimization
- ○ Adaboost

(h) **(1 Mark)** Which state of the art neural network architecture is used for handling variable-length sequential data, such as text generation or machine translation?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ○ Recurrent Neural Network (RNN)
- ✓ Transformer Network

(i) **(1 Mark)** Which technique is used for normalizing the input data in a neural network to ensure similar scales across different features?

- ○ Dropout
- ✓ Batch Normalization
- ○ L1 Regularization
- ○ L2 Regularization

(j) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient and the magnitude of the previous weight update?

- ○ Backpropagation through time (BPTT)
- ○ Resilient Propagation (RProp)
- ○ Levenberg-Marquardt Algorithm
- ✓ RMSPrp

(k) **(1 Mark)** Which neural network architecture is used for handling both sequential and hierarchical data, such as natural language parsing or speech recognition?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ✓ Recursive Neural Network (ReNN)
- ○ Radial Basis Function Network (RBFN)

(l) **(1 Mark)** Which technique is used for preventing overfitting in a neural network by adding a penalty term to the loss function based on the weights?

- ○ Dropout
- ○ Batch Normalization
- ✓ L2 Regularization

(m) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient and the squared history?

- ○ Backpropagation
- ○ Gradient Descent
- ○ Conjugate Gradient
- ✓ RMSPror

(n) **(1 Mark)** Which neural network architecture is used for handling both sequential and non-sequential data, such as sentiment analysis or document classification?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ○ Recurrent Neural Network (RNN)
- ○ Transformer Network

National University
of Computer & Emerging Sciences

Serial No 055118

Please Tick (✓) Campus: CFD ☐ ISB ☑ KHI ☐ LHR ☐ PWR ☐ Semester: SP ☑ SU ☐ FA ☐ 20____

FINAL EXAM ANSWER BOOK

Course Code & Title: Artificial Neural Networks

Roll No: _____ Section: _____ Student's Signature: _____ Date: _____

Serial No. of continuation sheet if attached: _____ Total No. of Extra Sheets Used: _____ Invigilator's Signature: _____

(THIS ANSWER BOOK CONTAINS PAGE 1-8)

DO NOT OPEN THE ANSWER BOOKLET OR START UNTIL INSTRUCTED

Instructions:

1. Please ensure that the area in your threshold is free of any material classified as 'useful in the paper or else there may a charge of cheating.
2. Read the question carefully to ensure clarity of context and understanding of meaning. Make assumptions whatever necessary, as neither the invigilator nor the teacher/examinar will address your queries or provide assistance in the examination hall.
3. Fit in all your answers in the answer booklet. You may use an extra sheet if required. If you do so, clearly mark question/part number on that page to avoid confusion.
4. Use only your own stationery and calculator. (If permitted by your teacher/examiner). If you do not have your own calculator, perform manual calculations.
5. Use only permanent ink pens. Only the questions attempted with permanent ink pen will be considered. Any part of paper done in lead pencil cannot be claimed for rechecking.
6. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, ear buds etc.) with you.
7. Return your Question Paper along with the answer booklet (including extra sheets, if used) to the invigilator before leaving the exam venue.

Q./Part No.	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Q-7	Q-8	Q-9	Q-10
Total Marks										
Obtained Marks	21	22	26	10	10	10	11	08		
CLO NO.										

Total Marks: 118

Total Marks Obtained

Examiner/Course Teacher

Date

(a).4)

(a)

$$J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\frac{\partial J}{\partial \theta_n} \rightarrow \frac{2}{m}\sum_{i=1}^{m}(\quad$$

for Simple Problem :

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

$$\frac{\partial J}{\partial \theta_0} = \frac{2}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) \rightarrow (i)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{2}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)x^{(i)} \rightarrow (ii)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{2}{4}\left[(\theta_0 + 2\theta_1 - 4) + (\theta_0 + 2\theta_1 - 1) + (\theta_0 + 4\theta_1 - 3) + (\theta_0 - 1)\right]$$

$$= \frac{1}{2}\theta_0 + \frac{3}{2}\theta_1 - 2 + \frac{1}{2}\theta_0 + \theta_1 - \frac{3}{2} + \frac{1}{2}\theta_0 + 2\theta_1 - \frac{3}{2} + \frac{1}{2}\theta_0 - \frac{1}{2}$$

$$= 2\theta_0 + 4.5\theta_1 - 4.5 \rightarrow (A)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{2}{9}\left[(\theta_0+3\theta_1-4)\times 3+(\theta_0+2\theta_1-1)\times 2 + (\theta_0+4\theta_1-3)\times 4+(\theta_0-1)\times 0\right]$$

$$= \frac{1}{2}\left[3\theta_0+9\theta_1-12+2\theta_0+4\theta_1-2 + 4\theta_0+16\theta_1-12\right]$$

$$= 4.5\theta_0 + 14.5\theta_1 - 13 \rightarrow (B)$$

From (A) and (B)

$$4.5\theta_0 + 14.5\theta_1 - 13 = 0$$

$$2\theta_0 + 4.5\theta_1 - 4.5 = 0$$

Multiply (A) with -2.25

$$4.5\theta_0 + 14.5\theta_1 - 13 = 0$$

$$-4.5\theta_0 - 10.125\theta_1 + 10.125 = 0$$

$$4.375\theta_1 - 2.875 = 0$$

$$\theta_1 = \frac{2.875}{4.375}$$

$$\theta_1 = 0.657$$

$$\theta_1 = 0.66$$

So

From (A)

$2\theta_0 + 4.5\theta_1 - 4.5 = 0$

$2\theta_0 + 2.96 - 4.5 = 0$

$2\theta_0 = 1.54$

$\Rightarrow \theta_0 = 0.77$

(b)

$h_\theta(x) = 0.77 + 0.66x$

$= 0.77 + 0.66(5)$

$= 4.07$

(c)

$h_\theta(x) = 0.77 + 0.66(1)$

$= 1.43$

(VII)

$h(1) = 6(0.0) + 2$

$= 2$

$h(2) = 6(0.101010) + 2$

$= 8.606606$

$h(3) = 6((2.020202) + 2$

$= 14.121212$

$h(4) = 6(3.030303030) + 2$

$= 20.181818$

$h(5) = 6(4.040404) + 2$

$= 26.242424$

$e_1 = 20.0 - 2$

$= 18$

$e_2 = 27.070710 - 9.2424$

$= 19.020304$

$e_3 = 36.202928 - 14.112112$

$= 22.090124$

$e_4 = 40.364533 - 20.187118$

$= 20.177337$

$e_5 = 60.567289 - 26.242424$

$= 34.324865$

$MSE = \frac{1}{5}(18^2 + 19.09^2 + 22.08^2 + 27.18^2 + 34.32^2)$

$= 617.99$

(b)

$y = 10x^2 + 6x + 2$

$h(1) = 10(0.5)^2 + 6(0.5) + 2$

$= 2$

$h(2) = 10(1.5)^2 + 6(1.5) + 2$

$= 19.25$

$= 19.25$

$h(3) = 10(2.020202)^2 + 6(9.090909) + 2$

$= 54.935372$

$h(4) = 10(3.030303)^2 + 6(3.030303) + 2$

$= 112.009181$

$h(5) = 10(4.040404)^2 + 6(6.060606) + 2$

$= 189.491069$

$e_1 = 18$

$e_2 = 8.81 \quad (27.08 - 18.27)$

$e_3 = 36.20 - 54.973$

$= -18.773$

$e_4 = 47.36 - 112.00$

$= -64.64$

$e_5 = 60.56 - 189.49$

$= -128.92$

$MSE = \frac{1}{5}(18^2 + 8.81^2 + (-18.73)^2 + (-64.64)^2 + (-128.92)^2]$

$= 4307.66$

As MSE of Model 1 is less than that of Model 2. We'll choose Model 1.

(c)

(a) $\hat{y} = \theta x + 9$

$= 6(10) + 9$

$y = 99$

(2.5)

```
def cost_function(X, y, theta):
    hypothesis = np.dot(X, theta)
    error = (y - hypothesis)**2
    return 1/2*m * sum(error)
```

error = error / (2 * len(X))

return error

(b)

$\frac{\partial J}{\partial \theta_1}$ Assuming $h_\theta = \theta_0 + \theta_1 x$

$$\frac{\partial J}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_0 - \theta_1 x\right)^2$$

$$= \frac{2}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_0 - \theta_1 x\right)(-x)$$

$$= -\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_0 - \theta_1 x\right) x$$

$$\frac{\partial J}{\partial \theta_0} = \frac{\partial}{\partial \theta_0} \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_0 - \theta_1 x\right)^2$$

$$= \frac{2}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \theta_0 - \theta_1 x\right)(-1)$$

$$= -\frac{1}{m} \sum_{i=1}^{m} \left(y - \theta_0 - \theta_1 x\right)$$

(VIII)

(a)

def affine_forward (x, w, bets):

hypothesis_1 = W ~~theta~~ . X . T

hypothesis_2 = np.sum (hypothesis_1)

hypothesis_3 = np.exp (-1 * bets * hypothesis_2)

dW = hypothesis_3 * (-2) * bets * hypothesis_2

dx = hypothesis_3 * (+2) * bets * hypothesis_2 * x

return (hypothesis_3, dx, dW)

(b)

(c)

(Tx)

class Descriminator(nn.Module):
    def __init__(self):
        super().__init__()
        self.layers = nn.Sequential(
            nn.Linear(1024, 512),
            nn.ReLU(),
            nn.Linear(512, 256),
            nn.ReLU(),
            nn.Linear(256, 64),
            nn.ReLU(),
            nn.Linear(64, 1))
    def forward(self, x):
        return self.layers(x)

net_D = Discriminator()
recon_loss_func = BCELoss()
optimizer = SGD(net_D.parameters(), 0.01)

data = transformation_data_name('path_to_image')
dataloader = DataLoader(data, shuffle=True)
epoch = 10

```
for i in range (epochs):
    for img, label in dataloader:
        out = net(img.flatten())
        loss = loss_func(out, label)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
```

(Q VIII)

(b)

$$\text{Loss} = L_i = (y^{(i)} - h^{(i)})^2$$

$$\text{overall loss} = L = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - h^{(i)})^2$$

⇒ Assuming there are n activation functions involved

$$L = \frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - f\left(p\sum(w_i^{(i)} - x_i)\right)\right]^2$$

Rough Work

$$\frac{dL}{dw_1} = \frac{d}{dw_1}\left[\frac{1}{m}\sum_{i=1}^{m} y^2 - e^{\left(-\beta\sum w_i^{(2)} - x_i\right)^2}\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m} -e^{-\beta\sum(w-x)} \frac{d}{dw_2}\left[-\beta\sum(w-x)^2\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m} -e^{-\beta\sum\left(w_2^{(2)}-x\right)^2}\left[-\beta(2)(w-x)\right]$$

$$\Rightarrow \frac{dL}{dw_2} = \left[\frac{1}{m}\sum_{i=1}^{m} 2\beta\, e^{-\beta\sum\left(w_{i1}^{(2)} - x_{(i)}\right)^2}\left(w_{i1}^{(2)} - x_{(i)}\right)\right]$$

$$\frac{dL}{dw_1} = \frac{dL}{dh_1} \times \frac{dh_1}{dw_1}$$

$$\frac{dL}{dh_1} = \frac{d}{dh_1}\left[\frac{1}{m}\sum y_i - e^{\left(-\beta\sum w_{1i}^{(1)} - x_i\right)^2}\right]$$

$$= \frac{d}{h_1}\left[\frac{1}{m}\sum - e^{\left(-\beta\sum(w-x)^2\right)}\right]\left[-\beta(2)(w-x)(1)\right]$$

$$= \frac{1}{m}\sum -e^{\left(-\beta\sum(w-x)^2\right)}\left(2\beta(w-x)\right)$$

$$\frac{dL}{dh_1} = \frac{1}{m}\sum 2\beta e^{-\beta\Sigma(w-x)^2}(w-x) \rightarrow (A)$$

$$\frac{dh_1}{dw_1} = \frac{d}{dw_1} e^{-\beta\Sigma(w-x)^2}$$

$$= e^{-\beta\Sigma(w-x)} \left[-\beta(2)(w-x)(1)\right]$$

$$= -2\beta e^{-\beta\Sigma(w-x)^2}(w-x) \rightarrow (B)$$

$$\frac{dL}{dw_1} = \frac{dL}{dh_1} \times \frac{dh_1}{dw_1}$$

from A & B

$$= \frac{1}{m}\sum -2\beta e^{-\beta\Sigma(w-x)^2}(w-x) \times -2\beta e^{-\beta\Sigma(w-x)^2}(w-x)$$

(o) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient and the momentum term?

- ○ Backpropagation
- ○ Gradient Descent
- ○ Stochastic Gradient Descent (SGD)
- ✓ Momentum-based Gradient Descent

(p) **(1 Mark)** What is the purpose of the momentum term in a neural network?

- ○ It controls the speed of convergence during training
- ○ It introduces non-linearity in the network
- ○ It allows shifting the activation function
- ✓ It helps accelerate the convergence and overcome local minima

(q) **(1 Mark)** Which technique is used for preventing overfitting in a neural network by randomly selecting a subset of the training examples for each iteration?

- ○ Dropout
- ○ Batch Normalization
- ○ L1 Regularization
- ✓ Mini-batch Gradient Descent

(r) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient, momentum and directionless history?

- ○ Backpropagation
- ○ Gradient Descent
- ✓ Adam Optimization
- ○ Adaboost

(s) **(1 Mark)** What is the purpose of the early stopping technique in a neural network?

- ✓ It prevents the network from overfitting the training data
- ○ It speeds up the convergence of the network
- ○ It allows shifting the activation function
- ○ It controls the learning rate of the network

(t) **(1 Mark)** Which neural network architecture is used for handling both sequential and spatial data, such as video processing or 3D image analysis?

- ○ Feedforward Neural Network (FNN)
- ○ Convolutional Neural Network (CNN)
- ○ Recurrent Neural Network (RNN)
- ✓ Long Short-Term Memory (LSTM) Network

(u) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient and the magnitude of the previous weight update?

- ○ Backpropagation through time (BPTT)
- ○ Resilient Propagation (RProp)
- ○ Levenberg-Marquardt Algorithm
- ✓ Momentum Optimization

(v) **(1 Mark)** What is the purpose of the dropout technique in a neural network?

- ✓ It prevents the network from overfitting the training data
- ○ It speeds up the convergence of the network
- ○ It allows shifting the activation function
- ○ It introduces non-linearity in the network

(w) **(1 Mark)** Which algorithm is used for updating the weights in a neural network by considering the direction of the negative gradient and quadratic derivative?

- ○ Backpropagation
- ○ Gradient Descent
- ○ Conjugate Gradient
- ○ RMS Prop

(x) **(1 Mark)** Choose the correct option describing the features of Artificial neural network

1. It is essentially machine learning algorithm.
2. It is useful when solving the problems for which the data set is very large.
3. They are able to extract features without input from the programmer.
4. These are systems modeled on the human brain and nervous system

Choose the most appropriate answer from the options given below:

- ○ All the statements are correct.
- ○ Only 2 and 3 are correct.
- ○ Only 1 and 4 are correct.
- ○ All the statements are not correct.

(y) **(1 Mark)** Which Boolean operation on two variables can be represented by a single perception layer?

1. X1 AND X2
2. X1 OR X2
3. X1 NOR X2
4. X1 XOR X2

Choose the most appropriate answer from the options given below:

- ○ 1 and 2 Only
- ○ 2 and 3 Only
- ○ 1, 2 and 3 Only
- ○ 1, 2, 3 and 4 Only

(z) **(5 Marks)** In a feed forward neural network with the following specifications:

Input layer has 4 neurons, hidden layer has 3 neurons and output layer has 2 neurons using the sigmoid activation function for given input values [0.5, 0.8, 0.2, 0.6] as well as the initial weights for the connections.

W1: [0.1, 0.3, 0.5, 0.2]
W2: [0.2, 0.4, 0.6, 0.2] Input layer to hidden layer weights
W3: [0.3, 0.5, 0.7, 0.2]
W4: [0.4, 0.1, 0.3]
W5: [0.5, 0.2, 0.4] Hidden layer to output layer weights

What is the output of the output layer when the given input values are passed through neural network? Round the answer to two decimal places :

- ○ [0.62, 0.68]
- ○ [0.72, 0.78]
- ○ [0.82, 0.88]
- ○ [0.92, 0.98]

Question 1	Answer	Question 2	Answer	Question 3	Answer
a	b	a	a	a	a
b	a	b	a	b	c
c	c	c	a	c	a
d	c	d	a	d	d
e	b	e	d	e	b
f	a	f	a	f	a
g	b	g	a	g	b
h	b	h	a	h	d
i	b	i	a	i	b
j	c	j	a	j	d
k	c	k	d	k	c
l	d	l	a	l	c
m	c	m	a	m	c
n	c	n	b	n	d
o	a	o	a	o	d
p	a	p	b	p	d
q	d	q	a	q	d
r	d	r	a	r	c
s	a	s	b	s	a
t	c	t	c	t	d
u	a	u	c	u	d
v	c	v	b	v	a
w	a	w	d	w	d
x	a	x	b	x	a
y	d	y	c	y	c
	24		23	z	a

26

Question IV ..... **(10 Marks)**

Consider the problem of predicting how well a student does in his/her second semester of university, given how well they did in their first semester. Specifically, let x be equal to the number of "A" grades (including A-, A, and A+ grades) that a student receives in their first semester of college. We would like to predict the value of y, which we define as the number of "A" grades they get in their second semester. Refer to the following training set of a small sample of different students' performances . Here each row is one training example. Recall that in linear regression, our hypothesis is $h_\theta(x) = \theta_0 + \theta_1 x$, and we use m to denote the number of training examples.

x	y
3	4
2	1
4	3
0	1

Figure 1: Example Dataset

(a) **(7 Marks)** What values of θ_0, θ_1 should we use. [Hint: Remember our goal is to find the parameters that minimize following cost function. $J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$. You may use simple Calculus rules to arrive at the solution]

(b) **(2 Marks)** How many $A's$ a student will get in his second semester, if he had 5 A's in his first semester.

(c) **(1 Mark)** How many $A's$ a student will get in his second semester, if he had 1 A's in his first semester.

Question V ..... **(10 Marks)**

Recall that in Linear Regression we try to optimize (minimize) following cost function to find the optimal (a separator that separates maximally both the classes) separating decision boundary:

$$J_\theta = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - h_\theta(x^{(i)}))^2$$

Here m is the number of training examples.

(a) **(5 Marks)** Now your goal is to write a function to compute the cost function for a given dataset.

```
def cost_function(self, X,Y, theta):
        '''
            Computes the Cost function for given
            input data (X) and labels (Y).

            Input:
                X: can be either a single n X d-dimensional
                   vector or n X d dimensional matrix of inputs
                theta: must  d X 1-dimensional vector for
                   representing vectors
                Y: Must be n X 1-dimensional label vector

            Return:
                Returns the cost of hypothesis with input parameters
        '''
```

(b) **(5 Marks)** Write the equation of derivative of given cost function with respect to $\theta_1 and \theta_0$

Question VI ..... **(10 Marks)**

Consider the problem of predicting how well a student does in his/her second semester of university, given how well they did in their first semester. Specifically, let x be equal to the number of "A" grades (including A-, A, and A+ grades) that a student receives in their first semester of college. We would like to predict the value of y, which we define as the number of "A" grades they get in their second semester. Refer to the following training set of a small sample of different students' performances . Here each row is one training example. Recall that in linear regression, our hypothesis is $h_\theta(x) = \theta_0 + \theta_1 x$, and we use m to denote the number of training examples.

x	y
3	4
2	1
4	3
0	1

Figure 2: Example Dataset

(a) **(7 Marks)** What values of θ_0, θ_1 should we use. [Hint: Remember our goal is to find the parameters that minimize following cost function. $J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$. You may use simple Calculus rules to arrive at the solution]

(b) **(2 Marks)** How many $A's$ a student will get in his second semester, if he had 5 A's in his first semester?

(c) **(1 Mark)** How many $A's$ a student will get in his second semester, if he had 1 A's in his first semester?

Question VII ..... **(10 Marks)**

From the following measured values of (x, y):

	X	Y
0	0.000000	20.000000
1	1.010101	27.080910
2	2.020202	36.202428
3	3.030303	47.364555
4	4.040404	60.567289

Figure 3: Example dataset.

(a) **(3 Marks)** A linear regression run resulted in the model: $y = 6x + 2$. Determine the error in each estimation and the mean sum of square of errors (MSE) (in other words what will be the cost function).

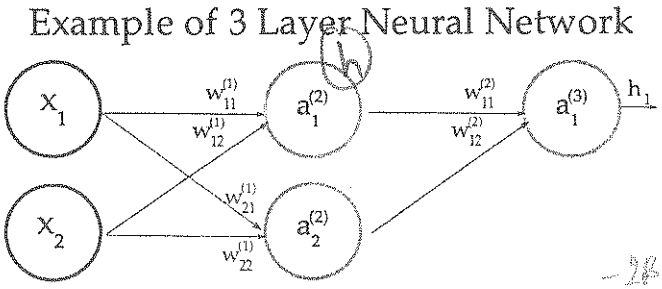
(b) **(4 Marks)** Another linear regression run resulted in the model: $y = 10 * x^2 + 6x + 2$. Determine the error in each estimation and the mean sum of square of errors (MSE) (in other words what will be the cost function).

(c) **(3 Marks)** What model among the both of the above will you use ? Now use the selected model to estimate y for $x = 15$.

Question VIII ..... **(15 Marks)**

RBF Neural Network A Radial Basis Function (RBF) Neural network is another class of neural networks in which we replace the affine layer with a Gaussian layer. For example, to convert the following given neural

network to RBF neural network we simply update the definition of affine functions $a_1^{(2)}, a_2^{(2)}$, and $a_1^{(3)}$ with their corresponding RBF definitions.



For instance, in our RBF $a_1^{(2)}$ will have following definition:

$$a_1^{(2)} = \exp(-\beta \sum_i (W_{1i}^{(2)} - X_i)^2)$$

instead of $a_1^{(2)} = \sum_i (W_{1i}^{(2)} * X_i)$. $a_2^{(2)}$ and $a_1^{(3)}$ also have similar forms. Here β is a hyperparameter and remains same for the neurons of a layer. Also note that there are no biases involved. For this current RBF neural network we will be using L_2 loss function that is, $L_i = (y^{(i)} - h_1^{(i)})^2$

(a) **(5 Marks)** Write code for forward propagation,

```
def affine_forward_RBF(x, w, beta):
    """
    Computes the forward pass for an RBF (fully-connected) layer.

    The input x has shape (N, d_1, ..., d_k) where x[i] is the ith input.

    Inputs:
    x - Input data, of shape (N, d_1, ..., d_k)
    w - Weights, of shape (D, M)
    beta - a scalar for the current layer

    Returns a tuple of:
    - out: output, of shape (N, M)
    - cache: (x, w, beta)
    """
```

(b) **(10 Marks)** Now given this definition of RBF networks, your goal is to derive derivatives for $W^{(2)}$ and $W^{(1)}$.

Question IX ..... **(10 Marks)**

Code the following network in pytorch. The fully connected neural network predicts whether an RGB image of size 32x32 is real or fake.

- There are 3 hidden layers
- All layers except the last one have ReLU() as activations.
- Use nn.Sequential to build the network
- Choose an appropriate loss function and an optimizer.
- Do a forward pass and a backward pass.
- The class name should be Discriminator and the object of the class should be netD=Discriminator()
- The dataloader is a tuple containing an image and a label.