We then incorporate this into the standard SAE loss $\mathcal{L}_{SAE}$, weighted by λ_2:

$$\mathcal{L}_{SAE} = MSE(a_i, a_i^{SAE}) + \lambda_1 \cdot \|a_i^{SAE}\|_1 + \lambda_2 \cdot \mathcal{L}_N, \quad (5)$$

where MSE is the mean-squared error, a_i denotes the MLP activations of L_i and a_i^{SAE} denotes the reconstructed activations from the SAE.

To update the average embedding $h_{i,j,avg}$ for a neuron $N_{i,j}^{SAE}$ across training, we use a momentum-based update:

$$h_{i,j,avg} = m \cdot h_{i,j,avg} + (1 - m) \cdot h_{i,b}, \quad (6)$$

where m controls the balance between the existing embedding and the new embedding. This allows information from earlier embeddings to decay, putting more weight on the more recent embeddings which should better reflect what the neuron is learning to respond to. We set m to 0.9 in our experiments.

Intuitively, this new loss term pushes each neuron to respond to a single feature, as the distance between the combined neuron embedding and the current neuron embedding will increase if the combined neuron embedding contains multiple features.

4. Results

4.1. Feature Clusters

4.1.1. Experimental Setup

We show that neuron embeddings can effectively capture semantic similarity between a neuron's dataset examples by applying feature clustering to GPT2-small. We processed 10,000 input examples from OpenWebText (Gokaslan & Cohen, 2019) with the model and retrieved the model's internal activations using the TransformerLens library (Nanda & Bloom, 2022). For each neuron, we collected any example that induced at least 75% of the maximum observed activation of the neuron [2], up to a maximum of 100 examples per neuron. We chose to keep the first 100 examples over this activation threshold, rather than the top 100 most activating examples, to better capture the diversity of behaviours that occur below maximal activation. A basic implementation of this process took a few hours to run on a single GPU, but could likely be made dramatically more efficient with more engineering effort, to enable it to scale to larger models.

We then applied feature clustering to each neuron to group the dataset examples into their distinct features. We analyse some examples of neurons to demonstrate the efficacy of the technique, and also provide a UI for navigating the full model [3]. The UI additionally provides insight into which tokens were important for neuron activation using the method from (Foote et al., 2023), as well as links to similar features from other neurons identified via Nearest Neighbour search on the embeddings of the central example in each feature cluster.

4.1.2. Neuron Examples

Figure 2 shows the results of applying clustering to a neuron in the 7th layer (of 12) of the model. The dendrogram shows the results of the hierarchical clustering of the neuron embeddings, where each branch point shows the average distance between the embeddings in the two clusters (where a cluster contains one or more points). The examples separate into six clusters (colour and numerically coded), with snippets of the dataset examples in each cluster shown. The highlighting indicates the strength of neuron activation on each token, and the brightest token is the key token which we use to produce the neuron embedding. Note we also show the following five tokens after the key token for context, but these do not influence the neuron embedding as the model is auto-regressive.

The clusters each contain a distinct behaviour, and successfully group examples that represent the same concept with different wording. The hierarchy captures similarity at multiple levels, with highly similar examples with the same key token clustering first, with these sub-clusters then merging into the full clusters. Additionally, similar but distinct clusters, such as the red and green cluster which both relate to numbers of events, also appear closer together in the hierarchy.

This phenomenon, where a neuron has multiple related but distinct behaviours, is quite common in GPT2-small, and may be related to feature splitting (Bricken et al., 2023), where a neuron in a Sparse Auto-Encoder (SAE) represents multiple closely related features, which then split into increasingly fine-grained features as the size of the Auto-Encoder is scaled up. Applying feature clustering to neurons in SAEs may be able to identify which neurons contain multiple sub-features which might split after scaling up, which could be useful for measuring the prevalence of these neurons and better understanding feature splitting more generally.

Figure 3 shows an example of a different class of neuron, where there is a common primary behaviour and a rarer secondary behaviour. In this case, dataset examples for the primary behaviour (orange) outnumber those for the rare

[2] Maximum activation was obtained from Neuroscope (Nanda, 2022), which measured neuron activation on a much larger dataset

[3] https://feature-clusters.streamlit.app/