Neural Networks

1. Introduction to Numpy

Center for Cognitive Science
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava

Thursday 22nd February, 2024

Numpy

- np.array, np.zeros, np.ones, np.eye
- np.random.rand – uniform, np.random.randn – normal/gaussian
- +, -, *, np.sin(. . .), np.exp(. . .) all work elementwise
- +=, -=, *= also work
- np.dot(a,b) = a.dot(b) = a @ b
- np.transpose(a) = a.transpose() = a.T
- transpose on a vector does nothing!
  - use np.inner or np.outer
  - alternatively, vector(x) or np.atleast_2d(x).T returns a matrix posing as a column vector
- np.linalg.norm
- np.min/max/mean/std(x, axis=0)

Conventions, vol. 1

c – scalar
$\boldsymbol{x}$ – column vector ("western" algebra)
x_i – i-th element of a vector (scalar)
$\boldsymbol{W}$ – matrix
$\boldsymbol{w}_i$ – i-th row of matrix (vector)
$w_{i,j}$ – element in the i-th row and j-th column (scalar)

Conventions, vol. 2

$c\boldsymbol{x}$, $c\boldsymbol{W}$ – scalar/elementwise multiplication – shape is preserved
$\boldsymbol{WX}$ – matrix multiplication
$\boldsymbol{Wx}$ – matrix-vector multiplication (i.e. applying transformation W to a vector x)
$\boldsymbol{xy}$ – mistake
$\boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y}$ – scalar/dot/inner product – result is a scalar
$\boldsymbol{x} \times \boldsymbol{y}$ – vector/cross product – result is a vector
$\boldsymbol{xy}^T = \boldsymbol{x} \otimes \boldsymbol{y}$ – outer product – result is a matrix

Task: linear regression using Numpy

Prepare data

- create vector of length N, $\boldsymbol{x} = (0, 1, ..., N-1)$
- generate slope k and shift q
- compute vector $\boldsymbol{y}$:
- $\boldsymbol{y} = k\boldsymbol{x} + q$
- add noise to each value of $\boldsymbol{y}$

Task: linear regression using Numpy

Compute regression

- create matrix $\boldsymbol{X} = \begin{bmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ x_{N-1} & 1 \end{bmatrix}$
- find predicted slope $\hat{k}$ and shift $\hat{q}$:
  $\hat{k},\ \hat{q} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$
- compute predicted vector $\hat{\boldsymbol{y}}$:
  $\hat{\boldsymbol{y}} = \hat{k}\boldsymbol{x} + \hat{q}$

Task: linear regression using Numpy

Result