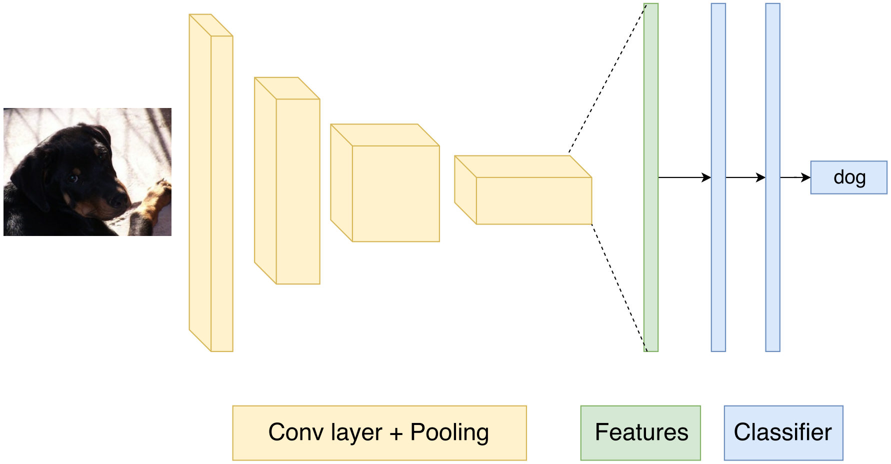
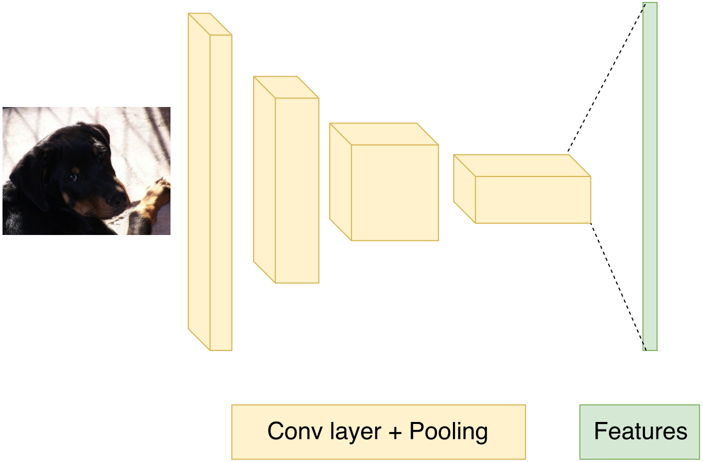
Adversarial Feature Learning
J.Donahue, T. Darrell, P. Krähenbühl

Pierre Ménard

March 22, 2018


dog
Conv layer + Pooling
Features
Classifier



How to learn features without labels ?

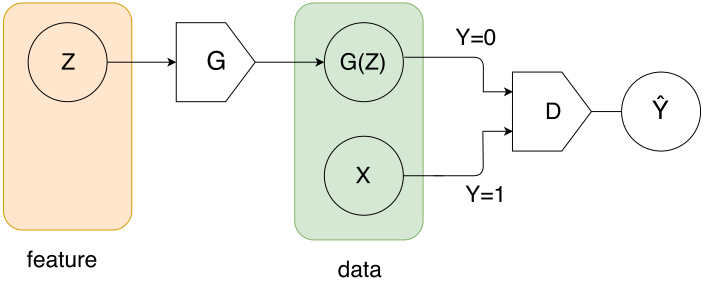
Generative Adversarial Network

Inputs data: $X \sim P_X$ on Ω_X random noise: $Z \sim P_Z$ on Ω_Z

Generator $G : \Omega_Z \mapsto \Omega_X$

Goal: Find G s.t. $G(Z) \overset{\mathcal{L}}{\approx} X$

Generative Adversarial Network



Inputs data: $X \sim P_X$ on Ω_X random noise: $Z \sim P_Z$ on Ω_Z

Generator $G : \Omega_Z \mapsto \Omega_X$

Goal: Find G s.t. $G(Z) \overset{\mathcal{L}}{\approx} X$

Discriminator: $D : \Omega_X \mapsto [0, 1]$, $Y \sim \mathcal{B}(1/2)$

Goal: Find D s.t.

$$D(YX + (1 - Y)G(Z)) \approx \mathbb{P}(Y = 1 | YX + (1 - Y)G(Z))$$

Generator $G : \Omega_Z \mapsto \Omega_X$

data: $X \sim P_X$ generated data: $G(Z) \sim P_G$

Discriminator $G : \Omega_X \mapsto [0,1]$, $Y \sim \mathcal{B}(1/2)$

Log loss for D:

$$-\mathbb{E}\Big[Y \log(D(X)) + (1-Y)\log\Big(1 - D(G(Z))\Big)\Big].$$

Generator $G : \Omega_Z \mapsto \Omega_X$

data: $X \sim P_X$ generated data: $G(Z) \sim P_G$

Discriminator $G : \Omega_X \mapsto [0,1]$, $Y \sim \mathcal{B}(1/2)$

Log gain for D:

$$V(D,G) = \mathbb{E}\Big[Y \log(D(X)) + (1-Y)\log\Big(1 - D(G(Z))\Big)\Big] .$$

Minimax objective

$$\min_G \max_D V(D,G) .$$

Divergences

Kullback-Leibler divergence between P and Q

$$\mathrm{KL}(P, Q) = \begin{cases} \displaystyle\int_\Omega \ln\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right)\mathrm{d}P & \text{if } P \ll Q; \\ +\infty & \text{else.} \end{cases}$$

Jensen-Shannon divergence between P and Q

$$\mathrm{JS}(P, Q) = \frac{1}{2}\mathrm{KL}\left(P, \frac{P+Q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(Q, \frac{P+Q}{2}\right)$$

$$
\begin{aligned}
V(D,G) &= \mathbb{E}\Big[Y\log(D(X)) + (1-Y)\log\Big(1-D(G(Z))\Big)\Big] \\
&= \int_{\Omega_X} \frac{1}{2}\log\left(D(x)\right)\mathrm{d}P_X(x) + \frac{1}{2}\log\left(1-D(x)\right)\mathrm{d}P_G(x)\,\mathrm{d}x
\end{aligned}
$$

$$\begin{aligned} V(D,G) &= \mathbb{E}\Big[Y\log(D(X)) + (1-Y)\log\Big(1-D(G(Z))\Big)\Big] \\ &= \int_{\Omega_X} \frac{1}{2}\log\left(D(x)\right)\mathrm{d}P_X(x) + \frac{1}{2}\log\left(1-D(x)\right)\mathrm{d}P_G(x)\,\mathrm{d}x \end{aligned}$$

Optimal discriminator:

$$D_G^* = \arg\max_D V(D,G) = \frac{\mathrm{d}P_X}{\mathrm{d}P_X + \mathrm{d}P_G}$$

Objective at G^*:

$$V(D_G^*, G) = \mathrm{JS}(P_X, P_G) - \log(2)$$

$$V(D,G) = \mathbb{E}\Big[Y\log(D(X)) + (1-Y)\log\big(1-D(G(Z))\big)\Big]$$
$$= \int_{\Omega_X} \frac{1}{2}\log\,(D(x))\mathrm{d}P_X(x) + \frac{1}{2}\log\,(1-D(x))\mathrm{d}P_G(x)\,\mathrm{d}x$$

Optimal discriminator:

$$D_G^* = \arg\max_D V(D,G) = \frac{\mathrm{d}P_X}{\mathrm{d}P_X + \mathrm{d}P_G}$$

Objective at G^*:

$$V(D_G^*, G) = \mathrm{JS}(P_X, P_G) - \log(2)$$

Global minimum

$$G^* = \arg\min_G V(D_G^*, G) \quad \Leftrightarrow \quad P_X = P_G^*$$

$$C(G^*) = -\log(2)$$

$$V(D,G) = \mathbb{E}\Big[Y\log(D(X)) + (1-Y)\log\Big(1 - D(G(Z))\Big)\Big]$$
$$= \int_{\Omega_X} \frac{1}{2}\log\left(D(x)\right)\mathrm{d}P_X(x) + \frac{1}{2}\log\left(1 - D(x)\right)\mathrm{d}P_G(x)\,\mathrm{d}x$$

Optimal discriminator:

$$D^*_{G^*} = \arg\max_D V(D, G^*) = \frac{1}{2}$$

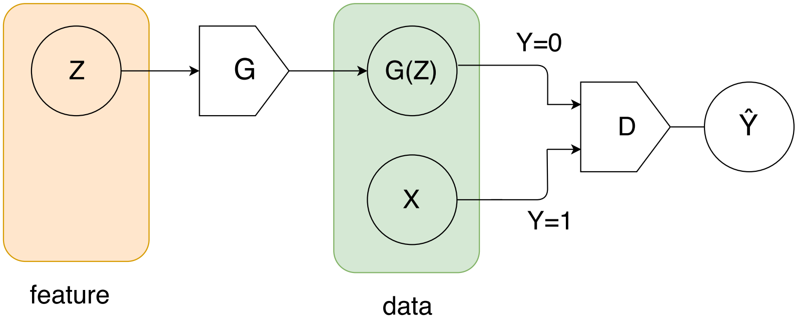
Objective at G^*:

$$V(D^*_G, G) = \mathrm{JS}(P_X, P_G) - \log(2)$$

Global minimum

$$G^* = \arg\min_G V(D^*_G, G) \quad \Leftrightarrow \quad P_X = P^*_G$$

$$C(G^*) = -\log(2)$$



How to learn the inverse mapping $X \mapsto Z$?

Bidirectional Generative Adversarial Network

Inputs data: $X \sim P_X$ random noise: $Z \sim P_Z$

Generator $G : \Omega_Z \mapsto \Omega_X$

Goal: Find G s.t. $G(Z) \overset{\mathcal{L}}{=} X$

Bidirectional Generative Adversarial Network

Inputs data: $X \sim P_X$ random noise: $Z \sim P_Z$

Generator $G : \Omega_Z \mapsto \Omega_X$

Goal: Find G s.t. $G(Z) \overset{\mathcal{L}}{=} X$

Encoder $E : \Omega_X \mapsto \Omega_Z$

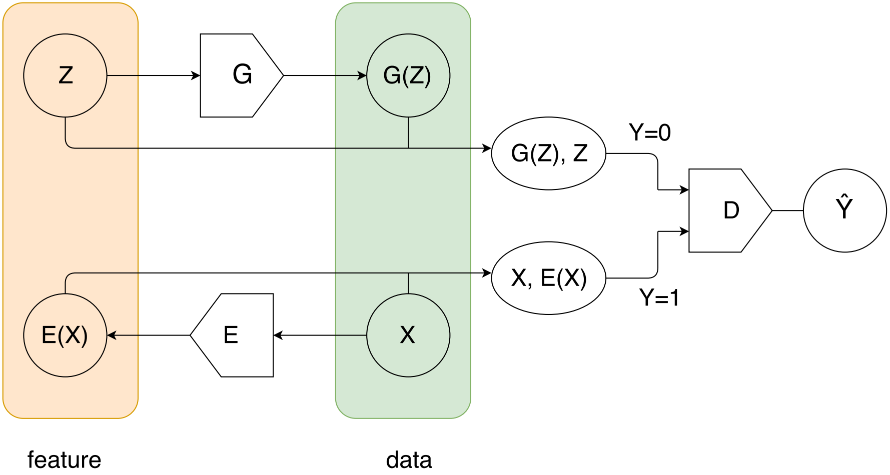
Goal: Find E s.t. $Z \overset{\mathcal{L}}{=} E(X)$

Discriminator: $D : \Omega_X \times \Omega_Z \mapsto [0, 1]$, $Y \sim \mathcal{B}(1/2)$

Goal: Find D s.t.

$$D\Big(Y(X, E(X)) + (1 - Y)(G(Z), Z)\Big) \approx \mathbb{P}\Big(Y = 1 | Y(X, E(X)) + (1 - Y)(G(Z), Z)\Big)$$

Bidirectional Generative Adversarial Network



Minimax objective

$$V(D, G, E) = \mathbb{E}\Big[Y \log\Big(D(X, E(X))\Big) + (1 - Y) \log\Big(1 - D(G(Z), Z)\Big)\Big]$$

Minimax objective: $\min_{(G,E)} \max_{D} \; V(D, G, E)$

Minimax objective

$$V(D,G,E)=\mathbb{E}\Big[Y\log\Big(D(X,E(X))\Big)+(1-Y)\log\Big(1-D(G(Z),Z)\Big)\Big]$$

Minimax objective: $\min_{(G,E)}\max_{D}\ V(D,G,E)$

$$((X,E(X))\sim P_{XE} \qquad\qquad (G(Z),Z)\sim P_{GZ}$$

Optimal discriminator: $D^*_{G,E}=\arg\max_D V(D,G,E)=\dfrac{\mathrm{d}P_{XE}}{\mathrm{d}P_{XE}+\mathrm{d}P_{GZ}}$

$$V(D^*_{G,E},G,E)=\mathrm{JS}(P_{XE},P_{GZ})-\log(2)$$

Minimax objective

$$V(D,G,E) = \mathbb{E}\Big[Y\log\Big(D(X,E(X))\Big) + (1-Y)\log\Big(1-D(G(Z),Z)\Big)\Big]$$

Minimax objective: $\min\limits_{(G,E)} \max\limits_{D} \; V(D,G,E)$

$$((X,E(X)) \sim P_{XE} \qquad\qquad (G(Z),Z) \sim P_{GZ}$$

Optimal discriminator: $D^*_{G,E} = \arg\max\limits_{D} V(D,G,E) = \dfrac{\mathrm{d}P_{XE}}{\mathrm{d}P_{XE}+\mathrm{d}P_{GZ}}$

$$V(D^*_{G,E},G,E) = \mathrm{JS}(P_{XE},P_{GZ}) - \log(2)$$

Global minimum:

$$(G^*,E^*) = \arg\min_{(G,E)} V(D^*_{G,E},G,E) \quad \Leftrightarrow \quad P_{XE^*} = P_{G^*Z}$$

Minimax objective

$$V(D, G, E) = \mathbb{E}\left[Y \log\Big(D(X, E(X))\Big) + (1 - Y) \log\Big(1 - D(G(Z), Z)\Big)\right]$$

Minimax objective: $\min\limits_{(G,E)} \max\limits_{D} \; V(D, G, E)$

$$((X, E(X)) \sim P_{XE} \qquad\qquad (G(Z), Z) \sim P_{GZ}$$

Optimal discriminator: $D^*_{G^*,E^*} = \arg\max\limits_{D} V(D, G^*, E^*) = \dfrac{1}{2}$

$$V(D^*_{G,E}, G, E) = \mathrm{JS}(P_{XE}, P_{GZ}) - \log(2)$$

Global minimum:

$$(G^*, E^*) = \arg\min_{(G,E)} V(D^*_{G,E}, G, E) \quad \Leftrightarrow \quad P_{XE^*} = P_{G^*Z}$$

Theorem

For (G, E) an optimal generator-encoder, then $G(E(x)) = x$ P_X a.s. on Ω_X and $E(G(z)) = z$ P_Z a.s. on Ω_Z.

Proof. $R_X = \left\{x \in \Omega_X \;:\; G(E(x)) \neq x\right\}$ We need to prove that $P_X(R_X) = 0$.

Theorem

For (G, E) an optimal generator-encoder, then $G(E(x)) = x$ P_X a.s. on Ω_X and $E(G(z)) = z$ P_Z a.s. on Ω_Z.

Proof. $R_X = \left\{x \in \Omega_X \ : \ G(E(x)) \neq x\right\}$ We need to prove that $P_X(R_X) = 0$.

$$R = \left\{(x, z) \in \Omega_X \times \Omega_Z \ : \ z = E(x) \wedge x \in R_X\right\}$$

$$\begin{aligned} P_X(R_X) &= \int_{\Omega_X} \mathbb{I}_{\{(x,E(x))\in R\}} \mathrm{d}P_X(x)\mathrm{d}x \\ &= P_{XE}(R) = P_{GZ}(R) \\ &= \int_{\Omega_Z} \mathbb{I}_{\{(G(z),z)\in R\}} \mathrm{d}P_{\Omega_Z}(z)\mathrm{d}z \\ &= \int_{\Omega_Z} \mathbb{I}_{\{z=E(G(z))\wedge G(E(G(z)))\neq G(z)\}} \mathrm{d}P_{\Omega_Z}(z)\mathrm{d}z = 0 \end{aligned}$$

Empirical evaluation

On **MNIST**

BiGAN	D	LR	JLR	AE (ℓ_2)	AE (ℓ_1)
97.39	97.30	97.44	97.13	97.58	97.63

Table 1: One Nearest Neighbors (1NN) classification accuracy (%) on the permutation-invariant MNIST (LeCun et al., 1998) test set in the feature space learned by BiGAN, Latent Regressor (LR), Joint Latent Regressor (JLR), and an autoencoder (AE) using an ℓ_1 or ℓ_2 distance.

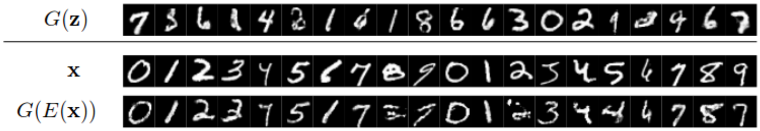
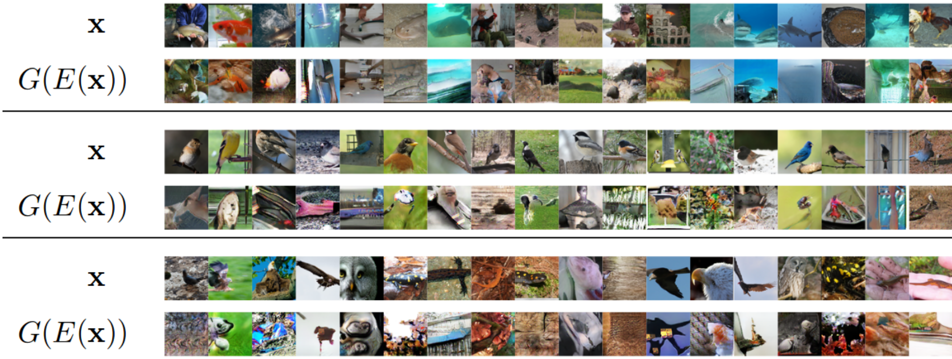


Figure 2: Qualitative results for permutation-invariant MNIST BiGAN training, including generator samples $G(\mathbf{z})$, real data $\mathbf{x}$, and corresponding reconstructions $G(E(\mathbf{x}))$.

On **ImageNet**



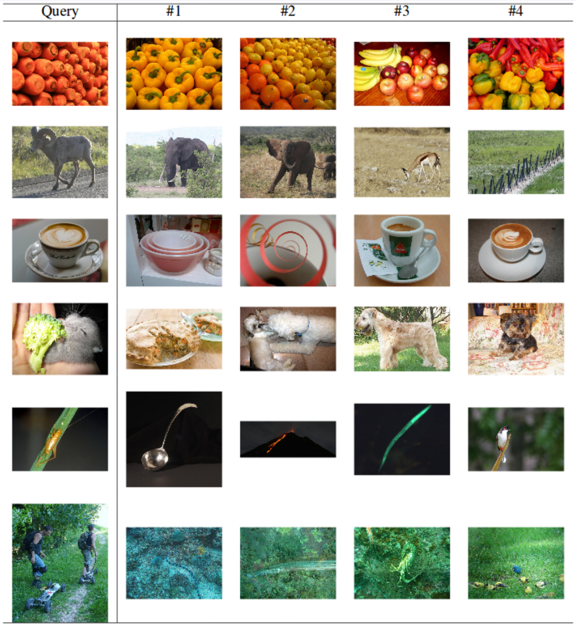


Figure 5: For the query images used in Krähenbühl et al. (2016) (left), nearest neighbors (by minimum cosine distance) from the ImageNet LSVRC (Russakovsky et al., 2015) training set in the *fc6* feature

Convolutional filters learned

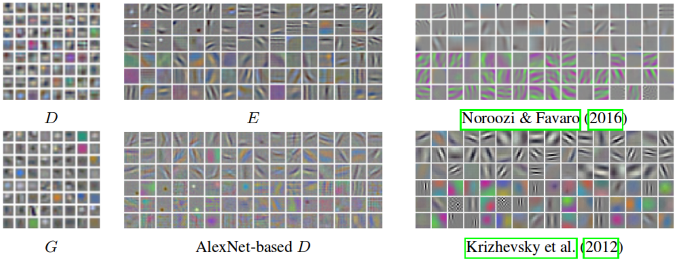
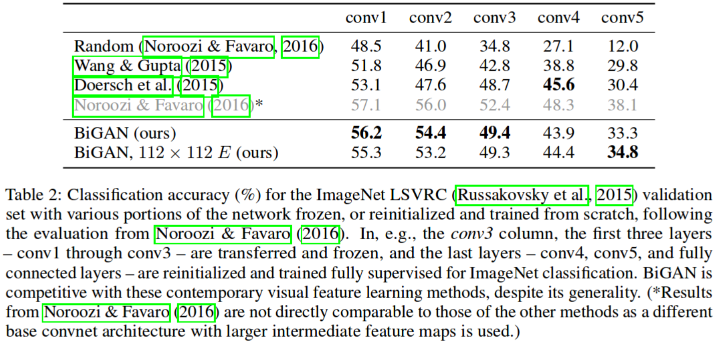


Figure 3: The convolutional filters learned by the three modules (D, G, and E) of a BiGAN (left, top-middle) trained on the ImageNet (Russakovsky et al., 2015) database. We compare with the filters learned by a discriminator D trained with the same architecture (bottom-middle), as well as the filters reported by Noroozi & Favaro (2016), and by Krizhevsky et al. (2012) for fully supervised ImageNet training (right).

ImageNet classification



	conv1	conv2	conv3	conv4	conv5
Random (Noroozi & Favaro, 2016)	48.5	41.0	34.8	27.1	12.0
Wang & Gupta (2015)	51.8	46.9	42.8	38.8	29.8
Doersch et al. (2015)	53.1	47.6	48.7	**45.6**	30.4
Noroozi & Favaro (2016)*	57.1	56.0	52.4	48.3	38.1
BiGAN (ours)	**56.2**	**54.4**	**49.4**	43.9	33.3
BiGAN, 112 × 112 *E* (ours)	55.3	53.2	49.3	44.4	**34.8**

Table 2: Classification accuracy (%) for the ImageNet LSVRC (Russakovsky et al., 2015) validation set with various portions of the network frozen, or reinitialized and trained from scratch, following the evaluation from Noroozi & Favaro (2016). In, e.g., the *conv3* column, the first three layers – conv1 through conv3 – are transferred and frozen, and the last layers – conv4, conv5, and fully connected layers – are reinitialized and trained fully supervised for ImageNet classification. BiGAN is competitive with these contemporary visual feature learning methods, despite its generality. (*Results from Noroozi & Favaro (2016) are not directly comparable to those of the other methods as a different base convnet architecture with larger intermediate feature maps is used.)

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles

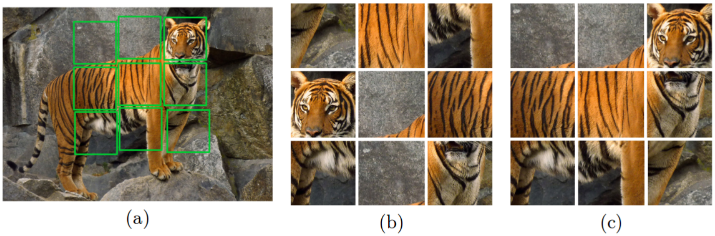


Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].