# Mixed Domain Mini Project

You are given a pickle file which contains a dictionary, the keys of the dictionary are:

- x_train: train data which shape is (2400, 2)
- y_train: labels of the train data
- d_train: domains of the train data
- x_valid: validation data which shape is (300, 2)
- y_valid: labels of the validation data
- d_valid: domains of the validation data

The data might be loaded using:

```
import pickle
data = pickle.load(open("./data.pkl","rb"))
```

One might plot the data using:

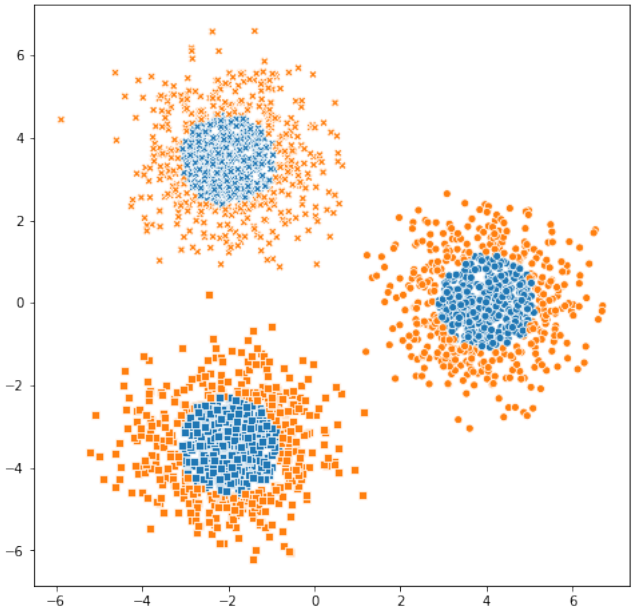
```
import matplotlib.pyplot as plt
import seaborn as sns

f, (ax1) = plt.subplots(nrows=1, ncols=1, figsize=(8, 8))
sns.scatterplot(
    x=data['x_train'][:, 0], y=data['x_train'][:, 1],
    hue=data['y_train'], style=data['d_train'], legend=False
)
plt.show()
```

The plot shows the train data, each Gaussian distribution is considered as a separate domain; each domain is plotted using different shape (cross, square, and circle). Each Gaussian distribution has two classes in it which are plotted using different colors (blue and orange) so here we are dealing with a binary-classification task.

The goal of this mini-project is to create a multi-domain feedforward neural network (FFNN).

- **The FFNN is composed of two hidden layers; the 1st one has 20 hidden units and the 2nd one has 10 hidden units.**




```
h1 = FFNN(input,hidden_units=20)
h2 = FFNN(h1, hiddent_units=10)
output = FFNN(h2,2)
```

- For each of the following questions run your code **using different random states** and report the **average accuracy and standard deviation** on the validation set provided
- Each one of the models should be trained for **30,000** epochs using an **Adam** optimizer and a learning rate of **1e-3**.
- A **ReLU** activation function should be used in each of the hidden layers.

In order to reproduce the candidate's results the random seed should be defined as a parameter. As random seeds you can use the following list [0,10,1234,99,2021] and report the average accuracy and standard deviation for each of the experiments.

## Question 1: Implement a basic FFNN

– One should write a class which contains a constructor and a forward function.

– One should train a multi-domain model over all data using the above mentioned random seeds and report the average accuracy and the standard deviation.

## Question 2: Implement a multi-domain model using knowledge distillation

Knowledge Distillation (KD) consists of transfering the knowledge from a model (also called teacher) to another model (also called student). In general cases, KD is used to transfer knowledge from big to smaller models and it showed significant improvements for the small models performances. KD might be formulated as follows:

$$L_{CE} = \mathcal{H}_{CE}(y, S(X))$$

$$L_{KD} = D_{KL}(\sigma(\frac{z_t(X)}{\tau}), \sigma(\frac{z_S(X)}{\tau}))$$

$$L = (1 - \alpha)L_{CE} + \alpha L_{KD}$$

where $\mathcal{H}_{CE}$ represents the cross entropy between the true label $y$ and the student network prediction $S(X)$ for a given input $X$, $D_{KL}$ KL is the KL divergence between the teacher and student predictions softened using the temperature parameter $\tau$, $z(X)$ is the network output before the softmax layer (logits), and $\sigma(.)$ indicates the softmax function. The term $\alpha$ in the above equation is a hyper-parameter which controls the amount of contribution from the cross entropy and KD loss.

In our case, we want to transfer knowledge from a single-domain teachers to a multi-domain student. To do so, one needs to start by training single domain models (using the same architecture mentionned above); in the current case, one needs to train 3 single-domain models.

After training single-domain models, one needs to train a multi-domain model using KD. In our case the KD loss should be computed between the student's logits and the in-domain teacher's logits of the current sample $X$. In other words, for a sample $X$ from domain $K$, the student must distill knowledge from the teacher trained on domain $K$.

**For KD experiments, you can set the default value of $\alpha$ to 0.5. The value of $\alpha$ should be set as parameter in order to change it easily.**

Similar to the previous experiments, one needs to run experiments using different random seeds and report the average accuracy and the standard deviation.

## Question 3: Try to come up with a technique that might boost the results for the multi-domain model.