

AGIMatematic

Documentação Matemática Completa de AGI

1. Sistema de Percepção

1.1 Processamento Visual

- Convolução 2D:**

$$F(i, j) = \sum_m \sum_n K(m, n) I(i - m, j - n)$$

onde K é o kernel e I é a imagem.

- Normalização em Lote:**

$$y = \gamma \left(\frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta$$

onde:

- μ : média do batch
- σ : desvio padrão
- γ, β : parâmetros treináveis
- ϵ : valor pequeno para estabilidade

1.2 Processamento de Linguagem

- Self-Attention:**

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

- Multi-Head Attention:**

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O$$

onde

$$\text{head}_i = \text{Attention}(QW^{Q_i}, KW^{K_i}, VW^{V_i})$$

2. Sistema de Memória

2.1 Memória de Trabalho

- **LSTM Gates:**

$$\begin{aligned}f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\c_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\c_t &= f_t \odot c_{t-1} + i_t \odot c_t \\h_t &= o_t \odot \tanh(c_t)\end{aligned}$$

2.2 Memória Associativa

- **Hopfield Network Update:**

$$\begin{aligned}E &= -\frac{1}{2} \sum_i \sum_j w_{ij} s_i s_j \\s_i(t+1) &= \text{sign} \left(\sum_j w_{ij} s_j(t) \right)\end{aligned}$$

3. Sistema de Raciocínio

3.1 Inferência Probabilística

- **Bayes Generalizado:**

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad \text{onde} \quad P(E) = \sum_i P(E|H_i)P(H_i)$$

3.2 Raciocínio Causal

- **Structural Causal Model:**

$$X_i = f_i(\text{PA}_i, U_i)$$

Para intervenções:

$$\text{do}(X = x): P(Y|\text{do}(X = x)) = \sum_Z P(Y|X = x, Z)P(Z)$$

4. Sistema de Aprendizado

4.1 Gradient Descent

- **Atualização de Pesos:**

$$w_t = w_{t-1} - \eta \nabla L(w_{t-1})$$

- **Adam Optimizer:**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$w_t = w_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

4.2 Q-Learning

- **Q-Value Update:**

$$Q(s, a) = Q(s, a) + \alpha \left[R + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

- **Double Q-Learning:**

$$Q_1(s, a) = Q_1(s, a) + \alpha \left[R + \gamma Q_2 \left(s', \arg \max_{a'} Q_1(s', a') \right) - Q_1(s, a) \right]$$

5. Sistema de Decisão

5.1 Planejamento

- **Value Iteration:**

$$V_{k+1}(s) = \max_a \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_k(s') \right]$$

- **Policy Iteration:**

$$\pi_k(s) = \arg \max_a \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_k(s') \right]$$

5.2 Multi-Objective Optimization

- **Pareto Front:**

$$P = \{x \in X \mid \neg \exists y \in X: y \text{ dominates } x\}$$

onde y domina x se $\forall i: f_i(y) \geq f_i(x) \wedge \exists j: f_j(y) > f_j(x)$

6. Sistema de Auto-Melhoria

6.1 Architecture Search

- **Neural Architecture Search:**

$$A^* = \operatorname{argmax}_A \mathbb{E}_{(x,y) \sim D} [L(w^*(A), x, y)]$$

onde $w^* = \operatorname{argmin}_w \mathbb{E}_{(x,y) \sim D_{\text{train}}} [L(w, A, x, y)]$

6.2 Meta-Learning

- **MAML Update:**

$$\begin{aligned}\theta' &= \theta - \alpha \nabla_{\theta} L_{\tau}(f_{\theta}) \\ \theta &= \theta - \beta \nabla_{\theta} \sum_{\tau} L_{\tau}(f_{\theta'})\end{aligned}$$

7. Integração de Sistemas

7.1 Information Flow

- **Entropy:**

$$H(X) = -\sum p(x) \log p(x)$$

- **Mutual Information:**

$$I(X; Y) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

7.2 System Synchronization

- **Phase Locking:**

$$\frac{d\theta_i}{dt} = \omega_i + K \sum_j \sin(\theta_j - \theta_i)$$

8. Métricas de Performance

8.1 Error Metrics

- **Cross Entropy Loss:**

$$L = - \sum_i y_i \log(\hat{y}_i)$$

- **KL Divergence:**

$$D_{KL}(P||Q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

8.2 Performance Bounds

- **PAC Learning:**

$$P(|err(h) - err_S(h)| \leq \epsilon) \geq 1 - \delta$$

onde

$$m \geq O\left(\frac{1}{\epsilon^2} \left(\ln|H| + \ln \frac{1}{\delta}\right)\right)$$

9. Restrições de Segurança

9.1 Value Alignment

- **Inverse Reward Learning:**

$$R^* = \operatorname{argmax}_R P(D|\pi_R^*)P(R)$$

onde

$$\pi_R^* = \operatorname{argmax}_{\pi} \mathbb{E}[\sum \gamma^t R(s_t, a_t)]$$

9.2 Robustness

- **Adversarial Training:**

$$\min_{\theta} \mathbb{E} \left[\max_{\|\delta\| \leq \epsilon} L(x + \delta, y; \theta) \right]$$

10. Otimização de Recursos

10.1 Memory Management

- **Memory Access:**

$$P_{\text{hit}} = 1 - \left(1 - \frac{1}{n}\right)^k$$

onde n é o número de slots de memória e k é o número de acessos.

10.2 Compute Allocation

- **Load Balancing:**

$$\text{Load}_i = \frac{\lambda_i}{\mu_i}$$

$$\text{Balance} = \max_i \text{Load}_i - \min_i \text{Load}_i$$

Adilson Oliveira

Esta documentação abrange os fundamentos matemáticos essenciais para a construção de uma AGI. Cada componente descrito aqui requer ajustes e otimizações específicas para o contexto de aplicação, necessitando de experimentação e refinamento contínuos para alcançar a inteligência geral artificial.