MLSM 2020 - Quiz 4
26/10/2020 - 04/11/2020 at 23:59

Questions based on Lecture 6 and 7

(1) (1.0 pt.)

Let $\mathbf{K}_1$ and $\mathbf{K}_2$ be two kernel matrices. They are positive semi-definite and symmetric, with the same size. Which of these three operations does not provide a correct kernel matrix in the general case?

Be aware, the one false statement needs to be selected! Think about the differences between the expressions of the elements of the product matrices.

(1) The linear combination of $\mathbf{K}_1$ and $\mathbf{K}_2$ with non-negative real weights.
(2) The itemwise (pointwise, Hadamard) product of $\mathbf{K}_1$ and $\mathbf{K}_2$.
(3) The matrix product of $\mathbf{K}_1$ and $\mathbf{K}_2$.

(2) (1.0 pt.)

A polynomial kernel has the form $\kappa_{pol}(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^q$. Let $\kappa(\mathbf{x}, \mathbf{z})$ be any valid kernel. Let the kernel $\kappa(\mathbf{x}, \mathbf{z})$ substitute the $\mathbf{x}^T\mathbf{z}$ term in the polynomial kernel, namely we have $\kappa_{pol\_mod}(\mathbf{x}, \mathbf{z}) = (\kappa(\mathbf{x}, \mathbf{z}) + c)^q$. Which of these statements is true?

Hint: How is the kernel function defined in the corresponding Hilbert space? How can it be represented?

(1) κ_{pol_mod} is a valid, positive semi-definite kernel for any κ.
(2) There are cases where κ_{pol_mod} is not positive semi-definite.
(3) κ_{pol_mod} is only a valid kernel if κ is a Gaussian one.

(3) (1.0 pt.) Let the feature vector of a polynomial kernel, $\kappa_{pol}(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^q$, be represented explicitly. What is the dimension of the explicit feature vector? Assume that the degree of the polynomial is 3, and the polynomial kernel is defined on the vector space of dimension 2.

Hint: See the representation of the polynomial kernel in Lecture 6.

(1) 6
(2) 10
(3) 15

(4) (2.0 pt.)

Implement a regression problem via the backprogation algorithm of the neural network based learner, Lecture 7, Slides about the "Backpropagation training algorithm". Both the input and the output are one dimensional real numbers. The network contains an input layer with one node, a hidden layer with $H = 5$ nodes, and on output layer with one node. The activation function in the hidden layer is the sigmoid function, see on the corresponding slides. No bias is considered when the weights are applied.

The data description, and the learning parameters for the neural network are given by the following python code.

```
import numpy as np

  ## Load the data
  m0=100                          ## m0+1 gives the index of the 0 input
  m=2*m0+1                        ## number of points
  xx=2*np.arange(m+1)/m-1         ## input data is in the range [-1,+1], with steps 0.01
  rr=xx**2                        ## output values: a parabola

  ## set the learner parameters
  niteration=100                  ## number of iterations
  H=5                             ## number of nodes in the hidden layer
  eta=0.3                         ## learning speed, step size

  ## Random initialization of the edge weights
  ## Set the seed to a fixed number
  np.random.seed(12345)
  ## weights between the input and the hidden layers
  W=2*(np.random.rand(H)-0.5)/100          ## random uniform in [-0.01,0.01]
  ## weights between the hidden and the output layers
  V=2*(np.random.rand(H)-0.5)/100          ## random uniform in [-0.01,0.01]
```

Be aware, the initialization of the weights needs to follow the scheme shown above to receive the proper result. The data points are processed sequentially from -1 to 1 without randomization.

Assume that the vector yy of size m contains the predicted values for all input examples. Which interval contains the error after $niteration = 100$, at the input value 0, when the error is computed in this way $rr[m0 + 1] - yy[m0 + 1]$?

(1) $[+0.2, +0.3]$
(2) $[-0.1, +0.1]$
(3) $[-0.3, -0.2]$