Project Case	BINUS UNIVERSITY Software Laboratory Center
COMP7117 Artificial Neural Network	
Computer Science	**O202-COMP7117-BS01-00**
Valid on *Odd Semester Year 2019/2020*	**Revision 01**

1. Seluruh kelompok tidak diperkenankan untuk:
   The whole group is not allowed to:
   - Melihat sebagian atau seluruh proyek kelompok lain,
     Seeing a part or the whole project from other groups
   - Menyadur sebagian maupun seluruh proyek dari buku,
     Adapted a part or the whole project from the book
   - Mendownload sebagian maupun seluruh proyek dari internet,
     Downloading a part or the whole project from the internet,
   - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,
     Working with another theme which is not in accordance with the existing theme in the matter of the project,
   - Melakukan tindakan kecurangan lainnya,
     Committing other dishonest actions,
   - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.
     Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.

2. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.
   If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be "Zero"

3. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.
   Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted

4. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.
   Don't forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.

5. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:
   Marking percentage for this subject is described as follows:

Tugas Mandiri *Assignment*	**Proyek** *Project*	**UAP** *Final Exam*
40%	60%	-

6. Software yang digunakan pada matakuliah ini adalah sebagai berikut:
Software will be used in this subject are described as follows:

Software *Software*
Visual Studio Code Python 3.6 SciPy Scikit TensorFlow 1.10

7. Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:
File extensions should be included in assignment and project collection for this subject are described as follows:

Tugas Mandiri *Assignment*	**Proyek** *Project*
PY	PY

Soal
Case

Tech AI

You are working in **Tech AI,** an AI company that builds AI-based solutions for companies around the world. **Tech AI** currently has two projects, which are clustering and classification task. As a programmer at **Tech AI**, you are asked to build the model based on the existing dataset.

1. Clustering (Self-Organizing Map)

A certain marketplace has a project for **Tech AI**. They gave us the data of survey from their customers. Now, they want to know **how many classes of people** are spending money on their marketplace. To do this, you are going to use **Kohonen Self-Organizing Map** technique to **cluster the data**.

a. Dataset Description

Content

The given dataset contains **200 data of customers** including gender, age, and income.

Feature Description

The table below shows the feature descriptions in the dataset.

Table 1. Table of Feature Descriptions for Clustering

Category	Column	Description	Possible Value
Features	customer_id	The id of the customer	1 to 200
	gender	The gender of the customer	String
	age	The origin of the product	18 to 70
	annual_income	The annual income of customer in k ($)	15 to 137
	spending_score	The transaction rate of the customer	1 to 99

b. Feature Selection

Instead of using **the actual value** for the clustering, you are asked to **create features derived** from the **actual data**. The features requested are:

Feature	Derivation Formula

Gender	If (gender is "Male"):
	Gender = 0
	else if (gender is "Female"):
	Gender=1

Income	annual_income
Age	age
Spending Rate	spending_score

c. Feature Extraction

After the four features are extracted, you are asked to use **Principal Component Analysis (PCA)** to both clean the data and reduce the dimensionality even further.

The steps that you need to take are as follows:

1. **Select the features** as defined in the **Feature Selection** section
2. **Normalize** the data
3. **Analyze** the data with **Principal Component Analysis** to obtain the new components
4. Take the **highest 3 principal components** as the **input of your neural network**

d. Architecture

You need to **create your own architecture design** that will be **able to solve the given problem**.

Consider the following when building your architecture:

- **Number of input nodes** required
- **Number of clusters**

These considerations will be **accounted for in the grading process**.

e. Training

The training procedure of the neural network are as follows:

1. **Epoch** for the trainings is **5000**
2. **For each data** in the dataset, **find the winning node** by using **nearest distance**
3. **Update the neighbors around** the winning node in a **square pattern**
4. **Update the weight** of the network

f. Visualization

After the training is complete, use **matplotlib** to **visualize the clusters** generated by the self-organizing map.

2. Classification

A certain zoo has asked **Tech AI** to create a **model for animal classification based on their attributes**.

a. Dataset Description

Content

The given dataset contains **101 data of animals** that are **already labeled with the corresponding classes.** Each animal is also **accompanied by their information,** in which will define the classification of the animal.

Feature Description

The table below shows the feature descriptions in the dataset.

Table 2. Table of Feature Descriptions for Classification

Categor y	Column	Description	Possible Value
Features	animal_name	The name of the animal	String
	hair	Whether the animal has hair or not	Boolean(0,1)
	feathers	Whether the animal has feathers or not	Boolean(0,1)
	eggs	Whether the animal lay eggs or not	Boolean(0,1)
	milk	Whether the animal	Boolean(0,1)

		produces milk or not	
	airborne	Whether the animal is airborne or not	Boolean(0,1)
	aquatic	Whether the animal is aquatic or not	Boolean(0,1)
	predator	Whether the animal is predator or not	Boolean(0,1)
	toothed	Whether the animal has tooth or not	Boolean(0,1)
	backbone	Whether the animal has backbone or not	Boolean(0,1)
	breathes	Whether the animal breathes or not	Boolean(0,1)
	venomous	Whether the animal is venomous or not	Boolean(0,1)
	fins	Whether the animal has fins or not	Boolean(0,1)
	legs	The number of the legs of the animal	0,2,4,5,6,8
	tail	Whether the animal has tail or not	Boolean(0,1)
	domestic	Whether the animal is a domestic animal or not	Boolean(0,1)
	catsize	Whether the animal size is catsize or not	Boolean(0,1)
Output	class_type	The class of the animal	Mammal Fish Bird Invertebrate Bug Amphibian Reptile

b. Feature Selection

From the given dataset, here are the **features** that are going to be used by the model:

Feature Name (Column)
hair
feathers
eggs
milk
toothed

backbone
breathes
fins
legs
tail
domestic
catsize

While the **output** of the system will be:

Label Name
Class Type

c. **Feature Extraction**

Due to the **large number of features** that need to be considered in building the neural network, you want to **simplify the data** to make your network trains faster. While **reducing the complexity of the data is important**, **preserving the variance and relationship between the data is also important**. To solve those problems, your approach in **reducing the dimensionality** of the data is by using **Principal Component Analysis** technique.

The steps that you want to take are as follows:

1. **Select the features** as defined in the Feature Selection section
2. **Normalize** the data
3. **Analyze** the data with **Principal Component Analysis** to obtain the new components
4. Take the **highest 5 principal components** as the **input of your neural network**

d. **Architecture**

You need to **create your own architecture design** that will be **able to solve the given problem**.

Consider the following when building your architecture:

- **Number of input nodes** required
- **Number of output nodes** (classes) required
- **Whether hidden layer is required or not** (whether the case is a linearly separable case or not)

These considerations will be **accounted for in the grading process**.

e. **Training**

The training of the neural network use **70% of the dataset that picked randomly**. The training is done with **gradient descent** as the optimization formula for **5,000 epochs**. In addition, during the training, **20% of the dataset** should be used as the **validation dataset**.

The training procedure are as follows:

A. **Initialization**

The initialization step needs to be run once before starting the training iteration:

1. Take the **output** of the **Principal Component Analysis** as the **features**
2. **Initialize** the **weights** and **biases randomly**

B. **Iteration**

For **5,000 epochs**, repeat the following:

1. **Calculate the error** by comparing the output of the neural network to the target in the dataset using **mean squared error** (**MSE**)
2. **Update** the **weights and biases** using **gradient descent optimization**
3. **For every 100 epochs**, **print** the **current error** and **epoch number** to the console
4. **After reaching the 500th epoch**, **calculate the validation error** by passing the validation dataset. After that, **record the validation error** and **save the model to file**
5. **For every 500 epochs**, **get the new validation error** by passing in the validation dataset. If the **validation error is lower** than the previous validation error, **save the model to file**. If the **validation error is higher**, **do not save the model**

f. **Evaluation**

The neural network is to be **evaluated** based on the accuracy with **30% of the dataset after the training process** finished. The **accuracy** is calculated as follows:

$$ccuracy=\frac{number\ of\ correct\ result}{number\ of\ evaluation\ data}*100\%$$

Reference

- The dataset is obtained from Kaggle https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python and https://www.kaggle.com/uciml/zoo-animal-classification.