Fast Lossless Compression with Integer-Only Discrete Flows

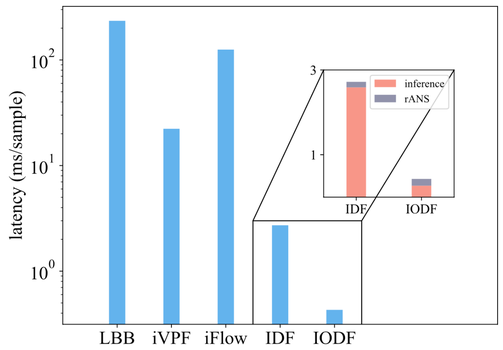
Siyu Wang[1], Jianfei Chen[1,*], Chongxuan Li[2], Jun Zhu[1,*], Bo Zhang[1]

[1]Tsinghua University; [2]Renmin University of China

ICML International Conference On Machine Learning

1. Neural compressors are not time-efficient

Current DGM-based neural compressors still suffer from high inference latency, among which Integer Discrete Flows[1] (IDF) is the most tim-efficient. IDF compresses at around 1MB/s yet far from practical demand. The time bottleneck in the coding process is network inference.



IDF views data x and latent representation z both in discrete integer space:

$$\mathcal{X} = \mathcal{Z} = \mathbb{Z}^d ,$$

and designs a bijective function $f_\theta(\cdot)$ between x and z. The additive coupling layer is the basic building block of IDF:

$$\begin{bmatrix} z_a \\ z_b \end{bmatrix} = \begin{bmatrix} x_a \\ x_b + t_\theta(x_a) \end{bmatrix}$$

The data distribution can be estimated as

$$p_X(x) = p_Z\big(f_\theta(x)\big).$$

Then rANS[2] algorithm is used to encode z into a bit stream with an average code length that approximates the entropy of p_X.

2. Integer-Only Discrete Flows

IODF consists of a novel network architecture for $t_\theta(\cdot)$, where most computations are achieved by **efficient integer operations**, and **redundant convolution filters can be pruned out with learnable binary gates.**

2.1. Integer-only network architecture

Integer Convolution A quantizer Q is used to represent a real-valued tensor r in a hybrid format with an integer tensor $\hat{r}$ and a real-valued scalar s_r:

$$r \approx \tilde{r} = Q(r) = s_r \hat{r}.$$

By quantizing inputs and weights of the convolution, **most computations can be performed with the INT8 kernel**, leading to a significant speedup.

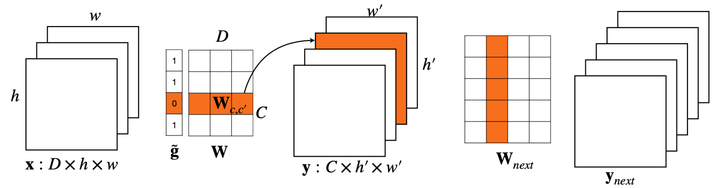
$$y_c = \sum_{c'=1}^{D} W_{c,c'} \circledast x_{c'} + b_c \;\rightarrow\; \hat{y}_c \approx \frac{s_W s_x}{s_y} \sum_{c'=1}^{D} \widehat{W}_{c,c'} \circledast \hat{x}_{c'} + b_c, c \in \{1, \dots, C\}$$

x is a $D{\times}h{\times}w$ input tensor, W is a $C{\times}D{\times}k{\times}k$ convolution kernel tensor, and y is a $C{\times}h'{\times}w'$ output tensor. They are all in the hybrid format: $x \approx s_x\hat{x}, W \approx s_W\widehat{W}, y \approx s_y\hat{y}$.

To make the inference of the integer-only model more efficient on hardware, IODF replaces dense blocks in IDF with residual blocks for its more regular and computation-intensive architecture and fewer connections across layers,

2.2. Learnable binary gated convolutions

IODF reduces redundant filters in the convolutions by adding learnable binary gates, where the masked gates can be removed at the inference time.

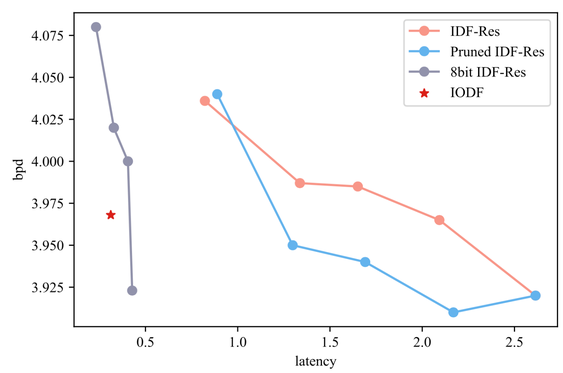


Denote a learnable binary gate by $\tilde{g} = b(g) = \mathbb{I}(g > 0.5), g \in [0,1]^C$, then a gated convolution can is defined as (omitting the bias):

$$y = \mathrm{Gconv}(x; W, g) = B(\tilde{g}) \odot \mathrm{Conv}(x, W) = \sum_{c'=1}^{D} \big(\tilde{g}_c W_{c,c'}\big) \circledast x_{c'}.$$

$B(\tilde{g})$ is a broadcast operation to a $C{\times}h'{\times}w'$ tensor with entries $B(\tilde{g})_{c,i,j} = \tilde{g}_c$. $\tilde{g}_c = 0$ relates to disabling a filter W_c and zeroing an output feature map y_c. Then the corresponding entries in the next convolution layer's weight that apply on this feature map are also removed.

2.3 Compression performance



We furtherly deploy models on a Tesla T4 GPU with TensorRT library[3] and test their inference latency.

Table1. Evaluate compression rate and inference latency of (1) pure IDF-Dense and IDF-ResNets; (2) INT8 quantized IDF-Dense and IDF-Res; (3) FP32 IDF-Res with half of the FLOPS pruned; and (4) IODF which is INT8 quantized IDF-Res with half of the FLOPS pruned.

	Coding bpd	Inference Latency 4	8	16
ImageNet32				
IDF-Dense	3.900	8.38	5.08	4.08
IDF-Res	3.926	4.19	3.19	3.59
8bit IDF-Dense	3.921	5.38	2.90	1.74
8bit IDF-Res	3.934	2.08	1.09	0.64
Pruned IDF-Res	3.947	3.27	2.04	1.60
IODF	3.979	1.79	0.94	0.54
Speedup	-	**4.7**×	**5.4**×	**7.6**×
ImageNet64				
IDF-Dense	3.638	18.65	15.45	13.93
IDF-Res	3.640	12.50	11.89	9.30
8bit IDF-Dense	3.666	8.98	5.57	4.35
8bit IDF-Res	3.673	3.03	2.02	1.61
Pruned IDF-Res	3.666	7.75	6.45	6.55
IODF	3.695	2.79	1.71	1.34
Speedup	-	**6.9**×	**9.0**×	**10.4**×

Table2. Compression performance on high-resolution image dataset CLIC. Bandwidth is measured in MB/s and GPU memory usage in GB.

Model	IDF-Dense	IDF-Res	8bit IDF-Res	Pruned IDF-Res	IODF	PNG
BPD	**2.438**	2.430	2.499	2.451	2.505	3.62
Bandwidth	0.84	1.28	7.57	1.95	**9.17**	29.8
Memory	3.2	2.8	1.7	2.4	**1.7**	*

3. Conclusion

- An efficient integer-only neural architecture for discrete flows.
- An effective algorithm for pruning out redundant filters in IDF with learnable integer gates.
- Deployment on a Tesla T4 GPU and up to 10X speedup compared to pure IDF during training.

References

[1] Hoogeboom, E., Peters, J., van den Berg, R., and Welling, M. Integer discrete flows and lossless compression. In Advances in Neural Information Processing Systems, 2019.

[2] Duda, J. Asymmetric numeral systems. arXiv preprint arXiv:0902.0271, 2009.

[3] NVIDIA. Tensorrt. https://developer.nvidia. com/tensorrt, 2018.

Primary Contact: Siyu Wang (siyuwang19@mails.tsinghua.edu.cn); Codes: https://github.com/thu-ml/IODF