

Laboratorio N.4 Redes Neuronales Artificiales Para Predecir Riesgo Cardiovascular

Brayan Camilo Salazar González

Jhoan Sebastián Riaño Herrera

Daniel Alejandro Soto

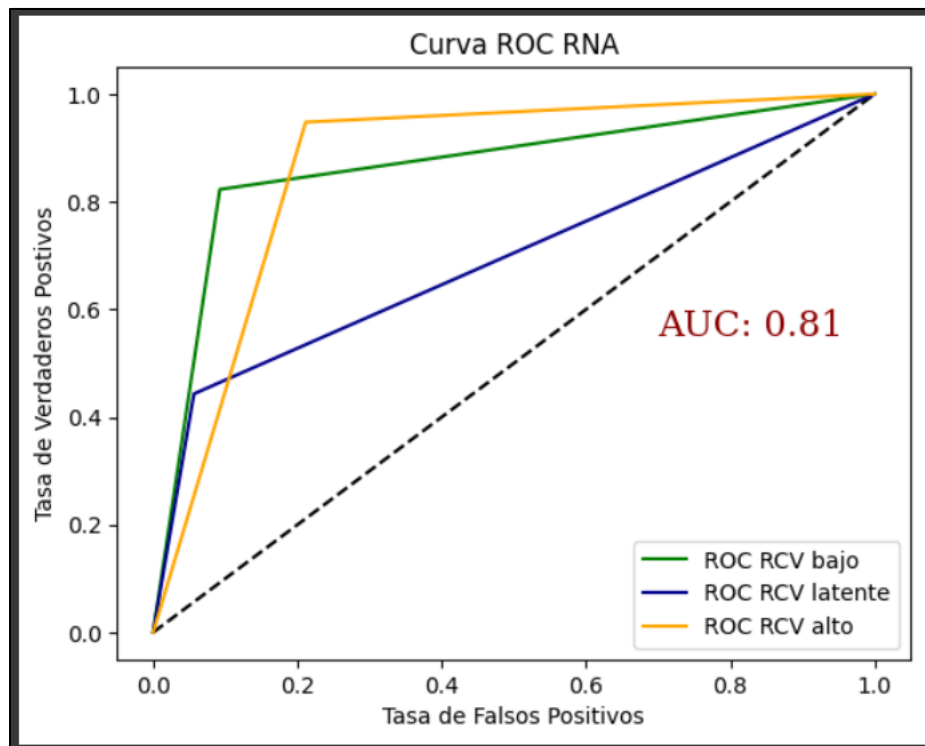
Jhonnatan Alexander Pérez

Joseph Oswald Quiroz Mejia

Desarrollo del laboratorio: Partiendo de la explicación dada por el docente, se plantea aumentar el valor de desempeño de la curva ROC (métricas de desempeño), para predecir el riesgo cardiovascular a partir de una base de datos limpia que posee 35 variables de datos.

Se ejecuta el código proporcionado por el docente donde el resultado de la curva ROC es de 81%

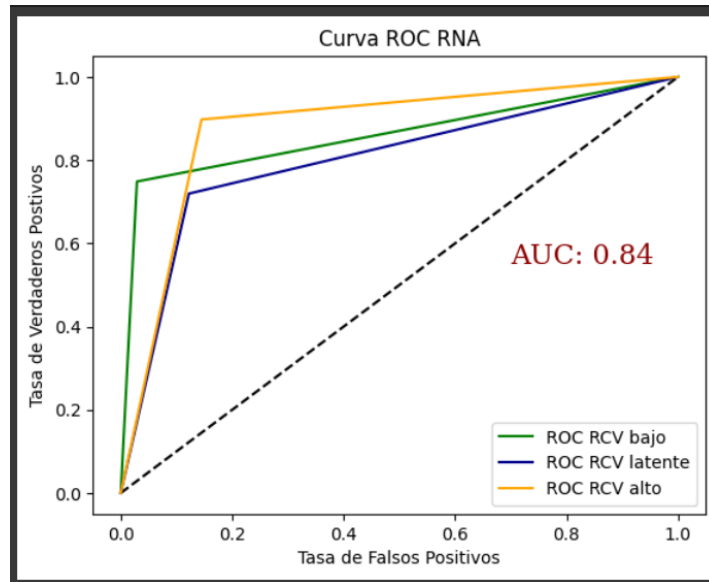
```
[1448] # Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(1, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape)
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```



Se dispone a aumentar este valor modificando específicamente el número de neuronas en la primera capa oculta y aumentando el número de capas ocultas de la siguiente manera.

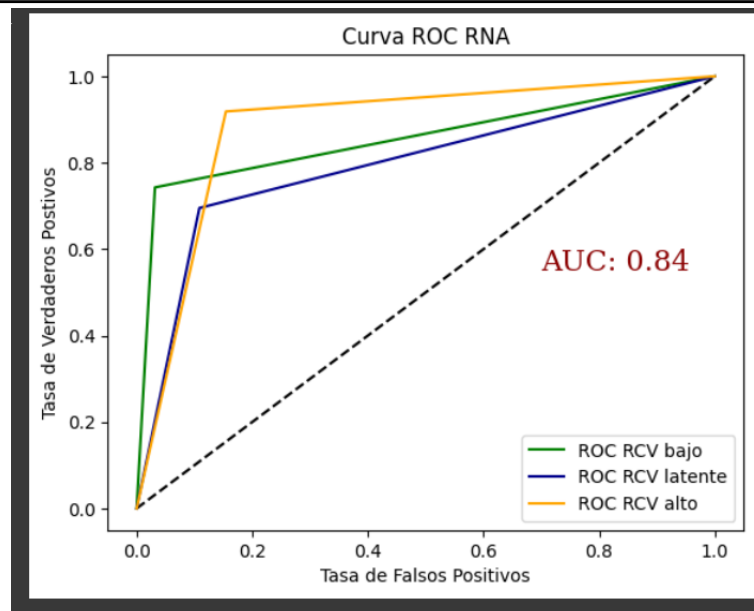
- 2 neuronas en la primera capa oculta con un resultado de la curva ROC de 84%.

```
# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(2, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```



- 5 neuronas en la primera capa oculta con un resultado de la curva ROC de 84%.

```
# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(5, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape) y #neuronas en la primera
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

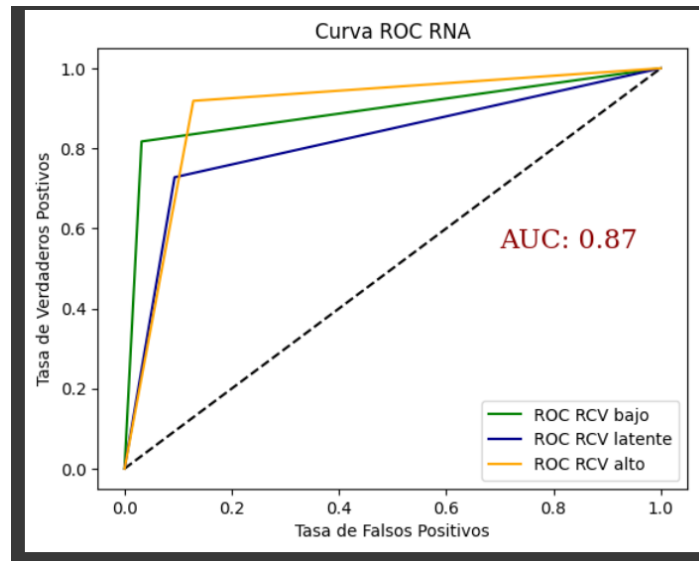


- 7 neuronas en la primera capa oculta con un resultado de la curva ROC de 87%.

```

# 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
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```

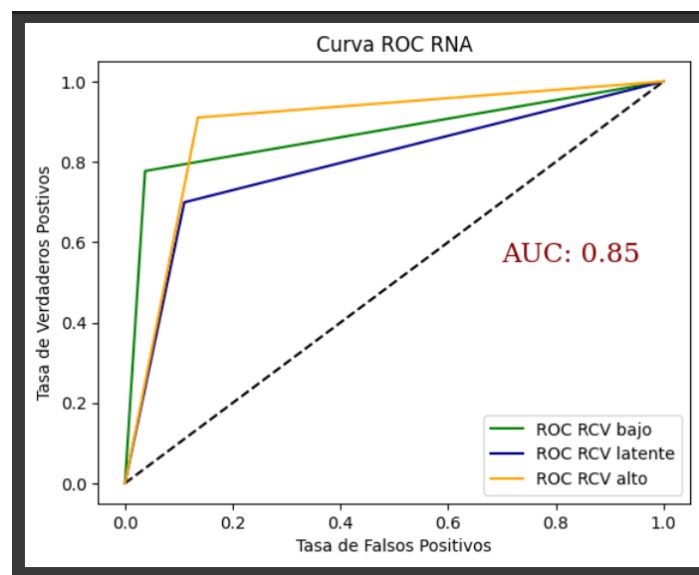


- 8 neuronas en la primera capa oculta con un resultado de la curva ROC de 85%.

```

# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(8, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```

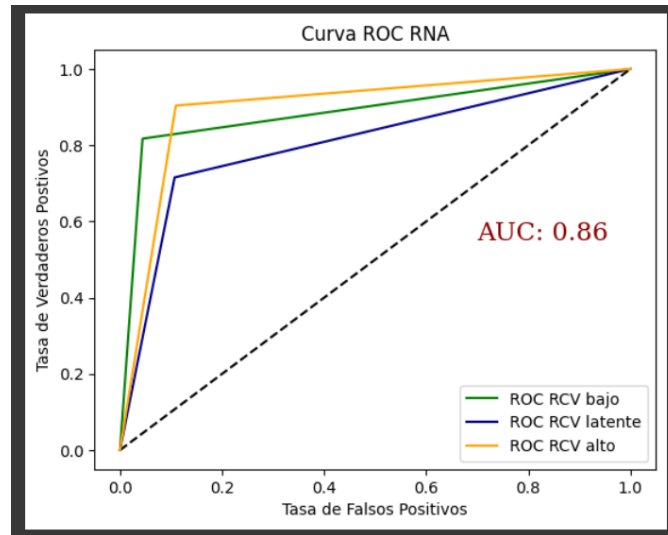


- 9 neuronas en la primera capa oculta con un resultado de la curva ROC de 86%.

```

# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(9, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape)
# modelRNA.add(Dense(1, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```

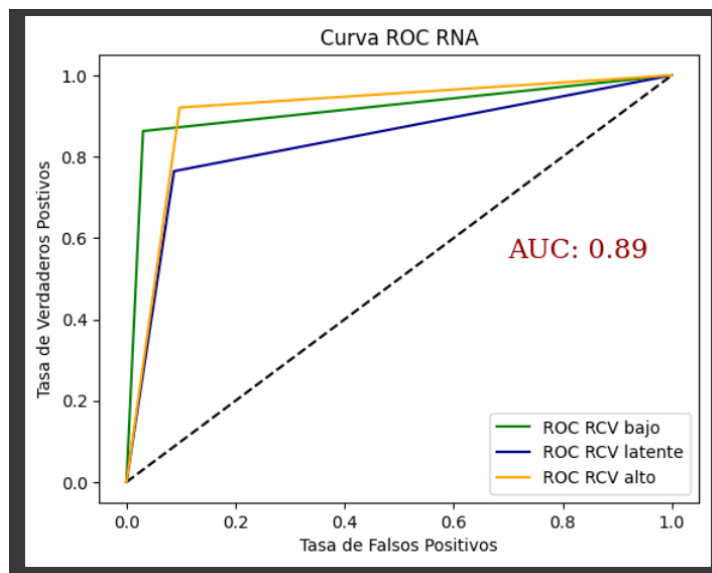


- 7 neuronas en la primera capa oculta, además se habilita la segunda capa oculta con 7 neuronas de igual forma dando como resultado de la curva ROC 89%.

```

# 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)
modelRNA.add(Dense(7, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```

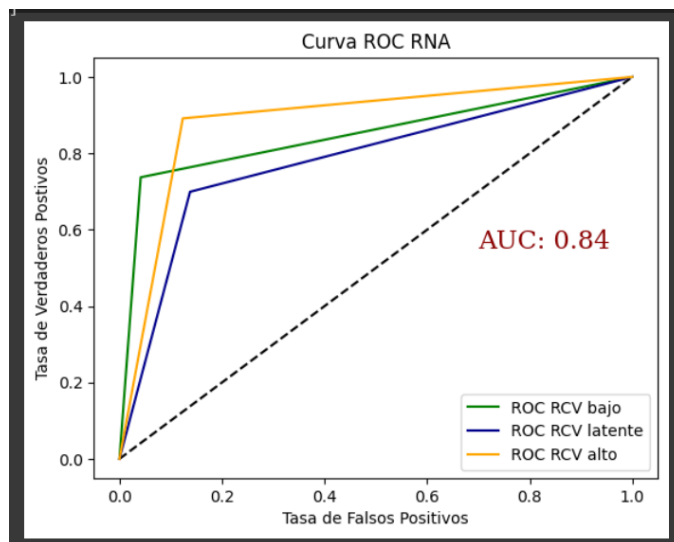


- 45 neuronas en la primera capa oculta con un resultado de la curva ROC 84%.

```

# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(45, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape=(None, 35))
#modelRNA.add(Dense(7, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```

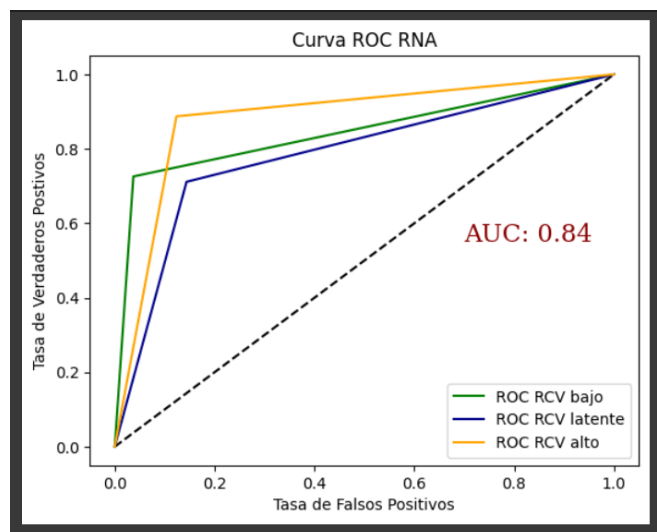


- 45 neuronas en la primera capa oculta y 45 neuronas en la segunda capa oculta dando como resultado de la curva ROC 84%.

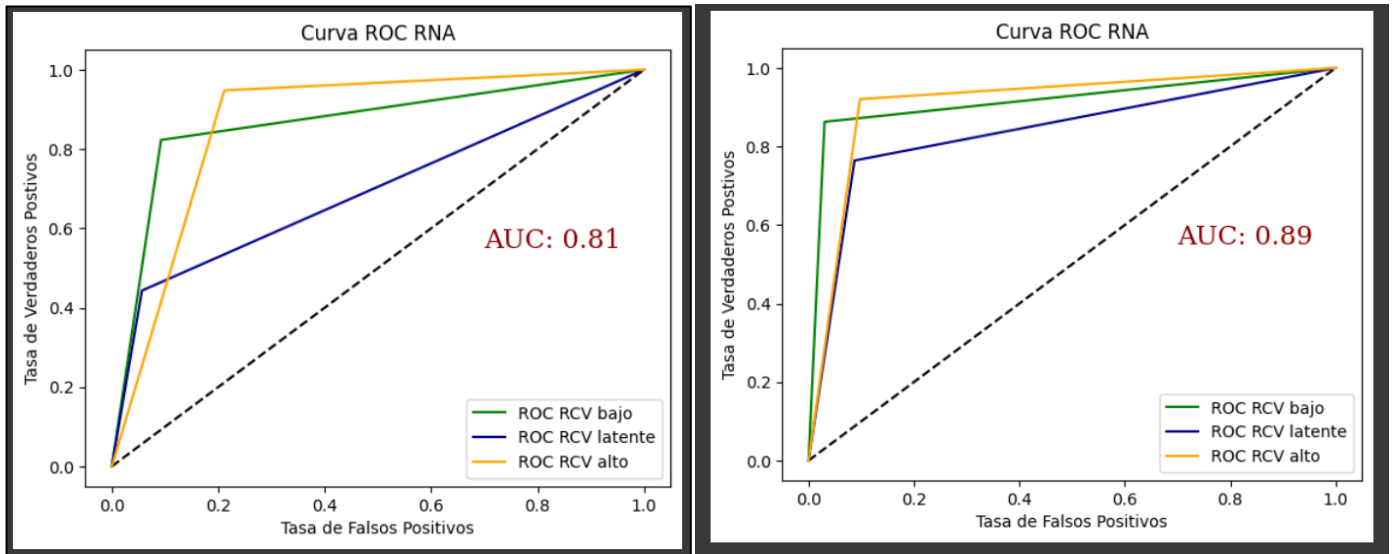
```

# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(45, batch_input_shape=(None, 35), activation='relu')) ## neuronas en la capa de entrada (batch_input_shape=(None, 35))
modelRNA.add(Dense(45, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))

```



Comparación



- Se observó que el aumentar el número de neuronas y en número de capas ocultas no conduce a un mejor resultado, en la experimentación se evidencia que la mejor configuración es tener dos capas ocultas cada una con 7 neuronas donde su métrica de desempeño es de 89%, de igual forma la configuración de una sola capa oculta 7 neuronas tiene un resultado de 87% que no es nada malo.
- Al comparar las graficas se observa que el ROC RCV latente mejora considerablemente pasando del 42% aproximadamente de tasa de verdaderos positivos al 75% aproximadamente.
- El ROC RCV bajo es más efectivo en la predicción de riesgo cardiovascular con un valor de 89% sin comprometer la tasa de falsos positivos donde es casi 0 lo que indica que el modelo tuvo una bastante mejoría, en el caso del RCV alto mejora la predicción específicamente en el eje de falsos positivos pasando del 20% al 10% aproximadamente.