Ex 11

Montag, 17. Juli 2017 16:08

Exercises for the course
Machine Learning 2
Summer semester 2017

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 11

In this exercise sheet, we refer to the sections of the paper "*Methods for interpreting and understanding deep neural networks*" linked via ISIS.

Exercise 1: Experts and Prototypes (40 P)

Consider the linear model $y = \boldsymbol{w}^\top \boldsymbol{x} + b$ mapping some input $\boldsymbol{x}$ to an output y. We would like to interpret the output y by building a prototype $\boldsymbol{x}^\star$ in the input domain following the activation maximization techniques outlined in Section 3.

(a) *Find* the prototype $\boldsymbol{x}^\star$ obtained by activation maximization as formulated in Section 3.1.

(b) *Find* the prototype $\boldsymbol{x}^\star$ obtained by activation maximization as formulated in Section 3.2. We assume that the data is represented by the Gaussian expert $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where $\boldsymbol{\mu}$ and Σ are the mean and covariance.

(c) *Find* the prototype $\boldsymbol{x}^\star$ obtained by activation maximization as formulated in Section 3.3. The data is generated as (i) $\boldsymbol{z} \sim \mathcal{N}(0, I)$ and (ii) $\boldsymbol{x} = A\boldsymbol{z} + \boldsymbol{c}$, where A and $\boldsymbol{c}$ are the parameters of the generator.

(d) *Relate* the prototypes obtained for the three approaches above, in particular under which regularizers, experts and generator, the found prototypes are mutually equivalent.

Exercise 2: Sensitivity Analysis and Taylor Decomposition (30 P)

Let us consider a data point $\boldsymbol{x}$ and its prediction by a homogeneous linear model $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$. We would like to explain the prediction using the methods described in Section 4.

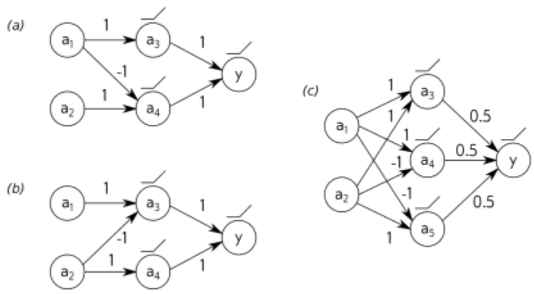
(a) *Compute* the explanation for the prediction $f(\boldsymbol{x})$ using sensitivity analysis as described in Section 4.1.

(b) *Compute* the explanation for the prediction $f(\boldsymbol{x})$ using Taylor decomposition (Section 4.2) at root point $\widetilde{\boldsymbol{x}} = \boldsymbol{0}$.

(c) *Compute* the explanation for the prediction $f(\boldsymbol{x})$ using Taylor decomposition at root point $\widetilde{\boldsymbol{x}}$ chosen to be the nearest (in the Euclidean sense) from $\boldsymbol{x}$. (Hint: You can use the Lagrange multipliers to find this root point.)

Exercise 3: Layer-Wise Relevance Propagation (30 P)

We would like to test the dependence of layer-wise relevance propagation (LRP) on the structure of the neural network. For this, we consider the function $y = \max(a_1, a_2)$, where $a_1, a_2 \in \mathbb{R}^+$ are the input activations. This function can be implemented as a ReLU network in multiple ways. Three examples are given below.



Because of the positive activations, an appropriate rule for both layers is LRP-$\alpha_1\beta_0$ defined in Section 5.1.

(a) *Give* for each network an analytic solution for the obtained scores R_1 and R_2 obtained by application this propagation rule at each layer.

(b) *Discuss* which implementation of the "max" function (a, b, or c) gives the most intuitive explanations.

$\boxed{1}$ linear model $y = \vec{w} \cdot \vec{x} + b$ → find a prototype

a) Consider a DNN classifier mapping data points $\boldsymbol{x}$ to a set of classes $(\omega_c)_c$. The output neurons encode the modeled class probabilities $p(\omega_c|\boldsymbol{x})$. A prototype $\boldsymbol{x}^\star$ representative of the class ω_c can be found by optimizing:

$$\max_{\boldsymbol{x}} \ \log p(\omega_c|\boldsymbol{x}) - \lambda \|\boldsymbol{x}\|^2.$$

wie hängt y mit $p(\omega_c|x)$ zusammen? $y = \log p(\omega_c|x)$?

→ prototype could be $\underbrace{\operatorname{argmax}_x \ y(\vec{x}) - \lambda \|\vec{x}\|^2}$ // λ ist hier kein Lagrange-Multiplikator

$\rightarrow$ prototype could be $\arg\max_x \underbrace{y(\vec{x}) - \lambda \|\vec{x}\|^2}_{L}$ // λ ist hier kein Lagrange-Multiplikator

mit $L = y(\vec{x}) - \lambda\|\vec{x}\|^2 \rightarrow \frac{\partial L}{\partial \vec{x}} = \vec{w} - 2\lambda\vec{x} \overset{!}{=} 0 \rightarrow \vec{x}^* = \frac{\vec{w}}{2\lambda}$

b) $p(x) \sim \mathcal{N}(\vec{\mu}, \Sigma)$ $\max_{\boldsymbol{x}} \log p(\omega_c|\boldsymbol{x}) + \log p(\boldsymbol{x})$.

$$\arg\max_{\vec{x}} \vec{w}\cdot\vec{x} + b + \log\left(\frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})\right)\right)$$

$$\frac{\partial L}{\partial \vec{x}} = \vec{w}^T + \frac{1}{\mathcal{N}(\mu,\Sigma)} \cdot \mathcal{N}(\mu,\Sigma) \cdot \frac{\partial}{\partial\vec{x}}\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right)$$

$$= \vec{w}^T - \frac{1}{2}\frac{\partial}{\partial \vec{x}}\left[(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})\right]$$

use: $\frac{\partial(\vec{x}^T A \vec{x})}{\partial \vec{x}} = 2\vec{x}^T A$ if A is symmetric

$$= \vec{w}^T - (\vec{x}-\vec{\mu})^T \Sigma^{-1} \overset{!}{=} 0$$

$$\vec{w}^T = (\vec{x}-\vec{\mu})^T\Sigma^{-1} = \vec{x}^T\Sigma^{-1} - \vec{\mu}^T\Sigma^{-1}$$

$$\vec{w}^T + \vec{\mu}^T\Sigma^{-1} = \vec{x}^T\Sigma^{-1} \quad | \cdot \Sigma$$

$$\vec{w}^T\Sigma + \vec{\mu}^T = \vec{x}^T \quad |^T$$

$$\Sigma^T\vec{w} + \vec{\mu} = \vec{x}^*$$

c) $\vec{x} = A\underset{\uparrow\,\text{code}}{\vec{z}} + \vec{c}$

$\max_{\boldsymbol{z}\in\mathcal{Z}} \log p(\omega_c\,|\,g(\boldsymbol{z})) - \lambda\|\boldsymbol{z}\|^2$,

optimize in code space

$$\frac{\partial}{\partial\vec{z}}\left(\vec{w}\cdot(A\vec{z}+\vec{c}) + b) - \lambda\|z\|^2\right)$$

$\frac{\partial(\vec{a}^T\vec{x})}{\partial\vec{x}} = \vec{a}$

$$= \frac{\partial}{\partial\vec{z}}(\underbrace{\vec{w}^T\cdot A}_{\vec{a}^T}\vec{z}) - 2\lambda\vec{z}$$

$$= A\cdot\vec{w} - 2\lambda\vec{z} \rightarrow \vec{z}^* = \frac{A^T\cdot\vec{w}}{2\lambda}$$ (like part a if $A = \mathbb{1}$)

$\boxed{2}$ $f(x) = \vec{w}^T\vec{x}$

a) $R_i(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_i}\right)^2$. $\rightarrow \frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i}\sum_j w_j x_j = w_i$

$\rightarrow R_i(\vec{x}) = w_i^2$

b) $R_i(\boldsymbol{x}) = \frac{\partial f}{\partial x_i}\cdot x_i$. $\rightarrow R_i(\vec{x}) = \frac{\partial f}{\partial x_i}\Big|_{\vec{x}=\vec{x}_0} \cdot (x_i - x_{0i}) = w_i x_i$

b) $R_i(\boldsymbol{x}) = \frac{\partial f}{\partial x_i} \cdot x_i.$ $\rightarrow$ $R_i(\vec{x}) = \frac{\partial f}{\partial x_i}\Big|_{\vec{x}=\vec{x}_0} \cdot (x_i - x_{0i}) = w_i x_i$

c) min distance while $\vec{w}^T \vec{x} = 0$

$\rightarrow L = \|\vec{x} - \vec{x}_0\|^2 + \lambda \vec{w}^T \vec{x}_0$

$\frac{\partial L}{\partial \vec{x}_0} = -2(\vec{x} - \vec{x}_0) + \lambda \vec{w} \overset{!}{=} 0 \rightarrow \vec{x} - \frac{\lambda}{2} \vec{w} = \vec{x}_0 \quad \Big| \vec{w}^T \cdot$

$\vec{w}^T \vec{x} = \frac{\lambda}{2} \|\vec{w}\|^2$

$\Rightarrow \vec{x} - \frac{\lambda}{2} \vec{w} = \vec{x}_0 , \qquad \lambda = \frac{2 \cdot \vec{w}^T \vec{x}}{w^2}$

$\rightarrow \vec{x}_0 = \vec{x} - \frac{(\vec{w}^T \cdot \vec{x}) \vec{w}}{\|\vec{w}\|^2} = \vec{x} - (\hat{w}^T \vec{x}) \hat{w}$

$R_i(\vec{x}) = \frac{\partial f}{\partial x_i}\Big|_{\vec{x}=\vec{x}_0} \cdot (x_i - x_{0i}) = w_i \cdot \left(x_i - x_i + \frac{1}{w^2}(\vec{w}^T \cdot \vec{x}) w_i\right)$

$= \frac{w_i^2}{\|\vec{w}\|^2} (\vec{w}^T \cdot \vec{x})$