Introduction, PCA, Assignment 1-2

(Neural Networks Implementation and Application Tutorial)

Tsimafei Prakapenka, Nicholas Gareth Jennigs

16th November 2022

Overview

- Introduction
- Requirements, Materials, Assignments
- Previous assignment
- Linear Algebra, PCA
- Current assignment
- QA

Hello

Who am I?

Hello

Who am I?

Who are you?

🤔

Introduction

Choose and answer at least two questions:

- On scale from 1-10 how proficient are you in programming, mathematics, and neural networks?
- What topics of Neural Networks excite you the most?
- What topics of Neural Networks excite you the least?
- What programming languages do you know?

Also the following:

- Who is your groupmate?
- What are you expecting from tutorials?
- How best can the tutorial sessions be helpful to your needs?

Requirements

Tutorial Requirements (exam admission)

- 60% of mandatory points ($\sim$10 assignments, 10 points each)
- Tutorial points only for exam admission (no final grade influence)

Tutorial Bonus Points

- $\sim$2pts for extra exercises in the assignments (if available)
- up to 2pt for answering a question in a tutorial + presenting your HW solution

Final Project

- None

Transfer from last year

- Yes
- Assignments recommended (because of the exam)

What's available

- Lectures by Prof. Klakow (onsite)
- Tutorials (online with Tsimafei and onsite with Nicholas)
- Corrected homework
- Consultations
  - Only in specific cases
  - By default **no** email and **no** personal chat
  - Ask questions during the lecture / tutorials
- Public forum (please use Piazza)
  - Ask questions
  - Other students will also benefit from the answers
  - You can answer someone else's issue

Assignments

- Mandatory groups of 2 or 3
- Jupyter notebook templates
  - Assignment + solution in the same notebook
  - Use Google Colab or local runtime
  - Write solutions in Python files and import them
  - Submitted notebook must only contain your analysis and outputs
- Only one submission per group
  - Submit through Teams

Dates / Times

- Lecture:
  - Tuesday 14:15-15:45
- Tutorials:
  - Tsimafei: Wednesday 16:15-17:45
  - Noon: Thursday 14:15-15:45
- Assignments
  - Released (usually) by Wednesday 08:00 (available in Teams)
  - Deadline (next) by Wednesday 08:00 (submit in Teams)
- Exam: TBD

Tutorial Content

- Review of the topics covered in class + live coding sessions
- Presentation of the past assignment
- Discussing the current assignment

Organization

Questions?

Assignment 1

- Does anybody want to present solution?

Assignment 1

- Does anybody want to present solution?
- How long does it take? Any other feedback?

Linear Algebra Basics

Few definitions (+how are they implemented in Python/Numpy/PyTorch)

- Scalars
- Vectors
- Matrices
- Tensors

Linear Algebra Basics

Few definitions (+how are they implemented in Python/Numpy/PyTorch)

- Scalars
- Vectors
- Matrices
- Tensors

Identify the following objects (Python lists):

- `[5.0, 3.0]`
- `5.0`
- `[True]`
- `[[5, 1], [0, 4]]`
- `[[True, False], [False, True]]`
- `[ [[0,1], [0,1], [0,1]], [[0,1], [0,1], [0,1]] ]`

Linear Algebra Basics

A few operations and properties involving matrices:

- Transpose
- Inverse
- Dot product (i.e. matrix multiplication)
  - $C = AB$, $C_{i,j} = \sum_k A_{i,k} B_{k,j}$

Common Properties:

- $A(B + C) = AB + AC$
- $A(BC) = (AB)C$
- $AB \neq BA$
- $(AB)^T = B^T A^T$

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)
  - $A = U \Sigma V^T$

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)
  - $A = U\Sigma V^T$
- Principal Component Analysis:

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)
  - $A = U \Sigma V^T$
- Principal Component Analysis:
  - Eigendecomposition or SVD of covariance matrix $W = \frac{A^T A}{n-1}$

Linear Algebra Basics

Definitions:

- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)
  - $A = U \Sigma V^T$
- Principal Component Analysis:
  - Eigendecomposition or SVD of covariance matrix $W = \frac{A^T A}{n-1}$
  - Assume ordering of {eigen,singular}values from highest to lowest

Linear Algebra Basics

Definitions:

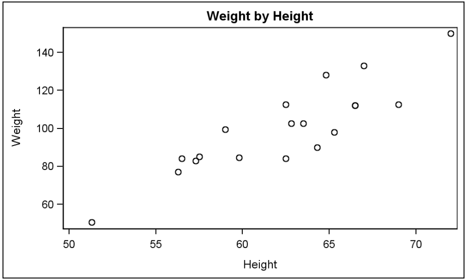
- Eigenvector, Eigenvalue
  - $Av = \lambda v, v \neq \overrightarrow{0}$
- Eigendecomposition
  - $A = Q \cdot L \cdot Q^{-1}$
- Singular value decomposition (SVD)
  - $A = U \Sigma V^T$
- Principal Component Analysis:
  - Eigendecomposition or SVD of covariance matrix $W = \frac{A^T A}{n-1}$
  - Assume ordering of $\{$eigen,singular$\}$values from highest to lowest
  - Project to k dimensions: $A_k = AQ_k$

Linear Algebra Basics - True or False? 🧐

$$A = \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$$

1. Is $v_1 = (1, -1)$ an eigenvector of A?
2. Is $v_2 = (2, 1)$ an eigenvector of A?
3. Is $v_3 = (2, 2)$ an eigenvector of A?

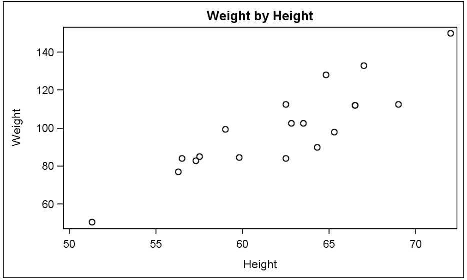
PCA



Questions 🧐

- What will be the first principal component?

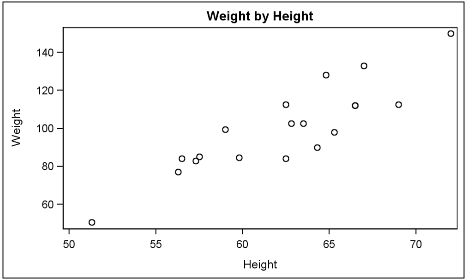
PCA



Questions 🧐

- What will be the first principal component?
- Does anyone know how PCA works?

PCA



Questions 🧐

- What will be the first principal component?
- Does anyone know how PCA works?
- What does it mean that we take only k largest principal components?

Standardization

- Is not normalization! ($x' = \frac{x}{|x|}$)
- $X = \frac{X - \text{mean}(X)}{\text{std}(X)}$
- Compute either:
  - With Numpy: `X = (X-X.mean())/np.std(X)`
  - With Scikit: `StandardScaler().fit_transform(X)`
- Why do we need standardization for PCA? 🤔

Assignment 2

- Any questions?

Resources

1. Course Website: lsv.uni-saarland.de/neural-networks-implementation-and-application-winter-2021-2022-2
2. Piazza: https://piazza.com/class/l9so16qqvk34hu
3. Tutorial repository https://github.com/tsimafeip/uds-nnia-tutorial-2223 (adapted from last year edition by Vilém Zouhar and Noon Pokaratsiri Goldstein)
4. Lecture & tutorial teams channels