

## Matrices in NumPy

TCTI-VKAAI-17: Applied Artificial Intelligence

Huib Aldewereld

## Inhoudsopgave

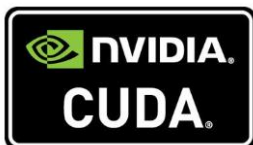
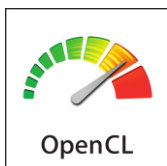


- NumPy
  - Vectoren
  - Matrices
  - Rijen en kolommen
- Broadcasting
- Recap

## NumPy



- Lineaire algebra wegens GPU acceleratie
- OpenCL of CUDA voor bare-metal performance
- NumPy als middle ground
  - Leesbaarheid en performance



3

## Vectoren



- Numpy arrays
 

```
a = numpy.array([1, 2, 3])
```
- Idempotent
 

```
a == numpy.array(a)
```
- Optellen, scalair vermenigvuldigen met gebruikelijke operatoren
- Inner product  $\langle \mathbf{u} | \mathbf{v} \rangle$  m.b.v `numpy.dot(u, v)` (niet `u*v`!)

`u.dot(v)` werkt ook!

4

## Matrices



- Meer NumPy arrays!

```
M = numpy.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

- Optellen, scalair vermenigvuldigen als vectoren
- Matrix product ook m.b.v. `numpy.dot(M, N)` (niet  $M*N$ !)

5

## Rijen en kolommen



- Vectoren standaard richtingloos
- Standaard-interpretatie: rijen

- Expliciet kenbaar maken:

<code>numpy.array([1, 2, 3])</code>	# Vector
<code>numpy.array([[1, 2, 3]])</code>	# Rij-vector
<code>numpy.array([[1], [2], [3]])</code>	# Kolom-vector

6



## Inhoudsopgave

- NumPy
  - Vectoren
  - Matrices
  - Rijen en kolommen
- Broadcasting
- Recap

7



## Broadcasting

- Voor veel operaties eisen aan dimensies vectoren/matrices
- NumPy extrapoleert:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{bmatrix}$$

- Voorwaarde: dimensies die extrapoleren moeten 1 zijn:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \text{⚡}$$

8



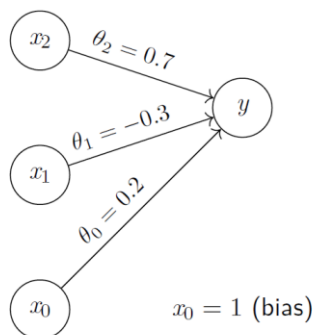
## Inhoudsopgave

- NumPy
  - Vectoren
  - Matrices
  - Rijen en kolommen
- Broadcasting
- Recap

9



## Recap: ons eerste netwerkje



$$\mathbf{x} = \begin{bmatrix} 1 \\ 0.2 \\ 0.3 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} 0.2 \\ -0.3 \\ 0.7 \end{bmatrix}$$

$$y = \sigma(\langle \boldsymbol{\theta} | \mathbf{x} \rangle)$$

```
def sigmoid(x):
    return 1 / (1 + (math.e ** (-x)))
```

```
x = numpy.array([1, 0.2, 0.3]) # Bias and two input values
theta = numpy.array([0.2, -0.3, 0.7])
y = sigmoid(numpy.dot(theta, x)) # 0.5866175789173301
```

10