

Integrantes:

Cristhian Balaguera

Daniel Gordillo

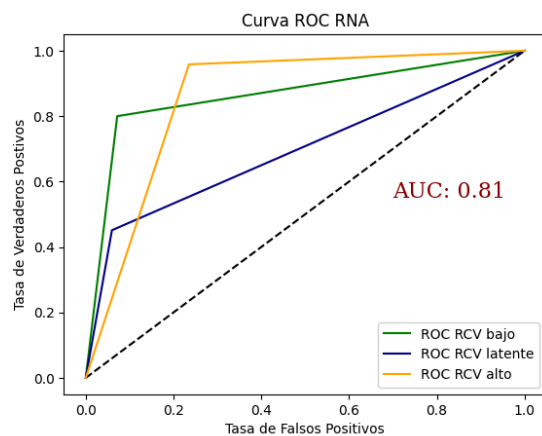
Paula Rios

Oscar Velasquez

Actividad Clasificación RCV

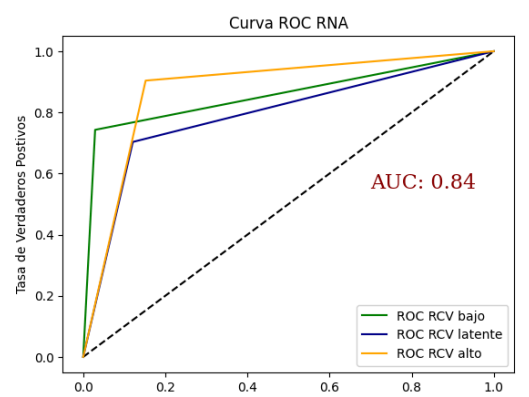
Estado inicial

- **Capa entrada:** 35 neuronas
- **Capa 1 oculta:** 1 neurona
- **Capa salida:** 3 neuronas



Prueba 1

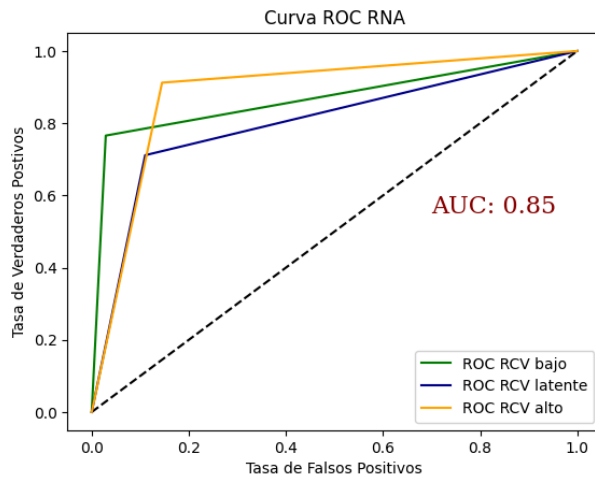
- **Capa entrada:** 35 neuronas
- **Capa 1 oculta:** 5 neurona
- **Capa salida:** 3 neuronas



Prueba 2

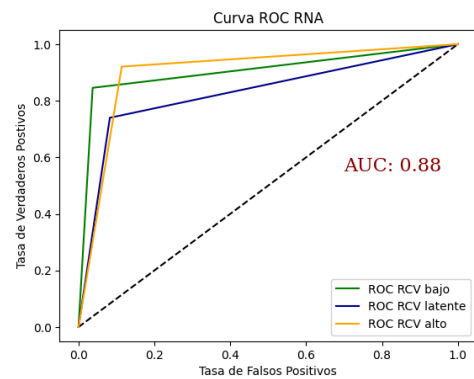
- **Capa entrada:** 35 neuronas
- **Capa 1 oculta:** 10 neuronas
- **Capa 2 oculta:** 10 neuronas

- **Capa salida: 3 neuronas**



Prueba 3

- **Capa de entrada: 35 neuronas**
- **Capa oculta 1: 10 neuronas**
- **Capa oculta 2: 10 neuronas**
- **Capa oculta 3: 10 neuronas**
- **Capa oculta 4: 10 neuronas**
- **Capa oculta 5: 5 neuronas**
- **Capa de salida: 3 neuronas**



```
Epoch 105/200
53/53 [=====] - 0s 4ms/step - loss: 0.2294 - accuracy: 0.9818 - val_loss: 0.2840 - val_accuracy: 0.8619
Epoch 180/200
53/53 [=====] - 0s 3ms/step - loss: 0.2291 - accuracy: 0.9854 - val_loss: 0.2870 - val_accuracy: 0.8690
Epoch 181/200
53/53 [=====] - 0s 3ms/step - loss: 0.2293 - accuracy: 0.9818 - val_loss: 0.2810 - val_accuracy: 0.8667
Epoch 182/200
53/53 [=====] - 0s 3ms/step - loss: 0.2283 - accuracy: 0.9842 - val_loss: 0.2869 - val_accuracy: 0.8643
Epoch 183/200
53/53 [=====] - 0s 3ms/step - loss: 0.2286 - accuracy: 0.9818 - val_loss: 0.2834 - val_accuracy: 0.8595
Epoch 184/200
53/53 [=====] - 0s 4ms/step - loss: 0.2288 - accuracy: 0.9824 - val_loss: 0.2825 - val_accuracy: 0.8619
Epoch 152/200
53/53 [=====] - 0s 4ms/step - loss: 0.2288 - accuracy: 0.9848 - val_loss: 0.2830 - val_accuracy: 0.8595
Epoch 153/200
53/53 [=====] - 0s 3ms/step - loss: 0.2285 - accuracy: 0.9865 - val_loss: 0.2893 - val_accuracy: 0.8690
Epoch 154/200
53/53 [=====] - 0s 3ms/step - loss: 0.2296 - accuracy: 0.9800 - val_loss: 0.2858 - val_accuracy: 0.8667
Epoch 155/200
53/53 [=====] - 0s 3ms/step - loss: 0.2286 - accuracy: 0.9818 - val_loss: 0.2798 - val_accuracy: 0.8667
Epoch 156/200
53/53 [=====] - 0s 3ms/step - loss: 0.2288 - accuracy: 0.9818 - val_loss: 0.2881 - val_accuracy: 0.8571
Epoch 157/200
53/53 [=====] - 0s 4ms/step - loss: 0.2282 - accuracy: 0.9818 - val_loss: 0.2851 - val_accuracy: 0.8571
Epoch 158/200
53/53 [=====] - 0s 3ms/step - loss: 0.2301 - accuracy: 0.9818 - val_loss: 0.2880 - val_accuracy: 0.8571
Epoch 159/200
53/53 [=====] - 0s 3ms/step - loss: 0.2282 - accuracy: 0.9868 - val_loss: 0.2882 - val_accuracy: 0.8548
Epoch 160/200
53/53 [=====] - 0s 3ms/step - loss: 0.2286 - accuracy: 0.9868 - val_loss: 0.2814 - val_accuracy: 0.8571
Epoch 200/200
53/53 [=====] - 0s 3ms/step - loss: 0.2288 - accuracy: 0.8978 - val_loss: 0.3854 - val_accuracy: 0.8571
```

	precision	recall	f1-score	support
0	0.846	0.846	0.846	175
1	0.771	0.740	0.755	246
2	0.902	0.921	0.911	479
accuracy			0.857	900
macro avg	0.840	0.835	0.837	900
weighted avg	0.855	0.857	0.856	900

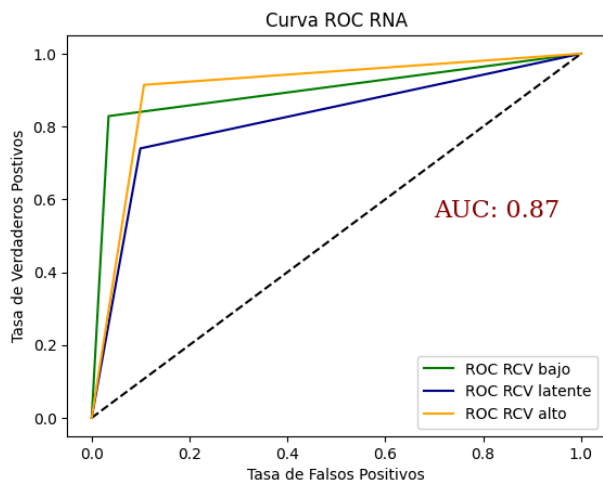
El tiempo de cómputo con RNA fue: 43.84 segundos

```
# 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 capa oculta
modelRNA.add(Dense(10))
modelRNA.add(Dense(10))
modelRNA.add(Dense(10))
modelRNA.add(Dense(5))
modelRNA.add(Dense(3, activation='softmax'))
```

Prueba 4

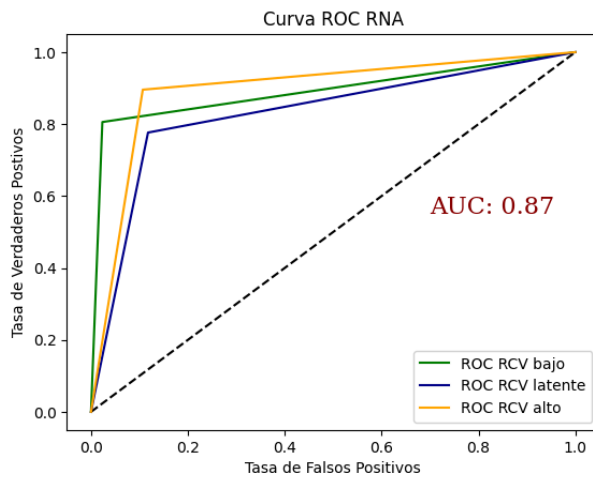
- **Capa entrada: 35 neuronas**
- **Capa 1 oculta: 10 neuronas**
- **Capa 2 oculta: 5 neuronas**

- **Capa salida:3 neuronas**



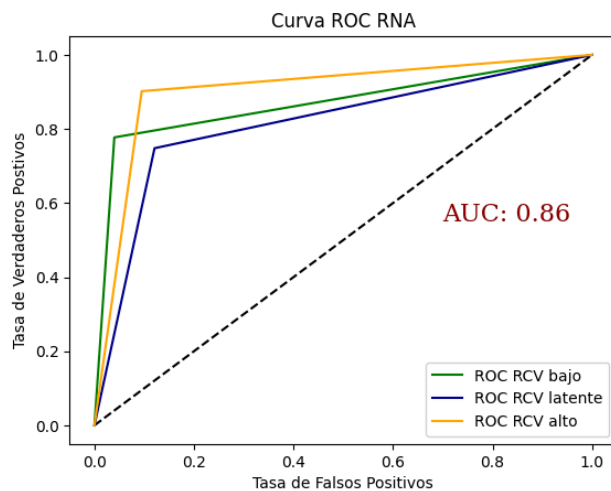
Prueba 5

- **Capa entrada: 35 neuronas**
- **Capa 1 oculta: 7 neuronas**
- **Capa 2 oculta: 7 neuronas**
- **Capa 3 oculta: 7 neuronas**
- **Capa salida:3 neuronas**



Prueba 6

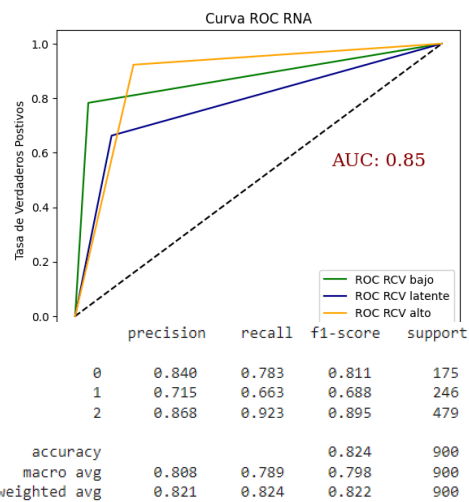
- **Capa entrada: 35 neuronas**
- **Capa 1 oculta: 10 neuronas**
- **Capa 2 oculta: 10 neuronas**
- **Capa 3 oculta: 15 neuronas**
- **Capa salida:3 neuronas**



Prueba 7

- **Capa de entrada:** 35 neuronas
- **Capa oculta 1:** 10 neuronas
- **Capa oculta 2:** 10 neuronas
- **Capa oculta 3:** 50 neuronas
- **Capa de salida:** 3 neuronas

```
Epoch 187/200
53/53 [-----] - 0s 6ms/step - loss: 0.1824 - accuracy: 0.9119 - val_loss: 0.3795 - val_accuracy: 0.8198
Epoch 188/200
53/53 [-----] - 0s 5ms/step - loss: 0.1828 - accuracy: 0.9095 - val_loss: 0.3798 - val_accuracy: 0.8214
Epoch 189/200
53/53 [-----] - 0s 5ms/step - loss: 0.1828 - accuracy: 0.9143 - val_loss: 0.3795 - val_accuracy: 0.8198
Epoch 190/200
53/53 [-----] - 0s 5ms/step - loss: 0.1821 - accuracy: 0.9089 - val_loss: 0.3799 - val_accuracy: 0.8167
Epoch 191/200
53/53 [-----] - 0s 5ms/step - loss: 0.1809 - accuracy: 0.9125 - val_loss: 0.3856 - val_accuracy: 0.8119
Epoch 192/200
53/53 [-----] - 0s 4ms/step - loss: 0.1823 - accuracy: 0.9113 - val_loss: 0.3753 - val_accuracy: 0.8198
Epoch 193/200
53/53 [-----] - 0s 5ms/step - loss: 0.1816 - accuracy: 0.9131 - val_loss: 0.3718 - val_accuracy: 0.8333
Epoch 194/200
53/53 [-----] - 0s 5ms/step - loss: 0.1814 - accuracy: 0.9089 - val_loss: 0.3761 - val_accuracy: 0.8238
Epoch 195/200
53/53 [-----] - 0s 5ms/step - loss: 0.1804 - accuracy: 0.9113 - val_loss: 0.3720 - val_accuracy: 0.8262
Epoch 196/200
53/53 [-----] - 0s 5ms/step - loss: 0.1814 - accuracy: 0.9143 - val_loss: 0.3757 - val_accuracy: 0.8286
Epoch 197/200
53/53 [-----] - 0s 5ms/step - loss: 0.1808 - accuracy: 0.9137 - val_loss: 0.3821 - val_accuracy: 0.8143
Epoch 198/200
53/53 [-----] - 0s 5ms/step - loss: 0.1804 - accuracy: 0.9143 - val_loss: 0.3788 - val_accuracy: 0.8198
Epoch 199/200
53/53 [-----] - 0s 6ms/step - loss: 0.1810 - accuracy: 0.9089 - val_loss: 0.3748 - val_accuracy: 0.8167
Epoch 200/200
53/53 [-----] - 0s 3ms/step - loss: 0.1803 - accuracy: 0.9161 - val_loss: 0.3786 - val_accuracy: 0.8143
```



El tiempo de cómputo con RNA fue: 40.89 segundos

```
# Definir la arquitectura del modelo de la RNA
modelRNA = models.Sequential()
modelRNA.add(Dense(10, 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(10))
modelRNA.add(Dense(50))
modelRNA.add(Dense(3, activation='softmax'))
```

Conclusiones

A pesar de que solo se evidencian 7 test realizados en el documento, extraoficialmente se realizaron más test con el fin de encontrar una tendencia dentro de las redes neuronales para llegar al punto más óptimo posible. De las varias pruebas que se realizaron se destacan las siguientes conclusiones:

- El número de neuronas se puede asimilar entre 5 a 10 neuronas por capa oculta.
- Si se coloca otro número de neuronas ya sean menores a 5, entre 5 y 10 o después de 10 el resultado suele empeorar.
- La anterior condición sólo exceptúa al número 50 ya que con este también suele mejorar los resultados.
- Para obtener una conclusión mas rapida en cuanto al posible resultado de la estructura neuronal configurada, se puede destacar la condición de si durante el proceso de entrenamiento del modelo, al momento de acercarse al Epoch=100, el valor de "accuracy" supera el 0.8600 el modelo puede ser óptimo y tener resultados del "AUC" entre 0.87 a 0.89.
- El último modelo creado, representa la configuración más óptima encontrada durante el proceso de pruebas.
- Posiblemente el resultado del test #10 sea una mera casualidad de optimización. Esto se concluye ya que luego de que inoportunamente se fuera la luz y apagara el equipo de pruebas, se volvió a ejecutar la misma configuración y esta no superó más del valor de AUC=0.88.
- Al realizar entrenamientos con más de 5 capas ocultas, estas empiezan a tener un rendimiento lineal no mayor a AUC=0.87.
- El aumento de la cantidad de capas no siempre implica que el valor de AUC se acerque más a 1.
- El aumento excesivo de neuronas disminuye el valor del AUC.
- El valor de AUC tiende a ser más alto manteniendo la cantidad de neuronas menor a 15.