FRIteza

Image classification a modern approach

Yannick Kuhar

Artificial Inteligence project

Code and data related to this project are available at https://github.com/yannickKuhar/UIProject

May 2, 2022

prof. dr. Ivan Bratko | Mentor

The age we live in has brought us an abundance of data. Images and videos make up a good part of all globally collected data. Image classification is likely one of the most if not the most important method of digital image analysis. In some tasks machine learning models have already surpassed human accuracy. Therefore, it is a fascinating field to study. In this study we will focus on a presentation of image classification with classical machine learning models. We will give a brief overview of the field and then present three well known models.

Image classification, Machine learning

1. Introduction

Machine learning is a branch of artificial intelligence, its main idea is to improve models automatically based on experience. Most commonly with the use of data. Given advances in multimedia technology image and video data has become vilely available. Tasks such facial retention or recognition of any other object on an image are widely studied as well as vitally important tasks such the classification of medical images. In our work, however we will return to the basics—the classification of handwritten digits, images of clothing and a diverse dataset with one hundred classes.

2. Methods

In this section, we will present the machine learning in general and the methods used in our experiments. Lastly, we will go into more detail regarding our methodology.

Machine learning. In artificial intelligence systems, experience influences future actions. We will focus on machine learning, which is used for analysis and processing of data e.g. sets of images, video games, etc. The result of the learning process can be a set of rules, a function or a probability distribution. Known machine learning algorithms are neural networks, decision trees, support vector machines, etc. Machine learning is divided into regression, classification and clustering, or supervised and unsupervised learning. We will focus on **supervised learning**.

Supervised learning. A supervised learning model uses a training set, which consists of input-output pairs i.e. $\{(x_0, y_0), (x_1, y_1), ..., (x_N, y_N)\}$, where x_i is an input and y_i the output class. The outputs y_i are generated by some unknown function $f(x_i) = y_i$. Supervised learning aims to learn the function h, which approximates f. Function h is called **the hypothesis**. If f is a discrete function then this is a **classification** problem, but if if is a continuous function then this is a **regression** problem. We will focus on **classification**.

The quality of the hypothesis is measured with a test set that is the same shape as the training set, but contains new examples that the model has not yet seen. In classification, we use classification accuracy **CA** by calculating the percentage of correctly predicted elements, so whether y_i is equal to $h(x_i)$.

Support vector machines. Support vectors machines (SVM) are one of the most successful in machine learning [1]. SVM set the optimal hyperplane into the transformed attribute space. Data points closest to the hyperplane are called support vectors. The distance between support vectors and the hyperplane is called the margin. A hyperplane is optimal if the margin is maximal.

In most cases, a linear hyperplane is not sufficient, therefore SVM transform the original space using a nonlinear function into a more complex space, where linear separation is possible. This transformation is performed with a kernel function. They do not have to transform every single data point, only the support vectors.

Due to this property, SVM model larger datasets with numerous attributes well and are resistant to overfitting. Generally, SVM are known to have a high classification accuracy [1].

Random forest. Random forest was originally invented to improve the performance of decision trees [1]. The main idea of this method is to train a sequence of decision trees, usually around one hundred. Each three would then predict new data points, the final prediction is then determined by a majority vote. Each decision tree has one vote.

It's known as a robust method because it reduces the variance of the tree models [1]. Its classification accuracy is generally amongst the highest of all known models.

Naive Bayes classifier. Naive Bayes classifier is based on Bayes' probability theorem:

$$P(r_k|V) = P(r_k) \prod_{i=1}^{a} \frac{P(r_k|v_i)}{P(r_k)},$$

where $P(r_k)$ is the a priori probability class r_k and $P(r_k|v_i)$ is the probability of class r_k dependent on the value of the attribute i, v_i. The learning algorithm approximates the a priori probabilities of each class under the assumption that the attributes are independent [1].

Data. In our experiments we will use the datasets presented in Table 1. All datasets are evenly distributed [2–4].

Dataset name	Number of classes	Dimensionality	Number of examples
MNIST	10	784	70 000
CIFAR100	100	1024	60 000
Fashion-MNIST	10	784	70 000

Table 1. Basic characteristics of the datasets used in our experiments, showing the number of classes, the dimensionality of the data and the number of examples each dataset holds.

MNIST. MNIST is a set of handwritten digits, centered on a (28×28) sized image [2]. Some examples are show on Figure 1.

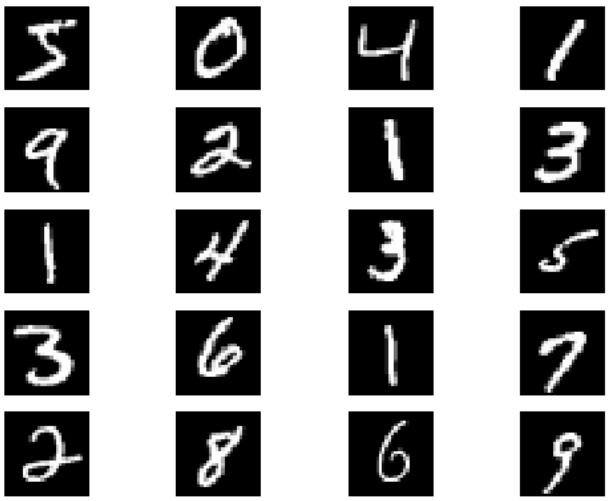

Figure 1. Twenty examples from the MNIST dataset.

FMNIST. Fashion-MNIST is a dataset comprised of images of clothing. The dataset contains $70,000$ images of size (28×28). The classes are shown in Table 2, while some examples are show on Figure 2.

Figure 2. Twenty-five examples from the FMNIST dataset.

Oznaka	Opis
0	T-shirt/Top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandals
6	Shirt
7	Sneaker
8	Bag
9	Ankle boots

Table 2. Classes of the Fashion-MNIST dataset.

CIFAR100. The CIFAR100 dataset had $60,000$ examples of images sized (32×32). The images depict a wide verity of things, such as airplanes, birds, cats, ships, dogs, etc. It has 100 classes, 600 examples each [3].

Methodology. The program, which can be seen on figure 3, has two input parameters:

- **model**: either SVM, Naive Bayes classifier, Random forest or Majority classifier,
- **data**: either MNIST, FMNIST or CIFAR100.

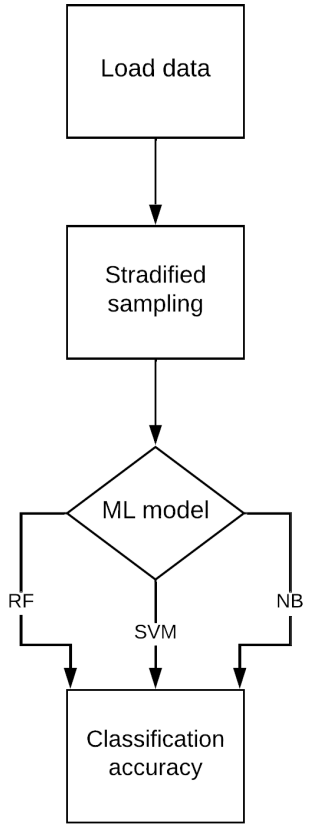

Figure 3. The architecture of our testing framework.

Based on these parameters, the data is loaded and is preprocessed into a matrix. Before we train a machine learning model with our newly constructed dataset, we must split it into a training and a testing set, typically in a 70:30 ratio. Random sampling is a method commonly used to perform this task. When using it, every data point has an equal chance of being selected in the sample. However, random sampling is not as precise as some alternatives and is error-prone [5].

Therefore, we decided to use Stratified sampling in our work. As its first step, Stratified sampling divides the constricted dataset into smaller subgroups, called strata, each sharing a common characteristic. A random sample is then taken from each strata. This ensures each group is represented in the population needed [5]. In our case, this means that the class distributions in the training and testings sets are the same, allowing us to evaluate our models more accurately.

One of the supported machine learning models is then trained and evaluated using classification accuracy.

3. Results

In this section, we will present the results obtained using our methodology, as described in chapter two. We will also give a brief description of the parameters used in all methods.

Parameters. All machine learning models have all parameters set to their defaults according to Sklearn documentation [6].

Accuracy table. All obtained results are presented in Table 3. Random forest was the most successful model, followed by SVM. Naive Bayes trails behind and lastly there is the Majority classifier. Based on Occam's razor we can say that all models are useful as the outperform the simplest possible prediction function.

Model/Data	MNIST	FMNIST	CIFAR100
SVM	0.92	0.83	0.09
Random Forest	0.96	0.87	0.12
Naive Bayes	0.57	0.61	0.07
Majority Classifier	0.11	0.1	0.01

Table 3. The classification accuracies of each model and evaluated on all datasets, rounded up to the second decimal.

Given their characteristics the successes of Random forest and SVM models were expected, Random forest more so as the ensemble approach is known to improve accuracy [1]. Naive Bayes is less successful as due to its naive assumption that all attributes are independent.

4. Discussion

In this project, we implemented a framework to perform image classification tasks. The purpose of this project was to see which method fares the best and to present a general overview of machine learning, an important subfield of artificial intelligence.

Our results show that Random forest performs the best, while Naive Bayes fares the worst. All models however outperform the majority classifier and are according to Occam's razor thus considered useful.

Some interesting upgrades to this project would be to compare additional machine learning algorithms as well as use additional datasets. One subfield of machine learning we havent addressed is deep learning as neural networks are far more complex than the classical models. One factor that greatly influences their performance is topology, for which special selection method are needed or a simple trial and error approach. Therefore, we considered them beyond the scope of this work.

Bibliography

1. Kononenko I, Šikonja MR (2010) *Inteligentni sistemi.* (Založba FE in FRI).
2. LeCun Y, Cortes C (2010) MNIST handwritten digit database. Dosegljivo: http://yann.lecun.com/exdb/mnist/, [Dostopano: 2019-07-19].
3. Krizhevsky A, Nair V, Hinton G (2009) Cifar-100 (Canadian institute for advanced research).
4. Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.
5. Acharya AS, Prakash A, Saxena P, Nigam A (2013) Sampling: Why and how of it. *Indian Journal of Medical Specialties* 4(2):330–333.
6. Pedregosa F et al. (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.