1. Redes neuronales
   1. Definición
   2. Resumen histórico
   3. Funcionamiento de una red neuronal
      1. Estructura
      2. Función activadora
   4. Clasificación de las redes neuronales
   5. Aplicación de las redes neuronales a vehículos autónomos
2. Unity
   1. Unity como entorno de simulación
3. Implementación de una red neuronal a la simulación del recorrido de un vehículo autónomo
4. Bibliografía
5. **Redes neuronales**
   1. **Definición**

Para definir una red neuronal podemos remontarnos a Rubiolo, quien a su vez cita a Haykin, quien define una Red Neuronal Artificial (RNA) como un procesador en paralelo y distribuido que tiene la capacidad de almacenar conocimiento basado en experiencia. Las RNAs presentan similitudes con el cerebro humano en cuanto a que el conocimiento es adquirido mediante procesos de aprendizaje y las conexiones neuronales son utilizadas para almacenar este conocimiento. Cuentan con características tales como: aprender mediante experiencias, adaptabilidad, capacidad de generalización, reconocimiento de patrones de ruido, pragmatismo y capacidad de operar problemas con múltiples variables (Rubiolo, 2014).

De manera similar, Mercado et al. definen una red neuronal como un conjunto de elementos simples de procesamiento llamados nodos o neuronas, conectadas entre sí por conexiones que tienen un valor numérico modificable, su funcionamiento de forma general consiste normalmente en sumar valores de entrada proporcionados por otras unidades, comparar esta cantidad con un umbral y si éste es superado, enviar una señal de activación o salida hacia las otras unidades del sistema. Asimismo, se puede hacer una analogía entre los conceptos de neurona, entrada, salida y peso de las neuronas de una RNA con los conceptos de soma, dendrita, axón y sinapsis de una neurona biológica (Mercado et al., 2015).

Montaño compara la estructura conexionista de una RNA con la arquitectura tradicional Von Neumann que constituye la estructura de procesamiento de la mayoría de los ordenadores: Mientras que un procesador con estructura Von Neumann realiza tareas de manera secuencial una tras otra una serie de expresiones binarias, en una red neuronal las unidades de procesamiento se ejecutan simultáneamente. Otra diferencia notable es que en la arquitectura Von Neumann la memoria se almacena de forma discreta en ubicaciones físicas independientes y definidas. Mientras que en las redes neuronales se almacenan a lo largo de los parámetros del sistema (Montaño, 2002).

Entre las ventajas de una estructura de procesamiento con redes neuronales se encuentra su naturaleza descentralizada que ofrece resistencia ante un funcionamiento defectuoso o la pérdida parcial de la estructura de procesamiento debido su estructura distribuida no centralizada (Montaño, 2002).

* 1. **Resumen histórico**

Los orígenes de las redes neuronales se remontan a mitades del siglo XX: en 1943 McCulloch y Pitts proponen un modelo de red neuronal, en 1949 se propone la regla Hebbiana de aprendizaje, y en 1958 Frank Rosemblatt introduce las redes neuronales de una sola capa denominadas perceptrones, en años posteriores la disciplina es abandonada considerablemente debido a las limitaciones de los perceptrones demostradas por Minsky y Papera en 1969. En los años 80 hay un resurgimiento de los modelos con estructuras de perceptrones multicapa y de redes radiales en la década de 1990 (Gómez, 2006).

* 1. **Funcionamiento de una red neuronal**
     1. **Estructura de una red neuronal**

Habiendo entendido la red neuronal como un procesador distribuido de unidades de cómputo independientes denominadas neuronas, es posible definir a las neuronas como estructuras constituidas usualmente por cuatro elementos funcionales y que realizan funciones computacionales simples, Pérez describe esta estructura usual de una neurona de la siguiente manera:

* **Elemento receptor**, a donde llegan las señales de entrada provenientes de otras neuronas, llegando ponderas por sus respectivos pesos que indican la intensidad de la conexión entre la neurona fuente y la neurona receptiva en cuestión.
* **Elemento sumador**, realiza la suma algebraica de las señales de entrada ponderadas por peso.
* **Función activadora**, es una función no lineal de umbral que es utilizada para determinar si la neurona es activada y dispare una salida.
* **Elemento de salida**, es la señal producida dependiendo del resultado de la función activadora.

El funcionamiento de una RNA con perceptrón multicapa con capas (incluyendo capa de entrada y salida), con cada neurona de la capa recibiendo información de las neuronas de la capa anterior , la neurona la procesa y envía la señal de salida a cada una de las neuronas de la capa está dado por la expresión:

Donde es el número de neuronas de la capa, es el peso de la conexión entra la neurona de la capa y la neurona de la capa , es el umbral de la neurona de la capa . es la función de activación de la neurona de la capa (Fernandez, 2007).

* + 1. **Función activadora**

Las funciones activadoras más usadas en RNAs son la función escalón y la curva logística (Pérez, 2019).

* Función escalón:
* Logística sigmoide: .

Dinamarca ofrece una lista más amplía de funciones activadoras que incluyen (Dinamarca, 2018):

* Tangente hiperbólica
* Softmax o exponencial normalizada
* Unidad lineal rectificada
* Leaky ReLU
* Unidad lineal rectificada paramétrica
* Thresholded ReLU.
  1. **Clasificación de las redes neuronales**

Los criterios para la clasificación de las redes neuronales más importantes son (Gómez, 2006):

* Por su topología:
  + Capa simple (single layer)
  + Multicapa (multilayer)
  + Recurrente
  + Auto organizada
* Por su aplicación:
  + Clasificación
  + Clustering
  + Aproximación de funciones
  + Predicción
* Tipo de conexión:
  + Estática
  + Dinámica
* Método de aprendizaje:
  + Supervisada
  + No supervisada
  1. **Aplicación de las redes neuronales a vehículos autónomos**

Park et al. presentan una propuesta para la implementación de redes neuronales convolucionales en vehículos autónomos equipados con una cámara y dos sensores LiDAR al frente del vehículo (Park et al., 2021). Tian et al. comparan el uso de redes neuronales recurrentes con redes neuronales convolucionales para el mismo fin.

1. **Unity**

Unity es un motor de juegos utilizado mayormente para el desarrollo de videojuegos, provee las herramientas necesarias para la integración de elementos gráficos, y auditivos, además de permitir la integración del equipo de trabajo y programación de los entornos lógicos, mecánicos y de físicas de juegos (Nieminen, 2021). Permite importar arte y recursos en 2D O 3D provenientes de otro software y ensamblarlos en escenas o ambientes; permite añadir luz, audio, efectos especiales, física y animación, interactividad y lógica de juego; y editar, depurar y optimizar el contenido de sus plataformas de destino.

* 1. **Unity como entorno de simulación**

Unity además permite la simulación de procesos más generales como líneas de producción (Reinikka, 2019) por lo que destaca su uso en simulación de entornos físicos similares al abordado en este trabajo.

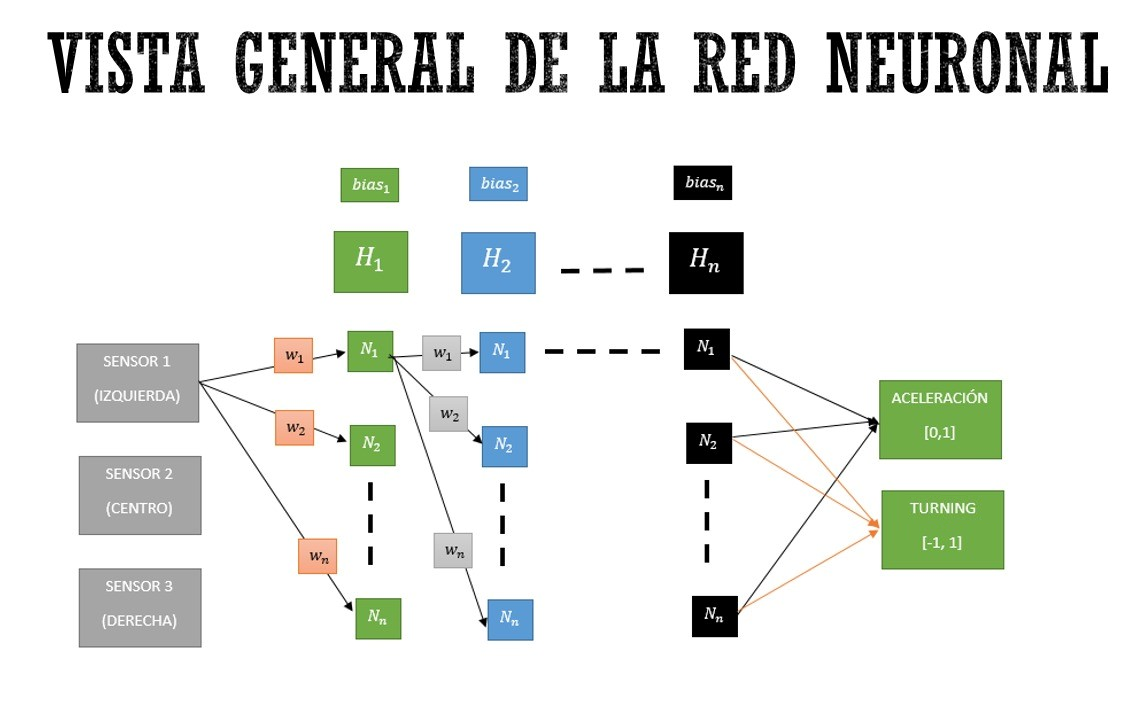
1. **Implementación de una red neuronal a la simulación del recorrido de un vehículo autónomo**

Para la realización del proyecto se utilizó la plataforma Unity. Unity es un motor de juegos, este software proporciona a los creadores de juegos el conjunto necesario de características para construir juegos de forma rápida y eficiente.

Se creó un entornó virtual en Unity simulando una carretera a través de la cual debe transitar un vehículo automotor.

Para la implementación de la red neuronal, las variables de entrada corresponden a los datos obtenidos por los sensores que el carro integra en las principales direcciones: izquierda (), derecha () y centro (). Los valores de salida son la aceleración con valores entre y turning (valor de giro) con valores en que describen la trayectoria a seguir a continuación por el vehículo.

Si se considera una capa oculta dos neuronas y con pesos y respectivamente y el sesgo, que no se ve afectado por las variables de entrada, se tiene:



1. **Bibliografía**

* Dinamarca, A (2018). Aprendizaje y Análisis de Redes Neuronales Artificiales Profundas, tesis para la obtención del grado como Licenciada en Ciencias Básicas con Orientación en Física. Universidad Nacional de Cuyo.
* Fernandez, D (2007). Desarrollo de una nueva herramienta basada en redes neuronales para el diseño de protecciones ligeras cerámica-metal frente al impacto de alta velocidad, tesis doctoral. Universidad Carlos III de Madrid.
* Gómez, M (2006). Redes neuronales aplicadas a los negocios: Análisis de un caso, tesis para obter el grado en ingeniera industrial. Instituto tecnológico de Buenos Aires.
* Mercado D., Pedraza, L., Martínez, E. (2015). Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo. PROSPECTIVA, vol. 13, núm. 2.
* Montaño, J (2002). Redes Neuronales Artificiales aplicadas al Análisis de Datos, tesis doctoral. Universitat De Les Illes Balears.
* Nieminen, T. (2021) Unity game engine in visualization, simulation and modelling, tesis de grado en Programme in Computer Sciences. Tampere University.
* Park, M., Kim, H., Park, S. (2021). A Convolutional Neural Network-Based End-to-End Self-Driving Using LiDAR and Camera Fusion: Analysis Perspectives in a Real-World Environment, Electronics 2021, 10, 2608.
* Pérez, G (2019). Esquema de control basado en una red neuronal artificial de bajo orden, tesis para obtener el grado de maestro en Ciencias en Ingeniería Electrónica. Tecnológico Nacional de México.
* Reinikka, J. (2019) Production line simulation made with Unity and controlled by TwinCAT, tesis para obtener el grado en Information Technology. Seinäjoki University of Applied Sciences.
* Rubiolo M, (2014). Desarrollo de nuevos modelos y algoritmos basados en redes neuronales para tareas de minería de datos, tesis para obtener el grado de doctor en ingeniería. Universidad Tecnológica Nacional.
* Tian, Y., Jana, S., Pei, K., Ray, B. (2018). DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars. International Conference on Software Engineering ’18, May 27-June 3, 2018, Gothenburg, Sweden.