

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INFORMÁTICOS

**MODELADO DE COMPORTAMIENTO DE
CONDUCTORES CON TÉCNICAS DE
INTELIGENCIA COMPUTACIONAL**

TESIS DOCTORAL

Alberto Díaz Álvarez

Máster en Ciencias y Tecnologías de la Computación

DIRECCIÓN

Dr. Francisco Serradilla García

Doctor en Inteligencia Artificial

Dr. Felipe Jiménez Alonso

Doctor en Ingeniería Mecánica

Madrid 12 de febrero de 2018

Índice general

Inteligencia Computacional 9

Índice de figuras

1. La *Habitación China* de John Searle es un experimento mental por el que se trata de demostrar la invalidez del Test de Turing. Partimeos de un Test de Turing donde la máquina ha aprendido a hablar chino. Reemplazamos la máquina por un humano sin idea de chino pero con un manual de correspondencias de ideogramas. Cuando una persona le manda mensajes en chino, esta otra responde usando el manual, por lo que podemos afirmar que la persona, y por tanto la máquina, no saben chino. 10
2. Diferentes objetivos perseguidos por la ai 12
3. Ilustración de una sección del neocórtex humano 14
4. Modelo de neurona artificial de McCulloch y Pitts 15
5. Three training epochs on a multilayer perceptron with a 0.5 dropout rate (i.e. 50 % probability for a neuron to be disabled) in its hidden layer. The grey neurons are the ones disabled each epoch. 17
6. *Modus ponendo ponens* vs. *modus tollendo tollens* 21
7. Diferencias entre *modus ponens* tradicional y generalizado 21
8. Diagrama general de un fis 22
9. Esquema de agente y sus propiedades 24
10. Arquitectura básica de un agente 26
11. Diferencias entre un agente sin y con modelo de entorno 26
12. Arquitecturas de agente según su comportamiento 27
13. Diferencias entre colaboración y competitividad de agentes 29

Índice de cuadros

Inteligencia Computacional

Todo elemento dentro de un entorno se ve influenciado por multitud de variables que determinan en mayor o menor grado su comportamiento. Sin embargo, en la mayoría de los casos es muy complicado determinar el grado de efecto de estas variables, identificar las relaciones que existen entre las mismas o incluso determinar cuáles son.

La definición que esta tesis da de la **Inteligencia Computacional (IC)** es la de la rama de la **Inteligencia Artificial (IA)** que engloba a todas aquellas técnicas que tratan de *aprender*¹ soluciones a problemas a través de la observación y el análisis de información, ya sea presente en conjuntos de datos, en el entorno o ambos.

Sin embargo, no existe una definición globalmente aceptada ya que, dependiendo del autor es considerada desde un sinónimo de la **IA** hasta un campo completamente diferenciado.

Por ello, la primera sección del capítulo ofrecerá una visión histórica de la aparición del concepto para justificar el por qué de la definición. El resto del capítulo introducirá las nociones de agente y aprendizaje y ofrecerá una explicación en detalle de las técnicas sobre las que se apoya el trabajo teórico de esta tesis: Las **Red Neuronal Artificial** y la **Lógica Difusa**.

De *Inteligencia Artificial* a *Inteligencia Computacional*

Es difícil precisar el comienzo del interés del ser humano por la emular la inteligencia humana. Los silogismos en la antigua grecia para modelar el conocimiento como reglas o los autómatas mecánicos de los filósofos modernos (siglos XVII al XIX) donde los cuerpos vivos son como un reloj son sólo dos ejemplos de este hecho.

Podemos aventurarnos a decir que a principios del siglo XX se comienza a gestar el área de la **IA** con los trabajos relacionados con los principios del **conexionismo**².

Hacia mediados del siglo XX, estas ideas del conexionismo saltaron al campo de la computación cuando Warren S. McCulloch y Walter Pitts publicaron su trabajo “*A logical calculus of the ideas immanent in nervous activity*” [McCulloch and Pitts, 1943]³ donde se describe el

¹ Veremos más detalles sobre el concepto de aprendizaje en este contexto y sus implicaciones en el área de la **Inteligencia Artificial (IA)** más adelante

² El enfoque del **conexionismo** postula que tanto la *mente* como el *conocimiento* son comportamientos complejos que emergen de redes formadas por unidades sencillas (i.e. neuronas) interconectadas. Se puede considerar a Santiago Ramón y Cajal como principal precursor de esta idea por sus trabajos acerca de la estructura de las neuronas y sus conexiones (e.g. [Cajal, 1888] y [Ramón and Cajal, 1904])

³ Muchos autores prefieren nombrar este hito, junto con el trabajo “*The organization of behavior*” [Hebb, 1968] de Donald O. Hebb como el punto de partida del área debido a su connotación computacional

primer modelo artificial de una neurona.

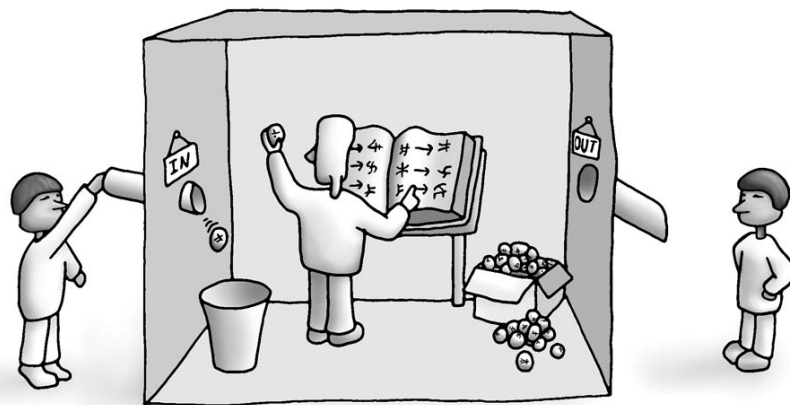
El trabajo suscitó tanta expectación que se comenzó a especular sobre la posibilidad de emular la inteligencia humana en máquinas. Uno de los resultados fue la publicación de un artículo por parte de Alan Turing que comenzaba con la frase “*Can machines think?*” [Turing, 1950], introduciendo el famoso Test de Turing ⁴ por el que el autor pretendía establecer una metodología para determinar si una máquina podía ser considerada inteligente y por tanto podría llegar a pensar ⁵.

Pocos años después de la publicación del artículo, en el año 1956, se celebró la **Conferencia Dartmouth** [McCarthy et al., 1956]. En ésta, el tema de la conferencia fue la pregunta del artículo de Turing, y el área nació con entidad propia tras acuñarlo John McCarthy como **Inteligencia Artificial**.

A partir de este momento, la investigación en el área recibió mucha atención por parte de investigadores y gobiernos. Después de todo era un área nueva, muy prometedora y con mucho trabajo por delante. Tras su nacimiento el campo comenzó a dar resultados, aunque quizá la expectación y las promesas no permitían ver que los resultados se obtenían en problemas relativamente simples, muy formales y en general estériles, donde en realidad no era necesaria demasiada información para generar un conocimiento del entorno en el que los modelos se movían.

⁴ El **Test de Turing** es una metodología para probar si una máquina es capaz de exhibir comportamiento inteligente similar al del ser humano. En ella, dos humanos (H_1 y H_2) y una máquina (M) están separados entre sí pero pudiendo intercambiarse mensajes de texto. H_2 envía preguntas a H_1 y M y éstos le responden. Si H_2 no es capaz de identificar qué participante es la máquina, se puede concluir que ésta es inteligente.

⁵ El concepto de “pensar” es un tema controvertido incluso en el propio ser humano: ¿es inherentemente biológico? ¿surge de la mente? Tanto si sí como si no, ¿de qué forma lo hace? Por ello existen detractores de la validez del Test de Turing. Un ejemplo es el experimento de la habitación china (Figura 1), donde se demuestra la invalidez argumentando que la máquina ha aprendido a realizar acciones sin *entender* lo que hace y por qué lo hace. Sin embargo, ¿qué garantías tenemos de que el humano sí es capaz? Si los ordenadores operan sobre símbolos sin comprender el verdadero contenido de éstos, ¿hasta qué punto los humanos lo hacen de forma diferente?.



jolyon.co.uk

Figura 1: La *Habitación China* de John Searle es un experimento mental por el que se trata de demostrar la invalidez del Test de Turing. Partimeos de un Test de Turing donde la máquina ha aprendido a hablar chino. Reemplazamos la máquina por un humano sin idea de chino pero con un manual de correspondencias de ideogramas. Cuando una persona le manda mensajes en chino, esta otra responde usando el manual, por lo que podemos afirmar que la persona, y por tanto la máquina, no saben chino.

Dado que los estudios estaban dominados por aquellos relacionados con las ideas del conexionismo, la publicación del libro “*Perceptrons*” [Minsky and Papert, 1969] de Marvin Minsky y Seymour Papert en 1969 supuso un varapalo para las investigaciones. En el se expusieron las limitaciones de los modelos de **Red Neuronal Artificial (RNA)** desarrollados hasta la fecha, y el impacto fue de tal envergadura que la investigación en el área se abandonó casi por completo.

Concretamente el conexionismo prácticamente desapareció de la literatura científica durante dos décadas. Es lo que se conoce como el primer *AI Winter* ⁶.

El interés por el campo volvió de nuevo a principios de los 80 con la aparición en escena de los primeros *Sistemas Expertos*, los cuales se consideran como el primer caso de éxito en la IA ([Russell et al., 2003]). A finales de la década, sin embargo, empezaron a resurgir de nuevo los enfoques conexionistas, debido en gran parte a la aparición de nuevas técnicas de entrenamiento en perceptrones multicapa y por el concepto de activación no lineal en neuronas [Rumelhart et al., 1985, Cybenko, 1989]). En este momento los sistemas expertos empezaron a perder interés frente al nuevo avance del conexionismo ⁷. Algunos autores identifican esta década como el segundo *AI Winter*, dado que la investigación e inversiones sobre *Sistemas Expertos* se vieron disminuidas.

TODO!Me he quedado aquí!!!!

Junto con el resurgir del conexionismo, otras técnica alineadas como la *Lógica Difusa (FL)* o los *Algoritmo Genéticos (AGs)* también ganaban popularidad, y entre ellas retroalimentaban los éxitos gracias a sus sinergias. Esto provocó una explosión de terminologías para diferenciar las investigaciones en curso de la propia IA clásica. Por un lado se evitaba el conflicto, nombrando las áreas de trabajo con un término más acorde con el comportamiento o técnica utilizada. Por otro, se separaba de las connotaciones negativas que fue cosechando la IA con el paso de los años (i.e. promesas, pero no resultados).

Lo verdaderamente interesante es ver la evolución de la literatura durante estos años. En el nacimiento del campo, se buscan literalmente máquinas que piensen como humanos, o al menos seres racionales, con mente. Con el paso de los años (y los continuos choques contra la realidad), la literatura va tendiendo hacia la búsqueda de conductas y comportamientos inteligentes cada vez más específicos. Este hecho se hace más patente en este momento, donde cada investigación se nombra de cualquier forma menos con el término IA (e.g. *Machine Learning (ML)*, *Recommender Systems (RS)*, o *Natural Language Programming (NLP)*). Es evidente que la IA se puede observar desde diferentes puntos de vista, todos perfectamente válidos. En [Russell et al., 2003], tras un análisis de las definiciones existentes en la literatura por parte de diferentes autores, se hace énfasis en este hecho mostrando los diferentes puntos de vista a la hora de hablar de lo que es la IA. El resumen se puede observar en la figura 2.

Volviendo al tema de la terminología, muchas de las técnicas se fueron agrupando dentro de diferentes áreas. Una de ellas es la conocida como ci. Dado que persigue el mismo objetivo a largo plazo y que surge de la propia ai parece lógico mantenerla como un subconjunto y no como un nuevo campo del conocimiento humano. Sin embargo, algunos autores abogan por que la ci es un campo diferenciado de la

⁶ El **AI Winter** no sólo se produjo por el efecto gurú del libro *Perceptrons*, aunque éste fue la gota que colmó el vaso. A la emoción inicial por los avances le siguieron muchos años de promesas incumplidas, investigación sin resultados significativos, limitaciones de hardware, aumento de la complejidad del software (los comienzos de la crisis del software [Dijkstra, 1972]). Todo ello provocó un desinterés y una disminución de la financiación que se retroalimentaron la una a la otra.

⁷ Esto no debió de sentar bien a los autores prolíficos en *Sistemas Expertos*. Mientras que el enfoque en estos sistemas es el clásico en la computación, donde los problemas (en este caso el conocimiento experto) son resueltos mediante operaciones sobre un lenguaje de símbolos, el enfoque del conexionismo postula que la mente, el comportamiento inteligente, emerge de modelos a más bajo nivel. Por ello, algunas voces se alzaron contra lo que se consideraba el *enfoque incorrecto* de la IA. Después de todo, los modelos desarrollados en los métodos clásicos son fáciles de interpretar mientras que los del enfoque conexionista no son del todo deducibles, más aún si estos problemas son de naturaleza estocástica.

ai.

Podemos definir la ci como la “rama de la ai que aporta soluciones a tareas específicas de forma *inteligente* a partir del aprendizaje mediante el uso de *datos experimentales*”. A diferencia de la aproximación clásica de la ai, se buscan aproximaciones a las soluciones y no las soluciones exactas. Esto es debido a que muchos problemas son de naturaleza compleja, ya sea por la relación entre sus múltiples variables, a la falta de información o a la imposibilidad de traducirlos a lenguaje binario.

Figura 2: Diferentes objetivos perseguidos por la ai. Las filas diferencian entre pensamiento o comportamiento mientras que las columnas separan entre inteligencia humana o el ideal de la inteligencia (racionalidad). Fuente: *Artificial Intelligence: A Modern Approach* (3rd Ed.), [Russell et al., 2003].

Thinking Humanly "The exciting new effort to make computers think . . . <i>machines with minds</i> , in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning .. ." (Hellman, 1978)	Thinking Rationally "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly "The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	Acting Rationally "Computational Intelligence is the study of the design of intelligent agents." (Poole et al, 1998) "AI ... is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

⁸ <http://cis.ieee.org/>

Se puede establecer el año 1994 como en el que la ci nace formalmente como área, coincidiendo con el cambio de nombre del *IEEE Neural Networks Council* a *IEEE Computational Intelligence Society* ⁸. Poco antes, en 1993, Bob Marks presentaba las que él consideraba diferencias fundamentales entre la ai clásica y la ci, resumiéndolas en la siguiente frase:

“Neural networks, genetic algorithms, fuzzy systems, evolutionary programming, and artificial life are the building blocks of CI.”

Durante estos años ganaba popularidad también el concepto del en contraposición con el . El engloba las técnicas que buscan resolver problemas con información incompleta o con ruido. Debido a que el conjunto de técnicas definidas como consituyentes del son las mismas que se usan en la **Inteligencia Computacional (IC)** algunos autores consideran ambos términos equivalentes. Nosotros consideramos que el es un punto de vista de la computación y que la **IC** es un área de la Inteligencia Artificial hace uso de métodos del .

TODO!Añadir como colofón que hoy en día la inteligencia computacional es un campo con muchas aplicaciones prácticas y muchísimos research topics abiertos.

y son la forma de referirse a la computación convencional frente al . El basa sus técnicas en aquellas basadas en modelos analíticos definidos de forma precisa y que en ocasiones requieren mucho tiempo de cómputo. Están basados en lógica binaria, análisis numérico, algoritmos y respuestas exactas. El por otro lado es tolerante a la imprecisión y al ruido y tiende a llegar a soluciones aproximadas de manera más rápida. Se basa en modelos aproximados, emergencia de algoritmos y modelos estocásticos.

El rol del aprendizaje en la *Inteligencia Computacional*

El cambio más notorio entre los dos puntos de vista de la *IA* tradicional y la de la *IC* es el concepto de entrenamiento, es decir, pasar de “desarrollar un programa para resolver un problema” a “entrenar un modelo para que aprenda la solución”. Éste es el concepto de **aprendizaje**, y al proceso de ajuste del modelo a la solución buscada **entrenamiento**.

Las técnicas de aprendizaje se clasifican dependiendo de la forma en la que se entrenan. *Grosso modo*, cuando al modelo se presentan ejemplos de preguntas y respuestas para que las aprendan se dice que estamos realizando un *aprendizaje supervisado* mientras que, si lo que le presentamos es un conjunto de datos en bruto del que extraer relaciones o patrones, hablamos de un *aprendizaje no supervisado* ^{9 10}.

En el caso concreto del **aprendizaje supervisado**, vamos a suponer que disponemos de un modelo denotado por $M_V(V, I) = O$ donde V es el conjunto de variables que determinan el comportamiento de M_V y donde I es un conjunto de valores (características) de entrada para las que el modelo obtiene un conjunto O de valores de salida. Entonces, la forma de entrenar al modelo, es decir el *algoritmo de entrenamiento* se encargaría de, a partir de un conjunto de la forma $D = (I_i, O_i) | \forall i \in \mathbb{N}$, donde cada O_i es la salida esperada del modelo a la entrada I_i , modificar los valores de las variables del conjunto V para ajustar lo más posible O_i a O dado I_i .

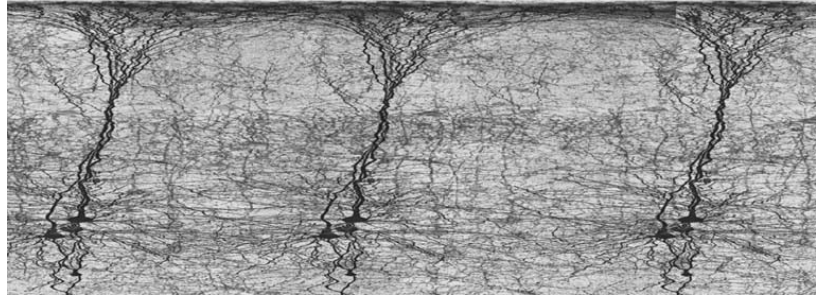
Los algoritmos de entrenamiento dependen tanto del modelo a usar (no es lo mismo un algoritmo de entrenamiento para una red neuronal recurrente que para un perceptrón multicapa), y suelen ser usado principalmente para la solución a problemas de **clasificación** (i.e. determinar si un elemento dada sus características pertenece o no a determinado conjunto) y de **regresión**, esto es, ajustar las salidas de un modelo para ajustarse lo máximo posible al valor real del sistema modelado.

⁹ El **aprendizaje no supervisado** es el proceso por el cual un modelo aprende a partir de datos en brutos sus relaciones y extrae patrones, sin saber qué son esos datos ni recibir supervisión (a diferencia del aprendizaje supervisado donde los datos incluyen un valor de entrada y su salida correspondiente). En general, los algoritmos pertenecientes a esta categoría se dedican al problema del *clustering*, es decir, identificar grupos de elementos cercanos en el espacio basándose en la suposición de que el comportamiento de dos elementos es más parecido cuanto más cerca están el uno del otro. Algunos ejemplos de técnicas que basan su entrenamiento en un esquema no supervisado son los mapas autoorganizados (SOM), los autoencoders o las redes de creencia profunda (DBN de Deep Belief Network).

Esta tesis se dedica al modelado de comportamiento entrenando a partir de datos reales de conducción, por lo que el discurso se centrará únicamente en el esquema de aprendizaje supervisado.

¹⁰ Algunos autores hacen uso de técnicas pertenecientes a ambos paradigmas en forma de aproximación híbrida para suplir deficiencias u optimizar/acelerar el aprendizaje. Un claro ejemplo lo podemos ver en [Hinton, 2006], donde los autores hacen uso de *autoencoders* como técnica no supervisada para la inicialización de los pesos de una red neuronal y posteriormente realizan un entrenamiento supervisado para la optimización es éstos.

Figura 3: Sección del neocórtex humano, región asociada a las capacidades cognitivas y que supone alrededor de un 76 % del volumen total del cerebro humano. Está distribuido en 6 capas y miles de columnas que las atraviesan, cada una con alrededor de 10,000 neuronas y un diámetro de 0,5mm. Como dato anecdótico, se estima que sólo en el neocórtex humano existen alrededor de 20,000 millones de neuronas, cada una de las cuales conectada a entre 100 y 100,000 neuronas vecinas ([Pakkenberg and Gundersen, 1997]). Esto supone entre $2 \cdot 10^{12}$ y $2 \cdot 10^{15}$ conexiones. Fuente: Blue Brain Project EPFL, <http://bluebrain.epfl.ch/>.



Deep learning

El paradigma de los Agentes Inteligentes

Tipos de entorno

Arquitecturas en sistemas de agentes

Sistemas multiagente

Red Neuronal Artificiales

¹¹ Aún apoyándose en la topología y funcionamiento del cerebro humano para realizar el símil, lo cierto es que dichos modelos distan aún de considerarse *cerébros artificiales*. La red neuronal más compleja hasta la fecha es la propuesta en [Trask ANDREWTRASK et al.,], con alrededor de 160,000 parámetros a ser ajustados (podemos abstraernos y pensar en ellos como conexiones entre neuronas). Si comparamos esta cifra sólo con las del neocórtex (figura 3) hace que, tecnológicamente hablando, nos quedemos con la sensación de estar aún a años luz de aproximarnos a la complejidad de un cerebro humano.

Son herramientas que tratan de replicar las funciones cerebrales de un ser vivo de una manera muy fundamental, esto es, desde sus componentes más básicos, las neuronas. Para ello se basan en estudios de neurobiología y de ciencia cognitiva moderna del cerebro humano ¹¹.

Una RNA es independiente del problema a solucionar. Se la puede considerar como una caja negra que aprende las relaciones que subyacen en los datos de un problema para abstraer el modelo a partir de éstos. Estas características de aprendizaje y abstracción son los factores determinantes por los que son usadas en prácticamente todas las áreas de la ciencia y de la ingeniería ([Du and Swamy, 2006]).

El primer trabajo en la disciplina se le atribuye a los investigadores McCulloch-Pitts por su modelo de neurona artificial ilustrado en la figura 4 ([McCulloch and Pitts, 1943]). Existen diferentes tipologías y formas de operar con redes, pero todas funcionan de la misma manera: unidades (e.g. neuronas) interconectados mediante enlaces por los que fluye la información de manera unidireccional, donde algunas de dichas unidades sirven de entrada al sistema (i.e. entradas o sensores), otras sirven de salida del sistema (i.e. salidas y actuadores) y otras como elementos internos (i.e. ocultas), y donde las conexiones se ajustan mediante un proceso denominado *entrenamiento*.

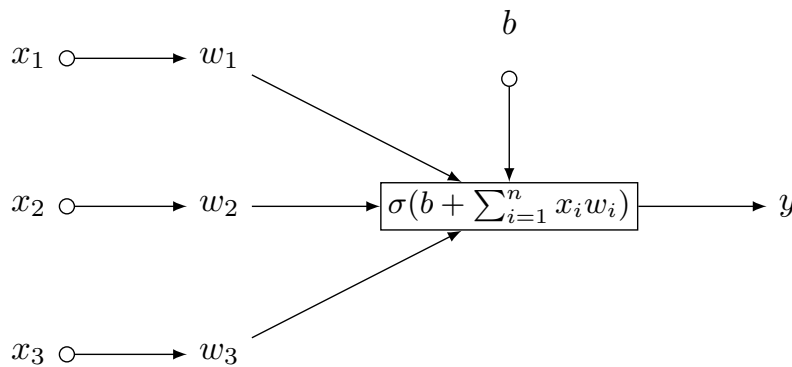


Figura 4: Variación de la representación del modelo de neurona artificial propuesto por McCulloch y Pitts. En éste, cada una de las entradas x_i es incrementada o inhibida aplicando el producto con su peso asociado w_i . La activación vendrá determinada por la aplicación de una función (denominada “de activación”) a la suma de los valores. Esta variación en concreto incluye una entrada x_0 y un peso w_0 como bias de la neurona para la variación dinámica del umbral de activación.

Estructura y clasificación de redes neuronales

Hablar del modelo de neurona artificial, de feed-forward vs recurrentes y del efecto de las diferentes funciones de activación.

En esta tesis el trabajo se centra en dos topologías concretas dentro de las redes de tipo *feed-forward*, el **Perceptrón Multicapa** y la **Red neuronal convolucional**, que son con las que se cerrará la sección.

Perceptrones multicapa

Redes de convolución

Este primer modelo de neurona proponía una función escalón para determinar si la neurona se activaba o no (analogía al funcionamiento de la neurona artificial). **TODO!**Parrafito sobre la limitación de la neurona singular para dar hilo a las topologías.

Posteriormente aparecieron nuevos modelos de neuronas con diferentes funciones de activación. De éstas, las más comunes son las de tipo sigmoide ¹². **TODO!**Actualizar esto porque creo que empieza a ser más común la ReLU, que se usa hasta en la sopa. Una o dos ilustraciones mostrando los diferentes tipos (grafiquitas) de las funciones de activación en redes neuronales. Para que sea de un poco más de 1990 y no esta mierda. Poner escalón, sigmoidal y tanh (como clásicas) y ReLU y ajustad ReLU (o como coño se llamen) como modernas.

¹² Concretamente la función logística de pendiente 1 definida como:

$$\frac{1}{1 + e^{-x}} \quad (1)$$

EXISTEN DIFERENTES TOPOLOGÍAS DE REDES NEURONALES o arquitecturas dependiendo de qué forma toma el grafo que modela las neuronas y sus conexiones. En este caso, las redes neuronales pueden ser de dos tipos:

- **Feed-Forward.** Sus grafos no contienen ningún ciclo (figura XXX)
| **TODO!**Esa pedaaaaaaso de figura. A lo mejor hacer una que incluya tres ejemplos.. Es la topología más usada en aplicaciones prácticas debido a su sencillez y su efectividad. En ellas el flujo de

información sigue un camino desde las entradas hasta las salidas, sin ninguna retroalimentación. No es requisito que las neuronas se agrupen en capas, aunque suele ser la estructura común. A las redes de más de dos capas ocultas (i.e. las capas que se encuentran entre la capa de neuronas de entrada y la capa de neuronas de salida) se las denomina “profundas” o *deep*. Algunos tipos pertenecientes a esta categoría pueden ser el Perceptrón [Rosemblat, 1957], el Perceptrón multicapa [Rumelhart et al., 1986], el algoritmo LVQ y su sucesor los Mapas Auto-Organizados [Kohonen, 1998].

- **Recurrentes.** Sus grafos contienen uno o más ciclos, de tal manera que el flujo de información de salida de una neurona puede llegar a afectar a su propio estado. Estas topologías representan de una forma más fiel las bases biológicas de las RNA, pero son más complejas a la hora de operar y entrenar. Algunos casos particulares de este tipo de arquitectura son las Redes de Hopfield [Hopfield, 1982] o las memorias LSTM (del inglés Long-Short Term Memory) [Hochreiter & Schmidhuber, 1997].

TODO!Más cosas a añadir del paper de modelling behavior of lane change execution

ANNs are CI techniques used as a general solution in prediction and classification problems. They are composed by single processing units called neurons and are usually classified depending on how are arranged and connected those neurons. Depending on its architecture, the ANN can be trained on a problem by means of a supervised training scheme (i.e. by showing tuples of input \rightarrow output) or non-supervised training scheme (i.e. by showing a whole dataset and letting them to find patterns and solutions by themselves). They have been used successfully in several fields like speech recognition [20], series and sequences prediction [21] or terrain classification [22].

In this paper, for mimicking the lane-change acceptance of a particular user, two different families of ANN are proposed: MLPs and CNNs. Figure 2. Example of two feed-forward architectures: (a) A Multilayer Perceptron architecture; (b) A Convolutional Neural Network architecture. Figure 2. Example of two feed-forward architectures: (a) A Multilayer Perceptron architecture; (b) A Convolutional Neural Network architecture. Both families rely on the same model of artificial neuron, the McCulloch-Pitts model [23], and belong to the category feed-forward, i.e. a Directed Acyclic Graph (DAG) where the inputs are presented by one end of the DAG (called input layer) and the outputs are collected at the other end (called output layer).

In these two models, the connections are represented with a real value called weight imitating the behaviour of biological neurons with the strength of connection between axon and dendrites. Typically, the training process belongs to the supervised scheme and is based on the modification of these weights to obtain the desired response, following the principles of the Hebbian theory.

One of the most difficult tasks on these models is their hyper-parameter tuning, and it is difficult because ANNs involve a lot of them. Apart from the family, and the arrangement of their neurons that fortunately, in the most of cases, comes imposed by the family, there are number of neurons, number of layers, training parameters (depending on the learning algorithm), and they follow the non-free lunch theorem [24]. The problems derived from that are two: the high bias or under-fitting and the high variance or over fitting.

To avoid under-fitting, the classical technique is to increase the size of the network, both deepness and/or number of neurons. Over-fitting, on the other hand, is a little bit harder because it can be motivated for the lack of data, the number of layers, the number of neurons, etc. In these cases, the solution may be the augmentation of training set, either naturally or artificially, and/or regularization techniques. We will talk later about the data augmentation techniques used in this work.

The regularization technique used in this experiment is the one called Dropout [25]. It's a simple technique where in each training epoch some random neurons are, indeed, dropped. This technique slows down the convergence process a bit, also avoiding the neurons to rely only in few of their inputs, thus reducing the over-fitting. Figure 3 shows an example of this techniques through three different epochs.

TODO! Quizá esto quede bien al lado como ejemplo de dropout.

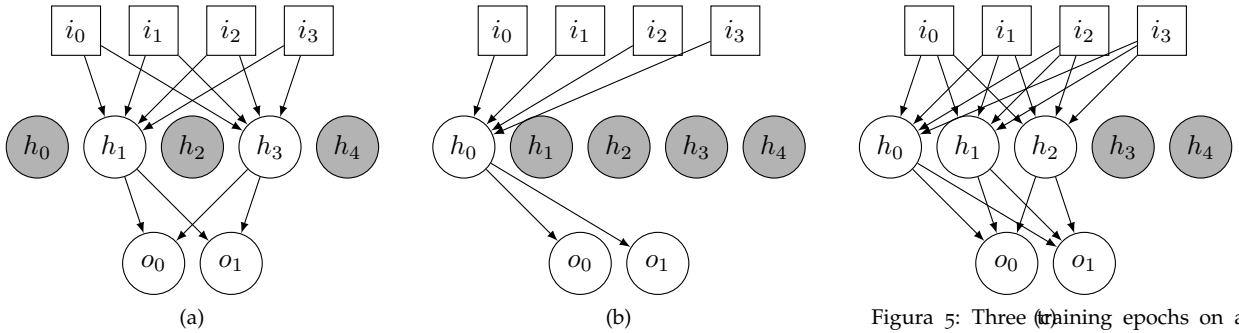


Figura 5: Three training epochs on a multilayer perceptron with a 0.5 dropout rate (i.e. 50 % probability for a neuron to be disabled) in its hidden layer. The grey neurons are the ones disabled each epoch.

Multilayer Perceptrons In a MLP, the neurons are arranged in layers so that all the neuron outputs on one layer are the inputs for all the neurons in the next layer. The first and last layers are called input layer and output layer respectively. The inner layers are called hidden layers. As shown in Figure 2 (b) (a two-layered MLP), in this architecture the inputs are usually presented as vectors. As they have been used extensively in several areas with great success, we use them here to make a comparison of the improvement of the use of CNNs over MLPs.

Convolutional Neural Networks As its name suggests, a CNN uses convolutions, i.e. mathematical operations that filter regions of a structure (generally in the form of a matrix or a cube) for pattern identi-

fication and/or structure transformations. A CNN has also an arrangement of layers, but they work differently. Firstly, this architecture is separated into two groups of layers or phases that can be called pattern identification phase and prediction phase. The pattern recognition phase works with layers dealing with pattern recognitions and input sub-sampling while the prediction phase works with an MLP using the output of the pattern recognition phase as input. Secondly, the inputs are arranged as n -dimensional matrices as opposed to the MLP that are always presented as vectors; this fact allows us working more easily with properties and patterns arranged in an n -dimensional space like images (2-dimension matrix) or videos (3-dimension matrix). The layer types in that phase are as follows: 1. Convolutional Layer. Given a n -dimensional input, a convolution or filter is an n -dimensional pattern (composed by neurons) that travels along the n -dimensional space to generate a new n -dimensional space. A convolutional layer is a layer composed by m filter and generate a set of m new n -dimensional spaces which will be the new input for another layer. 2. Local Response Normalization (LRN) layer [26]. This layer is intended to enhance the difference between the existent values in a space. They are commonly placed after a convolution layer to increase the contrast of the values in the space. In this paper, every convolution layer is followed by an LRN layer. 3. Pooling layer. Pooling is grouping together elements. A pooling layer is intended to subsample a space to ease its management. Given an n -dimensional space, a pooling filter of dimension travels the space in steps of size generating one value per step. When placed after convolutions, it is expected to subsample the space without losing the recognized pattern while reducing computation costs and avoiding overfitting. The most common operation used in pooling is the max operation.

TODO!Desarrollar un poquito el tema de los grafos computacionales indicando que si bien no se tratan de una técnica dentro de la inteligencia computacional, sí que se usan para representar operaciones.

Grafos computacionales

En la actualidad el concepto de grafo computacional se relaciona directamente con las redes neuronales, y por ello se introduce el concepto en este apartado. Sin embargo, no se trata de un concepto exclusivo de esta técnica. De hecho, ni siquiera es un concepto perteneciente al área de la inteligencia computacional, sino que es simplemente una forma de representar operaciones sobre datos.

Formalmente, un **grafo computacional** es un grafo dirigido donde los vértices representan operaciones sobre datos mientras que las aristas representan el flujo de dichos datos. La figura XXX ilustra un ejemplo de un grafo computacional.

Una ventaja al representar un modelo como grafo computacional

es que ayuda a abstraerse de la formas de las entradas y las salidas, facilitando el trabajo de operaciones en batch. Otra ventaja, todavía mayor, es que, al organizar de entrada a salida (en la figura XXX de izquierda a derecha) las operaciones que se necesitan para obtener una salida a partir de una entrada, en los casos donde el objetivo es optimizar la salida permiten fácilmente representar el gradiente al organizarlo del modo contrario (es decir, de las salidas a la entrada o, en el caso de la figura XXX de derecha a izquierda).

TODO! Poner aquí un ejemplo de grafo computacional, por ejemplo una recta o algo así. Algo que tenga al menos dos pasos y explicar cómo se calcula la derivada y por qué esto viene genial. Luego, en el siguiente capítulo se puede explicar el backpropagation con esto (ver <http://colah.github.io/posts/2015-08-Backprop/>)

Aprendizaje en Red Neuronal Artificiales

TODO! Falta un **huevazo**. Por lo menos hay que desarrollar las arquitecturas de MLP y CNN (que son las que uso), explicar el descenso del gradiente y ADAM (y cómo se llega a él explicando los conceptos de momento y su puta madre), explicar también cuáles son los problemas de High bias y High variance o overfitting y underfitting y ilustraciones bonitas del to. También hay que explicar qué es un grafo computacional, aunque quizá esto mejor después de la lógica difusa y antes de agentes inteligentes. También hay que hablar de qué es el aprendizaje profundo, deep learning vs shallow learning.

¿Qué es el aprendizaje profundo?

Lógica Difusa

Ampliando la teoría de conjuntos

Aquí dentro hay que hablar también de operaciones entre conjuntos. Como aquí hablaremos de sistemas de orden tipo 1, en los bordes igual estaría genial hablar de qué son los de tipo 2, 3, ...

Razonamiento

Sistemas de inferencia difusa

Hablar de los tipos de sistemas (Tipos Mamdani y Sugeno diferentes).

Grafos computacionales

fl

La **lógica** nace en el siglo IV a.C. dentro de la física Aristotélica, que permaneció inalterada hasta la revolución científica (alrededor del siglo XVI. d.C.), momento en que se separó y permaneció como disciplina paralela perteneciente más al campo de la filosofía que de la física y la matemática. Empezó a relacionarse de nuevo con la matemática a principios del siglo XIX y a principios del siglo XX la lógica y la teoría de conjuntos pasaron a convertirse en partes indispensables la una de la otra. Por ello suelen ir de la mano cada vez que se habla de la una y de la otra. La evolución de la teoría de conjuntos (Cantor, finales del siglo XIX, buscar referencia) y su unión con la lógica es una época bastante convulsa dentro de la historia de la matemática.

La lógica matemática (y por extensión la teoría de conjuntos) tiene como misión servir de fundamento del razonamiento matemático. Se basa en la definición precisa y con rigor de un razonamiento evitando cualquier tipo de ambigüedad y de contradicción. Es por ello que la lógica tradicional no suele servir como fundamento de razonamientos del mundo real.

Los conceptos que se manejan en el mundo real suelen ser vagos, llenos de imprecisiones. Además tienden a ser nombrados cualitativamente, no cuantitativamente, y cuando existe una correspondencia, ésta suele estar marcada por la subjetividad de los términos.

Explicar lógica difusa y control difuso. Indicar los controladores difusos de segundo, tercer y sucesivos niveles.

Teoría de conjuntos difusos

A diferencia de los conjuntos tradicionales, los conjuntos difusos expresan el grado de pertenencia de un elemento a la categoría representada por el conjunto. La definición podría escribirse de la siguiente manera:

TODO!Creo que habría que definir antes qué es un dominio

Definición 1: Sea X una colección de elementos. Se define al **conjunto difuso** F como un conjunto ordenado de pares de la forma $F = (x, \mu_F(x)) | x \in X$, siendo $\mu_F(x) \in [0, 1] \forall x \in X$.

La función de la definición 1 se denomina **función de pertenencia**, y caracteriza unívocamente a un conjunto difuso del dominio de X .

TODO!Quizá aquí habría que decir qué es una partición de un dominio

Operaciones entre conjuntos

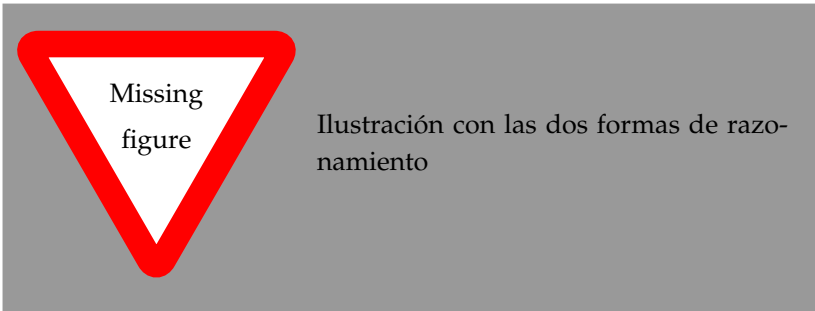
La unión, intersección y el complemento son operaciones básicas en la teoría de conjuntos. **TODO!** hablar aquí de t_{norm} , t_{conorm} y complemento, pero someramente. No hay que enrollarse demasiado.

Razonamiento

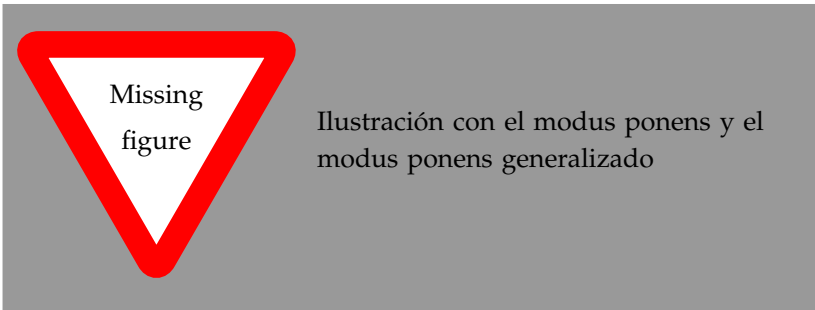
Al igual que en la lógica tradicional, en la **FL** el razonamiento o inferencia es la manera de extraer conclusiones a partir de premisas en función de un conjunto de reglas.

Estas reglas se expresan como implicaciones, definidas típicamente en lógica difusa como $A \rightarrow B \equiv A \wedge B$.

Las dos formas de extraer conclusiones a partir de premisas en FL son el *modus ponens* generalizado (del que hablaremos) y el *modus tollens* generalizado, modificaciones sobre los procesos de inferencia *modus ponens* y *modus tollens*¹⁴, dos formas similares de razonamiento (figura 6). Nosotros centraremos nuestro discurso en la primera.



EL MODUS PONENS GENERALIZADO es una generalización del *modus ponens* de la lógica tradicional donde, en lugar de expresar las reglas de forma absoluta, se expresan de forma aproximada. En la figura 7 se ilustra las diferencias fundamentales entre ambos modos.



Para determinar qué grado le asignamos a un consecuente a partir de las premisas parciales y las reglas que dirigen el razonamiento se utiliza un método denominado **regla composicional de inferencia**.

Una regla $A' \rightarrow B'$ se puede representar como una implicación caracterizada por una función $I(\mu_A(x), \mu_B(y))$ ([Ful,]).

TODO!Explicar mejor porque es terrorífico.

...

Tengo que hablar también del softmax, el por qué usarlo. También de la entropía cruzada y de por qué usamos este loss sobre el resto.

He visto una cosa curiosa que es el "negative sampling". Podría

¹³ La **implicación** en lógica se representa como $A \rightarrow B$, donde A es cualquier operación de premisas y B la conclusión que arrojan.

En lógica tradicional, el valor de verdad de una implicación es equivalente al de la expresión $\neg A \vee B$. Sin embargo, en lógicas multivaluadas (y por tanto en lógica difusa) esta equivalencia da lugar a razonamientos que se pueden considerar contraintuitivos.

En el caso concreto de la lógica difusa se han propuesto muchas cantidad de equivalencias. Sólo en los trabajos [Kiszka et al., 1985] se analizan 72 alter-

Figura 6: Formas de razonamiento en lógica tradicional usadas por obstante el *modus tollens* debido a su rendimiento (en la implicación de Mamdani la T -norma se implementa como el operador mínimo).

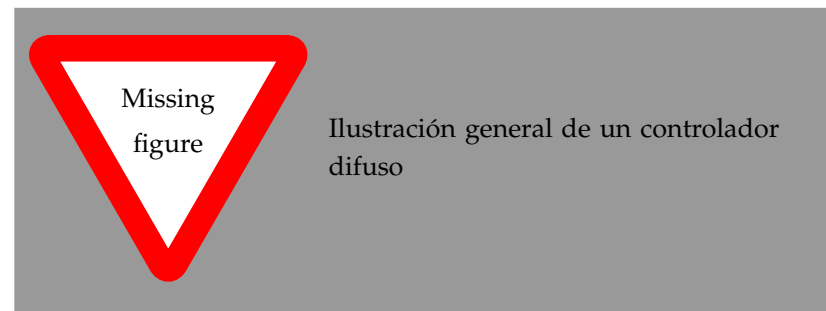
¹⁴ En realidad se llaman *modus ponendo ponens* ("la forma que al afirmar, afirma") y *modus tollendo tollens* ("la forma que al negar, niega").

hablar de ello y usarlo porque haría el entrenamiento mucho más rápido. Por lo que me ha parecido ver, se usa en lugar de la capa softmax y de lo que va es transformar la capa softmax (que es coñazo de calcular) en una capa de $k+1$ clasificadores binarios, donde hay 1 correcto y k incorrectos. Se usa en clasificación.

En [Ma, 2004] hay un capítulo de razonamiento que parece que está guay. Revisarlo un poco a fondo a ver si merece la pena tirar or ahí.

Los (o) son el caso de éxito de la lógica difusa que más resultados ha cosechado tanto a nivel académico como a nivel industrial. Se trata sistemas que utilizan el razonamiento difuso para inferir una respuesta a partir de un conjunto de entradas.

Figura 8: Diagrama del esquema general de un fis.



Habitualmente son descritos como un componente dividido en tres bloques conceptuales:

- **Fuzzificación.** Traducir los valores de entrada en crudo del dominio sobre el que está definida cada variable lingüística a sus respectivos grados de pertenencia a conjuntos difusos a través de sus funciones de pertenencia. **TODO!** Ojo, algunos controladores toman como valores de entrada conjuntos difusos según [Ma, 2004]. Habrá que buscar sobre ello.
- **Inferencia.** Realiza todo el proceso de razonamiento difuso a partir del conjunto de reglas que dan significado a este controlador difuso.
- **Defuzzificación.** Traduce el conjunto difuso resultado del proceso de inferencia a valores de los dominios sobre los que están definidos dichos conjuntos difusos. **TODO!** En un Sugeno, la salida es una función directamente así que se podría especificar que en un tipo Sugeno, se puede ver como que la salida son sólo singletons, manteniendo la generalización del proceso de funcionamiento de un fis.

Esta división se ilustra en la figura 8.

HAY VARIOS TIPOS DIFERENTES DE , aunque tienden a seguir el esquema básico de un controlador difuso típico (figura 8).

Sistemas de tipo Mamdani Son la primera aproximación de propuestos

Sistemas de tipo Takagi-Sugeno ...

Hablar someramente de los tres tipos clásicos que se usan, e indicar que al final los más usados son el Mamdani y el Sugeno. Añadir también quizá una tabla comparativa entre los tres o al menos entre los dos principales:

El consecuente de un fis de tipo Mamdani siempre es un conjunto difuso. Por tanto, el proceso de sacar un valor crisp es costoso. Lo bueno, se mantiene significado semántico de las salidas. El consecuente en un Sugeno es un valor, y se puede decir que no necesita proceso de defuzzificación. Si embargo, la respuesta pierde significado semántico si la suma de la fuerza de salida no es 1 (no entiendo qué quiero decir con esto).

Agentes inteligentes

Si echamos un poco la vista atrás, en la figura 2 se mostraban los cuatro objetivos perseguidos por la ai. En uno de ellos en particular se la entiende como el estudio del conseguir que las entidades (e.g. sistemas, software, ...) actúen de la manera más inteligente posible. A dichas entidades se las conoce como *agentes*, concretamente en este contexto como *agentes inteligentes*¹⁵. Sin embargo, si es difícil encontrar un consenso en la definición de agente más lo es a la hora de definir cuándo la conducta de éstos es inteligente.

Lo que sí existe es una serie de características comunes que se repiten a lo largo de la literatura (figura 9):

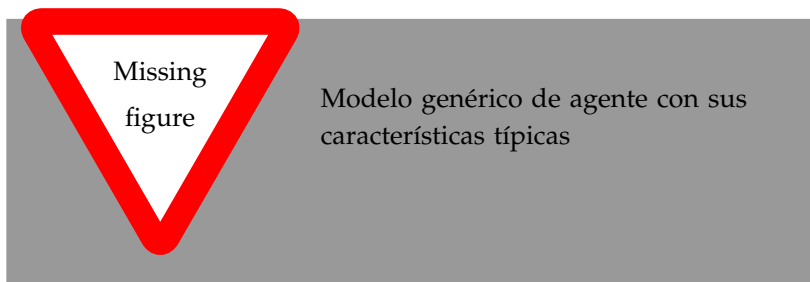
- Operan siempre en un **entorno**, ya sea éste físico (e.g. una red de carreteras para un vehículo autónomo) o virtual (e.g. un cliente de correo electrónico para un clasificador de spam).
- Tienen la capacidad de **percibir** el entorno por medio de *sensores* y de **actuar** sobre él por medio de *actuadores*.
- Son **autónomos** en el sentido de que pueden actuar sin intervención externa (e.g. humana u otros agentes) teniendo control sobre su estado interno y su comportamiento. Algunos autores les presuponen una autonomía absoluta mientras que otros hablan de que sólo es necesaria cierta autonomía parcial.
- Tienen **objetivos** a cumplir, actuando para ello sobre el entorno de la manera que les indique su comportamiento.

¹⁵ En realidad los autores prefieren denominarlo *agente racional*, dado que captura la esencia de lo que es un comportamiento inteligente. Sin embargo, según esta definición, hasta un elemento tan rudimentario como un termostato puede ser considerado como elemento inteligente, ya que realiza siempre la mejor acción para cumplir sus objetivos, por simples que puedan parecer. Dónde está el límite entre qué es y que no es un agente inteligente cae dentro de los dominios de la filosofía.

- Pueden ser **sociales**, es decir, tienen la capacidad de comunicarse con otras entidades (e.g. otros agentes) para llevar a cabo sus objetivos.

Por tanto nosotros usaremos la siguiente definición: Un agente es una entidad física o virtual que realiza una acción ¹⁶ de manera total o parcialmente autónoma dada una secuencia de percepciones del entorno en el que se ubica.

¹⁶ En [Russell et al., 2003] se define como “...just something that acts” alegando que la palabra *agent* proviene del latín *agere*. Para clarificar esto, *agere* es la forma verbal para *hacer*, pero imprime un significado de movimiento/acción. El agente es una entidad que tiene propiedades que le permiten interactuar con el entorno. Aun cuando no existe una definición única de agente, se puede decir que un agente es una entidad que posee propiedades que le permiten interactuar con el entorno, la planificación de acciones sobre un entorno dependiendo de las percepciones que le llegan de éste y tiene la capacidad de comunicarse con el resto de elementos, incluidos otros agentes.



Pero, ¿qué hace a un agente inteligente? Según algunos autores, el hecho de que posea unos objetivos y autonomía suficiente para cumplirlos ya denota inteligencia (**TODO!** encontrar el trabajo y citar). Según otros, es necesario que el comportamiento sea flexible, esto es, que sea reactivo (reacciona ante el entorno que percibe), proactivo (iniciativa para tratar de cumplir sus objetivos) y social (capaz de interactuar con otros agentes para cumplir sus objetivos) [Wooldridge et al., 1995]. Y otros directamente exigen, además, un comportamiento racional a la hora de cumplir los objetivos para calificarlo de inteligente (**TODO!** encontrar el trabajo y citar).

Por tanto, asumiremos la definición ofrecida por [Russell et al., 2003] donde, se indica que un agente es considerado **agente inteligente** cuando éste realiza la mejor acción posible (según un criterio de medida). En este contexto, “la mejor acción posible” se refiere en términos de objetivos y comprensión del entorno, que puede ser o no correcta ¹⁷.

Las nociones de agentes inteligentes y la de *ci* van de la mano. Esto es debido a que su definición funciona a la perfección para las técnicas de la *ci*, esto es, agentes autónomos que perciben el entorno (problema) y actúan de la mejor manera posible sobre él (resuelven) de acuerdo a su conocimiento del medio y su estado interno (en base a algoritmos como *ann*, *fl*, ...). Por ello desde mediados de los años 1990 el concepto de agente inteligente ha ganado tanta popularidad ¹⁸.

Tipos de entorno

La tupla (*entorno, agente*) es esencialmente una metáfora para referirse a la tupla (*problema, solución*) por lo que existen casi tantos

¹⁷ Que la comprensión del entorno no sea total es un factor clave que diferencia la racionalidad de la omnisciencia. La omnisciencia significa conocer el resultado de toda acción antes de realizarla y por tanto implica el conocimiento de absolutamente todos los detalles del entorno. La racionalidad existe dentro de un contexto de conocimiento limitado.

¹⁸ Tanto es así que en algunos trabajos se define el objetivo de la *ai* como la implementación de la función agente, esto es, la función que realiza la correspondencia de una percepción a una acción, para un problema dado.

entornos diferentes como problemas.

Afortunadamente es posible caracterizar los entornos de acuerdo a un conjunto de propiedades o dimensiones. Este conjunto es usado por la totalidad de la literatura a la hora de caracterizar entornos:

- **Observable.** Un entorno es **totalmente observable** cuando el agente es capaz de captar toda la información relevante para la toma de una decisión y no necesita mantener ningún modelo interno del entorno, **parcialmente observable** cuando la información obtenida es incompleta o tiene ruido y **no observable** cuando el agente no posee sensores.
- **Multiagente o. monoagente.** Un entorno es **multiagente** cuando requiere de múltiples agentes interactuando para llegar a una solución mientras que es **monoagente** cuando sólo requiere de uno para ello.
- **Determinista o. no determinista.** Si el estado del entorno actual depende totalmente del estado anterior, se dice que el entorno es **determinista**. Si no es así, se considera **no determinista o estocástico**¹⁹.
- **Episódico o. secuencial.** Un entorno en el que las acciones se dividen atómicamente donde cada una de ellas conlleva un ciclo de (percepción, decisión, acción) y sin relación una con otra se denomina episódico. Si en lugar de ello la acción del agente puede afectar a las decisiones futuras se dice que el entorno es **no episódico o secuencial**.
- **Estático o. dinámico.** Si durante la toma de decisión en entorno no cambia, se dice que el entorno es **estático**. En caso contrario, se dice que es **dinámico**.
- **Discreto o. continuo.** Esta dimensión en realidad se divide en cuatro, estado del entorno, tiempo en el entorno, percepciones y acciones. La dimensión es **discreta** cuando ésta se divide en una partición discretizada, y **continua** cuando no. Por ejemplo, en el Juego de la Vida de Conway, si se modela en un sistema multiagente, tanto el estado (i.e. tablero) como el tiempo (i.e. turnos) como las percepciones y acciones están discretizadas. Sin embargo, en un entorno de conducción automática se puede determinar que las cuatro dimensiones son continuas.
- **Conocido o. desconocido.** Un entorno es **conocido** cuando es posible determinar cuál va a ser el resultado de una acción. Si por el contrario no es posible, entonces se dice que es **desconocido**.

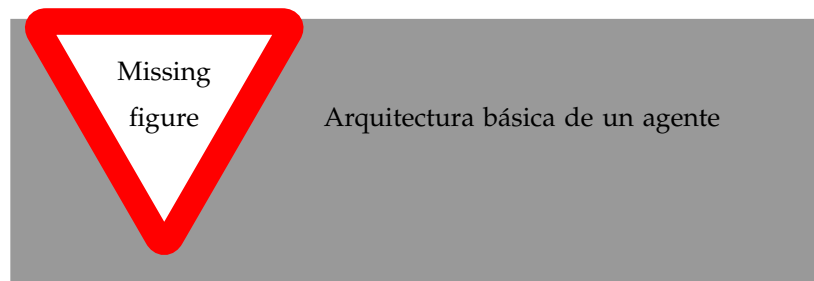
¹⁹ En general, los entornos del mundo real tienden a ser tan complejos que es imposible para un agente abarcar todos los aspectos medibles de éste. Por lo tanto, sea o no la naturaleza del entorno determinista, en general se suele suponer éste como no determinista.

Arquitecturas

Existe una serie de arquitecturas básicas o tipos de agentes que dependen principalmente de cómo perciben el entorno y de qué forma

se comportan aunque, dependiendo de los autores, las nomenclaturas, tipologías y esquemas pueden variar. Por ello, hemos decidido ofrecer una abstracción donde poner de manifiesto las partes comunes y no comunes entre arquitecturas.

Figura 10: Arquitectura básica de un agente. Aunque existen múltiples arquitecturas diferentes, todas se basan en la misma estructura. El agente percibe el entorno, lo interpreta y toma la decisión de cómo actuar sobre él.

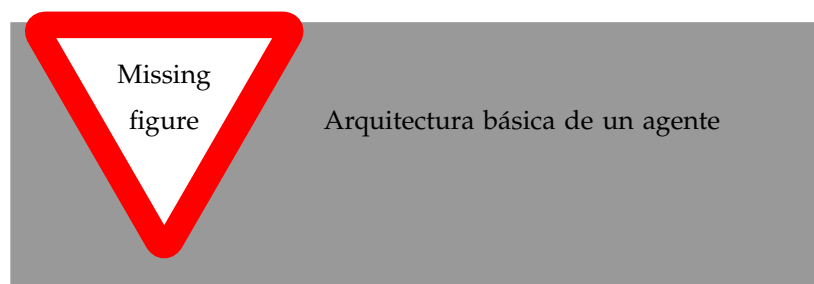


La figura 10 muestra el esquema de las partes principales de un agente. En general, toda arquitectura de agente inteligente está cortada por el mismo patrón y obedece al siguiente funcionamiento:

1. El agente, a través de sus **sensores**, percibe el entorno en el que éste se mueve.
2. De acuerdo a cómo recordamos el entorno (llamémoslo **modelo del entorno**), el agente genera una **interpretación del entorno** tal y como supone el agente que es. Esto es, percibe el entorno y, de acuerdo a sus sensaciones, lo entiende de una determinada forma.
3. Esta interpretación del entorno es pasada a un proceso de **inferencia** el cual, en función la implementación para la consecución de sus objetivos, generará una serie de acciones a realizar sobre el entorno.
4. Estas acciones serán ejecutadas sobre el entorno a través de una serie de **actuadores**, provocando probablemente una modificación en éste que será percibida de nuevo en momentos sucesivos.

La primera diferencia clave surge en la manera que se ofrece al bloque de inferencia la interpretación del entorno y genera la primera clasificación (figura 11):

Figura 11: Ilustración de la diferencia entre un agente sin modelo de entorno y uno con modelo de entorno. Cada acción realizada por el agente con modelo de entorno tiene en cuenta el estado del entorno en momentos pasados. El agente sin modelo de entorno actúa tal y como interpreta el entorno en cada momento, como si sufriese de amnesia.



- **Sin modelo de entorno.** Si el agente ofrece su interpretación del entorno directamente, sin hacer uso de información histórica sobre el entorno que se ha movido. Otras formas de denominar a estos agentes es como *agentes reactivos* o *simple-reflex agents* ([Russell et al., 2003]). Sin embargo, los términos *reactivo* o *reflex* para algunos autores se refieren a la forma de inducción de acciones a partir de percepciones, y por ello preferimos la denominación *sin modelo de entorno*.
- **Con modelo de entorno.** El agente genera su interpretación más detallada del entorno a partir de las percepciones que llegan desde los sensores y de el histórico del entorno que mantiene. Otras formas de llamarlo es *agentes con estado* o *Model-based*, pero lo hemos denominado de esta manera para diferenciar que el modelo que se mantiene en este punto pertenece únicamente al entorno.

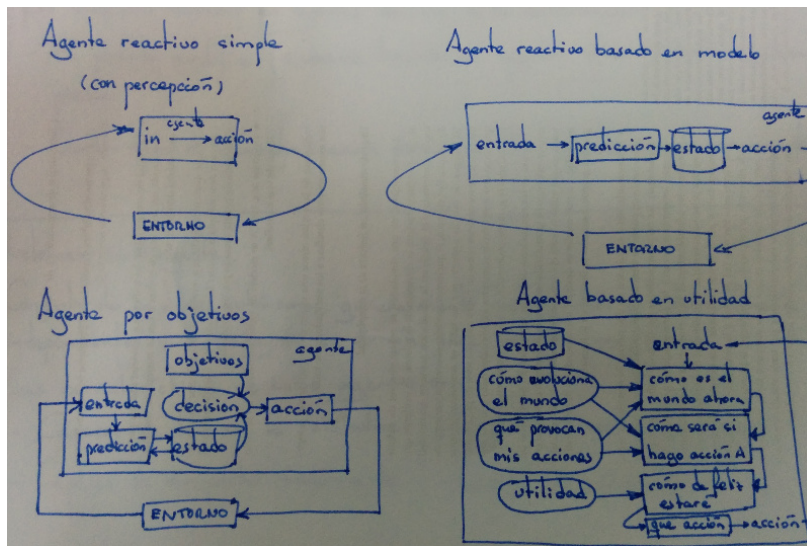


Figura 12: Distintas arquitecturas de agentes en función del comportamiento. Dependiendo de las acciones a realizar, se identifican tres tipos, los reactivos que aplican una acción sin proceso deductivo y los basados en modelo y utilidad (en algunos contextos denominados deliberativos) que basan su comportamiento en alguna forma de deducción.

La siguiente clasificación viene motivada por la forma de deducir el conjunto de acciones a ser aplicadas por parte de los sensores. En este sentido podemos identificar tres tipos distintos de agentes (figura 12):

- **Reactivos.** Son aquellos donde el uso de un proceso de razonamiento explícito es demasiado costoso para producir una conducta en un tiempo aceptable. Se suelen implementar como correspondencias (percepción \rightarrow acción) sin ningún razonamiento adicional.
- **Basados en objetivos.** Plantean una deducción de forma que determinan cuál sería el estado del entorno tras aplicar varias o todas las acciones que puede realizar. En base a los resultados, selecciona la acción que se corresponde con sus propios objetivos.
- **Basados en utilidad.** Éstos plantean una deducción similar a los basados en objetivos con la diferencia de que, mientras los primeros sólo diferencian entre entorno objetivo o no objetivo, éstos

asignan un valor (i.e. *utilidad*) a cada uno de los escenarios de entorno posibles para seleccionar el mejor (e.g. el que mayor utilidad tiene).

En la literatura se describen muchos tipos de agente, como por ejemplo los agentes BDI (Believe-Desire-Intention) o los agentes lógicos (i.e. el entorno se representa con reglas lógicas y se infiere mediante métodos como por ejemplo deducción lógica o prueba de teoremas). Sin embargo, éstos pueden definirse en los términos aquí expuestos (figuras 10, 11 y 12).

mas

Son aquellos sistemas compuestos de dos o más agentes que interactúan de alguna manera para llegar a una solución.

Cuando los agentes son inteligentes y el problema cae dentro del dominio de la IA, el ámbito de estudio es el de la , la rama dedicada a la resolución de problemas mediante procesamiento descentralizado.

Desde el punto de vista de la ingeniería de sistemas, y a pesar del aumento de complejidad, los mas, al ser sistemas inherentemente descentralizados, ofrecen múltiples ventajas frente a los sistemas centralizados tradicionales:

- Los sistemas son más robustos y fiables frente a fallos, ya que los agentes son autónomos e independientes del resto.
- La modificación del sistema se puede realizar sobre la marcha, agente a agente sin necesidad de parar el sistema al completo.
- Su diseño fuerza a desacoplar las dependencias entre agentes.
- Son inherentemente paralelizables y por tanto pueden llegar a ser más eficientes que sus homólogos centralizados. Este punto es quizá el más controvertido, ya que esta ganancia en eficiencia se puede perder rápidamente en función de la cantidad de comunicación existente entre agentes.
- Debido al nivel de complejidad alcanzado en los sistemas existentes en la actualidad, la computación se distribuye a través de múltiples sistemas, normalmente heterogéneos. La tendencia además es a la alza. La definición de los mas hace natural su implementación en este tipo de arquitecturas.

Desde el punto de vista de la IA podemos añadirles la ventaja de que permiten el estudio de conductas complejas de poblaciones a partir del comportamiento de sus elementos básicos, facilitando el estudio de modelos de teorías sobre éstos.

LA COMUNICACIÓN ENTRE AGENTES, se trata de una característica clave en un , ya que para denominarse de esta manera dos o más agentes deben interactuar (i.e. comunicarse) entre si. Esta interacción puede implementarse de diversas maneras ²⁰ y siempre toman una o las dos formas siguientes (figura 13):

- **Cooperación.** Los agentes intercambian información entre sí para llegar a una solución. Esta solución puede ser fragmentada (i.e. cada agente posee parte de la solución y se comunican para ir avanzando de forma común hacia la solución global) o poseerla uno o varios agentes que hacen uso de más agentes para ir avanzando la solución.
- **Competición.** Los agentes compiten dentro de un entorno, generalmente mediante la adquisición de recursos limitados. Un ejemplo de este tipo de sistemas multiagente puede ser aquellos sistemas de vida artificial.

²⁰ Las formas clásicas de comunicación son el de paso de mensajes, los sistemas de pizarra y la estigmergia. Para los dos primeros existen dos propuestas para estándar de lenguaje de comunicación, kqml ([Finin et al., 1994]) y acl ([Poslad, 2007]). La tercera forma de comunicación suele ser muy dependiente del problema y no se apoya en lenguajes estándares. Se trata de una forma de comunicación basada en la modificación del entorno, como la efectuada por las hormigas en la búsqueda de alimento, donde éstas dejan rastros de feromonas modificando el entorno para modificar el comportamiento del resto de la colonia.

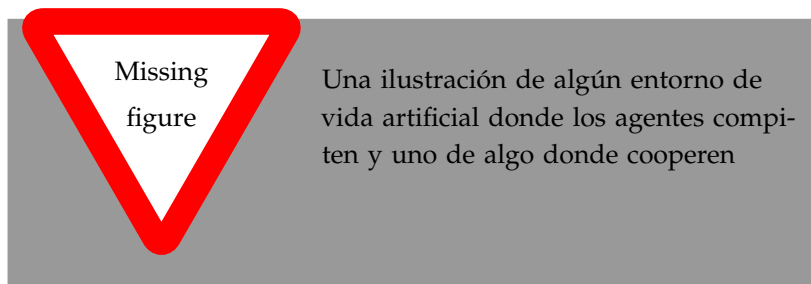


Figura 13: La comunicación entre agentes puede ser de dos tipos: *colaborativa*, donde los agentes tratan de llegar a una solución intercambiándose información y *competitiva*, donde los agentes compiten unos contra otros en un entorno.

Bibliografía

[Ful,]

[par, 2010] (2010). Directive 2010/40/eu of the european parliament and of the council of 7 july 2010 on the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport text with eea relevance. *Official Journal of the European Union*, 50:207.

[Aghabayk et al., 2013] Aghabayk, K., Forouzideh, N., and Young, W. (2013). Exploring a local linear model tree approach to car-following. *Computer-Aided Civil and Infrastructure Engineering*, 28(8):581–593.

[Ahmed, 1999] Ahmed, K. I. (1999). Modeling Drivers ' Acceleration and Lane Changing Behavior. *Transportation*, Ph.D:189.

[Al-Shihabi and Maurant, 2001] Al-Shihabi, T. and Maurant, R. R. (2001). A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators. *The 5th International Conference on Autonomous Agents.*, (June):286–291.

[Alexiadis et al., 2004] Alexiadis, V., Colyar, J., Halkias, J., Hranac, R., and McHale, G. (2004). The next generation simulation program. *ITE Journal (Institute of Transportation Engineers)*, 74(8):22–26.

[Balmer et al., 2004] Balmer, M., Cetin, N., Nagel, K., and Raney, B. (2004). Towards truly agent-based traffic and mobility simulations. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 1:60–67.

[Bando et al., 2013] Bando, T., Takenaka, K., Nagasaka, S., and Taniguchi, T. (2013). Unsupervised drive topic finding from driving behavioral data. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 177–182. IEEE.

[Barlovic et al., 1998] Barlovic, R., Santen, L., Schadschneider, A., and Schreckenberg, M. (1998). Metastable states in cellular automata for traffic flow. *Eur. Phys. J. B*, 5:793–800.

[Behrisch et al., 2011] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.

- [Bender et al., 2015] Bender, A., Agamennoni, G., Ward, J. R., Worrall, S., and Nebot, E. M. (2015). An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3325–3336.
- [Bexelius, 1968] Bexelius, S. (1968). An extended model for car-following.
- [Bingham, 2001] Bingham, E. (2001). Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research*, 131(2):232–241.
- [Brilon and Wu, 1999] Brilon, W. and Wu, N. (1999). Evaluation of cellular automata for traffic flow simulation on freeway and urban streets. *Traffic and Mobility*.
- [Casas et al., 2011] Casas, J., Perarnau, J., and Torday, A. (2011). The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach. *Procedia-Social and Behavioral Sciences*, 20:251–262.
- [Chaib-draa and Levesque, 1994] Chaib-draa, B. and Levesque, P. (1994). Hierarchical model and communication by signs, signals, and symbols in multi-agent environments. *on Modelling Autonomous Agents in a Multi- . . .*
- [Chakroborty and Kikuchi, 2003] Chakroborty, P. and Kikuchi, S. (2003). Calibrating the membership functions of the fuzzy inference system: Instantiated by car-following data. *Transportation Research Part C: Emerging Technologies*, 11(2):91–119.
- [Chan et al., 2012] Chan, K. Y., Dillon, T. S., Singh, J., and Chang, E. (2012). Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg-Marquardt Algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):644–654.
- [Chandler et al., 1958] Chandler, R. E., Herman, R., and Montroll, E. W. (1958). Traffic Dynamics: Studies in Car Following. *Operations Research*, 6(2):165–184.
- [Clymer, 2002] Clymer, J. (2002). Simulation of a vehicle traffic control network using a fuzzy classifier system. In *Proceedings 35th Annual Simulation Symposium. SS 2002*, pages 285–291. IEEE Comput. Soc.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.
- [Das and Bowles, 1999] Das, S. and Bowles, B. (1999). Simulations of highway chaos using fuzzy logic. *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397)*, pages 130–133.

- [Das et al., 1999] Das, S., Bowles, B. A., Houghland, C. R., Hunn, S. J., and Zhang, Y. (1999). Microscopic simulations of freeway traffic flow. In *Proceedings of the Thirty-Second Annual Simulation Symposium IEEE Computer Society*, pages 79–84. IEEE Comput. Soc.
- [Dia, 2002] Dia, H. (2002). An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transportation Research Part C: Emerging Technologies*, 10(5):331–349.
- [Díaz Álvarez et al., 2014] Díaz Álvarez, A., Serradilla García, F., Naranjo, J. E., Anaya, J. J., and Jiménez, F. (2014). Modeling the driving behavior of electric vehicles using smartphones and neural networks. *IEEE Intelligent Transportation Systems Magazine*, 6(3):44–53.
- [Dijkstra, 1972] Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, 15(10):859–866.
- [Dingus et al., 2006] Dingus, T. A., Klauer, S. G., Neale, V. L., Petersen, A., Lee, S., Sudweeks, J., Perez, M., Hankey, J., Ramsey, D., Gupta, S., et al. (2006). The 100-car naturalistic driving study, phase ii-results of the 100-car field experiment. Technical report.
- [Dougherty et al., 1993] Dougherty, M. S., Kirby, H. R., and Boyle, R. D. (1993). The use of neural networks to recognise and predict traffic congestion. *Traffic engineering & control*, 34(6):311–4.
- [Dresner and Stone, 2004] Dresner, K. and Stone, P. (2004). *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems : AAMAS 2004 : New York City, New York, USA : July 19-23, 2004*. IEEE Computer Society.
- [Du and Swamy, 2006] Du, K. and Swamy, M. (2006). Neural networks in a softcomputing framework.
- [Ehlert and Rothkrantz, 2001] Ehlert, P. a. M. and Rothkrantz, L. J. M. (2001). A reactive driving agent for microscopic traffic simulation. *Modelling and Simulation 2001*, pages 943–949.
- [Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. *Proceedings of the third international conference on Information and knowledge management - CIKM '94*, pages 456–463.
- [Fix and Armstrong, 1990] Fix, E. and Armstrong, H. (1990). Modeling human performance with neural networks. *1990 IJCNN International Joint Conference on Neural Networks*, pages 247–252 vol.1.
- [Fritzsche and Ag, 1994] Fritzsche, H.-t. and Ag, D.-b. (1994). A model for traffic simulation. *Traffic Engineering & Control*, 35(5):317–321.
- [Galis and Rao, 2000] Galis, a. and Rao, S. (2000). -"Application of Agent Technology to Telecommunication Management Services".

- [Gazis et al., 1959] Gazis, D. C., Herman, R., and Potts, R. B. (1959). Car-Following Theory of Steady-State Traffic Flow. *Operations Research*, 7(4):499–505.
- [Gipps, 1981] Gipps, P. G. (1981). A behavioural car-following model for computer simulation.
- [Gipps, 1986] Gipps, P. G. (1986). A model for the structure of lane-changing decisions. *Transportation Research Part B*, 20(5):403–414.
- [Gu et al., 2015] Gu, M., Billot, R., Faouzi, N.-e. E., Lyon, D., and Has-sas, S. (2015). Multi-Agent Dynamic Coupling for Cooperative Vehicles Modeling. pages 4276–4277.
- [Halati et al., 1997] Halati, A., Lieu, H., and Walker, S. (1997). CORSIM-corridor traffic simulation model. *Traffic Congestion and Traffic Safety in the*.
- [Hatipkarasulu, 2003] Hatipkarasulu, Y. (2003). A Variable Response Time Lag Module for Car Following Models Using Fuzzy Set Theory. (225).
- [Hebb, 1968] Hebb, D. (1968). The organization of behavior.
- [Hidas, 2002] Hidas, P. (2002). Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies*, 10(5-6):351–371.
- [Hinton, 2006] Hinton, G. E. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.
- [Hou et al., 2011] Hou, H., Jin, L., Niu, Q., Sun, Y., and Lu, M. (2011). Driver Intention Recognition Method Using Continuous Hidden Markov Model. *International Journal of Computational Intelligence Systems*, 4(3):386–393.
- [Hunt and Lyons, 1994] Hunt, J. G. and Lyons, G. D. (1994). Modelling dual carriageway lane changing using neural networks. *Transportation Research Part C*, 2(4):231–245.
- [Huval et al., 2015] Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., Mujica, F., Coates, A., and Ng, A. Y. (2015). An Empirical Evaluation of Deep Learning on Highway Driving. *arXiv*, pages 1–7.
- [Imprialou et al., 2016] Imprialou, M.-I. M., Quddus, M., Pitfield, D. E., and Lord, D. (2016). Re-visiting crash-speed relationships: A new perspective in crash modelling. *Accident Analysis & Prevention*, 86:173–185.
- [Jia et al., 2003] Jia, H., Juan, Z., and Ni, A. (2003). Develop a car-following model using data collected by 'five-wheel system'. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1:346–351.

- [Jin et al., 2016] Jin, J., Ma, X., Koskinen, K., Rychlik, M., and Kosonen, I. (2016). Evaluation of fuzzy intelligent traffic signal control (fits) system using traffic simulation. In *Transportation Research Board 95th Annual Meeting*, number 16-4359.
- [Jin, 2006] Jin, W.-L. (2006). A kinematic wave theory of lane-changing vehicular traffic.
- [K. et al., 1996] K., A., M., B.-A., H., K., and R., M. (1996). Models of freeway lane changing and gap acceptance behavior. *Transportation and traffic theory*, 13:501–515.
- [Kerner et al., 2008] Kerner, B., Klenov, S., and Brakemeier, A. (2008). Testbed for wireless vehicle communication: A simulation approach based on three-phase traffic theory. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 180–185. IEEE.
- [Khodayari et al., 2012] Khodayari, A., Ghaffari, A., Kazemi, R., and Braunstingl, R. (2012). A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects. In *IEEE on Systems, Man, and Cybernetics*, volume 42, pages 1440–1449.
- [Kikuchi and Chakroborty, 1992] Kikuchi, S. and Chakroborty, P. (1992). Car-following model based on fuzzy inference system. 1(1365).
- [Kiszka et al., 1985] Kiszka, J. B., Kochańska, M. E., and Sliwińska, D. S. (1985). The Influence of Some Parameters on the Accuracy of a Fuzzy Model. *Industrial Applications of Fuzzy Control*, pages 187–230.
- [Krajewicz et al., 2012] Krajewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.
- [Krajewicz et al., 2002] Krajewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility) – an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187.
- [Krauss et al., 1997] Krauss, S., Wagner, P., and Gawron, C. (1997). Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597–5602.
- [Kuge et al., 2000] Kuge, N., Yamamura, T., Shimoyama, O., and Liu, A. (2000). A Driver Behavior Recognition Method Based on a Driver Model Framework. *Structure*, 109(1dm):469–476.
- [Laval and Daganzo, 2006] Laval, J. A. and Daganzo, C. F. (2006). Lane-changing in traffic streams. *Transportation Research Part B: Methodological*, 40(3):251–264.

- [Lerner et al., 2010] Lerner, N., Jenness, J., Singer, J., Klauer, S., Lee, S., Donath, M., Manser, M., and Ward, N. (2010). An exploration of vehicle-based monitoring of novice teen drivers. *Final Report. NHTSA, Report No. DOT HS, 811:333*.
- [Lipton et al., 2015] Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
- [Liu and Li, 2013] Liu, R. and Li, X. (2013). Stability analysis of a multi-phase car-following model. *Physica A: Statistical Mechanics and its Applications*, 392(11):2660–2671.
- [Ma, 2004] Ma, X. (2004). Toward An Integrated Car-following and Lane- changing Model by A Neural Fuzzy Approach. pages 1–15.
- [Margolus, 1993] Margolus, N. (1993). CAM-8: a computer architecture based on cellular automata *.
- [Maye et al., 2011] Maye, J., Triebel, R., Spinello, L., and Siegwart, R. (2011). Bayesian on-line learning of driving behaviors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4341–4346. IEEE.
- [McCarthy et al., 1956] McCarthy, J., Minsky, M., Rochester, N., and Shannon, C. (1956). Dartmouth conference. In *Dartmouth Summer Research Conference on Artificial Intelligence*.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- [McDonald et al., 1997] McDonald, M., Wu, J., and Brackstone, M. (1997). Development of a fuzzy logic based microscopic motorway simulation model. *IEEE Conference on*.
- [Michon, 1985] Michon, J. A. (1985). A critical view of driver behavior models: what do we know, what should we do? In *Human behavior and traffic safety*, pages 485–524. Springer.
- [Minderhoud, 1999] Minderhoud, M. (1999). *Supported Driving: Impacts on Motorway Traffic Flow*. PhD thesis.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). Perceptrons.
- [Muñoz et al., 2010] Muñoz, J., Gutierrez, G., and Sanchis, A. (2010). A human-like torcs controller for the simulated car racing championship. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 473–480. IEEE.
- [Munoz et al., 2001] Munoz, L., Gomes, G., Yi, J., Toy, C., Horowitz, R., and Alvarez, L. (2001). Integrated meso-microscale traffic simulation of hierarchical ahs control architectures. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 82–87. IEEE.

- [Nagel and Schreckenberg, 1992] Nagel, K. and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic.
- [Nagel et al., 1998] Nagel, K., Wolf, D. E., Wagner, P., and Simon, P. (1998). Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437.
- [OICA, 2015] OICA (2015). Motorization rate 2014 – worldwide.
- [Osogami et al., 2012] Osogami, T., Imamichi, T., Mizuta, H., Morimura, T., Raymond, R., Suzumura, T., Takahashi, R., and Idé, T. (2012). IBM Mega Traffic Simulator.
- [Pakkenberg and Gundersen, 1997] Pakkenberg, B. and Gundersen, H. J. (1997). Neocortical neuron number in humans: effect of sex and age. *The Journal of comparative neurology*, 384(2):312–20.
- [Panwai and Dia, 2007] Panwai, S. and Dia, H. (2007). Neural agent car-following models. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):60–70.
- [Pipes, 1953] Pipes, L. (1953). An operational analysis of traffic dynamics. *Journal of applied physics*.
- [Poslad, 2007] Poslad, S. (2007). Specifying Protocols for Multi-Agent Systems Interaction. *ACM Transactions on Autonomous and Adaptive Systems*, 2(4, Article 15):25.
- [Quaassdorff et al., 2016] Quaassdorff, C., Borge, R., Pérez, J., Lumberras, J., de la Paz, D., and de Andrés, J. M. (2016). Microscale traffic simulation and emission estimation in a heavily trafficked roundabout in madrid (spain). *Science of The Total Environment*, 566:416–427.
- [Ramón and Cajal, 1904] Ramón, S. and Cajal, S. (1904). *Textura del Sistema Nervioso del Hombre y de los Vertebrados*, volume 2. Madrid Nicolas Moya.
- [Rasmussen, 1986] Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. North-Holland.
- [Reuschel, 1950] Reuschel, A. (1950). Fahrzeugbewegungen in der Kolonne. *Osterr. Ing. Archiv*, 4(1):193–215.
- [Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, DTIC Document.
- [Russell et al., 2003] Russell, S. J., Norvig, P., Canny, J. F., and Malik, Jitendra M, E. D. D. (2003). *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River.
- [Sasoh and Ohara, 2002] Sasoh, A. and Ohara, T. (2002). Shock Wave Relation Containing Lane Change Source Term for Two-Lane Traffic Flow. *Journal of the Physical Society of Japan*, 71(9):2339–2347.

- [Sekizawa et al., 2007] Sekizawa, S., Inagaki, S., Suzuki, T., Hayakawa, S., Tsuchida, N., Tsuda, T., and Fujinami, H. (2007). Modeling and recognition of driving behavior based on stochastic switched ARX model. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):593–606.
- [Shiose et al., 2001] Shiose, T., Onitsuka, T., and Taura, T. (2001). Effective information provision for relieving traffic congestion. In *Proceedings Fourth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2001*, pages 138–142. IEEE.
- [Simonelli et al., 2009] Simonelli, F., Bifulco, G. N., and Martinis, V. D. (2009). Human-Like Adaptive Cruise Control Systems through a Learning Machine Approach. *Applications of Soft Computing*, pages 240–249.
- [Sparmann, 1978] Sparmann, U. (1978). *Spurwechselvorgänge auf zweispurigen BAB-Richtungsfahrbahnen*.
- [Suzumura and Kanezashi, 2012] Suzumura, T. and Kanezashi, H. (2012). Highly scalable x10-based agent simulation platform and its application to large-scale traffic simulation. *Proceedings of the 2012 IEEE/ACM 16th*.
- [Toledo et al., 2003] Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. (2003). Modeling Integrated Lane-Changing Behavior. *Transportation Research Record*, 1857(1):30–38.
- [Toledo et al., 2007] Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. (2007). Integrated driving behavior modeling. *Transportation Research Part C: Emerging Technologies*, 15(2):96–112.
- [Tordeux et al., 2011] Tordeux, A., Lassarre, S., and Roussignol, M. (2011). A study of the emergence of kinematic waves in targeted state car-following models of traffic. <http://cybergeog.revues.org>.
- [Trask ANDREWTRASK et al.,] Trask ANDREWTRASK, A., Gilmore DAVIDGILMORE, D., and Russell MATTHEWRUSSELL, M. Modeling Order in Neural Word Embeddings at Scale.
- [Turing, 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.
- [Van Hoorn et al., 2009] Van Hoorn, N., Togelius, J., Wierstra, D., and Schmidhuber, J. (2009). Robust player imitation using multiobjective evolution. In *2009 IEEE Congress on Evolutionary Computation*, pages 652–659. IEEE.
- [Van Ly et al., 2013] Van Ly, M., Martin, S., and Trivedi, M. M. (2013). Driver classification and driving style recognition using inertial sensors. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1040–1045. IEEE.

- [Wagner et al., 1997] Wagner, P., Nagel, K., and Wolf, D. E. (1997). Realistic multi-lane traffic rules for cellular automata. *Physica A: Statistical Mechanics and its Applications*, 234(3-4):687-698.
- [Wegener et al., 2008] Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., and Hubaux, J.-P. (2008). TraCI. *Proceedings of the 11th communications and networking simulation symposium on - CNS '08*, (August 2016):155.
- [Wei et al., 2000] Wei, H., Lee, J., Li, Q., and Li, C. (2000). Observation-Based Lane-Vehicle Assignment Hierarchy: Microscopic Simulation on Urban Street Network. *Transportation Research Record*, 1710(1):96-103.
- [Weidemann and Reiter, 1992] Weidemann, R. and Reiter, U. (1992). Microscopic Traffic Simulation, The Simulation System-Mission. *University Karlsruhe, Germany*, 2:54.
- [Wiedemann, 1974] Wiedemann, R. (1974). Simulation des Straßenverkehrsflusses. Technical report.
- [Wooldridge et al., 1995] Wooldridge, M., Jennings, N. R., Adorni, G., Poggi, A., Allen, J. F., Bates, J., Bell, J., BELNAP, N., PERLOFF, M., Bratman, M. E., Israel, D. J., Pollack, M. E., Brooks, R., Brooks, R. A., Bussmann, S., Demazeau, Y., Castelfranchi, C., Chaib-Draa, B., Moulin, B., Mandiau, R., Millot, P., Chang, E., Chapman, D., Chellas, B. F., Cohen, P. R., Levesque, H. J., Cohen, P. R., Perrault, C. R., Cutkosky, M., Engelmores, R., Fikes, R., Genesereth, M., Gruber, T., Mark, W., Tenenbaum, J., Weber, J., Downs, J., Reichgelt, H., Emerson, E. A., Halpern, J. Y., Fagin, R., Halpern, J. Y., Vardi, M. Y., Fisher, M., Gasser, L., Gasser, L., Braganza, C., Herman, N., Genesereth, M. R., Ketchpel, S. P., Georgeff, M. P., Greif, I., Guha, R. V., Lenat, D. B., Haas, A. R., Halpern, J. Y., Halpern, J. Y., Moses, Y., Halpern, J. Y., Vardi, M. Y., Hayes-Roth, B., Hewitt, C., Huang, J., Jennings, N. R., Fox, J., Jennings, N. R., JENNINGS, N. R., Jennings, N., Varga, L., Aarnts, R., Fuchs, J., Skarek, P., Kaelbling, L. P., Kraus, S., Lehmann, D., Kripke, S. A., Maes, P., Maes, P., McCabe, F. G., Clark, K. L., Mukhopadhyay, U., Stephens, L. M., Huhns, M. N., Bonnell, R. D., Müller, J. P., Pischel, M., Thiel, M., Newell, A., Simon, H. A., Norman, T. J., Long, D., PAPAOGLOU, M. P., LAUFMANN, S. C., SELLIS, T. K., Perlis, D., Perlis, D., Perloff, M., Poggi, A., Reichgelt, H., Sacerdoti, E. D., Searle, J. R., Shoham, Y., Thomas, B., Shoham, Y., Schwartz, A., Kraus, S., Thomason, R. H., Varga, L., Jennings, N. R., Cockburn, D., Vere, S., Bickmore, T., Wavish, P., Graham, M., Weerasooriya, D., Rao, A., Ramamohanarao, K., and Wooldridge, M. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(02):115.
- [Wu et al., 2003] Wu, J., Brackstone, M., and McDonald, M. (2003). The validation of a microscopic simulation model: A methodological case study. *Transportation Research Part C: Emerging Technologies*, 11(6):463-479.

- [Wymann et al., 2013] Wymann, B., Espi, E., Guionneau, C., Dimitrakakis, C., and Sumner, A. (2013). TORCS : The open racing car simulator e. at <http://torcs. . . .>, pages 1–4.
- [y Cajal, 1888] y Cajal, S. R. (1888). *Estructura de los centros nerviosos de las aves*.
- [Yang and Koutsopoulos, 1996] Yang, Q. and Koutsopoulos, H. N. (1996). A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3 PART C):113–129.
- [Zhang et al., 2011] Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., and Chen, C. (2011). Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639.
- [Zheng et al., 2013] Zheng, J., Suzuki, K., and Fujita, M. (2013). Car-following behavior with instantaneous driver-vehicle reaction delay: A neural-network-based methodology. *Transportation Research Part C: Emerging Technologies*, 36:339–351.
- [Zheng and McDonald, 2005] Zheng, P. and McDonald, M. (2005). Application of Fuzzy Systems in the Car-Following Behaviour Analysis. pages 782–791. Springer Berlin Heidelberg.

Cómo citar esta tesis

Si deseas citar esta tesis, lo primero gracias. Me alegro de que te sirva para tu investigación. Si lo deseas, incluye el siguiente código en bibtex:

```
@phdthesis{blazaid20167,  
  author = {Alberto Díaz Álvarez},  
  abstract = {XXX},  
  pages = {XXX},  
  title = {Modelado de comportamiento de conductores con técnicas de Inteligencia Computacional},  
  url = {XXX},  
  year = {12 de febrero de 2018}  
}
```

Acerca del código fuente

La presente tesis lleva consigo numerosas horas de programación y, por tanto, muchísimas líneas de código. Éste se encuentra en formato electrónico como datos adjuntos a la memoria y no como capítulo o anexo a ésta, una forma más manejable para su consulta y a la vez respetuosa con el medio ambiente. No obstante sí es posible que existan pequeños fragmentos de código para apoyar explicaciones. En caso de necesitar los fuentes y no estar disponibles los datos anexos a la memoria, puedes contactar directamente conmigo en alberto.diaz@upm.es.