



Inteligencia Artificial
Ingeniería en Sistemas de Información
Año 2021

Trabajo Práctico N° 2

Entrega Final

GRUPO N° 13

Apellido y Nombres	Dirección de E-Mail
Romano, Leandro	leandroromano@frba.utn.edu.ar
Moncarz, Fernando Nicolás	nmoncarz@frba.utn.edu.ar
Flighelman, Germán	gflighelman@frba.utn.edu.ar
Delaguardia, Gonzalo	gdelaguardia@frba.utn.edu.ar
Orlando, Leonardo Jorge	lorlando@frba.utn.edu.ar
Conde, Nicolás Ariel	nconde@frba.utn.edu.ar

Fecha de Presentación	06/06/2021
-----------------------	------------

Universidad Tecnológica Nacional
Facultad Regional Buenos Aires

TP 2

Leandro Romano,
Fernando Nicolás Moncarz,
Germán Flighelman,
Delaguardia Gonzalo,
Leonardo Jorge Orlando,
Nicolás Ariel Conde

Universidad Tecnológica Nacional. Facultad Regional Buenos Aires.
Buenos Aires. Argentina

Resumen. El presente trabajo presenta la implementación de una red neuronal artificial, perteneciente a los sistemas inteligentes, para poder distinguir una mina terrestre de una roca

Palabras Claves: Roca, Mina, Red Neuronal Artificial, Inteligencia artificial

1. Introducción

El problema consiste en distinguir una roca común de un cilindro metálico, similar a una mina terrestre, a través de datos generados por sensores especializados.

El desarrollo de un sistema inteligente que permita distinguir entre los objetos puede ser de utilidad en el campo humanitario y bélico, además de profundizar en el campo académico de los sistemas inteligentes.

2. Elementos del Trabajo y Metodología

Modelo

Se utilizará una Red Neuronal Artificial, del tipo Backpropagation. La topología de la misma incluirá una capa de entrada con 60 neuronas, 2 capas ocultas, y la capa de salida tendrá solo una neurona, ya que el resultado es binario. Una de las dos capas ocultas se comunica con todos los nodos de entrada a la red, y la otra capa previene el sobreentrenamiento, también conocido como overfitting. Para entrenar al modelo, y contar con una variedad de métricas, el modelo realizó 1000 epochs o corridas.

Inicialmente, la primera capa oculta consta de 128 neuronas con una función de activación *relu* - función de activación lineal rectificada - y penalidades de regularización de tipo gaussianas, de tipo *l2*.

Utilizamos entre la primera capa oculta y la de salida, una capa de tipo dropout, para evitar el sobreentrenamiento, permitiendo de esta forma, que otros pesos puedan tomar protagonismo.

Dataset

El dataset consiste de patrones obtenidos al rebotar una señal de sonar contra un objeto, en este caso, una roca o un cilindro, desde una variedad de ángulos y condiciones ambientales, desde una distancia de aproximadamente

10 metros. La señal del sonar utilizada se varía ascendentemente en frecuencia para completar el conjunto de datos.

Tenemos entonces por cada objeto, una tupla de 60 valores, todos entre 0 y 1. Además, está identificado cada elemento para poder clasificarlos: en caso de ser una roca, se lo identifica con la letra 'R', caso contrario, se lo identifica con la letra 'M'.

Si bien los datos están distribuidos de manera normal, los mismos no se encuentran estandarizados, es decir, no son una distribución normal centrada en 0 con desvío estándar 1. Para mejorar el desarrollo de nuestro sistema inteligente, se realizó la mencionada estandarización.

Sobre el total del conjunto de datos, se decidió utilizar la cuarta parte del mismo para validar el sistema y las tres cuartas partes restantes, son empleadas para el entrenamiento.

Tecnologías

Se optó por implementar el sistema inteligente utilizando el lenguaje de programación Python, dado que el mismo cuenta con una gran colección de librerías especializadas para la tarea, entre las cuales, se utilizan para el desarrollo, las librerías **TensorFlow**, **Keras** y **Sklearn**, que facilitan la creación de modelos de sistemas inteligentes y generar conjuntos de entrenamiento y validación; **Numpy**, que facilita el manejo de arreglos en el lenguaje; y **Pandas**, que simplifica el manejo de *Data Frames*, utilizados para manejar el dataset sobre el cual se desarrolla el sistema inteligente.

3. Resultados

Primer Modelo

- Topología
 - Una capa de entrada de 60 neuronas
 - Una capa intermedia u oculta de 128 neuronas con función de activación es *relu*
 - Una capa intermedia dropout, que descarta el 10% de los inputs
 - Una capa de salida de 1 neurona con función de activación es *sigmoidal*
 - Con un coeficiente de entrenamiento $\alpha = 0.01$

Matriz de Confusión

	Si	No
Si	29.336	1.664
No	3.488	17.512

- Métricas
 - Exactitud: 0.9009230737686157
 - Precisión: 0.8756753513216973
 - Recuperación: 0.9614212039709091
 - Perdida: 0.5420284813195467

Segundo Modelo

- Topología
 - Una capa de entrada de 60 neuronas
 - Una capa intermedia u oculta de 128 neuronas con función de activación es *relu*
 - Una capa intermedia dropout que descarta el 10% de los inputs
 - Una capa intermedia u oculta de 64 neuronas con función de activación es *relu*
 - Una capa intermedia dropout que descarta el 10% de los inputs
 - Una capa de salida de 1 neurona con función de activación es *sigmoidal*
 - Con un coeficiente de entrenamiento $\alpha = 0.001$

Matriz de Confusión

	Si	No
Si	29.845	1.155
No	3.603	17.397

- Métricas
 - Exactitud: 0.9084999949336052
 - Precisión: 0.8703776785731315
 - Recuperación: 0.974769027709961
 - Pérdida: 0.42525842021405696

4. Discusión

Este sistema inteligente resuelve el problema de forma efectiva con ambos modelos planteados y se puede notar esto gracias a las métricas:

- La **exactitud** en ambos modelos mayor al 90%, lo que indica que el modelo parece bueno
- Si se observa la **precisión**, podemos decir que ambos modelos cumplen satisfactoriamente, ya que obtuvieron un valor mayor al 87%
- Buena **recuperación**, ya que ambos se recuperan en el 97% de los casos correctamente
- La **pérdida** en ambos modelos es menor al 54%, valor aceptable viendo el resto de las métricas

5. Conclusión

Luego de ajustar la topología, agregando dos subcapas y disminuyendo el coeficiente de entrenamiento, pudimos notar una mejora en la pérdida, donde el valor medio cayó 10 puntos. Esto hace que el segundo modelo sea más efectivo. Pero el resto de las métricas no otorgaron beneficios lo suficientemente importantes como para justificar el cambio de topología planteado, y no consideramos que esto haya sido rentable por la suma de complejidad que se le adiciona al modelo y el costo agregado a la hora del entrenamiento del mismo.

Se pueden proponer diversas mejoras para el análisis realizado. Por ejemplo, se podría haber tenido en cuenta la métrica **F1 Score**, para combinar las métricas analizadas y definir objetivamente la exactitud real de los modelos. Además, se podrían haber realizado *benchmarks*, para medir el tiempo de entrenamiento y el consumo de recursos de ambos modelos, para poder analizar con mayor certeza la eficiencia.

6. Referencias bibliográficas

- [Sonar Mines vs Rocks dataset](#)
- [Keras](#)
- [Dropout layer by Keras](#)
- [Pandas](#)
- [Pandas read_csv](#)
- [Introducción a Dropout](#)