Estimating bounds for bit truncated word embeddings

Desh Raj

August 2, 2020

Abstract

Dense word embeddings are a staple feature in several natural language processing tasks. However, since the embeddings themselves consist of floating-point vectors, they are hardly memory-efficient, especially when storing in mobile devices. In this paper, we attempt to approximate these vectors using fixed-point integers by the process of bit truncation, and find lower bounds for such a truncation while maintaining the task performance.

1 Introduction

With the popularity of dense embeddings such as skip-gram, GloVe, and more recently, FastText, natural language processing tasks almost always comprise a feature layer that initializes words using pre-trained embeddings. Usually, these embeddings are of several hundred dimensions, with each dimension being a floating-point number (sometimes restricted to a given range). While these vectors have been found to perform consistently well on several NLP tasks such as text classification, question-answering, machine translation, etc., they are not particularly memory-efficient. For instance, an embedding data file comprising 1 million vocabulary words, each represented by 100-dimensional vectors, would occupy at least 800 megabytes of storage space. As such, it may not be feasible to store and use them as part of a learning module on a mobile device.

To alleviate this issue, [LSR16] suggests the use of bit truncation, which essentially involves bucketing the floating point values to the nearest integer such that the vectors may be represented using much less memory (since fixed-point integers are much smaller than floating-point numbers). The authors further demonstrated empirically that 64-bit floats may be reduced to 8-bit integers without compromising on the performance on several simple tasks such as word similarity.

In this paper, we try to find bounds for bit size till which the vectors may be reduced without significantly affecting the task performance. For this, we first obtain a loose general bound such that distinct vectors are not mapped to the same vector. Thereafter, we attempt to find a tighter bound particularly for the case of the word similarity task, by hypothesizing that performance is consistent as long as the relative Euclidean distance between randomly selected pairs of points remain unchanged.

2 General loose bound: Distinct vectors must remain distinct

Let us first define some formalisms. Let D be the distribution of n-dimensional word vectors, i.e., $D \sim [0,1]^n$. Suppose our vocabulary, denoted by S, consists of m vectors picked i.i.d. from D. Let k be the reduced number of bits, such that the range of values of transformed vectors lies in $[-2^{k-1}, @^{k-1} - 1]$. For instance, if $k = 8$, vector dimensions lie in $[-128, 127]$. Let $\mathbf{x}$, $\mathbf{y}$, etc., denote particular word vectors in S, s.t. $\mathbf{x} = x_1, x_2, \ldots, x_n$. Suppose after transformation, the final distribution is denoted by D'.

For any vector transformation system, we must at least ensure that most of the original vectors, if distinct, should be mapped to distinct vectors in the final vector space. For this, we bound the probability that any two vectors $\mathbf{x}$ and $\mathbf{y}$ (which are initially distinct) are mapped to the same vector. Let $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ denote the vectors obtained on transforming $\mathbf{x}$ and $\mathbf{y}$, respectively.

$$\begin{aligned}\mathsf{P}[\hat{\mathbf{x}} = \hat{\mathbf{y}}] &= \bigcap_{i=1}^{n} [\hat{x_i} = \hat{y_i}] \\ &= (\frac{1}{2^k})^n \\ &= 2^{-nk}\end{aligned} \tag{1}$$

Since the set contains a total of m samples, using union bound, the total probability of failure (i.e., any pair of distinct vectors getting mapped to the same vector) becomes

$$\begin{aligned}\mathsf{P}_{\hat{\mathbf{x}},\hat{\mathbf{y}} \sim S}[\hat{\mathbf{x}} = \hat{\mathbf{y}}] &= \bigcup_{\hat{\mathbf{x}},\hat{\mathbf{y}} \in S} 2^{-nk} \\ &= \binom{m}{2} 2^{-nk} \\ &\approx m^2 2^{-nk}\end{aligned} \tag{2}$$

For a confidence ratio 1-δ, the probability computed in Eqn. 2 must be less than δ, i.e.,

$$\begin{aligned}\implies m^2 2^{-nk} &\leq \delta \\ \implies 2\log m - nk &\leq \log \delta \\ \implies k &\geq \frac{1}{n} \log \frac{m^2}{\delta}\end{aligned} \tag{3}$$

If we consider $n = 100$, $m = 10^6$, and $\delta = 10^{-3}$, the required condition for discretization turns out to be $k \geq 0.38$, which is trivially true since k is a positive integer. This shows that in the general setting, discretization of word embeddings would almost never lead to a loss in uniqueness. However, this also emphasizes the need for a tighter bound, which we derive in the following section.

3 Tighter bound for word similarity task

In the previous section, to obtain a loose lower bound on the number of truncated bits, the inherent task assumed was vector uniqueness. To obtain tighter bounds therefore, we would need to assume other (more complex) tasks that the embeddings are required for. A common objective in this regard is the word similarity task, which is defined thus.

Given a set S of words, the task of *word similarity* is to find, for each word x in the set, the most similar word y in the same set, according to some notion of similarity. In the case of the evaluation of word embeddings, the similarity score in vector space is usually computed by taking the cosine or the Euclidean distance between the two vectors.

For the truncated embeddings to perform well on a word similarity task, the relative distances between point pairs should remain unchanged after truncation. More formally, given four word vectors $\mathbf{x_1}$, $\mathbf{y_1}$, $\mathbf{x_2}$, and $\mathbf{y_2}$, the following should hold

$$\begin{aligned}&(L_2(\mathbf{x_1}, \mathbf{y_1}) - L_2(\mathbf{x_2}, \mathbf{y_2})) \\ &\cdot (L_2(\hat{\mathbf{x_1}}, \hat{\mathbf{y_1}}) - L_2(\hat{\mathbf{x_2}}, \hat{\mathbf{y_2}})) > 0,\end{aligned} \tag{4}$$

where $\hat{\mathbf{x}}$ denotes $\mathbf{x}$ after truncation, and $L_2(\mathbf{x}, \mathbf{y})$ is the Euclidean distance.

Let a denote the value of embedding along any dimension. From quantization theory, the maximum quantization error along this dimension is given as $\frac{\max(a)-\min(a)}{2^{k+1}}$, if there are k bits used. For sake of simplicity, let us assume that the embedding values along any dimension are distributed according to the normal distribution, and restricted to the range [-1,1]. Then, the max quantization error along any dimension becomes 2^{-k}. Furthermore, any vector, upon truncation, will move a maximum squared distance of $n.2^{-2k}$. With this change, the squared distance between any two vectors, upon truncation, will change by a maximum quantity $n.2^{1-2k}$, and the maximum change in the difference between the squared distances of any two pairs of points would then be $n.2^{2-2k}$. Let us denote this quantity as λ, i.e.,

$$\lambda = \frac{n}{2^{2k-2}}. \tag{5}$$

For the relative distance to change after truncation (i.e., for (4) to be violated), the initial distance between the given point pairs must be less than this distance. It must be noted here that this is only a necessary condition and not a sufficient condition. In fact, the majority of point pairs meeting this criteria would satisfy (4) after truncation. This is due to the distribution of points in high-dimensional space.

We will now compute the probability that the initial difference in squared distance between the point pairs is less than λ computed above. For this computation, we assume (as stated earlier), that the embedding values along each dimension are distributed randomly according to the same normal distribution $\mathcal{N}(0, \sigma^2)$. While this is arguably a restrictive assumption, it is nevertheless necessary to approximate the embeddings through known distributions for sake of computational simplicity.

With such an assumption, we get

$$\begin{aligned} (x_i - y_i) &\sim \left(\mathcal{N}(0, 2\sigma^2)\right), \\ \text{and} \quad (x_i - y_i)^2 &\sim \Gamma(\frac{1}{2}, 4\sigma^2), \end{aligned} \tag{6}$$

where $\Gamma(\alpha, \beta)$ is the gamma distribution with shape parameter α and scale parameter β. Taking the sum of all n dimensions (which are identically distributed), we obtain

$$\begin{aligned} \sum_{i=1}^{n} (x_i - y_i)^2 &\sim \sum_{i=1}^{n} \Gamma(\frac{1}{2}, 4\sigma^2) \\ &= \Gamma(\frac{n}{2}, 4\sigma^2). \end{aligned} \tag{7}$$

(7) gives the distribution of the squared distance between any two word embeddings in the vector space. From mathai1993noncentral, we can then write the density function for the difference of two identical gamma distributions as

$$g(u) = \begin{cases} \frac{u^{\alpha-1}}{\Gamma(\alpha)(2\beta)^{\alpha}} W_{(0,\frac{1}{2}-\alpha)}(\frac{2u}{\beta}), & \text{if } u > 0 \\ \frac{(-u)^{\alpha-1}}{\Gamma(\alpha)(2\beta)^{\alpha}} W_{(0,\frac{1}{2}-\alpha)}(\frac{-2u}{\beta}), & \text{if } u \leq 0 \end{cases} \tag{8}$$

where $\Gamma(z)$ is the gamma function ($\Gamma(z) = (z-1)!$), and $W_{(\kappa,\mu)(z)}$ is the Whittaker function, defined in terms of Kummer's confluent hypergeometric functions U and M as follows.

$$\begin{aligned} W_{(\kappa,\mu)}(z) = \exp(\frac{-z}{2}) z^{\mu+\frac{1}{2}} \\ U(\mu - \kappa + \frac{1}{2}, 1 + 2\mu; z) \end{aligned} \tag{9}$$

$$\begin{aligned} U(a, b; z) = \frac{\Gamma(1-b)}{\Gamma(a+1-b)} M(a, b; z) \\ + \frac{\Gamma(b-1)}{\Gamma(a)} z^{1-b} M(a+1-b, 2-b; z) \end{aligned} \tag{10}$$

$$M(a, b; z) = \sum_{n=0}^{\infty} \frac{a^{(n)} z^n}{b^{(n)} n!}, \tag{11}$$

where $a^{(0)} = 1$ and $a^{(n)} = a(a+1)\dots(a+n-1)$ is the rising factorial. Substituting the values of κ and μ in (9), and then using (10) and (11) to substitute U, we get

$$\begin{aligned} W = \exp(\frac{-z}{2}) \sum_{n=0}^{\infty} \frac{z^n}{2^{(n)} n!} \\ \left(\frac{\Gamma(2\alpha - 1)}{\Gamma(\alpha)} z^{1-\alpha} + \frac{\Gamma(1-2\alpha)}{\Gamma(1-\alpha)} z^{\alpha} \right) \end{aligned} \tag{12}$$

We now use the definition of the Gamma function, and use Roman factorials to approximate the factorial for negative integers in the second term of (12) to obtain

$$\begin{aligned} W = \exp(\frac{-z}{2}) \sum_{n=0}^{\infty} \frac{z^n}{2^{(n)} n!} \\ \left[\frac{(2\alpha-2)!}{(\alpha-1)!} z^{1-\alpha} + \frac{(-1)^{\alpha} \alpha!}{(2\alpha)!} z^{\alpha} \right] \end{aligned} \tag{13}$$

Using (13) in (8) and discarding the piecewise function in the negative domain (since we are only concerned with the absolute difference in distances between vector pairs), we get, $\forall u \geq 0$,

$$\begin{aligned} g(u) = \exp(\frac{-u}{\beta}) \sum_{n=0}^{\infty} \frac{(\frac{2u}{\beta})^n}{2^{(n)} n!} \\ \left[\alpha \binom{2\alpha}{\alpha} \frac{2^{-2\alpha}}{(2\alpha-1)} + \frac{(-1)^{\alpha}}{(2\alpha)! \beta^{2\alpha}} u^{2\alpha-1} \right] \end{aligned} \tag{14}$$

Let us now approximate the summation term in the above equation.

$$\begin{aligned} \sum_{n=0}^{\infty} \frac{(\frac{2u}{\beta})^n}{2^{(n)} n!} &= \sum_{n=0}^{\infty} \frac{2^n (\frac{u}{\beta})^n}{(2.3 \ldots (2+n-1)) n!} \\ &\leq \sum_{n=0}^{\infty} \frac{1}{n!} (\frac{u}{\beta})^n \exp(\frac{u}{\beta}) \end{aligned} \tag{15}$$

The final approximation in the above equation is valid since the series can be shown to be convergent using ratio test. Using (15) in (14), we get

$$g(u) \leq \left[\alpha \binom{2\alpha}{\alpha} \frac{2^{-2\alpha}}{(2\alpha-1)} + \frac{(-1)^{\alpha}}{(2\alpha)! \beta^{2\alpha}} u^{2\alpha-1} \right] \tag{16}$$

On substituting the values for α and β from (7) and taking integral from 0 to λ, we get

$$\int_0^{\lambda} g(u) du \leq \left[\frac{n}{2^n} \binom{n}{\frac{n}{2}} \frac{n}{2^{2k}} + \frac{1}{\sigma^{2n} n!} (\frac{n}{2^{2k}})^n \right]. \tag{17}$$

Furthermore, the coefficients in both the terms are $\mathcal{O}(\frac{1}{2^n n!})$. Since this is an extremely small quantity, the first term is negligible, and the second term is only finitely large if $\frac{n}{2^{2k}} > 1$, i.e., if $k < \log \sqrt{n}$. With this constraint, we can then compute the final bound on k for which the probability of relative distance between any pair of vectors (selected randomly from the m samples) getting inverted upon truncation is less than δ, as follows.

$$\begin{aligned} \frac{m^4}{\sigma^{2n} n!} (\frac{n}{2^{2k}})^n \leq \delta \\ \implies k \geq \frac{1}{2n} (4 \log m + n \log n - \log(n!) \\ - 2n \log \sigma - \log \delta). \end{aligned} \tag{18}$$

If we assume $\sigma \approx 1$ and use Stirling's approximation for $\log(n!)$, we finally obtain

$$k \geq \frac{1}{2n} \log(\frac{m^4 e^n}{\delta}). \tag{19}$$

This is the required lower bound for the number of bits to ensure performance consistency in the word similarity task with probability at least $1 - \delta$.

References

[LSR16] Shaoshi Ling, Yangqiu Song, and Dan Roth. Word embeddings with limited memory. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 387, 2016.