Exercise 6

Exercise 1a: Write a test in Python that verifies empirically that the kernel is positive semi-definite and run it.

```
In [39]: import numpy as np

In [40]: def conv_kernel(x,y):
             return np.sum(np.square(np.convolve(x,y)))

In [41]: def perform_psd_test(kernel_function, inputs):
             i,j=0,0
             A=np.zeros((len(inputs),len(inputs)))

             for x in inputs:
                 j=0
                 for y in inputs:
                     A[i,j]=kernel_function(x,y)
                     j+=1
                 i+=1

             return np.all(np.linalg.eigvals(A) >= 0)

In [44]: def random_ts_tests():
             # perform psd test with random time series multiple times
             outcomes=[]
             for i in range(1000000):
                 # generate 2 random time series of length 10
                 inputs=[]
                 for j in range(2):
                     inputs.append(np.random.rand(100)*10)

                 outcomes.append(perform_psd_test(conv_kernel, inputs))
             return(np.all(outcomes))

In [45]: random_ts_tests()

Out[45]: True
```

The positive semi definiteness of the Gram matrix has been shown with 1000000 different pairs of time series of length 100. The positive semi definiteness of the kernel function has therefore been shown empirically.

Exercise 1b:

K is said to be positive semi-definite kernel if and only if:

$$\sum_{i=1}^{m}\sum_{j=1}^{m} K(x_i, x_j)c_i c_j \geq 0$$

we have the convolution kernel:

$$K(x, x^{'}) = ||x * x^{'}||^2$$

and:

$$||x * x^{'}||^2 = ||x^{'} * x||^2$$

then:

$$\begin{aligned}
\sum_{ij}^{m} c_i c_j k_{ij} &= \sum_{ij}^{m} c_i c_j < x_i * x_i, x_j * x_j > \\
&= \sum_{ij}^{m} < c_i(x_i * x_i), c_j(x_j * x_j) > \\
&= Trace\Big((\sum_i c_i(x_i * x_i))(\sum_j c_j(x_j * x_j)) \Big) \\
&= Trace(Z^\top Z) \\
&= ||Z||^2 \geq 0
\end{aligned}$$

Where:

$$Z = \Big(\sum_i c_i(x_i * x_i) \Big)$$

we can conclude that the feature map(starting point of kernel trick) is given by: $\phi(x) = x * x$ since:

$$k(x, x^{'}) =< \phi(x), \phi(x^{'}) >=< x * x, x' * x' >$$

Exercise 2a:

We want to show that

$$\sum_{i=1}^{K}\sum_{j=1}^{K} \alpha_i \alpha_j k(x_i, x_j) \geq 0\,. \tag{1}$$

First we insert the definition of the kernel and rearrange

$$\begin{aligned}
&= \sum_{i=1}^{K}\sum_{j=1}^{K} \alpha_i \alpha_j \sum_{m=1}^{M} \beta_m \sum_{n=1}^{N-m+1} I(u_{m,n}(x_i) = u_{m,n}(x_j)) \\
&= \sum_{m=1}^{M} \beta_m \sum_{n=1}^{N-m+1} \sum_{i=1}^{K}\sum_{j=1}^{K} \alpha_i \alpha_j \cdot I(u_{m,n}(x_i) = u_{m,n}(x_j))\,.
\end{aligned}$$

Fix m and n and note that the indicator function is symmetric and transitive, that is for all $1 \leq i, j, k \leq K$ whenever $u_{m,n}(x_i) = u_{m,n}(x_j)$ we have $u_{m,n}(x_j) = u_{m,n}(x_i)$, and whenever $u_{m,n}(x_i) = u_{m,n}(x_j)$ and $u_{m,n}(x_j) = u_{m,n}(x_k)$ it follows that $u_{m,n}(x_i) = u_{m,n}(x_k)$. For the sum above it follows that if a term $\alpha_i \alpha_j$ does not vanish, $\alpha_j \alpha_i$ does not vanish as well. Further if $\alpha_i \alpha_k$ does not vanish, so does $\alpha_j \alpha_k$. Define the set $S = \{i \mid \exists j \neq i \text{ s.t. } I(u(x_i) = u(x_j)) = 1\}$. We can then rewrite the sum as

$$= \sum_{m=1}^{M} \beta_m \sum_{n=1}^{N-m+1} \left((\sum_{i \in S} \alpha_i)^2 + \sum_{i \notin S} \alpha_i^2 \right)_{m,n} ,$$

where $(\sum_{i \in S} \alpha_i)^2 = \alpha_{i_1}^2 + 2\alpha_{i_1}\alpha_{i_2} + 2\alpha_{i_1}\alpha_{i_3} + ... + \alpha_{i_2}^2 + 2\alpha_{i_2}\alpha_{i_3} + ... + \alpha_{i_3}^2 + 2\alpha_{i_3}\alpha_{i_4} + ...$ consists of all the terms that do not vanish as described above.
The term in the brackets is clearly greater or equal to zero, since $\beta_m \geq 0$ for all m, it follows that inequality (1) holds.

Exercise 2b:

For the special case $M = 1$ the kernel reduces to $k(x, x') = \sum_{n=1}^{N} I(u_{1,n}(x) = u_{1,n}(x'))$. We want to find a map ϕ such that $k(x, x') = \phi(x)^T \phi(x')$. Define the set of all gene sequences to be $\mathcal{A}^N$ (where we have assumed that all sequences are of length N and the language $L = \{A, C, T, G\}$. We define $\phi : \mathcal{A}^N \to \mathbb{R}^{N \cdot |L|^m}$, $\phi : x \mapsto (1 \text{ if } x_n = l)_{l,n}$ for $l \in L$ and $1 \leq n \leq N$. For example $\phi(AC) = (1, 0, 0, 0, 0, 1, 0, 0)^T$. The corresponding 'encoding vector' is $(A1, C1, T1, G1, A2, C2, T2, G2)$ where the number stands for the position n of the letter l in the sequence.

Exercise 2c:

For the special case $M = 2$ and $\beta_0 = 0, \beta_1 = 1$ the kernel function reduces to $k(x, x') = \sum_{n=1}^{N} I(u_{2,n}(x) = u_{2,n}(x'))$. As in (b) we define $\phi : \mathcal{A}^N \to \mathbb{R}^{N \cdot |L|^m}$, $\phi : x \mapsto (1 \text{ if } x_n = l)_{l,n}$ for $l \in L$ and $1 \leq n \leq N$. The 'encoding vector' now corresponds to all possible combinations of length two and all $N - 1$ possible positions. For example when $N = 3$, the 'encoding vector' is $(AA1, AC1, AT1, AG1, CA1, ..., GT2, GG2)$.