Why Deep Learning rocks

A philosophical note

Andrey Ustyuzhanin, Maxim Borisyak, Mikhail Usvyatsov, Alexander Panin

Yandex School of Data Analysis
Yandex Data Factory

Laboratory of Methods for Big Data Analysys
National Research University Higher School of Economics

No free lunch

No free lunch theorem

No free lunch theorem states that **in average** all learning algorithms are equally bad at learning.
Examples:

- crazy algorithm:

$$f(x) = \left\lfloor \left(\left\lceil \sum_i x_i \right\rceil \mod 17 + 1027 \right)^{\pi} \right\rfloor \mod 2$$

- SVM

perform equally well **in average**.

No free lunch

$$X = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 0 \\ 1 \\ 1 \\ ? \end{pmatrix} = y$$

No free lunch

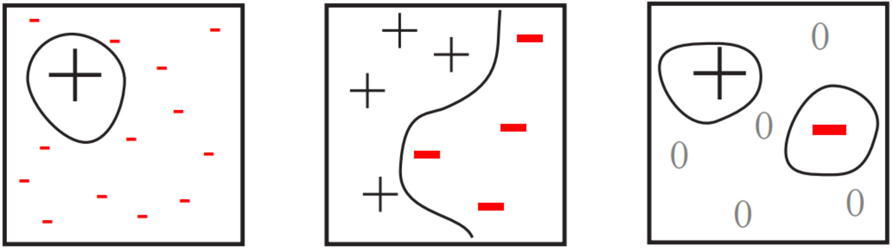
$$X = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 0 \\ 1 \\ 1 \\ ? \end{pmatrix} = y \qquad X = \begin{pmatrix} 2 \\ 7 \\ 1 \\ 0 \\ 4 \\ 3 \\ 5 \\ 6 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 0 \\ 1 \\ 1 \\ ? \end{pmatrix} = y \qquad X = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 0 \\ 1 \\ ? \\ 3 \end{pmatrix} = y$$

No free lunch

$$X = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 0 \\ 1 \\ 1 \\ 2 \end{pmatrix} = y \qquad X = \begin{pmatrix} 2 \\ 7 \\ 1 \\ 0 \\ 4 \\ 3 \\ 5 \\ 6 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 0 \\ 1 \\ 1 \\ 2 \end{pmatrix} = y \qquad X = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \\ 3 \end{pmatrix} = y$$

$$y = \left| \sum_{i=0}^{2} 2^i x_i - 4 \right| = |x - 4|$$

No free lunch theorem



Possible learning algorithm behaviours in **problem space:**

- + - better than the average;
- — - worse than the average.

Are Machine Learning algorithms useless?

Are Machine Learning algorithms useless?

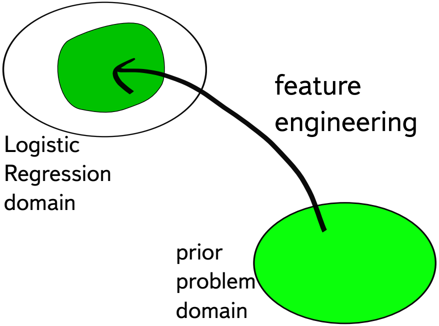
No.

Are Machine Learning algorithms useless?

- Machine Learning algorithms have data scientists;
- data scientists are additional source of prior information;
- prior information helps to bypass No Free Lunch Theorem.

Traditional Machine Learning

- analyse the problem and make assumptions;
- pick an algorithm from a toolkit (e.g. logistic regression);
- provide assumptions suitable for the algorithm **(feature engineering)**.



Discussion

- this approach works well for traditional datasets with a small number of features:
- e.g. Titanic dataset:

passenger class	name	sex	age	fare	...

Essentially, performance of the algorithm depends data scientist's ability to generate features.

- but our abilities are limited.

Kitten

Kitten seen by a machine

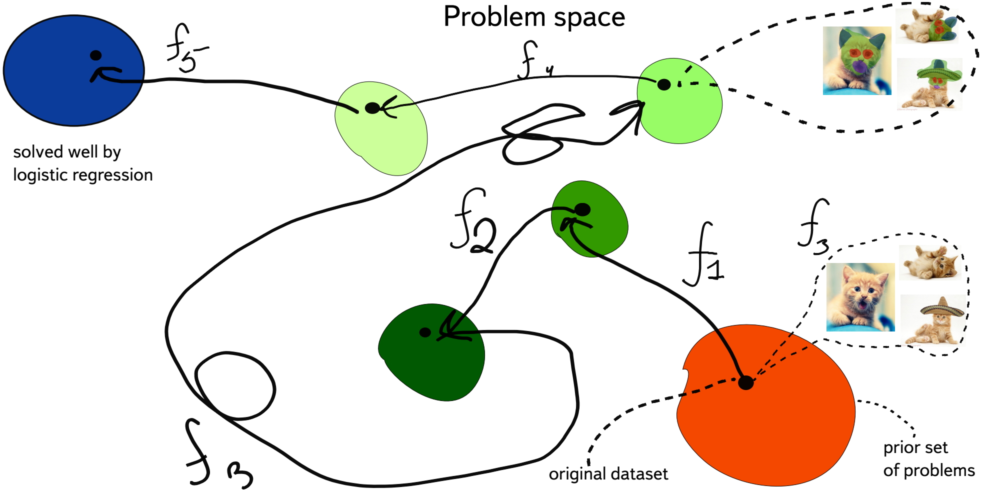
```
[[ 22  25  28  32  29 ...,  58  36  35  34  34]
 [ 26  29  30  31  36 ...,  65  38  42  41  42]
 [ 27  28  31  30  40 ...,  84  58  51  52  44]
 [ 27  26  27  29  43 ...,  90  70  60  57  43]
 [ 20  26  28  28  31 ...,  83  73  62  52  45]
 ...,
 [173 187 180 183 184 ..., 170 227 244 219 199]
 [193 199 194 188 185 ..., 181 197 201 209 187]
 [175 177 156 166 171 ..., 226 215 194 185 182]
 [161 159 160 187 178 ..., 216 193 220 211 200]
 [178 180 177 185 164 ..., 190 184 212 216 189]]
```

Solution?

- edge detection;
- image segmentation;
- eyes, ears, nose models;
- fit nose, ears, eyes;
- average color of segments;
- standard deviation of color segments;
- goodness of fit for segments;
- kitten's face model;
- feed it to Logistic Regression

* not an practical solution

Solution?


Problem space
f_5
f_4
solved well by
logistic regression
f_2
f_1
f_3
f_3
original dataset
prior set
of problems

Solution?

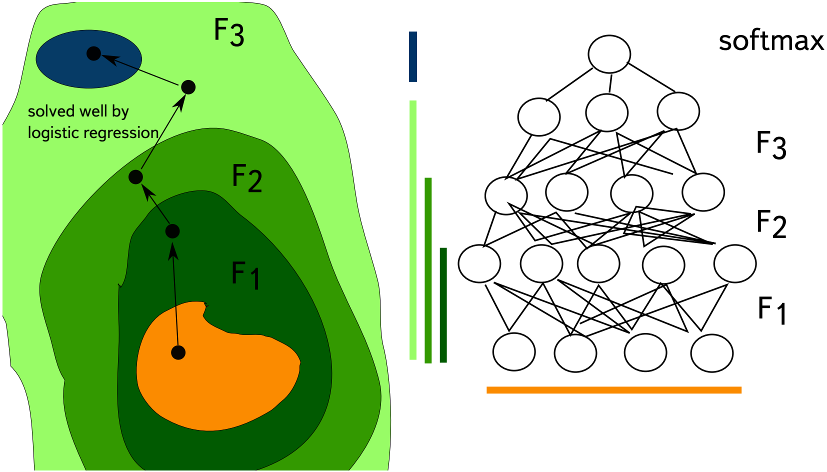
Perhaps, more Machine Learning and less Human Engineering?

Deep Learning

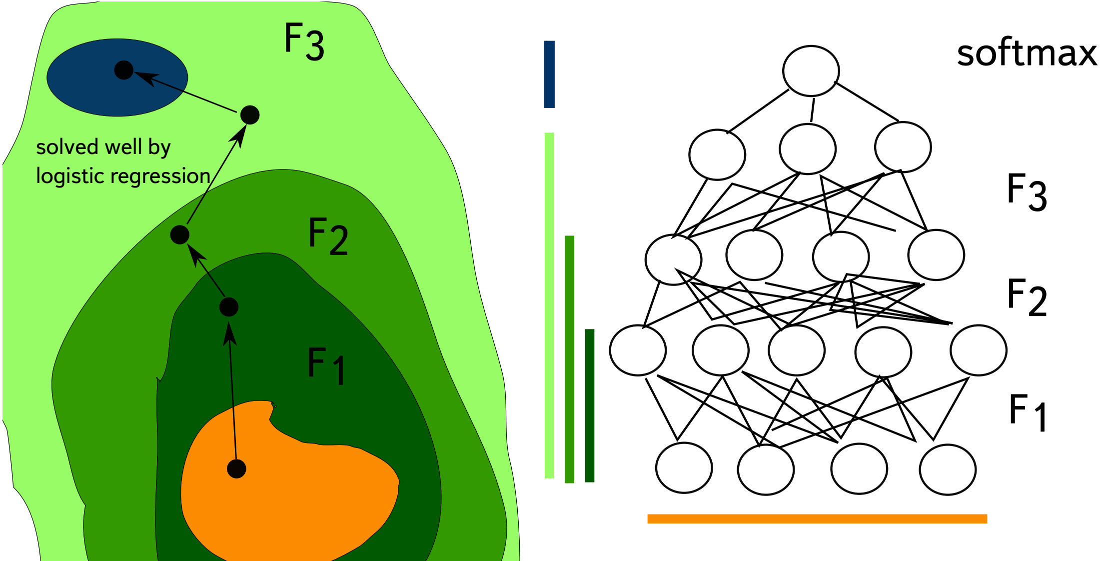
Deep Learning

Let's learn features!

Original picture



Deep Learning

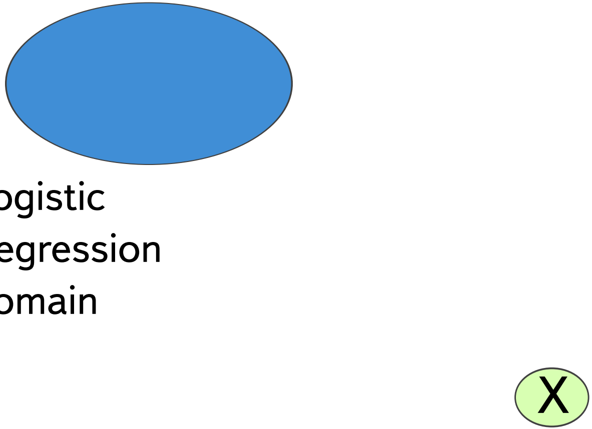


How

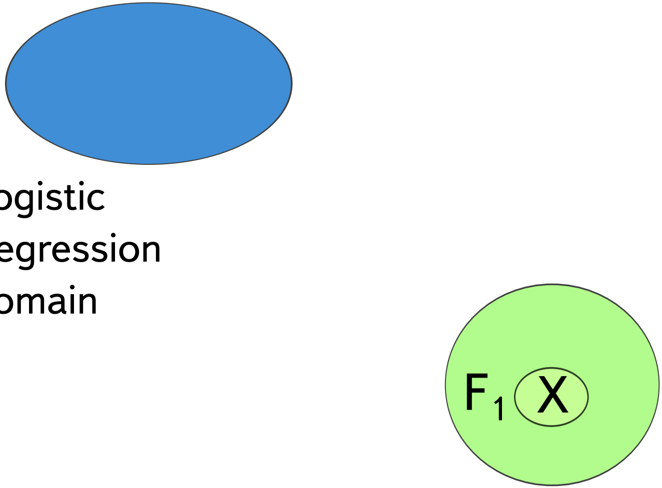
- apply some abstract simple transformation to the original input:

$$X \to f(X) \;\cdots\; y$$

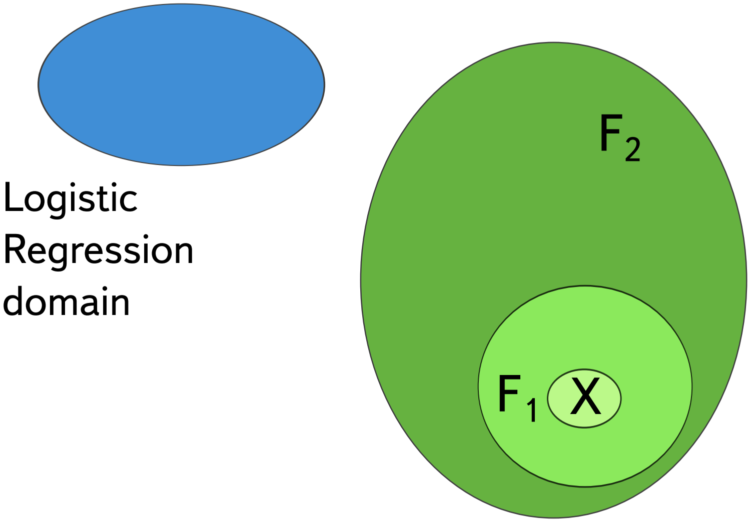
Original picture



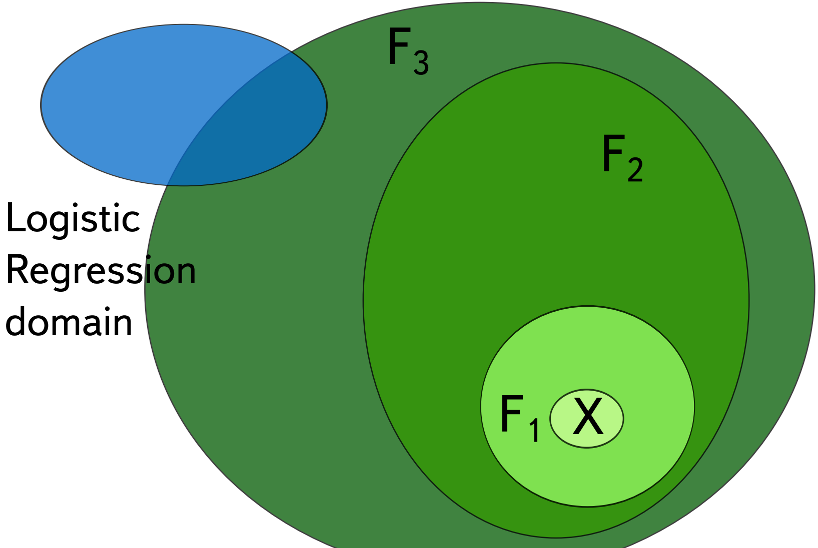
Some transformation applied



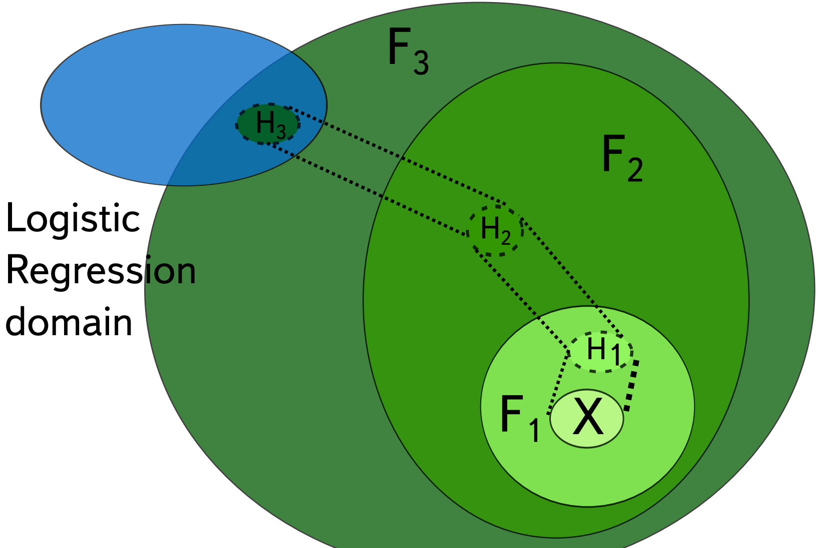
How



How



How



Kitten

- use convolutions;
- use convolutions again;
- and again;
- and again;
- ...
- logistic regression.

Why deep?

- new set of features is generated from previous one by a simple learnable transformation;
- each step increases complexity of feature generation;
- high-level features (kitten or puppy) are complex ones thus requires a lot of steps;
- therefore, deep.

Deep Learning

- is not a superior algorithm;
- is not a single algorithm;
- is a framework;
- very flexible framework;
- allows to express our assumptions in much more general way.

Why DL rocks

Solves much harder problems:

- purely a human factor:
  - research time;
  - limits of our intuition and understanding of the world; A framework:
  - algorithms are like constructor;
  - possible to solve almost every possible problem:
    - classification;
    - regression;
    - clasterisation;
    - sample generation...

Downsides

- learning features requires data;
  - big datasets;
  - big computational resources (GPUs);
- there is almost always a better algorithm:
  - with hand-made features;
  - probably constructed by a super-intelligent alien.

Summary

Summary

Deep Learning:

- a flexible framework;
- allows to express you knowledge easier;
- solves much harder problems.