Disentanglement by penalizing correlation

Mikael Kågebäck & Olof Mogren
Chalmers University of Technology, Sweden
[kageback,mogren]@chalmers.se

Abstract

An important reason for the success of deep neural networks this is their capability to automatically learn representations of data in levels of abstraction, increasingly disentangling the data as the internal transformations are applied. In this paper we propose a novel regularization method that actively penalize covariance between dimensions of the hidden layers in a network, driving the model towards a more disentangled solution. This makes the network learn linearly uncorrelated representations which increases interpretability while obtaining good results on a number of tasks, as demonstrated by our experimental evaluation. Further, the proposed technique effectively disables superfluous dimensions, compressing the representation to the dimensionality of the underlying data. Our approach is computationally cheap and can be applied as a regularizer to any gradient-based learning model.

1 Introduction

A good data representation should uncover underlying factors in the data while being useful for some task. Deep networks learn representations of increasing abstraction, disentangling the causes of variation in the underlying data (Bengio et al., 2013). Formal definitions of disentanglement are lacking, although Ver Steeg & Galstyan (2015); Achille & Soatto (2017) both use the total correlation as a measure of disentanglement. Inspired by this, we consider a simpler objective: a representation disentangles the data well when its components do not correlate, and we explore the effects of penalizing this linear dependence betweeen different dimensions in the representation.

We propose L_Σ regularization, a novel regularization scheme that penalizes the correlation between the dimensions of the learned representations. The approach is very versatile and can be applied to any gradient-based machine learning model that learns its own distributed vector representations. Compared to previous work on learning independent nonlinear representations our approach is simpler, and does not impose restrictions on the model used. The approach strongly encourages the model to find the dimensionality of the data, something that is verified by the experimental evaluation. This can be of great utility when pruning a network, or to decide when a network needs a larger capacity. The disabling of activations in the internal representation can be viewed as (and used for) dimensionality reduction. The proposed approach allows for interpretability of the activations computed in the model, such as isolating specific underlying factors. The solution is computationally cheap, and can be applied without modification to many gradient-based machine learning models that learns distributed representations. Moreover, we present an extensive experimental evaluation on a range of tasks on different data modalities using different model layouts, which shows that the proposed approach disentangles the data well; we do get uncorrelated components in the resulting internal representations, while retaining the performance of the models on their respective task.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2 Disentanglement by penalizing correlation

We present a novel regularizer based on the covariance of the activations in a neural network layer over a batch of examples. The aim of the regularizer is to penalize the covariance between dimensions in the layer to decrease linear correlation.

Definition The covariance regularization term (L_Σ) for a layer, henceforth referred to as the coding layer, is computed as $L_\Sigma = \frac{1}{p^2}||\mathcal{C}||_1$ where p is the dimensionality of the coding layer, $||\mathcal{C}||_1 = \sum_{i,j=1}^{N} |\mathcal{C}_{ij}|$ is the element wise L1 matrix norm of $\mathcal{C}$, and $\mathcal{C} \in \mathcal{R}^{p\times p}$ is the sample covariance of the activations in the coding layer over N examples $\mathcal{C} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{H} - \mathbf{1}_N\bar{\mathbf{h}})^T(\mathbf{H} - \mathbf{1}_N\bar{\mathbf{h}})$.Further, $\mathbf{H} = [\mathbf{h}_1; ...; \mathbf{h}_N]$ is a matrix of all activations in the batch, $\mathbf{1}_N$ is an N-dimensional column vector of ones, and $\bar{\mathbf{h}}$ is the mean activation.

Usage As L_Σ has the structure of a regularizer, it can be applied to most gradient based models without changing the underlying architecture. In particular, L_Σ is simply computed based on select layers and added to the error function, e.g. $Loss = Error + \lambda L_\Sigma$

3 Experiments

This section describes the experimental evaluation of L_Σ regularization in different settings.

3.1 Evaluation metrics

Mean Absolute Pearson Correlation (MAPC) Pearson correlation report the normalized linear correlation between variables $\in [-1, 1]$. MAPC is the average absolute value of the correlation between all dimensionss, i.e. $\mathbf{MAPC} = (2/(p^2 - p))\sum_{i<j}^{p} |\mathcal{C}_{ij}|/\sqrt{\mathcal{C}_{ii}}\sqrt{\mathcal{C}_{jj}}$
Covariance/Variance Ratio (CVR) Mean absolute Pearson correlation becomes ill defined when the variance of one (or both) of the variables approaches zero. We define a related measure where all variances are summed for each dimension. More precise, the score is computed as: $\mathbf{CVR} = \frac{1}{p^2}\frac{||\mathcal{C}||_1}{\mathbf{tr}(\mathcal{C})}$where $||\mathcal{C}||_1$ is defined as in Sec 2. The intuition behind CVR is simply to measure the fraction of all information that is captured in a linear uncorrelated fashion within the coding layer.
Top d-dimension Variance/total variance (TdV) TdV measure to what degree the total variance is captured inside the variance of the top d dimensions.
90% Utilized Dimensions (UD$_{90\%}$) The number of dimensions that needs to be kept to retain 90% of the total variance.

3.2 Dimensionality reduction

The purpose of this experiment is to investigate if it is possible to disentangle independent data that has been projected to a higher dimension using a random projection, i.e. we would like to find the principal components of the original data.

The model we employ in this experiment is an auto encoder consisting of a linear $p = 10$ dimensional coding layer and a linear outputlayer. The model is trained using the proposed covariance regularization L_Σ on the coding layer. The data is generated by sampling a $d = 4$ dimensional vector of independent features $z \sim N(0, \Sigma)$, where $\Sigma \in \mathcal{R}^{d\times d}$ is constrained to be non-degenerate and diagonal. However, before the data is fed to the autoencoder it is pushed through a random linear transformation $x = \Omega z$. The goal of the model is to reconstruct properties of z in the coding layer while only having access to x. The model is trained on 10000 iid random samples for 10000 epochs. 9 experiments were performed with different values for the regularization constant λ. The first point on each curve (in Fig 1 and 2) is $\lambda = 0$, i.e. no regularization, followed by 8 points logarithmically spaced between 0.001 and 1. Each experiment is repeated 10 times using a different random projection Ω and the average is reported.

The result of the experiment is reported using all four metrics defined in Sec 3.1. The result in terms of MAPC and CVR is reported in Fig 1. The first thing to notice is that L_Σ consistently lead to lower correlation while incurring less MSE penalty compared to L_1. Further, looking at the MAPC it is interesting to notice that it is optimal for a very small values of L_Σ. This is because higher amounts of L_Σ leads to lowering of the dimensionality of the data, see Fig 2, which in turn yields

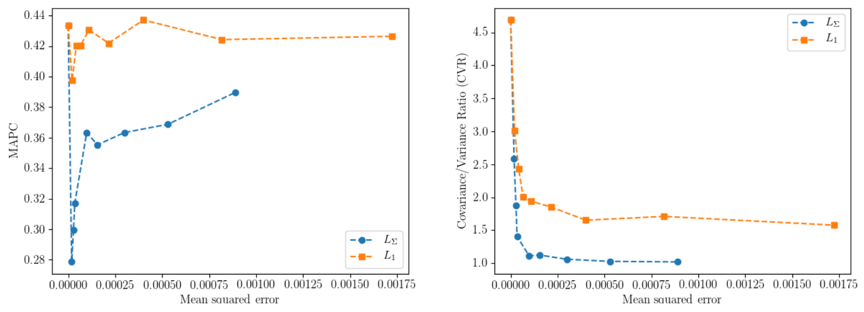


Figure 1: The amount of residual linear correlation after training the model with L_Σ and L_1 regularization respectively, measured in MAPC (left) and CVR (right). The first point on each curve corresponds to $\lambda = 0$, i.e. no regularization, followed by 8 points logarithmically spaced between 0.001 and 1. All scores are averaged over 10 experiments using a different random projection (Ω).

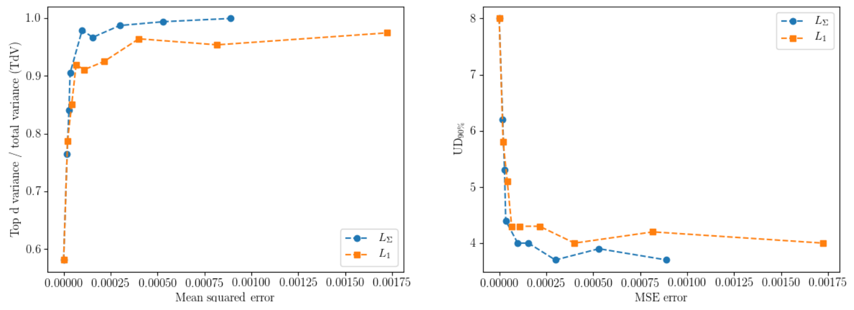


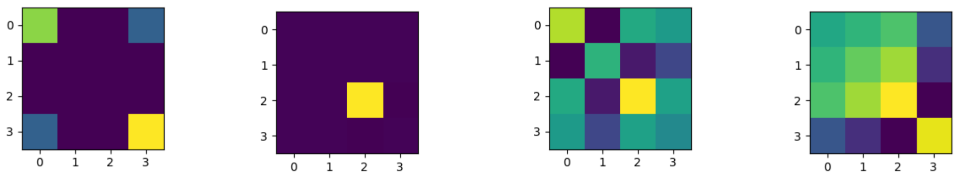
Figure 2: The resulting dimensionality the coding layer after training the model with L_Σ and L_1 regularization respectively, measured in TdV (left) and $UD_{90\%}$ (right). The first point on each curve corresponds to $\lambda = 0$, i.e. no regularization, followed by 8 points logarithmically spaced between 0.001 and 1.All scores are averaged over 10 experiments using a different random projection (Ω).

unpredictable Pearson correlation scores between these inactivated neurons. However, this effect is compensated for in CVR for which L_Σ quickly converges towards the optimal value of one, which in turn indicates no presence of linear correlation. Turning the attention to dimensionality reduction, Fig 2 shows that L_Σ consistently outperform L_1. Further, looking closer at the TdV score, L_Σ is able to compress the data almost perfectly, i.e. TdV=1, at a very small MSE cost while L_1 struggle even when accepting a much higher MSE cost. Further, the $UD_{90\%}$ scores again show that L_Σ achieves a higher compression at lower MSE cost. In this instance the underlying data was of 4 dimensions which L_Σ quickly achieves. At higher amounts of L_Σ the dimensionality even occasionally fall to 3, however, this is due to the threshold of 90%.

3.3 Deep network of uncorrelated features

In Sec 3.2 we showed that we can learn a minimal orthogonal representation of data that is generated to ensure that each dimension is independent. However, in reality it is not always possible to encode the necessary information, to solve the problem at hand, in an uncorrelated coding layer. However, using a deep network it should be possible to learn such a nonlinear transformation that enables uncorrelated features in higher layers. To test this in practice on a problem that has this property but still is small enough to easily understand we turn to the XOR problem.

It is well known that the XOR problem can be solved by a neural network of one hidden layer consisting of a minimum of two units. However, instead of providing this minimal structure we would like the network to discover it by itself during training. Hence, the model used is intentionally over-specified consisting of two hidden layers of four logistic units each followed by a one dimensional logistic output layer. The model was trained on XOR examples, e.g. [1,0]=1, in a random order until convergence with L_Σ applied to both hidden layers with $\lambda = 0.2$.



(a) Layer 1, L_Σ (b) Layer 2, L_Σ (c) Layer 1, baseline (d) Layer 2, baseline

Figure 3: Covariance matrices of the hidden layers when trained while applying L_Σ regularization (a and b) to solve the XOR problem. Layer one has learned to utilize unit zero and three while keeping the rest constant, and in layer two only unit two is utilized. This learned structure is the minimal solution to the XOR problem. The baseline model (c and d) is trained without L_Σ and depicts a less interpretable solution to XOR.

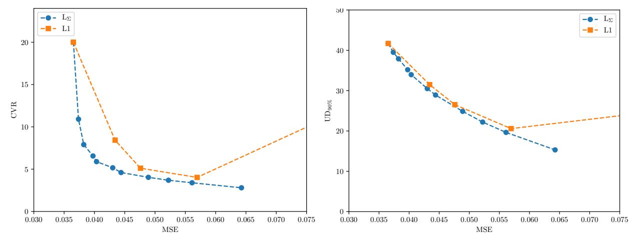


Figure 4: Results on CIFAR-10 test set using L_Σ and L^1, respectively. **Left:** CVR against MSE. **Right:** UD$_{90\%}$ against MSE. Each point in the plot: $\lambda \in [0.0, 0.2, ..., 10.24]$.

	CVR	**UD$_{90\%}$**	**MSE**
L_Σ	6.56	35.18	0.0398
L^1	4.03	20.59	0.0569
W/O	20.00	41.69	0.0365

Table 1: Results from the experiments on CIFAR.

As can be seen in Figure 3 the model was able to learn the optimal structure of exactly 2 dimensions in the first layer and one dimension in the second, whereas the baseline did not. Further, as expected, the first layer do encode a negative covariance between the two active units while the second layer is completely free from covariance. Note that, even though the second hidden layer is not the output of the model it does encode the result in that one active neuron.

3.4 Non-linear uncorrelated convolutional features

Convolutional autoencoders have been used to learn visual features. Here, we will see that it is possible to train a deep convolutional autoencoder on real-world data and learn representations that have low covariance, while retaining the reconstruction quality.

To keep it simple, the encoder part of the model used two convolutional layers and two fully connected layers, with a total of roughly 500.000 parameters in the whole model. The regularization was applied to the coding layer which has 84 dimensions, giving a bottleneck effect. The model was trained and evaluated on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009), containing 32x32 pixel colour images. Official test set: 10,000 images, 5,000 images from the training set of 50,000 was set aside for validation. We compare the results from using L_Σ regularization with L^1 regularization and with no regularization at all (W/O). The model was trained using Adam (Kingma & Ba, 2015) with early stopping. Initial learning rate: 0.001, batch size: 100, λ: 0.08. The reported scores are averages from training the model five times with different initialization.

The results (see Table 1) show that the high-level features become more disentangled and has a lower CVR (6.56) using L_Σ regularization. Without regularization, the score is 20.00, and with L^1 regularization the score is 4.03. The model with L_Σ regularization obtains an MSE of 0.0398, roughly the same as without regularization (0.0365), both of which are much better than using L^1 regularization, with an MSE of 0.0569. Figure 4 shows the CVR score vs the MSE, illustrating that L_Σ leads to more disentangled representations.As you increase the regularization factor L_Σ regularization pushes down the CVR quickly, while retaining an MSE error that is almost constant. L^1 regularization also lower CVR, although slower, and at a higher MSE cost. The UD$_{90\%}$ results show that L_Σ encourages representations that concentrate the variation.

4 Related work

Different notions of independence have been proposed as useful criteria to learn disentangled representations. Principal component analysis (PCA; Pearson, 1901) can find linearly uncorrelated variables. Nonlinear PCA often refers to neural autoencoders (Kramer, 1991) without specific regularization. Independent component analysis (ICA; Hyvärinen et al., 2004) has a somewhat stronger requirement of statistical independence. Dinh et al., (2015; 2017) used the substitution rule of differentiation as a motivation for the model. Using a fixed factorial prior, they encouraged the model to learn independent representations. Brakel & Bengio (2017) used adversarial training to make a generative network learn a factorized, independent distribution $p(\mathbf{z})$. Our approach is more flexible and portable, as it can be applied as a regularization to learn uncorrelated components in any gradient-based model that learns internal representations.

5 Conclusions

In this paper, we have presented L_Σ regularization, a novel regularization scheme based on penalizing the covariance between dimensions of the internal representation learned in a hierarchical model. The proposed regularization scheme helps models learn linearly uncorrelated variables in a non-linear space. While techniques for learning independent components follow criteria that are more strict, our solution is flexible and portable, and can be applied to any feature-learning model that is trained with gradient descent. Our method has no penalty on the performance on tasks evaluated in the experiments, while it does disentangle the data. We saw that our approach performs well applied to a standard deep convolutional autoencoder on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009); the resulting model performs comparable to the model without regularization, while we can also see that the covariances between dimensions in the internal representation decrease drastically.

Acknowledgments

The authors would like to acknowledge the project *Towards a knowledge-based culturomics* supported by a framework grant from the Swedish Research Council (2012–2016; dnr 2012-5738), and the project *Data-driven secure business intelligence* grant IIS11-0089 from the Swedish Foundation for Strategic Research (SSF).

References

A. Achille and S. Soatto. Emergence of Invariance and Disentangling in Deep Representations. *ICML Workshop on Principled Approaches to Deep Learning*, 2017.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

Philémon Brakel and Yoshua Bengio. Learning independent features with adversarial nets for non-linear ICA. *arXiv preprint arXiv:1710.05050*, 2017.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *ICLR*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *ICLR*, 2017.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

Greg Ver Steeg and Aram Galstyan. Maximally informative hierarchical representations of high-dimensional data. In *Artificial Intelligence and Statistics*, pp. 1004–1012, 2015.