Formulación Matemática Detallada de Kistmat_AI

1. Generación de Datos y Preprocesamiento

Sea P el conjunto de problemas matemáticos generados. Cada problema $p \in P$ se define como:

$$p = (t, s, d, c)$$

Donde:

- $t \in T$: texto del problema (espacio de todos los textos posibles)
- $s \in \mathbb{C}$: solución (número complejo para abarcar soluciones reales e imaginarias)
- $d \in [1, 3]$: nivel de dificultad (número real entre 1 y 3)
- $c \in C$: concepto matemático (conjunto finito de conceptos)

La función de generación de datos G está parametrizada por la etapa de aprendizaje $e \in E$ y el nivel de dificultad d:

$$G : E \times [1, 3] \rightarrow P^n$$

Donde n es el número de problemas generados, típicamente entre 4000 y 5000.

Para cada etapa e, existe una función específica de generación G_e:

$$G_e(d) = \{p_i = (t_i, s_i, d, c_i) \mid i = 1, \ldots, n\}$$

Por ejemplo, para la etapa "elementary1":

$$G_{elementary1}(d) = \{(f(a_i, b_i), eval(f(a_i, b_i)), d, op_i) \mid i = 1, \ldots, n\}$$

Donde $f(a, b) = "a \text{ op } b"$, $a_i, b_i \sim U(1, \lfloor 10d \rfloor)$, $op_i \in \{+, -\}$, y $eval$ es la función de evaluación de la expresión.

2. Tokenización

La función de tokenización T_e para la etapa e se define como:

$$T_e : T \rightarrow \mathbb{N}^m$$

Donde m es la longitud máxima de la secuencia (MAX_LENGTH en el código).

Para problemas básicos:

$$T_{basic}(t) = [hash(w_i) \mod V \mid w_i \in split(t)]$$

Para problemas avanzados:

$$T_{advanced}(t) = [f_{token}(w_i) \mid w_i \in split(t)]$$

Donde:

$$f_{token}(w) = \begin{cases} hash(w) \mod V & \text{si w es alfabético} \\ ord(w) & \text{si w es dígito o símbolo} \\ \lfloor float(w) \cdot 100 \rfloor & \text{en otro caso} \end{cases}$$

Para problemas de cálculo:

$$T_{calculus}(t) = [\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k]$$

Donde α_i son coeficientes normalizados y β_i son exponentes normalizados de los términos en la expresión de cálculo.

3. Modelo Principal (Kistmat_AI)

3.1 Capa de Embedding

$$E = Emb(x) \in R^{m \times d_e}$$

Donde $x \in N^m$ es el vector de tokens de entrada, d_e es la dimensión del embedding.

3.2 LSTMs Bidireccionales

Para cada capa LSTM bidireccional $l = 1, 2$:

$$\vec{h}_{l,t} = LSTM_l(\vec{h}_{l,t-1}, E_t)$$

$$\backslash cevh_{l,t} = LSTM_l(\backslash cevh_{l,t+1}, E_t)$$

$$H_l = [\vec{h}_{l,1}, \ldots, \vec{h}_{l,m}] \oplus [\backslash cevh_{l,1}, \ldots, \backslash cevh_{l,m}]$$

Donde $\oplus$ denota la concatenación, $H_0 = E$, y $H_l \in R^{m \times 2d_h}$, siendo d_h la dimensión oculta del LSTM.

3.3 Capa de Dropout

$$H_{drop} = Dropout(H_2, p = 0.5)$$

3.4 Atención Multi-Cabeza

Para cada cabeza de atención $i = 1, \ldots, h$:

$$Q_i = H_{drop} W_i^Q, \quad K_i = H_{drop} W_i^K, \quad V_i = H_{drop} W_i^V$$

$$head_i = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

$$\mathrm{MultiHead}(H_{drop}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O$$

Donde $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{2d_h \times d_k}$ y $W^O \in \mathbb{R}^{hd_k \times 2d_h}$.

3.5 Consulta de Memoria

$$q = \mathrm{MemoryQuery}(\mathrm{MultiHead}(H_{drop}))$$

Donde $q \in \mathbb{R}^{d_q}$ y d_q es la dimensión de la consulta de memoria.

3.6 Razonamiento y Salida

$$r = \mathrm{ReLU}(W_r \mathrm{MultiHead}(H_{drop}) + b_r)$$

$$y = W_o r + b_o$$

Donde $W_r \in \mathbb{R}^{d_r \times 2d_h}$, $b_r \in \mathbb{R}^{d_r}$, $W_o \in \mathbb{R}^{2 \times d_r}$, $b_o \in \mathbb{R}^2$, y $y \in \mathbb{R}^2$ representa la parte real e imaginaria de la solución predicha.

4. Sistema de Memoria Integrado

4.1 Memoria Externa

Sea $M_{ext} \in \mathbb{R}^{n_{ext} \times d_v}$ la matriz de embeddings de memoria externa, donde n_{ext} es el número de entradas y d_v es la dimensión del valor.

$$\mathrm{sim}_{ext}(q) = qM_{ext}^T$$

$$m_{ext} = \mathrm{TopK}(\mathrm{sim}_{ext}(q), k) \cdot M_{ext}$$

4.2 Memoria Formulativa

Sea $F = \{(f_i, e_i, T_i) \mid i = 1, \ldots, n_f\}$ el conjunto de fórmulas almacenadas, donde f_i es la fórmula, $e_i \in \mathbb{R}^{d_f}$ es su embedding, y T_i es el conjunto de términos asociados.

$$\mathrm{sim}_{form}(q, T) = qE_T^T$$

$$m_{form} = \mathrm{TopK}(\mathrm{sim}_{form}(q, T), k) \cdot E_T$$

Donde E_T es la matriz de embeddings de las fórmulas cuyos términos intersectan con los de la consulta.

4.3 Memoria Conceptual

Sea $C = \{(c_i, e_i) \mid i = 1, \ldots, n_c\}$ el conjunto de conceptos almacenados, donde c_i es el concepto y $e_i \in \mathbb{R}^{d_c}$ es su embedding.

$$\mathrm{sim}_{\mathrm{conc}}(q) = qE_c^T$$

$$m_{\mathrm{conc}} = \mathrm{TopK}(\mathrm{sim}_{\mathrm{conc}}(q), k) \cdot E_c$$

4.4 Memoria a Corto Plazo

Sea $S = [s_1, \ldots, s_{n_s}]$ la lista de estados recientes, donde $s_i \in \mathbb{R}^{d_s}$.

$$\mathrm{sim}_{\mathrm{short}}(q) = qS^T$$

$$m_{\mathrm{short}} = \mathrm{TopK}(\mathrm{sim}_{\mathrm{short}}(q), k) \cdot S$$

4.5 Memoria a Largo Plazo

Sea $L = \{(l_i, \alpha_i) \mid i = 1, \ldots, n_l\}$ el conjunto de memorias a largo plazo, donde $l_i \in \mathbb{R}^{d_l}$ es la memoria y α_i es su importancia.

$$\mathrm{sim}_{\mathrm{long}}(q) = qL^T \odot \alpha$$

$$m_{\mathrm{long}} = \mathrm{TopK}(\mathrm{sim}_{\mathrm{long}}(q), k) \cdot L$$

4.6 Memoria de Inferencia

Sea $I = \{(i_j, \beta_j) \mid j = 1, \ldots, n_i\}$ el conjunto de inferencias almacenadas, donde $i_j \in \mathbb{R}^{d_i}$ es la inferencia y β_j es su confianza.

$$\mathrm{sim}_{\mathrm{inf}}(q) = qI^T \odot \beta$$

$$m_{\mathrm{inf}} = \mathrm{TopK}(\mathrm{sim}_{\mathrm{inf}}(q), k) \cdot I$$

4.7 Integración de Memorias

$$M = [m_{\mathrm{ext}}; m_{\mathrm{form}}; m_{\mathrm{conc}}; m_{\mathrm{short}}; m_{\mathrm{long}}; m_{\mathrm{inf}}]$$

$$m_{\mathrm{integrated}} = \mathrm{Attention}(q, M, M)$$

5. Razonamiento Simbólico

Sea S el espacio de expresiones simbólicas y $f_s : S \to C$ la función de evaluación simbólica.

Para una ecuación lineal $ax + b = c$:

$$f_s(ax + b = c) = \frac{c - b}{a}$$

Para una ecuación cuadrática $ax^2 + bx + c = 0$:

$$f_s(ax^2 + bx + c = 0) = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para límites:

$$f_s(\lim_{x \to a} g(x)) = \lim_{h \to 0} g(a + h)$$

Para derivadas:

$$f_s(\frac{d}{dx}(\sum_{i=0}^{n} a_i x^i)) = \sum_{i=1}^{n} i a_i x^{i-1}$$

6. Aprendizaje por Refuerzo (continuación)

Sea s_t el estado en el tiempo t, a_t la acción tomada, r_t la recompensa recibida, y $\pi_\theta(a|s)$ la política parametrizada por θ. El gradiente del objetivo $J(\theta)$ se define como:

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)(\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'})]$$

Donde $\gamma \in [0, 1]$ es el factor de descuento.

Para el caso específico de Kistmat_AI, definimos:

- Estado s_t: El problema matemático actual y el contexto del aprendizaje.
- Acción a_t: La predicción del modelo para la solución del problema.
- Recompensa r_t: Una función del error entre la predicción y la solución real:

$$r_t = f(y_t, \hat{y}_t) = \begin{cases} 1 & \text{si } |y_t - \hat{y}_t| < \epsilon \\ -\frac{|y_t - \hat{y}_t|}{\max(|y_t|, 1)} & \text{en otro caso} \end{cases}$$

Donde y_t es la solución real, $\hat{y}_t$ es la predicción del modelo, y ϵ es un umbral de tolerancia.

La actualización de los parámetros se realiza mediante:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta_t)$$

Donde α es la tasa de aprendizaje.

7. Entrenamiento Curricular

Sea $E = \{e_1, \ldots, e_9\}$ el conjunto de etapas de aprendizaje, donde:

$$e_1 = \text{elementary1}, \ldots, e_9 = \text{university}$$

Para cada etapa e_i, definimos un umbral de preparación τ_i:

$$\tau = [\tau_1, \ldots, \tau_9] = [0.95, 0.93, 0.91, 0.89, 0.87, 0.85, 0.83, 0.81, 0.80]$$

La función de pérdida L_{e_i} para la etapa e_i se define como:

$$L_{e_i}(\theta) = \frac{1}{|P_{e_i}|} \sum_{p \in P_{e_i}} \|y_p - \hat{y}_p\|^2$$

Donde P_{e_i} es el conjunto de problemas para la etapa e_i, y_p es la solución verdadera y $\hat{y}_p$ es la predicción del modelo.

El criterio de avance a la siguiente etapa se define como:

$$\text{Avanzar si: } R^2(P_{e_i}^{val}) > \tau_i$$

Donde $R^2(P_{e_i}^{val})$ es el coeficiente de determinación en el conjunto de validación de la etapa e_i.

8. Evaluación

8.1 Error Cuadrático Medio (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \|y_i - \hat{y}_i\|^2$$

Donde n es el número de muestras, y_i es la solución verdadera y $\hat{y}_i$ es la predicción del modelo.

8.2 Coeficiente de Determinación (R²)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \|y_i - \hat{y}_i\|^2}{\sum_{i=1}^{n} \|y_i - \bar{y}\|^2}$$

Donde $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ es la media de los valores verdaderos.

8.3 Error Absoluto Medio (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

9. Optimización

El objetivo general de optimización se puede expresar como:

$$\min_{\theta} \sum_{e \in E} \lambda_e L_e(\theta) + \lambda_R R(\theta)$$

Donde λ_e son los pesos para cada etapa de aprendizaje, $L_e(\theta)$ es la función de pérdida para la etapa e, λ_R es el factor de regularización, y $R(\theta)$ es el término de regularización (por ejemplo, L2: $R(\theta) = \|\theta\|^2$).

La actualización de los parámetros se realiza mediante el algoritmo Adam:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla_\theta L(\theta_t) \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2)(\nabla_\theta L(\theta_t))^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t
\end{aligned}
$$

Donde β_1, β_2 son los factores de decaimiento para las estimaciones del primer y segundo momento, η es la tasa de aprendizaje, y ϵ es un pequeño valor para evitar la división por cero.