



MÁQUINAS DE VECTORES DE SOPORTE Y PERCEPTRÓN MULTICAPA

Práctica de Laboratorio 3 - 2020II

Tópicos de Inteligencia Artificial

I. Pautas

- Fecha de Entrega: 04/Dic/2020, a través de Moodle.
- Lenguaje de Programación: Python.

II. Implementación (6)

Implemente las siguientes funciones:

1. `Leer_Datos`
2. `Normalizar_Datos`
3. `Calcular_Accuracy`
4. `Crear_k_folds`
5. `Sigmoidal`
6. (0.25) `Calcular_Funcion_Costo`: Recibe la salida de la última capa (i.e., las activaciones de la última capa), y ; y devuelve el valor de la función de Costo asociado a los datos (X, y) .
7. (0.75) `dS`: Recibe los datos D y calcula para dichos datos la derivada de la función sigmoidal.
8. (2.00) `Forward`: Recibe los datos X y un diccionario de parámetros W (i.e., los pesos de cada capa), donde cada elemento del diccionario es una matriz de pesos; y realiza la etapa de propagación. Devuelve un diccionario de activaciones A , donde cada elemento del diccionario son las activaciones de cada capa.
9. (2.50) `Backward`: Recibe los datos de entrada X , las activaciones A de cada capa, un diccionario de parámetros W (i.e., los pesos de cada capa), las salidas y y la tasa de aprendizaje. Ejecuta la etapa de retropropagación. Devuelve los pesos actualizados.
10. (0.50) `Gradiente_Descendiente`: Recibe X , y , un diccionario de parámetros W (i.e., los pesos de cada capa inicializados aleatoriamente), el número de iteraciones (épocas) y la tasa de aprendizaje. Actualiza W usando el gradiente descendiente. Devuelve los valores actualizados de W y el costo de cada iteración.

III. Experimentos (11)

Importante: Para que los experimentos sean calificados sus resultados deben estar dentro del informe.

1. (3) Experimento 1: Buscar los mejores parámetros de entrenamiento para los conjuntos “Propio(+2Clases)” y “Titanic”, usando validación cruzada (*k-fold cross validation*, con $k = 3$). En la validación cruzada se entrena con unos parámetros específicos y se calcula el promedio de los *accuracies* obtenido al ejecutar el algoritmo del gradiente descendiente k veces. En cada vez, uno de los *folds* es usado como conjunto de prueba y el resto, i.e. los otros dos, como conjunto de entrenamiento. Los *folds* son conjuntos disjuntos dos a dos del conjunto de datos original.

Dichos promedios deben ser mostrados en dos tablas, una para “Propio(+2Clases)” y otra para “Titanic”. En cada tabla, se mostrará el *Accuracy* promedio obtenido, al variar los parámetros de entrenamiento, a saber, tasa de aprendizaje, número de épocas, número de capas ocultas (pruebe con 1, 2 y 3 capas ocultas) y número de neuronas por capa (pruebe con diferentes valores según su criterio).

2. (8) Experimento 3: En este experimento debe usar el Scikit-learn de Python para implementar una SVM usando los conjuntos de datos “Propio(+2Clases)” y “Titanic”. Luego, encuentre los mejores valores para sus parámetros usando validación cruzada. Los promedios de los *accuracies* deben ser mostrados en dos tablas, una para cada conjunto. En cada tabla, se mostrará el *Accuracy* promedio obtenido al variar los parámetros de los *kernels*: lineal, polinomial, gaussiano, y el parámetro C .

IV. Informe (3)

Importante: El informe solo recibirá puntaje si está terminado, si está bien redactado y si los experimentos tienen discusión.

El informe debe estar redactado en Latex (plantilla LNCS) y tener la siguiente estructura:

- Título “Práctica de Laboratorio Nro. 3”.
- Datos del alumno conforme la plantilla.
- Introducción
- Implementación: Hacer referencia al código.
- Experimentos y Resultados: La sección debe estar dividida en tantas secciones como experimentos existan y cada experimento debe discutido.
- Conclusiones.
- Referencias: Si las hubieran.