Постановки задач оптимизации в машинном обучении Часть 2: обзор задач

Воронцов Константин Вячеславович

(Лаборатория машинного интеллекта МФТИ)

ЛЕТНЯЯ ШКОЛА

• Управление, информация, оптимизация •
«Сириус», Сочи • 23–29 августа 2020

Содержание

Общая оптимизационная задача машинного обучения

Дано: выборка объектов $\{x_i\}_{i=1}^{\ell}$

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

Критерий: минимум эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) \;\to\; \min_w$$

где $L_i(w)$ — функция потерь модели $a(x, w)$ на объекте x_i,

или минимум регуляризованного эмпирического риска

$$\sum_{i=1}^{\ell} L_i(w) + \sum_{j=1}^{r} \tau_j R_j(w) \;\to\; \min_w$$

где R_j — регуляризаторы, τ_j — коэффициенты регуляризации

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*:
$$a(x, w) = \langle x, w\rangle = \sum_{j=1}^n w_j f_j(x), \qquad w \in \mathbb{R}^n$$
2. Фиксируется функция потерь, например, *квадратичная*:
$$L_i(w) = \bigl(a(x_i, w) - y_i\bigr)^2$$
3. Метод обучения — *метод наименьших квадратов*:
$$\sum_{i=1}^\ell \bigl(a(x_i, w) - y_i\bigr)^2 \to \min_w$$
4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:
$$Q(X^k) = \frac{1}{k} \sum_{i=1}^k \bigl(a(\tilde{x}_i, w) - \tilde{y}_i\bigr)^2$$

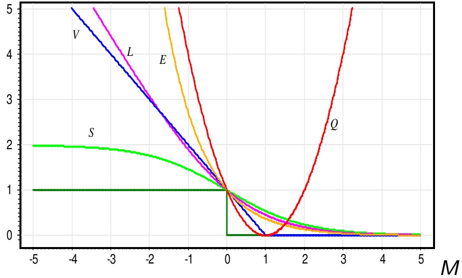
Оптимизационная задача обучения классификация

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*:
$$a(x, w) = \operatorname{sign}\langle x, w\rangle = \operatorname{sign} \sum_{j=1}^n w_j f_j(x)$$
2. Функция потерь — пороговая или её верхняя оценка:
$$L_i(w) = \bigl[a(x_i, w) y_i < 0\bigr] = \bigl[\langle x_i, w\rangle y_i < 0\bigr] \leqslant \mathscr{L}\bigl(\langle x_i, w\rangle y_i\bigr)$$
3. Метод обучения — *минимизация эмпирического риска*:
$$\sum_{i=1}^\ell \bigl[\langle x_i, w\rangle y_i < 0\bigr] \leqslant \sum_{i=1}^\ell \mathscr{L}\bigl(\langle x_i, w\rangle y_i\bigr) \to \min_w$$
4. Проверка по тестовой выборке $X^k = (\tilde{x}_i, \tilde{y}_i)_{i=1}^k$:
$$Q(X^k) = \frac{1}{k} \sum_{i=1}^k \bigl[\langle \tilde{x}_i, w\rangle \tilde{y}_i < 0\bigr]$$

Непрерывные верхние оценки пороговой функции потерь

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



$[M < 0]$	— пороговая функция потерь
$V(M) = (1 - M)_+$	— кусочно-линейная (SVM)
$H(M) = (-M)_+$	— кусочно-линейная (Hebb's rule)
$L(M) = \log_2(1 + e^{-M})$	— логарифмическая (LR)
$Q(M) = (1 - M)^2$	— квадратичная (FLD)
$S(M) = 2(1 + e^{M})^{-1}$	— сигмоидная (ANN)
$E(M) = e^{-M}$	— экспоненциальная (AdaBoost)

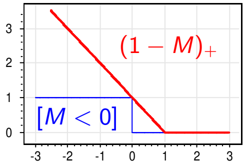
Метод опорных векторов SVM (двухклассовый)

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

Кусочно-линейная функция потерь:

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

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



Важнейшие свойства SVM:

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

Мультиколлинеарность и регуляризация в линейных моделях

Пусть построена линейная модель $\langle w, x\rangle$
Пусть имеется мультиколлинеарность: $\exists u \in \mathbb{R}^n$: $\forall x\ \langle u, x\rangle = 0$
Тогда решение не единственно: $\forall \gamma \in \mathbb{R}\ \ \langle w, x\rangle = \langle w + \gamma u, x\rangle$

Причины мультиколлинеарности:

- признаки (почти) линейно зависимы
- объектов меньше, чем признаков

Последствия мультиколлинеарности:

- слишком большие веса $|w_j|$ разных знаков
- неустойчивость модели: малое $\delta x \to$ большое $\delta\langle w, x\rangle$
- переобучение: $Q(X^k) \gg Q(X^\ell)$

Основной способ уменьшить переобучение:

- регуляризация (weight decay): $\|w\| \to \min$

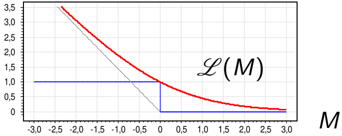
Логистическая регрессия (двухклассовая)

Линейная модель классификации $a(x,w) = \mathrm{sign}\langle x, w\rangle$
Логарифмическая функция потерь:

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

Логарифмическая функция потерь:

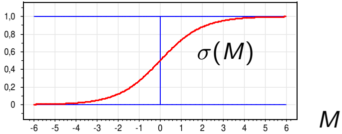
$$\mathscr{L}(M) = \ln\bigl(1 + e^{-M}\bigr)$$



Модель условной вероятности:

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

где $\sigma(M)$ — сигмоидная функция



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:

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

где $\mathrm{SoftMax}\colon \mathbb{R}^Y \to \mathbb{R}^Y$ переводит произвольный вектор в нормированный вектор дискретного распределения.

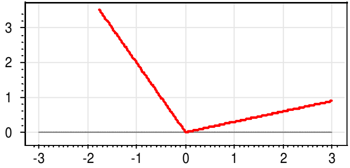
Максимизация правдоподобия (log-loss) с регуляризацией:

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

Квантúльная регрессия

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

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



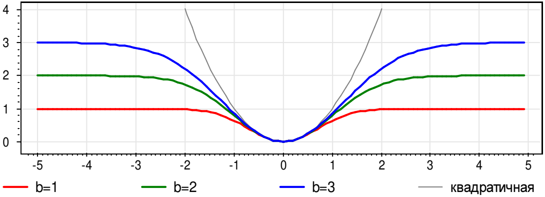
Модель регрессии: линейная $a(x_i, w) = \langle x_i, w \rangle$.

Сведение к задаче линейного программирования:
замена переменных $\varepsilon_i^+ = \bigl(a(x_i) - y_i\bigr)_+$, $\varepsilon_i^- = \bigl(y_i - a(x_i)\bigr)_+$;

$$\begin{cases} \sum\limits_{i=1}^{\ell} C_+\varepsilon_i^+ + C_-\varepsilon_i^- \to \min\limits_w; \\ \langle x_i, w \rangle - y_i = \varepsilon_i^+ - \varepsilon_i^-; \\ \varepsilon_i^+ \geqslant 0; \quad \varepsilon_i^- \geqslant 0. \end{cases}$$

Робастная регрессия

Функция Мешалкина: $\mathscr{L}(\varepsilon) = b\bigl(1 - \exp\bigl(-\frac{1}{b}\varepsilon^2\bigr)\bigr)$, $\varepsilon = a - y$



Модель регрессии: $a(x, w)$

Постановка оптимизационной задачи:

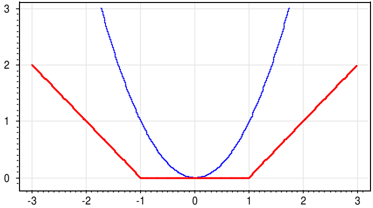
$$\sum_{i=1}^{\ell} \exp\Bigl(-\frac{1}{b}\bigl(a(x_i, w) - y_i\bigr)^2\Bigr) \to \max_w$$

Численное решение — методом Ньютона-Рафсона

SVM-регрессия

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

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



Постановка оптимизационной задачи:

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

Сводится к выпуклой задаче квадратичного программирования

Регуляризаторы, штрафующие сложность линейной модели

Регуляризатор — аддитивная добавка к основному критерию:

$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(\langle x_i, w\rangle, y_i\bigr) + \tau\, \text{штраф}(w) \;\to\; \min_{w}$$

где $\mathscr{L}(a, y)$ — функция потерь, τ — коэффициент регуляризации

L_2*-регуляризация* (гребневая регрессия, SVM):

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

L_1*-регуляризация* (LASSO, ElasticNet — для отбора признаков):

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

L_0*-регуляризация* (критерии Акаике AIC, байесовский BIC):

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

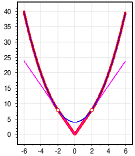
Негладкие регуляризаторы для отбора признаков

Общий вид регуляризаторов (μ — параметр селективности):

$$\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle, y_i\big) + \tau \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_w .$$

Регуляризаторы с эффектом группировки зависимых признаков:

Elastic Net: $R_\mu(\alpha) = \mu|\alpha| + \alpha^2$

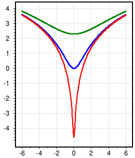


Support Features Machine (SFM):

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

Relevance Features Machine (RFM):

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



Задача обучения ранжированию (learning to rank)

Дано: $X^\ell = \{x_1, \dots, x_\ell\}$ — обучающая выборка
$i \prec j$ — правильный порядок на парах (x_i, x_j)

Найти: модель ранжирования $a\colon X \to \mathbb{R}$ такую, что

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

Критерий: число неверно упорядоченных пар (x_i, x_j)
или аппроксимированный попарный эмпирический риск:

$$\sum_{i \prec j} \bigl[a(x_j, w) < a(x_i, w)\bigr] \leqslant \sum_{i \prec j} \mathscr{L}\bigl(\underbrace{a(x_j, w) - a(x_i, w)}_{M_{ij}(w)}\bigr) \to \min_w$$

где $\mathscr{L}(M)$ — убывающая функция *парного отступа* $M_{ij}(w)$

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

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

или максимум апостериорной вероятности

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

где γ — вектор гиперпараметров априорного распределения

Задача восстановления смеси плотностей распределения

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

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

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

или максимум апостериорной вероятности

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

где γ — вектор гиперпараметров априорного распределения

Задача кластеризации (clustering)

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

Найти:
— центры кластеров $\mu_j \in \mathbb{R}^n$, $j = 1, \dots, K$
— какому кластеру принадлежит каждый объект $a_i \in \{1, \dots, K\}$

Критерий: минимум внутрикластерных расстояний

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

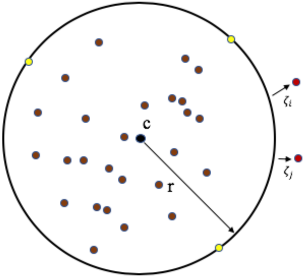
в случае евклидовой метрики

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

Метод OSVM — одноклассовый SVM

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

Найти: центр $a \in \mathbb{R}^n$ и радиус r шара, охватывающего всю выборку кроме аномальных объектов-выбросов



Критерий: минимизация радиуса шара и суммы штрафов за выход из шара:

$$\nu r^2 + \sum_{i=1}^{\ell} \mathscr{L}\bigl(\underbrace{r^2 - \|x_i - a\|^2}_{\text{margin}_i}\bigr) \to \min_{a,r}$$

При $\mathscr{L}(M) = (-M)_+$ свойства решения аналогичны SVM:

- Выпуклая задача квадратичного программирования
- Решение *разрежено* — зависит только от *опорных объектов*
- Обобщение на нелинейные модели: $\langle x_i, x_j \rangle \to K(x_i, x_j)$

Задача частичного обучения (semi-supervised learning, SSL)

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):

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

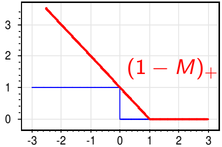
При построении модели классификации, $a_i = a(x_i, w)$:

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

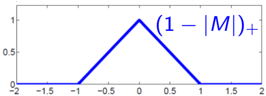
Метод TSVM — трансдуктивный SVM

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

- Функция потерь $\mathscr{L}(M) = (1 - M)_+$ штрафует за уменьшение отступа



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



Обучение весов w, w_0 по частично размеченной выборке:

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

Задачи низкорангового матричного разложения

- Понижение размерности для классификации/регрессии
- Формирование векторных представлений объектов
- Восстановление пропущенных значений в матрице

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

$$\|Z - XY\| = \sum_{(i,j)\in\Omega} \mathscr{L}\Bigl(z_{ij} - \sum_t x_{it} y_{tj}\Bigr) \to \min_{X,Y}$$

Почему на практике отказываются от классического SVD:

- неквадратичная функция потерь $\mathscr{L}$
- неотрицательное матричное разложение: $x_{it} \geqslant 0$, $y_{tj} \geqslant 0$
- 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. **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.

Дистилляция моделей или суррогатное моделирование

Обучение сложной модели $a(x, w)$ «долго, дорого»:

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

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

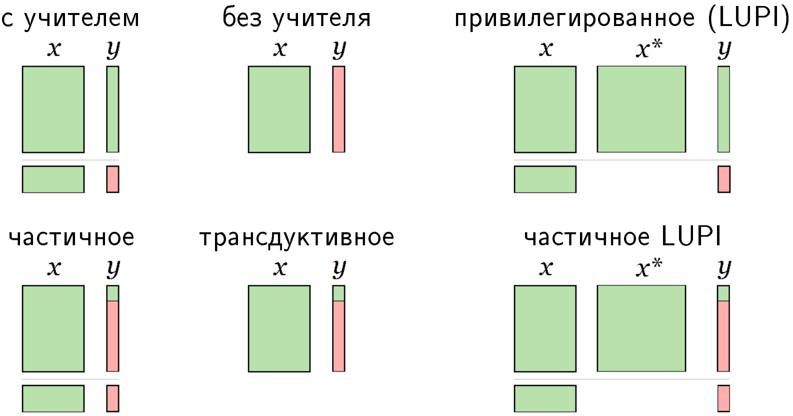
$$\sum_{i=1}^{k} \mathscr{L}\bigl(b(x'_i, w'), a(x'_i, w)\bigr) \;\to\; \min_{w'}$$

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:

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

Модель-ученик обучается повторять ошибки модели-учителя:

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

Совместное обучение модели-ученика и модели-учителя:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(a(x_i, w), y_i\big) + \lambda \mathscr{L}\big(a(x_i^*, w^*), y_i\big) + \\ + \mu \mathscr{L}\big(a(x_i, w), a(x_i^*, w^*)\big) \;\to\; \min_{w, w*}$$

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 $\mathscr{L}_i$:

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

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

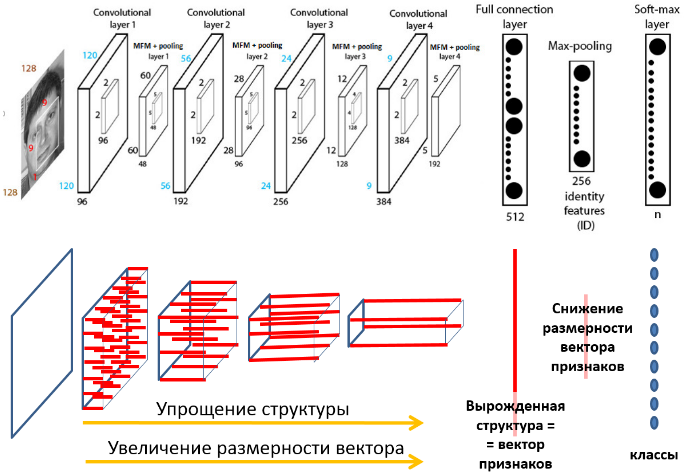
$$\sum_{i=1}^{m} \mathscr{L}_i'\bigl(f(x_i', \alpha), g'(x_i', \beta')\bigr) \to \max_{\beta'}$$

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

$$\sum_{i=1}^{m} \mathscr{L}_i'\bigl(f(x_i', \alpha), g'(x_i', \beta')\bigr) \to \max_{\alpha, \beta'}$$

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

Свёрточные сети глубокого обучения

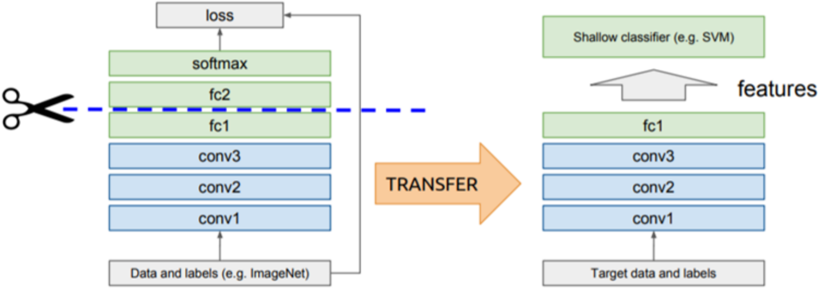


Визильтер Ю.В., Горбацевич В.С. Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей. ММРО-2017.

Пред-обученные (pre-trained) нейронные сети

Свёрточная сеть для обработки изображений:

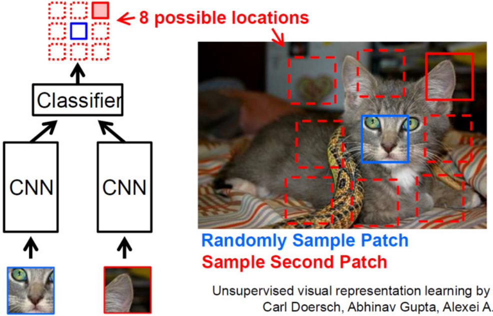
- $f(x, \alpha)$ — свёрточные слои для векторизации объектов
- $g(x, \beta)$ — полносвязные слои под конкретную задачу



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

Самостоятельное обучение (self-supervised learning)

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

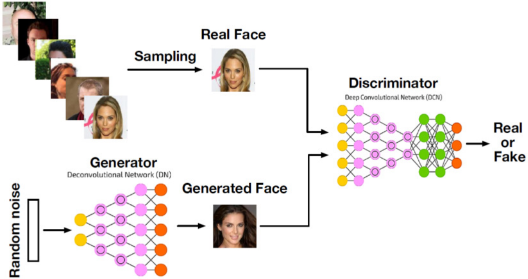


Unsupervised visual representation learning by context prediction, Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

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

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:

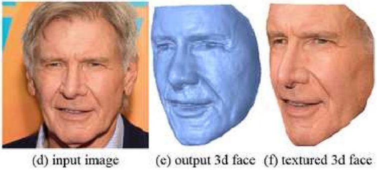
$$\sum_{i=1}^m \ln D(x_i,\beta) + \ln\bigl(1 - D(G(z_i,\alpha),\beta)\bigr) \;\to\; \max_{\beta}$$

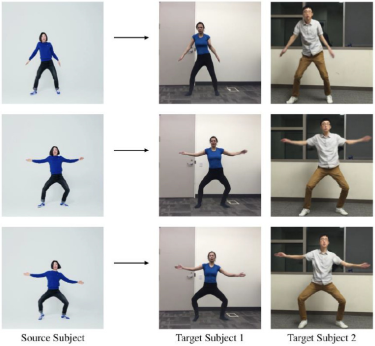
обучение генеративной модели G по случайному шуму $\{z_i\}_{i=1}^m$:

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

Ian Goodfellow et al. Generative Adversarial Nets. 2014

Примеры GAN для синтеза изображений и видео





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.

Вместо резюме. Типология задач машинного обучения

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)
   - матричные разложения (matrix factorization)
   - обучение многообразий (manifold learning)
6. Глубокое обучение (deep learning)
7. Обучение близости/связей (similarity/relational learning)
8. Обучение структуры модели (structure learning)
9. Привилегированное обучение (privileged learning, distilling)
10. Состязательное обучение (adversarial learning)
11. Динамическое обучение (online/incremental learning)
12. Активное обучение (active learning)
13. Обучение с подкреплением (reinforcement learning)
14. Перенос обучения (transfer learning)
15. Многозадачное обучение (multitask learning)
16. Мета-обучение (meta-learning, AutoML)