Influence of Data Dimension Reduction, Feature Scaling and Activation Function on Machine Learning Performance

Grzegorz Słowiński

University of Technology and Economics, ul. Jagiellońska 82f, 03-301 Warsaw, Poland

Abstract

A dataset containing over 13k samples of dry beans geometric features is being analysed using machine learning (ML) and deep learning (DL) techniques with the goal to automatically classify the bean specie. The obtained geometrical data has quite a lot redundancy. Many features are strongly correlated. This work analyses the influence of data dimension reduction (DDR) (elimination of excess strongly correlated features) and features scaling (FS), often called normalization, on the machine learning performance (measured in terms of accuracy and approximate training time). Additionally also an influence of activation function (sigmoid vs. ReLU) on artificial neural network performance has been checked.

Keywords

machine learning, deep learning, data dimension reduction, features scaling, activation function

1. Introduction

Classification of dry beans is of some economic importance. Manual classification is labour intensive, etc. Over 13 k samples of dry beans of 7 various species were photographed and their geometry was measured via computer vision techniques in [1]. Then the set was analysed via several machine learning (or data science) and deep learning (or artificial neural network) techniques. The overall accuracy obtained was 87.92-93.13%, depending on technique used.

The dataset used in [1] has been published in the UCI machine learning repository [2]. In this work, a collection of beans was used as material for investigation how machine learning process is influenced by the following factors: 1) data dimension reduction, 2) features scaling (or data normalization) and 3) in case of neural networks, how their performance depends on activation function used (ReLU vs. sigmoid).

The research question examined in this work is: How do data dimension reduction, feature scaling and activation function influence machine learning performance? The above question is related to concurency, specification and programming in the following way. Among topics of CS&P 2021 one can find: Model checking and testing - this work checks different ML models, knowledge discovery and data mining - machine learning belong to this field, soft computing - artificial neural networks are are categorized as a kind of soft-computing.

1.1. Data Dimension Reduction

[1]29th International Workshop on Concurrency, Specification and Programming (CS&P'21)
EMAIL: grzegorz.slowinski@uth.edu.pl
ORCID: 0000-0001-9770-5063


In the work [1] data dimension has not been reduced, although many features are strongly correlated. This work investigates the effect of data dimension reduction on performance (computing time and accuracy).

1.2. Feature scaling

In the handbook [4], page 72 Aurelien Geron, states: "One of the most important transformations you need to apply to your data is feature scaling. With few exceptions, Machine Learning algorithms don't perform well when the input numerical attributes have very different scales." This work verifies this statement and investigates what ML methods really needs feature scaling.

1.3. Activation Function

In work [1] ANN with sigmoid activation in hidden layers has been applied. This work investigates how ANN performance depends on activation function used. Two activation function are compared: ReLU and sigmoid.

2. Tools

The entire analysis was done using Python and its ML frameworks: numpy, pandas, matplotlib, seaborn, scikit-learn and keras. Google Colab a free cloud version of jupyter notebook was used. The reader can find the Python scripts under link [3]. Parameters of compute engine used were: Intel(R) Xeon(R) CPU @ 2.30GHz, 12,69 GB RAM, no graphical processing unit (GPU) acceleration. Majority of experiments performed were shallow learning that do not need GPU support. As the dry beans dataset is relatively simple, the artificial neural network (ANN) applied was also rather simple and GPU support was not crucial for ANN training. Training times were in range from milliseconds to a few minutes.

3. Data

The dataset under study consists of 13611 samples. A sample amounts to 16 geometrical features and a label identifying the specie of the bean. The species are as follows: Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira. The features are: Area, Perimeter, MajorAxisLength, MinorAxisLength, AspectRatio, Eccentricity, ConvexArea, EquivDiameter, Extent, Solidity, Roundness, Compactness, ShapeFactor1, ShapeFactor2, ShapeFactor3, and ShapeFactor4. A detailed explanation how the features were calculated is presented in [1].

Table 1.
Correlation between beans features

	Area	Peri-meter	Major Axis Length	Minor Axis Length	Aspect Ratio	Eccentri-city	Convex Area	Equiv Diameter	Extent	Solidity	Round-ness	Compact-ness	Shape Factor 1	Shape Factor 2	Shape Factor 3	Shape Factor 4
Area	1.000	0.967	0.932	0.952	0.242	0.267	1.000	0.985	0.054	-0.197	-0.358	-0.268	-0.848	-0.639	-0.272	-0.356
Perimeter	0.967	1.000	0.977	0.913	0.385	0.391	0.968	0.991	-0.021	-0.304	-0.548	-0.407	-0.865	-0.768	-0.408	-0.429
MajorAxisLength	0.932	0.977	1.000	0.826	0.550	0.542	0.933	0.962	-0.078	-0.284	-0.596	-0.568	-0.774	-0.859	-0.568	-0.483
MinorAxisLength	0.952	0.913	0.826	1.000	-0.009	0.020	0.951	0.949	0.146	-0.156	-0.210	-0.015	-0.947	-0.471	-0.019	-0.264
AspectRatio	0.242	0.385	0.550	-0.009	1.000	0.924	0.243	0.304	-0.370	-0.268	-0.767	-0.988	0.025	-0.838	-0.979	-0.449
Eccentricity	0.267	0.391	0.542	0.020	0.924	1.000	0.269	0.319	-0.319	-0.298	-0.722	-0.970	0.020	-0.860	-0.981	-0.449
ConvexArea	1.000	0.968	0.933	0.951	0.243	0.269	1.000	0.985	0.053	-0.206	-0.362	-0.270	-0.848	-0.641	-0.274	-0.362
EquivDiameter	0.985	0.991	0.962	0.949	0.304	0.319	0.985	1.000	0.028	-0.232	-0.436	-0.328	-0.893	-0.713	-0.330	-0.393
Extent	0.054	-0.021	-0.078	0.146	-0.370	-0.319	0.053	0.028	1.000	0.191	0.344	0.354	-0.142	0.238	0.348	0.149
Solidity	-0.197	-0.304	-0.284	-0.156	-0.268	-0.298	-0.206	-0.232	0.191	1.000	0.607	0.304	0.153	0.344	0.308	0.702
Roundness	-0.358	-0.548	-0.596	-0.210	-0.767	-0.722	-0.362	-0.436	0.344	0.607	1.000	0.768	0.230	0.783	0.763	0.472
Compactness	-0.268	-0.407	-0.568	-0.015	-0.988	-0.970	-0.270	-0.328	0.354	0.304	0.768	1.000	-0.009	0.869	0.999	0.484
ShapeFactor1	-0.848	-0.865	-0.774	-0.947	0.025	0.020	-0.848	-0.893	-0.142	0.153	0.230	-0.009	1.000	0.469	-0.008	0.249
ShapeFactor2	-0.639	-0.768	-0.859	-0.471	-0.838	-0.860	-0.641	-0.713	0.238	0.344	0.783	0.869	0.469	1.000	0.873	0.530
ShapeFactor3	-0.272	-0.408	-0.568	-0.019	-0.979	-0.981	-0.274	-0.330	0.348	0.308	0.763	0.999	-0.008	0.873	1.000	0.484
ShapeFactor4	-0.356	-0.429	-0.483	-0.264	-0.449	-0.449	-0.362	-0.393	0.149	0.702	0.472	0.484	0.249	0.530	0.484	1.000

Correlation analysis (see table 1) has shown that several of the features are strongly (positively or negatively) correlated. This is due to the fact that basically all of them are kind of geometric measures. In the original work [1] the issue of strong correlation between features has not been addressed. Generally strongly (over 0,9) features bring little extra information, so its elimination should reduce computational complexity (speed up training) with little if any loss in classification accuracy.

It is also sometimes suggested that feature scaling (often called normalization) can improve performance [4], pages 72-73. This is also investigated. To give a brief visualisation of beans dataset, the pair-plot with selected features (less correlated) has been done, see figure 1.

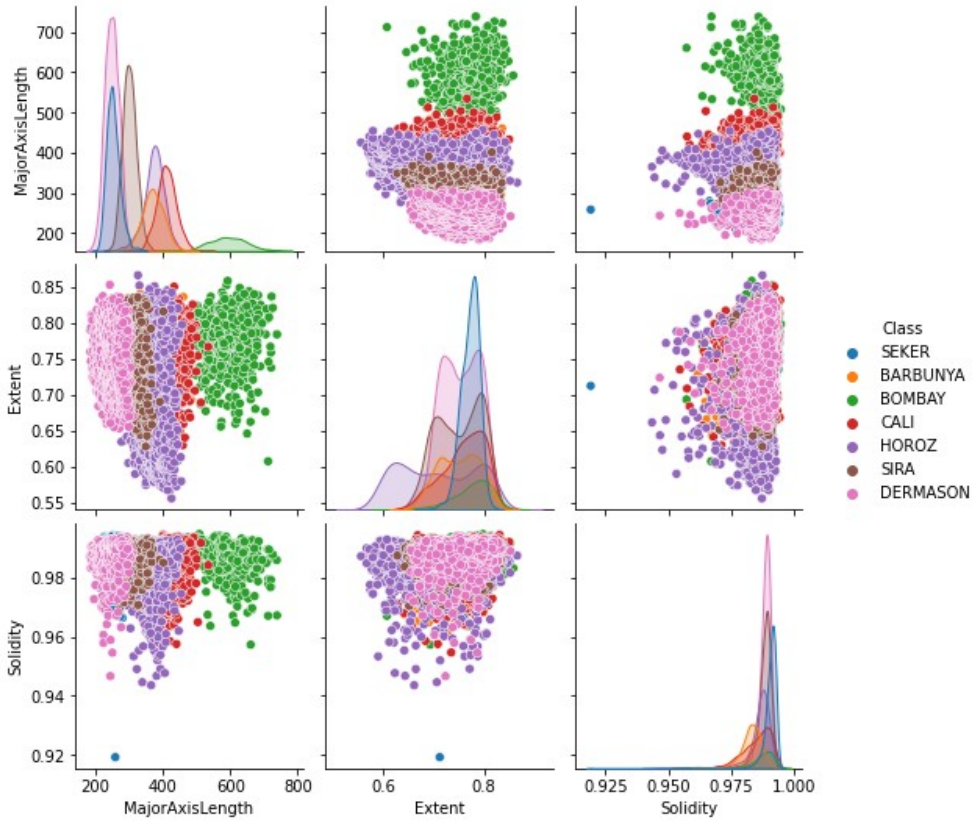

Figure 1: Pair-plot of selected (low corelated) bean features.

4. Shallow learning results

The methods tried were: Naive Bayes Classifier, Decision Tree, Random Forest, Support Vector Classifier.

4.1. Naive Bayes Classifier

Results for Gaussian naive Bayes classifier are shown in table 2. One can see that DDR or FS has small effect on training time. Using DDR or FS (or both) significantly increased accuracy from 77.23% to 89.83-91.00%.

Table 2.
Gaussian naive Bayes classifier performance

Data	Accuracy	Approx. training time
Full, not scaled	77.23%	18.2 ms
Dimension reduced, not scaled	91.00%	16.3 ms
Full, scaled	89.83%	17.0 ms
Dimension reduced, scaled	90.78%	15.8 ms

4.2. Decision tree

Results for decision tree are shown in table 3. Decision tree applied was limited to 16 leaf nodes and maximum depth of 5. One can see that FS has no effect on accuracy and little effect on training time. This probably connected with the fact that DT analyses one feature at the time, so it not cares what is the ratio of specific feature range to other features. DDR shorten training time with limited accuracy decrease.

Table 3.
Decision tree classifier performance

Data	Accuracy	Approx. training time
Full, not scaled	88.87%	128 ms
Dimension reduced, not scaled	88.24%	71 ms
Full, scaled	88.87%	129 ms
Dimension reduced, scaled	88.24%	70 ms

Decision tree is known to be sensitive for data “rotation”, see [4] p 188. DT analyses only one feature at the time. Strongly correlated features gives little extra information, but can present information in a slightly different manner, suitable for decision tree.

4.3. Random Forest Classifier

Results for the random forest (RF) are shown in table 4. The random forest consisted of 150 trees. No limits (max leaves, max depth and etc.) were put on trees. One can observe that training times are longer that for single decision tree (which is reasonable as here we have a set of decision trees). The accuracies are high. DDR shortened training time and allowed for slightly higher accuracy (0,14-0,18 % point). This is quite interesting that although DDR slightly reduced accuracy on single tree it improved accuracy on RF. Similarly to decision tree, SF practically has little effect on training time.

Table 4.
Random forest classifier performance

Data	Accuracy	Approx. training time
Full, not scaled	93.06%	4.69 s
Dimension reduced, not scaled	93.24%	2.69 s
Full, scaled	93.10%	4.79 s
Dimension reduced, scaled	93.24%	2.59 s

4.4. Support Vector Classifier

Results for support vector classifier (SVC) is shown in table 5. Polynomial kernel has been used. Generally SVC is much more “heavier” model than gaussian classifier, decision tree or random forest. Training times much longer. One can see that DDR or FS has small effect on SVC accuracy. DDR on

not scaled features reduced training time. Feature scaling significantly increased training time and increased accuracy a little (about 1% point). The longest training time was observed for DDR and SF data. The training time was 9 times longer than for DDR and not SF data. The author cannot explained this effect.

Table 5.
Support vector classifier performance

Data	Accuracy	Approx. training time
Full, not scaled	91.81%	42 s
Dimension reduced, not scaled	91.81%	29 s
Full, scaled	93.24%	88 s
Dimension reduced, scaled	92.95%	266 s

5. Artificial neural network

For an artificial neural network (ANN) the data needs additional treatment. First, the names of bean species were labelled with numbers and then these numbers 0-6 were codded as so called ”one-hot”. The reason of using ”one-hot” encoding is well explained for example in [5] p. 376 or [6] pp. 190-194.

Three experiments has been performed to analyse: 1) influence of data dimension reduction, 2) influence of features scaling and 3) influence of activation function (sigmoid vs. ReLU). The ANN architecture was kept similar (as much as possible) to described in [1]. All ANNs had 3 hidden layers with 17, 12, 3, neurons respectively. However here ReLU function has be used as “default” option. Output layer consisted of 7 neurons with softmax activation function – one for each class. Generally training lasted for 16 epochs. However, as it was obvious that ANN with sigmoid activation is undertrained, this net was trained for 48 epochs. The training process is presented in figure 2. The performance summary is presented in table 6.

Table 6.
ANNs performance, 17-12-3 architecture, Adam optimiser, 16 epochs

Data	Activation function in hidden layers	Epochs of training	Approx. training time	Accuracy
16 features, scaled	ReLU	16	14 s	92.66%
8 features, scaled	ReLU	16	8 s	93.24%
8 features, not scaled	ReLU	16	9 s	26.74%
8 features, scaled	sigmoid	48	41 s	88.14%

It can be visible that:

1. feature scaling (or data normalisation) is very important for ANN’s. An attempt to train without prior data scaling failed. Only 55,82% accuracy has been obtained. Perhaps bigger network can manage this issue by rescaling data in a few first layers, but it will influence training time and accuracy.
2. ReLU works significantly better than sigmoid function as an activation function. ReLU network trains faster and reaches better accuracy.
3. Data dimension reduction shortens training time nearly by half and increases accuracy by about 0,58 % point.

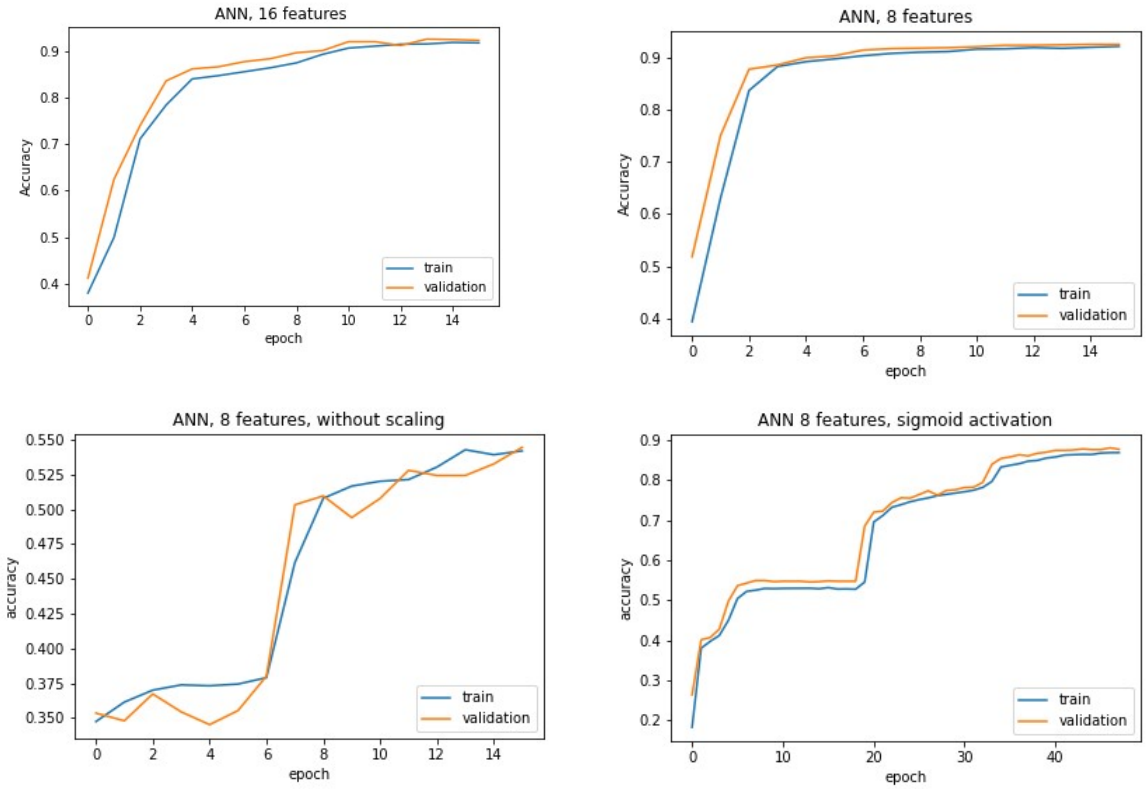

Figure 2: Training of different ANNs

6. Conclusions

Influence of data dimension reduction, data scaling (or normalisation) and activation function has been investigated. The influence depends on machine learning technique used.

Generally data dimension reduction reduces training time with rather limited influence on accuracy Data scaling is a must in case of artificial neural network. Omitting data scaling decreased accuracy from about 93% to about 56%. In case of shallow learning techniques its influence is smaller, it sometimes help a little with accuracy, sometimes not.

Generally scaling had no effect on decision tree and random forest performance. In case of support vector classifier scaling resulted in huge training time increase. Author cannot explain this effect.

The highest accuracy observed was 93,24%. It was obtained 3 times with: 1) random forest with 8 features (scaled and not scaled), 2) ANN, 8 features, scaled and 3) SVC, 16 features, scaled. It is quite intriguing that exactly the same, maximum result repeated 3 time.

7. References

[1] Murat Koklu, Ilker Ali Ozkan, Multiclass classification of dry beans using computer vision and machine learning techniques, Computers and Electronics in Agriculture 174 (2020) 105507

[2] Dry beans dataset at UCI repository: https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset, access 23.06.2021

[3] Colab notebook containing computation scripts for this work: https://colab.research.google.com/drive/1l5lH1QgesDX8CbbkqcnmlbqwcXfksGQB?usp=sharing

[4] Aurelien Geron, Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly, 2019

[5] Jake VanderPlass, Python Data Science Handbook, O'Reilly, 2017

[6] Francois Chollet, Deep Learning with Python, Manning Publications, 2018