Random Superposition

Jeff Edmonds
York jeff@yorku.ca Nov 2024

I got excited when I learned from 3brown2blue about neural networks being able to store more features than there are dimensions, i.e. superposition. It referenced the paper, *Toy Models of Superposition*[1]. It is excellent, yet does not seem to contain any of my thoughts. Maybe my thoughts are obvious and previously published, but I wrote them up anyway. None of it is hard. Sorry the writing is rough. I am hoping to find co-authors who will find out how the paper fits into the literature, run some experiments, help rewrite this paper, and give me some more problems to think about.

The problem definition is the same as that in [1], except me being a math and not a machine learning guy, I wanted to choose the vectors W_i randomly instead of machine learning them. Agreeing with their experimental results, I formally prove matching upper and lower bounds $m = \Theta(k \log n)$ on the relationship between the number of features n, the number of features stored k, and the number of dimensions m. I also consider when antipodal pairs of feature directions are useful. I would be curious to know how much better their learned W_i perform than my randomly chosen ones.

Many Features in Small Dimensions: Features get represented by directions allowing for $V(\text{“king”}) - V(\text{“man”}) + V(\text{“woman”}) = V(\text{“queen”})$. Think of a neural network with some small number m of neurons in some layer. The values there forms a vector $h \in \mathcal{R}^m$ in m-dimensional space. The goal is to linearly represent a large number n of features within these m dimensions when each input is sparse in that it has at most k of the n features.

Data Compression: Let sparse input $x \in \{0,1\}^n$ indicate which features are present, where $x_i = 1$ if f_i is present and this occurs for at most k features. We want to map each such x to an m bit string h in a way that allows for the recovery of x.

Results: Here are my results (likely not new).

> **Theorem 1** *Achieving perfect recovery of x from h requires $m \geq k(\log(n/k) - 1)$, which is $\Omega(k \log n)$ as long as $k \leq n^{.99}$.*
>
> This result is referenced in [1] to be by [52, 53, 54].
>
> **Proof:** The number of n bit strings with at most k ones is $\binom{n}{\leq k} \approx \left(\frac{n}{ek}\right)^k$. The number of m bit strings is 2^m. A one-to-one mapping requires $2^m \geq \left(\frac{n}{ek}\right)^k$ or $m \geq k(\log(n/k) - 1)$. ■
>
> **Theorem 2** *If some error is allowed, then $m \leq \mathcal{O}(k \log(n))$ is achievable by a simple neural network. The mapping $h = Wx$ is linear with randomly chosen directions W_i. The reverse mapping is linear $x' = W^T h$ followed by a Rectilinear Activation Unit ReLU (or Threshold).*
>
> This agrees with the experimental observation in [1] that doubling the number of hidden dimensions m, doubles the number of features k the model learned.

[1] https://transformer-circuits.pub/2022/toy_model/index.html

Theorem 3 *The number n of features represented can be doubled by pairing the features into antipodal pairs, i.e., $W_{2i+1} = -W_{2i}$, if the sparsity is $\frac{k}{n} \leq \frac{1}{2}$ and a $(1+2\left(\frac{k}{n}\right)^2)$ factor increase in error.*

This agrees with the experimental observation in [1] that antipodal pairs appear between $1-S = \frac{k}{n} = 1$ and 0.3.

A Random/Learned Directions/Mapping: The set up is the same as that in [1], except they learn the vectors W_i and we choose them randomly.

For $i \in [n]$, let $W_i \in \mathcal{K}^m$ be a randomly chosen direction in the m-space representing feature f_i. Let sparse input $x \in \{0,1\}^n$ indicate which features are present, where $x_i = 1$ if f_i is present and this occurs for at most k features. In m-space, $h = \sum_j x_j W_j = Wx$ is the vector representing input x as the linear combination of the feature vectors. The dot product $W_i \cdot h$, lets us know how feature f_i align with this stored value h. It is also a good approximation of x_i. Hence, define $x'_i = W_i \cdot h$. These are computed for all features with $x' = W^T h = W^T W x$. We get a better approximation of x_i by subtracting off the expected error ϵ and passing it through some nonlinear activation function, namely $x''_i = ReLU(x'_i - 2\epsilon)$. The total error taken over the set of inputs considered is $L = \sum_x \sum_i (x''_i - x_i)^2$.

Exponentially Many Almost Orthonormal Vectors: Clearly in m-dimensional space, you can only have m orthonormal vectors. But quite surprisingly, if we allow them to be not quite orthogonal, we can have exponentially many.

Lemma 1 *$n = e^{\delta^2 m/4}$ randomly chosen vectors $W_i \in \mathcal{K}^m$ are likely almost orthonormal with all $|W_i| = 1$ and $|W_i \cdot W_j| \leq \delta = \sqrt{\frac{4 \ln n}{m}}$.*

Proof: For $i \in [n]$, let $W_i \in \mathcal{K}^m$ be a randomly chosen direction in the m space, by randomly choosing each coordinate of each W_i with the normal distribution with mean 0 and standard deviation $\frac{1}{\sqrt{m}}$. The expected length will be $\text{Exp}|W_i|^2 = m(\frac{1}{\sqrt{m}})^2 = 1$. Their length can be scaled to be exactly 1. These are orthogonal if $|W_i \cdot W_j| = 0$. The expected value of $W_i \cdot W_j$ is 0 and standard deviation of $\frac{1}{\sqrt{m}}$.[2] Chernoff says that the probability of deviating by more than c such standard devastations is $e^{-c^2/2}$. Setting $\delta = \frac{c}{\sqrt{m}}$ with $c = \sqrt{4 \ln n}$ gives the probability that $|W_i \cdot W_j| \leq \delta$ is $p = e^{-2 \ln n} = n^{-2}$. But between the n vectors there are $\frac{1}{2}n^2$ such pairs for which $|W_i \cdot W_j| \leq \delta$ must be true. The union bound gives that the probability that all of these events succeed is at least $\frac{1}{2}n^2 p = \frac{1}{2}$. ■

Error Terms: Lets bound the error.

Lemma 2 *The error $x'_i - x_i$ is $|\sum_{j \neq i} x_j W_i^T W_j|$.*

[2]For independent zero expectation variables, the variance of the product (sum) is the product (sum) of their variance. Hence, the variance of $[W_i]_k$ is $(\frac{1}{\sqrt{m}})^2$, of $[W_i]_k[W_j]_k$ is $(\frac{1}{\sqrt{m}})^4$ and of $\sum_k [W_i]_k[W_j]_k$ in $m \times (\frac{1}{\sqrt{m}})^4 = \frac{1}{m}$. It's standard deviation is the square root of that. Alternatively, each of the m terms acts as a random walk, so the difference between the number of positive and negative terms is likely to be $\sqrt{m}$. Each such step $[W_i]_k[W_j]_k$ is likely to have magnitude $(\frac{1}{\sqrt{m}})^2$ for a total path length of $\sqrt{m} \times (\frac{1}{\sqrt{m}})^2 = \frac{1}{\sqrt{m}}$.

Proof: The value x_i for feature f_i is first approximated by $x'_i = W_i \cdot h = W_i^T W x = W_i^T(\sum_j x_j W_j) = x_i W_i^T W_i + |\sum_{j \neq i} x_j W_i^T W_j|$. We are assuming that $W_i^T W_i = |W_i|^2 = 1$. This gives $x'_i - x_i$ as stated.

This might better be understood by considering the matrix $W^T W$. If the directions W_i are almost orthonormal with $|W_i| = 1$ and $|W_i \cdot W_j| \leq \delta$, then $W^T W$ will be the $n \times n$ matrix with 1 on the diagonal and randomly chosen values with magnitude at most δ on the off diagonals. When computing $x' = W^T W x$, the i^{th} diagonal 1 gives $x'_i = x_i + error_i$. Its the off diagonals that give $error = |\sum_{j \neq i} x_j W_i^T W_j|$ as stated.

Lemma 3 *The error $error_i = x'_i - x_i$ is always bounded by $\epsilon = \delta k$ and by $\epsilon = \delta c' \sqrt{k}$ with probability $1 - e^{-c'^2/2}$.*

Given a fixed matrix W, there will be inputs x whose features are worst case making $error_i = \delta k$ and then making $|x''_i - x_i|$ be one. However, if the inputs x are chosen randomly or if the W is chosen randomly after the $\{x\}$ are fixed, then this will only occur with probability $e^{-c'^2/2}$. This will add $e^{-c'^2/2}|\{x\}||\{i\}|$ to $L = \sum_x \sum_i (x''_i - x_i)^2$.

Proof: Because x is sparse, $x_j \neq 0$ for at most k features f_i. Hence, the sum $error = |\sum_{j \neq i} x_j W_i^T W_j|$ has at most k non-zero terms. Because $|W_i \cdot W_j| \leq \delta$, the magnitude of each of these terms is bounded. In the worst case, all the terms are positive, giving a sum of $\epsilon = \delta k$. However, because $W_i^T W_j$ is randomly positive or negative for a random W, most of these terms cancel. Consider the number of positive terms minus the number of negative terms. The expectation of this difference is zero and the variance is $\sqrt{k}$. Chernoff says that the probability of deviating by more than c' such standard devastations is $e^{-c'^2/2}$. Hence, with this probability, a bound of $\epsilon = \delta c' \sqrt{k}$ suffices. ■

Error Measure L: Lets bound the total error $L = \sum_x \sum_i (x''_i - x_i)^2$.

Lemma 4 *Suppose the error is bounded by $error_i < \epsilon < \frac{1}{2}$.*
1) $x''_i = Threshold_{\frac{1}{2}}(x'_i)$ gives the perfect answer $x''_i = x_i$.
2) $x''_i = ReLU(x'_i - 2\epsilon)$ gives $|x''_i - x_i| \leq 3\epsilon$.

Proof: In the first case, when $x_i = 1$, $x'_i = 1 \pm error_i > \frac{1}{2}$, and $x''_i = Threshold_{\frac{1}{2}}(x'_i) = 1$. When $x_i = 0$, $x'_i = 0 \pm error_i < \frac{1}{2}$, and $x''_i = Threshold_{\frac{1}{2}}(x'_i) = 0$.

In the second case, When $x_i > 3\epsilon$, $x'_i - 2\epsilon = x_i \pm error_i - 2e > 0$. Hence, $|x''_i - x_i| = |ReLU(x'_i - 2\epsilon) - x_i| = |ReLU(x_i \pm error_i - 2\epsilon) - x_i| = |x_i \pm error_i - 2\epsilon - x_i| = |\pm error_i - 2\epsilon| \leq 3\epsilon$. When $x_i \in [-\epsilon, \epsilon]$, $x'_i - 2\epsilon = x_i \pm error_i - 2\epsilon \leq 0$. Hence, $|x''_i - x_i| = |ReLU(x'_i - 2\epsilon) - x_i| = |0 - x_i| \leq \epsilon$ ■

In conclusion, the error from these last two source will total $\exp L = \sum_x \sum_i (x''_i - x_i)^2 = ((3\epsilon)^2 + e^{-c'^2/2})|\{x\}||\{i\}|$.

Bounding m:

Proof of Theorem 2:

Lemma 1 gives $\delta = \sqrt{\frac{4 \ln n}{m}}$ and Lemma 3 gives $\epsilon = \delta c' \sqrt{k} = \sqrt{\frac{4 \ln n}{m}} c' \sqrt{k}$. Hence, $m = \frac{4c'^2}{\epsilon^2} k \ln n \leq \mathcal{O}(k \ln(n))$. Note Lemma 3 says c' should be a large constant and Lemma 4 says ϵ should be a small constant ■

Antipodal Pairs: We will now prove Theorem 3. We let $x_i = 1$ with probability $(1-S) = \frac{k}{n}$.

Antipodal: Let $W_{2i+1} = -W_{2i}$ and perpendicular to all other feature vectors f_j. The calculations for these two features will be:

$$W^T W x = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \text{ ReLU gives } \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \text{ Both correct.}$$

$$W^T W x = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \text{ ReLU gives } \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \text{ Both wrong.}$$

Note that the only time this technique gets a wrong answer is when both features appears. It gets two wrong in this case so $\sum_i (x_i'' - x_i)^2 = 2$. This occurs with probability $(\frac{k}{n})^2$. Hence, the expected error will be $2(\frac{k}{n})^2$.

One Direction: Lets have one feature represented and the other ignored.

$$W^T W x = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}.$$

Note that the only time this technique gets a wrong answer is when the second feature appears. It gets one wrong in this case. This occurs with probability $\frac{k}{n}$. Hence, the expected error will be $\frac{k}{n}$.

Compare Antipodal to One Direction: We should go with the antipodal technique when $2(\frac{k}{n})^2 \leq (\frac{k}{n})$ which occurs when $\frac{k}{n} \leq \frac{1}{2}$, i.e., sparse.

Total Error: In conclusion, the error from these last three source will total $\exp L = \sum_x \sum_i (x_i'' - x_i)^2 = ((3\epsilon)^2 + e^{-c'^2/2} + 2(\frac{k}{n})^2)|\{x\}||\{i\}|$.

Compare Antipodal to Smaller n: Theorem 2 gives $\epsilon^2 = \frac{4k \ln n}{m}$. Let ϵ' be the same after halving n because the feature have been paired with antipodal directions, i.e., $\epsilon'^2 = \frac{4k \ln(n/2)}{m} = \epsilon^2 - \frac{4k}{m}$.

So using antipodal directions increases $\exp L$ by $2(\frac{k}{n})^2$ and decreases it by $\frac{4k}{m}$. Hence, it should be used when $2(\frac{k}{n})^2 \leq \frac{4k}{m}$ or $m \geq \frac{2n^2}{k}$.

The experimental results in [1] had $n = 20; m = 5; k = 1$ or 2. Our relation is then $5 \geq \frac{2 \cdot 20^2}{2}$ which is wrong. Hence there must be something wrong with this analysis.