Estrategias de recomienzo para el entrenamiento de perceptrones multicapa

Manuel Cebrián & Iván Cantador

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

28049 Madrid

{manuel.cebrian, ivan.cantador}@uam.es

Resumen

El entrenamiento de un perceptrón multicapa puede considerarse un algoritmo Las Vegas, i.e., un algoritmo aleatorio que se detiene al alcanzar un error de entrenamiento requerido, y cuyo tiempo de ejecución es una variable aleatoria discreta. En este trabajo se modeliza como tal el entrenamiento de un perceptrón multicapa sobre un conjunto de patrones bien conocido, Tiroides, observándose que el tiempo necesario para terminar su aprendizaje no es monótonicamente decreciente con el tiempo ya transcurrido. Aplicando sobre él dos estrategias de recomienzos se buscará minimizar el tiempo esperado de aprendizaje. La primera hará uso del conocimiento completo de la distribución del tiempo de entrenamiento y conseguirá el mínimo global. La segunda, siendo universal, no hará uso de conocimiento a priori y proporcionará un tiempo esperado de entrenamiento que sólo es un factor logarítmico más lento que el de la estrategia óptima. Los resultados obtenidos muestran que el tiempo esperado de entrenamiento se ve drásticamente reducido con ambas estrategias.

1 Introducción

Considerando la finalización del proceso de construcción de un perceptrón multicapa (PMC) al alcanzarse un valor prefijado δ del error de entrenamiento, y sin tener en cuenta el tamaño de los conjuntos de datos empleados, cuatro son los factores que pueden determinar el tiempo de entrenamiento necesario, medido en épocas y denotado a partir de ahora como la variable aleatoria discreta T. El primero de ellos se corresponde con la arquitectura de la red neuronal

$$\mathcal{A}^{H}_{n_0,n_1,n_2,\ldots,n_H,n_{H+1}},$$

donde H es el número de capas ocultas, n_0 y n_{H+1} son respectivamente los números de unidades de las capas de entrada y de salida, y $n_1, n_2, \ldots, n_H$, los números de unidades en cada capa oculta. En general, cuanto mayor es el número de unidades, más rápidamente se minimiza el error sobre el conjunto de entrenamiento. El segundo factor está relacionado con el algoritmo de aprendizaje $\mathcal{L}$ que se emplee para actualizar los pesos de la red. Dependiendo de la estrategia utilizada la convergencia del error de entrenamiento se realizará de manera más o menos estable. El tercero es el error de entrenamiento requerido δ, que se calcula sobre el conjunto de patrones $\mathcal{P}$. Cuanto más bajo sea, más se tardará en alcanzar y mayor será la probabilidad de que no sea alcanzado, por poder llegar a mínimos locales superiores en valor al mismo. Estos tres factores se fijan por el diseñador a la hora de abordar un problema. No pasa lo mismo con el cuarto: la inicialización de los pesos del perceptrón. Estos pesos se inicializan en general de forma aleatoria y son por tanto un vector aleatorio $\overrightarrow{W}_0$ con

una distribución de probabilidad $\overrightarrow{\mathcal{W}}$ determinada por el diseñador, situación que tiene un gran impacto en el entrenamiento [1, 2, 6].

Empíricamente es fácil comprobar (véase Sección 3) la existencia de una gran varianza en el número de épocas necesario para que el entrenamiento tenga éxito, i.e., para que el error de entrenamiento alcance el valor δ fijado. El motivo radica en el hecho de que ciertos $\overrightarrow{W}_0$ pueden provocar la convergencia hacia óptimos locales, tardándose mucho tiempo (e incluso indefinidamente) en llegar a la precisión requerida. Para afrontar este problema diversos algoritmos de inicialización de pesos han sido diseñados (e.g. [2, 11]). Todos ellos establecen condiciones para la distribución de probabilidad $\overrightarrow{\mathcal{W}}$ basadas en información del conjunto de datos, de forma que se minimice el tiempo medio de entrenamiento.

Nuestro enfoque, aunque con el mismo objetivo, es radicalmente diferente y sigue el marco teórico expuesto en [3, 7, 8]. Consideramos el aprendizaje del PMC como un algoritmo Las Vegas, i.e una algoritmo aleatorio (por la inicialización aleatoria de sus pesos $\overrightarrow{W}_0$) que se detiene al alcanzar un error de entrenamiento δ prefijado, y cuyo tiempo de entrenamiento es una variable aleatoria discreta (T). Estos algoritmos trabajan sobre dos supuestos:

i. La única observación posible es la duración de la ejecución;

ii. El sistema tiene conocimiento *absoluto* o *nulo* sobre la distribución del tiempo de ejecución.

Con estas premisas, atacamos el problema de minimizar el tiempo de entrenamiento esperado. Para ello, usamos la siguiente estrategia: entrenamos al PMC durante un número de épocas fijo t_1. Si se alcanza la precisión requerida δ antes de t_1, hemos acabado. De lo contrario, inicializamos los pesos del perceptrón con otro nuevo $\overrightarrow{W}_0$ y ejecutamos el entrenamiento durante t_2 épocas. Así continuamos indefinidamente hasta que el entrenamiento alcanza el error δ. Este esquema se ve representado en la Figura 1.

procedimiento RECOMIENZOS($\mathcal{A}, \mathcal{P}, \mathcal{L}, \delta, \mathcal{S}$)
  $epocas \leftarrow 0$
  $i \leftarrow 1$
  do
    $s \leftarrow$ SEMILLA_ALEATORIA()
    $W_0^{(i)} \leftarrow$ GENERAR_PESOS(s)
    $PMC \leftarrow$ INICIALIZAR_PMC($\mathcal{A}, \mathcal{L}, W_0^{(i)}$)
    $t_i \leftarrow \mathcal{S}[i]$
    $\delta' \leftarrow$ ENTRENAR_PMC($\mathcal{P}$, t_i)
    $epocas \leftarrow epocas + t_i$
    $i \leftarrow i + 1$
  while $\delta' > \delta$
  return *epocas*
end

Figura 1: Esquema general de entrenamiento del PMC usando la estrategia de recomienzos $\mathcal{S}$.

Estas estrategias, denominadas *recomienzos*, vienen determinadas por una secuencia infinita de tiempos de reinicio $(t_1, t_2, t_3, \ldots)$, tras los cuales el algoritmo se detiene, establece aleatoriamente sus pesos, y vuelve a ejecutarse. Existen resultados formales (e.g. [7, 8]) que cuantifican el aumento de eficiencia que puede obtenerse usando varias de estas estrategias. Sin embargo, en este trabajo se estudiará un ejemplo de clasificación concreto, Tiroides [9], sobre el que se aplicarán dos estrategias de recomienzos y se buscará minimizar el tiempo esperado de entrenamiento.

El artículo está organizado de la siguiente manera. Primeramente, en la Sección 2, se establecen las características que la distribución de T debe cumplir para poder beneficiarse de las estrategias de recomienzos. A continuación, en la Sección 3, se presenta el problema de clasificación Tiroides, sobre el que se justificarán y aplicarán las citadas técnicas. En la Sección 4, se supone conocida la distribución de la variable T y se aplica una estrategia que proporciona el mínimo global del valor esperado de T. Invirtiendo el escenario, en la Sección 5 la distribución de T es desconocida y se aplica entonces una estrategia *universal* de recomienzos para obtener un tiempo esperado que sólo es un factor logarítmico más lento. Finalmente, en la Sección 6 se discuten los resultados obtenidos y las futuras líneas de trabajo.

2 Distribuciones de probabilidad aptas para estrategias de recomienzos

Siguiendo el esquema de un algoritmo Las Vegas, una ejecución se debe reiniciar cuando el tiempo de finalización condicionado al tiempo transcurrido sea menor que el tiempo esperado de entrenamiento. Esto puede formalizarse haciendo que T_{τ^K} sea el tiempo de finalización tras K recomienzos en los tiempos $\tau, 2\tau, \ldots, K\tau$, y $T_{\tau^0} = T$ sea el caso sin recomienzos. El siguiente teorema (demostrado en [8]) da la condición necesaria y suficiente para que una variable aleatoria T pueda beneficiarse del uso de recomienzos.

Teorema 1. El tiempo esperado de finalización bajo cero ($E[T]$), $K \geq 1$ ($E[T_{\tau^K}]$) y un número ilimitado de recomienzos ($E[T_\tau]$) se relacionan de la siguiente manera:

$$\begin{aligned} E[T_\tau] &< \ldots < E[T_{\tau^{K+1}}] < E[T_{\tau^K}] < \ldots \\ &< \ldots < E[T] \leftrightarrow E[T_\tau] < E[T], \quad (1) \end{aligned}$$

y

$$E[T] < E[T - \tau | T > \tau] \leftrightarrow E[T_\tau] < E[T]. \quad (2)$$

De la Ecuación (2) podemos extraer que la condición suficiente para usar estrategias de recomienzos es que $E[T] < E[T - \tau | T > \tau]$ para algún $\tau \in \mathbb{Z}^+$. Una vez se haya identificado un τ que cumpla la Ecuación (2), sabemos gracias a la Ecuación (1) que la mejor estrategia consiste en usar un número ilimitado de recomienzos. Es importante reseñar que el Teorema 1 es válido también cuando T es una variable aleatoria no discreta y $\tau \in \mathbb{R}^+$.

La pregunta que debe plantearse entonces es: ¿qué tipo de distribuciones cumplen la Ecuación (2) para al menos un valor? Las distribuciones típicas que la satisfacen son aquellas que asintóticamente tienen colas de tipo Pareto-Lévy, *viz.*

$$P[T > t] \sim Ct^{-\alpha}, \quad (3)$$

donde α es una constante en el intervalo (0,2), y t es mayor que cero. Estas distribuciones se denominan comúnmente distribuciones de *colas pesadas* y son las que más se benefician de estrategias de recomienzos [3, 5]; aunque no son las únicas, pues se han encontrado distribuciones con colas exponencialmente decrecientes que también han incrementado su eficiencia gracias a esta técnica [5].

3 Un ejemplo práctico: Tiroides

Para poder motivar el uso de estrategias de recomienzo en la construcción de un PMC, es necesario comprobar en primer lugar la gran variabilidad existente en el tiempo de entrenamiento medio empleado por estos algoritmos.

Como ya se introdujo anteriormente, el estudio se ha realizado sobre el problema Tiroides, conjunto de datos bien conocido en la literatura y obtenido del repositorio de la UCI [9]. Su elección, debida a los bajos errores de clasificación que sobre él se consiguen mediante PMCs, permite mostrar que a pesar de ser un problema relativamente sencillo de resolver por un perceptrón (obteniendo errores de clasificación inferiores al 1 %), provoca una alta variabilidad en el número de épocas de entrenamiento necesario para alcanzar un porcentaje dado de aciertos en clasificación, independientemente de la arquitectura seleccionada. La Tabla 1 muestra este hecho. Los resultados expuestos se obtuvieron con un PMC con una única capa oculta, entrenado mediante descenso por gradiente estocástico durante 10000 épocas, y promediados en 10 iteraciones de validación cruzada de 10 particiones. La primera columna es el número de unidades ocultas, la segunda y tercera se corresponden con la media y la desviación típica del número de épocas necesarias para alcanzar un porcentaje de clasificación correcta $\delta = 98\,\%$ (sobre el conjunto de entrenamiento). Llaman la atención las desviaciones típicas del número de épocas de entrenamiento que se efectuaron en el experimento. Mientras que para $n_1 = 6$ los resultados son "discretos", pues esta arquitectura empleó de media 587.6 épocas con una desviación de 55.1, valores como $n_1 = 2, 3, 4, 5, 9$ obtuvieron desviaciones mucho más grandes en relación a los valores medios alcanzados.

De estos últimos se ha elegido $n_1 = 3$ como arquitectura fija para el resto de pruebas

n_1	$E[T]$	$\sigma[T]$
0	9828.8	116.9
1	8551.7	2547.5
2	5516.8	3885.6
3	1872.2	2436.9
4	2339.7	2848.8
5	1680.2	1355.6
6	587.6	55.1
7	482.4	296.9
8	490.5	464.1
9	1637.4	2554.8

Tabla 1: Media y desviación típicas del número de épocas necesarias para alcanzar un porcentaje de clasificación correcta $\delta = 98\,\%$, para PMCs con distinto número de neuronas n_1 en la (única) capa oculta.

y razonamientos teóricos asociados a las mismas. En primer lugar porque, aunque tenga un número de unidades pequeño, alcanza un alto poder computacional: su porcentaje de aciertos llegaba en los experimentos a más del $99\,\%$. En segundo lugar, porque el número de épocas medio es también relativamente bajo: 1872.2, y en tercer lugar, por la desviación del número medio de épocas, que es incluso mayor que la propia media, hecho muy favorable para la aplicación de estrategias de recomienzos.

En la sección anterior enunciamos la condición suficiente para el uso de recomienzos en un algoritmo Las Vegas. La Figura 2 muestra el tiempo medio de finalización condicionado a un tiempo de entrenamiento transcurrido τ. Podemos ver claramente que existen muchos valores de τ (la gran mayoría) que verifican la Ecuación (2). Es posible comprobar empíricamente que T es una distribución de colas pesadas. Para ello estimamos el exponente de decrecimiento de la Ecuación (3) usando el estimador de Hill [4]:

$$\hat{\alpha}_r = \left(r^{-1} \sum_{j=1}^{r} \ln X_{n,n-j+1} - \ln X_{n,n-r} \right),$$

donde n es el tamaño muestral, $X_{n,1} \leq X_{n,2} \leq \ldots \leq X_{n,n}$ es la muestra ordenada y $r < n$ es un valor de truncamiento que nos permite observar sólo los valores de la cola. Un valor muy

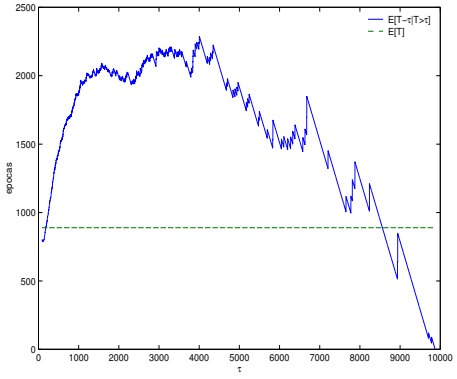

Figura 2: $E[T - \tau | T > \tau]$ como función de τ.

usado en la práctica es $r = \frac{1}{10}n$. Los resultados proporcionan una estimación $\hat{\alpha}_r = 1.942$, consistente con nuestra hipótesis de que la cola derecha de T tiene decaimiento polinómico y reune por tanto las condiciones idóneas para mejorar su rendimiento usando recomienzos.

4 Estrategias cuando la distribución es conocida

Recordemos que se pretende encontrar una estrategia $\mathcal{S}$ que minimice el tiempo medio de entrenamiento[1] $E[\mathcal{S}]$, i.e. el tiempo medio en alcanzar el error de entrenamiento δ usando la estrategia de recomienzos $\mathcal{S} = (t_1, t_2, t_3, \ldots)$, con $t_i \in \mathbb{Z}^+ \cup \{\infty\}$. Definimos p como la distribución de la variable aleatoria T, de forma que $p(t)$ denote la probabilidad de finalizar el entrenamiento la época t, con $t \in \mathbb{Z}^+ \cup \{\infty\}$. Suponiendo el completo conocimiento de la función de distribución $p(t)$, el siguiente teorema proporciona la estrategia óptima (se puede encontrar la demostración de este resultado en [7]).

Teorema 2. Sea S_t la estrategia definida como $\mathcal{S}_t = (t, t, \ldots)$, y l_p el valor definido como

$$l_p \equiv \inf_{t<\infty} E[\mathcal{S}_t]$$

[1] En un abuso de notación usaremos $E[\mathcal{S}] \equiv E[T_{\mathcal{S}}]$ a lo largo de todo el artículo.

La estrategia $\mathcal{S}_{t^*} = (t^*, t^*, \ldots)$ es la estrategia óptima para p, siendo t^* un valor que cumple

$$t^* = \begin{cases} t & \text{tal que } E[S_t] = l_p \\ \infty & \text{si } \nexists\ l_p. \end{cases}$$

Este resultado reduce el problema de encontrar la estrategia óptima a calcular el valor de t que minimiza la ecuación

$$E[S_t] = \frac{1}{q(t)} \left(t - \sum_{t'<t} q(t') \right)$$

donde $q(t) = \sum_{t' \leq t} p(t')$ es la función de distribución acumulada de p. Es posible encontrarse con funciones de distribución para las cuales no exista un mínimo finito (e.g. [7]); en ese caso, y tal como indica el Teorema 2, la estrategia de recomienzos será $(\infty, \infty, \ldots)$, i.e. no usar ninguna estrategia de recomienzos.

En general, existen dos limitaciones prácticas para esta optimización. En primer lugar, no se suele conocer la función de distribución $q(t)$ del tiempo de finalización del entrenamiento, aunque siempre se puede estimar a través de su función de distribución empírica $\hat{q}(t)$. En segundo lugar, suele existir un número de épocas mínimo t_{min} por debajo del cual $\hat{p}(t) = 0$, y un valor máximo t_{max} de esfuerzo computacional en el cual se detiene la ejecución aunque no se haya alcanzado δ, por lo que $\hat{q}(t_{max}) = 1$ (en nuestro ejemplo $t_{min} = 85$ y $t_{max} = 10000$). Por ello, el problema práctico consiste en muestrear lo máximo posible la variable T, y encontrar el t que minimice la expresión

$$\frac{1}{\hat{q}(t)} \left(t - \sum_{t'<t} \hat{q}(t') \right)$$

con $t \in \{t_{min}, t_{min}+1, \ldots, t_{max}-1, t_{max}\}$.

La Figura 3 muestra el tiempo esperado de entrenamiento para varios valores t, i.e. para varias estrategias S_t. Los tiempos esperados son estimaciones calculadas usando la función de distribución empírica $\hat{q}(t)$ obtenida a partir de las 1000 muestras de tiempos de entrenamiento del problema descrito en la sección anterior. Observamos que $t^* = 418$ y

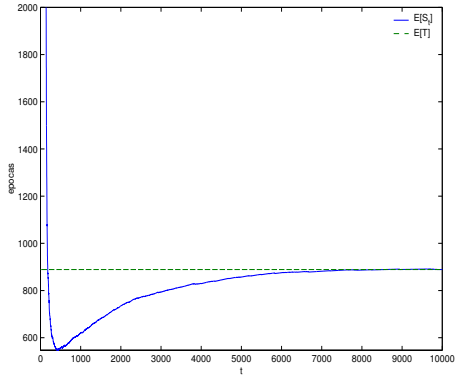

Figura 3: Tiempo esperado de entrenamiento para varios valores t de recomienzo usando la estrategia $\mathcal{S}_t$.

$l_p = 546.876$, siendo el tiempo medio sin recomienzos $E[T] = 888.850$. El uso de la estrategia $\mathcal{S}_{t^*}$ aumenta la eficiencia del entrenamiento en un 40 %, siempre en términos esperados.

Puede darse la situación en la que no se conozca $q(t)$, pero en la que se tenga algún conocimiento del valor de $q(t^*)$. En [7] se da una estrategia $\widetilde{\mathcal{S}}_t = (t_1, t_2, t_3, \ldots)$ con t_i de la forma:

$$t_i = 2^k, \quad k \left\lfloor \frac{1}{q(t)} \right\rfloor < i \leq (k+1) \left\lfloor \frac{1}{q(t)} \right\rfloor,$$

consiguiendo un rendimiento sólo un factor constante por encima de la estrategia óptima,

$$E[\widetilde{\mathcal{S}}_{t^*}] = O(E[\mathcal{S}_{t^*}]). \tag{4}$$

En la Figura 4 se aplica esta estrategia a varios valores de $q(t)$, encontrándose el óptimo en $q(t) = 0.5740$, que es justamente $q(t^*)$. El tiempo esperado que se consigue es 656.970, que supone un 26 % de mejora sobre $E[T]$. Tal y como dice la Ecuación (4), la eficiencia de esta estrategia está ligeramente poco por debajo (14 %) de la de la óptima.

5 Estrategias cuando la distribución es desconocida

A pesar de que la existencia de una estrategia óptima es un resultado muy interesante, puede resultar de poco valor en algunos escenarios

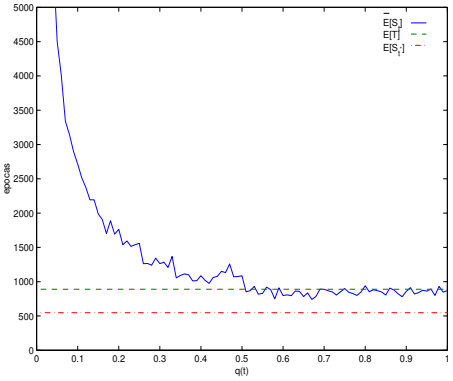

Figura 4: Tiempo esperado de entrenamiento para varios valores $q(t)$ usando la estrategia $\widetilde{\mathcal{S}}_t$. Para cada $q(t)$ se estimó $E[\widetilde{\mathcal{S}}_{t^*}]$ mediante 100 simulaciones de tipo Montecarlo de la variable aleatoria $\widetilde{\mathcal{S}}_t$ (ver Figura 5).

de uso de PMCs. Inicialmente, la primera vez que entrenamos un PMC, disponemos de muy poca (o ninguna) información a priori sobre la forma de su función de distribución, y ésta, en general, varía mucho respecto al problema sobre el que se aplique. Es más, en algunas ocasiones puede que necesitemos entrenar el PMC una sola vez, por lo que no sería útil entrenar muchas veces para obtener información sobre su distribución. Por tanto, lo que nosotros desearíamos en este escenario es disponer de una estrategia *universal*, que sea eficiente para todas las distribuciones de tiempos de entrenamientos posibles.

Sorprendentemente, dicha estrategia existe, y su coste esperado asintótico esta bastante cerca del de la estrategia óptima. Formalmente la estrategia se define como $\mathcal{S}^{\text{univ}} = (t_1, t_2, t_3, \ldots)$, donde

$$t_i = \begin{cases} 2^{k-1}, & \text{si } i = 2^k - 1 \\ t_{i-2^{k-1}+1}, & \text{si } 2^{k-1} \leq i < 2^k - 1. \end{cases}$$

Luby et al. presentan esta estrategia en [7] y enuncian los siguientes dos teoremas que juntos demuestran su optimalidad asintótica cuando la distribución es desconocida:

procedimiento SIMULAR$_{\widetilde{\mathcal{S}}_t}(muestra, q(t))$
  $i \leftarrow 1$
  $t_i \leftarrow 2$
  $epocas \leftarrow 0$
  do
    $T \leftarrow$ MONTECARLO$(muestra)$
    if $T \leq t_i$
      $epocas \leftarrow epocas + T$
      break
    end if
    $epocas \leftarrow epocas + t_i$
    $i \leftarrow i + 1$
    if i **mod** $\lfloor 1/q(t) \rfloor = 0$
      $t_i \leftarrow 2t_i$
    end if
  while $T > t_i$
  return $epocas$
end

Figura 5: Procedimiento para simular la variable aleatoria $\widetilde{\mathcal{S}}_t$.

Teorema 3. Para todas las distribuciones p,

$$E[\mathcal{S}^{\text{univ}}] \leq 192 E[\mathcal{S}_{t^*}](\log_2 E[\mathcal{S}_{t^*}] + 5). \quad (5)$$

Teorema 4. Para cualquier estrategia $\mathcal{S}$,

$$\sup_{p:l_p=l} E[\mathcal{S}] \geq \frac{1}{8} E[\mathcal{S}_{t^*}] \log_2(E[\mathcal{S}_{t^*}]).$$

Por tanto, es posible encontrar un p para cualquier estrategia que haga que su tiempo esperado sea, al menos, $O(E[\mathcal{S}_{t^*}] \log_2 E[\mathcal{S}_{t^*}])$, mientras que ese es el caso peor para la estrategia S^{univ}. En términos esperados, S^{univ} es lo mejor que se puede conseguir sin conocimiento a priori sobre la distribución p.

Sin embargo, a pesar de todas estas bondades teóricas, los resultados con esta estrategia sobre el problema Tiroides distan mucho de ser óptimos. Esto se debe a que los Teoremas 3 y 4 son resultados asintóticos, e incluyen factores logarítmicos y constantes que de deben ser tenidos en cuenta en aplicaciones prácticas. En nuestro ejemplo concreto, el factor logarítmico y los factores constantes de (5) son demasiado grandes (tres órdenes de magnitud en total). Tras realizar 1000 simulaciones de tipo Montecarlo de la variable $\mathcal{S}^{\text{univ}}$ (ver Figura 6), obtenemos una estimación de $E[\mathcal{S}^{\text{univ}}] = 4,141$,

```
procedimiento SIMULAR_{S^univ}(muestra)
  i ← 1
  t_i ← 1
  epocas ← 0
  do
    T ← MONTECARLO(muestra)
    if T ≤ t_i
      epocas ← epocas + T
      break
    end if
    epocas ← epocas + t_i
    i ← i + 1
    if log_2(i + 1) ∈ Z^+
      t_i ← 2^{log_2(i+1)−1}
    else
      t_i ← t_{i−2^{⌊log_2(i+1)⌋}+1}
    end if
  while T > t_i
  return epocas
end
```

Figura 6: Procedimiento para simular la variable aleatoria $\mathcal{S}^{\text{univ}}$.

casi 5 veces más lento que $E[T]$, i.e., sin usar ninguna estrategia de recomienzos. Este comportamiento "defectuoso" de la estrategia universal también ha sido observado en [3, 5].

Si observamos la estrategia universal con detenimiento,

$$S^{univ} = (1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, \ldots),$$

podemos deducir la razón que su eficiencia esté por debajo de lo esperado. La estrategia supone que existe una probabilidad netamente positiva de que el entrenamiento finalice en 1 época, cuando en los experimentos realizados con Tiroides ninguna de las ejecuciones toma menos de 85 épocas. Esto provoca que las ejecuciones de longitud $1, 2, 4, 8, 16, 32$ y 64 tengan una probabilidad bajísima de éxito. Por tanto, necesitamos que el tamaño de nuestros recomienzos crezca más rápido que los de la estrategia de Luby.

Para resolver este problema, e inspirándose en la estrategia de Luby, Walsh diseñó una nueva estrategia de recomienzos que se aplicó con mucho éxito a problemas de satisfacción de restricciones [10]. En ella, cada nuevo recomienzo es un factor constante γ más grande

```
procedimiento SIMULAR_{S^Walsh}(muestra, γ)
  i ← 1
  t_i ← 1
  epocas ← 0
  do
    T ← MONTECARLO(muestra)
    if T ≤ t_i
      epocas ← epocas + T
      break
    end if
    epocas ← epocas + t_i
    i ← i + 1
    t_i ← γ^i
  while T > t_i
  return epocas
end
```

Figura 7: Procedimiento para simular la variable aleatoria $\mathcal{S}^{\text{Walsh}}$.

que el anterior

$$\mathcal{S}^{\text{Walsh}} = (1, \gamma, \gamma^2, \gamma^3, \ldots), \quad \gamma > 1.$$

La estrategia se beneficia de que la probabilidad de éxito es alta cuando el valor de recomienzo t_i es cercano al óptimo t^*. Aumentando el valor de recomienzo geométricamente nos aseguramos de estar cerca del óptimo en pocas iteraciones. A partir de ese punto, la estrategia espera encontrar la solución en pocos recomienzos antes de que el valor de t_i se aleje demasiado del óptimo.

La Figura 8 muestra los valores esperados de S^{Walsh} usando varios valores estándar de γ. Podemos ver que en todos ellos se acelera el entrenamiento, con mejoras que están entre el $23\,\%$ ($\gamma = 8$) y el $9\,\%$ ($\gamma = 2$).

6 Conclusiones y trabajo futuro

En este trabajo hemos modelizado el entrenamiento un PMC como un algoritmo Las Vegas. Sobre este enfoque hemos mostrado en un problema concreto, Tiroides, que el tiempo necesario para entrenar un PMC puede no seguir una distribución de probabilidad estándar. Esta distribución está caracterizada por tener ejecuciones extremadamente largas con una frecuencia más alta de lo esperado. A continuación hemos introducidos las estrategias

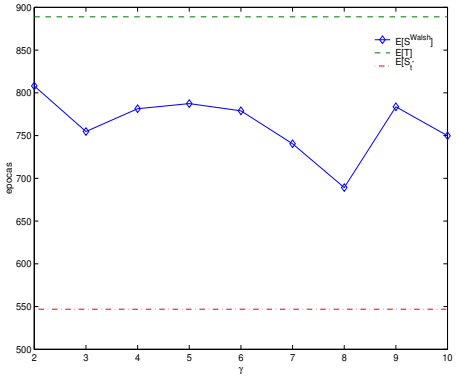

Figura 8: Tiempo esperado de entrenamiento usando la estrategia $\mathcal{S}^{\text{Walsh}}$ para varios γ. Para cada γ se estimó $E[\mathcal{S}^{\text{Walsh}}]$ mediante 1000 simulaciones de tipo Montecarlo (ver Figura 7) de la variable aleatoria $\mathcal{S}^{\text{univ}}$.

de recomienzos, y hemos comprobado empíricamente que el problema Tiroides tiene las características idóneas para aumentar su eficiencia con estas estrategias. Cuatro estrategias distintas se han aplicado sobre el problema: las dos primeras ($\mathcal{S}_{t^*}$ y $\widetilde{\mathcal{S}}_{t^*}$) suponen conocimiento absoluto de la distribución del tiempo de ejecución, consiguiendo una disminución del tiempo esperado de entrenamiento del 40 %. Las dos segundas ($\mathcal{S}^{\text{univ}}$ y $\mathcal{S}^{\text{Walsh}}$) suponen conocimiento nulo de la distribución, y consiguen mejoras que están entre el 9 % y el 23 %, también en términos esperados.

Es posible aumentar la eficiencia de las estrategias de recomienzos si se relajan los supuestos sobre los que se fundamentan los algoritmos Las Vegas; recordemos: (i) la única observación posible es la longitud de la ejecución; (ii) el sistema tiene conocimiento absoluto o nulo de la distribución del tiempo de entrenamiento. En [5] se formulan estrategias de recomienzos que consideran observaciones en tiempo real de la actividad de los algoritmos, en vez de asumir escenarios de completo o nulo conocimiento sobre su función de distribución. Sus métodos explotan dinámicamente la actualización de los conocimientos sobre la función de distribución del tiempo de ejecución. Una futura linea de trabajo es estudiar la eficiencia de estrategias de recomienzos para PMCs que tengan información *época a época* de indicadores de la actividad del PMC tales como los errores cuadráticos medios de entrenamiento y validación.

Referencias

[1] W. H. Delashmit, M. T. Manry, *Enhanced Robustness of Multilayer Perception Training*, Systems and Computers, 2002.

[2] W. Duch, R. Adamczak, N. Jankowski, *Initialization and optimization of multilayered perceptrons*, Third Conference on Neural Networks and Their Applications, 1997.

[3] C. Gomes, *Complete Randomized Backtrack Search*, Kluwer, 2003.

[4] B.M. Hill, *A simple general approach to inference about the tail of a distribution*, Annals of Statistics, 1975.

[5] H.A. Kautz, E. Horvitz, Y. Ruan, C.P. Gomes, B. Selman, *Dynamic Restart Policies*, AAAI/IAAI, 2002.

[6] Y. LeCun, L. Bottou, G. Orr, and K. Muller, *Efficient BackProp*, Neural Networks: Tricks of the trade, 1998.

[7] M. Luby, A. Sinclair, D. Zuckerman, *Optimal Speedup of Las Vegas Algorithms*, Information Processing Letters, 1993.

[8] A. V. Moorsel, K. Wolter, *Analysis and Algorithms for Restart*, First International Conference on Quantitative Evaluation, 2004.

[9] P. Murphy, D. Aha, *UCI Repository of Machine Learning Databases*, Tech. Report, University of California, 1997.

[10] T. Walsh, *Search in a Small World*, IJCAI, 1999.

[11] N. Weymaere, J.P. Martens, *On the initialization and optimization of multilayer perceptrons*, IEEE Transactions on Neural Networks, 1994.