# Nordic probabilistic AI school

## Introduction to probabilistic programming languages (PPLs)

Helge Langseth, Andrés Masegosa, and Thomas Dyhre Nielsen

June 14, 2022

# Plan for this week

**Day 1:** Probabilistic programming

- Introduction to probabilistic programming
- Probabilistic programming in Pyro

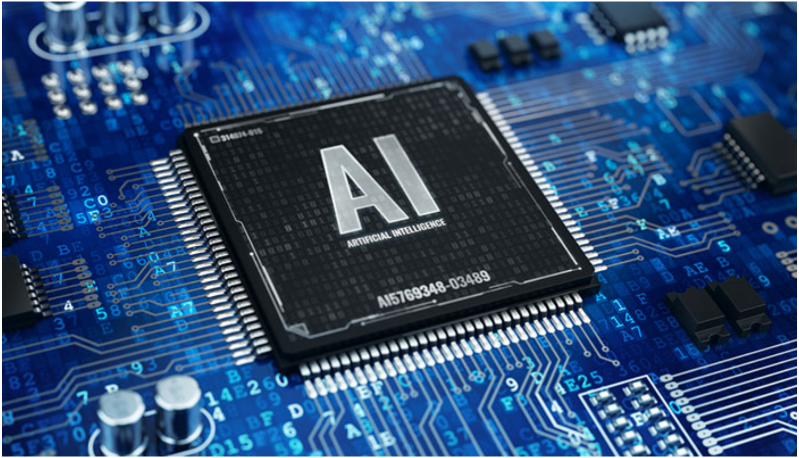
**Day 2:** Variational inference

- Recap of variational inference (variational inference as optimization)
- Derivation and implementation of selected examples
  - Bayesian linear regression
  - Factor analysis
  - . . .

**Day 3:** Variational inference – cont'd

- Black box variational inference
- Variational inference in Pyro
- Variational auto-encoders

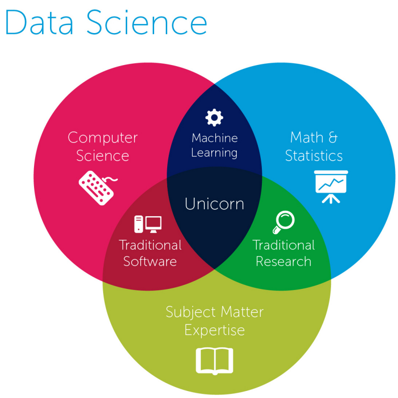
# Machine Learning Systems



The development of **machine learning systems** requires enormous efforts.

-

# Machine Learning Systems

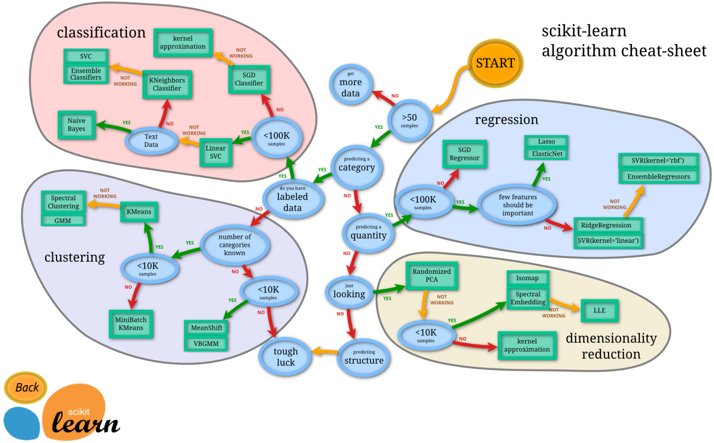


Copyright © 2014 by Steven Geringer Raleigh, NC.
Permission is granted to use, distribute, or modify this image,
provided that this copyright notice remains intact.

The development of **machine learning systems** requires enormous efforts.

- **It requires of highly qualified experts.**

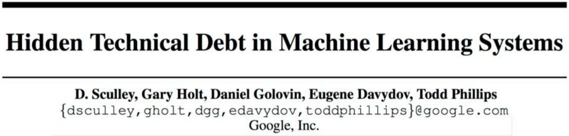
# Machine Learning Systems



## The development of machine learning systems requires enormous effort.

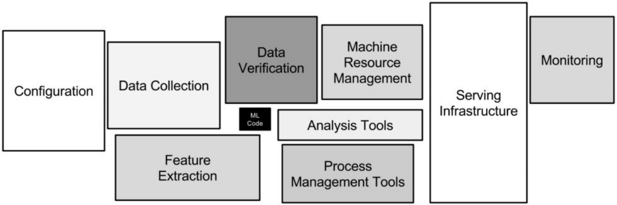
- It is necessary to have highly qualified experts.
- **It is difficult to find the ML model most suitable for an application.**

# Machine Learning Systems



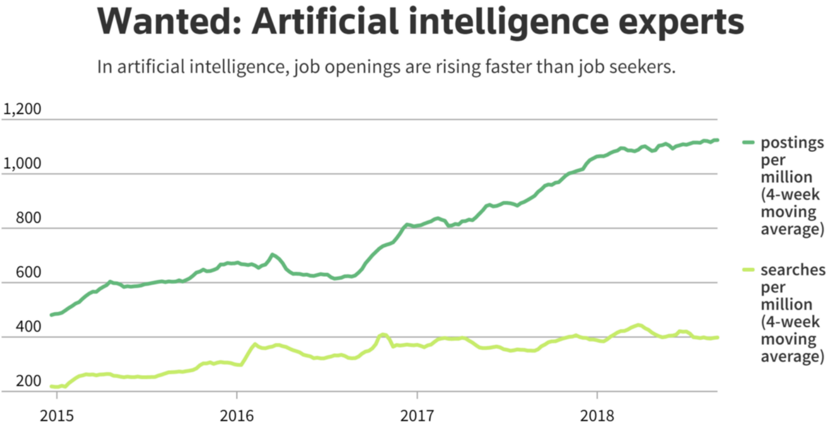
## Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.



### The development of machine learning applications requires enormous effort.

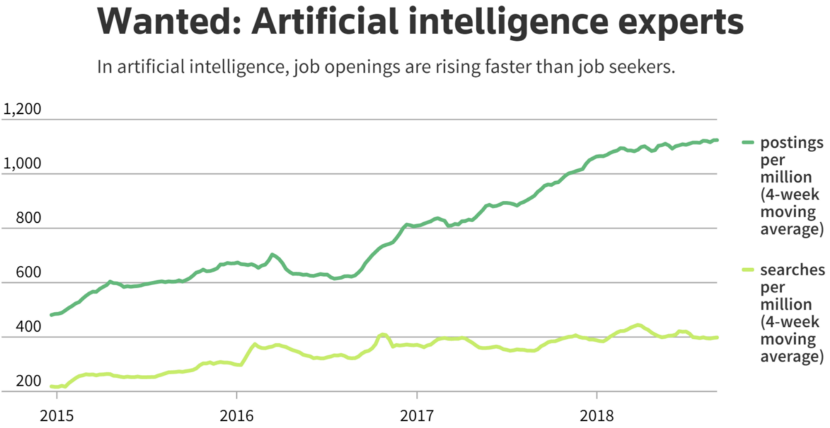
- It is necessary to have highly qualified experts.
- It is difficult to find the ML model most suitable for an application.
- **Programming a ML model is a complex task where many problems are intermingled**.



## Consequences:

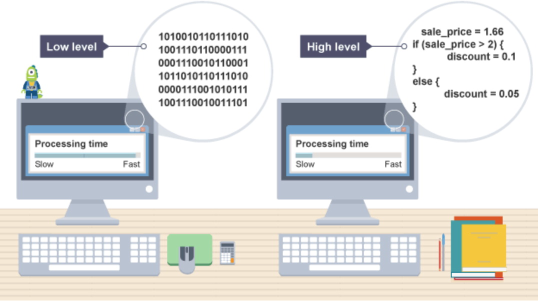
- Shortage of AI experts (and high salaries).

# Developing Machine Learning Systems



## Consequences:

- Shortage of AI experts (and high salaries).
- Only big corporations have the resources for developing ML systems.

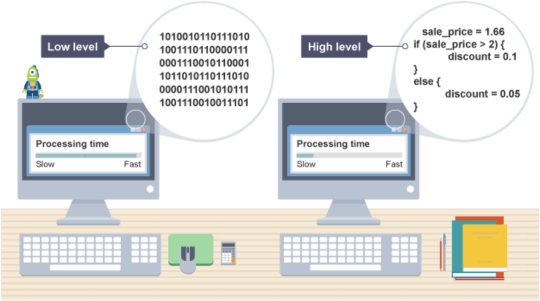


```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
 sale_price = 1.66
if (sale_price > 2) {
         discount = 0.1
}
else {
         discount = 0.05
}
```

## Similar situation than 50 years ago:

# Programming Languages



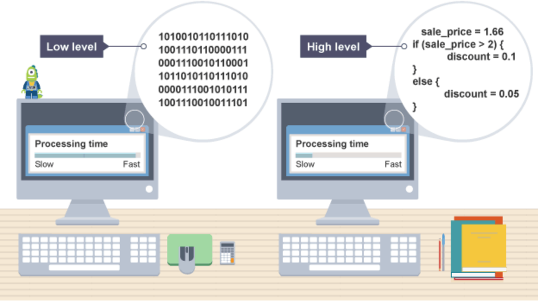
```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
 sale_price = 1.66
if (sale_price > 2) {
         discount = 0.1
}
else {
          discount = 0.05
}
```

## Similar situation than 50 years ago:

- People used to program in low-level programming languages.

# Programming Languages



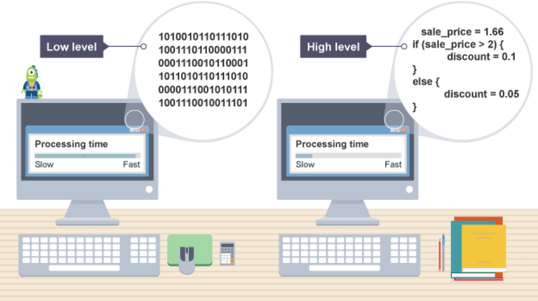
```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
 sale_price = 1.66
if (sale_price > 2) {
         discount = 0.1
}
else {
        discount = 0.05
}
```

## Similar situation than 50 years ago:

- People used to program in low-level programming languages.
- Programming was complex and demand high-expertise.

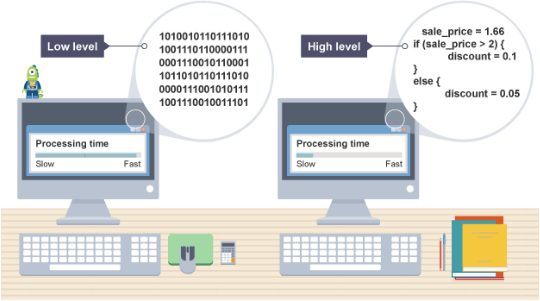
# Programming Languages



## Similar situation than 50 years ago:

- People used to program in low-level programming languages.
- Programming was complex and demand high-expertise.
- Focus on application and low-level hardware details.

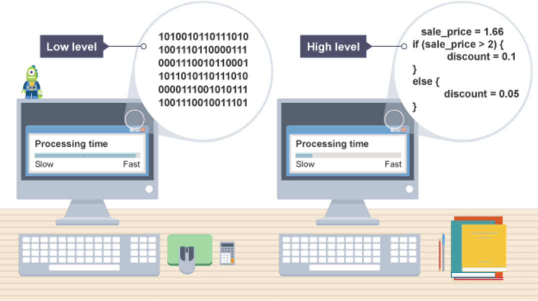
# Programming Languages



```
sale_price = 1.66
if (sale_price > 2) {
        discount = 0.1
}
else {
        discount = 0.05
}
```

## High-level programming languages brought many advantages:

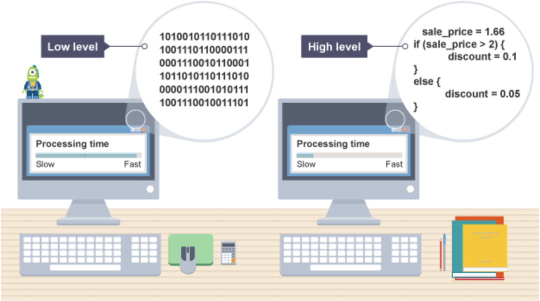
# Programming Languages



## High-level programming languages brought many advantages:

- Programmers focused on the applications.

# Programming Languages



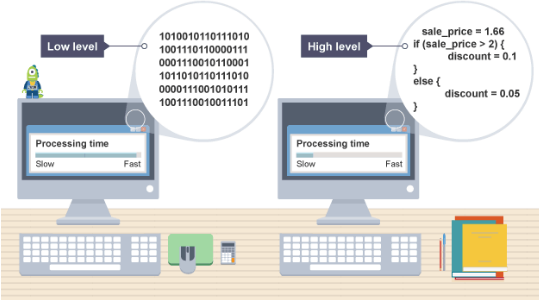
```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
  sale_price = 1.66
if (sale_price > 2) {
          discount = 0.1
}
else {
         discount = 0.05
}
```

## High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.

# Programming Languages



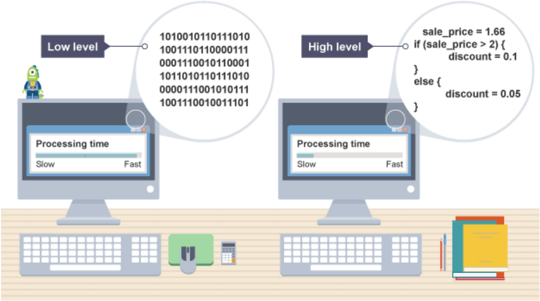
```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
sale_price = 1.66
if (sale_price > 2) {
        discount = 0.1
}
else {
        discount = 0.05
}
```

## High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.
- High gains in productivity.

# Programming Languages



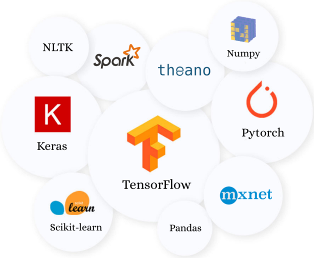
```
1010010110111010
1001110110000111
0001110010110001
1011010110111010
0000111001010111
1001110010011101
```

```
 sale_price = 1.66
if (sale_price > 2) {
         discount = 0.1
}
else {
         discount = 0.05
}
```

## High-level programming languages brought many advantages:

- Programmers focused on the applications.
- Hardware Experts focused on compilers.
- High gains in productivity.
- “Democratization” of the software development.

# Big Data and Machine Learning Ecosystem



Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
`https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c`

## Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries

# Big Data and Machine Learning Ecosystem

NLTK
Spark
theano
Numpy
Keras
Pytorch
TensorFlow
mxnet
learn
Scikit-learn
Pandas

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
`https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c`

## Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.

# Big Data and Machine Learning Ecosystem

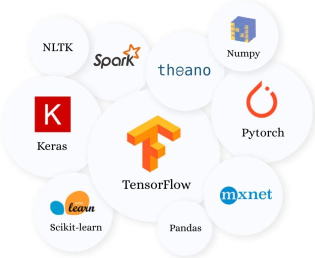
NLTK
Spark
theano
Numpy
Keras
Pytorch
TensorFlow
mxnet
learn
Scikit-learn
Pandas

Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
`https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c`

## Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.
- Hiding under the hood **low level details**.

# Big Data and Machine Learning Ecosystem



Claire D. Costa. Best Python Libraries for Machine Learning and Deep Learning.
`https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c`

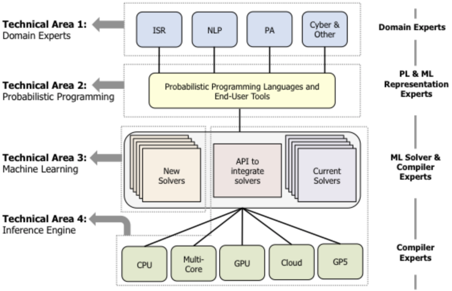
## Big Data and Machine Learning Libraries:

- **High-quality**, well-maintained and open-source libraries
- They try to provide **high-level abstractions**.
- Hiding under the hood **low level details**.
- Increase the **adoption** of these technologies.

# What about in Probabilistic AI?

Which are the "high-level libraries" in Probabilistic AI?

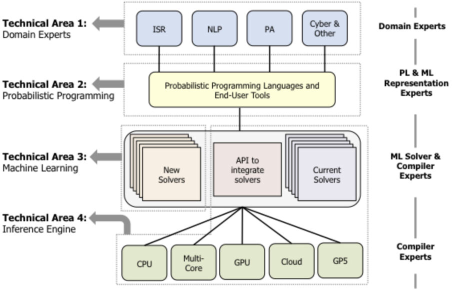
# Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture

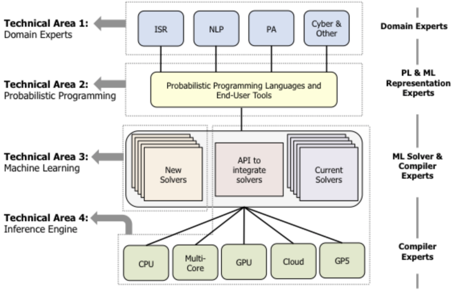
# Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.

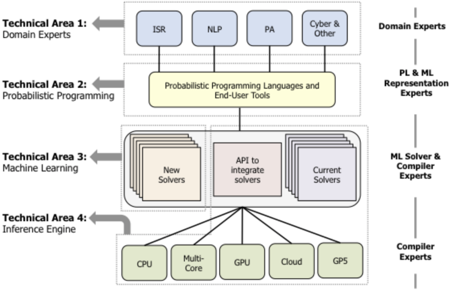
# Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.
- ML experts will focus on the development of new ML solvers.

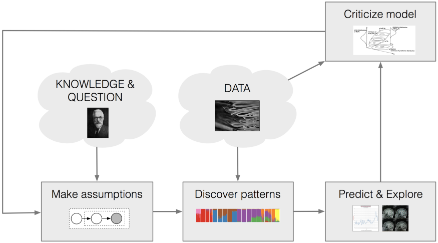
# Why PPLs?



PPLs as high-level programming languages for **probabilistic machine learning systems**:

- Stacked architecture
- Different Domain Experts will code their models using the same language.
- ML experts will focus on the development of new ML solvers.
- Compile experts will focus on running these ML solvers on specialized hardware.

# Box’s Loop



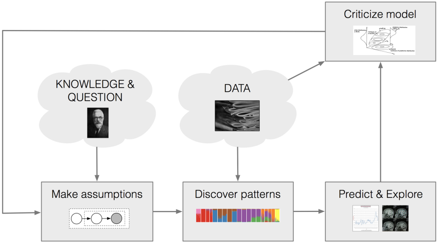
[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

## Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.

# Why PPLs?

## Box's Loop



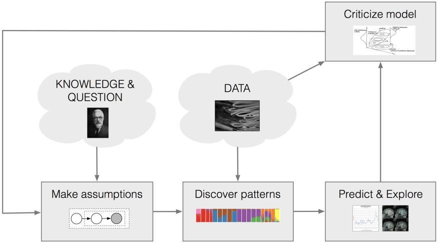
[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

### Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.

# Why PPLs?

## Box's Loop



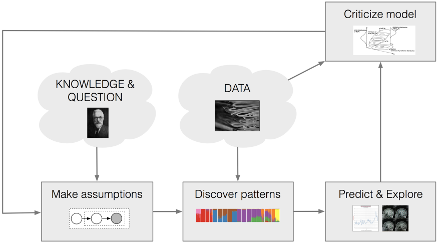
[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

### Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.
- Reduce the necessary level of expertise.

# Why PPLs?

## Box's Loop

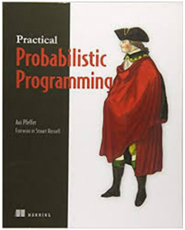


[Box, 1980; Rubin, 1984; Gelman+ 1996; Blei, 2014]

### Benefits of PPLs for developing probabilistic machine learning systems:

- Simplify probabilistic machine learning model code.
- Reduce development time and cost to encourage experimentation.
- Reduce the necessary level of expertise.
- "Democratization" of the development of probabilistic ML systems.

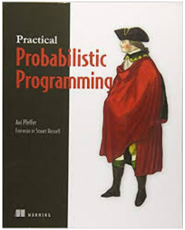
# PPLs



## 1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.

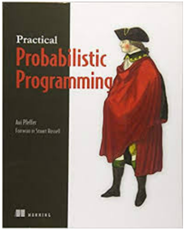
# PPLs



## 1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).

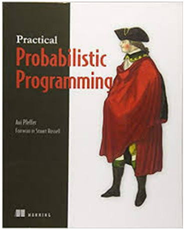
# PPLs



## 1st Generation of PPLs :

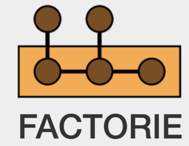
- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).
- Inference engine based on Monte Carlo methods.

# PPLs



## 1st Generation of PPLs :

- Bugs, WinBugs, Jags, Figaro, etc.
- Turing-complete probabilistic programming languages. (i.e. they can represent any computable probability distribution).
- Inference engine based on Monte Carlo methods.
- They did not scale to large data samples/high-dimensional models.



## 2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.

# PPLs

infer.net

FACTORIE

AMiDST TOOLBOX

## 2nd Generation of PPLs :

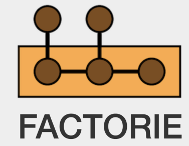
- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.

# PPLs

## 2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.
- They did scale to large data samples/high-dimensional models.

# PPLs



## 2nd Generation of PPLs :

- Infer.net, Factorie, Amidst, etc.
- Inference engine based on message passage algorithms and/or variational inference methods.
- They did scale to large data samples/high-dimensional models.
- Restricted probabilistic model family (i.e. factor graphs, conjuage exponential family, etc.)

# PPLs

## 3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.

# PPLs

## 3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.

# PPLs

## 3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.

# PPLs

## 3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.

# PPLs

## 3rd Generation of PPLs :

- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).

# PPLs

## 3rd Generation of PPLs :

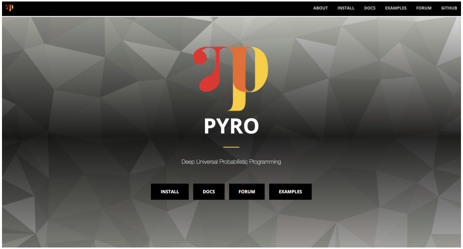
- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).
    - Specialized hardware like GPUs, TPUs, etc.

# PPLs

## 3rd Generation of PPLs :

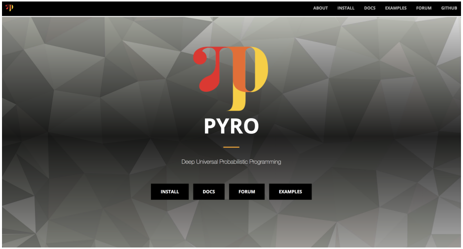
- TensorFlow Probability, Pyro, PyMC3, InferPy, etc.
- Black Box Variational Inference and Hamiltonian Monte-Carlo.
- They did scale to large data samples/high-dimensional models.
- Turing-complete probabilistic programming languages.
- Rely on deep learning frameworks (TensorFlow, Pytorch, Theano, etc).
    - Specialized hardware like GPUs, TPUs, etc.
    - Automatic differentiation methods.

# Pyro



Deep Universal Probabilistic Programming

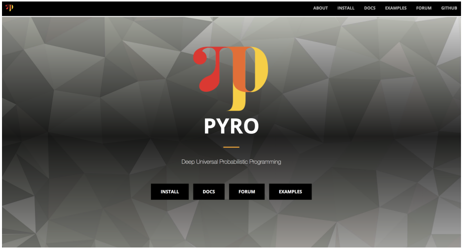
# Pyro



## Pyro’s main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)

# Pyro



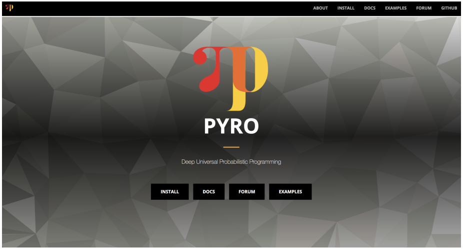
PYRO

Deep Universal Probabilistic Programming

## Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).

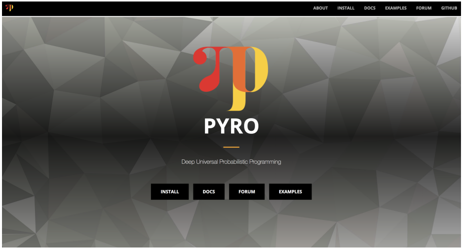
# Pyro



## Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.

# Pyro



## Pyro's main features (www.pyro.ai) :

- Initially developed by UBER (the car riding company).
- Actively maintained by community contributors (there is dedicated team at Broad Institute)
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.

https://github.com/PGM-Lab/2022-ProbAI