

¿Cuándo es publicado el artículo Computing Machinery and Intelligence?

1950 y propone la prueba de turing

¿Qué es la prueba de turing?

Es una herramienta de evaluación de la capacidad de una máquina para exhibir un comportamiento inteligente similar al de un ser humano o indistinguible de este

¿En qué nos basamos para decir que algo es inteligente?

En una persona que se comunica con un dispositivo informático con una entidad, dicha entidad puede ser humana o máquina inteligente. Si después de una conversación la persona no es capaz de distinguir si la entidad es humana o máquina, entonces, en caso de que sea máquina, la podemos considerar inteligente.

¿Cuáles son las características que debe tener una máquina que apruebe el test de Turing? (RRAR)

- Reconocimiento de lenguaje natural
- Razonamiento
- Aprendizaje
- Representación del conocimiento

¿Qué incluye la prueba de turing?

Comunicación usando cámara de video/sonido y el paso de objetos entre las entidades a través de una compuerta.

¿Qué tipo de sistemas hay?

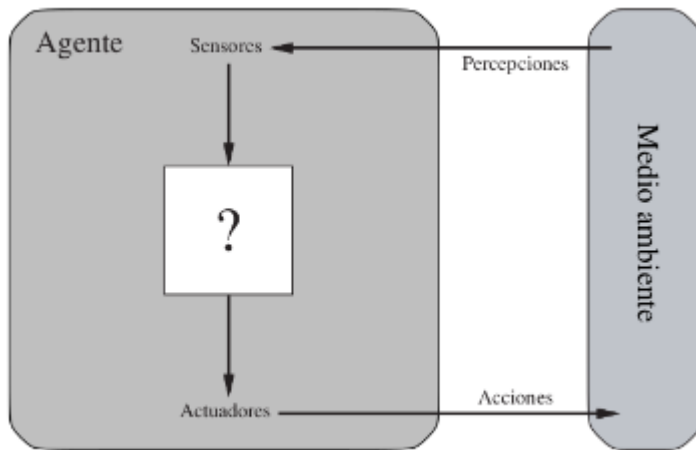
Sistemas que piensan como humanos, que piensan racionalmente, que actúan como humanos y que actúan racionalmente - agentes.

¿Qué es un agente inteligente?

Los actos de un agente inteligente deben basarse en razonamiento y en las conclusiones obtenidas a partir de la información que se posee.

Toman la decisión más conveniente en base a la información y al tiempo del que disponen.

Grupo de agentes, cada uno con capacidades diferentes, y que trabajan juntos colaborando para resolver un problema.



Partes de un agente inteligente (MAPS SFP)

- Medio ambiente
- Sensores
- Actuadores
- Percepción
- Secuencia de percepciones
- Función de agente
- Programa del agente

Agente racional

Es aquel que hace lo correcto. Hacer lo correcto es mejor que hacer algo incorrecto, pero ¿qué significa hacer lo correcto? Como primera aproximación, se puede decir que lo correcto es aquello que permite al agente obtener un resultado mejor.

Las **medidas de rendimiento** son **criterios** que **determinan el éxito** en el comportamiento del agente. Al situar un agente en el medio, este genera una secuencia de acciones dependiendo de las percepciones que recibe.

La **racionalidad** en un momento determinado depende de cuatro **factores**:

1. La **medida** de rendimiento que define el criterio de éxito
2. El **conocimiento** del medio en el que habita el agente
3. Las **acciones** que el agente puede llevar a cabo
4. La **secuencia** de percepciones del agente hasta este momento

En cada posible secuencia de percepciones, un agente racional deberá emprender aquella acción que supuestamente maximice su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en el conocimiento que el agente mantiene almacenado.

Propiedades de los entornos

Observabilidad

- Totalmente observable: los sensores del agente le proporcionan acceso al estado completo del entorno. Detectando todos los aspectos relevantes en la toma de decisiones.
- Parcialmente observable: debido al ruido y a sensores poco exactos.

Determinismo

- Determinista: si el siguiente estado del entorno está totalmente determinado por su estado actual y la acción ejecutada por el agente.
- Estocástico: no determinista.
- Entorno estratégico: medio determinista excepto para las acciones de otros agentes.

Periodicidad

- Episódico: cuando la experiencia del agente se divide en episodios atómicos independientes, donde cada episodio consiste en la percepción del agente y la realización de una acción posterior.
- Secuencialidad: no existe división y una decisión presente puede afectar a decisiones futuras.

Dinamismo

- Estático: el entorno no cambia mientras el agente está deliberando.
- Dinámico: el entorno sí cambia
- Semi-dinámico: el entorno no cambia con el paso del tiempo, pero el rendimiento del agente cambia.

Discretitud

- Discreto: el entorno tiene un número finito de estados distintos
- Continuo: no es posible enumerar los estados

Individualidad

- Individual: un solo agente resolviendo un problema
- Multiagente: varios compitiendo o cooperando

Agentes y búsqueda

Los agentes deben **maximizar su medida de rendimiento**. Se puede lograr si se puede definir un objetivo y tratar de satisfacerlo.

Los objetivos ayudan a organizar el comportamiento limitando las metas que intenta alcanzar el agente.

Lo importante es la **formulación del objetivo**, basado en la situación actual y la medida de rendimiento del agente.

El proceso de **examinar diferentes secuencias posibles de acciones que permitan cumplir con el objetivo**, y entonces **escoger la mejor secuencia** (maximizar la medida de rendimiento) se llama **búsqueda**.

Estado inicial

El estado en el que inicia el agente. Descripción de acciones disponibles para el agente.

Test objetivo

Determina si un estado es un estado objetivo

Función costo de camino

Permite calcular un valor numérico a cada posible solución que refleje la medida de rendimiento.

Algoritmo genético

Es una técnica de búsqueda basada en la evolución. De la evolución tomamos la reproducción de individuos y mutaciones.

Se dice que es una técnica de aprendizaje ya que permite evaluar posibles soluciones, descartar las que no son tan buenas soluciones y recombinar las mejores para encontrar la solución óptima, o una solución aceptable.

Elementos

- Genes
- Cromosomas (Individuos)
- Población
- Selección
- Reproducción o cruce

- Función de aptitud
- Criterio de finalización

Estructura de un Algoritmo genético

- Una población inicial con individuos creados al azar
- Evaluar la población con la función de aptitud
- Seleccionar los mejores individuos para la reproducción
- Reproducir individuos y obtener hijos que puedan resolver el problema
- Introducción de mutaciones en algunos individuos.
- Se finaliza cuando se cumple el criterio de finalización.

Representación de genes

Es importante decidir la forma que usaremos para representar las soluciones, esto implica la forma en representar cada gen de los individuos y su significado.

Representación binaria

Es una representación simple y de las más utilizadas. Cada gen puede tomar el valor 1 o 0 y el individuo se representa por una cadena binaria.

Representación de valor entero

Para genes de valores discretos, no siempre podemos limitar el espacio de booleanos.

Representación de valor decimal

Para los problemas en los que queremos definir los genes utilizando variables continuas en lugar de discretas, la representación de valor decimal es la más natural.

Representación de permutación

En muchos problemas, la solución está representada por un orden de elementos. En tales casos, la representación por permutación es la más adecuada.

Población

La población es importante ya que de ella salen las soluciones. La diversidad de la población debe mantenerse de lo contrario podría conducir a una convergencia prematura.

El tamaño de la población no debe mantenerse muy grande, ya que puede hacer que un algoritmo genético se desacelere.

Una población pequeña podría no ser suficiente para que la reproducción genere buenas soluciones.

Por lo tanto, el tamaño óptimo de la población debe decidirse por ensayo y error.

Modelos de población

Estado estable

Se generan uno o dos hijos en cada iteración y reemplazan a uno o dos individuos de la población.

Generacional

Se generan n descendientes, donde n es el tamaño de la población, y toda la población se reemplaza por la nueva al final de la iteración.

Función de aptitud

Es una función que toma a un individuo como entrada o indica que tan *buena* solución puede ser dicho individuo.

El cálculo de la función se realiza muchas veces por lo que debe ser un cálculo rápido.

Cuando se busca maximizar o minimizar, la mayoría de los casos la función de aptitud y la función objetivo son la misma.

Pero para problemas más complejos con múltiples objetivos se pueden desarrollar varias funciones de aptitud

Una función de aptitud debe poseer las siguientes características:

- Debe ser lo suficientemente rápida para calcular
- Debe medir cuantitativamente qué tan apta es una solución dada o qué tan aptos son los individuos para reproducirse.

Selección

Es el proceso de seleccionar a los padres que se reproducen y se recombinan para crear descendientes para la próxima generación.

La selección de los padres es muy importante para la tasa de convergencia del algoritmo genético, ya que los mejores padres llevan a soluciones mejores y más adecuadas.

Se debe tener cuidado para evitar que una solución extremadamente adecuada se apodere de toda la población en unas pocas generaciones, lo que conduce a una pérdida de diversidad.

El apoderamiento de toda la población por una solución extremadamente adecuada se conoce como **convergencia prematura** y es una condición indeseable en un algoritmo genético.

Selección por ruleta

Es una forma de selección proporcional a la aptitud en la que la probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores.

Simula una ruleta donde los elementos más aptos poseen una sección mayor de la ruleta.

Es simple pero ineficiente. Posee complejidad $O(n^2)$. Presenta el problema de que el individuo menos apto puede ser seleccionado más de una vez.

Es de notar que este método no sirve cuando los valores de la función de aptitud pueden ser valores negativos.

$S = \text{suma de los valores de aptitud}$

- Repetir n veces (n es el tamaño de la población)
- Generar un número aleatorio A entre 0 y S
- Recorrer secuencialmente los individuos, sumando los valores de aptitud, hasta que la suma sea mayor o igual a A
- El individuo que haga que la suma exceda el límite A es el seleccionado.

Selección por torneo (k-way)

Se eligen subgrupos de cantidad k (normalmente de $k = 2$ individuos) de individuos de la población y los miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.

Es fácil de paralelizar y sirve para valores negativos.

Pasos:

- Escoger un número de k de individuos (generalmente 2) de forma aleatoria
- Compararlos en base a su aptitud
- El ganador del "torneo" es el individuo más apto.
- Repetir un total de n veces para seleccionar n padres. $n = \text{población}$.

Selección por rango

También funciona con números negativos y es usado cuando los individuos tienen valores de aptitud muy cercanos (esto pasa usualmente al final del algoritmo genético)

A cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este rango, en lugar de las diferencias absolutas en aptitud.

Reproducción o cruce

Cruce en un punto

Se selecciona un punto al azar n que representa a un gen y se generan dos hijos al combinar de 0 a n genes del padre A y de n a la longitud de genes del padre B.

Cruce multipunto

Se seleccionan x puntos de cruce, se separan los genes de los padres en grupos que representen los puntos de cruce, y luego se combinan los grupos de los padres.

Mutación

Representa un pequeño cambio aleatorio en algún gen de cierto individuo dando como resultado una nueva solución. Se usa para introducir y mantener diversidad genética en la población

Las mutaciones deben aparecer con probabilidad muy baja, ya que una cantidad grande de mutaciones reduce al algoritmo a una búsqueda aleatoria.

Mutación bit-flip

Se selecciona uno o varios bits y se invierten

Random resetting

Es una extensión de bit flip, pero con genes enteros. Se escoge un gen aleatorio y se cambia su valor por un valor aleatorio, dentro de los permitidos para el gen.

Mutación de intercambio

Seleccionamos dos genes de un mismo individuo y los intercambiamos uno con otro.

Mutación scramble

Se selecciona un subset de genes y se *revuelven* aleatoriamente.

Mutación de inversión

Se selecciona un subset de genes y se reordenan de forma inversa.

Supervivientes para la nueva generación

En un **modelo de población generacional**, la población es reemplazada por los individuos hijos.

Esta es una de las más usadas.

En un **modelo de población elitista**, la nueva población se selecciona en base a edad o a mejor aptitud.

Selección por aptitud

Los individuos menos aptos de la generación actual son reemplazados por los hijos más aptos. La selección de los menos aptos se puede hacer usando alguno de los métodos de selección.

La decisión de cuántos individuos reemplazar puede ser arbitraria.

Condición de finalización

Es importante determinar el criterio para dar por finalizada la ejecución de un algoritmo genético.

Suele suceder que el algoritmo genético mejora las soluciones en las primeras generaciones para luego ralentizarse en el mejoramiento de las soluciones.

Generalmente se requiere de una condición de finalización que permite encontrar una solución muy cercana a la óptima.

Pueden ser:

1. Criterio por cantidad de generaciones
2. Criterio por porcentaje de efectividad de la solución o por rango de valor
3. Criterio por estancamiento en la población
4. Criterio cuando se alcance un valor específico.

Redes neuronales

Redes interconectadas

Cada neurona recibe estímulos eléctricos de otras neuronas. La neurona procesa esos estímulos y en ciertas ocasiones se activa y dispara para estimular a otras neuronas a las que está conectada.

Las conexiones necesarias entre neuronas se hacen más fuertes. Se llegan a acostumbrar a que si alguna se dispara las otras también deben dispararse, casi en automático.

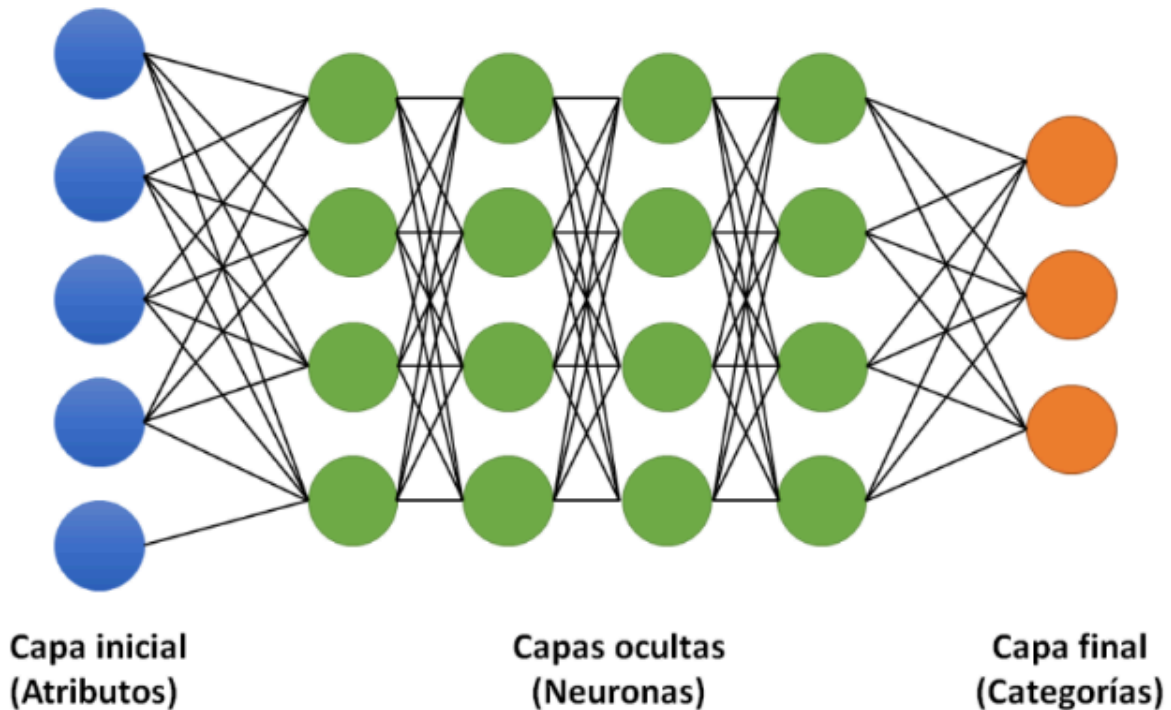
¿Cómo tomamos una decisión?

- Factores de entrada para apoyar la decisión
- Evaluar los factores y tomar la decisión
- Iniciar todo para conseguir una decisión correcta

Perceptrón simple

Consta de:

- Factores o entradas
- Pesos de factores
- Función de activación
- Umbral
- Sesgo: lo inverso al umbral



Cada perceptrón puede ser ajustado *manualmente* a nivel de pesos y sesgo, pero una red neuronal por lo general tiene miles o millones de perceptrones y cientos de capas.

En la mayoría de redes neuronales con valores grandes en los factores puede generar un efecto mariposa no deseado.

Aprendizaje supervisado

1. Iniciar los pesos y umbral (sesgo) de forma aleatoria
2. Aplicar las entradas a la red
3. Calcular la salida z
4. Comparar el resultado obtenido con el deseado. A la diferencia entre ambos se llama "error" e
5. Modificar los pesos y umbral según el error obtenido
6. Repetir hasta que el error esté dentro de un rango aceptable ($<1\%$)

Necesitamos un factor de aprendizaje λ , el cual indica la velocidad de aprendizaje. Un valor alto puede hacer que no se ajusten los pesos correctamente, y un valor bajo puede hacer que se tarde mucho el aprendizaje.

El valor generalmente usado es 0.2

Redes multicapa

Sin función de activación, la red neuronal solo puede resolver problemas lineales. Algo no deseado.

Dicha función junto con múltiples capas, muchas neuronas y suficiente entrenamiento puede ajustarse a funciones complejas.



STEP FUNCTION

Funcion $g(x)$: 0 if $x < 0$

1 if $x \geq 0$

Derivada $g'(x)$: 0

Rango: $-\infty, \infty$

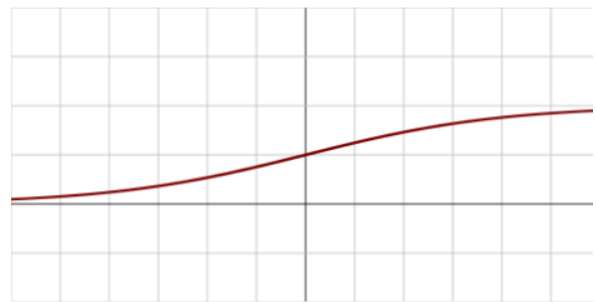


SIGMOIDE O LOGISTICA (σ)

Funcion $g(x)$: $1 / (1 + e^{-x})$

Derivada $g'(x)$: $g(x) (1 - g(x))$

Rango: 0, 1



Buena para clasificación binaria

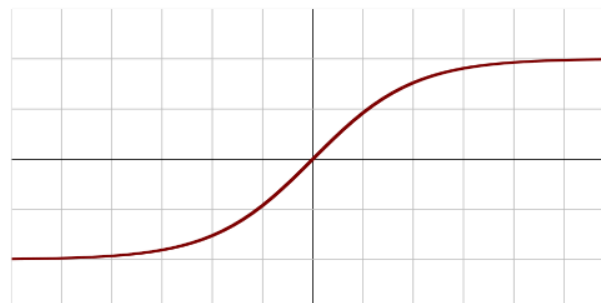


TANGENTE HIPERBOLICA (\tanh)

Funcion $g(x)$: $(e^x - e^{-x}) / (e^x + e^{-x})$

Derivada $g'(x)$: $1 - g(x)^2$

Rango: -1, 1



Aprendizaje mas rapido pero costosa al calcular



RECTIFIED LINEAR UNIT: ReLU

Funcion $g(x)$: 0 if $x \leq 0$
 x if $x > 0$
 $= \max(x, 0)$

Derivada $g'(x)$: 0 if $x < 0$
 1 if $x > 0$

Rango: $0, \infty$



Sin los problemas de las anteriores pero genera neuronas muertas

Backpropagation

El aprendizaje supervisado para una red neuronal con capas ocultas utiliza el algoritmo de backpropagation.

Consiste en 4 pasos:

1. Forward pass: las entradas son ingresadas en la capa de entrada, se multiplican por los pesos y suma el sesgo, luego se pasa el resultado a la función de activación y el resultado se pasa a la capa siguiente.
2. La salida de la red se compara con el valor esperado, se calcula la diferencia $e = (y - z)$
3. Backpropagation
4. Actualización de los pesos.