Deep Learning

project

Stanisław Kaźmierczak, Maciej Żelaszczyk, Adam Żychowski

Faculty of Mathematics and Information Science,
Warsaw University of Technology

2021

Overview

1. Assessment
2. Project I - multilayer perceptron
3. Project II - convolutional neural networks
4. Project III - recurrent neural networks
5. Project IV - generative adversarial networks

Overview

- **4 projects**
- for each project: max 25 points, 4 weeks
- groups of 2 people
- first week of delay: -5 points; second week of delay: additional -10 points (-15 points in total); exceeding 2 weeks of delay results in 0 points for the project
- grades: [0;50] $\rightarrow$ 2, [51;60] $\rightarrow$ 3, [61;70] $\rightarrow$ 3.5, [71;80] $\rightarrow$ 4, [81;90] $\rightarrow$ 4.5, [91;100] $\rightarrow$ 5,

Timetable

1.	23.02	**Introduction**
2.	02.03	Tutorial hours
3.	09.03	Initial presentation of the first part
4.	16.03	**Project I (first part) deadline**
5.	23.03	**Project I (second part) deadline**
6.	30.03	Verification of project plan
7.	13.04	Initial presentation of the second project
8.	20.04	**Project II deadline**
9.	27.04	Verification of project plan
10.	11.05	Initial presentation of the third project
11.	18.05	**Project III deadline**
12.	25.05	Verification of project plan
13.	01.06	Consultation
14.	08.06	Initial presentation of the fourth project
15.	15.06	**Project IV deadline**

Rules

Project I - multilayer perceptron

Topic: **Multilayer perceptron (MLP) employing backpropagation algorithm**

Dataset for the first part of the project will be provided during the first class.

Dataset for the second part: MNIST
`https://www.kaggle.com/c/digit-recognizer`

Project I - multilayer perceptron

- low-level implementation is required; you can use only some basic packages like NumPy, pandas, etc.
- MLP parameters are:
  - number of hidden layers and number of neurons in hidden layers
  - activation function
  - bias presence
  - batch size
  - number of iterations
  - learning rate
  - momentum
  - problem type: classification or regression
- during the project presentation, you will be asked to train and test your network on new (unseen before) datasets
- take care of reproducibility by initializing a random number generator with a constant seed

Project I - multilayer perceptron

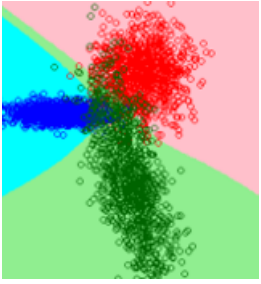
- elements to analyze:
  - how does activation function affect the model's accuracy? Experiment with sigmoid and two other activation functions. The activation function in an output layer should be chosen accordingly to the problem;
  - how does the number of hidden layers and number of neurons in hidden layers impact the model's accuracy? Analyze different architectures;
  - how does the loss function affect the model's accuracy? Consider two different loss functions for both classification and regression.
- the application should plot training and test error
- user should be able to track learning process iteration by iteration (visualization of edges' weights) as well as a propagated error (visualization of an error on each edge)
- visualization of a training set and classification/regression result (as a background)

Assessment: source code, report

Project I - multilayer perceptron

	A	B	C	D
1	x,y,cls			
2	-0.00292545510455966,0.722530109807849,1			
3	0.504570618271828,-0.789261351339519,2			
4	0.894131738692522,-0.720432524569333,1			
5	0.220206981524825,0.12407802278176,2			
6	0.267815329600126,0.925336269196123,1			
7	-0.27340721199289,0.479693677276373,1			
8	-0.287346473895013,-0.972735119983554,2			
9	0.467718373052776,0.81061793724075,1			
10	0.993102677632123,-0.181434742640704,1			

	A	B
1	x,y	
2	-5,-1253	
3	-4.99,-1247.368296	
4	-4.98,-1241.753168	
5	-4.97,-1236.154592	
6	-4.96,-1230.572544	
7	-4.95,-1225.007	
8	-4.94,-1219.457936	
9	-4.93,-1213.925328	
10	-4.92,-1208.409152	
11	-4.91,-1202.909384	
12	-4.9,-1197.426	
13	-4.89,-1191.958976	
14	-4.88,-1186.508288	
15	-4.87,-1181.073912	



Project I - multilayer perceptron

- The second part of the project consists in fitting the implemented model to the deep learning *Hello world* dataset - MNIST
- The achieved score should be registered in Kaggle platform

Assessment: source code, extended report (or supplement to the report), presentation

Project I - multilayer perceptron

Useful resources:

- https://www.deeplearningbook.org
- http://neuralnetworksanddeeplearning.com/
- https://www.coursera.org/specializations/deep-learning
- https://drive.google.com/drive/folders/0B5DSlxnH–fzR1hxd3VYOUhua2c

Project II - convolutional neural networks

Topic: **Image classification with convolutional neural networks**

Dataset: CIFAR-10
`https://www.kaggle.com/c/cifar-10/`

Project II - convolutional neural networks

- you can utilize code from external sources (books, articles, blogs) provided that:
  - reference is cited in the report
  - some modifications to the original solution are applied
- violation of the above or any other kind of plagiarism result in a failing grade
- application of pre-trained models (AlexNet, VGG, etc.) is permitted (and even recommended as a part of the experiment)
- full discretion in terms of packages use
- crucial elements taken into account while assessment:
  - report quality
  - achieved accuracy registered on Kaggle
  - the originality of the solution

Project II - convolutional neural networks

- Conspectus of the project should include:
  - problem description
  - goals of research
  - planned methods along with references
  - data description
- take care of reproducibility by initializing a random number generator with a constant seed.
- to obtain statistically significant results, each experiment should be repeated multiple times.

Assessment: source code, report, presentation

Project II - convolutional neural networks

Useful resources:

- Google Colab or GPU to speed up training
- ensembling (soft voting or majority/hard voting)
- data augmentation
- https://benchmarks.ai/cifar-10
- https://www.robots.ox.ac.uk/ vgg/practicals/cnn/index.html
- https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/
- https://medium.com/kaggle-blog/profiling-top-kagglers-bestfitting-currently-1-in-the-world-58cc0e187b

Project III - recurrent neural networks

Topic: **Speech commands classification with recurrent neural networks**

Dataset: Speech Commands Dataset
`https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data`

Project III - recurrent neural networks

- test and compare different network architectures (at least one of them should be Long short-term memory (LSTM))
- investigate influence of parameters change on the obtained results
- present confusion matrix (with appropriate discussion)
- in case of accuracy or efficiency problem a subset of classes can be selected and tested (e.g. only “yes” and “no” commands)
- please pay special attention on “silence” and “unknown” classes - test different approaches (e.g. separate network for their recognition)

Assessment: source code, report, presentation

Project III - recurrent neural networks

Useful resources:

- https://www.kaggle.com/davids1992/speech-representation-and-data-exploration
- https://towardsdatascience.com/recognizing-speech-commands-using-recurrent-neural-networks-with-attention-c2b2ba17c837
- https://www.coursera.org/lecture/nlp-sequence-models/recurrent-neural-network-model- ftkzt
- https://pathmind.com/wiki/lstm

Project IV - generative adversarial networks

Topic: **Image generation with generative adversarial networks**

Dataset: 10% sample of the LSUN Bedrooms Dataset
`https://www.kaggle.com/jhoward/lsun_bedroom`

Your only need the data from the *sample* directory ($\sim$ 4.7GB in total).

Project IV - generative adversarial networks

- test and compare different network architectures (at least one of them should converge to generate satisfactory images)
- potential architectures (can use different ones): vanilla GAN with CNNs, DC-GAN, Wasserstein GAN, WGAN-GP, SN-GAN, StyleGAN1, StyleGAN2 (with or without data augmentations)
- calculate the Fréchet Inception Distance (FID) for your generated images and compare it to results from literature
- assess your results qualitatively
- investigate the influence of hyperparameters on obtained results

Project IV - generative adversarial networks

- discuss sets of hyperparameters which help in overcoming training collapse and mode collapse
- select two of your generated images together with their latent vectors; interpolate linearly between the two latent vectors to generate 8 additional latent vectors; use these 8 vectors to generate images from your model; present the 10 generated images (8 newly generated and 2 generated previously) and discuss the importance of the results
- discuss any additional findings
- no required language or libraries; suggested language: Python; suggested libraries: PyTorch or TensorFlow

Assessment: source code, report, presentation

Project IV - generative adversarial networks

Useful resources:

- Original GAN paper: https://arxiv.org/pdf/1406.2661.pdf
- DC-GAN paper: https://arxiv.org/pdf/1511.06434.pdf
- StyleGAN2 with data augmentations: https://arxiv.org/pdf/2006.06676.pdf
- FastAI on GANs (skip to relevant part): https://course18.fast.ai/lessons/lesson12.html
- Machine Learning Mastery (general overview): https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
- DeepLearning.AI (for those interested; ability to access content for free): https://deeplearning.ai/program/generative-adversarial-networks-gans-specialization/

The End