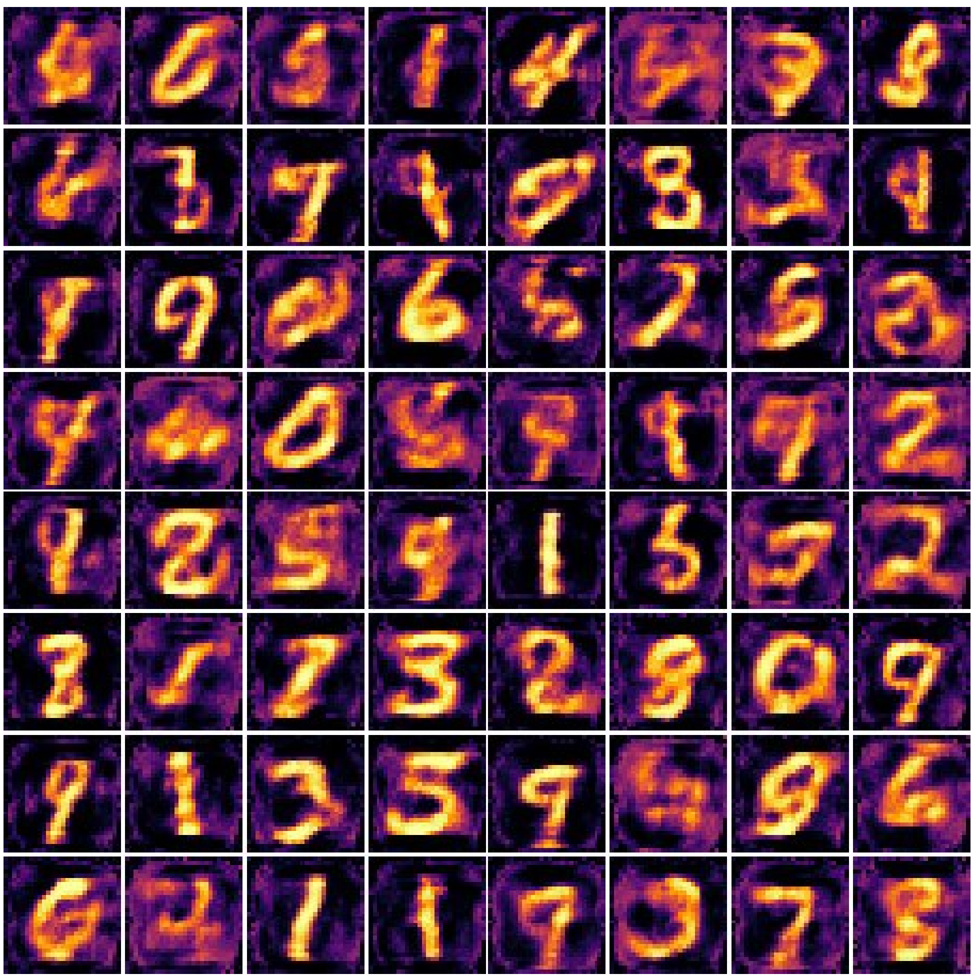
BinaryGAN: a generative adversarial network with binary neurons

October 22 2018, by Ingrid Fadelli



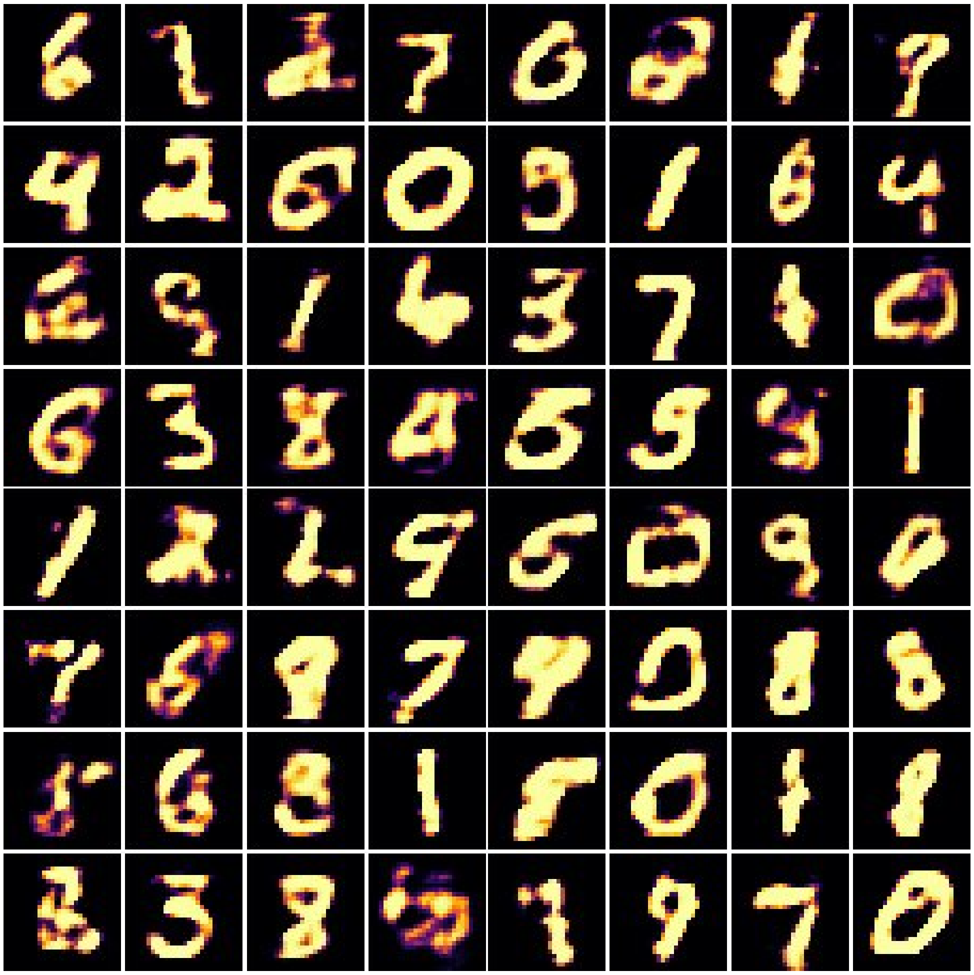
Preactivated outputs (i.e., the real-valued, intermediate values right before the binarization operation; see Section 2.2) for the proposed model implemented by MLPs and trained with the WGAN-GP objective. Proposed model with DBNs (preactivated). Credit: Dong and Yang.

Researchers at the Research Center for IT Innovation of Academia Sinica, in Taiwan, have recently developed a novel generative adversarial network (GAN) that has binary neurons at the output layer of the generator. This model, presented in a paper pre-published on arXiv, can directly generate binary-valued predictions at test time.

So far, GAN approaches have achieved remarkable results in modeling continuous distributions. Nonetheless, applying GANs to discrete data has been somewhat challenging so far, particularly due to difficulties in optimizing the model distribution toward the target data distribution in a high-dimensional discrete space.

Hao-Wen Dong, one of the researchers who carried out the study, told *Tech Xplore*, "I am currently working on music generation in the Music and AI Lab at Academia Sinica. In my opinion, composing can be interpreted as a series of decisions—for instance, regarding the instrumentation, chords and even the exact notes to use. To move toward achieving the grand vision of a solid AI composer, I am particularly interested in whether deep generative models such as GANs are able to make decisions. Therefore, this work examined whether we can train a GAN that uses binary neurons to make binary decisions using backpropagation, the standard training algorithm."

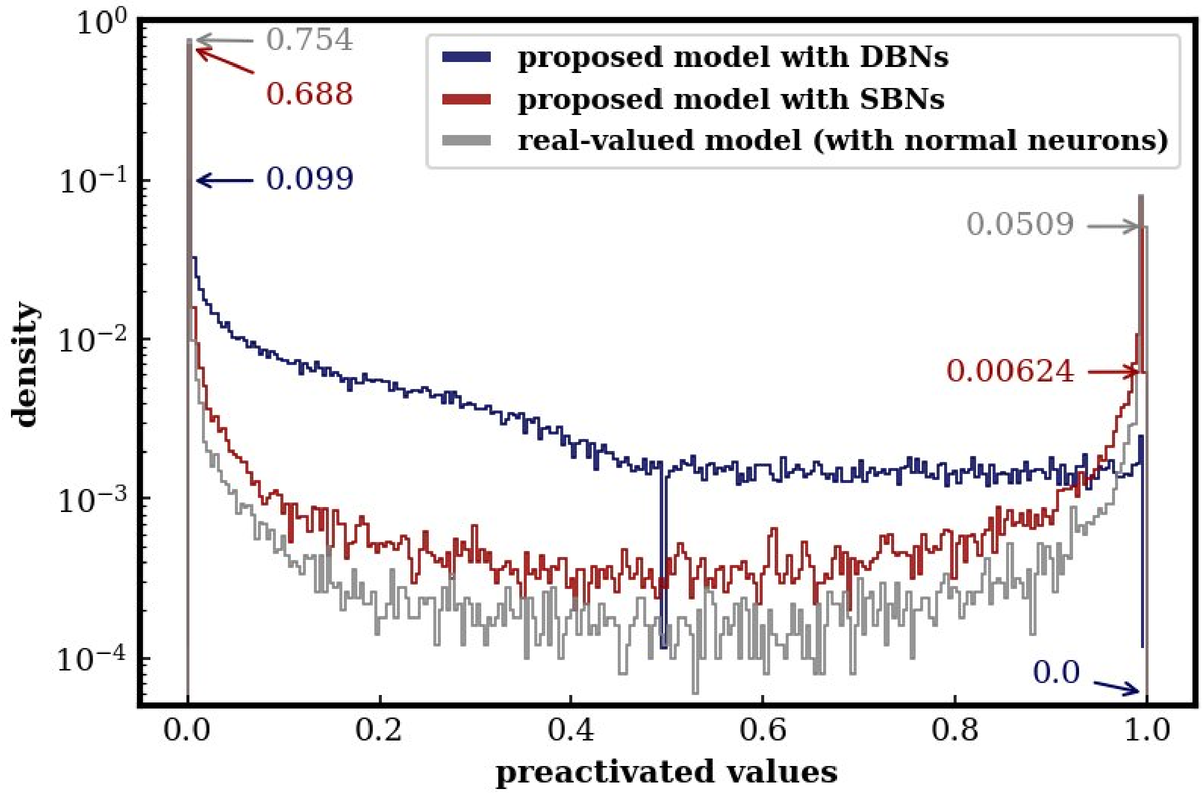
Dong and his advisor Yi-Hsuan Yang developed a model that can directly generate binary-valued predictions at test time. They then used it to generate binarized MNIST digits and compare the performance of different types of binary neurons, GAN objectives and network architectures.



Sample generated digits and preactivated outputs (i.e., the real-valued, intermediate values right before the binarization operation; see Section 2.2) for the proposed model implemented by MLPs and trained with the WGAN-GP objective. Proposed model with SBNs (pre-activated). Credit: Dong and Yang.

"In a nutshell, BinaryGAN is a GAN that adopts binary neurons, neurons that output either a one or a zero, at the output layer of the generator," Dong said. "A GAN has two main components: the generator and the discriminator. The generator aims to produce fake data samples that are able to fool the discriminator into classifying generated samples as real. On the other hand, the goal of the discriminator is to distinguish fake from real data. The feedback provided by the discriminator is then used to improve the generator. After the training, the generator can then be used to generate new data samples."

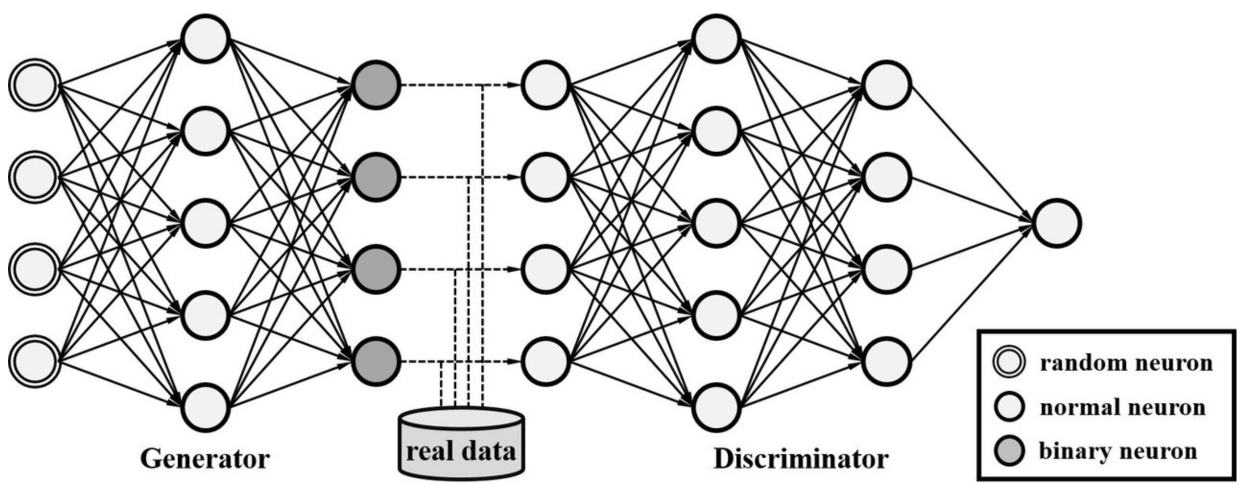
The researchers were able to effectively train BinaryGAN, their generative model with binary neurons. Their findings also suggest that the use of gradient estimators could be a promising approach to model discrete distributions with GANs.



Histograms of the preactivated outputs for the proposed model and the probabilistic predictions for the real-valued model. The two models are both implemented by MLPs and trained with the WGAN-GP objective. Credit: Dong and Yang.

"With the use of gradient estimators, we were able to train BinaryGAN, using the backpropagation algorithm," Dong said. "Moreover, the binarization adopted in the model resulted in distinct characteristics of the intermediate representations learned by the deep neural networks. This also emphasizes the importance of including binarization operations in the training, so that these binarization operations can also be optimized."

Dong and Yang are now looking to apply a GAN that adopts binary neurons to the realization of a conditional computation graph. In this instance, some components would be activated and deactivated, according to the decisions made by the network's binary neurons.



System diagram of the proposed model implemented by MLPs. Note that binary neurons are only used at the output layer of the generator. Credit: Dong and Yang.

"This is important for it allows us to build a more complex model that relies on decisions made at early layers of the network," Dong said. "For example, we could build an AI composer that learns to first decide the instrumentation and chords and then composes accordingly."

More information: — Training generative adversarial networks with binary neurons by end-to-end backpropagation. arXiv: 1810.04714v1 [cs.LG]. arxiv.org/pdf/1810.04714.pdf

— github.com/salu133445/binarygan