Differencing Neural Networks

David Shriver Sebastian Elbaum Matthew B. Dwyer

University of Virginia

Problem

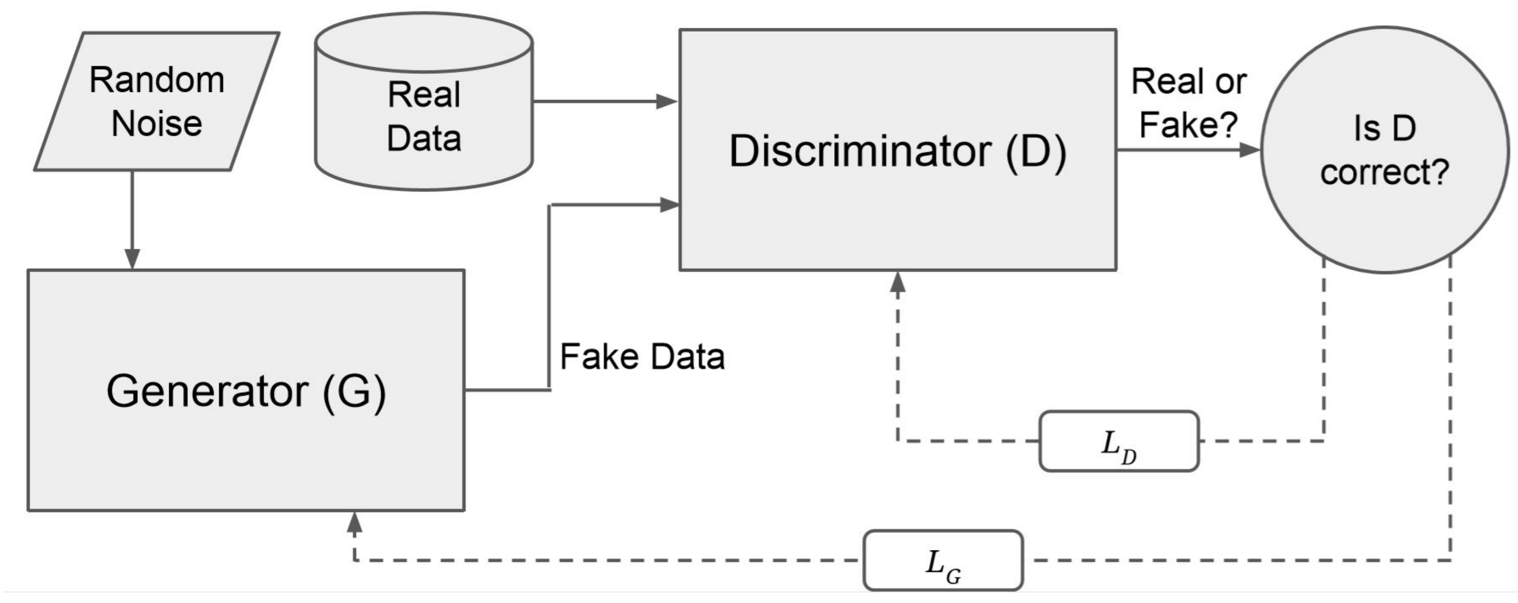
Many versions of a neural network may be trained over time. New architectures, data, or training procedures can lead to the evolution of the DNN model over time.

Two versions of a DNN may produce different outputs for the same input.

Not all inputs are relevant. Random noise may cause different outputs, but is unlikely to appear in practice.

Insight

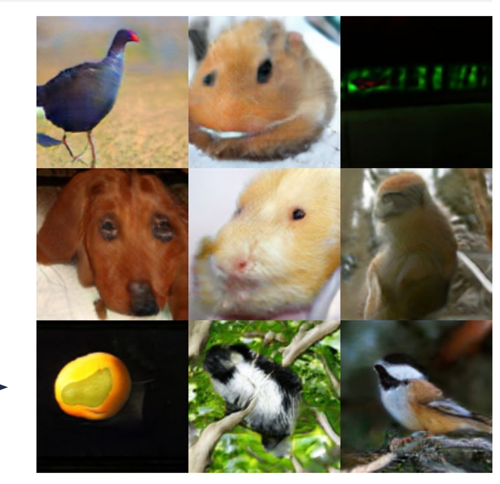
Generative adversarial networks (GANS) can be trained to generate inputs that are near to a given data distribution, and can learn to generate "realistic" images.



Examples of images generated using GANs.

MNIST

ImageNet



Solution

DiffNN$_1$

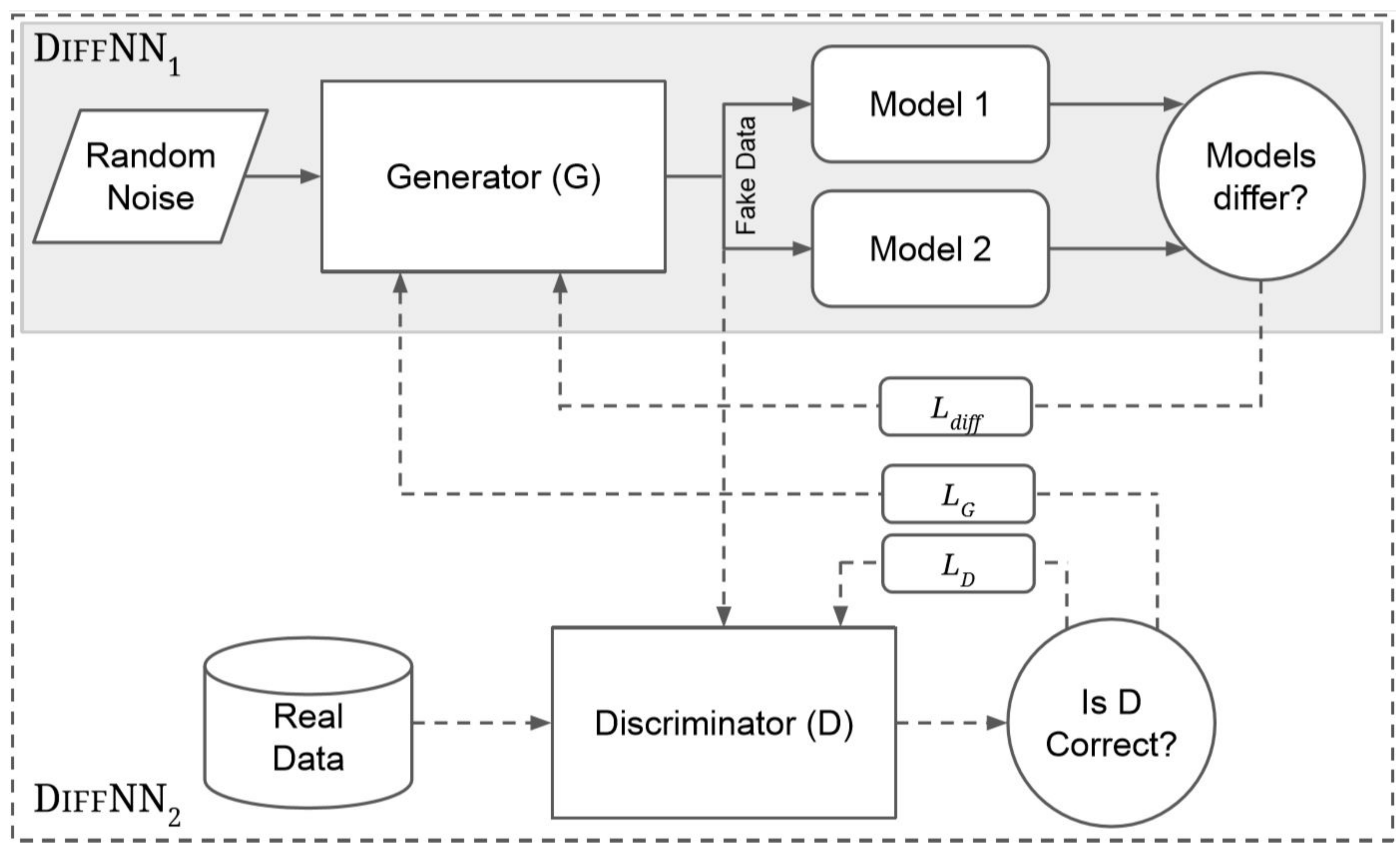
Given a pre-trained GAN, we use the generator to randomly sample inputs and check whether the two DNNs produce different outputs.

DiffNN$_2$

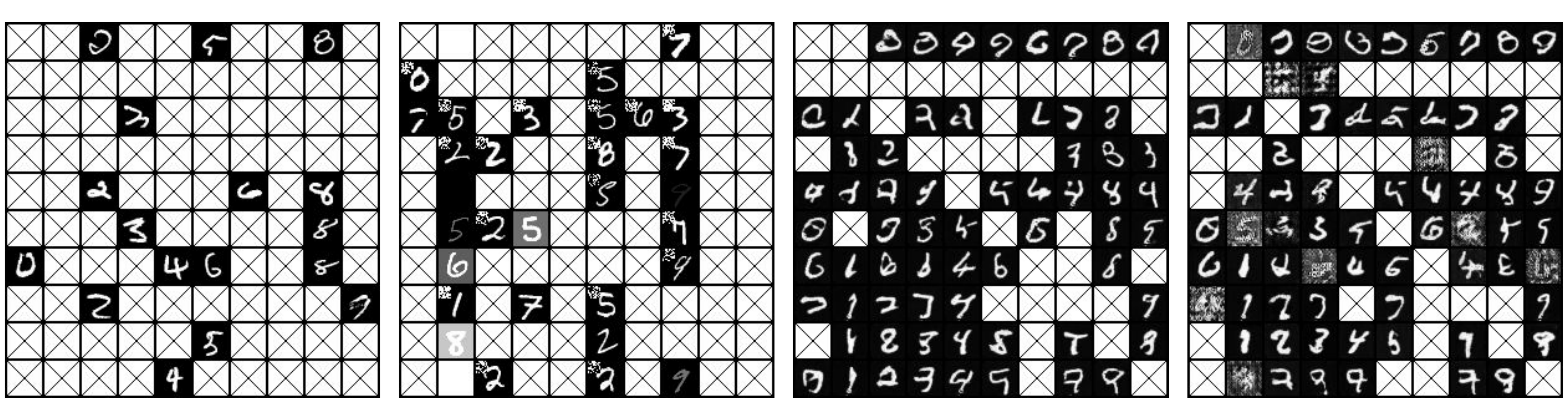
We modify the GAN training procedure to bias the generator towards inputs that are more likely to differentiate two DNNs.

We introduce an additional loss function for the generator that assigns high cost value to non-differentiating inputs.

$$L^{(c_1,c_2)}_{diff(\mathcal{N}_1,\mathcal{N}_1)}(x) = -\log(\mathcal{N}_1(x)[c_1]) - \log(\mathcal{N}_2(x)[c_2])$$

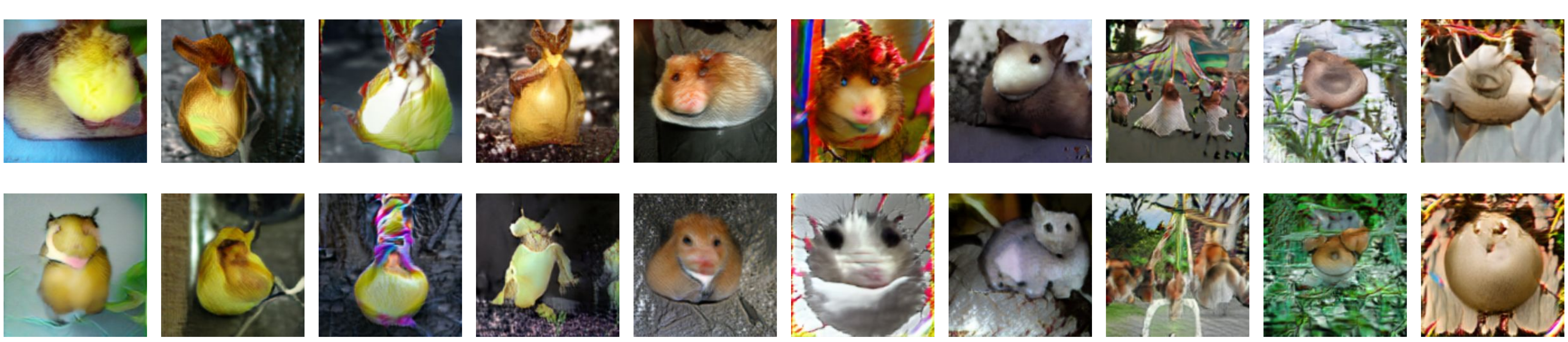


Results



Left:
Differentiating images for two MNIST networks using (from left to right) the test set, DeepXplore, DiffNN$_1$, and DiffNN$_2$.

Bottom:
Differentiating images between two ImageNet networks using DiffNN$_2$.



This work has been funded in part by NSF.