Implementation and evaluation of K-nearest neighbors (KNN) algorithm for handwritten digit recognition.

Project Proposal – DataScience SoSe 2021
Project 5 Group 2

Maximilian Hingerl, Emma Kray,
Nina Gutzeit, Johannes Müller

Project goals

Effektivität von 97,5% bei der digit Erkennung

Eigene Implementierung einer PCA

KNN selbst implementieren

Kreative Idee??

Komplett neuer Datensatz oder Weiterführung der Digit recognition

Digitalisierung von Labordaten/Patientendaten

Code, der neue Daten in richtiges Format bringt

Dataset

- Training set - 60.000 images
- Test set - 10.000 images
- Size-normalized (28*28 pixels)
- Centered
- As .csv
  - One line → one image
  - First column represents label

Logical algorithm flow

Input: Image

Data normalization

Principal component analysis

K nearest neighbours of training set

Classification of test set images

Output: class membership based on Knn

1. Milestone: implementation of data normalization

Check for and resolve errors

- Duplicates?
- NA values? All values between 0-255?
- Correct labels in training dataset?
- Correct image orientation
- Identify outliers

Standardizing, optimizing for kNN

- **Standardization**: Z-Transformation
- or: **Normalization/ Re-scaling:** [0, 1]
- **Noise reduction**: Clipping → binary?

2. Milestone: implementation of PCA

Benefits of PCA

- **Reduces training time** by decreasing the dimensionality
- **Removes noise** by reducing data set to only relevant variables
- **Makes visualization possible** as it reduces multi-dimensional data sets to the PC

Implementation of PCA

- Write on PCA code including steps for scaling the data, calculating the covariance matrix, and the eigenvalues and -vector
- Visualize the PCA

3. Milestone: implement classification algorithm

Delivers : class/label of the test data set

- Should return the class (digit between 0 and 9) of the test data

Planned analysis steps:

- Write KNN-function on our own
- Test performance by optimizing the number of k neighbors

K-nearest neighbors algorithm

Calculate the distance (for example, euclidian or Manhattan Distance) between the test data and the training data

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

$$d_{Manhattan}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Find K-nearest neighbours of the test data

Look up which label the k nearest neighbours have and decide based on that, which label fits best to our test data

Requirement: data needs to be standardized
→ can be achieved through Z-transformation

Image source: Carl Hermann, DataAnalysis3FS_WS2021_Woche2.pdf

4. Milestone: testing the algorithm

kNN is an exception to general workflow for building/testing supervised machine learning models	Instead:	Important
• No model to train • No validation required/possible	• finding a fitting number of k, giving the best result	• Density of our data • Keep the size of the test set small

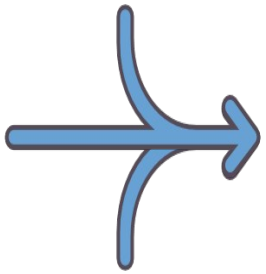
Timeline

12.05.21 - 15.06.21

Emma:
Data Normalization

Maxi & Nina:
PCA

Johannes:
kNN



16.06.21 - 20.06.21

Connecting our components

21.06.21 - 15.07.21

Optimization & performance evaluation

Possible application

Siehe Markdown Ideensammlung

- Nicht fertig
- Fragen offen

Thank you for your attention!