



1821 Universidad de Buenos Aires

Resolución Consejo Directivo

Número:

Referencia: EX-2025-02356031- -UBA-DMESA#FCEN - POSGRADO – Sesión
23/06/2025

VISTO:

La nota presentada por la Dirección del Departamento de Computación, mediante la cual eleva la información del curso de posgrado **Curso de Actualización en Visión Computacional Basada en Redes Neuronales Artificiales (MDM)** para el año 2025,

CONSIDERANDO:

lo actuado por la Comisión de Doctorado,

lo actuado por este Cuerpo en la sesión realizada el día 23 de junio de 2025,

en uso de las atribuciones que le confiere el Artículo 113° del Estatuto Universitario,

**EL CONSEJO DIRECTIVO DE LA FACULTAD
DE CIENCIAS EXACTAS Y NATURALES**

RESUELVE:

ARTÍCULO 1°: Aprobar el nuevo curso de posgrado **Curso de Actualización en Visión Computacional Basada en Redes Neuronales Artificiales (MDM)** de 50 horas de duración, que será dictado por el Dr. José Ignacio Orlando.

ARTÍCULO 2°: Aprobar el programa del curso de posgrado **Curso de Actualización en Visión Computacional Basada en Redes Neuronales Artificiales (MDM)** que como anexo forma parte de la presente Resolución, para su dictado en el segundo cuatrimestre de 2025.

ARTÍCULO 3°: Aprobar un puntaje máximo de dos y medio (2,5) puntos para la Carrera de Doctorado.

ARTÍCULO 4°: Establecer un arancel de **CATEGORÍA BAJA**, estableciendo que dicho arancel estará sujeto a los descuentos y exenciones estipulados mediante la Resolución CD N.º 1072/19. Disponer que los fondos recaudados ingresen en la cuenta presupuestaria habilitada para tal fin, y sean utilizados de acuerdo a la Resolución 072/03.

.

ARTÍCULO 5°: Disponer que, de no mediar modificaciones en el programa, la carga horaria y el arancel, el presente Curso de Posgrado tendrá una vigencia de cinco (5) años a partir de la fecha de la presente Resolución.

ARTÍCULO 6°: Comuníquese a todos los Departamentos Docentes, a la Dirección de Estudiantes y Graduados, a la Biblioteca de la FCEyN y a la Secretaría de Posgrado con copia del programa incluida. Cumplido, pase COMPUTACION#FCEN y resérvese.

ANEXO

PROGRAMA

Al término del curso, se espera que los/as estudiantes hayan adquirido los conocimientos necesarios para:

- Modelar soluciones basadas en redes neuronales artificiales a problemas clásicos de visión (clasificación, segmentación semántica, detección de objetos y generación de imágenes).
- Implementar, entrenar y evaluar redes neuronales artificiales, en especial redes neuronales convolucionales.
- Interpretar bibliografía reciente del área, tanto desde el punto de vista conceptual como en lo que refiere a su notación matemática.
- Desarrollar códigos propios o extender librerías de terceros basadas en el *framework* de aprendizaje profundo *PyTorch* y en el lenguaje de programación Python.
- Hacer un uso eficiente del hardware de computación de alto desempeño (en particular, unidades de procesamiento gráfico o GPUs).

Temario:

- Unidad 1. Introducción a la visión computacional ¿Qué es la visión computacional? ¿Por qué es necesaria? Enfoques de abordaje multidisciplinario de la visión computacional. Aplicaciones reales de la visión artificial. Problemas típicos de visión: clasificación de imágenes, detección de objetos, segmentación semántica, generación de imágenes artificiales. Breve repaso histórico de la visión computacional. Redes neuronales convolucionales.
- Unidad 2. Imágenes en *PyTorch*. Representación en máquina de las imágenes: matrices y tensores. Intensidad de gris. Colores. Espacios de color. Formatos de archivo de imagen. Compresión de imágenes y artefactos. El *framework* para aprendizaje profundo *PyTorch*. ¿Cómo se estructura el código *PyTorch*? Tensores. Imágenes como tensores. Lotes de imágenes en *PyTorch*.
- Unidad 3. Introducción a las redes neuronales. ¿Qué es el aprendizaje profundo?

Las redes neuronales. Aspectos biológicos. La neurona artificial. Arquitectura de un perceptrón multicapa. Funciones de activación no lineal. Aprendizaje de redes neuronales. La función de pérdida. Optimización por gradiente descendente estocástico. *Backpropagation*. Separación de datos en entrenamiento, validación y test. Curvas de aprendizaje. *Learning rate* y monitoreo. *Early stopping*. Aumentación de imágenes. Regularización.

- Unidad 4. Fundamentos para el entrenamiento de modelos en *PyTorch*. El *training loop*. La clase `torch.nn.Module`: funciones de pérdida, componentes de una red neuronal. La clase `torch.optim`: optimizadores por gradiente descendente. Lectura de datos mediante `torch.utils.data`. Organización del desarrollo en *PyTorch*.
- Unidad 5. Clasificación de imágenes con redes neuronales. Definición del problema de clasificación. Desafíos. Ejemplos. Imágenes y etiquetas. Preprocesamiento: mejoras de contraste, normalización/estandarización. El perceptrón multicapa como clasificador de imágenes. Funciones de pérdida. Implementación en *PyTorch*: *DataLoader*, transformaciones, arquitectura. Límites del perceptrón multicapa. Convoluciones. Capas convolucionales. Capas de *pooling*. Redes neuronales convolucionales. Métricas de evaluación.
- Unidad 6. Evolución histórica de las redes neuronales convolucionales de clasificación. Funciones de activación no lineal alternativas. Manipulación de la capacidad de una red. Underfitting y overfitting. Regularización: *early stopping*, *weight decay*, dropout, *data augmentation*. Normalización de activaciones por lotes o instancias. Incremento de la profundidad de una red. Ajuste de hiperparámetros. Estrategias para entrenamiento eficiente. Conexiones residuales. VGG, *GoogLeNet (Inception)*, *ResNet*, *MobileNet*, *EfficientNet*.
- Unidad 7. Transferencia de aprendizaje y métodos de atribución para problemas de clasificación. Motivación. Redes neuronales convolucionales como extractoras de características. *Fine-tuning*. Métodos de atribución. ¿Por qué usarlos? ¿Cómo funcionan? Métodos basados en gradiente. Métodos basados en oclusiones.
- Unidad 8. Segmentación semántica de imágenes. Definición del problema de segmentación. Diferencias con clasificación. Imágenes y etiquetas. Requerimientos para un modelo de segmentación. Funciones de pérdida. Redes totalmente convolucionales. Capas de *unpooling*. U-Nets. Métricas de evaluación. Resumen histórico.
- Unidad 9. Conceptos avanzados de segmentación de imágenes Problemas de segmentación desbalanceados. Segmentación 3D. Ensembles. Calibración de modelos de segmentación. Funciones de pérdida. Uso de modelos de segmentación semántica preentrenados en *PyTorch*. Cómo entrenar modelos

propios de segmentación.

- Unidad 10. Detección de objetos en imágenes. Definición del problema de detección de objetos. Diferencias con clasificación y segmentación. Las funciones de pérdida multitarea. Primeras ideas. *Region proposals*. R-CNN, *Fast R-CNN*, *Faster R-CNN*, *Mask R-CNN*. Métricas de evaluación. Resumen histórico.
- Unidad 11. Generación de imágenes artificiales. Definición del problema de generación de imágenes. Autocodificadores. Autocodificadores para eliminación de ruido. Autocodificadores variacionales. Redes neuronales generativas adversarias (GANs). *Conditional GANs*. *pix2pix*. *CycleGANs*.

BIBLIOGRAFIA

- Rowel Atienza. *Advanced Deep Learning with Keras*. Packt. 2018.
- Michael Avendi. *PyTorch Computer Vision Cookbook*. Packt. 2020.
- François Chollet. *Deep Learning with Python*. Manning. 2018.
- Adrian Rosebrock. *Deep Learning for Computer Vision*. pyimagesearch. 2017.
- Eli Stevens, Luca Antiga, Thomas Viehmann. *Deep Learning with PyTorch*. Manning. 2020.
- Vishnu Subramanian. *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch*. Packt. 2018.
- John Hany, Greg Walters. *Hands-On Generative Adversarial Networks with PyTorch 1.x*. Packt. 2019.