

## A Parallel Neuron Detection Algorithm

Inspired by [Zhao et al. \(2024a\)](#); [Wang et al. \(2025a\)](#), the neuron detection method in Equation 2 can be done parallel. While Equation 2 considers the change in the final output embedding, the parallel methods described here efficiently calculate the change in the output of the *specific layer containing the neuron* when that neuron is deactivated. This layer-wise impact serves as a proxy or component for the overall impact.

In this context, let  $X \in \mathbb{R}^{l \times d_{model}}$  be the input hidden states to a given layer, where  $l$  is the sequence length and  $d_{model}$  is the hidden dimension of the model. For a neuron  $\mathcal{N}$  within this layer, its impact is measured as  $\|f(X; \Theta) - f(X; \Theta_{\ominus \mathcal{N}})\|_2$ , where  $f(X; \Theta)$  is the layer’s output with parameters  $\Theta$ , and  $f(X; \Theta_{\ominus \mathcal{N}})$  is the output when neuron  $\mathcal{N}$  (a specific row or column in  $\Theta$ ) is deactivated (its parameters set to zero).

### A.1 Feed-Forward Network (FFN) Neurons

A standard FFN layer in modern transformer models can be expressed as:

$$\text{FFN}(X) = (\text{SiLU}(XW_{gate}) \odot (XW_{up})) W_{down} \quad (7)$$

where  $X \in \mathbb{R}^{l \times d_{model}}$  is the input to the FFN layer,  $W_{gate}, W_{up} \in \mathbb{R}^{d_{model} \times d_{inter}}$ , and  $W_{down} \in \mathbb{R}^{d_{inter} \times d_{model}}$ . Here,  $d_{inter}$  is the intermediate dimension of the FFN. The symbol  $\odot$  denotes element-wise multiplication. Let  $H_{act} = \text{SiLU}(XW_{gate}) \odot (XW_{up})$  be the intermediate activation matrix,  $H_{act} \in \mathbb{R}^{l \times d_{inter}}$ . Thus, the FFN output is  $Y_{FFN} = H_{act}W_{down} \in \mathbb{R}^{l \times d_{model}}$ .

We consider a neuron  $\mathcal{N}_{inter,k}$  to be associated with the  $k$ -th dimension of the intermediate representation  $H_{act}$ . Deactivating such a neuron means that the  $k$ -th column of  $H_{act}$ , denoted  $H_{act}[:, k]$ , is effectively zeroed out before the multiplication with  $W_{down}$ . This deactivation corresponds to zeroing out the parameters that produce this  $k$ -th intermediate feature, e.g., the  $k$ -th column of  $W_{up}$  (i.e., neuron  $\mathcal{N}$  is  $W_{up}[:, k]$ ) and  $W_{gate}$ , or by zeroing out parameters that read from it, e.g., the  $k$ -th row of  $W_{down}$  (i.e., neuron  $\mathcal{N}$  is  $W_{down}[k, :]$ ).

Let  $Y_{FFN, \ominus \mathcal{N}_{inter,k}}$  be the output when the  $k$ -th intermediate neuron is deactivated. The change in the layer’s output is:

$$\Delta Y_{FFN,k} = Y_{FFN} - Y_{FFN, \ominus \mathcal{N}_{inter,k}}$$

If  $H'_{act}$  is  $H_{act}$  with its  $k$ -th column zeroed, then  $Y_{FFN, \ominus \mathcal{N}_{inter,k}} = H'_{act}W_{down}$ . So,

$$\Delta Y_{FFN,k} = (H_{act} - H'_{act})W_{down}$$

The matrix  $(H_{act} - H'_{act})$  is zero everywhere except for its  $k$ -th column, which consists of the elements  $H_{act}[:, k]$ . Let this difference matrix be  $\delta H_k$ . Then  $\Delta Y_{FFN,k} = \delta H_k W_{down}$ . This resulting  $l \times d_{model}$  matrix is formed by the outer product of the  $k$ -th column of  $H_{act}$  and the  $k$ -th row of  $W_{down}$ :

$$\Delta Y_{FFN,k} = H_{act}[:, k](W_{down})_{k,:}$$

The impact of the  $k$ -th intermediate FFN neuron is then the L2 norm of this change:

$$\|\Delta Y_{FFN,k}\|_2 = \|H_{act}[:, k](W_{down})_{k,:}\|_2 \quad (8)$$

This computation can be performed in parallel for all  $k \in \{1, \dots, d_{inter}\}$  to obtain the impact of all intermediate neurons in the FFN layer.

### A.2 Self-Attention Network Neurons

The output of a self-attention layer (for simplicity, we describe a single attention head; multi-head attention involves similar computations per head) can be given by:

$$Y_{Attn} = \text{Softmax} \left( \frac{(XW_Q)(XW_K)^T}{\sqrt{d_k}} \right) (XW_V) \quad (9)$$

Let  $Q = XW_Q \in \mathbb{R}^{l \times d_{attn}}$ ,  $K = XW_K \in \mathbb{R}^{l \times d_{attn}}$ , and  $V = XW_V \in \mathbb{R}^{l \times d_{attn}}$ , where  $d_{attn}$  is the dimension of queries, keys, and values for the attention mechanism.  $d_k$  is the scaling factor, typically the dimension of the key/query vectors (e.g.,  $d_k = d_{attn}$ ). Let  $A = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) \in \mathbb{R}^{l \times l}$ . The layer output is  $Y_{Attn} = AV \in \mathbb{R}^{l \times d_{attn}}$ . (An additional output projection  $W_O$  might follow this, which would be multiplied subsequently).

### 616 A.2.1 Neurons in $W_V$

617 Consider a neuron  $\mathcal{N}_{V,k}$  defined as the  $k$ -th column of  $W_V$ , i.e.,  $W_V[:, k]$ . Deactivating this neuron  
 618 sets  $W_V[:, k]$  to zero, which in turn makes the  $k$ -th column of  $V = XW_V$ , denoted  $V[:, k]$ , zero. Let  
 619  $V'$  be the matrix  $V$  with its  $k$ -th column zeroed. The change in the layer's output is:

$$\Delta Y_{Attn,k}^{(V)} = AV - AV' = A(V - V')$$

620 The matrix  $(V - V')$  is zero everywhere except for its  $k$ -th column, which is  $V[:, k]$ . Let this  
 621 difference matrix be  $\delta V_k$ . Then  $\Delta Y_{Attn,k}^{(V)} = A(\delta V_k)$ . This  $l \times d_{attn}$  matrix has  $AV[:, k]$  (the matrix  
 622  $A$  multiplied by the vector  $V[:, k]$ ) as its  $k$ -th column, and zeros in other columns. The impact of  
 623 neuron  $\mathcal{N}_{V,k}$  is:

$$\left\| \Delta Y_{Attn,k}^{(V)} \right\|_2 = \|AV[:, k]\|_2 \quad (10)$$

624 where the norm is effectively taken over the  $l \times 1$  vector  $AV[:, k]$  that forms the  $k$ -th column of the  
 625 change matrix. This can be calculated in parallel for all  $k \in \{1, \dots, d_{attn}\}$ .

### 626 A.2.2 Neurons in $W_Q$

627 Consider a neuron  $\mathcal{N}_{Q,k}$  defined as the  $k$ -th column of  $W_Q$ , i.e.,  $W_Q[:, k]$ . Deactivating this neuron  
 628 sets  $W_Q[:, k]$  to zero. This makes the  $k$ -th column of  $Q = XW_Q$ , denoted  $Q[:, k]$ , zero. Let  $Q'$  be the  
 629 matrix  $Q$  with its  $k$ -th column zeroed. The original unnormalized attention scores are  $S_{raw} = \frac{QK^T}{\sqrt{d_k}}$ .  
 630 The new unnormalized attention scores with  $\mathcal{N}_{Q,k}$  deactivated are  $S'_{raw} = \frac{Q'K^T}{\sqrt{d_k}}$ . The change in  
 631 the unnormalized scores due to deactivating  $\mathcal{N}_{Q,k}$  is  $\Delta S_{raw,k} = S_{raw} - S'_{raw} = \frac{(Q-Q')K^T}{\sqrt{d_k}}$ . The  
 632 matrix  $(Q - Q')$  is zero everywhere except for its  $k$ -th column, which is  $Q[:, k]$ . Thus,

$$\Delta S_{raw,k} = \frac{(Q[:, k])(K[:, k])^T}{\sqrt{d_k}}$$

633 This  $l \times l$  matrix represents the change in raw attention scores attributable to the interaction involving  
 634 the  $k$ -th column of  $Q$  and the  $k$ -th column of  $K$ .

635 Let  $A_{orig} = \text{softmax}(S_{raw})$  be the original attention probability matrix. Let  $A_{\ominus \mathcal{N}_{Q,k}} =$   
 636  $\text{softmax}(S_{raw} - \Delta S_{raw,k})$  be the attention probability matrix when neuron  $\mathcal{N}_{Q,k}$  is deactivated. The  
 637 change in the layer's output is:

$$\Delta Y_{Attn,k}^{(Q)} = A_{orig}V - A_{\ominus \mathcal{N}_{Q,k}}V = (A_{orig} - A_{\ominus \mathcal{N}_{Q,k}})V$$

638 The impact of neuron  $\mathcal{N}_{Q,k}$  is:

$$\left\| \Delta Y_{Attn,k}^{(Q)} \right\|_2 = \|(A_{orig} - A_{\ominus \mathcal{N}_{Q,k}})V\|_2 \quad (11)$$

639 To calculate this efficiently for all  $k \in \{1, \dots, d_{attn}\}$  (corresponding to each column neuron in  $W_Q$ ):

- 640 1. Compute the original  $S_{raw} = \frac{QK^T}{\sqrt{d_k}}$  and  $A_{orig} = \text{softmax}(S_{raw})$ .
- 641 2. For each  $k$ , compute the specific change term  $\Delta S_{raw,k} = \frac{Q[:, k](K[:, k])^T}{\sqrt{d_k}}$ . This step can be  
 642 parallelized by constructing a tensor  $\Delta S_{raw} \in \mathbb{R}^{d_{attn} \times l \times l}$  where the slice  $\Delta S_{raw}[k, :, :] =$   
 643  $\Delta S_{raw,k}$ .
- 644 3. For each  $k$ , compute the adjusted scores  $S_{adjusted,k} = S_{raw} - \Delta S_{raw}[k, :, :]$ .
- 645 4. For each  $k$ , compute  $A_{\ominus \mathcal{N}_{Q,k}} = \text{softmax}(S_{adjusted,k})$ .
- 646 5. For each  $k$ , calculate the impact norm  $\|(A_{orig} - A_{\ominus \mathcal{N}_{Q,k}})V\|_2$ .

### 647 A.2.3 Neurons in $W_K$

648 The impact of deactivating a neuron  $\mathcal{N}_{K,k}$  (the  $k$ -th column of  $W_K$ ) is calculated symmetrically to  
 649 that of  $\mathcal{N}_{Q,k}$ . The same change term  $\Delta S_{raw,k} = \frac{Q[:, k](K[:, k])^T}{\sqrt{d_k}}$  is used, reflecting the idea that this  
 650 term captures the interaction component associated with the  $k$ -th features of both  $Q$  and  $K$ . The  
 651 procedure then follows steps 3-5 as outlined for  $W_Q$  neurons, using this  $\Delta S_{raw,k}$  to find the adjusted  
 652 attention matrix and the resulting impact.

## B Neuron Detection Corpus

This section provides additional details regarding the corpus used for neuron detection, as mentioned in the main text. Our methodology relies on the OSCAR corpus for both identifying language-related neurons through activation patterns and quantifying their functional contribution via perplexity changes upon deactivation.

### B.1 OSCAR Corpus

The OSCAR (Open Super-large Crawled Aggregated coRpus) corpus (Abadji et al., 2022) is a massive multilingual collection of texts obtained by language classification and filtering of the Common Crawl dataset. Common Crawl is a publicly available web crawl spanning petabytes of data. OSCAR further processes this raw data to produce monolingual corpora across a wide range of languages, making it a valuable resource for training large language models and conducting cross-lingual research.

Key characteristics of the OSCAR corpus include:

- **Large Scale:** It contains hundreds of gigabytes to terabytes of text data for many languages.
- **Multilingual Coverage:** It supports a vast number of languages, facilitating studies that require diverse linguistic data.
- **Data Cleaning:** Efforts are made to clean and filter the crawled data, though the quality can vary depending on the language and the nature of web content.
- **Accessibility:** OSCAR is publicly available, promoting reproducibility and broader research in NLP.

For our study, we sample 1000 sentences for each target language from its respective monolingual section within the OSCAR corpus. This sampled data serves as the basis for analyzing neuron activations and evaluating perplexity changes. The diversity and scale of OSCAR help in capturing a wide array of linguistic phenomena necessary for robustly identifying language-specific neural correlates.

### B.2 Illustration of Sample Sentences

To provide a concrete illustration of the data used, Table 2 presents conceptual example sentences from the OSCAR corpus for the five languages central to our analysis: English (en), Chinese (zh), Swahili (sw), German (de), and French (fr).

Table 2: Illustrative sample sentences from the OSCAR corpus for the selected languages. These are conceptual examples, as actual sentences are randomly sampled.

Language	Conceptual Example
English (en)	The quick brown fox jumps over the lazy dog.
Chinese (zh)	敏捷的棕色狐狸跳过了懒惰的狗。
Swahili (sw)	Mbweha mwepesi wa kahawia anaruka juu ya mbwa mvivu.
German (de)	Der schnelle braune Fuchs springt über den faulen Hund.
French (fr)	Le renard brun rapide saute par-dessus le chien paresseux.

The sentences sampled for each language are then further used to observe which neurons are consistently activated during processing. A similar set of sentences is then used to measure the perplexity of the model when specific neurons or sets of neurons are deactivated, thereby quantifying their functional importance to that language.