Tarea 2 EL4106 - Semestre Otoño 2019

Profesor: Javier Ruiz del Solar
Auxiliar: Patricio Loncomilla
Ayudantes: Gabriel Azócar, Nicolás Cruz, Francisco Leiva, Giovanni Pais

Fecha enunciado: 17 de Abril
Plazo entrega tarea: 30 de Abril

El objetivo de esta tarea implementar un clasificador de fallas de motores basado en análisis de señales de corriente. Se utilizará la base de datos Dataset for Sensorless Drive Diagnosis Data Set, que es una base de datos tomadas del UC Irvine Machine Learning Repository. Contiene 48 características extraídas de la señal de corriente que alimenta a un motor síncrono. Hay 11 clases. Se seleccionó un subconjunto de la base de datos original debido a su gran tamaño. Para mayor información revise la página del repositorio:

https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis

La base de datos estará disponible en u-cursos.

Se usará tensorflow para entrenar y calibrar un clasificador de fallas y poder observar los efectos de cambiar el tamaño de la red y la función de activación.

Se pide:

1) Teoría:
   a) Defina lo que es una Red Neuronal Artificial, junto con todas sus componentes básicas (neurona, capa de entrada, capa oculta, capa de salida, pesos, bias, función de activación).
   b) Defina lo que es un *batch*, y de qué modo se usa al entrenar la red
   c) ¿Cuál es el número sugerido (heurístico) de neuronas de la capa oculta para una red feed-forward de 1 capa oculta?
   d) Explique lo que es el sobreajuste y cómo se previene.
   e) Defina lo que es el *accuracy*.
   f) Defina la Matriz de Confusión, como calcularla, cómo normalizarla y como interpretarla.

2) En esta tarea se usarán tres conjuntos de datos: entrenamiento, validación y prueba. El conjunto de validación se usará para determinar cuál es la mejor arquitectura. Los archivos de entrenamiento+validación (sensorless_tarea2_train.txt) y el de prueba (sensorless_tarea2_test.txt) se entregan por separado.

3) Dividir la base de datos sensorless_tarea2_train.txt en 2 conjuntos representativos: entrenamiento (80%) y validación (20%). Compruebe la representatividad de éstos, verificando si la proporción de cada clase se mantiene cercana a la proporción del conjunto completo. La base de datos contiene 48 columnas para las características, además de una columna extra que contiene la clase (de 1 a 11, se transforman al rango 0-10 tras ser leídas). Se recomienda representar las clases usando one-hot encoding, lo cual consiste en representar la salida de la red mediante un vector conteniendo las clases. Por ejemplo, [1 0 0 0 0 0 0 0 0 0 0] representa la clase “0”, [0 1 0 0 0 0 0 0 0 0 0] la clase “1”, etcétera.

4) Clasificador Multiclase
   a) Entrenar una red neuronal con el conjunto de entrenamiento, utilizando tensorflow. Utilice una red neuronal de una sola capa oculta con el numero óptimo teórico de neuronas de ésta. Se debe usar función de activación sigmoidal (ya entregada en código base).
   b) Implementar un código para medir el tiempo de entrenamiento.

c) Generar las salidas de la red neuronal para el conjunto de validación y clasificar utilizando el índice del máximo de la salida. Por ejemplo, en [0.1 0.3 0.9 0.1 0.2 0.7 0.6 0.3 0.6 0.4] el máximo es 0.9, que es el tercer índice, por lo tanto se clasifica como clase 2 (ya entregado en código base).
d) Genere una figura que muestre el *accuracy* sobre el conjunto de validación a medida que la red se entrena (se puede graficar después del entrenamiento, almacenando los datos intermedios en un arreglo).
e) Encuentre y analice la Matriz de Confusión usando el conjunto de validación (ya entregado).

5) Análisis
   a) Repita el punto 3) variando la cantidad de neuronas en la capa oculta, usando el conjunto de validación (3 valores distintos)
   b) Usando la mejor arquitectura obtenida, entrene la red usando como no-linealidad de la capa oculta tanto sigmoide como ReLU, evaluando en el conjunto de prueba. Debido a la variabilidad causada por los pesos iniciales, entrene cada variante de la red tres veces (modificando la variable random_state), usando el conjunto de prueba esta vez para evaluar la matriz de confusión.
   c) Compare y concluya sobre los resultados obtenidos. Explique los efectos de variar la cantidad de neuronas en la capa oculta y como se ven reflejados en el desempeño de la red. Comente sobre: tiempo de entrenamiento, capacidad de la red, sobreajuste y cantidad de muestras necesarias. Indique el desempeño relativo al considerar sigmoidal y ReLU, considerando la velocidad de convergencia, el promedio y la desviación estándar de la suma de la diagonal de la matriz de confusión.

Se entrega un código base, el cual es funcional, pero no considera el uso del conjunto de test, ni el gráfico del accuracy, ni la medición del tiempo de ejecución.

Los informes y códigos deben ser subidos a u-cursos a más tardar a las 23:59 del 30 de abril. Incluir un corto archivo de texto explicando cómo se utiliza su programa. Las tareas atrasadas serán penalizadas con un punto base más un punto de descuento adicional por cada día de atraso.

Importante: La evaluación de la tarea considerará el correcto funcionamiento del programa, la inclusión de los resultados de los pasos pedidos en el informe, la calidad de los experimentos realizados y de su análisis, la inclusión de las partes importantes del código en el informe, así como la forma, prolijidad y calidad del mismo. Se adjunta la pauta de corrección junto a la tarea, la cual indica la estructura y contenido esperados para el informe.