El Grupo Integrado de Ingeniería ha estado trabajando en el MDB desde hace más de 10 años. Este se compone de 5 grandes bloques que contienen modelos matemáticos cada uno de ellos y se quieren aprender por medio de mecanismos de aprendizaje automático. El proyecto nace porque el método que está utilizando la el grupo para obtener este aprendizaje es costoso y los resultados que han tenido no han sido buenos. Es por esto que se propuso tratar de resolver el problema mediante Deep Learning.

Lo primero que se hizo fue tener un objetivo al cual tratar de llegar (tener un modelo cuyas predicciones sean casi perfectas). Para esto se decidió entrenar una red utilizando Mini-Batch, y los resultados obtenidos fueron excelentes. A partir de ese objetivo, se comenzaron a probar técnicas de aprendizaje online que simularan el mundo real en el cual se encontraría el robot. Los resultados fueron buenos y el problema fue resuelto, como se esperaba.

Conclusiones

* Sí se cumplió con los objetivos propuestos pues se pudo encontrar un método de entranmiento online que lograra aprender en tiempo real. Sus predicciones son muy buenas y además el tiempo de ejecución es bajo.

Si se compara el método resultante con el que se estaba utilizando, hay una mejora positiva en cuanto a rendimiento y eficiencia.

* Se entregó el código con los entrenamiento Batch, Mini-Batch, entrenamiento estocástico, estocástico con memoria y estocástico con memoria y Mini-Batch. Además de clases para guardar datos en un csv. Se codificó un programa que grafica un csv con la librería matplotlib.

Se entregaron bitácoras de varios entrenamientos, incluyendo el entrenamiento final de cada pnode, se realizó una validación cruzada con el pnode0 para comprobar la generalización en el aprendizaje de la red.

Las bitácoras incluyen:

* + 3 archivos csv con los siguientes datos:
    - Error de entrenamiento y test por iteración (Formato: iteración, entrenamiento,prueba)
    - Tiempo de cada iteración (Formato: iteración,segundos)
    - Predicciones finales (Formato: registro,predicción,etiqueta)
  + 3 gráficas mostrando los siguientes datos:
    - Curvas de entrenamiento y test
    - Tiempos por iteración
    - Gráfica de dispersión de las predicciones
  + Un archivo que contiene información de cada entrenamiento:
    - Número de entrenamiento
    - Tamaño de memoria utilizado
    - Pasos de entrenamiento
    - Valor del learning rate
    - Error cuadrático medio de entrenamiento
    - Raíz del error cuadrático medio de las pruebas
    - Raíz del error cuadrático medio del test final
    - Desviación estándar del test final
    - Optimizador utilizado
    - Número de neuronas por cada capa de la red neuronal
    - Número del pnode utilizado para entrenar
    - Nombre de las funciones de activación utilizadas por capa
    - Booleano que indica que se utilizó Dropout en las capas intermedias
    - Tipo de entrenamiento utilizado

El entrenamiento aún está en prueba, los resultados que se han obtenido han sido buenos pero se tiene que seguir comparando con otros datasets y distintos hiperparámetros para ver sus comportamiento ante escenarios distintos.

El método de eliminar el elemento de la memoria cuando está llena, puede ser distinto a una cola, se pueden probar otras técnicas como eliminar datos repetidos.

Se puede crear una función de coste para la red que combine el error cuadrático medio y la desviación estándar. (en caso de los entrenamientos con mini-batch o batch).

Se puede realizar la integración con el MDB, utilizando TensorFlow.

Experiencias

Mi trabajo realizado en el GII de la UDC me brindó mucho conocimiento en materia de Deep Learning. El campo aún se encuentra en investigación y cada vez hay más técnicas que ayudan al buen funcionamiento de las redes neuronales profundas.

El proyecto cumplió con sus objetivos y se realizó lo descrito en el anteproyecto.

Fue una experiencia demasiado enriquecedora viéndolo desde un ángulo técnico y social.