Machine learning notes

Sadržaj

Notation

Objects

Variables are generally denoted by italic serif letters. Most constants are denoted by upright serif letters. Random variables are underlined. Vectors and sequences are denoted by lowercase bold letters. Matrices and multidimensional-arrays are denoted by uppercase bold letter. Sets are denoted by uppercase blackboard-bold letters. Latin or Greek letters can be used for any type of object.

a, A, θ	Variable (commonly scalar or function)
$\boldsymbol{a}$, $\boldsymbol{\theta}$	Vector or sequence (commonly column vector)
$\boldsymbol{A}$, $\boldsymbol{\Theta}$	Matrix or multidimensional array
$\mathbb{A}$	Set or multiset
a, A, $\uptheta$	Constant
$\mathbf{a}$, $\boldsymbol{\uptheta}$	Vector or sequence constant
$\mathbf{A}$, $\mathbf{\Theta}$	Matrix or multidimensional array constant
$\mathbb{A}$	Set constant
$\underline{a}$, $\underline{A}$, $\underline{\theta}$	Random variable
$\underline{\boldsymbol{a}}$, $\underline{\boldsymbol{\theta}}$	Random vector or sequence
$\underline{\boldsymbol{A}}$, $\underline{\boldsymbol{\Theta}}$	Random matrix or multidimensional array
$\underline{\mathbb{A}}$	Random set or multiset
a, riječ	Textual label not representing an object

Constants

$\{\}$	Enpty set
e	The number that satisfies $\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^x = \mathrm{e}^x$
$\mathbf{0}$	Null-vector
$\mathbf{e}_i$	i-th canonical basis vector
$\mathbf{1}$	The sum of all canonical basis vectors
$\mathbf{I}$, $\mathbf{I}_n$	Identity matrix (with n rows/columns)
$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$	A standard set

$\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$	The set of all non-negative/positive real numbers

Defining sets and arrays

$a..b$	Shorthand notation for $a, .., b$
$\{a..b\}$	A subset of integers from a to b
$\{f(a)\colon P(a)\}$, $\{f(a)\}_{P(a)}$	A set with elements defined by a function f and a predicate P
$\{f(a)\}_a$	A set with elements defined by a function f and variables a from an implicitly defined set
$\{a_1 .. a_n\}$, $\{a_i\}_{i=1..n}$	A set with n elements
$[x_1, .., x_n]$	A row vector
$[a_i]_i, [a_{i,j}]_{i,j}, [a_{i,j,k}]_{i,j,k}$	A multidimensional array with an implicit or undefined number of elements
$[a, b)$	A semi-closed interval

Subscript and superscript

U donjem i gornjem indeksu oznake mogu biti oznake drugih matematičkih objekata ili slova ili riječi koje ne predstavljaju matematičke objekte. Redni brojevi elemenata vektora ili višedimenzionalnih nizova se, ako nije određeno drugačije, pišu u donjem indeksu oznake vektora u uglatim zagradama. Npr. i-ti element vektora $\boldsymbol{a} = [a_1, .., a_n]^\mathsf{T}$ je $\boldsymbol{a}_{[i]} = a_i$. Indeksi kod n-dimenzionalnih nizova mogu biti i vektori iz $\mathbb{N}^n$, ili kombinacije vektora manje dimenzije sa skalarima.

a_d^g	Varijabla s oznakama u donjem i gornjem indeksu
$\boldsymbol{a}_{[i]}$	i-ti element vektora $\boldsymbol{a}$
$\boldsymbol{a}_{[i_1:i_2]}$	Vektor kojeg čine elementi $\boldsymbol{a}_{[i_1]}, \boldsymbol{a}_{[i_1+1]}, .., \boldsymbol{a}_{[i_2]}$
$\boldsymbol{a}_{[(i_1 .. i_n)]}$	Vektor kojeg čine elementi $\boldsymbol{a}_{[i_1]}, \boldsymbol{a}_{[i_2]}, .., \boldsymbol{a}_{[i_n]}$
$\boldsymbol{A}_{[i,j]}$	Element i, j matrice $\boldsymbol{A}$
$\boldsymbol{A}_{[i,:]}$	i-ti redak matrice $\boldsymbol{A}$
$\boldsymbol{A}_{[:,i_1:i_2,j]}$	2-D odsječak 3-D niza $\boldsymbol{A}$
$\boldsymbol{A}_{[\boldsymbol{i}]}$	Element $\boldsymbol{A}_{\left[\boldsymbol{i}_{[1]}, .., \boldsymbol{i}_{[n]}\right]}$ n-D niza

$\boldsymbol{A}_{[\boldsymbol{i}_1:\boldsymbol{i}_2]}$	Podniz $\boldsymbol{A}_{\left[\boldsymbol{i}_{1\,[1]}:\boldsymbol{i}_{2\,[1]},..,\boldsymbol{i}_{1\,[n]}:\boldsymbol{i}_{2\,[n]}\right]}$ n-D niza
$\boldsymbol{A}_{[\boldsymbol{i}_1:\boldsymbol{i}_2;:]}$	Podniz $\boldsymbol{A}_{\left[\boldsymbol{i}_{1\,[1]}:\boldsymbol{i}_{2\,[1]},..,\boldsymbol{i}_{1\,[n-1]}:\boldsymbol{i}_{2\,[n-1]},:\right]}$ n-D niza

Operacije linearne algebre i operacije s nizovima

$\langle \boldsymbol{a} \mid \boldsymbol{b} \rangle$, $\boldsymbol{a}^\mathsf{T}\boldsymbol{b}$	Skalarni produkt
$\boldsymbol{a}\boldsymbol{b}^\mathsf{T}$	Vanjski produkt
$\boldsymbol{a} \odot \boldsymbol{b}$	Umnožak po elementima; Hadamardov produkt
$\boldsymbol{a} \oslash \boldsymbol{b}$	Dijeljenje po elementima
$\boldsymbol{a}^{\odot b}$	Potenciranje po elementima
$\boldsymbol{A}\boldsymbol{B}$	Matrično množenje
$\boldsymbol{A}^{-1}$	Inverz matrice
$\boldsymbol{A}^\mathsf{T}$	Transponiranje
$\operatorname{diag}(\boldsymbol{a})$	Dijagonalna matrica kojoj dijagonalu čini vektor $\boldsymbol{a}$
$\det(\boldsymbol{A})$	Determinanta matrice $\boldsymbol{A}$
$\lVert \boldsymbol{a} \rVert_2$	L^2-norma vektora $\boldsymbol{a}$
$\lVert \boldsymbol{a} \rVert_p$	L^p-norma vektora $\boldsymbol{a}$
$\lVert \boldsymbol{A} \rVert_p$	Matrična L^p-norma matrice $\boldsymbol{A}$
$\lVert \boldsymbol{A} \rVert_\mathsf{F}$	Frobeniusova norma matrice $\boldsymbol{A}$
$\boldsymbol{a} \mathbin{+\!\!+} \boldsymbol{b}$	Konkatenacija vektora (stupaca) $\boldsymbol{a} \in \mathbb{R}^n$ i $\boldsymbol{b} \in \mathbb{R}^m$ u vektor iz $\mathbb{R}^{n+m}$
$\boldsymbol{A} \mathbin{+\!\!+} \boldsymbol{B}$	Konkatenacija nizova po prvoj dimenziji
$\operatorname{vec}(\boldsymbol{A})$	Funkcija koja preslikava niz iz $\mathbb{R}^{d_1 \times \cdots \times d_n}$ u $\mathbb{R}^{d_1 \cdots d_n}$
$\dim(\boldsymbol{a})$	Dimenzija vektora
$\dim(\boldsymbol{A})$	Vektor dimenzija niza; $[d_1,..,d_n]$ za $\boldsymbol{A} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$

Diferencijalni račun

$\frac{\mathrm{d}y}{\mathrm{d}x}$, $\frac{\mathrm{d}}{\mathrm{d}x}f(x)$	Derivacija $y = f(x)$ po x

$\frac{\partial y}{\partial x}$, $\frac{\partial}{\partial x} f(x)$	Parcijalna derivacija $y = f(x)$ po x
$\nabla_{\boldsymbol{x}} y$, $\nabla_{\boldsymbol{x}} f(x)$, $\left(\frac{\partial y}{\partial \boldsymbol{x}}\right)^{\mathsf{T}}$	Gradijent $y = f(\boldsymbol{x})$ po $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$, $\nabla_{\boldsymbol{X}} f(x)$	Gradijent $y = f(\boldsymbol{x})$ po $\boldsymbol{X}$
$\frac{\partial^2 y}{\partial \boldsymbol{x} \partial \boldsymbol{x}^{\mathsf{T}}}$, $\boldsymbol{H}_f(\boldsymbol{x})$, $\boldsymbol{H}$	Hessijan iz $\mathbb{R}^{n \times n}$ za $f \colon \mathbb{R}^n \to \mathbb{R}$ i $y = f(\boldsymbol{x})$
$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$, $\boldsymbol{J}_f(\boldsymbol{x})$, $\boldsymbol{J}$	Jakobijeva matrica iz $\mathbb{R}^{m \times n}$ za $f \colon \mathbb{R}^n \to \mathbb{R}^m$ i $\boldsymbol{y} = f(\boldsymbol{x})$
$\int_{\mathbb{A}} f(x) \, \mathrm{d}x$, $\int_{x \in \mathbb{A}} f(x)$	Određeni integral funkcije $f(x)$ po $x \in \mathbb{A}$
$\int f(x) \, \mathrm{d}x$, $\int_x f(x)$	Određeni integral funkcije $f(x)$ po $x \in \mathbb{A}$, gdje je $\mathbb{A}$ implicitan

Teorija vjerojatnosti

Svakoj slučajnoj varijabli $\underline{a}$ jednoznačno je dodijeljena jedna razdioba $\mathrm{p}(\underline{a})$ (ili $\mathrm{P}(\underline{a})$) i funkcija gustoće vjerojatnosti (koja može biti poopćena funkcija) $p_{\underline{a}}(a) = \mathrm{p}(\underline{a} = a)$. $\mathrm{P}(A)$ označava vjerojatnost događaja A, a $P_{\underline{a}}$ funkciju vjerojatnosti slučajne varijable $\underline{a}$. Mogući su i kraći zapisi $\mathrm{p}(a)$ i $\mathrm{P}(a)$, gdje se po slovu koje označava vrijednost pretpostavlja slučajna varijabla označena istim slovom bez serifa. Mogu se koristiti i druge oznake za funkciju vjerojatnosti ili funkciju gustoće vjerojatnosti.

$(\underline{a} \mid \underline{b} = b)$, $(\underline{a} \mid b)$	Uvjetna slučajna varijabla
$(\underline{a}, \underline{b})$	Združena slučajna varijabla
$\underline{a} \perp \underline{b}$	*Slučajne varijable $\underline{a}$ i $\underline{b}$ su nezavisne*
$\underline{a} \not\perp \underline{b}$	*Slučajne varijable $\underline{a}$ i $\underline{b}$ su zavisne*
$\underline{a} \perp \underline{b} \mid \underline{c}$	*Slučajne varijable $\underline{a}$ i $\underline{b}$ su uvjetno nezavisne uz poznat ishod slučajne varijable $\underline{c}$*
$\underline{a} \not\perp \underline{b} \mid \underline{c}$	*Slučajne varijable $\underline{a}$ i $\underline{b}$ su uvjetno zavisne uz poznat ishod slučajne varijable $\underline{c}$*
p, q	Razdioba ili funkcija gustoće vjerojatnosti
$\mathbb{A}$	Događaj
$\{R(\underline{a})\}$	Događaj definiran predikatorm slučajne varijable $\underline{a}$

$\mathrm{P}(\{R(\underline{a})\})$, $\mathrm{P}(R(\underline{a}))$	Vjerojatnost događaja $\{R(\underline{a})\}$
$\mathrm{P}(\underline{a})$, $\mathrm{p}(\underline{a})$, $\mathcal{D}$	Razdioba slučajne varijable $\underline{a}$; P ako je $\underline{a}$ diskretna slučajna varijabla, a p ako nije ili ako se ne zna
$\mathrm{P}(\underline{a} = a)$, $P_{\underline{a}}(a)$, $\mathrm{P}(a)$	Vjerojatnost događaja $\{\underline{a} = a\}$
$\mathrm{p}(\underline{a} = a)$, $p_{\underline{a}}(a)$, $\mathrm{p}(a)$	Gustoća vjerojatnosti događaja $\{\underline{a} = a\}$
$p_{\underline{a}\mid\underline{b}}(a)$, $\mathrm{p}(a \mid b)$	Gustoća vjerojatnosti događaja $\{\underline{a} = a \mid \underline{b} = b\}$
$p_{\underline{a},\underline{b}}(a, b)$, $\mathrm{p}(a, b)$	Gustoća vjerojatnosti događaja $\{\underline{a} = a, \underline{b} = b\}$
$\underline{a} \sim q$, $\mathrm{p}(\underline{a}) = q$	*Slučajna varijabla $\underline{a}$ ima razdiobu q*
$\underline{a} \sim \mathbb{A}$	*Slučajna varijabla $\underline{a}$ ima takvu razdiobu da svi elementi (multi)skupa $\mathbb{A}$ imaju vjerojatnost proporcionalnu višestrukosti ($\frac{1}{\lvert\mathbb{A}\rvert}$ za običan skup)*
$a \sim q$	*a se izvlači iz razidiobe q*
$a \sim \underline{a}$, $a \sim \mathrm{p}(\underline{a})$	*a se izvlači iz razidobe $\mathrm{p}(\underline{a})$*
$\mathop{\mathbf{E}}_{a\sim\underline{a}} f(a)$, $\mathop{\mathbf{E}}_{\underline{a}} f(a)$	Očekivanje funkcije slučajne varijable $\underline{a}$
$\mathop{\mathbf{D}}_{a\sim\underline{a}} f(a)$, $\mathop{\mathbf{D}}_{\underline{a}} f(a)$	Disperzija (varijanca) funkcije slučajne varijable $\underline{a}$
$\mathrm{Cov}(\underline{a}, \underline{b})$	Kovarijanca
$\mathcal{N}(\mu, \sigma^2)$	Normalna razdioba s učekivanjem μ i varijancom σ^2
$\mathcal{U}(\mathbb{A})$	Uniformna razdioba nad skupom $\mathbb{A}$

Information theory

$\mathrm{I}(\mathbb{A})$	Information content of event $\mathbb{A}$
$\mathrm{H}(\underline{a})$	Entropy
$\mathrm{h}(\underline{a})$	Differnetial entropy
$\mathrm{I}(\underline{a}; \underline{b})$	Mutual information
$\mathrm{H}(\underline{a} \mid \underline{b})$	Conditional entropy
$\mathrm{H}_{\underline{b}}(\underline{a})$	Cross entropy
$\mathrm{D}_{\underline{b}}(\underline{a})$	Relative entropy (Kullback-Leibler divergence)

Grafovi

$\mathrm{pa}_G(a)$	Skup čvorova roditelja čvora a u grafu G
$\mathrm{ch}_G(a)$	Skup čvorova djece čvora a u grafu G
$\mathrm{pred}_G(a)$	Skup čvorova prethodnika čvora a u grafu G
$\mathrm{succ}_G(a)$	Skup čvorova nasljednika čvora a u grafu G

Other

$\mathbb{A} \to \mathbb{B}$	A set of functions with domain $\mathbb{A}$ and codomain $\mathbb{B}$
$f \in (\mathbb{A} \to \mathbb{B})$	Function definition; a function mapping from its domain $\mathbb{A}$ to its codomain $\mathbb{B}$
$x \mapsto f(x)$	Function definition; a function mapping x from its domain to $f(x)$ in its codomain
$f + g$	Sum of functions
fg	Product of functions
$f * g$	Convolution of functions
$\langle f \vert g \rangle$	Scalar product of functions
$\lvert \mathbb{A} \rvert$	Set cardinality
$\delta(\cdot)$	Dirac delta
$[\![\cdot]\!]$	Iverson bracket; $[\![P]\!] = \begin{cases} 1, & P \equiv \top \\ 0, & P \equiv \bot \end{cases}$
$f[\mathbb{A}]$	Image of $\mathbb{A}$ (subset of the domain) under f; $f[\mathbb{A}] \coloneqq \{f(a) \mid a \in \mathbb{A}\}$
$f^{-1}[\mathbb{B}]$	Inverse image (preimage) of $\mathbb{B}$ (subset of the codomain) under f; $f^{-1}[\mathbb{B}] \coloneqq \{a \mid f(a) \in \mathbb{B}\}$

1 Probability

Probability of an event $\mathrm{P}(E)$

The distribution of a random variable $\underline{x}$ is denoted $\mathrm{p}_{\underline{x}}$. If $\underline{x}$ is known to be discrete, its distribution can be denoted with $\mathrm{P}_{\underline{x}}$.

$$\mathrm{P}_{\underline{x}}(x) = \mathrm{P}(\underline{x} = x). \tag{1}$$

$$\mathrm{p}_{\underline{x}}(x) = \lim_{\epsilon \to 0} \frac{\mathrm{P}(\underline{x} \in B_\epsilon(x))}{\int_{x' \in B_\epsilon(x)} \mathrm{d}x'}. \tag{2}$$

$$\mathrm{P}_{\underline{x}}(x \in A) = \int_{x \in A} \mathrm{d}\mathrm{p}_{\underline{x}}(x). \tag{3}$$

1.1 Functions of random variables

For any function $f \in \mathbb{A} \to \mathbb{B}$, we will use the same symbol to denote an equivalent function that maps random variables taking values in $\mathbb{A}$ to random variables taking values in $\mathbb{B}$:

$$\mathrm{P}(f(\underline{a}) \in \mathbb{B}_1) \coloneqq \mathrm{P}\big(\underline{a} \in f^{-1}[\mathbb{B}_1]\big), \tag{4}$$

where $f^{-1}[\mathbb{B}_1] \coloneqq \{a \mid f(a) \in \mathbb{B}_1\}$ is what we call the preimage of $\mathbb{B}_1 \subseteq \mathbb{B}$ under f.

Conditional expectation:

$$\mathbf{E}(\underline{x} \mid \underline{y}) = (y \mapsto \mathbf{E}(\underline{x} \mid y))(\underline{y}). \tag{5}$$

The conditional expectation $E(\underline{x} \mid \underline{y})$ is a function of the random variable $\underline{y}$ and, thus, is a random variable as well.

2 Statistics

2.1 Monte Carlo integration

Monte Carlo integration (approximation) is a method of approximating integrals that can be expressed as expectations of some random variables.

Let $u \in \mathbb{A} \to \mathbb{R}$ be a function. Let $I \coloneqq \int u(x) \,\mathrm{d}x$ be an integral that is hard to compute. If we express u as the product of a function f and a probability density function (distribution) p, $u(x) = f(x)p(x)$, the integral I can be expressed as the expectation of $f(\underline{x})$, where $\underline{x}$ is distributed according to p:

$$I = \int f(x) \,\mathrm{d}x = \int f(x)p(x) \,\mathrm{d}x = \mathop{\mathbf{E}}_{x \sim p}(f(x)). \tag{6}$$

This expectation can be approximated with the following estimator:

$$\underline{\hat{I}}_n \coloneqq \frac{1}{n} \sum_{i=1..n} f(\underline{x}_i), \tag{7}$$

where $\underline{x}_i \sim p$. The estimator $\underline{\hat{I}}_n$ is unbiased if $\underline{x}_i$ are independent and it is valid if variances of $u(\underline{x}_i)$ are bounded.

2.2 Rejection sampling

2.3 Importance sampling

$I := \int_{x\in\mathbb{B}} f(x)\,\mathrm{d}x$

3 Information theory

3.1 Information-theoretic measures

functionals

3.1.1 Basic concepts

Information content of an event – optimal message length for the event $\{\underline{x} = x\}$:

$$\mathrm{I}(\underline{x} = x) = -\ln \mathrm{P}(x). \tag{8}$$

Entropy (Shannon entropy) of a random variable (or a distribution) – expected message length for optimally encoded elementary events of $\underline{x}$:

$$\mathrm{H}(\underline{x}) = \mathop{\mathbf{E}}_{\underline{x}} \mathrm{I}(\underline{x} = x) = -\mathbf{E} \ln \mathrm{P}(\underline{x}). \tag{9}$$

The same formula applies for joint distributions, e.g. for the distribution $\mathrm{P}(\underline{x}, \underline{y})$, we denote entropy (also called **joint entropy**) by $\mathrm{H}(\underline{x}, \underline{y})$. Entropy is a common measure of uncertainty.

Cross entropy – expected message length if the optimal code for $\mathrm{P}_{\underline{y}}$ is used, but $\mathrm{P}_{\underline{x}}$ is sampled.

$$\mathrm{H}_{\underline{y}}(\underline{x}) = \mathop{\mathbf{E}}_{\underline{x}} \mathrm{I}(\underline{y} = x) = -\mathop{\mathbf{E}}_{\underline{x}} \ln \mathrm{P}(\underline{y} = x). \tag{10}$$

Relative entropy (Kullback–Leibler (KL) divergence) – difference of cross entropy and entropy, measures how much $\mathrm{P}_{\underline{y}}$ differs from $\mathrm{P}_{\underline{x}}$:

$$\mathrm{D}_{\underline{y}}(\underline{x}) = \mathrm{H}_{\underline{y}}(\underline{x}) - \mathrm{H}(\underline{x}) = \mathop{\mathbf{E}}_{\underline{x}}(\mathrm{I}(\underline{y} = x) - \mathrm{I}(\underline{x} = x)) = \mathop{\mathbf{E}}_{\underline{x}} \ln \frac{\mathrm{P}(x)}{\mathrm{P}(\underline{y} = x)}. \tag{11}$$

Mutual information of random variables is the expectation of how much knowing the outcome of one of them gives information (or reduces the uncertainty) about the other:

$$\mathrm{I}(\underline{x};\underline{y}) = \mathrm{H}(\underline{x}) - \mathrm{H}(\underline{x} \mid \underline{y}) = \mathrm{H}(\underline{y}) - \mathrm{H}(\underline{y} \mid \underline{x}) = \mathrm{H}(\underline{x}) + \mathrm{H}(\underline{y}) - \mathrm{H}(\underline{x},\underline{y}). \tag{12}$$

If there is a common condition, we can use $\mathrm{I}(\underline{x};\underline{y} \mid z)$ as a shorter notation for $\mathrm{I}((\underline{x} \mid z);(\underline{y} \mid z))$. If we want to express mutual information between e.g. $\underline{x}$ and $\underline{y} \mid z$, we do it like this: $\mathrm{I}(\underline{x};(\underline{y} \mid z))$, without ambiguity.

Mutual information can also be expressed as relative entropy:

$$\mathrm{I}(\underline{x};\underline{y}) = \mathrm{D}_{\mathrm{P}_{\underline{x}}\mathrm{P}_{\underline{y}}}(\mathrm{P}_{\underline{x},\underline{y}}) \tag{13}$$

$$\mathrm{I}(\underline{x};\underline{y}) = \mathrm{D}_{\mathrm{P}[\underline{x}]\mathrm{P}[\underline{y}]}(\mathrm{P}[\underline{x},\underline{y}]) \tag{14}$$

$$\mathrm{I}(\underline{x};\underline{y}) = \mathrm{D}_{\mathrm{P}(\underline{x})\mathrm{P}(\underline{y})}(\mathrm{P}(\underline{x},\underline{y})) \tag{15}$$

$$\mathrm{I}(\underline{x};\underline{y}) = \mathrm{D}_{[\underline{x}][\underline{y}]}([\underline{x},\underline{y}]). \tag{16}$$

3.1.2 Conditional measures

Conditional counterparts of the information-theoretic measures have random variables in the condition-part of the expression that represents the argument of a measure. e.g. **Conditional entropy** is defined like this:

$$\mathrm{H}(\underline{x} \mid \underline{y}) = \mathop{\mathbf{E}}_{\underline{y}} \mathrm{H}(\underline{x} \mid y). \tag{17}$$

Similarly, conditional cross-entropy can be defined like this:

$$\mathrm{H}_{\underline{y}}(\underline{x} \mid \underline{z}) = \mathop{\mathbf{E}}_{\underline{z}} \mathrm{H}_{\underline{y}}(\underline{x} \mid z), \tag{18}$$

conditional mutual information like this:

$$\mathrm{I}(\underline{x};\underline{y} \mid \underline{z}) = \mathop{\mathbf{E}}_{\underline{z}} \mathrm{I}(\underline{x};\underline{y} \mid z). \tag{19}$$

3.1.3 Differential counterparts

3.1.4 Information theory and measure theory

https://en.wikipedia.org/wiki/Information_theory_and_measure_theory

3.2 Kolmogorov complexity

3.3 Minimum description length

4 Machine learning

An Occam's razor thought. If the model (hypothesis search space) is simpler (smaller), we are more likely to find the correct hypothesis. If the correct hypothesis is complex, there will be more hypotheses consistent with the data and we are less likely to find the correct one anyway.

5 Uncertainty in machine learning

5.1 Expressing uncertainty

The basic and most complete way to express uncertainty are probability distributions. From a probability distribution, other uncertainty measures can be derived. Some common ones are the distribution of a derived random variable (a random variable which is a function of the original one), a parameter or a property of the distribution (e.g. the probability of the most certain value of the random variable or the entropy of the distribution).

5.2 Epistemic and aleatory uncertainty

Epistemička nesigurnost (nesigurnost modela) je nesigurnost u model ili parametre. Ona se može smanjiti uz više podataka/informacija. *Epistemička nesigurnost predikcije* dolazi od nesigurnosti u model/parametre.

Kad parametre modela procjenjujemo točkasto, nemamo aposteriornu razdiobu parametara i ne znamo kakva je epistemička nesigurnost (ne možemo ju izraziti), ali ju možemo smanjiti uz više podataka. Kod bayesovske procjene parametara ili kod ansabla možemo procijeniti epistemičku nesigurnost.

Aleatorna nesigurnost (predikcije) je nesigurnost koja dolazi od višeznačnosti podataka i ograničenja modela. Aleatorna nesigurnost se ne može smanjiti uz više podataka, ali bi se mogla smanjiti uz bolje podatke, tj. podatke koji imaju značajke koje sadrže više korisnih informacija, ili model koji pronalazi bolje značajke.

Kod diskriminativnog modela izlazna razdioba $p(y \mid x, \theta_{\mathsf{MAP}})$ izražava aleatornu nesigurnost.

Je li ukupna nesigurnost zbroj epistemičke i aleatorne? Mislim da procjena ukupne nesigurnosti ovisi o tome koliko je dobro procijenjena epistemička nesigurnost. Što je lošija procjena aposteriorne razdiobe parametara, to je procjena epistemičke nesigurnosti lošija.

5.3 Nesigurnost i izvanrazdiobni primjeri

Neka je D_{train} razdioba iz koje su došli primjeri za učenje. Diskrimanativni model uči funkciju $p(y \mid x)$. Ako je gustoća vjerojatnosti $D_{\text{train}}(x)$ jako mala (ili 0), moguće je da nije bilo sličnih primjera u skupu za učenje i model može dati bilo kakvu predikciju za taj primjer. Takve primjeri su *izvanrazdiobni primjeri*.

Ipak, pokazano je se da se (kod nekih modela) izvanrazdiobni primjeri često mogu dosta dobro prepoznavati na temelju izlazne razdiobe modela s točkasto procijenjenim parametrima.

5.4 Successfulness of epistemic uncertainty estimation with bayesian inference approximation

6 Adversarial examples and generalization

Even for models that perform similar to humans on testing data, it has been shown that, by perturbing input examples even inperceptibly for humans, the models can be made to significantly change their predictions, i.e. make confident wrong predictions (Goodfellow et al., 2014, Szegedy et al., 2013). Such perturbed input examples are called **adversarial examples**. The existence of adversarial examples indicates that such models are probably performing well for somewhat wrong reasons, without actually *understanding data*.

6.1 Defining and finding adversarial examples

Let $\mathbb{X}$ be the input space, and $d \in (\mathbb{X} \times \mathbb{X} \to \mathbb{R}^+)$ a **distance function** that can be used to define similarity between inputs. For each example $\boldsymbol{x}$ we can also define its **neighbourhood** as $B_\epsilon = \{\boldsymbol{x}' \colon d(\boldsymbol{x}', \boldsymbol{x}) \leq \epsilon\}$, ϵ being the maximum distance from the example.

Ideally, the neighbourhood of an example $\boldsymbol{x}$ should be the set of *perceptually similar* examples that all belong to the same class as $\boldsymbol{x}$ (their true class may be at most ambiguous), but it is hard to define such a neighbourhood. A practical and common way of defining the neighbourhood function (for images) is to let the distance function d be a L^p distance where p is usually ∞ or 2. Note that if an example is very near the true class boundary, such a neighbourhood may contain examples actually belonging to another class.

Finding an adversarial example can be defined as an optimization problem of maximizing the loss with respect to the input with the constraint that the input is in the neighbourhood $B_\epsilon(\boldsymbol{x})$:

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,max}_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} L(y, h(\boldsymbol{x}')), \tag{20}$$

where y is either the true label or the predicted label. Let $\hat{h}(\boldsymbol{x}) = \arg\max_{y'} h(\boldsymbol{x}')_{[y']}$ denote the function that assigns the label with the highest probability to an input. An objective can also be to find the $\tilde{\boldsymbol{x}}$ closest to $\boldsymbol{x}$ such that the classifier misclassifies it (Moosavi-Dezfooli et al., 2016):

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}' :\ \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}) \neq y} d(\boldsymbol{x}', \boldsymbol{x}). \tag{21}$$

The described objectives, where it only matters that the adversarial example is misclassified, are objectives for **untargeted adversarial attacks**. There are also **targeted adversarial attacks**, where the objective is to create an adversarial example such that the model classifies it as some desired class. Targeted attack objectives corresponding to equations (20) and (21) are:

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} L(y_\mathsf{t}, h(\boldsymbol{x}')) \tag{22}$$

and

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}' :\ \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}) = y_\mathsf{t}} d(\boldsymbol{x}', \boldsymbol{x}), \tag{23}$$

where y_t denotes the adversarial target label.

Adversarial examples can also be generated without knowledge of the true label. Such adversarial examples are called **virtual adversarial examples**. Miyato et al. (2017) propose the following objective for adversarial training:

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} D((\underline{\boldsymbol{y}} \mid \underline{\boldsymbol{x}}, \underline{\boldsymbol{\theta}}), (\underline{\boldsymbol{y}} \mid \underline{\boldsymbol{x}} = \boldsymbol{x}', \underline{\boldsymbol{\theta}})), \tag{24}$$

where D is some non-negative function that represents distance between distributions. Other (untargeted) attacks can also produce virtual adversarial examples by using the predicted label $\hat{h}(\boldsymbol{x})$ instead of the true label in the loss.

6.1.1 Transferability

Common (naturally trained) CV models (algorithms) are biased similarly with respect to having adversarial examples – they are nonrobust in similar ways.

Can this bias be easily overcome? Unfortunately, there seems not to be much evidence indicating this.

Overdependence on semantically low-level features.

6.1.2 Adversarial training

Kurakin et al. (2016) note that by using the true label in the loss in untargeted attacks (y in equation (20)) can cause

6.1.3 Distance metrics for images

Usually, an L^p distance ($d(\boldsymbol{x}', \boldsymbol{x}) = \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p$) is used as a distance metric for adversarial examples.

Scale-invariant norms. Let $\boldsymbol{x}$ denote some image (or perturbation) and $\boldsymbol{x}_\lambda$ the same image with dimensions scaled by λ. By having more pixels, $\boldsymbol{x}_\lambda$ has a greater norm. The scaled image contains λ^2 the number of pixels of the original image, and every pixel approximately repeated λ^2 times. In order to make the norm of the scaled image equal to the norm of the original image, the first though might be to divide the norm by λ^2. As the following shows, this would work for $p = 1$, but not otherwise:

$$\begin{aligned}
\|\boldsymbol{x}_\lambda\|_p &= \left(\sum_{u \in \{0..\lambda H\}} \sum_{v \in \{0..\lambda W\}} \left|\boldsymbol{x}_{\lambda[u,v]}\right|^p \right)^{\frac{1}{p}} && (25) \\
&\approx \left(\sum_{u \in \{0..H\}} \sum_{v \in \{0..W\}} \lambda^2 \left|\boldsymbol{x}_{\lambda[u,v]}\right|^p \right)^{\frac{1}{p}} && (26) \\
&= \left(\lambda^2 \sum_{u \in \{0..H\}} \sum_{v \in \{0..W\}} \left|\boldsymbol{x}_{\lambda[u,v]}\right|^p \right)^{\frac{1}{p}} && (27) \\
&= \lambda^{\frac{2}{p}} \left(\sum_{u \in \{0..H\}} \sum_{v \in \{0..W\}} \left|\boldsymbol{x}_{\lambda[u,v]}\right|^p \right)^{\frac{1}{p}} && (28) \\
&= \lambda^{\frac{2}{p}} \|\boldsymbol{x}\|_p. && (29)
\end{aligned}$$

For non-infinite p, if we have 2 perceptually similar perturbation of different resolutions, the higher-resolution one will have a greater norm by a factor of $(\lambda^2)^{\frac{1}{p}}$. Hence, we can define scale invariant equivalents of L^p norms by dividing the norm by the scale factor of the image. The scale factor can be relative to an image with area 1, so we can use the number of pixels as λ^2. We can define scale-invariant norms like this:

$$\|\boldsymbol{x}\|_{\mathsf{s}p} := \frac{\|\boldsymbol{x}\|_p}{n^{\frac{1}{p}}} = \left(\frac{1}{n} \sum_i |x_i|^p \right)^{\frac{1}{p}}, \tag{30}$$

where n is the number of pixels in $\boldsymbol{x}$. Such norms could probably enable more informative comparison of norms between different-resolution images and different datasets, and easier hyperparameter choice for adversarial training.

Maybe something similar could be done about objects of different scale?

...

Expectation of scale-invariant p-norms of high-dimensional uniformly distributed random vectors. Let $\underline{\boldsymbol{x}}_n = (\underline{x}_i)_{i \in \{1..n\}}$ be a random vector with n independent elements $\underline{x}_i \sim \mathcal{U}([-\epsilon, \epsilon])$. Assuming n is very large, its scale-invariant p-norm can[1] be approximated with a single sample $\boldsymbol{x}$:

$$
\begin{aligned}
\mathbf{E}\|\underline{\boldsymbol{x}}\|_{\mathsf{s}p}^{p} \approx \|\boldsymbol{x}\|_{\mathsf{s}p} &= \left(\frac{1}{n}\sum_i \left|\boldsymbol{x}_{[i]}\right|^p\right)^{\frac{1}{p}} && \text{(approximation with a single sample)} \\
&\approx \left(\mathbf{E}\left(\left|\underline{\boldsymbol{x}}_{[i]}\right|^p\right)\right)^{\frac{1}{p}} && \text{(IID elements, large } n\text{)} \\
&= \left(\int_0^1 \left|\underline{\boldsymbol{x}}_{[i]}\right|^p \mathrm{d}x\right)^{\frac{1}{p}} \\
&= \left(\frac{1}{p+1}\right)^{\frac{1}{p}} = (p+1)^{-\frac{1}{p}},
\end{aligned}
$$

which is a monotonically increasing functon of p.

$$
\mathbf{E}\|\underline{\boldsymbol{x}}\|_{\mathsf{s}p}^{p} \approx \|\boldsymbol{x}\|_{p} = \left(\frac{1}{n}\sum_i \left|\boldsymbol{x}_{[i]}\right|^p\right)^{\frac{1}{p}}
$$

Local p-norm and hiererchical norm. Let $\boldsymbol{k}$ denote a non-negative 2-D kernel with $\|\boldsymbol{k}\|_1 = 1$, e.g. Gaussian cenetered at $(0, 0)$. Let $\boldsymbol{x}$ denote an image perturbation and assume that it has a single channel for simplicity. We can define the local p-norm around a pixel (i, j) as

$$
\mathrm{LocalNorm}_{p,\boldsymbol{k}}(\boldsymbol{x})_{[i,j]} = \left(|\boldsymbol{x}|^p * \boldsymbol{k}\right)_{[i,j]}^{\frac{1}{p}}, \tag{31}
$$

where the absolute value and powering operatons are elementwise.

We can then define a bi-level hierarchical (p_1, p_2)-norm as $\|\mathrm{LocalNorm}_{p_1,\boldsymbol{k}}(\boldsymbol{x})\|_{p_2}$. This can be generalizad to a multi-level norm $(p_1, .., p_n)$-norm by chaining multiple local norms with potentially different kernels until the last, global, p_n-norm.

Why?

[1] TODO: prove that the approximation is good for large n

6.2 Making adversarially robust classifiers

7 Generative adversarial networks

7.1 Getting the probability of the example from the generator

first paragraph

case 1) normal generator

case 2) invertible generator

8 Paper summaries

8.1 The Conditional Entropy Bottleneck (Anonymous, 2018)

URL: https://openreview.net/forum?id=rkVOXhAqY7.

9 p

Neka je odnos između slučajnih varijabli $\underline{x}$ i $\underline{y}$ definiran funkcijom f koja ishode jedne slučajne varijable deterministički preslikava u ishode druge, što označavamo ovako: $\underline{y} = f(\underline{x})$. Ako su $\underline{x}$ i $\underline{y}$ diskretne slučajne varijable, onda je razdioba slučajne varijable $\underline{y}$ definirana ovako:

$$P_{\underline{y}}(y) = \sum_{x\colon f(x)=y} P_{\underline{x}}(x). \tag{32}$$

Ako su $\underline{x}$ i $\underline{y}$ kontinuirane slučajne varijable s vrijednostima iz $\mathbb{R}$ i f je injektivna, može se pokazati (Elezović, 2007) da vrijedi

$$p_{\underline{y}}(y) = p_{\underline{x}}(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right|. \tag{33}$$

Neka je $C_{\underline{x}}(x) := \int_{-\infty}^{x} p_{\underline{x}}(x')\,\mathrm{d}x'$. Vrijednosti iz intervala $(x, x+\epsilon)$ na kojem je f monotono rastuća preslikavaju se u interval $(f(x), f(x+\epsilon))$. Granice su obrnute ako je f monotono padajuća na tom intervalu. Budući da $\mathrm{P}(\underline{x} \in (x, x+\epsilon)) = \mathrm{P}(\underline{y} \in (f(x), f(x+\epsilon)))$, vrijedi

$$C_{\underline{x}}(x+\epsilon) - C_{\underline{x}}(x) = C_{\underline{y}}(f(x+\epsilon)) - C_{\underline{y}}(f(x)). \tag{34}$$

Ako obje strane jednadžbe dijelimo s ϵ i pustimo $\epsilon \to 0$,

$$\lim_{\epsilon \to 0} \frac{C_{\underline{x}}(x+\epsilon) - C_{\underline{x}}(x)}{\epsilon} = \lim_{\epsilon \to 0} \frac{C_{\underline{y}}(f(x+\epsilon)) - C_{\underline{y}}(f(x))}{\epsilon}. \tag{35}$$

Redom, prema definiciji derivacije, pravilu derivacije složene funkcije i definiciji funkcija $C_{\underline{\boldsymbol{x}}}$ i $C_{\underline{\boldsymbol{y}}}$ kao integrala gustoće vjerojatnosti, slijedi:

$$\frac{\mathrm{d}}{\mathrm{d}x} C_{\underline{x}}(x) = \frac{\mathrm{d}}{\mathrm{d}x} C_{\underline{y}}(f(x)), \tag{36}$$

$$\frac{\mathrm{d}}{\mathrm{d}x} C_{\underline{x}}(x) = \frac{\mathrm{d}}{\mathrm{d}f(x)} C_{\underline{y}}(f(x)) \frac{\mathrm{d}}{\mathrm{d}x} f(x), \tag{37}$$

$$p_{\underline{x}}(x) = p_{\underline{y}}(f(x)) \frac{\mathrm{d}}{\mathrm{d}x} f(x). \tag{38}$$

Može se pokazati da je za monotono padajuće intervale desna strana jednadžbe (38) pomnožena s -1, iz čega uz jednadžbu (38) slijedi

$$p_{\underline{x}}(x) = p_{\underline{y}}(y) \left| \frac{\mathrm{d}y}{\mathrm{d}x} \right|, \tag{39}$$

gdje je $f(x)$ zamijenjen s y. Množenjem toga s $\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = \left|\frac{\mathrm{d}y}{\mathrm{d}x}\right|^{-1}$ slijedi jednadžba (33). To pravilo se može poopćiti i na vektore. Onda vrijedi (Murphy, 2012)

$$p_{\underline{\boldsymbol{y}}}(\boldsymbol{y}) = p_{\underline{\boldsymbol{x}}}(\boldsymbol{x}) \left| \det \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}} \right|. \tag{40}$$

Neka je $\underline{z}$ zbroj slučajnih varijabli $\underline{x}$ i $\underline{y}$. Onda vrijedi

$$p_{\underline{z}}(z) = \int p_{\underline{x},\underline{y}}(x, z-x) \, \mathrm{d}x. \tag{41}$$

Ako su $\underline{x}$ i $\underline{y}$ nezavisne, onda to postaje konvolucija:

$$p_{\underline{z}}(z) = \int p_{\underline{x}}(x) p_{\underline{y}}(z-x) \, \mathrm{d}x =: (p_{\underline{x}} * p_{\underline{y}})(z). \tag{42}$$

9.1 PDF of vector r.v. defined via a function of a vector r.v.

Let $f \in (\mathbb{R}^n \to \mathbb{R}^m)$ and $\underline{\boldsymbol{y}} = f(\underline{\boldsymbol{x}})$. We want to compute the PDF of $\underline{y}$, or, equivalently, the distribution $\mathrm{p}(\underline{\boldsymbol{y}})$.

$$\frac{\partial \underline{\boldsymbol{y}}}{\partial \underline{\boldsymbol{x}}} \in \mathbb{R}^{m \times n} \tag{43}$$

For easier analysis, let's assume that $m = 1$, i.e. $\underline{\boldsymbol{y}}$ is a scalar, and denote it with $\underline{y}$. We want to compute its PDF.

10 Dense anomaly detection for dense prediction based on reconstruction error

Pretpostavljamo duboki diskriminativni model $h(\boldsymbol{x};\boldsymbol{\theta})$ s parametrima $\boldsymbol{\theta}$ koji ulaz $\boldsymbol{x}$ preslikava u vektor $\boldsymbol{y}$ koji predstavlja izlaznu razdiobu $\mathrm{p}(\underline{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$.

previsoka sigurnost (postizanje male pogreške na skupu za učenje, kalibracija temperaturnim skaliranjem)

kriva klasifikacija izvanrazdiobnih primjera

neprijateljski primjeri

(Hendrycks i Gimpel, 2016)

(Guo et al., 2017)

(Lee et al., 2017)

(Liang et al., 2017)

Neki pristupi za prepoznavanje anomalija/izvanrazdiobnih primjera (detaljnije opisati i s referencama):

- iz predikcije – očekujemo manju vjerojatnost i veću nesigurnost za izvanrazdioben primjere,
- iz neke skrivene reprezentacije – možemo analizirati razdiobe logita ili nečega drugoga i pomoću toga propoznavati izvanrazdiobne primjere,
- eksplicitnim učenjem razlikovanja razdiobe skupa za učenje od neke pozadinske razdiobe,
- korištenje generativnog modela za generiranje primjera iz područja male gustoće vjerojatnosti i korištenje njih kao izvanrazdiobnih primjera
- korištenjem generativnog modela kod kojeg je moguće izračunati gustoću vjerojatnosti za primjer,
- korištenjem rekonstrukcijske pogreške autoenkodera.

Neki pristup ise mogu kombinirati.

10.1 Autoencoders and GAN-s

.

10.2 Korištenje rekonstrukcijske pogreške autoenkodera za propoznavanje onoga što model ne zna da ne zna

Pretpostavljamo duboki diskriminativni model $h(\boldsymbol{x};\boldsymbol{\theta})$ s parametrima $\boldsymbol{\theta}$ koji ulaz $\boldsymbol{x}$ preslikava u vektor $\boldsymbol{y}$ koji predstavlja izlaznu razdiobu $\mathrm{p}(\underline{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$.

10.2.1 Korištenje autoenkodera za prepoznavanje izvanrazdiobnih primjera

Sabokrou et al. (2018) za otkrivanje anomalija u slici predlažu korištenje autoenkodera (s jako velikom skrivenom reprezentacijom) kojemu se kod učenja kao ulaz daje zašumljena slika. Uz autoenkoder se dodaje diiskriminator koji se uči a razlikuje izlaz autoenkodera od stvarnih primjera za učenje. Kao gubitak se koristi težinski zbroj kvadratne rekonstrukcijske pogreške i suparničkog gubitka. Kao primjeri se koriste mali izrazani dijelovi većih slika. Za prepoznavanje anomalija koristi se izlaz diskriminatora za rekonstruirani primjer.

Pidhorskyi et al. (2018) isto predlažu pristup s autoenkoderom i superničkim gubitkom. Kod njih gubitak ima 3 komponente: (1) suparnički gubitak koji potiče da primjeri za učenje "pokrivaju" cijelu zadanu (Gaussovu) razdiobu skrivene reprezentacije, (2) suparnički gubitak koji potiče da rekonstruirani primjeri budu iz razdiobe skupa za učenje (kao kod Sabokrou et al. (2018)) i (3) rekonstrukcijski gubitak. Kao mjera za procjenu je li primjer izvan razdiobe se koristi procjena $\mathrm{p}(\boldsymbol{z} \mid \mathbb{D})$ koja ovisi o udaljenosti od "manifolda". Trebam još pručiti kako se točno dobiva.

Pretpostavljamo duboki diskriminativni model $h(\boldsymbol{x};\boldsymbol{\theta})$ s parametrima $\boldsymbol{\theta}$ koji ulaz $\boldsymbol{x}$ preslikava u vektor $\boldsymbol{y}$ koji predstavlja izlaznu razdiobu $\mathrm{p}(\underline{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$. Želimo prepoznavati izvanrazdiobne primjere pomoću autoenkodera.

Neke ideje u vezi autoenkodera:

- Koristiti dekoder s heteroskedastičkom (Kendall i Gal, 2017) nesigurnošću u rekonstrukciju (modelirati $\mathrm{p}(\underline{\boldsymbol{x}} \mid \boldsymbol{z})$) i $-\ln \mathrm{p}(\underline{\boldsymbol{x}} \mid \boldsymbol{z})$ za empirijski gubitak.
- Isprobati rekonstrukciju neke skrivene reprezentacije klasifikatora kako bi se u rekonstrukcijskoj pogrešci naglasile značajke bitne za klasifikaciju (semantički bitne). Možemo h rastaviti na dvije funkcije: $h(\boldsymbol{x}) = (f_2 \circ f_1)(\boldsymbol{x})$ pa onda učime autoenkoder rekonstruirati $f_1(\boldsymbol{x})$. Ako kao kao f_1 koristimo bijekciju, možemo vidjeti kako izgleda rekonstrukcija ulaza koja odgovara rekonstruiranoj reprezentaciji.
- Isprobati klasifikaciju na temelju skrivene reprezentacije autoenkodera $\boldsymbol{z}$, koristiti i klasifikacijski gubitak za učenje kodera, vidjeti kako izgledaju rekonstrukcije. (Isprobati CEB?)
- Minimalna reprezentacija autoenkodera onemogućuje neprijateljske primjere kojima je cilj postići dobru rekonstrukciju anomalije, pogotovo ako

pretpostavimo dovoljno dobru funkciju rekonstrukcijske pogreške ili diskriminator.

- Je li dobro poticati da skup za učenje pokriva cijelu razdiobu $\mathrm{p}(\underset{\sim}{z})$? Onda će različite klase biti odmah jedna uz drugu – malo izmijenimo $\boldsymbol{z}$ i dođemo u područje visoke gustoće za neku drugu klasu. Možda valja učiti razdiobu $\mathrm{p}(\underset{\sim}{z} \mid \mathbb{D})$ i znati koja su područja niže gustoće (margine) (kako?).
- Dodati šum na ulaz autoenkodera. Možda bi valjalo nešta između gaussovog šuma i "rupa" za popunjavanje.

Osnovni model koji bih htio isprobati (na velikim slikama) je ovakav:

$$\boldsymbol{x} \xmapsto{f_1} \boldsymbol{h} \xmapsto{f_2} \boldsymbol{y} \quad \text{(klasifikator)}, \tag{44}$$

$$\boldsymbol{h} \xmapsto{e} \boldsymbol{z} \xmapsto{d} \boldsymbol{h}_\mathsf{r} \quad \text{(autoenkoder skrivene reprezentacije)}. \tag{45}$$

Treba odrediti točan opis modela.

Možemo isprobati i klasifikaciju na temelju rekonstrukcije:

$$\boldsymbol{h}_\mathsf{r} \xmapsto{f_2} \boldsymbol{y}. \tag{46}$$

Možemo isprobati i klasifikaciju na temelju skrivene reprezentacije:

$$\boldsymbol{z} \xmapsto{f_z} \boldsymbol{y}, \tag{47}$$

i istovremeno učenje klasifikacije i rekonstrukcije.

Bilo bi zanimljivo vidjeti kako izgleda rekonstrukcija ulazne slike na temelju izlaza autoenkodera ovisno o tome koji skriveni sloj se kodira:

$$\boldsymbol{h}_\mathsf{r} \xmapsto{f_1^{-1}} \boldsymbol{x}_\mathsf{r} \quad \text{(inverz prvog dijela klasifikatora s ulazom } \boldsymbol{h}_\mathsf{r}\text{)}. \tag{48}$$

Rekonstrukciju ulazne slike možemo dobiti ako koristimo neki model koji je bijektivan, npr. i-RevNet.

$$\min_h L_\mathsf{c}(\boldsymbol{y}, \boldsymbol{y}^*) \tag{49}$$

$$\min_{d,e} L_\mathsf{r}(\boldsymbol{h}, \boldsymbol{h}_\mathsf{r}) \tag{50}$$

10.2.2 Što bih još htio isprobati

Kombinaciju Lee et al. (2017) i korištenja izvanrazdiobnih primjera u učenju.

Literatura

Neven Elezović. *Vjerojatnost i statistika: Slučajne varijable*. Element, 2007.

Ian J. Goodfellow, Jonathon Shlens, i Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. URL http://arxiv.org/abs/1412.6572.

Chuan Guo, Geoff Pleiss, Yu Sun, i Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017. URL http://arxiv.org/abs/1706.04599.

Dan Hendrycks i Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016. URL http://arxiv.org/abs/1610.02136.

Alex Kendall i Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *CoRR*, abs/1703.04977, 2017. URL http://arxiv.org/abs/1703.04977.

Alexey Kurakin, Ian J. Goodfellow, i Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL http://arxiv.org/abs/1611.01236.

Kimin Lee, Honglak Lee, Kibok Lee, i Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *CoRR*, abs/1711.09325, 2017.

Shiyu Liang, Yixuan Li, i R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017. URL http://arxiv.org/abs/1706.02690.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, i Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *CoRR*, abs/1704.03976, 2017. URL http://arxiv.org/abs/1704.03976.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, i Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. U *CVPR*, stranice 2574–2582. IEEE Computer Society, 2016.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020, 9780262018029.

Stanislav Pidhorskyi, Ranya Almohsen, i Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. U *NeurIPS*, stranice 6823–6834, 2018.

Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, i Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. U *CVPR*, stranice 3379–3388. IEEE Computer Society, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, i Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.