# Transformer モデルのニューロンには局所的に概念についての知識がエンコードされている

有山 知希 [1]　Benjamin Heinzerling[2,1]　乾 健太郎 [1,2]
[1] 東北大学　[2] 理化学研究所
tomoki.ariyama.s3@dc.tohoku.ac.jp
benjamin.heinzerling@riken.jp,　inui@ecei.tohoku.ac.jp

## 概要

事前学習言語モデルには，マスク穴埋め問題を解くことが出来ることから何らかの形で知識が保存されていると考えられているが，その保存形態については未だよく分かっていない．そこで本研究では，知識が言語モデル内のパラメータに「局所的に」エンコードされているという仮定の下，メモリとしても働くとされる Transformer の Feed-Forward 層に着目して，ある概念についての知識がエンコードされている特定のニューロンが存在することを確認した．また，概念を表す単語の品詞によって，知識の保存形態が異なる可能性が示唆された．

## 1　はじめに

事前学習言語モデルの中には，穴埋め文，例えば「__が出来ることの一つはニャーと鳴くことです。」という文が与えられた場合に，穴埋め部分に「子猫」が入ると予測できるものが存在する．このような言語モデルは学習の結果，図 1 に示すように何らかの形でモデル内に子猫についての知識が保存されていると考えられる．しかし，この知識がどのような形で言語モデル内に保存されているか，ということは未だ解明されていない．

そこで我々は，ある概念[1)]についての知識は言語モデルの一部のパラメータにエンコードされている，すなわち「局所的に」保存されているという仮定を置き，Transformer[1] の Feed-Forward 層（以下，「FF 層」と呼ぶ）を調査することにした．これは Transformer の FF 層は key-value メモリと同様の働きをすることが Geva らによって報告されており [2]，そのため FF 層には概念についての知識がエンコードされている可能性が高いと考えたためである．


**図 1**　子猫に関する知識はどのように保存されているか？

Transformer の FF 層を分析する手法については，FF 層の中間表現をニューロンと見立てた時に，入力に反応する特定のニューロンを帰属法を用いて探す手法が Dai らによって提案されている [3]．そこで我々はこの手法を用い，ある概念についての知識が特定のニューロンにエンコードされているかを調査し，その知識がエンコードされていると判断されたニューロンの活性値を編集すると，その概念を正解とする穴埋め問題を解く際の正解を選ぶ確率が変化することを確認する．

## 2　手法

実験手法を説明するにあたり，まず本論文で用いる「ニューロン」という言葉について説明する．本論文におけるニューロンとは，Transformer の encoder を構成する一モジュールである FF 層において，第一線形層の出力を活性化関数にかけたものを指す．FF 層の式は入力を $x$，第一・第二線形層の重み・バイアス項をそれぞれ $W_1, b_1, W_2, b_2$ で表し，活性化関数に GELU[4] を用いることにすると，次のよ

1)　本論文において「概念」とは，名詞や固有名詞等で表される「エンティティ」や，動詞や形容詞等で表される「動作」「性質」などを全て含めた単語，として定義する

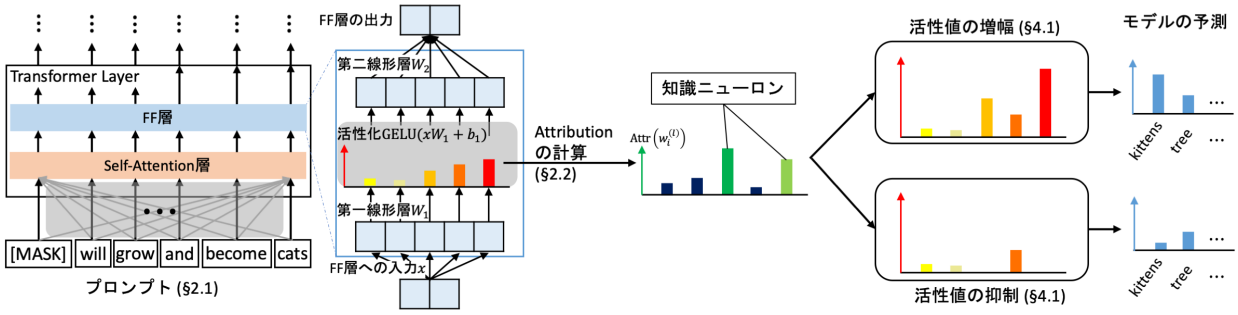


**図 2**　知識帰属法と実験のイメージ．知識帰属法では，モデルがプロンプトのマスク部分を予測する際の，FF 層における各ニューロンの活性値を用いて各貢献度を計算し，それらを元に知識ニューロンを探し出す．実験では，知識ニューロンの活性値を抑制・増幅したモデルに再度プロンプトを解かせ，モデルの予測がどのように変化するかを観察する．

うに表される．

$$\mathrm{FF}(x) = (\mathrm{GELU}(xW_1 + b_1))W_2 + b_2$$

すなわち，“$\mathrm{GELU}(xW_1 + b_1)$” の部分がニューロンに対応し，その値がニューロンの活性値となる．

## 2.1　知識ニューロンを探すためのタスク

ある概念に紐づくニューロン（以下，「知識ニューロン」と呼ぶ）を探すためのタスクとして，穴埋め文（以下，「プロンプト」と呼ぶ）の穴埋め部分を予測するタスクを言語モデルに解かせる．プロンプトは，その概念が穴埋め部分，すなわち [MASK] トークンとなるようにデータセットから作成する．プロンプトの一例を以下に示す．下記例の [MASK] トークンに対応する概念は “kittens” である．

- [MASK] will grow and become cats.

## 2.2　知識帰属法

この節では，Dai ら [3] によって提案された，知識ニューロンを探すための手法である知識帰属法について説明する (図 2 参照)．

知識ニューロンを探すため，事前学習言語モデルにおける各ニューロンの，「言語モデルが，あるプロンプト $x$ について正しい答えを出力する確率 $\mathrm{P}_x(\hat{w}_i^{(l)})$」に貢献する度合いを測定する．ここで確率 $\mathrm{P}_x(\hat{w}_i^{(l)})$ は，$y^*$ を正しい答え，$w_i^{(l)}$ を $l$ 番目の FF 層の $i$ 番目のニューロン，$\hat{w}_i^{(l)}$ をそのニューロンの活性値とすると，次の式 1 のように表される．

$$\mathrm{P}_x(\hat{w}_i^{(l)}) = p(y^* | w_i^{(l)} = \hat{w}_i^{(l)}) \tag{1}$$

この確率について，Sundararajan ら [5] の “Integrated Gradients” という帰属法を用い，$w_i^{(l)}$ を 0 から事前学習言語モデルによって計算された元の活性値 $\bar{w}_i^{(l)}$ まで徐々に変化させ，それに伴って変化する，確率 $\mathrm{P}_x(\hat{w}_i^{(l)})$ に対する勾配 $\frac{\partial \mathrm{P}_x(\alpha \bar{w}_i^{(l)})}{\partial w_i^{(l)}}$ を積分することで，今考えているニューロン $w_i^{(l)}$ の貢献度 $\mathrm{Attr}(w_i^{(l)})$ を計算することができる．

$$\mathrm{Attr}(w_i^{(l)}) = \bar{w}_i^{(l)} \int_{\alpha=0}^{1} \frac{\partial \mathrm{P}_x(\alpha \bar{w}_i^{(l)})}{\partial w_i^{(l)}} d\alpha \tag{2}$$

この値が大きいほど，プロンプト $x$ に強く反応するニューロンであると判定する．この手法を用いて，あるプロンプトについてモデル内の全てのニューロンの貢献度を計算し，その中での貢献度の閾値 $t$ を超えるニューロンのみを選ぶことにする．

しかし，上述のようにある一つのプロンプトについてのニューロンを探し出しても，それらのニューロンは本当に知識ニューロンであるとは限らない．なぜならば，そのプロンプトの構文情報を表現してしまっているような「偽陽性の」ニューロンが存在している可能性があるためである．そこで目的とする知識ニューロンを，先述の貢献度によって選出されたニューロンに対して精製作業を加えた以下の方法で手に入れる：

1. ある概念が正解となるプロンプトを，構文や含まれる語彙が異なるようにして複数用意する
2. 各プロンプトについて，各ニューロンの貢献度を計算する
3. 各プロンプトについて，閾値 $t$ を超える貢献度を持つニューロンのみを選出する
4. 全てのプロンプト間での共有率の閾値 $p$% を設定し，$p$% 以上のプロンプトで共有されているニューロンのみを残す

最後のステップで残ったニューロンは，各プロンプトで共有されている要素，すなわち概念と紐づくニューロン（=知識ニューロン）である．

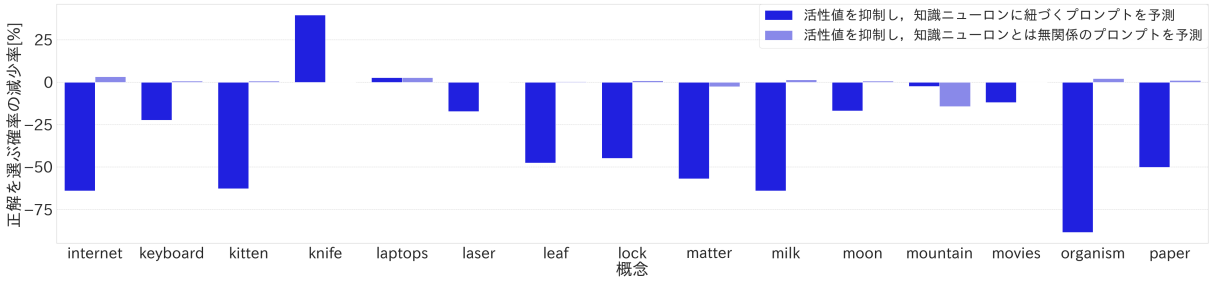


**図 3**　活性値を抑制した際の各プロンプトに対する正解を選ぶ確率の変化例

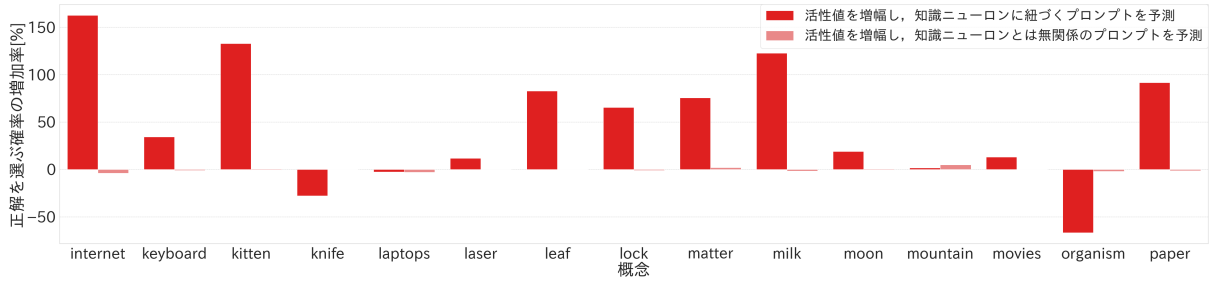


**図 4**　活性値を増幅した際の各プロンプトに対する正解を選ぶ確率の変化例

# 3　実験

Dai ら [3] は 2.2 節の手法を用いて高々数十種類の関係（例えば “born in ”）の知識ニューロンが存在するかを調べるに留まっていた．一方我々は，それよりはるかに種類の多い概念についても個別の概念に対応する知識ニューロンが存在するかを調べる．

## 3.1　設定

プロンプトについては，英語のデータセットである LAMA[6] の ConceptNet 部分を用いて relation の subject 部分をマスクし，最終的に 742 個の概念について作成した．事前学習言語モデルには，HuggingFace Transformers[7] で公開された “bert-base-uncased” を使用した．2.2 節で述べた各プロンプトにおける閾値 $t$ は最も大きい貢献度の 0.2 倍とし，プロンプト間の共有率 $p$ は 50% に設定した．なお，実験に使用したコードはすべて公開する[2]．

## 3.2　知識ニューロンと知識のエンコード

2.2 節の手法により得られた知識ニューロンの活性値 $\bar{w}_i^{(l)}$ を編集したモデルに「その知識ニューロンが紐づく概念が正解のプロンプト」と「全く関係のない概念が正解のプロンプト」のそれぞれを解かせて正解を選ぶ確率を調査し，活性値を変化させる前の確率との変化量を比較する．もし無関係の概念のプロンプトを解いた際の正解を選ぶ確率にのみ変化が見られなければ，その知識ニューロンには紐づく概念についての知識のみがエンコードされていると考えられる[3]．また，活性値は 0 にする（抑制）元の活性値 $\bar{w}_i^{(l)}$ の 2 倍にする（増幅）の二通りの編集方法を実験する．直感的には，抑制は知識ニューロンを取り除く操作でモデルはその知識ニューロンが紐づく概念を忘れ，逆に増幅は知識ニューロンが紐づく概念を強化する．

## 3.3　概念の品詞によって，活性値の編集による影響に違いがあるか

言語モデルが概念の品詞を区別できるとすれば，それは品詞によって知識の保存形態が異なるためであるという仮定の下，各概念を品詞ごとに分けた上で 3.2 節と同様の実験を行い，品詞の違いによる活性値編集の影響の差異を観察する．実験では，品詞分類は形態素解析器である nltk[8] を用いて「名詞」と「動詞・形容詞・副詞」の二種類に分類した．

# 4　実験結果

## 4.1　知識ニューロンと知識のエンコード

図 3 に抑制時のグラフを，図 4 に増幅時のグラフを示す．各図の概念は，冒頭で例として取り上げた子猫 (=“kitten”) と辞書順にその周辺の綴りを持つものを選出した．図 3，4 より，その知識ニューロン

2）　https://github.com/tomokiariyama/concept-neurons.git

3）　無関係の概念には一律で “london” を用いた．

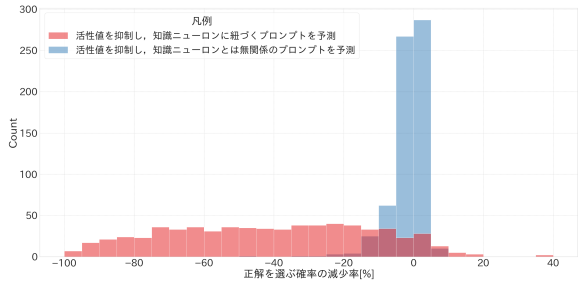


**図 5**　活性値を抑制した際の全概念についての正解を選ぶ確率の変化

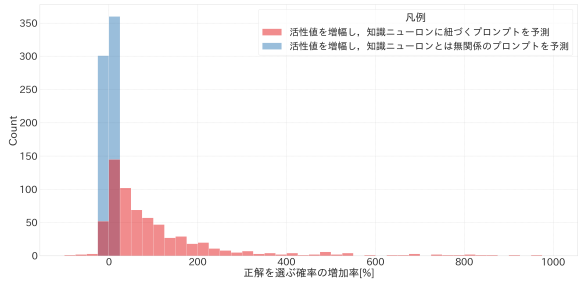


**図 6**　活性値を増幅した際の全概念についての正解を選ぶ確率の変化

が紐づく概念が正解のプロンプトを解かせると，活性値の編集に応じて正解を選ぶ確率が大きく増減するが，無関係の概念が正解のプロンプトに対しては影響がほとんど見られなかった．しかし全ての概念についてそのような傾向が見られたわけではなく，“knife” のように知識ニューロンの活性値の抑制に伴って紐づくプロンプトの正解を選ぶ確率が増加かつ活性値の増幅に伴って正解を選ぶ確率が減少するといった事例や，“organism” のように抑制・増幅に関わらず紐づくプロンプトの正解を選ぶ確率が減少する事例も見られた．

また，使用した全ての概念についての，正解を選ぶ確率の変化をヒストグラムに表したものを図 5，6 に示す．この図 5，6 から，知識ニューロンの活性値を編集した際，その知識ニューロンに紐づくプロンプトを解かせた場合には正解を選ぶ確率に対して影響が見られ，紐づかないプロンプトを解かせた場合にはほとんど影響がないことが確認できる．すなわち，多くの知識ニューロンは確かに紐づく概念についての知識がエンコードされているニューロンであり，知識が局所性を伴って Transformer の FF 層の中にエンコードされていることが確認された．

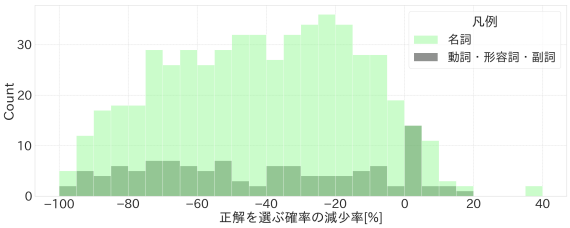


**図 7**　活性値を抑制してそのニューロンに紐づくプロンプトを予測した際の，品詞ごとの正解を選ぶ確率

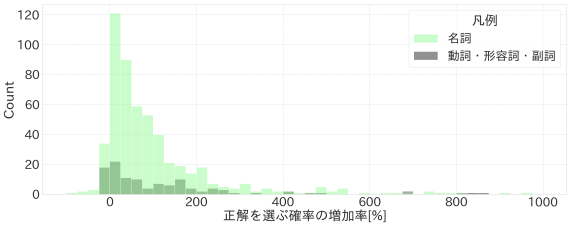


**図 8**　活性値を増幅してそのニューロンに紐づくプロンプトを予測した際の，品詞ごとの正解を選ぶ確率

### 4.2　概念の品詞によって，活性値の編集による影響に違いがあるか

図 7 に知識ニューロンの活性値を抑制してその知識ニューロンが紐づくプロンプトを解かせ，概念の品詞に応じて正解を選ぶ確率の変化を表したものを，図 8 に同様に活性値を増幅した場合の結果を示す．グラフの概形を比較すると，図 8 では概形に大きな違いは認められなかったものの，図 7 では名詞は –25 ~ –20% 付近にピークがあるのに対し，動詞・形容詞・副詞は 0 ~ 5% 付近が最も件数が多い．これは，名詞の方がより局所的な傾向を持つことを示しており，品詞によって知識の保存形態が異なる可能性が示唆された．

## 5　おわりに

本研究では，概念についての知識が Transformer の FF 層の中で局所的にエンコードされていることを確認した．ある概念についての知識が保存されていると考えられる知識ニューロンの活性値を編集した際，多くの場合その概念についてのプロンプトのみが解けなくなり，他の概念についてのプロンプトに対しては影響が殆ど見られないことも確認した．また，概念を品詞ごとに分類して影響の差異を調べた結果，品詞によって知識の保存形態が異なる可能性が示唆された．一方で，なぜ知識ニューロンの活性値を編集しても影響がほとんど見られない，もしくは想定とは逆の影響が観察される概念が存在するのか，ということは研究課題として残っている．

## 謝辞

本研究は，JST/CREST (JPMJCR20D2) および JSPS 科研費 (JP19H04425，JP21K17814) の助成を受けた．

## 参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)**, 2017.

[2] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 5484–5495.

[3] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. **arXiv:2104.08696**, 2021.

[4] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). **arXiv preprint arXiv:1606.08415**, 2016.

[5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In **Proceedings of the 34th International Conference on Machine Learning**, Vol. 70, pp. 3319–3328, 2017.

[6] A. H. Miller P. Lewis A. Bakhtin Y. Wu F. Petroni, T. Rocktäschel and S. Riedel. Language models as knowledge bases? In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 2463–2473.

[7] Hugging Face. Transformers, （2022-01 閲覧）. https://huggingface.co/docs/transformers/index.

[8] NLTK :: Natural Language Toolkit, （2022-01 閲覧）. https://www.nltk.org/.