Deep Learning

Exercise 1 – Intro to EDF

Christian Reiser
Autonomous Vision Group
University of Tübingen / MPI-IS

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

MAX-PLANCK-GESELLSCHAFT

ellis
European Laboratory for Learning and Intelligent Systems

Outline

- Setup for Exercises
- Vector Calculus
- Numpy Tutorial

Setup for Exercises

ILIAS

- We organize the exercises using the ILIAS system
  `https://ovidius.uni-tuebingen.de/ilias3`
- Exercise sheets will be available in the ILIAS system. Please be aware of the **submission deadline**.
- You are eligible to finish the homework within a group up to 2 people, but **each person must submit a solution.**
- If you have any questions about the material, please ask at the **forum** on ILIAS.
- For personal questions send an email via ILIAS to the responsible TA.

.

Development environment setup

- Follow the instructions for your OS to install the Python package manager `conda`: `https://docs.conda.io/projects/conda/en/latest/user-guide/install/`
- Download the archive for exercise 1 and open a terminal in the `code` directory
- Creates a new environment named `deep_learning_ex_1` and installs required packages (numpy, etc.): `conda env create -f environment.yml`
- Before launching your notebook you need to activate the environment: `conda activate deep_learning_ex_1`
- Run this command from the directory where the jupyter notebooks are located: `jupyter notebook`

Vector Calculus

Derivative of a vector-to-vector function

Let $\mathbf{f}\colon \mathbb{R}^n \mapsto \mathbb{R}^m, \mathbf{y} = \mathbf{f}(\mathbf{x})$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

where $\frac{\partial f_i}{\partial x_j}(\mathbf{x})$ is the derivative of $\mathbf{f}$'s i-th output w.r.t. to the j-th component x_j of $\mathbf{f}$'s input vector $\mathbf{x}$.

Two frequent special cases

Scalar-to-scalar function, $n = m = 1$. Then $f \colon \mathbb{R}^1 \mapsto \mathbb{R}^1$.

$$f'(x) = \frac{\partial f}{\partial x}(x) \in \mathbb{R}^{1\times 1} \equiv \mathbb{R}$$

Vector-to-scalar function, $n > 1, m = 1$. Then $f \colon \mathbb{R}^n \mapsto \mathbb{R}^1$.

$$(\nabla_{\mathbf{x}} f)(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{1\times n}$$

The alternative notation $(\nabla_{\mathbf{x}} f)(\mathbf{x})$ is read as the gradient of f at $\mathbf{x}$ and is exclusviely used for vector-to-scalar functions.

Chain rule for vector-to-vector functions

Let $\mathbf{f} \colon \mathbb{R}^n \mapsto \mathbb{R}^m, \mathbf{f}(\mathbf{x}) = \ldots$ and $\mathbf{g} \colon \mathbb{R}^m \mapsto \mathbb{R}^p, \mathbf{g}(\mathbf{y}) = \ldots$.

If

$$\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{f}(\mathbf{x}))$$

then $\mathbf{h} \colon \mathbb{R}^n \mapsto \mathbb{R}^p$ and:

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}) = \frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\mathbf{f}(\mathbf{x}))\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})$$

Here the p by m matrix $\frac{\partial \mathbf{g}}{\partial \mathbf{y}}(\mathbf{f}(\mathbf{x}))$ gets multiplied by the m by n matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x})$ to form the resulting p by n matrix $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x})$.

Special case of chain rule

Let $\mathbf{f}\colon \mathbb{R}^1 \mapsto \mathbb{R}^n, \mathbf{f}(x) = \ldots$ and $g\colon \mathbb{R}^n \mapsto \mathbb{R}^1, g(\mathbf{y}) = \ldots$.
If

$$h(x) = g(\mathbf{f}(x))$$

then $h\colon \mathbb{R}^1 \mapsto \mathbb{R}^1$ and:

$$\frac{\partial h}{\partial x}(x) = \frac{\partial g}{\partial \mathbf{y}}(\mathbf{f}(x))\frac{\partial \mathbf{f}}{\partial x}(x) = \sum_{i=1}^{n} \frac{\partial g}{\partial y_i}(\mathbf{f}(x))\frac{\partial f_i}{\partial x}(x)$$

Here the 1 by n matrix $\frac{\partial g}{\partial \mathbf{y}}(\mathbf{f}(x))$ gets multiplied by the n by 1 matrix $\frac{\partial \mathbf{f}}{\partial x}(x)$ to form the resulting 1 by 1 matrix $\frac{\partial h}{\partial x}(x)$.

Numpy Tutorial

Questions?