Universidad Simón Bolívar   
Departamento de Computación y Tecnología de la Información

CI5438 – Inteligencia Artificial II

Enero-Marzo 2017

Integrantes:

Leonardo Martínez Carné: 11-10567

Nicolás Mañán Carné: 06-39883

Joel Rivas Carné: 11-10866

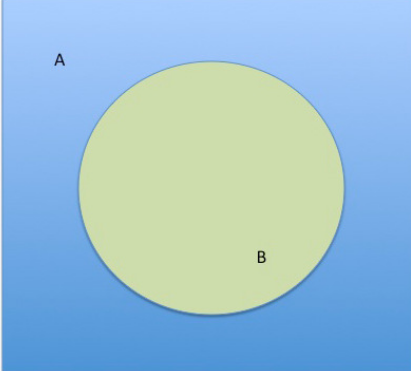
**Proyecto 2 – Redes Neuronales**

**Resumen**

En el siguiente proyecto se realizaran 3 ejercicios relacionados sobre redes neuronales, siendo este tema una base importante para el desarrollo de IA (Inteliencia Artificial).

Ejercicio 1: Se busca construir un algoritmo que implemente backpropagation en una red multicapa feedforward.

Ejercicio 2: Se realizara una serie de pruebas a la red neuronal construida anteriormente. El experimento trata sobre la clasificación de patrones, en donde se pretende lograr una clasificación de puntos en un plano perteneciente a dos regiones predeterminadas (A y B). Estas dos regiones corresponden a figuras geométricas, el área de A es el cuadrado menos el área que ocupa B, siendo el área de B limitada por una circunferencia centrada en (10,10) con radio 6, incrustada al interior del área de A. Como se muestra en el siguiente dibujo:

 Cuya ecuación canónica de esta circunferencia es:

**(x − 10) 2 + (y − 10) 2 = 36**

Se entiende por patrón un punto (x,y) dentro del rectángulo y el área al que este pertenece (A o B). Se suministraron tres conjuntos de entrenamiento de 500,100 y 2000 patrones ya clasificados, en donde se entrenara la red. Adicionalmente, se generara otro trío de conjuntos de datos de igual tamaño.

Se tomaron un conjunto de prueba de 10201 puntos de un barrido completo de la región cuadrada correctamente etiquetados. Se evaluaran, mostraran y analizaran las configuraciones en base a: error en entrenamiento, error en prueba, Falsos positivos y Falsos negativos.

Ejercicio 3: Se entrenara la red neuronal para construir dos clasificadores sobre los datos del conjunto *Iris Data Set* (<http://archive.ics.uci.edu/ml/datasets/Iris>). Uno que separe las “Iris Setosa” del resto (Clasificador Binario) y el otro que separa cada una de las clases. Estos conjuntos de datos se probaran con redes de 4 a 10 neuronas en la capa intermedia, usando como conjunto de entrenamiento los siguientes porcentajes de los datos para cada uno de ellos: 50%,60%,70%,80%,90%.

**Detalles de implementación/experimentación**

Ejercicio 1: Se uso una lista de lista, donde cada lista interna representa una capa de la red neuronal y cada capa contiene a su vez una lista de diccionarios que representan las neuronas.

Ejercicio 2: Antes que nada, se hizo un pre procesamiento de los datos, aplicando el método de normalización Min-Max. Posteriormente, se ejecutó la red neuronal con un alpha de 0.1 y 20000 iteraciones para la convergencia.

Por otra parte, se creo un script de python3 para generar los 10201 puntos para el conjunto de prueba. Igualmente, otro script para generar el trío adicinal de conjuntos de entrenamiento de 500,1000 y 2000.

Ejercicio 3: En principio, se aplico el método de normalización Min-Max y se transformaron los valores nominales a valores numéricos ( Iris Setosa = 0, Iris Versicolor = 1, Iris Virginica = 2 ). Luego, se implementó un script en python3 para crear los dos conjuntos: Uno que separé los “Iris Setosa” del resto (Clasificador binario) y uno que separe cada una de las 3 clases. Y por ultimo, se seleccionaron los conjuntos de datos entrenamiento de 50%,60%,70%,80% y 90% manualmente y de manera aleatoria.

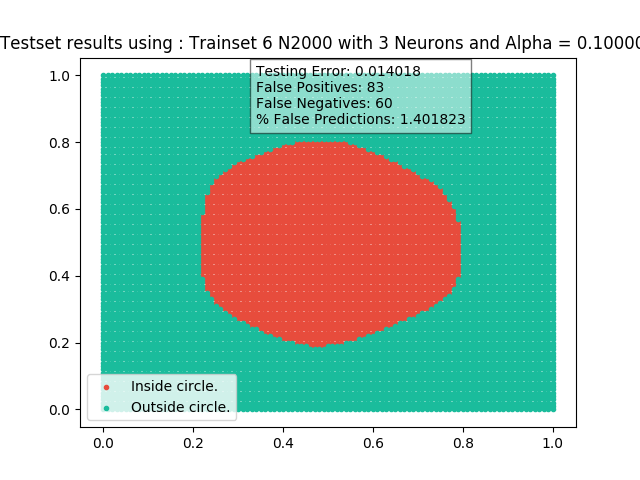
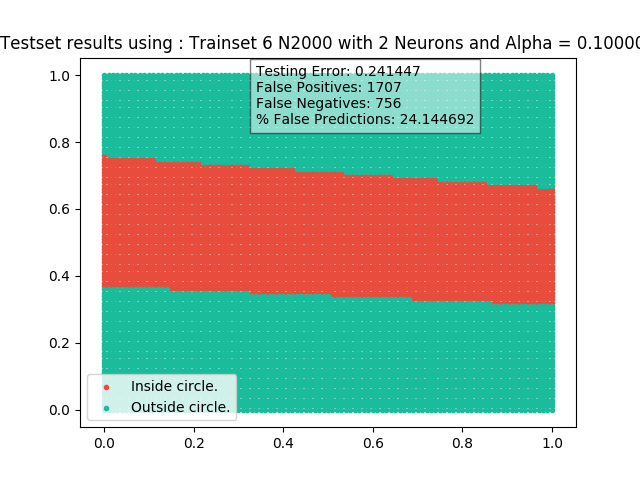
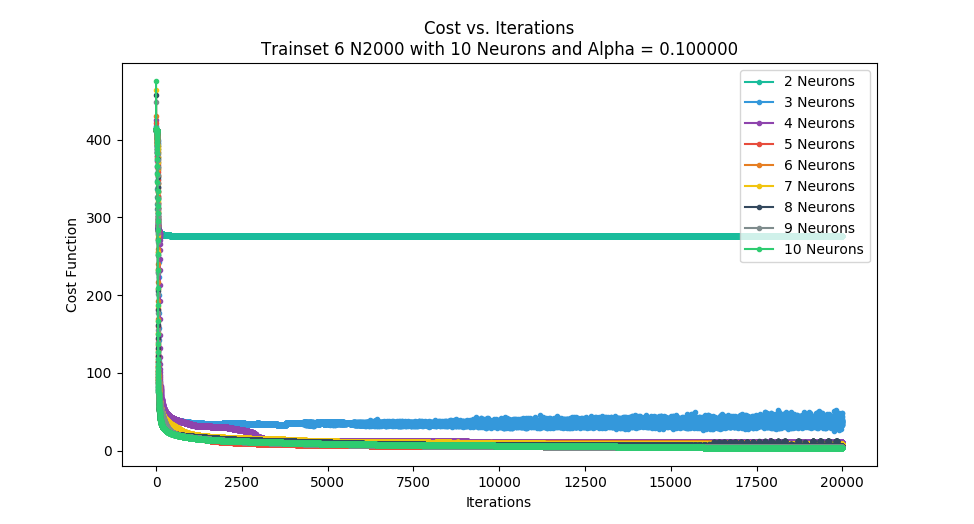
**Presentación y discusión de los resultados**

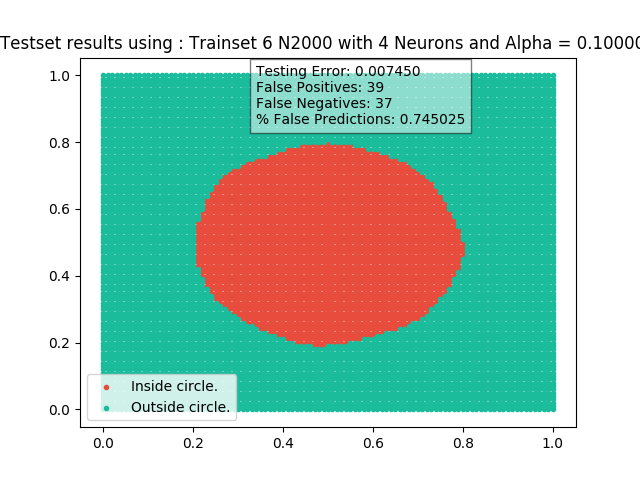
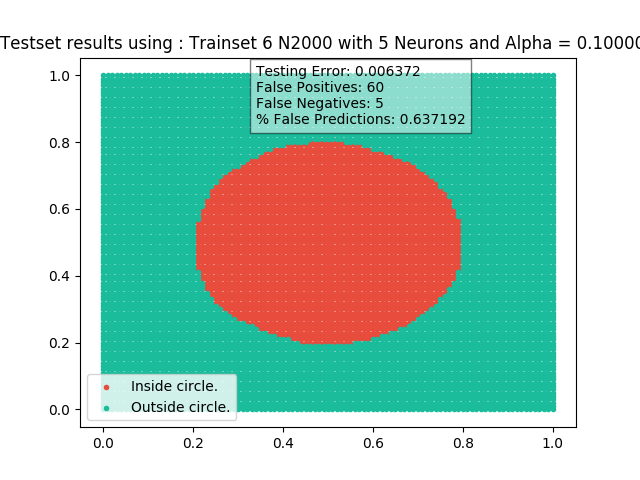
**Ejercicio 2:**

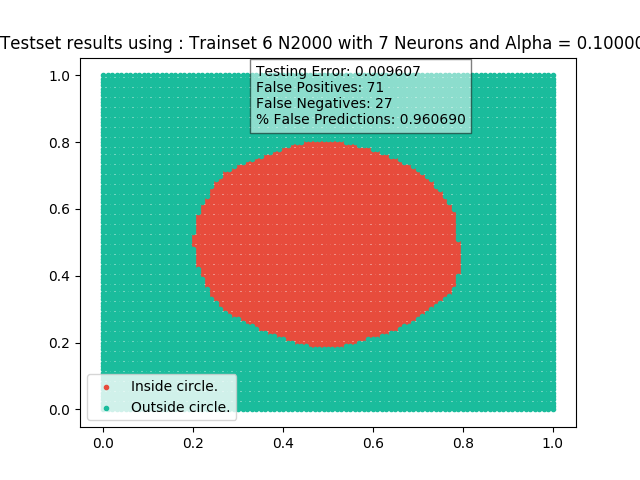
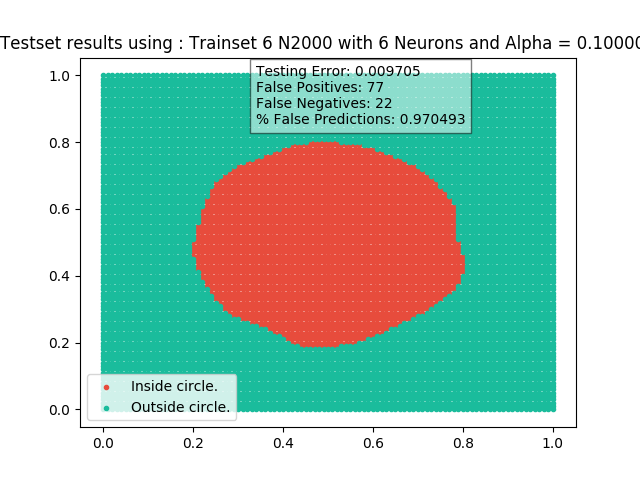
A continaución, se muestran un conjunto de resultados obtenidos al crear las diferentes redes neuronales con un número de neuronas entre 2 y 10.

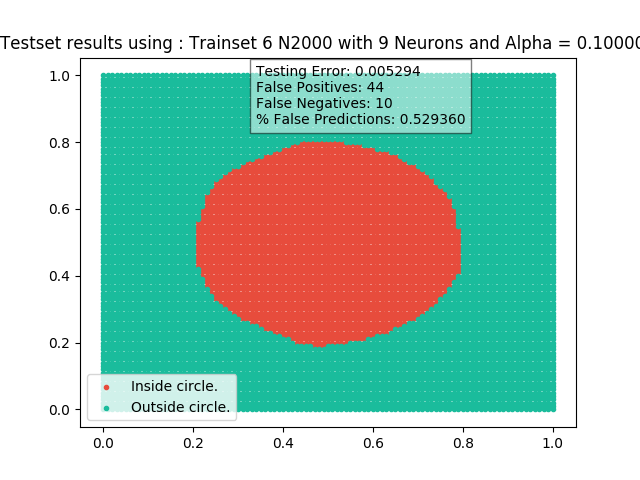
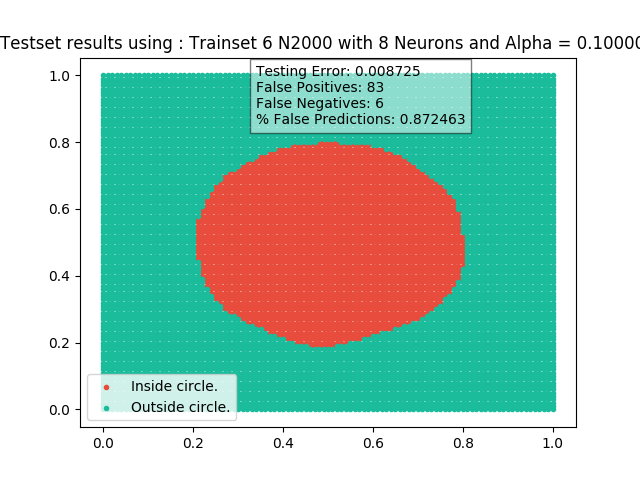
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Conjunto de Entrenamiento** | **Nº Neuronas** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **1 N = 500** | **Costo Mínimo** | 60.507 | 5.658 | 4.0868 | 1.8659 | 0.7903 | 0.4776 | 0.8760 | 0.7839 | 0.2295 |
| **Error de Prueba (MSE)** | 0.2433 | 0.0334 | 0.0281 | 0.0178 | 0.0147 | 0.0232 | 0.0208 | 0.0178 | 0.0170 |
| **Falsos Positivos** | 1657 | 296 | 209 | 129 | 70 | 137 | 136 | 125 | 87 |
| **Falsos Negativos** | 825 | 45 | 78 | 53 | 80 | 100 | 77 | 57 | 87 |
| **% Predicciones Erróneas** | 24.33 | 3.3428 | 2.8134 | 1.7841 | 1.4704 | 2.3233 | 2.0880 | 1.7841 | 1.7057 |
| **2**  **N = 1000** | **Costo Mínimo** | 136.399 | 12.104 | 7.9229 | 3.4122 | 3.294 | 2.951 | 1.1384 | 2.9655 | 1.3150 |
| **Error de Prueba (MSE)** | 0.2489 | 0.030 | 0.0255 | 0.010 | 0.010 | 0.009 | 0.0099 | 0.011 | 0.0111 |
| **Falsos Positivos** | 2129 | 280 | 241 | 96 | 87 | 84 | 88 | 98 | 111 |
| **Falsos Negativos** | 441 | 29 | 20 | 15 | 16 | 9 | 13 | 15 | 3 |
| **% Predicciones Erróneas** | 24.899 | 3.029 | 2.5585 | 1.088 | 1.009 | 0.9116 | 0.9900 | 1.107 | 1.1175 |
| **3**  **N = 2000** | **Costo Mínimo** | 275.195 | 24.250 | 10.771 | 10.8423 | 8.4337 | 8.45183 | 5.7612 | 6.9938 | 6.6888 |
| **Error de Prueba (MSE)** | 0.2394 | 0.020 | 0.0112 | 0.018 | 0.010 | 0.0104 | 0.0058 | 0.009 | 0.0072 |
| **Falsos Positivos** | 1628 | 1 | 21 | 5 | 12 | 11 | 12 | 1 | 37 |
| **Falsos Negativos** | 815 | 211 | 94 | 183 | 96 | 96 | 48 | 97 | 37 |
| **% Predicciones Erróneas** | 23.9486 | 1.0782 | 1.127 | 1.8429 | 1.058 | 1.0489 | 0.588 | 0.960 | 0.725 |
| **4 N = 500** | **Costo Mínimo** | 68.020 | 4.8609 | 1.1704 | 1.3035 | 0.6959 | 0.6197 | 0.6961 | 0.5305 | 0.3404 |
| **Error de Prueba (MSE)** | 0.2484 | 0.0242 | 0.0183 | 0.0168 | 0.012 | 0.015 | 0.0184 | 0.017 | 0.015 |
| **Falsos Positivos** | 1865 | 186 | 161 | 124 | 88 | 120 | 133 | 127 | 112 |
| **Falsos Negativos** | 669 | 61 | 26 | 48 | 38 | 42 | 55 | 47 | 44 |
| **% Predicciones Erróneas** | 24.8407 | 2.421 | 1.8331 | 1.1686 | 1.2351 | 1.588 | 1.8429 | 1.705 | 1.529 |
| **5**  **N = 1000** | **Costo Mínimo** | 137.4770 | 9.1461 | 4.514 | 4.379 | 2.918 | 2.981 | 2.929 | 2.709 | 2.224 |
| **Error de Prueba (MSE)** | 0.2421 | 0.0196 | 0.01480 | 0.0121 | 0.0134 | 0.01186 | 0.0160 | 0.0166 | 0.015 |
| **Falsos Positivos** | 1708 | 61 | 89 | 67 | 80 | 107 | 124 | 129 | 144 |
| **Falsos Negativos** | 762 | 139 | 62 | 57 | 57 | 14 | 40 | 41 | 41 |
| **% Predicciones Erróneas** | 24.2133 | 1.9605 | 1.4802 | 1.2155 | 1.3430 | 1.1861 | 1.6076 | 1.666 | 1.5194 |
| **6**  **N = 2000** | **Costo Mínimo** | 275.4866 | 25.76 | 10.54 | 4.476 | 8.666 | 7.901 | 8.866 | 4.1513 | 3.47566 |
| **Error de Prueba (MSE)** | 0.2414 | 0.0140 | 0.0074 | 0.0063 | 00009 | 0.0096 | 0.00857 | 0.00529 | 0.0076 |
| **Falsos Positivos** | 1707 | 83 | 39 | 60 | 77 | 71 | 83 | 44 | 60 |
| **Falsos Negativos** | 756 | 60 | 37 | 5 | 22 | 27 | 6 | 10 | 18 |
| **% Predicciones Erróneas** | 24.144 | 1.4018 | 0.7450 | 0.637 | 0.9704 | 0.9607 | 0.0872 | 0.5293 | 0.7646 |

A continuación se muestran la grafica de Iteraciones vs. Costo para el mejor conjunto de entrenamiento identificado, así como la progreción de las predicciones de los puntos del conjunto de pruebas para los diferentes números de neuronas. Este mejor conjunto fue tomado base a el menor error de prueba MSE y el %total de predicciones erróneas respecto al testset.



****

****

****

****

Puede observarse en las graficas otenidas para 6 neuronas en adelante, que la grafica generada de los puntos clasificados por la red neuronal varía muy poco, esto es porque el % de predicciones erróneas es realmente bajo (<0.9%).

**Ejercicio 3:**

**Parte 1.**

Primer conjunto de datos, usando 500 épocas y tasa de aprendizaje = 0.1. En todos los casos, se obtuvieron como Error de Prueba (MSE), Falsos Positivos, Falsos Negativos y % Predicciones Erróneas como cero (0). Por lo tanto, en las tablas de resultados que se muestran a continuación se obviarán estos parámetros.

**Conjunto de entrenamiento de 50%**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 0.031647 | 0.032323 | 0.026179 | 0.033217 | 0.026074 | 0.028804 | 0.023701 |
| **Costo Maximo** | 23.529952 | 32.139896 | 29.403197 | 23.848601 | 32.337585 | 35.362520 | 36.414706 |

**Conjunto de entrenamiento de 60%**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 0.035502 | 0.035601 | 0.030731 | 0.036882 | 0.030367 | 0.032104 | 0.028115 |
| **Costo Maximo** | 27.772692 | 37.980620 | 34.435932 | 28.472308 | 38.095737 | 42.209713 | 43.629493 |

**Conjunto de entrenamiento de 70%**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 0.029401 | 0.030713 | 0.024606 | 0.031029 | 0.024495 | 0.027130 | 0.022214 |
| **Costo Maximo** | 31.542141 | 43.570894 | 38.866699 | 32.302637 | 43.626981 | 49.022311 | 50.837242 |

**Conjunto de entrenamiento de 80%**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 0.031215 | 0.032218 | 0.027113 | 0.032654 | 0.026743 | 0.028823 | 0.024852 |
| **Costo Maximo** | 35.302040 | 48.631755 | 42.955022 | 36.292927 | 48.427737 | 55.691535 | 57.977840 |

**Conjunto de entrenamiento de 90%**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 0.031715 | 0.032685 | 0.027884 | 0.033032 | 0.027475 | 0.029469 | 0.025691 |
| **Costo Maximo** | 38.087858 | 53.055956 | 45.690604 | 38.93293 | 52.094009 | 62.411200 | 65.069206 |

Podemos observar que el costo mínimo se acerca bastante a cero, por lo tanto, el error en la predicción es bastante pequeño. Es decir, en la mayoría de los casos se espera tener una predicción correcta para la clasificación binaria. Esto se comprueba al haber obtenido 0 en los errores de prueba.

**Parte 2.**

Para la segunda parte del Ejercicio 3, en la clasificación de las 3 clases para el conjunto de pruebas, se obtienen los siguientes resultados, usando 500 épocas y tasa de aprendizaje = 0.1. Similar a los resultados de la parte 1, El error y los falsos positivos/negativos para las tres clasificaciones se vuelven 0 a partir del entrenamiento con 70% de los datos de prueba. Por lo tanto, se mostrarán estos atributos en las tablas correspondientes.

Conjunto de entrenamiento de 50%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 1.956026 | 1.724866 | 1.807889 | 1.738804 | 1.787320 | 1.815346 | 1.783438 |
| **Costo Maximo** | 33.639047 | 36.668418 | 35.487963 | 36.402884 | 40.110593 | 45.645755 | 48.452859 |
| **Error de Prueba (MSE)** | 0.026667 | 0.026667 | 0.026667 | 0.026667 | 0.026667 | 0.026667 | 0.026667 |
| **Falsos Positivos**  **Iris-Setosa** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Negativos**  **Iris-Setosa** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Positivos**  **Iris-Versicolor** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **Falsos Negativos**  **Iris-Versicolor** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Positivos**  **Iris-Virginica** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Negativos**  **Iris-Virginica** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **% Total de Predicciones Falsas** | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

Conjunto de entrenamiento de 60%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 1.984459 | 1.803100 | 1.881171 | 1.827861 | 1.869659 | 1.868986 | 1.863630 |
| **Costo Maximo** | 38.381915 | 41.621181 | 41.435530 | 41.757116 | 46.199514 | 53.189468 | 58.035362 |
| **Error de Prueba (MSE)** | 0.016667 | 0.016667 | 0.016667 | 0.016667 | 0.016667 | 0.016667 | 0.016667 |
| **Falsos Positivos**  **Iris-Setosa** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Negativos**  **Iris-Setosa** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Positivos**  **Iris-Versicolor** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **Falsos Negativos**  **Iris-Versicolor** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Positivos**  **Iris-Virginica** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Falsos Negativos**  **Iris-Virginica** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| **% Total de Predicciones Falsas** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Conjunto de entrenamiento de 70%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 1.775510 | 1.628615 | 1.695151 | 1.650556 | 1.665571 | 1.669830 | 1.659271 |
| **Costo Maximo** | 42.629978 | 47.014808 | 46.916276 | 46.820422 | 53.061240 | 60.392167 | 67.516309 |

Conjunto de entrenamiento de 80%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 1.906754 | 1.751057 | 1.822988 | 1.771836 | 1.798815 | 1.806180 | 1.793136 |
| **Costo Maximo** | 46.564774 | 51.775103 | 51.609070 | 50.751200 | 59.949303 | 67.583808 | 76.917313 |

Conjunto de entrenamiento de 90%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nº Neuronas** | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Costo Minimo** | 1.866960 | 1.759531 | 1.816440 | 1.779187 | 1.802952 | 1.802069 | 1.801186 |
| **Costo Maximo** | 50.241914 | 55.724549 | 57.067957 | 55.047730 | 65.051474 | 73.203559 | 86.173810 |

Podemos observar que en las primeras instancias hubieron falsos negativos/positivos que involucraban a una de las clases por clasificar. Sin embargo, este error fue disminuyendo a medida que se incluía un mayor porcentaje del conjunto inicial para el entrenamiento de la red neuronal.

**Conclusiones**

* Las redes neuronales son un modelo bastante eficiente para la clasificación de datos. Además, el uso de Backpropagation potencia el aprendizaje de este modelo de manera significativa.
* La selección adecuada de la tasa de aprendizaje puede disminuir el número de iteraciones necesarias en la minimización de la función de costo.
* La selección de un mayor número de neuronas para las capas ocultas de la red neuronal ayuda a minimizar más la función de costos en la mayoría de los casos, pero no neceariamente afecta de forma importante la tasa de errores de predicción que la red pueda obtener. Lo que hace que la selección del número adecuado de neuronas para las capas ocultas sea un proceso metodológico, esto para identificar cuáles configuraciones de arquitectura modelan mejor la red para la clasificación deseada.
* Se evidencia la importancia de un conjunto de entrenamiento balanceado en ejemplos de todas las clases a clasificar para que el aprendizaje de la red sea más efectivo.