FERM: A FEature-space Representation Measure for Improved Model Evaluation

Yeu Shin Fu[0,1], Wenbo Ge[0,1] and Jo Plested[0,2]

[0]Equal contribution
[1]Australian National University, Canberra, ACT 2601, Australia
[2]University of New South Wales, Northcott Dr, Campbell ACT 2612, Australia

Abstract

Understanding whether a particular dataset and task are well represented by a deep learning model can be as crucial as the model's prediction accuracy in many applications. Currently, best prediction performance for large, modern datasets is often achieved by complex and difficult to interpret deep learning models. As deep learning model size and complexity increases compared to the size of the training dataset, the capacity of the model to overfit to inappropriate features and perform poorly or unreliably also increases. Unreliability may not be obvious in traditional performance measures during evaluation so it is important to also consider how well the model is representing the current data distribution. There has previously been little work focusing on measuring this space. We introduce several measures that we collectively name FERM: A FEature-space Representation Measure for determining how well the current feature space representation models the current data distribution and task. We compared our new measures and potential candidates from other related research areas. We demonstrated that our new method, along with two others, have excellent potential to be used for measuring how well a trained model is currently representing a dataset and task. These findings have many implications for deep learning research and applications, including, evaluating when the current model is no longer representing new data well to reduce the frequency of computationally expensive retraining of models, assessing for hard to evaluate failure modes such as model biases that result in particular input samples being poorly represented, guidance on the best hyperparameters to use when updating models with limited new data.

Keywords

Representation learning, Feature space evaluation, Deep learning,

1. Introduction

With the successes of deep learning in the past decade when applied to modelling large well formed and stable data distributions, recent focus has turned to modelling datasets that are:

1. Not well formed because they are very different to the source dataset in the case of some transfer learning applications.
2. Not stable over time in the case of online learning tasks.
3. Are difficult to model as they have long tailed distributions including rare minority classes for example or other non-standard distributions.

With this new focus comes the obvious question of how to evaluate whether an existing trained deep learning model is representing the current data distribution well. Several works have looked at related problems, including:

- transferability, being how well a model trained on a related source task is likely to perform when fine-tuned on a target task [1, 2, 3, 4]
- analysis of deep learning feature spaces and how those produced by pretrained models differ from those with random initialisation [5, 6, 7].

As far as we are aware there is no research that has looked at evaluating specifically how well a particular deep learning model model represents a data distribution for a particular task.

A successful measure of this kind will have important implications for many challenges in modeling real world data that is continuously changing, has too few training examples to learn from random initialisation, or has a non-standard distribution. This measure could be used to reduce the frequency of computationally expensive retraining of models to incorporate new data, flag predictions that cannot be relied upon because of biases in the model training, and provide guidance on the best hyperparameters and other settings required to achieve optimal performance on new data while not decreasing the performance on old data.

We make the following contributions:

1. New evaluation measures for determining how well the current feature space representation models the current data distribution and task.

EBeM'22: Workshop on AI Evaluation Beyond Metrics, July 25, 2022, Vienna, Austria
guyver.fu@anu.edu.au (Y. S. Fu); wenbo.ge@anu.edu.au (W. Ge); j.plested@unsw.edu.au (J. Plested)
0000-0002-0429-3061 (W. Ge); 0000-0001-9342-4539 (J. Plested)

2. A thorough comparison of our new measures and potential candidates from other related research areas.

2. Related Work

There have been limited previous investigations into measures of how well the data is being represented by a deep learning model. There are however many potential methods that could be adapted for this purpose from other fields including:

1. Recent methods designed for measuring the "transferability" of a pretrained deep learning model [2, 1, 3]. The logic for this being that how well modelled a source dataset is would likely be strongly correlated with how transferable the current model is. If the pretrained model weights produce a poorly modelled feature space transfer learning is likely to perform poorly with those weights.
2. Methods designed for measuring how well clustered a high dimensional space is. The logic here is that a well modeled feature space for classification is one where the data points are well clustered and separated into their classes in feature space ready to be classified by the final classification algorithm. There are many clustering measures that fail in high dimensional spaces or with high number of classes which mean that they are not useful for many deep learning feature spaces. However, there are several that do work well in these spaces [8].
3. Adapting methods designed to measure distance in high dimensions. A major problem with measuring the feature space is the high dimensionality. We propose a new method of measuring clustering based on the Fisher Score [9] that is commonly used as a clustering measure in two dimensions. We replace the Euclidean distance measure in the Fisher Score with cosine similarity, which is known to be an effective distance measure in high dimensions, along with other adaptations.

Several research areas that are related to measuring the feature space are outlined below.

2.1. Exploring the feature space in deep transfer learning

Several methods have been proposed for analysing the feature space from a pretrained model applied to a new target dataset in transfer learning [5, 6]. However these methods are focused on analysing how pretrained weights stabilise and improve training on the target dataset, and prevent over fitting. They do not look at how well fixed weights represent the current target dataset without fine-tuning and thus when and how to perform fine-tuning.

2.2. Exploring and visualising the deep learning feature space

There are many methods that work on visualising either:

- the feature activations within a deep neural network [10, 11]
- the final feature space [12, 13, 14, 15]
- the predictions and their accuracy [16, 17].

While some of these methods, particularly those in item two above, do result in a projection of the feature space into a lower dimensional visualisation that would be easier to measure, they focus on visual inspection rather than on measurement. They also don't analyse the loss of information, and thus intra-class separation, by projecting from a high dimensional space to a low dimensional space that can be visualised.

2.3. Interpreting Model Predictions

There has been a large amount of work done in interpreting model predictions and producing measures and visualisations that show how much a prediction should be trusted [18, 17]. These models focus on analysing and interpreting the importance of input features, rather than the final learned feature space.

2.4. Metric Learning

Metric learning techniques aim to find a feature embedding space that optimises some predefined distance metric given pairs of examples that are classified as either the same or different [19, 20, 21]. This problem has been well studied. Our problem is the opposite in that we already have an embedding space and we wish to find a metric that measures how well our current embedding is separating our current samples into the same and different classes or clusters. There may be some potential to repurpose scores designed for the metric learning space, however we leave this to future work as we have focused on the most promising closely related measures in this work.

3. Methodology

4. Notation

- $x \in \mathcal{X}$ where x is an input and $\mathcal{X}$ is the domain
- $X = \{x_1, x_2, ..., x_n\}$ where X is the set of inputs
- $\mathcal{Y}$ is the finite set of labels
- $C_k = \{x_{k,1}, x_{k,2}, ..., x_{k,m_k}\}$ is the set of inputs that belong to class k with m_k samples, and thus $C_1 \cup C_2 \cup ... \cup C_K = X$ where K is the number of classes
- θ is the trained model, which can be decomposed as $\theta(x) = h(w(x))$
- w is the feature extractor that maps an input x to a representation (or embedding) $r = w(x)$
- r is the feature representation
- h is a classifier (or head) that takes the representation r as input and returns a probability distribution over $\mathcal{Y}$.
- $\mathcal{R}_i = \{r_{i,1}, r_{i,2}, ..., r_{i,m}\} = w(C_i)$ is the feature representation of the inputs in a class, processed by the feature extractor w
- We define $P(A, B)$ as a function that operates on two sets, A and B, and gives the unordered set of all unique pairs from A and B. That is,

$$P(C_1, C_2) = \{(x_{1,1}, x_{2,1}), (x_{1,1}, x_{2,2}), ... \\ , (x_{1,m_k}, x_{2,m_k-1}), (x_{1,m_k}, x_{2,m_k})\}$$

- We can also say that, when $A = B$, $P(A, A) = P(A)$ and instead gives the unordered set of unique pairs, excluding pairs with itself. That is,

$$P(C_k, C_k) = P(C_k) = \{(x_{k,1}, x_{k,2}), (x_{k,1}, x_{k,3}), ... \\ , (x_{k,m_k-1}, x_{k,m_k})\}$$

4.1. Scoring the feature space

The aim of this work was to quantify how well constructed a feature space is by creating or finding a measure that gives high scores when the feature space is well formed and low scores when the feature space is malformed. Here, we think of a well formed feature space as one where there is high similarity/tight clustering within a class (intra-class) and low similarity/sparse clustering between classes (inter-class). Figure 1 shows a well formed 1,500 dimensional feature space reduced using T-SNE into the normalised top-2 representative dimensions so it can be visualised. Note that the data points from all classes are grouped tightly within their class and mostly well separated from other classes.

The motivation for a score that measures how well constructed the feature space is, is three-fold:

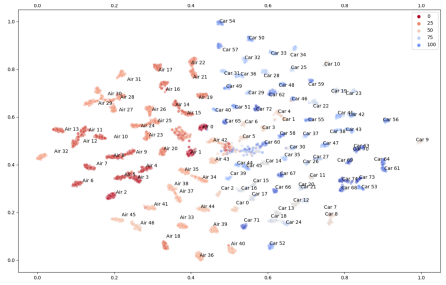

Figure 1: A well formed feature space. Each class is numbered and allocated a different colour. Class centroids are labelled.

1. It would allow for the detection of when re-training is needed. When the input data distribution slowly changes, re-training or updating of the model is required. Human monitoring of model inputs to detect domain shift would be time consuming, require expert knowledge, and may not be possible for a human to judge at all in the case of highly complex high dimensional input spaces. Furthermore, it is not computationally efficient to constantly update the model with every new sample or even at frequent regular intervals. As a result, it is useful to quantify domain shifts in order to batch update the model.
2. It may quantify how to perform transfer learning. Using a trained source model with a target dataset, a well formed feature space would likely mean that the model is easier to be re-trained for the target task and needs lower learning rate, momentum and higher weight regularisation. Whereas a malformed feature space will likely mean that effective transfer learning will be harder to achieve and require more precise attention to finding suitable hyperparameters [22, 23].
3. It would also allow for detection of when a model should not be relied upon to predict particular data points that are too far removed from the standard data distribution the model has been changed on. For example facial recognition models performing poorly on minority races [24].

4.2. Proposed scores

See Section 4 for a list of mathematical notation used in this report.

4.2.1. Proposed score

We propose several scores that use cosine similarity to quantify the level of inter-class similarity vs intra-class similarity. We expect that a well formed feature space as

shown in Figure 1 should have high intra-class similarity and low inter-class similarity.

Our measure is based on adapting the Fisher Score [9] which is known to perform poorly in high dimensions, by replacing the Euclidian distance with cosine similarity which is known to perform well in high dimensions.

Cosine similarity is defined as:

$$S_c(a,b) = \cos\angle(a,b) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{a^\intercal b}{\sqrt{a^\intercal a}\sqrt{b^\intercal b}} \quad (1)$$

where a and b are vectors, $\cdot$ is the inner dot product, and $\|\cdot\|$ is the magnitude of the vectors.

We define our first FERM:

$$\text{FERM 1} = \frac{1}{K}\sum_{k=1}\frac{\frac{2}{m_k^2-m_k}\sum_{r_i,r_j\in P(C_k)} S_c(r_i,r_j)}{\frac{1}{m_k(n-m_k)}\sum_{r_i,r_j\in P(C_k,X\setminus C_k)} S_c(r_i,r_j)} \quad (2)$$

The intuition is quite simple: the numerator is the sum of cosine similarities of all unique pairs in a class, normalised by the number of unique pairs (i.e., an average). The denominator is the sum of cosine similarities of all unique pairs between samples in the class and samples out of the class, normalised by the number of unique pairs (i.e., an average). This provides a ratio of intra-class similarity and inter-class similarity. This ratio is then averaged across all classes, resulting in FERM 1.

We can then define our second FERM:

$$\text{FERM 2} = \frac{\sum_{k=1}\frac{2}{m_k^2-m_k}\sum_{r_i,r_j\in P(C_k)} S_c(r_i,r_j)}{\sum_{k,l\ k<l}\frac{1}{m_k m_l}\sum_{r_i,r_j\in P(C_k,C_l)} S_c(r_i,r_j)} \quad (3)$$

The intuition is similar to the first FERM. The numerator remains the same after incorporating the out sum (an average of cosine similarities of all unique pairs in a class, across all classes), but the denominator is now an average of cosine similarities of unique pairs between samples in the class and samples out of the class *that has not yet been accounted for*. Although, in the first measure, only the unique pairs of samples in and out of a class are averaged, further repeating this (the outer sum) results in double counting across classes. FERM 2 prevents this double counting.

We define our third FERM through the use of a centroid in terms of cosine similarities, a so called 'angular centroid'. In the same way that the average Euclidean distance of one point to several other points can be represented as the distance of that one point to a Euclidean centroid of points, the average angle between one point and several other points can be represented as the angle between that one point and an 'angular centroid' of points. The centroid for a class k is defined as:

$$G_k = \frac{1}{m_k}\sum_{r_i\in C_k}\frac{r_i}{\|r_i\|} \quad (4)$$

This can be interpreted as normalising all samples to the unit hyper-sphere, then finding the centroid point on the unit hyper-sphere by adding all normalised samples together and normalising the combined vector. We can then define our third FERM:

$$\text{FERM 3} = \frac{\sum_{k=1}\frac{2}{m_k^2-m_k}\sum_{r_i,r_j\in P(C_k)} S_c(r_i,r_j)}{\sum_{k,l\ k\neq l}\frac{1}{m_k(K-1)}\sum_{r_i\in C_k} S_c(r_i,G_l)} \quad (5)$$

The numerator term is still the same, but now the denominator is the average of cosine similarity of samples within a class to the centroids of other classes.

Using the same notation above, we can then define our fourth FERM:

$$\text{FERM 4} = \frac{\sum_{k=1}\frac{2}{m_k^2-m_k}\sum_{r_i,r_j\in P(C_k)} S_c(r_i,r_j)}{\sum_{k,l\ k\neq l}\frac{1}{K-1} S_c(G_k,G_l)} \quad (6)$$

This further simplifies the calculation of the denominator to a comparison of the centroid of a class to the centroids of other classes.

For all FERMs a higher score means better clustering. As each individual FERM score and thus the numerator and denominator are within the bounds $[-1,1]$, a positive score above 1.0 reflects more intra-class similarity compared to inter-class similarity.

4.3. Data sets

We have selected the following datasets.

4.3.1. Source Dataset

ImageNet 1K (ImageNet) [25] A general image dataset containing 1,000 common image classes with at least 1,000 total images in each class for a total of just over 1.3 million images in the training set. We use ImageNet as the source dataset for all our experiments.

4.3.2. Target Datasets

Caltech-256 (Caltech) [26] Pictures of objects belonging to 256 categories, with at least 80 images per category. The Caltech dataset is a general image classification dataset similar to ImageNet but with orders of magnitude fewer training examples. It is generally considered to be the most similar target dataset to ImageNet and fixed weights pretrained on ImageNet tend to perform about as well as fine-tuned weights [22, 23].

FGVC Aircraft (Aircraft) [27] Contains 100 different makes and models of aircraft with 6,667 training examples and 3,333 test examples. The Aircraft dataset is a fine-grained image classification dataset that is very different

from ImageNet. Fixed weights pretrained on ImageNet perform extremely poorly on this dataset [22, 23].

Stanford Cars (Cars) [28] Contains 196 different makes and models of cars with 8,144 training examples and 8,041 test examples. The Cars dataset is also a fine-grained image classification dataset that is very different from ImageNet and fixed weights pretrained on ImageNet also perform extremely poorly on this dataset [22, 23].

Describable textures (DTD) [29] Consists of 3,760 training examples of texture images jointly annotated with 47 attributes. While the DTD dataset is conceptually very different to ImageNet recent results have shown that fixed weights pretrained on ImageNet perform reasonably well on this dataset compared to fine-tuned weights [22, 23].

The ratio of the fixed features to fine-tuned results for a model pretrained on ImageNet are shown in Table 4 for all datasets.

5. Experiments

We performed two sets of experiments:

1. Conducting experiments to compare the effectiveness of our score along with candidate scores from other fields in measuring how well a model trained on the ImageNet 1K source dataset represents a particular known and stable target dataset. We use datasets where it is well known how well fixed pretrained ImageNet 1K weights perform on them so they make a good basis for comparison.
2. Using the above measures to detect 'corruption' or domain shift in the feature space.

We further elaborate on each goal in the corresponding Sections 5.1 and 5.2 below.

In addition to our proposed measures, we explored several other clustering measures. These were chosen by reviewing [8] and removing clustering scores that were not stable as dimensionality increased (large perturbations or outliers), and similar in score between overlapping-clusters and well separated-clusters:

- Silhouette score [30]
- Davies Bouldin score [31]
- Calinski Harabasz score [32]
- Dunn score [33]
- RS index [8]
- Point Biserial Index [34]
- $C\sqrt{K}$ index [35].

We also investigated recent transferability scores that have been shown to perform well when measuring how well transfer learning will perform on a particular target dataset:

- H-score [2]
- LEEP [1]
- OTCE [3].

5.1. Stable target datasets

For each experiment we used the Inception v4 architecture [36] pretrained on ImageNet 1k. Using this model, we compared the different FERMs on the different target data sets: Aircraft, DTD, Cars, and Caltech-256. We also used ImageNet 1k as a target data set to determine a baseline score for each measure.

During this evaluation, two pipelines were constructed: one that utilises transformations of the data, and one that does not. When determining how well classes are clustered together, a forward pass of the unaltered data was initially used, providing us with the exact feature representation of that sample. During a standard deep learning training process, samples are randomly flipped, scaled, resized, and rotated. These samples incur a loss if classified incorrectly, and so we expect the model to still learn to classify those samples correctly. Therefore it is likely that the feature representation of these randomly transformed samples are still able to be represented in a well formed feature space. Assuming the model adequately classifies the transformed data, a measure that is robust to these transformations (that is, does not change much in the presence or absence of transformations) would be better than one that is not, as it would allow us to use this during the training process.

We explored the four proposed FERMs on the five target data sets (including ImageNet 1k) with the two different pipelines (with or without transformations). Each transformation experiment was also repeated five times, as the transformations are random.

5.1.1. Results

Comparisons between different FERMS across the different target data sets with and without transformations can be seen in Table 1. Note that results with transformations are reported as means and standard deviations as the experiments were repeated. The datasets in all tables are listed in order of the ratio of the performance of fixed features pretrained on ImageNet to the best fine-tuned model performance, using results from [22, 23] as a proxy for how well formed the feature space is.

With and without transformations ImageNet consistently scored highest, followed consistently by Caltech except with FERM 4. For FERM 1 and 2 Aircraft and Cars score much lower than ImageNet and Caltech and DTD is in between. This is the same ordering as our proxy for a well formed feature space.

Table 1
Different FERM measures with and without transformations on different target tasks. Source task is ImageNet, source model is trained Inceptionv4. Higher measure is better. FF/FT is the ratio of fixed feature results to fine-tuned results for an Inceptionv4 model pretrained on ImageNet. This is used as our proxy for how well formed the current feature embedding is. Standard deviation in brackets. Standard deviations of 0.00 were hidden for brevity.

Target task	FF/FT	FERM 1		FERM 2		FERM 3		FERM 4	
Transformations		False	True	False	True	False	True	False	True
Aircraft	0.633	1.11	1.08	1.14	1.11	1.04	0.97	0.97	0.88
Cars	0.677	1.22	1.16	1.22	1.16	1.06	0.98 (0.01)	0.93	0.83 (0.01)
DTD	0.946	1.33	1.31	1.36	1.34	0.91	0.89	0.63	0.61
Caltech-256	0.987	1.61	1.47	1.61	1.47	1.27	1.10	1.02	0.84
ImageNet 1k	1.0	1.82	1.82	1.81	1.81	1.47	1.47	1.21	1.21

Table 2
Different transferability measures without transformations on different target tasks. Source task is ImageNet 1k, source model is trained Inception v4. Higher measure is better.

Target task	LEEP	OTCE	H-score
Aircraft	-4.59	0.26	53.09
Cars	-5.25	0.32	160.41
DTD	-3.83	0.28	41.77
Caltech-256	-5.49	0.31	152.64
ImageNet 1k	-6.84	0.00	360.00

Table 3
Different transferability measures with transformations on different target tasks. Source task is ImageNet 1k, source model is trained Inception v4. Standard deviation in brackets. Standard deviations of 0.00 were hidden for brevity.

Target task	LEEP	OTCE	H-score
Aircraft	-4.59	0.28 (0.01)	49.76 (0.13)
Cars	-5.26	0.34	158.95 (0.12)
DTD	-3.83	0.29	41.42 (0.05)
Caltech-256	-5.51	0.34	131.00 (0.37)
ImageNet 1k	-6.84	0.00	360.00 (0.00)

Further results showing the comparison with all additional clustering measures and transferability scores can be seen in Tables 2 to 5.

5.1.2. Discussion

We know that fixed features pretrained on ImageNet 1k perform well on Caltech-256, moderately well on DTD, and poorly on Aircraft, and Cars [22, 23] as shown by our ratios of fixed features to fine-tuned performance in Table 1. We use this as a proxy for a well formed feature space and expect a good score to reflect the same knowledge, that is, a low score for Aircraft and Cars, a moderate score for DTD, a high score for Caltech-256, and a very high score for ImageNet.

Our results with and without random transformations to the data suggest that FERM 1, and 2 seem to be able to consistently do this. It seems that FERM 1, and 2 have potential as a way to measure how well formed the feature space is for a particular trained model and target task.

Of the transferability measures, LEEP is the only score that consistently ranks ImageNet 1k and Caltech-256 as most transferable, in the presence and absence of transformations, however it ranks DTD as least transferable in both cases, which is incorrect. Given the scores are in the same order as the number of classes in the dataset it seems likely that it's affected by the number of classes. H-score also seems to be strongly affected by the number of classes, as the scores are close to being proportional to the number of classes in the target dataset.

Of the clustering measures, Silhouette score, Davies Bouldin score, Point Biserial Index, and $C\sqrt{K}$ index seem to also consistently rank ImageNet 1k and Caltech-256 as the most transferable, in the presence and absence of transformations. However only Silhouette score ranks DTD as moderately transferable compared to the others. Point biserial may also be strongly affected by the number of classes, as the scores are again close to being proportional to the number of classes in the target dataset.

In summary when looking at only stable target datasets our proposed scores FERM 1 and 2 as well as the clustering measure Silhouette score are good candidates for measuring how well formed the feature space is for a given trained model and target task.

5.2. Detecting and quantifying domain shifts

We attempted to detect and quantify incremental domain shifts. As it is hard to concretely quantify different levels of domain shift, we reduce the problem down into detecting levels of 'corruption'. 'Corruption' is defined as the presence of the target data set mixed into the source data set, where the source data can be thought of as no domain shift, whilst the target data set can be thought of

Table 4
Different clustering measures without transformations on different target tasks. Source task is ImageNet 1k, source model is trained Inception v4

Target task	Silhouette	Davies Bouldin	Calinski Harabasz	Dunn	RS	Point biserial	$C\sqrt{K}$
Aircraft	-0.12	6.79	20.71	0.08	1.61	243.95	0.10
Cars	-0.10	4.29	5.75	0.21	1.61	1146.63	0.07
DTD	-0.04	4.65	9.07	0.09	1.81	248.99	0.14
Caltech-256	0.09	3.10	18.22	0.12	1.61	2334.25	0.06
ImageNet 1k	0.11	2.88	40.68	0.00	1.55	9823.30	0.03

Table 5
Different clustering measures with transformations on different target tasks. Source task is ImageNet 1k, source model is trained Inception v4. Standard deviation in brackets. Standard deviations of 0.00 were hidden for brevity.

Target task	Silhouette	Davies Bouldin	Calinski Harabasz	Dunn	RS	Point biserial	$C\sqrt{K}$
Aircraft	-0.13 (0.01)	7.79 (0.06)	10.86 (0.17)	0.06 (0.01)	1.75	280.96 (1.35)	0.10
Cars	-0.10	4.52 (0.03)	3.87 (0.11)	0.20 (0.02)	1.70 (0.01)	1218.97 (14.44)	0.07
DTD	-0.04	4.85 (0.02)	8.48 (0.10)	0.09 (0.02)	1.82	250.41 (2.05)	0.14
Caltech-256	0.03	3.59 (0.01)	12.65 (0.12)	0.00	1.70	2267.21 (18.45)	0.06
ImageNet 1k	0.11	2.88	40.68	0.00	1.55	9823.30	0.03

as complete domain shift. This can be then quantified by the percentage of target data in the source data set.

We again started with an Inception v4 model pretrained on ImageNet 1K. We then incrementally shifted the domain by either adding target data to the evaluation set or removing source data from the evaluation set. The source samples are derived from the ImageNet 1k validation set, whilst the target samples are derived from the training set of Aircraft. The Aircraft dataset was used in this case as it was the most poorly represented by the pretrained model in our previous experiments. Each time we added more 'corruption' we used all measures from our previous experiments to measure the feature space.

Specifically, we created the evaluation set by randomly choosing 200 classes from ImageNet 1k, and then randomly choosing the same number of samples across the classes. Aircraft was combined with this in a similar way, that is, randomly choosing the same number of samples across all 100 classes. The union of both creates the evaluation set.

The feature representation of a sample is defined as $r_i = w(x_i)$, where $w(\cdot)$ is the feature extractor from the trained source model. We expected that as the level of corruption increases (as more of the source data set is replaced by the target data set), the clustering of classes in the feature space degrades; features in the new class are not clustered well, and thus the overall clustering score should decrease.

Another way we approached the problem is by looking at transferability measures. Since measures of transferability are largest when the source task is the same as the target task, we hypothesized that at 0% corruption (i.e., there is no domain shift) transferability scores will be high, and will slowly degrade with increasing levels of corruption.

5.2.1. Results

For each different combination of source and target dataset we ran the experiment 10 times as the selection of the examples for each class was random. The classes chosen from ImageNet were fixed to allow for a fixed comparison. The change in each of the different scores as the domain shifts to the target data set of Aircraft can be seen in Figure 2. The scores have been normalised between 0 and 1. Although several of these were repeated and averaged, we did not plot the error bars as they are largely uninformative, as seen in Section 5.1.1.

5.2.2. Discussion

We expect a measure that is good at detecting domain shift to start with a normalised score of 1 (or 0 if inversely proportional) with no domain shift, and incrementally decrease to 0 (or increase to 1) as the domain is completely shifted. We also would like the measure to be monotonically decreasing (or increasing). The results in

Figure 2: Scores as domain shifts from ImageNet 1k to Aircraft. Only 200 randomly selected classes of ImageNet 1k were used. Percentages are aircraft as a portion of the whole dataset

Figure 2 show that only Point Biserial Index seems to be almost entirely monotonically trending. Ignoring the last point (0 samples of ImageNet 1k), H-score seems to have strong potential to detect domain shift however more investigation is required to see why the final point is so far out of sequence.

RS index, Davies Bouldin score, and Silhouette score seem to also have sections of monotonic trend. Further work is required to make a strong claim in the ability of these measures to detect and quantify domain shift.

The results of our FERMs are particularly interesting. If the points where there is only one example per class of either Aircraft or ImageNet are excluded (second from the left and right on the graph) the trend is almost monotonic from all ImageNet examples to all Aircraft examples. Also the point where the score reduces significantly from the original ImageNet score is approximately at the point where the dataset has shifted to the extent that its composition is more than 50% of the target dataset. The experiments with only one example from each class of either the source or the target dataset can be thought of as just adding noise, as intra-class distances cannot be measured with only one example for each class. Thought of in this way it is useful that our measure is strongly sensitive to this situation.

More extensive work should be done to compare our methods with the Point Biserial Index, and H-score across a broader range of domain shift applications.

6. Conclusion

We have created a selection of new scores for evaluating how well a particular dataset is being represented by the current model weights and architecture. We have performed extensive experiments to compare our new scores with measures from other fields that could have potential to be reused for this purpose. We compared the efficacy of these measures on both measuring how well existing model weights are representing a new stable target dataset, and detecting domain shift. The result of these experiments indicate that this new method, along with two others, have excellent potential to be used for measuring how well a dataset is currently being represented by a model.

Measures for this purpose have not been investigated before and our results have strong implications for the wider deep learning community. These measures have the potential to be used to:

1. Detect domain shift and predict the best response in terms of model retraining.
2. Detect when an existing model has biases that make it unreliable for use on rarer data.
3. Predict the optimal way to train or retrain a model with limited training examples for a new or changing target dataset.

There are a great many examples of ways these measures could be useful as an important part of an overall evaluation of a model, some of these are:

- Uncovering and quantifying biases in models. For example how well is a model that is trained on mostly Caucasian faces likely to perform in identifying faces from other races.
- Quantifying how well prediction models based on historical data are representing data from the last few years that has changed due to COVID and other modern challenges. Once quantified these measures could also give guidance on how to update models to better incorporate modern data.
- Highlighting when models are performing well on training and test data, but overfitting a poor representation that will not generalise well to new data. A classic example being the snow in the foreground being used to classify a husky versus a wolf in [17].

Acknowledgments

Thanks to Dawn Olley for editing services.

References

[1] C. Nguyen, T. Hassner, M. Seeger, C. Archambeau, Leep: A new measure to evaluate transferability of learned representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 7294–7305.

[2] Y. Bao, Y. Li, S.-L. Huang, L. Zhang, L. Zheng, A. Zamir, L. Guibas, An information-theoretic approach to transferability in task transfer learning, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 2309–2313.

[3] Y. Tan, Y. Li, S.-L. Huang, Otce: A transferability metric for cross-domain cross-task representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 15779–15788.

[4] A. T. Tran, C. V. Nguyen, T. Hassner, Transferability and hardness of supervised classification tasks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1395–1405.

[5] B. Neyshabur, H. Sedghi, C. Zhang, What is being transferred in transfer learning?, arXiv preprint arXiv:2008.11687 (2020).

[6] H. Liu, M. Long, J. Wang, M. I. Jordan, Towards understanding the transferability of deep representations, arXiv preprint arXiv:1909.12031 (2019).

[7] X. Shen, J. Plested, S. Caldwell, T. Gedeon, Exploring biases and prejudice of facial synthesis via semantic latent space, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.

[8] T. Nenad, M. Radovanovic, Clustering Evaluation in High-Dimensional Data, in: Unsupervised Learning Algorithms, Springer, 2016, pp. 71 – 107.

[9] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (1936) 179–188.

[10] A. Nguyen, J. Yosinski, J. Clune, Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, arXiv preprint arXiv:1602.03616 (2016).

[11] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, arXiv preprint arXiv:1506.06579 (2015).

[12] M. Aubry, B. C. Russell, Understanding deep features with computer-generated imagery, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2875–2883.

[13] T.-Y. Lin, S. Maji, Visualizing and understanding deep texture representations, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2791–2799.

[14] C. Vondrick, A. Khosla, H. Pirsiavash, T. Malisiewicz, A. Torralba, Visualizing object detection

features, International Journal of Computer Vision 119 (2016) 145–158.
[15] G. E. Hinton, S. Roweis, Stochastic neighbor embedding, Advances in neural information processing systems 15 (2002).
[16] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, J. Wilson, The what-if tool: Interactive probing of machine learning models, IEEE transactions on visualization and computer graphics 26 (2019) 56–65.
[17] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[18] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).
[19] K. Q. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, Advances in neural information processing systems 18 (2005).
[20] E. Xing, M. Jordan, S. J. Russell, A. Ng, Distance metric learning with application to clustering with side-information, Advances in neural information processing systems 15 (2002).
[21] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking., Journal of Machine Learning Research 11 (2010).
[22] S. Kornblith, J. Shlens, Q. V. Le, Do better imagenet models transfer better?, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2661–2671.
[23] J. Plested, X. Shen, T. Gedeon, Non-binary deep transfer learning for imageclassification, arXiv e-prints (2021) arXiv:2107.08585. arXiv:2107.08585.
[24] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on fairness, accountability and transparency, PMLR, 2018, pp. 77–91.
[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.
[26] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, authors.library.caltech.edu (2007).
[27] Y. Cui, F. Zhou, Y. Lin, S. Belongie, Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1153–1162.
[28] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, A. Vedaldi, Fine-Grained Visual Classification of Aircraft, Technical Report, Toyota Technological Institute at Chicago, 2013. arXiv:1306.5151.
[29] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
[30] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.
[31] D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE transactions on pattern analysis and machine intelligence (1979) 224–227.
[32] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics-theory and Methods 3 (1974) 1–27.
[33] J. C. Dunn, Well-separated clusters and optimal fuzzy partitions, Journal of cybernetics 4 (1974) 95–104.
[34] G. W. Milligan, A monte carlo study of thirty internal criterion measures for cluster analysis, Psychometrika 46 (1981) 187–199.
[35] D. Ratkowsky, A stopping rule and clustering method of wide applicability, Botanical gazette 145 (1984) 518–523.
[36] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.