446 Midterm 2 Solutions[1]

1. Question 1

(a) Python raises an exception (i.e. gives an error) when given the command

```
{[1, 2], 3}
```

TRUE. Recall that lists, dictionaries and sets are mutable data types in Python. Moreover, elements of sets must be immutable.

(b) Python's implementation of training a neural network is deterministic. That is, if we use the command `from tensorflow.keras.models import Sequential`, and then define `neural_model()` to be a neural network, then define `model = neural_model()` and train the network using `model.fit`, the output will be the same, regardless of how many different times I ask for an output, and regardless of any random seed that is provided to Python.

FALSE. We observed in class that the training the neural network is not deterministic since e.g. the training data is randomly ordered in different epochs.

(c) For classifying images from the MNIST dataset, we found in class that convolutional neural networks outperform a single layer neural network.

TRUE. We observed this in class. CNNs obtained less than one percent error on the training set, while single layer networks had around 7% error.

(d) For classifying images from the facial images dataset (which we brought into Python with the command `from sklearn.datasets import fetch_lfw_people`), we found in class that convolutional neural networks outperform a two layer neural network.

TRUE. We observed this in class. CNN with two convlutional layers obtained around 16% error, while a two layer neural network bad around 21% error on the training set.

2. Question 2

Suppose we run the commands

```
import pandas as pd
data = {
    "state": ["Ohio", "Ohio", "Ohio", "Nevada", "Nevada", "Nevada"],
    "year": [2000, 2001, 2002, 2001, 2002, 2003],
    "pop": [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]
}
frame = pd.DataFrame(data)
```

(a) What is the output of the following commands?

```
frame2 = frame.reindex(index = [3, 2, 5])
frame2
```

(b) What is the output of the following commands?

```
frame3 = frame2.set_index("year")
frame3
```

Solution.

[1]November 9, 2024,

```
    state  year  pop
3  Nevada  2001  2.4
2    Ohio  2002  3.6
5  Nevada  2003  3.2
```

```
          state    pop
year
2001  Nevada    2.4
2002    Ohio    3.6
2003  Nevada    3.2
```

3. Question 3

Suppose we have a DataFrame named `df2` that contains single season 2 point field goal percentages of basketball players, of the form

```
        player      percentage      year
0       Clark      .7              2015
1       Jones      .6              2010
2       Smith      .5              2009
...
```

Suppose we have a DataFrame named `df3` that contains single season 3 point field goal percentages of basketball players, of the form

```
        player      percentage      year
0       Hawkins    .7              2015
1       Clark      .6              2010
2       Jones      .5              2009
...
```

Write a Python program that finds a player with the largest sum of 2 point and 3 point field goal percentages, among all players whose names appear in both `df2` and `df3`. Also, explain in complete sentences why your program performs this task.

Solution.

```
import pandas as pd
import numpy as np

df = df2.set_index("player") + df3.set_index("player")
max_player = df["percentage"][df["percentage"] == np.max(df["percentage"])]
max_player.index[0]
```

The DataFrame `df` will have an index of player names, and the percentage columns will add from `df2` and `df3`. The command `df["percentage"] == np.max(df["percentage"])` will tell us which rows of `df` contain the largest percentages. So, the second line of the program will output the player(s) with the largest percentage value in `df`. Since `max_player` is technically a series with an index of player names, the last line of the program will output the player name with corresponding largest percentage in `df`.

4. Question 4

What is the output of the following program? Explain your reasoning.

```
import re
data = '''
"data-testid="bar-chart--results-bar" style="width:51%"
role="progressbar" aria-valuenow="51" class="jsx-4201391551
jsx-842384122 labeled-bar df white"><span data-testid=
"bar-chart--results-bar-percent" class="jsx-4201391551 jsx-842384122"
'''
search_string = r'jsx([\w-]{5})'

found_strings = re.findall(search_string, data)
found_strings
```

Solution.

```
['-4201', '-8423', '-4201', '-8423']
```

The search string looks for copies of `jsx` and reports the five subsequence characters (that are either alphanumeric or a minus sign). The output is a list of these matching strings. The command `[\w-]` asserts we are looking for alphanumeric or minus sign characters. Also `{5}` asks for five characters after `jsx`.

5. Question 5

Give an example showing that the singular value decomposition is not unique.

That is, find positive integers m, n, p and find a real $m \times n$ matrix A, $m \times m$ orthogonal matrices $U, \widetilde{U}$, $n \times n$ orthogonal matrices $V, \widetilde{V}$ and $p \times p$ diagonal matrices $D, \widetilde{D}$ (with $p \leq \min(m, n)$ and with nonzero diagonal entries) such that

$$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V = \widetilde{U} \begin{pmatrix} \widetilde{D} & 0 \\ 0 & 0 \end{pmatrix} \widetilde{V},$$

and such that either: $U \neq \widetilde{U}$, or $V \neq \widetilde{V}$, or $D \neq \widetilde{D}$.

Solution. When $m = n = p = 1$, we can write $A = 1 = 1 \cdot 1 \cdot 1$ where $U = D = V = 1$ and $A = (-1) \cdot 1 \cdot (-1)$ where $\widetilde{U} = \widetilde{V} = -1$.

Altneratively, we can use $m = n = p = 2$ and let $A = D = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ with U, V identity matrices, then use $\widetilde{U} = \widetilde{V} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.