Reading The Dictionary:

An Exploration Into the Interpretability of Dictionary Learning

By Kola Ayonrinde, Andrew Gritsevskiy, Hans Gundlach, Derik Kauffman
(authors listed in last name alphabetical order)

"Among these unhappy mortals is the writer of dictionaries, whom mankind have considered, not as the pupil, but the slave of science"

- Samuel Johnson, A Dictionary of the English Language

A central aim of neural network interpretability is to find meaningful concepts in the jungle of activations these networks create. Recent research has shown that a representation learning method, sparse dictionary learning, can extract meaningful features from a neural network. In this vein, we develop a dictionary learning system of our own on a one-layer transformer using a custom dataset. This paper outlines our design, the features we found, and the scaling laws we discovered. This includes the result that % of nonzero-feature scales inversely with autoencoder size in our model.

Background on Autoencoders for Interpretability :

Our dictionary learning system is based on the systems developed by Anthropic in "Towards Monosematincity." "Sparse Autoencoders Find Highly Interpretable Features in Language Models" also deserves responsibility for the development of dictionary learning as an interpretability method. An in-depth description of dictionary learning is also available on Kola Ayonrinde's blog.

Our Dataset, Transformer Design:

To incorporate a wide variety of features, we chose our dataset to incorporate a wide variety of languages with many symbols. Most of our data (around 70%) came from C4. The rest of our dataset includes Wikipedia from a variety of languages as well as a few other sources (see Appendix).

In the design of our transformer, we tried to keep close to Anthropic's design. Therefore, we chose to build a one-layer, 8-head transformer. We trained for 28000 iterations, with a batch size of 64 consisting of sequences of 1024 tokens. We used the gpt-2 tokenizer.

Our Autoencoder Design:

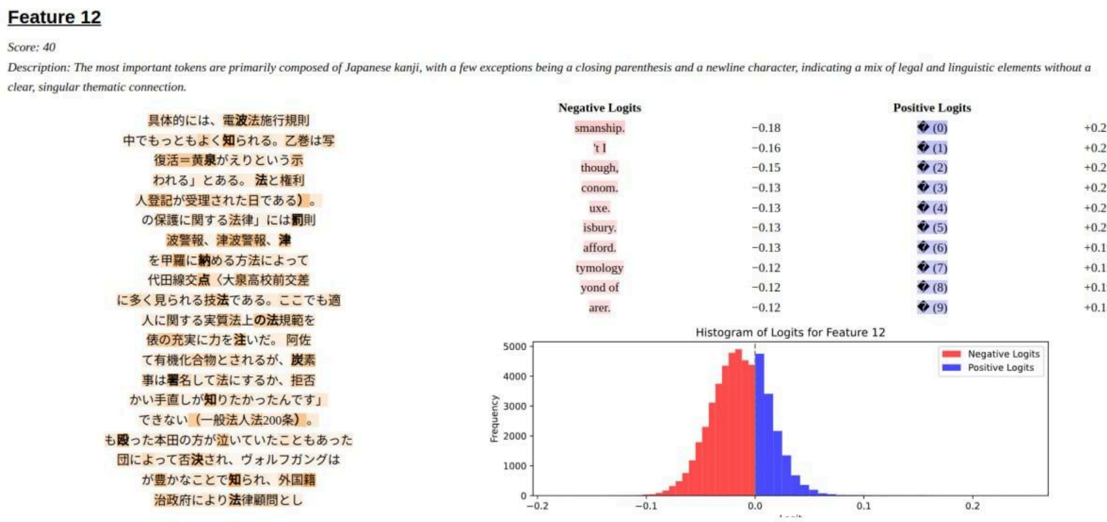
For our autoencoder, we followed a broadly similar design to Anthropic's as well. In order, to prevent dead features we used Anthropic's resampling feature outlined in Towards Monosemanticity. We trained autoencoders with 512, 1024, 2048, 4096, 8182, and 16384 features.

Examples of Interpretable Features:

Interpretable Features Found in Our 1024 and 4096 Autoencoder

Cool Features for Autoencoder 1024:

Feature 12: Japanese Legal Terminology Feature. We found this feature with autointerpretability using the openAI API.



Feature 12

Score: 40

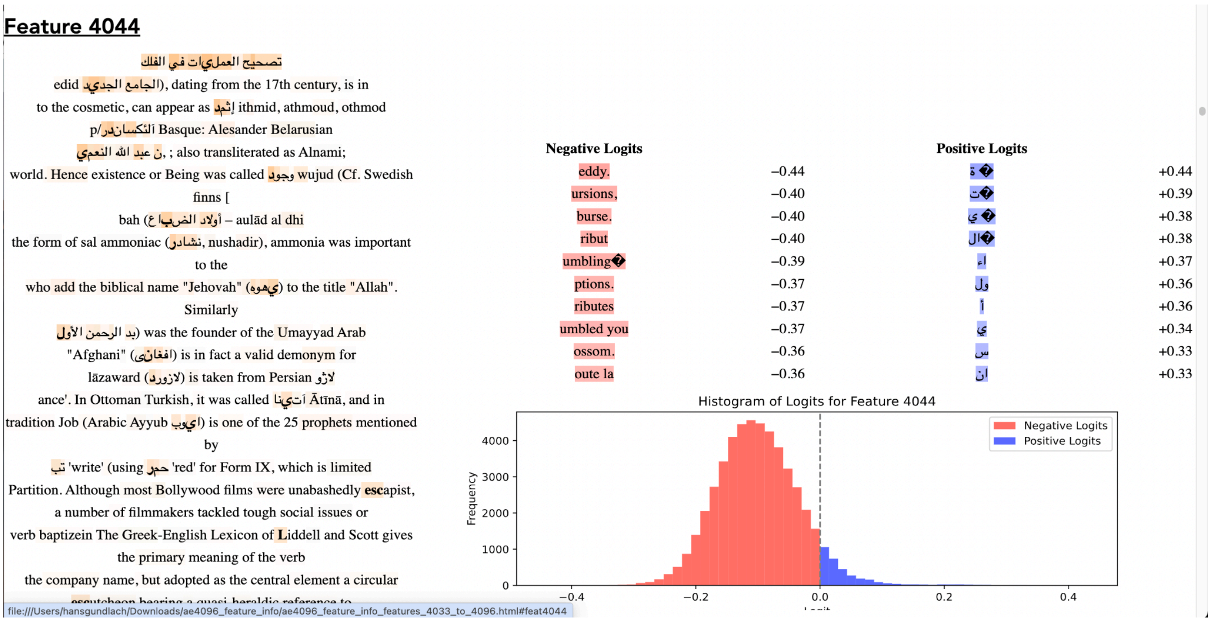
Description: The most important tokens are primarily composed of Japanese kanji, with a few exceptions being a closing parenthesis and a newline character, indicating a mix of legal and linguistic elements without a clear, singular thematic connection.

具体的には、電波法施行規則
中でもっともよく知られる。乙巻は写
復活＝黄泉がえりという示
われる」とある。 法と権利
人登記が受理された日である）。
の保護に関する法律」には罰則
波警報、津波警報、津
を甲羅に納める方法によって
代田線交点〈大泉高校前交差
に多く見られる技法である。ここでも適
人に関する実質法上の法規範を
俵の充実に力を注いだ。 阿佐
て有機化合物とされるが、炭素
事は署名して法にするか、拒否
かい手直しが知りたかったんです」
できない（一般法人法200条）。
も殴った本田の方が泣いていたこともあった
団によって否決され、ヴォルフガングは
が豊かなことで知られ、外国籍
治政府により法律顧問とし

Negative Logits		Positive Logits	
smanship.	−0.18	� (0)	+0.25
't I	−0.16	� (1)	+0.23
though,	−0.15	� (2)	+0.21
conom.	−0.13	� (3)	+0.21
uxe.	−0.13	� (4)	+0.20
isbury.	−0.13	� (5)	+0.20
afford.	−0.13	� (6)	+0.19
tymology	−0.12	� (7)	+0.19
yond of	−0.12	� (8)	+0.19
arer.	−0.12	� (9)	+0.18

Cool Features for Autoencoder 4096:

The arabic in english text feature.



Feature 4044

تصحيح العمليات في الفلك
edid الجامع الجديد), dating from the 17th century, is in
to the cosmetic, can appear as إثمد ithmid, athmoud, othmod
p/ألكسانى‌در Basque: Alesander Belarusian
ن عبد الله النعمي, ; also transliterated as Alnami;
world. Hence existence or Being was called وجود wujud (Cf. Swedish
finns [
bah (أولاد الضباع – aulād al dhi
the form of sal ammoniac (نشادر, nushadir), ammonia was important
to the
who add the biblical name "Jehovah" (يهوه) to the title "Allah".
Similarly
بد الرحمن الأول) was the founder of the Umayyad Arab
"Afghani" (افغانى) is in fact a valid demonym for
lāzaward (لازورد) is taken from Persian لاژو
ance'. In Ottoman Turkish, it was called آتينا Ātīnā, and in
tradition Job (Arabic Ayyub ايوب) is one of the 25 prophets mentioned
by
تب 'write' (using حمر 'red' for Form IX, which is limited
Partition. Although most Bollywood films were unabashedly **esc**apist,
a number of filmmakers tackled tough social issues or
verb baptizein The Greek-English Lexicon of **L**iddell and Scott gives
the primary meaning of the verb
the company name, but adopted as the central element a circular
escutcheon bearing a quasi heraldic reference to

Negative Logits		Positive Logits	
eddy.	−0.44	ة �	+0.44
ursions,	−0.40	ت�	+0.39
burse.	−0.40	ي �	+0.38
ribut	−0.40	ال�	+0.38
umbling�	−0.39	اء	+0.37
ptions.	−0.37	ول	+0.36
ributes	−0.37	أ	+0.36
umbled you	−0.37	ي	+0.34
ossom.	−0.36	س	+0.33
oute la	−0.36	ان	+0.33

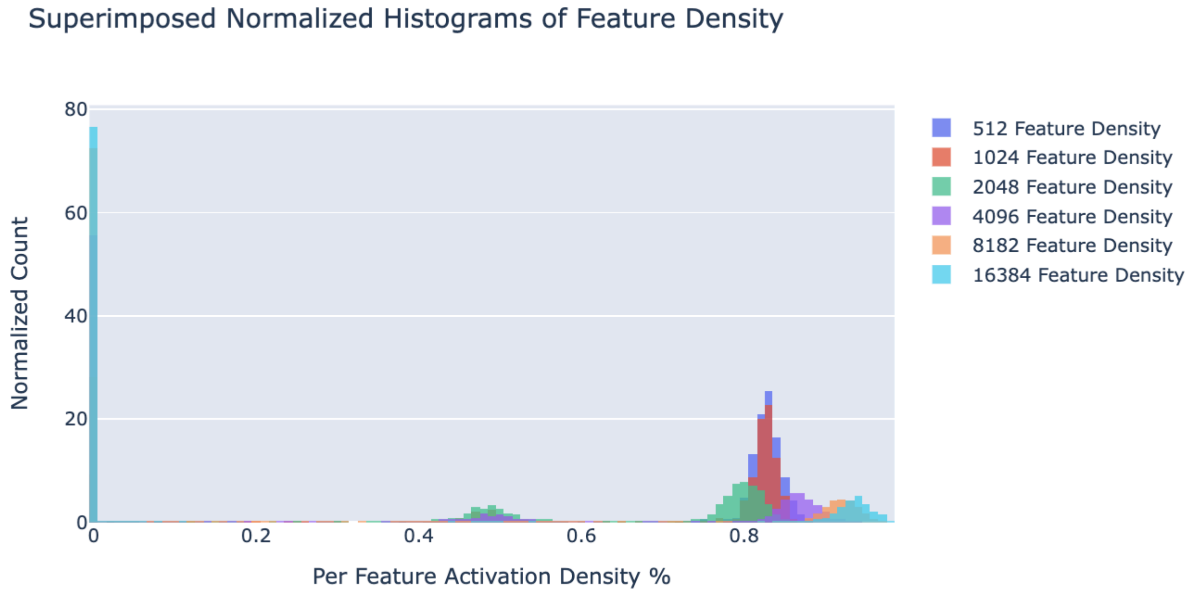
file:///Users/hansgundlach/Downloads/ae4096_feature_info/ae4096_feature_info_features_4033_to_4096.html#feat4044

Nevertheless, even though we were able to find interpretable features. It was hard to find features that had both interpretable max activations and interpretable logits that matched the max activations. We would guess that more than half of our features that had interpretable max-activations did not have interpretable logit attributions. This may suggest errors with our logit attribution or max-activation sampling. However, when we tried steering our transformer by adding our given logit attribution vector to the activations, our transformer produced outputs that were coherent with the given logit attribution.

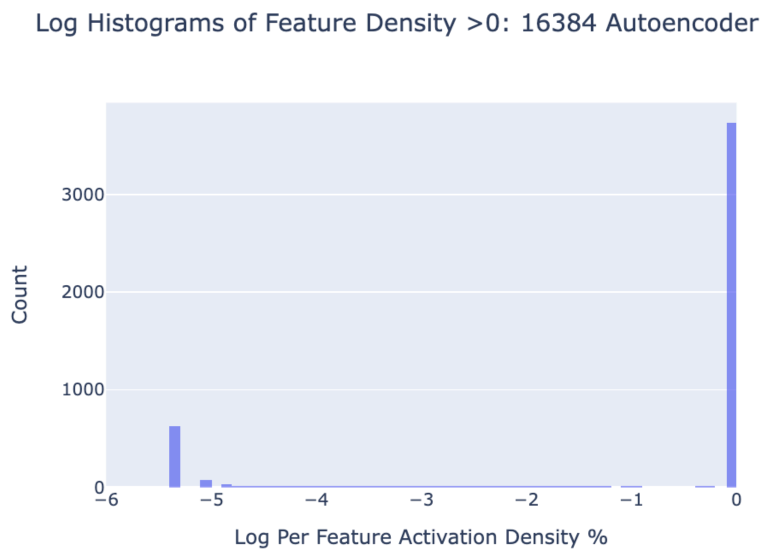
Scaling of Low-Density Features with Autencoder Size:

So how do the features change as we increase the size of the autoencoder? Naively, as the number of encoded features increases each feature should correspond to a more specific section of the data. For example, as the number of features increases we should have features for math in Russian vs just math. The process by which this happens is known as feature splitting. This means that as autoencoder size increases the density of an individual feature decreases. Overall this is the pattern we observed. However, when the number of features increased in our autoencoder we noticed that the number of "dead" features grew consistently. Here we use dead and ultralow density features (active on less than 1 in 10^6 tokens) synonymously as we were not able to distinguish features with a lower density than (1 in 10^6) in our largest autoencoders.

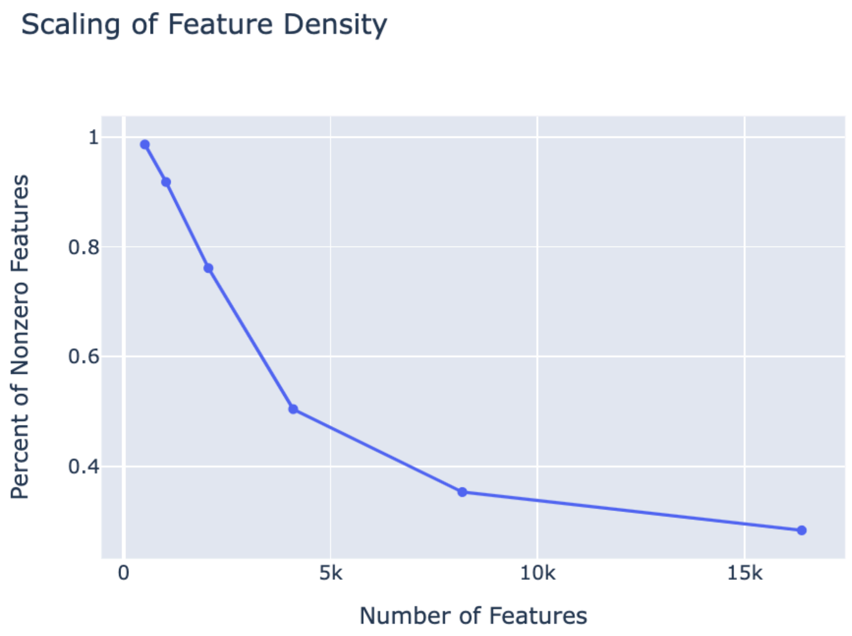
Below is a histogram (normalized w.r.t number of features) of the distribution of feature density. Each bin corresponds to features that are active on the given bin percentage of inputs (ie % features that have % feature density). Notice that for the model with 16834 features a large percentage are dead/ultralowdensity (see blue bar next to 0). The feature density histogram for each feature is given in the appendix.



What is happening with all these dead features are they simply ultra-low density? If we look at the log histogram of feature density filtering everything above 0, it looks like few features are of density 10^-4 % or below. The vast majority never are less active then 10^-5 %.



We find that the number of nonzero features scales inversely with the number of features used in our model. It seems that the number of “active” features stays relatively constant while the number of features grows leading to an inverse relation of nonzero to total features as autoencoder size grows.



References:

Ayonrinde, K. (2023). Dictionary Learning with Sparse AutoEncoders. Blog post. Retrieved from http://www.kolaayonrinde.com/2023/11/03/dictionary-learning.html

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., & Olah, C.

(2023). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*. Retrieved from https://transformer-circuits.pub/2023/monosemantic-features/index.html

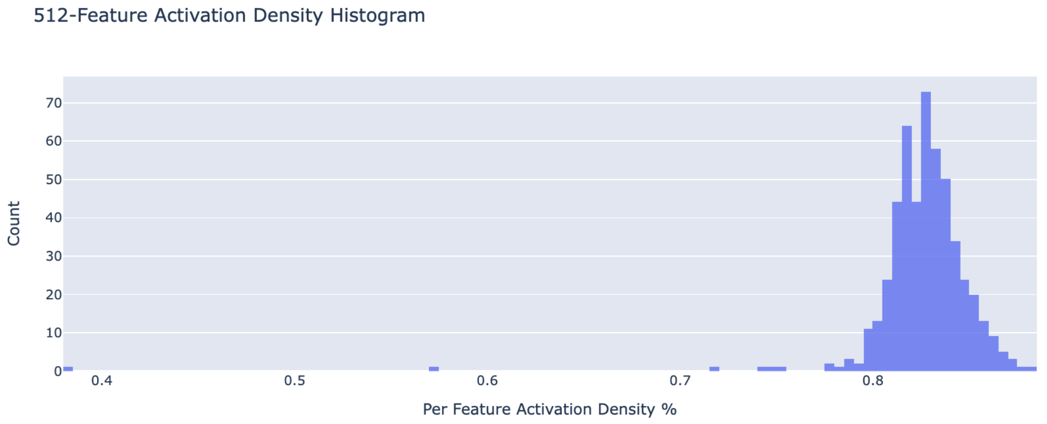
Cunningham, H., Ewart, A., Riggs, L., Huben, R., & Sharkey, L. (2023). Sparse Autoencoders Find Highly Interpretable Features in Language Models. arXiv preprint arXiv:2309.08600.

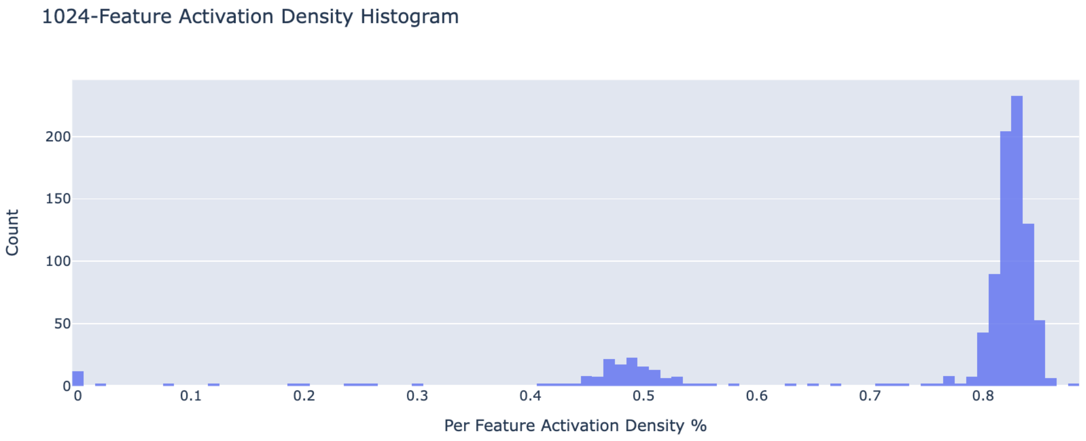
Appendix :

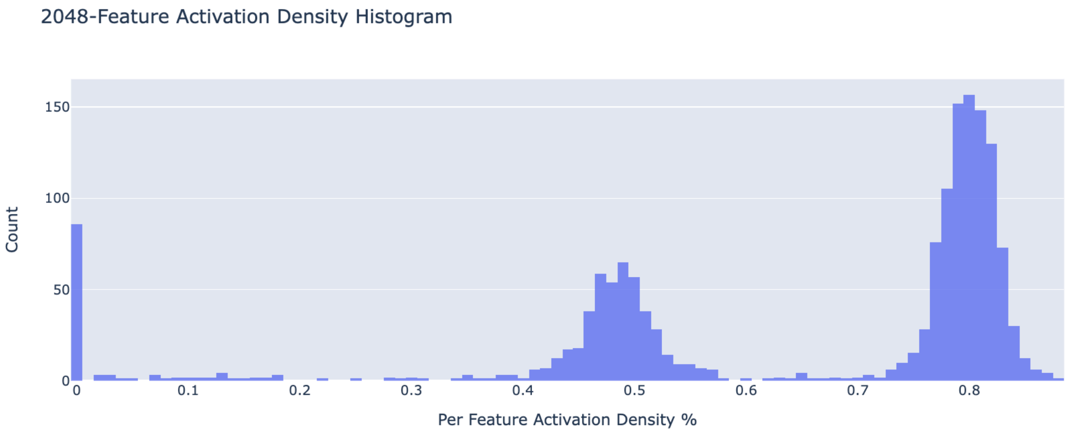
Dataset:
Below is our dataset composition in terms of the number of rows. Keep in mind that languages like Japanese have many more tokens per line than English. This biases our identified features to languages with a higher number of tokens per letter ie Japanese, arabic, etc.

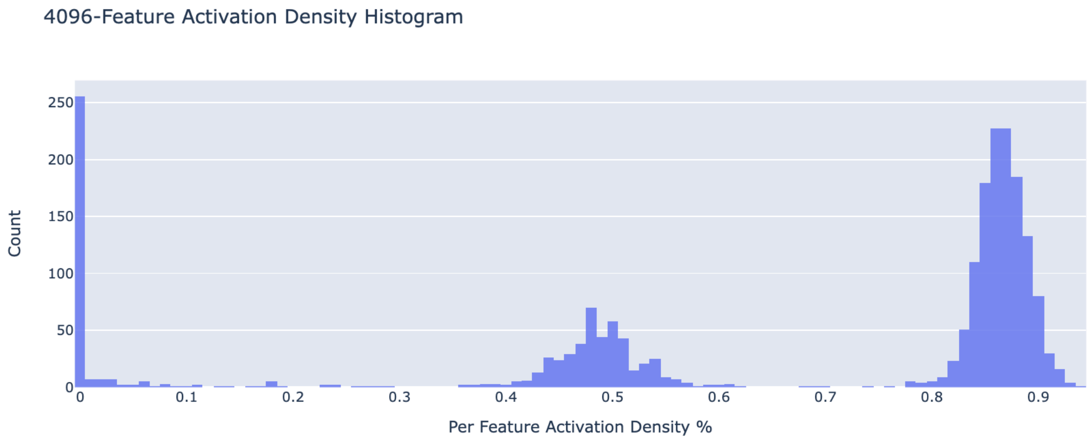
C4 -	70.42 %
Simple Wikipedia -	1.41 %
English Wikipedia -	14.08 %
French Wikipedia -	3.52 %
Arabic Wikipedia -	3.52 %
Yoruba Wikipedia -	1.06 %
Japanese Wikipedia -	2.46%
tiny stories -	3.52 %

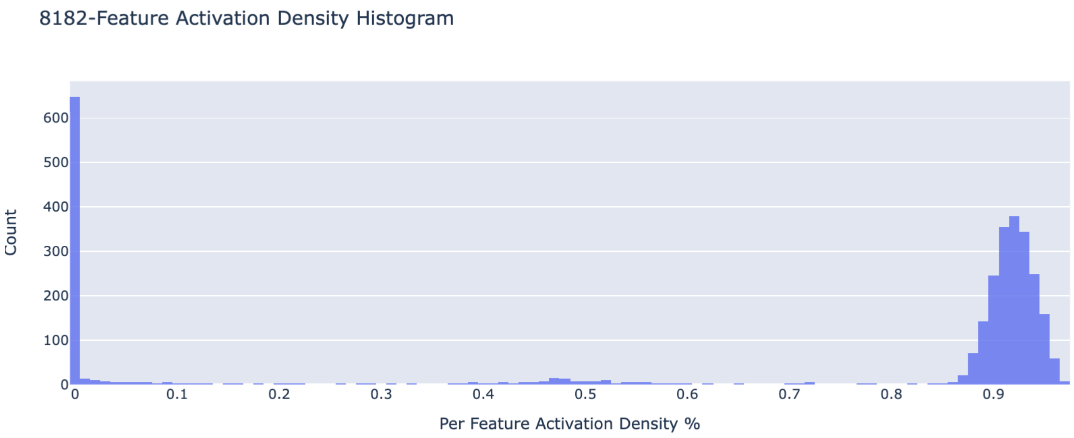
Feature Density Plot At Each Autoencoder Scale

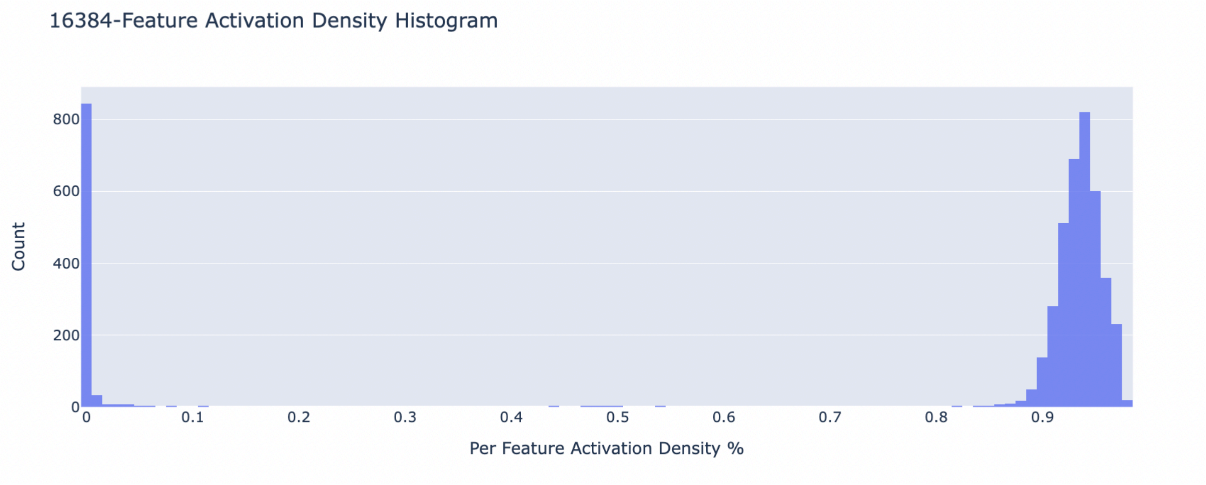



1024-Feature Activation Density Histogram
Count
200
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Per Feature Activation Density %


2048-Feature Activation Density Histogram
Count
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Per Feature Activation Density %


4096-Feature Activation Density Histogram
Count
250
200
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Per Feature Activation Density %


8182-Feature Activation Density Histogram


16384-Feature Activation Density Histogram