Linear Algebra - Lecture Notes

Deep Learning

Why Linear Algebra is interesting for Machine Learning? an example:

A good understanding of linear algebra is essential for understanding and working with many machine learning algorithms, especially deep learning algorithms. One of the simpler machine learning model is the linear classifier that classify a point based on it's "position" respect to an hyperplane.
Let's consider and hyperplane in $\mathbb{R}^n$. An hyper plane is composed of the set of points that satisfy a linear equation e.g.

$$w_1x_1 + w_2x_2 + \cdots + w_nx_n = b$$

- A point in an multidimensional space is represented by a vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
- we can represent the hyperplane with the vector of its weights $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

  and the bias b
- $\vec{w}$ is itself a vector, orthogonal to the hyperplane
- in $\mathbb{R}^2$ hyperplane is just a line

The hyperplane can be define as:

$$\vec{w} \cdot \vec{x} = b$$

Matrix

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix}$$

Matrix Transpose:

$(\mathbf{A}^T)_{i,j} = A_{j,i}$. $\mathbf{A}^T = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$

Symmetric Matrix

$\mathbf{A} = \mathbf{A}^T$

Diagonal Matrix

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & 0 & 0 & \dots & 0 \\ 0 & A_{2,2} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A_{n,n} \end{bmatrix}$$

Identity Matrix:

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Matrix (Dot) Product

$\mathbf{C} = \mathbf{AB}$
$C_{i,j} = \sum_k A_{i,k} \cdot B_{k,j}$, where $\mathbf{A} \in \mathcal{R}^{m \times n}, \mathbf{B} \in \mathcal{R}^{n \times p}$

Matrix - Vector

$$\mathbf{A} \cdot \vec{b} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} A_{1,1}b_1 + A_{1,2}b_2 + A_{1,3}b_3 \\ A_{2,1}b_1 + A_{2,2}b_2 + A_{2,3}b_3 \end{bmatrix}$$

Matrix - Matrix

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix} \cdot \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix} =$$
$$= \begin{bmatrix} A_{1,1}B_{1,1} + A_{1,2}B_{2,1} + A_{1,3}B_{3,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} + A_{1,3}B_{3,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} + A_{2,3}B_{3,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} + A_{2,3}B_{3,2} \end{bmatrix}$$

Power of a Matrix

$$\mathbf{A}^k = \mathbf{A} \cdot \mathbf{A} \cdot \ \ldots \ \cdot \mathbf{A}$$

Properties of Matrix Product

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

Hadamard product (Element-wise product)

$$\begin{aligned}\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \odot \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \\ = \begin{bmatrix} A_{1,1} \cdot B_{1,1} & A_{1,2} \cdot B_{1,2} \\ A_{2,1} \cdot B_{2,1} & A_{2,2} \cdot B_{2,2} \end{bmatrix}, \text{ where } \mathbf{A}, \mathbf{B} \in \mathbb{R}^{n,m}\end{aligned}$$

Dot product (vectors)

$< x, y >= x^T y$

Trace Operator

The trace operator gives the sum of all the diagonal entries of a matrix

$$Tr(\mathbf{A}) = \sum_i A_{i,i}$$

Matrix Inversion:

$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n$
$\mathbf{A}^{-1}$ do not always exists:

- **A** must be squared (but not all squared matrix are invertible);
  - if **A** is not squared it is possible to use **Moore-Penrose pseudoinverse**;
- its columns must be independent.

Tensors:
A tensor is an array of numbers that may have 0 or more more dimensions (more than 2).

- 0 dimensions, and be a scalar;
- 1 dimension, and be a vector;
- 2 dimensions, and be a matrix.
- ...

System of Linear Equations

$$\mathbf{A}\vec{x} = \vec{b}$$

same as

$$\begin{aligned}
&\mathbf{A}_{1,:}\vec{x} = \vec{b}_1\\
&\mathbf{A}_{2,:}\vec{x} = \vec{b}_2\\
&\vdots\\
&\mathbf{A}_{m,:}\vec{x} = \vec{b}_m
\end{aligned}$$

Let's $\vec{x} = [x_1, x_2, \dots, x_n]$ be a vector of unknown variables:

$$\begin{aligned}
&\mathbf{A}_{1,1}x_1 + \mathbf{A}_{1,2}x_2 \dots \mathbf{A}_{1,n}x_n = b_1\\
&\mathbf{A}_{2,1}x_1 + \mathbf{A}_{2,2}x_2 \dots \mathbf{A}_{2,n}x_n = b_2\\
&\vdots\\
&\mathbf{A}_{m,1}x_1 + \mathbf{A}_{m,2}x_2 \dots \mathbf{A}_{m,n}x_n = b_m
\end{aligned}$$

Norm

Norms are functions mapping vectors to non-negative values.

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$$

- $p = 1$ $\|x\|_1 = \sum_i |x_i|$
- $p = 2$ Euclidean norm (the Euclidean distance from the origin to the point identified by $\vec{x}$)
- $p = 2$ For **matrix** is known as **Frobenius norm**: $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} A_{i,j}^2}$

Special Vector and Matrices

Unit Vector

A vector with unit norm: $\|x\|_2 = 1$

Orthogonal Vectors

A vector $\vec{x}$ and a vector $\vec{y}$ are orthogonal to each other if $\vec{x}^T\vec{y} = 0$

Orthogonal Matrix

An orthogonal matrix is a **square** matrix $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$
and thus $\mathbf{A}^{-1} = \mathbf{A}^T$

Eigendecomposition

Eigendecomposition decompose a **square** matrix into a set of eigenvectors and eigenvalues. An **eigenvector** of a square matrix $\mathbf{A}$ is a nonzero vector $\mathbf{v}$ such that multiplication by $\mathbf{A}$ alters only the scale of $\vec{v}$

$$\mathbf{A}\vec{v} = \lambda\vec{v},$$

where λ is a scalar known as the **eigenvalue** corresponding to this eigenvector. If $\vec{v}$ is an eigenvector of $\mathbf{A}$, then so is any rescaled vector $s\vec{v}$ for $s \in \mathbb{R},\ s \neq 0$. Let's $\mathbf{V} = [\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n]$ and $\vec{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$, the eigendecomposition of $\mathbf{A}$ is then given by:

$$\mathbf{A} = \mathbf{V}diag(\vec{\lambda})\mathbf{V}^{-1}$$

Singular Value Decomposition (SVD)

More general decomposition than eigendecomposition. Every real matrix has an SVD.

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T,$$

where $\mathbf{A} \in \mathbb{R}^{n\times m},\ \mathbf{U} \in \mathbb{R}^{m\times n},\ \mathbf{D} \in \mathbb{R}^{m\times n}$ and it is diagonal, $\mathbf{U} \in \mathbb{R}^{n\times n}$.

- The elements along the diagonal of $\mathbf{D}$ are known as the **singular values** (*i.e.* square roots of the non-zero eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$);
- the columns of $\mathbf{U}$ are known as the **left-singular vectors** (*i.e.* eigenvectors of $\mathbf{A}\mathbf{A}^T$);
- The columns of $\mathbf{V}$ are known as as the **right-singular vectors** (*i.e.* eigenvectors of $\mathbf{A}^T\mathbf{A}$);

Exercise

Assume $\mathbf{A} = \mathbf{V}diag(\vec{\lambda})\mathbf{V}^{-1}$. Does these exist an eigendecomposition for $\mathbf{A}^2$? If so, what does it looks like?