

Training generative neural networks using Maximum Mean Discrepancy

Vlad Shakhuro



10 March 2017

Equality of probability distributions

Let x and y be two random variables: $x \sim p(x)$, $y \sim q(y)$

Given observations

$$X = \{x_1, \dots, x_N\}, \quad Y = \{y_1, \dots, y_M\}$$

we'd like to decide whether

$$p \stackrel{?}{=} q$$

Maximum Mean Discrepancy

$$\text{MMD}[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} (E_p f(x) - E_q f(y))$$

Theorem

Let \mathcal{F} be a unit ball in a universal RKHS \mathcal{H} , defined on the compact metric space \mathcal{X} with associated kernel $k(\cdot, \cdot)$. Then $\text{MMD}[\mathcal{F}, p, q] = 0$ iff $p = q$.

Gretton et al. A Kernel Two-Sample Test. JMLR 2012

Maximum Mean Discrepancy

Given that $k(x, y) = \langle \phi(x), \phi(y) \rangle$, we can obtain:

$$\begin{aligned} \text{MMD}^2[\mathcal{H}, p, q] &= \|E_p \phi(x) - E_q \phi(y)\|^2 = \\ E_{pp} \langle \phi(x), \phi(x') \rangle - E_{pq} \langle \phi(x), \phi(y) \rangle + E_{qq} \langle \phi(y), \phi(y') \rangle &= \\ E_{pp} k(x, x') - E_{pq} k(x, y) + E_{qq} k(y, y') \end{aligned}$$

Maximum Mean Discrepancy

Given that $k(x, y) = \langle \phi(x), \phi(y) \rangle$, we can obtain:

$$\text{MMD}^2[\mathcal{H}, p, q] = \|E_p \phi(x) - E_q \phi(y)\|^2 =$$

$$E_{pp} \langle \phi(x), \phi(x') \rangle - E_{pq} \langle \phi(x), \phi(y) \rangle + E_{qq} \langle \phi(y), \phi(y') \rangle =$$

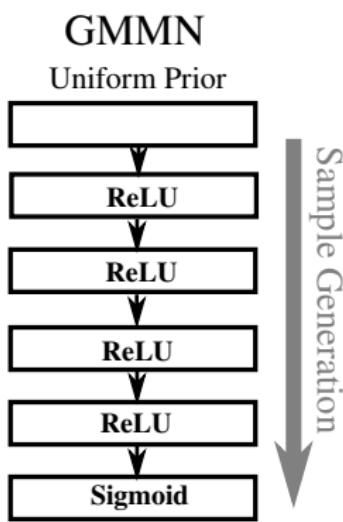
$$E_{pp} k(x, x') - E_{pq} k(x, y) + E_{qq} k(y, y')$$

Biased estimate for practical use:

$$\text{MMD}_b^2[\mathcal{H}, X, Y] =$$

$$\frac{1}{N^2} \sum_{n, n'} k(x_n, x_{n'}) - \frac{2}{MN} \sum_{n, m} k(x_n, y_m) + \frac{1}{M^2} \sum_{m, m'} k(y_m, y_{m'})$$

MMD neural network



Training:

given dataset $\{x_1^d, \dots, x_N^d\}$, prior $p(z)$, neural net $G(z; w)$

while not stop:

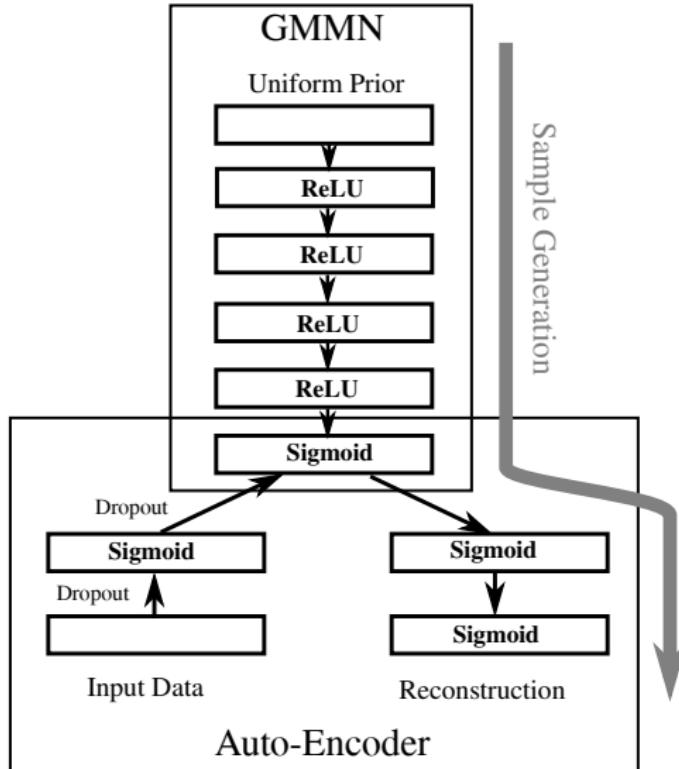
get a minibatch of data X^d

generate samples X^s

compute $\frac{\partial \text{MMD}}{\partial w}$ on X^d and X^s

update w using gradient

MMD + Autoencoder



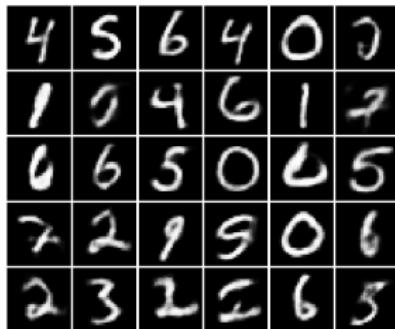
Results



(a) GMMN MNIST samples



(b) GMMN TFD samples



(c) GMMN+AE MNIST samples



(d) GMMN+AE TFD samples

Results (adversarial autoencoders)



(a) MNIST samples (8-D Gaussian)



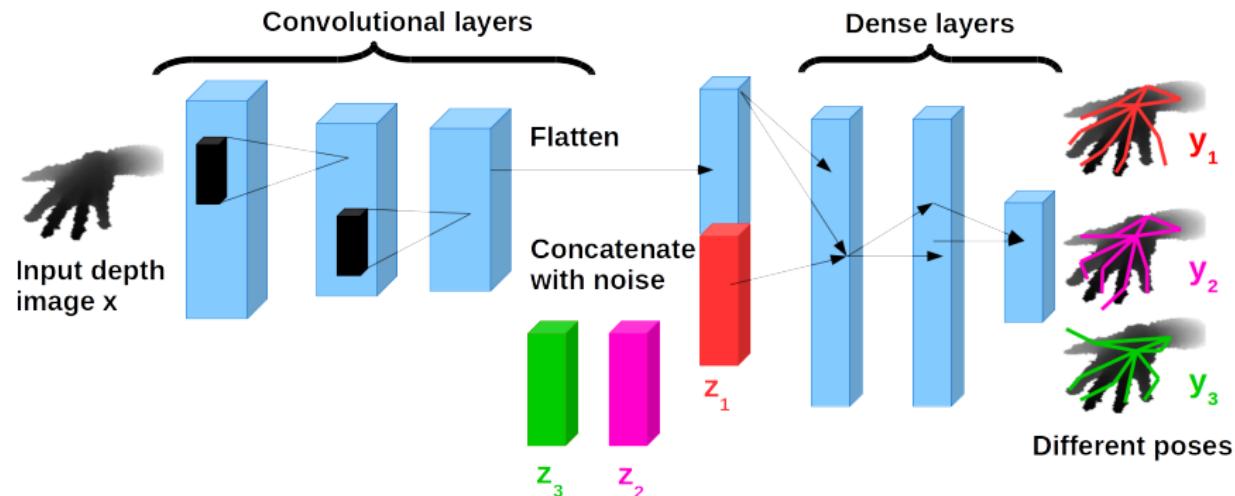
(b) TFD samples (15-D Gaussian)

Generative net for posterior distribution

Given training dataset $\{(x_1, y_1), \dots, (x_m, y_m)\}$,
we'd like to learn $p(y|x)$

Generative net for posterior distribution

Given training dataset $\{(x_1, y_1), \dots, (x_m, y_m)\}$,
we'd like to learn $p(y|x)$



Bouchacourt et al. DISCO Nets: DISsimilarity COefficient Networks. NIPS 2016

Objective function

We want $p_G(y|x)$ be similar to $p_T(y|x)$ evaluated with respect to task-specific loss function $\Delta(y, y')$

$$\text{DIV}_\Delta(p_T, p_G) = \int_x \int_y \int_{y'} \Delta(y, y') p_T(y|x) p_G(y'|x) p_T(x) dy dy' dx$$

Objective function

We want $p_G(y|x)$ be similar to $p_T(y|x)$ evaluated with respect to task-specific loss function $\Delta(y, y')$

$$\text{DIV}_\Delta(p_T, p_G) = \int_x \int_y \int_{y'} \Delta(y, y') p_T(y|x) p_G(y'|x) p_T(x) dy dy' dx$$

$$\begin{aligned} \text{DISC}_\Delta(p_T, p_G) &= \text{DIV}_\Delta(p_T, p_G) - \gamma \text{DIV}_\Delta(p_G, p_G) - (1 - \gamma) \text{DIV}_\Delta(p_T, p_T) \end{aligned}$$

Objective function

We want $p_G(y|x)$ be similar to $p_T(y|x)$ evaluated with respect to task-specific loss function $\Delta(y, y')$

$$\text{DIV}_\Delta(p_T, p_G) = \int_x \int_y \int_{y'} \Delta(y, y') p_T(y|x) p_G(y'|x) p_T(x) dy dy' dx$$

$$\text{DISC}_\Delta(p_T, p_G) = \text{DIV}_\Delta(p_T, p_G) - \gamma \text{DIV}_\Delta(p_G, p_G) - (1 - \gamma) \text{DIV}_\Delta(p_T, p_T)$$

$$F = \text{DIV}_\Delta(p_T, p_G) - \gamma \text{DIV}_\Delta(p_G, p_G)$$

Rao. Diversity and dissimilarity coefficients: A unified approach.
Theoretical Population Biology, 1982

Unbiased estimates

$$\text{DIV}_\Delta(p_T, p_G) = \frac{1}{NK} \sum_{n=1}^N \sum_{k=1}^K \Delta(y_n, G(z_k, x_n))$$

$$\begin{aligned} \text{DIV}_\Delta(p_G, p_G) = \\ \frac{1}{NK(K-1)} \sum_{n=1}^N \sum_{k,k' \neq k} \Delta(G(z_k, x_n), G(z_{k'}, x_n)) \end{aligned}$$

Unbiased estimates

$$\text{DIV}_\Delta(p_T, p_G) = \frac{1}{NK} \sum_{n=1}^N \sum_{k=1}^K \Delta(y_n, G(z_k, x_n))$$

$$\begin{aligned} \text{DIV}_\Delta(p_G, p_G) = \\ \frac{1}{NK(K-1)} \sum_{n=1}^N \sum_{k,k' \neq k} \Delta(G(z_k, x_n), G(z_{k'}, x_n)) \end{aligned}$$

If $\gamma = \frac{1}{2}$ and $\Delta(y, y') = \|y - y'\|_2$, then

$$\begin{aligned} F = \frac{1}{N} \sum_n \left(\frac{1}{K} \sum_k \|y_n - G(z_k, x_n)\|_2 - \right. \\ \left. \frac{1}{2K(K-1)} \sum_{k,k' \neq k} \|G(z_{k'}, x_n) - G(z_k, x_n)\|_2 \right) \end{aligned}$$

F is minimised iff $p_T = p_G$

Gneiting, Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association, 2007

Metrics

Table 2: Evaluation metrics. N is the number of testing example pairs $(\mathbf{x}_n, \mathbf{y}_n)$ and $\mathbf{y}_{\Delta_{metric}, n}$ is the prediction, specific to the metric, for the n^{th} input example \mathbf{x}_n . J is the number of joints of the hand.

Shorthand & Definition	Δ_{metric}	Formula
ProbLoss: Probabilistic Loss of K sampled poses.	$\Delta_{\text{ProbLoss}} = \ \cdot\ _2^\beta$	$\widehat{F}(\Delta_{\text{ProbLoss}}, \theta)$ with $\gamma = 0.5$
MeJEE (Mean Joint Euclidean Error) : per-joint Euclidean distance between the pointwise pose and the groundtruth, averaged by J and N .	$\Delta_{\text{MeJEE}} = \frac{1}{J} \sum_{j=1}^J \ \cdot\ _2^{\text{joint}j}$	$\frac{1}{N} \sum_{n=1}^N \frac{1}{J} \sum_{j=1}^J \ \mathbf{y}_n^j - \mathbf{y}_{\Delta_{\text{MeJEE}}, n}^j\ _2^{\text{joint}j}$
MaJEE (Max Joint Euclidean Error) per-joint maximal Euclidean distance between the pointwise pose and the groundtruth averaged by N .	$\Delta_{\text{MaJEE}} = \max_{j \in [1, J]} \ \cdot\ _2^{\text{joint}j}$	$\frac{1}{N} \sum_{n=1}^N \max_{j \in [1, J]} \ \mathbf{y}_n^j - \mathbf{y}_{\Delta_{\text{MaJEE}}, n}^j\ _2^{\text{joint}j}$
FF(d) (Fraction of Frames) : fraction of test examples that have all predicted joints of the pointwise pose below a given maximum Euclidean distance d in mm from the ground-truth.	$\Delta_{\text{FF}} = -\mathbb{1}_{\max_{j \in [1, J]} \ \cdot\ _2^{\text{joint}j} \leq d}$ (note the minus sign since we want to maximise FF)	$\frac{1}{N} \sum_{n=1}^N \mathbb{1}_{\max_{j \in [1, J]} \ \mathbf{y}_n^j - \mathbf{y}_{\Delta_{\text{FF}}, n}^j\ _2^{\text{joint}j} \leq d}$

Getting point prediction from samples

For fixed x , we generate K samples and get one that minimizes expected task-specific loss Δ , estimated using generated samples

$$y = \arg \min_{k=1 \dots K} \sum_{k'=1}^K \Delta(y_k, y_{k'})$$

Results

Table 3: Metrics values on the test set \pm SEM. Best performances in bold.

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
BASE $_{\beta=1, \sigma=1}$	103.8 \pm 0.627	25.2 \pm 0.152	52.7 \pm 0.290	86.040
BASE $_{\beta=1, \sigma=5}$	99.3 \pm 0.620	25.5 \pm 0.151	52.9 \pm 0.289	85.773
BASE $_{\beta=1, \sigma=10}$	96.3 \pm 0.612	25.7 \pm 0.149	53.2 \pm 0.288	85.664
DISCO $_{\beta=1, \gamma=0.5}$	83.8 \pm 0.503	20.9 \pm 0.124	45.1 \pm 0.246	94.438

Table 4: Metrics values on the test set \pm SEM for cGAN and DISCO Nets. Best performances in bold.

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
cGAN	442.7 \pm 0.513	109.8 \pm 0.128	201.4 \pm 0.320	0.000
cGAN _{init, fixed}	128.9 \pm 0.480	31.8 \pm 0.117	64.3 \pm 0.230	78.454
DISCO $_{\beta=1, \gamma=0.5}$	83.8 \pm 0.503	20.9 \pm 0.124	45.1 \pm 0.246	94.438

Table 5: DISCO Nets compared to state-of-the-art performances \pm SEM.

Model	MeJEE (mm)	MaJEE (mm)	FF (80mm)
NYU-Prior	20.7 \pm 0.150	44.8 \pm 0.289	91.190
NYU-Prior-Refined	19.7 \pm 0.157	44.7 \pm 0.327	88.148
NYU-Init	27.4 \pm 0.152	55.4 \pm 0.265	86.537
NYU-Feedback	16.0 \pm 0.096	36.1 \pm 0.208	97.334
DISCO $_{\beta=1, \gamma=0.5}$	20.7 \pm 0.121	45.1 \pm 0.246	93.250

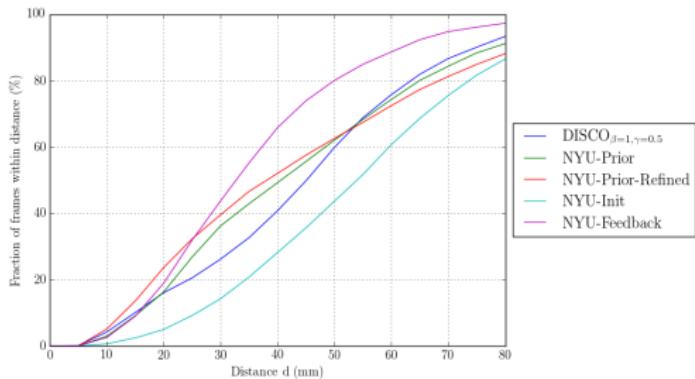


Figure 4: Fractions of frames within distance d in mm (by 5 mm). Best viewed in color.

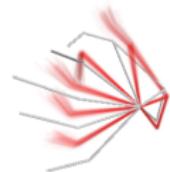
Results



(a) When there are no occlusions, DISCO Nets model low uncertainty on all joints.



(b) When the hand is half-fisted, DISCO Nets model the uncertainty on the location of the fingertip.



(c) Here the fingertips of all fingers but the forefinger are occluded and DISCO Nets model high uncertainty on them.

Figure 3: Visualisation of $DISCO_{\beta=1, \gamma=0.5}$ predictions for 3 examples from the testing dataset. The top image shows the input depth image, and the bottom image shows the ground-truth pose in grey with 100 candidate outputs superimposed in transparent red. Best viewed in color.

Summary

MMD is a theoretically grounded loss function for training generative models

MMD is easier and faster to implement and train than GANs

Further work on kernels for conditional prediction is required