Auxiliary feature learning for small dataset regularization
CE888 Assignment 1 - Project 1

Cynthia Masetto, 1802774

***Abstract*— Dimensionality reduction is often used as a preprocessing step in classification. There are dimensionality reduction techniques which transform the high dimensional data to new lower dimensional data, the autoencoders are now a popular method that are used to reduce the dimensions of the input features. This work perform dimensionality reduction autoencoders which features provided will help learn a discriminative classifier where the instances of the class are linearly separable from instances of other classes.**

I. Introduction

Nowadays, real-world data has high dimensionality. A lot of data preprocessing techniques have been suggested towards the optimization of the classification process. Dimensionality reduction is one of them and it's often used as a preprocessing step in classification. As the number of dimensions of a data increases, it becomes more difficult to process it. For that reason, classifiers with low-dimensional inputs perform a faster training and are able to learn a better classifier. Moreover, with small datasets overfitting could be avoided.

Dimension reduction methods are based on the assumption that dimension of data is artificially inflated and its intrinsic dimension is much lower. Principal Component Analysis (PCA) is one of the most popular linear dimension reduction, which transforms the high dimension data to lower dimension data. However, this is not the only method that can be used and one example are the autoencoders. The autoencoders are unsupervised Artificial Neural Networks trained so that they can reconstruct their input through an intermediate representation, typically of lower dimension.[10] This work is organised as follows: in the first part, the methods for dimensionality reduction will be explained a long with the autoencoders and simple linear classifiers, in the second part, data exploration and visualisation of 3 datasets obtained from Kaggle are perform, respectively, in the third part, a dimensionality reduction autoencoder that will learn a discriminative classifier is analysed, and finally, discussion and conclusion about the method are summarized.

II. Background

A. Dimensionality Reduction

Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality. Basically, the reduced representation should have a dimensionality that corresponds to the intrinsic dimensionality of the data. This means that the intrinsic dimensionality is the minimum number of parameters needed to account for the observed properties of the data. As a result, dimensionality reduction facilitates classification, visualisation, etc.[2]

Furthermore, dimensionality reduction can remove two types of "noise" from the input. The first noise is the independent random noise, which is uncorrelated with the input and the label, one example of this would be running Principal Components Analysis (PCA) or other unsupervised dimensionality reduction algorithm. The second noise are the unwanted degrees of freedom, which are possibly nonlinear, along which the input changes but the label does not. For that reason a more radical form of denoising requires the dimensionality reduction to be informed by the labels, that is why it is called supervised dimensionality reduction.[3]

On the other hand, dimensionality reduction has being performed using linear techniques such as: PCA, factor analysis and classical scaling[2]. However, the problem arises when complex nonlinear data appears in the picture and this kind of techniques cannot handle it.

For this reason, a number of nonlinear techniques for dimensionality reduction have been proposed. These techniques have the ability to deal with complex nonlinear data. Moreover, previous studies have shown that nonlinear techniques outperform their linear counterparts on complex artificial tasks. Some of those techniques are: Kernel PCA, Isomap, Maximum Variance Unfolding, Difussion Maps, Locally Linear Embedding, Laplacian Eigenmaps, Local Tangent Space Analysis, Sammon Mapping, Multilayer Autoencoders, Locally Linear Coordination and Manifold Charting.[2]

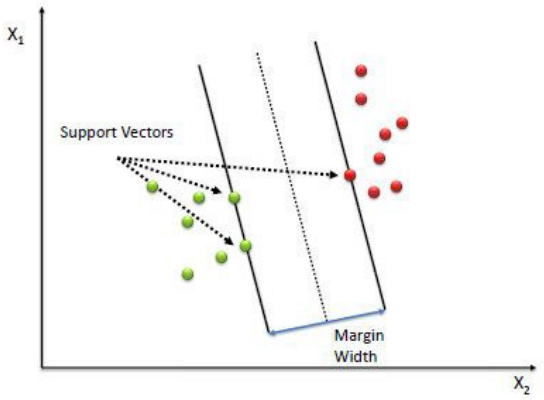
B. Discriminative Classifiers - Support Vector Machines

Discriminative classifiers are a popular approach to solving classification problems, one example of that are the Support vector machines, which are an approximate implementation of structural risk minimisation.[11] Support vector machines are binary classifiers, where the decision boundary is estimated to maximise the margin, the distance from the decision boundary to the closest points from each of the classes. Moreover, they have been found to yield good performance on a wide range of tasks and are suitable for use with data in high dimensional spaces.[11]

Support vector machines when used for classification, they separate a given set of binary labeled training data with a hyper-plane that is maximally distant from them. For cases in which no linear separation is possible, they can work in combination with the technique of "kernels", that automatically realizes a non-linear mapping to a feature space. The hyper-plane found by the SVM in feature space

corresponds to a non-linear decision boundary in the input space.

Fig. 1. Support Vector Machine



Source: Yadav, Ajay,*"Support Vector Machines"*, Towards Data Science, 2018.

As we can see in the Figure 1, the points that are closer to the hyper-plane are called "support vector points" an the distance of the vectors from the hyper-plane are called margins.[12]

C. Autoencoders

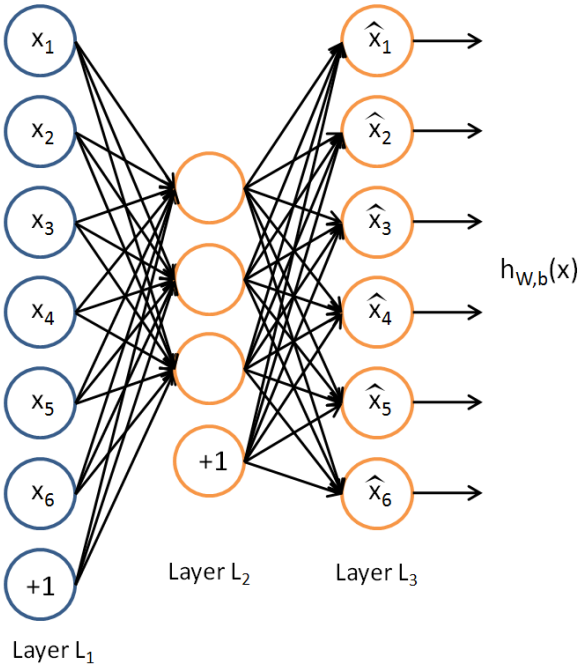
Autoencoder is an unsupervised tool for feature learning. Like other feature learning algorithms, the goal of autoencoder is to produce a good representation of the input data.[1] Autoencoders are widely used in different image classification tasks, speech emotion recognition and distribution estimation.

Specifically, multilayer autoencoders are feed-forward neural networks with an odd number of hidden layers, and share weights between the top and bottom layers.[2] Here, the network is trained to minimize the mean squared error between the input and the output of the network (ideally, the input and the output are equal).

According to Andrew Ng (2012), an autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs ($y^{(i)}$=$x^{(i)}$).

The autoencoder tries to learn an approximation to the identity function, so the output $\hat{x}$ is similar to x. In this case, the identity function is a particularly trivial function that tries to learn by placing constraints on the network such as: limiting the number of hidden units, that could give us some insights about the data. [4] Sometimes the autoencoder often ends up learning a low-dimensional representation very similar to PCAs. This happens when the number of hidden units is small, however when the number of hidden units is large more constraints on the network can be imposed and the autoencoder will still discover interesting structure in the data. However, the main weakness of autoencoders is that their training may be tedious.

Fig. 2. Example of an autoencoder

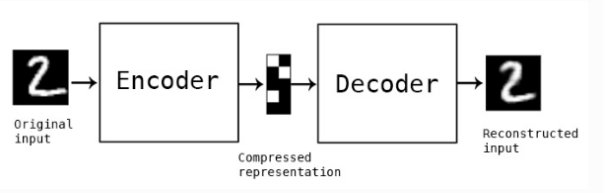


Source: Ng, Andrew, et al,*"Unsupervised Feature Learning and Deep Learning"*, Sandford University, 2012.

On the other hand, Chollet (2016) describes the autoencoder as a data compression algorithm where the compression and decompression functions are: data specific, which means that they compress data similar to what they have been trained on, lossy, which means that the decompressed outputs are degraded compared to the original inputs, and finally, they are learned automatically from examples rather than engineered by a human, which means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input.

The following image shows how an autoencoder reconstructs the input:

Fig. 3. Autoencoder reconstructing an input



Source: Chollet, Francois, *"Building Autoencoders in Keras"*, The Keras Blog, 2016.

An encoding, decoding and distance functions are needed to build an autoencoder. The distance function is the amount of information loss between the compressed representation of your data and the decompressed representation. Moreover, the encoder and decoder will be chosen to be parametric functions and to be differentiable with respect to the distance function, so the parameters of the encoding/decoding

functions can be optimize to minimize the reconstruction loss using the Stochastic Gradient Descent.[5]

Furthermore, autoencoders can learn data projections that are more interesting than other techniques like PCA, where they can perform dimensionality reduction. Although autoencoders can solve the problem of unsupervised learning, they are also a self-supervised technique where they apply supervised learning and the targets are generated from the input data. On the other hand, in order to get self-supervised models to learn interesting features it is necessary to have a synthetic target and a loss function.

There are several types of autoencoders. First, **regularized autoencoders** that use regularization terms in their loss functions to achieve desired properties. Within these autoencoderes, there are the sparse autoencoders which adds a penalty on the sparsity of the hidden layer. Here, regularization forces the hidden layer to activate only some of the hidden units per data sample. This means that the output from a deactivated node to the next layer is zero. This restriction forces the network to condense and store only important features of the data. The other kind of regularized autoencoders are the denoising ones, which a random noise is deliberately added to the input and network is forced to reconstruct the unadulterated input. Here, the decoder function resists small changes in the input and a neural network that resists noise in input results. [6] Moreover, within regularized autoencoders, there also exist the contractive autoencoders, here instead of adding noise to input, contractive autoencoders add a penalty on the large value of derivative of the feature extraction function.

Second, the **variational autoencoders**, which are based on nonlinear latent variable models. These autoencoders consist of two neural networks, the first one for learning the latent variable distribution and the second one, for generating the observables from a random sample obtained from latent variable distribution. These autoencoders, minimize the loss and also the difference between the assumed distribution of latent variables and the distribution resulting from the encoder.

Third, **undercomplete autoencoder**, where the size of the hidden layer is smaller than the input layer in udercomplete autoencoders, here by reducing the hidden layer size we force the network to learn the important features of the dataset. Once the training is over, the decoder part is discarded and the encoder is used to transform a data sample to feature subspace. If the decoder transformation is linear and loss function is MSE the feature subspace is the same as that of PCA. [6]

III. Methodology

Following the dimensionality reduction with autoencoders methodology, where the class information is used in order to define new discriminant targets, this approach will use autoencoders to learn the representation of the data and then train a simple linear classifier to classify the dataset into respective classes.

As part of the work the following datasets are proposed to the analysis:

1) Credit Card Fraud Detection [7]
2) Cervical Cancer Risk Classification[8]
3) Black Friday[9]

The databases were obtained through Kaggle, which is an online community of data scientist and machine learners that allows users to find and publish data sets, explore and build models in a web-based data-science environment. All datasets and codes are saved in the following **GitHub Repository**: `https://github.com/CynthiaMasetto/CE888-Assignmen1`

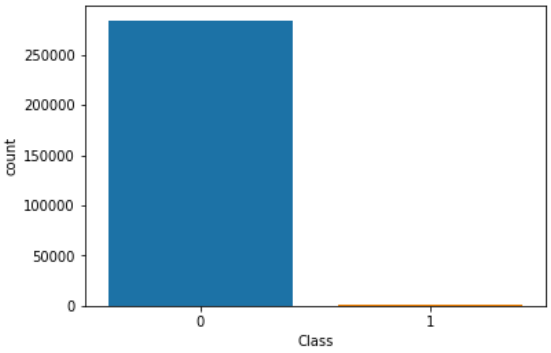
A. Credit Card Fraud Detection

This dataset contains transactions made by credit cards in September 2013. This dataset presents transactions that occurred in two days, where 492 frauds were found out of 284,807 transactions.[7]

For confidentiality reasons, the dataset contains only numerical input variables which are the result of a PCA transformation. Features V1,V2,...,V28 are the principal components obtained with PCA, the only features of this dataset which have not been transformed with PCA are *Time* and *Amount*. Feature *Time*, contains the seconds elapsed between each transaction and the first transaction in the dataset, on the other hand, feature *Amount* is the transaction amount, finally the feature *Class* is the response variable and it takes value 1 in case of fraud and 0 otherwise.[7]

In the following graph, the class can be seen:

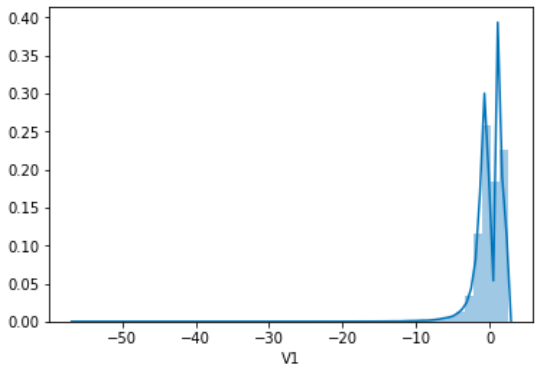
Fig. 4. Class "Fraud"



Source: Own analysis with data from kaggle.

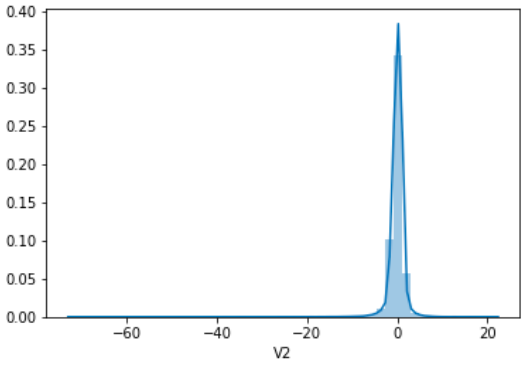
As it was mentioned, the Class = 1 accounts for 0.172% of all transactions which means that the data set is unbalanced. During future analysis this is going to be taken into account. Also, in this first approach, the distribution of the first two features of the dataset are shown on the following figures: At this moment, nothing could be tell about the features, but when the analysis starts we will be able to see which these features could represent.

Fig. 5. Feature "V1"



Source: Own analysis with data from kaggle.

Fig. 6. Feature "V2"



Source: Own analysis with data from kaggle.

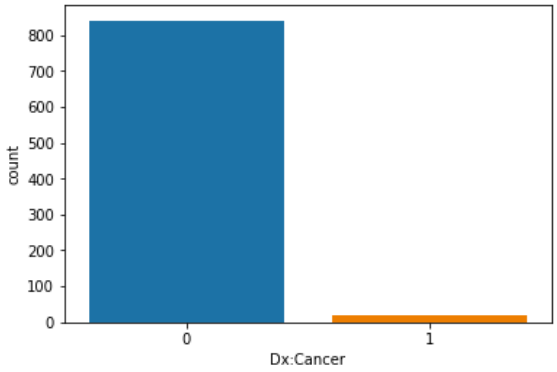
B. Cervical Cancer Risk Classification

The dataset is obtained from UCI repository, which is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms and contains a list of risk factors for cervical cancer. According to the World Health Organization (WHO), almost all cases of cervical cancer are caused by HPV. HPV is a very common virus that can be passed on through any type of sexual contact with a man or a woman. Cervical cancer occurs when the cells of the cervix grow abnormally and invade other tissues and organs of the body.[12] Worldwide, cervical cancer is the fourth most frequent cancer in women.

The following variables are contained in the dataset: age, number of sexual partners, first sexual intercourse, number of pregnancies, smokes, smokes (years and packs), hormonal contraceptives, hormonal contraceptives(years), IUD (with years), STDs, diagnosis, hinselmann, shciller, citology and biopsy. The class of diagnosis is 1 if cancer is positive and 0 otherwise.[8]

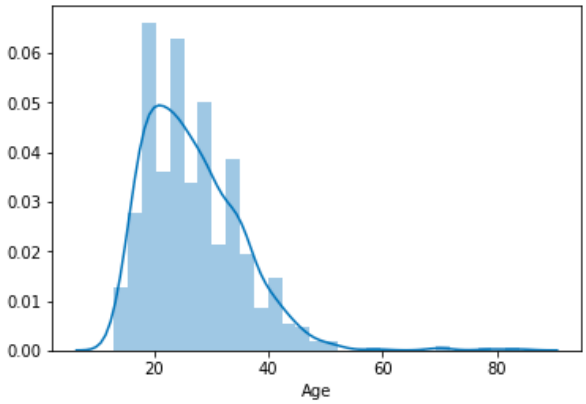
In this first approach, the graphs for class = 1 (positive cancer) and the age distribution is showed:

Fig. 7. Class cancer



Source: Own analysis with data from kaggle.

Fig. 8. Distribution of age



Source: Own analysis with data from kaggle.

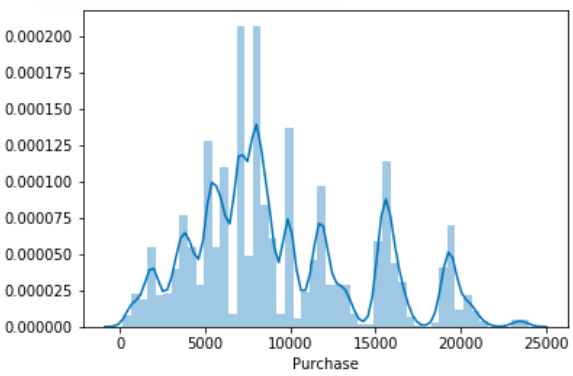
As we can see, in the graph, the distribution is centered between 20-40 years, however at this moment of the analysis we don't have any insights about how this features will affect the prediction of cancer.

C. Black Friday

The Black Friday is an informal name for the Friday following Thanksgiving Day in the United States. The day after Thanksgiving has been regarded as the beginning of America's Christmas shopping season since 1952. Many stores offer highly promoted sales on Black Friday, such are the volume of costumers in this date that during 2017 over $59.57 billion dollars were spent. The dataset is a sample of 555,000 observations about the transactions made in a retail store during Black Friday. The variables contained in the dataset are: user id, product id, gender, age, occupation, city, marital status, product category and purchase amount in dollars.[9]

The following graph shows the distribution of the purchases made:

Fig. 9. Purchases Distribution



Source: Own analysis with data from kaggle.

As we can see there is a lot of variation in the data and in further analysis we will take a deeper look into that.

IV. Experiments

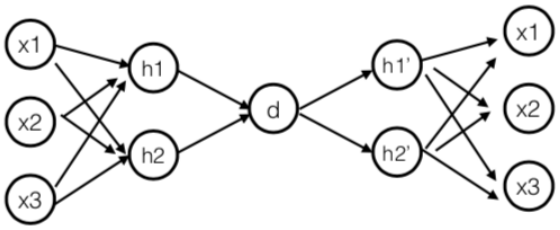
In this work autoencoders are used to reduce the dimension of the input features. Here, autoencoders are constructed by training the data in a way that the input and output are the same. All the datasets are use to perform this analysis and the features resulted from the autoencoders are used to train the classification algorithms, which in this case is the Support vector machine (SVM), which corresponds to a linear classifier that tries to identify which class a the input belongs to (class = 1 or class=0) by making a decision based on the value of a linear combination of the features. Moreover, the advantage of this method is that is not biased by outliers and it is not sensitive to overfitting.

Autoencoders are trained in a unsupervised manner in order to learn the representations of the input data. This data are then deformed back to project the actual data. This means that an autoencoder is a regression task where the network is asked to predict its input. In order to build an autoencoder, a fullyconnected neural layer as encoder and as decoder are created. Where, as mentioned before, the encoder is the part of the network that compresses the input into a latent space representation. The encoder layer encodes the input data as a compressed representation in a reduced dimension that will later be coded to fed the decoder. On the other hand, the decoder will decode the disorted version of the original input and encode the input back to the original dimension. The decoded input will be a lossy reconstruction of the original input and it is reconstructed from the latent space representation.

As shown in the image, a highly fine tuned autoencoder model should be able to reconstruct the same input which was passed in the first layer, the key is to use a rule for that, this rule is the equation that can be defined as the learning process.[14]

After training the autoencoder and using the selected features for the discriminative classifier, the model performance will be tested trhough the classification matrix, where accuracy, recall, precision will be measured.

Fig. 10. Autoencoder Architecture



Source: Kaggle, *"How autoencoders work? Intro and use cases"*, 2019.

V. Discussion

The goal of the autoencoder is to produce a good representation of the input data. As it was mentioned before, autoencoders are widely used in different image classification tasks, speech emotion recognition, distribution estimation and they can also perform dimensionality reduction. Once the autoencoder is trained, the features were used to perform a discriminative classifier analysis.

References

[1] Angshuman, Paul, et al., *"Discriminative Autoencoder"*, 2018.

[2] Van der Maaten, Laurens, et al., *"Dimensionality Reduction: Acomparative Review"*, Tiburg University, The Netherlands, 2009.

[3] Wang, Weiran, et al., *"The role of Dimensionality Reduction in Classification*, Association for the Advancement of Artificial Intelligence, University of California, 2014.

[4] Ng, Andrew, et al., *"Unsupervised Feature Learning and Deep Learning"*, Sandford University, 2012. Online resource: `http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial`

[5] Chollet, Francois, *"Building Autoencoders in Keras"*, The Keras Blog, 2016. Online resource: `https://blog.keras.io/building-autoencoders-in-keras.html`

[6] Kumar Pal, Ashwini*"Dimension Reduction Autoencoders"*, Machine Learning Blog, 2018. Online resource: `https://blog.paperspace.com/dimension-reduction-with-autoencoders/`

[7] Credit Card Fraud Detection. Anonymized credit card transactions labeled as fraudulent or genuine. 2018. Online resource: `https://www.kaggle.com/mlg-ulb/creditcardfraud`

[8] Cervical Cancer Risk Classification. Prediction of cancer indicators. Online resource: `https://www.kaggle.com/loveall/cervical-cancer-risk-classification`

[9] Black Friday. A study of sales through consumer behaviours. Online resource: `https://www.kaggle.com/mehdidag/black-friday`

[10] Nousi, Paraskevi, et al.,*"Deep learning algorithms for discriminant autoencoding"*, Department of Informatics, Aristotle University of Thessaloniki, Greece, 2017.

[11] Gales, M.J.F., et al.,*"Discriminative classifiers with adaptive kernes for noise robust speech recognition*, Cambridge University Engineering Department. 2009.

[12] Yadav, Ajay*"Support Vector Machines"*, Towards Data Science, 2018. Online resource: `https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589`

[13] WHO, World Health Organization, *"Cervical cancer"*, 2019. Online resource: `https://www.who.int/cancer/prevention/diagnosis-screening/cervical-cancer/en/`

[14] Kaggle, *"How autoencoders work? Intro and use cases"*, 2019. Online resource: `https://www.kaggle.com/shivamb/how-autoencoders-work-intro-and-usecases`

Plan: Breakdown of the work needed to complete this project		
Task	Dates	Time
Data Exploration	25th February - 1st March	1 week
Data Cleansing	25th February - 1st March	1 week
Data preprocessing	25th February - 1st March	1 week
Autoencoders training	4th March - 8th March	1 week
Feature selection	11th March - 15th March	1 week
Classification	11th March - 15th March	1 week
Evaluation	18th March - 23th March	1 and a half week
Reporting results	18th March - 23th March	1 and a half week