大規模言語モデルのニューロン分析による効率的な多言語拡張

飯森栄治 [1,2]　谷中瞳 [1,2,3]
東京大学 [1]　理化学研究所 [2]　東北大学 [3]
{iimori-eiji, hyanaka}@is.s.u-tokyo.ac.jp

概要

大規模言語モデル (LLM) は低資源言語の性能に課題がある．LLM の多言語処理の内部機序の分析のために，特定の言語入力に対して集中的に発火する言語ニューロンが特定されてきたが，これを用いた応用研究は少ない．本研究では，LLM の多言語拡張において Mixture of Experts (MoE) アーキテクチャに注目し，低資源言語の言語ニューロンの分析結果にしたがって MoE の層ごとのエキスパート配置を適切に決定する **NeuronMoE** を提案する．

1　はじめに

大規模言語モデル (LLM) の課題の一つとして，学習データが十分に確保できない低資源言語では，高資源言語に比べて性能が低下しやすいことが挙げられる [1]．これに対する解決策の一つとして，事前学習済みモデルの元言語（高資源言語）から対象言語（低資源言語）へと Mixture of Experts (MoE) アーキテクチャを利用して拡張する手法が提案されてきた [2, 3, 4, 5]．特に層単位の配置戦略 [6]（LayerMoE）では，各層における言語間の Attention 層出力の類似度を測定し，類似度が低い層（言語間の処理が異なる層）には多くのエキスパートを，類似度が高い層（言語間の処理が似ている層）には少数のエキスパートを配置することで，パラメータ削減を実現している．

しかし，言語間類似度に基づくエキスパート配置手法には課題がある．この手法は Attention 層の出力の類似度のみを用いており，モデルパラメータの 3 分の 2 を占める MLP 層を考慮していない．そのため，MLP 層における言語固有の処理パターンを捉えられず，より効率的な配置の可能性を見逃している可能性がある．

一方，言語ニューロンに関する研究 [7, 8, 9] では，Attention 層と MLP 層の両方において個々のニューロンが言語知識を符号化し，層間で不均一な分布（初期層と後期層に集中，中間層で疎）を示すことが明らかになっている．しかし，これらの知見を活用した応用研究は限定的である．

本研究では，LLM の効率的な多言語能力の拡張に向けて，Transformer の全構成要素（Attention 層と MLP 層）における言語ニューロンの分布に基づいて MoE のエキスパート配置を決定する手法として NeuronMoE を提案する．

本研究では以下の 2 つの問題に取り組む．**(1) ニューロンレベルの分析は，Attention 層出力の類似度に基づく配置手法と比べて，より効率的なエキスパート配置を実現できるか．(2) 多言語拡張時に MoE エキスパート内で言語ニューロンはどのような分布になっていくのか．**

問い (1) に対し，LLaMA-3.2-3B [10] での実験により，ニューロン配置が異なる低資源言語（ギリシャ語，トルコ語，ハンガリー語）で平均 40%のパラメータ削減（47-50 個 vs 84 個のエキスパート）を NeuronMoE は達成しつつ，LayerMoE と同等の性能を実現した．

問い (2) に対する分析では，対象言語のエキスパートは高資源言語と類似した言語ニューロンの分布パターン (初期層と後期層への集中) をもつすることが明らかになった．

2　関連研究

2.1　Mixture-of-Experts による多言語拡張

事前学習済み LLM を追加言語へ拡張する際の主要な課題は，新しい言語知識を獲得する過程で，元の言語能力の破滅的忘却が起こる点にある．一方，MoE アーキテクチャは，疎なエキスパート活性化を通じて，元の言語の能力を保持しつつ新しい言語へ適応できる．

初期の MoE 研究では，全層に同一数のエキスパートを割り当てる均一配置が一般的であった [11, 12]．しかし，これらは主として計算効率に焦点を当てて

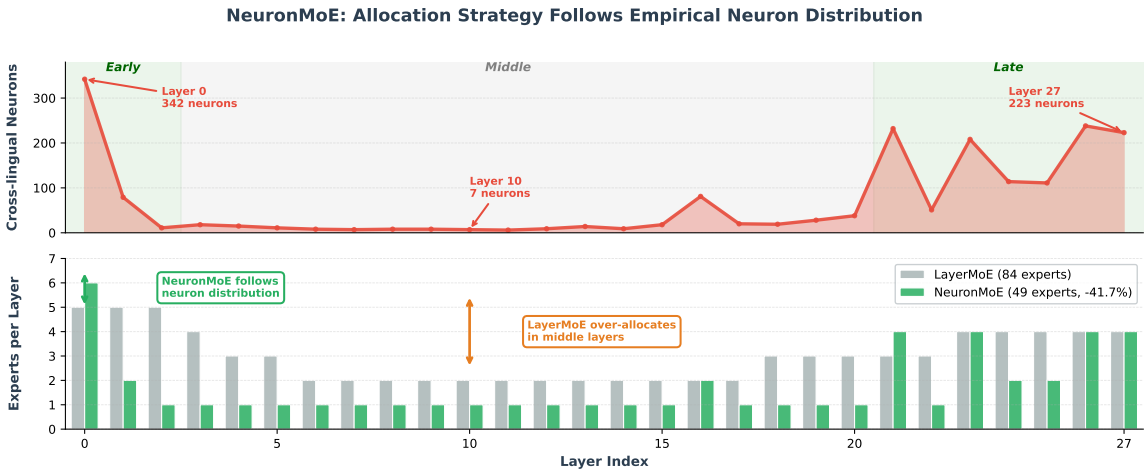

図 1　NeuronMoE の配置戦略の概要．**上：**言語間ニューロン多様性は層間で不均一な分布を示す (Layer 0 は 342 個，Layer 10 は 7 個のみ)．**下：**NeuronMoE はこの分布に従ってエキスパートを配置し (合計 49 個)，言語ニューロンが多い層に容量を集中させる．対照的に既存手法である LayerMoE は中間層に過剰配置する (合計 84 個)．

おり，層ごとの必要十分なエキスパート数な配置については検討されていなかった．[4] は MoE-LPR を提案し，言語のトークンの事前確率に基づくルーティングと 2 段階訓練フレームワークにより多言語拡張を実現した．同手法は元言語の破滅的忘却を効果的に緩和する一方で，全層で均一なエキスパート配置を採用しているため，パラメータ効率にはなお改善の余地が残る．

MoE-LPR を踏まえ，[6] は言語間類似度に基づく層単位のエキスパート配置 (LayerMoE) を導入した．具体的には，注意層の類似度に反比例する形でエキスパート数を割り当て，単一言語拡張および多言語拡張でパラメータ削減を達成した．

しかし，LayerMoE にはいくつかの課題が残されている．第一に，類似度計算が注意層のみに基づいており，MLP 層が考慮されていない．第二に，層レベルの類似度では，層内における詳細な言語処理の差異を十分に捉えられない．第三に，配置戦略は言語間の違いに関する間接的な指標に依存しており，ニューロンレベルの分析から得られる言語固有のニューロン分布といった直接的な測定値を活用できていない．

本研究では，Attention 層と MLP 層に対して言語ニューロンを分析し，エキスパート配置を決定すし，これらの課題に対処する．

2.2　多言語モデルにおける言語ニューロン

[7] は事前訓練モデルをニューロンレベルで分析し，少数のニューロンサブセットが特定の言語出力に大きく寄与することを示した．[8] は MLP 層における言語ニューロン識別のために，Language Activation Probability Entropy (LAPE) を提案した．[9] は，Attention 層と MLP 層の両方において特定言語に対して統計的に有意な活性化パターンを示すニューロンを言語ニューロンと定義した．ニューロンの特定手法が異なるものの，言語ニューロンが層間で不均一な分布を示すという共通の知見を得ている．具体的には，言語ニューロンは初期層と後期層に集中し，中間層では最小限となることが示されている．

3　手法

本手法は，言語ニューロンの分布を測定してエキスパート配置を決定することで，ニューロンレベル分析と MoE アーキテクチャ設計を結びつける．[9] の手法を採用し，Attention 層と MLP 層の両方を分析する．

3.1　言語ニューロン測定

[9] に従い，特定言語に対して統計的に有意な活性化パターンを示すニューロンを言語ニューロンと定義する．層 l のニューロン n と言語 ℓ について，多言語コーパス上での活性化値 $a_n^{(i)}$ を計算する ($y^{(i)}$ はサンプル i の言語ラベル)．

ニューロン n の言語 ℓ に対する言語特異性は，Average Precision (AP) を用いて測定される．AP は活性化値で降順ソートしたときに言語 ℓ のサンプルが

上位に集中する度合いを表す:

$$\mathrm{AP}(n,\ell) = \frac{1}{|\{i : y^{(i)} = \ell\}|} \sum_{k=1}^{N} \mathrm{Precision}(k) \cdot \mathbb{1}[y^{(\pi(k))} = \ell] \tag{1}$$

ここで π は活性化値の降順ソート，$\mathrm{Precision}(k)$ は上位 k 件中の言語 ℓ の割合である．AP が 1.0 に近いほど言語 ℓ に特異的なニューロンである．

各言語について，全層にわたり AP スコアが上位 1000 個の言語ニューロンを抽出する．層スコア $\mathcal{S}_l$ は，層 l における元言語と対象言語の言語ニューロンの和集合のサイズである．ギリシャ語拡張の場合，以下のように定式化される:

$$\mathcal{S}_l = \left| \bigcup_{\mathrm{lang} \in \{\mathrm{en},\ \mathrm{el}\}} N_{l,\mathrm{lang}} \right| \tag{2}$$

ここで $N_{l,\mathrm{lang}}$ は層 l における言語 lang の言語ニューロンの集合である．このスコアは，層が両言語を処理するために必要な言語ニューロンの総数を表す．同様の定式化がトルコ語 (en, tr) やハンガリー語 (en, hu) の拡張にも適用される．

3.2 エキスパート配置戦略

各層のエキスパート数 E_l は，ニューロン数 $\mathcal{S}_l$ を事前に定めた最小・最大エキスパート数の間で線形スケーリングして決定する:

$$E_l = E_{\min} + \mathrm{round}(\mathrm{norm}(\mathcal{S}_l) \times (E_{\max} - E_{\min})) \tag{3}$$

ここで，$\mathrm{norm}(\mathcal{S}_l) = (\mathcal{S}_l - \min_l \mathcal{S}_l)/(\max_l \mathcal{S}_l - \min_l \mathcal{S}_l)$ は全層にわたり正規化されたスコアである．本実験では $E_{\min} = 1$，$E_{\max} = 6$ と設定した．このアプローチは，言語ニューロン数が多い層にはより多くのエキスパートを配置し，言語ニューロン数が少ない層では最小限のパラメータ追加に抑える．

3.3 2 段階訓練プロセス

MoE-LPR フレームワーク [4] に従い，2 段階訓練アプローチを採用する．

Stage 1 (エキスパート初期化): 元の事前訓練モデルパラメータを凍結し，ニューロン配置戦略に従って各層に新しい MoE エキスパートを追加する．モデルはターゲット言語データで訓練し，これらのエキスパートを初期化する．この段階では元言語データを使用しないため，新規エキスパートはターゲット言語に特化した表現を学習する．各層のルーティング重みは一様分布で初期化される．

表 1 LLaMA-3.2-3B ギリシャ語拡張の性能比較.

		ARC		MMLU	
モデル	#Exp	EN	EL	EN	EL
Dense	-	51.11	31.93	56.45	41.17
LayerMoE	84	49.32	37.50	55.79	44.06
NeuronMoE	49	50.17	35.02	56.48	43.66

Stage 2 (ルーター訓練): 元言語の性能を保持するための少量の元言語データ（リプレイデータ）とターゲット言語データを混合してルーティング機構を訓練する．リプレイデータの比率は 1%未満に設定し，計算コストを最小限に抑えつつ破滅的忘却を防ぐ．この段階で，モデルは入力に応じて適切なエキスパートを選択する能力を獲得する．

4 実験設定

LayerMoE [6] との直接比較のため，同じ実験設定を用いた: 主要モデルとして LLaMA-3.2-3B(28 層) [10]，違うアーキテクチャのモデルとして Qwen-1.5-1.8B(24 層) [13] を使用した．訓練データには多言語コーパス CulturaX [14] から各言語 2B トークンを用いて，ギリシャ語 (EL)，トルコ語 (TR)，ハンガリー語 (HU) へモデルを拡張した．性能評価には多言語ベンチマーク (MMLU [15]，ARC Challenge [16]) を使用した．

全モデルは AdamW オプティマイザ，学習率 1e-4，バッチサイズ 32，15K ステップで訓練した．

5 結果

表 1 に主要な結果を示す．LLaMA-3.2-3B をギリシャ語に拡張する実験において，NeuronMoE は 41.7%のパラメータ削減 (49 個 vs 84 個のエキスパート) を達成し，ギリシャ語の性能を維持しつつ，英語の破滅的忘却を緩和した．具体的には，初期層 (層 0-2) と後期層 (層 24-27) には複数のエキスパート (2-6 個) を配置し，中間層 (層 3-23) には各層 1 個のエキスパートのみを配置した (詳細は付録表 4 参照)．これは，全層に均等に配置する LayerMoE とは対照的である．

元言語である英語と拡張対象の言語の性能のトレードオフはタスクに依存する．常識推論タスクである ARC Challenge では約 2%の性能劣化が見られるが，言語理解タスクである MMLU では劣化は 1%未満にとどまる．

Qwen-1.5-1.8B を用いた実験では，LLaMA と同様

表 2　トルコ語・ハンガリー語拡張結果.

言語	モデル	#Exp	ARC	MMLU
トルコ語:				
	Dense	-	33.56	42.35
	LayerMoE	84	36.55	43.51
	NeuronMoE	50	34.50	43.37
ハンガリー語:				
	Dense	-	34.85	43.68
	LayerMoE	84	39.73	44.86
	NeuronMoE	47	37.67	44.30

に 50.0%のパラメータ削減を達成したことから，提案手法はモデルのアーキテクチャによらず有効であることが示唆される．

トルコ語・ハンガリー語への拡張では，同一の戦略に基づき，47〜50 個のエキスパートを配置した．表 2 に示すように，NeuronMoE は LayerMoE と比較して ARC 性能の劣化を約 2%に抑えつつ，Dense ベースラインを大きく上回る性能を達成している．さらに，類型の異なる 3 言語において一貫した性能・効率のトレードオフが観察されたことは，本配置戦略が言語によらず有効であることを示唆する．

6　分析

訓練後の MoE モデルにおける言語ニューロン分布を分析し，NeuronMoE によるエキスパート配置の有効性を検証する．

6.1　MoE エキスパート内の言語ニューロン分布

疎な MoE アーキテクチャでは，各エキスパートがルーティングで選ばれた一部のトークンしか処理しないため，ニューロンの測定が難しい．そこで，本研究では [9] の手法を拡張し，各層でルーティング情報を追跡する．具体的には，サンプル s 内のトークンのうちエキスパート e に割り当てられたトークン集合 $T_e^{(s)}$ について，トークンレベルの活性化 $a_e^{(t)}$ を平均してサンプルレベルの表現 $h_e^{(s)}$ を得る．この表現に基づいて，エキスパート e 内のニューロン n の Average Precision (AP) スコアを計算し，ニューロンの活性化と言語ラベルの相関を測定する（詳細は付録 A.1 参照）．

表 3 に主要な層における高 AP ニューロン数を示す．層 0(6 個のエキスパート配置) では，expert_3 が 61 個 (0.31%)，expert_1 が 27 個 (0.14%) のギリシャ語ニューロンを持つ．層 27(4 個のエキスパート配置) では，ベースモデル部分が 179 個 (0.92%)，expert_2 が 405 個 (2.08%) のギリシャ語ニューロンを持ち，

表 3　エキスパートにおけるギリシャ語特化ニューロン数.

層	高 AP 数	比率 (%)
L0 expert_1	27	0.14
L0 expert_3	61	0.31
L26 expert_0	278	1.43
L26 expert_1	154	0.79
L27 base	179	0.92
L27 expert_2	405	2.08

特に expert_2 で多くの言語ニューロンを持つ．一方，層 26 では，expert_0 が 278 個 (1.43%)，expert_1 が 154 個 (0.79%) を持つ．単一エキスパートが配置された中間層 (層 3-20) では言語ニューロンがほとんど活性化せず，ほとんどの層で 10 個未満である．

6.2　配置戦略の一般化に関する考察

上記の観察は，なぜエキスパート配置戦略が言語・アーキテクチャ間で一般化するのかを説明する．本手法のエキスパート配置は元言語と追加言語の言語ニューロン分布のみに基づいて決定される．しかし訓練後の分析によると，対象言語であるギリシャ語も訓練を通じて，初期層と後期層に言語ニューロンが集中し，中間層では疎になるという元言語と同様の分布を獲得する．この結果は，Transformer が初期層で入力符号化，中間層で言語非依存の抽象推論，後期層で出力生成を行うという先行研究の知見 [8] と一致する．

したがって，ニューロン分布に基づいてエキスパートを配置することにより，対象言語の学習データが少ない状況でも，追加エキスパートが訓練されることにより，効率的に言語固有の処理パターンを捉えるのが可能となることを示唆する．

7　結論

本研究では，Attention 層と MLP 層における言語ニューロン分布に基づいて MoE エキスパートを配置する手法を提案した．3 言語 (ギリシャ語，トルコ語，ハンガリー語) での実験により，性能を維持しながら 40-50%のパラメータ削減を達成した．適切なエキスパート配置に基づく MoE によって言語ニューロンは初期・後期層に集中し，高資源言語に類似した分布パターンを形成することが示された．

謝辞

本研究は JST CREST JPMJCR2565，JST BOOST JPMJBY24H5 の支援を受けたものである．

参考文献

[1] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the nlp world. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 6282–6293, 2020.

[2] Shangjie Li, Xiangpeng Wei, Shaolin Zhu, Jun Xie, Baosong Yang, and Deyi Xiong. MMNMT: Modularizing multilingual neural machine translation with flexibly assembled MoE and dense blocks. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 4978–4990, Singapore, December 2023. Association for Computational Linguistics.

[3] Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 15913–15923, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[4] Hao Zhou, Zhijun Wang, Shujian Huang, Xin Huang, Xue Han, Junlan Feng, Chao Deng, Weihua Luo, and Jiajun Chen. Moe-lpr: Multilingual extension of large language models through mixture-of-experts with language priors routing. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 39, No. 24, pp. 26092–26100, Apr. 2025.

[5] Chong Li, Yingzhuo Deng, Jiajun Zhang, and Chengqing Zong. Group then scale: Dynamic mixture-of-experts multilingual language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 1730–1754, Vienna, Austria, July 2025. Association for Computational Linguistics.

[6] Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. Less, but better: Efficient multilingual expansion for LLMs via layer-wise mixture-of-experts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 17948–17963, Vienna, Austria, July 2025. Association for Computational Linguistics.

[7] Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4865–4880, Online, November 2020. Association for Computational Linguistics.

[8] Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 5701–5715, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 6919–6971, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024.

[11] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. **arXiv preprint arXiv:2006.16668**, 2020.

[12] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In **International conference on machine learning**, pp. 5547–5569. PMLR, 2022.

[13] Qwen Team. Introducing qwen1.5, February 2024.

[14] Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 4226–4237, Torino, Italia, May 2024. ELRA and ICCL.

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. **Proceedings of the International Conference on Learning Representations (ICLR)**, 2021.

[16] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. **arXiv:1803.05457v1**, 2018.

A　付録

A.1　MoE における言語ニューロン測定

疎な MoE アーキテクチャにおける言語ニューロンの測定手法を説明する．各エキスパートはルーティング機構により選ばれたトークンのサブセットのみを処理するため，標準的な手法をそのまま適用できない．

ルーティング情報の追跡: 各層 l において，ルーティング情報 $R_l = \{\text{expert\_ids}, \text{routing\_weights}\}$ を記録する．サンプル s の各トークン t に対して，どのエキスパートに割り当てられたかを追跡する．

トークンレベルからサンプルレベルへの集約: エキスパート e がサンプル s 内で処理したトークン集合を $T_e^{(s)}$ とする．エキスパート e のサンプル s に対する活性化表現 $h_e^{(s)}$ を以下のように計算する:

$$h_e^{(s)} = \frac{1}{|T_e^{(s)}|} \sum_{t \in T_e^{(s)}} a_e^{(t)} \tag{4}$$

ここで $a_e^{(t)}$ はエキスパート e のトークン t に対する活性化ベクトルである．

このサンプルレベルの表現 $h_e^{(s)}$ を用いて，本文で述べた AP スコアを計算することで，エキスパート e 内の各ニューロン n の言語 ℓ に対する言語特異性を測定できる．

A.2　エキスパート配置の詳細

表 4 に LayerMoE と NeuronMoE の層ごとのエキスパート配置を示す．LayerMoE は層間類似度に基づいて比較的均一にエキスパートを配置するのに対し，NeuronMoE は言語ニューロン分布に基づいて初期層と後期層に重点的にエキスパートを配置する．この配置の違いにより，NeuronMoE は合計 49 個のエキスパート (LayerMoE は 84 個) で同等の性能を達成している．

表 4　LLaMA-3.2-3B ギリシャ語拡張における層ごとのエキスパート配置比較．各セルの数字はその層範囲内の各層に配置されたエキスパート数を示す．

層	LayerMoE	NeuronMoE
0-2	5,5,5	6,2,1
3-10	4,3,3,2,2,2,2,2	1×8
11-18	2×7,3	1×7,2
19-27	3×4,4×5	1,1,4,1,4,2,2,4,4
合計	**84**	**49**