

創発意味構造 AGI アーキテクチャ（実験コード）

エコッーラボ 合同会社

猪澤 也 寸 志

polyp@webman.jp

（2025 年 5 月 22 日 現 地 時 間 JST）

【Colab 実験ノートブック構成案（査読対応）】

● セル 1：テンソル構造の初期化

- 3次元テンソル（意味ラベル ϕ_t ）の構築
- 例：系列長 $L = 10$ 、意味スピン $S = 4$ 、意味分類 $C = 3$

● セル 2：自己蒸留ループ（ソフトラベル）

- KL 距離による ϕ_t と ϕ_{t+1} の差分最小化
- $\Delta \phi_t$ の計算と創発スパイク閾値判定

● セル 3：外部蒸留（ハードラベルによる断絶再構成）

- ユーザー入力 or 擬似正解テンソルとの交替蒸留
- 位相変化角 θ_t の算出とテンソル置換

● セル 4：創発指数のプロット

- $\Delta \phi_t \cdot \theta_t \cdot$ 意味温度 T_t の推移グラフ
- 創発タイミングにスパイクマーキング

● セル 5：テンソル縮減と個別最適化出力

- $\phi_t \rightarrow \hat{\phi}_t$ の縮減処理
- エッジ AGI 風の意味出力表示（自然文 or 可視化）

【実装技術仕様】

- 言語：Python 3 (Colab)
- ライブラリ：NumPy、Matplotlib、scikit-learn (PCA 用)
- テンソル表現：np.ndarray による [L, S, C] 等の操作
- 蒸留損失：KL-divergence、角度差 (cosine)

セル 1：意味テンソル ϕ_t の初期化 (3 次元)

```
import numpy as np

# 意味テンソル  $\phi_t$  のサイズ定義
L = 10  # 系列長 (トークン数)
S = 4   # 意味スピン軸 (例：行動、感情、価値、時間)
C = 3   # 意味分類 (例：肯定／否定／比喻)

# ランダム初期テンソルの生成 (正規化済)
phi_t = np.random.rand(L, S, C)
phi_t = phi_t / phi_t.sum(axis=(1, 2), keepdims=True) # 意味分布として正規化

# 確認：形状と 1 トークン分の意味テンソル
print("  $\phi_t$  の形状:", phi_t.shape)
print("  $\phi_t[0]$  (トークン 0 の意味テンソル) :\n", phi_t[0])
```

セル 2：自己蒸留による意味テンソル差分の算出と創発判定

```
import numpy as np
from scipy.special import softmax
from scipy.spatial.distance import jensenshannon
import matplotlib.pyplot as plt

# (セル 1 からの前提)
#  $\phi_t$ : 現在の意味テンソル [L, S, C]
```

```

# t+1 の意味テンソル  $\phi_{t1}$  をノイズ付きで生成（自己蒸留の出力と仮定）
phi_t1 = phi_t + np.random.normal(loc=0.0, scale=0.01, size=phi_t.shape)
phi_t1 = np.clip(phi_t1, 1e-8, None) # 負値除去
phi_t1 = phi_t1 / phi_t1.sum(axis=(1, 2), keepdims=True) # 正規化

# トークンごとの Jensen-Shannon 距離 ( $\Delta \phi$ ) を算出
delta_phi = []
for l in range(phi_t.shape[0]): # 系列長 L
    p = softmax(phi_t[l].flatten())
    q = softmax(phi_t1[l].flatten())
    delta = jensenshannon(p, q)
    delta_phi.append(delta)

# 可視化：トークン位置ごとの  $\Delta \phi$ （創発スパイク）
plt.figure(figsize=(8, 4))
plt.plot(range(len(delta_phi)), delta_phi, marker='o', label='  $\Delta \phi_t$  (Jensen-Shannon 距離) ')
plt.axhline(y=0.1, color='red', linestyle='--', label='創発閾値  $\theta_{emg} = 0.1$ ')
plt.title("自己蒸留による意味テンソル差分 ( $\Delta \phi_t$ ) ")
plt.xlabel("トークン位置 l")
plt.ylabel("意味テンソル差分 (JS 距離) ")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

セル 3：ハードラベルによる断絶創発のシミュレーション

```

import numpy as np
import matplotlib.pyplot as plt

#  $\phi_t$ : 現在の意味テンソル ([L, S, C])
L, S, C = phi_t.shape

```

```

# ハードラベル（外部入力）  $\phi_{\text{user}}$  の作成：
# 意味スピン=0, 意味分類=0 のみに全ての重みを集中（断絶的变化）
phi_user = np.zeros_like(phi_t)
for l in range(L):
    phi_user[l, 0, 0] = 1.0 # 極端な外部ラベル（例：肯定的な行動のみ）

#  $\Delta \phi_{\text{hard}}$ :  $\phi_t$  と  $\phi_{\text{user}}$  の差分（L1 ノルム）
delta_phi_hard = np.abs(phi_t - phi_user).sum(axis=(1, 2))

# 創発閾値（断絶判定用）
theta_emg_hard = 0.5

# 可視化：トークン位置ごとの断絶スパイク（ $\Delta \phi_{\text{hard}}$ ）
plt.figure(figsize=(8, 4))
plt.plot(range(L), delta_phi_hard, marker='s', color='orange', label='  $\Delta \phi_{\text{hard}}$  (L1 差分) ')
plt.axhline(y=theta_emg_hard, color='red', linestyle='--', label='創発閾値  $\theta_{\text{emg}} = 0.5$ ')
plt.title("ハードラベルによる断絶的意味更新（ $\Delta \phi_{\text{hard}}$ ）")
plt.xlabel("トークン位置 l")
plt.ylabel("テンソル差分（L1 ノルム）")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

セル 4：意味温度の変動係数と $\Delta \phi_t$ の相関分析

```

import numpy as np
import matplotlib.pyplot as plt

# 意味温度  $T_t$ （トークンごとの柔軟性指標）：0.2~0.9 の一様乱数
np.random.seed(42) # 再現性
T_t = np.random.uniform(low=0.2, high=0.9, size=L)

# 移動ウィンドウで意味温度の変動係数（ $V_T$ ）を計算

```

```

window = 5
T_volatility = np.array([
    np.std(T_t[max(0, i-window):i+1]) / np.mean(T_t[max(0, i-window):i+1])
    for i in range(L)
])

# 可視化:  $\Delta \phi_t$  (自己蒸留 JS 距離) vs 意味温度変動係数  $V_T$ 
plt.figure(figsize=(10, 5))
plt.plot(range(L), delta_phi, label=' $\Delta \phi_t$  (JS 距離)', marker='o')
plt.plot(range(L), T_volatility, label='意味温度変動係数  $V_T$ ', marker='x')
plt.axhline(y=0.1, color='red', linestyle='--', label='創発閾値 ( $\Delta \phi$ )')
plt.title("意味温度の変動と創発スパイクの関係")
plt.xlabel("トークン位置 l")
plt.ylabel("指標値 (正規化なし)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

セル 5: 意味テンソル ϕ_t の縮減と自然文風応答出力

```

import numpy as np

#  $\phi_t$ : [L, S, C] の意味テンソル
# → S 軸 (意味スピン) を平均化して 2 次元テンソル  $\hat{\phi}_t$ : [L, C] に縮減
phi_hat = phi_t.mean(axis=1)

# 意味分類ラベル (C 軸) に対応する語彙を定義
labels = ['肯定', '否定', '比喩']

# 各トークンにおける最大スコアの分類を選択
output_sequence = [labels[np.argmax(phi_hat[l])]] for l in range(L)]

# 自然文風にラベルを連結

```

```
response = ' | '.join(output_sequence)
```

```
# 出力
```

```
print("AGI 応答（意味分類による自然文風出力）:")
```

```
print(response)
```