# Machine Learning and Artificial Intelligence

Form the past to the future

# Parameters in ML

**The keys to Artificial Intelligence**

## Dataset

Informations  form real life.

## Algorithms

The way of understanding the dataset.

## Neural Networks

The combination of algorithms.

## Output

What do you want to do with the dataset.

# The traditional Methods.

There are some traditional ML libraries like the Encog and Accord.

Provided by Microsoft Research

# Microsoft Computational Network Toolkit

CNTK describes neural networks as a series of computational steps via a directed graph.

# CNTK

**What is it?**

01

OPEN-SOURCE

CNTK is Microsoft’s open-source, cross-platform toolkit for learning and evaluating deep neural networks.

02

SUPPORT

CNTK expresses(nearly) arbitrary neural networks by composing simple building blocks into complex computational networks, supporting relevant network types and applications.

03

PRODUCTION-READY

CNTK is production-ready: state-of-the art accuracy, efficient and scales to multi-GPU/ multi-server.

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

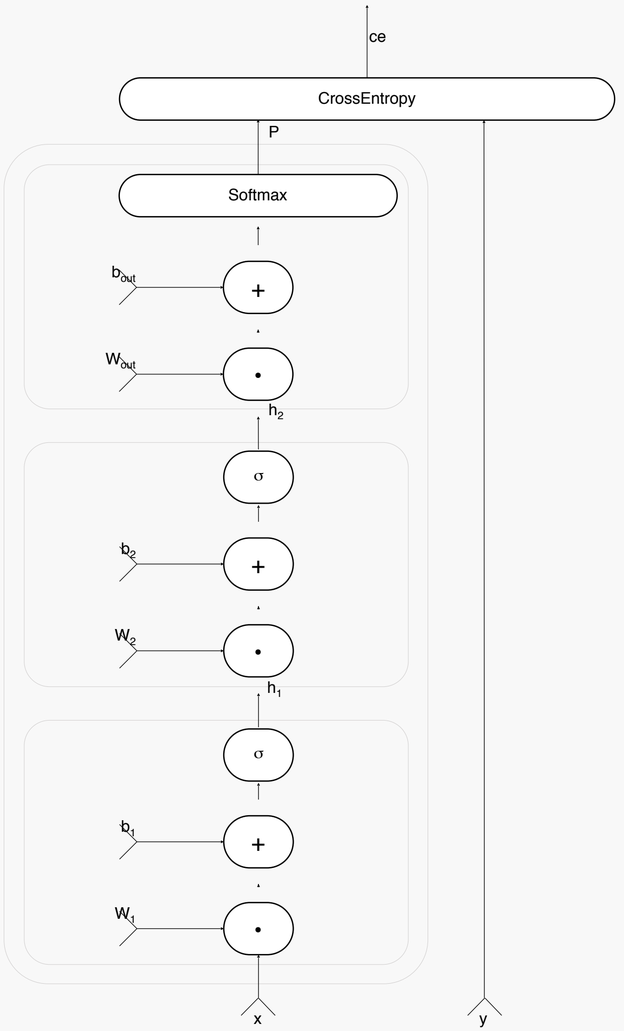
$$ce = y^{\mathrm{T}} \log P$$
$$\Sigma_{\text{corpus}} ce = \max$$

Canonical Deep network

# Example: 2-hidden layer feed-forward NN

h1 = Sigmoid (W1 * x + b1)
h2 = Sigmoid (W2 * h1 + b2)
P = Softmax (Wout * h2 + bout)
ce = CrossEntropy (y, P)

# Canonical Deep network




h1 = Sigmoid (W1 * x + b1)

h2 = Sigmoid (W2 * h1 + b2)

P = Softmax (Wout * h2 + bout)

ce = CrossEntropy (y, P)

All the nodes are functions
Edges are values.
It provides you with automatic adjustment.

# CNTK Architecture

## Reader

- task-specific deserializer
- automatic randomization

## Network

- network definition
- CPU/GPU execution engine

## Learner

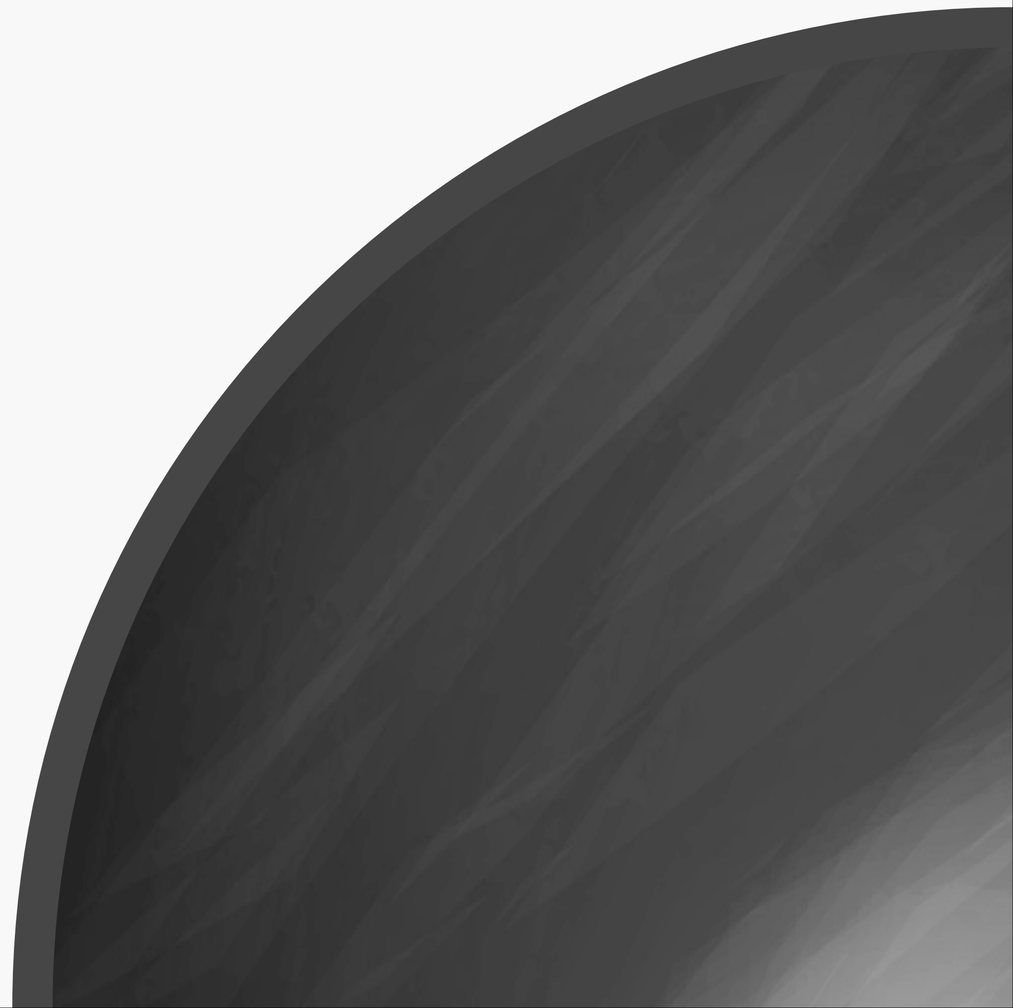
- Stochastic Gradient Decent) (momentum, AdaGrad, …)
- minibatching, packing, padding

# How to read the dataset

- standard formats: images, speech (HTK, Kaldi)
- convert to CNTK Text Format

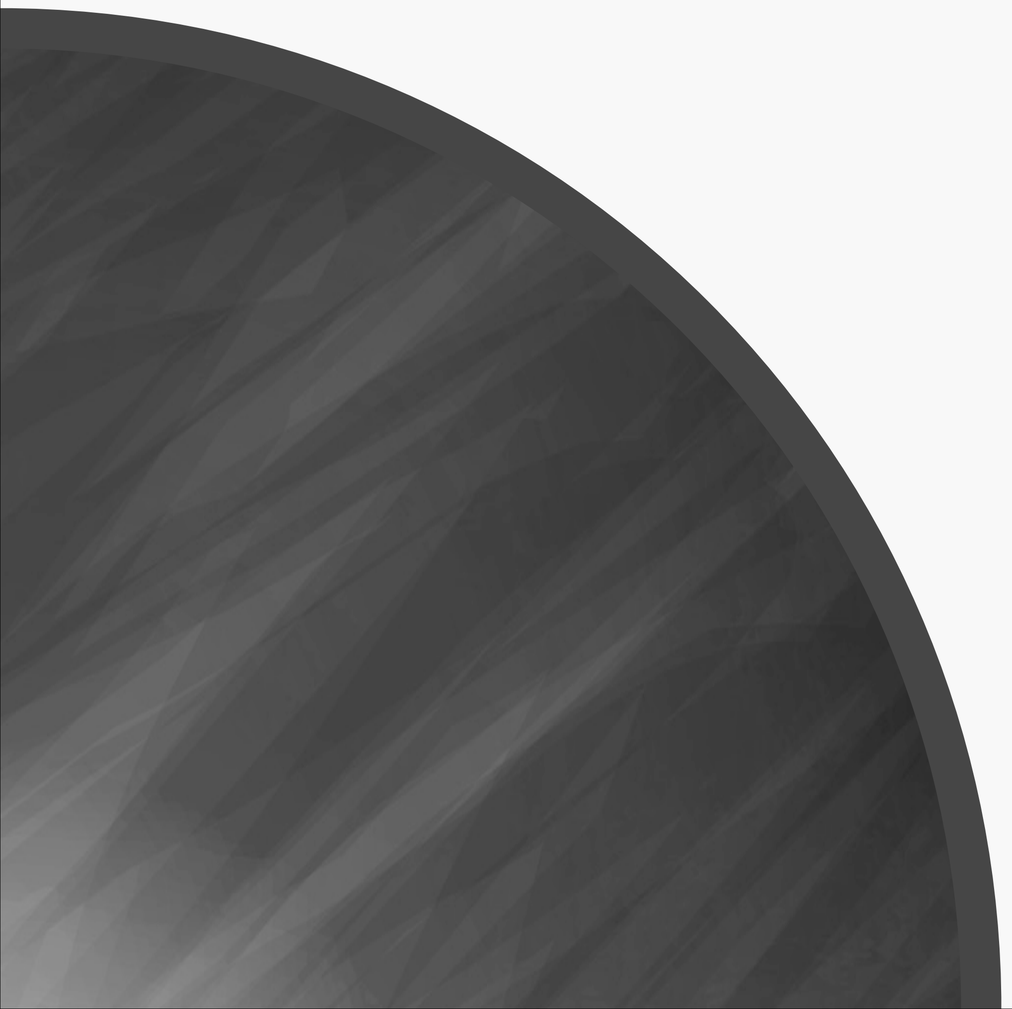
```
sed -e 's/^/<s> /' -e 's/$/ <\/s>/' < en.txt > en.txt1
sed -e 's/^/<s> /' -e 's/$/ <\/s>/' < fr.txt > fr.txt1
paste en.txt1 fr.txt1 | Scripts/txt2ctf.py --map en.dict fr.dict > ef.ctf
```

- big data: implement custom deserializer in C++ or Python



# How to build the network

- network specification consists of:
  - the network function's formula
    - including learnable parameters
    - (but no gradients, which are automatically determined by the system)
  - inputs
  - the output(s) and training/evaluation criteria
- network descriptions are called "brain scripts"
  - custom network description language "BrainScript"



# CNTK BrainScript

- direct, down-to-earth, easily understandable syntax
- high-level composability; custom functions/function objects (e.g. LSTM and GRU are expressed in BrainScript, not C++)
- powerful yet easy-to-use library for standard layer types (written in BrainScript)

# How to: learner

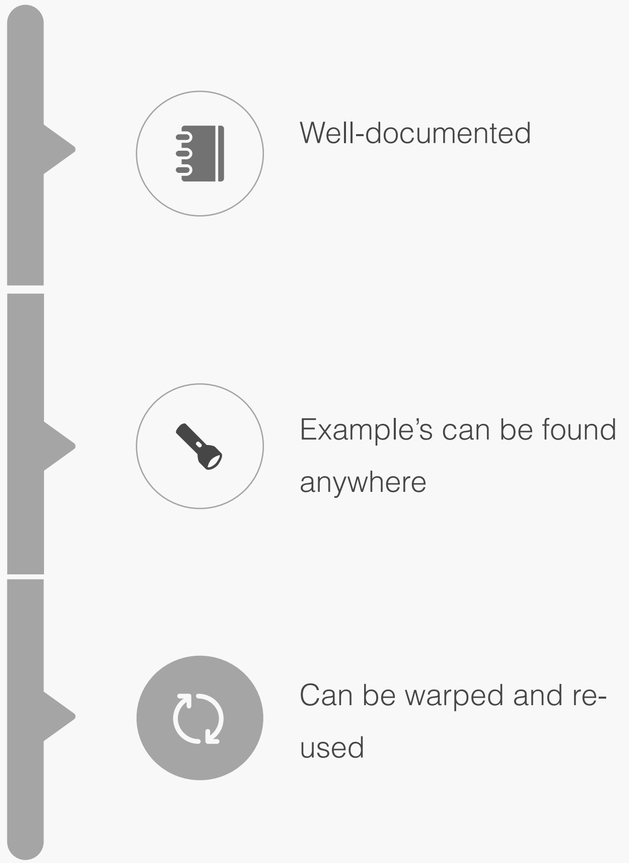
```
SGD = {
    maxEpochs = 50
    minibatchSize = $mbSizes$
    learningRatesPerSample = 0.007*2:0.0035
    momentumAsTimeConstant = 1100
    AutoAdjust = { … }
    ParallelTrain = { … }
}
```

- all learning parameters at a glance
- various SGD(stochastic gradient decent) variants (momentum, Adam, …)
- MB-size agnostic learning rate and momentum
- auto-adjustment of learning rate and minibatch size
- multi-GPU/multi-server parallelization

# Traditional

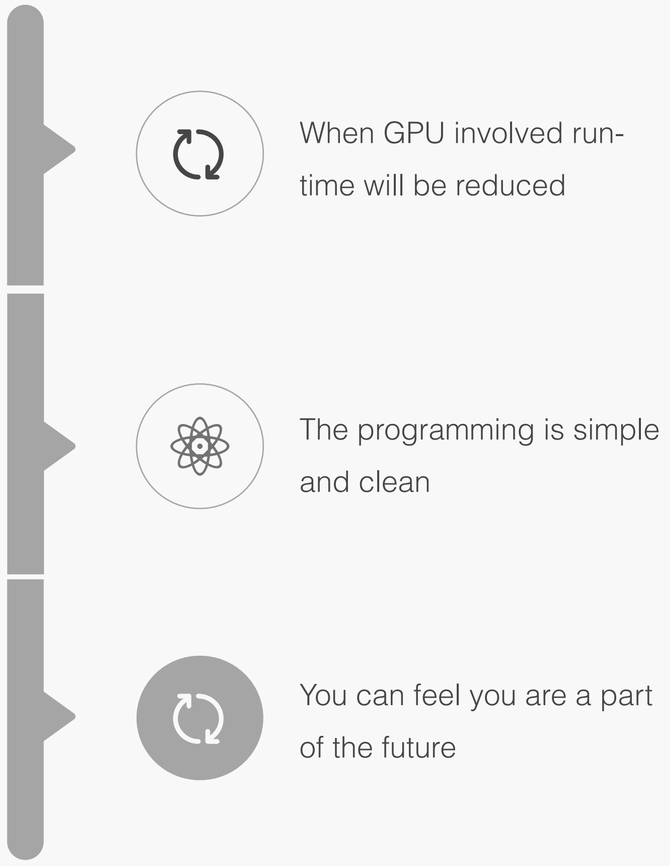
There are lots of traditional ML libraries can be found online, nearly supported by all kinds of programming language.

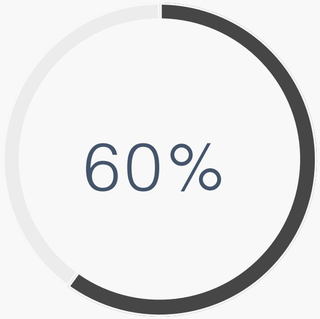
- Encog Java, C#, F#
- Accord C#



# CNTK

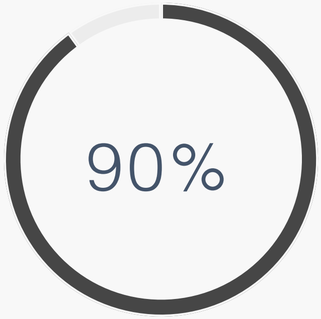
CNTK is relatively new, but it has huge potential, when the dataset is big and multi-dimensions.






## Single CPU

When running a big dataset.




## GPU involved

1000000 pictures dataset, single CPU takes 20 m, when using GPU takes 2 m.

# One more thing

# We can’t define the machine learning.

Give the possibility to the great minds.

# Bach style Prelude 29 Emmy Cope

Btech 451

Thank you.