

Diplomado en Inteligencia Artificial Aplicada

Sumilla curricular y temario por módulos

Programa de formación aplicada en inteligencia artificial moderna: deep learning, procesamiento de lenguaje natural, sistemas de recomendación, visión por computadora, modelos generativos de difusión, aprendizaje por refuerzo, despliegue de modelos, sensoramiento remoto y graph machine learning.

Modalidad: Teórico-práctica con laboratorios en Python

Nivel: Intermedio aplicado

Duración total: 144 horas distribuidas en 9 módulos

Producto final: Proyecto de inteligencia artificial aplicada

Entorno de trabajo: Google Colab / entorno local con Python

Elaborado para postulantes al programa

24 de mayo de 2026

Índice

1. ¿De qué trata este diplomado?	2
2. Eje conceptual: representaciones latentes	2
3. Perfil de ingreso y egreso	3
4. Estructura del diplomado	3
5. Metodología de enseñanza	4
6. Evaluación	4
Módulo 1: Deep Learning Fundamentals	6
Módulo 2: Procesamiento de Lenguaje Natural	8
Módulo 3: Sistemas de Recomendación	10
Módulo 4: Aplicaciones de Deep Learning en Visión	12
Módulo 5: Despliegue de Modelos de Machine Learning	14
Módulo 6: Aprendizaje por Refuerzo	16
Módulo 7: Modelos Generativos de Difusión	18
Módulo 8: Deep Learning para Sensoramiento Remoto	20
Módulo 9: Graph Machine Learning	22
7. Proyecto final	24
8. Ruta de aprendizaje	24
9. Bibliografía y recursos	25

1. ¿De qué trata este diplomado?

El Diplomado en Inteligencia Artificial Aplicada es un programa de formación práctica en los métodos de IA más relevantes en la actualidad. Está diseñado para personas que ya tienen una base en programación y estadística, y quieren dar el paso hacia el desarrollo y despliegue de modelos de inteligencia artificial en problemas reales.

El programa recorre nueve módulos que cubren dominios distintos pero comparten una lógica común: transformar datos complejos en representaciones que permitan clasificar, predecir, recomendar o generar. En NLP, el texto se convierte en vectores que capturan significado. En recomendación, las interacciones usuario-ítem se descomponen en perfiles latentes. En visión, una red aprende jerarquías visuales que van de bordes simples a conceptos complejos. En grafos, los nodos adquieren representaciones que codifican sus relaciones. Esta idea transversal conecta los módulos y permite que el aprendizaje de cada uno refuerce a los demás.

El diplomado combina exposición conceptual, laboratorios guiados en Python con datos reales o reproducibles, y entregas aplicadas por módulo. Cada entrega construye el portafolio del participante y puede escalar en complejidad hasta convertirse en el proyecto final.

Al concluir el programa, el participante será capaz de formular un problema de datos como problema de IA, seleccionar un enfoque adecuado, entrenar y evaluar un modelo con criterio técnico, documentar sus decisiones y presentar sus resultados a una audiencia técnica y no técnica.

2. Eje conceptual: representaciones latentes

Una idea transversal al diplomado es que el aprendizaje automático moderno, en casi todos sus dominios, puede entenderse como aprender a representar datos de forma más compacta y útil.

- En **NLP**: el texto pasa de matrices dispersas de palabras a vectores densos (embeddings) o representaciones contextuales (transformers) que capturan significado.
- En **sistemas de recomendación**: la matriz usuario-ítem, mayoritariamente vacía, se aproxima mediante vectores latentes que describen preferencias no observadas directamente.
- En **visión por computadora**: las capas de una red convolucional aprenden progresivamente representaciones que van de bordes simples a patrones semánticos complejos.
- En **modelos de difusión**: la red aprende a distinguir la estructura de los datos del ruido, lo que le permite generar muestras nuevas a partir de ruido puro.
- En **graph machine learning**: nodos y aristas adquieren embeddings que codifican no solo las propiedades de una entidad, sino su posición relacional en la red.

Esta lógica compartida hace que el aprendizaje de un módulo facilite la comprensión de los siguientes. Un participante que entiende la factorización matricial del módulo de recomendación

tiene una ventaja real al estudiar embeddings de nodos en el módulo de grafos.

3. Perfil de ingreso y egreso

Perfil de ingreso

- Programación básica en Python: estructuras de datos, funciones y uso de librerías.
- Estadística descriptiva básica: medias, varianza, distribuciones y correlación.
- Álgebra lineal elemental: vectores, matrices y producto matricial.
- Contacto previo con machine learning supervisado (regresión, clasificación).
- Capacidad de trabajar con notebooks en Google Colab o entorno equivalente.

Perfil de egreso

Al finalizar el diplomado, el participante será capaz de:

- Identificar qué tipo de modelo de IA corresponde a un problema de datos dado.
- Preparar datos estructurados, textuales, visuales, geoespaciales y relacionales.
- Entrenar, evaluar e interpretar modelos de deep learning en múltiples dominios.
- Construir un prototipo funcional de inferencia y documentarlo para revisión o entrega.
- Reconocer las limitaciones técnicas, sesgos y consideraciones éticas relevantes en los modelos que desarrolla.
- Presentar resultados de IA a una audiencia mixta técnica y de negocio.

4. Estructura del diplomado

Mód.	Nombre	Horas	Entrega principal
1	Deep Learning Fundamentals	18 h	Red neuronal entrenada y documentada.
2	Procesamiento de Lenguaje Natural	20 h	Pipeline NLP con dos enfoques comparados.
3	Sistemas de Recomendación	14 h	Motor de recomendación con evaluación offline.
4	Aplicaciones de Deep Learning	16 h	Modelo de visión con análisis de errores.
5	Despliegue de Modelos (MLOps)	18 h	API dockerizada con monitoreo básico.

6	Aprendizaje por Refuerzo	12 h	Agente entrenado en entorno simulado.
7	Modelos Generativos de Difusión	12 h	Experimento generativo documentado.
8	Deep Learning para Sensoramiento Remoto	14 h	Mapa predictivo sobre imágenes satelitales.
9	Graph Machine Learning	20 h	Análisis de red con modelo predictivo.
Total 144 h			Proyecto final de IA aplicada.

5. Metodología de enseñanza

Cada módulo sigue una estructura de cuatro componentes aplicados en todas las sesiones:

1. **Fundamento conceptual:** exposición del modelo, sus supuestos y la matemática mínima necesaria para entender qué está haciendo el algoritmo y por qué funciona.
2. **Laboratorio guiado:** el instructor desarrolla un notebook en Python paso a paso, explicando cada decisión de implementación con datos reales o reproducibles.
3. **Práctica autónoma:** el participante resuelve una variante del laboratorio, ajustando parámetros, usando datos distintos o resolviendo una tarea relacionada.
4. **Entrega del módulo:** producto breve que integra lo aprendido y alimenta el portafolio. Puede escalar en complejidad hasta convertirse en el proyecto final.

La evaluación privilegia la capacidad de aplicar sobre la de memorizar. Un análisis bien documentado de por qué un modelo falla vale más que un resultado correcto sin interpretación.

6. Evaluación

Componente	Peso	Evidencia esperada
Entregas por módulo	40 %	Notebooks documentados, scripts y reportes técnicos breves.
Proyecto final	35 %	Repositorio, modelo evaluado y prototipo funcional de inferencia.

Exposición final	15 %	Presentación de 15 minutos para audiencia técnica y ejecutiva.
Participación	10 %	Discusión de casos y retroalimentación entre participantes.

Módulo 1: Deep Learning Fundamentals

Duración: 18 horas

Sumilla

El módulo introduce los conceptos esenciales del aprendizaje profundo: cómo una red neuronal aprende a partir de datos, qué papel cumplen la función de pérdida y el optimizador, y cómo identificar los problemas más comunes durante el entrenamiento. Se trabajan redes densas, funciones de activación, backpropagation, regularización y validación. Al final del módulo, el participante puede entrenar y evaluar un modelo básico en PyTorch, interpretar curvas de aprendizaje con criterio y documentar sus experimentos de forma reproducible. Este módulo es la base técnica de todos los que siguen.

Objetivos

- Describir el flujo de entrenamiento: datos, modelo, función de pérdida, optimizador y evaluación.
- Implementar y entrenar redes densas y CNN simples en PyTorch.
- Diagnosticar sobreajuste, subajuste y fuga de datos a partir de curvas de aprendizaje.
- Aplicar regularización (dropout, weight decay, batch normalization) y ajuste básico de hiperparámetros.

Contenidos

1. Diferencias entre IA, machine learning y deep learning. Cuándo usar cada enfoque.
2. Tensores y operaciones matriciales como base computacional de las redes neuronales.
3. Perceptrón y redes multicapa: arquitectura, pesos y sesgo.
4. Funciones de activación: ReLU, sigmoid, tanh, GELU y cuándo utilizarlas.
5. Funciones de pérdida: MSE para regresión, entropía cruzada para clasificación.
6. Descenso por gradiente y backpropagation: intuición y mecánica.
7. Optimizadores: SGD, Adam y variantes. Tasa de aprendizaje y scheduling.
8. Regularización: dropout, weight decay, batch normalization y early stopping.
9. Partición de datos, validación cruzada y prevención de fuga de información.
10. Introducción a CNN: filtros, pooling y mapas de características.
11. Panorama de arquitecturas avanzadas: RNN, autoencoders, transformers.
12. Buenas prácticas: reproducibilidad, semillas, logging y documentación de experimentos.

Laboratorios

- Clasificación tabular con una red densa: entrenamiento, curvas y matriz de confusión.

- Clasificación de imágenes con una CNN básica en PyTorch.
- Diagnóstico de sobreajuste: comparación con y sin regularización.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Clasificador de riesgo crediticio:** red densa sobre datos tabulares de solicitudes de crédito. Incluye análisis de variables relevantes, curvas de aprendizaje y métricas de negocio (precisión, recall y umbral de decisión).
- **Detector de anomalías en manufactura:** autoencoder simple entrenado sobre datos de sensores industriales para identificar comportamientos fuera de distribución.
- **Reconocimiento de dígitos manuscritos:** CNN básica sobre MNIST o Fashion-MNIST, con análisis de errores por clase y visualización de filtros aprendidos.
- **Proyecto integrador con módulo 4:** los modelos entrenados aquí pueden ampliarse al trabajar transfer learning en el Módulo 4.

Entrega del módulo

Notebook documentado con un modelo entrenado, curvas de aprendizaje comentadas y análisis de errores.

Módulo 2: Procesamiento de Lenguaje Natural

Duración: 20 horas

Sumilla

El módulo cubre el procesamiento de texto desde sus fundamentos hasta los modelos preentrenados modernos. La primera parte trabaja con representaciones clásicas: bolsa de palabras, TF-IDF y modelos estadísticos simples para clasificación y descubrimiento de tópicos. La segunda parte introduce embeddings, redes neuronales para texto y la arquitectura transformer, culminando en el uso práctico de modelos preentrenados de Hugging Face para clasificación, búsqueda semántica y generación de texto. El módulo es intencionalmente amplio porque NLP es el dominio con mayor penetración práctica en empresas e instituciones en la actualidad.

Objetivos

- Preprocesar texto: limpieza, tokenización, normalización y representación vectorial.
- Construir pipelines de clasificación y modelado de tópicos con métodos clásicos.
- Evaluar clasificadores NLP con métricas adecuadas para clases desbalanceadas.
- Aplicar modelos preentrenados para clasificación, búsqueda semántica y generación de texto.

Contenidos

Bloque A — Representaciones clásicas

1. Tareas principales de NLP: clasificación, extracción, búsqueda y generación.
2. Complejidad del lenguaje: ambigüedad, polisemia, contexto y variación.
3. Tokenización, normalización, lematización y n-gramas.
4. Matriz documento-término, bolsa de palabras y TF-IDF.
5. Clasificadores para texto: Naive Bayes, regresión logística y SVM.
6. Modelado de tópicos con NMF: descubrimiento de estructura semántica no supervisada.
7. Reducción dimensional y visualización de documentos con UMAP y HDBSCAN.

Bloque B — Modelos profundos y preentrenados

8. Embeddings de palabras: Word2Vec, GloVe y FastText.
9. Redes recurrentes para texto: RNN, LSTM y GRU.
10. Modelos de lenguaje: predicción de la siguiente palabra como tarea de preentrenamiento.
11. Arquitectura transformer: atención multi-cabeza, encoders y decoders.
12. Fine-tuning de modelos preentrenados (BERT, RoBERTa) para clasificación.
13. Grandes modelos de lenguaje: prompting estructurado y evaluación de respuestas.

14. Riesgos del uso de LLMs: sesgos, alucinaciones, privacidad y uso responsable.

Laboratorios

- Pipeline de clasificación de sentimiento con TF-IDF y regresión logística.
- Topic modeling con NMF sobre una colección documental real.
- Comparación cuantitativa: TF-IDF vs. embeddings vs. transformer en la misma tarea.
- Fine-tuning ligero de BERT para clasificación binaria.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Analizador de reseñas de productos:** pipeline completo de clasificación de sentimiento (positivo / negativo / neutro) sobre reseñas de e-commerce, con comparación de métodos clásicos y transformer.
- **Motor de búsqueda semántica:** sistema de recuperación de documentos basado en embeddings de oraciones, con interfaz de consulta y comparación de resultados frente a búsqueda por palabras clave.
- **Clasificador de tickets de soporte:** modelo que categoriza automáticamente solicitudes de soporte técnico por área, prioridad o tipo de problema, incluyendo análisis de errores por categoría.
- **Explorador temático de noticias:** aplicación de topic modeling sobre un corpus de noticias para identificar tendencias por período de tiempo y visualizar la evolución de tópicos.
- **Proyecto integrador con módulo 5:** cualquiera de los modelos anteriores puede desplegarse como API REST en el Módulo 5.

Entrega del módulo

Pipeline NLP con al menos dos enfoques evaluados, reporte de métricas y análisis de cuándo falla cada uno.

Módulo 3: Sistemas de Recomendación

Duración: 14 horas

Sumilla

El módulo estudia cómo construir sistemas que sugieran ítems relevantes a usuarios a partir de sus comportamientos e interacciones anteriores. Se parte de enfoques simples (popularidad, reglas de negocio) y se avanza hacia filtrado colaborativo y factorización matricial. La idea central es que una matriz de interacciones usuario-ítem puede descomponerse en vectores latentes que representan preferencias no observadas directamente. El módulo concluye con criterios de evaluación offline y consideraciones de producción básicas.

Objetivos

- Formular un problema de recomendación a partir de datos de interacción.
- Implementar recomendadores por popularidad, basados en contenido y filtrado colaborativo.
- Aplicar factorización matricial con ALS para feedback explícito e implícito.
- Evaluar recomendadores con métricas top- k y discutir su conexión con métricas de negocio.

Contenidos

1. Motivación: sobrecarga de opciones y personalización como ventaja competitiva.
2. Tipos de datos de interacción: ratings, clicks, compras y tiempo de visualización.
3. Feedback explícito vs. feedback implícito: diferencias y tratamiento.
4. Recomendadores no personalizados: popularidad, tendencias y reglas de negocio.
5. Recomendación basada en contenido: similitud entre ítems a partir de sus atributos.
6. Filtrado colaborativo: similitud usuario-usuario e ítem-ítem con vecinos cercanos.
7. Factorización matricial: representación latente de usuarios e ítems.
8. ALS (Alternating Least Squares) para matrices dispersas con feedback implícito.
9. Recomendación híbrida: combinación de enfoques.
10. Evaluación offline: Precision@ k , Recall@ k , MAP, MRR y NDCG.
11. Evaluación online: CTR, conversión, retención y experimentos A/B.
12. Problemas prácticos: cold start, sesgo de popularidad, diversidad y cobertura.
13. Arquitectura de producción: generación de candidatos, ranking y re-ranking.

Laboratorios

- Baseline por popularidad y similitud ítem-ítem sobre un dataset de interacciones reales.

- ALS con feedback implícito usando la biblioteca `implicit`.
- Evaluación offline comparativa con métricas top- k .

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Recomendador de películas o series:** motor personalizado sobre MovieLens, con comparación entre popularidad, filtrado colaborativo y ALS. Análisis de cold start y diversidad de recomendaciones.
- **Recomendador de cursos:** sistema que sugiere cursos a estudiantes a partir de su historial de inscripciones, con evaluación de cobertura del catálogo y sesgo de popularidad.
- **Recomendador de artículos de noticias:** combinación de contenido (embeddings del artículo) y comportamiento (historial de lectura), con análisis de diversidad temática.
- **Proyecto integrador con módulos 2 y 9:** los embeddings de texto del Módulo 2 pueden enriquecer la recomendación basada en contenido; los grafos de interacción del Módulo 9 pueden reemplazar el filtrado colaborativo clásico.

Entrega del módulo

Motor de recomendación con al menos dos enfoques evaluados y comparación de métricas offline.

Módulo 4: Aplicaciones de Deep Learning en Visión

Duración: 16 horas

Sumilla

Este módulo trabaja deep learning aplicado a imágenes: cómo funcionan las redes convolucionales, por qué aprenden representaciones jerárquicas y cómo usarlas para clasificar, detectar y segmentar objetos en imágenes. El énfasis está en la transferencia de aprendizaje: usar modelos preentrenados en grandes colecciones de imágenes y adaptarlos eficientemente a problemas específicos con datos más limitados. Se revisan aplicaciones en salud, industria y medio ambiente, con atención especial a la evaluación e interpretación visual de los resultados.

Objetivos

- Aplicar transfer learning para clasificación de imágenes en dominios específicos.
- Preparar datasets de imágenes: aumentación, balanceo y pipelines de carga eficiente.
- Comprender la arquitectura encoder-decoder y aplicar U-Net para segmentación.
- Evaluar modelos de visión con métricas adecuadas y analizar errores visualmente.

Contenidos

1. CNN: filtros, pooling, receptive field y mapas de características.
2. Transfer learning: por qué funciona y cómo elegir qué capas congelar.
3. Arquitecturas preentrenadas: ResNet, EfficientNet y Vision Transformer (ViT).
4. Preparación de datos: aumentación, normalización y manejo de desbalance.
5. Clasificación de imágenes con fine-tuning sobre modelos preentrenados.
6. Detección de objetos: bounding boxes, IoU, mAP e introducción a YOLO.
7. Segmentación semántica: diferencia con detección y arquitecturas encoder-decoder.
8. U-Net: skip connections y su papel en la reconstrucción espacial detallada.
9. Interpretabilidad visual: Grad-CAM y análisis de activaciones intermedias.
10. Evaluación: accuracy, F1, IoU y coeficiente Dice.
11. Aplicaciones en salud, industria, agricultura y análisis ambiental.

Laboratorios

- Clasificación de imágenes médicas (rayos X o histología) con ResNet preentrenado.
- Segmentación de estructuras con U-Net sobre un dataset de imágenes pequeñas.
- Análisis visual de errores con Grad-CAM.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Clasificador de enfermedades en plantas:** detección de patologías foliares a partir de fotografías de campo, con análisis de errores por especie y condición de iluminación.
- **Segmentación de lesiones en imágenes médicas:** U-Net aplicado a la delimitación de masas pulmonares, lesiones de piel u otras estructuras en imágenes de acceso público.
- **Detector de defectos en líneas de producción:** clasificador o segmentador entrenado sobre imágenes de piezas industriales para identificar grietas, deformaciones o imperfecciones.
- **Proyecto integrador con módulo 8:** los modelos de visión entrenados aquí pueden extenderse directamente a imágenes satelitales en el Módulo 8, reemplazando imágenes RGB por datos multibanda.

Entrega del módulo

Modelo de visión con reporte de métricas, visualizaciones Grad-CAM y reflexión sobre limitaciones.

Módulo 5: Despliegue de Modelos de Machine Learning

Duración: 18 horas

Sumilla

Este módulo enseña a convertir un modelo entrenado en un servicio que puede usarse en producción. El punto de partida es reconocer que un notebook con un modelo entrenado no es una solución: para que genere valor, el modelo debe estar empaquetado, expuesto como API, monitoreado y mantenible. Se cubren los principios de MLOps, la organización de proyectos reproducibles, la creación de APIs con FastAPI, la contenedorización con Docker y los fundamentos de monitoreo de datos y modelos en operación.

Objetivos

- Organizar un proyecto de ML con estructura reproducible y versionamiento de código y modelos.
- Serializar y empaquetar un modelo entrenado como servicio de inferencia con FastAPI.
- Contenedorizar el servicio con Docker para despliegue reproducible.
- Diseñar una estrategia básica de monitoreo de datos, modelos y latencia.

Contenidos

1. Del notebook al producto: qué cambia estructuralmente y por qué importa.
2. Organización estándar de proyectos de ML: separación de datos, código, artefactos y configuración.
3. Gestión de entornos virtuales y dependencias con pip y conda.
4. Versionamiento de código con Git, datos con DVC y modelos con MLflow.
5. Pipelines de entrenamiento e inferencia: separación de responsabilidades.
6. Serialización de modelos: pickle, joblib y ONNX.
7. APIs de inferencia con FastAPI: endpoints, validación de entradas y salidas.
8. Pruebas básicas para modelos: datos de borde y distribuciones inesperadas.
9. Docker: imágenes, contenedores, Dockerfile y docker-compose.
10. Logging estructurado, trazabilidad de predicciones y gestión de errores.
11. Data drift y concept drift: cuándo un modelo empieza a degradarse sin que nadie lo vea.
12. Estrategias de reentrenamiento y versionamiento de modelos en producción.
13. Seguridad básica, gobernanza y auditoría de modelos en producción.

Laboratorios

- Empaquetar un clasificador entrenado como API REST con FastAPI.
- Crear un Dockerfile y probar el servicio en contenedor local.
- Implementar logging de predicciones, latencia y errores.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **API de clasificación de texto en producción:** tomar el clasificador NLP del Módulo 2, empaquetarlo como servicio REST, dockerizarlo y añadir logging de predicciones y latencia.
- **Servicio de inferencia de imágenes:** despliegue del modelo de visión del Módulo 4 como endpoint HTTP que recibe una imagen y devuelve una predicción con probabilidades por clase.
- **Pipeline de reentrenamiento automatizado:** sistema que detecta drift en los datos de entrada y dispara el reentrenamiento del modelo con nuevos datos, manteniendo versiones anteriores disponibles.
- **Proyecto integrador con cualquier módulo previo:** este módulo está diseñado para productivizar los modelos de los módulos anteriores. Cualquier entrega previa puede convertirse en un servicio desplegable.

Entrega del módulo

Servicio de inferencia dockerizado, documentado y con registro de predicciones funcional.

Módulo 6: Aprendizaje por Refuerzo

Duración: 12 horas

Sumilla

El módulo introduce un paradigma de aprendizaje distinto al supervisado: en lugar de aprender de ejemplos etiquetados, un agente aprende tomando decisiones y observando sus consecuencias. Se estudian los conceptos clave —estados, acciones, recompensas, políticas y funciones de valor— a través de los procesos de decisión de Markov. El módulo culmina en Q-learning tabular y una introducción práctica a Deep Q-Networks. El énfasis está en formular correctamente el problema y comprender la mecánica del aprendizaje por experiencia.

Objetivos

- Formular un problema de decisión secuencial como proceso de decisión de Markov.
- Distinguir los conceptos fundamentales: estado, acción, recompensa, política y retorno.
- Implementar Q-learning tabular con estrategia de exploración ϵ -greedy.
- Comprender los componentes de Deep Q-Networks: red aproximadora, replay buffer y target network.

Contenidos

1. Aprendizaje por experiencia vs. aprendizaje supervisado: diferencias conceptuales.
2. Agente, entorno, ciclo de interacción y señal de recompensa.
3. Procesos de decisión de Markov: estados, acciones y transiciones probabilísticas.
4. Recompensa inmediata, retorno acumulado y factor de descuento γ .
5. Políticas determinísticas y estocásticas.
6. Funciones de valor: $V(s)$ y función acción-valor $Q(s, a)$.
7. Ecuaciones de Bellman: consistencia entre valores de estados consecutivos.
8. Iteración de políticas e iteración de valor.
9. Q-learning: actualización temporal y estrategia ϵ -greedy.
10. Deep Q-Networks: red neuronal como aproximador de Q , replay buffer y target network.
11. Aplicaciones: videojuegos, robótica, asignación de recursos y control adaptativo.

Laboratorios

- Gridworld con iteración de valor: visualización de la política óptima.
- Q-learning en un entorno discreto (FrozenLake de Gymnasium).
- DQN básico en CartPole: entrenamiento y comparación con agente aleatorio.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Agente para un juego de navegación:** entrenamiento de un agente que aprende a resolver un laberinto o entorno gridworld con obstáculos dinámicos, comparando distintas estrategias de exploración.
- **Optimización de inventario:** formulación del problema de reposición de stock como MDP, donde el agente aprende cuándo y cuánto pedir para minimizar costos de almacenamiento y penalizaciones por quiebre.
- **Control de balanceo (CartPole):** DQN aplicado al problema de equilibrio de un péndulo invertido, con análisis de la curva de aprendizaje y sensibilidad a la tasa de exploración.
- **Proyecto integrador con módulo 5:** el agente entrenado puede empaquetarse como servicio de decisión automatizada aplicando los principios de MLOps del Módulo 5.

Entrega del módulo

Agente entrenado con comparación entre política inicial y aprendida, con curvas de recompensa comentadas.

Módulo 7: Modelos Generativos de Difusión

Duración: 12 horas

Sumilla

Los modelos de difusión son hoy la base de los sistemas de generación de imágenes más avanzados. Este módulo explica su lógica desde adentro: cómo se añade ruido progresivamente a una imagen en el proceso forward, y cómo una red neuronal aprende a revertir ese proceso para generar imágenes nuevas a partir de ruido puro. Se estudia la arquitectura U-Net para denoising, el condicionamiento por texto e imagen, y variantes como DDIM que aceleran el muestreo. El módulo concluye con una discusión rigurosa sobre los riesgos y consideraciones éticas de la IA generativa.

Objetivos

- Explicar el proceso forward (ruido progresivo) y el proceso reverse (denoising aprendido).
- Comprender el papel de U-Net, los embeddings de tiempo y el condicionamiento por contexto.
- Usar modelos de difusión preentrenados para generación, edición e inpainting.
- Evaluar generaciones con criterio técnico y reflexionar sobre sus implicaciones éticas.

Contenidos

1. Panorama de modelos generativos: VAEs, GANs y difusión. Qué los diferencia.
2. Proceso forward: definición del calendario de ruido y sus propiedades estadísticas.
3. Proceso reverse: la red aprende a predecir el ruido, no la imagen directamente.
4. DDPM: función de pérdida, arquitectura y procedimiento de entrenamiento.
5. U-Net para denoising: skip connections y preservación de detalles espaciales.
6. Embeddings de tiempo: cómo la red sabe en qué paso del proceso se encuentra.
7. Embeddings de contexto: condicionamiento textual con CLIP y cross-attention.
8. Generación condicional: texto-a-imagen e imagen-a-imagen.
9. DDIM: muestreo determinista y aceleración del proceso generativo.
10. Modelos de difusión latentes: eficiencia al operar en espacio comprimido.
11. Inpainting y control de generación.
12. Aplicaciones: síntesis de datos, diseño, segmentación aumentada y creatividad computacional.
13. Riesgos: deepfakes, propiedad intelectual, sesgo en distribuciones generadas y uso responsable.

Laboratorios

- Muestreo con DDPM sobre un dataset de imágenes pequeñas (MNIST o CIFAR-10).
- Generación condicional con un modelo preentrenado de Hugging Face Diffusers.
- Comparación de calidad y velocidad entre DDPM y DDIM a distintos pasos de muestreo.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Generación de datos sintéticos para aumentar datasets pequeños:** uso de modelos de difusión para generar imágenes adicionales de clases subrepresentadas y evaluación del impacto en un clasificador del Módulo 4.
- **Editor de imágenes por instrucción de texto:** sistema de inpainting y edición guiada por texto usando un modelo preentrenado, con interfaz de demostración y análisis de casos fallidos.
- **Generador de imágenes condicionado por categoría:** modelo entrenado sobre un dominio específico (ej. texturas, dígitos, señales) con evaluación cuantitativa mediante FID u otras métricas de calidad.
- **Proyecto integrador con módulo 4:** los datos sintéticos generados con difusión pueden usarse para entrenar o mejorar los clasificadores de visión desarrollados en el Módulo 4.

Entrega del módulo

Experimento generativo documentado: parámetros, resultados visuales, limitaciones y análisis ético.

Módulo 8: Deep Learning para Sensoramiento Remoto

Duración: 14 horas

Sumilla

Este módulo aplica las técnicas de visión por computadora al análisis de imágenes satelitales y aéreas, que tienen características particulares: múltiples bandas espectrales, resolución variable, georreferenciación y grandes extensiones espaciales. Se trabaja el pipeline completo: descarga y preprocesamiento de imágenes raster, generación de tiles para entrenamiento, entrenamiento de modelos de clasificación y segmentación, e inferencia con reconstrucción del mapa final. Las aplicaciones cubren cobertura terrestre, vegetación, infraestructura y análisis de cambios temporales.

Objetivos

- Comprender la estructura de imágenes satelitales: bandas, resolución, CRS y formato raster.
- Preparar tiles, máscaras y etiquetas para entrenamiento con imágenes multibanda.
- Aplicar CNN y U-Net a tareas de clasificación y segmentación geoespacial.
- Realizar inferencia por tiles y reconstruir el mapa de predicción completo.

Contenidos

1. Fundamentos de sensoramiento remoto: satélites, sensores y tipos de resolución.
2. Estructura raster: GeoTIFF, bandas espectrales, CRS y reproyección.
3. Índices espectrales: NDVI, NDWI y NDBI, y su interpretación temática.
4. Preprocesamiento: manejo de nubes, recortes, mosaicos y normalización por banda.
5. Generación de tiles y máscaras para entrenamiento supervisado.
6. Clasificación de cobertura terrestre: enfoques por píxel y por parche.
7. Segmentación semántica geoespacial con U-Net adaptada a datos multibanda.
8. Detección de objetos en imágenes satelitales: retos específicos del dominio.
9. Análisis multitemporal y detección de cambios: conceptos y pipeline básico.
10. Inferencia por tiles y reconstrucción del mapa de predicción.
11. Evaluación espacial: métricas de segmentación e interpretación territorial.
12. Limitaciones: errores de etiquetado, heterogeneidad espacial y generalización geográfica.

Laboratorios

- Clasificación de cobertura terrestre con imágenes Sentinel-2 (EuroSAT u equivalente).
- Segmentación semántica con U-Net sobre imágenes multibanda.
- Pipeline de inferencia por tiles y reconstrucción del mapa final.

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Monitor de deforestación:** modelo que detecta cambios en cobertura boscosa entre dos fechas a partir de imágenes Sentinel-2, con mapa de áreas afectadas y estimación de superficie.
- **Clasificador de uso de suelo urbano:** segmentación de zonas residenciales, industriales, comerciales y áreas verdes en imágenes de alta resolución de una ciudad.
- **Estimador de salud de cultivos:** clasificador sobre imágenes multibanda que identifica áreas de estrés hídrico o fitosanitario usando índices espectrales como NDVI.
- **Proyecto integrador con módulo 4:** la arquitectura U-Net desarrollada en el Módulo 4 se adapta directamente a datos geospaciales multibanda, extendiendo el proyecto de visión a imágenes satelitales.

Entrega del módulo

Mapa predictivo con pipeline documentado, métricas de evaluación y análisis de errores espaciales.

Módulo 9: Graph Machine Learning

Duración: 20 horas

Sumilla

Muchos problemas del mundo real no pueden modelarse adecuadamente con tablas o imágenes: las relaciones entre entidades importan tanto como sus atributos. Este módulo introduce el análisis y el aprendizaje automático sobre grafos. La primera parte cubre teoría de grafos aplicada: métricas de red, centralidad, detección de comunidades y visualización. La segunda parte avanza hacia representaciones aprendidas: embeddings de nodos, clasificación, predicción de enlaces y graph neural networks. El módulo cierra con aplicaciones en detección de fraude, recomendación, biología y conocimiento empresarial.

Objetivos

- Representar problemas relacionales como grafos con nodos, aristas y atributos.
- Calcular e interpretar métricas de red: grado, centralidad y coeficiente de clustering.
- Aplicar detección de comunidades y algoritmos de embedding de nodos.
- Implementar clasificación de nodos y predicción de enlaces con una GCN básica.

Contenidos

Bloque A — Análisis exploratorio de grafos

1. Grafos como estructura de datos: nodos, aristas, atributos y tipos (dirigido, ponderado, bipartito).
2. Representaciones matriciales: matriz de adyacencia y lista de aristas.
3. Visualización de grafos: layouts y lectura visual de estructura.
4. Métricas de red: grado, densidad, diámetro y coeficiente de clustering.
5. Centralidad: degree, betweenness, closeness, eigenvector y PageRank.
6. Detección de comunidades: modularidad y algoritmo de Louvain.

Bloque B — Representaciones aprendidas y GNNs

7. Embeddings de nodos: DeepWalk y Node2Vec como random walk sobre grafos.
8. Clasificación de nodos con atributos: enfoque supervisado en grafos.
9. Predicción de enlaces: formulación como clasificación binaria de pares.
10. Message passing: la operación fundamental de las graph neural networks.
11. Graph Convolutional Networks (GCN): arquitectura y propagación de información.
12. GraphSAGE y Graph Attention Networks (GAT): inducción y atención en grafos.

13. Aplicaciones: fraude, recomendación, redes sociales, biología y conocimiento empresarial.

Laboratorios

- Análisis exploratorio de una red real con NetworkX (red de coautoría o transacciones).
- Detección de comunidades y visualización de la estructura de la red.
- Clasificación de nodos con una GCN usando PyTorch Geometric (Cora o CiteSeer).

Proyectos del módulo

Proyectos sugeridos para este módulo:

- **Detector de fraude en redes de transacciones:** modelo que clasifica transacciones como fraudulentas o legítimas considerando el contexto relacional entre cuentas, dispositivos y comercios.
- **Análisis de redes de colaboración científica:** exploración de comunidades temáticas, identificación de autores puente e investigadores más influyentes usando métricas de centralidad y embeddings.
- **Predicción de conexiones en redes sociales:** modelo que estima qué pares de usuarios tienen mayor probabilidad de conectarse, basado en atributos y estructura de la red existente.
- **Proyecto integrador con módulo 3:** el grafo usuario–ítem puede reemplazar el filtrado colaborativo clásico, usando GCN para generar recomendaciones basadas en la estructura relacional de las interacciones.

Entrega del módulo

Análisis completo de una red real: métricas, comunidades, embeddings y modelo predictivo documentado.

7. Proyecto final

El proyecto final integra los aprendizajes del diplomado en una solución de IA evaluable, reproducible y comunicable. A diferencia de las entregas por módulo, el proyecto final plantea un problema real con datos propios o de acceso público, lo desarrolla con el rigor metodológico del programa y lo presenta ante una audiencia mixta.

El participante puede elegir una de las siguientes líneas de aplicación, o proponer una combinación de módulos que sea coherente con su contexto profesional o de investigación:

- **NLP aplicado:** clasificación de documentos, análisis de opiniones, sistemas de búsqueda semántica o extracción de información de texto no estructurado.
- **Visión por computadora:** clasificación, detección o segmentación de imágenes en salud, industria, educación o medio ambiente.
- **Recomendación:** motor personalizado de productos, cursos, contenidos o servicios con evaluación offline rigurosa.
- **Sensoramiento remoto:** clasificación de cobertura, detección de cambios o segmentación de infraestructura a partir de imágenes satelitales.
- **Graph ML:** detección de fraude, análisis de relaciones, recomendación basada en grafos o predicción de enlaces.
- **IA generativa:** prototipo con modelos de difusión o LLMs, con evaluación sistemática y análisis riguroso de riesgos.
- **Solución integrada:** combinación de módulos justificada por el problema (ej. NLP + recomendación, visión + MLOps, sensoramiento remoto + segmentación).

Entregables mínimos

1. Planteamiento del problema: contexto, usuario, criterio de éxito y métricas.
2. Dataset documentado con diccionario de datos y consideraciones éticas.
3. Baseline simple y modelo principal con comparación justificada.
4. Análisis de errores: cuándo falla el modelo, por qué y qué se podría mejorar.
5. Prototipo funcional: API, demo local o notebook interactivo ejecutable.
6. Repositorio organizado con instrucciones claras de reproducción.
7. Presentación de 15 minutos para audiencia técnica y ejecutiva.

8. Ruta de aprendizaje

Los módulos siguen una secuencia diseñada para que cada uno construya sobre los anteriores:

1. **Módulo 1 — Deep Learning Fundamentals:** base técnica común. Todo el programa asume este módulo completado.
2. **Módulo 2 — NLP:** primera aplicación de representaciones aprendidas; dominio de alto impacto práctico.
3. **Módulo 3 — Recomendación:** conecta con NLP a través de la idea de vectores latentes en matrices de interacción.
4. **Módulo 4 — Visión:** segunda aplicación central; asume comprensión de CNN del Módulo 1.
5. **Módulo 5 — MLOps:** se ubica a mitad del programa para que los participantes puedan aplicarlo al proyecto final desde ese punto.
6. **Módulo 6 — Aprendizaje por Refuerzo:** requiere madurez técnica; se ubica una vez que el participante ha entrenado modelos en varios dominios.
7. **Módulo 7 — Difusión:** especialización generativa; se apoya en la comprensión de U-Net del Módulo 4.
8. **Módulo 8 — Sensoramiento Remoto:** aplicación de visión a datos geoespaciales; extiende directamente el Módulo 4.
9. **Módulo 9 — Graph ML:** cierra el programa con el dominio más distinto estructuralmente, conectando con recomendación y NLP.

9. Bibliografía y recursos

- Goodfellow, I., Bengio, Y. y Courville, A. *Deep Learning*. MIT Press.
- Sutton, R. y Barto, A. *Reinforcement Learning: An Introduction*. MIT Press.
- Jurafsky, D. y Martin, J. *Speech and Language Processing* (3.^a ed., borrador abierto).
- Aggarwal, C. *Recommender Systems: The Textbook*. Springer.
- Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly.
- Hamilton, W. *Graph Representation Learning*. Morgan & Claypool.
- Ho, J. et al. *Denoising Diffusion Probabilistic Models*. NeurIPS 2020.
- Documentación oficial de PyTorch, TensorFlow/Keras y Hugging Face Transformers.
- Documentación oficial de FastAPI, Docker, MLflow y DVC.
- Documentación oficial de NetworkX y PyTorch Geometric.
- Documentación oficial de Rasterio, GeoPandas y librerías geoespaciales en Python.