Investigation on Robustness of Quantized Ternary Weights for Deep Neural Nets

Behzad Haghgoo, Ying Hang Seah, Golrokh Khoddambashi Emami {yinghang, bhaghgoo, golrokh} @stanford.edu

Introduction

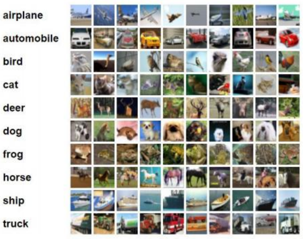
Deploying deep learning models onto devices with limited memory and computing power has always been a limitation for machine learning due to the heavy computation required. A possible solution is to compress DNNs using ternary weight quantization.

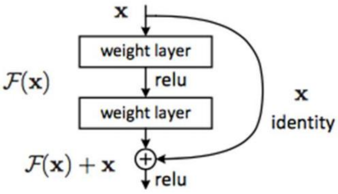
Ternary weight quantization is a new approach was proposed by Zhang and Liu to have the weights discretized into 3 values: -1, 0, 1. Ternary Weight Networks (TWN) appears to be a promising compressing model that has comparable performance compared to the full precision floating point weight networks (FP).

In our project, we are evaluating the relative robustness of TWNs vs. FP models against random noise and adversarial noise.

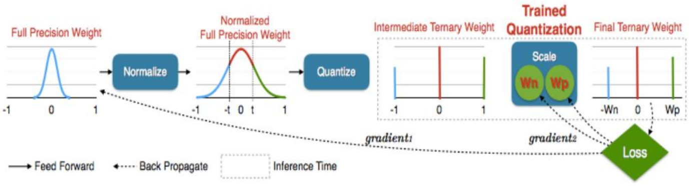
Dataset and Model

To investigate the robustness of TWNs vs FP, we are using a CIFAR-10 dataset on ResNet20. CIFAR-10 is an image classification benchmark containing images of size 32 * 32 RGB pixels in a training set of 50000 and a test set of 10000.




x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

TWN are trained using methods described in Trained Ternary Quantization by Zhu et al. The weights in each layer are quantized into 3 values, -Wn, 0, +Wp after each iteration of backpropagation.



Ternary Quantization

$$\alpha^*, \mathbf{W}^{t*} = \underset{\alpha, \mathbf{W}^t}{\arg\min}\, J(\alpha, \mathbf{W}^t) = ||\mathbf{W} - \alpha\mathbf{W}^t||_2^2 \quad \text{s.t.} \quad \alpha \geq 0,\ \mathrm{W}_i^t \in \{-1, 0, 1\},\ i = 1, 2, \ldots, n. \tag{1}$$

$$\mathrm{W}_i^t = f_t(\mathrm{W}_i|\Delta) = \begin{cases} +1, & \text{if } \mathrm{W}_i > \Delta \\ 0, & \text{if } |\mathrm{W}_i| \leq \Delta \\ -1, & \text{if } \mathrm{W}_i < -\Delta \end{cases} \tag{2}$$

$$\alpha^*, \Delta^* = \underset{\alpha \geq 0, \Delta > 0}{\arg\min} \left(|\mathbf{I}_\Delta|\alpha^2 - 2(\sum_{i \in \mathbf{I}_\Delta} |\mathrm{W}_i|)\alpha + c_\Delta\right) \tag{3}$$

$$\alpha_\Delta^* = \frac{1}{|\mathbf{I}_\Delta|} \sum_{i \in \mathbf{I}_\Delta} |\mathrm{W}_i|. \tag{4}$$

$$\Delta^* = \underset{\Delta > 0}{\arg\max} \frac{1}{|\mathrm{I}_\Delta|} \Big(\sum_{i \in \mathrm{I}_\Delta} |\mathrm{W}_i|\Big)^2 \tag{5}$$

References

[1] Zhu, Chenzhuo, et al. "Trained ternary quantization." arXiv preprint

[2] Li, Fengfu, Bo Zhang, and Bin Liu. "Ternary weight networks." arXiv preprint arXiv:1605.04711 (2016)

[3] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016

[4] GitHub. (2018). utkuozbulak/pytorch-cnn-adversarial-attacks. [online] Available at: https://github.com/utkuozbulak/pytorch-cnn-adversarial-attacks [Accessed 21 Mar. 2018].

Method

To investigate the relative robustness networks, we will evaluate the performance of the network against test images with random and adversarial noise added. On the original dataset, the FP ResNet20 achieved 91.75% top1 accuracy and the TWN ResNet20 achieved 91.71% top1 accuracy.

To study random noise, Gaussian noise, Poisson noise, Salt and Pepper noise, and Speckle noise are added to the test set and the noisy dataset is fed to FP and TWN and the performance were then further recorded.

Original Gaussian Poisson Salt and Pepper Speckle

To assess the robustness of the full precision and ternary models against adversarial noise, we calculate the absolute mean difference between the adversarial example and the original image.

Adversarial example for floating point network:

Original Photo - Car

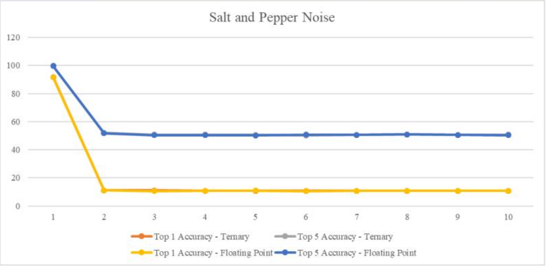
Adversarial Noise

Car to Airplane Adversarial Example

Results and Discussion

We used the average of $mean(abs(noise))$ over dataset images as our metric for robustness. This shows the difficulty of generating new adversarial examples to deceive the networks. As figures show, both in random noise and adversarial noise there is no meaningful difference in robustness.

	FP top 1	FP top 5	TWN top 1	TWN top 5
Clean dataset	91.750	99.760	91.710	99.750
Gaussian noise	19.090	72.540	19.930	72.520
Poisson noise	18.760	72.480	19.580	72.400
Speckle noise	10.340	52.380	10.430	52.590



	FP	TWN
Full Mean Difference	73.819317	73.819370

Conclusion

As a regularizer, ternary weight networks bring major advantages in memory and computation cost for small devices. In this project we showed there is no meaningful robustness loss relative to random and adversarial noise. This alongside Zhu et al.'s result that there is no significant loss in performance can prove ternary weight networks to be a reasonable replacement for floating point weight networks without loosing the advantages.

A next step for this project can be extending it to more comprehensive classifier networks like AlexNet to see how these results generalize, visualizing the activations can also help see how ternary weight networks and floating point weight networks differ in perceiving inputs.