UNIVERSIDAD ADOLFO IBAÑEZ
FACULTAD DE INGENIERÍA Y CIENCIAS

Tarea 3

ING560: Algebra Lineal y Optimización para Data Science

Profesor: Miguel Romero

Fecha de Entrega: 21 de Diciembre 2020

Indicaciones:

1. No se aceptarán atrasos.
2. La tarea se puede hacer en grupos de a lo más 3 personas.
3. Debe entregar un **informe** con sus respuestas a los problemas. Junto con la tarea se adjunta un archivo `main.py` el cual deberá completar. Este archivo incluye también código para entrenar y probar su red neuronal. Para la preparación del informe se recomienda el uso de LaTeX.
4. Puede bajar las últimas versiones de `Python`, `Numpy` y `PyTorch` desde https://www.python.org/, https://numpy.org/ y https://pytorch.org/.

Problema 1 (25%)

Responda brevemente a las siguientes preguntas:

1. (2.0 pts) Explique la diferencia entre descenso por el gradiente y descenso estocástico por el gradiente, en el contexto de entrenamiento de redes neuronales.
2. (2.0 pts) ¿Cuál es la diferencia entre descenso por el gradiente con y sin momentum? ¿Qué es lo que buscamos al agregar momentum?
3. (2.0 pts) Explique en qué consisten los métodos RMSProp y Adam.

Problema 2 (75%)

Utilizaremos el dataset MNIST utilizado en clases. Recuerde que este dataset consiste de 70.000 imágenes en blanco y negro de 28×28 pixeles las cuales corresponden a dígitos del 0 al 9. Del total de imágenes, 60.000 son de entrenamiento y 10.000 para testeo.

1. (1.5 pts) Implemente una clase `FFNN_1HL` que herede de `torch.nn.Module` y que corresponda a una red neuronal feed-forward con *una* capa escondida y funciones de activación ReLU. Implemente el constructor de su clase de manera que la dimensión de la capa de entrada, de la capa oculta y de la capa de salida sean parámetros. Utilizando el código dado en `main.py`, entrene y pruebe su red sobre el dataset MNIST. Como optimizador utilice descenso estocástico del gradiente simple con una tasa de aprendizaje de $\eta = 0.01$. Pruebe con dimensiones para la capa oculta de $1, 10, 100, 200$ y 500. ¿Qué puede observar con respecto al tiempo de ejecución, a la función de pérdida, y a la precisión de la red?
2. (1.5 pts) Ahora fije la dimensión de la capa oculta a 500. Entrene y pruebe su red sobre el dataset MNIST, para valores de tasa de aprendizaje de 0.1, 0.01, 0.001 y 0.0001. ¿Qué puede observar con respecto al tiempo de ejecución, a la función de pérdida, y a la precisión de la red?

3. (1.5 pts) Fije la dimensión de la capa oculta a 500 y la tasa de aprendizaje a $\eta = 0.01$. Entrene y pruebe su red sobre el dataset MNIST, utilizando las funciones de activación ReLU, sigmoid y tangente hiperbólica. Para estas dos últimas puede utilizar las funciones `torch.nn.Sigmoid()` y `torch.nn.Tanh()`. ¿Qué puede observar con respecto al tiempo de ejecución, a la función de pérdida, y a la precisión de la red?

4. (1.5 pts) Con dimensión de capa oculta 500, tasa de aprendizaje de $\eta = 0.01$, y función de activación ReLU, entrene y pruebe su red sobre el dataset MNIST, utilizando tres optimizadores: descenso estocástico del gradiente simple, descenso estocástico del gradiente con momentum (con parámetro `momentum=0.9`) y Adam (con los parámetros por defecto). Nuevamente, ¿Qué puede observar con respecto al tiempo de ejecución, a la función de pérdida, y a la precisión de la red?