

Wprowadzenie do TensorFlow

Wojciech Janowski



Czym jest TensorFlow?

- Niskopoziomowy framework stworzony przez Google, służący do zaawansowanych obliczeń numerycznych wykorzystujący specjalne grafy operacji (dataflow graphs)
- Używany do szeroko pojętego uczenia maszynowego, w tym sieci neuronowych
- Elastyczne środowisko wspierające architektury rozproszone, bazujące zarówno na CPU jak i GPU
- Obsługuje systemy Linux, macOS, Windows, istnieje też wersja mobilna iOS oraz Android



Historia

- W 2011 roku ekipa Google Brain stworzyła DistBelief, system do uczenia maszynowego, oparty na głębokich sieciach neuronowych
- W 2015 projekt przemianowano na TensorFlow
- Środowisko jest dystrybuowane na licencji open source
- Wersja 1.0.0 została opublikowana 11.02.2017
- Najnowsza wersja 1.4.0 miała premierę 2.11.2017



Schemat budowy

Frontend API
Python, C++, GO, Java

Backend, TensorFlow Core
C++, biblioteka Eigen



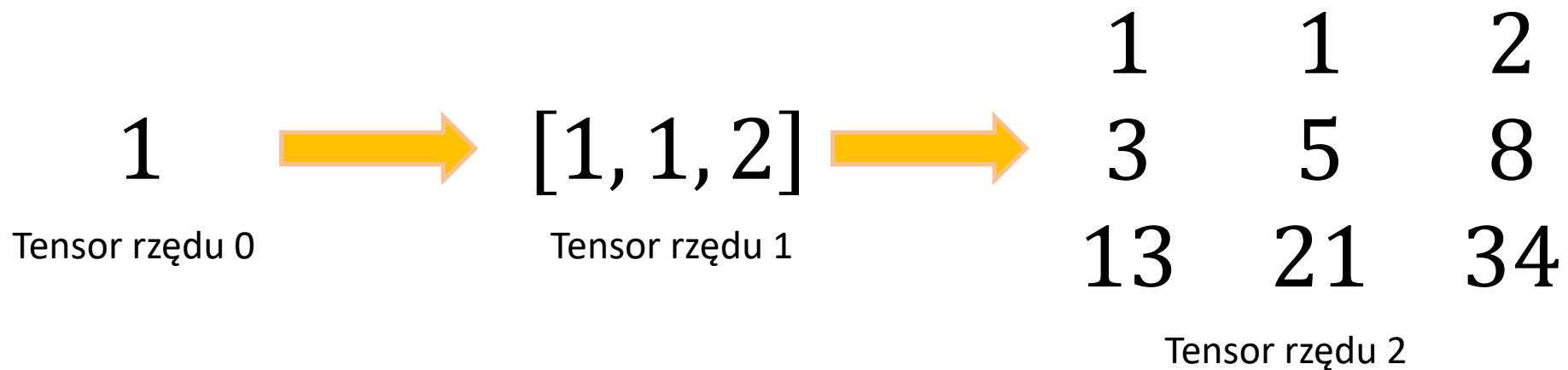
Warstwa sprzętowa
CPU & GPU & TPU
CUDA, OpenCL, XLA

Dane
Dataset API
Hadoop



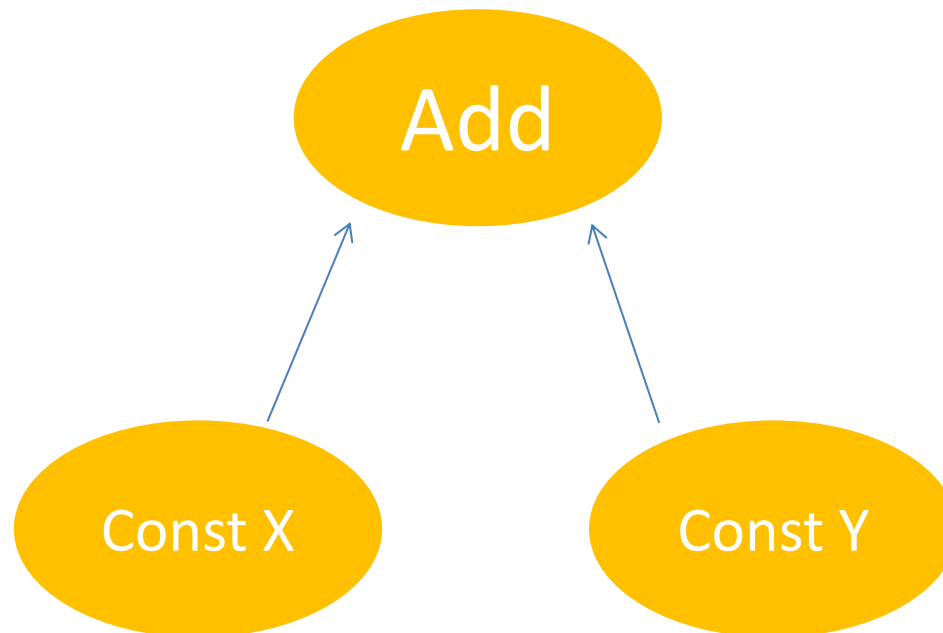
Odrobina matematyki

Tensor- uogólnienie pojęcie wektora



Graf operacji

- Operacje (stałe również) są węzłami
- Wielowymiarowe tensory zawierające dane stanowią krawędzie



Zalety

- Model dataflow umożliwia bardzo łatwe i efektywne równoleglenie operacji
- Możliwość podziału zadań na różne jednostki wykonawcze (CPU, GPU, TPU)
- Optymalizacja podczas kompilacji (eksperymentalne XLA)
- Niezależność od języka, dzięki temu model stworzony w Pythonie można bez problemu przenieść do C++



Wykonywanie grafu

- Pierwszą fazą jest stworzenie grafu poprzez połączenie różnych operacji i stałych
- Wywołanie metody `Session.run()` spowoduje wykonanie grafu

```
node1 = tf.constant(3.0, dtype=tf.float32)
node2 = tf.constant(4.0) # also tf.float32 implicitly

sess = tf.Session()
print(sess.run([node1, node2]))
```

```
[3.0, 4.0]
```



TensorBoard

- Narzędzie służące do wizualizacji oraz badania przebiegu działania grafu
- Ułatwia testowanie, umożliwia analizę procesu uczenia, zbieranie statystyk
- Generuje dokument HTML, który otwieramy w przeglądarce



Dataset API

- Klasyczne sieci neuronowe wymagają ogromnej ilości danych
- Zarządzanie nimi bywa kłopotliwe, zwłaszcza w systemach rozproszonych
- Dataset API umożliwia tworzenie i sterowanie potokami danych
- Łatwa, hurtowa edycji danych wejściowych/wyjściowych
- Generowanie permutacji z istniejących danych
- Gotowe do użycia zbiory danych (np *MNIST*)



Keras

- TensorFlow jest niskopoziomowym frameworkiem
- Wymaga od programisty uwagi i wiedzy o tym co robi 😊
- Konieczność ustawiania i konfiguracji wielu parametrów
- Z tego powodu powstał Keras, specjalna nakładka na TF znacznie upraszczająca pracę,
- Otrzymał oficjalne wsparcie Google, obecnie został na stałe zaimplementowany w TF Core jako pełnoprawne, możliwe do wyboru API



Tensor Processing Unit

- Funkcjonowanie sieci neuronowych opiera się na licznych operacjach macierzowych małej precyzji
- Idąc tym tokiem myślenia Google stworzyło specjalny procesor typu ASIC (Application-Specific Integrated Circuit), który wykonuje sprzętowo operacje na macierzach
- Zamiast 32bit lub 16bit danych, działa na 8 bitowych liczb



Dane techniczne

- Wykonany w 28nm, taktowany 700MHz, zużycie prądu 40W
- **Matrix Multiplier Unit (MXU)** - 65,536 8-bitowych jednostek multiply-and-add
- **Unified Buffer (UB)** - 24MB pamięci SRAM używana jako rejestry
- **Activation Unit (AU)** – sprzętowe funkcje aktywacji
- Podczas jednego cyklu zwykle CPU wykona kilka operacji
- CPU z jednostką wektorową przeprowadzi dziesiątki operacji
- GPU obliczy dziesiątki tysięcy operacji
- TPU wykona może wykonać do 128 tysięcy operacji 😊

