

Universidad de Ingeniería Y Tecnología

Global Academy



Introduction to Intelligent Control Systems

Proyecto 3

Castro Quispe, Eduardo Sebastián

Palma Rodríguez, Diego Alonso V.

Instructor: Diego Benítez, Ph.D.

Lima - Peru

2021-1

1. Redes Neuronales: *Gradient Descent*

1.1. Sonar Dataset

Para este proyecto hemos usado el dataset del repositorio de Machine Learning del UCI. Nuestro objetivo con este dataset es evaluar si el objeto es un mineral o una roca, para esto el dataset nos provee una serie de 60 ángulos para determinar esa entidad, por cada ángulo tenemos 208 datos. En esta caso, el valor 0 significa roca y 1 mineral.

1.2. Red neuronal

Con los códigos proveídos en clase hemos sido capaces de implementar una simple red neuronal para resolver el problema.

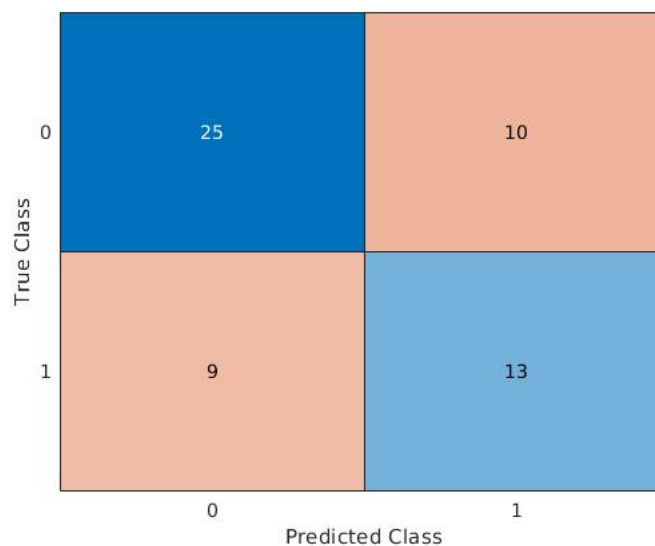
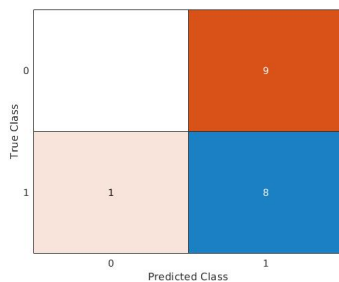


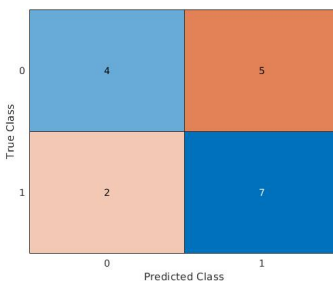
Figura 1: Confusion Matrix of a perceptron with 0,85 learning rate and 150 training samples

Este perceptron ha sido probado con 57 muestras. Su accuracy es del 67 % y su precisión es del 59 %, lo cual nos da una seguridad de la certitud de la red. Según los comentarios del dataset, el accuracy esperado es del 53 %, hemos superado por 14 % el rendimiento esperado.

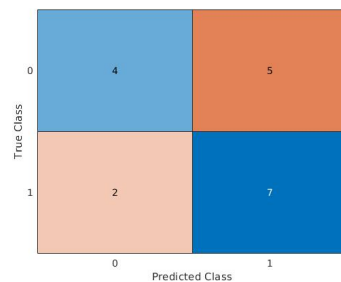
1.3. Distintos entrenamientos



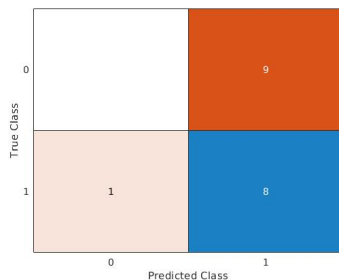
(a) 0.01 learning rate and 26 training samples



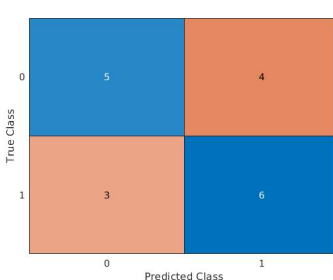
(b) 0.01 learning rate and 100 training samples



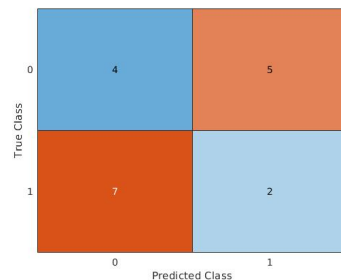
(c) 0.01 learning rate and 176 training samples



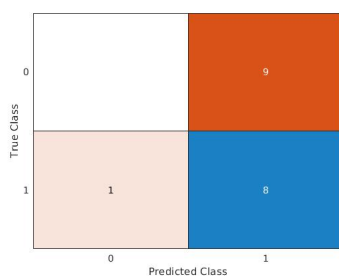
(d) 0.5 learning rate and 26 training samples



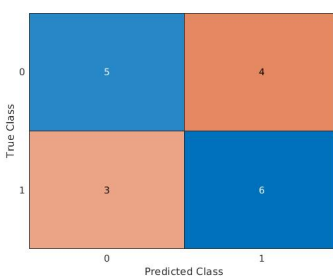
(e) 0.5 learning rate and 100 training samples



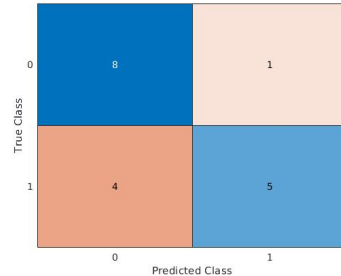
(f) 0.5 learning rate and 176 training samples



(g) 0.95 learning rate and 26 training samples



(h) 0.95 learning rate and 100 training samples



(i) 0.95 learning rate and 176 training samples

Figura 2: Entrenamiento con distintos tasas de aprendizajes y muestras de entrenamiento

Hemos variado entrenamientos con distintas tasas y muestras de entrenamiento. Hemos entrenado las tasas 0,01, 0,50, 0,95 y muestras 26, 100, 176, todos los entrenamientos fueron testeados con 18 datos.

	26 training samples	100 training samples	176 training samples
0,01 Learning rate	44 %	61 %	61 %
0,50 Learning rate	44 %	61 %	33 %
0,95 Learning rate	44 %	61 %	72 %

Tabla 1: Accuracy de los distintos entrenamientos

Podemos observar que el accuracy no varía mucho cuando los datos son pequeños, el learning rate deja de tener relevancia en ese punto. Por otro lado, cuando se tiene una cantidad considerable de datos es preferible aumentar el learning rate. El learning rate no varía el accuracy cuando tenemos datos menores o iguales a 100; sin embargo, cuando ya tenemos una cantidad superior a ella el learning rate se comporta de manera rara, teniendo el mejor resultado con el learning rate 0,95.

2. Redes Neuronales: Toolbox Neural Pattern Recognition

En esta sección utilizaremos el toolbox *Neural Pattern Recognition* de MATLAB para entrenar, validar y evaluar nuestra red neuronal. En primera instancia solo se hará uso de un perceptrón y luego se utilizará una red neuronal multicapa. El *dataset* utilizado es el mismo que el descrito en la sección 1.1. A continuación se muestra el detalle de ello.

2.1. Perceptrón

En esta subsección se evaluará el comportamiento de una red neuronal con una sola capa variando el porcentaje de muestras de entrenamiento. Para nuestros propósitos mantendremos el porcentaje de las muestras de validación en 5 %, puesto que no podemos prescindir de ellas en la aplicación.

En la tabla 2 se muestra el porcentaje de muestra de entrenamiento, validación y prueba que se utilizará.

Tabla 2: *Porcentaje de muestras*

	Entrenamiento	Prueba	Validación
% Muestras	60 %	35 %	5 %
	65 %	30 %	
	70 %	25 %	
	75 %	20 %	
	80 %	15 %	

A continuación, en la Fig.3 se muestra la matriz de confusión obtenida con cada conjunto de entrenamiento y prueba (Tabla 2).

En la Tabla 3 se muestra un resumen del *test accuracy* para cada conjunto de muestras. En general, se puede observar que el *accuracy* está entre un 78 % y 82 % utilizando un perceptrón y el método de *cross-entropy* del toolbox de MATLAB. Asimismo, el mejor *accuracy* se obtuvo para un conjunto de muestras de entrenamiento del 80 %.

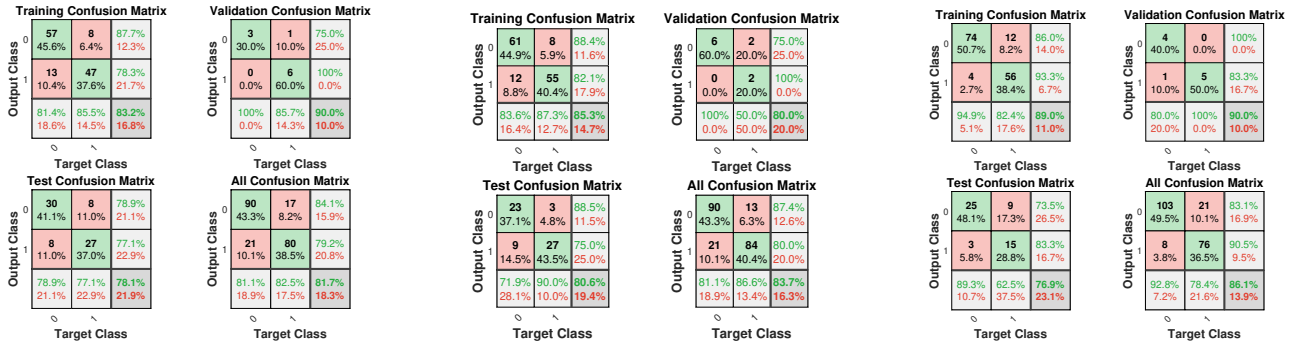
Tabla 3: *Perceptrón: Accuracy*

	Entrenamiento	Prueba	Validación	Test Accuracy
% Muestras	60 %	35 %	5 %	78.1 %
	65 %	30 %		80.6 %
	70 %	25 %		76.9 %
	75 %	20 %		76.3 %
	80 %	15 %		81.4 %

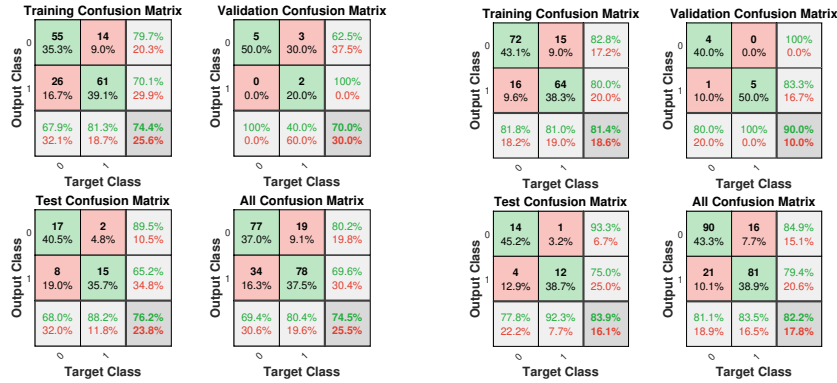
2.2. Red neuronal multicapa

En esta sección se utilizó una red neuronal multicapa en lugar de un perceptrón. Para ello se utilizaron $n = 5, 10, 15$ neuronas para las dos muestras que obtuvieron ua mayor precisión (Tabla 3). A partir de ello, se utilizó el toolbox de MATLAB para entrenar nuestra red. En la Figura 4 se evidencian las matrices de confusión para cada caso.

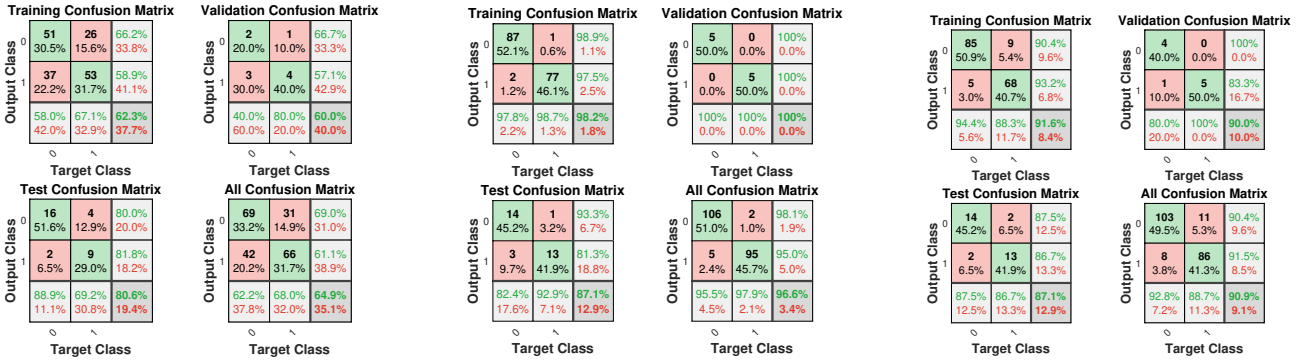
En la Tabla 4 se muestra un resumen de los *test accuracy* para cada caso. Como se puede observar, se tienen un mayor *accuracy* a medida que aumentamos el número de neuronas. Sin embargo, esto hace que



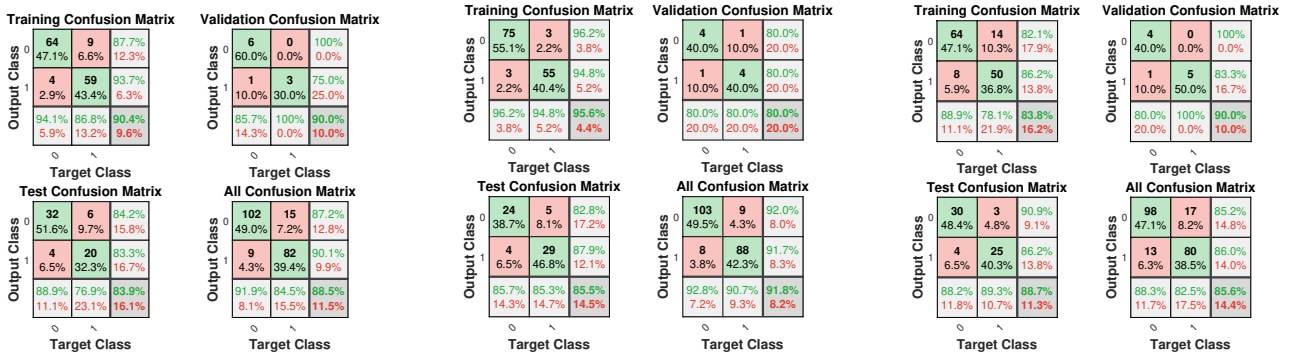
(a) Entrenamiento: 60 %, Prueba:35 % (b) Entrenamiento: 65 %, Prueba:30 % (c) Entrenamiento: 70 %, Prueba:25 %



(d) Entrenamiento: 75 %, Prueba: 20 % (e) Entrenamiento: 80 %, Prueba:15 %
Figura 3: Matrices de confusión



(a) 5 capas. E:1 80 %, P:15 % (b) 10 capas. E:80 %, P:15 % (c) 15 capas. E:80 %, P:15 %



(d) 5 capas. E:65 %, P:30 % (e) 10 capas. E:65 %, P:30 % (f) 15 capas. E:65 %, P:30 %
Figura 4: Matrices de confusión

aumente el costo computacional del entrenamiento de la red. En caso habláramos de millones de datos, bastaría con utilizar 5 neuronas, puesto que el *accuracy* es relativamente alto.

Tabla 4: *Red neuronal multicapa: Test Accuracy*

	Entrenamiento	Prueba	Validación	Nº de neuronas	Test Accuracy
% Muestras	65 %	30 %	5 %	5	83.9 %
				10	85.5 %
				15	88.7 %
	80 %	15 %		5	80.6 %
				10	87.1 %
				15	87.1 %

3. Redes Neuronales - Clasificador multiclase

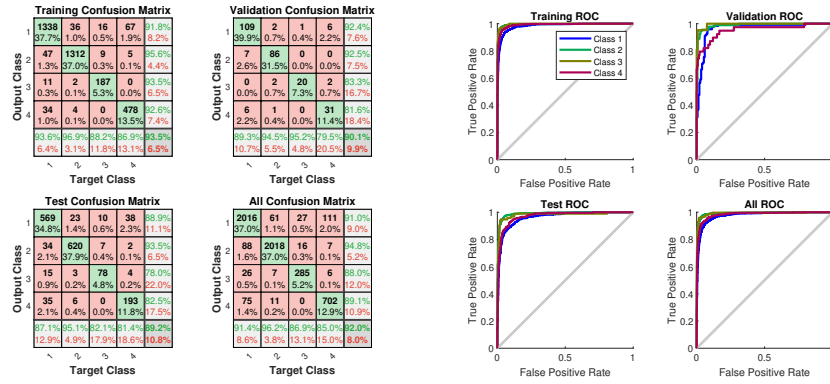
En esta sección desarrollaremos un clasificador multiclase mediante redes neuronales. Para ello utilizaremos el *dataset* de la navegación de un robot móvil seguidor de pared (Figura 5). Estos datos fueron recolectados mientras el robot SCITOS G5 navegaba por la habitación siguiendo la pared en el sentido de las agujas del reloj, durante 4 rondas, utilizando 24 sensores de ultrasonido dispuestos circularmente alrededor de su cintura. En ese sentido el *dataset* contiene los valores brutos de las mediciones de los 24 sensores de ultrasonidos y la etiqueta de clase correspondiente: *Move-Forward*, *Sharp-Right-Turn*, *Slight-Left-Turn* y *Slight-Right-Turn*. Las lecturas de los sensores se muestrean a una frecuencia de 9 muestras por segundo. Cabe mencionar que en total se proporcionan 5456 datos para cada una de los 24 sensores.



Figura 5: Robot móvil

Para poder entrenar nuestra red neuronal es necesario convertir las clases categóricas a valores numéricos, por lo que se convierte cada una de las clases a vectores binarios. La clase *Move-Forward* se representa con el vector $[1, 0, 0, 0]$; la clase *Sharp-Right-Turn*, con el vector $[0, 1, 0, 0]$; la clase *Slight-Left-Turn*, con el vector $[0, 0, 1, 0]$; y la clase *Slight-Right-Turn*, con el vector $[0, 0, 0, 1]$. Una vez realizado ello se ingresan los datos de entrada y las salidas deseadas. En base al conocimiento adquirido en las secciones previas, se utiliza una muestra del 65 % para el entrenamiento, el 30 % para la prueba y 5 % para la validación. Asimismo, se utilizan 15 neuronas ocultas.

A partir de ello se se entrena la red y se grafica la matriz de confusión. En la Figura 6a se observa la matriz de confusión obtenida. Se puede observar que se obtiene un *test accuracy* del 89.2%. Es decir, se obtiene una alta precisión en la predicción. Además de ello, se puede observar que el robot es más propenso a confundirse cuando tiene que seguir adelante o girar suavemente a la derecha. Esto también se puede observar en la gráfica del Test ROC en la Figura 6b.



(a) Matriz de confusión (b) ROC
Figura 6: Métricas de evaluación

4. Conclusiones

Se cumplieron con los objetivos de crear una red neural y entrenarla con distintos ratios y datos de aprendizaje. Hemos encontrado que los ratios de aprendizaje pierden valor en *datasets* pequeños de entrenamiento. Con una suficiente cantidad de datos el *learning rate* se comporta de manera inusual haciendo que a mayor *learning rate*, mayor *accuracy*. Además de ello, se logró implementar correctamente una red neuronal con 15 capas ocultas para una clasificación multiclase, de tal forma que logre predecir el movimiento que debe realizar el robot con un 89.2% de *accuracy*.

Con el objetivo de mejorar el presente proyecto se propone como trabajo futuro realizar, previo a la clasificación, un análisis de componentes principales de los datos. Lo que se busca con este análisis es reducir la cantidad de características y determinar cuáles son las que más influyen en la predicción de la clase. De este modo se aumentaría la eficiencia computacional en caso se tuviera una mayor cantidad de datos y se espera aumentar la precisión.