Report-Challenge 1

Da Vinchie Lisa, Pettenà Piero

November 21, 2023

Contents

1 Introduction

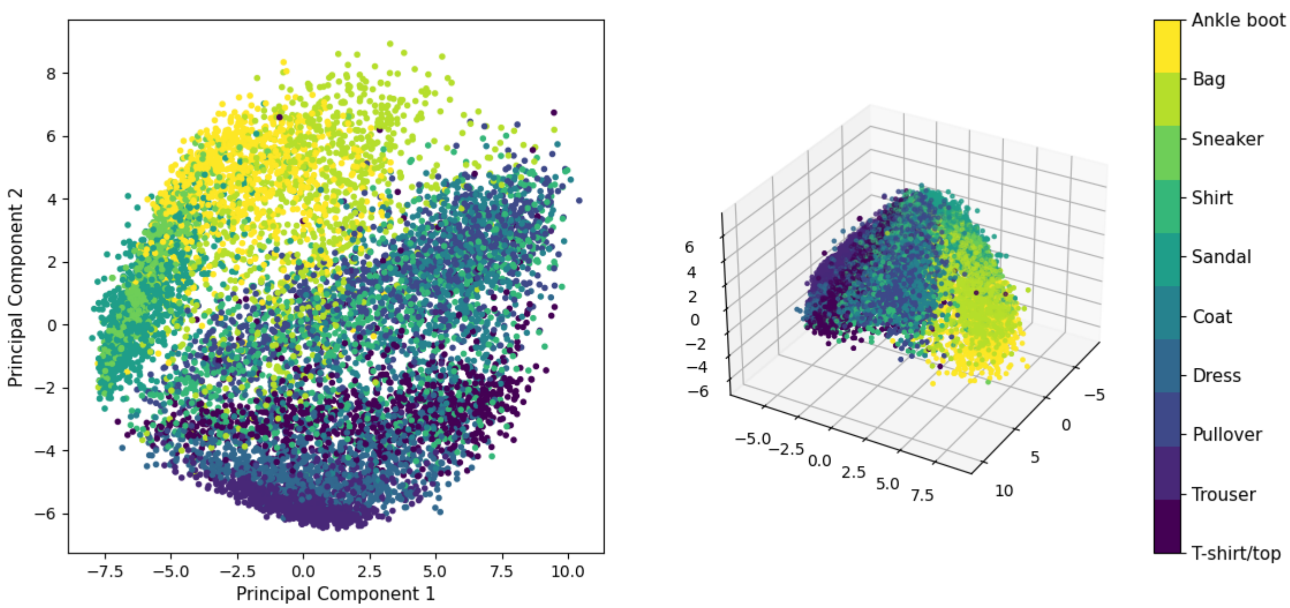
In this challenge, our goal is to find an effective method to classify the images of the *Fashion-MNIST* dataset, based on their content. The dataset is formed by black-and-white images of 28×28 pixel, each one representing a clothing item belonging to one of the following 10 cathegories:

1. T-shirt or top
2. Trouser
3. Pullover
4. Dress
5. Coat
6. Sandal

7. Shirt
8. Sneaker
9. Bag
10. Ankle boot

2 Exercise 1

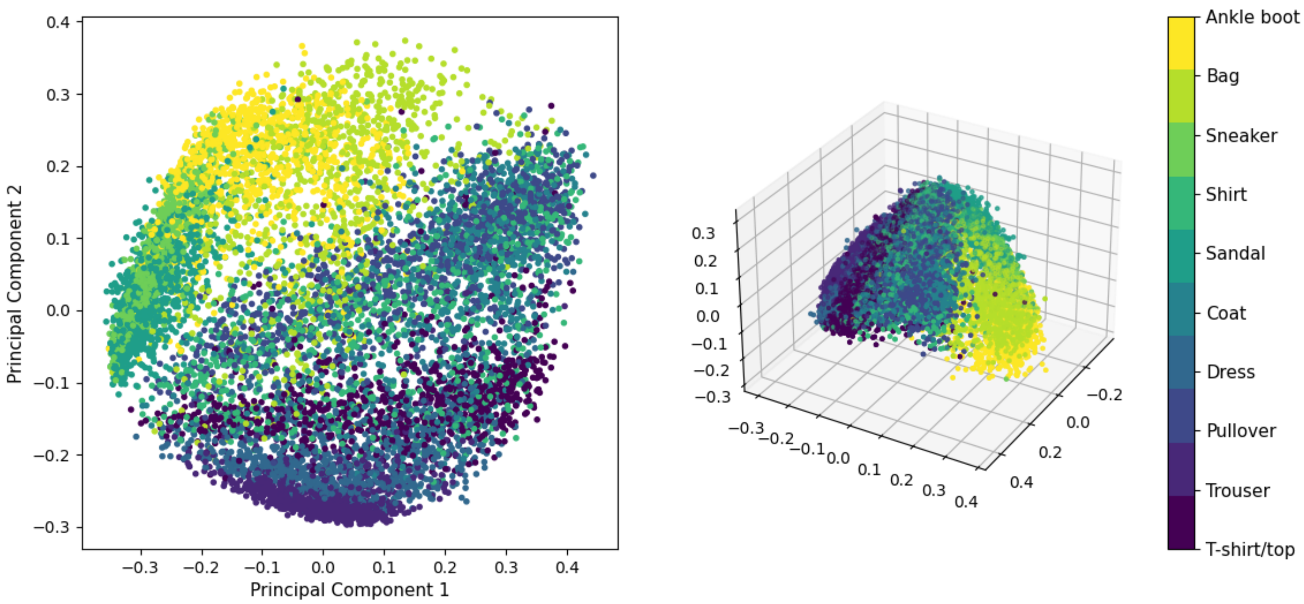
The goal of this exercise is to perform Principal Component Analysis (PCA) in order to find out how the clusters are separated. In order to do that, we performed two kinds of PCA: linear and kernel PCA; then, we plotted the first two and three principal components, along with the true labels.
We started by using a linear PCA, obtaining the clustering in Figure 1; although the datapoints appear to be grouped in clusters, it is clear how they are not well separated; moreover, it happens that some classes have datapoints that are very far from the centroid of the cluster, as happens for class 0.



(a) (b)

Figure 1: First two and three principal components obtained by performing linear PCA, plotted along with true labels

Performing Kernel PCA with a Radial Basis Function kernel, using the default value of the dispersion parameter $\gamma = 1/\#\text{samples} = 1/784$, does not lead to better results, since the classes are still very mixed up, as we can see in Figure 2.



(a) (b)

Figure 2: First two and three principal components obtained by performing kernel PCA with RBF kernel, plotted along with true labels

We tried to do the same with the polynomial and sigmoid kernels, as shown in Figure 3, the but none of them separates clearly the clusters. In particular, looking at the three-components plots, it seems that datapoints with labels *Ankle boot*, *Bag*, *Shirt* and *T-Shirt/Top* are easyer to separate, while the others are mixed up.

3 Exercise 2

Even though no PCA method separates the datapoints in a satisfying way, we decided that the kernel PCA with sigmoid kernel is the one that works better. Then, we performed unsupervised clustering on those data, using three methods: K-means clustering, spectral clustering and Gaussian Mixture, obtaining the results showed in Figure 4.

Trying to confront the obtained results with the original labels using an accuracy score is useless, since the names of the clusters will differ. In order to provide a quantitative measure of similarity, we decided to use the *adjusted rand index*: this method computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters, in the predicted and true clusterings. It has values that range between 0 and 1, with 1 meaning that the two clusters are exactly the same.

The results that we obtained are the following:

- K-means: 0.3628

- Spectral clustering: 0.4342
- Gaussian mixture: 0.3794

As we expected, the clusters that we found using unsupervised learnign do not resemble much the ones formed by the true labels.
This, probably, is due to the fact that the clusters are not clearly separable in the first place, and that maybe 10 clusters are too many: by looking at the eigenvalues of each principal component in Figure 5, we can see how the knee-point is clearly on the third component.

4 Exercise 3

In this exercise, we have to perform supervised classification, using the original images associated with one of the label sets found in the previous exercise; since the clustering made with the spectral method scored the highest rand inex, we decided to use this set of labels.
To perform supervised classification, we used 3 different methods:

- SVM, with different kernels;
- Fully Connected Neural Network;
- Convolutional Neural Network.

In all the three cases, we splitted our 10000 rows dataset into a train test, of 7000 rows, and a test set, of 3000 rows; this is necessary in order to evaluate the generality of the model.

4.1 Support Vector Machine

We performed this kind of supervised classification using the four kernels that we used for the PCA in the first exercise: linear, RBF, polynomial and sigmoid. The accuracy scores that we obtained with each kernel are the followings:

- Linear: 0.9657
- RBF: 0.9733
- Polynomial: 0.9617
- Sigmoid: 0.3453

4.2 Fully Connected Neural Network

We started by considering a simple neural network, with just 1 fully connected layer and a Softmax activation function; we trained it for different numbers of epochs between 0 and 20, using stochastic gradiend descent with learning rate 0.01 as the optimizer, anc aclculated the test accuracy using the cross entropy loss. As expected, the test accuracy increases with the number of epochs, as illustrated in Figure 6a.
We, then, tried a sligtly more complex neural network, with two fully connected layers and,

again, the softmax activation function; in this case, we also need to choose the number of neurons of the first layer, along with the number of neurons. We started by choosing a number of neurons of 850 and study how the accuracy grows with the number of epochs; as in Figure 6b, we can see how we obtain better results than in the previous model.

Fixing the number of epochs to 8 and changing the number of neurons between 50 and 1050, instead we obtaon the plot in Figure 7; here, the accuracy increases very slowly with the number of neurons. In conclusion, we can say that using two fully connected layers seems better than using just one; concerning the number of epochs and neurons, it seems better to prioritize the first one.

4.3 Convolutional Neural Network

To test the performance of convolutional neural networks, we followed a similar approach: we started by considering a simple CNN composed by one two-dimensional convolutional layer, with 2×2 max pooling, and one fully connected layer; the obtained test accuracy, varying for the number of epochs for which the model was trained, is reported in Figure 8a. Then, we tried a more complex CNN, with two convolutional layers, with 2×2 max pooling, and one fully connected layer; the accuracies obtained by varying the number of epochs and neurons are reported, respectively in Figures 8b and 9

As we can see, the model with 2 convolutional layers returns better results with less training epochs.

5 Results

6 Conclusion

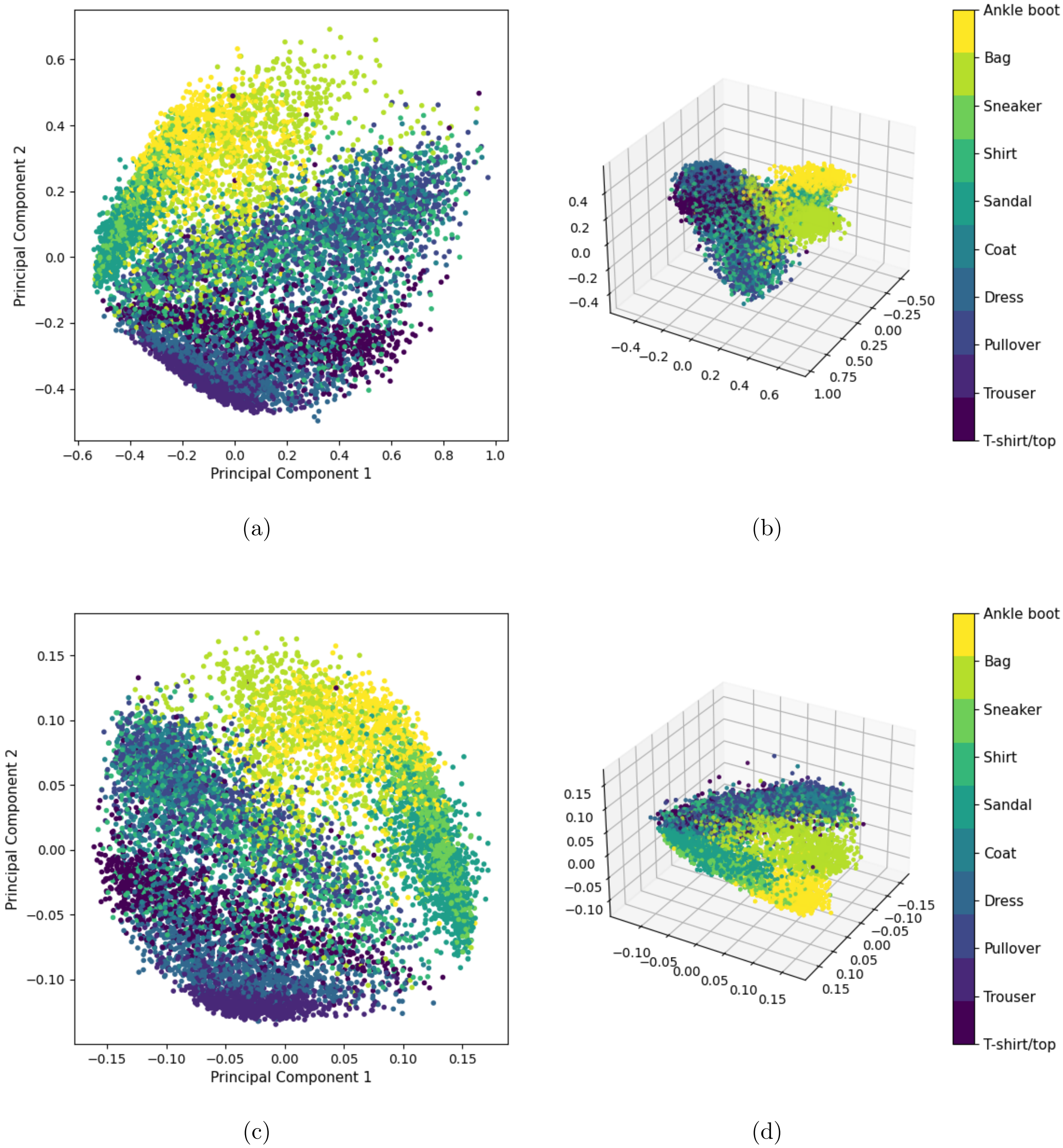


Figure 3: First two and three principal components obtained by performing kernel PCA with polynomial ((a) and (b)) and sigmoid ((c) and (d)) kernel, plotted along with true labels.

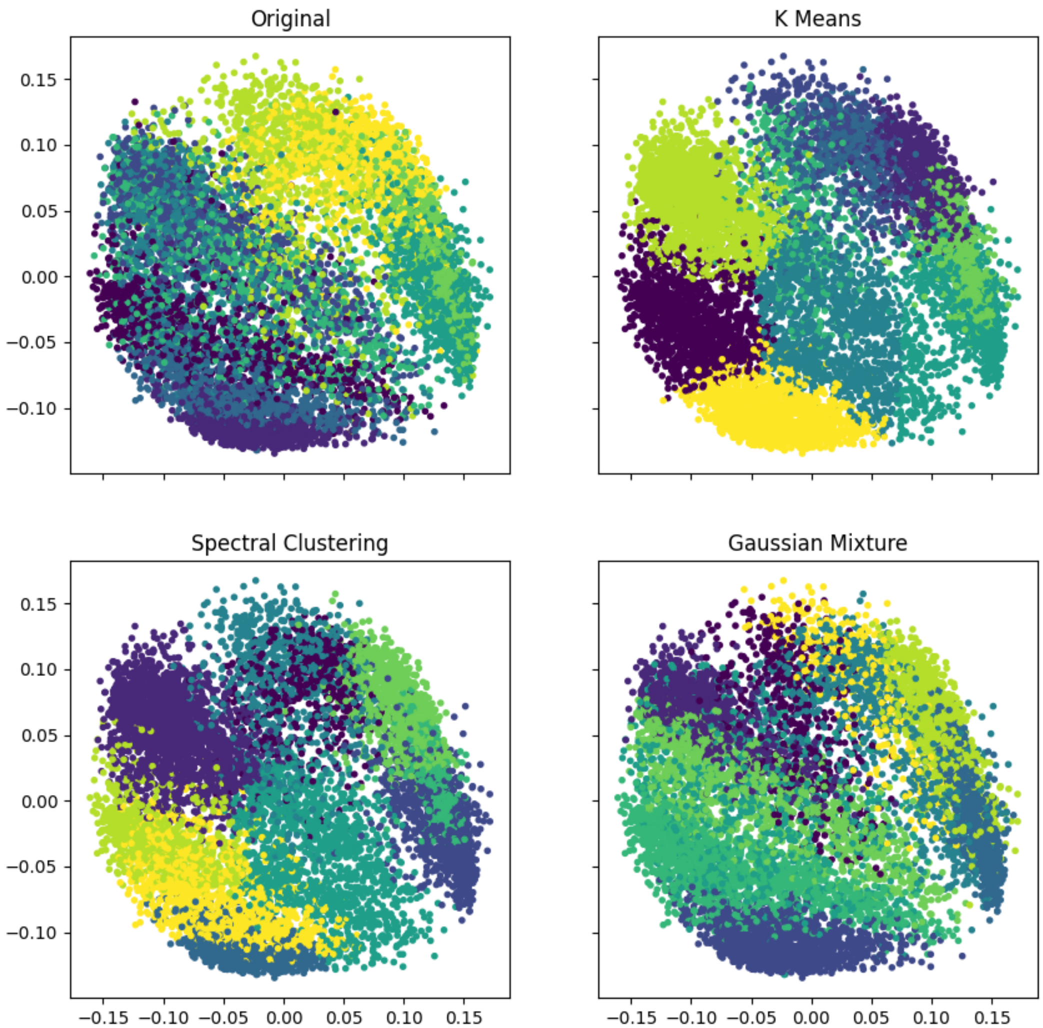


Figure 4: Unsupervised clustering obtained using K-means clustering, spectral clustering and gaussian mixture, confronted with the true labels.

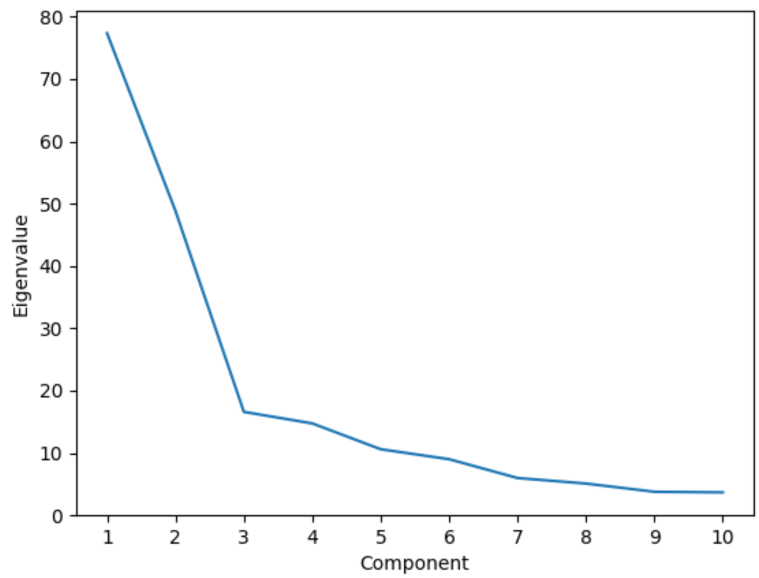
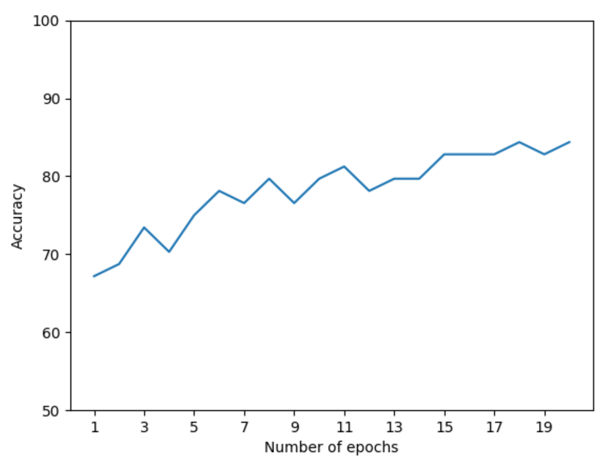
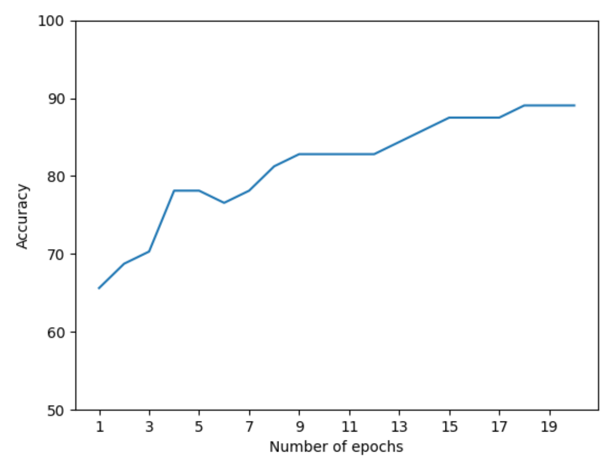


Figure 5: First 10 eigenvalues obtained by performing kernel PCA with sigmoid kernel.



(a)



(b)

Figure 6: Test accuracy obtained by using a 1-layer (Figure a) and 2-layer (Figure b) fully connected neural network trained for different numbers of epochs

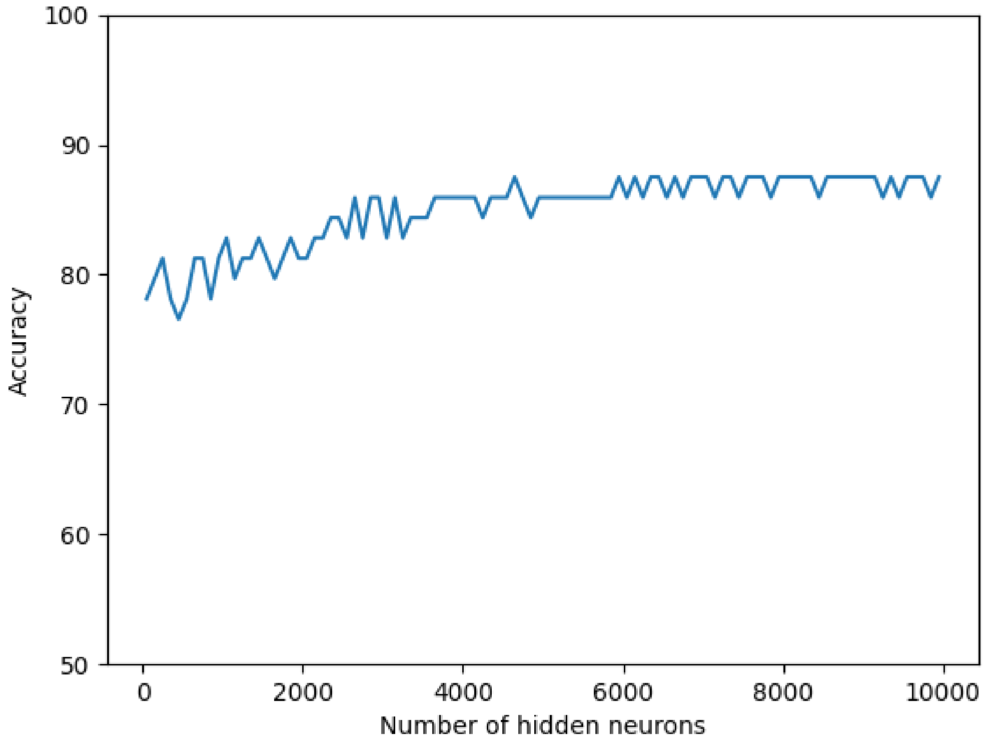
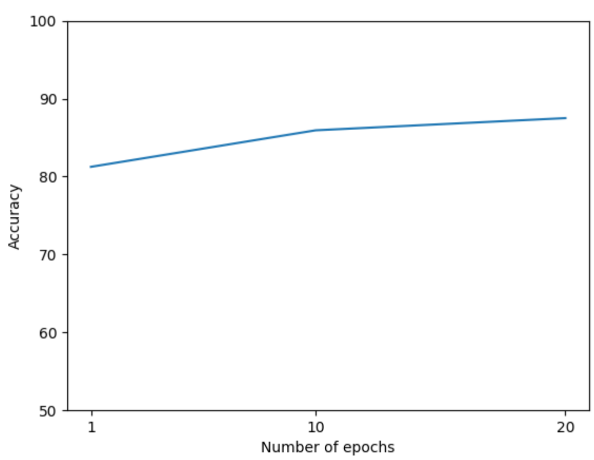
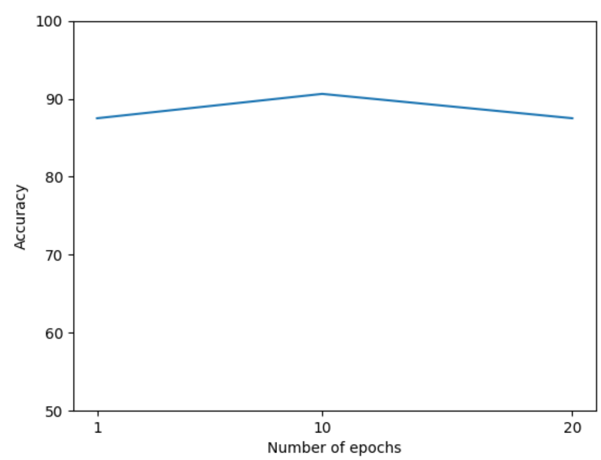


Figure 7: Test accuracy obtained by using a 2-layer fully connected neural network with different numbers of neurons per hidden layer, trained for 8 epochs



(a)



(b)

Figure 8: Test accuracy obtained by using a 1-layer (Figure a) and 2-layer with 100 neurons per hidden layer (Figure b) convolutional neural network trained for different numbers of epochs

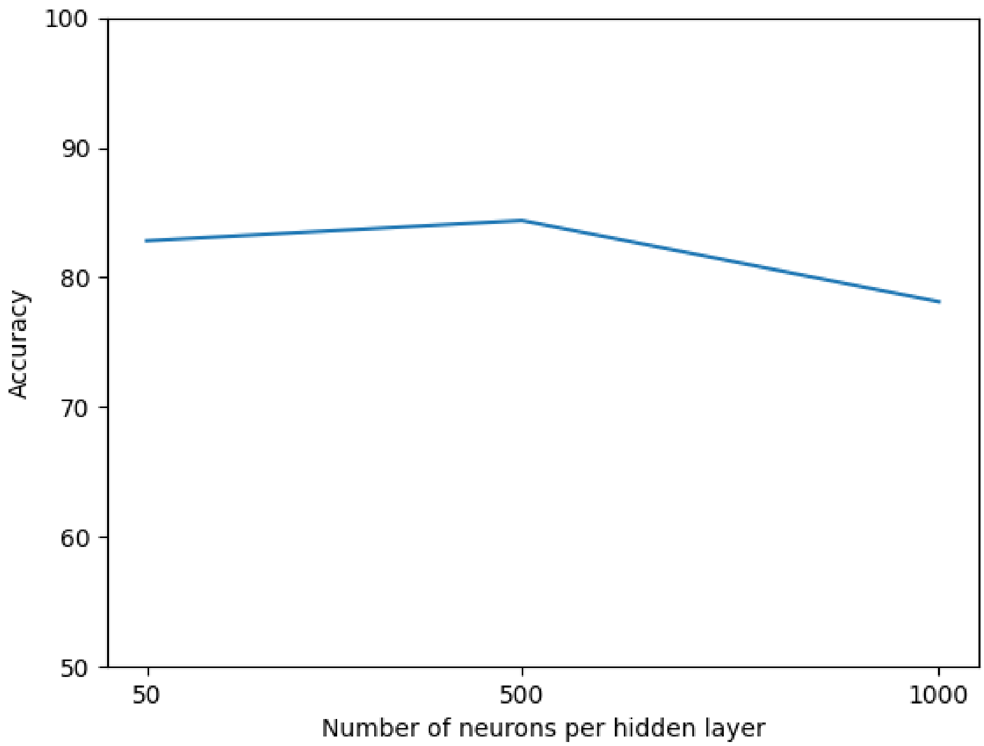


Figure 9: Test accuracy obtained by using a 2-layer fully connected neural network with different numbers of neurons per hidden layer, trained for 2 epochs.