Exercise 05: The quenched solution to randomised perceptron learning

In this exercise we basically follow the same procedures of the last exercise, but instead of having the **annealed** analytical solution we use the **quenched** one and compare it with the data we obtained from Exercise 03.

We hence have

$$\frac{R}{\sqrt{1-R}} = \frac{\alpha}{\pi}\, I(R) \tag{1}$$

where

$$I(R) = \int_{-\infty}^{+\infty} \frac{dv}{2\pi} \frac{\exp[-(1+R)\, v^2/2]}{H(-\sqrt{R}v)}$$

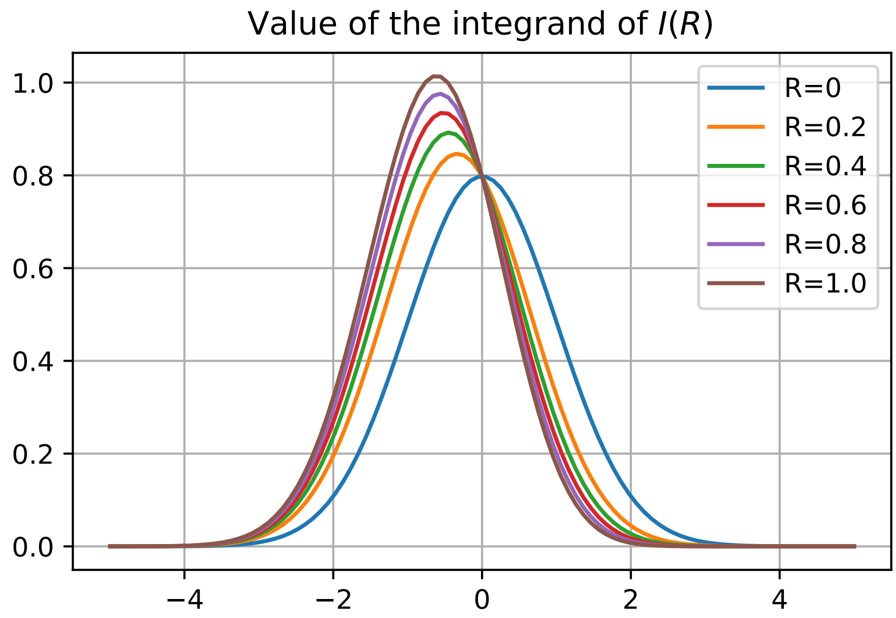
and

$$H(u) = \int_{u}^{+\infty} \frac{dx}{\sqrt{2\pi}} \exp(-x^2/2) = \frac{1}{2}\mathrm{erfc}(\frac{u}{\sqrt{2}}).$$

Recall that

$$\epsilon(R) = \frac{1}{\pi} \arccos(R).$$

We first make sure that the integrand that we chose can be effectively be integrated in a limited interval, such as $[-10, +10]$, without any heavy consequence over the computations. We can check that it has a bell-shape in the $[-5, +5]$ interval and elsewhere it is almost zero, for our domain of interest $R \in [0, 1.)$. We choose anyways, just to be safe, to integrate the function between $[-20, +20]$.



Step 1: Choose a parameter and represent $\epsilon(\alpha)$ in a parametric plot

Since it is present in both functions ϵ and α, I will choose R as my parameter of choice.

$$t \equiv R$$

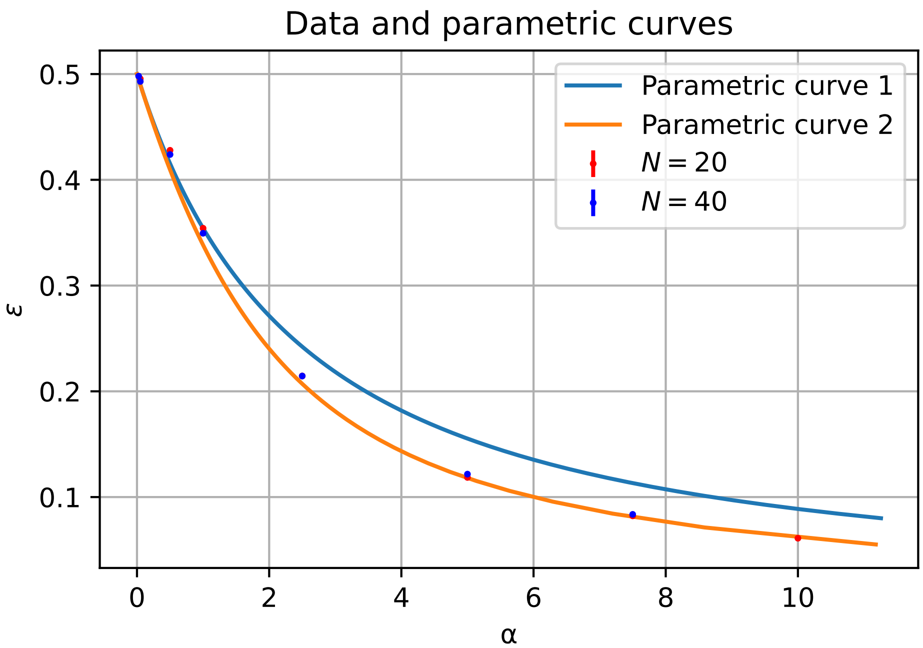
I will hence have the following parametric expressions:

$$\epsilon(t) = \frac{\arccos(t)}{\pi}$$

and

$$\alpha(t) = \frac{\pi t}{\sqrt{1-t}\; I(t)}.$$

The superimposition between data, parametric curve from the annealed computation (parametric curve 1) and parametric curve from the quenched computation (parametric curve 2) will look like this:

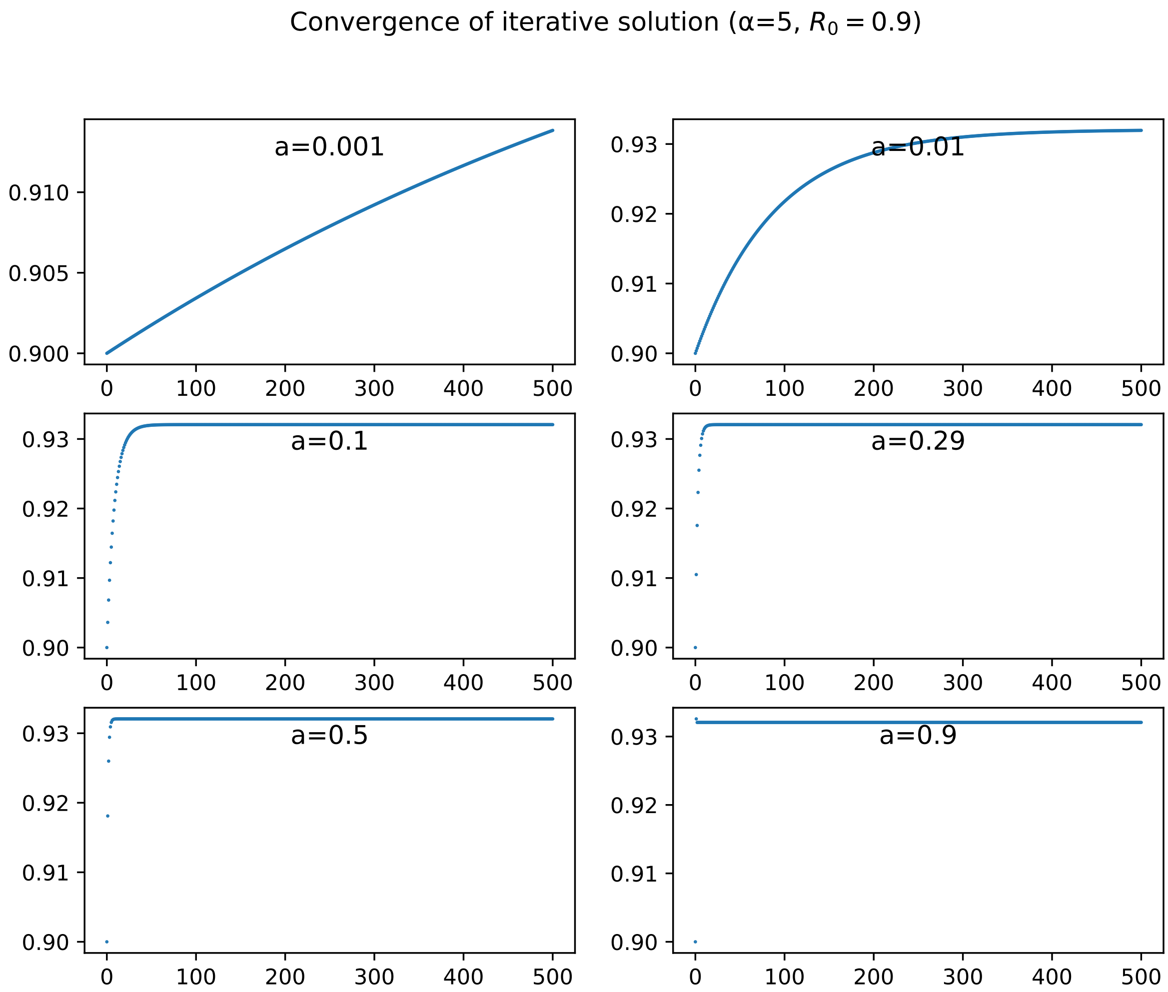


Step 2: Devise a strategy to explicitly solve equation (1) for, say, $\alpha = 5$

We will, for example, rewrite the equation in an iterative form such as

$$R \equiv f(R, \alpha) = 1 - (\frac{\pi R}{\alpha\, I(R)})^2$$

and see whether for different values of the a, we get a convergence. We can see in the following plots that the algorithm converges in all cases.



Step 3: Check the possible choices of f and verify if they converge

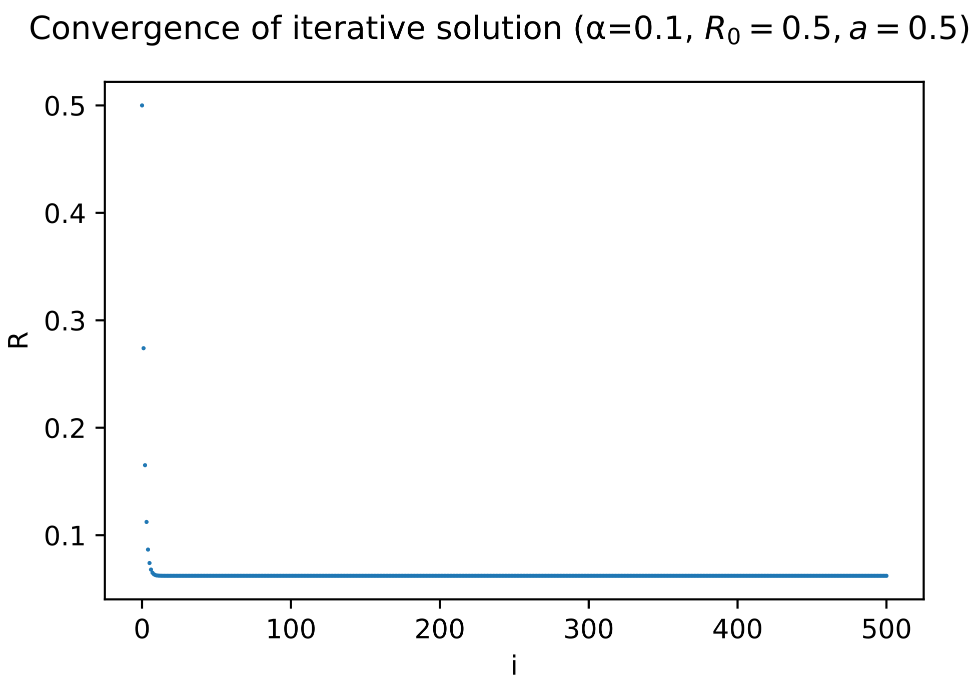
Two possibilities are proposed:

$$f_1(R, \alpha) = \frac{\alpha}{\pi}\sqrt{1-R}\; I(R)$$

and

$$f_2(R, \alpha) = 1 - (\frac{\pi R}{\alpha\; I(R)})^2$$

which is, by the way, the function that we used in step 2.



Instead if we try to do the same with $\alpha = 5$ we get:

```
Error of numerical integral is higher than 0.001: nan
```

```
/tmp/ipykernel_38261/2132753670.py:2: RuntimeWarning: invalid value encountered in sqrt
  return alpha * np.sqrt(1-R) * vI_func(R) / np.pi
/tmp/ipykernel_38261/514324215.py:5: IntegrationWarning: The occurrence of roundoff error is
  the requested tolerance from being achieved.  The error may be
  underestimated.
  integrate_result = integrate.quad(lambda x: I_integrand(R, x), -lim, +lim)
```

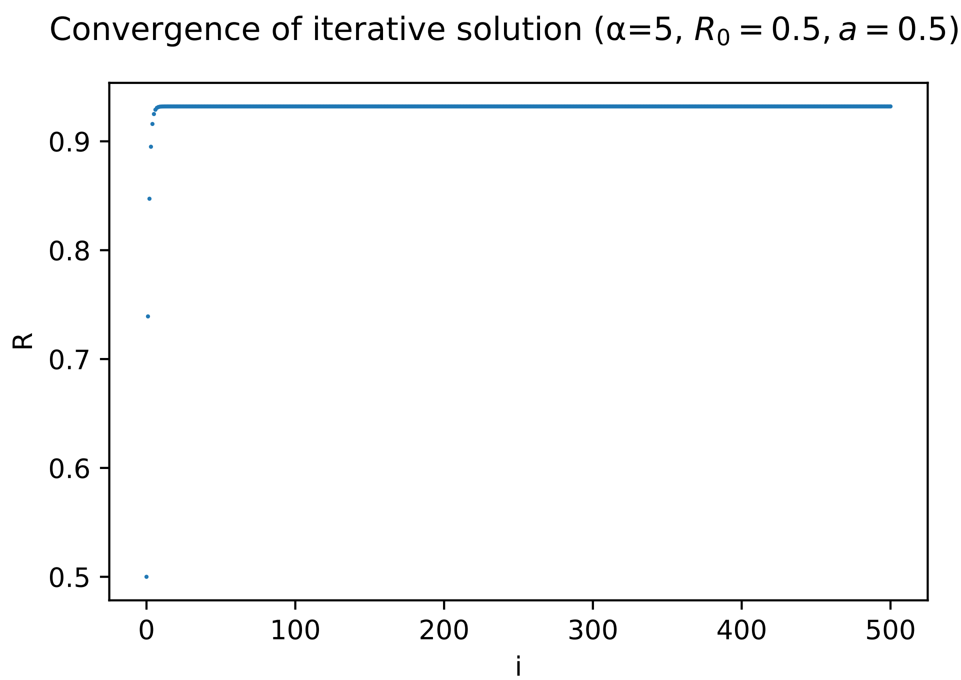
```
<Figure size 1650x1050 with 0 Axes>
```

As we can see the integration in this case explodes and yields to an error. This is probably due to convergence properties not being satisfied by f_1. Instead f_2, as we can see in the following, converges for high values of α but not for low ones.

```
Error of numerical integral is higher than 0.001: nan
```

```
/tmp/ipykernel_38261/514324215.py:2: RuntimeWarning: invalid value encountered in sqrt
  return 2. * np.exp(-(1.+R) * v*v / 2.) / (np.sqrt(2.*np.pi) * erfc(-np.sqrt(R/2.) * v))
/tmp/ipykernel_38261/514324215.py:2: RuntimeWarning: overflow encountered in exp
  return 2. * np.exp(-(1.+R) * v*v / 2.) / (np.sqrt(2.*np.pi) * erfc(-np.sqrt(R/2.) * v))
/tmp/ipykernel_38261/514324215.py:5: IntegrationWarning: The occurrence of roundoff error is
  the requested tolerance from being achieved.  The error may be
  underestimated.
  integrate_result = integrate.quad(lambda x: I_integrand(R, x), -lim, +lim)
```

```
<Figure size 1650x1050 with 0 Axes>
```

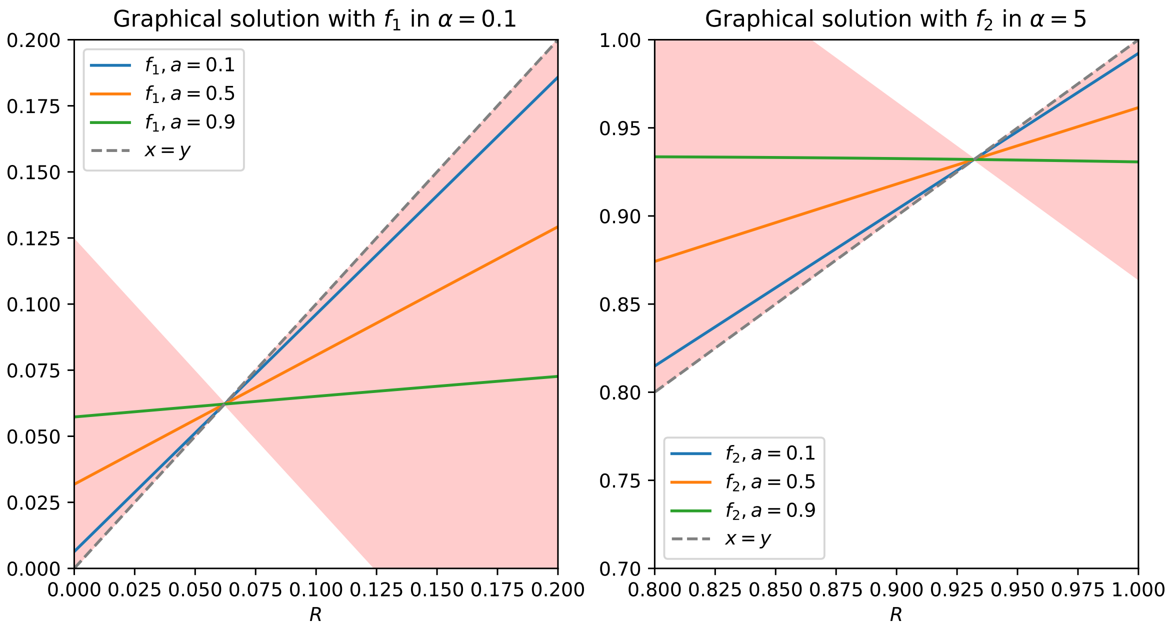


Step 4: Is any of the two choices suitable over the entire range?

No, both have limitations over the range where they yield to convergent results.

Step 5: Set up an automatic procedure to determine $\epsilon(\alpha)$ over $\alpha \in [0, 10]$

Let's see the recursive functions plotted in order to understand the situation better.

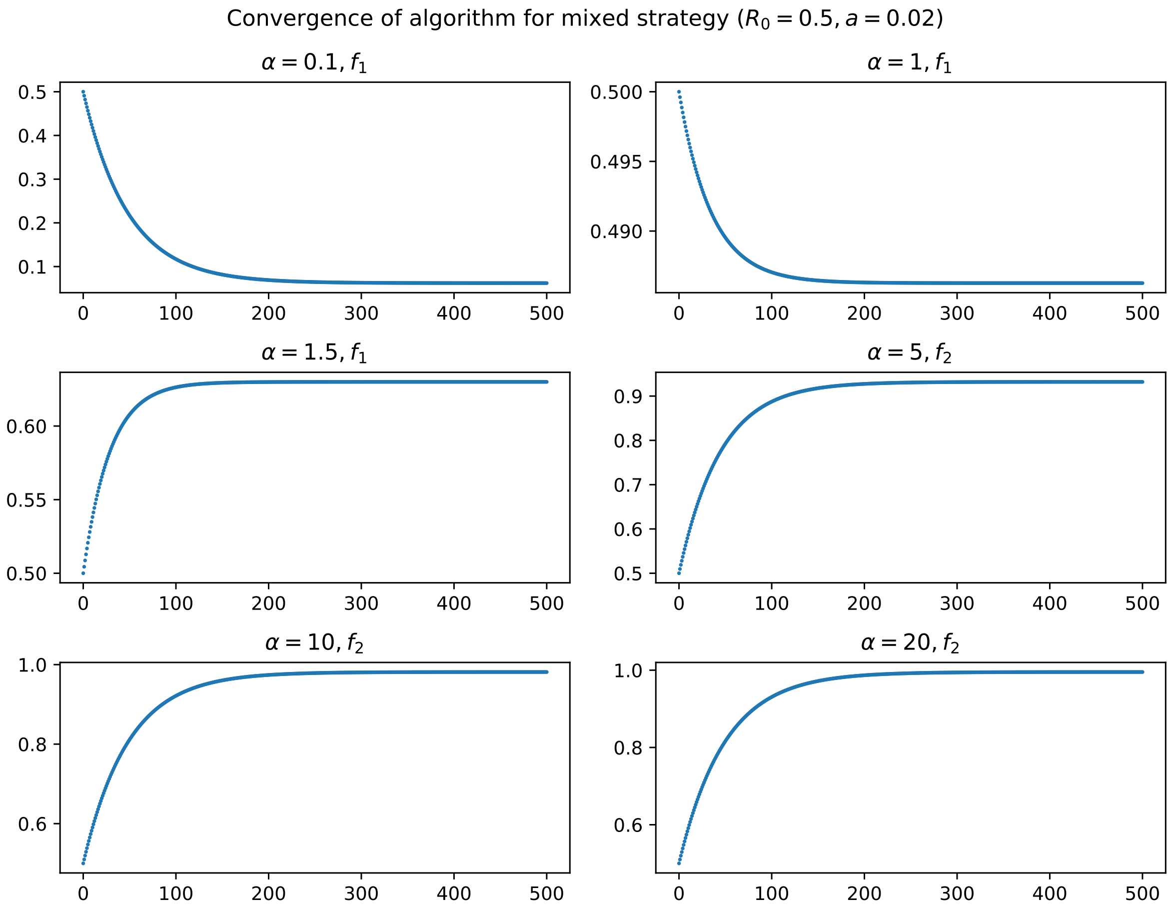


The choice of the function over the α's seems to give us some problems. Analytical computation of the absolute value of the parametric function seems to be a little bit too cumbersome, but we can see by tweaking the parameters which values do bring to convergence with which function. For example, it seems that choosing f_1 for $\alpha < 5$ and f_2 for values of $\alpha \geq 5$ is a good criterion for reaching convergence over the entire domain of α.

We will hence (arbitrarily) choose to use the following rule

$$f = \begin{cases} \frac{\alpha}{\pi}\sqrt{1-R}\ I(R) & \alpha < 5 \\ 1 - (\frac{\pi R}{\alpha\ I(R)})^2 & \alpha \geq 5 \end{cases}$$

with $a = 0.02$ (found good value by trial and error).


Convergence of algorithm for mixed strategy ($R_0 = 0.5, a = 0.02$)
$\alpha = 0.1, f_1$
$\alpha = 1, f_1$
$\alpha = 1.5, f_1$
$\alpha = 5, f_2$
$\alpha = 10, f_2$
$\alpha = 20, f_2$
0.1
0.2
0.3
0.4
0.5
0.490
0.495
0.500
0.50
0.55
0.60
0.6
0.7
0.8
0.9
1.0
0
100
200
300
400
500

Step 6: Plot the obtained curve on top of the parametric plot and the numerical results