Unsupervised Data Selection for Supervised Learning

Gabriele Valvano, Andrea Leo, Daniele Della Latta, Nicola Martini, Gianmarco Santini, Dante Chiappino and Emiliano Ricciardi

1IMT School for Advanced Studies Lucca, Lucca, Italy
2Department of Information Engineering, University of Pisa, Pisa, Italy
3Imaging department, Fondazione Gabriele Monasterio, Massa, Italy

Abstract

Recent research put a big effort in the development of deep learning architectures and optimizers obtaining impressive results in areas ranging from vision to language processing. However little attention has been addressed to the need of a methodological process of data collection. In this work we show that high quality data for supervised learning can be selected in an unsupervised manner and that by doing so one can obtain models capable to generalize better than in the case of random training set construction.

1 Introduction

In the last decade Deep Learning methods achieved state of the art performance in many computer vision tasks, and became popular among artificial intelligence studies. These methods are Representation Learning methods able to gradually decompose the inputs at multiple levels of representation by using a series of non-linear modules. During this process, the training data are non-linearly projected onto some typically low-dimensional surface approximating an ideal manifold representing any possible input for the specific task.

Supervised learning algorithms, where the learner gradually discovers a mapping function leading from the input samples to the associated outputs, proved to be particularly successful. It is commonly accepted that the success of supervised learning generally depends on the quantity of data available. Furthermore, it is common opinion that the quality of data also plays a role in the final performance of the model.

Unfortunately, despite a big quantity of unlabeled samples are often available, there is no rigorous way to determine in advance which are most informative samples that are worth to label. In addition, the number of labeled data is often limited, as the data labelling process is always associated to a cost in terms of money, time or human effort.

The aim of this work is to introduce a method for the collection of labeled samples based on quality rather than quantity. More specifically, we will show that high quality data for supervised learning can be selected in an unsupervised manner; in this way optimal performances can be obtained also from small training sets.

In order to improve the reproducibility of the work, we also share the Python code used for our experiments.

*Corresponding author: gabriele.valvano@imtlucca.it (G. Valvano).
Figure 1: Example of a circumscribed region of a smooth manifold which can be locally approximated by a planar surface. The learning algorithm learns to map here the training samples (black points). The blue circles symbolically represent the regions which the learner can physically explore thanks to data augmentation operations. Supposing that we can afford to label new samples after shaping the manifold on a large amount of data, the green points should be a better choice than the red ones because they would allow to examine larger unexplored regions of the manifold. Best viewed in color.

2 On the quality of data

The samples associated with clean, well defined labels are commonly referred as good quality samples. For example, if we want to train a neural network to segment a specific organ in a medical image, good labeled examples might consist of pairs of patient images and annotated manual segmentations. Because the process of manual segmentation may take a long time, it is common to compensate for the lack of labeled data by altering the collected ones to obtain new artificial samples. This process is called data augmentation and is performed by applying small random transformations to the data (i.e. rotation, mirroring, translation). These operations allow to immediately generate new plausible input-output pairs, similar to those of the original dataset but with small perturbations useful to improve the robustness of the model.

A big advantage of data augmentation is that it preserves the quality of the training data because we know a-priori the outputs to assign to the generated inputs (e.g., in a segmentation task, the same transformation is applied to both the input image and the output mask).

The generated samples allow the learner to shape new regions of the manifold which represents the training data, but have some limitations: they intrinsically present redundancy because they only show minor deviations from the factors of variation present in the data already collected. In other words, relying on very mild assumptions on the manifold (see Section 3), we can say that the artificial data will be mapped in a neighborhood of the real data representation on the manifold (blue circles in Figure 1). Consequently, if we consider each training sample and its versions in the augmented data as possible realizations of the same random variable, we can generalize associating to each input an Area of Representation (AoR) rather than a simple vector of coordinates on the manifold.

If we model the problem of collecting training data in terms of AoR, the random gathering of samples may result in the unlucky situation in which the AoRs of the inputs overlap or appear to be close on the manifold. Such a situation is awkward to handle because the generalization capability of the learning algorithms are mostly achieved by local interpolations among the regions of the manifold shaped during training. Then we believe that to obtain a better approximation of the ideal manifold it may be much more convenient the collection of labeled data represented in coordinates of the manifold spread enough: this will likely introduce relevant information about possible variations of the inputs.

For this reason, in the first place we wish to better specify the concept of quality of data: we refer to high quality samples as those samples which make it easier for the model to extract useful information to build a manifold as close as possible to the ideal one (i.e. able to represent any possible real input for that specific task). With the given architecture, these samples should let us train the best possible
predictor in terms of generalization capabilities on unseen data. We expect these samples to be able to show a statistically greater number of factors of variation in the data with respect to the case of random sampling in the input domain.

In particular, we assume that learning to represent the inputs on a manifold encouraging at first the smoothness of this manifold, and then exploiting it, can lead to the detection of the most important samples among the available unlabeled data. This is possible if this abstract representation preserves most of the information useful to fully characterize the input data. In such a case, our purpose is to approximate the manifold representing the input domain with a limited number of samples. We believe that if the manifold is smooth enough, we can discard those samples which appear to be particularly close in the feature space, and chose instead those samples which show very different factors of variations (i.e., the green dots in Figure 1).

3 On the latent representation

Due to data complexity, working directly in the input domain may be particularly difficult. For this reason we usually would like to map real-world data to a manifold \( M \) of much lower dimensionality. We would like this manifold to have several properties, among which [5]:

- **Smoothness**: the learned function \( f \) is s.t. \( x \approx y \) generally implies \( f(x) \approx f(y) \).
- **Sparsity**: for a given observation \( x \), the extracted features should be sparse. This means that only a small fraction of possible factors of variation are relevant, while small variations of \( x \) (i.e. random noise) are not encoded.
- **Natural clustering**: very different samples should tend to be well separated in the features space.

Together with the properties of **invariance** and **distribution** of the representation we would like to **disentangle** the factors of variation to achieve a complete and robust representation of the inputs [5], which can be used as a proxy to evaluate similarities among different samples.

3.1 Sampling data observing a smooth manifold

Autoencoders (AE) [8–11], are algorithms able to learn a direct encoding of the inputs and build a manifold representing them, while learning a decoding function mapping back from representation to input space. A probabilistic approach to the AE framework is that of Variational Autoencoders (VAE) [12]. Differently from the AEs, VAEs represent the encoded features as a probability distribution from which randomly sample to reconstruct the input. By doing so, VAEs enforce a continuous, smooth latent space representation where close features vectors should correspond to similar reconstructions.

In order to build a smooth and continuous manifold, in this work we employed the VAE framework. We then analyzed the latent representation for each volume in the training set, pruning away samples which appeared too close on the manifold. In this context we evaluated the local distances linearly approximating the regions of the manifold and using euclidean distance.

4 Experiments

In order to validate the method, we considered a problem of binary segmentation. In particular, the task was to train a model to detect the pixels belonging to brain tissue in medical images and return binary segmentation masks. For the experiments we used the open source software library TensorFlow; the code is available online[2].

4.1 Data

For our experiments we employed an open source dataset, the Neurofeedback Skull-stripped (NFBS) repository[3] containing a total of 125 Magnetic Resonance Imaging (MRI) volumes of the head and

---

2The code can be found here: [https://gitlab.com/gabriele_valvano/unsup_data_selection](https://gitlab.com/gabriele_valvano/unsup_data_selection).

3The NFBS dataset can be found here: [http://preprocessed-connectomes-project.org/NFB_skullstripped/](http://preprocessed-connectomes-project.org/NFB_skullstripped/).
their associated binary segmentation masks of the brain. These pairs consist of three-dimensional
grayscale arrays of 256x256x192 voxels; however, due to memory constraints on the available GPU
we undersampled the volumes to the size of 128x128x96.

The undersampled data was divided in a test set containing 25 volumes, a validation set of 10 volumes
and a train set of 90 volumes. All the 90 volumes of the training set were employed during the
unsupervised training of a Convolutional VAE (CVAE), but only few selected samples were used
for the supervised training process, mimicking what would happen if we could afford to manually
label only few cases. In our experiment, due to the high redundancy of the dataset (Table 1), a wide
usage of data augmentation makes it possible to train a neural network with a very limited amount of
samples. For this reason, we analyzed the performance of the supervised model training it on just 5%
of the training volumes, comparing the case of random subsampling with the case of data pruning
operated by removing the closest samples in the manifold constructed by the unsupervised CVAE.

To obtain independence from the MRI acquisition parameters, all the inputs were preprocessed
standardizing each voxel $x$ on the entire volume, as:

$$z = \frac{x - m}{s}$$

where $m$ represents the mean value of the MRI scan and $s$ its standard deviation.

Employed data augmentation operations were: random translation, rotation, mirroring, scaling and
shearing, plus an additional injection of random noise on the input images.

4.2 Selecting the Data on the Manifold

Accordingly to what stated before, the data selection procedure was carried out analyzing the features
encoded by the unsupervised model and detecting samples maximizing their reciprocal distance in
the encoded space. The greedy approach describing the data selection procedure is summarized in
the following steps:

- For each sample $j$ in the unlabeled dataset compute the features vector $a_j$.
- Iterate, removing one sample at a time, until reaching the maximum number of training
  samples chosen to train the supervised algorithm.
- At each step:
  - compute the Euclidean distance matrix $E$ containing distances $d_{jk}$ between each pair
    of vectors $(a_j, a_k)$ remaining in the dataset;
  - evaluate the pair of samples $(\tilde{x}, \tilde{y})$ corresponding to the minimum distance in the strictly
    upper triangular part of $E$;
  - between $\tilde{x}$ and $\tilde{y}$ discard the sample globally closer to the others in the feature space.

The global distance of a sample $j$ is evaluated as the mean distance $d_{jk}$ of $a_j$ from the
other feature vectors (i.e. is the mean value of $E[j, :]$).

4.3 Architecture and Metrics for Unsupervised Training

The CVAE used for the unsupervised training procedure (Figure 2) consisted of 12 encoding and
12 decoding layers employing 3D convolutions. In the center of the CVAE two vectors of 128
latent features were introduced to encode the mean and standard deviation vectors characterizing the
gaussian distribution from which to sample during the reconstruction of the input. Batch normalization
[13] was applied after each of the internal layers to accelerate network training [14].

As usual, the CVAE loss function consisted in the sum of two separate contributes: the generative
loss, which is the mean squared error $MSE$ (measuring how accurately the network reconstructs
the inputs), and a latent loss term, which is the KL divergence loss $D_{KL}$ (measuring how closely the
latent variables match a standard normal distribution):

$$Loss = E[MSE + D_{KL}]$$

$$= E \left[ \frac{1}{P} \sum_{p=1}^P |X_p - \hat{X}_p|^2 + \frac{1}{2} \sum_{q=1}^Q \exp(\hat{\sigma}_q(X)) + \hat{\mu}_q^2(X) - 1 - \sigma_q(X) \right]$$
where $E[\cdot]$ is the operator of expectation on the elements in a minibatch, $|X|$ and $|X|$ are respectively the input and its reconstruction, $\mu$ and $\sigma$ the mean and standard deviation vectors extracted by the CVAE.

An additional loss contribute was added to boost the mapping of the latent features on a smooth manifold and to encourage to symmetry (latent reconstruction loss). This additional term is evaluated as mean squared error of the reconstruction of the penultimate encoding layer activations as the output of the second decoding layer and generates a gradient contribute on these internal layers.

The weights, initialized according to [15], were regularized introducing an L2 loss term.

The cost function was optimized with Adam optimizer [16] using a learning rate of 0.01.

4.4 Architecture and Metrics for Supervised Training

For the segmentation task, a 3D extension of the U-net architecture [17] was employed. Even in this case, batch normalization layers [13, 14] were added after each convolutional layer.

Weighted cross-entropy was chosen as cost function, with class weights for each minibatch to compensate for the class imbalance and minimize the classification error:

$$H(y, \hat{y}) = -\sum_l w_l \cdot y_l \cdot \log(\hat{y}_l)$$

$$w_l = 1 - \frac{n_l}{N}; \quad l \in \{0, 1\}$$

where $y_l$ is the ground truth label for the $l$-th class and $\hat{y}_l$ is the network output over the same class; $w_l$ is the weight for the given class $l$, $n_l$ is the number of voxels with labels equal to $l$ and $N$ is the total number of voxels.

Network weights, initialized according to [15], were regularized with an L2 loss term.

In order to train the model we used Adam optimizer [16] with a learning rate of 0.001.

Finally, the segmentation quality was assessed using a popular segmentation metric, namely the Sørensen–Dice coefficient, defined as:

$$DSC = \frac{2|Y \cap \hat{Y}|}{|Y| + |\hat{Y}|}$$

where $|Y|$ and $|\hat{Y}|$ are the cardinalities (i.e. the number of voxels equal to one) of the ground truth segmentation mask and the predicted one, respectively.
Table 1: Performance obtained by the supervised model on the test set when employing just 5% or 50% of randomly selected samples, or all the training volumes.

| Size  | DSC (%) |
|-------|---------|
| 5%    | 95.33   |
| 50%   | 98.38   |
| 100%  | 98.64   |

Figure 3: Loss and DSC index evaluated on the validation set during training.

4.5 Results

Table 1 shows the performance in terms of DSC on the test set, obtained using the whole train set rather than just 5% or 50% of its volumes. The training subsets were obtained via random choice in the pool of available training samples.

Figure 3 illustrates the trend of the loss function (left) and the DSC index (right) measured on the validation set during the training of the supervised learner. Each plot represents the mean value and the standard deviation interval obtained mediating 5 different runs with different random initialization of the network weights. In the case of random sampling of the training volumes at each run there were also different training samples. That, however, was not the case of selection of data according to the approach we suggested: in fact, after the training of the unsupervised CVAE, our process of data pruning is deterministic.

In addition to the performance on the validation set, we also evaluated for each run the mean DSC index on the test set, obtaining a value of 96.08% for the random subsampling and of 96.39% when using our approach.

5 Discussion

The performances obtained by supervised model in Table 1 seem to confirm our hypothesis that an intrinsic redundancy lies in the train set. In fact, despite the use of more data evidently improves the DSC index, very good performance are already obtained with as few as 5% of the train volumes. This justified the subsequent analysis carried out on just 5% of the set.

One could argue that the manifold from which we sampled the training data (built during the unsupervised training) may differ from the one representing the data in the supervised task. Despite of this, our opinion is that since the unsupervised manifold should fully characterize not only the inputs, but also encode many possible variations of theirs, choosing the dissimilar samples on this
manifold should help the supervised learner as well, letting it improve its robustness to the wide variety of data it can encounter.

Experimental results seem to confirm this hypothesis: as can be seen, the model was able to improve its generalization capabilities making use of carefully chosen samples from the available unlabeled population. This suggests that high quality data can be effectively chosen using unsupervised learning and that employing these samples for supervised learning can improve the generalization capabilities of the model.

We see these results as an opportunity to encourage the community to pursue a more systematic investigation of the criterions behind data collection and we believe such an approach to be particularly well suited for tasks where the unlabeled population contains an inherent redundancy. For example, in medical imaging it is often the case that a big number of unlabeled data are available and that the process of data labeling is long and difficult; since there is big similarity among data of the same kind (i.e. same target organ, exam, acquisition protocol, etc.) this method could be successfully applied to get to the optimal compromise between time and performance.

Finally, we find intriguing the idea of studying the unlabeled population on the manifold because it offers a new interesting perspective, that is the possibility to discover outliers in the sample set: this may be of great interest to better understand how the input domain may change and to detect worst case scenarios. In this context, the proposed approach could also be of help for the creation of new high quality test sets to use as benchmark in many challenges on the web.

6 Conclusion

We showed that high quality data for supervised learning can be selected in an unsupervised manner. In particular, by encouraging the manifold representing the unlabeled data to be smooth enough, we were able to discard samples falling particularly close to others in the feature space, assuming them redundant in the content of information for the final task.

We tested our approach on an open source dataset and we showed the improved generalization capabilities of the model compared to the case of random sampling of the training set volumes.

This could suggest that a more systematic investigation of the criteria behind the labeled data collection may be desirable for many tasks in order to achieve more significant impact on real-world applications.

7 Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436, 2015.
[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[3] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.
[4] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484, 2016.
[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.
[6] Pim Moeskops, Max A Viergever, Adriëlle M Mendrik, Linda S de Vries, Manon JNL Benders, and Ivana Išgum. Automatic segmentation of mr brain images with a convolutional neural network. IEEE transactions on medical imaging, 35(5):1252–1261, 2016.

[7] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. Medical image analysis, 35:18–31, 2017.

[8] Y-lan Boureau, Yann L Cun, et al. Sparse feature learning for deep belief networks. In Advances in neural information processing systems, pages 1185–1192, 2008.

[9] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model. In Advances in neural information processing systems, pages 1137–1144, 2007.

[10] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103. ACM, 2008.

[11] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on International Conference on Machine Learning, pages 833–840. Omnipress, 2011.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[14] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? (no, it is not about internal covariate shift). arXiv preprint arXiv:1805.11604, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.