How many of a neuron's weights does its explanation need?

Will Sackfield, Open Interpretability · https://open-interpretability.com/research/pruning-keep

Abstract

Magnitude pruning removes most of a network's weights for a small loss, which suggests that a unit's behaviour rests on a few large weights. This paper tests that claim on single neurons whose explanations have passed an adversarial review. For each of the 127 VERIFIED neuron poles of Pythia-70m, we zeroed the smallest 50%, 75% or 90% of the neuron's input weights, with no retraining. Then we ran each explanation's own probes again. As a null, we zeroed the same number of weights at random. With magnitude pruning, 94 explanations keep at 50%, 73 at 75% and 63 at 90%. With random pruning, 73, 42 and 17 keep. On the poles where the two methods disagree, magnitude pruning wins 30 to 9, 37 to 6 and 49 to 3 (exact McNemar $p = 0.001$, 2×10^{-6} and 10^{-11}). Half of the explanations still pass with a tenth of the input weights left, although each pole's activations then correlate with their unpruned form at a median of only 0.54. Whether an explanation keeps is barely related to that correlation. Most failures are on specificity: a pruned neuron starts to fire on the near-misses that its explanation excludes. So the part of a neuron that an explanation describes rests on its largest input weights, and much of the rest of its activity does not.

Status: Preprint. Work left: #6164 (GPT-2, Qwen2.5-0.5B, the output weights and more random seeds).

1 Introduction

An explanation of a neuron says when the neuron fires. It does not say which of the neuron's weights make it fire that way. If the explanation rests on a few large weights, two things follow. The explanation is a description of a small part of the network, which can be found from the weights alone. And a change to the model that leaves those weights alone should leave the explanation alone too, which matters for any method that edits weights.

Pruning gives a direct test. PackNet zeroes 50% to 75% of a network's weights by magnitude, and the network keeps most of its accuracy after a short retraining (Mallya & Lazebnik, 2018). That result concerns a whole network and its task loss. It does not say whether a single unit keeps the function that someone has described in words, and PackNet's retraining hides how much the unit changed before it recovered.

This paper applies the pruning to one neuron at a time, with no retraining, and asks whether the neuron's VERIFIED explanation still passes. The registry holds 127 such explanations for Pythia-70m (Biderman et al., 2023), each with the probes that an adversarial reviewer wrote to break it.

The paper's contributions are these:

- A measurement of how many VERIFIED explanations keep when 50%, 75% or 90% of their neuron's input weights are zeroed by magnitude.
- A random-pruning null at the same levels, with paired tests, which shows that weight size, not weight count, decides what an explanation needs.
- Evidence that an explanation can keep while most of the neuron's activity changes, and that its failures are mostly failures of specificity.
- The rows for each pole, level and method, so that other analyses can use the same outcomes.

The paper bears on research question 1, *Are native MLP neurons clean?* (research questions). If a clean explanation rests on a few weights, the clean part of a neuron is small and can be located.

2 Related work

Magnitude pruning. Mallya and Lazebnik (2018) prune 50% or 75% of the weights in each layer of VGG-16 by lowest absolute magnitude, in one shot. After a short retraining, the top-1 error on ImageNet rises from 28.42 to 29.33 at 50% and to 30.87 at 75%. The freed weights then learn new tasks while the old weights stay fixed. Their pruning is per layer and their measure is the task's accuracy. Ours is per neuron, with no retraining, and our measure is whether an explanation of that neuron still passes its probes. Their review page in this repo proposed the present test.

Explanations after a change to the model. An earlier analysis in this repo ran the same keep test after instruction tuning (`docs/analyses/base-instruct.md`). On Qwen2.5-0.5B, 252 of 311 VERIFIED explanations keep on the instruct model, and the activation correlation does not separate the poles that keep from those that fail. A companion paper tests whether a weight-importance score predicts which explanations keep under instruction tuning, and finds that it does not (`papers/fisher-keep/`). Pruning is a cruder change than tuning, but it is fully controlled: the change touches one neuron's input weights, and nothing else.

3 Method

Model and units. The model is Pythia-70m, which has 6 layers of 2,048 MLP neurons (Biderman et al., 2023). A *pole* is one sign of a neuron's activation, positive or negative. A pole is *VERIFIED* when its natural-language explanation passed the registry's adversarial review. In that review, a reviewer wrote 20 hard negatives (near-misses that the explanation says should not fire) and 10 holdouts (new texts that it says should fire). The explanation passed a specificity gate of 0.80 and a sensitivity gate of 0.60. The registry holds 127 VERIFIED poles of Pythia-70m in all six layers: 118 positive and 9 negative.

Pruning. A neuron's input weights are its row of the MLP's first projection: 512 weights, one for each dimension of the residual stream. At a level f, we zero $\mathrm{round}(512f)$ of them, for $f = 0.5$, 0.75 and 0.9. *Magnitude* pruning zeroes the smallest $|w|$ first. *Random* pruning zeroes the same number, chosen at random with a seed for each neuron. The bias stays. There is no retraining, and the script saves no weights. Zeroing a neuron's own

input weights changes only that neuron at its layer, so we prune all the poles of one layer together and read them in one pass.

The keep test. This is the test of the base-instruct analysis. Each pole's own probes run again on the pruned model. The activation line is the record's line times the ratio of the pole's maximum after pruning to its maximum before. Both maxima come from the same 20,000 tokens of the registry's scan corpus. So "active" means the same share of the pole's own range before and after. An explanation *keeps* when both gates still pass. All 127 pass on the unpruned model, so all 127 enter every comparison.

Activation change. For each pole, we also give Pearson's r between its activations before and after pruning, over the same 20,000 tokens, with each window's first token left out.

Tests. The keep share has a 95% Wilson interval. An exact McNemar test compares magnitude and random pruning on the same poles: the two-sided binomial test of the poles that keep under one method and not the other.

Motion. Figure 2 animates the change of each pole's correlation from one pruning level to the next. We measured only the three levels, and the frames show nothing between them.

4 Does an explanation survive magnitude pruning?

Most explanations survive half of their weights going, and half survive nine tenths going. Magnitude pruning keeps 94 of 127 explanations at 50% (74%, 95% interval 66–81%), 73 at 75% (57%, 49–66%) and 63 at 90% (50%, 41–58%) (Figure 1, Table 1). The 64 that fail at 90% are not one kind. Layer 0's explanations keep most often, 17 of 22 at 90%. Layers 1 and 2 lose the most, with 9 of 27 and 5 of 16 keeping. The negative poles keep at about the same rate as the positive ones, 3 of 9 against 60 of 118, but they are too few to compare.

Magnitude pruning keeps most explanations, and random pruning of as many weights does not

Each mark is the share of Pythia-70m's 127 VERIFIED explanations that still pass their probes, with a 95% interval. With 90% of each neuron's input weights zeroed: 63 keep by magnitude, 17 at random.

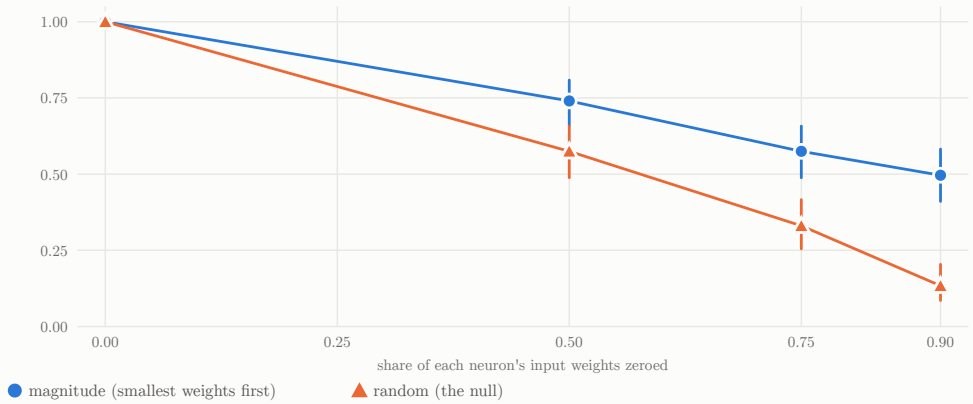

Figure 1. The share of Pythia-70m's 127 VERIFIED explanations that still pass their probes after the stated share of each neuron's input weights is zeroed. Circles are magnitude pruning, and triangles are random pruning of the same number of weights. The bars are 95% Wilson intervals.

Table 1. The explanations that keep, for each level and method, out of 127, with each pole's median correlation with its unpruned activations.

Zeroed	Magnitude: keep	95% interval	Median r	Random: keep	95% interval	Median r
50%	94 (74%)	66–81%	0.94	73 (57%)	49–66%	0.63
75%	73 (57%)	49–66%	0.78	42 (33%)	25–42%	0.39
90%	63 (50%)	41–58%	0.54	17 (13%)	9–20%	0.22

The keep share does not fall in step for every pole. 16 explanations keep at 90% but fail at 50% or 75%. The test reads a pole at a line scaled by its maximum, and each explanation has 30 probes. So a pole whose pass rates sit near a gate can cross it in either direction from one level to the next.

5 Do the weights that matter have to be the large ones?

They do. Random pruning of the same number of weights keeps far fewer explanations at every level: 73, 42 and 17 (Table 1). The gap grows with the level. On the poles where the two methods disagree, magnitude pruning keeps the explanation and random pruning does not in 30, 37 and 49 cases, and the reverse holds in 9, 6 and 3 cases. The exact McNemar p-values are 0.001, 1.6×10^{-6} and 1.0×10^{-11}. So what an explanation needs is the neuron's largest input weights, not merely some share of its weights.

6 Does the neuron's activity survive with its explanation?

Much of it does not. With magnitude pruning, each pole's activations correlate with their unpruned form at a median of 0.94 at 50%, 0.78 at 75% and 0.54 at 90% (Figure 2). Random pruning lowers the correlation faster, to 0.63, 0.39 and 0.22. At 90% magnitude pruning, half of the explanations keep, yet the typical pole's activations share less than a third of their variance with their unpruned form.

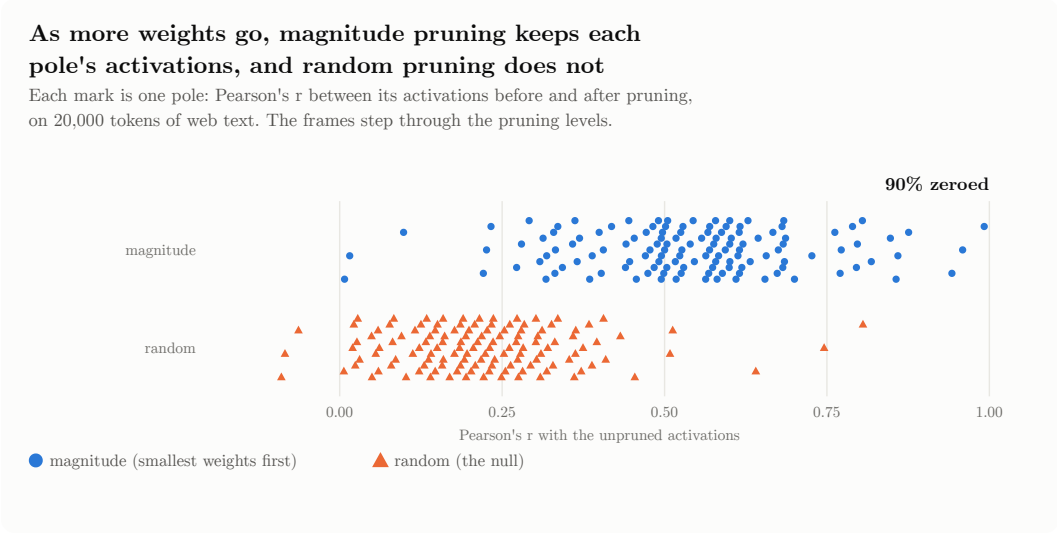

Figure 2. Each mark is one pole: Pearson's r between its activations before and after pruning, on 20,000 tokens of web text. Circles are magnitude pruning, and triangles random pruning. The frames step through the three levels and rest on 90%.

The correlation does not tell which explanations keep. At 90% magnitude pruning, the median correlation is 0.546 for the poles that keep and 0.542 for those that fail. The same holds at the other levels: 0.948 against 0.914 at 50%, and 0.783 against 0.777 at 75%. The correlation covers every token, most of which sit far below the pole's activation line. The probes test the top of the range, which the largest weights carry.

The failures are mostly failures of specificity. At 90% magnitude pruning, 37 of the 64 failures miss only the specificity gate, 23 only the sensitivity gate, and 4 both. At 50%, the split is 27, 4 and 2. A pruned neuron therefore starts to fire on the near-misses that its explanation excludes before it stops firing on what its explanation describes.

7 Discussion

A VERIFIED explanation of a Pythia-70m neuron rests on the neuron's largest input weights. Half of the explanations survive with a tenth of those weights, and random pruning of the same count destroys most of them. This agrees with the picture behind magnitude pruning, in which a few large weights carry a unit's function (Mallya & Lazebnik, 2018). We find it at the level of a single unit's described function, with no retraining afterwards.

The explanation and the neuron's full activity come apart under pruning. The largest weights keep the strong, selective responses that the probes test. The many small weights shape the rest of the activation distribution, which the correlation measures and the explanation does not describe. This suggests that an explanation describes the top of a neuron's range, and that this part is sparse in the weights. It also suggests a caution for comparing two models by activation correlation alone: a low correlation can hide an explanation that still holds.

The specificity failures say what the small weights do. When they go, a neuron fires on near-misses. So the small weights help to suppress the cases that sit near the explanation's boundary. An explanation that keeps at 90% therefore names a feature that the large weights detect on their own. An explanation that fails at 50% names one that needs the small weights to reject its near-misses.

For editing models, the result suggests a simple rule: a change that leaves a neuron's largest input weights alone will often leave its explanation alone. It does not show that such a change leaves the neuron's downstream effect alone, because the output weights and the neuron's full activity both matter there.

8 Limitations

- **One small model.** Pythia-70m has 2,048 neurons per layer. A larger model, with wider rows and more superposition, could need a larger share of its weights.
- **Input weights only, and no retraining.** We prune one neuron's input row, not the whole network, and we do not retrain. PackNet's setting differs in both ways.
- **One random draw per neuron.** The null uses one seeded draw for each neuron and level. Over 127 poles the counts are stable, but a single pole's random outcome is one

sample.

- **30 probes per explanation.** A pass rate near a gate can cross it by chance, which is why 16 explanations keep at 90% but fail at a lower level.
- **A rescaled line.** The keep test rescales the activation line by the change in the pole's maximum. So the test asks whether the pattern of the top of the range survives, not whether its absolute size does.
- **The registry's poles.** The 127 poles are those that the registry has VERIFIED, not a random sample of the model's neurons.

9 Conclusion

Half of Pythia-70m's VERIFIED explanations survive when 90% of their neuron's input weights are zeroed by magnitude, against 13% when the same number are zeroed at random. An explanation rests on a neuron's largest input weights, and it can hold while most of the neuron's activity changes.

References

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S. S., Raff, E., Skowron, A., Sutawika, L., & van der Wal, O. (2023). *Pythia: A suite for analyzing large language models across training and scaling.* In Proceedings of the 40th International Conference on Machine Learning (PMLR 202, pp. 2397–2430). PMLR.

Mallya, A., & Lazebnik, S. (2018). *PackNet: Adding multiple tasks to a single network by iterative pruning.* CVPR 2018. arXiv:1711.05769.

Appendices

A. Reproduce

The script reads each VERIFIED pole of Pythia-70m from `wiki/`, in float32 on the CPU. The random null uses seed 5122 (`SEED` in `openinterpretability/interference/pruning.py`), with one draw for each layer, neuron and matrix. The run took about 7 minutes on a laptop CPU.

```
uv run python scripts/pruning_keep.py --device cpu --out papers/pruning-keep/results.jsonl
uv run python scripts/paper_figures.py
```

`results.jsonl` holds a `summary` row and one `level` row for each level and method. It also holds one `pole` row for each pole, level and method, with the unpruned readings as method `none`. Each `pole` row gives the pole's pass rates, the gates it failed, whether it keeps, its scale ratio and its correlation. The poles that a run selects depend on the registry on the day it runs.

B. Who did the work

A model, Claude Opus 5.5 in Claude Code, wrote the plan, the code, the analysis and the text. The registry's agents wrote and reviewed the 127 explanations and their probes.