

# Capítulo 6

## Correspondencia de parámetros entre los modelos microscópico y macroscópico

En los Capítulos 3 y 5 se establece el marco teórico de dos modelos matemáticos que dan una posible explicación del mecanismo que rige la dinámica de población de las células T durante una infección aguda. Como se puede ver en las simulaciones correspondientes de estos modelos (ver Capítulo 4 y Sección 5.2) ambos pueden reproducir comportamientos similares, como son el de tolerancia e intolerancia al *patógeno*. Sin embargo, ambos modelos son notablemente distintos por dos razones principalmente:

1. El punto de vista desde el cual se aborda el algoritmo de decisión de las células T es distinto. Mientras que el modelo microscópico determina el algoritmo de comportamiento de cada célula de manera individual, el macroscópico presenta unas ecuaciones que gobiernan sobre toda la población de células.
2. Las ecuaciones diferenciales que conforman el modelo microscópico son de primer orden y su significado, desde el punto de vista biológico está bien definido. Esto es, los parámetros de este modelo, tales como el número de receptores de membrana de la célula ( $r_i$ ) o la tasa de cambio de estos receptores ( $\lambda_{xy}$ ) (ver Tabla 4.1), constituyen conceptos biológicos claros. Por su parte, el modelo macroscópico utiliza un sistema de ecuaciones de segundo grado, basado en las dinámicas *newtonianas* y en dos propiedades de la población: la elasticidad y la inercia. Los parámetros  $k$  y  $\lambda$  representan estas dos últimas propiedades en las ecuaciones, respectivamente. Sin embargo, desde el punto de vista biológico, el valor de estos parámetros tiene un significado difuso, pues no se conocen mecanismos que rijan estas dos propiedades.

A pesar de que el número de parámetros del modelo macroscópico es considerablemente menor, la elección de los parámetros  $k$  y  $\lambda$  es más compleja que la de los parámetros del modelo microscópico por la razón 2. Así las cosas, lo ideal sería poder establecer una correspondencia entre los parámetros de ambos modelos. De esta manera se podrían establecer los valores de los parámetros del modelo microscópico, que tienen un significado biológico claro, e inferir el valor de los parámetros del modelo macroscópico o viceversa. A lo largo de este capítulo se detalla cómo se ha abordado este problema mediante el uso de técnicas de inteligencia artificial (Sección 6.1) y se interpretan los resultados obtenidos (Sección 6.2).

## 6.1. Conjunto de datos y entrenamiento de la red neuronal

Como una primera aproximación a esta cuestión de correspondencia de parámetros, se propone la implementación de una red neuronal cuyo propósito es predecir el valor de los parámetros que se le deben asignar al modelo macroscópico teniendo como entrada aspectos característicos de una simulación. En otros términos, se podría decir que se busca hacer la función inversa del modelo. De esta manera, podemos hacer una simulación con unos parámetros concretos del modelo microscópico y obtener el valor de los parámetros del modelo macroscópico que se deberían usar para lograr un resultado similar.

Antes de poder implementar la red es necesario determinar con qué datos se va a trabajar. Más concretamente se deben establecer las entradas y las salidas que tendrá la red. En nuestro caso se toman las siguientes decisiones:

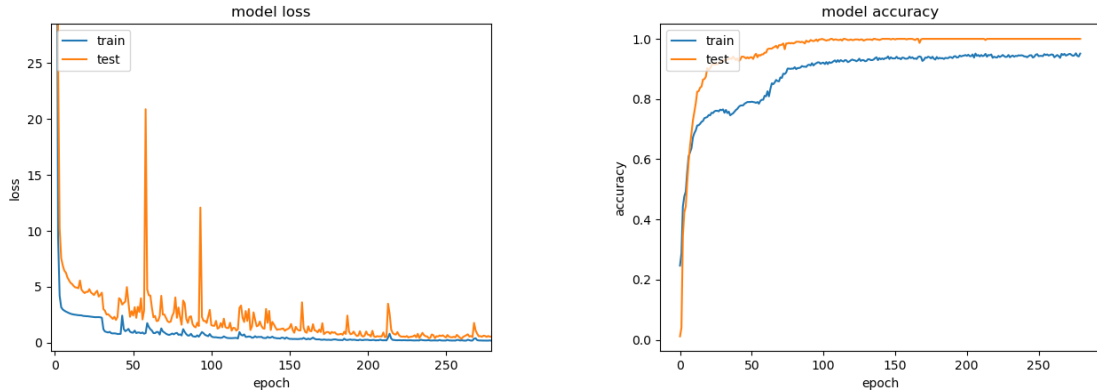
- Las simulaciones que se realizan para obtener los datos pertinentes se corresponden con situaciones de intolerancia al *patógeno*.
- La red neuronal consta de diez datos de entrada y cuatro de salida. Los seis primeros datos de entrada se corresponden con seis puntos de interés de cada simulación. Estos puntos son: el máximo número de células de patógeno alcanzado, el máximo número de células T alcanzado, el tiempo en el que se alcanzaron ambos y el tiempo en el que desaparecieron ambas poblaciones (en la Figura 6.2a pueden verse destacados los puntos mencionados), que denominaremos como  $max\_P$ ,  $max\_T$ ,  $t\_max\_P$ ,  $t\_max\_T$ ,  $t\_min\_P$ ,  $t\_min\_T$ , respectivamente. Los cuatro restantes datos de entrada son los parámetros  $\alpha$ ,  $\beta$ ,  $k$  y  $\lambda$  del modelo macroscópico con los cuales se han obtenido los seis valores anteriores. Por último, los cuatro parámetros de salida de la red se corresponden con los valores de los parámetros  $\alpha$ ,  $\beta$ ,  $k$  y  $\lambda$  predichos por la misma.
- El rango de valores para  $\alpha$ ,  $\beta$ ,  $k$  y  $\lambda$  se estableció con ayuda del modelo macroscópico adimensional (ver Figura 5.2), para ajustarnos lo más posible a una situación de intolerancia, y de tal manera que el número de simulaciones resultantes no fuera demasiado elevado, pero permitiendo suficiente variabilidad en los datos para abarcar el mayor número posible de situaciones. En concreto se establecieron los siguientes rangos:

- $\alpha \in [0, 75; 7]$
- $\beta \in [0, 1; 5]$
- $k, \lambda \in [0, 1; 2]$

Con estos rangos y a un paso<sup>1</sup> de 0,5 se obtienen unas 2080 simulaciones aproximadamente, de las cuales 1587 fueron casos de intolerancia. Los valores correspondientes a los puntos de interés de la simulación y sus parámetros se recogen en el archivo *data\_neural\_network\_csv* por filas y en el mismo orden que han sido mencionados ( $max\_P$ ,  $max\_T$ ,  $t\_max\_P$ ,  $t\_max\_T$ ,  $t\_min\_P$ ,  $t\_min\_T$ ,  $\alpha$ ,  $\beta$ ,  $k$  y  $\lambda$ ). Este documento da lugar al conjunto de datos de la red.

Como es habitual para el entrenamiento de una red neuronal, el 70 % del conjunto de los datos, tomado de forma aleatoria, se utilizó para el entrenamiento y el 30 % restante para *testear* la red. La implementación de la red está realizada en Python

<sup>1</sup>Con paso nos referimos al valor del incremento del parámetro en cada iteración.



(a) Valores de *loss* calculados para la red neuronal durante el entrenamiento.

(b) Valores de *accuracy* calculados para la red neuronal durante el entrenamiento.

Figura 6.1: Representación gráfica de los valores de *loss* y *accuracy* para cada *epoch* durante el entrenamiento de la red.

y el código correspondiente puede verse en el archivo *redNeuronal\_modeloMacro.py*. Esta cuenta con cinco capas densas y activaciones *ReLu*<sup>2</sup>. Esto es importante en la última capa, puesto que los parámetros no pueden tomar valores negativos.

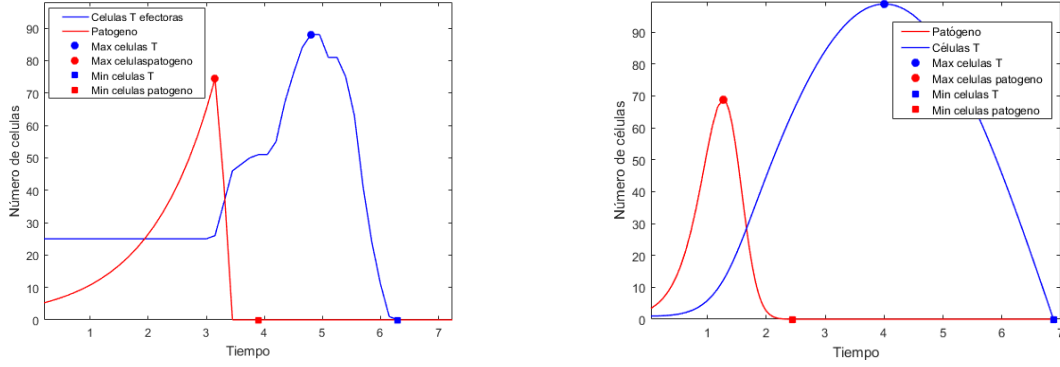
## 6.2. Resultados obtenidos por la red neuronal

En esta sección se exponen los resultados obtenidos tras el entrenamiento de la red, prestando atención a los valores de *loss* y *accuracy* alcanzados. Además, veremos un ejemplo real de la inferencia de parámetros dada por la red tras establecer como entrada una simulación del modelo microscópico.

Comencemos definiendo los conceptos de *epoch*, *loss* y *accuracy* para una red neuronal, de esta manera las gráficas y resultados que vienen a continuación no presentarán ningún impedimento terminológico: Se entiende por *epoch* cada pasada completa por todo el conjunto de datos de entrenamiento. Las redes neuronales, cuando entrenan, hacen varias pasadas por los datos y, en cada una de ellas, intentan minimizar una función de error. El concepto de *loss* está asociado a esto último, pues este es el valor que intentamos minimizar. Cuanto más pequeño es más precisas son las predicciones de la red. En nuestro caso, el valor de *loss* se corresponde con el error cuadrático medio. Por su parte, el valor de *accuracy* es una métrica utilizada para medir el rendimiento del algoritmo. Este valor se calcula una vez la red se ha entrenado y ha fijado todos sus parámetros. El valor de *accuracy* mide cómo de preciso es el modelo comparado con los datos reales. Por ejemplo, supongamos que tenemos 1000 muestras y nuestro modelo es capaz de clasificar bien 990 de ellas entonces, el valor de *accuracy* es del 99%.

En la Figura 6.1 podemos ver las gráficas correspondientes a los valores de *loss* y *accuracy* durante el entrenamiento de la red. Como se puede observar en la Figura 6.1a, el valor de *loss* consigue estabilizarse al mínimo en el conjunto de prueba una vez pasada la iteración 230 (ver Figura 6.1a). Por su parte, el valor de *accuracy* (ver Figura 6.1b) continúa incrementándose para el conjunto de entrenamiento hasta prácticamente la última

<sup>2</sup> $ReLu(x) = \max(0, x)$



(a) Simulación: caso de intolerancia al *patógeno* en el modelo microscópico. Parámetros y variables:  $t_{cycle} = 0,05$ ,  $t_{apo} = 0,1$ ,  $t_{next} = 0,15$ ,  $\alpha = 6,4$ ,  $\beta = 0,22$ ,  $\lambda_{pd} = 0,05$ ,  $\lambda_{Tp} = 6 * 10^{-5}$ ,  $\lambda_{pp} = 0,5 * 10^{-4}$ ,  $\mu_{pc} = 8$ ,  $\mu_{da} = 15$ .

(b) Simulación: caso de intolerancia al *patógeno* en el modelo macroscópico. Parámetros:  $\alpha = 3,5$ ,  $\beta = 0,29$ ,  $k = 0,3$ ,  $\lambda = 0,9$ ,  $P_m = 0$ .

Figura 6.2: Ejemplo con simulaciones del modelo microscópico y macroscópico con los valores de los parámetros predichos por la red neuronal. Casos de intolerancia al *patógeno*

iteración, lo que indica que el modelo no está sobreentrenando, a pesar de que en el conjunto de prueba se estabilice una vez pasada la iteración 100 aproximadamente. Estos resultados sugieren que el número de *epoch* utilizados para entrenar la red es el óptimo.

En el archivo *resultados.txt* se pueden ver algunos de los resultados obtenidos por la red, correspondientes a distintos valores de *accuracy*<sup>3</sup>. En el caso que nos ocupa ahora, detallaremos un ejemplo concreto obtenido a partir de los datos de una simulación del modelo microscópico, cumpliendo así con el propósito de esta red. En la Figura 6.2a podemos ver el resultado de la simulación del modelo microscópico, con los seis puntos de interés destacados. Concretamente el valor de esos parámetros es:  $max\_P = 74,4$ ,  $max\_T = 88$ ,  $t\_max\_P = 3,15$ ,  $t\_max\_T = 4,8$ ,  $t\_min\_P = 3,9$  y  $t\_min\_T = 6,3$ . Una vez la red estaba entrenada se introdujeron estos valores como entrada para obtener la predicción de los valores de los parámetros del modelo macroscópico. El resultado obtenido fue:  $\alpha = 3,5$ ,  $\beta = 0,29$ ,  $k = 0,3$  y  $\lambda = 0,9$ . En la Figura 6.2b puede verse la simulación del modelo macroscópico correspondiente a esos parámetros. Si comparamos ambas figuras observamos a simple vista que ambas presentan dos situaciones muy similares, si bien es cierto que los valores difieren ligeramente. En particular, la simulación del modelo macroscópico tiene en puntos de interés los siguientes valores:  $max\_P = 68,94$ ,  $max\_T = 98,82$ ,  $t\_max\_P = 1,27$ ,  $t\_max\_T = 4$ ,  $t\_min\_P = 2,45$  y  $t\_min\_T = 6,87$ . Si comparamos estos valores con los obtenidos con el modelo microscópico vemos el valor  $max\_P$  es menor en el modelo microscópico pero que los tiempos asociados a este ( $t\_max\_P$  y  $t\_min\_P$ ) también lo son. Lo que nos dice que, a pesar de que los valores no han sido exactos, la forma de la gráfica sí se preserva. Si prestamos atención a los valores referentes a las células T, vemos que el patrón ha cambiado, pues se alcanza un número mayor de células T en el modelo macroscópico y, sin embargo, este valor se alcanza antes que en el modelo microscópico. Esto nos indica que los parámetros de elasticidad e inercia no se han ajustado completamente, lo que hace que observemos ese pequeño desfase.

<sup>3</sup>El resultado de cada de las distintas simulaciones se representa por filas con el siguiente formato: [los seis puntos de interés generados tras una simulación del modelo macroscópico] => [el valor predicho por la red para los parámetros  $\alpha$ ,  $\beta$ ,  $k$  y  $\lambda$ ] (expected [el valor real de dichos parámetros]).