GE 461 Spring 2022 - Project 4 Telehealth Fall Detection

Zübeyir Bodur - 21702382
Computer Engineering
Bilkent University
Ankara, Turkey
zubeyir.bodur@ug.bilkent.edu.tr

Abstract

There is no abstract for this project

Index Terms

unsupervised learning, clustering, multi-layer perceptrons, support vector machines

I. Part A

A. Exploratory Data Analysis

The Principal Component Analysis (PCA) visualization of the Telehealth data set is given in Fig. 1. Current setting was reported to explain 46.21 % of the variance in the Telehealth data. It should be noted that, before using PCA, min-max normalization was used. The reason for using normalization was the outcomes in the 3^{rd} project of this course; that is, normalization has a huge effect on training an algorithm to learn about the data.

In fact, normalization has also has a good effect on visualization of the data as well. The original data has outliers and all of the data would be in the same line if we did not use normalization.

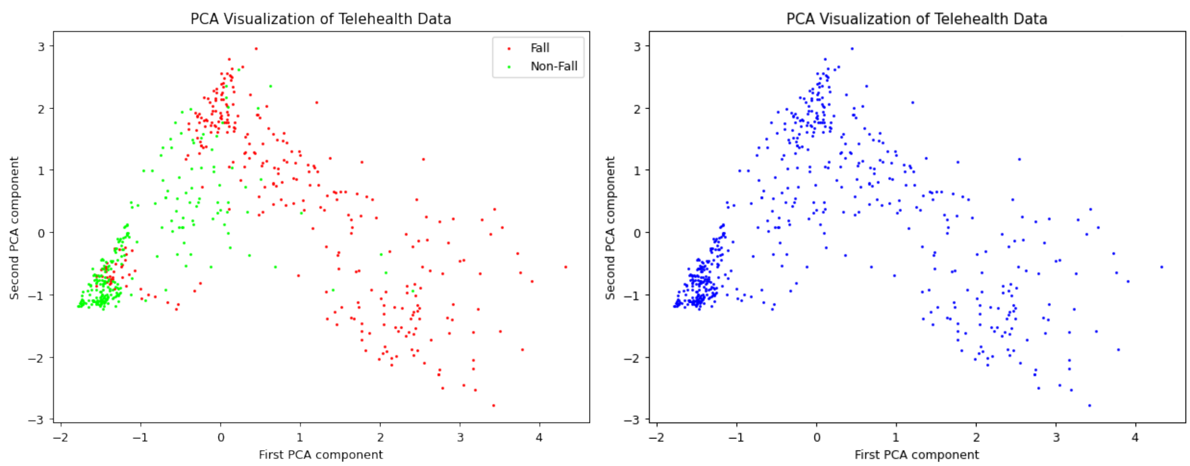

Fig. 1. PCA visualization of the Telehealth data set, with and without the labels

B. K-Means Clustering

After doing the proper pre-processing steps, namely normalization and dimensionality reduction, we can move on to the training step of the unsupervised learning part of this project.

While using K-Means, the K values ranging from 2 to 10 was used. The accuracy of each clustering result was computed based on weighted average of the accuracy of each cluster for each plot, and the results were put in Table I. While computing accuracy of a cluster, it was assumed that the majority of ground truth's of a cluster was considered as the predicted label of a cluster, then accuracy was computed accordingly. The clusters were plotted in Fig. 2-6.

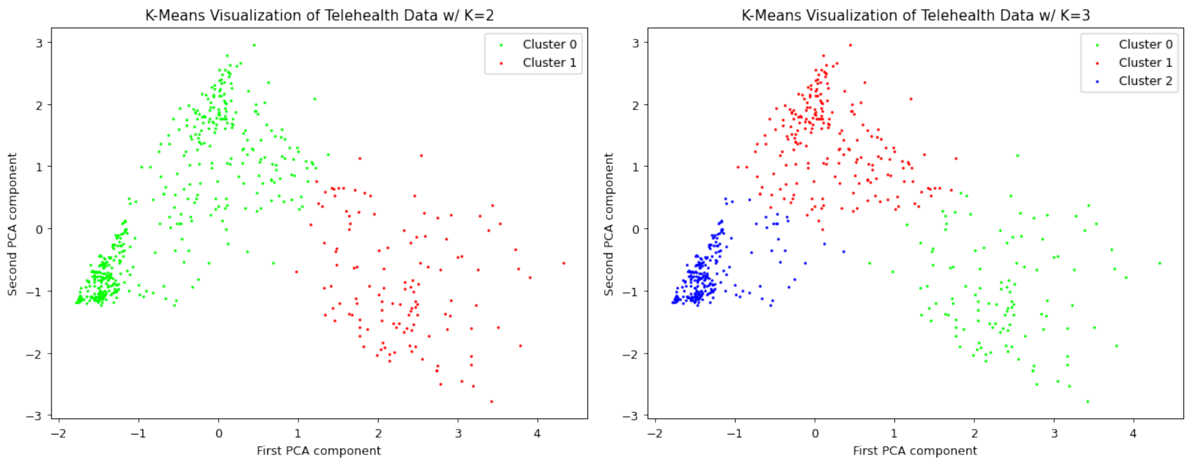

Fig. 2. K-Means clustering of the Telehealth data when K = 2 and 3

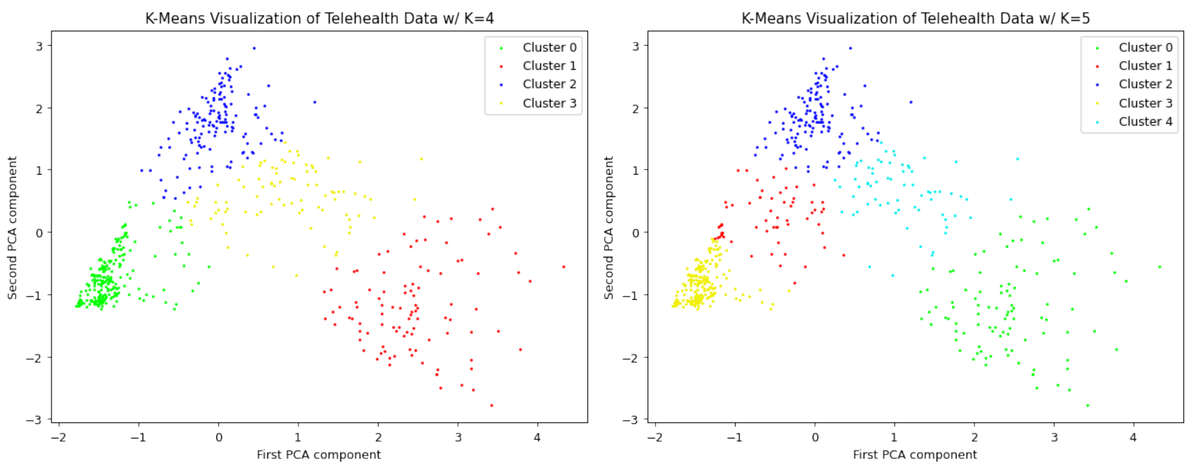

Fig. 3. K-Means clustering of the Telehealth data when K = 4 and 5

C. Results and Discussion

The weighted average of accuracies of each cluster is given in Table I. It can be inferred that, after value $K = 5$, the accuracy does not improve significantly, and in some occasions, it drops slightly. Therefore, minimum number of clusters required for this data set is 5.

TABLE I
MEAN ERRORS FOR TRAINING AND TEST PREDICTIONS.

K Value	Accuracy (%)
2	65.54
3	81.27
4	80.74
5	85.34
6	85.34
7	85.34
8	86.04
9	85.69
10	85.34

The percentages of accuracies show promising results about the normalization and the dimensionality of the data. It tells us that we can classify the incoming sensor data as Fall or Non-Fall using only 2 PCA components and a

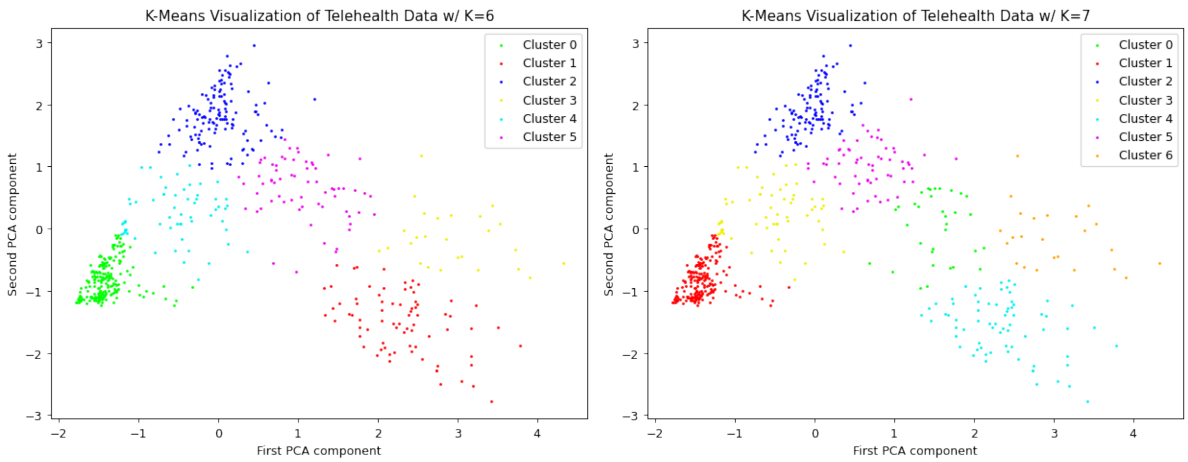

Fig. 4. K-Means clustering of the Telehealth data when K = 4 and 5

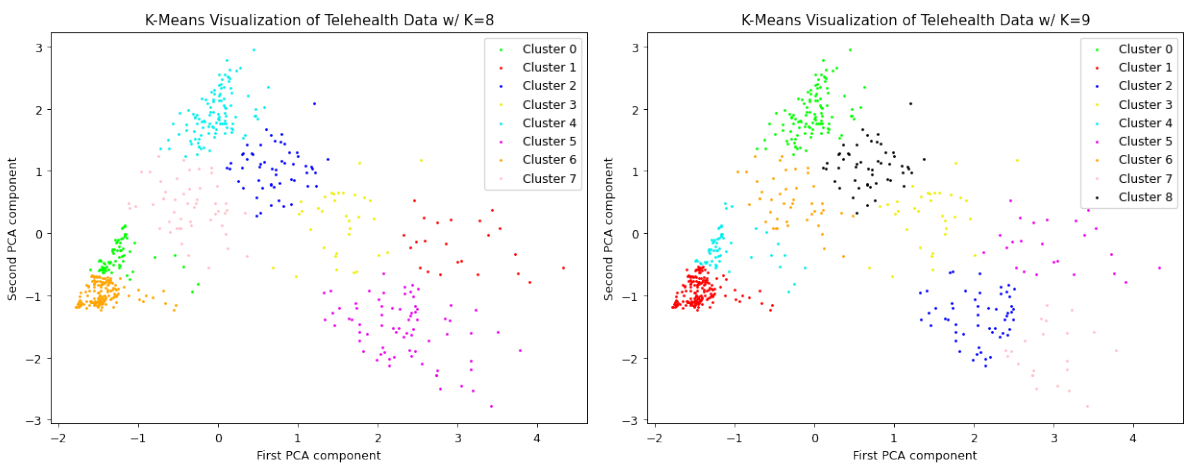

Fig. 5. K-Means clustering of the Telehealth data when K = 4 and 5

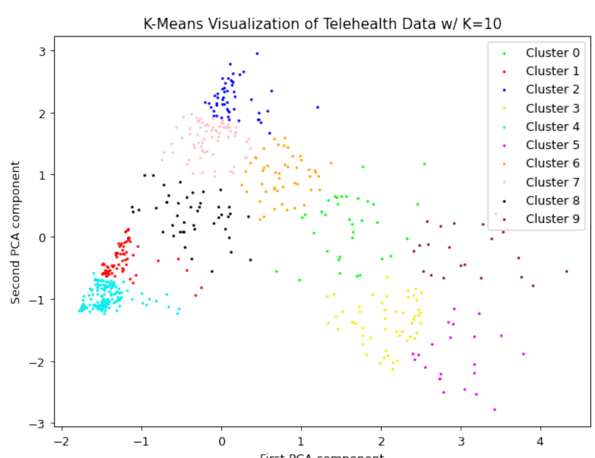

Fig. 6. K-Means clustering of the Telehealth data when K = 10

simple normalization method such as min-max normalization. In other words, this part of the assignment proves that sensor data is a decider for fall detection problem.

However, especially for applications in healthcare sector, accuracies such as 85 % should not be considered as a huge success due to the ethical outcomes. An accuracy above 95 % should be aimed, which is what I have done in Part B of this assignment.

In order to improve this, for instance, we will use more PCA components in order to explain more variance in our data.

II. Part B

We had observed that, from Project 2 of this course, using approximately 20-30 PCA components of a data set with 400 features performed the best in classification tasks. That's why, in this part of the project, we will be using first 25 PCA components to analyze the rest of the data. In addition, first 25 PCA components already explain 92.49 % of the variance in the data.

For both classifiers, the following procedure was followed to obtain a model that learns the data. After performing training (70 %) on each hyper-parameter setting in each classifier, the accuracy will be tested on the validation set (15 %). Then, the model with maximum accuracy will be selected to measure the accuracy using the test set (15 %).

A. Support Vector Machine (SVM)

a) Training Details: For this part, the following hyperparameters will be tuned:

- C = [10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2]
- Kernel Types = [”poly”, ”rbf”, ”sigmoid”]
- Degrees = [2, 3, 4, 5]. Those values were tuned for the polynomial kernel
- Kernel Coefficients = [”scale”, ”auto”]

Linear SVM will not be tuned because we already know that using non-linearity while learning a model is usually better, from Project 3 of this course. In fact, most classifiers use radial basis function (RBF) as the kernel for SVM's, but we will also try sigmoid and polynomial kernels for this data set.

b) Chosen Model and Results: From the Table A, we can observe that 11 models out of 72 scored 100 % validation accuracy. Since choosing RBF kernel is a common practice, we will be using the model with an RBF kernel. Therefore, the following model was chosen for testing, and had the following test accuracy:

- C = 1
- Kernel Type = RBF
- Kernel Coefficient = Auto
- Validation Accuracy = 100 %
- Testing Accuracy = 96.47 %

B. Multi-Layer Perceptron (MLP)

a) Training Details: For this part, the following hyperparameters will be tuned:

- Hidden Layer Size = [8x8, 16x16, 32x32, 64x64]
- Learning Rate (η) = [10^{-2}, $5 \cdot 10^{-2}$, 10^{-3}, $5 \cdot 10^{-4}$, 10^{-4}]
- Alpha (α) = [1, 10^{-1}, 10^{-2}, 10^{-3}]
- Solver = [”adam”, ”sgd”]. SGD refers to the Stochastic Gradient Descent algorithm.
- Activation Function = ReLU

We will use ReLU only, as our activation function. From the deep learning seminar's of this course, we can understand that some non-linear activation functions cause the gradient to vanish/explode. Therefore, we will use ReLU as our activation function and tune other parameters.

b) Chosen Model and Results: From the Table III, we can observe that 104 models out of 160 scored 100 % validation accuracy. In order to choose a model for testing, I have chosen the model that gave the best testing accuracy, which is given below.

- Hidden Layer Size = 8x8
- Learning Rate (η) = 10^{-2}
- Alpha (α) = 1
- Solver = Adam
- Activation Function = ReLU
- Validation Accuracy = 100 %
- Testing Accuracy = 97.65 %

C. Discussion

From the ratio of models that had 100 % accuracies, we can infer that MLP classifier was better than SVM classifier. Only 15.23 % of the SVM models had 100 % validation accuracies while all of the MLP models scored above 90 % and 65 % of them scored 100 %.

Moreover, the testing accuracy of the best models chosen were 1.18 % higher for the MLP classifier. This difference may not seem a lot in terms of making a decision. However, the following interpretation below shows how important it actually is to have 1 % more accurate results in healthcare applications.

The best model of the SVM classifier will miss detect 1 out of 28 actions recorded by the patient's sensor, while the MLP classifier will miss detect only 1 out of 43 actions recorded. Therefore, the MLP classifier chosen will be 33.42 % less prone to errors, which is a notable ratio, especially in a situation where the patients' health are concerned.

For both classifiers, though, we can infer that Telehealth data coming from wearible sensors are a decider for fall detection problem.

Appendix

TABLE II
VALIDATION RESULTS OF THE SVM CLASSIFIER

Regularization Parameter	Kernel Type	Degree	Kernel Coefficient	Validation Accuracy (%)
1	rbf	NULL	auto	100
1	sigmoid	NULL	scale	100
1	sigmoid	NULL	auto	100
10	poly	3	scale	100
10	rbf	NULL	scale	100
10	rbf	NULL	auto	100
10	sigmoid	NULL	auto	100
100	poly	2	scale	100
100	poly	3	scale	100
100	rbf	NULL	scale	100
100	rbf	NULL	auto	100
100	sigmoid	NULL	auto	100
1	poly	3	scale	98.82352941176471
1	rbf	NULL	scale	98.82352941176471
10	poly	2	scale	98.82352941176471
10	poly	5	scale	98.82352941176471
100	poly	5	scale	98.82352941176471
100	poly	2	auto	98.82352941176471
100	poly	3	auto	98.82352941176471
1	poly	2	scale	97.6470588235294
10	poly	4	scale	97.6470588235294
10	poly	2	auto	97.6470588235294
100	poly	4	scale	97.6470588235294
0.1	rbf	NULL	scale	96.47058823529412
10	sigmoid	NULL	scale	96.47058823529412
100	sigmoid	NULL	scale	96.47058823529412
1	poly	4	scale	95.29411764705881
1	poly	5	scale	95.29411764705881
10	poly	3	auto	95.29411764705881
100	poly	4	auto	95.29411764705881
0.1	sigmoid	NULL	scale	94.11764705882352
0.1	poly	2	scale	90.58823529411765
1	poly	3	auto	89.41176470588236
0.1	poly	3	scale	88.23529411764706
0.1	rbf	NULL	auto	88.23529411764706
1	poly	2	auto	88.23529411764706
0.1	sigmoid	NULL	auto	87.05882352941177
0.01	sigmoid	NULL	scale	84.70588235294117
100	poly	5	auto	84.70588235294117
0.1	poly	4	scale	81.17647058823529
10	poly	5	auto	80
10	poly	4	auto	78.82352941176471
0.1	poly	5	scale	76.47058823529412
0.01	poly	5	scale	60
0.001	poly	2	scale	58.82352941176471
0.001	poly	3	scale	58.82352941176471
0.001	poly	4	scale	58.82352941176471
0.001	poly	5	scale	58.82352941176471
0.001	poly	2	auto	58.82352941176471
0.001	poly	3	auto	58.82352941176471
0.001	poly	4	auto	58.82352941176471
0.001	poly	5	auto	58.82352941176471
0.001	rbf	NULL	scale	58.82352941176471
0.001	rbf	NULL	auto	58.82352941176471
0.001	sigmoid	NULL	scale	58.82352941176471
0.001	sigmoid	NULL	auto	58.82352941176471
0.01	poly	2	scale	58.82352941176471
0.01	poly	3	scale	58.82352941176471
0.01	poly	4	scale	58.82352941176471
0.01	poly	2	auto	58.82352941176471
0.01	poly	3	auto	58.82352941176471
0.01	poly	4	auto	58.82352941176471
0.01	poly	5	auto	58.82352941176471
0.01	rbf	NULL	scale	58.82352941176471
0.01	rbf	NULL	auto	58.82352941176471
0.01	sigmoid	NULL	auto	58.82352941176471
0.1	poly	2	auto	58.82352941176471
0.1	poly	3	auto	58.82352941176471
0.1	poly	4	auto	58.82352941176471
0.1	poly	5	auto	58.82352941176471
1	poly	4	auto	58.82352941176471
1	poly	5	auto	58.82352941176471

TABLE III
VALIDATION RESULTS OF THE MLP CLASSIFIER

Hidden Layer Size	Activation Function	Solver	Alpha	Learning Rate	Validation Accuracy (%)
(8, 8)	relu	adam	1	0.01	100
(8, 8)	relu	sgd	1	0.01	100
(8, 8)	relu	adam	0.1	0.01	100
(8, 8)	relu	sgd	0.1	0.01	100
(8, 8)	relu	adam	0.01	0.01	100
(8, 8)	relu	sgd	0.01	0.01	100
(8, 8)	relu	adam	0.001	0.01	100
(8, 8)	relu	sgd	0.001	0.01	100
(8, 8)	relu	adam	1	0.05	100
(8, 8)	relu	sgd	1	0.05	100
(8, 8)	relu	adam	0.1	0.05	100
(8, 8)	relu	sgd	0.1	0.05	100
(8, 8)	relu	adam	0.01	0.05	100
(8, 8)	relu	sgd	0.01	0.05	100
(8, 8)	relu	adam	0.001	0.05	100
(8, 8)	relu	sgd	0.001	0.05	100
(8, 8)	relu	adam	1	0.001	100
(8, 8)	relu	sgd	1	0.001	100
(8, 8)	relu	adam	0.1	0.001	100
(8, 8)	relu	sgd	0.1	0.001	100
(8, 8)	relu	adam	0.01	0.001	100
(8, 8)	relu	sgd	0.01	0.001	100
(8, 8)	relu	adam	0.001	0.001	100
(8, 8)	relu	sgd	0.001	0.001	100
(8, 8)	relu	adam	1	0.0005	100
(8, 8)	relu	adam	0.1	0.0005	100
(8, 8)	relu	adam	0.01	0.0005	100
(8, 8)	relu	adam	0.001	0.0005	100
(8, 8)	relu	adam	1	0.0001	100
(8, 8)	relu	adam	0.1	0.0001	100
(8, 8)	relu	adam	0.01	0.0001	100
(8, 8)	relu	adam	0.001	0.0001	100
(16, 16)	relu	adam	1	0.01	100
(16, 16)	relu	sgd	1	0.01	100
(16, 16)	relu	adam	0.1	0.01	100
(16, 16)	relu	adam	0.01	0.01	100
(16, 16)	relu	adam	0.001	0.01	100
(16, 16)	relu	adam	1	0.05	100
(16, 16)	relu	sgd	1	0.05	100
(16, 16)	relu	adam	0.1	0.05	100
(16, 16)	relu	adam	0.01	0.05	100
(16, 16)	relu	adam	0.001	0.05	100
(16, 16)	relu	adam	1	0.001	100
(16, 16)	relu	adam	0.1	0.001	100
(16, 16)	relu	adam	0.01	0.001	100
(16, 16)	relu	adam	0.001	0.001	100
(16, 16)	relu	adam	1	0.0005	100
(16, 16)	relu	adam	0.1	0.0005	100
(16, 16)	relu	adam	0.01	0.0005	100
(16, 16)	relu	adam	0.001	0.0005	100
(32, 32)	relu	adam	1	0.01	100
(32, 32)	relu	sgd	1	0.01	100
(32, 32)	relu	adam	0.1	0.01	100
(32, 32)	relu	sgd	0.1	0.01	100
(32, 32)	relu	adam	0.01	0.01	100
(32, 32)	relu	sgd	0.01	0.01	100
(32, 32)	relu	adam	0.001	0.01	100
(32, 32)	relu	sgd	0.001	0.01	100
(32, 32)	relu	adam	1	0.05	100
(32, 32)	relu	sgd	1	0.05	100
(32, 32)	relu	adam	0.1	0.05	100
(32, 32)	relu	sgd	0.1	0.05	100
(32, 32)	relu	adam	0.01	0.05	100
(32, 32)	relu	sgd	0.01	0.05	100
(32, 32)	relu	adam	0.001	0.05	100
(32, 32)	relu	sgd	0.001	0.05	100
(32, 32)	relu	adam	1	0.001	100
(32, 32)	relu	sgd	1	0.001	100
(32, 32)	relu	adam	0.1	0.001	100

Hidden Layer Size	Activation Function	Solver	Alpha	Learning Rate	Validation Accuracy (%)
(32, 32)	relu	sgd	0.1	0.001	100
(32, 32)	relu	adam	0.01	0.001	100
(32, 32)	relu	sgd	0.01	0.001	100
(32, 32)	relu	adam	0.001	0.001	100
(32, 32)	relu	sgd	0.001	0.001	100
(32, 32)	relu	adam	1	0.0005	100
(32, 32)	relu	adam	0.1	0.0005	100
(32, 32)	relu	adam	0.01	0.0005	100
(32, 32)	relu	adam	0.001	0.0005	100
(32, 32)	relu	adam	1	0.0001	100
(64, 64)	relu	adam	1	0.01	100
(64, 64)	relu	sgd	1	0.01	100
(64, 64)	relu	adam	0.1	0.01	100
(64, 64)	relu	sgd	0.1	0.01	100
(64, 64)	relu	adam	0.01	0.01	100
(64, 64)	relu	sgd	0.01	0.01	100
(64, 64)	relu	adam	0.001	0.01	100
(64, 64)	relu	sgd	0.001	0.01	100
(64, 64)	relu	adam	1	0.05	100
(64, 64)	relu	sgd	1	0.05	100
(64, 64)	relu	adam	0.1	0.05	100
(64, 64)	relu	sgd	0.1	0.05	100
(64, 64)	relu	adam	0.01	0.05	100
(64, 64)	relu	sgd	0.01	0.05	100
(64, 64)	relu	adam	0.001	0.05	100
(64, 64)	relu	sgd	0.001	0.05	100
(64, 64)	relu	adam	1	0.001	100
(64, 64)	relu	adam	0.1	0.001	100
(64, 64)	relu	adam	0.01	0.001	100
(64, 64)	relu	adam	0.001	0.001	100
(64, 64)	relu	adam	1	0.0005	100
(64, 64)	relu	adam	0.1	0.0005	100
(64, 64)	relu	adam	0.01	0.0005	100
(64, 64)	relu	adam	0.001	0.0005	100
(64, 64)	relu	adam	1	0.0001	100
(8, 8)	relu	sgd	1	0.0005	98.82352941176471
(8, 8)	relu	sgd	0.1	0.0005	98.82352941176471
(8, 8)	relu	sgd	0.01	0.0005	98.82352941176471
(8, 8)	relu	sgd	0.001	0.0005	98.82352941176471
(16, 16)	relu	sgd	0.1	0.01	98.82352941176471
(16, 16)	relu	sgd	0.01	0.01	98.82352941176471
(16, 16)	relu	sgd	0.001	0.01	98.82352941176471
(16, 16)	relu	sgd	0.1	0.05	98.82352941176471
(16, 16)	relu	sgd	0.01	0.05	98.82352941176471
(16, 16)	relu	sgd	0.001	0.05	98.82352941176471
(16, 16)	relu	sgd	1	0.001	98.82352941176471
(16, 16)	relu	sgd	0.1	0.001	98.82352941176471
(16, 16)	relu	sgd	0.01	0.001	98.82352941176471
(16, 16)	relu	sgd	0.001	0.001	98.82352941176471
(16, 16)	relu	adam	1	0.0001	98.82352941176471
(16, 16)	relu	adam	0.1	0.0001	98.82352941176471
(16, 16)	relu	adam	0.01	0.0001	98.82352941176471
(16, 16)	relu	adam	0.001	0.0001	98.82352941176471
(32, 32)	relu	sgd	1	0.0005	98.82352941176471
(32, 32)	relu	sgd	0.1	0.0005	98.82352941176471
(32, 32)	relu	sgd	0.01	0.0005	98.82352941176471
(32, 32)	relu	sgd	0.001	0.0005	98.82352941176471
(32, 32)	relu	adam	0.1	0.0001	98.82352941176471
(32, 32)	relu	adam	0.01	0.0001	98.82352941176471
(32, 32)	relu	adam	0.001	0.0001	98.82352941176471
(64, 64)	relu	sgd	1	0.001	98.82352941176471
(64, 64)	relu	sgd	0.1	0.001	98.82352941176471
(64, 64)	relu	sgd	0.01	0.001	98.82352941176471
(64, 64)	relu	sgd	0.001	0.001	98.82352941176471

Hidden Layer Size	Activation Function	Solver	Alpha	Learning Rate	Validation Accuracy (%)
(64, 64)	relu	sgd	1	0.0005	98.82352941176471
(64, 64)	relu	sgd	0.1	0.0005	98.82352941176471
(64, 64)	relu	sgd	0.01	0.0005	98.82352941176471
(64, 64)	relu	sgd	0.001	0.0005	98.82352941176471
(64, 64)	relu	adam	0.1	0.0001	98.82352941176471
(64, 64)	relu	adam	0.01	0.0001	98.82352941176471
(64, 64)	relu	adam	0.001	0.0001	98.82352941176471
(16, 16)	relu	sgd	1	0.0005	97.6470588235294
(16, 16)	relu	sgd	0.1	0.0005	97.6470588235294
(16, 16)	relu	sgd	0.01	0.0005	97.6470588235294
(16, 16)	relu	sgd	0.001	0.0005	97.6470588235294
(8, 8)	relu	sgd	1	0.0001	95.29411764705881
(8, 8)	relu	sgd	0.1	0.0001	95.29411764705881
(8, 8)	relu	sgd	0.01	0.0001	95.29411764705881
(8, 8)	relu	sgd	0.001	0.0001	95.29411764705881
(32, 32)	relu	sgd	1	0.0001	95.29411764705881
(32, 32)	relu	sgd	0.1	0.0001	95.29411764705881
(32, 32)	relu	sgd	0.01	0.0001	95.29411764705881
(32, 32)	relu	sgd	0.001	0.0001	95.29411764705881
(64, 64)	relu	sgd	1	0.0001	92.94117647058823
(64, 64)	relu	sgd	0.1	0.0001	92.94117647058823
(64, 64)	relu	sgd	0.01	0.0001	92.94117647058823
(64, 64)	relu	sgd	0.001	0.0001	92.94117647058823
(16, 16)	relu	sgd	0.1	0.0001	91.76470588235294
(16, 16)	relu	sgd	0.01	0.0001	91.76470588235294
(16, 16)	relu	sgd	0.001	0.0001	91.76470588235294
(16, 16)	relu	sgd	1	0.0001	90.58823529411765