UiT

THE ARCTIC UNIVERSITY OF NORWAY

FYS-2021 - Exercise Set 1

Vectorising Code

Department of Physics and Technology
Faculty of Science and Technology

Problem 1

The recommended programming language in this course is Python. The exercise group teachers will also be able to help with Matlab. Students can also choose other programming languages, but in that case it could be that the exercise group teachers are not able to provide detailed coding help.

(1a) For students who choose to use Python, but do not have much experience with the numerical library NumPy (or want to refresh their knowledge), it is recommended to do a few of the "warm-up" exercises that you can find here:

https://www.w3resource.com/python-exercises/numpy/basic/index.php

https://www.w3resource.com/python-exercises/numpy/index-array.php

https://www.w3resource.com/python-exercises/numpy/linear-algebra/index.php

https://www.w3resource.com/python-exercises/numpy/python-numpy-stat.php

Students who are familiar with Numpy and students who choose to use other programming languages can move on to the exercises below.

This course assumes you have a fair knowledge of how numerical programming, (i.e. working with vectors and matrices programatically), works. In the following exercises, we will look at writing efficient, vectorised code. Assume that we have vectors

$$\boldsymbol{a} \in \mathbb{R}^n \text{ with elements } \boldsymbol{a} = [a_1, \dots, a_n]^T$$

$$\boldsymbol{b} \in \mathbb{R}^n \text{ with elements } \boldsymbol{b} = [b_1, \dots, b_n]^T$$

Further assume we have a matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ & & \vdots & \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix}$$

The result vector is $\boldsymbol{y}$ which takes its dimensionality based on the problem.

In all problems you should assume that the dimensions of the vectors and matrices involved are such that the matrix multiplications (and inner products) are valid. Note that e.g. `y+=k` is the same as `y=y+k`.

(1b) Write this sum as a mathematical vector operation

$$y = \sum_{i=1}^{n} a_i b_i$$

(1c) Write this for loop as a mathematical vector operation and as vectorised code.

```
for i=1,...,n:
  y+=a_i b_i
```

(1d) Write this for loop as a mathematical vector operation and vectorised code (assume $N = n$)

```
for i=1,...,M:
  for k=1,...,n:
    y_i+=x_ik a_k
```

(1e) *Vectorise* the following algorithm (answer with code). This shouldn't be done with traditional matrix multiplications *alone*:

```
for i=1,...,M:
  for k=1,...,n:
    y+=x_ik a_k
```

(1f) Let $Z \in \mathbb{R}^{P\times M}$ and $z_{ij} \in Z$ be the entry in row i and column j.

Vectorise the following algorithm

```
for j=1,...,N:
  for i=1,...,P:
    for k=1,...,M:
      y_i+=z_ik x_kj
```

(1g) NB:*This part problem has different notation!*
Suppose you have a dataset with N samples and n features:

$$\boldsymbol{x}_1, \dots, \boldsymbol{x}_N, \boldsymbol{x}_i = [x_{i1}, \dots, x_{in}] \in \mathbb{R}^n$$

Similarly, you have parameter vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_P$, $\boldsymbol{w} \in \mathbb{R}^n$ and $\boldsymbol{a} \in \mathbb{R}^P$.

Implement an algorithm, *for every single sample* $\boldsymbol{x}_j = [x_{j1}, \dots, x_{jn}]$ in the dataset, that calculates:

$$v_i = \sum_{k=1}^{n} w_{ik} x_{jk}$$

$$y_j = \sum_{i=1}^{P} v_i a_i$$

Vectorise these expressions. You want to end up with a vector $\boldsymbol{y} \in \mathbb{R}^N$

Because of the potential lack of relevance, the next problems are considered 'bonus' exercises.

(1h) Implement an algorithm that does matrix multiplication (without using built-in matrix multiplication functions).

(1i) Show that for any matrix $\boldsymbol{A}$ the product $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric.

(1j) Implement a algorithm that calculates $\boldsymbol{A}^T\boldsymbol{A}$ using the property of symmetry. Generate a huge matrix $\boldsymbol{A}$ and compare the running time between the implementation you did in (1h) and the built in matrix multiplication function.