Sampling Components

Contents

Sampling methods.

class **anomalib.models.components.sampling.KCenterGreedy(*embedding*, *sampling_ratio*)**

Bases: `object`

Implements k-center-greedy method.

Parameters:

- **embedding** (*torch.Tensor*) – Embedding vector extracted from a CNN
- **sampling_ratio** (*float*) – Ratio to choose coreset size from the embedding size.

Example

```
>>> embedding.shape
torch.Size([219520, 1536])
>>> sampler = KCenterGreedy(embedding=embedding)
>>> sampled_idxs = sampler.select_coreset_idxs()
>>> coreset = embedding[sampled_idxs]
>>> coreset.shape
torch.Size([219, 1536])
```

get_new_idx()

Get index value of a sample.

Based on minimum distance of the cluster

Returns:

Sample index

Return type:

int

reset_distances()

Reset minimum distances.

Return type:

None

sample_coreset(*selected_idxs=None*)

Select coreset from the embedding.

Parameters:

selected_idxs (list[int] | None) – index of samples already selected. Defaults to an empty set.

Returns:

Output coreset

Return type:

Tensor

Example

```
>>> embedding.shape
torch.Size([219520, 1536])
>>> sampler = KCenterGreedy(...)
>>> coreset = sampler.sample_coreset()
>>> coreset.shape
torch.Size([219, 1536])
```

select_coreset_idxs(*selected_idxs=None*)

Greedily form a coreset to minimize the maximum distance of a cluster.

Parameters:

selected_idxs (`list`[`int`] | `None`) – index of samples already selected. Defaults to an empty set.

Return type:

`list`[`int`]

Returns:

indices of samples selected to minimize distance to cluster centers

`update_distances`(*`cluster_centers`*)

Update min distances given cluster centers.

Parameters:

cluster_centers (*list[int]*) – indices of cluster centers

Return type:

`None`

Previous
Normalizing Flows

Next
Filters