Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction

(2017)

Richard Zhang, Phillip Isola, Alexei A. Efros.

Notes

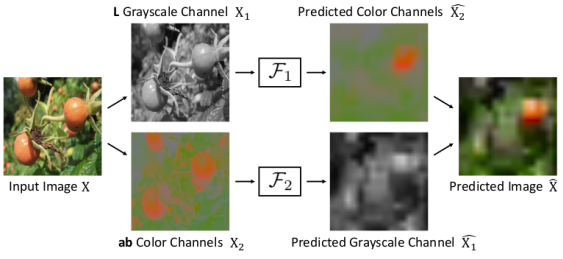
Contributions

The objective is to learn useful representation in a unsupervised manner, based only on raw data. The authors propose a new type of autoencoders called split brain autoencoder based on the draw back of the current designs and types of autoencoders.

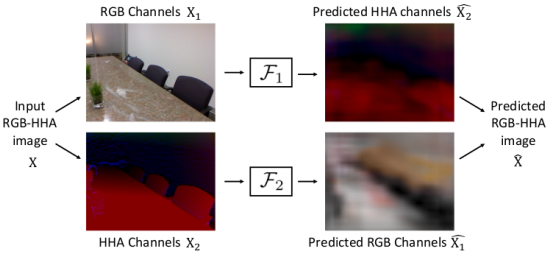
To avoid identity mapping, traditional autoencoder are limited by bottleneck, the smaller they are, the greater the forced representation and the smaller the information content that can be expressed. As alternatives, denoising autoencoders are trained to remove an *iid* noise that was added to the input, context encoders predict the masked region on image, on the other hand, both of these autoencoders suffer of a *domain gap*, where they trained on one type of data (noisy - masked) and tested on clean data. Cross-channel autoencoders, where we predict one subsets of data channels from an other, turns out to be quite effective and able to learn useful representation for downstream tasks, the problem with this cross channel encoding objective, is that the model is blind to a subset of the data (colors) that may contain useful information that will not be learned otherwise, this is an *input handicap*.

	auxiliary task type	**domain gap**	**input handicap**
Autoencoder [20]	reconstruction	no	no
Denoising autoencoder [44]	reconstruction	suffers	no
Context Encoder [35]	prediction	no	suffers
Cross-Channel Encoder [49, 28]	prediction	no	suffers
Split-Brain Autoencoder	**prediction**	**no**	**no**

The authors propose split brain autoencoder, taking the advantages from cross-channel encoders while being able to learn features from the entire input. This is done by introducing a single split to the network, resulting in two disjoint subnetworks, each one is trained as a cross-channel encoder. The pairs of tasks explored are RGB / depth prediction: predicting depth from images and images from depth, L / a,b channels. This way we'll be able to extract features from the entirety of the inputs.



(a) ***Lab* Images**



(b) **RGB-D Images**

Method

Cross-channel Autoencoders To learn a subset of the data using the rest, the inputs $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ are split into two: $\mathbf{X_1} \in \mathbb{R}^{H \times W \times C_1}$ and $\mathbf{X_2} \in \mathbb{R}^{H \times W \times C_2}$ with $C_1, C_2 \subseteq C$, then a deep network $\mathcal{F}$ is trained to predict a set of channels $\mathbf{X_2}$ from $\mathbf{X_1}$, and the network can be trained with either $L2$ loss:

$$\ell_2\left(\mathcal{F}\left(\mathbf{X}_1\right), \mathbf{X}_2\right) = \frac{1}{2} \sum_{h,w} \left\| \mathbf{X}_{2h,w} - \mathcal{F}\left(\mathbf{X}_1\right)_{h,w} \right\|_2^2$$

Or a classification loss, where the space of possibilities is quantified, which works better when there is multiple plausible predictions for a given input (a car can have many possible and plausible colorizations):

$$\ell_{cl}\left(\mathcal{F}\left(\mathbf{X_1}\right), \mathbf{X_2}\right) = -\sum_{h,w} \sum_{q} \mathcal{H}\left(\mathbf{X_2}\right)_{h,w,q} \log\left(\mathcal{F}\left(\mathbf{X_1}\right)_{h,w,q}\right)$$

Split-Brain Autoencoders Using two cross-channel autoencoders $\mathcal{F}_1$ and $\mathcal{F}_2$, we can predict both $\mathbf{X}_1$ and $\mathbf{X}_2$ for each other.

$$\mathcal{F}_1^* = \arg\min_{\mathcal{F}_1} L_1\left(\mathcal{F}_1\left(\mathbf{X}_1\right), \mathbf{X}_2\right) \quad \text{and} \quad \mathcal{F}_2^* = \arg\min_{\mathcal{F}_2} L_2\left(\mathcal{F}_2\left(\mathbf{X}_2\right), \mathbf{X}_1\right)$$

To train on the whole input $\mathbf{X}$, the two encoders can be concatenated layer wise at each level, and the resulting encoder $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$ is trained on the full input. For CNNs, such encoder can be implemented with each convolution operation `group` parameter is set to two, this way each half of the convolution, in each layer will only see the corresponding half of the inputs (the channels dimensions of $\mathbf{X_1}$ and $\mathbf{X_2}$ needs to be equal, $C_1 = C_2 = C/2$).

The authors propose another alternative **aggregation technique** for training a split brain encoder, which is to use one network without any grouping, and train it simultaneously to predict $\mathbf{X_1}$ and $\mathbf{X_2}$, but also a reconstruction objective $\mathbf{X}$ so that the network sees the whole input, with a hyperparamter $\lambda \in \left[0, \frac{1}{2}\right]$:

$$\mathcal{F}^* = \arg\min_{\mathcal{F}} \lambda L_1\left(\mathcal{F}\left(\mathbf{X_1}\right), \mathbf{X_2}\right) + \lambda L_2\left(\mathcal{F}\left(\mathbf{X_2}\right), \mathbf{X_1}\right) + (1 - 2\lambda) L_3(\mathbf{X}, \mathcal{F}(\mathbf{X}))$$

Results

- Split-Brain Autoencoder (cl,cl): A split-brain autoencoder, with one half performing colorization, and the other half performing grayscale prediction. The loss is classification.
- Split-Brain Autoencoder (reg,reg): same as above with an $L2$ loss.
- Ensembled $L \rightarrow ab$: Ensembling two networks, both prediction the colors, one with a classification objective, the other with a regression objective.
- $(L, ab) \rightarrow (ab, L)$: The aggregation technique, without the autoencoding loss term.
- $(L, ab, Lab) \rightarrow (ab, L, Lab)$: The aggregation technique, with $\lambda = 1/3$.

And for ablation studies, a number of autoencoder types can be tested:

- Cross-channel encoder: $L \rightarrow ab(reg)$, $L \rightarrow ab(reg)$, $ab \rightarrow L(reg)$ and $ab \rightarrow L(reg)$
- Autoencoder: $Lab \rightarrow Lab$
- Denoising Autoencoder: $Lab(drop50) \rightarrow Lab$

Task Generalization on ImageNet Classification [37]					
Method	**conv1**	**conv2**	**conv3**	**conv4**	**conv5**
ImageNet-labels [26]	19.3	36.3	44.2	48.3	50.5
Gaussian	11.6	17.1	16.9	16.3	14.1
Krähenbühl et al. [25]	17.5	23.0	24.5	23.2	20.6
[1]Noroozi & Favaro [31]	19.2	30.1	34.7	33.9	28.3
Doersch et al. [8]	16.2	23.3	30.2	31.7	29.6
Donahue et al. [9]	**17.7**	24.5	31.0	29.9	28.0
Pathak et al. [35]	14.1	20.7	21.0	19.8	15.5
Zhang et al. [49]	13.1	24.8	31.0	32.6	31.8
Lab→Lab	12.9	20.1	18.5	15.1	11.5
Lab(drop50)→Lab	12.1	20.4	19.7	16.1	12.3
L→ab(cl)	12.5	25.4	32.4	33.1	32.0
L→ab(reg)	12.3	23.5	29.6	31.1	30.1
ab→L(cl)	11.6	19.2	22.6	21.7	19.2
ab→L(reg)	11.5	19.4	23.5	23.9	21.7
(L,ab)→(ab,L)	15.1	22.6	24.4	23.2	21.1
(L,ab,Lab)→(ab,L,Lab)	15.4	22.9	24.0	22.0	18.9
Ensembled L→ab	11.7	23.7	30.9	32.2	31.3
Split-Brain Auto (reg,reg)	17.4	27.9	33.6	34.2	32.3
Split-Brain Auto (cl,cl)	**17.7**	**29.3**	**35.4**	**35.2**	**32.8**

Table 2: **Task Generalization on ImageNet Classification** To test unsupervised feature representations, we train linear logistic regression classifiers on top of each layer to perform 1000-way ImageNet classification, as proposed in [49]. All weights are frozen and feature maps spatially resized to be ∼9000 dimensions. All methods use AlexNet variants [26], and were pre-trained on ImageNet without labels, except for **ImageNet-labels**. Note that the proposed split-brain autoencoder achieves the best performance on all layers across unsupervised methods.

Dataset & Task Generalization on Places Classification [50]					
Method	**conv1**	**conv2**	**conv3**	**conv4**	**conv5**
Places-labels [50]	22.1	35.1	40.2	43.3	44.6
ImageNet-labels [26]	22.7	34.8	38.4	39.4	38.7
Gaussian	15.7	20.3	19.8	19.1	17.5
Krähenbühl et al. [25]	21.4	26.2	27.1	26.1	24.0
[1]Noroozi & Favaro [31]	23.0	32.1	35.5	34.8	31.3
Doersch et al. [8]	19.7	26.7	31.9	32.7	30.9
Wang & Gupta [46]	20.1	28.5	29.9	29.7	27.9
Owens et al. [33]	19.9	29.3	32.1	28.8	29.8
Donahue et al. [9]	**22.0**	28.7	31.8	31.3	29.7
Pathak et al. [35]	18.2	23.2	23.4	21.9	18.4
Zhang et al. [49]	16.0	25.7	29.6	30.3	29.7
L→ab(cl)	16.4	27.5	31.4	32.1	30.2
L→ab(reg)	16.2	26.5	30.0	30.5	29.4
ab→L(cl)	15.6	22.5	24.8	25.1	23.0
ab→L(reg)	15.9	22.8	25.6	26.2	24.9
Split-Brain Auto (cl,cl)	21.3	**30.7**	**34.0**	**34.1**	**32.5**

Table 3: **Dataset & Task Generalization on Places Classification** We train logistic regression classifiers on top of frozen pre-trained representations for 205-way Places classification. Note that our split-brain autoencoder achieves the best performance among unsupervised learning methods from `conv2-5` layers.

Task and Data Generalization on PASCAL VOC [12]							
	Classification [25] (%mAP)			**Detection [15] (%mAP)**		**Seg. [29] (%mIU)**	
frozen layers		**conv5**	**none**		**none**		**none**
fine-tuned layers	**Ref**	**fc6-8**	**all**	**Ref**	**all**	**Ref**	**all**
ImageNet labels [26]	[49]	78.9	79.9	[25]	56.8	[29]	48.0
Gaussian	[35]	–	53.3	[35]	43.4	[35]	19.8
Autoencoder	[9]	16.0	53.8	[35]	41.9	[35]	25.2
Krähenbühl et al. [25]	[9]	39.2	56.6	[25]	45.6	[9]	32.6
Jayaraman & Grauman [23]	–	–	–	[23]	41.7	–	–
Agrawal et al. [1]	[25]	–	52.9	[25]	41.8	–	–
Agrawal et al. [1]†	[9]	31.0	54.2	[25]	43.9	–	–
Wang & Gupta [46]	[25]	–	62.8	[25]	47.4	–	–
Wang & Gupta [46]†	[25]	–	63.1	[25]	47.2	–	–
Doersch et al. [8]	[25]	–	55.3	[25]	46.6	–	–
Doersch et al. [8]†	[9]	55.1	65.3	[25]	51.1	–	–
Pathak et al. [35]	[35]	–	56.5	[35]	44.5	[35]	29.7
Donahue et al. [9]†	[9]	52.3	60.1	[9]	46.9	[9]	35.2
Misra et al. [30]	–	–	–	[30]	42.4	–	–
Owens et al. [33]	▷	54.6	54.4	[33]	44.0	–	–
Owens et al. [33]†	▷	52.3	61.3	–	–	–	–
Zhang et al. [49]†	[49]	61.5	65.9	[49]	46.9	[49]	35.6
Larsson et al. [28] ◇	[28]	–	65.9	–	–	[28]	**38.4**
Pathak et al. [34] ◇	[34]	–	61.0	[34]	**52.2**	–	–
Split-Brain Auto (cl,cl)†	▷	**63.0**	**67.1**	▷	46.7	▷	36.0