

1. Datos Generales de la asignatura

Nombre de la asignatura:	Aprendizaje profundo
Clave de la asignatura:	MPSCO-0136
SATCA¹:	2 – 4 - 6
Carrera:	Maestría en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

La asignatura, Aprendizaje profundo, aporta al perfil de egreso del estudiante de la Maestría en Sistemas Computacionales los conocimientos, habilidades y competencias necesarios para diseñar, desarrollar e implementar soluciones basadas en redes neuronales profundas, capaces de procesar y analizar grandes volúmenes de datos, así como modelar patrones complejos mediante tecnologías y técnicas avanzadas. La importancia de esta asignatura radica en su capacidad para habilitar el desarrollo de sistemas inteligentes que resuelvan problemas complejos en diversas áreas como visión por computadora, procesamiento del lenguaje natural y análisis predictivo.

Aprendizaje Profundo se relaciona con la asignatura de Minería de Datos y Minería de Texto, ya que esta última proporcionan las técnicas necesarias para preprocesar y analizar grandes volúmenes de datos estructurados y no estructurados, los cuales son la base para entrenar modelos de aprendizaje profundo. Asimismo, se relaciona con la asignatura Bases de Datos y Bases de Conocimiento, dado que esta ofrece los fundamentos para la gestión eficiente de datos y el diseño de arquitecturas de almacenamiento, permitiendo una integración adecuada de los conjuntos de datos requeridos para la construcción y evaluación de modelos inteligentes.

Para abordar la asignatura de Aprendizaje profundo, el temario se organiza en cinco temas principales. El primer tema se centra en la programación con Python, proporcionando las bases para implementar soluciones computacionales. El segundo tema aborda la manipulación de tensores con bibliotecas como NumPy y PyTorch, fundamentales para el desarrollo de modelos eficientes. En el tercer tema, se estudia la visualización de datos utilizando herramientas como Matplotlib para interpretar y analizar patrones complejos. El cuarto tema introduce a los estudiantes en el diseño y entrenamiento de modelos de aprendizaje profundo con PyTorch, abarcando desde redes neuronales básicas hasta técnicas de regularización y optimización. Finalmente, en el quinto tema, se exploran las aplicaciones y ventajas de la transferencia de aprendizaje, permitiendo la adaptación de modelos preentrenados a problemas específicos para optimizar tiempo y recursos.

Intención didáctica

La extensión de los contenidos antes descritos debe ser adecuada de tal forma que sea posible cubrir los conceptos fundamentales de cada tema, así como proporcionar ejemplos prácticos y casos de estudio que permitan una comprensión más profunda. El enfoque de la asignatura debe ser práctico y orientado al desarrollo de proyectos que permitan a los estudiantes aplicar los conceptos aprendidos bajo un paradigma

¹ Sistema de Asignación y Transferencia de Créditos Académicos

colaborativo y dar solución a problemas de la vida real. El enfoque de la asignatura debe permitir a los estudiantes desarrollar competencias tales como resolución de problemas complejos, trabajo colaborativo, pensamiento crítico y analítico, comunicación efectiva, aprendizaje autónomo y la adaptabilidad y flexibilidad en el uso de diferentes tecnologías y herramientas de desarrollo. Por su parte, el docente debe actuar como facilitador de aprendizaje, proporcionando orientación y apoyo durante el proceso de aprendizaje, ejemplos claros y casos de estudio que permitan a los estudiantes comprender los conceptos teóricos y su aplicación práctica y retroalimentación constructiva que permita un desarrollo continuo de habilidades y competencias. Debe fomentar la participación, discusión e intercambio de ideas de los estudiantes en el desarrollo de proyectos para la solución de problemas de la vida real. Además, debe estar actualizado con las últimas tendencias y tecnologías en el campo de aprendizaje profundo.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Teziutlán. Enero 2025.	Instituto Tecnológico Superior de Teziutlán.	Elaboración de la especialidad de Inteligencia Artificial para la Maestría en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Diseña y desarrolla soluciones basadas en aprendizaje profundo que permiten procesar, analizar y modelar grandes volúmenes de datos mediante tecnologías y técnicas avanzadas, contribuyendo al desarrollo de sistemas inteligentes.

5. Competencias previas

- Utiliza herramientas de desarrollo de software y entornos de programación, demostrando habilidades para escribir, probar y depurar código en diferentes lenguajes de programación. - Identifica los conceptos básicos de manejo de datos y preprocesamiento, tales como limpieza, transformación y normalización, para preparar conjuntos de datos que serán utilizados en modelos de aprendizaje profundo. - Comprende los conceptos básicos de redes de computación y arquitectura de hardware, incluyendo el uso de GPUs, para optimizar la implementación y ejecución de modelos complejos de aprendizaje profundo. - Conoce los principios básicos de álgebra lineal y cálculo diferencial, identificando su relevancia en la manipulación de tensores y el diseño de modelos matemáticos para el aprendizaje profundo.

6. Temario

No.	Temas	Subtemas
1	Programación con Python	1.1. Introducción a la sintaxis básica y estructuras de control en Python. 1.2. Manejo de estructuras de datos. 1.3 Uso de funciones, módulos y paquetes. 1.4. Introducción a bibliotecas científicas para análisis de datos y cálculos básicos.
2	Manipulación de Tensores con Numpy y Torch	2.1. Creación, manipulación y operaciones básicas con tensores en NumPy. 2.2. Introducción a PyTorch: creación de tensores y diferencias con NumPy. 2.3. Operaciones avanzadas con tensores: transposición, concatenación, y redimensionamiento. 2.4. Uso de GPUs para acelerar operaciones con tensores en PyTorch.
3	Visualización de datos con Matplotlib	3.1. Creación de gráficos básicos. 3.2. Personalización de gráficos. 3.3. Representación de datos multidimensionales con gráficos avanzados. 3.4. Integración de Matplotlib con otras bibliotecas para análisis exploratorio.
4	Aprendizaje profundo con Pytorch	4.1. Introducción a redes neuronales. 4.2. Funciones de pérdida y optimizadores. 4.3. Entrenamiento y evaluación de modelos en conjuntos de datos básicos. 4.4. Monitoreo y visualización del proceso de entrenamiento.
5	Transferencia del aprendizaje	5.1. Introducción a la transferencia de aprendizaje: concepto y aplicaciones. 5.2. Uso de modelos preentrenados. 5.3. Ajuste fino (fine-tuning). 5.4 Evaluación del desempeño de modelos transferidos utilizando métricas estándar y comparación con modelos tradicionales.

7. Actividades de aprendizaje de los temas

1. Programación con Python	
Competencias	Actividades de aprendizaje
Específica(s): • Aplica estructuras y bibliotecas	1. Implementar un programa en Python que aplique estructuras de control. 2. Implementar un programa en Python

básicas de Python orientadas a la implementación de modelos de aprendizaje profundo. Genéricas: • Capacidad de análisis y síntesis. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de investigación.	que aplique estructuras de control y manejo de datos con relación al preprocesamiento de datos para aprendizaje profundo. 3. Implementar un programa en Python que integre bibliotecas científicas para el análisis de datos y cálculos básicos.
2. Manipulación de Tensores con Numpy y Torch	
Competencias	Actividades de aprendizaje
Específica(s): • Manipula tensores para realizar operaciones matemáticas, transformaciones y optimización de datos en la implementación de modelos de aprendizaje profundo. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de investigación. • Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. • Capacidad de trabajo en equipo.	1. Realizar ejercicios con NumPy para crear y manipular tensores, aplicando operaciones básicas como suma, multiplicación y redimensionamiento. 2. Implementar un ejercicio de clasificación utilizando tensores en PyTorch, creando un modelo simple y entrenándolo con un conjunto de datos pequeño. 3. Implementar tensores para la manipulación de datos, realizando transformaciones y preparando los datos para el entrenamiento de una red neuronal.
3. Visualización de datos con Matplotlib	
Competencias	Actividades de aprendizaje
Específica(s): • Visualiza y analiza los datos mediante gráficos y representaciones, con el fin de interpretar patrones y evaluar el rendimiento de los modelos de aprendizaje profundo. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de aplicar los conocimientos en la práctica.	1. Crear gráficos básicos a partir de un conjunto de datos y compararlos para visualizar tendencias o patrones. 2. Realizar un análisis exploratorio de datos (EDA) mediante la visualización de un conjunto de datos en Python, mostrando la distribución, correlaciones y otros aspectos clave. 3. Crear un gráfico de entrenamiento para un modelo de red neuronal en PyTorch, mostrando la evolución de la precisión y la función de pérdida.

- Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. - Capacidad de comunicación oral y escrita.	durante el entrenamiento.
4. Aprendizaje profundo con Pytorch	
Competencias	Actividades de aprendizaje
Específica(s): - Diseña, entrena y evalúa modelos de redes neuronales. Genéricas: - Capacidad de análisis y síntesis. - Capacidad de aplicar los conocimientos en la práctica. - Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. - Capacidad de comunicación oral y escrita.	- Implementar un modelo de red neuronal simple (MLP) para clasificar un conjunto de datos, entrenarlo y evaluar su desempeño. - Experimentar con diferentes funciones de pérdida y optimizadores (SGD, Adam) para entrenar un modelo en PyTorch y comparar resultados. - Desarrollar e implementar una red neuronal convolucional básica y evaluar su rendimiento en términos de precisión y tiempo de entrenamiento.
5. Transferencia del aprendizaje	
Competencias	Actividades de aprendizaje
Específica(s): - Aplica técnicas de transferencia de aprendizaje utilizando modelos preentrenados. Genéricas: - Capacidad de análisis y síntesis. - Capacidad de aplicar los conocimientos en la práctica. - Habilidad para buscar, procesar y analizar información procedente de fuentes diversas. - Capacidad de comunicación oral y escrita.	- Utilizar un modelo preentrenado para realizar clasificación en un nuevo conjunto de datos, utilizando fine-tuning. - Realizar un ajuste fino de un modelo preentrenado para adaptarlo a un conjunto de datos pequeño, evaluando el impacto en el desempeño y el tiempo de entrenamiento. - Implementar transferencia de aprendizaje para un problema específico y comparar los resultados con un modelo entrenado desde cero.

8. Práctica(s)

Con el objetivo de vincular y fortalecer el aprendizaje del saber con el saber hacer se recomienda la elaboración y desarrollo de las siguientes prácticas:

- Tema 1:
 - Implementación de un programa en Python que integre bibliotecas científicas para el análisis de datos y cálculos básicos.
- Tema 2:
 - Implementación de un modelo de clasificación simple, entrenado con un conjunto de datos pequeño.
 - Implementación de tensores para la manipulación de datos.
- Tema 3:
 - Análisis exploratorio de datos mediante visualización de datos.
 - Creación de un gráfico de entrenamiento para un modelo de red neuronal que muestre la precisión y la función de pérdida durante el entrenamiento.
- Tema 4:
 - Implementación de un modelo de red neuronal simple (MLP) para clasificar un conjunto de datos, entrenarlo y evaluar su desempeño.
 - Implementación de una red neuronal convolucional básica y evaluar su rendimiento en términos de precisión y tiempo de entrenamiento.
- Tema 5
 - Uso de un modelo preentrenado para realizar clasificación en un nuevo conjunto de datos utilizando fine-tuning.
 - Implementación de ajuste fino de un modelo preentrenado para su adaptación a un conjunto de datos pequeño

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo

en los estudiantes.

10. Evaluación por competencias

Para evaluar las actividades de aprendizaje se recomienda solicitar:

- Mapas conceptuales o mentales
- Reporte de investigación
- Reportes de prácticas
- Prototipos
- Tablas comparativas
- Estudio de casos
- Exposiciones en clase
- Portafolio de evidencias, entre otros.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar:

- Listas de cotejo.
- Guías de observación
- Rúbricas, entre otros.

11. Fuentes de información

- Bobadilla, J. (2021). Machine Learning y Deep Learning: Usando Python, Scikit y Keras. Colombia: Ediciones de la U.
- Chollet, F. (2021). Deep Learning with Python, Second Edition. Reino Unido: Manning.
- Martinez-Ramon, M., Ajith, M., Kurup, A. R. (2024). Deep Learning: A Practical Introduction. Reino Unido: Wiley.
- Meedeniya, D. (2023). Deep Learning: A Beginners' Guide. Reino Unido: CRC Press.
- Michelucci, U. (2022). Applied Deep Learning with TensorFlow 2 (p. 99). Springer, Berlín, Germany.
- Patterson, J., Gibson, A. (2017). Deep Learning: A Practitioner's Approach. Japan: O'Reilly.
- Sarkar, D., Bali, R., Ghosh, T. (2018). Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras. Alemania: Packt Publishing.