Лекция 4. Карты Кохонена, автокодировщики, перенос обучения, генеративно-состязательные сети

Александр Юрьевич Авдюшенко

МКН СПбГУ

10 марта 2022

Факультет
математики
и компьютерных
наук
СПбГУ

Пятиминутка

- Перечислите недостатки сверточных нейронных сетей
- Выпишите формулу простейшей (vanilla) рекуррентной сети
- Какие фильтры (gates) есть в LSTM?

Постановка задачи кластеризации

напоминание

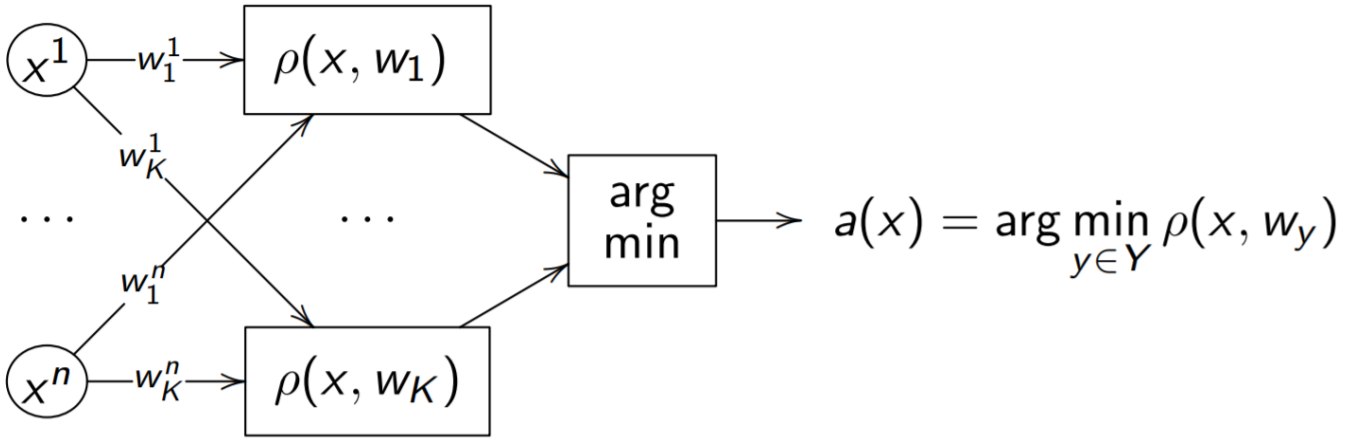
Дано:
$X^\ell = \{x_1, \ldots, x_\ell\}$ — обучающая выборка объектов $x_i \in \mathbb{R}^n$
$\rho : X \times X \to [0, \infty)$ — функция расстояния между объектами
Найти:
Y — множество кластеров, например, задаваемых своими центрами $w_y \in \mathbb{R}^n$
Пусть алгоритм кластеризации

$$a(x) = \arg\min_{y \in Y} \rho(x, w_y)$$

«Правило жёсткой конкуренции» (WTA, Winner Takes All)
Критерий: среднее внутрикластерное расстояние

$$Q(w; X^\ell) = \sum_{i=1}^{\ell} \rho^2(x_i, w_{a(x_i)}) \to \min_{w_y : y \in Y}$$

Сеть Кохонена — двухслойная нейронная сеть



Градиентный шаг в методе стохастического градиента:

$$w_y = w_y + \eta(x_i - w_y)[a(x_i) = y]$$

Если x_i относится к кластеру y, то w_y сдвигается в сторону x_i

---

T. Kohonen. Self-organized formation of topologically correct feature maps. 1982.

Стохастический градиентный спуск

Вход: выборка X^ℓ, темп обучения η, параметр λ
Выход: центры кластеров $w_1, \ldots, w_K \in \mathbb{R}^n$

1. инициализировать центры w_y, $y \in Y$
2. оценку функционала:

$$Q = \sum_{i=1}^{\ell} \rho^2(x_i, w_{a(x_i)})$$

3. **повторять**
   - выбрать объект x_i из X^ℓ (например, случайный)
   - вычислить кластер: $y = \arg\min_{y \in Y} \rho(x_i, w_y)$
   - сделать градиентный шаг: $w_y = w_y + \eta(x_i - w_y)$
   - оценить функционал: $Q = \lambda\rho^2(x_i, w_y) + (1 - \lambda)Q$
4. **пока** значение Q и/или веса w_y не сойдутся

Жесткая и мягкая конкуренция

Правило жёсткой конкуренции (WTA, Winner Takes All)

$$w_y = w_y + \eta(x_i - w_y)[a(x_i) = y], y \in Y$$

Недостатки правила WTA

- медленная скорость сходимости
- некоторые центры кластеров могут никогда не выбираться

Правило мягкой конкуренции (WTM, Winner Takes Most)

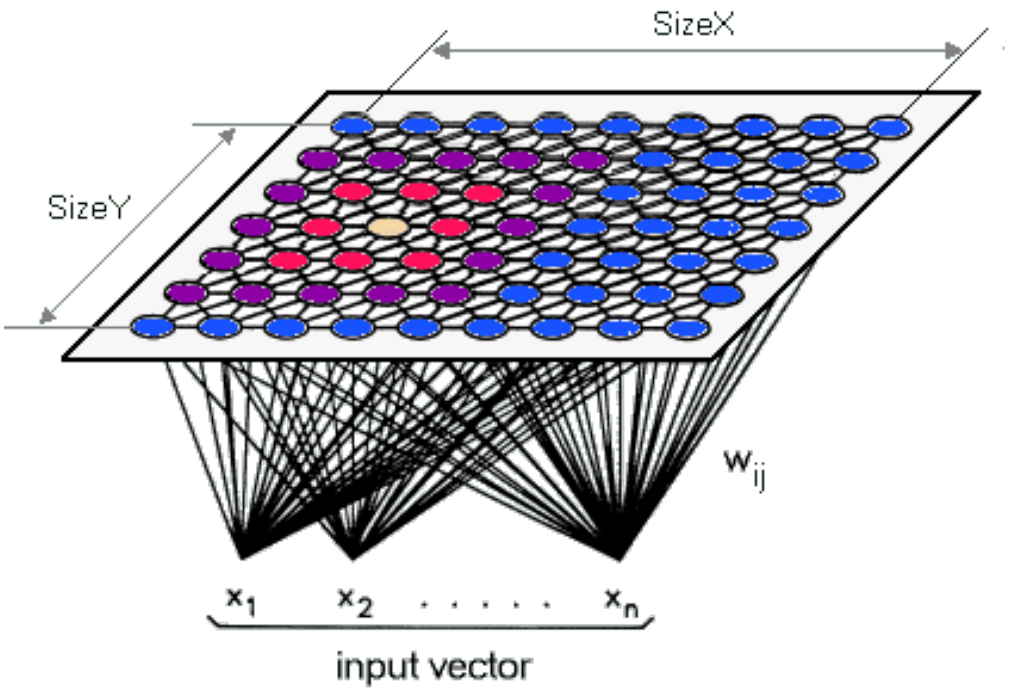
$$w_y = w_y + \eta(x_i - w_y)K(\rho(x_i, w_y)), y \in Y$$

где ядро $K(\rho)$ — неотрицательная невозрастающая функция
Теперь центры всех кластеров смещаются в сторону x_i, но чем дальше от x_i, тем меньше величина смещения

Карта Кохонена (Self Organizing Map, SOM)

Вводим прямоугольную сетку кластеров
$\{1, \ldots, SizeX\} \times \{1, \ldots, SizeY\}$
Каждому узлу (x, y) приписан нейрон Кохонена $w_{xy} \in \mathbb{R}^n$
Наряду с метрикой $\rho(x_i, w_{xy})$ вводится метрика на сетке:

$$r((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$$



Обучение карты Кохонена

Вход: выборка X^ℓ, темп обучения η
Выход: $w_{xy} \in \mathbb{R}^n$ — векторы весов,

1. инициализировать веса: $w_{xy} = \text{random}\left(-\frac{1}{2MH}, \frac{1}{2MH}\right)$
2. **повторять**
   - выбрать случайный объект x_i из X^ℓ
   - WTA: вычислить координаты кластера:

$$(a_i, b_i) = \arg\min_{(a,b)} \rho(x_i, w_{ab})$$

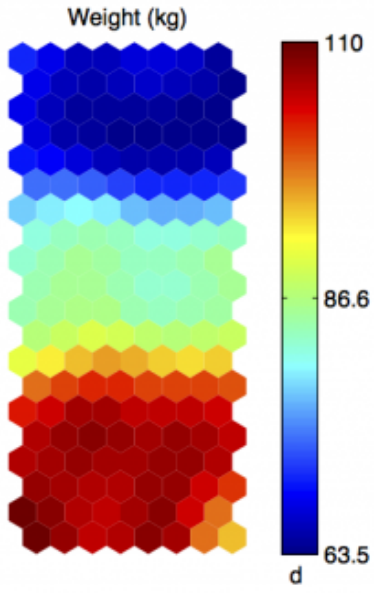
   - **для всех** $(a, b) \in$ Окрестность (a_i, b_i)
     WTM: сделать шаг градиентного спуска:
     $w_{ab} = w_{ab} + \eta(x_i - w_{ab})K(r((a_i, b_i), (a, b)))$
3. **пока** кластеризация не стабилизируется

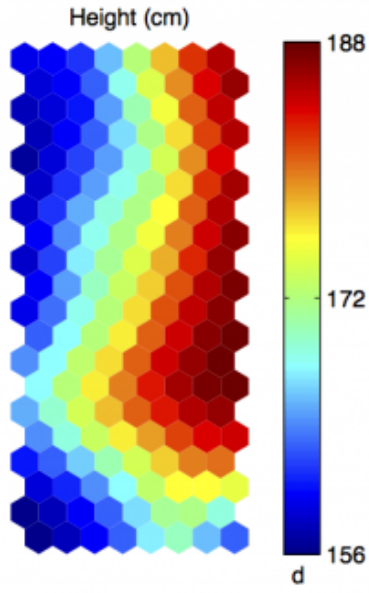
Интерпретация карт Кохонена

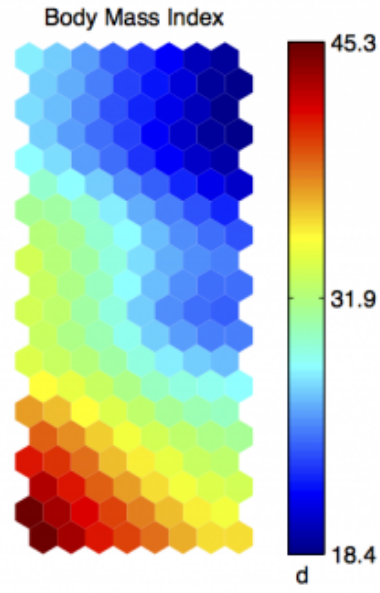
Два типа графиков — цветных карт $SizeX \times SizeY$

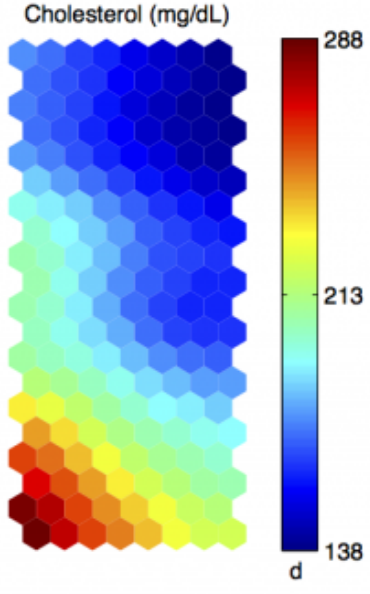
- Цвет узла (a, b) — локальная плотность в точке (a, b) — среднее расстояние до k ближайших точек выборки
- По одной карте на каждый признак: цвет узла (a, b) — значение j-й компоненты вектора w_{ab}

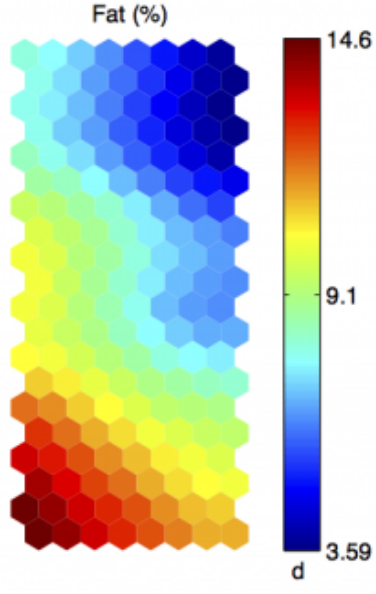
Посмотрим на карты Кохонена, построенные на шести признаках, собранных у 1000 человек.

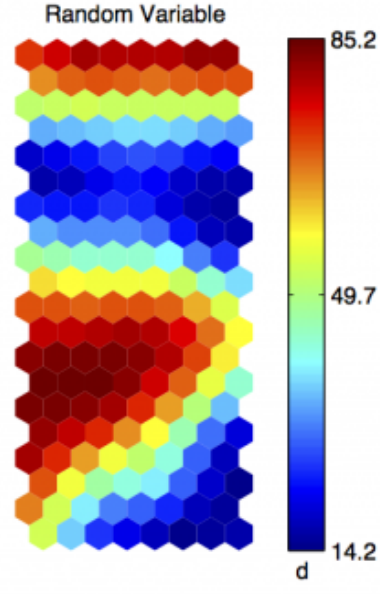

Weight (kg)
110
86.6
63.5
d


Height (cm)
188
172
156
d


Body Mass Index
45.3
31.9
18.4
d


Cholesterol (mg/dL)
288
213
138
d


Fat (%)
14.6
9.1
3.59
d


Random Variable
85.2
49.7
14.2
d

Достоинства и недостатки карт Кохонена

$+$ возможность визуального анализа многомерных данных

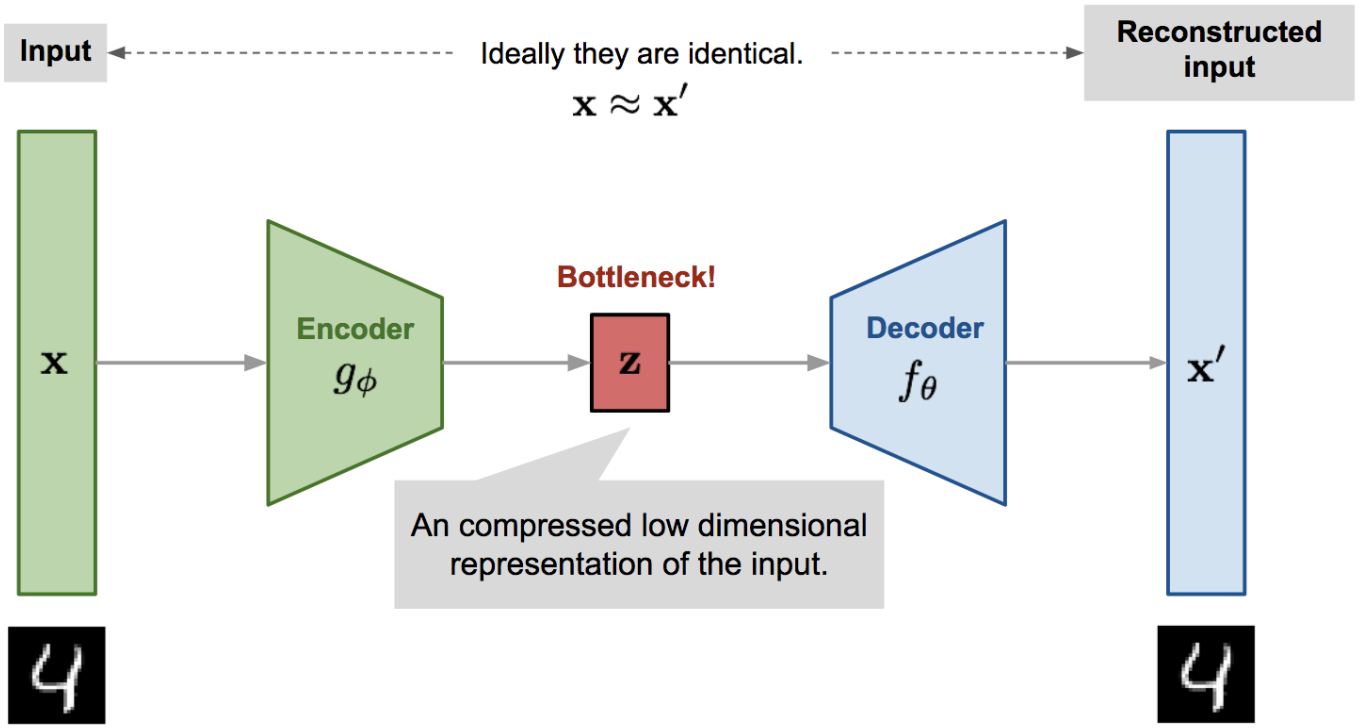
$-$ **Искажения**. Близкие в исходном пространстве могут перейти в далёкие точки на карте, и наоборот.

$-$ **Субъективность**. Карта зависит не только от кластерной структуры данных, но и от...

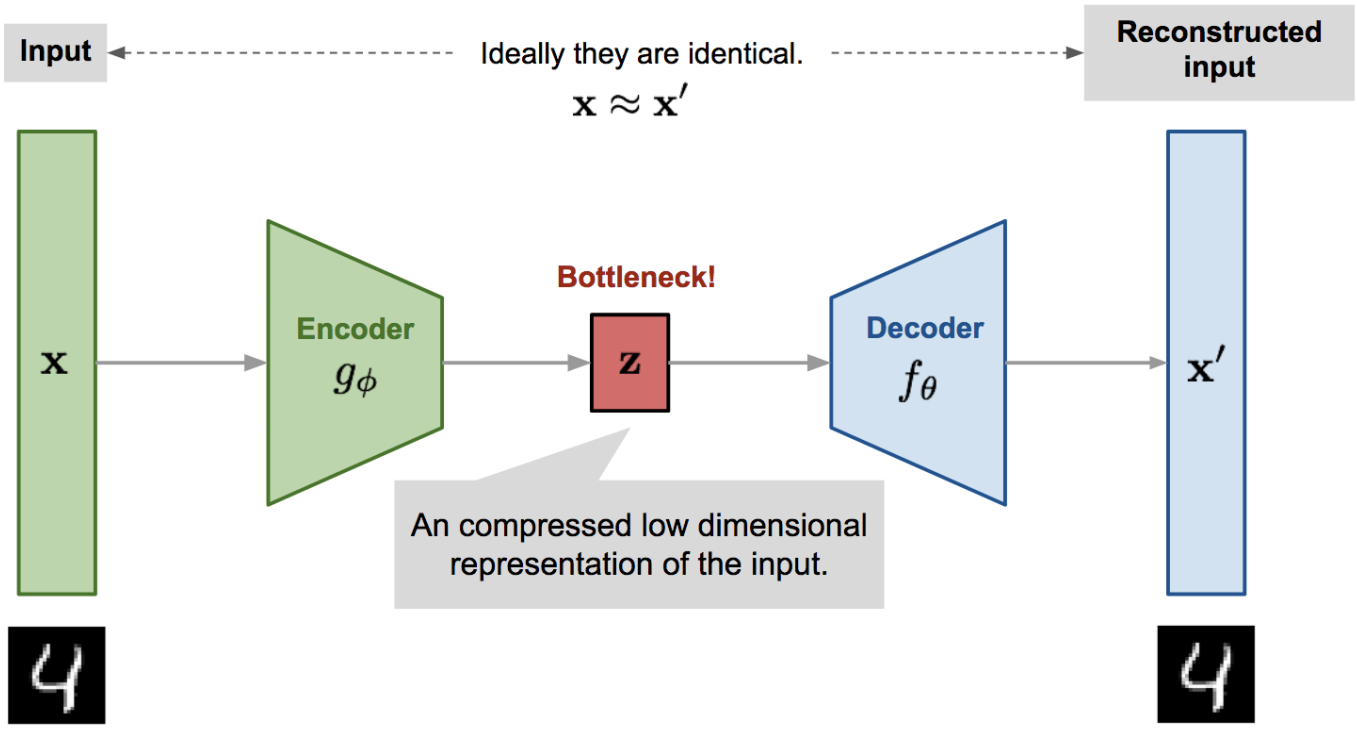
- свойств сглаживающего ядра
- (случайной) инициализации
- (случайного) выбора x_i в ходе итераций

Хорошо подходят для разведочного анализа даннных.

Автокодировщики



Автокодировщики



Вопрос

Какой метод из первой части курса похож на автокодировщик?

Способы использования автокодировщиков

- Генерация признаков (feature generation), например, для эффективного решения задач обучения с учителем
- Снижение размерности (dimensionality reduction)
- Сжатие данных с минимальными потерями
- Обучаемая векторизация объектов, встраиваемая в более глубокие нейросетевые архитектуры
- Генерация синтетических объектов, похожих на реальные

Rumelhart, Hinton, Williams. Learning Internal Representations by Error Propagation. 1986.

David Charte et al. A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. 2018.

Линейный автокодировщик и метод главных компонент

$$\mathscr{L}_{AE}(A,B) = \sum_{i=1}^{\ell} \|BAx_i - x_i\|^2 \to \min_{A,B}$$

Метод главных компонент: $F = (x_1 \dots x_\ell)^T$, $U^T U = I_m$, $G = FU$,

$$\|F - GU^T\|^2 = \sum_{i=1}^{\ell} \|UU^T x_i - x_i\|^2 \to \min_{U}$$

Автокодировщик обобщает метод главных компонент:

- не обязательно $B = A^T$ (хотя часто так делают)
- произвольные A, B вместо ортогональных
- нелинейные модели
- произвольная функция потерь $\mathscr{L}$ вместо квадратичной
- SGD оптимизация вместо сингулярного разложения (SVD)

Разреживающий автокодировщик (Sparse AE)

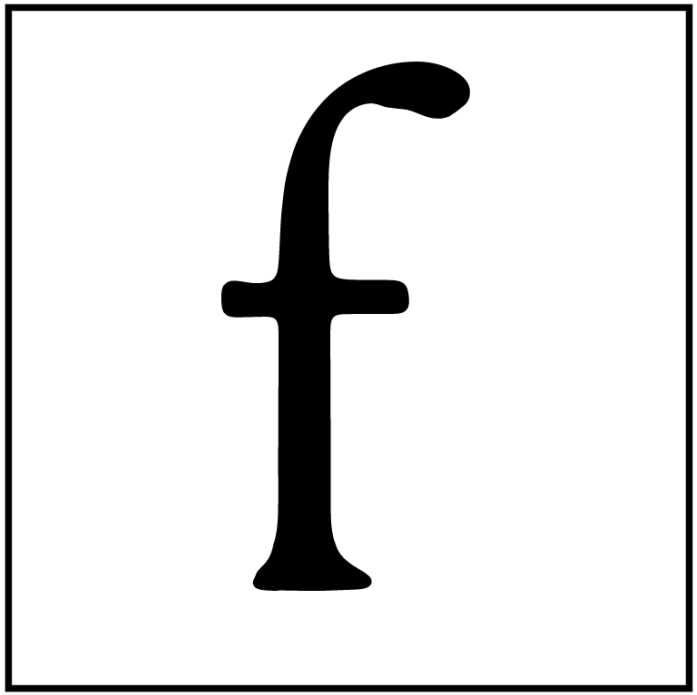
Напоминание из SVM

Если у функции потерь излом, то отбор объектов. Если в регуляризаторе, то признаков.

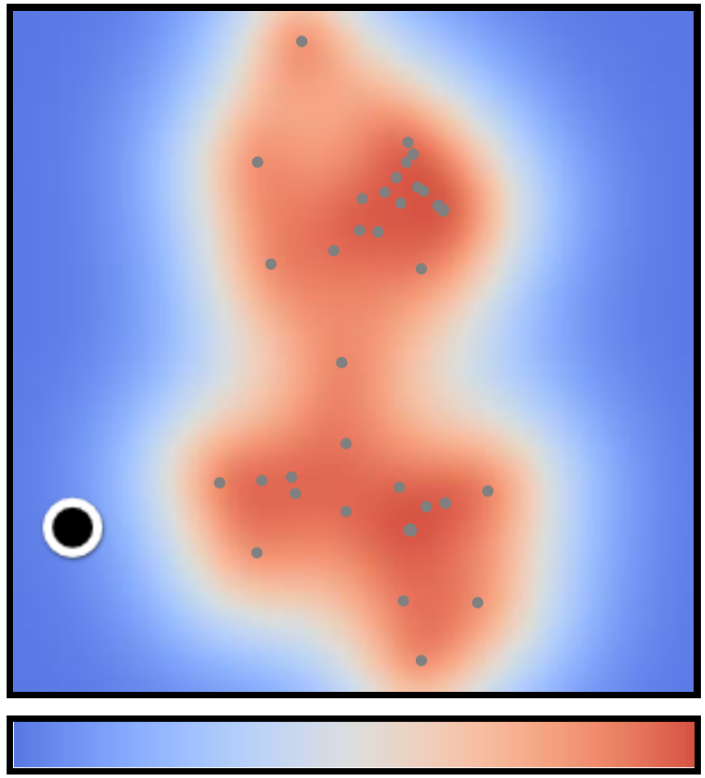
- Применение L_1 или L_2-регуляризации к векторам весов
- Применение L_1-регуляризации к кодовым векторам $z_i = Ax_i$
- Энтропийная регуляризация

D.Arpit et al. Why regularized auto-encoders learn sparse representation? 2015

Please drag the black and white circle around the heat map to explore the 2D font manifold!



Select Character: f



Unlikely Probability Likely

2d font manifold demonstration

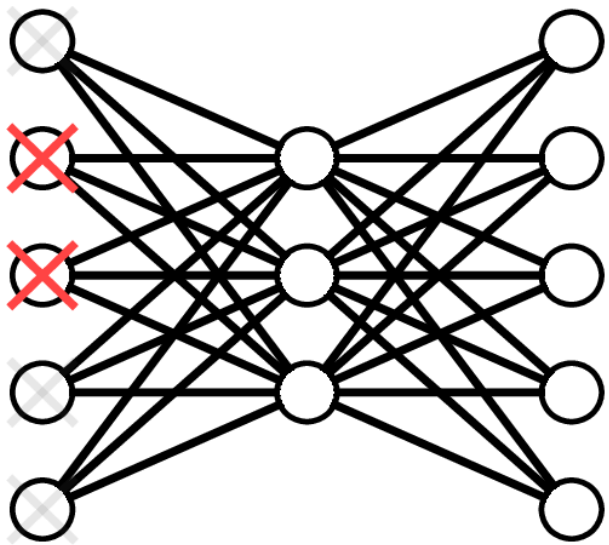
Шумоподавляющий автокодировщик (Denoising AE)

Устойчивость кодовых векторов z_i относительно шума в x_i:

$$\mathscr{L}_{DAE}(\alpha,\beta) = \sum_{i=1}^{\ell} E_{\tilde{x}\sim q(\tilde{x}|x_i)} \mathscr{L}(g(f(\tilde{x},\alpha),\beta),x_i) \to \min_{\alpha,\beta}$$

Вместо вычисления матожидания $E_{\tilde{x}}$ в методе стохастического градиента объекты x_i сэмплируются и зашумляются по одному: $\tilde{x} \sim q(\tilde{x}|x_i)$

- гауссовский шум: $\tilde{x} \sim N(x_i, \sigma^2 I)$
- обнуление компонент вектора x_i с вероятностью p_0



P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. ICML-2008.

Вариационный автокодировщик (Variational AE)

Строится генеративная модель, способная порождать новые объекты x, похожие на объекты выборки $X^\ell = \{x_1, \ldots, x_\ell\}$
$q_\alpha(z|x)$ — вероятностный кодировщик с параметром α
$p_\beta(\hat{x}|z)$ — вероятностный декодировщик с параметром β

$$\mathscr{L}_{VAE}(\alpha, \beta) = \sum_{i=1}^{\ell} \log p(x_i) = \sum_{i=1}^{\ell} \log \int q_\alpha(z|x_i) \frac{p_\beta(x_i|z)p(z)}{q_\alpha(z|x_i)} dz \geq$$

$$\geq \sum_{i=1}^{\ell} \int q_\alpha(z|x_i) \log p_\beta(x_i|z) dz - KL(q_\alpha(z|x_i)\|p(z)) \to \max_{\alpha,\beta}$$

D.P.Kingma, M.Welling. Auto-encoding Variational Bayes. 2013.
C.Doersch. Tutorial on variational autoencoders. 2016.

$$\sum_{i=1}^{\ell} \underbrace{E_{z\sim q_\alpha(z|x_i)} \log p_\beta(x_i|z)}_{\text{качество реконструкции}} - \underbrace{KL(q_\alpha(z|x_i)\|p(z))}_{\text{регуляризатор по } \alpha} \to \max_{\alpha,\beta}$$

где $p(z)$ — априорное распределение, обычно $N(0, \sigma^2 I)$

$$\sum_{i=1}^{\ell} \underbrace{E_{z\sim q_\alpha(z|x_i)} \log p_\beta(x_i|z)}_{\text{качество реконструкции}} - \underbrace{KL(q_\alpha(z|x_i)\|p(z))}_{\text{регуляризатор по } \alpha} \to \max_{\alpha,\beta}$$

где $p(z)$ — априорное распределение, обычно $N(0, \sigma^2 I)$
Репараметризация $q_\alpha(z|x_i): \ z = f(x_i, \alpha, \varepsilon), \ \varepsilon \sim N(0, I)$

Метод стохастического градиента:

- сэмплировать $x_i \sim X^\ell, \ \varepsilon \sim N(0, I), \ z = f(x_i, \alpha, \varepsilon)$
- градиентный шаг
  $\alpha = \alpha + h\nabla_\alpha[\log p_\beta(x_i|f(x_i, \alpha, \varepsilon)) - KL(q_\alpha(z|x_i)\|p(z))]$
  $\beta = \beta + h\nabla_\beta[\log p_\beta(x_i|z)]$

Генерация похожих объектов:

$$x \sim p_\beta(x|f(x_i, \alpha, \varepsilon)), \varepsilon \sim N(0, I)$$

Автокодировщики для обучения с учителем

Данные: неразмеченные $(x_i)_{i=1}^{\ell}$, размеченные $(x_i, y_i)_{i=\ell+1}^{\ell+k}$

Совместное обучение кодировщика, декодировщика и предсказательной модели (классификации, регрессии или др.)

$$\sum_{i=1}^{\ell} \mathscr{L}(g(f(x_i, \alpha), \beta), x_i) + \lambda \sum_{i=\ell+1}^{\ell+k} \tilde{\mathscr{L}}(\hat{y}(f(x_i, \alpha), \gamma), y_i) \to \min_{\alpha,\beta,\gamma}$$

Функции потерь:

$\mathscr{L}(\hat{x}_i, x_i)$ — реконструкция

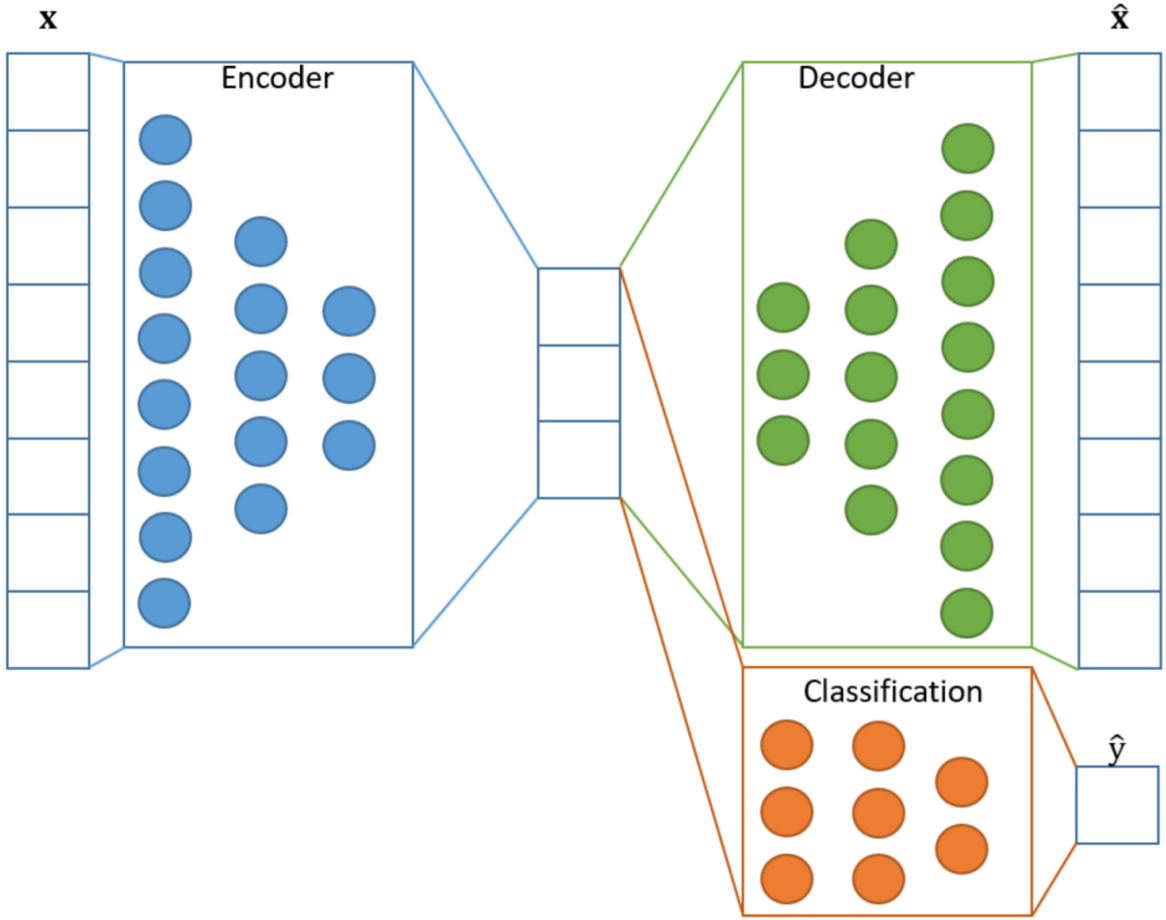
$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — предсказание

Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. 2020.

$z_i = f(x_i, \alpha)$ — кодировщик

$\hat{x}_i = g(z_i, \beta)$ — декодировщик

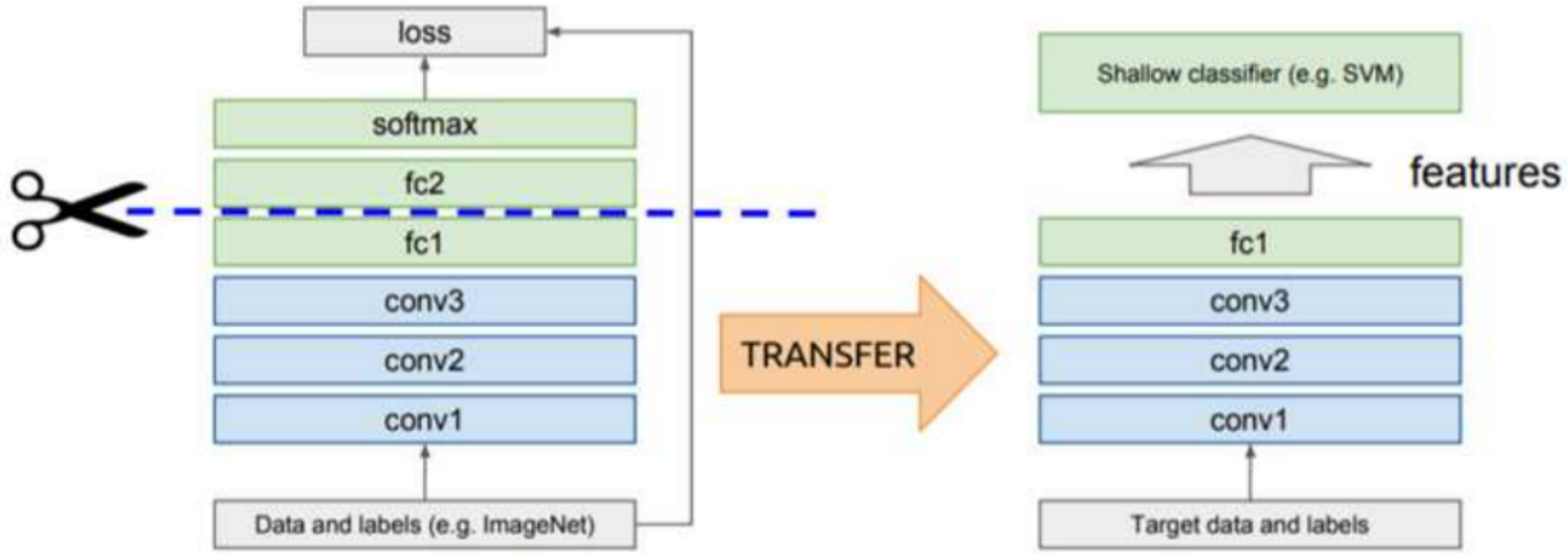
$\hat{y}_i = \hat{y}(z_i, \gamma)$ — классификатор



Пред-обучение нейронных сетей (pre-training)

Свёрточная сеть для обработки изображений:

- $z = f(x, \alpha)$ — свёрточные слои для векторизации объектов
- $y = g(z, \beta)$ — полносвязные слои под конкретную задачу



Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. How transferable are features in deep neural networks? 2014.

Перенос обучения (transfer learning)

- $f(x, \alpha)$ — универсальная часть модели (векторизация)
- $g(x, \beta)$ — специфичная для задачи часть модели

Базовая задача на выборке $\{x_i\}_{i=1}^{\ell}$ с функцией потерь $\mathscr{L}_i$:

$$\sum_{i=1}^{\ell} \mathscr{L}_i(f(x_i, \alpha), g(x_i, \beta)) \to \min_{\alpha,\beta}$$

Целевая задача на другой выборке $\{x'_i\}_{i=1}^{m}$ с другими $\mathscr{L}'_i, g'$:

$$\sum_{i=1}^{m} \mathscr{L}'_i(f(x'_i, \alpha), g'(x'_i, \beta')) \to \min_{\beta'}$$

при $m \ll \ell$ это может быть намного лучше, чем

$$\sum_{i=1}^{m} \mathscr{L}'_i(f(x'_i, \alpha), g'(x'_i, \beta')) \to \min_{\alpha,\beta'}$$

Sinno Jialin Pan, Qiang Yang. A Survey on Transfer Learning. 2009

Многозадачное обучение (multi-task learning)

- $f(x, \alpha)$ — универсальная часть модели (векторизация)
- $g(x, \beta)$ — специфичные части модели для задач $t \in T$

Одновременное обучение модели f по задачам X_t, $t \in T$:

$$\sum_{t \in T} \sum_{i \in X_t} \mathscr{L}_{ti}(f(x_{ti}, \alpha), g(x_{ti}, \beta_t)) \to \min_{\alpha, \{\beta_t\}}$$

Обучаемость (learnability): качество решения отдельной задачи $\langle X_t, \mathscr{L}_t, g_t \rangle$ улучшается с ростом объёма выборки $\ell_t = |X_T|$.

Learning to learn: качество решения каждой из задач $t \in T$ улучшается с ростом ℓ_t и общего числа задач $|T|$.

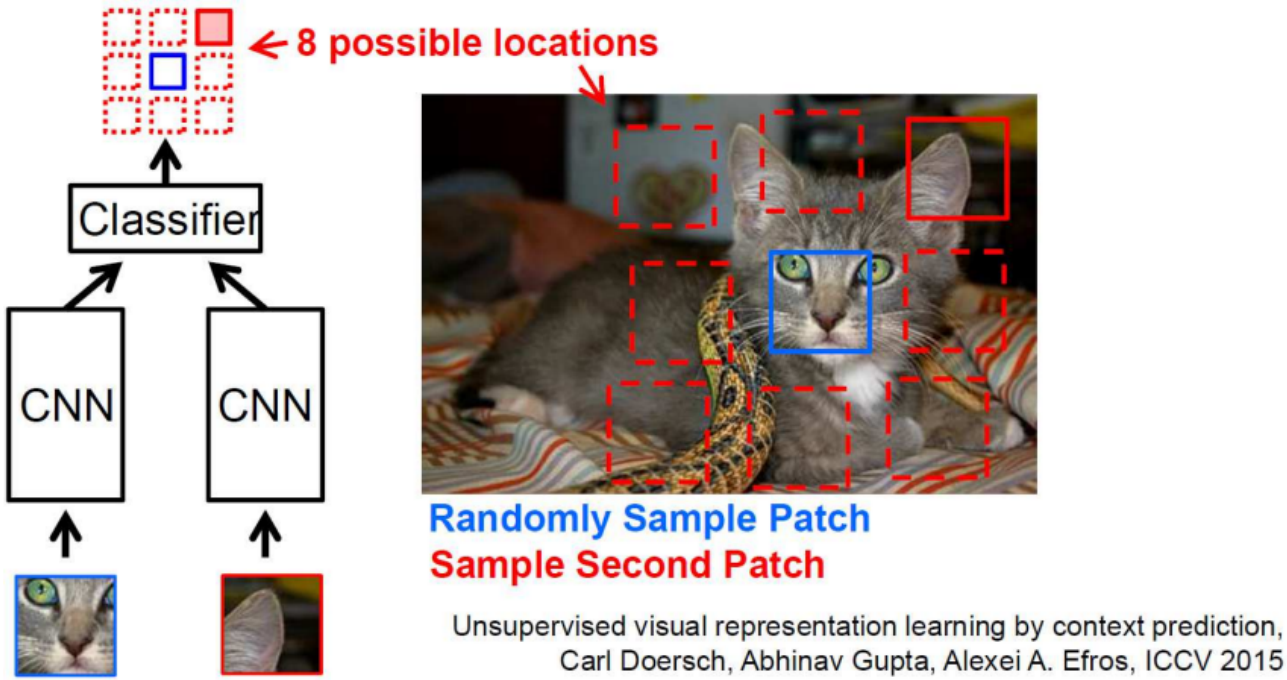
Few-shot learning: для решения задачи t достаточно небольшого числа примеров, иногда даже одного.

M.Crawshaw. Multi-task learning with deep neural networks: a survey. 2020

Y.Wang et al. Generalizing from a few examples: a survey on few-shot learning. 2020

Самостоятельное обучение (self-supervised learning)

Модель векторизации $z = f(x, \alpha)$ обучается предсказывать взаимное расположение пар фрагментов одного изображения



Преимущество: сеть выучивает векторные представления объектов без размеченной обучающей выборки. Их качество не уступает полученным по размеченному ImageNet.

Дистилляция моделей или суррогатное моделирование

Обучение сложной модели $a(x, w)$ «долгое, дорогое»:

$$\sum_{i=1}^{\ell} \mathscr{L}(a(x_i, w), y_i) \to \min_w$$

Обучение простой модели $b(x, w')$, возможно, на других данных:

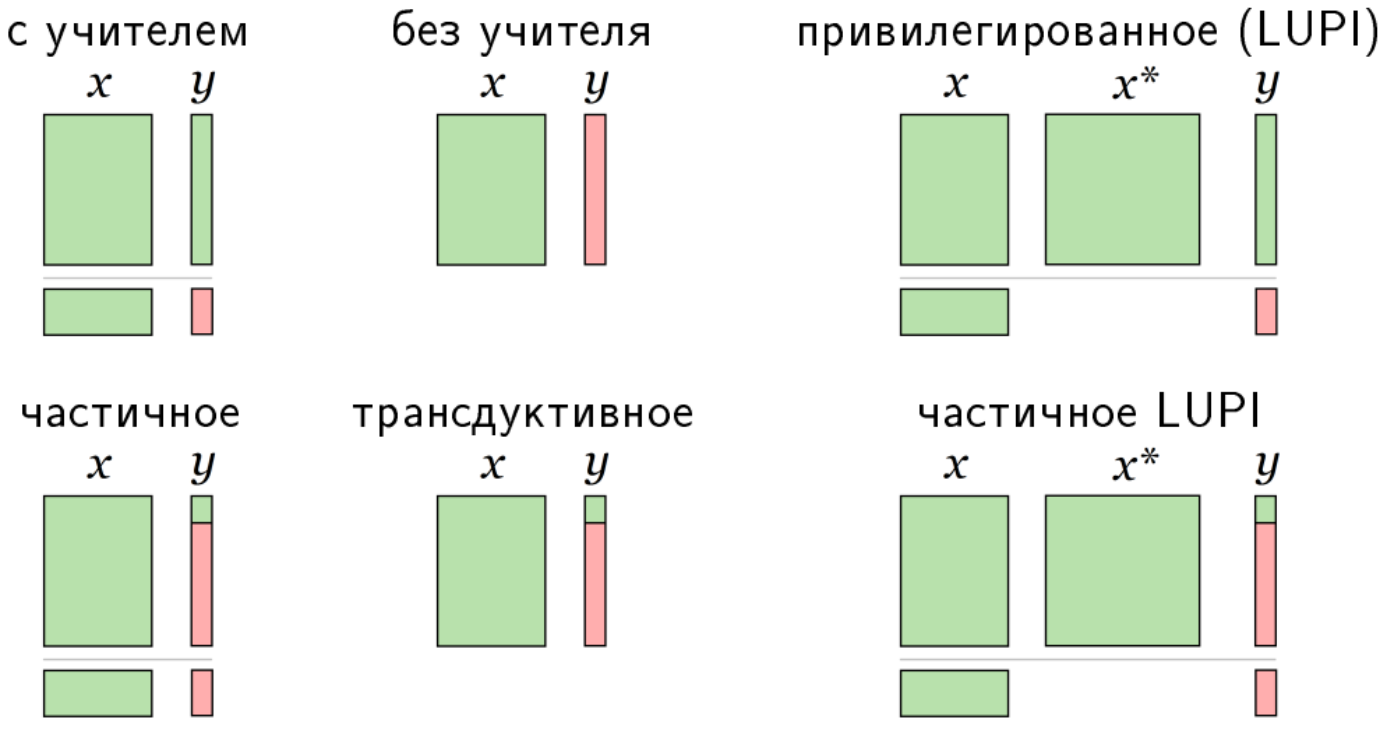
$$\sum_{i=1}^{k} \mathscr{L}(b(x'_i, w'), a(x'_i, w)) \to \min_{w'}$$

Примеры задач:

- замена сложной модели (климат, аэродинамика и др.), которая вычисляется на суперкомпьютере месяцами, «лёгкой» аппроксимирующей суррогатной моделью
- замена сложной нейросети, которая обучается неделями на больших данных, «лёгкой» аппроксимирующей нейросетью с минимизацией числа нейронов и связей

Обучение с использованием привилегированной информации

LUPI — Learning Using Priveleged Information



V.Vapnik, A.Vashist. A new learning paradigm: Learning Using Privileged Information // Neural Networks. 2009.

Примеры задач с привилегированной информацией x^*

- x — первичная (1D) структура белка
  x^* — третичная (3D) структура белка
  y — иерархическая классификация функции белка
- x — предыстория временного ряда
  x^* — информация о будущем поведении ряда
  y — прогноз следующей точки ряда
- x — текстовый документ
  x^* — выделенные ключевые слова или фразы
  y — категория документа
- x — пара (запрос, документ)
  x^* — выделенные асессором ключевые слова или фразы
  y — оценка релевантности

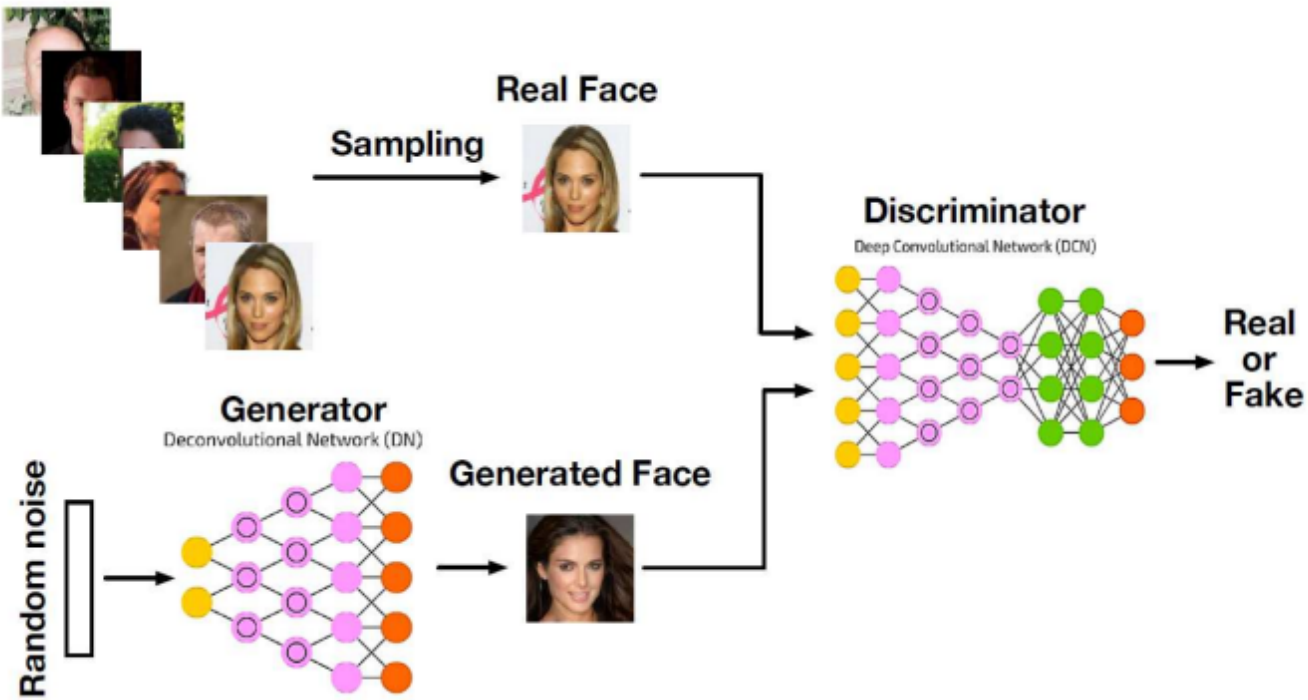
Задача обучения с привилегированной информацией

- Раздельное обучение модели-ученика и модели-учителя:
$$\sum_{i=1}^{\ell} \mathscr{L}(a(x_i, w), y_i) \to \min_w \qquad \sum_{i=1}^{\ell} \mathscr{L}(a(x_i^*, w^*), y_i) \to \min_w$$
- Модель-ученик обучается повторять ошибки модели-учителя:
$$\sum_{i=1}^{\ell} \mathscr{L}(a(x_i, w), y_i) + \mu \mathscr{L}(a(x_i, w), a(x_i^*, w^*)) \to \min_w$$
- **Совместное обучение** модели-ученика и модели-учителя:
$$\sum_{i=1}^{\ell} \mathscr{L}(a(x_i, w), y_i) + \lambda \mathscr{L}(a(x_i^*, w^*), y_i) + \mu \mathscr{L}(a(x_i, w), a(x_i^*, w^*)) \to \min_{w, w^*}$$

D.Lopez-Paz, L.Bottou, B.Scholkopf, V.Vapnik. Unifying distillation and privileged information. 2016.

Генеративная состязательная сеть (Generative Adversarial Net, GAN)

Генератор $G(z)$ учится порождать объекты x из шума z
Дискриминатор $D(x)$ учится отличать их от реальных объектов



Antonia Creswell et al. Generative Adversarial Networks: an overview. 2017.
Zhengwei Wang, Qi She, Tomas Ward. Generative Adversarial Networks: a survey and taxonomy. 2019.

Chris Nicholson. A Beginner's Guide to Generative Adversarial Networks. 2019.

Постановка задачи GAN

Есть выборка объектов $\{x_i\}_{i=1}^m$ из X

Обучаем

- вероятностную генеративную модель $G(z, \alpha) : x \sim p(x|z, \alpha)$
- вероятностную дискриминативную модель $D(x, \beta) = p(1|x, \beta)$

Критерии:

- обучение дискриминативной модели D:

$$\sum_{i=1}^{m} \ln D(x_i, \beta) + \ln(1 - D(G(z_i, \alpha), \beta)) \to \max_{\beta}$$

- обучение генеративной модели G по случайному шуму $\{z_i\}_{i=1}^m$:

$$\sum_{i=1}^{m} \ln(1 - D(G(z_i, \alpha), \beta)) \to \min_{\alpha}$$

Ian Goodfellow et al. Generative Adversarial Nets. 2014

StyleGAN demo

Посмотрим видео

Статьи тут: https://nvlabs.github.io/stylegan2/versions.html

Резюме

- Кластеризация и карты Кохонена
- Автокодировщики, включая совместное обучение с классификацией
- Многозадачное обучение (multi-task learning)
- Перенос обучения (transfer learning)
- Дистилляция и суррогатное моделирование
- Состязательные сети (GAN)

Резюме

- Кластеризация и карты Кохонена
- Автокодировщики, включая совместное обучение с классификацией
- Многозадачное обучение (multi-task learning)
- Перенос обучения (transfer learning)
- Дистилляция и суррогатное моделирование
- Состязательные сети (GAN)

Что ещё можно посмотреть?

- Лекцию 13 курса CS231n про GAN