



Nombre: Dana Carolina Ramírez Velázquez

Código: 220286547

División: Tecnologías para la integración Ciber-Humana

Carrera: Ingeniería en computación

Materia: Seminario de Inteligencia Artificial

Tarea: Practica 1 Ejercicio 4

Fecha: 15/10/23

## Introducción

- Un perceptrón multicapa es una arquitectura de red neuronal artificial que consta de múltiples capas de nodos (neuronas) interconectados. A diferencia de un perceptrón simple, que tiene solo una capa de entrada y una capa de salida, un perceptrón multicapa tiene al menos una capa oculta entre la capa de entrada y la capa de salida.
- En este ejercicio, tenemos 3 neuronas en la capa de salida que indican el tipo de planta que es.

## Desarrollo

- Funcion que particiona el dataframe para entrenamiento y testeo

```
11 def particionar(dataset, e, pTrain, name):
12     for i in range(e):
13
14         df = pd.read_csv(dataset, header=None)
15         p_train = pTrain / 100
16         df['is_train'] = np.random.uniform(0, 1, len(df)) <= p_train
17         train, test = df[df['is_train']==True], df[df['is_train']==False]
18         df = df.drop(['is_train'], axis=1)
19         train.pop('is_train')
20         test.pop('is_train')
21
22         trainData = pd.DataFrame(train)
23         testData = pd.DataFrame(test)
24         |
25         trainData.to_csv("./info/"+name+str(i+1)+"_trn.csv",header=False, index=False)
26         testData.to_csv("./info/"+name+str(i+1)+"_tst.csv",header=False, index=False)
27
```

- Funcion de entrenamiento del modelo, donde se utilizan las funciones softmax y relu

```

42 def feedforward(self, inputs):
43     layer_outputs = [inputs]
44     for i in range(len(self.weights)):
45         layer_input = np.dot(self.weights[i], layer_outputs[-1]) + self.biases[i]
46         if i == len(self.weights) - 1:
47             layer_output = self.softmax(layer_input) # Softmax activation for output layer
48         else:
49             layer_output = self.relu(layer_input) # ReLU activation for hidden layers
50         layer_outputs.append(layer_output)
51     return layer_outputs
52
53
54 def train(self, training_data, labels):
55     for epoch in range(self.epochs):
56         for inputs, label in zip(training_data, labels):
57             # Forward propagation
58             layer_outputs = self.feedforward(inputs)
59             predictions = layer_outputs[-1]
60
61             # Backpropagation
62             # Cross-entropy error calculation
63             errors = [-label * np.log(predictions)]
64             deltas = [errors[0] * self.softmax_derivative(predictions)]
65
66             # Calculate errors and deltas for hidden layers
67             for i in range(len(self.weights) - 2, -1, -1):
68                 error = deltas[0].dot(self.weights[i + 1])
69                 errors.insert(0, error)
70                 deltas.insert(0, errors[0] * self.relu_derivative(layer_outputs[i + 1]))
71
72             # Update weights and biases
73             for i in range(len(self.weights)):
74                 self.weights[i] += self.learning_rate * np.outer(deltas[i], layer_outputs[i])
75                 self.biases[i] += self.learning_rate * deltas[i]
76

```

- Funcion de prediccion y traduccion de datos a un arreglo

```

78
79 def predict(self, inputs):
80     layer_outputs = self.feedforward(inputs)
81     predictions = (layer_outputs[-1]).tolist()
82
83     print(predictions)
84     result = [-1, -1, -1]
85     a = predictions.index(max(predictions))
86     result[a] = 1
87     return result

```

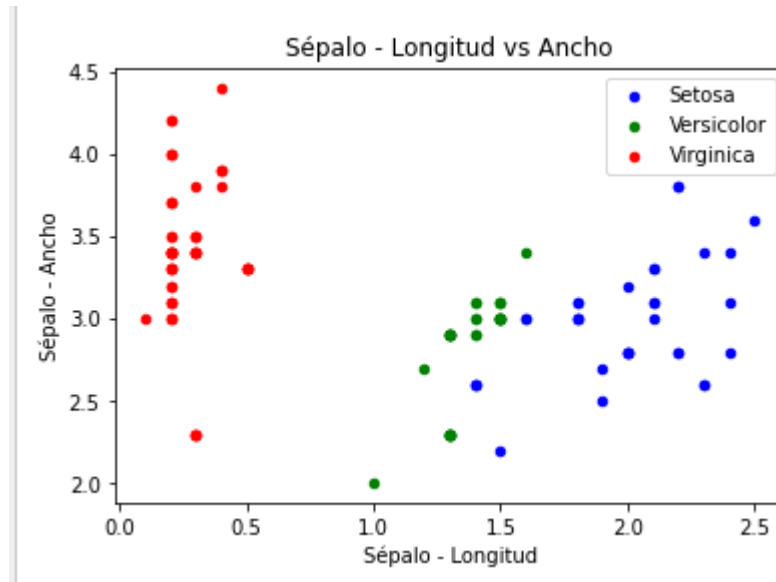
- Resultados

Leave-K-Out

Precisión en el conjunto de prueba: 46.43%

Desviación Estándar: 34.61%

Error Esperado: 53.57142857142857%



## Referencias:

- Mathivet, V. (2018). *Inteligencia artificial para desarrolladores: conceptos e implementación en c*. Ediciones ENI..
- Boden, M. A. (2017). *Inteligencia artificial*. Turner.
- Cerrillo Martínez, A. (2019). *El impacto de la inteligencia artificial en el derecho administrativo¿ nuevos conceptos para nuevas realidades técnicas?*.
- Leyva-Vázquez, M., & Smarandache, F. (2018). *Inteligencia Artificial: retos, perspectivas y papel de la Neutrosofía*. Infinite Study.
- Leyva-Vázquez, M., & Smarandache, F. (2018). *Inteligencia Artificial: retos, perspectivas y papel de la Neutrosofía*. Infinite Study.