Paradigmas de la inteligencia artificial

Para esta investigación hemos decidió elegir 2 paradigmas de la inteligencia artificial, los cuales son enfoque conexionista y paradigma simbólico, se mostrarán y explicarán 2 ejemplos utilizando estos paradigmas.

Enfoque conexionista

La inteligencia artificial conexionista, también conocida como IA basada en redes neuronales, se basa en la idea de que el procesamiento de la información se puede modelar utilizando redes de neuronas artificiales.

En este enfoque, los modelos de inteligencia artificial están compuestos por unidades de procesamiento interconectadas, que se asemejan a las neuronas del cerebro humano. Estas unidades de procesamiento, o neuronas artificiales, reciben entradas, realizan cálculos y generan salidas. A medida que se entrenan, las conexiones entre las neuronas se ajustan para mejorar el rendimiento del modelo.

Las redes neuronales artificiales han demostrado ser eficientes en el reconocimiento de patrones, el procesamiento de imágenes y el aprendizaje automático. A diferencia de la IA simbólica, la IA conexionista puede manejar información incierta y aprender de manera autónoma a partir de ejemplos. (iccsi, n.d.)

Como primer ejemplo que es basado en el paradigma conexionista hemos tomado como referencia un video en YouTube llamado "**Tu primera red neuronal en Python y Tensor Flow**" del canal ringa Tech se puede consultar el video donde se explica el ejemplo en el siguiente enlace https://www.youtube.com/watch?v=iX_on3VxZzk, el ejemplo lo hice en jupyter notebook, en los siguientes párrafos describiré que es lo que hace el programa conforme al entendimiento que tuve de la investigación y del video visto. Así como se anexaran capturas de pantalla de los pasos que hice en el programa

Descripción

El ejemplo utiliza una red neuronal simple para aprender la relación matemática entre temperaturas en grados Celsius y Fahrenheit. Este modelo se basa en el paradigma del conexionismo, que busca emular el funcionamiento del cerebro humano mediante redes de nodos interconectados (neuronas artificiales). El objetivo es entrenar una red neuronal para predecir temperaturas en Fahrenheit a partir de valores en Celsius, sin necesidad de programar explícitamente la fórmula.

Datos de entrenamiento: Se proporcionan dos arreglos de NumPy con valores de Celsius y sus equivalentes en Fahrenheit

Modelo: Se construye una red neuronal simple con una sola capa densa (Dense) de una neurona, utilizando TensorFlow y Keras. Esta capa tiene un solo peso (pendiente) y un sesgo (intercepción).

Definición de los datos de entrenamiento

Vamos a crear dos arreglos NumPy que representen temperaturas en Celsius y sus equivalentes en Fahrenheit. Estos datos se usarán para entrenar un modelo de aprendizaje automático que convierta temperaturas de Celsius a Fahrenheit.

```
celcius = np.array([-40,-10,0,8,15,22,38], dtype = float)
fahrenheit =  np.array([-40,14,32,46,59,72,100], dtype=float)
```

Creación del modelo de red neuronal

Vamos a construir un modelo simple de regresión lineal usando TensorFlow/Keras. Este modelo tendrá una sola capa densa que aprenderá a convertir temperaturas de Celsius a Fahrenheit basándose en los datos proporcionados.

```
capa = tf.keras.layers.Dense(units=1, input_shape=[1])
modelo = tf.keras.Sequential([capa])
```

Entrenamiento: El modelo se entrena durante 1000 épocas utilizando el optimizador Adam y la función de pérdida de error cuadrático medio (mean_squared_error), ajustando los pesos para minimizar la diferencia entre las predicciones y los valores reales.

Compilación del modelo

En esta sección, configuramos el modelo para el entrenamiento definiendo el optimizador y la función de pérdida. Esto prepara el modelo para aprender la relación entre Celsius y Fahrenheit.

```
modelo.compile(
    optimizer = tf.keras.optimizers.Adam(0.1),
    loss = 'mean_squared_error'
)
```

Entrenamiento del modelo

Ahora entrenaremos el modelo con los datos de temperaturas en Celsius y Fahrenheit. Este proceso ajustará los pesos de la capa densa para que el modelo aprenda a predecir Fahrenheit a partir de Celsius.

```
print ("Comenzando entrenamiento.....")
historial = modelo.fit(celcius, fahrenheit , epochs=1000, verbose=False)
print("Modelo entrenando!")
```

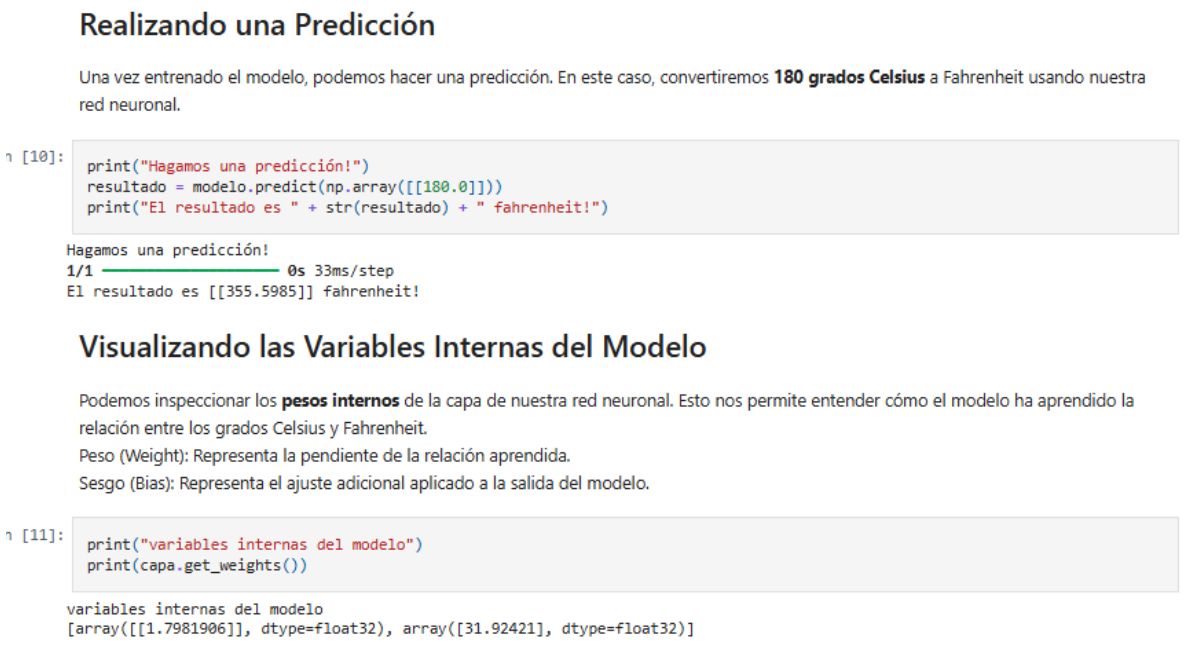
```
Comenzando entrenamiento.....
Modelo entrenando!
```

Predicción: Una vez entrenado, el modelo predice el valor en Fahrenheit para 180°C, resultando en aproximadamente 355.6°F (cercano al valor real de 356°F).

Análisis: Se visualiza la disminución de la pérdida durante el entrenamiento y se inspeccionan los pesos internos del modelo (pendiente ≈ 1.8 y sesgo ≈ 31.9)

que aproximan la fórmula que es Formula = 9/5 C + 32

9/5 es igual a 1.8 y 32 sería el sesgo



Realizando una Predicción

Una vez entrenado el modelo, podemos hacer una predicción. En este caso, convertiremos **180 grados Celsius** a Fahrenheit usando nuestra red neuronal.

[10]:
```
print("Hagamos una predicción!")
resultado = modelo.predict(np.array([[180.0]]))
print("El resultado es " + str(resultado) + " fahrenheit!")
```

```
Hagamos una predicción!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
El resultado es [[355.5985]] fahrenheit!
```

Visualizando las Variables Internas del Modelo

Podemos inspeccionar los **pesos internos** de la capa de nuestra red neuronal. Esto nos permite entender cómo el modelo ha aprendido la relación entre los grados Celsius y Fahrenheit.
Peso (Weight): Representa la pendiente de la relación aprendida.
Sesgo (Bias): Representa el ajuste adicional aplicado a la salida del modelo.

[11]:
```
print("variables internas del modelo")
print(capa.get_weights())
```

```
variables internas del modelo
[array([[1.7981906]], dtype=float32), array([31.92421], dtype=float32)]
```

Estas capturas son tomadas del repositorio de la tarea6 https://github.com/BryantTrujilloAcosta/Inteligencia-Artificial/blob/main/UNIDAD2/Tarea_6/Conversor_grados.ipynb

Bueno este fue un ejemplo muy básico y decidimos hacerlo para que sirva como el titulo del video del autor decía mi primera red neuronal. A continuación, nosotros decidimos a partir de este ejemplo mostrar como es que se usa el enfoque conexionista, las limitaciones y beneficios con base al ejemplo hecho.

Aplicación del Conexionismo en el Ejemplo

- Unidades: Una neurona procesa Celsius para predecir Fahrenheit mediante una combinación lineal (peso y sesgo).
- Conexiones: El peso y sesgo se ajustan durante el entrenamiento con el optimizador Adam.
- Aprendizaje Supervisado: Usa ejemplos (Celsius-Fahrenheit) para minimizar el error.
- Distribución del Conocimiento: La relación se aprende en los pesos, no en reglas explícitas.
- Generalización: Predice valores no vistos (ej. 180°C ≈ 355.6°F).
- Resumen: Emula el aprendizaje humano al inferir patrones de datos sin reglas predefinidas.

Limitaciones y Beneficios del Conexionismo en el Ejemplo

- **Sobre complejidad:** Red neuronal innecesaria para un problema simple con fórmula conocida
- **Dependencia de datos:** Requiere datos suficientes; con solo 7 pares, la precisión puede ser limitada.
- **Falta de precisión:** Aproxima la relación (ej. 355.6°F vs. 356°F real para 180°C).
- **Costo computacional:** 1000 épocas consumen recursos, injustificados para un problema simple.
- **Falta de interpretabilidad:** Pesos y sesgo no explican claramente el aprendizaje, a diferencia de la fórmula.
- **Limitación a linealidad:** Solo modela relaciones lineales; no sirve para problemas no lineales sin ajustes.

Paradigma simbólico

La IA simbólica es un modelo de investigación de la inteligencia artificial basado en la creación de representaciones de alto nivel ("legibles para el ser humano") de los problemas, la lógica y el conocimiento.

este enfoque permite a las empresas crear reglas que extraigan los elementos de un documento y lo asignen a una categoría. Estas reglas pueden escribirse de forma manual o generarse automáticamente (con la posibilidad de validarlas manualmente). (www.expert.ai).

Para el ejemplo sobre este paradigma simbólico se explicará acerca de Diagnóstico médico con MYCIN

es un sistema experto creado a partir de una serie de reglas causa-efecto, su base de datos consta de alrededor de unas 500 reglas. Para que el programa funcione, es necesario que un sujeto responda una serie de preguntas cuyas respuestas solo pueden ser *"si" y "no"*. Una vez obtenido el diagnóstico muestra la lista de las posibles bacterias culpables de la dolencia además de indicar su índice de confiabilidad. Su comportamiento es muy similar al de un doctor de verdad, de hecho, también es capaz de explicar cómo llegó a su conclusión y de recetar los medicamentos necesarios para acabar con las bacterias. (Azcoitia, n.d.)

El sistema funcionaba de la siguiente manera

La base de conocimiento de MYCIN es una base de datos que posee información y unas n reglas específicas sobre una materia determinada. Por Ejemplo, si la materia especifica de MYCIN fueran los Calculos Biliares y Renales entonces la estructura de la base de conocimientos debería tener ser la siguiente. (Rodrigo)

Tabla 1.

Objeto	Reglas	Atributo
Calculos Biliares	ha	Sido operado
	tiene	Dolores abdominales
	ha	Consumido grasas
	tiene	Orina amarilla
Calculos Renales	ha	Sido operado
	tiene	Dolores lumbares
	no tiene	Temperatura
	tiene	Dieta rica en calcio
ºUlceras Estomacales	ha	Bebido
	tiene	Consumo de café
	tiene	Dolores Abdominales

4.1. Ingreso de la Información en MYCIN.

Tabla 2.

Experto	¿ El paciente tiene dolores ?
Usuario	Si
Experto	¿ El paciente tiene fiebre ?
Usuario	Si
Experto	¿ El paciente tiene dolores en la zona cervical?
Usuario	No
Experto	¿ El paciente tiene dolores en la zona abdominal?
Usuario	Si
Experto	¿ El paciente ha sido operado antes?
Usuario	Si
Experto	Indique temperatura del paciente
Usuario	40°

Estas tablas son sacadas del documento de (Rodrigo) http://www.revistasbolivianas.ciencia.bo/pdf/rits/n1/n1a31.pdf

Aplicación del paradigma

con base a lo investigado podemos obtener estas conclusiones del como se aplica dicho paradigma

Base de conocimientos:

- Contenía reglas médicas basadas en el conocimiento de expertos en enfermedades infecciosas.
- Utilizaba una estructura de reglas del tipo "Si, Entonces.".

Motor de inferencia:

- Aplicaba lógica difusa y probabilística para determinar el diagnóstico y recomendar antibióticos.
- Usaba el factor de certeza (CF) para manejar la incertidumbre en los síntomas y diagnósticos.

Interacción con el usuario:

- Hacía preguntas al médico sobre síntomas, pruebas de laboratorio y antecedentes del paciente.
- Generaba un diagnóstico basado en las reglas y proponía tratamientos óptimos.

Explicación de sus decisiones:

- Permitía a los médicos preguntar por qué había llegado a una conclusión específica.

Limitaciones y Beneficios del paradigma simbólico en el Ejemplo

Beneficios

- Explicabilidad: Podía justificar sus decisiones mostrando las reglas usadas.
- Precisión: Basado en conocimientos médicos estructurados.
- Razonamiento lógico: Usaba reglas para diagnosticar y recomendar tratamientos.
- No dependía de grandes datos: No necesitaba entrenarse con miles de casos.

Limitaciones

- **No aprendía automáticamente:** Requería que expertos añadieran reglas manualmente.
- **Difícil de escalar:** Muchas reglas hacían que el sistema fuera complicado de mantener.
- **Rigidez:** No podía manejar casos imprevistos o síntomas combinados.
- **Interacción limitada:** No procesaba lenguaje natural de forma flexible.

Proceso de aprendizaje automático

Adquisición de datos.

La **adquisición de datos** se trata de recopilar y de obtener toda la información relevante y significativa a partir de diversas fuentes o dispositivos, este tipo de práctica es excelente para las empresas porque permite darles datos útiles y precisos para su posterior análisis, pudiendo hacer uso de ellos en distintos contextos. (sdindustrial, n.d.)

Procesamiento de datos.

El concepto de **procesamiento de datos** hace referencia a la acción de juntar y gestionar elementos o datos (bytes, kilobytes, megabytes, etc.) con la finalidad de elaborar información importante y útil que permita conseguir una constante mejora en todas las actividades. Asimismo, un buen procesamiento facilita tener en todo momento datos que muestren la realidad que vive la empresa, concediendo así una visión precisa para cualquier toma de decisiones. (Rojas, n.d.)

Entrenamiento del modelo.

El proceso de entrenamiento de un modelo de ML consiste en proporcionar datos de entrenamiento de los cuales aprender a un algoritmo de ML (es decir, el *algoritmo de aprendizaje*). El término *modelo de ML* se refiere al artefacto de modelo que se crea en el proceso de entrenamiento. (docs.aws.amazon.com, n.d.)

Los datos de entrenamiento deben contener la respuesta correcta, que se conoce como *destino* o *atributo de destino*. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir) y genera un modelo de ML que captura dichos patrones. (docs.aws.amazon.com, n.d.)

Evaluación del modelo.

La evaluación de modelos de aprendizaje automático es un proceso destinado a determinar la calidad y eficacia de los modelos desarrollados para diversas tareas predictivas o descriptivas en IA. Se basa en el uso de métricas y técnicas específicas para medir el rendimiento del modelo con datos nuevos, en particular datos que no ha visto durante su entrenamiento. El objetivo principal es garantizar que el modelo funciona satisfactoriamente en condiciones reales y que es capaz de generalizar correctamente más allá de los datos de entrenamiento. (es.innovatiana.com, n.d.)

Implementación del modelo.

El despliegue o implementación de modelos Machine Learning es el proceso para hacer que los modelos estén disponibles en entornos de producción, donde pueden proporcionar predicciones a otros sistemas de software. (aprendeia.com, n.d.)

Similitudes y Diferencias entre el Modelo Cognitivo y las Etapas del Aprendizaje Automático

Las similitudes y las diferencias es una conclusión basada en lo investigado decidimos nosotros ver cuáles eran estas características conforme al conocimiento adquirido en la investigación

Similitudes

Procesamiento de información: Ambos procesan datos (sensoriales en el humano, digitales en ML).
Aprendizaje y adaptación: Los dos mejoran su rendimiento con el tiempo basándose en la experiencia.
Toma de decisiones: Tanto los modelos cognitivos como los modelos de ML buscan hacer predicciones o inferencias.
Generalización del conocimiento: Ambos buscan aplicar lo aprendido en situaciones nuevas.

Diferencias

	Modelo cognitivo	Aprendizaje automático
Base de aprendizaje	Experiencias, emociones, contexto	Datos, patrones estadísticos
Estructura	Neuronas, redes neuronales biológicas	Usa algoritmos y modelos matemáticos
Flexibilidad	Altamente adaptable y creativo	Depende de la cantidad y calidad de datos
Errores y correcciones	Puede corregirse con reflexión lógica	Se ajusta a con técnicas de optimización
Memoria	Memoria a corto y largo plazo	Solo guarda datos relevantes

Bibliografía

(s.f.). Obtenido de sdindustrial: https://sdindustrial.com.mx/blog/adquisicion-de-datos/

(s.f.). Obtenido de https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/training-ml-models.html

aprendeia.com. (s.f.). Obtenido de https://aprendeia.com/guia-para-implementar-los-modelos-de-machine-learning/

Azcoitia, S. S. (s.f.). *telefonicatech*. Obtenido de https://telefonicatech.com/blog/mycin-el-comienzo-de-la-inteligencia#:~:text=MYCIN%20era%20capaz%20de%20identificar,la%20meningitis%20y%20la%20bacteriemia.

docs.aws.amazon.com. (s.f.). Obtenido de https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/training-ml-models.html

es.innovatiana.com. (s.f.). Obtenido de https://es.innovatiana.com/post/how-to-evaluate-ai-models

iccsi. (s.f.). Obtenido de https://iccsi.com.ar/inteligencia-artificial-simbolica-e-conexionista/

Rodrigo, A. T. (s.f.). *http://www.revistasbolivianas.ciencia.bo/*. Obtenido de http://www.revistasbolivianas.ciencia.bo/pdf/rits/n1/n1a31.pdf

Rojas, A. (s.f.). *rittalnet.cl*. Obtenido de https://rittalnet.cl/procesamiento-de-datos/

www.expert.ai. (s.f.). Obtenido de https://www.expert.ai/app/uploads/2022/03/WP-Hybrid-AI-The-value-of-symbolic-ai_ES.pdf