Date : 04/11/2022

(mid turm :

* unit 1,2,3(how much cover in part)
* numpy
* panda
* met plot lib)

perceptron

only for linerly sapratable

wait update

multi - layer perceptron

* hidden layer
* activation fuction (sigmoid function)
* backpropagation algorithm for learning

**Back propagation algorithm**

* forword pass
* backword pass
* gradient dicent

**Date : 09/11/2022 (10:30 to 12:30)**

**Midturm exam**

* kmeans – unsupervised
* unit 1 , 2
* unit 3 half
* peceptron
* back propogation
* Multy layer perceptron
* Solving of
* Python
  + theory (loop why it is uses)
  + What Is numpy
  + What is a functionality of numpy
  + No need of focus on coding
* Regression
* Metrics vectors
  + Orthogonal metrix
  + Symmetric metrix
* Veriants
* Mean median mod
* Data dispersion
  + Standerd deviation
* Formulas
* Probability
* Rendom veriable
* Sample space
* Probability dis\* function
* Bay’s propbabilty
* Co relation
* Co-veriant
  + Pearsons corelation
  + Spears man coralation
  + How to compute
* Leniar regression
* OLS for finding

Unit 2

* Application of ML
* Types of machine learning
  + Supervise
  + Unsupervised
  + Reinforcement
  + Semi supervise
  + Model base
  + Instance base
  + Online learning
  + Offline learning
  + Basic of super vise
    - Prediction
  + Classification
    - Logistic
    - Knn
    - Logistic regration
    - Designe tree
      * Feature selection
        + Forword selection
        + Backword
        + Mix
* Training set
* Testing set
* Validation set
* Hyper perameter
* Model peraeter
* Chllenges if ML
* Algorithm base challenges
* Data based challenger
* Measure performance
* What is metric
* Classification evaluation
* Positive class (most important class)
* Negative class
* Classification evaluation / performance measure of classification
* Confusion matrix
  + It is matrix that tabular-visualization of the ground truth vs predicted value

|  |  |  |  |
| --- | --- | --- | --- |
| Predicted value | | positive | negative |
| Ground value | positive | TP (true positive) | FN(false negative) |
| negative | FP(false positive) | TN(true negative) |

* + Accuracy = (TP + TN) / (TP + FN + FP + TN)
    - Not use for imbalanced data set
  + Precision = TP / (TP + FP)
  + Recall / Sensitivity / T = TPR / P
    - P = (TP + FN ) = **Actual positive**
    - **TPR**  =true positive rate
  + Specificity / TNR = TN / N
    - **TNR** = true negative rate
    - N = (FP + TN) = **Actual negative**
  + F1 – score = (2\*precision\*recall)/(precision + recall) **= Harmonic mean**
* Performance me
  + Mean absolute error
  + Mean square error
  + Root mean squared error
* **Decision tree**
  + Attribute selection procedure
  + Pure Partition
  + Information gain (ID3)
  + Gain ration
  + Gini Index
* **ID3**
  + Top down approach
  + Compute gain for all attribute
  + For every branch the remaining data count the gain for all attribute until there is a stopping condition
    - **Stopping condition**
      * No Data left
      * No attribute left
      * Pure partition
  + After that give a decision
  + Every branch have a decision depending a majority voting
  + **Entropy = -plog(p)**
  + **Gain(A) = Info(D) = Info A(D)**
  + **Info(D) =**
  + **Info A(D)**

**For lab part practical exam**

* Module1 : python
* Module2: model building
  + Data collection
  + Understanding data
  + Cleaning data
  + Model building
  + Evaluation of the model

**DATE : 09/11/2022 Main Class (12:30 to 1:30)**

**Regression MLP**

* One output neuron per output dimension
* Typical Regression MLP Aechitecture (# = numer of …)
  + # input neurons – one per input feature (e.g. , 28 x 28 = 784 for MNIST)
  + # hidden layers – Depends on the problem. Typically 1 to 5.
  + # neurons per hidden layer
  + # output neurons
  + Hidden activation
  + Output activation
  + Loss function – MSE or MAE/Huber
* **Classification MLP**
  + Multilabel Binary classification (Yes or NO)
    - Logistic activation function
  + Multiclass classification
  + Architecture
    - Input and hidden layers
    - # output neurons
    - Output layer activation
    - Loss function
* **Few terminologies**
  + Batch size
  + **Batch Gradient Descent**
  + **Stochastic Gradient Descent**
  + **Mini-Batch Gradient Descent**
  + **Epoch**
* **Activation function**
  + Activation function map a neuron’s input to that of the next layer
  + **Need of an Activation function**
    - Restricting value
    - Add non-linearity
  + List of activation function
    1. Sigmoid [0,1] = mid value = 0.5
    2. Tanh Function [-1,1] = mid value = 0
    3. Softmax
       - Not use in hidden layer coz dependancy
    4. RELU
* **Loss Function**
  + Mean Squared Error (Regression)
  + Mean Absolute Error (Regression)
  + Binary Cross Entropy (Classification)
  + Cross Entropy (Classification)

**Date: 29 Nov. 22**

**Coputer**