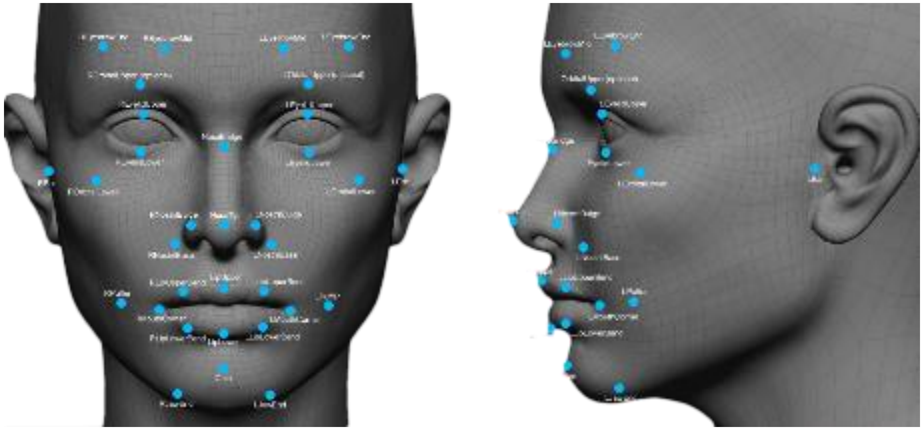


Seminar # 2

Seminar # 2 of MIPT course: Introduction to machine learning

Speaker: Alexey O. Seleznev
PhD in Computational Chemistry

5VISION TEAM

Moscow
2017

OUTLINE

- Discussion on the assignment results
  - Assignment 2
  - Assignment 3
- Hot topics in machine learning
- Autoencoders
- Deep generative models
  - Variational autoencoders
  - Generative adversarial networks
- References

Assignment Results

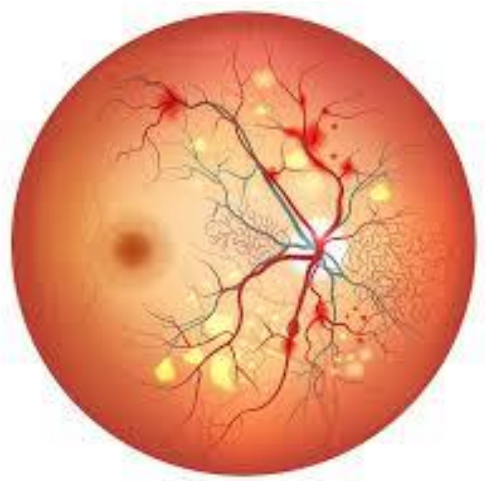
Assignment 2

❖ Leaderboard

1	–	Igor Prusov	0.30555	7	Mon, 15 May 2017 00:21:03
2	↑1	pepe	0.30373	4	Mon, 22 May 2017 10:24:49
		Baseline	0.15307		

❖ Solutions

- ➢ Tell about your solution
- ➢ How did you get to it?
- ➢ Was it an interesting task?



Assignment 3

- Leaderboard

1	new	SergeyVolodin	0.78512	4	Tue, 23 May 2017 15:16:46
2	new	pepe	0.71158	2	Mon, 22 May 2017 13:22:32 (-2.5h)
		Baseline	0.57282		

- Solutions
  - Tell about your solution
  - How did you get to it?
  - Was it an interesting task?

Hot Topics in Machine Learning

Hot Topics in Machine Learning

- Deep learning
  - Reasons for being hot:
    - State-of-the-art results in some applications (image recognition, neural translation, speech recognition, etc)
    - Concept of end-to-end learning (no need for hand-craft features)
    - Neurobiological inspiration
  - Examples of research topics:
    - Understanding of learning representations and learning process in general (including invention of new training techniques)
    - Learning a two-way transformation of the data into a space where variables are disentangled
    - Deep generative models
    - Models of long-term dependencies

Hot Topics in Machine Learning

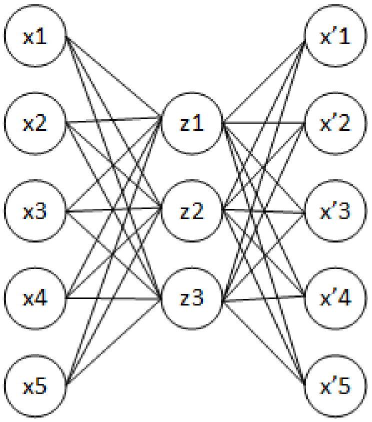
- Semi-supervised learning (a lot of unlabeled data; how to use it efficiently)
- Machine learning and reasoning (correlation vs causation)
- Approximate Bayesian inference
- Exploration vs exploitation dilemma in reinforcement learning (e.g. Bayesian reinforcement learning)
- Online learning
- Backpropagation through random operations
- Multi-agent systems

Autoencoders

Autoencoders

❖ Definition

➢ An autoencoder is a neural network that is trained to copy its input to its output



$$L(\boldsymbol{x}, g(f(\boldsymbol{x})))$$

❖ Our goal is to learn a hidden representation z of data

❖ Types of autoencoders (how to prevent learning the identical transform.)

➢ Undercomplete autoencoders (less z than x)

➢ Sparse autoencoders

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h})$$

Autoencoders

- Denoising Autoencoders (corrupted input)

$$L(\boldsymbol{x}, g(f(\tilde{\boldsymbol{x}})))$$

- Autoencoders penalizing derivatives (contractive autoencoders)

$$L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \lambda \sum_i ||\nabla_{\boldsymbol{x}} h_i||^2$$

  - Regularizer of this form forces the model to learn features that does not change much when x changes slightly
  - It allows finding local directions that are almost invariant to changes in input data (embeddings)

Autoencoders

- Applications
  - Dimensionality reduction
  - Manifold learning (especially contractive autoencoders)
  - Semantic hashing
  - Pre-training

Deep Generative Models

Deep Generative Models

- Generative vs Discriminative models
  - Discriminative models learn the (hard or soft) boundary between classes
  - Generative models model the distribution of individual classes

	Frequentist	Bayesian
Discriminative	$p(z\,;\,w,\beta)$	$p(z,\beta\,;\,w) = p(z \mid \beta\,;\,w) * p(\beta)$
Generative	$p(z,w\,;\,\beta)$	$p(z,w,\beta) = p(z,w \mid \beta) * p(\beta)$

z – category, w – input, beta - parameters

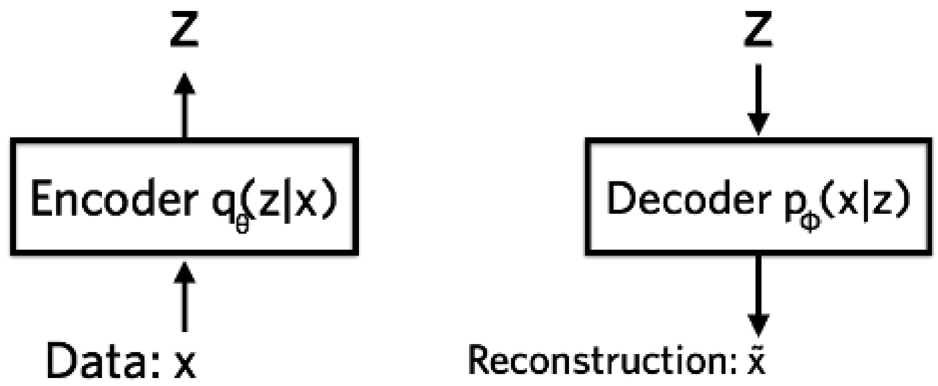
  - If generative models learn the full probability, they can restore the missing part (e.g. the input)

Deep Generative Models

- Examples of generative models:
  - Naïve Bayesian classificator
  - Linear and quadratic discriminant analysis
  - Bolzmann machines
  - Variational autoencoder
  - Generative adversarial networks

Variational Autoencoder

❖ Idea



➢ Let us maximize the lower bound of data probability

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log p_{\text{model}}(\boldsymbol{z}, \boldsymbol{x}) + \mathcal{H}(q(\mathbf{z} \mid \boldsymbol{x})) \\ &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{z}) - D_{\text{KL}}(q(\mathbf{z} \mid \boldsymbol{x}) || p_{\text{model}}(\mathbf{z})) \\ &\leq \log p_{\text{model}}(\boldsymbol{x}).\end{aligned}$$

➢ In contrast to a common autoencoder, here we have a stochastic unit with elements z

Goodfellow et al (book)
Kingma et al (2013)

Variational Autoencoder

- To get working equations, we have to suggest the form of distributions (e.g. Gaussian for z and Bernoulli for x)
- Reparametrization trick is necessary to allow gradient to go through the stochastic unit:

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \rightsquigarrow \quad \mathbf{z} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$$

- Two components in the loss function: reconstruction error and KL divergence

❖ Thus, variational autoencoder is composed of encoder and decoder trained simultaneously.

❖ To generate new data, we take random z points and project them to output by means of decoder

Variational Autoencoder

❖ Numbers generated by a Variational autoencoder

Generative Adversarial Networks

- Idea
  - GAN is based on a game theoretic scenario in which the generator competes with an adversary
  - Generator produces samples $\boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}^{(g)})$
  - Discriminator tries to distinguish between samples drawn from the training data and samples drawn from the generator
  - Discriminator estimates the probability $d(\boldsymbol{x}; \boldsymbol{\theta}^{(d)})$ of being a real sample

$$v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log d(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{model}}} \log (1 - d(\boldsymbol{x}))$$

$$g^* = \arg\min_g \max_d v(g, d)$$

Generative Adversarial Networks

❖ Examples of generated images

Goodfellow et al (book)

- Any questions about the course?

References

- I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. The MIT Press 2016
- D. Kingma, D. Rezende, S. Mohamed, M. Welling. *Semi-supervised learning with deep generative models*. In NIPS’2014