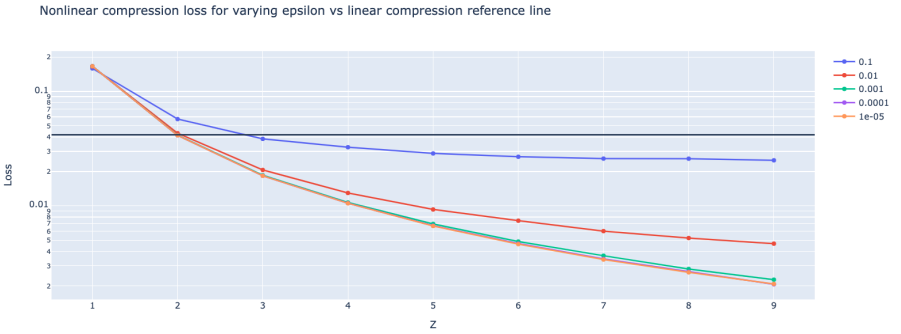
Regardless of whether large models end up using nonlinear compression, it should be possible to view directions being used with nonlinear compression as linear feature directions and reverse engineer the computation being used for compression like any other circuit. If this kind of encoding is pervasive throughout the network then it may merit some kind of automated decoding. It shouldn't pose a fundamental challenge to interpretability unless the model learns a scheme for doing complex computation while staying in a complicated nonlinear representation, which we suspect is unlikely.

To help provide intuition, the simplest example of what a nonlinear compression scheme might look like is compressing two [0,1) dimensions x and y into a single [0,1) dimension t:

$$t = \frac{\lfloor Zx \rfloor + y}{Z}$$

This works by quantizing the x dimension using some integer Z such that the floating point precision of t is split between x and y. This particular function needs the discontinuous floor function to compute, and the discontinuous fmod function to invert, but models can't compute discontinuous functions. However it's possible to replace the discontinuities with steep linear segments that are only some epsilon value wide.

We can compare the mean squared error loss on random uniform dense values of x and y and see that even with epsilons as large as 0.1 and Z values as small as 3 the nonlinear compression outperforms linear compression such as picking one of the dimensions or using the average:



Connection between compressed sensing lower bounds and the toy model

Here, we formalize the relationship between a compressed sensing lower bound and the toy model.

Let $T(x) : \mathbb{R}^n \to \mathbb{R}^n$ be the complete toy model autoencoder defined by $T(x) = \mathrm{ReLU}(W_2 W_1 x - b)$ for an $m \times n$ matrix W_1 and an $n \times m$ matrix W_2.

We derive the following theorem:

Theorem 1. Suppose that the toy model recovers all x with $T(x)$ such that $\|T(x) - x\|_2 \leq \varepsilon$ for sufficiently small ε and W_1 has the (δ, k) restricted isometry property. The inner dimension of the projection matrix W is $m = \Omega(klog(n/k))$.

We prove this result by framing our toy model as a compressed sensing algorithm. The primary barrier to doing so is that our optimization only searches for vectors that are close in ℓ_2 distance to the original vector and may not itself be exactly k-sparse. The following lemma resolves this concern through a denoising step:

Lemma 1. Suppose that we have a toy model $T(x)$ with the properties in Theorem 1. Then there exists a compressed sensing algorithm $f(y) : \mathbb{R}^m \to \mathbb{R}^n$ for the measurement matrix W_1.

Proof. We construct $f(y)$ as follows. First, compute $\tilde{x} = \mathrm{ReLU}(W_2 y - b)$, as in $T(x)$. This produces the vector $\tilde{x} = T(x)$ and so by supposition $\|T(x) - x\|_2 \leq \varepsilon$. Next, we threshold $\tilde{x}$ to obtain $\tilde{x}'$ by dropping all but its k largest entries. Lastly, we solve the optimization problem: $\mathrm{min}_{x'} \|x' - \tilde{x}'\|$ subject to $W_1 x' = y$, which is convex because x' and $\tilde{x}'$ have the same support. For sufficiently small ε (specifically, ε smaller than the $(k+1)$th largest entry in x), both $\tilde{x}$ and the nearest k-sparse vector to x have the same support, and so the the convex optimization problem has a unique solution: the nearest k sparse vector to x. Therefore, f is a compressed sensing algorithm for W_1 with approximation factor 1. .

Lastly, we use the deterministic compressed sensing lower bound of Do Ba, Indyk, Price, and Woodruff [49]:

Theorem 2 (Corollary 3.1 in [49]). Given a $k \times n$ matrix A with the restricted isometry property, a sparse recovery algorithm find a k-sparse approximation $\hat{x}$ of $x \in \mathbb{R}^n$ from Ax such that

$$\|x - \hat{x}\|_1 \leq C(k) \min_{x', \|x'\|_0 \leq k} \|x - x'\|_1$$

for an approximation factor $C(k)$. If $C(k) = O(1)$, then a sparse recovery algorithm exists only if $m = \Omega(k \log(n/k))$.

Theorem 1 follows directly from Lemma 1 and Theorem 2.