ODSL Block Course - Introduction to Machine Learning

For credits (ECTS), reports need to be handed in by October 31, 2022.
Please email your report as a single pdf to philipp.eller@tum.de.

1 Data transformations

During Tutoring Session

1. Obtain and explore the MNIST and Galaxy datasets.
   (a) The datasets are available in .npz format on the indico page
   (b) Explore the dataset by plotting some sample images, as we will use this also for future exercises.
2. De-correlated and whiten the datasets (interpret each pixel value of the detaset as a dimension)
   (a) Assess the covariance / correlation structure of the dataset (Since large matrices are hard to print out, you can try graphical representations, such as color maps).
   (b) Use one of the techniques discussed in the lecture (Eigen, Cholesky or SVD).
   (c) Assess the covariance / correlation structure of the now de-correlated dataset.

Homework

3. Perform a dimensionality reduction of the Galaxy dataset
   (a) As shown in the lecture, perform a PCA on the dataset and select N first components. (You can try out different values of N, for example. 2, 10, 100, ...)
   (b) Considering the two first principal components, discuss how different types of galaxies end up at different values
   (c) Work out the inverse transform and reconstruct data in the original space from your reduced dataset. How do the reconstructed images compare to the original ones?

2 Clustering

During Tutoring Session

1. Write down the expectation maximization (EM) steps for a simple 1d example with two gaussian as shown in the lecture.
   (a) Draw a few samples from two gaussians of your choice.
   (b) What is the expectation of each point belonging to either of the two given gaussians?
   (c) Given the expectations, what is the optimal choice for two gaussians?
   (d) Starting with two randomly assigned gaussians, can you successfully run the EM steps repeatedly and converge close to your truth?

Homework

2. Clustering:
   (a) Use a k-means implementation as discussed in the lecture to distinguish classes of samples (galaxies) in the galaxy dataset. (This will need to be performed on data that you reduced in dimensionality as in exercise 1.3.)
   (b) Test this for various dimensionalities of the PCA

(c) How good is your accuracy?
(d) Construct confusion matrices

3. Repeat the same for any other clustering algorithm of your choice

4. How do they compare to k-means in term of speed, accuracy and hyper parameters?

3 Optimization

During Tutoring Session

1. Consider a simple function like $f(x) = x^2$.

   (a) Devise an algorithm that uses a bisection strategy to approximate the minimum of $f(x)$ in a given interval $[a, b]$ via its analytic gradients.
   (b) Implement such an algorithm in a computer language, use analytic gradients.

Homework

2. Implement the Nelder-Mead simplex optimizer in n-dimensions.

   (a) Centroid calculation, point reflection, expansion, contraction and shrinking of a n-dimensional simplex need to be defined.
   (b) Implement a convergence criterion. One possibility would be to assess the difference in the values of the objective function of the simplex points, and/or the distance of the points themeselves.
   (c) Test you own implementation of the simplex algorithm on the Rosenbrock test function. Reproduce the 2d example as shown in the lecture.
   (d) Explore higher dimensions

3. Due to the Corona situation, your next physics exam will be online in the form of a multiple choice test with 50 questions and three answers A, B & C each (just one of which is correct in each case). The system allows you to submit infinitely many solutions and immediately reports the number of correct answers. Can you get a perfect score using a genetic algorithm?

   (a) Start with a population of N randomly assigned "individuals"—each individual is a solution to the complete test, i.e. 50 answers.
   (b) Get the score for each individual, and rank them according to their score.
   (c) With a probability proportional to their rank, select individuals for reproduction. (I.e. the best is twice as likely to be selected as the second best and so forth)
   (d) Randomly recombine 2 selected individuals to form a new one for the next generation. Recombination can be done by selecting an answer to each question randomly from the two "parents". Repeat this until you have a new generation of size N.
   (e) Add some mutation to the individuals, i.e. with a low probability change answers randomly.
   (f) Iterate this procedure until you pass the test. How many generations did it take? What did you find to be a good mutation rate and populations size?

4 Decision Trees

During Tutoring Session

1. ROC curve

   (a) To construct artificial data that resembles the output of a classifier, do the following: Assume two unit normal distributions separated by one standard deviation and produce random samples from each. Transform the resulting samples with the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ to map them onto the unit interval.

(b) Construct ROC curves by calculating TPRs and FPRs for many threshold values, one by one by calculating the TP, FP, TN and FN rates each time.
(c) Integrate the area under the curve as a measure for classification performance. Why is this a good metric?

2. Loss function In the lecture we discussed the binary cross-entropy (BCE) loss for classification. Generalize this loss for multi-label classification.

Homework

3. Decision Tree Classifier
   (a) Take the Galaxy dataset, and with PCA reduce its dimensionality to a smaller number (e.g. 10).
   (b) Split the dataset into mutually exclusive training and testing sets.
   (c) Train a decision tree algorithm for classifying smooth and round (class 0) and unbarred spirals (class 3) from the others (classes 1 and 2). Use an existing implementation from a library of your choice (a few were mentioned in the lecture).
4. Hyper parameters
   (a) Repeat with different parameter settings and discuss the results comparing resulting ROC curves, e.g. is there over training?
   (b) How does your best configuration look like and how does it perform?
5. Feature importance
   (a) Assess the importance of your input features to your best performing BDT. (Lecture 6 is needed for this)

5 Neural Nets

During Tutoring Session

1. Try out an auto-diff library (e.g. tensorflow) to automatically calculate derivatives of some test functions.
2. Use this in your simple bisection search from exercise 3.1.

Homework

3. Familiarize yourself with a deep learning framework of your choice (e.g. Keras).
   - Produce the one-hot encoding for the galaxy dataset.
   - Create a simple NN for galaxy classification, e.g. 2-3 dense layers.
   - Use an activation function of your choice (or try different ones) in your hidden layers. Use the softmax activation on your 4d output to make sure the 4 classes always sum up to 1.
   - Train your model with the categorical cross entropy loss.
   - How good does it classify? Can you outperform your BDT from Exercise 4?

6 Regression and AE

During Tutoring Session

1. **ML Challenge** - find an interesting dataset (you can also use the Boston housing dataset as discussed in the lecture), or look around (see e.g. kaggle.com) to perform regression on.
   (a) Train different regressors (linear, BDT, NN) to predict the the value of interest given the inputs
   (b) What happens if you use different loss functions, for example MSE vs. MAE loss? Or if you change other parameter settings?
   (c) What happens if you input values that are far outside of the range of the training set (extrapolation) for different regressors?

Homework

2. Linear auto encoder PCA equivalence

    (a) As in Exercise 1.3, do a dimensionality reduction of the galaxy dataset, but this time using a linear auto encoder.

    (b) Do you get an equivalent solution to PCA? If not, how do they differ? Can you give an explanation why the two should be equivalent?

    (c) How long did the training take compared to the PCA?

    (d) Now add in more layers with activation functions. Does the performance get better?