A1 Mean/stdev

reliminares
Dado un conjunto de m números

$$X = \{x_i\} \qquad i = 1..m$$

su media y desviación estándar vienen dadas por:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \quad ; \quad \sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2}$$

la expresión de la desviación estándar puede ser desarrollada de la siguiente manera:

$$\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2 = \frac{1}{m}(\sum x_i^2 - 2\mu\sum x_i + m\mu^2) = \frac{\sum x_i^2}{m} - 2\mu\frac{\sum x_i}{m} + \mu^2$$

con lo que finalmente

$$\sigma = \sqrt{\frac{\sum x_i^2}{m} - \left(\frac{\sum x_i}{m}\right)^2}$$

lo que nos permite calcular ambas la media y desviación estándar con una única pasada por los datos. Observa que cualquier sumatorio puede ser dividido en sumas parciales que se calculan en paralelo y luego se agregan.

A1.1 Calcula mean/stdev
1. Ejecuta el script create-data.py para crear 4 conjuntos de datos. Inspecciona el contenido los ficheros creados para familiarizarte con ellos.
2. Ejecuta el script a11-simple.py y trata de entender cómo usa el código en la libreria ct.py para obtener en paralelo sum(Xi) y sum(Xi^2) de cada fichero por separado.
3. Completa el script a11.py (que es igual que el a11-simple.py) para que al final calcule la media y desviación estandar agregando las sumas parciales obtenidas de cada fichero.

A1.2 Mide el speed up de ejecución paralela
1. Verifica en VirtualBox si en la configuración de tu máquina virtual puedes asignarle dos CPUs virtuales.
2. Ejecuta el script a11.py que completaste anteriormente creando un conjunto de ficheros con create-data.py según los siguientes casos:

1 fichero de 200000 datos
2 ficheros de 100000 datos cada uno
4 ficheros de 50000 datos cada uno
8 ficheros de 25000 datos cada uno

y reporta los tiempos de ejecución en cada caso en un archivo que se llame **`a12-reporte.txt`**

A1.3 Optimización estocástica
1. Abre y entiende el script a13-sgd-show.py y las funciones en sgd.py. Observa como se genera un dataset aleatorio cada vez que se invoca y resuelve un problema de regresión lineal por el método del

gradiente desciente batch, y por el método del gradiente descendiente estocástico.
2. Observa como en la optimización estocástica elegimos aleatoriamente elementos del dataset de entrenamiento y actualizamos los parámetros a optimizar con cada elemento.
3. Ejecuta a13-sgd-show.py y entiende las gráficas que genera.
4. Modifica a13-sgd-show.py para generar datasets de 2,3,4,5,6,7 y 8 dimensiones y sus gráficas asociadas. Colecciona las gráficas generadas en un documento llamado **`a13-reporte.pdf`** e incluye en el mismo la respuesta a las siguientes preguntas:

> **Pregunta 1**: ¿en qué casos la convergencia hacia un error mínimo es comparable en el caso batch y en el caso estocástico?
> **Pregunta 2**: ¿en qué casos la convergencia en el caso batch en mucho más rápida?
> **Pregunta 3**: ¿en qué casos la convergencia en el caso estocástico en mucho más rápida?

Crea un fichero zip con a11.py y tus reportes. Por ejemplo

```
zip TALLER-1-CCCCCC.zip a11.py a12-reporte.txt a13-reporte.pdf
```

donde CCCCCC es tu número de cédula; abre Dropbox desde un navegador dentro de la máquina virtual, accede a tu cuenta y sube el fichero zip en la carpeta compartida