
RESULTADO DE APRENDIZAJE

RdA de la asignatura:

- **RdA 2:** Aplicar modelos de aprendizaje automático supervisado y no supervisado, así como su validación y optimización, en la resolución de problemas tanto reales como simulados.

RdA de la clase:

- Comprender el método de retropropagación y su deducción matemática.
- Revisar las técnicas de optimización en el entrenamiento de redes neuronales.

INTRODUCCIÓN

Pregunta inicial: Si una red tiene miles de conexiones, ¿cómo puede saber exactamente qué conexión ajustar y en qué dirección, sin “probar a ciegas” una por una?

DESARROLLO

Actividad 1: Deducción del método de retropropagación

En esta actividad se explicará mediante clase magistral el método de retropropagación, incluyendo su propósito en redes neuronales y la deducción paso a paso del algoritmo usando la regla de la cadena para calcular gradientes eficientemente.

¿Cómo lo haremos?

- **Clase magistral:** Explicación detallada del método de retropropagación:

- Introducción a la retropropagación y su propósito en redes neuronales (optimizar pesos mediante gradiente descendente).
- Deducción paso a paso del algoritmo: desde la función de pérdida hasta el cálculo de gradientes usando la regla de la cadena.
- Forward pass (propagación hacia adelante) y backward pass (retropropagación de errores).

Actividad 2: Regularización y optimización en redes neuronales

En esta actividad se presentarán mediante clase magistral y ejercicio práctico en cuaderno de Jupyter las técnicas de regularización (L1, L2, Dropout, Early stopping) y diferentes optimizadores (SGD, Adam, RMSProp), evaluando su impacto en el rendimiento del modelo.

¿Cómo lo haremos?

- **Clase magistral:** Presentación de conceptos clave:

- Regularización: L1 (Lasso), L2 (Ridge) y Dropout (desactivación aleatoria de neuronas).
- Early stopping como estrategia para evitar el sobreajuste (detener entrenamiento cuando validation loss deja de mejorar).
- Diferentes optimizadores: SGD (Stochastic Gradient Descent), Adam (Adaptive Moment Estimation), RMSProp, y sus características.
- Learning rate adaptativo y su impacto en la convergencia del entrenamiento.

- **Implementación en Python:** Los estudiantes accederán a un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [15-Entrenamiento-Redes.ipynb](#).

- **Ejercicio práctico:** Implementación guiada de:

- Configurar y entrenar una red neuronal básica.
- Aplicar técnicas de regularización y evaluar resultados.
- Comparar el rendimiento utilizando diferentes optimizadores.
- Implementar early stopping y observar su impacto en el sobreajuste.

CIERRE

Verificación de aprendizaje:

1. ¿Cómo se utiliza la regla de la cadena en el cálculo de gradientes durante la retropropagación?
2. ¿Cómo afecta la regularización al sobreajuste y cuáles son las diferencias entre L1, L2 y Dropout?
3. ¿Cuáles son las diferencias principales entre SGD y Adam como optimizadores?

Preguntas tipo entrevista:

1. Tu red neuronal tiene un validation loss que oscila erráticamente durante el entrenamiento, mientras que el training loss disminuye suavemente. ¿Qué podría estar causando esto y cómo lo resolverías?
2. Estás entrenando una red muy profunda (50+ capas) y los gradientes en las primeras capas son extremadamente pequeños ($\approx 10^{-10}$). ¿Qué está pasando y cómo lo solucionarías?

Pregunta de investigación:

1. ¿Qué técnicas de regularización funcionan mejor en diferentes escenarios?
2. ¿Cómo podrías mejorar el desempeño de una red neuronal sin cambiar su arquitectura?
3. ¿Cómo afecta el tamaño del dataset al rendimiento del modelo y la necesidad de regularización?

Para la próxima clase: Realizar la Clase invertida: Redes Neuronales, disponible en el aula virtual y aquí: [04Est-ArbolDecision.pdf](#).