Input Selection for Bandwidth-Limited Neural Network Inference: Supplementary Materials

Stefan Oehmcke[*][†] Fabian Gieseke[‡][§][†]

1 Reproducibility

We provide all the source code used for our experiments via an open GitHub repository.[1] For training the models and assessing their quality, different machines, and GPU devices (NVIDIA K20, K40, GTX1080, and V100) were used.

The chosen parameters for our static and dynamic experiments are summarized in Table 2a and 2b, respectively. The scripts with our parameter settings to run the corresponding experiments can be found in the code repository (directory `exp_scripts`). The data transformations applied during training are listed in Table 1.

Table 1: Dataset augmentations and transforms

dataset	vert. flip	horiz. flip	crop	normalize
`mnist`				✓
`f-mnist`				✓
`cifar10`		✓	✓	✓
`svhn`		✓	✓	✓
`galaxy10`	✓	✓	✓	✓
`remote`	✓	✓	✓	✓
`ship`	✓	✓	✓	✓

To run the experiments, the necessary requirements need to be installed (e.g., via `pip install -r requirements.txt`). The individual experiments can be started via the command line. For instance, the `block` experiment on `svhn` (using the normal "any" selection) can be started via:

```
python create_mask.py
--dataset svhn  --mask-type static
--any-granularity subCXLIVdrant
--lambda-patience 5
--lambda-init 0.125
--lambda-factor 1.25 --n-epochs 300
--use-warmup-net 1 --lr 0.001
--n-repeats 10
```

By executing this command, a corresponding log file and checkpoints in the directory `runs` will be created. Note that the flag `--use-warmup-net 1` enables the use of pre-trained models and optimizers. The mask type (i. e., random, static, or dynamic) can be changed via the flag `--mask-type`, which is, together with the flag `--dataset`, the only required parameter. For instance, the dynamic mask experiment for `mnist` can be reproduced via the following command:

```
python create_mask.py
--dataset mnist --mask-type dynamic
--dynamic-mask linear
--any-granularity subpixel
--lambda-patience 5
--lambda-init 0.0005
--lambda-factor 1.5 --n-epochs 400
--use-warmup 1 --lr 0.0005
--n-repeats 10
```

The remaining experiments can be started in a similar fashion. An overview over the flags and options can be obtained via `python create_mask.py --help`.

2 Broader Impact

We expect that such selection masks will play an important role for many data-intensive domains to alleviate the data transfer problem between centralized storage servers and clients: In many cases, the collected data are stored on centralized storage servers. The transfer of data between such servers and the users has already become a severe bottleneck. For instance, practitioners in remote sensing already resort to reduced versions of the data since a full data transfer would be too time-consuming (i. e., instead of the full multi-spectral image data, reduced versions are often considered, such as channels computed via the so-called Normalized Difference Moisture Index (NDMI)). A similar situation is given in astronomy, where the data transfer will become a key bottleneck in future with projects producing exabytes of data per year [1–3]. Today's storage servers already provide advanced APIs to select or crop the data

[*]stefan.oehmcke@gmail.com
[†]Department of Computer Science, University of Copenhagen
[‡]fabian.gieseke@uni-muenster.de
[§]Department of Information Systems, University of Münster
[1]`https://github.com/StefOe/selection-masks`

(a) Parameters used for static experiments

granularity	dataset	λ_{init}	λ_{fac}	p	lr mask	epochs	any-init
`channel`	`remote`	0.1	1.25	10	0.001	300	3
`pixel`	`cifar10`	1	1.05	5	0.001	300	3
	`mnist`	0.0005	1.5	5	0.005	400	3
	`f-mnist`	0.0005	1.5	5	0.005	400	3
	`galaxy10`	1	1.5	2	0.001	300	3
`block`							
(12×12)	`svhn`	0.125	1.25	5	0.001	300	3

(b) Parameters used for dynamic experiments

granularity	mask model g	dataset	λ_{init}	λ_{fac}	p	lr mask	epochs
`pixel`	linear	`mnist`	0.0005	1.5	5	0.0005	400
		`f-mnist`	0.0005	1.5	2	0.0005	400
	convatt	`cifar10`	0.1	1.15	10	0.001	300
`block`							
(12×12)	convatt	`ship`	0.025	1.15	9	0.0005	400
(48×48)	convatt	`ship`	0.025	1.15	9	0.0005	400

prior to the transmission (e. g., the Planet API mentioned above can be used to select pieces of the data beforehand; similarly, services such as the Copernicus Open Access Hub allow to select subsets of the data and also provide previews of the data). The methods presented in this work offer the potential to alleviate the data transfer bottleneck in such domains, which, in the end, will lead to less resources that have to be spent for the overall infrastructure (e. g., more powerful servers, faster network connections, ...). Naturally, while our work addresses the two aforementioned application domains, the approaches developed are generally applicable to other domains as well (e. g., IoT data, medical image data, ...). We therefore believe that the results presented in this work will affect a broad range of domains and applications.

References

[1] D. Aguado, H. Jönsson, H. Zou, and et al., *The fifteenth data release of the sloan digital sky surveys: First release of manga-derived quantities, data visualization tools, and stellar library*, Astrophysical Journal, Sup. Series, 240 (2019).

[2] K. Grange, B. Alachkar, S. Amy, D. Barbosa, P. Bommimmeni, and P. Boven, *Square kilometre array: The radio telescope of the xxi century*, Astronomy Reports, 61 (2017), pp. 288–296.

[3] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. Al-Sayyad, S. F. Anderson, J. Andrew, and et al., *LSST: From Science Drivers to Reference Design and Anticipated Data Products*, The Astrophysical Journal, 873 (2019), p. 111.