ON THE LIMITS OF LEARNING REPRESENTATIONS WITH LABEL-BASED SUPERVISION

Jiaming Song, Russell Stewart, Shengjia Zhao & Stefano Ermon
Computer Science Department
Stanford University
{tsong,stewartr,zhaosj12,ermon}@cs.stanford.edu

1 INTRODUCTION

Advances in neural network based classifiers have accelerated the progress of automatic representation learning. Since the emergence of AlexNet (Krizhevsky et al., 2012), every winning submission of the ImageNet challenge (Russakovsky et al., 2015) has employed end-to-end representation learning, and due to the utility of good representations for transfer learning (Yosinski et al., 2014), representation learning has become as an important and distinct task from supervised learning. At present, this distinction is inconsequential, as supervised methods are state-of-the-art in learning transferable representations, which are widely transferred to tasks such as evaluating the quality of generated samples (Nguyen et al., 2016; Salimans et al., 2016).

In this work, however, we demonstrate that supervised learning is limited in its capacity for representation learning. Based on an experimentally validated assumption, we show that the existence of a set of features will hinder the learning of additional features. We also show that the total incentive to learn features in supervised learning is bounded by the entropy of the labels. We hope that our analysis will provide a rigorous motivation for further exploration of other methods for learning robust and transferable representations.

2 FEATURE LEARNING WITH DISCRIMINATIVE MODELS

Let $\mathbf{x} \in \mathbb{R}^d$ be observations drawn from a distribution $p_{\mathbf{X}}(\mathbf{x})$, and $\mathbf{y} \in \mathbb{R}^\ell$ be labels for $\mathbf{x}$ obtained through a deterministic mapping $\mathbf{y} = g(\mathbf{x})$. Assume we are operating in some domain (e.g. computer vision), and there exists a set of good features $\mathcal{F}_g$ that we would like to learn (e.g. a feature that denotes "tables"). These features will emerge from suitable weights of a deep neural network (e.g. a filter that detects tables), and thus must compete against an exponentially large set of bad, random features. Our goal is to learn all the good features from the dataset in the process of using a neural network to perform certain tasks.

We analyze the feature learning process by parameterizing the state of a network according to the set of features it has already learned. We then investigate the marginal value of learning an additional feature. If we have thus far learned $k-1$ features, $\{f_i\}_{i=1}^{k-1}$, we propose to measure the ease of learning the k-th feature according to the reduction in entropy of the labels when we add the new feature to improve the supervised learning performance.

$$\text{signal}(f_k) = I(\mathbf{Y}; f_k(\mathbf{X}) | f_1(\mathbf{X}), \ldots, f_{k-1}(\mathbf{X}))^1 \tag{1}$$

where we use the term "signal" of feature f_k to denote signal(f_k). This concept simply encodes our intuition that features will be easier to learn when they pertain more directly to the task at hand, and it aligns well with the "information gain" feature selection metric in Random Forests and Genomic studies (Schleper et al., 2005).

We informally use the term "learnability" to indicate the degree of incentive to learn features. The intuition is that features that add more predictive power to the current model (over a supervised task) have higher incentives to be learned.

Let us currently assume that the "learnability" of the feature corresponds to its signal (we will validate this in an experiment in Section 3). If we learn features one-by-one, then the existence of a

[1] In the remainder of the paper, We remove the $\mathbf{X}$ in $f(\mathbf{X})$ to ease notation.

set of features will decrease the conditional mutual information between a new feature and the label, and reduce the "signal", hence the "learnability". If the "signal" for a particular feature is small (or even worse, zero), then the model would receive little benefit in predicting the label correctly, hence it is unlikely for the model to learn this feature over others.

If we aim to learn k features, the sum of "signals" over all those features must be no greater than the entropy in the labels, since

$$\sum_{i=1}^{k} \text{signal}(f_i) = I(\mathbf{Y}; f_1, \ldots, f_k) \leq H(\mathbf{Y}) \tag{2}$$

which indicates that there is an upper bound for the sum of "signals". Suppose that we have already come up with features $f_1, \ldots, f_k$ that already reaches the maximum possible "signal", there will be no additional incentive to learn any new features. Therefore, if our assumption is true then we have an upper bound on the capacity of the model to learn features, which is the entropy of the labels and independent of the size of the dataset or the capacity of the model.

On the other hand, the existence of a set of features will reduce the "signal" of a new feature:

$$\text{signal}(f_k) = I(\mathbf{Y}; f_k | f_1, \ldots, f_{k-1}) \leq I(\mathbf{Y}; f_k | f_1, \ldots, f_{k-2}) \tag{3}$$

This suggests that the presence of a set of features for a supervised learning task may hinder the learning of additional features, regardless of their relation to the task (when considered as an independent feature). In the remainder of this paper, we will refer to this phenomenon under our assumption as "feature competition".

Intuitively, let us consider a dataset with images of "cats and dogs". If we have an "eye" feature that already allows us to discriminate cats versus dogs almost perfectly, there would be little incentive to learn an additional "mouth" feature, even though it is also highly related to the current task. Learning additional features that not directly related to the task, such as "tables", would be even more difficult. From an optimization perspective, the batch gradient will be close to zero when we have learned the "eye" feature; learning additional features can only be guided by the gradients from a small set of incorrectly classified samples.

3 Experimental Validations

In this section, we validate our assumption through an experiment, which suggests a high correlation between the "learnability" of a feature to its "signal". We consider our data to be images of size 28×56, where each 28×28 sub-image contains a digit[2], which we denote as x_l and x_r respectively (for "left" and "right"). We extract features using only the labels of x_l (which is denoted as y_l), and evaluate how much features from x_r is learned. The goal of this experiment is to answer the following question - **Does the "signal" for learning features in x_r correspond to its "learnability"?**

The experiment process is split into two phases termed "feature extraction" and "feature evaluation". In "feature extraction", we train a network over a dataset with (x_l, x_r) as input and y_l as labels. Features of x_l will be learned before x_r since only y_l is provided. Therefore, the conditional mutual information $I(\mathbf{Y}_l; \mathbf{X}_r | \mathbf{X}_l)$ is the total "signal" for learning all the features in x_r. We manually control $I(\mathbf{Y}_l; \mathbf{X}_r | \mathbf{X}_l)$ through two mechanisms to control the dataset:

1. Artificially introduce correlation between left label and right label by some probability ρ_r, and thus also between right digit and left label, which will increase the "signal" of the features for the right input.
2. Corrupting the left input by some probability $(1 - \rho_l)$ [3]. This will also increase the "signal" for the right input, since the left input becomes less informative about left label.

We parametrize the feature extraction model as a neural network. In particular, we learn two feature extractors f_l and f_r for x_l and x_r respectively. The concatenation of f_l and f_r is then feed through

[2]Digits are obtained from the MNIST dataset.

[3]The corruption is done by sampling $\mathbf{x}_l$ from a factored Gaussian distribution, where the mean and variance corresponds to the mean and variance of the MNIST training set.

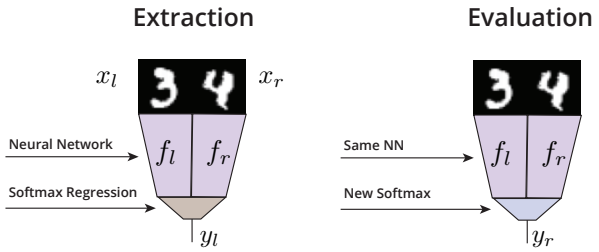
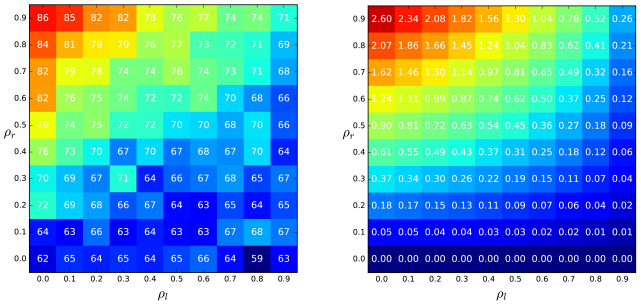


Figure 1: Illustration of the experiment.



(a) Test accuracy over $\mathbf{y}_r$ with different ρ_l and ρ_r.

(b) Calculated $I(\mathbf{Y}_l; \mathbf{X}_r|\mathbf{X}_l)$ given ρ_l and ρ_r.

Figure 2: Test accuracy over $\mathbf{y}_r$ and $I(\mathbf{Y}_l; \mathbf{X}_r|\mathbf{X}_l)$. The r-value between test accuracy and signals is 0.9213, which suggests that features having higher conditional mutual information with the labels are easier to learn.

a softmax regression classifier to make predictions. f_l, f_r and the softmax regression classifier are trained jointly.

In "feature evaluation", we measure the quality of f_r learned in "feature extraction". We perform this by training a softmax regression classifier with the concatenation of f_l and f_r as input and y_r (the label of x_r) as labels. We use "test accuracy" as a means to measure the overall quality (or equivalently, "learnability") of f_r. In this phase, we do not control the dataset, and only the softmax regression classifier is trained. Figure 1 illustrates the two phases of the experiment process.

Figure 2 contains heatmaps for the "signal" and test accuracy of y_r. We see a strong correlation between "signal" and "test accuracy", which suggests that features with higher "signal" will have higher incentive to be learned; this validates our assumption between "signal" and "learnability". Interestingly, we can observe the "feature competition" phenomenon in the upper right corners, where both inputs have high correlation to the y_l yet the quality of f_r is still low. This is because f_r competes with f_l for its "signal", which has a fixed upper bound of $H(\mathbf{Y}_l)$.

4 Conclusion

In this paper, we discuss certain limitations of supervised feature learning. We propose to measure the incentive to learn a new feature through conditional mutual information. Based on this measure, we propose the "feature competition" phenomenon in supervised learning and identify an upper bound on the capacity of supervised learning methods to extract features. Experimental results support our claims, and we observe a strong correlation between "signal" and "learnability", as well as strong "feature competition" between features. We hope that our analysis will motivate further explorations in representation learning without labels, such as generative representation learning.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2226–2234, 2016.

Christa Schleper, German Jurgens, and Melanie Jonuscheit. Genomic studies of uncultivated archaea. *Nature Reviews Microbiology*, 3(6):479–488, 2005.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Model	CNN	AE	GAN	WGAN
Accuracy	67.93 (84.31)	89.95 (82.18)	90.38 (82.27)	91.37 (84.97)

Table 1: Test accuracy for $\mathbf{y}_r$, given f learned by different architectures. The numbers in the brackets indicate using weights that are not trained, which is a baseline for the case of random features.

A Corner Cases of Figure 2

Figure 2 reflects some corner cases:

Bottom row y_r (hence also x_r) has no correlation with the y_l, hence there is no signal to learn features from the right part, and f_r would perform no better than random initialization of the weights.

Top left corner The x_r has high correlation with the y_l while the x_l doesnt (because of added noise), this has the highest signal, since this is essentially assigning the y_l to the x_r.

Top right corner Both inputs have high correlation to the y_l. Due to "feature competition", relatively few features corresponding to the x_r are learned.

B Feature Learning with Generative Models

If supervised learning is bounded in its capacity for feature extraction by the entropy of the labels, what of unsupervised learning? In this section, we show that one family of unsupervised learning methods, Generative Adversarial Networks (GAN Goodfellow et al. (2014)), is not impacted by feature competition under limited assumptions.

In GANs, a generator network, G, generates samples (with generative distribution $\mathcal{G}$), and a discriminator network, $D(\mathbf{x})$, attempts to distinguish between those samples and real data. D is presented with a labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $y = 1$ if $\mathbf{x} \sim \mathcal{D}$, and $y = 0$ if $\mathbf{x} \sim \mathcal{G}$; the two classes are balanced, so $H(y) = 1$. [4]

Assume that G and D have already learned $k-1$ features $f_1, \ldots, f_{k-1}$, where the discriminator cannot separate samples from $\mathcal{G}$ and $\mathcal{D}$ with only these features. This indicates that $\Pr(y = 1 | f_1(\mathbf{x}), \cdots, f_{k-1}(\mathbf{x})) = 0.5$ for all $\mathbf{x}$ in the dataset, and that $H(y|f_1, \ldots, f_{k-1}) = 1$. Thus, the discriminator is in a state of confusion. We measure the motivation of D for learning a new feature f_k to be

$$\begin{aligned} I(y; f_k | f_1, \ldots, f_{k-1}) &= H(y|f_1, \ldots, f_{k-1}) - H(y|f_1, \ldots, f_{k-1}, f_k) \\ &= 1 - H(y|f_1, \ldots, f_{k-1}, f_k) \\ &\geq 1 - H(y|f_k) \end{aligned} \tag{4}$$

where $H(y|f_k) \in [0, 1]$ is a measure of similarity between the distributions on the feature f_k for real and generated samples. $H(y|f_k) = 1$ if and only if $f_k(\mathbf{x})$ is identically distributed almost everywhere for G and D. Thus, if the generated distribution does not yet match the real data distribution along f_k, we will have positive signal to learn. Importantly, this lower bound has no dependence on previously learned features, $f_1, \ldots, f_{k-1}$. That is, when the discriminator is in a state of confusion, we have no feature competition.

C Experimental Validation of Generative Models

We have shown theoretically that one class of generative models, GANs, is not limited by the same upper bound on feature learning signal as discriminative models such as CNNs. We now empirically test these implications by revisiting the two-digit experiment from Section 2. We set $\rho_l = 1$ and $\rho_r = 0$, where the two digits are selected completely at random (and where feature learning for

[4] In this section, we consider the distribution over $\mathbf{x}$ to be the average of $\mathcal{G}$ and $\mathcal{D}$ for entropy and mutual information terms.

$\mathbf{x}_r$ using supervised learning methods performed worst). We consider four frameworks - a feed forward convolutional net (CNN); a traditional GAN (Goodfellow et al., 2014); a recently proposed Wasserstein GAN (WGAN, Arjovsky et al. (2017)) and an Autoencoder[5]. For the four frameworks, we use the same CNN architecture and set the output of f to be the 100 neurons at the second top layer. The results, shown in Table 1, demonstrate that in spite of the absence of labels, the features learned by all three generative models we considered, including GANs, AEs, and WGANs, were useful in the subsequent task of learning to recognize the right digit.

D MOTIVATION FOR BALANCING GANS VIA LOSS STATISTICS

Assume that the "state of confusion" assumption breaks for D, such that D has learned $l > 1$ more features than G has learned, and it can classify better than random guessing. Therefore, $H(y|f_1, \ldots, f_{k-1}) = 1$, $H(y|f_1, \ldots, f_{k+l-1}) < 1$, and

$$H(y|f_k, \ldots, f_{k+l-1}) < 1 \tag{5}$$

The motivation of D for learning a new feature f_{k+l} then becomes

$$\begin{aligned} I(y; f_{k+l}|f_1, \ldots, f_{k+1-1}) = &\ H(y|f_1, \ldots, f_{k+l-1}) - H(y|f_1, \ldots, f_{k+l}) \\ = &\ H(y|f_k, \ldots, f_{k+l-1}) - H(y|f_k, \ldots, f_{k+l}) \end{aligned} \tag{6}$$

which is no longer independent of $f_k, \ldots, f_{k+l-1}$ because of Equation 5. This is analogous to the supervised learning setting - D is simply trying to learn a new features f_{k+l}, given all the previous features $f_k, \ldots, f_{k+l-1}$ to optimize a fixed objective defined by features $f_1, \ldots, f_{k-1}$.

If D has learned $k + l - 1$ features and G has learned $k - 1$ features, then G is motivated to learn the proper distribution for feature f_k to minimize $H(y|f_1, \ldots, f_{k+l-1})$. However, this quantity will still be smaller than one even if we assume G learns the correct distribution on f_k, so the incentive for G becomes [6]

$$\begin{aligned} &H_{f_k}(y|f_1, \ldots, f_{k+l-1}) - H(y|f_1, \ldots, f_{k+l-1}) \\ &< 1 - H(y|f_1, \ldots, f_{k+l-1}) \\ &= H(y) - H(y|f_k, \ldots, f_{k+l-1}) = I(y; f_k, \ldots, f_{k+l-1}) \end{aligned} \tag{7}$$

Notice that the mutual information $I(y; f_k, \ldots, f_{k+l-1})$ is exactly the sum of motivation for the discriminator to learn features $f_k, \ldots, f_{k+l}$. This implies that if we continue to allow D and G to learn one feature at a time, which is the case where we do not attempt to balance GANs via loss statistics, G will not catch up with D in one step; D, on the other hand, the advantage in D will cause it to suffer from the "feature competition" challenge, where it has less incentive to learn features than it should.

One obvious method to counter this is by balancing GANs via loss statistics; although D and G suffers from feature competition during learning $f_k, \ldots, f_{k+l-1}$, G will catch up if it learns multiple features consecutively, so that it makes D confused again, where $V(D, G) = \log 4$. However, we are not promoting the strategy where G and D should be trained more whenever its loss exceeds some predetermined value, but we believe principled approaches to tackle this problem will be valuable to training of GANs.

[5] We do not split f into two networks in the convolution setting.

[6] We use H_{f_k} to denote the expectation is performed over the distribution when G has learned the proper distribution for feature f_k.