Paper reading notes of adversarial generation of continuous implicit shape representations

Sunday, December 27, 2020 4:23 PM

Adversarial generation of implicit shape representations.

- Given a random latent code, our generator produces a continuous 3D SDF, while the discriminator's job is to provide useful feedback to improve the generator.
- Signed distance functions: given a spatial point $p \in \mathbb{R}^3$, the $SDF(p) \in \mathbb{R}$ encodes the point's distance to its closest surface point, where the sign indicates whether p lies inside − or outside + the object.
  - SDF allows us to sample a surface point cloud $S = \{s_1, \dots, s_N\}$ of arbitrary cardinality N by translating uniformly sampled points $p_i \in P = \{p_1 \dots p_N\}$, to their closest surface point

$$s_i = p_i - SDF(p_i) \cdot \nabla_{p_i} SDF(p_i)$$

- Generator. Given a low dimensional encoding $z \in \mathbb{R}^d$ of a complete shape, the DeepSDF decoder g_θ learns the mapping

$$g_\theta(z, p) \approx SDF(p).$$

  g_θ is parameterized via trainable parameters θ, and is implemented as an MLP.

  - Since g_θ is conditioned on a latent code z, the same neural network can be used to model SDFs of multiple objects.
  - Note that g_θ processes only a single point, but can be trained against the GT SDF.
- We provide context to the discriminator by evaluating the generator for a batch of points P, and inject the latent code z to each point $p_i \in P$ individually:

$$g_\theta(z, P)_i = g_\theta(z, p_i)$$

Voxel based discriminator

- Use a 3D CNN as our discriminator d_θ, both $g_\theta(z,P)$ and $d_\theta(g_\theta(z,P), P)$ expect input points P to describe a regular grid of fixed resolution.

Point based discriminator

- We input any point by making use of the PointNet. as the discriminator d_θ.
- uniformly distributed points are first transformed independently into a high dimensional space, before a joint representation is obtained using the permutation-invariant max-pooling operation:

$$d_\theta(g_\theta(z,P), P) = \gamma_\theta\left(\max_{p \in P} h_\theta(g_\theta(z,P), P)\right) \in \mathbb{R}.$$

$P \sim U(-1,1)^3$,

γ_θ, h_θ are trainable MLPs.

- In contrast to PointNet, we also inject the SDF to each raw point, so g_θ takes both positional and SDF info.
- Compared to voxel-based approach, this formulation allows g_θ to know about any point in space, instead of solely being required to generate reasonable outputs for a fixed number of stationary points.

Zero ISO-surface decision boundary.

- While DeepSDF aggressively samples near the object surface, this solution cannot be applied since we do not know the object surface in advance.
- $g_\theta(z,P)$ already provides coarse-grained surface of an object. We strength the generator's representational power by sampling additional points near the surface of an object, based on the generated SDF values of uniformly distributed points.

$$\tilde{g}_\theta(z,P) = g_\theta(z,P) \bigcup_{\substack{p \in P \\ |g_\theta(z,p)| < \delta}} \{g_\theta(z, p - g_\theta(z,p) \cdot \nabla_p g_\theta(z,p) + \varepsilon)\}$$

$\varepsilon \sim N(0, \delta^2)$.

  - For each $p \in P$ that is sufficiently close to a predicted surface ($|g_\theta(z,p)| < \delta$), we project it onto the surface using the gradients of g_θ wrt p, and sample additional points following a Gaussian distribution. The discriminator then takes the refined point set:

$$\tilde{P} = P \cup \{p - g_\theta(z,p) \cdot \nabla_p g_\theta(z,p) + \varepsilon\}$$

  and their SDFs as input, and can hence more specifically draw its attention to the modeling of the zero iso surface decision boundary.

Training.

- We use pre-processed GT $SDF(P)$ from human designed meshes.
- We input both generated and real samples $d_\theta(g_\theta(z,P), P)$ and $d_\theta(SDF(P), P)$ to d_θ, optimizing the GAN minimax objective.