01/07/2021

Dive Into deep Learning.

The required skills needed before jumping into machine learning:

- Working with large datasets:
  - Linear Algebra.
- Optimization
  - Calculus
- Making prediction
  -

* Preliminaries

- Data Manipulation
- Data Preprocessing
- Linear Algebra
- Calculus
- Automatic Differentiation
- Probability
- Documentation.

Data Manipulation

- i) Acquire data
- ii) Process them once they are inside the computer.

n-dimensional array ~ Tensor

~~Import~~

import torch

- tensor - multidimensional array of numerical values
  - ▷ With one ~~vector~~ ~~tensor~~ axis - Corresponds in maths to a vector
  - ▷ With two axis - Corresponds to a matrix

Page 46

Concatenate = link things

Tips: it replicates the row & colums to match up.

Broadcasting Mechanism

- if we have matrix 3x1 & 1x2
- We broadcast it to 3x2

eg:

tensor [0]
[1]
[2]

tensor [0, 1]

for a + b

tensor ([[0, 1]
[1, 2]
[2, 3]]).

Indexing & Slicing.

2.2 Data Preprocessing.

- We need to write & load data

- Import Pandas Package to do Data Preprocessing.

- Handling Missing data
  - "NaN" = not a number.
  - Method - Imputation & deletion.

Imputation = Replaces missing values with substituted ones.
Deletion = Ignores missing values.

* Learn to create & Read .csv file later.

4/7/21

2.3 Linear Algebra.

Scalars

- $\in$ = "in" denotes membership in a set.

Vectors

eg - training a model to predict the risk that a loan defaults.

- associate each applicants with a vector corresponds
  - ~ Income
  - ~ length of employement
  - ~ number of previous defaults.
  - ~ Other factors.

Matrices

Matrices with ~~bt~~ bold - faced, capital letters

eg $(X, Y \text{ \& } Z)$ ~ Code tensors with two axes.

$A \in R^{m \times n}$

- shape of A is $(m \times n)$ or $m \times n$

Tensors.

Tensor refer to algebraic objects

- Vectors = first order tensors
- Matrices = second order tensors.

Reduction.

Σ = sum, in code we call the function ( )

eg

input

```
x = torch.arange (4, dtype = torch.float32)
x, x.sum ( )
```

Output

(tensor([0, 1, 2, 3.]), tensor (6.))

- We can specify axes along the tensor to reduced via ~~at~~ Σ.

eg

To reduce the row dimension (~~axis = 0~~ axis 0) by summing up elements of all the rows, we specify axis = 0.

Reducing matrix along raw & colums via Σ

Σ all the element of the matrix.

Dot product

eg y = torch.ones (4, dtype = torch.float32)
x, y, torch.dot (x, y)

output (tensor ([0., 1., 2., 3.]), tensor ([1., 1., 1., 1.]), tensor (6.))

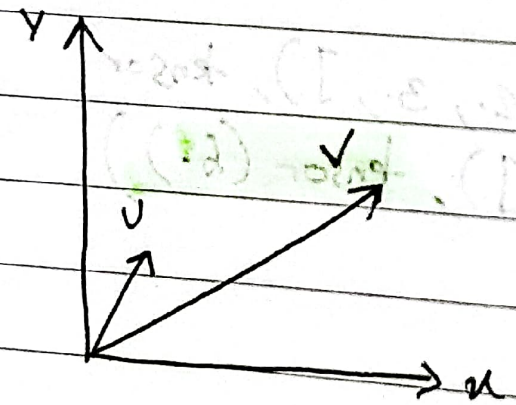
Matrix - Vector Product

Norms :

- Tells how big a vector.
- It focus on magnitude of components.
- A vector norm is a function f that maps a vector to a scalar.

based on youtube

Vector Norms :



Find how $v > u$

L_1 Norm ; $\|u\|_1 = |u_1| + |u_2|$

L_2 Norm ; $\|u\|_2 = \sqrt{u_1^2 + u_2^2}$

Property :

$$\|AB\|_p \leq \|A\|_p \|B\|_p$$

Why? Real world Implementation

$$\|A^k\|_p = \max_{\|U\|_p = 1} \|A^k U\|_p$$

Matrix is a Linear Transformation

Let λ_i be EigVal of A
with EigVec U_i

$$A \times U_1 = \lambda_i \times U_1$$

$$A \begin{bmatrix} U_i & \cdots & U_i \end{bmatrix} = \lambda_i \begin{bmatrix} U_i & \cdots & U_i \end{bmatrix} \Rightarrow AU_i = \lambda_i U_i$$

$$\|AU_1\|_p = |\lambda_i| \|U_i\|_p \leq \|A\|_p \|U_i\|_p \Rightarrow$$

absolute

$$\cancel{\|A_p\|} \; \|A\|_p \geq \|\lambda_i\|$$

$$\|A\|_p \geq |\lambda_i| \; \forall$$

$$\|A\|_p \geq \max |\lambda_i|$$

$$\|A^k\|_p \geq \max |\lambda_i|^k$$

Based on Deep Learning Book

$$f(\alpha x) = \underbrace{|\alpha|}_{\text{absolute value.}} f(x)$$

$$f(x+y) \leq f(x) + f(y)$$

$$f(x) \geq 0$$