Технології графічного процесінгу & розподілених обчислень

Лекція 7: Розподілене навчання

Кочура Юрій Петрович
iuriy.kochura@gmail.com
@y_kochura

Сьогодні

- Чому розподілене навчання?
- Паралелізм даних vs. паралелізм моделі

Чому розподілене навчання?

Використання спеціального апаратного забезпечення дозволяє значно скоротити час навчання

Чому розподілене навчання?

Використання спеціального апаратного забезпечення дозволяє значно скоротити час навчання

Коротший час навчання забезпечує швидшу ітерацію для досягнення ваших цілей моделювання

Розподілення не автоматичне

```
Sun Apr 23 11:32:07 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.161.03   Driver Version: 470.161.03   CUDA Versio
|-------------------------------+----------------------+----------
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util
|                               |                      |
|===============================+======================+==========
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |
| N/A   56C    P8    10W /  70W |  12105MiB / 15109MiB |      75%
|                               |                      |
+-------------------------------+----------------------+----------
|   1  Tesla T4            Off  | 00000000:00:05.0 Off |
| N/A   59C    P8    10W /  70W |      0MiB / 15109MiB |      0%
|                               |                      |
+-------------------------------+----------------------+----------
```

Способи розподіленого навчання

Паралелізм даних

- Синхронний паралелізм даних
- Асинхронний паралелізм даних

Паралелізм моделі

Способи розподіленого навчання

Паралелізм даних

- Синхронний паралелізм даних
- Асинхронний паралелізм даних

Паралелізм моделі

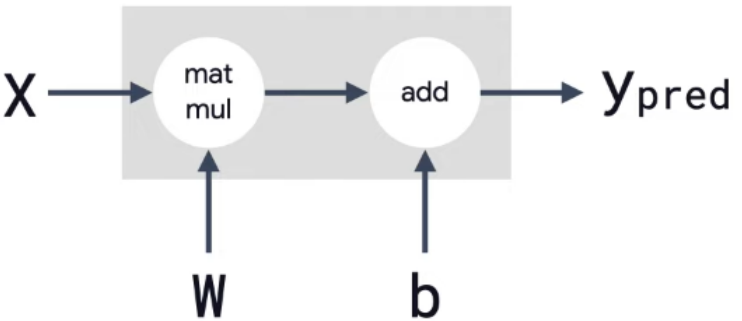
```
model.fit(x, y, batch_size=32)
```

```
model.fit(x, y, batch_size=32)
```

```
model.fit(x, y, batch_size=(32 * NUM_GPUS))
```

Лінійна модель

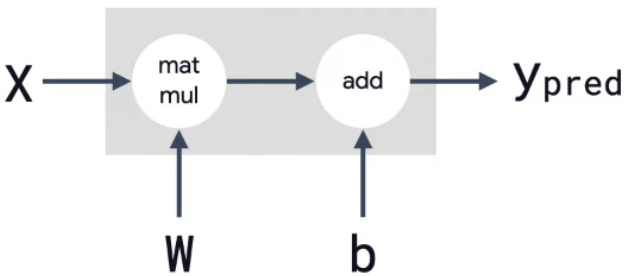
$$y_{\text{pred}} = WX + b$$

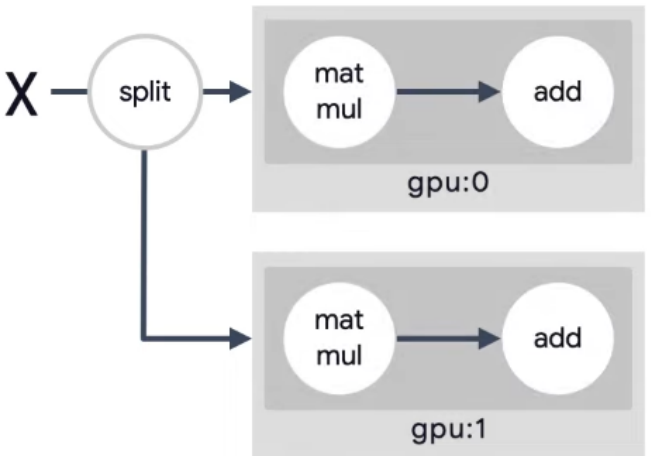



```
tf.keras.layers.Dense(units=1)
```

Паралелізм даних

$$y_{\text{pred}} = WX + b$$






```
tf.keras.layers.Dense(units=1)
```

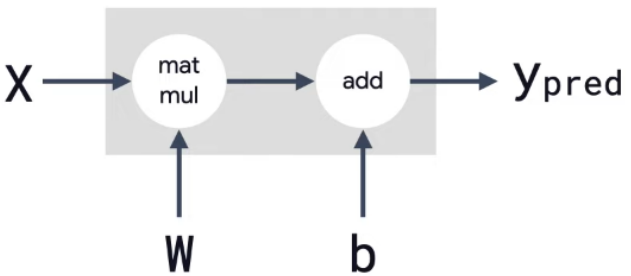
Способи розподіленого навчання

Паралелізм даних

Паралелізм моделі

Паралелізм моделі

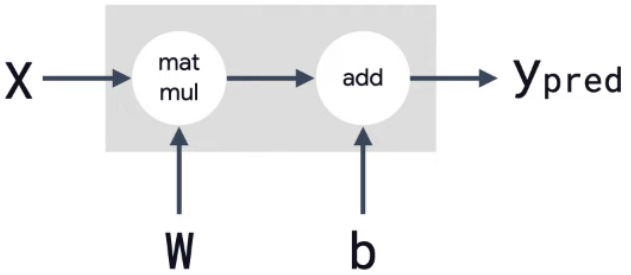
$$y_{\text{pred}} = WX + b$$



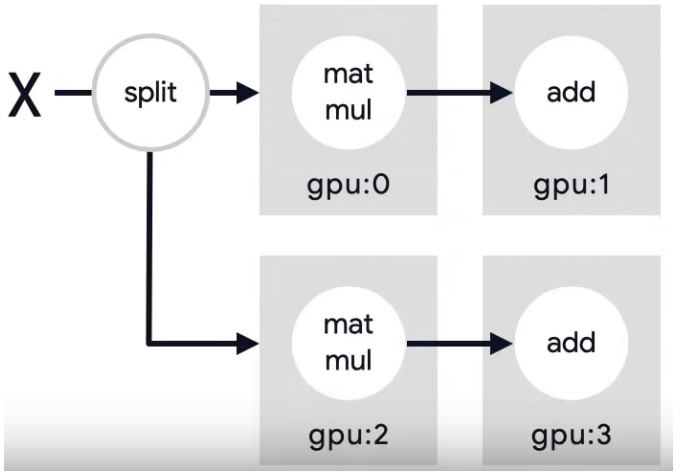

```
tf.keras.layers.Dense(units=1)
```

Комбінація

$$y_{\text{pred}} = WX + b$$




```
tf.keras.layers.Dense(units=1)
```



Стратегії навчання

- Агрегація градієнтів: об’єднання градієнтів із кількох графічних процесорів може бути дорогим у плані обчислень. Щоб зменшити цю вартість, можна рідше об’єднувати градієнти або використовувати методи стиснення, щоб зменшити розмір градієнтів.
- Перекриття зв’язку та обчислення: поки один GPU обчислює градієнти, інші GPU можуть отримувати дані та готуватися до наступної партії. Це може допомогти скоротити загальний час навчання.
- Балансування робочого навантаження: залежно від розміру моделі та обсягу даних деякі графічні процесори можуть виконувати роботу швидше, ніж інші. Щоб максимізувати продуктивність, вам слід збалансувати робоче навантаження на всі графічні процесори.

Література

1. A friendly introduction to distributed training
2. Deep learning on the parameter server
3. Scaling Distributed Machine Learning with the Parameter Server
4. Overview of how TensorFlow does distributed training
5. Deep learning on the parameter server

Кінець 🏁