Deadline: Friday Feb 5th , 10:00 Upload a `.pdf` file via LearnWeb. (e.g. exported Jupyter notebook)

1 Theory: Principal Componanet Analysis (10 points)

Principal Component Analysis (PCA) is a dimensionality reduction technique which aims at projecting a data set $X \in \mathbb{R}^{N \times M}$ via latent principal components $V \in \mathbb{R}^{K \times M}$ for $K << M$. The procedure aims at learning both the latent components and a linear combinations of the components via weights $Z \in \mathbb{R}^{N \times K}$, such that the original data is approximated via the following loss L:

$$\underset{Z,V}{\operatorname{argmin}} L = ||X - Z \cdot V||^2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(X_{i,j} - \sum_{k=1}^{K} Z_{i,k} V_{k,j} \right)^2 \tag{1}$$

Another method to compute the PCA of a data set is through gradient descent, where the latent data Z and the principal components Z are updated via computing the full gradient over L, as shown in Algorithm 1.

Algorithm 1 Compute PCA through Gradient Descent

Require: Original Data $X \in \mathbb{R}^{N \times M}$, Number of latent dimensions K, Learning Rate η, Number of epochs E
Ensure: Low-rank data $Z \in \mathbb{R}^{N \times K}$, Principal components $V \in \mathbb{R}^{K \times M}$

```
for 1, ..., E do
    for i = 1, ..., N j = 1, ..., M, k = 1, ..., K do
        V_{k,j} ← V_{k,j} − η ∂L/∂V_{k,j}
        Z_{i,k} ← Z_{i,k} − η ∂L/∂Z_{i,k}
    end for
end for
return Z, V
```

Derive the update rule gradients

A [5p] $\frac{\partial L}{\partial V_{k,j}}$.

B [5p] $\frac{\partial L}{\partial Z_{i,k}}$

2 Practice: Principal Component Analysis (10 points)

As in Tutorial 8, load the IRIS dataset via `sklearn.datasets.loadiris` and construct a 3:1 training-test split via `sklearn.modelselection.traintestsplit`; use 2020 as the random seed for part A and B.

A [2p] Compute the principal components using **only the training data**. What is the transformation from the old features to the new features?

B [2p] Make a plot of **the whole dataset**, using the first two principal components. (from part **??**)

C [6p] For $k = 1, 2, 3$ train two linear classifiers (you can use `sklearn`'s `LogisticRegression`): one on the original data, and one on the embedded data. Compare their performance by computing the mean and standard deviation of the test accuracy over 1000 independent runs for each classifier.