MM811 PROGRAMMING ASSIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

This assignment focuses on creating autoregression models that memorize training samples, to understand that learning the global distribution leads to memorization. You are required to submit an experiment report with necessary process description, key code, network architectures, etc. There is no restriction on how to write your report, as long as it is clear and easy to follow. For example, you can use a table to describe network architecture like Tab. 1, a formula to describe your model like Eq. 1, or some reference (van den Oord et al., 2016) if you utilize existing architectures. There is no page limitation; you can write as many pages as you want, as long as you organize it well. You can use the template of this document for your report. There is also one sample report attached, but remember that there is no restriction on your report format, as long as it is clear and well organized. I attached the entire MNIST dataset. You can work on a subset of 256 images. You can also work on 1024 images or the entire dataset if you want more challenge.

1 IMAGE CLASSIFICATION ARCHITECTURE, 40%

Design and implement a network for image classification. Full marks for this section once your accuracy is $> 90\%$. The input of the network should be images, and the output should be the label of images. You can use NLLLoss or CrossEntropyLoss.

2 CAPTURING BINARY LATENT REPRESENTATIONS, 40%

Design an Autoencoder architecture to extract the binary latent representation of images. The size of the latent should be $N \times 3 \times 4 \times 4$. Full marks for this section once the average Peak Signal-to-Noise Ratio (PSNR) is greater than 25.

3 FROM AUTOREGRESSIVE MODEL TO MEMORIZATION MODEL, 15%

Using the discrete tokens you obtain, implement an autoregressive model with different types of input and output. As your tokens are binary in channel, you can convert the signed binary values into decimal values for implementation. As this section is a little bit challenging, you can use only 100 images for experiments. The autoregression model can be expressed as:

$$p(\mathbf{Z}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathbf{z}_{<i}) \tag{1}$$

There are many different ways to implement this model, based on the design of different inputs and outputs. You are required to implement three different types of autoregressive models:

- **Sequence to Token, 5%**: The input of the network is prior sequences and the output is the next token. Full marks once over 80% of generated sequences are training sequences.
- **Sequence to Sequence, 5%**: Both the input and output are token sequences, using masking implementation. Full marks once over 80% of generated sequences are training sequences.
- **Sequence to Histogram, 5%**: The input of the network is prior sequences and the output is no longer a token, but a histogram describing the probability of the next token. Full marks once over 99% of generated sequences are training sequences, with over 512 images are used for learning.

Table 1: Details of our network architecture.

	Type	weight	stride	padding	Data size
Encoder	Input				$N \times 3 \times 32 \times 32$
	Conv2d	$64 \times 3 \times 4 \times 4$	2	1	$N \times 64 \times 16 \times 16$
	LeakyReLU				
	Conv2d	$256 \times 64 \times 4 \times 4$	2	1	$N \times 256 \times 8 \times 8$
	LeakyReLU				
	Conv2d	$512 \times 256 \times 4 \times 4$	2	1	$N \times 512 \times 4 \times 4$
	LakyReLU				
	Conv2d	$512 \times 8196 \times 1 \times 1$	1	0	$N \times 8196 \times 4 \times 4$
	Conv2d	$8196 \times \text{NoD} \times 1 \times 1$	1	0	$N \times \text{NoD} \times 4 \times 4$
Latents					$N \times \text{NoD} \times 4 \times 4$
Decoder	Linear	$\text{NoD} \times 8196 \times 1 \times 1$	1	0	$N \times 8196 \times 4 \times 4$
	Linear	$8196 \times 1024 \times 1 \times 1$	1	0	$N \times 1024 \times 4 \times 4$
	LeakyReLU				
	ConvT2d	$512 \times 1024 \times 4 \times 4$	1	0	$N \times 512 \times 4 \times 4$
	LeakyReLU				
	ConvT2d	$256 \times 512 \times 4 \times 4$	2	1	$N \times 256 \times 8 \times 8$
	ConvT2d	$64 \times 256 \times 4 \times 4$	2	1	$N \times 64 \times 16 \times 16$
	ConvT2d	$3 \times 64 \times 4 \times 4$	2	1	$N \times 3 \times 32 \times 32$
	Tanh				
Refine	Conv2d	$32 \times 3 \times 1 \times 1$	3	1	$N \times 32 \times 32 \times 32$
	LeakyReLU	$\alpha = 0.01$			$N \times 32 \times 32 \times 32$
	Conv2d	$3 \times 32 \times 1 \times 1$	3	1	$N \times 3 \times 32 \times 32$
	Output				$N \times 3 \times 32 \times 32$

NoD: number of dimension.

4 Advanced Assignment, 5%

Replace the input of your own non-image dataset; it could be text, motion, audio, music, or any dataset (text data recommanded). Reproduce the memorization model experiments, creating an autoregression model based on memorization. You can use a subset of the dataset under extremely limited conditions, such as only 2 training instances of dataset, as long as you can provde the reproduce the memorization.

References

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.