

Tarea 4

Martín Valderrama Illesca

MA7260 - Aprendizaje Profundo en Diagnóstico y Pronóstico de Fallas,
Primavera 2020

1. Introducción

En esta tarea se realiza una detección de anomalías en datos de Stripping, donde 10 sensores entregan datos cada 20 minutos por una ventana de tiempo de 6 horas. Se pretende entrenar un modelo de autoencoder para detectar anomalías en el proceso.

El autoencoder es un modelo que reconstruye la señal a partir de la misma señal. La señal pasa a través de un cuello de botella para luego ser reconstruida a la salida. La detección se realizará comparando el error de reconstrucción de la señal obtenida a la salida. Se compara dicho error con un umbral para discriminar la fase de operación nominal con la de operación anómala.

A continuación se detalla la metodología seguida, los resultados obtenidos y un análisis de los mismos. Se finaliza con una conclusión.

2. Metodología

Para entrenar el modelo se cargan los datos proporcionados por el cuerpo docente, donde se encuentran los conjuntos de entrenamiento, validación y test. Estos datos ya vienen preprocesados, es decir, escalados de tal manera que los sensores entregan datos comparables entre sí para alimentar el modelo.

Estos datos son entregados a la red para su entrenamiento, donde la red compara la serie de tiempo original con la reconstruida y se optimiza reduciendo el error cuadrático medio (MSE) entre las señales. El conjunto de validación es utilizado para obtener métricas de mse durante el entrenamiento.

Una vez entrenado el modelo, se pasa el conjunto de test por el modelo y se obtienen las reconstrucciones respectivas. Se calcula el rmse (raíz cuadrada del mse) al que se le aplica un umbral (th) con el que se determina si la serie corresponde a una operación anómala o nominal.

Para visualizar los resultados se utilizan gráficos del rmse, matriz de confusión y curva ROC.

3. Arquitectura

La arquitectura de la red utilizada se detalla a continuación. Esta fue inspirada a partir del workshop de autoencoders del curso. Se utilizan capas densas junto con dropout.

```
encoder = Dense(units=1800, activation='relu')(input_layer)
encoder = Dense(units=1800, activation='relu')(encoder)
encoder = Dropout(rate=0.1)(encoder)
encoder = Dense(units=1000, activation='relu')(encoder)
encoder = Dense(units=1000, activation='relu')(encoder)
encoder = Dropout(rate=0.1)(encoder)
encoder = Dense(units=700, activation='relu')(encoder)
encoder = Dense(units=700, activation='relu')(encoder)

latent_dim = 500
latent_vector = Dense(units=latent_dim, activation='relu')(encoder)

decoder = Dense(units=700, activation='relu')(latent_vector)
```

```

decoder = Dense(units=700, activation='relu')(decoder)
decoder = Dropout(rate=0.1)(decoder)
decoder = Dense(units=1000, activation='relu')(decoder)
decoder = Dense(units=1000, activation='relu')(decoder)
decoder = Dropout(rate=0.1)(decoder)
decoder = Dense(units=1800, activation='relu')(decoder)
decoder = Dense(units=1800, activation='relu')(decoder)

```

La entrada y la salida del modelo corresponden a datos de dimensión de 18×10 , la que corresponde a la concatenación de los 10 sensores en un vector de una sola dimensión.

4. Resultados

El resultado del entrenamiento se puede ver a continuación en la figura 1.

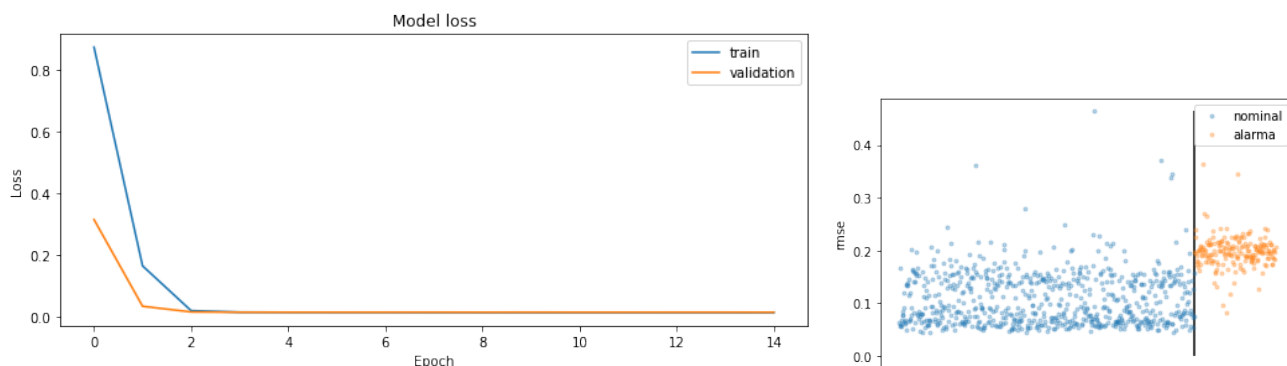


Figura 1: Fase de entrenamiento por 15 épocas del autoencoder: evolución de mse sobre conjunto de entrenamiento y test. A la derecha se observa la separación por error de reconstrucción obtenida.

Se observa que el rmse decrece con el tiempo, y que al pasar el conjunto de test por la red, se obtiene una separación más o menos evidente en cuanto a su valor para operación nominal (azul) y en anomalía (naranja). Para visualizar de mejor manera la separación que se busca, se presenta la figura 2 donde el umbral es el que se busca optimizar. Dicho umbral separa las operaciones anómalas de las nominales según su error de reconstrucción.

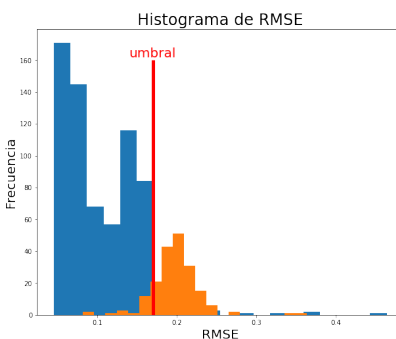


Figura 2: Histograma de RMSE donde se busca el umbral que separa operación anómala de las nominales.

4.1. ROC

Para encontrar los umbrales idóneos para la separación, se realiza una exploración de los umbrales y su rendimiento según la curva ROC. La curva ROC permite ver el desempeño de los reales positivos frente a los falsos positivos.

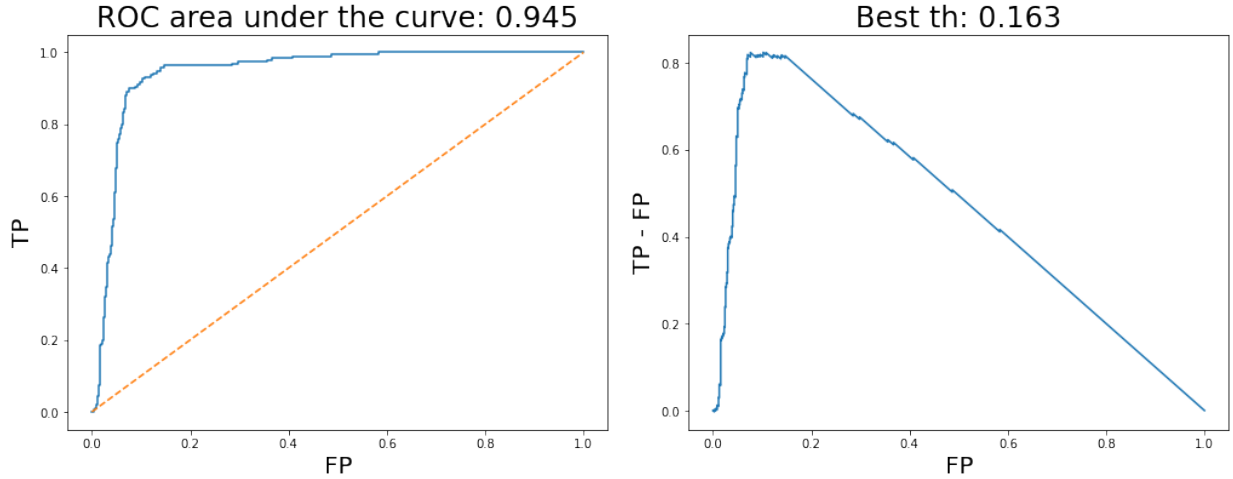


Figura 3: Curva ROC resultante en la exploración del umbral a seleccionar. A la derecha se muestra la curva TP - FP en función de FP para ilustrar los puntos en que se quiere maximizar la obtención de datos reales positivos, frente a la menor cantidad de falsos positivos.

En la figura 3 se ilustra la ROC obtenida, reales positivos (TP) frente a falsos positivos (FP). Esta curva es utilizada para determinar el mejor umbral que logra comparar los beneficios de un modelo, donde el umbral se determina en función de cuantos TP quiero frente a los FP que estoy dispuesto a aceptar. O, en otras palabras, cuanto estoy dispuesto a aceptar que el modelo se equivoque sabiendo detectar de buena forma los reales positivos, o aciertos.

4.2. Selección de Umbral

Para seleccionar el umbral se puede tomar la curva ROC obteniendo la separación de la figura 4 izquierda. O se puede optar por un criterio sobre el conjunto de test, como se muestra en la derecha; donde se desea obtener un total de falsos negativos (decir operación nominal cuando en realidad se está en operación anómala) determinado.

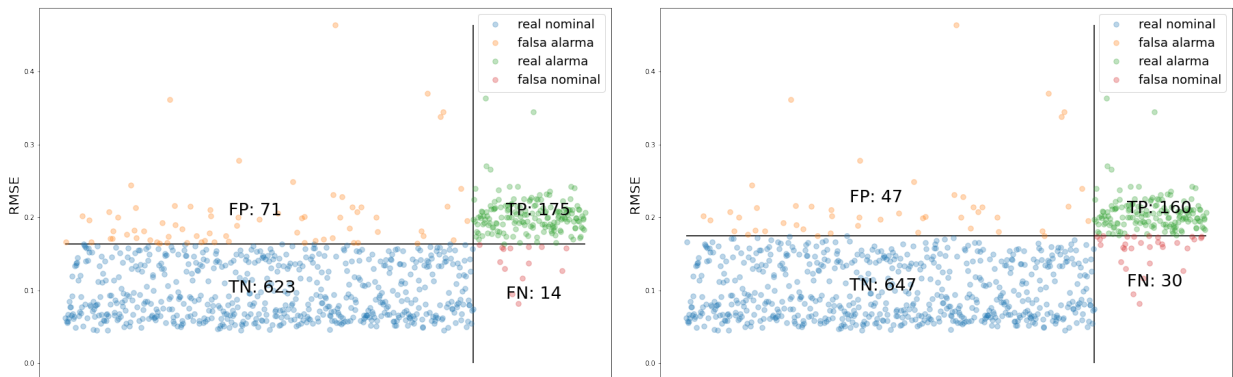


Figura 4: Separación obtenida desde el umbral de la mejor operación que optimiza la cantidad de TP frente a FP.

Las matrices de confusión obtenidas son presentadas en la figura 5.

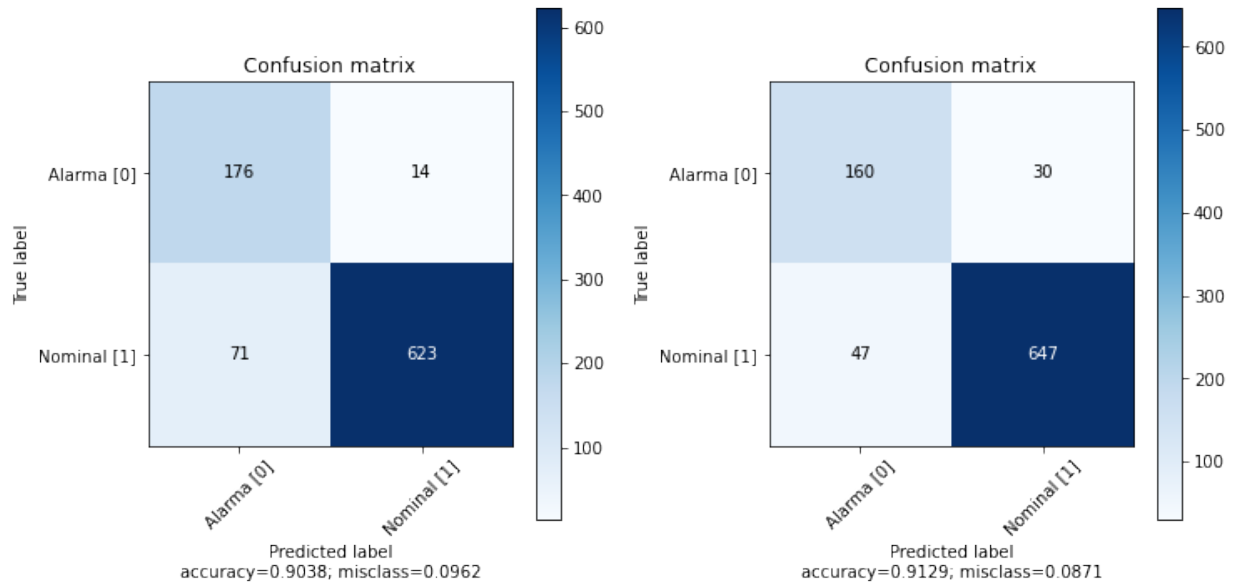


Figura 5: Matrices de confusión

Otra forma de ver el problema es la obtención de la menor cantidad de errores, es decir, la suma entre falsas alarmas y falsos nominales. La cual entrena una figura 6

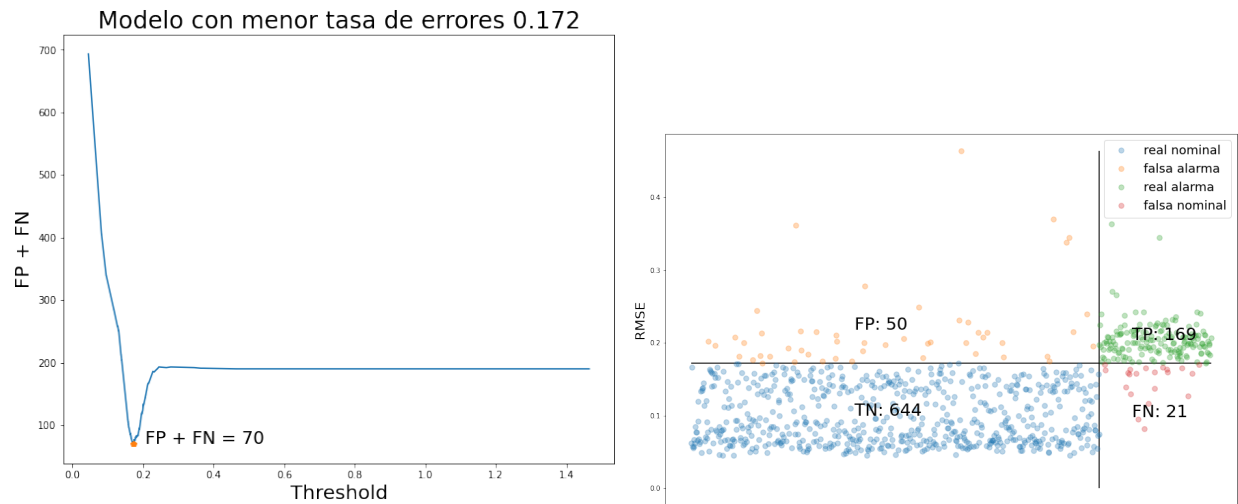


Figura 6: Separación lograda de la búsqueda de la menor cantidad de errores. A la izquierda se encuentra el umbral para la menor cantidad de errores en el conjunto de test, a la derecha se presenta la separación obtenida.

Su matriz de confusión está en la figura 7.

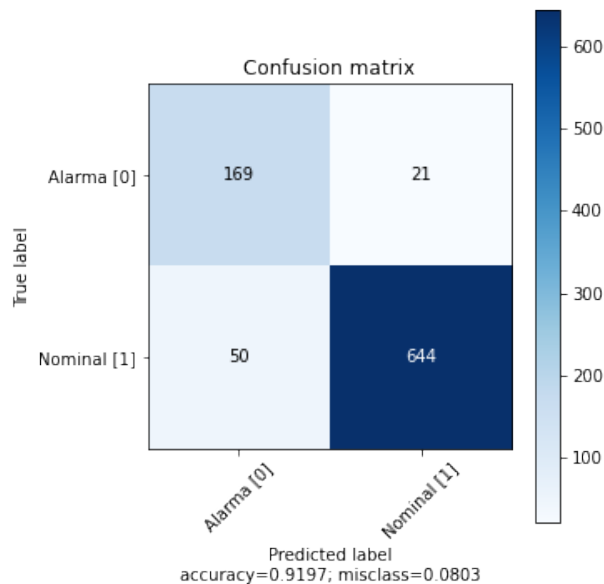


Figura 7: Matriz de confusión para umbral que obtiene menor cantidad de errores entre FP y FN.

5. Conclusión

De esta tarea se logra encontrar un modelo de autoencoder capaz de reconstruir una señal para un modo de operación nominal, y a partir de las reconstrucciones obtenidas de datos anómalos, detectar por medio de la comparación del rmse con un umbral, determinar el estado de anomalía de la serie de tiempo.

El modelo utilizado tiene un gran tamaño en cuanto a capas neuronales y a cantidad de neuronas. Esto funciona bien porque el modelo logra ajustarse de forma muy precisa a los datos nominales, mientras que los datos de falla no tienen una buena representación (no los ha visto la red), por lo que obtendrán un error de reconstrucción mayor.

En este trabajo también se utilizan las herramientas de ROC que permiten encontrar los puntos de umbral óptimos según el comportamiento que se quiera del modelo. Donde se puede obtener un modelo sensible y con un bajo umbral para detectar la mayor cantidad de anomalías posible, pero que también clasifica muchas veces estados normales de operación como anómalos. O tener un modelo poco sensible a las anomalías que, sin embargo, pase estados anómalos por nominales y ponga en riesgo la operación.

La utilidad de la ROC es más económica y de carácter de gestión de los activos.

En cuanto a la creación y entrenamiento de autoencoders, como se dijo anteriormente, se obtiene un autoencoder con un comportamiento satisfactorio que es capaz de entregar buenas reconstrucciones para señales nominales, mientras que las anómalas tiene un desempeño pobre, que es lo esperado.