Exploring distances in the latent space between ood and id samples

Constantin Gabriel-Adrian, Hernest Mihai
Supervisor: Elena Burceanu

Introduction

The problem of generalization occurs as a natural result of the current training paradigms: we usually train on a specific dataset and we validate it's performance using samples that are in the same distribution, however concrete real world data usually are not part of the same distribution.
We explore distances in the latent space between ood and id samples on *Camelyon17* dataset [1] which represents distribution shift in medical images. We present the difference in accuracy between samples in and out of distribution while exploring ways to predict the distribution of an image. Examples of what we set out to quantify are: *Contrastive loss* [2] which measures the similarity between 2 distributions, *Triplet loss* [3] which chooses 3 anchors: 2 positive and one negative. They are used with distance metric learning networks(DML) such as *Siamese networks*.

Dataset

We investigate this distance on the *Camelyon17* dataset which offers us medical images of regions with or without tumor tissue. It is composed of 4 datasets, and we used one third of the data:

- 99804 *train* images
- 11075 *id_val* images
- 11518 *ood_val* images
- 28068 *ood_test* images

In order to avoid bias, we randomly select the samples used while maintaining the ratio between train and validation sets.

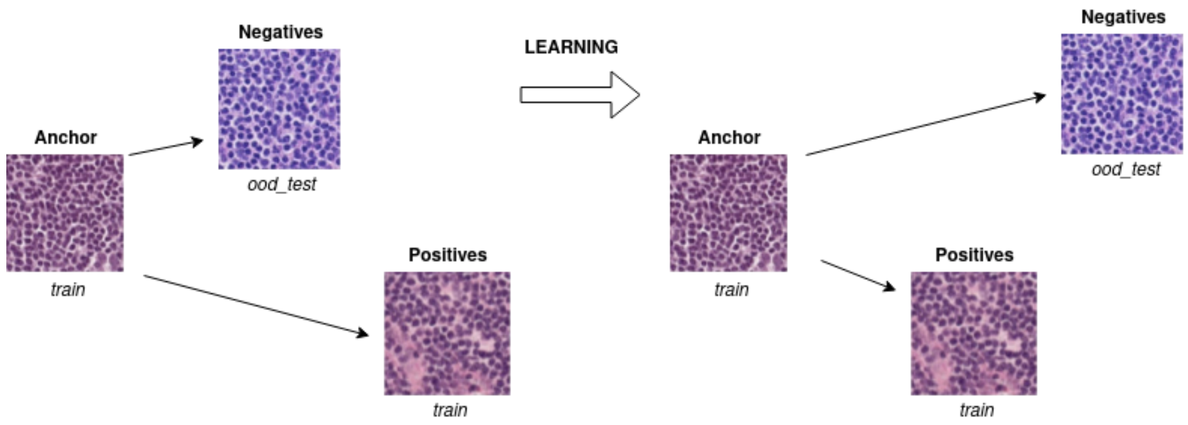
The Proposed Approach

We first train 3 models for label classification with the purpose of measuring the difference in accuracy between in and out of distribution data. We then attempt to predict for a random sample whether or not it is part of the same distribution as the training dataset. For this task, we explore 2 approaches:

Measure the distance between embeddings One way we achieve this is by using the feature extractor from our model to compute the embedding for the given sample and to calculate for the training data the centroid in the embedding space for each class. Next, we analyze the distance between these two embeddings using multiple metrics such as cosine similarity, L1, L2 and Minkowski distances.

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Contrastive learning Another way is to learn the aforementioned distance in a contrastive manner. In order to use contrastive learning to differentiate between samples in and out of distribution, we assign different labels for different distributions. Since these approach uses pairs of samples, we consider a pair to be positive (i.e. similar images) if they are part of the same distribution. The others are considered negative pairs.



Hyperparameters Optimization

- loss function: *CrossEntropy*
- optimizer: *SGD with momentum*
- number of epochs: *5*
- batch size: *32*

Conclusions and Feature Work

The cosine similarity performs better as a metric compared to our approach of implementing a contrastive learning model.
We remark a difference between in and out of distribution data. This gap is a lot more evident for class 1 compared to class 0 (when both images show tissue with no tumor). It is probably due to the fact that a tissue with no tumor looks similar even though the photo was taken with different medical devices.
Either way, the results look promising for both cosine similarity and contrastive learning approach. To further inspect this claim, we could analyze the results of the presented approaches on a different dataset such as *CIFAR10* and compare the results.

References

[1] Bandi, Peter and Geessink, Oscar and Manson, Quirine and Van Dijk, Marcory and Balkenhol, Maschenka and Hermsen, Meyke and Bejnordi, Babak Ehteshami and Lee, Byungjae and Paeng, Kyunghyun and Zhong, Aoxiao and others, "From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge" in IEEE Transactions on Medical Imaging, 2018

[2] Y. Sun andothers. "Deep learning face representation by joint identification-verification". in "Advances in Neural Information Processing Systems": (2014), pages 1988–1996.

[3] Elad Hoffer and Nir Ailon "Deep metric learning using triplet network" in International Workshop on Similarity-Based Pattern Recognition, Springer, pages 84–92 2015.

[4] Kawin Ethayarajh, "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings" in abs/1909.00512, 2009

Experiments and Methodology

The proposed network architectures are *ResNet18*, *DenseNet121* and *ResNeXt50*, all of them pre-trained on *ImageNet*. In order to leverage the network's previously existing knowledge, we replace the classifier for each aforementioned architecture with a linear layer that has 2 output neurons.

Feature extraction We train only the classifier on top of the features extracted from the pre-trained model.

Fine-tuning We retrain the entire model from the aforementioned (pre-trained) weights.

Prediction performance of models using different types of training

Trained approaches	Backbone	Accuracy		
		id_val	***ood_val***	***ood_test***
Feature extraction	ResNet18	89.79	81.17	75.61
	DenseNet121	93.32	85.03	76.98
	ResNeXt50	89.65	78.38	55.57
Fine-tuning	ResNet18	97.99	87.64	77.62
	DenseNet121	98.49	85.78	**79.53**
	ResNeXt50	**98.78**	**87.87**	72.22

We proceed to analyze the distance between the centroid of each class in the embedding space of training data and the embedding of a random sample in order to test if we can predict beforehand whether or not a given sample comes from another distribution. For each sample from both validations datasets, we measure the average distance with multiple metrics L1, L2, Minkowski and cosine similarity.

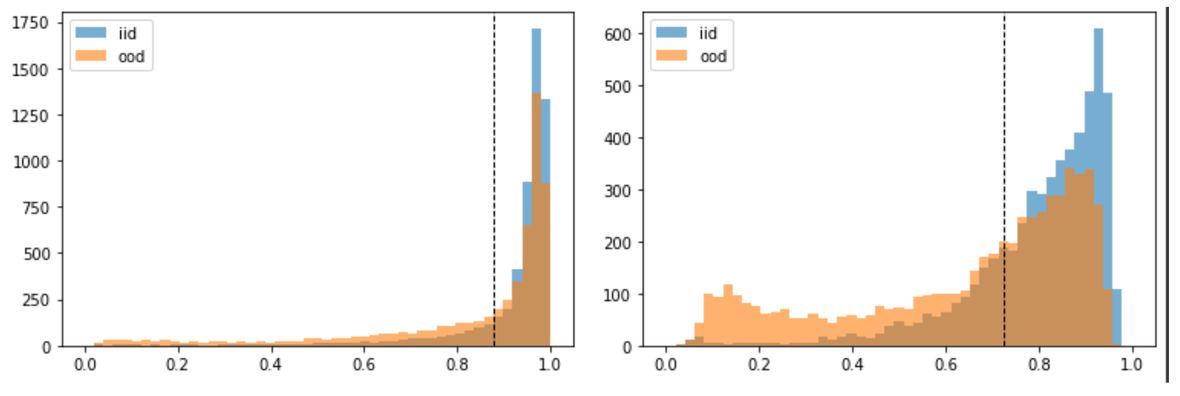
Distance metrics for each trained model

Model	Metric	No tumor		Tumor	
		id_val*	**ood_val***	**id_val***	**ood_val***
ResNet18	L1	62.41 ± 28.90	68.86 ± 30.06	839 ± 605	926 ± 718
	L2	5.99 ± 2.87	6.23 ± 2.34	42.03 ± 29.40	45.44 ± 35.77
	cosine	0.87 ± 0.13	0.73 ± 0.24	0.77 ± 0.16	0.72 ± 0.19
	Mink.	3.17 ± 1.47	3.25 ± 1.17	16.17 ± 11.15	17.28 ± 13.76
DenseNet121	L1	2042 ± 828	2094 ± 822	5861 ± 4817	4683 ± 1137
	L2	47.03 ± 20.78	47.71 ± 21.21	107 ± 89.48	82.50 ± 22.87
	cosine	0.65 ± 0.14	0.58 ± 0.20	0.42 ± 0.35	0.23 ± 0.34
	Mink.	16.25 ± 7.40	16.58 ± 7.72	31.66 ± 25.81	23.93 ± 7.36
ResNeXt50	L1	682 ± 179	683 ± 184	606 ± 137	754 ± 196
	L2	6.07 ± 3.14	6.74 ± 3.31	29.30 ± 25.24	27.75 ± 14.35
	cosine	**0.92 ± 0.11**	**0.82 ± 0.22**	**0.80 ± 0.15**	**0.65 ± 0.25**
	Mink.	2.46 ± 1.23	2.77 ± 1.30	9.06 ± 7.64	8.49 ± 4.40

* metrics are reported as mean ± standard deviation

Even though it usually performs poorly on transformers [4], the cosine similarity seems to be the most clear-cut metric. This behaviour is probably due to the fact that our layers are anisotropic. The geometric interpretation of anisotropy is that the image representations all occupy a narrow cone in the vector space rather than being uniform in all directions; the greater the anisotropy, the narrower this cone. This is exactly what CrossEntropy strives to achieve: vectors of the same class to form a cone along an axis.

Cosine similarity for both classes using fine-tuned ResNeXt50 (left: no tumor | right: tumor)



As expected, the in distribution samples are characterized by a greater similarity with the centroid than out of distribution ones.

Now we try to learn the aforementioned distance in a contrastive manner. For this, we use a *Siamese Network* with the feature extractor from our *ResNet18* model. We create positive pairs (i.e. similar images, they are part of the same distribution) and negative pairs. Since we need out of distribution samples, we now use the *test* dataset as part of the training process. Once again we see that out of distribution samples are more dissimilar.

Dissimilarity metric for contrastive learning

Model	id_val	ood_val
ResNet18	0.220 ± 0.3	0.234 ± 0.33