

## Laboratorio 4

Johan Steven Benavides Guarnizo-88593  
Sebastian Morales Devia - 73487

Universidad ECCI  
Facultad de Ingeniería

Elias Buitrago Bolivar

Bogota  
2024

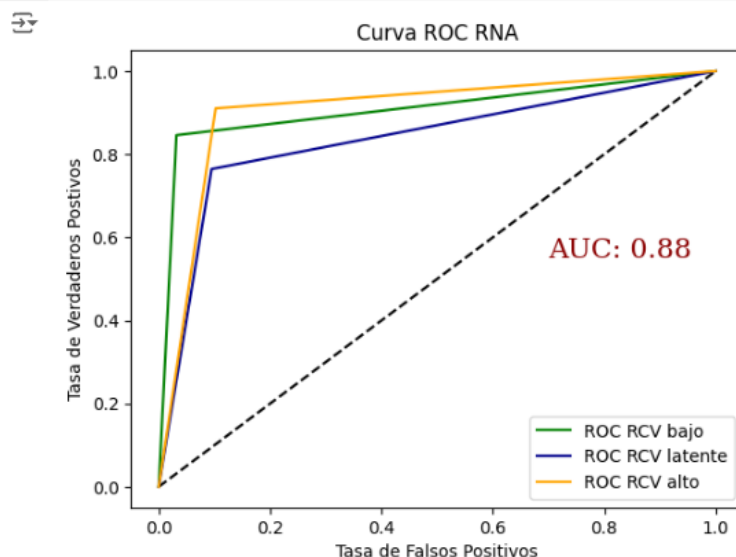
1. En la prueba numero 1 se utilizaron 2 capas ocultas de 4 y 22 neuronas respectivamente y una capa de salida con 3 neuronas

```
[66] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(4, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera capa oculta
modelRNA.add(Dense(22, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

```
[67] # compile the keras (tensorflow) flow graph
modelRNA.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),
                 loss='binary_crossentropy',
                 metrics=['accuracy'])
```

```
[69] training_log = modelRNA.fit(X_train,
                                Y_train,
                                epochs=200,
                                batch_size=32,
                                validation_data=(X_valid, Y_valid),
                                verbose=1)
```

```
# Curva ROC
# https://stackabuse.com/understanding-roc-curves-with-python/
Y_pred = np_utils.to_categorical(y_pred)
auc = roc_auc_score(Y_test, Y_pred)
fpr, tpr, threshold = roc_curve(Y_test[:,0], Y_pred[:,0])
font = {'family': 'serif',
        'color': 'darkred',
        'weight': 'normal',
        'size': 16,
        }
plt.plot(fpr, tpr, color='green', label='ROC RCV bajo')
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC RNA')
fpr, tpr, threshold = roc_curve(Y_test[:,1], Y_pred[:,1])
plt.plot(fpr, tpr, color='darkblue', label='ROC RCV latente')
fpr, tpr, threshold = roc_curve(Y_test[:,2], Y_pred[:,2])
plt.plot(fpr, tpr, color='orange', label='ROC RCV alto')
plt.text(0.7, 0.55, 'AUC: %.2f' % auc, fontdict=font)
plt.legend()
plt.show()
```



2. En la prueba numero 2 se utiliza 2 capas ocultas con 7 y 15 neuronas respectivamente y una capa de 3 neuronas

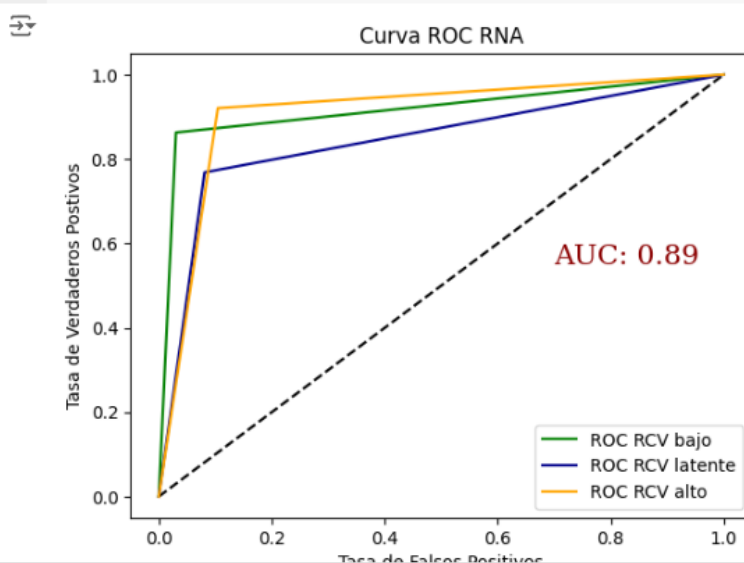
```
[228] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(7, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera capa oculta
modelRNA.add(Dense(15, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

## ✓ Entrenamiento del modelo de RNA

```
✓ [230] # Inicializar el reloj para calcular tiempo de cómputo
0 s t0 = process_time()
```

```
✓ [231] training_log = modelRNA.fit(X_train,
1 m Y_train,
epochs=200,
batch_size=32,
validation_data=(X_valid, Y_valid),
verbose=1)
```

```
✓ # Curva ROC
s # https://stackabuse.com/understanding-roc-curves-with-python/
Y_pred = np_utils.to_categorical(y_pred)
auc = roc_auc_score(Y_test, Y_pred)
fpr, tpr, threshold = roc_curve(Y_test[:,0], Y_pred[:,0])
font = {'family': 'serif',
'color': 'darkred',
'weight': 'normal',
'size': 16,
}
plt.plot(fpr, tpr, color='green', label='ROC RCV bajo')
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC RNA')
fpr, tpr, threshold = roc_curve(Y_test[:,1], Y_pred[:,1])
plt.plot(fpr, tpr, color='darkblue', label='ROC RCV latente')
fpr, tpr, threshold = roc_curve(Y_test[:,2], Y_pred[:,2])
plt.plot(fpr, tpr, color='orange', label='ROC RCV alto')
plt.text(0.7, 0.55, 'AUC: %.2f' % auc, fontdict=font)
plt.legend()
plt.show()
```



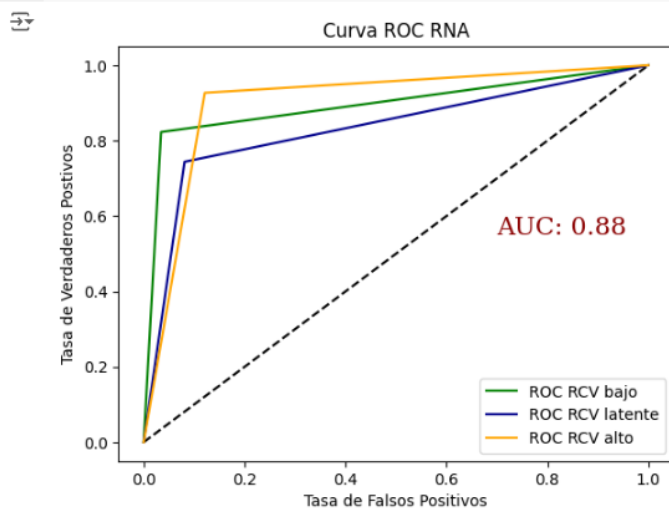
3. En la prueba numero 3 se utiliza 4 capas ocultas con 4, 25, 25 y 26 y una capa de 3 neuronas

```
[300] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(4, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera capa oculta
modelRNA.add(Dense(25, activation='relu'))
modelRNA.add(Dense(25, activation='relu'))
modelRNA.add(Dense(26, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

```
✓ 0 s [320] # Inicializar el reloj para calcular tiempo de cómputo
      t0 = process_time()
```

```
✓ 48 s [321] training_log = modelRNA.fit(X_train,
                                         Y_train,
                                         epochs=200,
                                         batch_size=32,
                                         validation_data=(X_valid, Y_valid),
                                         verbose=1)
```

```
✓ 0 s # Curva ROC
# #https://stackabuse.com/understanding-roc-curves-with-python/
Y_pred = np_utils.to_categorical(y_pred)
auc = roc_auc_score(Y_test, Y_pred)
fpr, tpr, threshold = roc_curve(Y_test[:,0], Y_pred[:,0])
font = {'family': 'serif',
        'color': 'darkred',
        'weight': 'normal',
        'size': 16,
        }
plt.plot(fpr, tpr, color='green', label='ROC RCV bajo')
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC RNA')
fpr, tpr, threshold = roc_curve(Y_test[:,1], Y_pred[:,1])
plt.plot(fpr, tpr, color='darkblue', label='ROC RCV latente')
fpr, tpr, threshold = roc_curve(Y_test[:,2], Y_pred[:,2])
plt.plot(fpr, tpr, color='orange', label='ROC RCV alto')
plt.text(0.7, 0.55, 'AUC: %.2f' % auc, fontdict=font)
plt.legend()
plt.show()
```



4. En la prueba numero 4 utiliza 3 capas ocultas con 4, 12, 19 y una capa de 3 neuronas

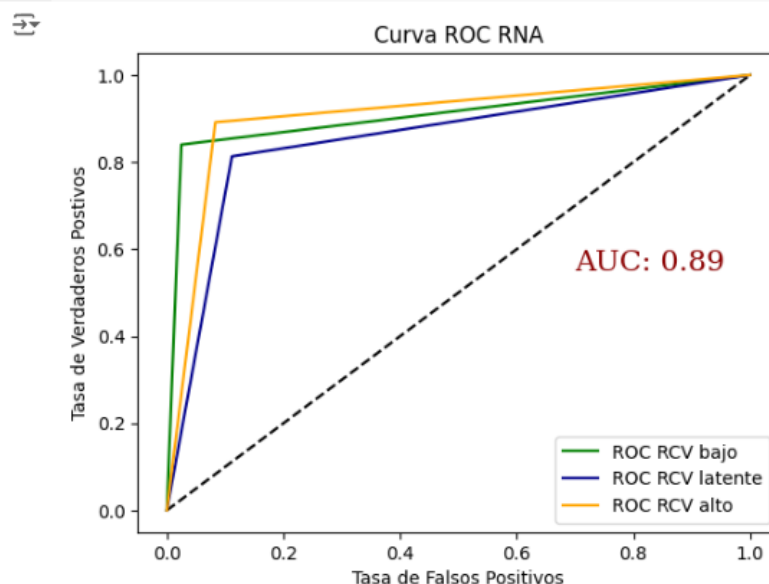
```
0s [338] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(4, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera capa oculta
modelRNA.add(Dense(12, activation='relu'))
modelRNA.add(Dense(19, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

## ✓ Entrenamiento del modelo de RNA

```
0s [340] # Inicializar el reloj para calcular tiempo de cómputo
t0 = process_time()
```

```
0s [341] training_log = modelRNA.fit(X_train,
                                   Y_train,
                                   epochs=200,
                                   batch_size=32,
                                   validation_data=(X_valid, Y_valid),
                                   verbose=1)
```

```
5s [342] # Curva ROC
# #https://stackabuse.com/understanding-roc-curves-with-python/
Y_pred = np_utils.to_categorical(y_pred)
auc = roc_auc_score(Y_test, Y_pred)
fpr, tpr, threshold = roc_curve(Y_test[:,0], Y_pred[:,0])
font = {'family': 'serif',
        'color': 'darkred',
        'weight': 'normal',
        'size': 16,
        }
plt.plot(fpr, tpr, color='green', label='ROC RCV bajo')
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC RNA')
fpr, tpr, threshold = roc_curve(Y_test[:,1], Y_pred[:,1])
plt.plot(fpr, tpr, color='darkblue', label='ROC RCV latente')
fpr, tpr, threshold = roc_curve(Y_test[:,2], Y_pred[:,2])
plt.plot(fpr, tpr, color='orange', label='ROC RCV alto')
plt.text(0.7, 0.55, 'AUC: %.2f' % auc, fontdict=font)
plt.legend()
plt.show()
```



5. En la prueba numero 3 utiliza 3 capas ocultas con 7, 7, 7 y una capa de 3 neuronas

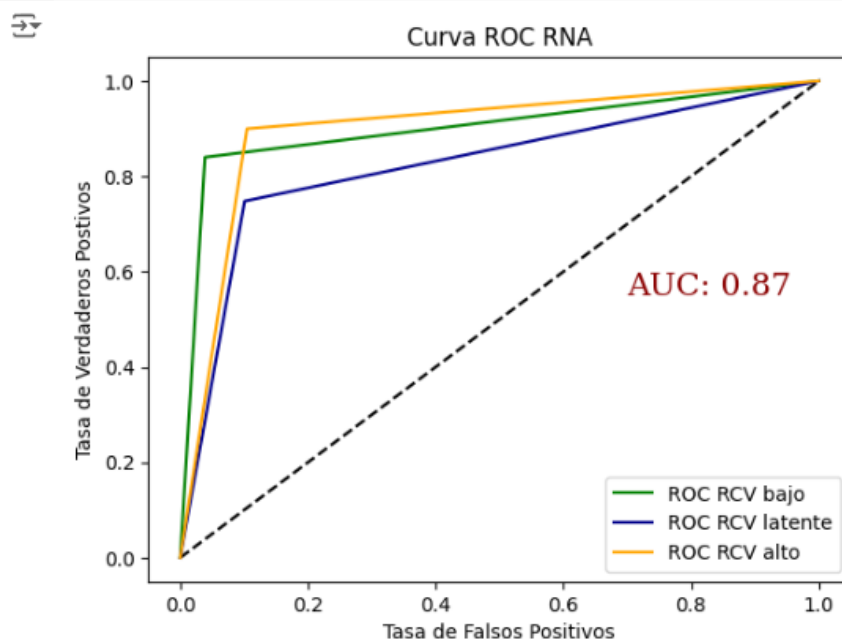
```
[407] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(7, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera capa oculta
modelRNA.add(Dense(7, activation='relu'))
modelRNA.add(Dense(7, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

## Entrenamiento del modelo de RNA

```
[409] # Inicializar el reloj para calcular tiempo de cómputo
t0 = process_time()
```

```
[410] training_log = modelRNA.fit(X_train,
                                Y_train,
                                epochs=200,
                                batch_size=32,
                                validation_data=(X_valid, Y_valid),
                                verbose=1)
```

```
# Curva ROC
# #https://stackabuse.com/understanding-roc-curves-with-python/
Y_pred = np_utils.to_categorical(y_pred)
auc = roc_auc_score(Y_test, Y_pred)
fpr, tpr, threshold = roc_curve(Y_test[:,0], Y_pred[:,0])
font = {'family': 'serif',
        'color': 'darkred',
        'weight': 'normal',
        'size': 16,
        }
plt.plot(fpr, tpr, color='green', label='ROC RCV bajo')
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
plt.title('Curva ROC RNA')
fpr, tpr, threshold = roc_curve(Y_test[:,1], Y_pred[:,1])
plt.plot(fpr, tpr, color='darkblue', label='ROC RCV latente')
fpr, tpr, threshold = roc_curve(Y_test[:,2], Y_pred[:,2])
plt.plot(fpr, tpr, color='orange', label='ROC RCV alto')
plt.text(0.7, 0.55, 'AUC: %.2f' % auc, fontdict=font)
plt.legend()
plt.show()
```



## Conclusiones

En la práctica realizada para mejorar la Curva ROC de una Red Neuronal Artificial (RNA), se llevaron a cabo diversas pruebas, cinco de las cuales se detallan en el documento actual. Se observó que a medida que se aumenta el número de capas ocultas, el resultado tiende a acercarse más a 1. Por otro lado se puede ver que tanto como con 4 capas ocultas como también con 2 se llegó a 0.89 entonces es algo curioso ya que no se sabe si es mejor o peor aumentar ya que las dos al final se acercaron bastante a 1.

Al final, se observó que agregar más capas ocultas con un alto número de neuronas puede acercar el resultado más hacia 1. Esto se debe a que estas configuraciones permiten realizar pruebas más complejas y variadas que no son posibles con solo dos capas ocultas. Como resultado, esto podría mejorar significativamente la precisión en las predicciones futuras.