

История и методология машинного обучения

Лекция 2. Постановки оптимизационных задач

Воронцов Константин Вячеславович

`vokov@forecsys.ru`

`http://www.MachineLearning.ru «User:Vokov»`

МФТИ • 26 марта 2020

- 1 Обучение с учителем**
 - Регрессия и классификация
 - Регуляризация
 - Ранжирование
- 2 Обучение без учителя**
 - Восстановление плотности
 - Кластеризация и частичное обучение
 - Понижение размерности и обучение представлений
- 3 Некоторые неклассические парадигмы обучения**
 - Обучение с привилегированной информацией
 - Перенос обучения (transfer learning)
 - Генеративные состязательные сети (GAN)

Общая оптимизационная задача машинного обучения

Дано: обучающая выборка $\{x_i: i = 1, \dots, \ell\}$

Найти: вектор параметров w модели $a(x, w)$

Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} \mathcal{L}_i(a(x_i, w)) \rightarrow \min_w$$

или минимум регуляризованного эмпирического риска

$$\sum_{i=1}^{\ell} \mathcal{L}_i(a(x_i, w)) + R(w) \rightarrow \min_w$$

где \mathcal{L}_i — функция потерь модели w на объекте x_i

Оптимизационная задача восстановления регрессии

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$

- 1 Фиксируется модель регрессии, например, *линейная*:

$$a(x, w) = \langle x, w \rangle = \sum_{j=1}^n w_j f_j(x), \quad w \in \mathbb{R}^n$$

- 2 Фиксируется функция потерь, например, *квадратичная*:

$$\mathcal{L}_i(a) = (a - y_i)^2$$

- 3 Метод обучения — *метод наименьших квадратов*:

$$Q(w) = \sum_{i=1}^{\ell} (a(x_i, w) - y_i)^2 \rightarrow \min_w$$

- 4 Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^k (a(\tilde{x}_i, w) - \tilde{y}_i)^2$$

Оптимизационная задача обучения классификация

Обучающая выборка: $X^\ell = (x_i, y_i)_{i=1}^\ell$, $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$

- 1 Фиксируется модель классификации, например, *линейная*:

$$a(x, w) = \text{sign}\langle x, w \rangle = \text{sign} \sum_{j=1}^n w_j f_j(x)$$

- 2 Функция потерь — пороговая или её **верхняя оценка**:

$$\mathcal{L}_i(a) = [ay_i < 0] = [\langle x_i, w \rangle y_i < 0] \leq \mathcal{L}(\langle x_i, w \rangle y_i)$$

- 3 Метод обучения — *минимизация эмпирического риска*:

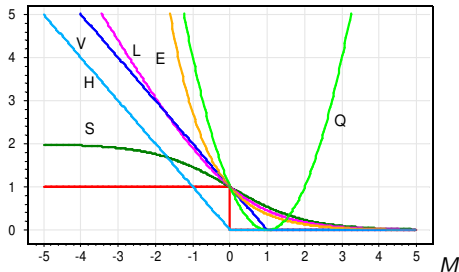
$$Q(w) = \sum_{i=1}^{\ell} [\langle x_i, w \rangle y_i < 0] \leq \sum_{i=1}^{\ell} \mathcal{L}(\langle x_i, w \rangle y_i) \rightarrow \min_w$$

- 4 Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:

$$\bar{Q}(w) = \frac{1}{k} \sum_{i=1}^k [\langle \tilde{x}_i, w \rangle \tilde{y}_i < 0]$$

Непрерывные верхние оценки пороговой функции потерь

Часто используемые непрерывные функции потерь $\mathcal{L}(M)$:



$[M < 0]$

$$V(M) = (1 - M)_+$$

$$H(M) = (-M)_+$$

$$L(M) = \log_2(1 + e^{-M})$$

$$Q(M) = (1 - M)^2$$

$$S(M) = 2(1 + e^M)^{-1}$$

$$E(M) = e^{-M}$$

— пороговая функция потерь

— кусочно-линейная (SVM)

— кусочно-линейная (Hebb's rule)

— логарифмическая (LR)

— квадратичная (FLD)

— сигмоидная (ANN)

— экспоненциальная (AdaBoost)

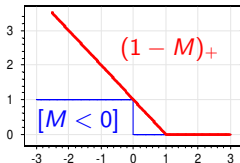
Метод опорных векторов SVM (двухклассовый)

$M_i(w, w_0) = y_i(\langle w, x_i \rangle - w_0)$ — отступ в линейной модели

Кусочно-линейная функция потерь:

$$\sum_{i=1}^{\ell} (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}$$

- *Функция потерь* штрафует объекты за приближение к границе классов
- *Регуляризация* максимизирует зазор между классами и штрафует мультиколлинеарность



Важнейшие свойства SVM:

- Задача выпуклого программирования, решение единственно
- Решение *разрежено* — зависит только от *опорных объектов*
- Обобщение на нелинейные модели: $\langle x, x_i \rangle \rightarrow K(x, x_i)$

Логистическая регрессия (двухклассовая)

Линейная модель классификации $a(x, w) = \text{sign}\langle x, w \rangle$

Логарифмическая функция потерь:

$$\sum_{i=1}^{\ell} \ln(1 + \exp(-\langle w, x_i \rangle y_i)) + \frac{\tau}{2} \|w\|^2 \rightarrow \min_w$$

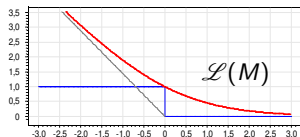
Логарифмическая функция потерь:

$$\mathcal{L}(M) = \ln(1 + e^{-M})$$

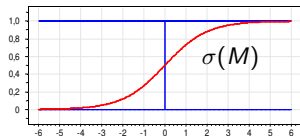
Модель условной вероятности:

$$P(y|x, w) = \sigma(M) = \frac{1}{1 + e^{-M}},$$

где $\sigma(M)$ — сигмоидная функция



M



M

Логистическая регрессия (многоклассовая)

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x, w) = \arg \max_{y \in Y} \langle w_y, x \rangle, \quad x, w_y \in \mathbb{R}^n$$

Вероятность того, что объект x относится к классу y :

$$P(y|x, w) = \frac{\exp\langle w_y, x \rangle}{\sum_{z \in Y} \exp\langle w_z, x \rangle} = \text{SoftMax}_{y \in Y} \langle w_y, x \rangle,$$

где $\text{SoftMax}: \mathbb{R}^Y \rightarrow \mathbb{R}^Y$ переводит произвольный вектор в нормированный вектор дискретного распределения.

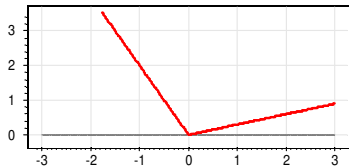
Максимизация правдоподобия (log-loss) с регуляризацией:

$$-\sum_{i=1}^{\ell} \ln P(y_i|x_i, w) + \frac{\tau}{2} \sum_{y \in Y} \|w_y\|^2 \rightarrow \min_w.$$

Квантильная регрессия

Функция потерь, $\varepsilon = a(x_i, w) - y_i$:

$$\mathcal{L}(\varepsilon) = \begin{cases} C_+ |\varepsilon|, & \varepsilon > 0 \\ C_- |\varepsilon|, & \varepsilon < 0; \end{cases}$$



Модель регрессии: линейная $a(x_i, w) = \langle x_i, w \rangle$.

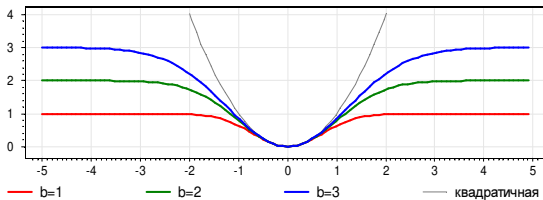
Сведение к задаче линейного программирования:

замена переменных $\varepsilon_i^+ = (a(x_i) - y_i)_+$, $\varepsilon_i^- = (y_i - a(x_i))_+$;

$$\begin{cases} Q = \sum_{i=1}^{\ell} C_+ \varepsilon_i^+ + C_- \varepsilon_i^- \rightarrow \min_w; \\ \langle x_i, w \rangle - y_i = \varepsilon_i^+ - \varepsilon_i^-; \\ \varepsilon_i^+ \geq 0; \quad \varepsilon_i^- \geq 0. \end{cases}$$

Робастная регрессия

Функция Мешалкина: $\mathcal{L}(\varepsilon) = b(1 - \exp(-\frac{1}{b}\varepsilon^2))$, $\varepsilon = f - y$



Модель регрессии: $a(x) = f(x, w)$

Постановка оптимизационной задачи:

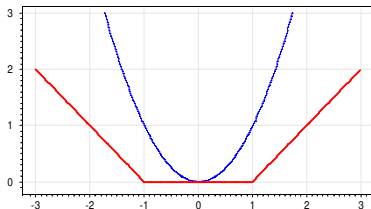
$$\sum_{i=1}^{\ell} \exp\left(-\frac{1}{b}(f(x_i, w) - y_i)^2\right) \rightarrow \max_w$$

Численное решение методом Ньютона-Рафсона

SVM-регрессия

Модель регрессии: $a(x) = \langle x, w \rangle - w_0$, $w \in \mathbb{R}^n$, $w_0 \in \mathbb{R}$.

Функция потерь: $\mathcal{L}(\varepsilon) = (|\varepsilon| - \delta)_+$



Постановка оптимизационной задачи:

$$\sum_{i=1}^{\ell} (|\langle w, x_i \rangle - w_0 - y_i| - \delta)_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}.$$

Сводится к выпуклой задаче квадратичного программирования

Регуляризаторы, штрафующие сложность модели

Регуляризатор — аддитивная добавка к основному критерию:

$$Q(w) = \sum_{i=1}^{\ell} \mathcal{L}_i(\langle x_i, w \rangle) + \tau \text{штраф}(w) \rightarrow \min_w$$

где τ — коэффициент регуляризации

L₂-регуляризация (гребневая регрессия, SVM):

$$\text{штраф}(w) = \|w\|_2^2 = \sum_{j=1}^n w_j^2.$$

L₁-регуляризация (LASSO, ElasticNet):

$$\text{штраф}(w) = \|w\|_1 = \sum_{j=1}^n |w_j|.$$

L₀-регуляризация (критерии Акаике AIC, байесовский BIC):

$$\text{штраф}(w) = \|w\|_0 = \sum_{j=1}^n [w_j \neq 0].$$

Негладкие регуляризаторы для отбора признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell} \mathcal{L}_i(a(x_i, w)) + \tau \sum_{j=1}^n R_{\mu}(w_j) \rightarrow \min_w .$$

Elastic Net:

$$R_{\mu}(\alpha) = \mu|\alpha| + \alpha^2$$

Support Features Machine (SFM):

$$R_{\mu}(\alpha) = \begin{cases} 2\mu|\alpha|, & |\alpha| \leq \mu; \\ \mu^2 + \alpha^2, & |\alpha| \geq \mu; \end{cases}$$

Relevance Features Machine (RFM):

$$R_{\mu}(\alpha) = \ln(\mu\alpha^2 + 1)$$

Определения и обозначения

X — множество объектов

$X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка

$i \prec j$ — правильный порядок на парах $(i, j) \in \{1, \dots, \ell\}^2$

Задача:

построить ранжирующую функцию $a: X \rightarrow \mathbb{R}$ такую, что

$$i \prec j \Rightarrow a(x_i) < a(x_j)$$

Линейная модель ранжирования:

$$a(x, w) = \langle x, w \rangle$$

где $x \mapsto (f_1(x), \dots, f_n(x)) \in \mathbb{R}^n$ — вектор признаков объекта x

Пример. Задача ранжирования поисковой выдачи

D — коллекция текстовых документов (documents)

Q — множество запросов (queries)

$D_q \subseteq D$ — множество документов, найденных по запросу q

$X = Q \times D$ — объектами являются пары «запрос, документ»:

$$x \equiv (q, d), \quad q \in Q, \quad d \in D_q$$

Y — упорядоченное множество рейтингов

$y: X \rightarrow Y$ — оценки релевантности, поставленные ассессорами:
чем выше оценка $y(q, d)$, тем релевантнее документ d запросу q

Правильный порядок определён только между документами, найденными по одному и тому же запросу q :

$$(q, d) \prec (q, d') \Leftrightarrow y(q, d) < y(q, d')$$

Основные подходы к ранжированию

- Point-wise — поточечный
- Pair-wise — попарный
- List-wise — списочный

Переход к гладкому функционалу качества ранжирования:

$$Q(w) = \sum_{i \prec j} \underbrace{[a(x_j) - a(x_i) < 0]}_{\text{Margin}(i,j)} \leq \sum_{i \prec j} \mathcal{L}(a(x_j, w) - a(x_i, w)) \rightarrow \min_w$$

где $a(x, w)$ — модель ранжирования с вектором параметров w ;
 $\mathcal{L}(M)$ — убывающая непрерывная функция отступа $\text{Margin}(i, j)$:

- $\mathcal{L}(M) = (1 - M)_+$ — RankSVM, аналог SVM
- $\mathcal{L}(M) = \ln(1 + e^{-M})$ — RankNet, аналог LR
- $\mathcal{L}(M) = \exp(-M)$ — RankBoost, аналог AdaBoost

Задача восстановления плотности распределения

Дано: обучающая выборка $\{x_i: i = 1, \dots, \ell\}$

Найти: вектор параметров θ в модели $p(x|\theta)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) \rightarrow \max_{\theta}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta) + \ln p(\theta|\gamma) \rightarrow \max_{\theta}$$

где γ — вектор гиперпараметров априорного распределения

Задача восстановления смеси плотностей распределения

Дано: обучающая выборка $\{x_i: i = 1, \dots, \ell\}$

Найти: параметры w_j, θ_j в модели $p(x|\theta, w) = \sum_{j=1}^K w_j p(x|\theta_j)$

Критерий: максимум правдоподобия

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta, w) \rightarrow \max_{\theta, w}$$

или максимум апостериорной вероятности

$$\sum_{i=1}^{\ell} \ln p(x_i|\theta, w) + \ln p(\theta, w|\gamma) \rightarrow \max_{\theta, w}$$

где γ — вектор гиперпараметров априорного распределения

Задача кластеризации (clustering)

Дано: обучающая выборка $\{x_i \in \mathbb{R}^n : i = 1, \dots, \ell\}$

Найти:

— центры кластеров $\mu_j \in \mathbb{R}^n, j = 1, \dots, K$

— кластеризации объектов $a_i \in \{1, \dots, K\}$

Критерий: минимум внутрикластерных расстояний

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 \rightarrow \min_{\{a_i\}, \{\mu_j\}}$$

в случае евклидовой метрики

$$\|x - \mu_j\|^2 = \sum_{d=1}^n (f_d(x) - \mu_{jd})^2$$

Задача частичного обучения (semi-supervised learning)

Дано:

$X^k = \{x_1, \dots, x_k\}$ — размеченные объекты (labeled data);
 $\{y_1, \dots, y_k\}$

$U = \{x_{k+1}, \dots, x_\ell\}$ — неразмеченные объекты (unlabeled data).

Найти: классификации $\{a_{k+1}, \dots, a_\ell\}$ неразмеченных объектов

Критерий без модели классификации (transductive learning):

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^k [a_i \neq y_i] \rightarrow \min_{\{a_i\}, \{\mu_j\}}$$

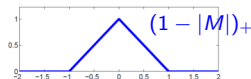
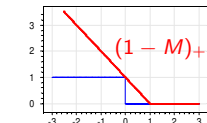
При построении модели классификации, $a_i = a(x_i, w)$:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 + \lambda \sum_{i=1}^k \mathcal{L}(a(x_i, w), y_i) \rightarrow \min_{\{a_i\}, \{\mu_j\}, w}$$

Метод TSVM — трансдуктивный SVM

$M_i = (\langle w, x_i \rangle - w_0) y_i$ — отступ объекта x_i

- Функция потерь $\mathcal{L}(M) = (1 - M)_+$ штрафует за уменьшение отступа
- Функция потерь $\mathcal{L}(M) = (1 - |M|)_+$ штрафует за попадание объекта внутрь разделяющей полосы



Обучение весов w, w_0 по частично размеченной выборке:

$$Q(w, w_0) = \sum_{i=1}^k (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 + \\ + \gamma \sum_{i=k+1}^{\ell} (1 - |M_i(w, w_0)|)_+ \rightarrow \min_{w, w_0}$$

Задачи низкорангового матричного разложения

- Понижение размерности в задачах регрессии
- Понижение размерности в задачах классификации
- Формирование сжатого представления данных

Дано: матрица $Z = \|z_{ij}\|_{n \times m}$, $(i, j) \in \Omega \subseteq \{1..n\} \times \{1..m\}$

Найти: матрицы $X = \|x_{it}\|_{n \times k}$ и $Y = \|y_{tj}\|_{k \times m}$ такие, что

$$\|Z - XY\|^2 = \sum_{(i,j) \in \Omega} \left(z_{ij} - \sum_t x_{it} y_{tj} \right)^2 \rightarrow \min_{X, Y}$$

Дополнительные ограничения, вынуждающие отказаться от SVD:

- неквадратичная функция потерь
- неотрицательное матричное разложение: $x_{it} \geq 0$, $y_{tj} \geq 0$
- разреженные данные: $|\Omega| \ll nm$

Примеры прикладных задач матричного разложения

- 1 **Выявление интересов в рекомендательных системах (recommender systems, collaborative filtering)**

$$z_{iu} = \sum_t p_{it} q_{tu}$$

дано: z_{iu} — рейтинги товаров i , поставленные пользователем u ;

найти: p_{it} — профиль интересов товара i ;

q_{tu} — профиль интересов пользователя u .

- 2 **Разделение смеси химических веществ по данным жидкостной хроматографии**

$$z_{t\lambda} = \sum_i x_{ti} y_{i\lambda}$$

дано: $z_{t\lambda}$ — выход сканирующего УФ-детектора;

найти: x_{ti} — хроматограмма i -го вещества, t — время;

$y_{i\lambda}$ — спектр i -го вещества, λ — длина волны.

Примеры прикладных задач матричного разложения

- 3 Латентный семантический анализ коллекций текстов (тематическое моделирование)

$$z_{wd} = \sum_t \varphi_{wt} \theta_{td}$$

дано: $z_{wd} = p(w|d)$ — частоты слов w в документах d ;

найти: $\varphi_{wt} = p(w|t)$ — распределения слов w в темах t ,

$\theta_{td} = p(t|d)$ — распределения тем t в документах d .

- 4 Оценивание экспрессии генов по данным ДНК-микрочипов с учётом кросс-гибридизации

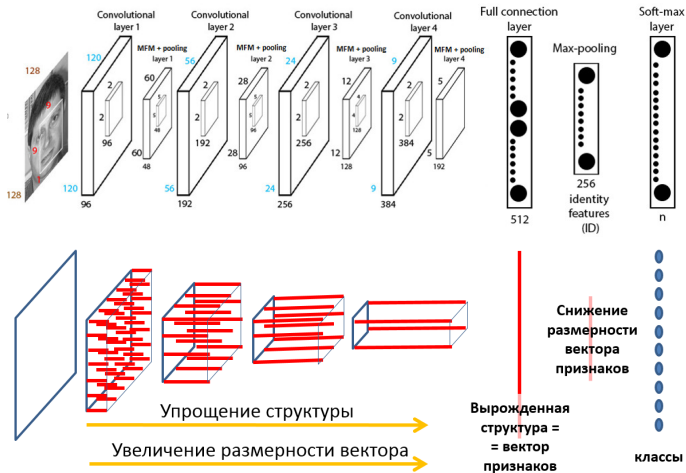
$$z_{pk} = \sum_g a_{pg} c_{gk}$$

дано: z_{pk} — интенсивность свечения p -й пробы на k -м чипе;

найти: a_{pg} — коэффициент сродства p -й пробы g -му гену,

c_{gk} — концентрация g -го гена на k -м чипе.

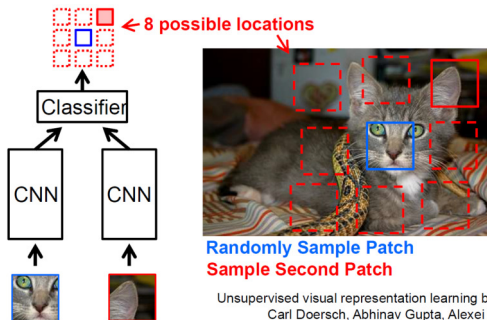
Свёрточные сети глубокого обучения



Визильтер Ю.В., Горбачевич В.С. Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей. ММРО-2017.

Самостоятельное обучение (self-supervised learning)

В компьютерном зрении сеть учится предсказывать взаимное расположение двух фрагментов на одном изображении

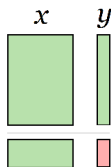


Преимущество: сеть выучивает векторные представления объектов без размеченной обучающей выборки.

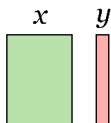
Обучение с использованием привилегированной информации

LUPI — Learning Using Privileged Information

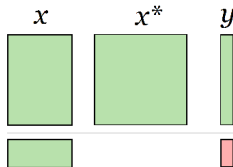
с учителем



без учителя



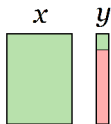
привилегированное (LUPI)



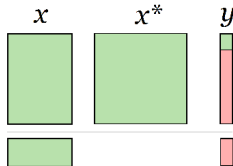
частичное



трандуктивное



частичное LUPI



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
- x — предыстория временного ряда
 x^* — информация о будущем поведении ряда
 y — прогноз следующей точки ряда
- x — данные баллистокордиографии
 x^* — данные ЭКГ (мониторирование по Холтеру)
 y — диагноз
- x — документ
 x^* — выделенные ключевые слова или фразы
 y — категория документа
- x — пара (запрос, документ)
 x^* — выделенные ассессором ключевые слова или фразы
 y — ассессорская оценка релевантности

Задача обучения с привилегированной информацией

Раздельное обучение модели-ученика и **модели-учителя**:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) \rightarrow \min_w \quad \sum_{i=1}^{\ell} \mathcal{L}(a(x_i^*, w^*), y_i) \rightarrow \min_{w^*}$$

Модель-ученик обучается повторять ошибки **модели-учителя**:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) + \mu \mathcal{L}(a(x_i, w), a(x_i^*, w^*)) \rightarrow \min_w$$

Совместное обучение модели-ученика и **модели-учителя**:

$$\sum_{i=1}^{\ell} \mathcal{L}(a(x_i, w), y_i) + \lambda \mathcal{L}(a(x_i^*, w^*), y_i) + \\ + \mu \mathcal{L}(a(x_i, w), a(x_i^*, w^*)) \rightarrow \min_{w, w^*}$$

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

Перенос обучения (transfer learning)

$f(x_i, \alpha)$ — часть модели, универсальная для всех задач

$g(x_i, \beta)$ — часть модели, специфичная для каждой задачи

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

$$\sum_{i=1}^{\ell} \mathcal{L}_i(f(x_i, \alpha), g(x_i, \beta)) \rightarrow \max_{\alpha, \beta}$$

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

$$\sum_{i=1}^m \mathcal{L}'_i(f(x'_i, \alpha), g'(x'_i, \beta')) \rightarrow \max_{\beta'}$$

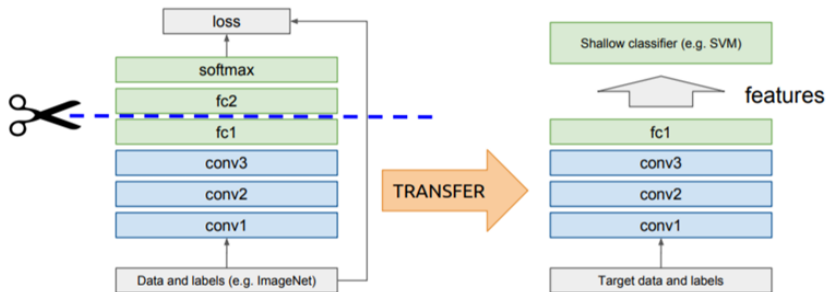
при $m \ll \ell$ это может быть намного лучше, чем

$$\sum_{i=1}^m \mathcal{L}'_i(f(x'_i, \alpha), g'(x'_i, \beta')) \rightarrow \max_{\alpha, \beta'}$$

Пред-обученные (pre-trained) нейронные сети

Свёрточная сеть для обработки изображений:

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

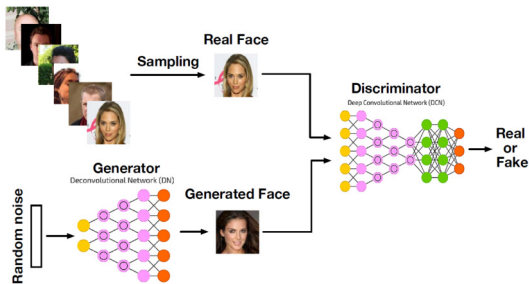


Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. How transferable are features in deep neural networks? 2014.

Генеративная состязательная сеть (Generative Adversarial Net)

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.

<https://pathmind.com/wiki/generative-adversarial-network-gan>. 2019.

Постановка задачи GAN

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

Найти:

вероятностную генеративную модель $G(z, \alpha): x \sim p(x|z, \alpha)$

вероятностную дискриминативную модель $D(x, \beta) = p(1|x, \beta)$

Критерий:

обучение дискриминативной модели D :

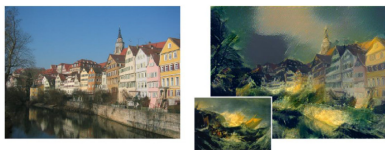
$$\sum_{i=1}^m \ln D(x_i, \beta) + \ln(1 - D(G(z_i, \alpha), \beta)) \rightarrow \max_{\beta}$$

обучение генеративной модели G :

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

Ian Goodfellow et al. Generative Adversarial Nets. 2014

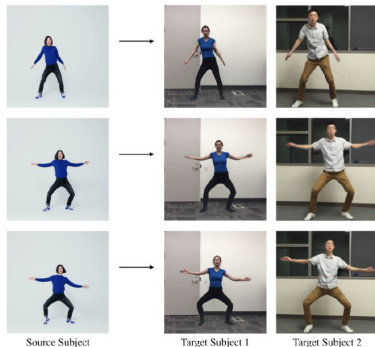
Примеры GAN для синтеза изображений и видео



(d) input image

(e) output 3d face

(f) textured 3d face



Source Subject

Target Subject 1

Target Subject 2

Chuan Li, Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. 2016.

Xiaoxing Zeng, Xiaojiang Peng, Yu Qiao. DF2Net: A Dense Fine Finer Network for Detailed 3D Face Reconstruction. ICCV-2019.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, Alexei A. Efros. Everybody Dance Now. ICCV-2109.

- 1 Предварительная обработка (data preparation)
 - извлечение признаков (feature extraction)
 - отбор признаков (feature selection)
 - восстановление пропусков (missing values)
 - фильтрация выбросов (outlier detection)
- 2 Обучение с учителем (supervised learning)
 - классификация (classification)
 - регрессия (regression)
 - ранжирование (learning to rank)
 - прогнозирование (forecasting)
- 3 Обучение без учителя (unsupervised learning)
 - кластеризация (clustering)
 - поиск ассоциативных правил (association rule learning)
 - восстановление плотности (density estimation)
 - одноклассовая классификация (anomaly detection)
- 4 Частичное обучение (semi-supervised learning)
 - трансдуктивное обучение (transductive learning)
 - обучение с положительными примерами (PU-learning)

- 5 Обучение представлений (representation learning)
 - обучение признаков (feature learning)
 - обучение многообразий (manifold learning)
 - матричные разложения (matrix factorization)
 - глубокое обучение (deep learning)
- 6 Обучение близости/связей (similarity/relational learning)
- 7 Обучение структуры модели (structure learning)
- 8 Состязательное обучение (adversarial learning)
- 9 Обучение с подкреплением (reinforcement learning)
- 10 Привилегированное обучение (privileged learning)
- 11 Динамическое обучение (online/incremental learning)
- 12 Активное обучение (active learning)
- 13 Мета-обучение (meta-learning)
- 14 Перенос обучения (transfer learning)
 - inductive transfer, learning to learn — просто синонимы
 - многозадачное обучение (multitask learning)