Глубокое обучение

Обучение без учителя

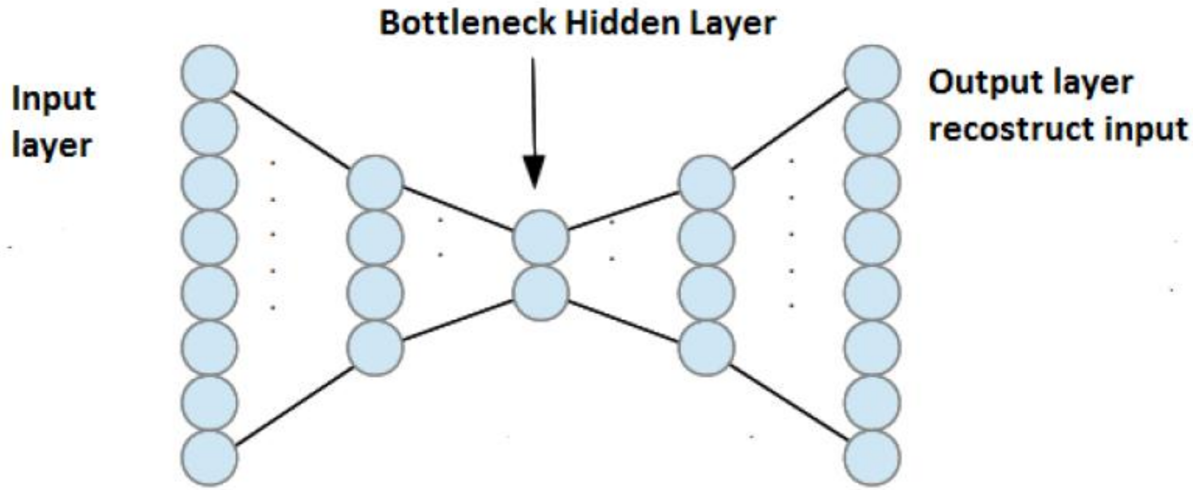
Дьяконов А.Г.

Московский государственный университет
имени М.В. Ломоносова (Москва, Россия)

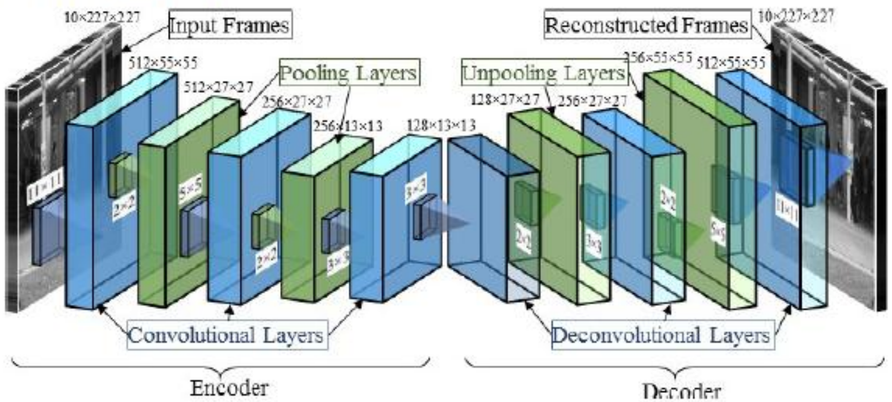
<table>
<tr><th colspan="4">Обучение без учителя
Unsupervised Learning</th></tr>
<tr><th>Невероятностные модели</th><th colspan="3">Вероятностные (генеративные) модели</th></tr>
<tr><td rowspan="3">• Sparse Coding +
• Autoencoders
• k-means, ...</td><th colspan="2">Плотность в явном виде (explicit density)</th><th>Плотность в неявном виде (implicit density)</th></tr>
<tr><th>Tractable Models</th><th>Non-Tractable Models</th><td rowspan="2">• GAN
• Momet Matching Networks</td></tr>
<tr><td>• Fully observed Belief Nets
• NADE
• PixelRNN</td><td>• Boltzmann Machines ~
• VAE
• Helmholtz Machines</td></tr>
</table>

Автокодировщики (Auto-encoders)

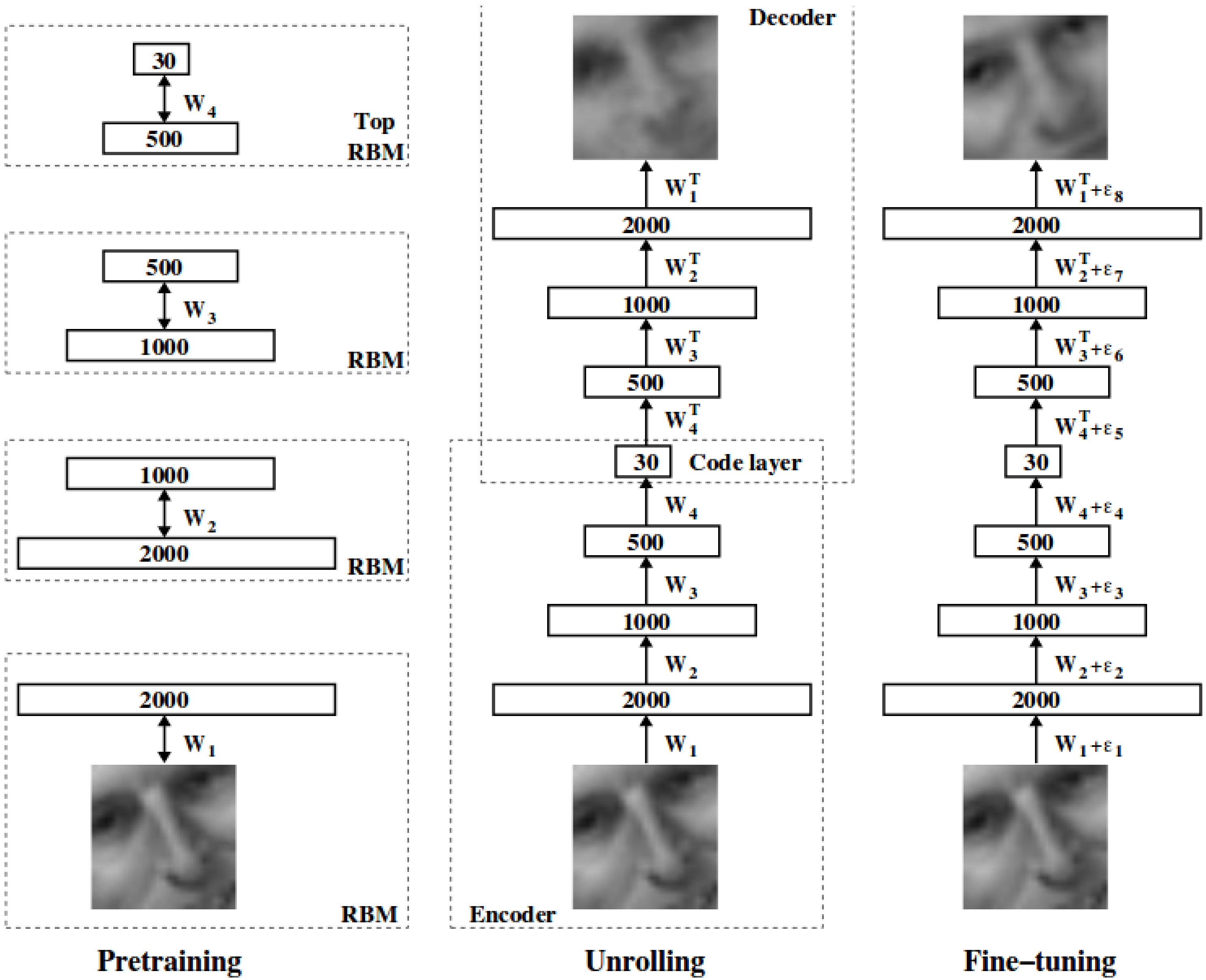
- Fully Contented Auto-encoder



- Convolution Auto-encoder



Глубокие автокодировщики



Зачем нужны автокодировщики

- сокращение размерности
- выделение признаков (для других алгоритмов)
- предобучение
- специальные задачи (ниже: устранение шума)

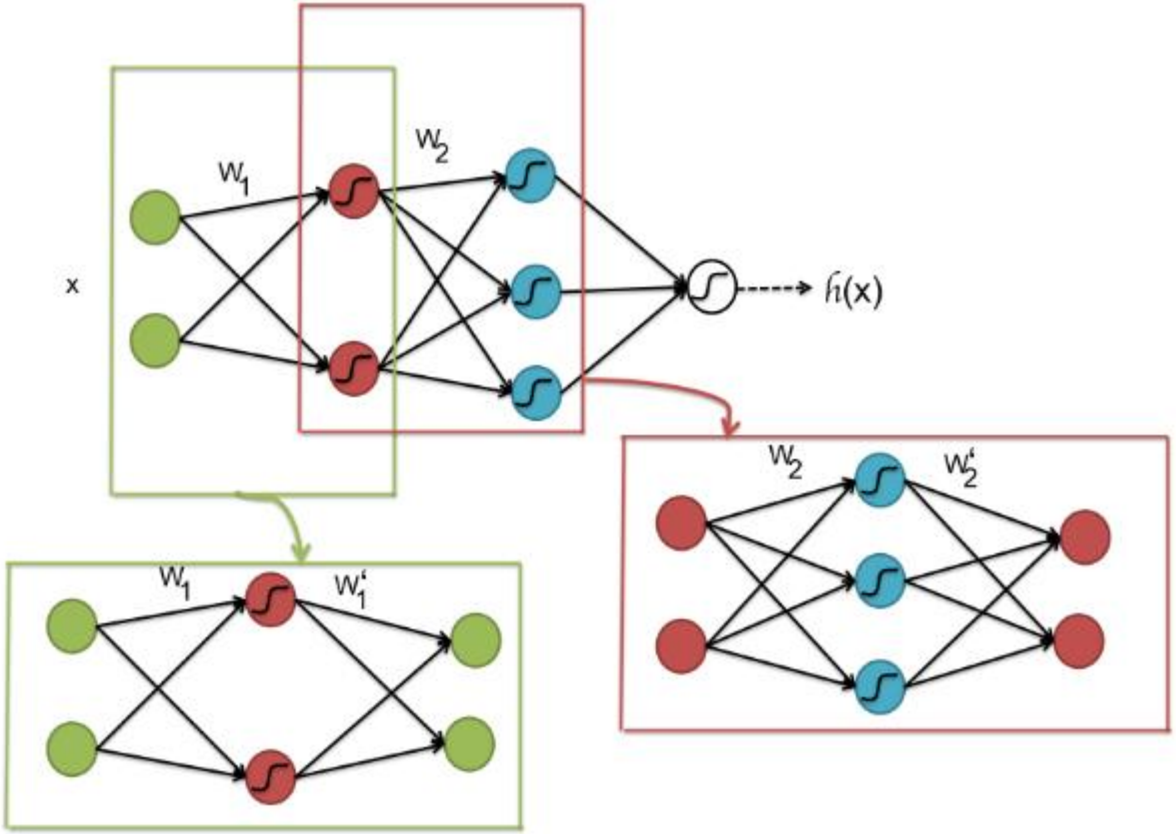
Denoising Autoencoder

увеличение выборки с помощью
дополнения к ней зашумлённых изображений
зашумления разного типа

- больше данных
- правильнее формируемые признаки

Предобучение с помощью автокодировщика

послойно обучить автокодировщики



первый слой должен воспроизводить вход

второй слой должен воспроизводить первый и т.д.

обучить последний слой, используя размеченные данные

обучить всю сеть, используя размеченные данные

https://cs.stanford.edu/~quocle/tutorial2.pdf

Sparse Coding

Постановка задачи. Выборка $\{x_i\}_{i=1}^{m}$, базисы $\{b_i\}_{i=1}^{k}$, хотим представить в виде разреженной линейной комбинации (sparse linear combination)

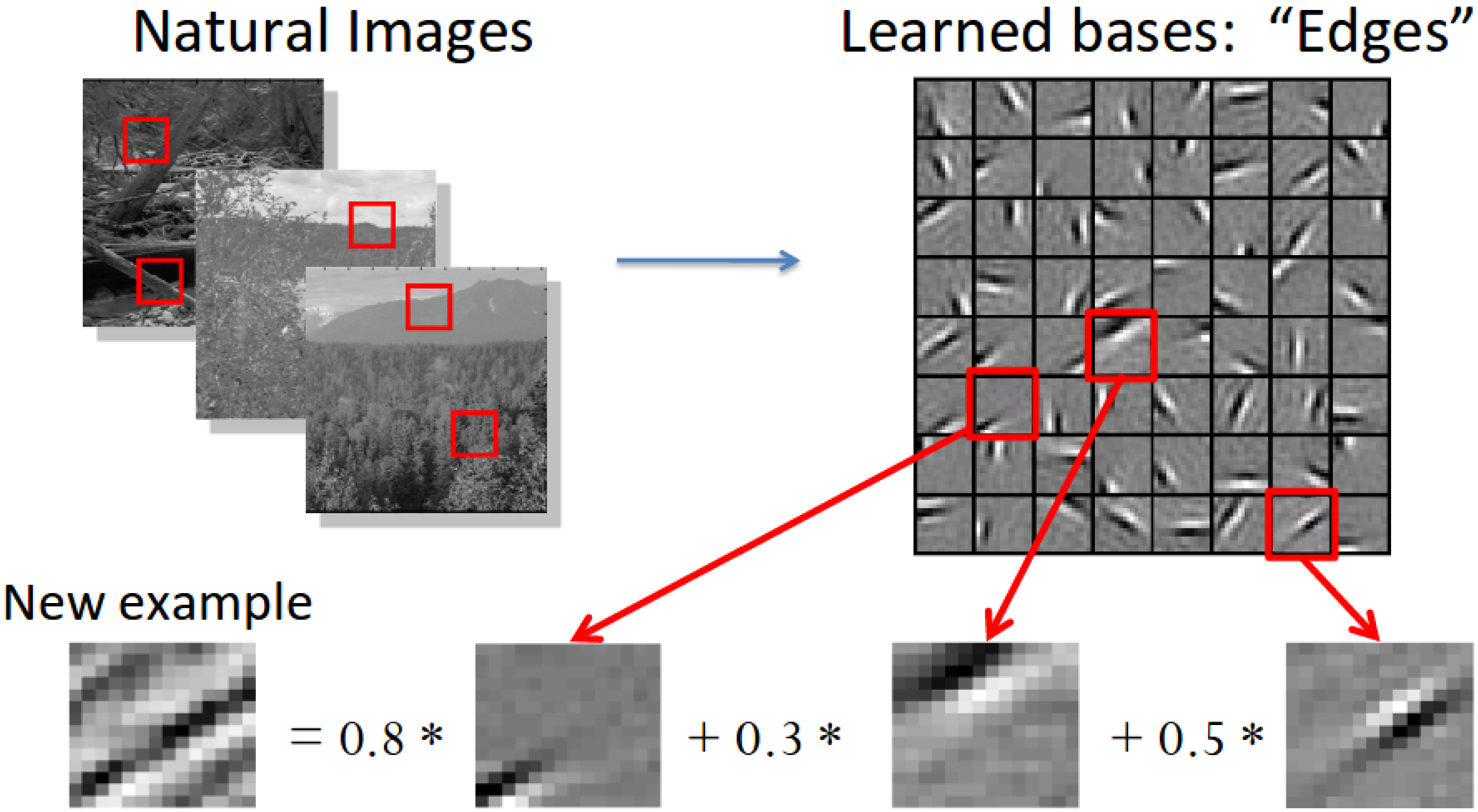
$$x_i \approx \sum_{j=1}^{k} \alpha_{ij} b_j :$$

в основном все α_{ij} нулевые!

$$\sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} \alpha_{ij} b_j \right\|_2^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |\alpha_{ij}| \to \min_{\alpha, b}$$

Alternative optimization – попеременно фиксировать базис и коэффициенты

Sparse Coding

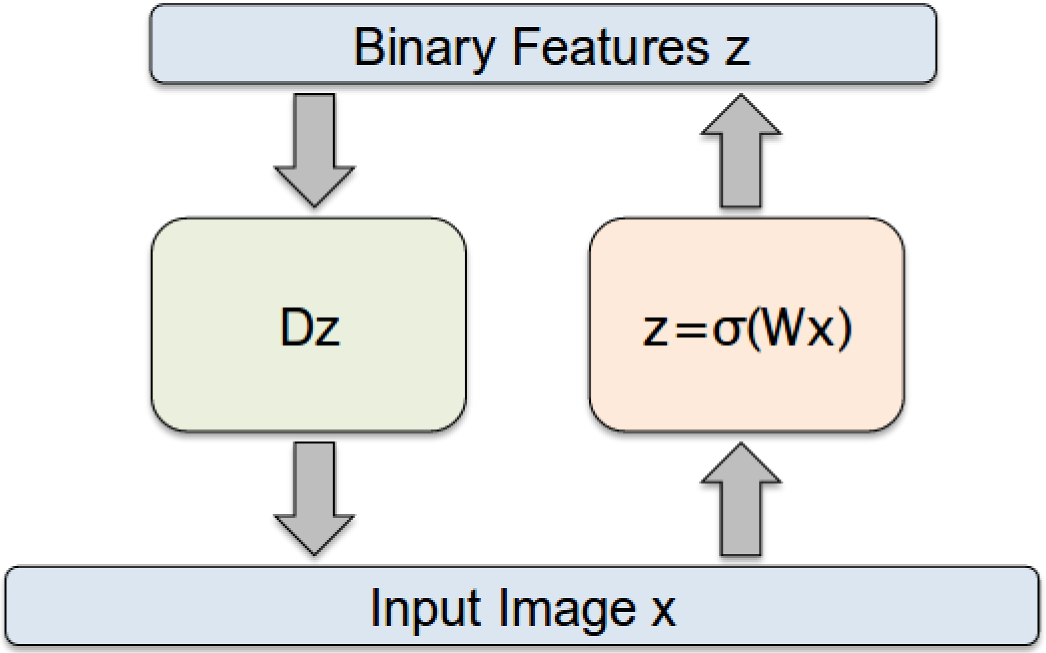


$$x = 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{65}$$

Sparse Coding

вписывается в парадигму автокодировщика!

Kavukcuoglu, Ranzato, Fergus, LeCun, 2009



k скрытых бинарных нейронов

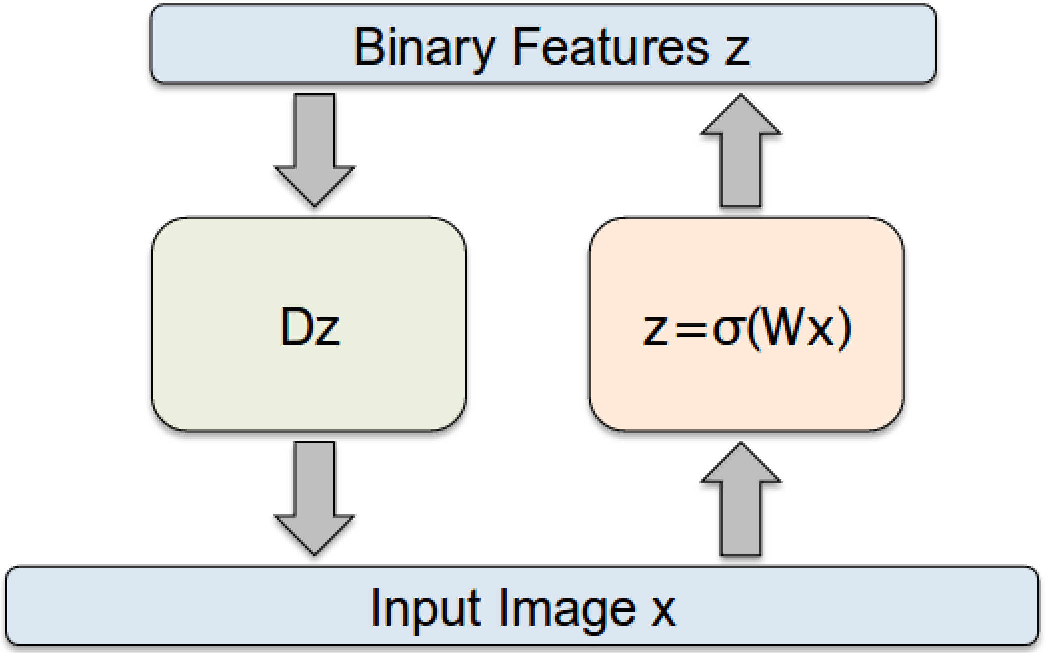
$$X_j \approx \sum_{t=1}^{k} D_{jt} \sigma\left(\sum_{i=1}^{n} W_{ti} X_i \right)$$

+ L1 регуляризация на значения $z = \sigma(Wx)$

Sparse Coding

вписывается в парадигму автокодировщика!

Kavukcuoglu, Ranzato, Fergus, LeCun, 2009



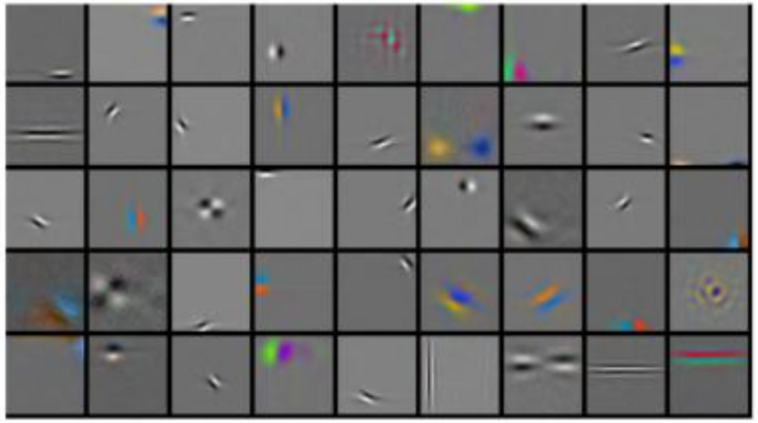
k скрытых бинарных нейронов

$$\| Dz - x \|_2^2 + \lambda \| z \|_1 + \| \sigma(Wx) - z \|_2^2 \to \min_{D,W,z}$$

Использование RBM

4 million **unlabelled** images

Learned features (out of 10,000)



REUTERS

Associated Press

Reuters dataset:
804,414 **unlabeled**
newswire stories
Bag-of-Words

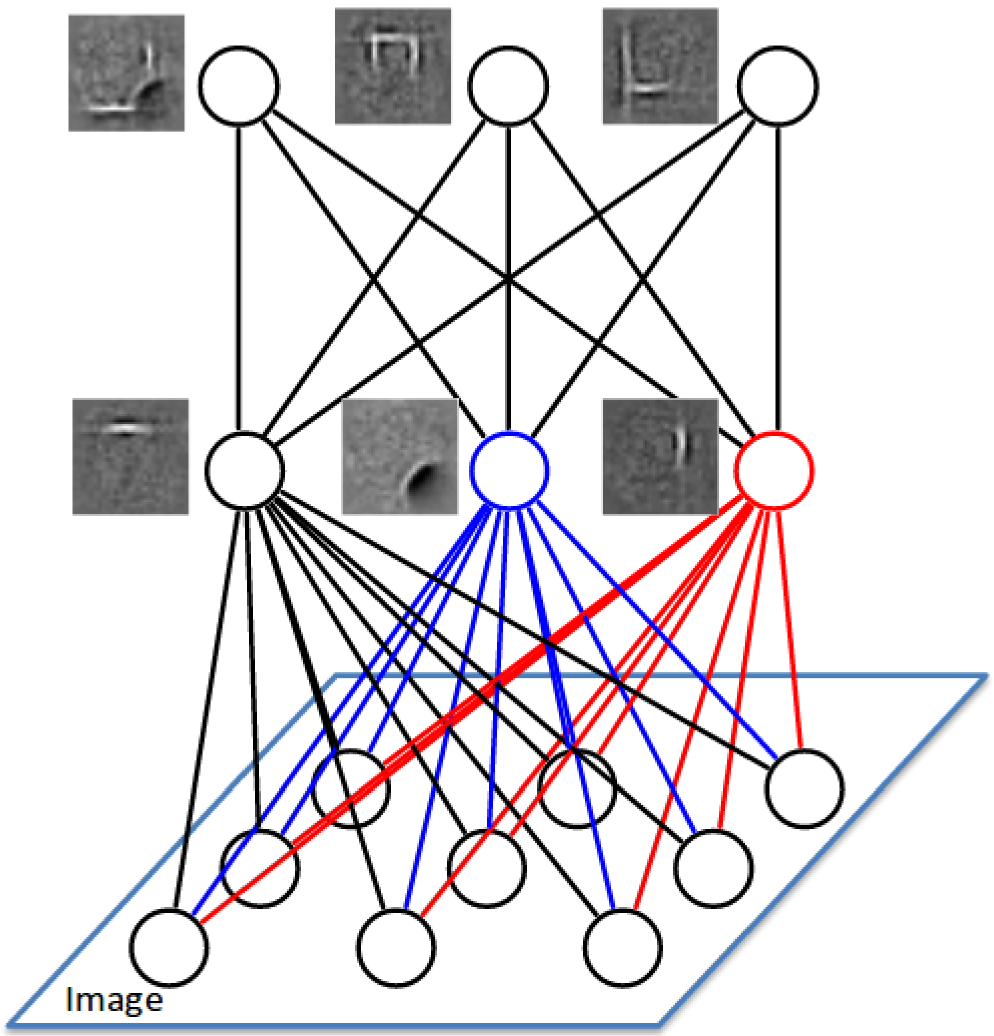
Learned features: ``topics''

russian	clinton	computer	trade	stock
russia	house	system	country	wall
moscow	president	product	import	street
yeltsin	bill	software	world	point
soviet	congress	develop	economy	dow

Использование RBM

$$P_\theta(v,h)=\frac{1}{Z(\theta)}\exp\left(\sum_{ij}W_{ij}v_i h_j+\sum_i b_i v_i+\sum_j a_j h_j\right)$$

Глубокие RBM (Deep Boltzmann Machines)



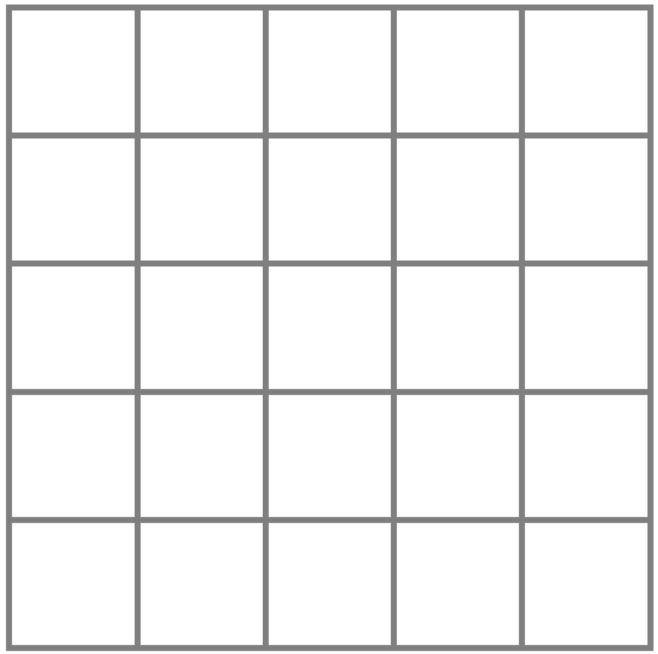
Многоуровневое пространство скрытых переменных (разные степени детализации)

Salakhutdinov, Hinton, 2009

SOM – Самоорганизующиеся карты Кохонена

Хотим отобразить пространство в регулярную структуру (например, узлы решётки решётки)

Решётка задаёт соседство узлов (топологию)



Самоорганизующиеся карты Кохонена

Алгоритм обучения

1. Подаём объект x

2. Находим самый близкий узел

$$(a,b)=\arg\max_{(i,j)}\mathrm{sim}(x,w_{ij})$$

соревнование ~ нейрон-победитель, пример $\mathrm{sim}(x,w_{ij})=-\|x-w_{ij}\|$

3. Коррекция весов

$$w_{ij}\leftarrow w_{ij}+\eta K((a,b),(i,j))x$$

веса нейронов около победителя корректируются больше...

~ кооперация, пример $K((a,b),(i,j))=\exp(-|a-i|+|b-j|)$

$$(a,b)=\arg\max_{(i,j)}\mathrm{sim}(x,w_{ij})$$

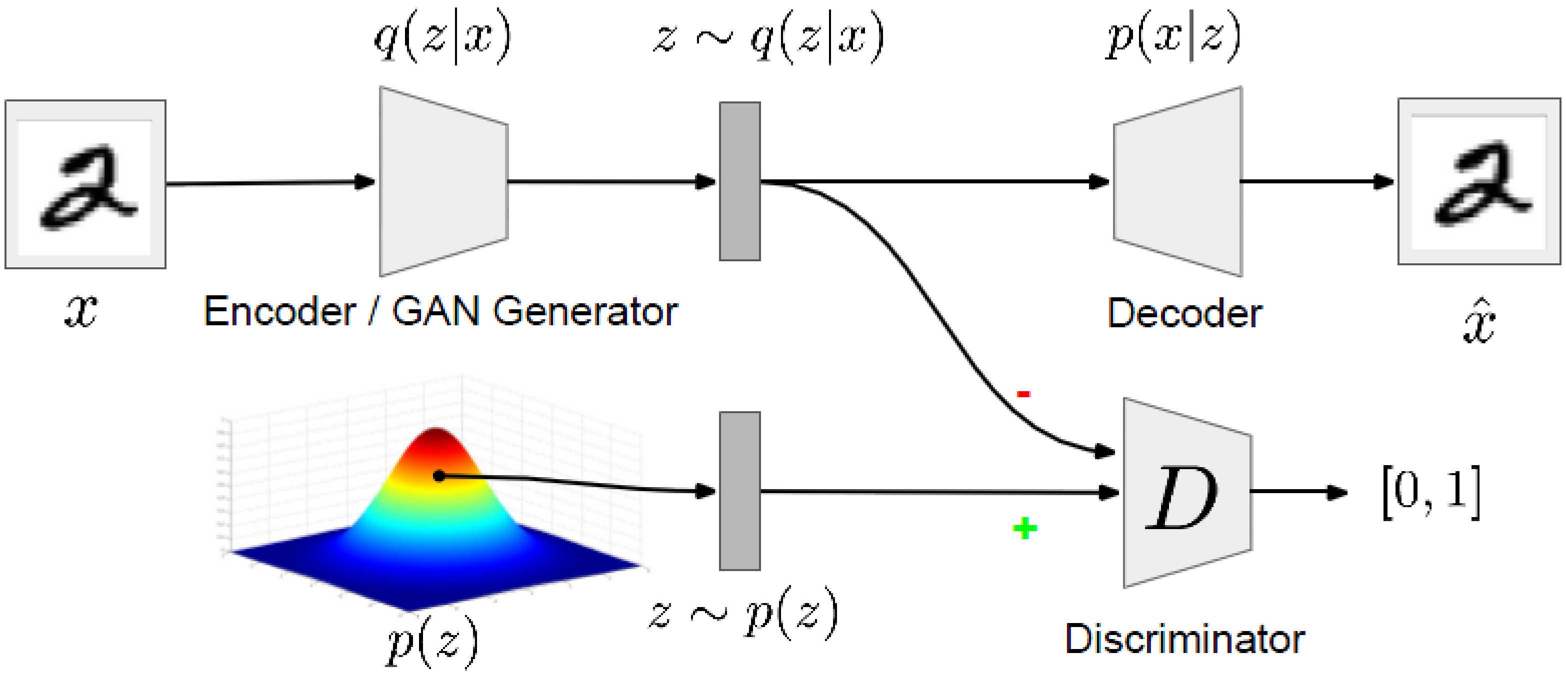
Регуляризация в автокодировщиках

- + шум (в Denoising Autoencoders)
- + специальное слагаемое к функции ошибки (в Sparse Autoencoders)
- + специальное слагаемое к функции ошибки для стабильности латентного представления (используем Якобиан в Contractive Autoencoders)

Adversarial Autoencoders (AAE)

Цель: наделить структурой пространство скрытых переменных

+ adversarial loss для соответствия распределения ПСП произвольному априорному



Отличие AAE от VAE

$$\mathcal{L} = \mathbb{E}_x \underbrace{\left[\mathbb{E}_{q(z|x)}[-\log p(x|z)]\right]}_{\text{Reconstruction Error}} + \mathbb{E}_x \underbrace{\left[\mathrm{KL}(q(z|x)||p(z))\right]}_{\text{KL Regularizer}}$$

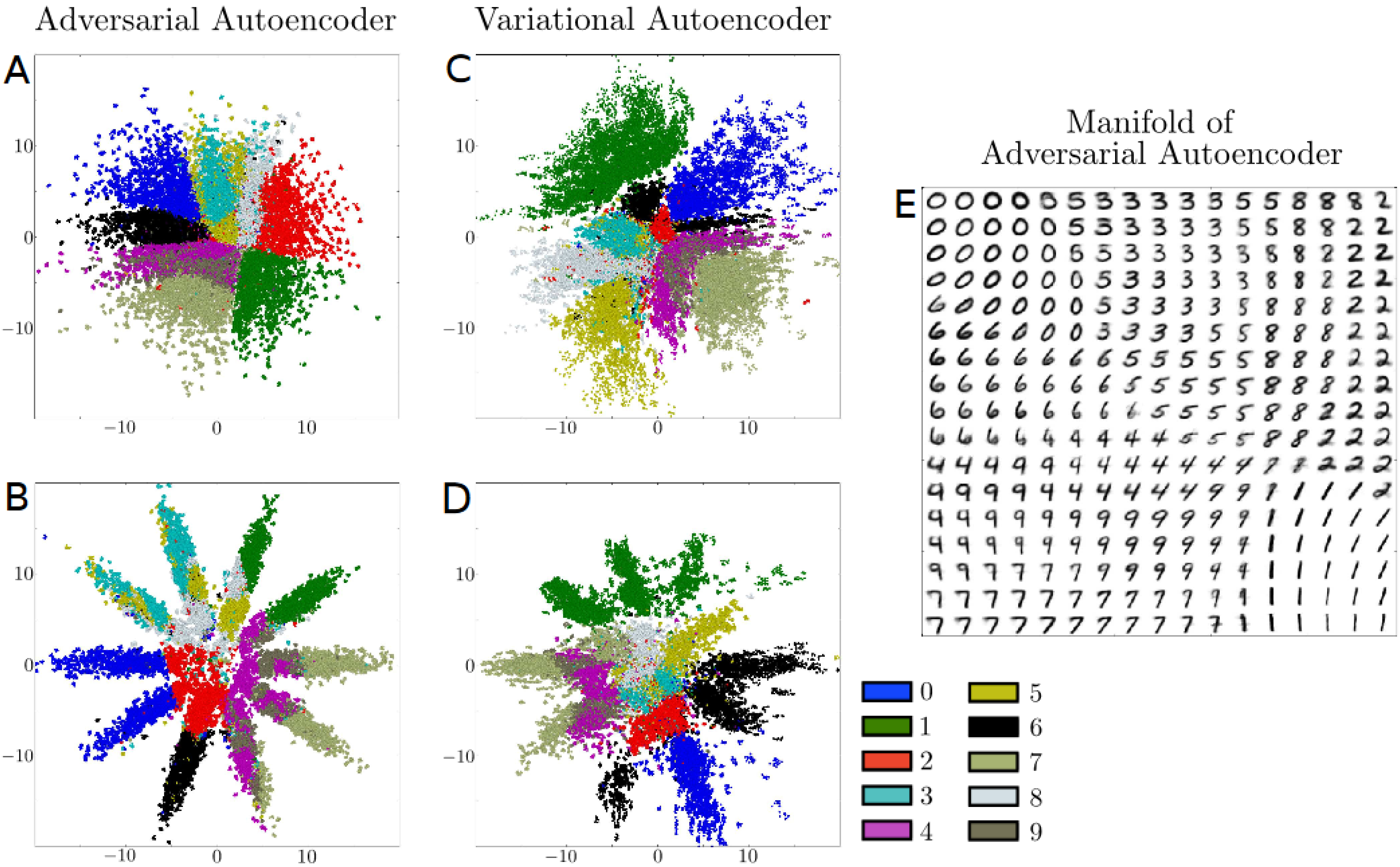
Replaced by adversarial loss in AAE

в VAE мы обратное распространение через KL ⇒ должна быть аналитическая запись p(z)

в AAE достаточно сэмплировать из априорного распределения

ниже (см. рис.) разные априорные распределения: нормальное в 2D и смесь нормальных

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey «Adversarial Autoencoders» // https://arxiv.org/abs/1511.05644


Adversarial Autoencoder
Variational Autoencoder
Manifold of
Adversarial Autoencoder
A
B
C
D
E
10
0
−10
0 1 2 3 4 5 6 7 8 9

Wasserstein Auto-Encoders (доклад на ICLR 2018) – обобщение AAE

$$D_{\mathrm{WAE}}(P_X, P_G) := \inf_{Q(Z|X)\in\mathcal{Q}} \underbrace{\mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\big[c\big(X, G(Z)\big)\big]}_{\textbf{Reconstruction}\text{ cost}} + \lambda \cdot \underbrace{\mathcal{D}_Z(Q_Z, P_Z)}_{\substack{\textbf{Regularizer}\text{ encourages the}\\ \text{encoded distribution to match}\\ \text{the prior}}}$$

VAE&GAN – минимизируют меру различия распределения данных и модели

WAE – «регуляризованное» расстояние Вассерштейна между модельным и целевым распределением

WAE = AAE при $c(x,y)=\| x-y \|_2^2$

WAE использует не обязательно состязательную (adversarial) меру D_Z

Расстояние Вассерштейна-1

$$W(p, p') = \inf E_{(x,y)\sim\Pi(p,p')} \| x - y \|$$

где $\Pi(p, p')$ – множество всех совместных распределений с соответвующими проекциями

Arjovsky et al 2017

Эквивалентно:

$$W(p, p') = \sup_{\|f\|_L \le 1} \mathbf{E}_{x\sim p} f(x) - \mathbf{E}_{x\sim p'} f(x)$$

Sup по липшицевым функциям с константой Липшица $\le$ 1

На практике f аппроксимируется с помощью НС, где все веса клиппируются, чтобы лежать на компактном пространстве (например на гиперкубе со стороной ε)