Laboratorio No 4

Diego Nicolas Garcia Vargas

Marco David Suarez

Leydi Dayana Galindo

Elías Buitrago Bolivar

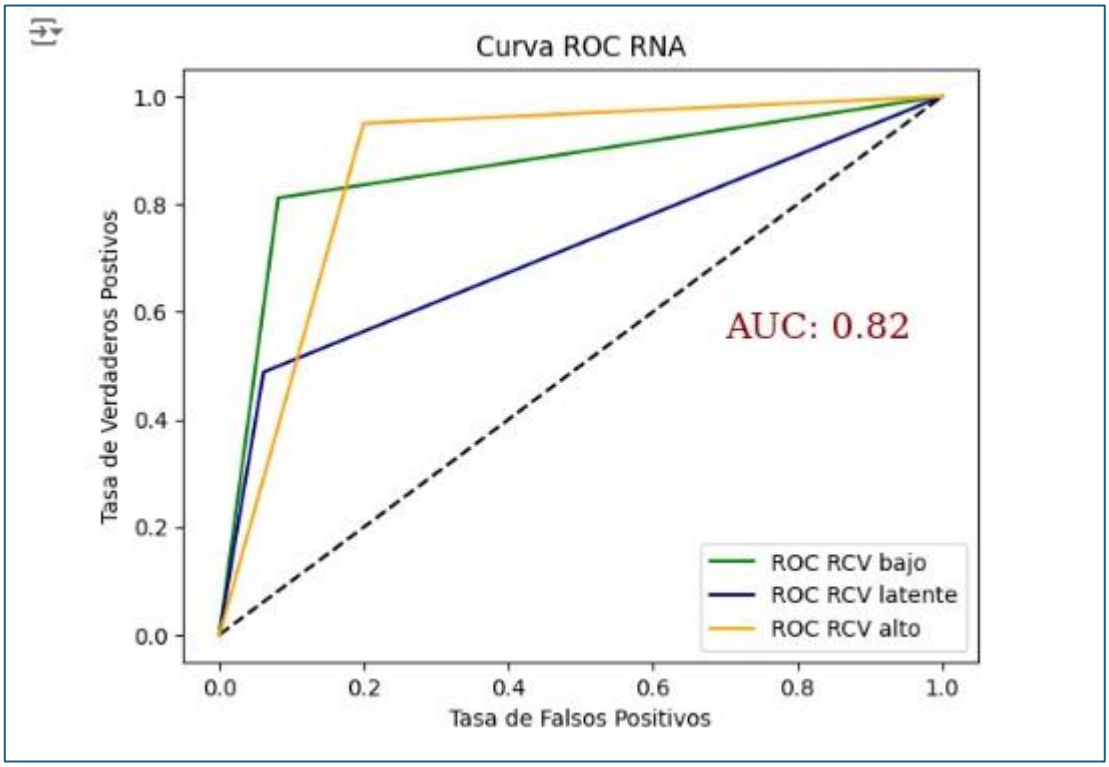
Seminario Big Data y Gestión de la Información

Universidad ECCI

Julio del 2024

Desarrollo

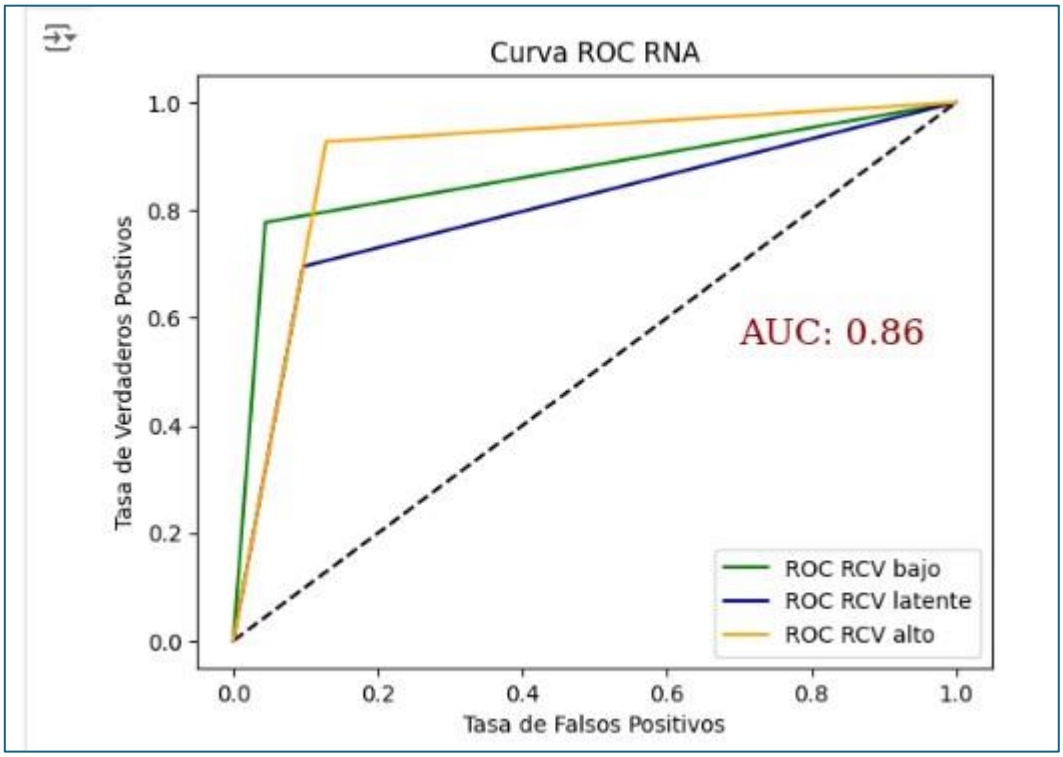
Inicialmente la arquitectura del modelo de red neuronal tiene 1 capa de entrada con 35 neuronas, una capa oculta de 1 neurona y una capa de salida con 3 neuronas. A lo que se obtiene una curva ROC de:



Al modelo anterior se le modifica la capa oculta para que tenga 8 neuronas y se optiene:

```
[22]  1 # Definir la arquitectura del modelo de la RNA
      2 modelRNA = models.Sequential()
      3 modelRNA.add(Dense(8, batch_input_shape=(None, 35), activation='relu')) ## ne
      4 #modelRNA.add(Dense(1, activation='relu'))
      5 modelRNA.add(Dense(3, activation='softmax'))
```

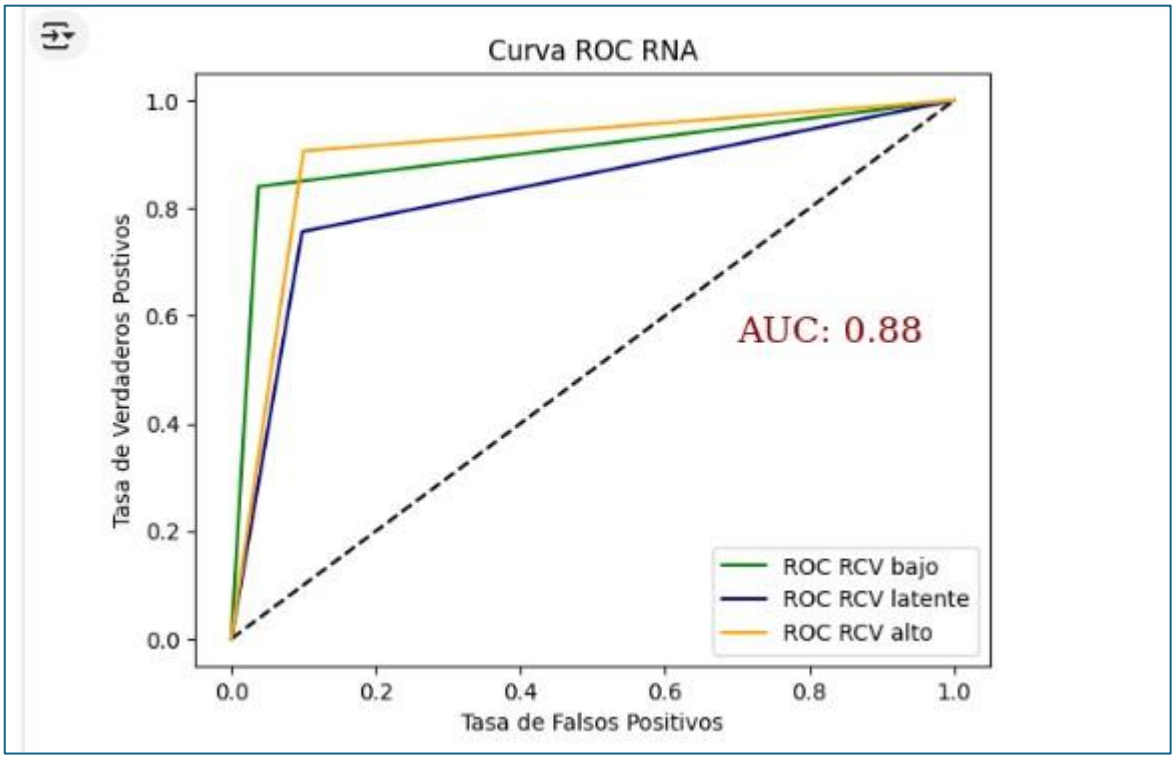
```
[23]  1 # compile the keras (tensorflow) flow graph
      2 modelRNA.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),
      3               loss='binary_crossentropy',
      4               metrics=['accuracy'])
```



Se agrega una nueva capa de oculta de 8 neuronas obteniendo:

```
[42]  1 # Definir la arquitectura del modelo de la RNA
      2 modelRNA = models.Sequential()
      3 modelRNA.add(Dense(8, batch_input_shape=(None, 35), activation='relu')) ## neu
      4 modelRNA.add(Dense(8, activation='relu'))
      5 modelRNA.add(Dense(3, activation='softmax'))
```

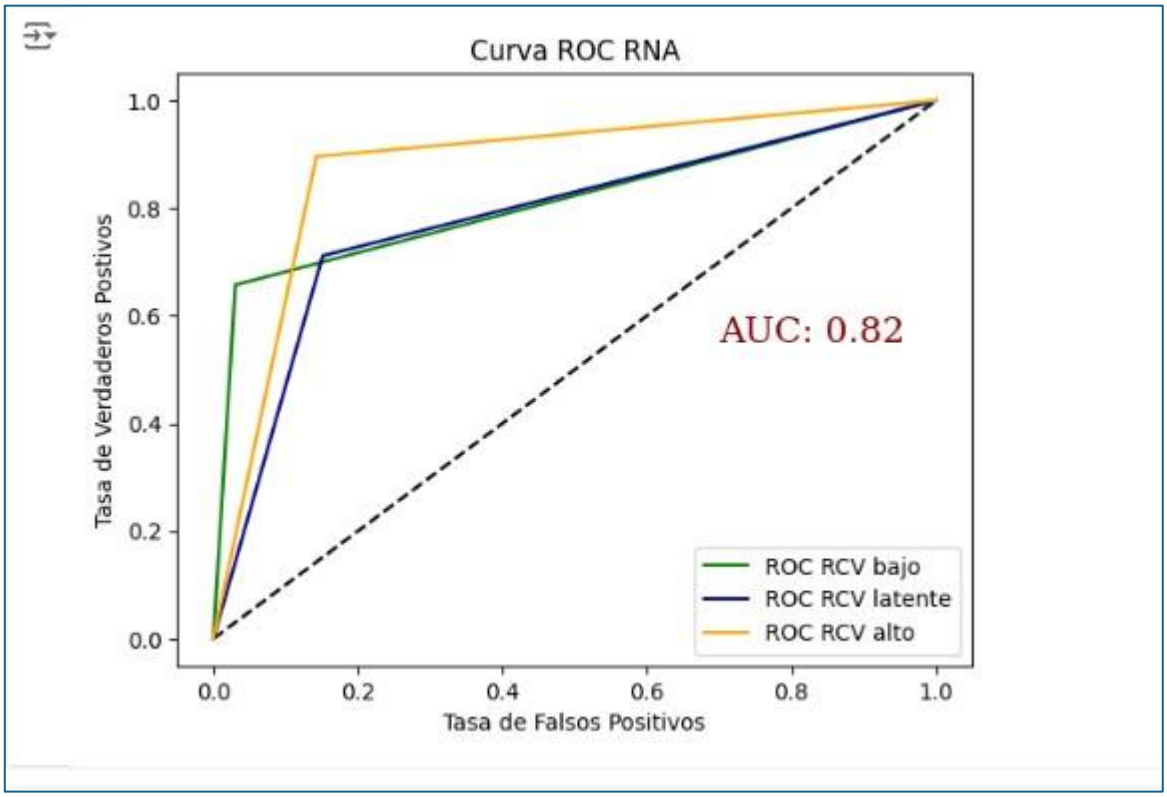
```
[43]  1 # compile the keras (tensorflow) flow graph
      2 modelRNA.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),
      3                  loss='binary_crossentropy',
      4                  metrics=['accuracy'])
```



Finalmente se agrega otra capa oculta de 8 neuronas, obteniendo:

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

```
[43]
# compile the keras (tensorflow) flow graph
modelRNA.compile(optimizer=optimizers.RMSprop(learning_rate=0.001),
                 loss='binary_crossentropy',
                 metrics=['accuracy'])
```


Curva ROC RNA
Tasa de Verdaderos Postivos
Tasa de Falsos Positivos
AUC: 0.82
ROC RCV bajo
ROC RCV latente
ROC RCV alto
0.0
0.2
0.4
0.6
0.8
1.0

Conclusiones

Estos resultados muestran que, aunque hacer la red neuronal más compleja con más neuronas y capas ocultas puede mejorar el rendimiento al principio, hay un punto en el que añadir más complejidad empeora el rendimiento por problemas como el sobreajuste. Es clave encontrar un equilibrio entre la capacidad del modelo para aprender patrones complejos y su capacidad para funcionar bien con nuevos datos. Estos hallazgos subrayan lo importante que es seleccionar y ajustar bien la arquitectura de las redes neuronales para optimizar su rendimiento.