

CNTK

刘士君

shliup@microsoft.com

CNTK “Cognitive Toolkit”

- CNTK 微软**开源、跨平台**，深度神经网络的学习和评估工具。
- CNTK 通过**组合**简单的模块成复杂的计算架构，来表达任意的神经网络，支持相关的网络类型和应用。
- CNTK 产品级准确度、高效，支持多GPU/多机分布式。

- 支持各种开源模型

- 由Microsoft Speech researchers (Dong Yu et al.) 2012创建, “Computational Network Toolkit”
- 2015年初开源 (CodePlex)
- 2016一月Github
- Oct 2016 (beta) Python支持, 新品牌“Cognitive Toolkit”
- 由微软产品组使用
- 外部的贡献者 MIT and Stanford

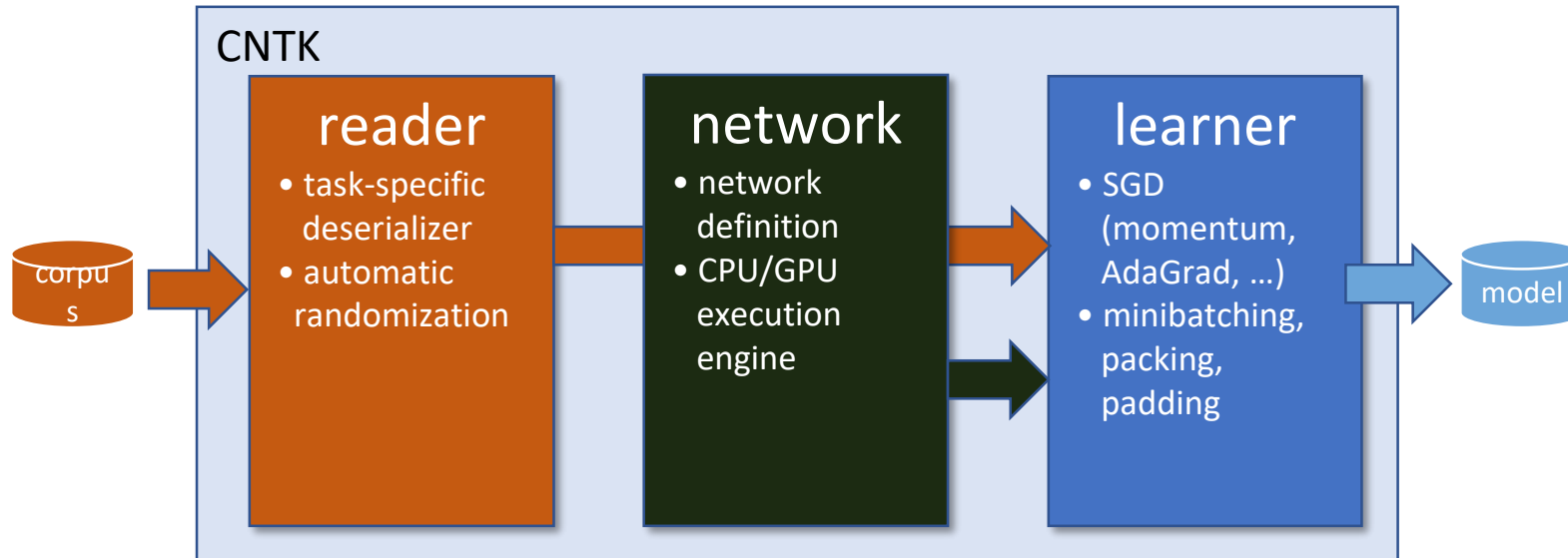
- Linux, Windows, docker, cudnn5, 下一步: CUDA 8

- Python and C++ API (beta; C#/.Net on roadmap)

Microsoft Cognitive Toolkit Market

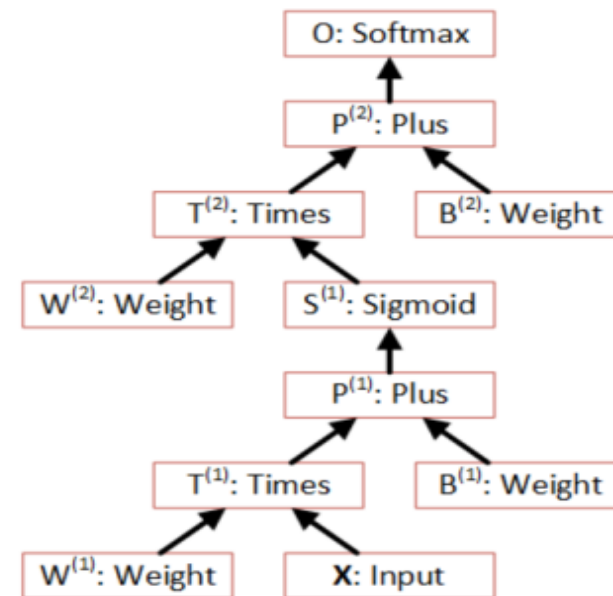
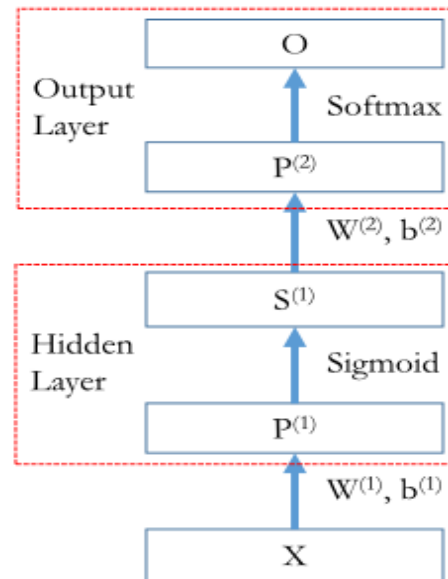
	Developer/Sponsor	Release date	GitHub Stars	Search Results
Tensorflow	Google	2015. 11	57,336	3,920,000
Caffe	BVLC	2013. 09	17,872	416,000
Cognitive Toolkit	Microsoft	2016. 01	10,717	975,000
MXNET	DMLC	2015. 04	9,677	395,000
Torch7	Facebook	2012. 01	6,856	495,000
Theano	Community	2011. 08	6,287	1,020,000

CNTK “Cognitive Toolkit”



Computational Graph

- A generalization of machine learning models that can be described as a series of computational steps.



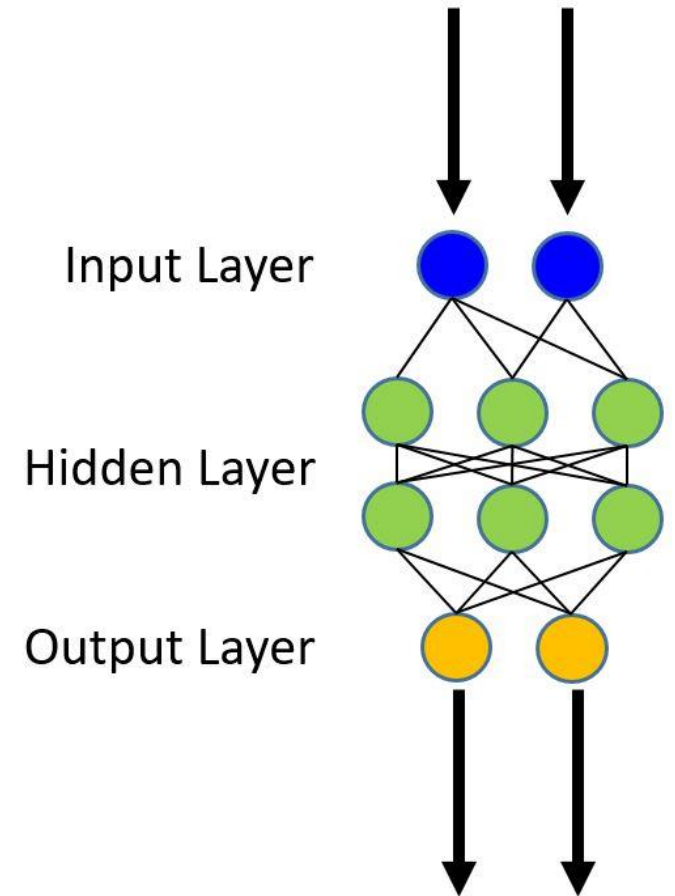
example: 2-hidden layer feed-forward NN

$$h_1 = \sigma(W_1 x + b_1)$$

$$h_2 = \sigma(W_2 h_1 + b_2)$$

$$P = \text{softmax}(W_{\text{out}} h_2 + b_{\text{out}})$$

with input $x \in \mathbb{R}^M$



example: 2-hidden layer feed-forward NN

$$h_1 = \sigma(W_1 x + b_1)$$

$$h_2 = \sigma(W_2 h_1 + b_2)$$

$$P = \text{softmax}(W_{\text{out}} h_2 + b_{\text{out}})$$

with input $x \in \mathbb{R}^M$ and one-hot label $y \in \mathbb{R}^J$
and cross-entropy training criterion

$$ce = y^T \log P$$

$$\sum_{\text{corpus}} ce = \max$$

example: 2-hidden layer feed-forward NN

$$h_1 = \sigma(W_1 x + b_1)$$

$$h_2 = \sigma(W_2 h_1 + b_2)$$

$$P = \text{softmax}(W_{\text{out}} h_2 + b_{\text{out}})$$



$$h1 = \text{sigmoid} (x @ w1 + b1)$$

$$h2 = \text{sigmoid} (h1 @ w2 + b2)$$

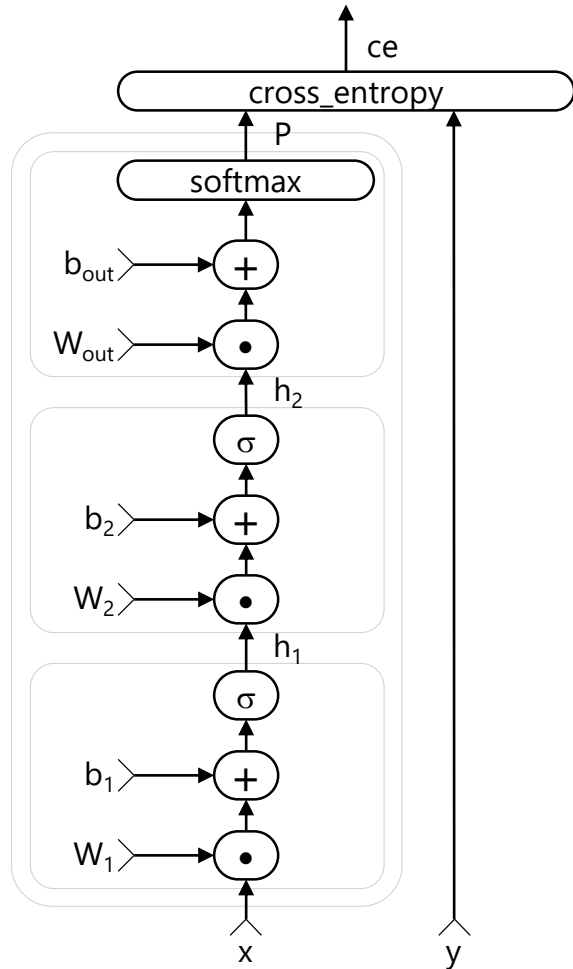
$$P = \text{softmax} (h2 @ wout + bout)$$

with input $x \in \mathbb{R}^M$ and one-hot label $y \in \mathbb{R}^J$
and cross-entropy training criterion

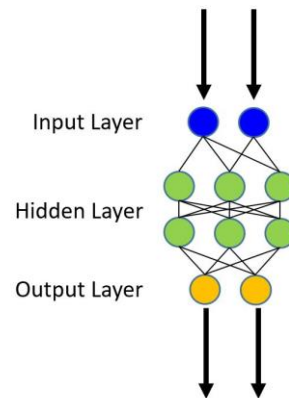
$$ce = y^T \log P$$

$$\sum_{\text{corpus}} ce = \max$$

$$ce = \text{cross_entropy} (P, y)$$



$$\begin{aligned}
 h1 &= \text{sigmoid} (x @ w1 + b1) \\
 h2 &= \text{sigmoid} (h1 @ w2 + b2) \\
 P &= \text{softmax} (h2 @ wout + bout) \\
 ce &= \text{cross_entropy} (P, y)
 \end{aligned}$$



```

learner = momentum_sgd(parameters,
lr_schedule,
mm_schedule,l2_regularization_weight)

```

```

trainer = Trainer(tl_model, (ce, pe), learner)

```

实验环境：Azure CPU F8 / Ubuntu 16.04

- `sudo apt-get install openmpi-bin`
- `pip install https://cntk.ai/PythonWheel/GPU/cntk-2.0-cp35-cp35m-linux_x86_64.whl`
- `python -c "import cntk; print(cntk.__version__)"`
- `python -m cntk.sample_installer`

Azure GPU VM

- N-series VM
 - NC VM: Telsa K80
 - NV VM: Telsa M60
- Support System:
 - CentOS
 - Ubuntu
 - Windows Server

INSTANCE	CORES	RAM	DISK SIZES	GPU
NC6	6	56.00 GiB	340 GB	1X K80
NC12	12	112.00 GiB	680 GB	2X K80
NC24	24	224.00 GiB	1,440 GB	4X K80
NC24r	24	224.00 GiB	1,440 GB	4X K80

Region: East US / South Central US / West US 2 / North Europe

INSTANCE	CORES	RAM	DISK SIZES	GPU
NV6	6	56.00 GiB	340 GB	1X M60
NV12	12	112.00 GiB	680 GB	2X M60
NV24	24	224.00 GiB	1,440 GB	4X M60

Region: East US / South Central US / North Central US / West US 2 / North Europe / West Europe / Southeast Asia / Japan East

预装和配置了多种常用的机器学习开发环境和工具: 如Ubuntu版本

- CNTK, TensorFlow, MXNet, Caffe, Caffe2, DIGITS, H2O, Keras, Theano, Torch
- NVIDIA driver, CUDA, and cuDNN
- 其他 :
 - Microsoft R Server 9.0 with Microsoft R Open 3.3.2
 - Anaconda Python 2.7 and 3.5
 - JupyterHub with sample notebooks
 - [Apache Drill](#) for querying non-relational data using SQL
 - Spark local 2.0.2 with PySpark and SparkR Jupyter kernels
 - Single node local Hadoop (HDFS, Yarn)
 - Azure command-line interface
 - Visual Studio Code, IntelliJ IDEA, PyCharm, Atom
 - JuliaPro, a curated distribution of Julia Language and tools
 - Vowpal Wabbit for online learning
 - xgboost for gradient boosting



00840



01170



00356

Visual Geometry Group
102类