

ARTIFICIAL INTELLIGENCE

Lab Manual

[Fall/ Spring 20\_\_]

|  |  |
| --- | --- |
| Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Student Id: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
|  |  |

|  |  |
| --- | --- |
| Prepared By: *Dr. Noman Islam* |  |
| Instructor: *Dr. Noman Islam* |  |

**LIST OF EXPERIMENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment** |  |
| **1** | \_\_/\_\_/\_\_ | To study and implement a chatbot using DialogFlow |  |
| **2** | \_\_/\_\_/\_\_ | To setup the environment and familiarize with Python |  |
| **3** | \_\_/\_\_/\_\_ | To study and implement algorithms in Python |  |
| **4** | \_\_/\_\_/\_\_ | To study and implement Graph search algorithms in Python |  |
| **5** | \_\_/\_\_/\_\_ | To study and understand numpy library |  |
| **6** | \_\_/\_\_/\_\_ | To study and implement pandas library |  |
| **7** | \_\_/\_\_/\_\_ | To study and implement Artificial Neural Network using Keras |  |
| **8** | \_\_/\_\_/\_\_ | To study and implement Convolutional Neural Network using Keras |  |
| **9** | \_\_/\_\_/\_\_ | To study and implement LSTM using Keras |  |
| **10** | \_\_/\_\_/\_\_ | To study and implement a web application in Django |
| **11** | \_\_/\_\_/\_\_ | To study and implement generative models based on LSTM to generate text in Keras |

**Lab 1: To study and implement a chatbot using DialogFlow**

The objective of this lab is to develop an admission chatbot using DialogFlow. DalogFlow gives users new ways to interact with your product by building engaging voice and text-based conversational interfaces powered by AI. You can connect with users on the Google Assistant, Amazon Alexa, Facebook Messenger, and other popular platforms and devices.

**Lab Tasks:**

1. Login to dialogflow.com by using your Gmail account.
2. Create an agent Admission that can answer questions related to admission.
3. Create an intent with name “Ask programs” such that user can ask for the admission programs at Iqra University.
4. Provide example questions to train the agent on this intent.
5. Provide sample responses as answer to this question.
6. Test your intent by asking questions
7. Create an intent named “Program Details” with parameter “degree program” and “query”
8. Provide example questions to train the agent on this intent such as “What is the percentage requirement for BS(CS)”
9. Provide sample responses as answer to this question.
10. Provide a default value for the degree program.
11. Test your intent by asking questions.
12. Now make degree program required and provide a default prompt.
13. Test your intent by asking questions.
14. Provide a speech response and then test your intent.
15. Add a question “How about another degree program (such as BBA)”
16. Set the “query” as context variable
17. Now publish your chatbot

**Lab 2: To setup the environment and familiarize with Python**

The objective of this lab is to set up the Python environment and get some familiarity with the language.

To set up the environment, follow the steps below:

1. Download and install Anaconda. Anaconda is the leading open data science platform powered by Python
2. Download and install PyCharm. PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language.

**Lab Tasks:**

1. Write a small program in Python to print your CV.
2. Write a program that takes the month (1…12) as input. Print whether the season is summer, winter, spring or autumn depending upon the input month.
3. To determine whether a year is a leap year, follow these steps:
   1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
   2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
   3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
   4. The year is a leap year (it has 366 days).
   5. The year is not a leap year (it has 365 days).

Write a program to input an year as integer. Using if…else, determines whether the input is a leap year or not.

1. Write a program that takes a line as input and finds the number of letters and digits in the input
2. Write a program that takes a sentence as input. Compute the frequency of each words and prints them.

**Lab 3: To study and implement basic algorithms in Python**

In this lab, we will familiarize ourselves with functions, classes and other advanced constructs of python.

**Lab Tasks:**

1. Write a program to generate a dictionary that contains (i,sqrt(i)), where *i* is an integer between 1 and n. *n* is a number input by the user.
2. Write a simple calculator program using functions add, sub, mul and div. The program should accepts two numbers and an operator and calls the corresponding function to perform the operation.
3. Write a function that generates a list with values that are square of number between 1 and 20.
4. Define a class named Shape with static method printType. Define methods draw() and area(). Now define two class Rectangle and Triangle. Rectangle has two attributes length and width. The Triangle class has attributes a,b and c. Override the two methods of shape class. Demonstrate the functionality of class by creating its objects.
5. Using recursion, write a program to calculate the reverse of a string.

**Lab 4: To study and implement Graph search algorithms in Python**

In this lab, we are going to implement searching algorithms in Python. There are two popular searching algorithms i.e. Depth First Search (Fig. 3a) and Breadth First Search (Fig 3b).

|  |
| --- |
| DFS(G,v) ( v is the vertex where the search starts )  Stack S := {}; ( start with an empty stack )  for each vertex u, set visited[u] := false;  push S, v;  while (S is not empty) do  u := pop S;  if (not visited[u]) then  visited[u] := true;  for each unvisited neighbour w of u  push S, w;  end if  end while  END DFS() |
| **3a:** Pseudo-code for Depth First Search |
| BFS(Graph, root):  create empty set S  create empty queue Q  root.parent = NIL  add root to S  Q.enqueue(root)  while Q is not empty:  current = Q.dequeue()  if current is the goal:  return current  for each node n that is adjacent to current:  if n is not in S:  add n to S  n.parent = current  Q.enqueue(n) |
| **3b:** Pseudo-code for Breadth First Search |

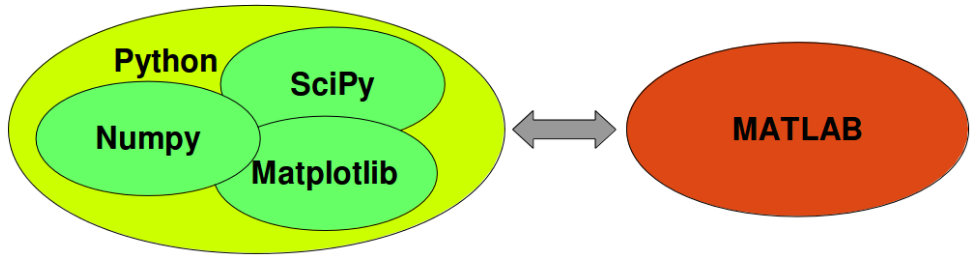
**Fig 3:** Pseudo-code for Graph Searching algorithms

**Lab Task:**

1. Provide the implementation of DFS and BFS algorithms in Python.

**Lab 5: To study and understand numpy library**

In this lab, we are going to explore numpy. NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines.



**Lab Task:**

Open the Python Notebook provided with this lab and perform the tasks.

a. Import the "numpy" library as "np".

In [ ]:

​

b. Create an array of shape (2, 3, 4) of zeros.

In [ ]:

​

c. Create an array of shape (2, 3, 4) of ones

In [ ]:

​

d. Create an array with values 0 to 999 using the "np.arange" function

In [ ]:

​

e. Create an array from the list [2, 3.2, 5.5, -6.4, -2.2, 2.4] and assign it to the variable "a"

In [ ]:

​

f. Do you know what a[1] will equal? Print it to see

In [ ]:

​

g. Try printing a[1:4] to see what that equals

In [ ]:

​

h. Create a 2-D array from the following list and assign it to the variable "a": [[2, 3.2, 5.5, -6.4, -2.2, 2.4], [1, 22, 4, 0.1, 5.3, -9], [3, 1, 2.1, 21, 1.1, -2]]

In [ ]:

​

i. Can you guess what the following slices are equal to? Print them to check your understanding. a[:, 3] a[1:4, 0:4] a[1:, 2]

In [ ]:

​

j. Create a 2-D array of shape (2, 4) containing two lists (range(4), range(10, 14)) and assign it to the variable "arr".Print the shape of the array. Print the size of the array. Print the maximum and minimum of the array

In [ ]:

​

In [ ]:

​

In [ ]:

​

k. Continue to use the array "arr" as defined above.Print the array re-shaped to (2, 2, 2).Print the array transposed.Print the array flattened to a single dimension. Print the array converted to floats.

In [ ]:

​

l. Create an an array counting from 1 to 20 inclusive

In [ ]:

​

m. The array of multiples of 3 greater than 0 and less than 30

In [ ]:

​

n. The array of 8 equally spaced floats x where 0 ≤ x ≤ 1

In [ ]:

​

o. Use np.arange and reshape to create the array A = [[1 2 3 4] [5 6 7 8]]

In [ ]:

​

p. Use np.array to create the array B = [1 2]

In [ ]:

​

q. Use broadcasting to add B to A to create the final array A + B

In [ ]:

​

In [ ]:

​

In [ ]:

​

**Lab 6: To study and implement pandas library**

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

**Lab Task:**

Open the Python Notebook provided with this lab and perform the tasks.

1. Create a data series with marks of students : 75, 80, 79, 60

In [ ]:

​

2. Create a data frame with name of students, id and marks

In [ ]:

​

3. Now read the file 'data.csv' in panda

In [ ]:

​

4. What are the columns in the dataframe?

In [ ]:

​

5. Sort the data based on Marks obtained. Fill all the 'na' cells with 0

In [ ]:

​

6. Display the top 10 rows

In [ ]:

​

7. Display the last 10 rows

In [ ]:

​

8. Display only the odd rows

In [ ]:

​

9. Display only those students who got failed in examination

In [ ]:

​

10. Find out the basic statistical info about data

In [ ]:

​

11. How many students got A, B, C, F?

In [ ]:

​

12. What are the mean scores for students who got A, B, C, F?

In [ ]:

​

**Lab 7: To study and implement Artificial Neural Network using Keras**

Keras is a powerful easy-to-use Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in a few short lines of code. Install Keras by using the following command:

> pip install keras

**Lab Tasks:**

1. Initialize the random number generator

from keras.models import Sequential

from keras.layers import Dense

import numpy

# fix random seed for reproducibility

numpy.random.seed(7)

1. Load the data

# load pima indians dataset

dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")

# split into input (X) and output (Y) variables

X = dataset[:,0:8]

Y = dataset[:,8]

Now create a model:

# create model

model = Sequential()

model.add(Dense(12, input\_dim=8, activation='relu'))

model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

1. Compile the model

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

1. Fit the model

model.fit(X, Y, epochs=150, batch\_size=10)

1. Evaluate the model

scores = model.evaluate(X, Y)

print("\n%s: %.2f%%" % (model.metrics\_names[1], scores[1]\*100))

1. Perform Predictions

predictions = model.predict(X)

# round predictions

rounded = [round(x[0]) for x in predictions]

print(rounded)

**Lab 8: To study and implement Convolutional Neural Network using Keras**

Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs. A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. In this lab, you will discover how to develop and evaluate deep learning models for object recognition in Keras.

**Lab Tasks:**

* Load the CIFAR-10 dataset in Keras.
* Create the following model:

model = Sequential()

model.add(Conv2D(32, (3, 3), input\_shape=(3, 32, 32), padding='same', activation='relu', kernel\_constraint=maxnorm(3)))

model.add(Dropout(0.2))

model.add(Conv2D(32, (3, 3), activation='relu', padding='same', kernel\_constraint=maxnorm(3)))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu', kernel\_constraint=maxnorm(3)))

model.add(Dropout(0.5))

model.add(Dense(num\_classes, activation='softmax'))

* Fit and evaluate the above model. What is the accuracy of your model?

**Lab 9: To study and implement LSTM using Keras**

The Long Short-Term Memory network or LSTM network is a type of recurrent neural network used in deep learning. In this lab, you will discover how to develop LSTM networks in Python using the Keras deep learning library to address a demonstration time-series prediction problem.

**Lab Tasks:**

1. Download the dataset from https://datamarket.com/data/set/22u3/international-airline-passengers-monthly-totals-in-thousands-jan-49-dec-60#!ds=22u3&display=line
2. Develop an LSTM network model and train it on the downloaded dataset.
3. Fit and evaluate the model. What is the accuracy of your model?

**Lab 10: To study and implement a web application in Django**

In this lab, we will study how can we implement a small web application in Django. Django is a framework for the development of web-based applications in Python. Django was designed to help developers take applications from concept to completion as quickly as possible. Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box. Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

**Lab Tasks:**

1. Install django by using the following command:

$ pip install Django

1. Create a small project in Django by typing the following commands:

$ django-admin startproject mysite

1. Create a small view and map it to the URL.

**Lab 11: To study and implement generative models based on LSTM to generate text in Keras**

In this lab, we will use LSTM to build a generative model that can produce text similar to the data it is trained on.

**Step 1: Necessary imports**

import numpy as np

import keras

import numpy as np

from keras import layers

import random

import sys

**Step 2: Reweight distribution**

def reweight\_distribution(original\_distribution, temperature=0.5):

distribution = np.log(original\_distribution) / temperature

distribution = np.exp(distribution)

return distribution / np.sum(distribution)

**Step 3: Training data**

def loadFile():

text = open('cricket.txt').read().lower()

return text

text = loadFile()

**Step 4: Vectorize the data**

maxlen = 60

step = 3

sentences = []

next\_chars = []

for i in range(0, len(text) - maxlen, step):

sentences.append(text[i: i + maxlen])

next\_chars.append(text[i + maxlen])

print('Number of sequences:', len(sentences))

chars = sorted(list(set(text)))

print('Unique characters:', len(chars))

char\_indices = dict((char, chars.index(char)) for char in chars)

print('Vectorization...')

x = np.zeros((len(sentences), maxlen, len(chars)), dtype=np.bool)

y = np.zeros((len(sentences), len(chars)), dtype=np.bool)

for i, sentence in enumerate(sentences):

for t, char in enumerate(sentence):

x[i, t, char\_indices[char]] = 1

y[i, char\_indices[next\_chars[i]]] = 1

**Step 5: Define a sample function**

def sample(preds, temperature=1.0):

preds = np.asarray(preds).astype('float64')

preds = np.log(preds) / temperature

exp\_preds = np.exp(preds)

preds = exp\_preds / np.sum(exp\_preds)

probas = np.random.multinomial(1, preds, 1)

return np.argmax(probas)

**Step 6: Develop the model**

model = keras.models.Sequential()

model.add(layers.LSTM(128, input\_shape=(maxlen, len(chars))))

model.add(layers.Dense(len(chars), activation='softmax'))

optimizer = keras.optimizers.RMSprop(lr=0.01)

model.compile(loss='categorical\_crossentropy', optimizer=optimizer)

model.fit(x, y, batch\_size=128, epochs=10)

**Step 7: Generate the text**

start\_index = random.randint(0, len(text) - maxlen - 1)

generated\_text = text[start\_index: start\_index + maxlen]

print('--- Generating with seed: "' + generated\_text + '"')

sys.stdout.write(generated\_text)

for i in range(2000):

sampled = np.zeros((1, maxlen, len(chars)))

for t, char in enumerate(generated\_text):

sampled[0, t, char\_indices[char]] = 1.

preds = model.predict(sampled, verbose=0)[0]

next\_index = sample(preds, 0.5)

next\_char = chars[next\_index]

generated\_text += next\_char

generated\_text = generated\_text[1:]

sys.stdout.write(next\_char)

**Lab Tasks:**

1. Develop a desktop application based on above code to generate text similar to training data.
2. Develop a web application based on above code.