Understanding complex information-processing systems

Marr (1982)

Computational theory
What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?

Representation and algorithm
How can this computational theory be implemented? What is the representation for the input and output, and what is the algorithm for the transformation?

Hardware implementation
How can the representation and algorithm be realized physically?
Notation

- **Indices**: $i, j$ indices of units ($i$ sending, $j$ receiving)
- **Activation**: $a_j$ activation of unit $j$
- **Net Input**: $n_j$ summed net input to unit $j$
- **Weight**: $w_{ij}$ weight on connection from unit $i$ to unit $j$
- **Input**: $e_j$ external input to unit $j$
- **Threshold**: $\theta_j$ threshold for unit $j$
- **Bias**: $b_j$ bias (tonic input) to unit $j$ ($= -\theta_j$)

Types of units

**Binary threshold unit**

- $n_j = \sum_i a_i w_{ij} + e_j$
- $a_j = \begin{cases} 1 & \text{if } n_j > \theta_j \\ 0 & \text{otherwise} \end{cases}$

If “bias” $b_j = -\theta_j$, this is the same as

- $n_j = \sum_i a_i w_{ij} + e_j + b_j$
- $a_j = \begin{cases} 1 & \text{if } n_j > 0 \\ 0 & \text{otherwise} \end{cases}$

Will generally omit $b_j$ and $e_j$ in equations

- Bias $b_j$ can be treated as weight $w_{ij}$ from special unit with fixed activation $a_j = 1$.
- External input $e_j$ can be treated as incoming activation $a_i$ across connection with fixed weight $w_{ij} = 1$.

**Linear units**

- $a_j = n_j = \sum_i a_i w_{ij}$

**Rectified linear units (ReLUs)**

- $a_j = \max(0, n_j)$

**Sigmoidal (“logistic”, “semi-linear”) units**

- $a_j = \sigma(n_j) = \frac{1}{1 + \exp(-n_j)}$

**Binary stochastic units**

- $\rho(a_j = 1) = \frac{1}{1 + \exp(-n_j)}$

**Continuous time-averaged (cascaded) units** [two alternatives]

- $n_j[t] = \tau \sum_i a_i w_{ij} + (1 - \tau) n_j[t-1]$
- $a_j[t] = \tau \sigma(n_j[t]) + (1 - \tau) a_j[t-1]$

**Interactive activation**

(Jets & Sharks model; Schema model; McClelland & Rumelhart letter/word model)

- $n_j[t] = \sum_i a_i[t-1] w_{ij} + e_j[t]$
- $a_j[t] = (1 - \text{decay}) a_j[t-1] + \begin{cases} \frac{n_j[t-1]}{a_j[t-1] - \min} & \text{if } n_j[t] > 0 \\ n_j[t] & \text{otherwise} \end{cases}$

decay = 0.1 max = 1.0 min = -0.2