INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
INTELIGENCIA ARTIFICIAL
ING. EN SISTEMAS COMPUTACIONALES

Laboratorio 11: Redes Neuronales

Alumno: Hurtado Morales Emiliano - 2021630390

Maestro: Andrés García Floriano

Grupo: 6CV3

Fecha de entrega: 28/11/2024

Ciclo Escolar: 2025 – 1

Introducción

El presente laboratorio se enfocó en el análisis y comparación de dos algoritmos de clasificación supervisada ampliamente utilizados en aprendizaje automático: el Perceptrón Multicapa (MLP) y la Red Neuronal RBF (simulada mediante SVM con kernel RBF). Estos algoritmos fueron aplicados a tres datasets de referencia disponibles en la biblioteca scikit-learn: Iris, Wine, y Breast Cancer. Los objetivos principales del laboratorio fueron evaluar el desempeño de estos clasificadores bajo diferentes métodos de validación y analizar su capacidad para clasificar correctamente las muestras en sus respectivas clases, considerando tanto la precisión global como los errores de clasificación observados en las matrices de confusión.

Los datasets seleccionados representan problemas de clasificación supervisada de distinta complejidad:

- **Iris:** Un problema sencillo con clases bien separadas en un espacio de baja dimensionalidad.
- **Wine:** Un problema intermedio, con una mayor cantidad de características y relaciones algo más complejas entre las clases.
- **Breast Cancer:** Un problema más desafiante debido a su naturaleza binaria y la presencia de características altamente correlacionadas que complican la separación entre clases.

Para evaluar la efectividad de los algoritmos, se utilizaron tres métodos de validación: Hold-Out (70/30), 10-Fold Cross-Validation, y Leave-One-Out (LOO). Cada método proporciona una perspectiva distinta sobre la generalización de los modelos y permite observar su desempeño bajo diferentes condiciones de partición del dataset. Este análisis es fundamental para comprender cómo los algoritmos responden a datos nuevos y para identificar sus fortalezas y limitaciones en distintos contextos.

Explicación de los Algoritmos

1. Perceptrón Multicapa (MLP)

El Perceptrón Multicapa (MLP, por sus siglas en inglés) es un modelo de red neuronal artificial diseñado para resolver problemas de clasificación y regresión. Se basa en un conjunto de capas de neuronas conectadas entre sí, donde:

- **Capa de entrada:** Recibe las características del dataset (atributos de las muestras).
- **Capas ocultas:** Procesan la información mediante funciones de activación no lineales.
- **Capa de salida:** Genera las predicciones (clasificaciones).

```
"Perceptrón Multicapa": MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000,
random_state=42),
```

- **Capa oculta:** Una única capa de 100 neuronas.
- **Función de activación:** ReLU (Rectified Linear Unit), que introduce no linealidad al modelo.
- **Optimización:** El algoritmo adam, un método basado en gradientes estocásticos con momentos adaptativos, que combina eficiencia computacional y convergencia estable.
- **Máximo de iteraciones:** 1000, para asegurar la convergencia del modelo durante el entrenamiento.

Componentes Principales:

1. **Neurona Artificial:** Cada neurona aplica una operación matemática básica:

$$z = w1x1 + w2x2 + \cdots + wnxn + b$$

donde wi son los pesos, xi son las características de entrada, y b es el sesgo. El resultado z se pasa a través de una función de activación para introducir no linealidad:

$$a = ReLU(z) \text{ o } a = sigmoide(z)$$

2. **Funciones de Activación:**
    - **ReLU (Rectified Linear Unit):**

$$f(z) = max(0,z)$$

    - Es la función de activación utilizada más comúnmente en capas ocultas. Permite una convergencia más rápida al evitar problemas como el "desvanecimiento del gradiente".
    - **Softmax (en la capa de salida):** En tareas de clasificación multiclase, se usa Softmax para convertir las salidas en probabilidades:

$$P(y = i|x) = e^{zi} / \sum_{j=1}^{C} e^{zj}$$

3. **Entrenamiento:**
    - **Retropropagación del error:** Es un algoritmo que ajusta los pesos de la red minimizando una función de pérdida (generalmente, la entropía cruzada para clasificación). Utiliza el gradiente descendente para actualizar los parámetros:

$$wi \leftarrow wi - \eta \partial L / \partial wi$$

- donde η es la tasa de aprendizaje y L es la función de pérdida.

Ventajas:

- Puede modelar relaciones no lineales complejas entre características.
- Es altamente flexible al permitir personalizar la arquitectura (número de capas y neuronas).

Limitaciones:

- Requiere un ajuste cuidadoso de hiperparámetros como la tasa de aprendizaje (η), el número de capas y neuronas.
- Puede ser computacionalmente costoso para datasets grandes o con muchas dimensiones.

2. Red Neuronal RBF (Simulada con SVM y Kernel RBF)

Una Red Neuronal de Función de Base Radial (RBF, por sus siglas en inglés) es un modelo que utiliza funciones gaussianas para medir la similitud entre las entradas y ciertos "centros" o puntos de referencia. Su arquitectura se compone de:

- **Capa de entrada:** Recibe las características del dataset.
- **Capa oculta:** Aplica funciones RBF centradas en puntos específicos.
- **Capa de salida:** Combina las activaciones para realizar la clasificación.

```
# Clasificador RBF (simulado usando SVM con kernel RBF)
def clasificador_rbf():
    """
    Devuelve un clasificador basado en SVM con kernel RBF.
    El kernel RBF permite manejar problemas no lineales.
    """
    return SVC(kernel='rbf', gamma='scale', probability=True)
```

- **Kernel RBF (Radial Basis Function):** Una función de similitud basada en una distribución gaussiana, que permite modelar relaciones no lineales.
- **Parámetro gamma:** Ajustado automáticamente con la opción gamma='scale', que calcula el valor en función de la varianza de las características del dataset.
- **Soporte de probabilidad:** Se habilitó la opción probability=True para realizar cálculos probabilísticos que pueden ser útiles en análisis posteriores.

Kernel RBF en SVM:

En este laboratorio, se simuló una red RBF utilizando un clasificador **SVM con kernel RBF**. En este contexto:

- **Kernel RBF:** Mide la similitud entre dos puntos x y x′ utilizando una función gaussiana:

$$K(x, x') = e^{-\gamma \|x - x'\|^2}$$

- donde γ controla la amplitud del kernel.

Componentes Principales:

1. **Hipótesis de Separación:** Las SVM buscan encontrar un hiperplano que maximice el margen entre las clases en el espacio de características. En problemas no lineales, el kernel RBF transforma los datos a un espacio de mayor dimensionalidad donde las clases son más separables.
2. **Parámetro γ:**
    - Si γ es pequeño, el modelo considera relaciones globales, lo que puede llevar a un subajuste (underfitting).
    - Si γ es grande, el modelo se enfoca en relaciones locales, lo que puede llevar a un sobreajuste (overfitting).
3. **Margen Máximo:** El modelo encuentra el hiperplano que maximiza la distancia entre las clases, minimizando al mismo tiempo los errores de clasificación.

Ventajas:

- Excelente desempeño en problemas no lineales.
- Requiere menos datos de entrenamiento en comparación con redes neuronales profundas.

Limitaciones:

- Sensible a los valores de γ y el parámetro de regularización C.
- Computacionalmente intensivo para datasets grandes debido al cálculo del kernel para cada par de muestras.

3. Métodos de Validación: Para evaluar los algoritmos, se emplearon tres métodos de validación ampliamente aceptados:

- **Hold-Out (70/30):** Divide el dataset en 70% para entrenamiento y 30% para prueba. Este método es rápido y sencillo de implementar, pero su desempeño puede depender de la aleatoriedad de la partición inicial.
- **10-Fold Cross-Validation:** Divide los datos en 10 subconjuntos de igual tamaño. En cada iteración, uno de estos subconjuntos se utiliza para prueba y los otros nueve para entrenamiento. Este método reduce la variabilidad entre particiones y proporciona un mejor estimador de la generalización.
- **Leave-One-Out (LOO):** Realiza tantas iteraciones como muestras en el dataset, utilizando una muestra para prueba y el resto para entrenamiento en cada iteración. Aunque es computacionalmente intensivo, maximiza el uso de los datos y produce resultados precisos en problemas con pocos datos.

Resultados Esperados

Se anticipó que los clasificadores presentarían las siguientes características de desempeño:

1. **Dataset Iris:**

```
Dataset: Iris

Clasificador: Perceptrón Multicapa

Validación: Hold-Out

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 19                    0                   0
Verdadero versicolor              0                   13                   0
Verdadero virginica               0                    0                  13
Accuracy: 1.0000

Validación: 10-Fold

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 49                    1                   0
Verdadero versicolor              0                   47                   3
Verdadero virginica               0                    2                  48
Accuracy: 0.9600

Validación: Leave-One-Out

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 49                    1                   0
Verdadero versicolor              0                   47                   3
Verdadero virginica               0                    3                  47
Accuracy: 0.9533
```

```
Clasificador: Red Neuronal RBF

Validación: Hold-Out

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 19                    0                   0
Verdadero versicolor              0                   13                   0
Verdadero virginica               0                    0                  13
Accuracy: 1.0000

Validación: 10-Fold

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 50                    0                   0
Verdadero versicolor              0                   47                   3
Verdadero virginica               0                    3                  47
Accuracy: 0.9600

Validación: Leave-One-Out

Matriz de Confusión:
                    Predicho setosa  Predicho versicolor  Predicho virginica
Verdadero setosa                 50                    0                   0
Verdadero versicolor              0                   48                   2
Verdadero virginica               0                    3                  47
Accuracy: 0.9667
```

2. **Dataset Wine:**

```
Dataset: Wine

Clasificador: Perceptrón Multicapa

Validación: Hold-Out

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                19                 0                 0
Verdadero class_1                 0                20                 1
Verdadero class_2                 0                 0                14
Accuracy: 0.9815

Validación: 10-Fold

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                59                 0                 0
Verdadero class_1                 0                68                 3
Verdadero class_2                 0                 1                47
Accuracy: 0.9778

Validación: Leave-One-Out

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                59                 0                 0
Verdadero class_1                 0                68                 3
Verdadero class_2                 0                 1                47
Accuracy: 0.9775
```

```
Clasificador: Red Neuronal RBF

Validación: Hold-Out

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                19                 0                 0
Verdadero class_1                 0                21                 0
Verdadero class_2                 0                 1                13
Accuracy: 0.9815

Validación: 10-Fold

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                58                 1                 0
Verdadero class_1                 0                70                 1
Verdadero class_2                 0                 2                46
Accuracy: 0.9775

Validación: Leave-One-Out

Matriz de Confusión:
                  Predicho class_0  Predicho class_1  Predicho class_2
Verdadero class_0                58                 1                 0
Verdadero class_1                 0                70                 1
Verdadero class_2                 0                 1                47
Accuracy: 0.9831
```

3. **Dataset Breast Cancer:**

```
Dataset: Breast Cancer

Clasificador: Perceptrón Multicapa

Validación: Hold-Out

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                 61                2
Verdadero benign                     1              107
Accuracy: 0.9825

Validación: 10-Fold

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                201               11
Verdadero benign                     4              353
Accuracy: 0.9737

Validación: Leave-One-Out

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                202               10
Verdadero benign                     5              352
Accuracy: 0.9736
```

```
Clasificador: Red Neuronal RBF

Validación: Hold-Out

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                 61                2
Verdadero benign                     3              105
Accuracy: 0.9708

Validación: 10-Fold

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                202               10
Verdadero benign                     5              352
Accuracy: 0.9736

Validación: Leave-One-Out

Matriz de Confusión:
                    Predicho malignant  Predicho benign
Verdadero malignant                204                8
Verdadero benign                     5              352
Accuracy: 0.9772
```

4. **Métodos de Validación:**
    - **Hold-Out:** Resultados rápidos pero con mayor variabilidad.
    - **10-Fold Cross-Validation:** Un buen balance entre tiempo de ejecución y confiabilidad.
    - **Leave-One-Out:** Precisión altamente confiable, pero a un costo computacional mayor.

Conclusión

El análisis realizado permitió observar el desempeño relativo de los algoritmos Perceptrón Multicapa y Red Neuronal RBF bajo diferentes condiciones. Los resultados muestran que:

- El MLP es adecuado para problemas lineales o con clases bien definidas, destacándose en el dataset Iris.
- La Red Neuronal RBF sobresale en problemas más complejos, como Breast Cancer, debido a su capacidad para modelar relaciones no lineales.

En cuanto a los métodos de validación, 10-Fold Cross-Validation proporcionó el mejor balance entre confiabilidad y tiempo de ejecución, mientras que Leave-One-Out es más preciso, pero computacionalmente costoso.

Este laboratorio resalta la importancia de seleccionar el clasificador y método de validación adecuados para cada problema. En futuros trabajos, se sugiere explorar técnicas como la optimización de hiperparámetros y el uso de conjuntos de datos con mayor número de clases para evaluar aún más la escalabilidad de los algoritmos.

La comprensión obtenida de este análisis puede servir como base para aplicaciones prácticas de clasificación supervisada en diversos dominios, como diagnóstico médico, análisis de datos financieros o reconocimiento de patrones.