Introduction to Machine Learning
Answers to Exercise 3 - Kernels & Neural Networks

Jingtao Min

April 14, 2022

1 Kernels

(a) Given dataset $X = \{\mathbf{x}_i\}_{i=1,2} = \{(-3,4),(1,0)\}$, and feature map $\phi(\mathbf{x}) = (x^{(1)}, x^{(2)}, \|\mathbf{x}\|)$, the mapped features are given by:

$$\phi(\mathbf{x}_1) = (-3,4,5), \quad \phi(\mathbf{x}_2) = (1,0,1) \tag{1}$$

the Gram matrix (inner product matrix) is given by:

$$\mathbf{G} = \begin{bmatrix} \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_1)\rangle & \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_2)\rangle \\ \langle\phi(\mathbf{x}_2),\phi(\mathbf{x}_1)\rangle & \langle\phi(\mathbf{x}_2),\phi(\mathbf{x}_2)\rangle \end{bmatrix} = \begin{bmatrix} 50 & 2 \\ 2 & 2 \end{bmatrix} \tag{2}$$

(b) Valid kernels.

(1) $k(x,y) = \frac{1}{1-xy}$ where $x,y \in (-1,1)$ is a valid kernel. This is an inner product kernel $k(x,y) = h(\langle x,y\rangle)$ where $h(z) = (1-z)^{-1}$ $(z \in (-1.1))$. We note that the Taylor series of $h(z)$:

$$h(z_0+dz) = \sum_{n=0}^{\infty} \frac{h^{(n)}(z_0)}{n!} dz^n = \sum_{n=0}^{\infty} \frac{(1-z_0)^{-(n+1)}}{n!} dz^n \tag{3}$$

has (strictly) positive coefficients for all $z_0 \in (-1,1)$. Therefore, according to the inner product kernel property, this is a valid kernel.

(2) $k(x,y) = 2xy$ with $x,y \in \mathbb{N}$ is a valid kernel. This is again an inner product kernel with $h(z) = 2^z$ where $z \in \mathbb{N}$. It is also apparent that its derivatives are all (strictly positive), i.e.

$$h(z_0+dz) = \sum_{n=0}^{\infty} \frac{h^{(n)}(z_0)}{n!} dz^n, \qquad h^{(n)} = \frac{d^n}{dz^n} 2^z = (\ln 2)^n 2^z > 0 \, (\forall z \in \mathbb{N}) \tag{4}$$

Therefore it is also a valid kernel.

(3) $k(x,y) = \cos(x+y)$ with $x,y \in \mathbb{R}$ is NOT a valid kernel. One can verify this with a simple counterexample: $x = \frac{\pi}{4}$, $y = \frac{3\pi}{4}$. The resulting kernel matrix:

$$\mathbf{K} = \begin{bmatrix} \cos\frac{\pi}{2} & \cos\pi \\ \cos\pi & \cos\frac{3\pi}{2} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \qquad |\lambda\mathbf{I} - \mathbf{K}| = \lambda^2 - 1 = 0 \quad \Longrightarrow \quad \lambda = \pm 1 \tag{5}$$

has eigenvalues -1, and is hence not positive semi-definite. Therefore it is not a valid kernel.

(4) $k(x,y) = \cos(x-y)$ with $x,y \in \mathbb{R}$ is a valid kernel. This can be decomposed into valid inner product kernels with trigonometric features:

$$k(x,y) = \cos(x,y) = \cos x \cos y + \sin x \sin y = \langle\cos(x),\cos(y)\rangle + \langle\sin(x),\sin(y)\rangle \tag{6}$$

Since $h(z) = z$ has non-negative derivatives, inner product kernel $k_0(u,v) = \langle u,v\rangle$ is of course valid. It so follows that $k_c(x,y) = k_0(\cos(x),\cos(y))$ and $k_s(x,y) = k_0(\sin(x),\sin(y))$ are both valid, and so is their sum $k(x,y) = \cos(x-y)$.

(5) $k(x, y) = \max(x, y)$ where $x, y \in \mathbb{R}^+$ is NOT a valid kernel. One can verify this with a simple counterinstance: $0 < x < y$. The resulting kernel matrix:

$$\mathbf{K} = \begin{bmatrix} \max(x,x) & \max(x,y) \\ \max(y,x) & \max(y,y) \end{bmatrix} = \begin{bmatrix} x & y \\ y & y \end{bmatrix} \qquad |\lambda \mathbf{I} - \mathbf{K}| = (\lambda - x)(\lambda - y) - y^2 = \lambda^2 - (x+y)\lambda + y(x-y) = 0 \tag{7}$$

will always have negative eigenvalue since $y(x - y) < 0$, hence is not positive semi-definite. Therefore it is not a valid kernel.

(6) $k(x, y) = \frac{\min(x,y)}{\max(x,y)}$ with $x, y \in \mathbb{R}^+$ is a valid kernel. Invoking the valid kernel $k_m(x, y) = \min(x, y)$ and the nonlinear mapping $\phi(z) = z^{-1}$, we can decompose the kernel as:

$$k(x, y) = \frac{\min(x, y)}{\max(x, y)} = \min(x, y) \cdot \min\left(\frac{1}{x}, \frac{1}{y}\right) = k_m(x, y)\, k_m\left(\phi(x), \phi(y)\right) \tag{8}$$

According to the composition of valid kernels, the resulting kernel is valid.

(c) Assuming $k(x, y)$ is a valid kernel, the following kernels:

(a) $k_a(x, y) = f(k(x, y))$ is a valid kernel where $f : \mathbb{R} \mapsto \mathbb{R}$ is a polynomial with non-negative coefficients. The kernel k_a would take the explicit form:

$$k_a(x, y) = \sum_{n=0}^{N} a_n \left[k(x, y)\right]^n, \quad a > 0 \tag{9}$$

According to the product rule, $k_n(x, y) = [k(x, y)]^n$ is a valid kernel; and according to scaling and summation rule, $k_a = \sum a_n k_n$ is also valid as $a_n > 0$.

(b) $k_b(x, y) = f(k(x, y))$ where f is an arbitrary polynomial might not be valid. The simple counterexample would be $f(z) = -z$. This would convert any positive definite kernel to negative definite.

(c) $k_c(x, y) = \exp\left(k(x, y)\right)$ is a valid kernel. A plausible proof comes from the fact that exponential function can be approximated to arbitrary precision by its Taylor series, which has strictly positive coefficients:

$$k_c(x, y) = \exp\left(k(x, y)\right) \approx k_{c,N}(x, y) = \sum_{n=0}^{N} \frac{1}{n!}\left[k(x, y)\right]^n = \sum_{n=0}^{N} \frac{1}{n!} k_n(x, y) \tag{10}$$

And thus the kernel k_c, N approximated by $N + 1$ terms in the series must be valid. Using strict language, one should be able to prove $k_c = \lim_{N \to +\infty} k_{c,N}$ is valid.

(d) $k_d(x, y) = g(x)k(x, y)g(y)$ where $g : \mathbb{R} \mapsto \mathbb{R}^+$ is a valid kernel. This can be viewed as a product of a known valid kernel and an inner product kernel with feature mapping:

$$k_d(x, y) = k(x, y)\langle g(x), g(y)\rangle = k(x, y) \cdot h(\langle g(x), g(y)\rangle), \qquad h(z) = z \tag{11}$$

Therefore the kernel is valid.

(e) $k_e(x, y) = h(x)k(x, y)h(y)$ where $h : \mathcal{X} \mapsto \mathbb{R}$ is a valid kernel for the same reason above.

(f) $k_f(x, y) = k(\phi(x), \phi(y))$ is a valid kernel.

1.1 Kernelized Hinge Loss

(a) Consider the l^2-regularized hinge loss:

$$L_h(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\right) + \lambda \mathbf{w}^T \mathbf{w} \tag{12}$$

In this case the features are just linear features ($\phi(\mathbf{x}) = \mathbf{x}$), thus the weights $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i = \mathbf{X}^T \boldsymbol{\alpha}$ in kernel formulation. Plugging in this expression:

$$L_h(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \boldsymbol{\alpha}^T \mathbf{X}\mathbf{x}_i\right) + \lambda \boldsymbol{\alpha}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\alpha} \tag{13}$$

(b) Top-left: neural network (1 hidden layer with ReLU); top-right: Gaussian kernel SVM; bottom-left: polynomial kernel (order=2) SVM; bottom-right: linear SVM.

2 Neural Networks

2.1 Grade Prediction

(a) The unit output in the first hidden layer:

$$a_i^{(1)} = \sigma\left(\sum_k w_{ki}^{(1)} x_k\right), \qquad \sigma(z) = \frac{1}{1+e^{-z}} \tag{14}$$

(b) The unit output in the 2nd hidden layer:

$$a_i^{(2)} = \sigma\left(\sum_k w_{ki}^{(2)} a_k^{(1)}\right), \qquad \sigma(z) = \frac{1}{1+e^{-z}} \tag{15}$$

(c) Final output:

$$f = w_1^{(3)} a_1^{(2)} + w_2^{(3)} a_2^{(2)} \tag{16}$$

(d) Suppose the 2nd hidden layer is subject to dropout, with a retaining probability of 0.4. We invoke the random variable S_i that controls the existence of $a_i^{(2)}$ during training. Expectation of output function f with dropout applied during training:

$$\begin{aligned}
\mathbb{E}\left[f|(x_1,x_2,x_3)\right] &= \mathbb{E}\left[w_1^{(3)} a_1^{(2)} S_1 + w_2^{(3)} a_2^{(2)} S_2\right] \\
&= w_1^{(3)} a_1^{(2)} \mathbb{E}[S_1] + w_2^{(3)} a_2^{(2)} \mathbb{E}[S_2] \\
&= 0.4\left[w_1^{(3)} a_1^{(2)} + w_2^{(3)} a_2^{(2)}\right]
\end{aligned} \tag{17}$$

(e) Variance of the output function:

$$\begin{aligned}
\mathrm{Var}\left[f|(x_1,x_2,x_3)\right] &= \mathbb{E}\left[(f-\mathbb{E}[f])^2\right] = \mathbb{E}\left[\left(\sum_i w_i^{(3)} a_i^{(2)} (S_i - p)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{ij} w_i^{(3)} w_j^{(3)} a_i^{(3)} a_j^{(3)} (S_i-p)(S_j-p)\right] \\
&= \sum_{ij} w_i^{(3)} w_j^{(3)} a_i^{(3)} a_j^{(3)}\, \mathbb{E}\left[(S_i-p)(S_j-p)\right] \\
&= \sum_{ij} w_i^{(3)} w_j^{(3)} a_i^{(3)} a_j^{(3)} \times \begin{cases} \mathbb{E}\left[(S_i-p)^2\right] = p(1-p) & (i=j) \\ \mathbb{E}\left[S_i-p\right]\mathbb{E}\left[S_j-p\right] = 0 & (i\neq j) \end{cases} \\
&= \sum_{ij} w_i^{(3)} w_j^{(3)} a_i^{(3)} a_j^{(3)} \cdot p(1-p)\delta_{ij} = p(1-p)\sum_i \left(w_i^{(3)} a_i^{(3)}\right)^2 \\
\mathrm{Var}\left[f|(x_1,x_2,x_3)\right] &= 0.24\left[(w_1^{(3)} a_1^{(2)})^2 + (w_2^{(3)} a_2^{(2))^2}\right]
\end{aligned} \tag{18}$$

(f) Expectation of loss function, with inputs and label as random variables:

$$\begin{aligned}
\mathbb{E}[L] &= \mathbb{E}\left[(y-f)^2\right] = \mathbb{E}\left[y^2 + f^2 - 2yf\right] \\
&= Y^2 + \mathbb{E}[f^2] - 2Y\mathbb{E}[f] \\
&= Y^2 + (\mathbb{E}[f])^2 + \mathrm{Var}[f] - 2Y\mathbb{E}[f] \\
&= Y^2 - 2Y\mathbb{E}[f] + \mathrm{Var}[f] + (\mathbb{E}[f])^2
\end{aligned} \tag{19}$$

(g) During training, if the unit $a_1^{(2)}$ is dropped out while $a_2^{(2)}$ is kept, the derivative with respect to $w_{21}^{(1)}$ is given by:

$$\begin{aligned}\frac{\partial L}{\partial w_{21}^{(1)}} &= \frac{\partial L}{\partial f}\frac{\partial f}{\partial \mathbf{a}^{(2)}}\frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{a}^{(1)}}\frac{\partial \mathbf{a}^{(1)}}{\partial w_{21}^{(1)}} = \frac{\partial L}{\partial f}\frac{\partial f}{\partial a_2^{(2)}}\frac{\partial a_2^{(2)}}{\partial a_1^{(1)}}\frac{\partial a_1^{(1)}}{\partial w_{21}^{(1)}} \\ &= 2(f-y)\cdot w_2^3 \cdot \sigma'\left(\sum_k w_{k2}^{(2)} a_k^{(1)}\right) w_{12}^{(2)} \cdot \sigma'\left(\sum_k w_{k1}^{(1)} x_k\right) x_2 \\ &= 2(f-y)\, w_2^3\, \sigma'\left(w_{12}^{(2)} a_1^{(1)} + w_{22}^{(2)} a_2^{(1)}\right) w_{12}^{(2)}\, \sigma'\left(w_{11}^{(1)} x_1 + w_{21}^{(1)} x_2 + w_{31}^{(1)} x_3\right) x_2\end{aligned} \tag{20}$$

2.2 Expressiveness

(a) Note the output using one layer:

$$Y = \sigma\left(w_0 + w_1 x_1 + w_2 x_2\right) = \frac{1}{1+\exp\{-(w_0 + w_1 x_1 + w_2 x_2)\}} \tag{21}$$

For constructing a logical OR function $Y = x_1 \vee x_2$ with threshold value 0.5, the boundaries are partitioned by $\exp(-z) = 1$ or $z = 0$. The requirements are explicitly stated:

$$\begin{aligned} w_0 &< 0 \\ w_0 + w_1 &\geq 0 \\ w_0 + w_2 &\geq 0 \\ w_0 + w_1 + w_2 &\geq 0 \end{aligned} \tag{22}$$

Choosing from the allowed set of values $\{-0.5, 0, 1\}$, we have:

$$\begin{aligned} w_0 &= -0.5 \\ w_1 &= 1 \\ w_2 &= 1 \end{aligned} \tag{23}$$

(b) For implementation of a logical AND function $Y = x_1 \wedge x_2$, we have requirements:

$$\begin{aligned} w_0 &< 0 \\ w_0 + w_1 &< 0 \\ w_0 + w_2 &< 0 \\ w_0 + w_1 + w_2 &\geq 0 \end{aligned} \tag{24}$$

Choosing from the allowed set of values $\{-2, -1.5, -1, -0.5, 0, 0.5, 1\}$, we have:

$$\begin{aligned} w_0 &= -2 \\ w_1 &= 1 \\ w_2 &= 1 \end{aligned} \tag{25}$$