

Instituto tecnológico de Culiacán

Materia:

Inteligencia Artificial

Docente:

ZURIEL DATHAN MORA FELIX

Alumnos:

Peña Vizcarra Jatniel Alejandro

Carrasco Medina Carlos Ivan

Tarea:

Investigación de paradigmas

Carrera:

Ing. Sistemas Computacionales

Grupo:

11:00-12:00

Fecha:

09/03/25



1. Enfoque Simbólico:

El enfoque simbólico es uno de los paradigmas más antiguos de la IA, basado en la representación del conocimiento mediante símbolos y reglas. Este enfoque ha sido utilizado con éxito en sistemas expertos, como los sistemas de diagnóstico médico.

Ejemplo 1: Sistemas Expertos en Diagnóstico Médico

Descripción: Los sistemas expertos se utilizan para emular el proceso de toma de decisiones de un experto humano. En el ámbito médico, estos sistemas ayudan a diagnosticar enfermedades basándose en síntomas ingresados y reglas de inferencia.

Aplicación del paradigma:

Los sistemas expertos en diagnóstico médico utilizan reglas predefinidas basadas en conocimiento médico estructurado. Por ejemplo, el sistema MYCIN, desarrollado en la Universidad de Stanford, utilizaba lógica simbólica para diagnosticar infecciones bacterianas. Cada enfermedad se representa como un conjunto de reglas "si-entonces", lo que permite razonar de manera estructurada.

Ventajas:

- Precisión y consistencia: Los sistemas expertos pueden aplicar reglas médicas de manera uniforme, reduciendo errores humanos y mejorando la precisión en diagnósticos.
- Disponibilidad: Estos sistemas pueden operar continuamente, ofreciendo soporte médico en todo momento.

Limitaciones:

- Rigidez: La dependencia de reglas predefinidas limita la capacidad del sistema para adaptarse a casos no contemplados o situaciones nuevas.
- Mantenimiento complejo: Actualizar y mantener las bases de conocimiento puede ser laborioso y requiere intervención humana constante.

Ejemplo en Python usando reglas lógicas:

```
from experta import Fact, Rule, KnowledgeEngine

class Diagnostico(Fact):
    pass

class SistemaExperto(KnowledgeEngine):
    @Rule(Diagnostico(sintoma="fiebre"), Diagnostico(sintoma="tos"))
    def diagnostico_gripe(self):
        print("El paciente podría tener gripe.")

engine = SistemaExperto()
engine.reset()
```

```
engine.declare(Diagnostico(sintoma="fiebre"))
engine.declare(Diagnostico(sintoma="tos"))
engine.run()
```

Ejemplo 2: Asistentes Virtuales como Siri

Descripción: Los asistentes virtuales utilizan IA simbólica para interpretar comandos de voz y responder con información relevante. Los sistemas siguen una serie de reglas predefinidas para interactuar con los usuarios.

Aplicación del paradigma:

Los asistentes virtuales como Siri utilizan IA simbólica para procesar lenguaje natural. Cuando un usuario dice "¿Cuál es el clima hoy?", Siri descompone la oración en tokens y aplica reglas gramaticales para identificar la intención del usuario. Luego, usa bases de conocimiento estructuradas para generar una respuesta.

Ventajas:

- Interacción natural: Permiten a los usuarios comunicarse mediante lenguaje natural, mejorando la experiencia del usuario.
- Automatización de tareas: Pueden gestionar tareas rutinarias, liberando tiempo para actividades más complejas.

Limitaciones:

- Comprensión limitada: Pueden tener dificultades para entender contextos complejos o ambiguos.
- Dependencia de actualizaciones: Requieren actualizaciones constantes para adaptarse a nuevos comandos o expresiones.

Ejemplo en Python usando NLTK para tokenización de comandos:

```
import nltk
from nltk.tokenize import word_tokenize

nltk.download('punkt')

entrada = "¿Cuál es el clima hoy?"
tokens = word_tokenize(entrada.lower())

if "clima" in tokens:
    print("Buscando información del clima...")
```

2. Enfoque Conexionista:

El enfoque conexionista se basa en la simulación de redes neuronales artificiales que imitan el funcionamiento del cerebro humano. Este paradigma es fundamental en el desarrollo de modelos de aprendizaje automático y reconocimiento de patrones.

Ejemplo 1: Reconocimiento de Voz

Descripción: Las redes neuronales profundas (DNN) se utilizan en aplicaciones como el reconocimiento de voz. Los modelos aprenden de grandes volúmenes de datos de voz para mejorar la precisión de la transcripción.

Aplicación del paradigma:

Los sistemas de reconocimiento de voz utilizan redes neuronales profundas (DNN). Estas redes aprenden patrones en grandes volúmenes de datos de voz para predecir la secuencia de palabras más probable. Por ejemplo, modelos como Wav2Vec 2.0 han demostrado reducir el error en reconocimiento de voz sin supervisión directa.

Ventajas:

- Adaptabilidad: Las redes neuronales pueden aprender y mejorar con más datos, aumentando su precisión.
- Procesamiento en tiempo real: Permiten transcripciones y respuestas inmediatas, mejorando la eficiencia.

Limitaciones:

- "Caja negra": La falta de interpretabilidad en las decisiones de las redes neuronales dificulta la comprensión de su funcionamiento.
- Sesgos inherentes: Pueden perpetuar prejuicios presentes en los datos de entrenamiento, afectando la equidad del sistema.

Ejemplo en Python usando transformers para transcripción de audio:

```
from transformers import Wav2Vec2ForCTC, Wav2Vec2Processor
import torchaudio

modelo = Wav2Vec2ForCTC.from_pretrained("facebook/wav2vec2-large-960h")
procesador = Wav2Vec2Processor.from_pretrained("facebook/wav2vec2-large-960h")

audio, tasa_muestreo = torchaudio.load("audio.wav")
entrada = procesador(audio, sampling_rate=tasa_muestreo,
return_tensors="pt").input_values
salida = modelo.generate(entrada)

texto = procesador.batch_decode(salida)
print("Texto transcrito:", texto)
```

Ejemplo 2: Reconocimiento de Imágenes

Descripción: El reconocimiento de imágenes se realiza utilizando redes neuronales convolucionales (CNN) que analizan y reconocen patrones en imágenes, como en la identificación de rostros o objetos en fotos.

Aplicación del paradigma:

Las redes neuronales convolucionales (CNN) aprenden representaciones jerárquicas de las imágenes. Capas iniciales detectan bordes, mientras que capas más profundas identifican formas y patrones complejos. Se aplican en sistemas como Google Photos para organizar imágenes automáticamente.

Ventajas:

- Alta precisión: Las redes neuronales convolucionales han demostrado ser efectivas en la identificación de objetos y patrones visuales.
- Aplicaciones diversas: Se utilizan en campos como la medicina, seguridad y entretenimiento.

Limitaciones:

- Necesidad de grandes volúmenes de datos: Requieren conjuntos de datos extensos y variados para entrenarse adecuadamente.
- Vulnerabilidad a ataques adversariales: Pequeñas alteraciones en las imágenes pueden engañar al sistema, comprometiendo su fiabilidad.

Ejemplo en Python usando TensorFlow para clasificar imágenes:

```
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.preprocessing import image
import numpy as np

modelo = MobileNetV2(weights='imagenet')
img = image.load_img("imagen.jpg", target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = tf.keras.applications.mobilenet_v2.preprocess_input(img_array)

predicciones = modelo.predict(img_array)
etiqueta =
tf.keras.applications.mobilenet_v2.decode_predictions(predicciones)
print("Clasificación:", etiqueta[0][0])
```

3. Enfoque Bioinspirado:

La inteligencia computacional inspirada en la vida se refiere a técnicas que emulan procesos naturales para resolver problemas complejos. Este enfoque abarca algoritmos evolutivos, sistemas inmunológicos artificiales y otras metodologías inspiradas en la biología.

Ejemplo 1: Algoritmos Genéticos en Optimización

Descripción: Los algoritmos genéticos simulan el proceso de evolución natural para resolver problemas complejos de optimización, como la mejora de sistemas logísticos o la programación de recursos.

Aplicación del paradigma:

Los algoritmos genéticos simulan la evolución biológica para encontrar soluciones óptimas. Se utilizan en diseño de circuitos electrónicos, donde diferentes configuraciones se "cruzan" y evolucionan hasta obtener el mejor diseño.

Ventajas:

- Exploración amplia de soluciones: Pueden encontrar soluciones óptimas en espacios de búsqueda complejos.
- Flexibilidad: Son aplicables a una variedad de problemas sin requerir modificaciones significativas.

Limitaciones:

- Coste computacional elevado: Pueden requerir muchos recursos y tiempo para converger a una solución óptima.
- No garantizan la mejor solución: Pueden estancarse en óptimos locales, sin alcanzar la solución global óptima.

Ejemplo en Python con DEAP para resolver una función de optimización:

```
from deap import base, creator, tools, algorithms
import random

creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

def evaluar(individuo):
    return sum(individuo),

toolbox = base.Toolbox()
toolbox.register("attr_bool", random.randint, 0, 1)
toolbox.register("individual", tools.initRepeat, creator.Individual,
    toolbox.attr_bool, 10)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("evaluate", evaluar)
```

```
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

pop = toolbox.population(n=100)
algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=40,
verbose=True)
```

Ejemplo 2: Redes Neuronales Artificiales

Descripción: Las redes neuronales artificiales están inspiradas en el cerebro humano y se utilizan para tareas de clasificación, predicción y reconocimiento de patrones.

Aplicación del paradigma: Este enfoque se basa en la imitación de la estructura de las neuronas biológicas para procesar información. Redes como las perceptrones multicapa (MLP) y las redes neuronales recurrentes (RNN) se utilizan en reconocimiento de voz, traducción automática y predicción de series temporales, como en el mercado financiero o la climatología.

Ventajas:

- Capacidad de aprendizaje: Pueden aprender de datos y mejorar su rendimiento con el tiempo.
- Versatilidad: Son aplicables en diversas áreas, desde reconocimiento de patrones hasta predicciones complejas.

Limitaciones:

- Opacidad: La complejidad de las redes dificulta la interpretación de sus decisiones.
- Necesidad de datos etiquetados: Requieren grandes cantidades de datos etiquetados para un entrenamiento efectivo.

4. Enfoque Probabilístico:

El enfoque probabilístico en IA se centra en la gestión de la incertidumbre y la toma de decisiones basadas en modelos estadísticos. Aunque ofrece flexibilidad, también implica desafíos en términos de complejidad computacional.

Ejemplo 1: Sistemas de Recomendación

Descripción: Los sistemas de recomendación utilizan modelos probabilísticos para predecir qué productos o contenidos le interesarán a un usuario, basándose en el análisis de su comportamiento pasado.

Aplicación del paradigma:

Los sistemas de recomendación usan modelos probabilísticos, como el filtrado bayesiano, para predecir productos que pueden interesar a los usuarios en plataformas como Netflix.

Ventajas:

- Personalización: Ofrecen recomendaciones adaptadas a las preferencias individuales de los usuarios.
- Mejora de la experiencia del usuario: Facilitan la navegación y descubrimiento de contenido relevante.

Limitaciones:

- Privacidad: El análisis de datos personales puede generar preocupaciones sobre la privacidad de los usuarios.
- Efecto de burbuja: Pueden limitar la exposición del usuario a contenido diverso, reforzando sesgos existentes.

Ejemplo en Python con Surprise para predicción de películas:

```
from deap import base, creator, tools, algorithms
import random

creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

def evaluar(individuo):
    return sum(individuo),

toolbox = base.Toolbox()
toolbox.register("attr_bool", random.randint, 0, 1)
toolbox.register("individual", tools.initRepeat, creator.Individual,
    toolbox.attr_bool, 10)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("evaluate", evaluar)
toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

pop = toolbox.population(n=100)
algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=40,
    verbose=True)
```

Ejemplo 2: Detección de Fraude en Transacciones Financieras

Descripción: Los modelos probabilísticos, como las redes bayesianas, se usan para identificar patrones inusuales en las transacciones bancarias y detectar posibles fraudes.

Aplicación del paradigma: Los sistemas de detección de fraude utilizan modelos probabilísticos para asignar una probabilidad de riesgo a cada transacción. Técnicas como los modelos de Markov ocultos o los árboles de decisión probabilísticos analizan factores como la ubicación del usuario, la

frecuencia de compras y el monto de la transacción. Instituciones financieras como PayPal y Visa implementan estos modelos para prevenir fraudes en tiempo real.

Ventajas:

- Identificación temprana: Detectan patrones sospechosos rápidamente, permitiendo acciones preventivas.
- Reducción de pérdidas: Ayudan a minimizar pérdidas financieras al identificar actividades fraudulentas.

Limitaciones:

- Falsos positivos: Pueden identificar erróneamente transacciones legítimas como fraudulentas, afectando la experiencia del cliente.
- Adaptación de los defraudadores: Los delincuentes pueden desarrollar nuevas técnicas para evadir los sistemas de detección.

Ejemplo en Python con pgmpy para modelar una red bayesiana:

```
from pgmpy.models import BayesianNetwork
from pgmpy.inference import VariableElimination

modelo = BayesianNetwork([('Transacción', 'Fraude'), ('Ubicación', 'Fraude')])
inferencia = VariableElimination(modelo)
resultado = inferencia.map_query(variables=['Fraude'],
evidence={'Transacción': 1, 'Ubicación': 0})
print("Probabilidad de fraude:", resultado)
```

5. Enfoque Computacional:

Ejemplo 1: Simulación de Proteínas

Este enfoque se refiere al uso de técnicas computacionales avanzadas para abordar problemas matemáticos y científicos. Por ejemplo, el aprendizaje automático ha permitido avances en la comprensión de problemas matemáticos clásicos, como el problema de Hadwiger-Nelson.

Descripción: La IA se utiliza en biología computacional para predecir la estructura de proteínas, un paso importante para el desarrollo de nuevos medicamentos.

Aplicación del paradigma:

La IA se usa para predecir la estructura de proteínas con modelos como AlphaFold. Estos modelos computacionales simulan cómo se pliegan las proteínas para facilitar el descubrimiento de nuevos medicamentos.

Ventajas:

- Aceleración en el descubrimiento de fármacos: Permiten predecir estructuras proteicas, facilitando el desarrollo de medicamentos.
- Reducción de costes: Disminuyen la necesidad de experimentos físicos costosos y laboriosos.

Limitaciones:

- Complejidad

Ejemplo en Python usando AlphaFold (requiere acceso a su API):

```
from alphafold.model import AlphaFold
modelo = AlphaFold()
estructura_predicha = modelo.predecir("secuencia_de_proteína")
print("Estructura predicha:", estructura_predicha)
```

Ejemplo 2: Modelado del Clima

Descripción: Los modelos climáticos computacionales permiten predecir cambios en el clima, simular el impacto del cambio climático y ayudar en la toma de decisiones políticas.

Aplicación del paradigma: Se emplean simulaciones computacionales basadas en ecuaciones diferenciales y modelos de predicción climática. Algoritmos avanzados analizan grandes volúmenes de datos atmosféricos para predecir patrones climáticos a corto y largo plazo. Por ejemplo, modelos como el **Community Earth System Model (CESM)** permiten simular la evolución del clima global y evaluar escenarios futuros bajo diferentes condiciones de emisiones de gases de efecto invernadero.

Ejemplo en Python con climatelearn para predicción climática:

```
from climatelearn.data import NOAA
data = NOAA()
data.load_data()
predicciones = data.predict_future_weather()
print("Predicción climática:", predicciones)
```

La adquisición de datos

La adquisición de datos es definida como el proceso de tomar información de la realidad mediante una señal como entrada, puede ser corriente o voltaje, para su posterior procesamiento y análisis, almacenamiento u otro tipo de manipulación (Chase et al., 2012; Sharma et al., 2012; Singh and Sharma, 2012)

Preprocesamiento

Los procesos de análisis de datos requieren preprocesamiento antes de la aplicación de técnicas o algoritmos, con el fin de incrementar la calidad de estos y adecuarlos a los

formatos necesarios para su procesamiento, principalmente cuando los datos provienen de diferentes fuentes.

Entrenamientos del modelo

El entrenamiento del modelo se basa en hacer un algoritmo que se aprenda a hacer una tarea en específico por ejemplo el aprendizaje no supervisado busca encontrar patrones y estructuras en datos no etiquetados, lo que permite agrupar y detectar anomalías (Sandoval, 2018, #). En este tipo de aprendizaje hay dos algoritmos (entrenamientos): el de clasificación y el de regresión.

La evaluación de modelos

La evaluación de modelos se usa antes de que manejar cualquier dato cuando se quiere realizar aprendizaje automatizado se requiere planificar con anticipación y utilizar técnicas que se adapten a los propósitos, ahí es donde entra la evaluación de modelos este nos permite asegurar que el modelo no solo funciona bien en los datos de entrenamiento, sino que también es capaz de generalizar a nuevos datos, lo que es fundamental para aplicaciones prácticas.

La implementación de modelos

La implementación de modelos este es el proceso de pasar del diseño de un modelo hasta su uso en una organización o entorno de producción. La implementación no es simplemente un paso posterior al diseño, sino un proceso de construcción conjunta entre el modelo y la organización en la que se inserta.

Componente	Similitudes	Diferencias
Proceso de adquisición	Se basan en la recopilación y procesamiento de información.	<ul style="list-style-type: none"> - El modelo cognitivo se enfoca en cómo el cerebro procesa y almacena información - mientras que el aprendizaje automático se centra en cómo adquiere habilidades o conocimientos a través de fases específicas
Preprocesamiento	requieren preparación de datos o información antes de su uso.	<ul style="list-style-type: none"> - En el modelo cognitivo se refiere a la percepción y atención - En el aprendizaje automático implica la adecuación de datos para su análisis
Entrenamiento de modelos	implican un proceso de adaptación basado en la experiencia o datos.	<ul style="list-style-type: none"> - El modelo cognitivo se "entrena" mediante la experiencia y la práctica (aprendizaje humano). - En el aprendizaje automático, el entrenamiento se realiza mediante algoritmos que ajustan parámetros para minimizar errores

Evaluación de modelos	Requieren que haya una validación para asegurar la efectividad.	<ul style="list-style-type: none"> - En el modelo cognitivo, la evaluación es interna (autoevaluación, retroalimentación). - En el aprendizaje automático, la evaluación es externa (validación con datos de prueba, métricas de precisión, recall, entre otras)
Implementación de modelos	aplican lo que se ha aprendido en un contexto práctico.	<ul style="list-style-type: none"> - El modelo cognitivo se implementa en la toma de decisiones y resolución de problemas en la vida real - En el aprendizaje automático, la implementación implica desplegar el modelo en un entorno de producción para su uso automatizado

Referencias.

Definiciones de los paradigmas y se obtuvieron los problemas de donde mismo aunque la mayoría fueron solo un listado de aplicaciones que se lograban con los paradigmas pero no daban una información detallada

- <https://sistemasinformatica.istmo.tecnm.mx/cursos/inteligencia-artificial/2-representacion-del-conocimiento/2-2-paradigmas-de-la-inteligencia-artificial/>
- <https://cibernetica.wordpress.com/2021/05/13/modelos-de-la-inteligencia-artificial-el-conexionista/>
- <https://riuma.uma.es/xmlui/bitstream/handle/10630/4686/LECCION%20INAUGURAL%202010.pdf?sequence=1&isAllowed=y>
- <https://unidigitaliasac.unizar.es/sites/unidigitaliasac/files/archivos/03.%20Una%20visi%C3%B3n%20conceptual%20de%20la%20IA.pdf>
- <https://elpais.com/ciencia/cafe-y-teoremas/2024-11-19/el-aprendizaje-automatico-ayuda-a-atacar-problemas-matematicos-clasicos.html>

De estos 3 se obtuvieron las aplicaciones

- <https://sistemasinformatica.istmo.tecnm.mx/cursos/inteligencia-artificial/2-representacion-del-conocimiento/2-2-paradigmas-de-la-inteligencia-artificial/>
- <https://riuma.uma.es/xmlui/bitstream/handle/10630/4686/LECCION%20INAUGURAL%202010.pdf?sequence=1&isAllowed=y>
- <https://unidigitaliasac.unizar.es/sites/unidigitaliasac/files/archivos/03.%20Una%20visi%C3%B3n%20conceptual%20de%20la%20IA.pdf>



En adelante

- [Sistema de adquisición de datos de bajo costo con la plataforma arduino](#)
- <https://dialnet.unirioja.es/descarga/articulo/9869574.pdf>
- [Pre-procesamiento de datos educativos desde un enfoque de dominio específico. | Respuestas](#)
- [\(PDF\) MODELOS DE APRENDIZAJE AUTOMÁTICO: APLICACIÓN Y EFICIENCIA](#)
- [1811.12808](#)
- [Models implementation: A state of the art - ScienceDirect](#)