Introduction to Tensor Algebra

basics

Vectors

- Ordered sequence of two or more numbers
- Geometrically, some position in n-dimensional space
- Coordinate system does not matter in ML as long as it is consistent

vectorisation

data → numbers representing point in vector space → fed into model

- Vectors in ML signify input data ∋ {bias, weight}
- Automates synonyms, clustering documents

vector search

Matrix

- Collection of vectors
- Rectangular array w/ order rows x columns

Tensors

- Generalised matrices
- dim (tensor) = rank
- matrices, vectors, scalars ∈ tensors
  (dim 1) (dim 0)
- dynamic: transform in specific ways when interacting with other mathematical entities } not all tensors are matrices: not all matrices fulfill dynamism; do not transform according to other entities

unknown shape, rank

tensors in ML

- dot product, cross product
  - neural net
    - comp. weighted sum
  - grad descent
    - grad of loss fn
    - step direction, magnitude in the parameter update
  - → normalisation, regularisation

recommender

deep learning

index placement is important

index placement is important

index placement is important

v^i: contravariant

w_i: dual / covariant

operations

trace(matrix)
sum of diagonal elements

orthogonality
orthogonal vectors ⇒ dot product zero
orthogonal matrix ⇒
- $A^T = A^{-1}$
- $\det(A) = 1/-1$
- rows, columns are orthonormal

multiplication by scalar
$(\lambda+\psi)C = \lambda C + \psi C$

Hadamard multiplication
matrices of same dimension
↓
element-wise multiplication
$A_{m\times n} \odot B_{m\times n}$

inverse
$AA^{-1} = A^{-1}A = I$
$(AB)^{-1} = B^{-1}A^{-1}$
$(AB)^T = B^TA^T$

For rectangular:
$(A^TA)^{-1}A^T$ → wide (full row rank)
$A^T(AA^T)^{-1}$ → tall (full col rank)
} Moore-Penrose pseudoinverse

eigenvalues
$tr(A)$ = sum of eigenvalues
$\det(A)$ = prod " "

singular
$\det(A) = 0$
features linearly dependent → rows & columns

non-singular
$\det(A) \neq 0$
features linearly independent

Outer product: matrix multiplication

- Result rank = sum of ranks of input tensor
- Used extensively in backpropagation [weight update] } Efficient calculation

weight × gradient

weight update $\Delta w_{ij} \propto$ outer prod of 2 vectors

Matrix inner product: dot product

multiply corr. elements → sum products → scalar | $a \cdot b = \sum a_i b_i$

NOTE: inner prod on matrices tells us how similar they are

- Weighted sum
- Cosine similarity = $\frac{a \cdot b}{|a|\,|b|}$
- Projection → PCA
- Attention in transformers

Hadamard product:

Element-wise multiplication of matrices w/ same order

- Broadcasting

Kronecker vs Tensor Outer Product

→ matrix specific concept of outer product

Hadamard vs Kronecker } → create large matrix blocks $A \otimes B$

Hadamard ↓ same dim; Kronecker ↓ different dim

Vector cross product

forming vector normal to plane formed by two others

traditional cross product → 3D