Neural Networks Weights Quantization: Target None-retraining Ternary (TNT)

Tianyu Zhang*, Lei Zhu†, Qian Zhao‡, and Kilho Shin§

*WeBank, †Harbing engineering university, ‡University of Hyogo and §Gakusyuin University

Introduction Target Non-retraining Ternary (TNT)

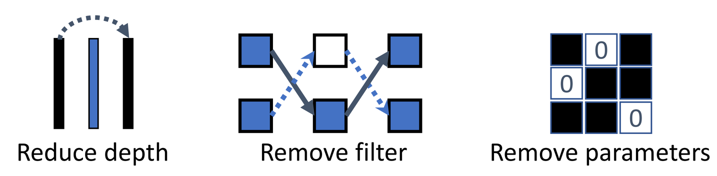
Motivation

Deep neural networks (DNNs) are widely used in many resource constrained edge devices, such as mobiles, robots, cars, and satellites, however, such devices have:

- **less** memory
- **low** powerful CPUs
- **limited** batteries

Flexibility of neural network compression

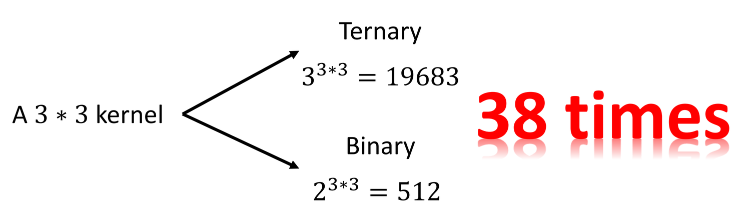
- Some parameters have limited effect
- Expression redundancy
- Reduce the performance of the model



Ternary Quantization

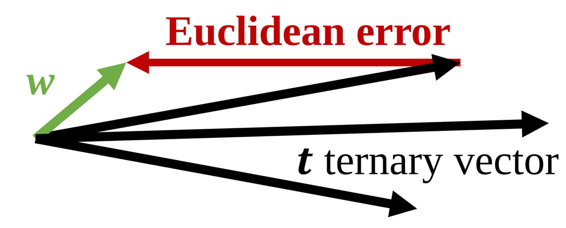
Parameters quantization of DNNs is one technique that represents DNNs' weights only using 1 or 2 bits. Rather than binary quantization, the ternary quantization approach attracts more attention by:

- Better representing ability than binary
- Reducing memory requirements
- Simplifying multiplication operations



Euclidean error method

Finding a scalar λ and a $\boldsymbol{t}$, where the members of $w^*$are -1, 0, or 1, to minimize **the Euclidean error**.



$$\min \|\boldsymbol{w} - \lambda \boldsymbol{t}\|_2$$

Problems of Euclidean error

The Euclidean error are widely used in many approaches, however, they have some practical problems:

- Searching range is too large, which is 3^N and N is the dimension of the vectors
- Converting needs training and fine-tuning, which is TIME CONSUMING
- Result CANNOT be controlled

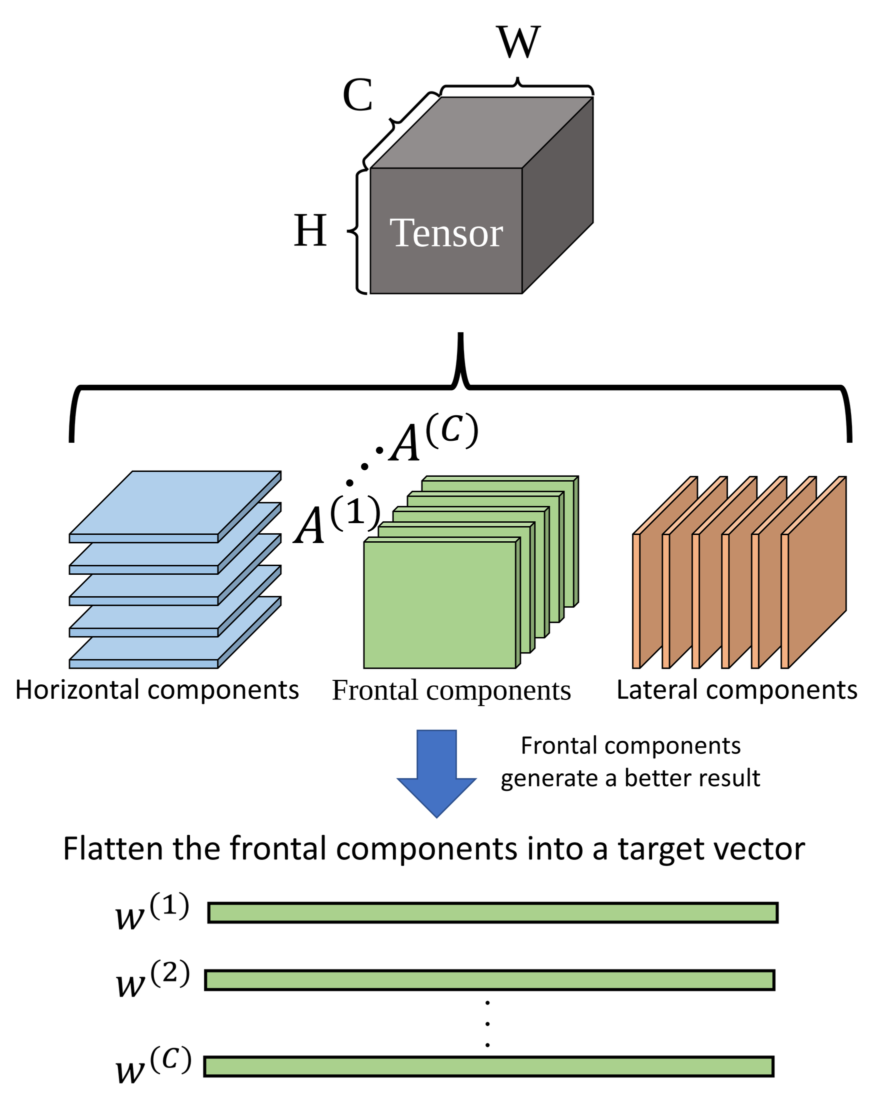
Target Non-retraining Ternary (TNT) cosine similarity based

- Reducing searching range to **N**, e.g., reducing **2187** times searching range of a 3*3 kernel.
- Non-retrain and Non-fine-tuning and converting time only costs **O(Nlog N)**
- Result is controllable

Methodology

Tensor Decomposition and Vectorization

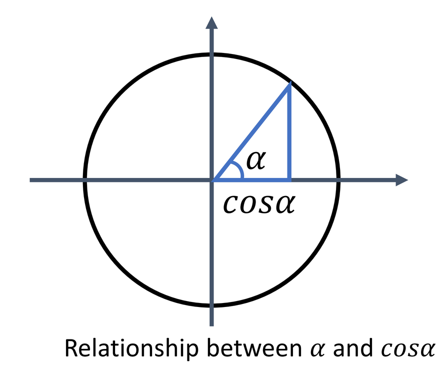
Weights of a DNN are stored in tensor which is H * C * W
Therefore, a tensor has three different decomposing ways.



Cosine Similarity

The **minimal intersecting angle** between w and t is equal to the maximum **cosine similarity** between them

$$\underset{t}{\operatorname{argmin}}\, \alpha = \underset{t}{\operatorname{argmax}} \frac{\boldsymbol{w}\cdot\boldsymbol{t}}{\|\boldsymbol{w}\|_2\|\boldsymbol{t}\|_2} = \underset{t}{\operatorname{argmax}} \frac{\widehat{\boldsymbol{w}}\cdot\boldsymbol{t}}{\|\boldsymbol{t}\|_2}$$



Relationship between α and $cos\alpha$

The maximum cosine similarity is decided by the ternary vector t, according to The dot product between a normalized w and a ternary vector

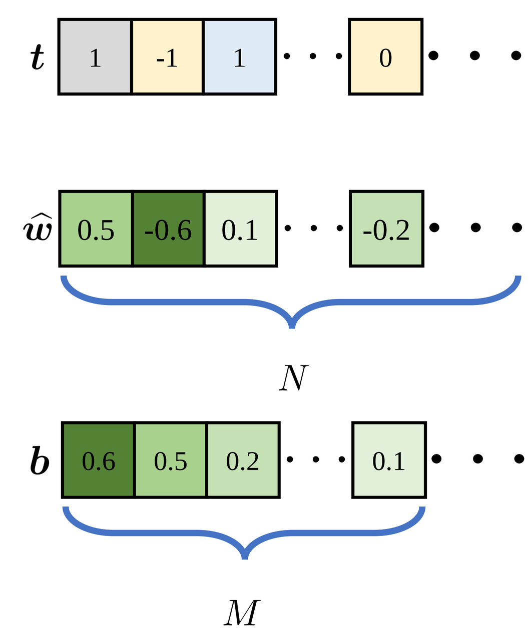
$$\underset{t}{\operatorname{argmax}} \frac{\widehat{\boldsymbol{w}}\cdot\boldsymbol{t}}{\|\boldsymbol{t}\|_2} = \underset{t_i}{\operatorname{argmax}} \frac{\sum_{i=1}^{N} a_i t_i}{\sqrt{\sum_{i=1}^{N}(t_i)^2}}$$

Sorting $|w_i|$in a decreasing order to obtain b, constraining the number of t, we have

$$\frac{\sum_{i=1}^{N} a_i t_i}{\sqrt{M}} \le \frac{\sum_{j=1}^{M} b_j}{\sqrt{M}}$$

Therefore, the searching range is only N, and the maximum cosine similarity is defined by

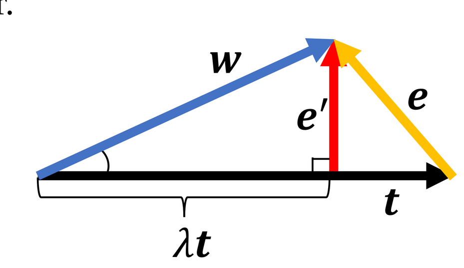
$$\operatorname{argmax}\left\{\frac{\sum_{i=1}^{M} b_i}{\sqrt{M}} \middle| M = 1, \ldots, N\right\}$$



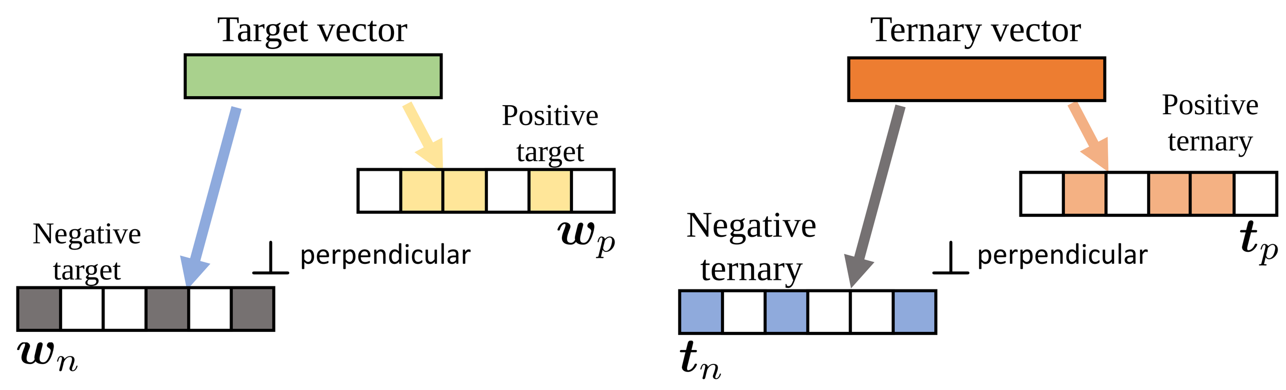
$$\frac{0.5+0.6+0.1+0}{\sqrt{3}} \le \frac{0.5+0.6+0.2+0}{\sqrt{3}}$$

Scalar-Tuning

- Introducing a scalar $\boldsymbol{\lambda}$, which can be obtained by orthogonal projection, to reduce the error further.



- Extracting the positive entries and negative entries from a target vector respectively to generate a positive vector and a negative vector.



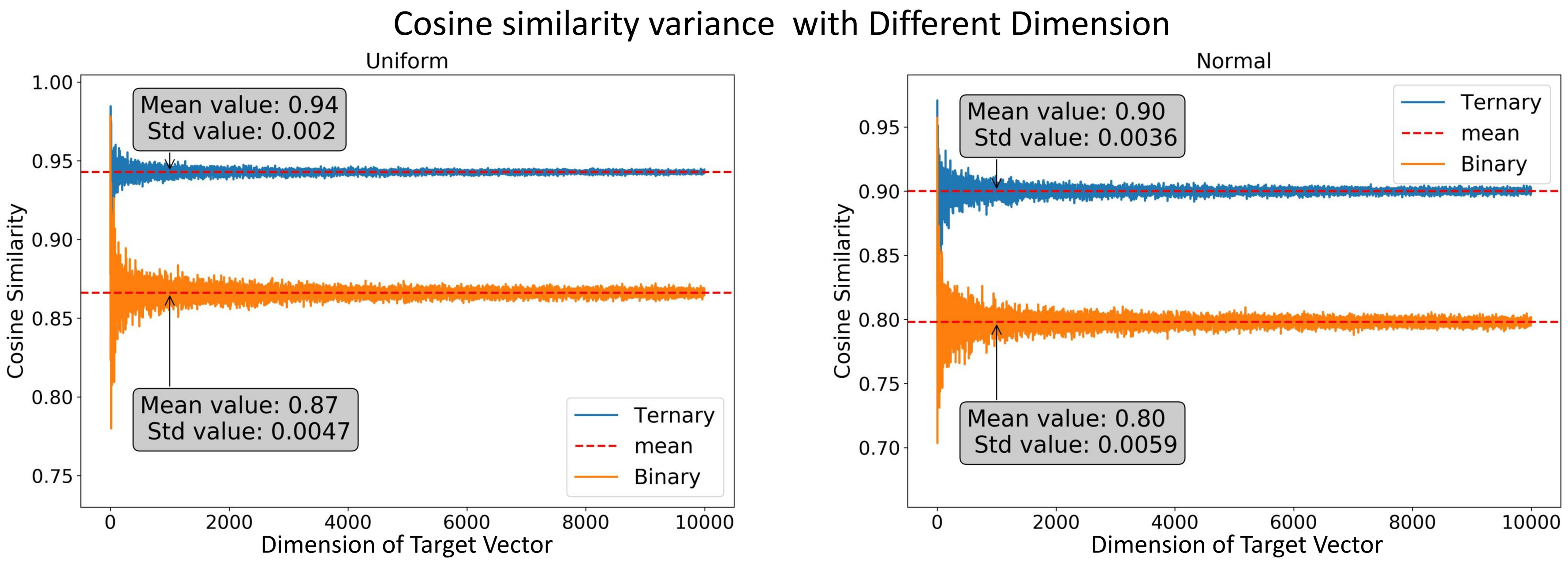
$$\left\|\boldsymbol{w} - \frac{\boldsymbol{w}\cdot\boldsymbol{t}}{\boldsymbol{t}\cdot\boldsymbol{t}}\boldsymbol{t}\right\|_2^2 \ge \left\|\boldsymbol{w}_p - \frac{\boldsymbol{w}_p\cdot\boldsymbol{t}_p}{\boldsymbol{t}_p\cdot\boldsymbol{t}_p}\boldsymbol{t}_p\right\|_2^2 + \left\|\boldsymbol{w}_n - \frac{\boldsymbol{w}_n\cdot\boldsymbol{t}_n}{\boldsymbol{t}_n\cdot\boldsymbol{t}_n}\boldsymbol{t}_n\right\|_2^2$$

- we can find **two scalars** that tuning positive and negative entries in ternary vector to reduce the error between target and ternary.

$$\lambda_p = \frac{\boldsymbol{w}_p\cdot\boldsymbol{t}_p}{\|\boldsymbol{t}_p\|} \qquad \lambda_n = \frac{\boldsymbol{w}_n\cdot\boldsymbol{t}_n}{\|\boldsymbol{t}_n\|}$$

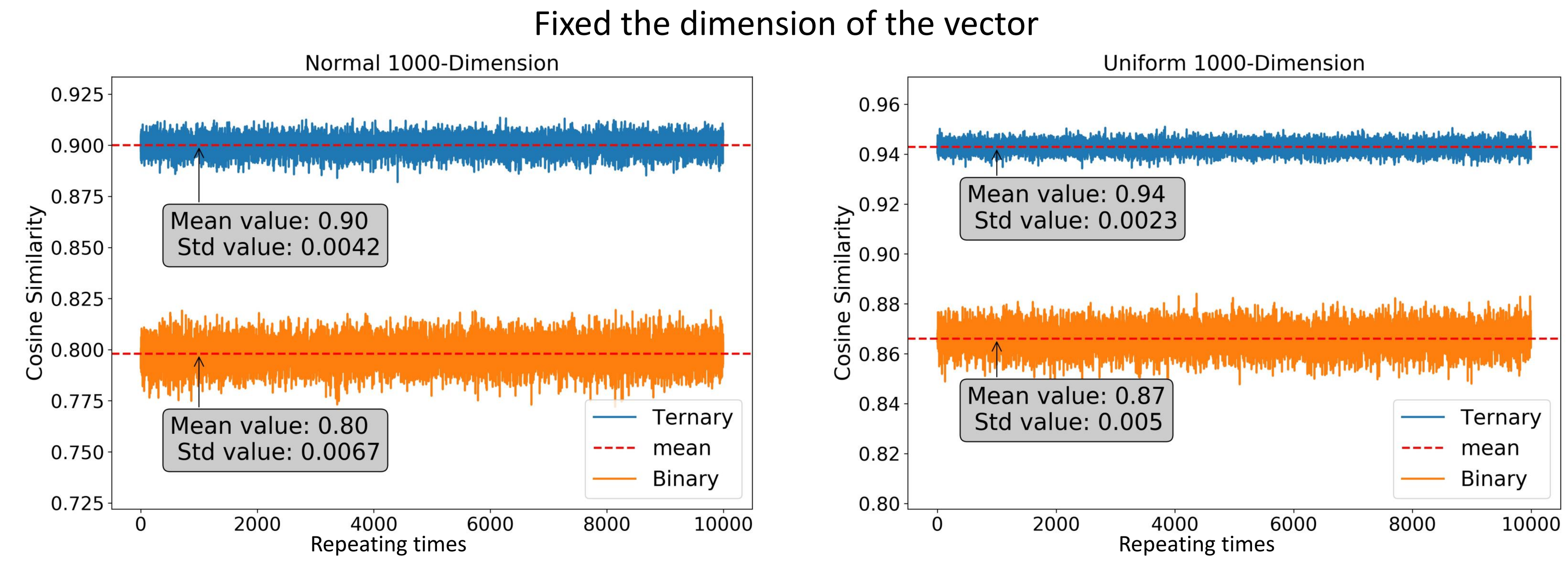
Simulation results

Cosine similarity variance with Different Dimension



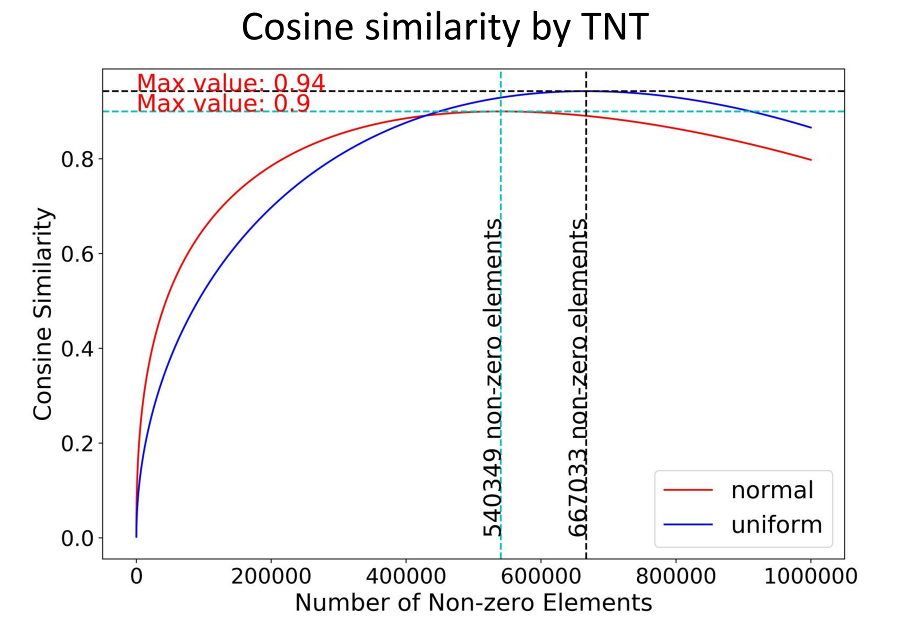
- With dimension increasing the cosine similarity becomes stable
- Uniform distribution has a higher similarity and lower variance

Fixed the dimension of the vector



The converting results is stable and uniform has a better results

Cosine similarity by TNT



- Normal distribution has a lower converting similarity
- Uniform distribution can have more zeros to reduce memory

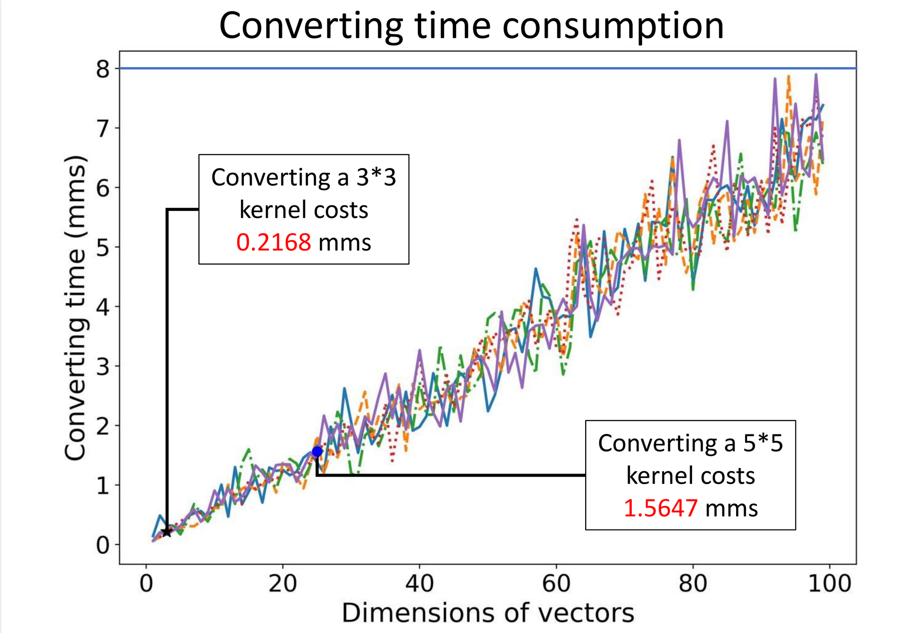
The Results on Converting Neural Network

Network	Base Line	TNT	parameter	Converting times
LeNet-5 (MNIST)	99.18%	98.97%	1,663,370	7.803s
VGG-7 (CIFAR-10)	91.31%	89.09%	7,734,410	88.288s
VGG-16 (ImageNet)	64.26%, 85.59%	56.26%, 80.25%	12,976,266	115.863s

Without retraining and fine-tuning

- LeNet-5 lost **0.21%** accuracy
- VGG-7 lost **2.22%** accuracy
- VGG-16 lost **5.34% Top-1** accuracy

Converting time consumption



- Converting 5 different vectors dimension from 1 to 100
- Converting a 100-D vector less than 8 mms

Contributions

TNT is an **efficient** parameters quantization method for neural network. According to this research we approached the following Contributions:

1. Reducing the searching range from 3^N to N
2. Constricting the searching time in **Nlog(N)**
3. Guaranteeing the best ternary vectors can be found
4. Showing that the initial parameters have an obvious affection on the weight converting result

The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019