TensorFlow

Open-Source Bibliothek für maschinelles Lernen

Matthias Täschner
Seminar Deep Learning WS1718
Abteilung Datenbanken
Universität Leipzig

Motivation

- Renaissance bei ML und KNN
- Forschung bei DNN
- fortgeschrittene Bild- und Spracherkennung
- AlphaGo (Google DeepMind)
- skalierbare ML-Modelle für Einzel- und verteilte Systeme
- spezialisierte Hardware (TPU)

TensorFlow

- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- Anwendung
- Bewertung und Fazit

Entwicklung

- Google / Google Brain Projekt
- 2011 - DistBelief
  - skalierbar, verteilte Systeme
  - Google Search / Ads / Maps / Youtube
- 2015 - TensorFlow
  - Nachteile von DistBelief ausgleichen
  - Datenfluss-Modell
  - Cluster-Systeme bis Mobilgeräte
- Open Source, www.tensorflow.org

TensorFlow

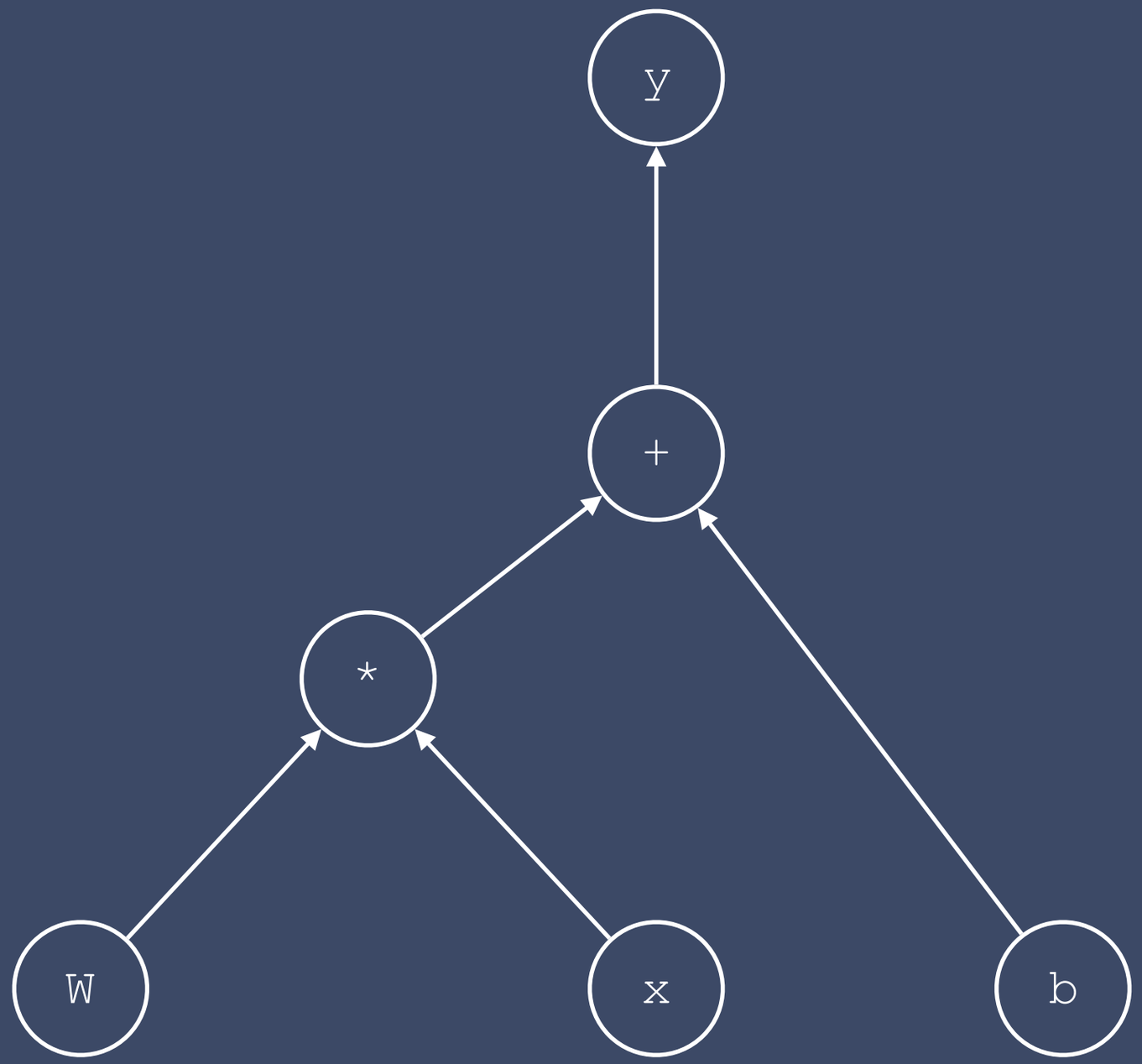
- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- Anwendung
- Bewertung und Fazit

Datenfluss-Modell

- Datenfluss-Graph repräsentiert
  - Berechnungen und
  - Zustände in ML-Algorithmus
- gerichteter Graph mit
  - Knoten -> Operationen
  - Kanten -> Datenfluss zwischen Knoten

Datenfluss-Modell

$$y = W * x + b$$



Datenfluss-Modell

Operationen als Knoten

- Kombination oder Transformation von Daten
- 0…n Eingaben, 0…n Ausgaben
- spezifische Implementierung für Ausführung auf spezifischer Recheneinheit -> Kernel

Datenfluss-Modell

Operationen als Knoten

Kategorie	Beispiel
math. Operation	Add, Sub, Mul, …
Array Operation	Concat, Shuffle, …
Matrix Operation	MatMul, MatInverse, …
zustandsor. Op.	Variable, Assign, …
neuronale Netze	SoftMax, Sigmoid, ReLU, …
CheckPoint Op.	Save, Restore
Flusssteuerung	Switch, Merge, …

vgl. Abadi, 2015

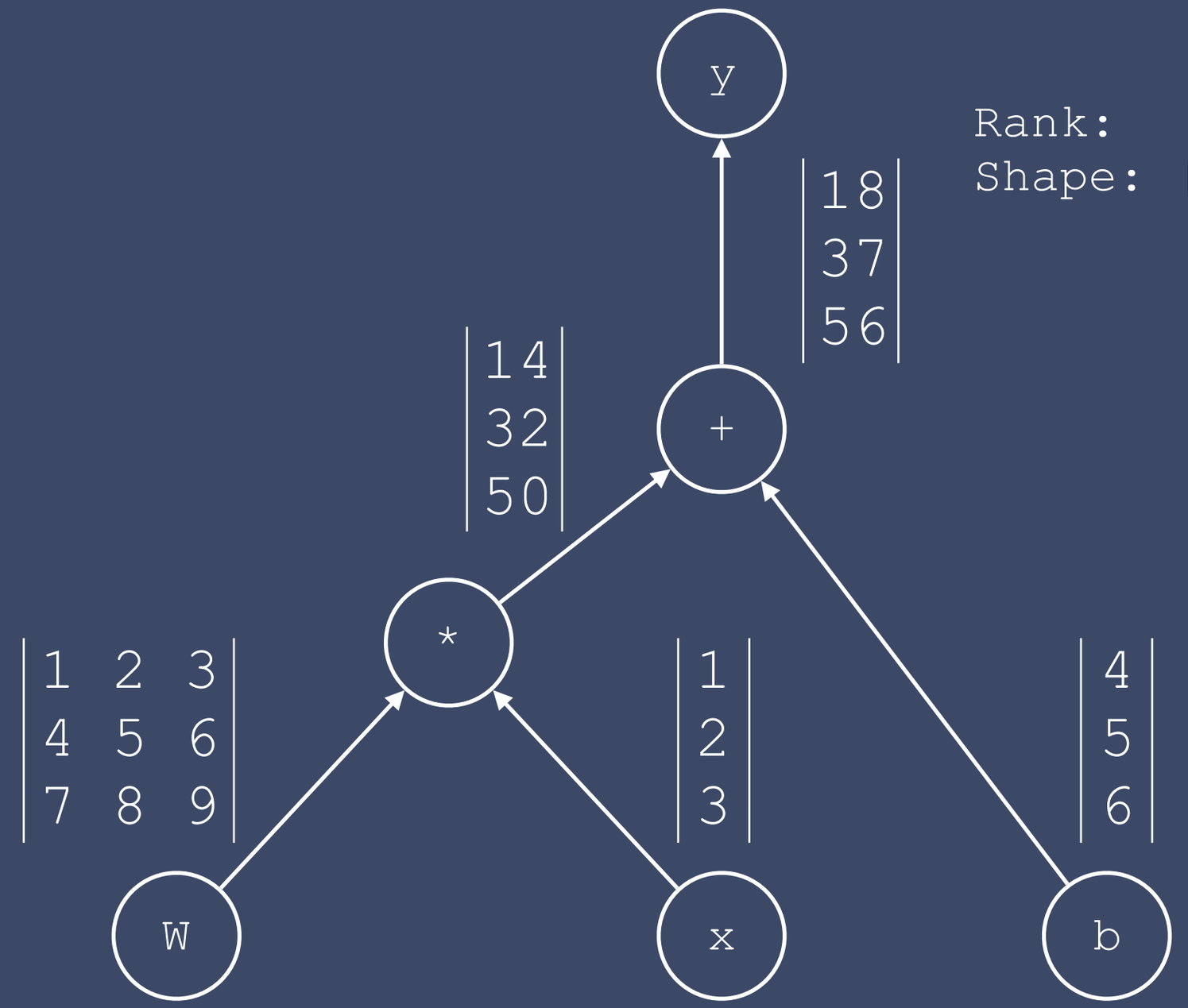
Datenfluss-Modell

Tensoren als Kanten

- Format für Datenfluss zwischen Operationen
- multidimensionales Array, homogene Werte, statischer Typ
- „Rank“, „Shape“
- Identifikatoren für Zugriff auf referenzierte Werte im Speicher

Datenfluss-Modell

Tensoren als Kanten



TensorFlow

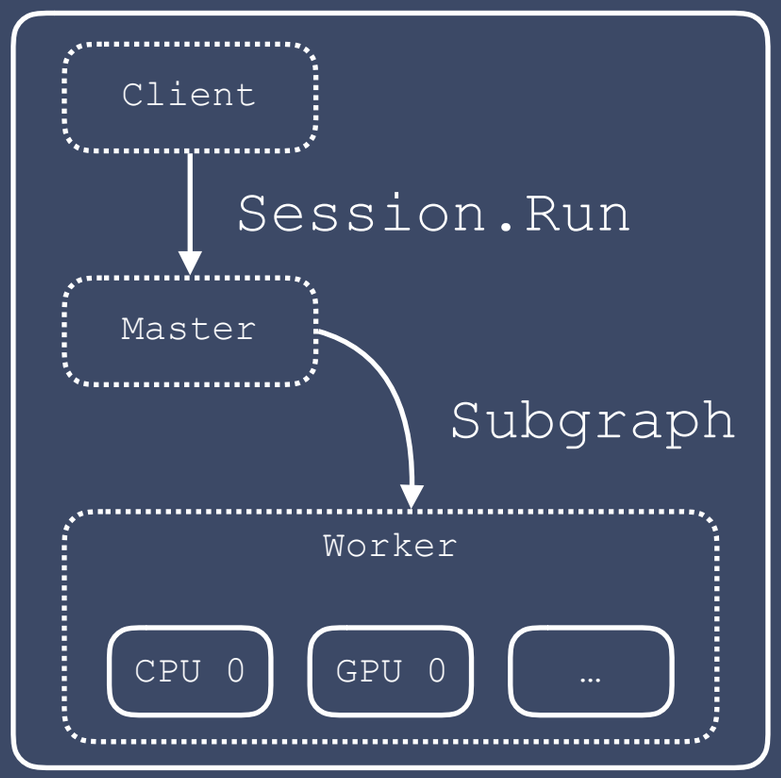
- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- Anwendung
- Bewertung und Fazit

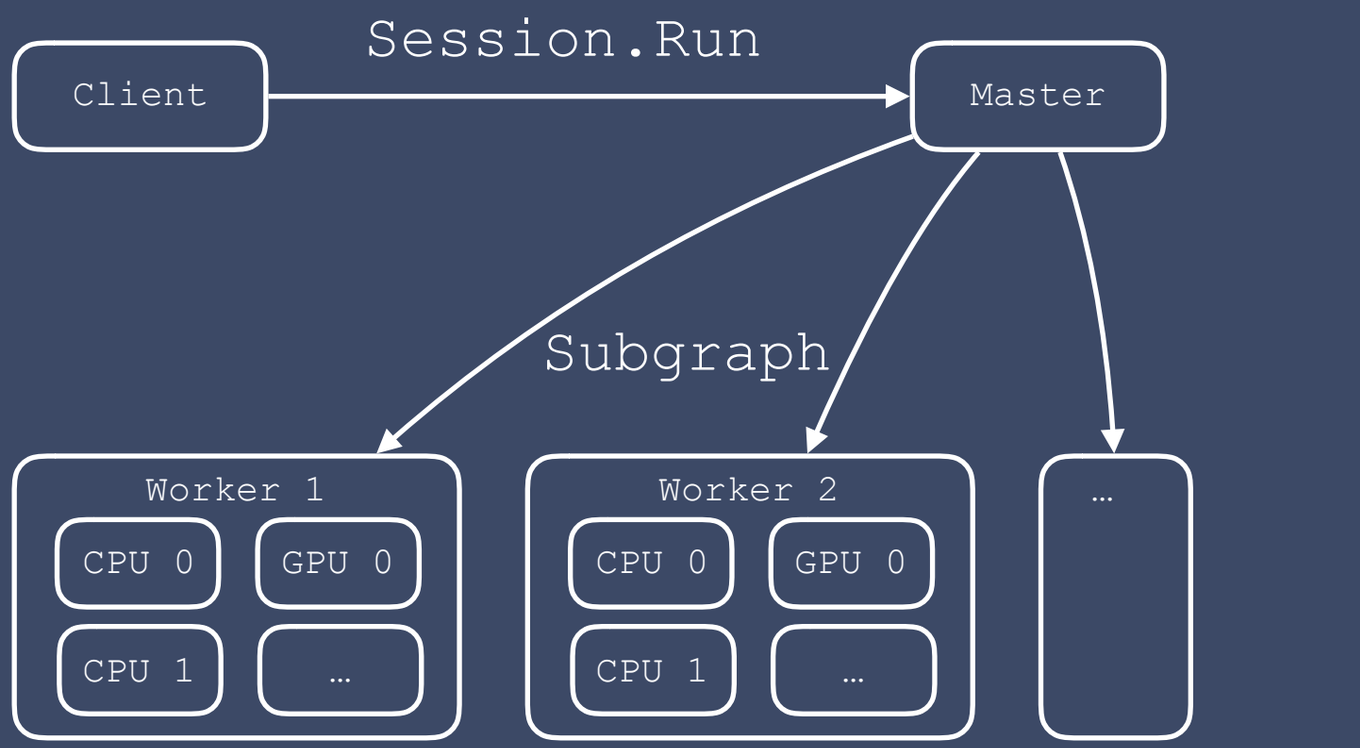
Implementierung

vgl. Abadi, 2016

Implementierung

- **Client** kommuniziert via **Session** mit **Master**
- Master orchestriert **Worker**
- **Worker** verwalten Recheneinheiten





vgl. Abadi, 2015

Implementierung

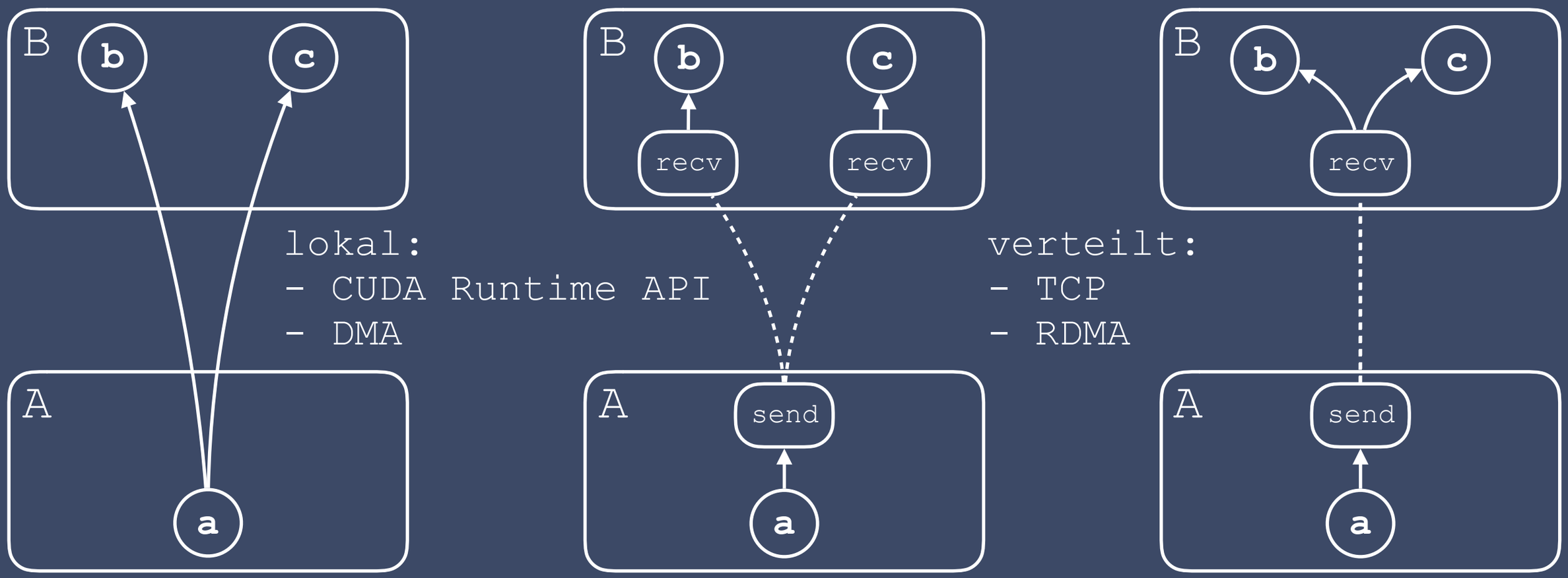
Verteilte Systeme / Recheneinheiten

- Knotenplatzierung mit Kostenmodell via Simulation
  - Größe von Eingabe- / Ergebnis-Tensor
  - Rechenzeit der Operation
  - Schätzung oder Messungen
- Platzierung auf Recheneinheit mit entsprechendem Kernel

Implementierung

Verteilte Systeme / Recheneinheiten

- Datenaustausch zwischen platzierten Subgraphen



vgl. Abadi, 2015 und Goldsborough, 2016

Implementierung

Erweiterungen

- Gradientenberechnung via Back Propagation
- Fehlertoleranz bei Abbruch
- Ausführung von Teilgraphen
- Vorgaben bei Knotenplatzierung
- Flusssteuerung über Operatoren
- Optimierungen für Performance

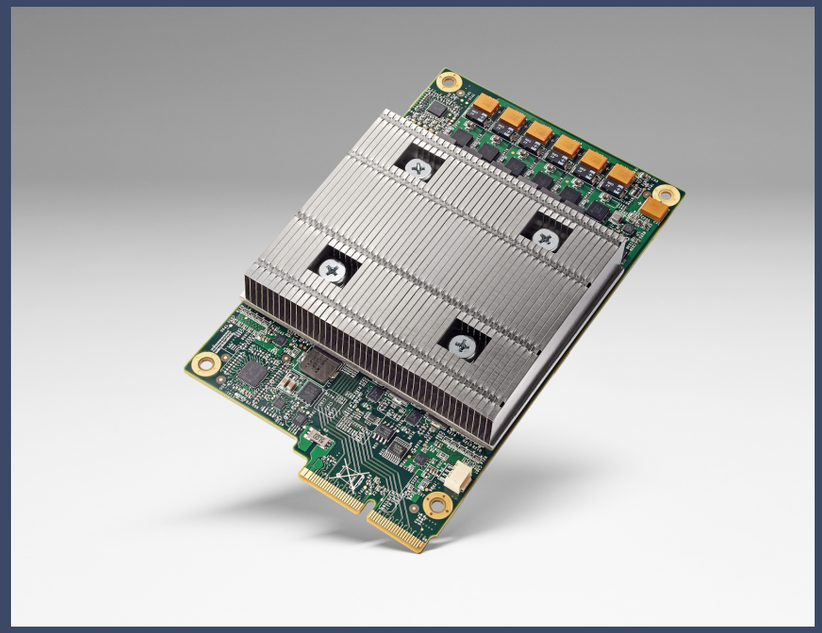
TensorFlow

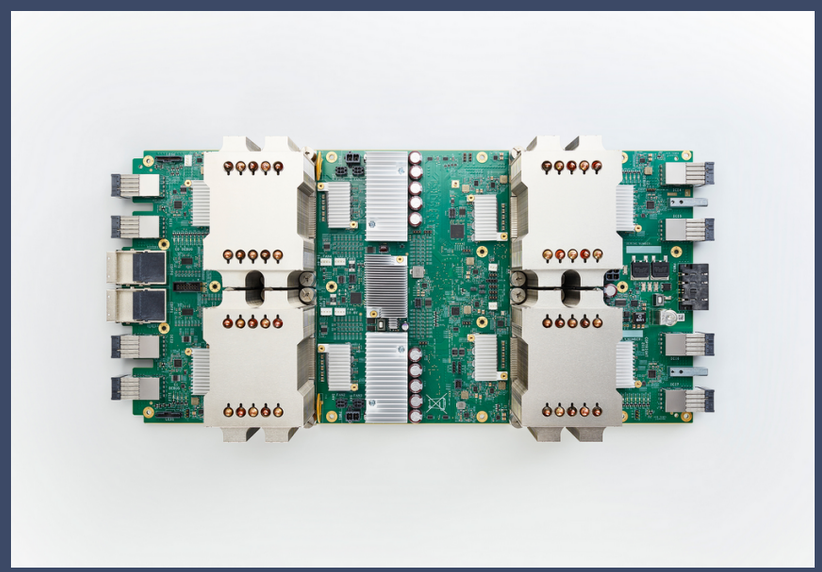
- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- Anwendung
- Bewertung und Fazit

Tensor Processing Unit

*y = act (W * x + b)*

- Operationen in KNN
  - Matrix-Multiplikation
  - Addition
  - Aktivierungsfunktion





Quelle: https://cloudplatform.googleblog.com, 2017

Tensor Processing Unit

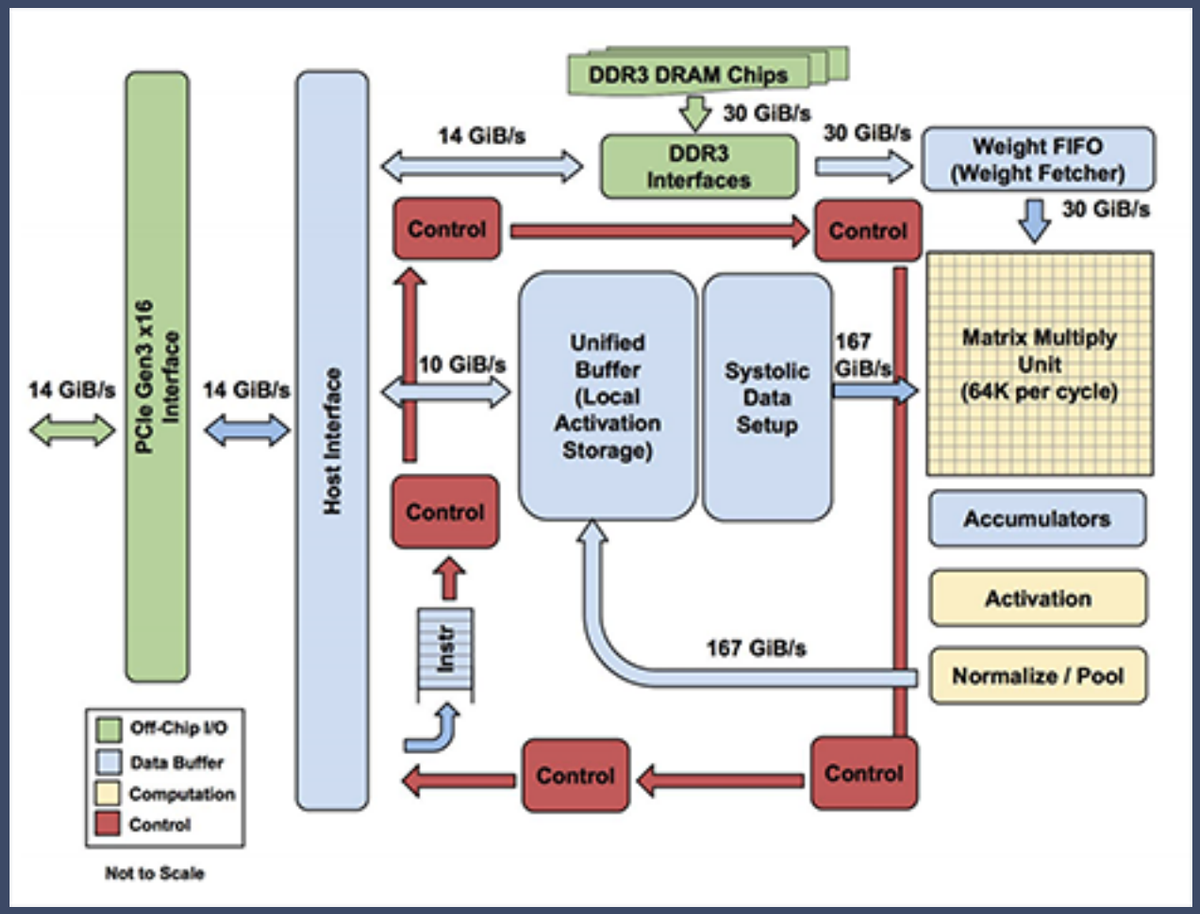
Matrix Multiplier Unit (MXU)

- systolisches Array
- 256x256 ALU - 8bit Int Mul

Geschwindigkeit:

- 15x Nvidia K80
- 30x Intel Haswell (18 Core)

vgl. Jouppi, 2017



Quelle: https://cloud.google.com/blog/big-data/, 2017

TensorFlow

- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- **Anwendung**
- Bewertung und Fazit

Anwendung

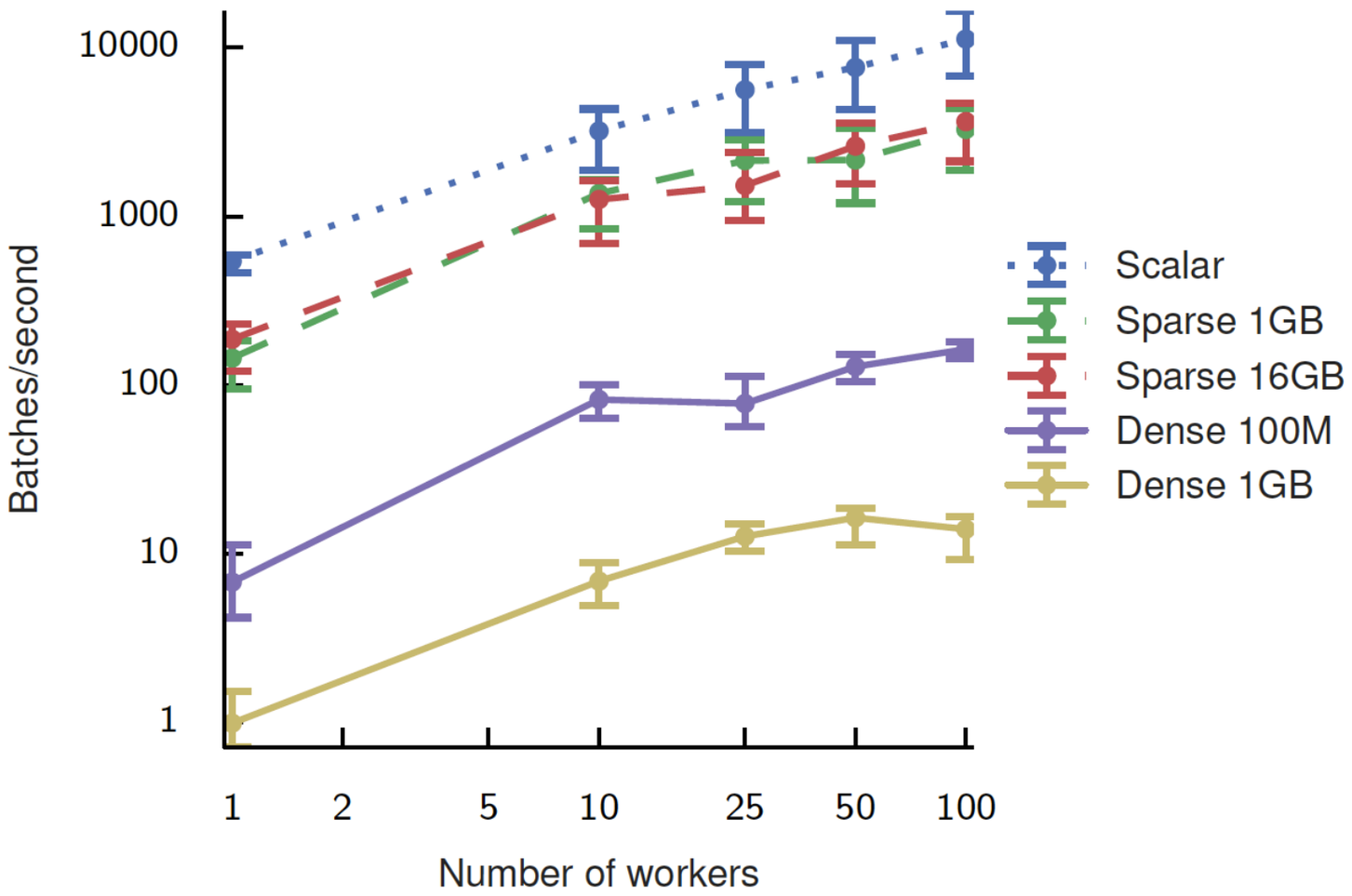
- Erkennung handgeschriebener Zahlen mit neuronalen Netzen
  -> log. Regression
  -> Convolutional Neural Network
- Visualisierung mit TensorBoard

TensorFlow

- Entwicklung
- Datenfluss-Modell
- Implementierung
- Tensor Processing Unit
- Anwendung
- Bewertung und Fazit

Bewertung und Fazit

Library	Training step time (ms)			
	AlexNet	Overfeat	OxfordNet	GoogleNet
Caffe [38]	324	823	1068	1935
Neon [58]	87	**211**	**320**	**270**
Torch [17]	**81**	268	529	470
TensorFlow	**81**	279	540	445



Geschwindigkeit:
CNN auf Intel Core i7-5930K (6 Core) mit NVIDIA Titan X

Skalierung:
- „null training step“
- 16 PS Tasks / Worker

Time / TrainStep
Skalar:
1.8ms - 8.8ms
Dense:
147ms - 613ms / 1s - 7s
Sparse:
5ms - 20ms

vgl. Abadi, 2016

Literatur

Martín Abadi et al., „TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems“, 2015

Martín Abadi et al., „TensorFlow: A System for Large-Scale Machine Learning“, 2016

Jeffrey Dean et al., „Large Scale Distributed Deep Networks“, 2012

Norman P. Jouppi et al., „In-Datacenter Performance Analysis of a Tensor Processing Unit“, 2017

Peter Goldsborough, „A Tour of TensorFlow“, 2016

Gerhard Völkl, „Python-Tutorial, Teil 1: Maschinelles Lernen mit TensorFlow“, 2017

Gerhard Völkl, „Python-Tutorial, Teil 2: Neuronale Netze und Deep Learning“, 2017

https://www.tensorflow.org

Fragen?