Supplement to Learning Augmentation Network via Influence Functions

Donghoon Lee Hyunsin Park Trung Pham Chang D. Yoo
Korea Advanced Institute of Science and Technology (KAIST)
{iamdh, hs.park, trungpx, cd_yoo}@kaist.ac.kr

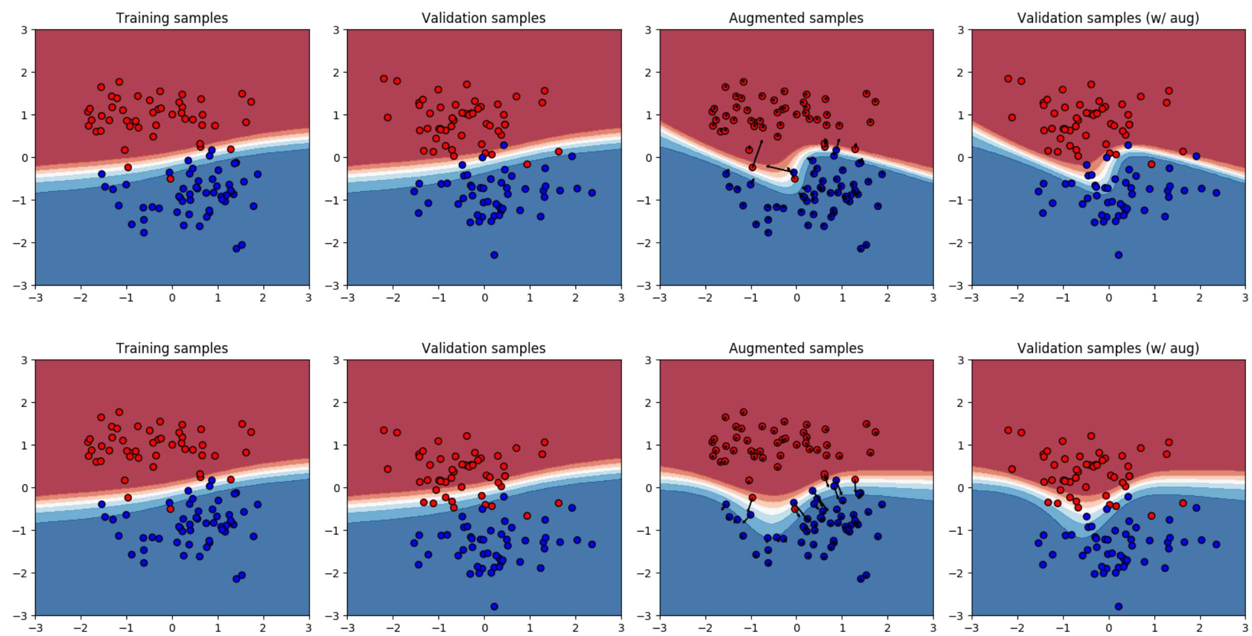

Figure 1: Toy example on a synthetic binary classification dataset. Red and blue dots represent data points of classes red and blue, respectively. These are overlaid with a contour plot of predictive probability. The columns represent: (i) training samples and the predictive probability contour plot from model parameters learned using training samples, (ii) validation samples and the predictive probability contour plot from model parameters learned using training sample, (iii) augmented samples that maximize influence and the predictive probability contour plot from model parameters learned using augmented samples, and (iv) validation samples with the predictive probability contour plot from model parameters learned using augmented samples.

A Analyzing using toy 2D dataset

To visualize the effect of the proposed method, experiments were conducted on a toy 2D dataset. Transitions along the x-axis and y-axis were considered in the augmentation model, with the influence computed by validation samples. A simple 3-layer neural network was considered for classification. In Figure 1, red and blue dots represent data points of classes red and blue respectively, overlaid with a contour plot of predictive probability.

In the first row of Figure 1, the influence value of augmented samples near the mis-classified validation sample is high. Thus, the augmentation model learns transitions towards the mis-classified validation samples. In the second row of Figure 1, training and validation data are generated. Validation data is then moved to the downside to make a mismatch with the training data. In this setting, the influence value of the downside transition is high, so the augmentation model learns the downside transition.

B Architectures and Hyper-parameters

Table 1: Architectures and optimization hyperparameters for experiments on MNIST dataset.

Encoder	
Input	$x \in \mathbb{R}^{28\times 28\times 1}$
Conv	4×4, stride 1, 1$\rightarrow$ 16, ReLU
Conv	4×4, stride 2, 16$\rightarrow$ 32, ReLU
Conv	4×4, stride 2, 32$\rightarrow$ 64, ReLU
Reshape	$7 \times 7 \times 64 \rightarrow 3136$
Linear	3136$\rightarrow$128, tanh
Output	$\tau \in \mathbb{R}^{128}$

Transformation Model	
Input	$\tau \in \mathbb{R}^{128}$
Linear	$128 \rightarrow 3136$
Reshape	$3136 \rightarrow 7 \times 7 \times 64$
ConvTr	4×4, stride 2, 64$\rightarrow$ 32, ReLU
ConvTr	4×4, stride 2, 32$\rightarrow$ 16, ReLU
ConvTr	4×4, stride 1, 16$\rightarrow$ 8, Identity
Output	$s \in \mathbb{R}^{28\times 28\times 6}$

Discriminator (image space)	
Input	$x \in \mathbb{R}^{28\times 28\times 1}$
Conv	4×4, stride 1, 1$\rightarrow$ 16, LeackyReLU 0.2
Conv	4×4, stride 2, 16$\rightarrow$ 32, LeackyReLU 0.2
Conv	4×4, stride 2, 32$\rightarrow$ 64, LeackyReLU 0.2
Reshape	$7 \times 7 \times 64 \rightarrow 3136$
Linear	3136$\rightarrow$ 1,
Output	$D(x) \in \mathbb{R}^1$

Optimizer	
G, D	ADAM, lr=.0002, $\beta = (.5, .999)$, batch=128
E, G	ADAM, lr=.01, $\beta = (.9, .99)$, batch=128

Table 2: Architectures and optimization hyperparameters for experiments on CIFAR-10 and CIFAR-100 datasets.

Encoder	
Input	$x \in \mathbb{R}^{32\times 32\times 3}$
Conv	4×4, stride 1, 1$\rightarrow$ 16, ReLU
Conv	4×4, stride 2, 16$\rightarrow$ 32, ReLU
Conv	4×4, stride 2, 32$\rightarrow$ 64, ReLU
Conv	4×4, stride 2, 64$\rightarrow$ 128, ReLU
Reshape	$4 \times 4 \times 128 \rightarrow 2048$
Linear	2048$\rightarrow$128, tanh
Output	$\tau \in \mathbb{R}^{128}$

Transformation Model	
Input	$\tau \in \mathbb{R}^{128}$
Linear	$128 \rightarrow 2048$
Reshape	$2048 \rightarrow 4 \times 4 \times 128$
ConvTr	4×4, stride 2, 128$\rightarrow$ 64, ReLU
ConvTr	4×4, stride 2, 64$\rightarrow$ 32, ReLU
ConvTr	4×4, stride 2, 32$\rightarrow$ 16, ReLU
ConvTr	4×4, stride 1, 16$\rightarrow$ 18, Identity
Output	$s \in \mathbb{R}^{32\times 32\times 18}$

Discriminator (image space)	
Input	$x \in \mathbb{R}^{28\times 28\times 1}$
Conv	4×4, stride 1, 1$\rightarrow$ 16, LeackyReLU 0.2
Conv	4×4, stride 2, 16$\rightarrow$ 32, LeackyReLU 0.2
Conv	4×4, stride 2, 32$\rightarrow$ 64, LeackyReLU 0.2
Conv	4×4, stride 2, 64$\rightarrow$ 128, LeackyReLU 0.2
Reshape	$4 \times 4 \times 128 \rightarrow 2048$
Linear	2048$\rightarrow$ 1,
Output	$D(x) \in \mathbb{R}^1$

Optimizer	
G, D	ADAM, lr=.0002, $\beta = (.5, .999)$, batch=128
E, G	ADAM, lr=.01, $\beta = (.9, .99)$, batch=128

Table 3: Architectures and optimization hyperparameters for experiments ImageNet dataset.

Encoder	
Input	$x \in \mathbb{R}^{224 \times 224 \times 3}$
Conv	7×7, stride 1, 1$\rightarrow$ 16, ReLU
Conv	4×4, stride 2, 16$\rightarrow$ 16, ReLU
Conv	4×4, stride 2, 16$\rightarrow$ 32, ReLU
Conv	4×4, stride 2, 32$\rightarrow$ 32, ReLU
Conv	4×4, stride 2, 32$\rightarrow$ 64, ReLU
Conv	4×4, stride 2, 64$\rightarrow$ 64, ReLU
Conv	4×4, stride 2, 64$\rightarrow$ 128, ReLU
Reshape	$4 \times 4 \times 128 \rightarrow 2048$
Linear	2048$\rightarrow$128, tanh
Output	$\tau \in \mathbb{R}^{128}$

Transformation Model	
Input	$\tau \in \mathbb{R}^{128}$
Linear	$128 \rightarrow 2048$
Reshape	$2048 \rightarrow 4 \times 4 \times 128$
ConvTr	7×7, stride 1, 128$\rightarrow$ 64, ReLU
ConvTr	4×4, stride 2, 64$\rightarrow$ 64, ReLU
ConvTr	4×4, stride 2, 64$\rightarrow$ 32, ReLU
ConvTr	4×4, stride 2, 32$\rightarrow$ 32, ReLU
ConvTr	4×4, stride 2, 32$\rightarrow$ 16, ReLU
ConvTr	4×4, stride 2, 16$\rightarrow$ 16, ReLU
ConvTr	4×4, stride 1, 16$\rightarrow$ 18, Identity
Output	$s \in \mathbb{R}^{224 \times 224 \times 18}$

Discriminator (image space)	
Input	$x \in \mathbb{R}^{224 \times 224 \times 3}$
Conv	7×7, stride 1, 1$\rightarrow$ 16, LeackyReLU 0.2
Conv	4×4, stride 2, 16$\rightarrow$ 16, LeackyReLU 0.2
Conv	4×4, stride 2, 16$\rightarrow$ 32, LeackyReLU 0.2
Conv	4×4, stride 2, 32$\rightarrow$ 32, LeackyReLU 0.2
Conv	4×4, stride 2, 32$\rightarrow$ 64, LeackyReLU 0.2
Conv	4×4, stride 2, 64$\rightarrow$ 64, LeackyReLU 0.2
Conv	4×4, stride 2, 64$\rightarrow$ 128, LeackyReLU 0.2
Reshape	$4 \times 4 \times 128 \rightarrow 2048$
Linear	2048$\rightarrow$128,
Output	$D(x) \in \mathbb{R}^{1}$

Optimizer	
G, D	ADAM, lr=.0002, $\beta = (.5, .999)$, batch=128
E, G	ADAM, lr=.01, $\beta = (.9, .99)$, batch=128