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 of units** ($i$ sending, $j$ receiving)
- **Activation** of unit $j$ ($a_j$)
- **Summed net input** to unit $j$ ($n_j$)
- **Weight** on connection from unit $i$ to unit $j$ ($w_{ij}$)
- **Threshold** for unit $j$ ($\theta_j$)
- **External input** to unit $j$ ($e_j$)
- **Bias** (tonic input) to unit $j$ ($b_j$)

**Linear units**

\[ a_j = n_j = \sum_i a_i w_{ij} \]

**Rectified linear units (ReLUs)**

\[ a_j = \max(0.0, n_j) \]

**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]

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

\[ a_j^{[t]} = \tau \sigma(n_j^{[t]}) + (1 - \tau) a_j^{[t-1]} \]

**Interactive activation**

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

\[ n_j^{[t]} = \sum_i a_i^{[t-1]} w_{ij} + e_j^{[t]} \]

\[ a_j^{[t]} = (1 - \text{decay}) a_j^{[t-1]} + \begin{cases} n_j^{[t]} (\max - a_j^{[t-1]}) & \text{if } n_j^{[t]} > 0 \\ n_j^{[t]} (a_j^{[t-1]} - \min) & \text{otherwise} \end{cases} \]

\[ \text{decay} = 0.1 \quad \max = 1.0 \quad \min = -0.2 \]
Constraint satisfaction

- Units represent **hypotheses** about parts of a problem
- Weights code **constraints** on how hypotheses can combine (i.e., the degree to which they are consistent or inconsistent)
- Possible **solutions** correspond to particular patterns of active units
- External input introduces **bias** to favor one possible solution over others

**Example: Map coloring**

Assign colors to regions so that no adjacent regions have the same color.

- **Hypotheses**
  - An assignment of a color to a region

- **Constraints**
  - Adjacent regions must be assigned different colors
  - Only one color can be assigned to each region
  - Each region must be assigned a color

- **Biases**
  - Initial color preference for a given region

**Necker Cube**
Maximizing Goodness (= minimizing Energy)

Global measure of degree to which activations satisfy weight constraints

\[ G (\text{Goodness} = -\text{Energy}) = \sum_{i<j} a_i a_j w_{ij} \]

How should unit \( k \) behave locally so as to increase global Goodness?

- Set \( a_k = 1 \) if \( G_{a_k=1} > G_{a_k=0} \) (or, equivalently, \( G_{a_k=1} - G_{a_k=0} > 0 \))

\[ G_{a_k=1} = \sum_i a_i w_{ik} + \sum_{i<j \neq k} a_i a_j w_{ij} \]

\[ G_{a_k=0} = \sum_{i<j \neq k} a_i a_j w_{ij} \]

\[ G_{a_k=1} - G_{a_k=0} = \sum_i a_i w_{ik} \]

- Set \( a_k = 1 \) if \( \sum_i a_i w_{ik} > 0 \) (= binary threshold unit)

\textit{Sigmoid units:} increase activation as net input increases (monotonicity)