Modeling Uncertainty with Hedged Instance Embedding

Seong Joon Oh[1*] Kevin Murphy[2] Jiyan Pan[2] Joseph Roth[2] Florian Schroff[2] Andrew Gallagher[2]

[1]Clova AI Research, LINE Plus Corp. [2]Google Research [*]Intern project at Google Research

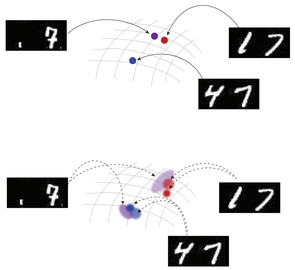
Problem & Motivation

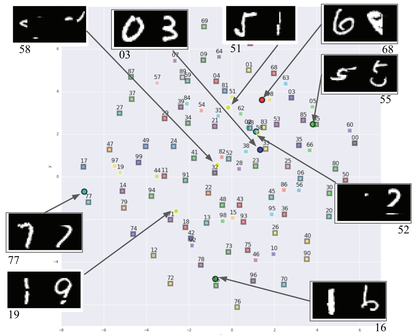
How to properly represent uncertain inputs for instance embedding?

Point embeddings cannot handle uncertain inputs.

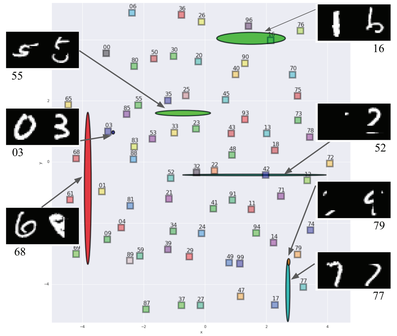
Hedged embeddings properly model them as probability distributions (e.g. MoG) over the embedding space.

- Improved performance.
- Uncertainty measure.
- Meaningful candidate answers.





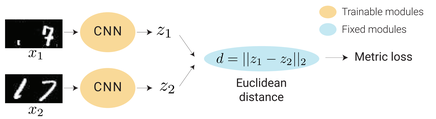
Point embeddings: uncertain inputs are mapped to arbitrary locations.



Hedged embeddings: distributions for uncertain inputs cover plausible candidate classes.

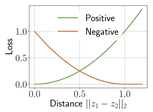
Point Embeddings

Deterministic Siamese network



Contrastive loss

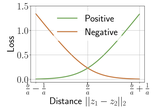
$$\mathcal{L}_{\text{con}} = \begin{cases} ||z_1 - z_2||_2^2 & \text{if match} \\ \max(M - ||z_1 - z_2||_2, 0)^2 & \text{if non-match} \end{cases}$$



Match probability loss

$$\mathcal{L}_{\text{mp}} = \begin{cases} -\log p(m|z_1, z_2) & \text{if match} \\ -\log(1 - p(m|z_1, z_2)) & \text{if non-match} \end{cases}$$

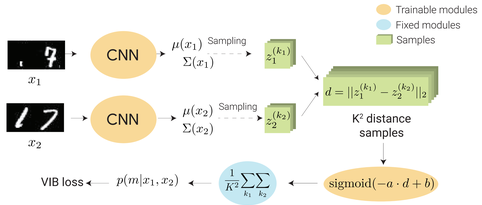
$$p(m|z_1, z_2) := \text{sigmoid}(-a||z_1 - z_2||_2 + b)$$



- Probabilistic treatment of contrastive loss.
- (a,b) are learnable; no need to manually select margin M.

Hedged Embeddings

Stochastic Siamese network



Single Gaussian:

- Shared CNN trunk.
- Two 1-layer branches for (μ, σ).

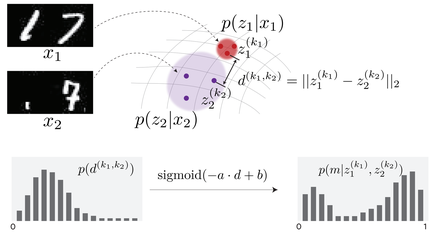
Mixture of Gaussians (C centroids):

- Shared CNN trunk.
- 2C 1-layer branches for (μc, σc).

Training objective: Variational information bottleneck

$$\mathcal{L}_{\text{VIB}} := -\mathbb{E}_{z_1 \sim p(z_1|x_1), z_2 \sim p(z_2|x_2)} \left[\log p(m = \hat{m}|z_1, z_2)\right] + \beta \cdot \left[\text{KL}(p(z_1|x_1)||r(z_1)) + \text{KL}(p(z_2|x_2)||r(z_2))\right]$$

Match probability computation

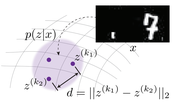


$$p(m|x_1, x_2) \approx \frac{1}{K^2}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K} p\left(m|z_1^{(k_1)}, z_2^{(k_2)}\right)$$

Uncertainty measure: self-mismatch

$$\eta(x) := 1 - p(m|x, x) \geq 0$$

For an uncertain input, two samples from the embedding are unlikely to match.



N-Digit MNIST Experiments

Dataset

https://github.com/google/n-digit-mnist

28 x (N x 28) images of 10^N classes embedded onto D dim space.

Introduce corruption (black patch) for uncertainty evaluation.

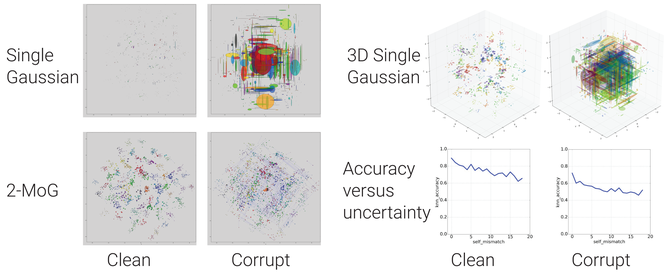
N=3 sample "958"

Random corruption

HIB performs better than point embeddings (kNN).

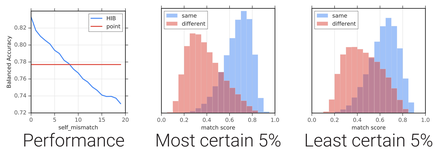
	$N=2, D=2$			$N=2, D=3$			$N=3, D=2$			$N=3, D=3$		
	Point	1-Gauss	2-MoG	Point	1-Gauss	2-MoG	Point	1-Gauss	2-MoG	Point	1-Gauss	2-MoG
clean	0.871	0.879	0.888	0.942	0.953	0.939	0.554	0.591	0.540	0.795	0.770	0.766
corrupt	0.583	0.760	0.757	0.874	0.909	0.885	0.274	0.350	0.351	0.522	0.555	0.598

HIB represents uncertainty - it knows when it doesn't know.



Pet Identification Experiment

Pet images (128 x 128 x 3) of 118k identities embedded onto D=20 dim space.



Performance | Most certain 5% | Least certain 5%

- Realistic dataset.
- HIB accuracy (77.6%) is similar to point (77.7%).
- HIB knows what it does not know.