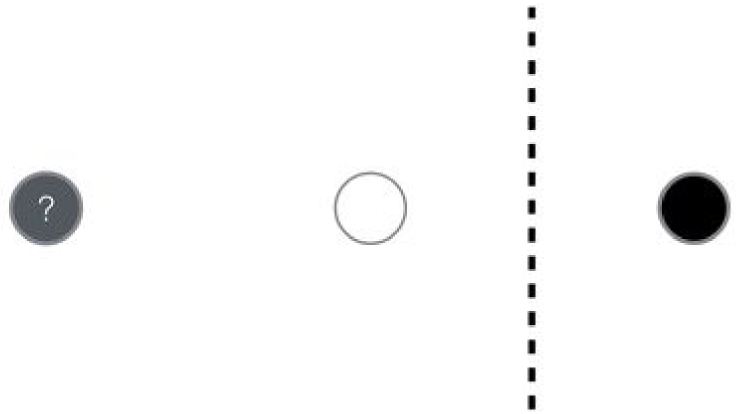
Semi-Supervised Learning

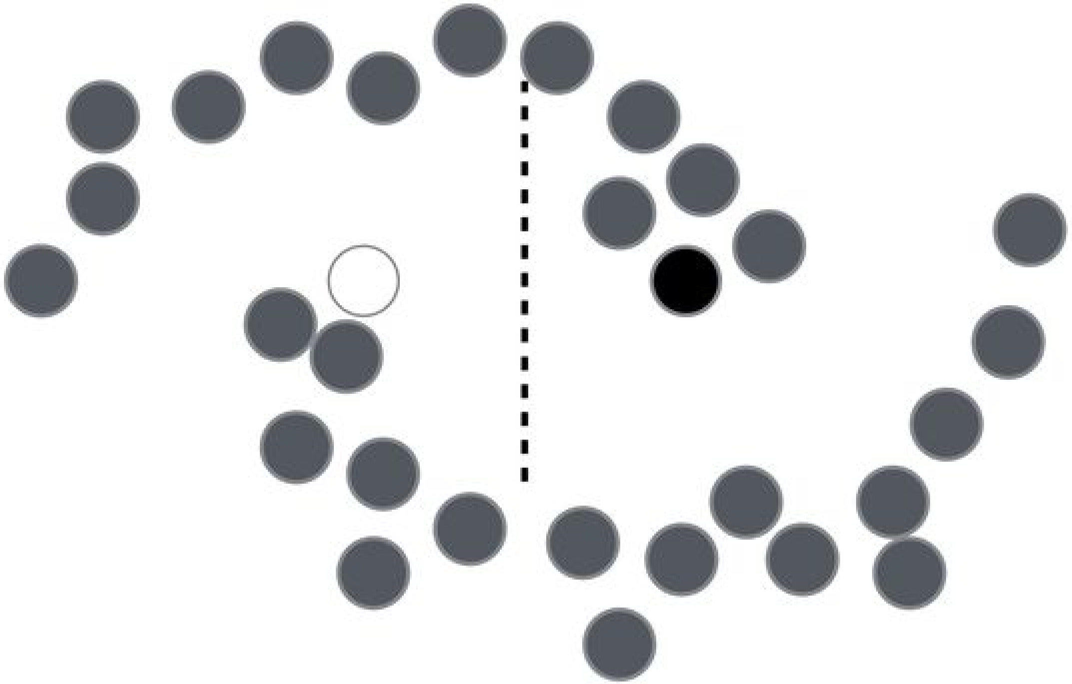
with Ladder Networks

Гущенко-Чеверда Иван, 141

Задача Semi-Supervised Learning



Задача Semi-Supervised Learning



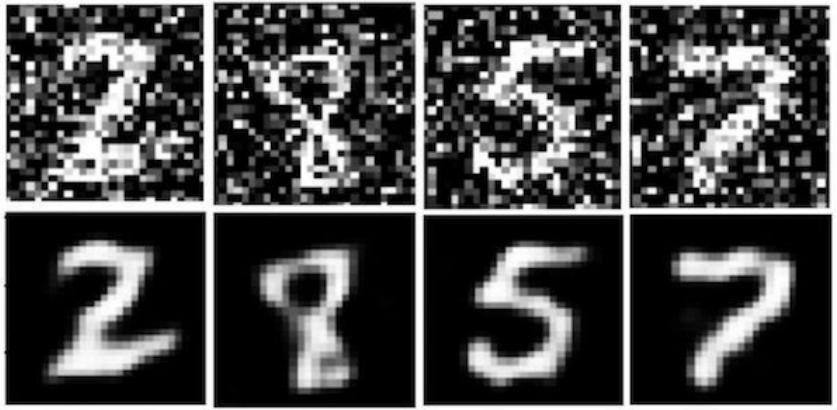
Как работать со сложными данными?

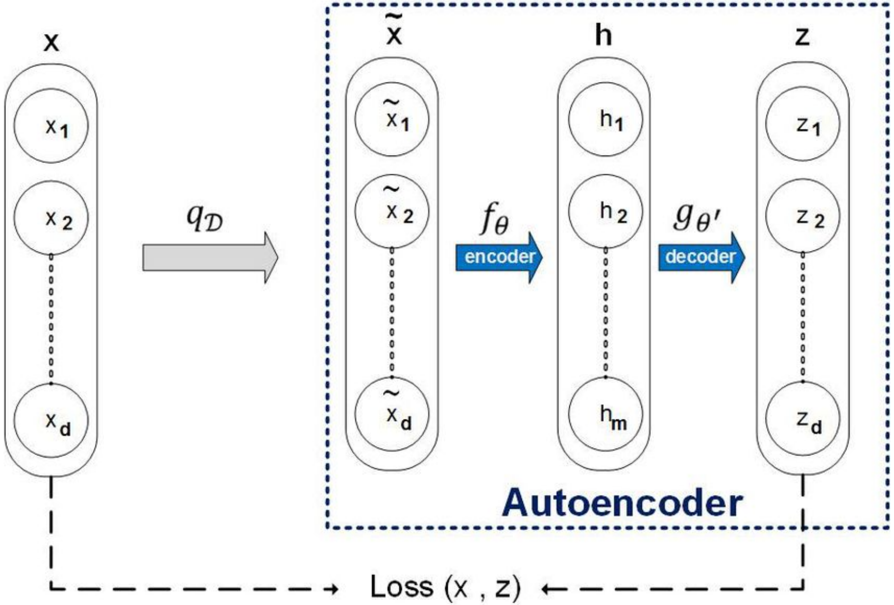
Предложенный метод позволит адаптировать нейронные сети для извлечения пользы из неразмеченных данных на ряду с размеченными.

Свойства:

- Не имеет ограничение на глубину базовой модели.
- Адаптируется под разные архитектуры.
- Масштабируется. Время итерации увеличивается в константу раз.

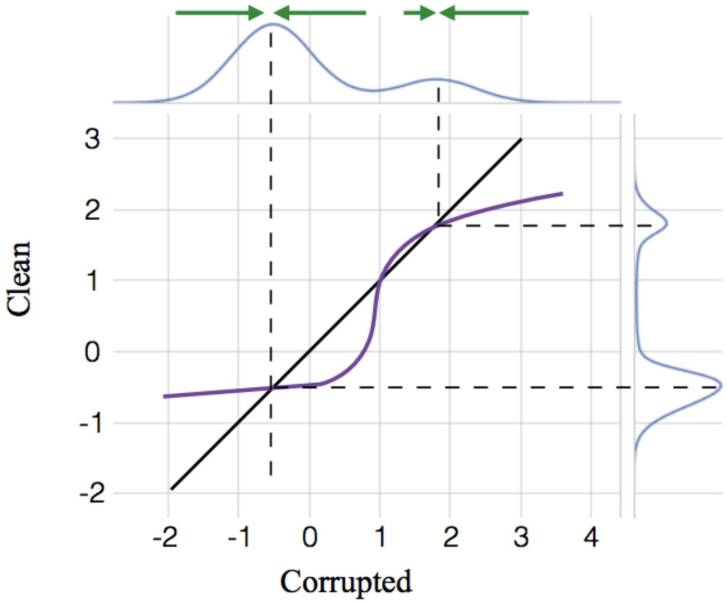
Denoising autoencoder





Denoising source separation

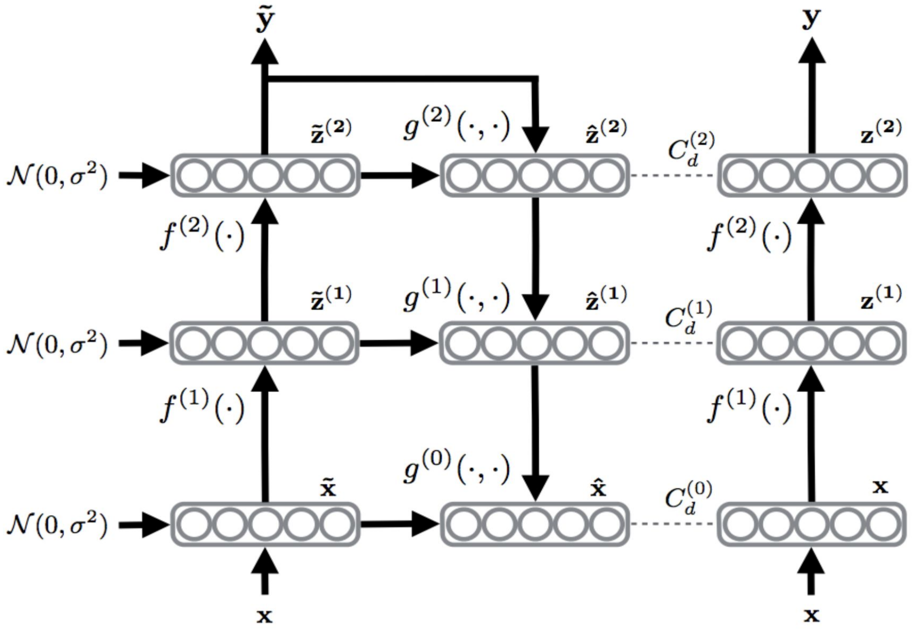
Вместо восстановления входов алгоритм восстанавливает $\hat{\mathbf{z}} = g(\tilde{\mathbf{z}})$, где $\mathbf{z} = f(\mathbf{x})$. Минимизируем $\|\hat{\mathbf{z}} - \mathbf{z}\|^2$.



Архитектура Ladder Network

- Noisy encoder
- Clean encoder
- Decoder

Архитектура Ladder Network



Noisy encoder

\# Corrupted encoder and classifier

$\tilde{\mathbf{h}}^{(0)} \leftarrow \tilde{\mathbf{z}}^{(0)} \leftarrow \mathbf{x}(n) + \texttt{noise}$

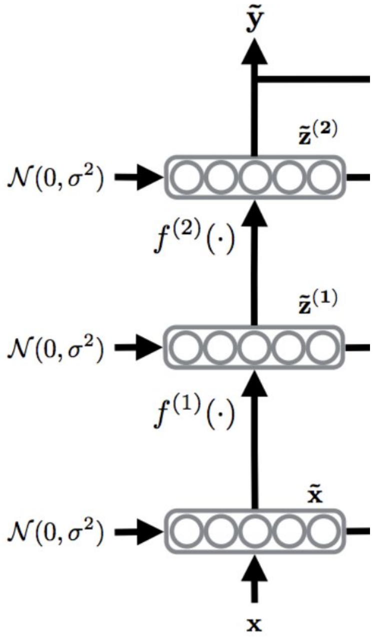
for l = 1 **to** L **do**

$\quad \tilde{\mathbf{z}}^{(l)} \leftarrow \texttt{batchnorm}(\mathbf{W}^{(l)}\tilde{\mathbf{h}}^{(l-1)}) + \texttt{noise}$

$\quad \tilde{\mathbf{h}}^{(l)} \leftarrow \texttt{activation}(\boldsymbol{\gamma}^{(l)} \odot (\tilde{\mathbf{z}}^{(l)} + \boldsymbol{\beta}^{(l)}))$

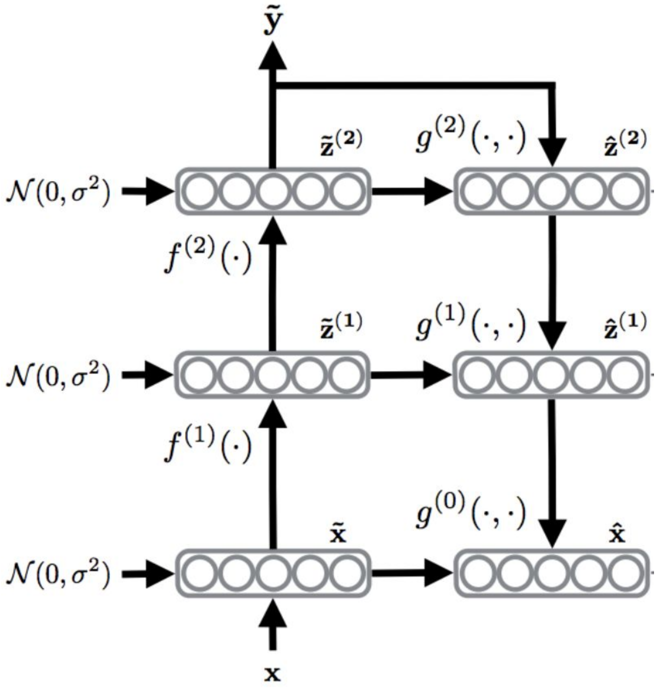
end for

$P(\tilde{\mathbf{y}} \mid \mathbf{x}) \leftarrow \tilde{\mathbf{h}}^{(L)}$



Decoder and loss

for l = L **to** 0 **do**
  if l = L **then**
    $\mathbf{u}^{(L)} \leftarrow \texttt{batchnorm}(\tilde{\mathbf{h}}^{(L)})$
  else
    $\mathbf{u}^{(l)} \leftarrow \texttt{batchnorm}(\mathbf{V}^{(l+1)} \hat{\mathbf{z}}^{(l+1)})$
  end if
  $\forall i : \hat{z}_i^{(l)} \leftarrow g(\tilde{z}_i^{(l)}, u_i^{(l)})$ # Eq. (2)
  $\forall i : \hat{z}_{i,\mathrm{BN}}^{(l)} \leftarrow \frac{\hat{z}_i^{(l)} - \mu_i^{(l)}}{\sigma_i^{(l)}}$
end for
# Cost function C for training:
$\mathrm{C} \leftarrow 0$
if $t(n)$ **then**
  $\mathrm{C} \leftarrow -\log P(\tilde{\mathbf{y}} = t(n) \mid \mathbf{x}(n))$
end if
$\mathrm{C} \leftarrow \mathrm{C} + \sum_{l=0}^{L} \lambda_l \left\| \mathbf{z}^{(l)} - \hat{\mathbf{z}}_{\mathrm{BN}}^{(l)} \right\|^2$ # Eq. (3)



Выбор функции g

Идеальный denoising для гауссовской случайной величины:

$$\hat{z} = g(\hat{z}) = v * \tilde{z} + (1 - v) * \mu = (\tilde{z} - \mu) * v + \mu$$

Мы хотим, чтобы мы могли провести идеальный denoising для

$$p(\mathbf{z}^{(l)} \mid \mathbf{z}^{(l+1)}) = \prod_i p(z_i^{(l)} \mid \mathbf{z}^{(l+1)})$$

распределенных нормально.

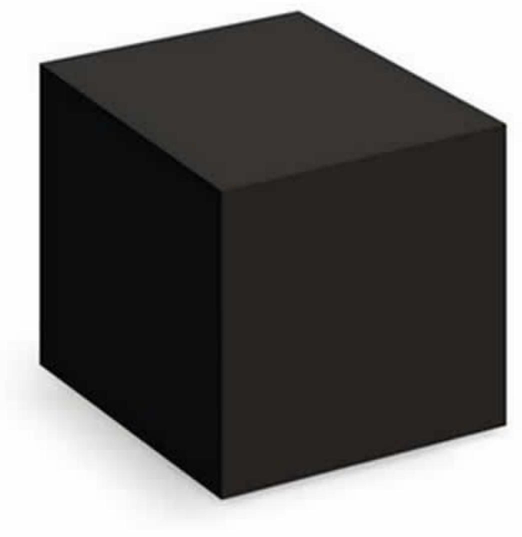
Выбор функции g

$$\hat{z}_i^{(l)} = g_i(\tilde{z}_i^{(l)}, u_i^{(l)}) = \left(\tilde{z}_i^{(l)} - \mu_i(u_i^{(l)})\right) \upsilon_i(u_i^{(l)}) + \mu_i(u_i^{(l)})$$

$$\mu_i(u_i^{(l)}) = a_{1,i}^{(l)} \texttt{sigmoid}(a_{2,i}^{(l)} u_i^{(l)} + a_{3,i}^{(l)}) + a_{4,i}^{(l)} u_i^{(l)} + a_{5,i}^{(l)}$$

$$\upsilon_i(u_i^{(l)}) = a_{6,i}^{(l)} \texttt{sigmoid}(a_{7,i}^{(l)} u_i^{(l)} + a_{8,i}^{(l)}) + a_{9,i}^{(l)} u_i^{(l)} + a_{10,i}^{(l)},$$

Почему алгоритм работает?



Permutation invariant MNIST

Test error % with # of used labels	100	1000	All
Semi-sup. Embedding (Weston *et al.*, 2012)	16.86	5.73	1.5
Transductive SVM (from Weston *et al.*, 2012)	16.81	5.38	1.40*
MTC (Rifai *et al.*, 2011)	12.03	3.64	0.81
Pseudo-label (Lee, 2013)	10.49	3.46	
AtlasRBF (Pitelis *et al.*, 2014)	8.10 ($\pm$ 0.95)	3.68 ($\pm$ 0.12)	1.31
DGN (Kingma *et al.*, 2014)	3.33 ($\pm$ 0.14)	2.40 ($\pm$ 0.02)	0.96
DBM, Dropout (Srivastava *et al.*, 2014)			0.79
Adversarial (Goodfellow *et al.*, 2015)			0.78
Virtual Adversarial (Miyato *et al.*, 2015)	2.12	1.32	0.64 ($\pm$ 0.03)
Baseline: MLP, BN, Gaussian noise	21.74 ($\pm$ 1.77)	5.70 ($\pm$ 0.20)	0.80 ($\pm$ 0.03)
Γ-model (Ladder with only top-level cost)	3.06 ($\pm$ 1.44)	1.53 ($\pm$ 0.10)	0.78 ($\pm$ 0.03)
Ladder, only bottom-level cost	1.09 ($\pm$0.32)	0.90 ($\pm$ 0.05)	0.59 ($\pm$ 0.03)
Ladder, full	**1.06** ($\pm$ 0.37)	**0.84** ($\pm$ 0.08)	**0.57** ($\pm$ 0.02)

MNIST

Table 2: CNN results for MNIST

Test error without data augmentation % with # of used labels	100	all
EmbedCNN (Weston *et al.*, 2012)	7.75	
SWWAE (Zhao *et al.*, 2015)	9.17	0.71
Baseline: Conv-Small, supervised only	6.43 ($\pm$ 0.84)	0.36
Conv-FC	0.99 ($\pm$ 0.15)	
Conv-Small, Γ-model	**0.89** ($\pm$ 0.50)	

CIFAR10

Table 3: Test results for CNN on CIFAR-10 dataset without data augmentation

Test error % with # of used labels	4 000	All
All-Convolutional ConvPool-CNN-C (Springenberg *et al.*, 2014)		9.31
Spike-and-Slab Sparse Coding (Goodfellow *et al.*, 2012)	31.9	
Baseline: Conv-Large, supervised only	23.33 ($\pm$ 0.61)	9.27
Conv-Large, Γ-model	**20.40** ($\pm$ 0.47)	

Дополнительно: Вариации алгоритма

	100		1000		60000	
Variant	**AER (%)**	**SE**	**AER (%)**	**SE**	**AER (%)**	**SE**
Gaussian	1.064	± 0.021	0.983	± 0.019	0.604	± 0.010
GatedGauss	1.308	± 0.038	1.094	± 0.016	0.632	± 0.011
MLP [4]	1.374	± 0.186	0.996	± 0.028	0.605	± 0.012
MLP [2, 2]	1.209	± 0.116	1.059	± 0.023	0.573	± 0.016
MLP [2, 2, 2]	1.274	± 0.067	1.095	± 0.053	0.602	± 0.010
AMLP [4]	1.072	± 0.015	**0.974**	± 0.021	0.598	± 0.014
AMLP [2, 2]	1.193	± 0.039	1.029	± 0.023	**0.569**	± 0.010
AMLP [2, 2, 2]	**1.002**	± 0.038	0.979	± 0.025	0.578	± 0.013

Ссылки

- Semi-Supervised Learning with Ladder Networks(2015) (https://arxiv.org/abs/1507.02672)
- Deconstructing the Ladder Network Architecture(2016) (http://proceedings.mlr.press/v48/pezeshki16.pdf)