

Apuntes de Clase

Rayforth Brenes Rodríguez
Ingeniería en Computación
Tecnológico de Costa Rica

Clase del 03 de Junio, 2025

Resumen

Este documento presenta un resumen comprensivo de los temas abordados en la clase sobre técnicas avanzadas de machine learning. Se exploran metodologías para el desarrollo de sistemas conversacionales inteligentes mediante RAG (Retrieval Augmented Generation), implementación de modelos de clasificación usando transfer learning con arquitecturas autoencoder, y técnicas de reducción dimensional a través del Análisis de Componentes Principales. Adicionalmente, se detallan los requerimientos para las asignaciones académicas correspondientes.

1. Introducción

La sesión académica se centró en tres pilares fundamentales del aprendizaje automático moderno: el desarrollo de asistentes conversacionales inteligentes, la aplicación de transfer learning en problemas de clasificación de imágenes, y las técnicas de reducción dimensional. Estos temas se abordan tanto desde una perspectiva teórica como práctica, con énfasis en las implementaciones requeridas para las asignaciones del curso.

2. Sistema Conversacional con RAG

2.1. Marco Conceptual

El objetivo principal consiste en implementar un asistente conversacional capaz de responder consultas basándose en documentación académica mediante la técnica de Retrieval Augmented Generation (RAG) complementada con agentes especializados.

2.2. Arquitectura del Sistema RAG

El proceso de implementación sigue una metodología estructurada:

1. **Extracción y Preparación:** Procesamiento completo del contenido textual de los documentos fuente
2. **Enriquecimiento de Metadatos:** Incorporación de información contextual relevan-

te (autoría, fechas, identificadores de archivo)

3. **Segmentación Inteligente:** División del contenido en fragmentos (chunks) con parámetros configurables de tamaño y solapamiento
4. **Vectorización Semántica:** Transformación de cada fragmento mediante modelos de embeddings textuales
5. **Almacenamiento Vectorial:** Persistencia de representaciones vectoriales en base de datos especializada (ej. FAISS)
6. **Recuperación por Similitud:** Búsqueda de fragmentos relevantes usando métricas de similitud coseno
7. **Generación Contextualizada:** Utilización de modelos de lenguaje para generar respuestas coherentes

2.3. Implementación de Agentes

El sistema requiere la implementación de dos herramientas especializadas:

- **Herramienta RAG:** Encargada del proceso de recuperación desde la base vectorial (implementación obligatoria con LangChain)
- **Herramienta de Búsqueda Web:** Para consultas que requieren información externa

Un agente orquestador, basado en LLM, toma decisiones sobre qué herramienta utilizar según el contexto de la consulta, manteniendo memoria conversacional durante la sesión.

2.4. Consideraciones de Implementación

- Modelo recomendado: GPT-3.5-turbo (balance costo-eficiencia)
- Interfaz de usuario: Streamlit para ejecución web
- Preprocesamiento crítico para manejo de caracteres especiales
- Búsquedas web únicamente bajo solicitud explícita del usuario

3. Clasificación con Transfer Learning

3.1. Planteamiento del Problema

El proyecto se enfoca en clasificación multi-clase de especies de mariposas utilizando transfer learning con arquitecturas autoencoder, específicamente explorando el impacto del pre-entrenamiento no supervisado en el rendimiento final.

3.2. Configuración del Dataset

- **Fuente:** Dataset "Butterflies 100 species"
- **Selección:** 30 especies con mayor representación muestral
- **Redistribución:** Transferencia de 20 muestras por especie desde entrenamiento hacia testing
- **Resolución:** Imágenes redimensionadas a 224×224 píxeles (mínimo 128×128)

3.3. Stack Tecnológico Requerido

PyTorch Lightning: Framework para estructuración del código de entrenamiento

- LightningDataModule para gestión de datos
- LightningModule para definición de modelos

- Implementación de métodos: training_step, testing_step, configure_optimizers

Hydra: Sistema de gestión de configuraciones mediante archivos YAML

Weights & Biases (WandB): Plataforma de monitoreo y visualización de experimentos

Callbacks: Early stopping y ajuste dinámico de learning rate

3.4. Diseño Experimental

3.4.1. Hipótesis Principal: Efectividad del Pre-entrenamiento

Pregunta de investigación: ¿Puede el pre-entrenamiento de un autoencoder con datos no etiquetados mejorar el rendimiento de clasificación comparado con entrenamiento desde cero?

Configuraciones experimentales:

- Configuración A: 70 % datos sin etiquetas / 30 % etiquetados
- Configuración B: 90 % datos sin etiquetas / 10 % etiquetados

Metodología:

Fase 1 - Pre-entrenamiento del Autoencoder:

- Arquitectura base: U-Net con skip connections obligatorias
- Entrenamiento no supervisado con datos sin etiquetas
- Posibilidad de modificaciones arquitectónicas

Fase 2 - Entrenamiento de Clasificadores:

- Modelo Baseline:** Entrenamiento desde cero usando arquitectura del encoder sin pesos pre-entrenados
- Modelo Transfer Learning Fijo:** Encoder pre-entrenado congelado + clasificador entrenable
- Modelo Fine-tuning:** Encoder pre-entrenado + clasificador, ambos re-entrenables

3.4.2. Hipótesis Secundaria: Denoising y Análisis de Representaciones

Objetivo: Evaluar el impacto del ruido (salt-and-pepper) en la calidad de las representaciones latentes.

Protocolo:

1. Entrenamiento de denoising autoencoder con ruido artificial
2. Visualización del espacio latente mediante t-SNE
3. Aplicación de clustering K-Means sobre representaciones latentes
4. Análisis cualitativo de agrupaciones resultantes

3.5. Entregables y Evaluación

- Selección de los tres mejores modelos basada en métricas de rendimiento
- Implementación de cuantización de modelos
- Análisis comparativo de latencia, tamaño y rendimiento
- Documentación completa en Jupyter Notebook
- Informe técnico detallado

4. Análisis de Componentes Principales (PCA)

4.1. Fundamentos Teóricos

PCA constituye una técnica fundamental de reducción dimensional que identifica direcciones de máxima variabilidad en los datos mediante transformaciones lineales ortogonales.

4.2. Objetivos de PCA

- **Reducción Dimensional:** Disminución del número de variables preservando información relevante
- **Maximización de Varianza:** Identificación de direcciones de mayor variabilidad
- **Eliminación de Correlaciones:** Creación de variables independientes (ortogonales)
- **Selección de Características:** Identificación automática de variables informativas

4.3. Conceptos Matemáticos Fundamentales

Matriz de Covarianza: Representa la variabilidad conjunta entre variables

$$\text{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Eigenvalores y Eigenvectores: Solución del problema característico

$$A\mathbf{v} = \lambda\mathbf{v}$$

donde A es la matriz de covarianza, \mathbf{v} son los eigenvectores y λ los eigenvalores correspondientes.

4.4. Algoritmo de Implementación

1. **Estandarización:** Normalización de características ($\mu = 0, \sigma^2 = 1$)
2. **Cálculo de Matriz de Covarianza:** $\Sigma = \frac{1}{n-1} X^T X$
3. **Descomposición Eigenvalue:** Resolución de $\det(\Sigma - \lambda I) = 0$
4. **Selección de Componentes:** Ordenamiento por eigenvalores y selección de los k principales
5. **Transformación de Datos:** Proyección $Y = XW_k$ donde W_k contiene los k eigenvectores principales

4.5. Aplicaciones y Limitaciones

Casos de Uso Óptimos:

- Datasets con alta dimensionalidad y variables correlacionadas
- Necesidad de variables independientes garantizadas
- Visualización de datos multidimensionales

Limitaciones Principales:

- Los componentes resultantes carecen de interpretabilidad física
- Asume relaciones lineales entre variables
- Sensible a la escala de las variables originales

5. Conclusiones

La clase abordó tres metodologías complementarias que representan el estado del arte en inteligencia artificial aplicada. La implementación de sistemas RAG demuestra la evolución hacia asistentes conversacionales más contextualizados y precisos. El uso de transfer learning con autoencoders presenta una aproximación innovadora para aprovechar datos no etiquetados en problemas de clasificación supervisada. Finalmente, PCA mantiene su relevancia como técnica fundamental para el preprocesamiento y análisis exploratorio de datos.

La integración de estas técnicas en proyectos prácticos consolida el aprendizaje teórico y desarrolla competencias esenciales para la investigación y desarrollo en machine learning.