# 신경망 네트워크와 수학적 기반

# Superposition

- $f : \mathbf{R}^n \to \mathbf{R}^m$ means $f$ is a function that maps $n$-vectors to $m$-vectors
- we write $f(x) = (f_1(x), \ldots, f_m(x))$ to emphasize components of $f(x)$
- we write $f(x) = f(x_1, \ldots, x_n)$ to emphasize components of $x$
- $f$ satisfies *superposition* if for all $x, y, \alpha, \beta$

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

- such an $f$ is called *linear*

## Linear equations

# Matrix-vector product function

- **with $A$ an $m \times n$ matrix, define $f$ as $f(x) = Ax$**
- **$f$ is linear:**

$$
\begin{aligned}
f(\alpha x + \beta y) &= A(\alpha x + \beta y) \\
&= A(\alpha x) + A(\beta y) \\
&= \alpha(Ax) + \beta(Ay) \\
&= \alpha f(x) + \beta f(y)
\end{aligned}
$$

- **converse is true: if $f : \mathbf{R}^n \to \mathbf{R}^m$ is linear, then**

$$
\begin{aligned}
f(x) &= f(x_1 e_1 + x_2 e_2 + \cdots + x_n e_n) \\
&= x_1 f(e_1) + x_2 f(e_2) + \cdots + x_n f(e_n) \\
&= Ax
\end{aligned}
$$

with $A = \left[\begin{array}{cccc} f(e_1) & f(e_2) & \cdots & f(e_n) \end{array}\right]$

## Linear equations
# Examples

- **reversal:** $f(x) = (x_n, x_{n-1}, \ldots, x_1)$

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & ⋰ & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}$$

- **running sum:** $f(x) = (x_1, x_1 + x_2, x_1 + x_2 + x_3, \ldots, x_1 + x_2 + \cdots + x_n)$

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

# Affine functions

- function $f : \mathrm{R}^n \rightarrow \mathrm{R}^m$ is *affine* if it is a linear function plus a constant, *i.e.*,

$$f(\mathrm{x}) = Ax + b$$

- same as:

$$f(\alpha\, \mathrm{x} + \beta\, y) = \alpha\, f(x) + \beta\, f(y)$$

holds for all $x$, $y$, and $\alpha$, $\beta$ with $\alpha + \beta = 1$

- Affine functions sometimes (incorrectly) called linear

# Taylor series approximation

- suppose $f : \mathrm{R}^n \to \mathrm{R}^m$ is differentiable
- first order Taylor approximation $\hat{f}$ of $f$ near $z$:

$$\begin{aligned}\hat{f}_i(x) &= f_i(z) + \frac{\partial f_i}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f_i}{\partial x_n}(z)(x_n - z_n) \\ &= f_i(z) + \nabla f_i(z)^T(x - z)\end{aligned}$$

- in compact notation: $\hat{f}(x) = f(z) + Df(z)(x - z)$
- $Df(x)$ is the $m \times n$ *derivative* or *Jacobian* matrix of $f$ at $z$

$$Df(z)_{ij} = \frac{\partial f_i}{\partial x_j}(z), \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- $\hat{f}(x)$ is a very good approximation of $f(x)$ for $x$ near $z$
- $\hat{f}(x)$ is an affine function of $x$

## Linear equations

# Systems of linear equations

- set (or *system*) of $m$ linear equations in $n$ variables $x_1, \ldots, x_n$:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n &= b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n &= b_2 \\ &\vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n &= b_m \end{aligned}$$

- $n$-vector $x$ is called the variable or unknowns
- $A_{ij}$ are the *coefficients*; $A$ is the coefficient matrix
- $b$ is called the *right-hand side*
- can express very compactly as $Ax = b$

# Systems of linear equations

- systems of linear equations classified as
  - under-determined if $m < n$ ($A$ wide)
  - square if $m = n$ ($A$ square)
  - over-determined if $m > n$ ($A$ tall)
- $x$ is called a *solution* if $Ax = b$
- depending on $A$ and $b$, there can be
  - no solution
  - one solution
  - many solutions