

Statistical Machine Learning (BE4M33SSU)

Seminar 1. Machine Learning Examples

Czech Technical University in Prague

Example 1: ImageNet (visual object classification)

ImageNet Challenge



- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



4

- ◆ **Training set:** $\mathcal{T}^m = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \dots, m\}$, where:
 - \mathcal{X} are images from ImageNet,
 - \mathcal{Y} is a set of output classes (ILSVRC 2012 defines $|\mathcal{Y}| = 1000$ of them).

Example 1: ImageNet (visual object classification)

	input	conv3-64	conv3-64	MP	conv3-128	conv3-128	MP	conv3-256	conv3-256	conv3-256	MP	conv3-512	conv3-512	conv3-512	MP	conv3-512	conv3-512	conv3-512	MP	FC - 4096	FC - 4096	FC - 1000	softmax
parameters		1.7k	37k		74k	147k		295k	590k	590k		1.2M	2.4M	2.4M		2.4M	2.4M	2.4M		103M	16.7M	4M	
activations	150k	3.2M	3.2M	800k	1.6M	1.6M	400k	800k	800k	800k	200k	400k	400k	400k	100k	100k	100k	100k	25k	4096	4096	1000	1000
	224 x 224 x 3	224 x 224 x 64		112 x 112 x 64	112 x 112 x 128		56 x 56 x 128	56 x 56 x 256		28 x 28 x 256	28 x 28 x 512		14 x 14 x 512	14 x 14 x 512		7 x 7 x 512			1 x 1 x 4096	1 x 1 x 4096	1 x 1 x 1000		

- ◆ **Class of prediction strategies:** VGGNet (Zisserman, et al., 2014), i.e. a convolutional neural network with fixed structure. Note that a convolutional neural network with p layers is a function composition $h_{\theta}(x) = (f_{\theta_p}^p \circ f_{\theta_{p-1}}^{p-1} \circ \dots \circ f_{\theta_1}^1)(x)$. Its outputs are interpreted as class probabilities.
- ◆ **Loss function:** negative log-likelihood of class probabilities (a.k.a. cross entropy)

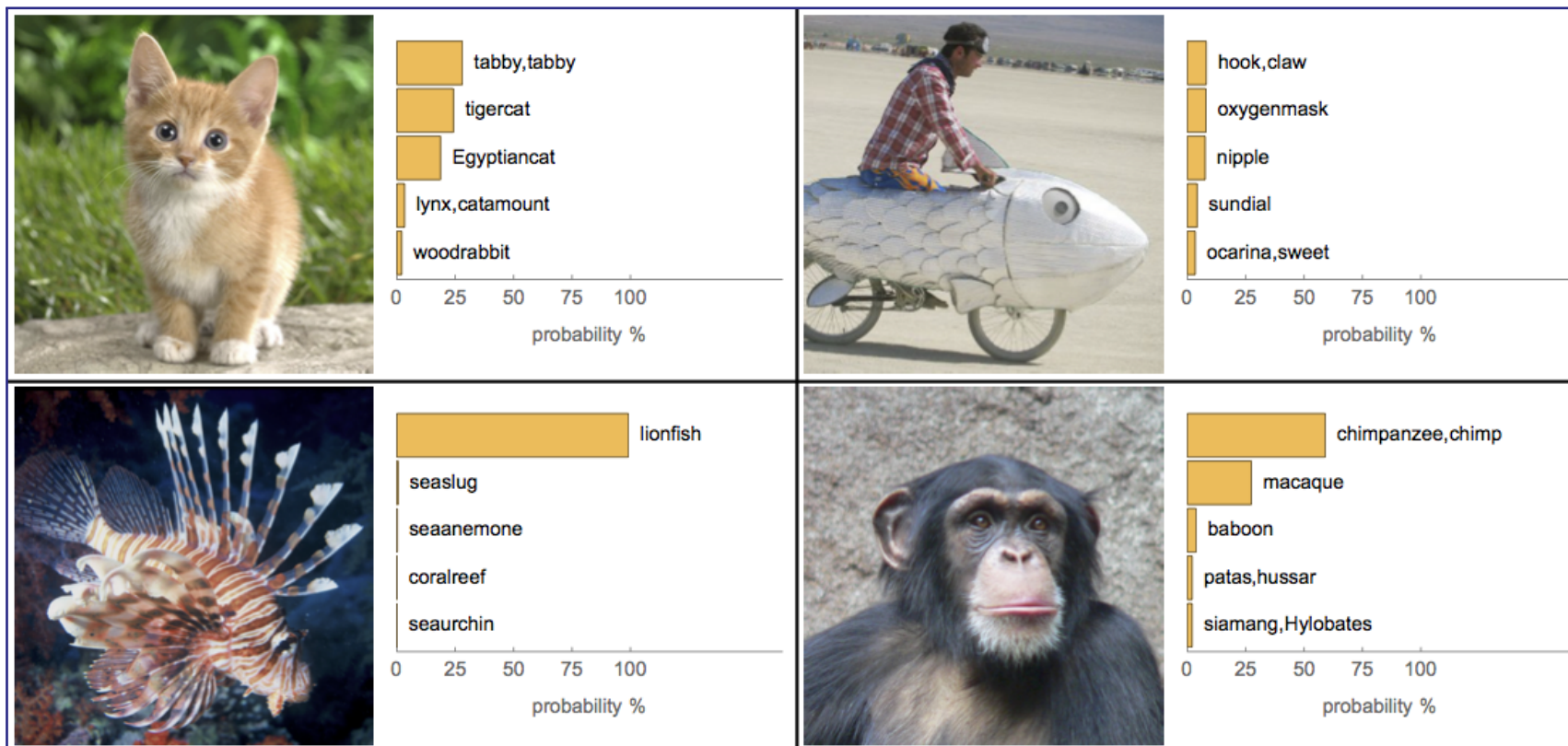
$$\ell(y^i, h(x^i)) = - \sum_{c \in \mathcal{Y}} \mathbb{I}\{y^i = c\} \log(h_c(x^i)).$$

- ◆ **Learning approach:** empirical risk minimisation, gradient descent

$$R_{\mathcal{T}^m}(\theta) = \frac{1}{m} \sum_{i=1}^m \ell(y^i, h_{\theta}(x^i)) \rightarrow \min_{\theta}$$

Example 1: ImageNet (visual object classification)

◆ Results by VGGNet



◆ More details in lectures on **deep learning**

Example 2: licence plate recognition



Online app estimating the Travel Time for cars in Prague based on the number plate recognition: <https://unicam.camea.cz/Discoverer/TravelTime3/map>

Example 2: licence plate recognition

Input image $x \in \mathcal{X}$ of size $[H \times W]$



Model of synthetic license plate images

A set of templates $w = (w_a | a \in \mathcal{A})$ for each character from \mathcal{A}



A **segmentation** $y = (s_1, \dots, s_L) \in \mathcal{Y}(x)$, where $s = (a, k)$, $a \in \mathcal{A}$ is a character code and $k \in \{1, \dots, W\}$ is a character position, together with templates w defines a synthetic image:



An admissible segmentation $y \in \mathcal{Y}(x)$ ensures that the templates are not overlapping and that the synthetic image has the same width as the input image x :

$$k(s_1) = 1, W = k(s_L) + \omega(s_L) - 1, \text{ and } k(s_i) = k(s_{i-1}) + \omega(s_{i-1}), \forall i > 1$$

where $\omega: \mathcal{A} \rightarrow \mathcal{N}$ are widths of the templates.

Example 2: licence plate recognition

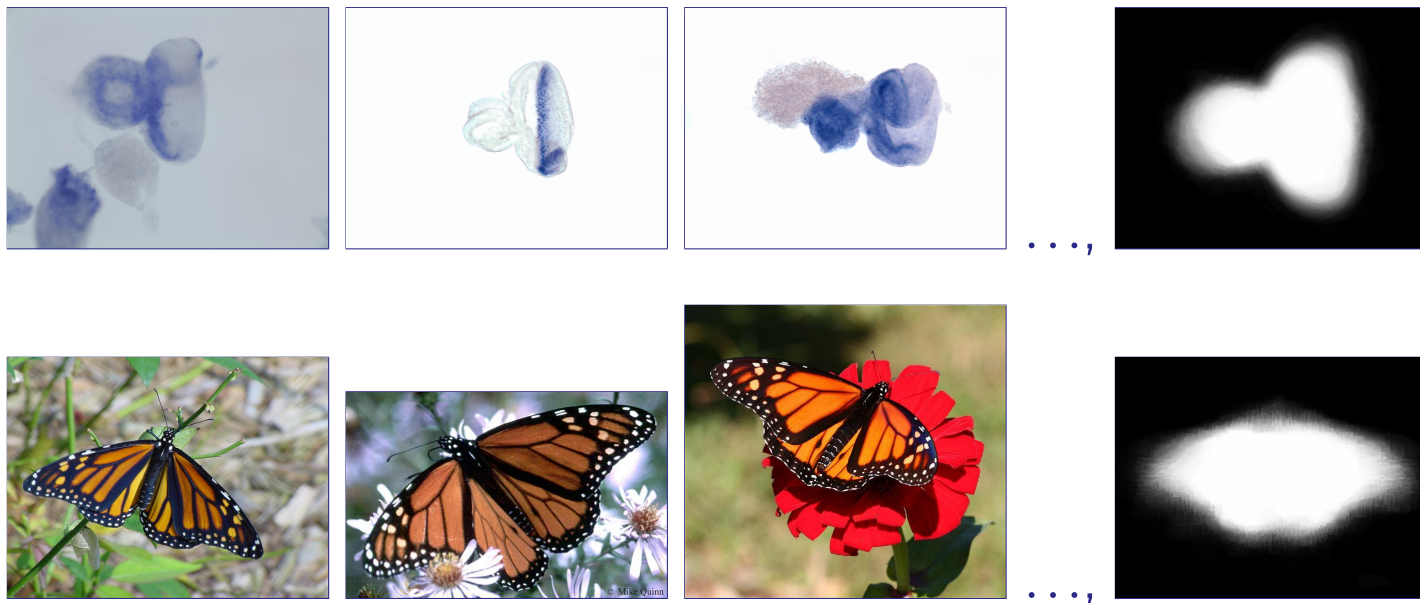
- ◆ We want a classifier which outputs the segmentations $y \in \mathcal{Y}(x)$ defining a synthetic image most similar (measured by correlation) to the input image x :

$$\hat{y} = h(x; w) = \arg \max_{(s_1, \dots, s_L) \in \mathcal{Y}(x)} \underbrace{\sum_{i=1}^{L(y)} \sum_{j=1}^{\omega(a(s_i))} \left\langle \text{col}(x, j + k(s_i) - 1), \text{col}(w_{a(s_i)}, j) \right\rangle}_{\left\langle \text{■ ULK 68-39 ■}, \text{■ ULK 68-39 ■} \right\rangle}$$

- ◆ **Problem:** How to construct the templates $w = \{w_a | a \in \mathcal{A}\}$ so that the classifier $h(x; w)$ predicts a segmentation with small Hamming distance to the correct one ?
- ◆ **Solution:** Select the templates w so that the classifier $h(x; w)$ performs well on a training set $\{(x^1, y^1), \dots, (x^m, y^m)\}$ and simultaneously control the over-fitting.
- ◆ More details in the lecture on **Structured Output Support Vector Machines**.

Example 3: Joint segmentation & registration

Given: set of images, each containing an object instance, and a shape model



Task: segment & register each image to the reference frame (shape model)

- ◆ image $\mathbf{x} = \{x_i \in \mathbb{R}^3 \mid i \in D'\}$, binary segmentation $\mathbf{y} = \{y_i \in \{0, 1\} \mid i \in D\}$
- ◆ shape model $p(\mathbf{y}) = \prod_{i \in D} p_i(y_i)$, with binomial distributions $p_i(y_i = 0, 1)$.
- ◆ appearance model $p_{\theta}(x_j \mid (T\mathbf{y})_j)$, $j \in D'$, where
 - T is an affine transformation,
 - $p_{\theta_0}(x_j \mid y'_j = 0)$, $p_{\theta_1}(x_j \mid y'_j = 1)$ are two mixtures of Gaussians.

Example 3: Joint segmentation & registration

◆ loss function $\ell(\mathbf{y}, \mathbf{y}') = \sum_{i \in D} \mathbb{I}\{y_i \neq y'_i\}$, i.e. Hamming distance

(1) Segmentation for **known** T and θ : minimise expected Hamming distance between true and estimated segmentation \Rightarrow

$$\mathbf{y} = h_{T, \theta}(\mathbf{x}) = \{h_i(\mathbf{x}) \mid i \in D\}$$

$$h_i(\mathbf{x}) = \arg \max_{y_i=0,1} p_{\theta}((T^{-1}\mathbf{x})_i \mid y_i) \cdot p_i(y_i)$$

(2) How to estimate unknown T and θ ? See lecture on the **EM-Algorithm**.