Linear Algebra and Learning from Data

Multiplication $Ax$ and $AB$
Column space of $A$
Independent rows and basis
Row rank = column rank

Neural Networks and Deep Learning / new course and book
math.mit.edu/learningfromdata
By rows
\[
\begin{bmatrix}
2 & 3 \\
2 & 4 \\
3 & 7
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= \begin{bmatrix}
2x_1 + 3x_2 \\
2x_1 + 4x_2 \\
3x_1 + 7x_2
\end{bmatrix}
\]

By columns
\[
\begin{bmatrix}
2 & 3 \\
2 & 4 \\
3 & 7
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= x_1 \begin{bmatrix}
2 \\
3
\end{bmatrix}
+ x_2 \begin{bmatrix}
3 \\
4
\end{bmatrix}
\]
\( b = (b_1, b_2, b_3) \) is in the column space of \( A \) exactly when \( Axb = b \) has a solution \((x_1, x_2)\)

\[
b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ is not in } \mathbf{C}(A).
\]

\[
Ax = \begin{bmatrix} 2x_1 + 3x_2 \\ 2x_1 + 4x_2 \\ 3x_1 + 7x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ is unsolvable.}
\]

The first two equations force \( x_1 = \frac{1}{2} \) and \( x_2 = 0 \).

Then: \( 3 \left( \frac{1}{2} \right) + 7(0) = 1.5 \) (not 1).

What are the column spaces of

\[
A_2 = \begin{bmatrix} 2 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 7 & 10 \end{bmatrix} \text{ and } A_3 = \begin{bmatrix} 2 & 3 & 1 \\ 2 & 4 & 1 \\ 3 & 7 & 1 \end{bmatrix}
\]
If column 1 of $A$ is not all zero, put it into $C$.
If column 2 of $A$ is not a multiple of column 1, put it into $C$.
If column 3 of $A$ is not a combination of columns 1 and 2, put it into $C$. Continue.
At the end $C$ will have $r$ columns ($r \leq n$).
They will be a “basis” for the column space of $A$.
The left out columns are combinations of those basic columns in $C$. 
If \( A = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix} \) then \( C = \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \) \( n = 3 \) columns in \( A \) \( r = 2 \) columns in \( C \)

If \( A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix} \) then \( C = A \). \( n = 3 \) columns in \( A \) \( r = 3 \) columns in \( C \)

If \( A = \begin{bmatrix} 1 & 2 & 5 \\ 1 & 2 & 5 \\ 1 & 2 & 5 \end{bmatrix} \) then \( C = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \) \( n = 3 \) columns in \( A \) \( r = 1 \) column in \( C \)
\[ A = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} = CR \]

All we are doing is to put the right numbers in \( R \). Combinations of the columns of \( C \) produce the columns of \( A \). Then \( A = CR \) stores this information as a matrix multiplication.
The number of *independent columns* equals the number of *independent rows*

Look at $A = CR$ by rows instead of columns. $R$ has $r$ rows. **Multiplying by $C$ takes combinations.** Since $A = CR$, we get every row of $A$ from the $r$ rows of $R$. Those $r$ rows are independent — a **basis for the row space of $A$**.
Column-row multiplication of matrices

\[ AB = \begin{bmatrix} a_1 & \ldots & a_n \\ \vdots & \ddots & \vdots \\ a_1 & \ldots & a_n \end{bmatrix} \begin{bmatrix} b_1^* \\ \vdots \\ b_n^* \end{bmatrix} = a_1 b_1^* + a_2 b_2^* + \cdots + a_n b_n^*. \]

The \( i, j \) entry of \( a_k b_k^* \) is \( a_{ik} b_{kj} \).

Add to find \( c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} = \text{row } i \cdot \text{ column } j. \)

\[ A = LU \quad A = QR \quad S = Q\Lambda Q^T \quad A = X\Lambda X^{-1} \quad A = U\Sigma V^T \]
### Deep Learning by Neural Networks

<table>
<thead>
<tr>
<th></th>
<th>Key operation</th>
<th>Composition $F = F_3(F_2(F_1(x_0)))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Key rule</td>
<td>Chain rule for derivatives</td>
</tr>
<tr>
<td>3</td>
<td>Key algorithm</td>
<td>Stochastic gradient descent</td>
</tr>
<tr>
<td>4</td>
<td>Key subroutine</td>
<td>Backpropagation</td>
</tr>
<tr>
<td>5</td>
<td>Key nonlinearity</td>
<td>ReLU$(x) = \max(x, 0) = \text{ramp function}$</td>
</tr>
</tbody>
</table>
Feature vector $x_0$
Three components for each training sample
$y_1$ at layer 1
$y_1 = A_1 x_0 + b_1$

ReLU

$x_1$ at layer 1
$x_1 = \text{ReLU} (y_1)$

ReLU

Output $x_2$
$x_2 = A_2 x_1 + b_2$
Theorem

Suppose we have $N$ hyperplanes $H_1, \ldots, H_N$ in $m$-dimensional space $\mathbb{R}^m$. Those come from $N$ linear equations $a_i^T x + b_i = 0$, in other words from $Ax = b$. Then the number of regions bounded by the $N$ hyperplanes (including infinite regions) is probably $r(N, m)$ and certainly not more:

$$r(N, m) = \sum_{i=0}^{m} \binom{N}{i} = \binom{N}{0} + \binom{N}{1} + \cdots + \binom{N}{m}.$$  

Thus $N = 1$ hyperplane in $\mathbb{R}^m$ produces $\binom{1}{0} + \binom{1}{1} = 2$ regions (one fold). And $N = 2$ hyperplanes will produce $1 + 2 + 1 = 4$ regions provided $m \geq 2$. When $m = 1$ we have 2 folds in a line, which only separates the line into $r(2, 1) = 3$ pieces.
The theorem will follow from the recursive formula

\[ r(N, m) = r(N - 1, m) + r(N - 1, m - 1) \]

**Figure:** The \( r(2, 1) = 3 \) pieces of \( H \) create 3 new regions. Then the count becomes \( r(3, 2) = 4 + 3 = 7 \) flat regions in the continuous piecewise linear surface \( z = F(x_1, x_2) \).
Backpropagation: Reverse Mode Graph for Derivatives of $x^2(x + y)$

Figure: Reverse-mode computation of the gradient $(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y})$ at $x = 2, y = 3$. 
$AB$ first or $BC$ first? Compute $(AB)C$ or $A(BC)$?

**First way**

$AB = (m \times n) (n \times p)$ has $mnp$ multiplications

$(AB)C = (m \times p) (p \times q)$ has $mpq$ multiplications

**Second way**

$BC = (n \times p) (p \times q)$ has $npq$ multiplications

$A(BC) = (m \times n) (n \times q)$ has $mnq$ multiplications