Automated Invariance Testing for Machine Learning Models Using Sparse Linear Layers

Anonymous Authors[1]

Abstract

Machine learning testing and evaluation are largely overlooked by the community. In many cases, the only way to conduct testing is through formula-based scores, e.g., accuracy, f1, etc. However, these simple statistical scores cannot fully represent the performance of ML model. Therefore, new testing frameworks are attracting more attention. In this work, we propose a novel invariance testing approach that does not utilise traditional statistical scores. Instead, we train a series of sparse linear layers which are more easily to be compared due to their sparsity. We then use different divergence functions to numerically compare them and fuse the difference scores into a visual matrix. Additionally, testing using sparse linear layers allows us to conduct a novel testing oracle: associativity: by comparing merged weights and weights obtained by combined augmentation. Finally, we assess whether a model is invariant by checking the visual matrix, the associativity, and its sparse layers. We show that by using our testing framework, inter-rater reliability can be significantly improved.

1. Introduction

Invariance qualities are important properties of a machine learning model in different fields, e.g., face recognition, object detection, segmentation, etc. Different approaches have been used to improve invariance quality, e.g., augmentation (Shorten & Khoshgoftaar, 2019). However, when evaluating how robust a model is against a certain type of transformation $\mathcal{T}$, e.g., rotation/brightness/scaling, in some cases, manual analysis is still required. Such labour-intensive analysis often leads to inconsistent and unreliable assessment.

In order to simplify and automate the invariance testing procedure, formula-based scores are often used to approximate invariance qualities (Goodfellow et al., 2009). For example, accuracy that all transformed input data are predicted correctly (Gao et al., 2020). Machine learning testing, including invariance testing, should be more sophisticated than computing an accuracy score (Zhang et al., 2020). Similar to the Anscombe's quartet shown in Figure 1, formula-based scores are just one statistical measure which does not suffice to approximate the distribution. Therefore, it is desired to have a more sophisticated invariance testing approach.

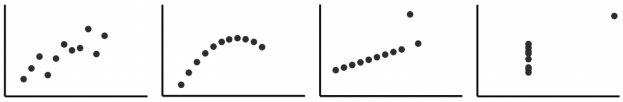

Figure 1. The four sets of points, i.e., the Anscome's quartet, have exactly the same statistical measures, e.g., mean, standard deviation, correlation, etc. However they are differently distributed.

Recently, (Liao et al., 2021) proposed an automatic invariance testing procedure where they used visualisation of all possible statistical scores to assist evaluation of invariance qualities. And they used bagging methods to automate the testing procedure and counteract noisy invariance labels, as suggest by (Frénay et al., 2014). They proposed different "medical-imaging-like" modalities (Kumar et al., 2013) to facilitate testing on different locations. This work, instead, treats each model as a feature extractor, and fully focus on the final feature vectors, and provide a novel testing approach to evaluate invariance qualities with trained sparse linear layers instead of statistical scores.

From transfer learning (Pan & Yang, 2009), metric learning (Kulis et al., 2012), to self-supervised learning (Jing & Tian, 2020), machine learning models are often used as feature extractors. Furthermore, for any machine learning models, it is always desired to investigate how robust/invariant the final feature vectors are (Zhang et al., 2020). Therefore, in this work, we focus on testing invariance qualities for machine-learned feature extractors.

In order to conduct invariance testing for feature extractors, we leverage the elastic net (or sparse linear layers) (Zou

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.

& Hastie, 2005) to better visualise the performance of the feature extractors. The contributions of this work are:

- We propose a novel invariance testing approach for machine-learned feature extractors to replace simple formula-based statistical scores, e.g., accuracy.
- A novel invariance testing approach including a new testing oracle that will lead to more consistent assessment than manual testing via a few random examples.
- We evaluate invariance qualities for a small model repository consisting of 600 CNNs, and provide our assessment labels.
- Our testing framework can offer a set of actionable items which can potentially improve the performance of the feature extractors, e.g., by selecting or combining weights using the sparse linear layers.

2. Related works

Invariance testing broadly falls into robustness testing (Carlini & Wagner, 2017). However, unlike out-of-distribution detection (Yang et al., 2021; Shen et al., 2021) or adversarial attacks (Szegedy et al., 2013; Liao, 2019; Tramèr et al., 2017), our work focuses on evaluating invariance qualities for feature extractors as a whole.

Debugging (Wong et al., 2021) for ML models and other actionable items are also related to this work. After the invariance testing has been conducted, one can design a sequence of debugging processes. These debugging processes can be used for testing/training data cleaning/selection (Metzen et al., 2017), skew detection in training and testing data (Kim et al., 2019), and for neuron rectification(Wong et al., 2021). Specifically, (Wong et al., 2021) included a human-in-the-loop debugging process where CNN models are treated as feature extractors, and sparse linear layers are retrained to select the most "important" features. From those "important" features, people hired from Amazon mTurk (Crowston, 2012) are asked to complete questionnaires to help developers locate "erroneous" features. And these "erroneous" features are manually set to zero to improve models' ability. Similarly, after invariance testing, it would be desirable to have some actionable items, e.g., deactivating "erroneous" features.

(Liao et al., 2021) proposed an automatic invariance testing for machine learning models using visualisation and different statistical functions. They conducted invariance tests on different locations of the models. Our framework instead, focuses on testing feature extractors, and thus focuses on the final features that the models produce. Another related topic to this work is the relationships between activations from different CNNs, e.g., (Morcos et al., 2018; Raghu et al., 2017). However, instead of testing the invariance quality, they tend to eliminate the effect of transformation, so that they are able to find the relationships between activations from different CNNs or different layers without being affected by transformations. This type of measurement can be used for testing equivalence quality (Zhang, 2019) instead of invariance qualities.

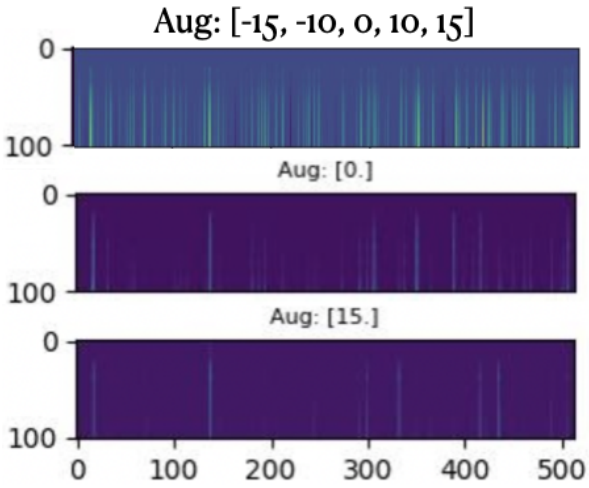

Figure 2. Upper row: sparse linear layers trained using combined augmentation. Middle / lower row: sparse linear layers trained using original training set X, and transformed training set $t_{15^\circ}(X)$ respectively.

Finally, metamorphic testing (Segura et al., 2018) is also related to this topic. For example (Dwarakanath et al., 2018) used images to identify implementation bugs, and (Shen et al., 2018) proposed mutation analysis on different mutation operations, e.g., delete a neuron. In this work, we also propose a new metamorphic testing oracle: associativity.

3. Sparse Linear Layers

Rationale: When testing feature extractor, normally one would train a linear classifier (or a feature selection method), e.g., SVM (Chandra & Bedi, 2021) or multi-layer perceptron (Ruck et al., 1990). However, when conducting invariance evaluation, these traditional methods will have two major issues: (1) training, i.e., gradient descent, for linear layers is not stable (Johnson & Zhang, 2013), e.g., when the loss function is not smooth, resulting in different sets of weights for different runs. (2) the number of selected features are often large, e.g., $\geq$60% in our case. These make comparison between selected features difficult and unreliable. Therefore, we use sparse linear layers to solve the two issues and better visualise differences between selected features.

In order to test whether a feature extractor $F : x \mapsto F(x) \in \mathbb{R}^d$ is invariant, where x is a data object and d is the dimen-

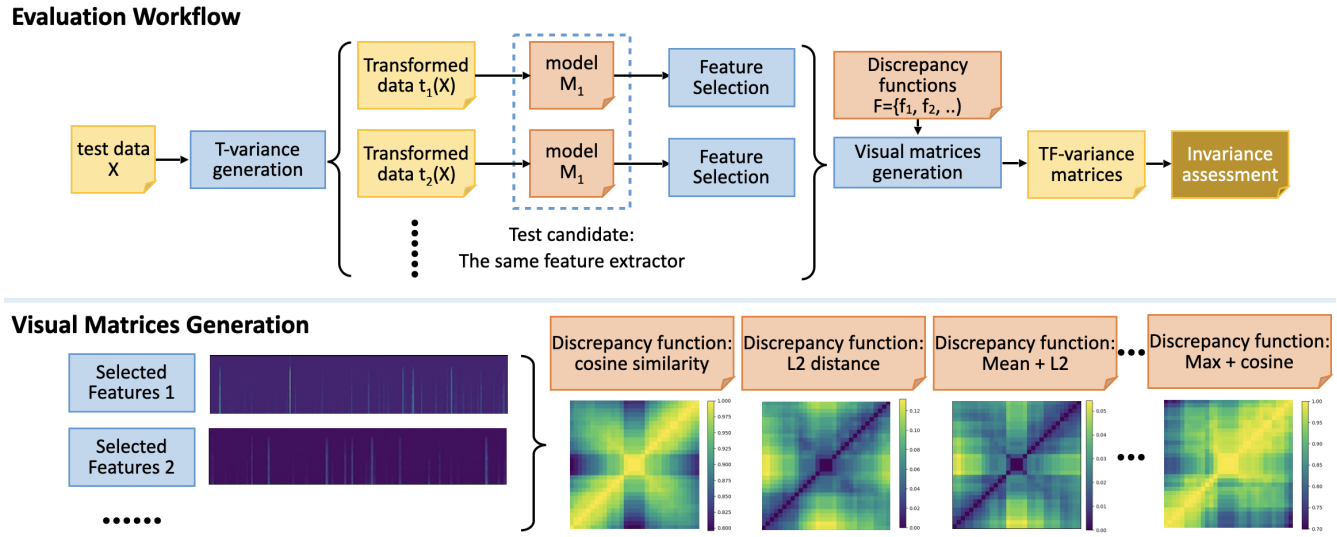

Figure 3. Testing procedure using sparse linear layers: We first define τ transformation operations $t()$, e.g., to rotate $\{-15°, -14°, \ldots, 15°\}$. And then we apply each $t()$ to all the data objects in the original testing suite X to form a transformed testing suite $t(X)$. For each feature extractor or model $\mathbf{M}$, we train τ sets of sparse linear layers $\{\theta_j | j = 1, 2, \ldots, \tau\}$ using the τ transformed testing suite $t(X)$. And then we use different discrepancy functions to evaluate the differences between the τ sets of sparse linear layers. We then use the visual patterns to analyse the invariance quality of $\mathbf{M}$.

sion of the extracted feature vectors $F(x)$, we propose that we firstly train a series of stable sparse linear classifiers (Zou & Hastie, 2005; Defazio et al., 2014). We use the variable Z to represent the feature vector, and an instance is denoted as $z = F(x)$. Similarly we have Y for the ground truth and $\tilde{Y}$ for the output of the sparse linear layers. This way, for sparse linear layers, i.e., weights ω and bias β, we have: $\tilde{Y} = Z^T\omega + \beta$. Similarly, the loss function of the sparse linear layers can be defined as $\mathcal{L}(X, Y) = \frac{1}{N}\sum(y - \tilde{y})^2$, where N is the number of input data objects. Then training a sparse linear layer is to obtain a set of weights, which are denoted as θ=$\{\omega, \beta\}$:

$$\min_{\theta} \mathcal{L}(X, Y) + \lambda R_\alpha(\theta) \tag{1}$$

where $R_\alpha = (1 - \alpha)\|\theta\|_2^2/2 + \alpha\|\theta\|_1$, which is referred to as elastic net regularisation (Zou & Hastie, 2005) for the parameter α and λ. α is a hyper-parameter controlling the regularisation terms varying between LASSO ($\alpha = 1$) and ridge ($\alpha = 0$) regression. Therefore, we can obtain a series of such weights θ when using different λ_1, λ_2, ..., λ_k:

$$\theta_i = \min_{\theta} \mathcal{L}(X, Y) + \lambda_i R_\alpha(\theta) \tag{2}$$

the greater the value of the parameter λ_i, the sparser the linear classifier is (more zero weights). This way, we will know which features contribute to the final decisions the most, and we can largely reduce the number of features that we need to visualise / consider when testing a feature extractor.

In order to test whether a feature extractor is robust against a certain type of transformation $\mathcal{T}$, we can define a sequence of transformation operations, e.g., to rotate -15°, -10°, 10°, and 15°. We denote such transformation functions as $t()$. We can then apply the same transformation function $t()$ to all data objects in the training set X, and use the transformed data objects $t(X)$ to train a sequence of sparse linear classifiers. Similarly, we will have a regularisation path:

$$\theta_{i,t} = \min_{\theta} \mathcal{L}(t(X), Y) + \lambda_i R_\alpha(\theta) \tag{3}$$

We can then visualise and compare the difference between different regularisation paths $\{\theta_{i,t} | i = 1, 2, ..., k\}$ obtained using different transformation function $t()$. We suggest arrange the transformation operations in descending or ascending order for better visualisation purpose, e.g., for rotation:

$$t_r = \{-15°, -14°, \ldots, -1°, 0°, 1°, \ldots, 14°, 15°\}$$

When visualising sparse linear layers, we fix the number of λ, and we denote the entire series of the k sparse linear layers as θ_t by omitting the i subscription (which indicates the ith lambda values). For example, in Figure 2 we show sparse linear layers $\theta_{t_{15°}}$ in the last row, and the ith line represents the ith lambda value.

4. Invariance Testing Framework

The workflow for the proposed framework is shown in Figure 3. As described above, we use sparse linear layers to

conduct the feature selection process. For each feature extractor (or model) $\mathbf{M}$, the outputs of the feature selection process are multiple sets of sparse linear layers θ_{t_j} for j in the targeted/interested interval, e.g., [-15°, 15°] for $\mathcal{T}$ being rotation.

We can then measure the difference between all the θ_{t_j}. However, if we consider τ transformation operations in the targeted/interested interval, e.g., τ angles for rotation, we will need to make $C(\tau, 2) = \binom{\tau}{2}$ comparisons for the τ sets of sparse linear layers. Therefore, we can use an invariance matrix to help scaling up the testing procedure. We define an *invariance matrix* as:

$$\Delta(\mathbf{M}, X, \mathrm{dif}) = \begin{bmatrix} \delta_{n,0} & \delta_{n,1} & \cdots & 0 \\ \vdots & \vdots & \iddots & \vdots \\ \delta_{1,0} & 0 & \cdots & \delta_{1,n} \\ 0 & \delta_{0,1} & \cdots & \delta_{0,n} \end{bmatrix}$$

where M is the testing candidate (a feature extractor), X is a test suite that is used to train multiple sets of sparse linear layers, and we define $\delta_{i,j} = \mathrm{dif}(\theta_{t_i}, \theta_{t_j})$ as the distance between θ_{t_i} and θ_{t_j}.

Similar to the testing framework proposed by (Liao et al., 2021), in order to compare the difference(s) between θ_{t_i} and θ_{t_j}, we can use different $\mathrm{dif}()$ functions, e.g., L2 distance/cosine similarity etc, to generate different variance matrices. This way, for each feature extractor, we are able to generate multiple variance matrices, which is similar to multi-modalities in the medical imaging testing process (Kumar et al., 2013). Examples can be found in Figure 3.

4.1. A New Test Oracle: Associativity

We also propose a new testing oracle on associativity: whether sparse layers obtained using combined augmentation (i.e., $t_{1,2,..,\tau}$) are the combination of the sparse layers obtained using individually augmented data, i.e., t_i, for i=$\{1, 2, ..., \tau\}$. To be more specific:

$$\theta_{\bigcap_{k=1}^{\tau} t_k} \stackrel{?}{=} \bigcap_{k=1}^{\tau} \theta_{t_k} \tag{4}$$

where τ is the number of transformation operations, and we define the union operation on a set of sparse linear layers θ_{t_k} for $k = 1, \ldots, \tau$ as "to choose the greatest value for each weight from the τ sets of sparse linear layers". We can use either L2 distance or cosine similarity to evaluate the associativity.

Finally, we are able to assess if the model is invariant or not by checking the variance matrices, the trained sparse linear layers, and the associativity score. We evaluated the invariance qualities on a small model repository consisting of 600 CNNs. The model repository, including the metadata e.g. model structure, learning rate, batch size etc, and our assessment labels will be publicly provided.

Table 1. Inter-rater reliability: Cohen's kappa scores

	Random examples	Our proposed approach
Cohen's	0.195	0.708

5. Evaluation experiments

We use the proposed invariance testing process to evaluate a model repository of 600 CNNs. In order to confirm whether the proposed testing process can assist researchers to judge the invariance qualities of models more reliably and consistently, we ask two ML professionals to assess if those models are invariant or not by 1) checking a few random examples, or by 2) checking the multiple criteria we propose, i.e., the variance matrix, sparse linear layers, and the associativity score. Note that the metadata, e.g., hyper-parameters used for training the models are not provided. In Table 1 we show that the proposed method is more consistent and reliable than checking only a few random examples. Additionally, the judgements using the proposed testing approach have a similar inter-rater reliability (IRR) score to many NLP tasks (El Dehaibi & MacDonald, 2020).

6. Discussion and Conclusion

Machine learning testing has been largely ignored by the community. While much efforts have been made to improve the performance of ML models, the only prevalent evaluation tool is to compute one single formula-based statistical score, e.g., accuracy, f1, MAP. However, similar to the Anscome's quartet (Figure 1), we cannot fully rely on statistical scores to judge the distribution of the data. We believe evaluation and testing tasks are a multi-criteria decision problem. Therefore, we do not intent to solve the problem by using only one statistical score. Instead, we train several sets of sparse linear layers, and evaluate whether a model is invariant by comparing these sparse linear layers. Additionally, we propose a new invariance testing oracle: associativity. Combined all the above non-statistical criteria, our testing approach can assist researchers to judge whether a model is invariant or not more reliably and consistently. And we conduct evaluation experiments to support our claims.

To conclude, this work revisited the invariance testing task, and converted it from a simple statistical formula-based score to a multi-criteria decision making process. The proposed testing approach utilises sparse linear layers and includes a novel testing oracle, which leads to a more reliable and consistent judgement on whether models are invariant or not.

References

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Chandra, M. A. and Bedi, S. Survey on svm and their application in image classification. *International Journal of Information Technology*, 13(5):1–11, 2021.

Crowston, K. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the future of ict research. methods and approaches*, pp. 210–221. Springer, 2012.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Dwarakanath, A., Ahuja, M., Sikand, S., Rao, R. M., Bose, R. J. C., Dubash, N., and Podder, S. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 118–128, 2018.

El Dehaibi, N. and MacDonald, E. Investigating inter-rater reliability of qualitative text annotations in machine learning datasets. In *Proceedings of the Design Society: DESIGN Conference*, volume 1, pp. 21–30. Cambridge University Press, 2020.

Frénay, B., Kabán, A., et al. A comprehensive introduction to label noise. In *ESANN*. Citeseer, 2014.

Gao, X., Saha, R. K., Prasad, M. R., and Roychoudhury, A. Fuzz testing based data augmentation to improve robustness of deep neural networks. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 1147–1158. IEEE, 2020.

Goodfellow, I., Lee, H., Le, Q., Saxe, A., and Ng, A. Measuring invariances in deep networks. *Advances in neural information processing systems*, 22:646–654, 2009.

Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

Kim, J., Feldt, R., and Yoo, S. Guiding deep learning system testing using surprise adequacy. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 1039–1049. IEEE, 2019.

Kulis, B. et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364, 2012.

Kumar, A., Kim, J., Cai, W., Fulham, M., and Feng, D. Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data. *Journal of digital imaging*, 26(6):1025–1039, 2013.

Liao, Z. Simultaneous adversarial training-learn from others' mistakes. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pp. 1–7. IEEE, 2019.

Liao, Z., Zhang, P., and Chen, M. Ml4ml: Automated invariance testing for machine learning models. *arXiv preprint arXiv:2109.12926*, 2021.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Morcos, A. S., Raghu, M., and Bengio, S. Insights on representational similarity in neural networks with canonical correlation. *arXiv preprint arXiv:1806.05759*, 2018.

Pan, S. J. and Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10): 1345–1359, 2009.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *arXiv preprint arXiv:1706.05806*, 2017.

Ruck, D. W., Rogers, S. K., and Kabrisky, M. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

Segura, S., Towey, D., Zhou, Z. Q., and Chen, T. Y. Metamorphic testing: Testing the untestable. *IEEE Software*, 37(3):46–53, 2018.

Shen, W., Wan, J., and Chen, Z. Munn: Mutation analysis of neural networks. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 108–115. IEEE, 2018.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Wong, E., Santurkar, S., and Madry, A. Leveraging sparse linear layers for debuggable deep networks. *arXiv preprint arXiv:2105.04857*, 2021.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

Zhang, J. M., Harman, M., Ma, L., and Liu, Y. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 2020.

Zhang, R. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.