Overview of Machine Learning from Optimizational Point of View

Konstantin Vorontsov

CMCM 2021 • MIPT • October 27–29, 2021

Contents

General optimization problem for many Machine Learning tasks

Given: a training set of objects $\{x_i\}_{i=1}^{\ell}$

Find: parameters w of the predictive model $a(x, w)$

Minimize the empirical risk

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

where $L_i(w)$ is a loss function of the model $a(x, w)$ at the object x_i

More generally, minimize the regularized empirical risk

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

where R_j is regularization criterion, τ_j is regularization coefficient

Regression as optimization problem

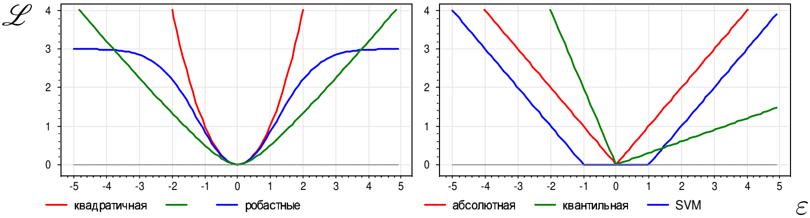
Given: a training set of objects $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \mathbb{R}$

Find: parameters w of the regression model $a(x, w)$

Minimize the empirical risk

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

Unimodal loss function $\mathscr{L}(\varepsilon)$ of the difference $\varepsilon = a(x, w) - y$:



Classification as optimization problem

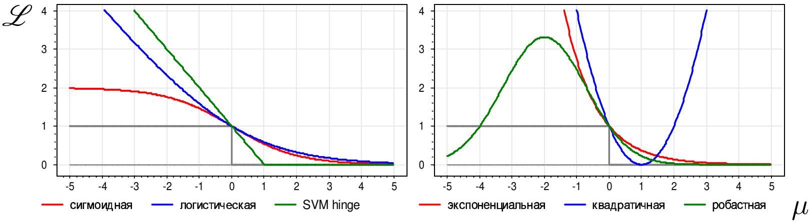
Given: a training set of objects $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in \{-1, +1\}$

Find: w of the classification model $a(x, w) = \operatorname{sign} g(x, w)$

Minimize the empirical risk

$$\sum_{i=1}^{\ell} \big[g(x_i, w) y_i < 0\big] \leqslant \sum_{i=1}^{\ell} \mathscr{L}\big(g(x_i, w) y_i\big) \to \min_{w}$$

Decreasing loss function $\mathscr{L}(\mu)$ of the margin $\mu = g(x, w) y$:



Multi-class classification as optimization problem

Given: a training set of objects $(x_i, y_i)_{i=1}^{\ell}$, $y_i \in Y$, $|Y| < \infty$

Find: w_y of the classification model $a(x, w) = \arg\max\limits_{y \in Y} g(x_i, w_y)$

The model of the class probability for a given object:

$$P(y|x, w) = \frac{\exp g(x, w_y)}{\sum\limits_{z \in Y} \exp g(x, w_z)} = \operatorname*{SoftMax}_{y \in Y} g(x, w_y), \quad y \in Y$$

where SoftMax: $\mathbb{R}^Y \to \mathbb{R}^Y$ is a smooth transformation of a vector into a normalized vector of a discrete distribution.

Maximize the log-likelihood of the data (log-loss):

$$-\sum_{i=1}^{\ell} \ln P(y_i|x_i, w) \to \min_w$$

Regularizers that penalize the complexity of a linear model

Regularizer is an additive complexity penalty to the main criterion:

$$\sum_{i=1}^{\ell} \mathscr{L}\big(\langle x_i, w\rangle, y_i\big) + \tau \, \mathrm{penalty}(w) \;\to\; \min_{w}$$

where $\mathscr{L}(a, y)$ is a loss function, τ is regularization coefficient

L_2-regularization (ridge regression, SVM):

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

L_1-regularization (LASSO, ElasticNet for feature selection):

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

L_0-regularization (Akaike/Bayes Information Criteria AIC/BIC):

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

Non-smooth regularizers for feature selection

A general form of a regularizer with selectivity parameter μ:

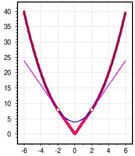
$$\sum_{i=1}^{\ell} \mathscr{L}\bigl(\langle x_i, w\rangle, y_i\bigr) + \tau \sum_{j=1}^{n} R_\mu(w_j) \;\to\; \min_{w}.$$

Regularizer with grouping effect for multi-collinear features:

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

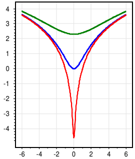
Support Features Machine (SFM):

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



Relevance Features Machine (RFM):

$$R_\mu(w) = \ln\bigl(\mu w^2 + 1\bigr)$$



Learning to Rank

Given: a training set of objects $\{x_i\}_{i=1}^{\ell}$
$i \prec j$ — partial order relation on object pairs (x_i, x_j)

Find: parameters w of the ranking model $a(x, w)$

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

Minimize number of misordered pairs (x_i, x_j) or approximated pairwise empirical risk:

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

where $\mathscr{L}(\mu)$ is a decreasing loss function of pairwise margin $\mu_{ij}(w)$

Density Estimation

Given: a training set of objects $\{x_i\}_{i=1}^{\ell}$

Find: parameters θ of the density model $p(x|\theta)$

Minimize Likelihood Estimation (MLE)

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

or Maximum A Posteriori (MAP) estimation:

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

where γ is a hyperparameter of a prior distribution

Mixture Density Estimation

Given: a training set of objects $\{x_i\}_{i=1}^{\ell}$

Find: parameters w_j, θ_j of the mixture $p(x|\theta, w) = \sum\limits_{j=1}^{K} w_j p(x|\theta_j)$

Minimize Likelihood Estimation (MLE)

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

or Maximum A Posteriori (MAP) estimation:

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

where γ is a hyperparameter of a prior distribution

Clustering

Given: a training set of objects $\{x_i \in \mathbb{R}^n \colon i = 1, \dots, \ell\}$

Find:

— centers of clusters $\mu_j \in \mathbb{R}^n$, $j = 1, \dots, K$

— what cluster $a_i \in \{1, \dots, K\}$ each object x_i pertains to

Minimize the average intra-cluster distances:

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

in the case of the Euclidean metric

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

Semi-Supervised Learning

Given: labeled $(x_i, y_i)_{i=1}^{k}$ and unlabeled $(x_i)_{i=k+1}^{\ell}$ data

Find: classification $(a_i)_{i=k+1}^{\ell}$ of unlabeled objects

Minimize the combined clustering/classification criterion:

- with no classification model (Transductive Learning):

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

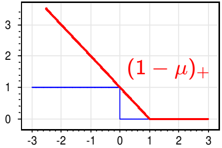
- with classification model, $a_i = a(x_i, w)$:

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

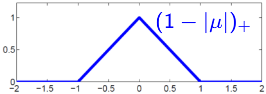
Transductive Learning of a margin-based classifier

$\mu_i(w) = g(x_i, w)y_i$ is margin of the x_i object

- loss function $\mathscr{L}(\mu) = (1-\mu)_+$ penalizes labeled objects for margin decreasing



- loss function $\mathscr{L}(\mu) = \bigl(1-|\mu|\bigr)_+$ penalizes unlabeled objects for falling into the gap between classes



Minimize the combined clustering/classification criterion:

$$\sum_{i=1}^{k} \bigl(1-\mu_i(w)\bigr)_+ + \gamma \sum_{i=k+1}^{\ell} \bigl(1-|\mu_i(w)|\bigr)_+ \to \min_w$$

Low-rank matrix factorization

- Generation of better feature vector representation of objects
- Recovering missing values in a matrix

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

Find: matrixes $X = \|x_{it}\|_{n\times k}$ и $Y = \|y_{tj}\|_{k\times m}$

Minimize

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

Why the classic SVD is abandoned in practice:

- non-square loss function $\mathscr{L}$
- non-negative matrix factorization: $x_{it} \geqslant 0$, $y_{tj} \geqslant 0$
- sparse data: $|\Omega| \ll nm$
- orthogonality is unnecessary or not interpretable

Autoencoders: unsupervised learning

Given a training set of objects $\{x_i\}_{i=1}^{\ell}$

Find:

encoder $f\colon X \to Z$ that produces code vector $z = f(x, \alpha)$
decoder $g\colon Z \to X$ that reconstructs vector $\hat{x} = g(z, \beta)$ from z

Minimize

the reconstruction error under square loss $\mathscr{L}(\hat{x}, x) = \|\hat{x} - x\|^2$:

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

Examples of autoencoders:

$f(x, A) = \underset{m\times n}{A}\, x, \quad g(z, B) = \underset{n\times m}{B}\, z$ — linear

$f(x, A) = \sigma(Ax), \quad g(z, B) = \sigma(Bz)$ — neural

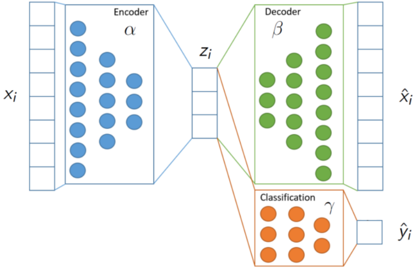
Autoencoders for supervised learning

Given labeled $(x_i, y_i)_{i=1}^{k}$ and unlabeled $(x_i)_{i=k+1}^{\ell}$ data

Find:

$z_i = f(x_i, \alpha)$ — encoder
$\hat{x}_i = g(z_i, \beta)$ — decoder
$\hat{y}_i = \hat{y}(z_i, \gamma)$ — predictor

Loss function:
$\mathscr{L}(\hat{x}_i, x_i)$ — for reconstruction
$\tilde{\mathscr{L}}(\hat{y}_i, y_i)$ — for prediction



Minimize the combined reconstruction/prediction criterion:

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

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

Graph Factorization

Given a set $(i,j) \in E$ of edges of the graph $\langle V, E\rangle$,
similarities S_{ij} between vertices of the edge (i,j)

For example, $S_{ij} = [(i,j) \in E]$ is binary adjacency matrix

Find: vector representation (embedding) of vertices such that adjacent vertices would have similar vectors

Minimize the reconstruction error of graph edges:

- in the case of undirected graph and symmetric S matrix

$$\sum_{(i,j)\in E} \bigl(\langle z_i, z_j\rangle - S_{ij}\bigr)^2 \to \min_{Z}, \quad Z \in \mathbb{R}^{V\times d}$$

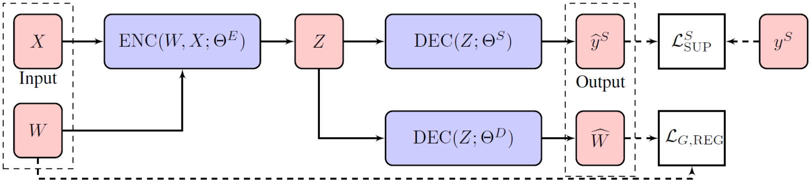
- in the case of directed graph and asymmetric S matrix

$$\sum_{(i,j)\in E} \bigl(\langle \varphi_i, \theta_j\rangle - S_{ij}\bigr)^2 \to \min_{\Phi,\Theta}, \quad \Phi, \Theta \in \mathbb{R}^{V\times d}$$

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

GraphEDM: a big family of graph autoencoders

Graph Encoder Decoder Model generalizes more than 30 models:



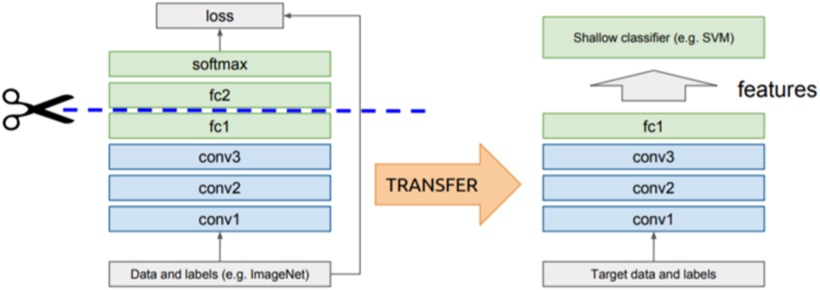
$W \in \mathbb{R}^{V \times V}$ is input data about edges
$X \in \mathbb{R}^{V \times n}$ is input feature data about vertices
$Z \in \mathbb{R}^{V \times d}$ is vector representation (embedding) of vertices
$\mathrm{DEC}(Z; \Theta^D)$ is decoder reconstructing the edge data
$\mathrm{DEC}(Z; \Theta^S)$ is decoder solving an applied supervised task
y^S is (semi-)supervised data about vertices or edges
$\mathcal{L}$ is loss function

I.Chami et al. Machine learning on graphs: a model and comprehensive taxonomy. 2020.

Pre-training of neural networks

Convolutional Neural Network (CNN) for image classification:

- $z = f(x, \alpha)$ is convolutional layers for image vectorization
- $y = g(z, \beta)$ is feedforward layers for vector classification



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

Transfer learning

$f(x, \alpha)$ is the universal part of the model (object vectorization)
$g(x, \beta)$ is a specific part of the model targeted for an applied task

Base task on a dataset $\{x_i\}_{i=1}^{\ell}$ with loss $\mathscr{L}_i$:

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

Target task on another dataset $\{x_i'\}_{i=1}^{m}$, with another $\mathscr{L}_i'$, g':

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

if $m \ll \ell$ then pre-training vectorizer $f(x_i, \alpha)$ could be better than

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

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

Multi-task learning

$f(x, \alpha)$ is the universal part of the model (object vectorization)
$g_t(x, \beta)$ is a specific part of the model targeted for the task $t \in T$

Joint training of the model f from datasets X_t of tasks $t \in T$:

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

The property of *learnability*: we learn the task $\langle X_t, \mathscr{L}_t, g_t \rangle$ better by augmenting data size $|X_t|$

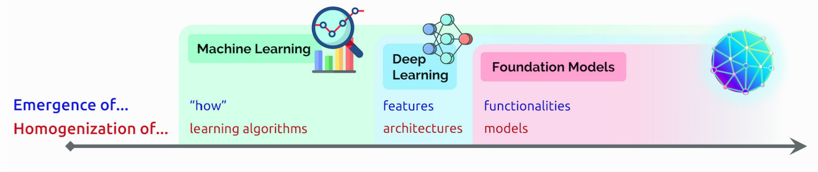
Learning to learn: we learn each of the tasks $\langle X_t, \mathscr{L}_t, g_t \rangle$ better by augmenting the number of tasks $|T|$

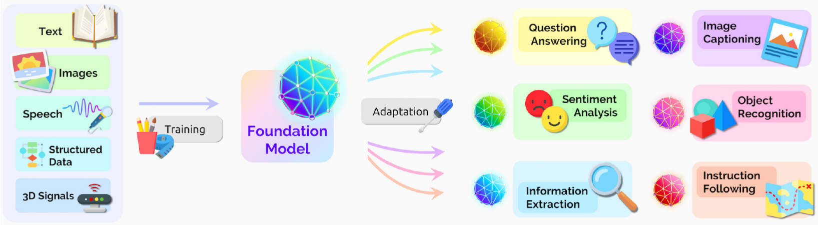
Few-shot learning: to solve the problem t, a small number of examples may be enough, sometimes even one

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

Foundation Models

Multi-task learnable data vectorization is a recent trend in AI/ML





R.Bommasani et al. (Center for Research on Foundation Models, Stanford University)
On the opportunities and risks of foundation models // CoRR, 20 August 2021.

Distillation and surrogate modeling

Learning a resource intensive heavy model $a(x, w)$

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

Learning a light model $b(x, w')$, possibly on other dataset

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

Examples:

- approximation of a heavy model, which is calculated on a supercomputer for months (climate, aerodynamics, etc.), by a light surrogate model
- approximation of a heavy neural network which learns for weeks on big data, by a light neural network with fewer neurons and connections

Learning Using Privileged Information (LUPI)

x_i^* — information about x_i available only for training

Student model and teacher model are learned separately:

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

Student model learns from responses of the teacher model:

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

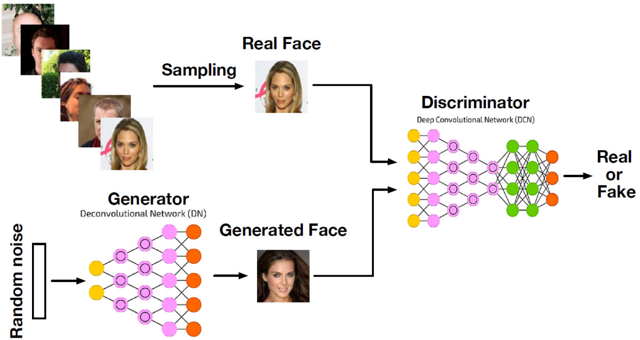
Student model and teacher model are learned together:

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

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

Generative Adversarial Net (GAN)

Generator $G(z)$ learns to generate realistic objects x from noise z
Discriminator $D(x)$ learns to distinguish is object real or fake



Antonia Creswell et al. Generative Adversarial Networks: an overview. 2017.
Zhengwei Wang et al. 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.

Generative Adversarial Net (GAN)

Given a training set of objects $\{x_i\}_{i=1}^{\ell}$

Find two probabilistic models:

- model $G(z, \alpha)$ generates $x \sim p(x|z, \alpha)$ from noise z
- model $D(x, \beta) = p(1|x, \beta)$ recognizes if object x is real

Minimax in the antagonistic game of generator vs. discriminator:

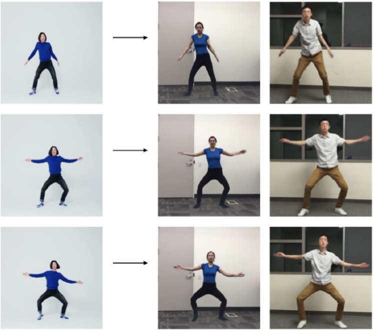
- *discriminator* $D(x, \beta)$ learns to maximize log-likelihood in order to better distinguish real object x from the fake one
- *generator* $G(z, \alpha)$ learns to minimize log-likelihood in order to generate realistic objects x

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

Ian Goodfellow et al. Generative Adversarial Nets. 2014

Examples of how GAN generates realistic image and video

(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-2019.

Optimization criteria induce a typology of ML tasks

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)
   - восстановление плотности (density estimation)
   - поиск ассоциативных правил (association rule learning)
   - одноклассовая классификация (anomaly detection)
4. Частичное обучение (semi-supervised learning)
   - трансдуктивное обучение (transductive learning)
   - обучение с положительными примерами (PU-learning)

Optimization criteria induce a typology of ML tasks

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. Перенос обучения (transfer learning)
9. Многозадачное обучение (multitask learning)
10. Привилегированное обучение (privileged learning, distilling)
11. Состязательное обучение (adversarial learning)
12. Обучение структуры модели (structure learning)
13. Динамическое обучение (online/incremental learning)
14. Активное обучение (active learning)
15. Обучение с подкреплением (reinforcement learning)
16. Мета-обучение (meta-learning, AutoML)