SISTEMAS INTELIGENTES I

**Práctica 1: Análisis y Diseño de Arquitecturas Neuronales Supervisadas para la Clasificación de Patrones (Backpropagation)**

**2020/21**

26 de diciembre de 2020

**José Amusquívar Poppe | Prashant Jeswani Tejwani**

Universidad de Las Palmas de Gran Canaria

Escuela de Ingeniería en Informática

Índice

[Estudio y análisis de la arquitectura BPNN 3](#_Toc59734423)

[Proceso de aprendizaje de la BPNN 4](#_Toc59734424)

[Análisis del conjunto de datos 5](#_Toc59734425)

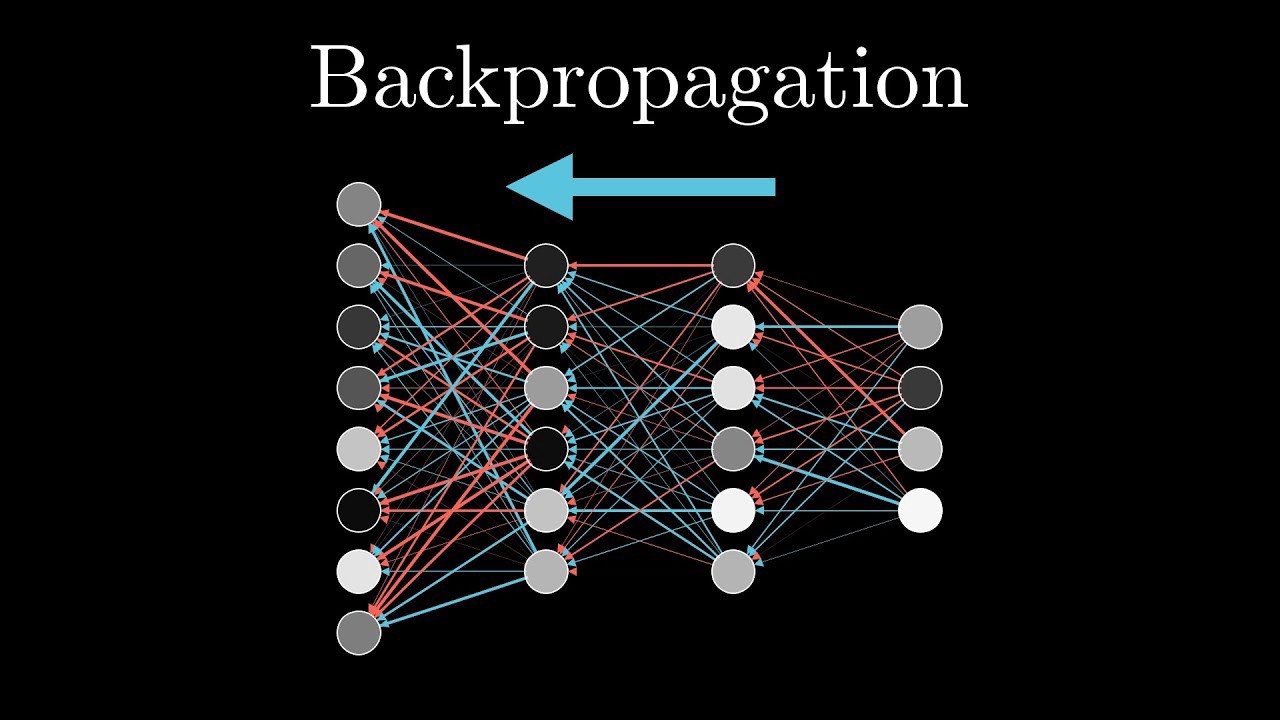
[Modelo 1 5](#_Toc59734426)

[Modelo 2 5](#_Toc59734427)

[Comparativa de los modelos 5](#_Toc59734428)

# Estudio y análisis de la arquitectura BPNN

Cuando necesitamos representar problemas complejos, no nos basta únicamente con un simple perceptrón, sino que necesitamos una red de perceptrones interconectados entre ellos.

Para el entrenamiento de una red se debe tener en cuenta que la salida de cada neurona no va a depender únicamente de las entradas del problema, sino que también depende de las salidas que ofrezcan el resto de las neuronas. Por este mismo motivo también podemos afirmar que el error cometido por una neurona no solo va a depender de que sus pesos sean los correctos o no, sino que dependerá del error que traiga acumulado del resto de neuronas que le precedan en la red.

Por lo que necesitamos un algoritmo eficiente que nos permita adaptar todos los pesos de una red multicapa, no sólo los de la capa de salida.

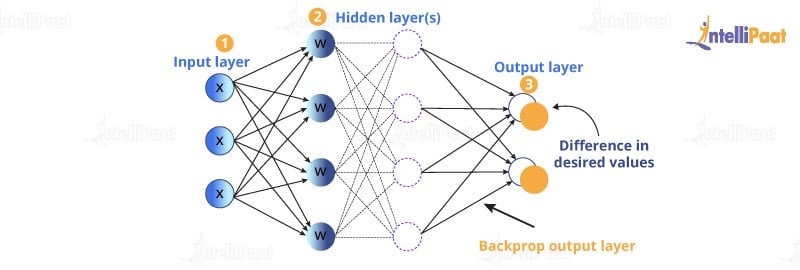
La **arquitectura de *Backpropagation*** consiste en interconectar varias unidades de procesamiento en capas (las neuronas de cada capa no se interconectan entre sí). Sin embargo, cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa, esto es, cada neurona transmitirá su señal de salida a cada neurona de la capa siguiente. Es decir, una red neuronal de retro propagación es una red neuronal multicapa de retroalimentación que consta de una capa de entrada, una (o varias) capa oculta y una capa de salida.

# Proceso de aprendizaje de la BPNN

Las redes *Backpropagation* tienen un método de entrenamiento supervisado. El objetivo de la retro propagación es optimizar los pesos (los cuales normalmente son inicializados aleatorios) para que la red neuronal pueda aprender a asignar correctamente entradas arbitrarias a salidas. El algoritmo conlleva una fase de propagación hacia adelante y otra fase de propagación hacia atrás.

**Propagación hacia adelante**: Esta fase de propagación hacia adelante se inicia cuando se presenta un conjunto de datos en la capa de entrada de la red. Las unidades de entrada toman el valor de su correspondiente elemento de entrada y se calcula el valor de activación (es necesario que la función de activación sea derivable). A continuación, las demás capas realizarán la fase de propagación hacia adelante que determina el nivel de activación de las otras capas hasta obtener la salida de la red (predicción de la red). Se realiza el cálculo del error cometido por la red mediante una función de pérdida.

**Propagación hacia atrás**: Una vez se ha completado la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de todos los pesos de las conexiones empiezan por la capa de salida y continua hacia atrás a través de todas las capas de la red hasta la capa de entrada. El objetivo con la propagación hacia atrás es actualizar cada uno de los pesos en la red para que provoquen que la salida real esté más cerca de la salida objetivo, minimizando así el error para cada neurona de salida y la red como un todo.



Cabe destacar que algunas redes *Backpropagation* utilizan unidades llamadas *bias* como parte de cualquiera de las capas ocultas y de la capa de salida. Estas unidades presentan constantemente un nivel de activación de valor 1. Además, esta unidad está conectada a todas las unidades de la capa inmediatamente superior y los pesos asociados a dichas conexiones son ajustables en el proceso de entrenamiento. La utilización de esta unidad tiene un doble objetivo: mejorar las propiedades de convergencia de la red y ofrecer un nuevo efecto umbral sobre la unidad que opera.

# Análisis del conjunto de datos

Se ha hecho uso del conjunto de datos: **Organización y selección de Jornadas de Conducción  
adecuadas en el Transporte de Mercancías Peligrosas**.

El conjunto de datos se compone de las siguientes características:

– *Fecha y hora de la observación*– *Longitud del Vehículo*– *Carril de circulación*– *Velocidad de circulación*– *Peso del Vehículo*– *Número de ejes*– *Temperatura del aire*– *Humedad relativa*  
– *Tipo de precipitación*– *Intensidad de la precipitación*– *Dirección del viento*– *Velocidad del viento*– *Estado carretera*– *Accidente (SI/NO)*

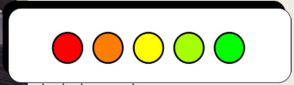
Antes de pasar el conjunto de datos a los modelos se ha decidido preprocesarlos para eliminar datos inconsistentes u observaciones sin valores:

Primeramente, se ha modificado la columna de la etiqueta a predecir (Accidente) estableciendo un 1 si se ha producido un accidente o 0 para lo contrario.

Luego, se ha realizado una codificación one-hot (*one-hot encoding)* a las columnas no numéricas: tipo de precipitación, intensidad de precipitación y estado de carretera.

Finalmente, la columna de la fecha y hora se ha optado por preservar el mes, día y hora, por lo que se han añadido tres nuevas columnas para los respectivos datos.

La salida de las redes será un valor que expresará el porcentaje de tener o no un accidente dadas las características. Dependiendo del porcentaje obtenido, el resultado se reducirá en 5 tipos de señales:



Siendo el rojo (76% - 100%) , naranja (51% - 75%) , amarillo (41% - 50%) , verde claro (21% - 40%) y verde oscuro (0% – 20%) probabilidades de que haya un accidente.

# Modelo 1

# Modelo 2

# Comparativa de los modelos

Accuracy

Loss