

# **UNIVERSIDAD NACIONAL DEL ALTIPLANO**

**FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,  
ELECTRÓNICA Y SISTEMAS**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**“ALGORITMOS GENÉTICOS EN APRENDIZAJE DE MÁQUINA:  
FEATURE SELECTION, OPTIMIZACIÓN DE HIPERPARÁMETROS Y  
NEUROEVOLUCIÓN”**

**CURSO:**

**APRENDIZAJE MÁQUINA**

**DOCENTE:**

**FERNANDEZ CHAMBI MAYENKA**

**PRESENTADO POR:**

**CCALLO CHAMBI MARK WILL**

**CUYO ZAMATA ROBERT**

**QUISPE TICONA ANGEL PEDRO**

**SEMESTRE: OCTAVO**

**2025 – I**

## “ALGORITMOS GENÉTICOS EN APRENDIZAJE DE MÁQUINA: FEATURE SELECTION, OPTIMIZACIÓN DE HIPERPARÁMETROS Y NEUROEVOLUCIÓN”

REPOSITORIO: <https://github.com/MarkTWill/Algoritmos-gen-ticos-git>

### Objetivo

Comprender la aplicación de los algoritmos genéticos en el aprendizaje de máquina, a través de ejemplos prácticos en diferentes contextos. La actividad se centra en tres áreas: selección de características, optimización de hiperparámetros y neuroevolución. Cada ejemplo demuestra cómo se aplica el ciclo del algoritmo genético: representación, inicialización, función de aptitud, selección, cruzamiento, mutación y criterio de terminación.

### 1. Algoritmo Genético para Feature Selection

En la selección de características, los algoritmos genéticos ayudan a identificar qué variables del conjunto de datos aportan más información al modelo. Esto no solo mejora la precisión, sino que también reduce el tiempo de entrenamiento y evita el sobreajuste.

Ciclo del algoritmo genético aplicado:

- Representación: cromosomas binarios donde 1 = característica seleccionada y 0 = descartada.
- Inicialización: población inicial de cromosomas generada aleatoriamente.
- Función de aptitud: basada en la precisión de un modelo SVM usando solo las características seleccionadas.
- Selección: método de torneo.
- Cruzamiento: de un punto para intercambiar características.
- Mutación: inversión aleatoria de bits.
- Terminación: número de generaciones o convergencia en el fitness.

Ejemplo: en el dataset Iris, el AG puede reducir de 4 características iniciales a 2, manteniendo un nivel de precisión similar o superior al modelo con todas las variables.

### 2. Algoritmo Genético para Hyperparameter Optimization

Los **hiperparámetros** son valores definidos antes del entrenamiento que afectan el rendimiento del modelo, como **C**, **gamma** o el tipo de **kernel** en un SVM. Escogerlos manualmente es complejo y consume tiempo, por lo que los algoritmos genéticos ofrecen una forma eficiente de encontrar combinaciones óptimas.

Ciclo del algoritmo genético aplicado:

- **Representación:** cromosomas con valores de hiperparámetros.
- **Inicialización:** población generada aleatoriamente.
- **Función de aptitud:** precisión en validación cruzada.
- **Selección:** por torneo o ruleta.
- **Cruzamiento:** mezcla de configuraciones.

- **Mutación:** cambio aleatorio de un hiperparámetro.
- **Terminación:** límite de generaciones o mejora mínima.

Ejemplo: en un **SVM**, el AG puede encontrar que  $[C=1, \text{gamma}=0.01, \text{kernel}=\text{linear}]$  logra la mayor exactitud en los datos de prueba.

### 3. Algoritmo Genético para Neuroevolution

Consiste en aplicar un algoritmo genético (AG) para optimizar arquitecturas de redes neuronales. El objetivo es demostrar cómo los principios evolutivos, como selección, cruce y mutación, pueden guiar la búsqueda de configuraciones más eficientes de modelos de aprendizaje profundo.

Cada individuo (cromosoma) de la población representa una red neuronal candidata, definida por el número de neuronas en dos capas ocultas y la tasa de aprendizaje. A partir de una población inicial aleatoria, se evalúa el desempeño de cada red entrenándola brevemente sobre el dataset MNIST, tomando como métrica de aptitud el accuracy alcanzado.

Posteriormente, se aplican las fases clásicas de los algoritmos genéticos:

- Selección por torneo, eligiendo a los mejores individuos.
- Cruce, combinando parámetros de dos padres para generar descendencia.
- Mutación aleatoria, que introduce diversidad en la población.

Este proceso evolutivo se repite por varias generaciones hasta identificar la red más apta, es decir, aquella con mayor precisión en la clasificación.

El código desarrollado muestra de manera clara y práctica cómo el algoritmo genético puede automatizar el diseño de redes neuronales, convirtiéndose en una técnica poderosa dentro de la inteligencia artificial evolutiva.

### Conclusión

En conclusión, los AG muestran cómo la inspiración en la biología evolutiva puede aplicarse exitosamente a la inteligencia artificial, ofreciendo soluciones innovadoras a problemas complejos. Los ejemplos desarrollados son prácticos, replicables y demuestran la relevancia de los algoritmos genéticos en el contexto actual del aprendizaje de máquina.