

Why Deep Learning rocks

A philosophical note

Andrey Ustyuzhanin, Maxim Borisjak, Mikhail Usvyatsov,
Alexander Panin

No free lunch

Terminology

Machine Learning is about learning algorithms A that:

- › defined on sample set \mathcal{X} (e.g. \mathbb{R}^n) and targets \mathcal{Y} (e.g. $\{0, 1\}$);
- › take a problem (dataset) $D = (X, y) \subseteq \mathcal{X} \times \mathcal{Y}$;
- › learn relation between \mathcal{X} and \mathcal{Y} ;
- › and return prediction function:

$$\begin{aligned} A(D) &= f \\ f : \mathcal{X} &\rightarrow \mathcal{Y} \end{aligned}$$

No free lunch theorem

No free lunch theorem states that **on average** by all datasets all learning algorithms are equally bad at learning.

Examples:

- › crazy algorithm:

$$f(x) = \left\lfloor \left(\left[\sum_i x_i + \theta \right] \mod 17 + 1027 \right)^\pi \right\rfloor \mod 2$$

- › SVM

perform equally well **on average**.

IQ test: try to learn yourself!

First question from MENSA website:

Following the pattern shown in the number sequence below, what is the missing number?

1, 8, 27, ?, 125, 216

Possible answers:

- > 36
- > 45
- > 46
- > 64
- > 99

IQ test: try to learn yourself!

First question from MENSA website:

Following the pattern shown in the number sequence below, what is the missing number?

X_{train}	1	2	3	5	6
y_{train}	1	8	27	125	216

$$X_{\text{test}} = (4,)$$

IQ test: try to learn yourself!

My solution:

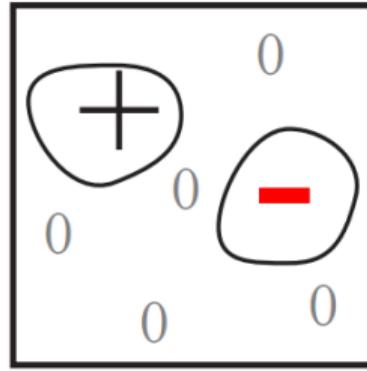
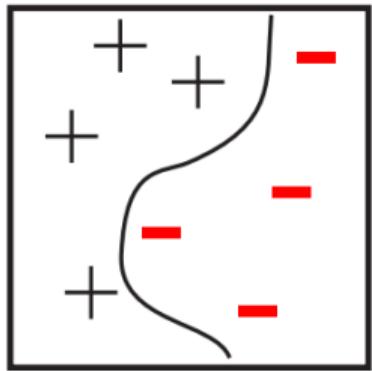
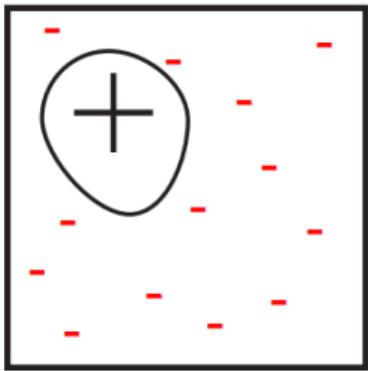
$$y = \frac{1}{12}(91x^5 - 1519x^4 + 9449x^3 - 26705x^2 + 33588x - 14940)$$

› fits perfectly!

My answer:

› 99

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?

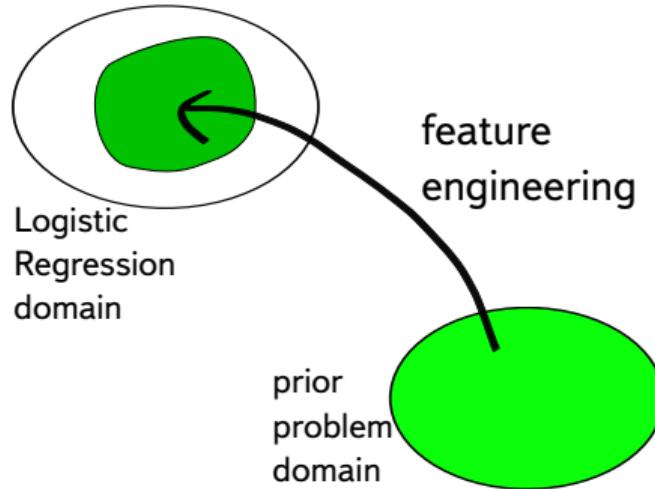
No.

Are Machine Learning algorithms useless?

- › No Free Lunch theorem applies to:
 - › one learning algorithm;
 - › against all possible problems.
- › in real world we have:
 - › **data scientist** with prior knowledge of the world;
 - › problem description;
 - › data description;
 - › a set of standard algorithms.

Traditional Machine Learning (simplified)

- › 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 on:

- › knowledge of the domain;
- › feature generation skills;
- › understanding of assumptions behind standard algorithms.

Kitten

Let's try to detect kittens!



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]

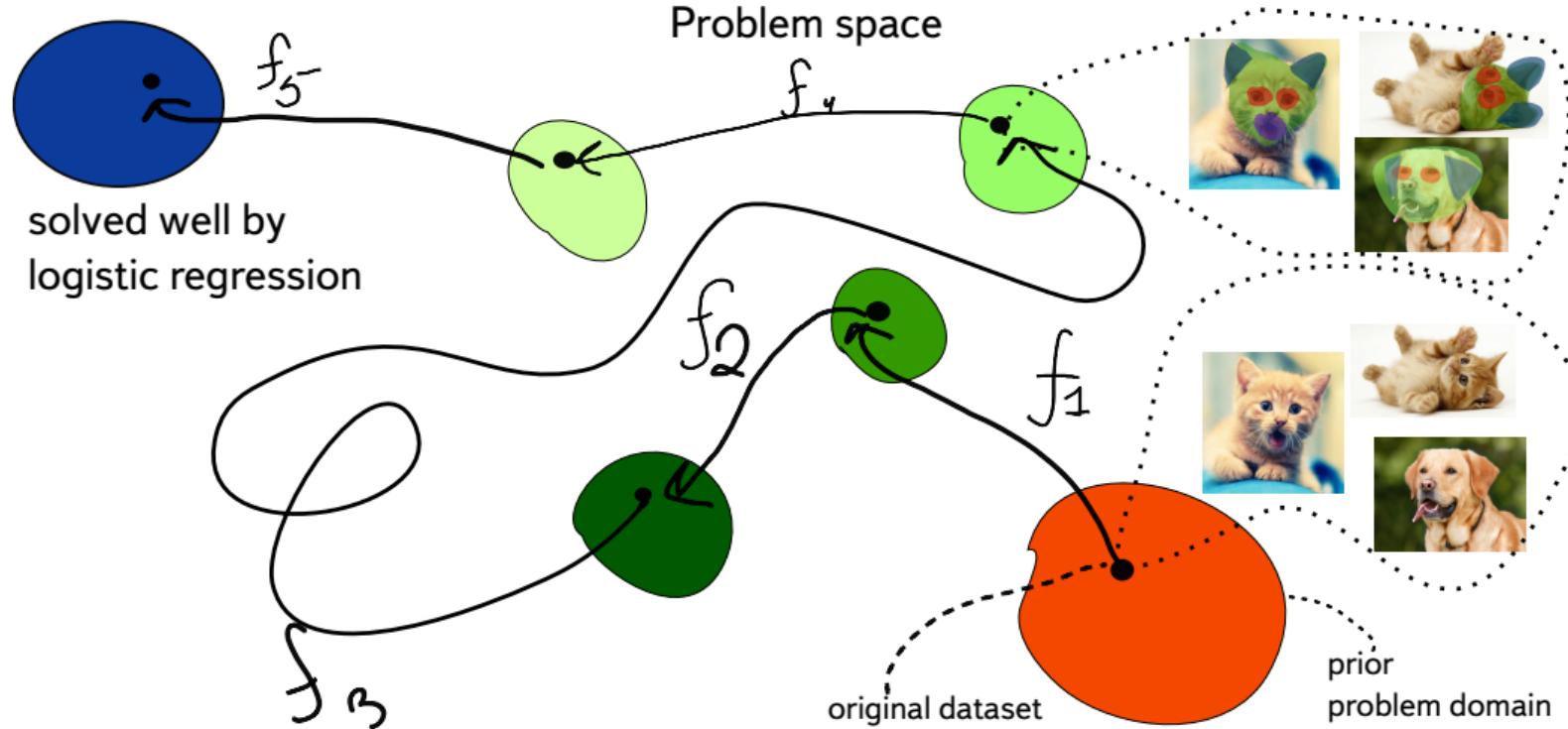
...,

[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;
- › logistic regression.

Solution?



Solution?

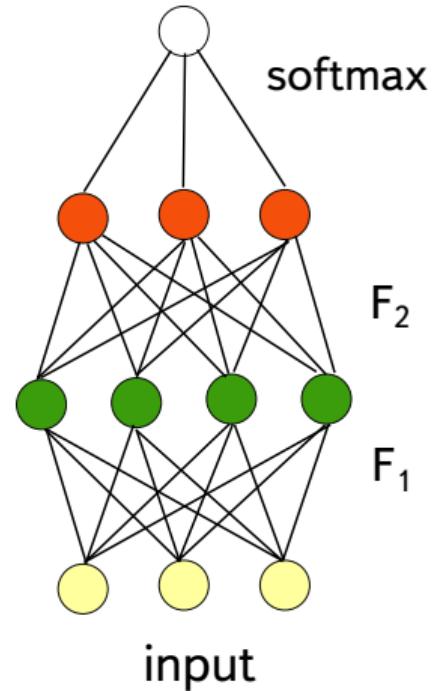
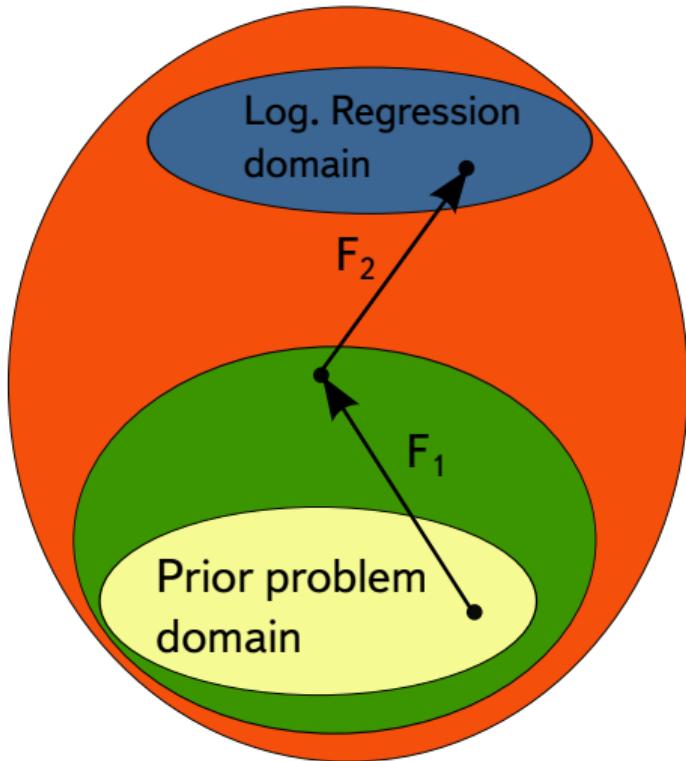
Perhaps, more Machine Learning and less Human Engineering?

Deep Learning

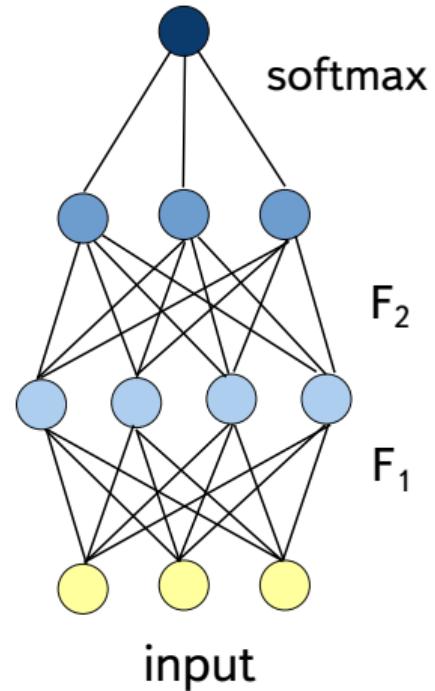
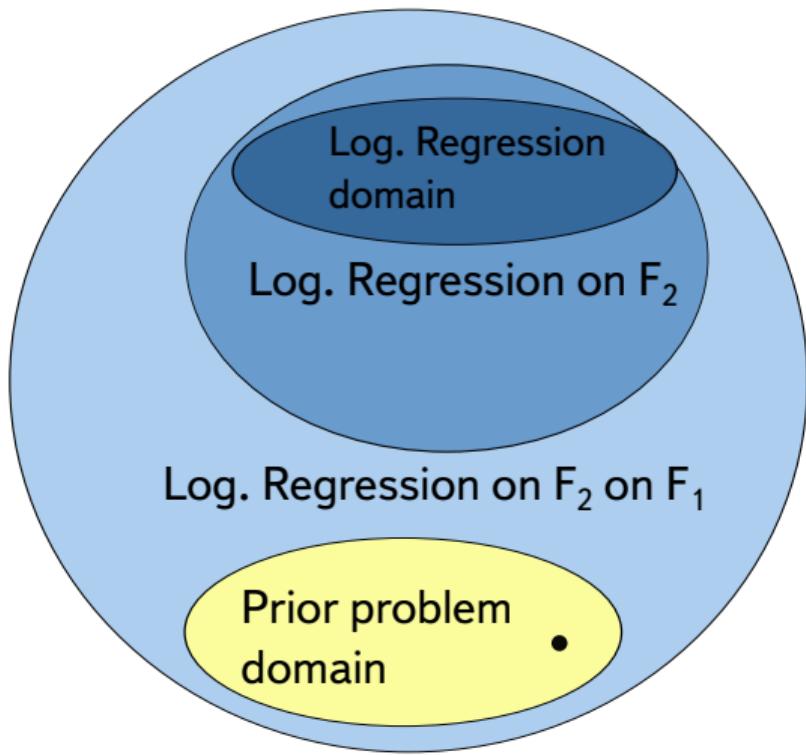
Deep Learning

Let's learn features!

Deep Learning



Deep Learning



Kitten

Traditional approach:

- › 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;
- › logistic regression.

Deep Learning:

- › convolutions;
- › logistic regression.

Deep Learning

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

Why DL rocks

- › can crack much harder problems;
 - › it is easier to formulate models for features than features itself;
- › easy to construct networks:
 - › merge together;
 - › bring new objectives;
 - › inject something inside network;
 - › build networks inside networks;
 - › any differentiable magic is allowed.

Example

A problem contains groups of features:

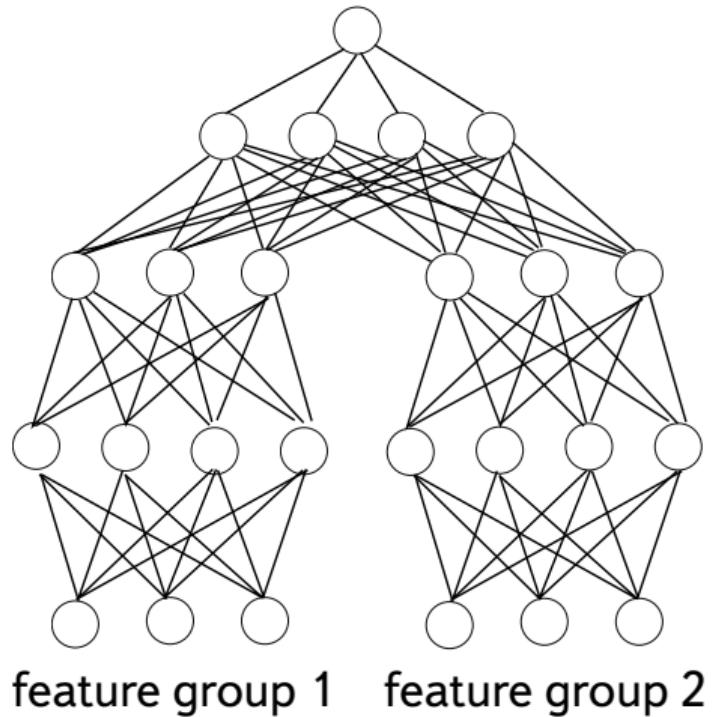
- › image;
- › sound features;

Prior knowledge:

- › features from different group should not interact directly;

Example of a solution:

- › build a subnetwork upon each group of features;
- › merge them together.



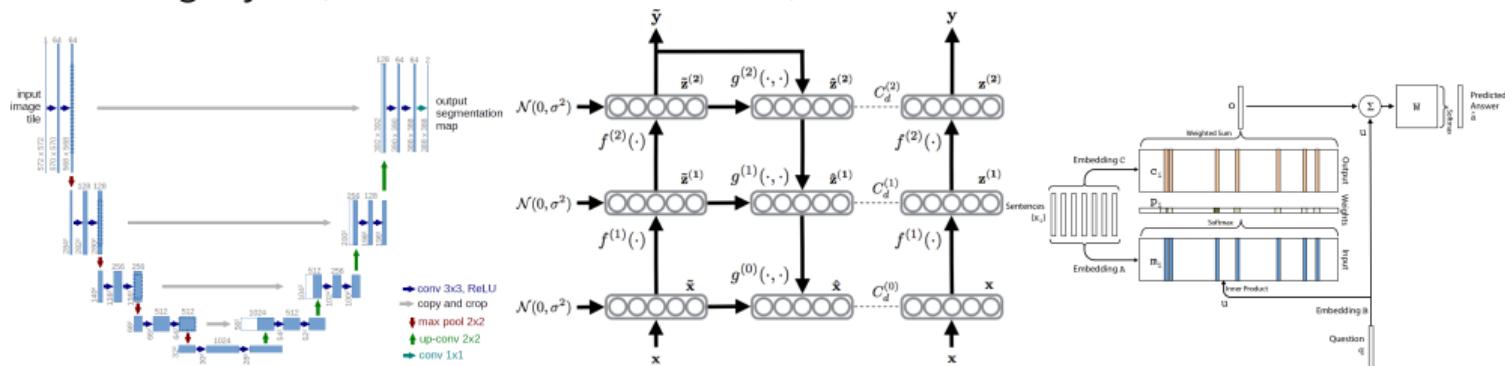
Almost Free Lunch

Machine Learning Algorithm

- › parametrized model - how to produce predictions;
- › search procedure:
 - › initial guess for parameters;
 - › optimization procedure.

Hacking model

- › hacking layers:
 - › restrictions on weights: convolutions, ...;
 - › new operations: pooling, kernels, ...;
 - › specific unit behaviour: GRU, LSTM units;
- › combining layers, architecture of network;



Images show: U-net, ladder net, end-to-end memory network.

Hacking model

- › restrictions on search space:
 - › regularization, e.g.:

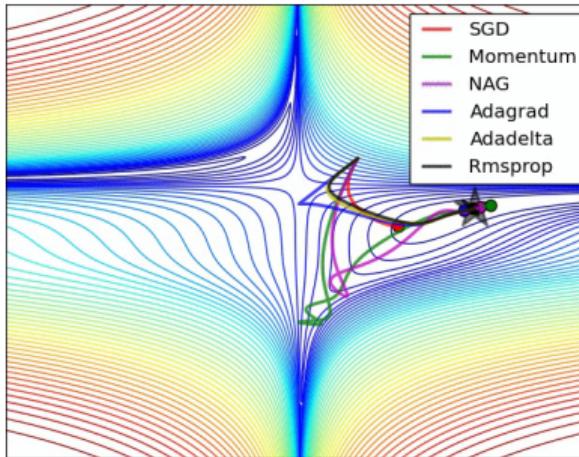
$$\mathcal{L} = \mathcal{L}_{\text{cross-entropy}} + \alpha \|W\|_2^2$$

- › regularization with respect to solution W_0 of a similar problem:

$$\mathcal{L} = \mathcal{L}_{\text{cross-entropy}} + \alpha \|W - W_0\|_2^2$$

Hacking search procedure

- › SGD-like methods:
 - › adam, adadelta, adamax, rmsprop;
 - › nesterov momentum;
- › quasi-Newton methods;



Hacking search procedure

- › data augmentation:
 - › shifts, rotations, ...:
 - › searching for a network that labels shifted, rotated, ... samples the same way as original ones;
 - › random noise:
 - › pushing separation surface farther from samples;
- › interference with network:
 - › drop-out, drop-connect:
 - › searching for a robust network.

Hacking search procedure

- › hacking objectives:
 - › introducing loss for each layer:

$$\mathcal{L} = \mathcal{L}_n + \sum_{i=1}^{n-1} C_i \mathcal{L}_i$$

where:

- › \mathcal{L}_i - loss on i -th layer.
- › Deeply Supervised Networks:
 - › searches for network that obtains good intermediate results.

Hacking initial guess

- › solution for a similar problem as initial guess for search;
- › pretraining on a similar dataset:
 - › unsupervised pretraining on unlabeled samples;
 - › supervised pretraining.

Almost Free Lunch

Any magic is allowed!

... almost any magic.

Summary

Summary

No Free Lunch theorem:

- › Machine Learning is about using prior knowledge about the problem wisely.

Deep Learning:

- › a flexible framework;
- › allows to express prior knowledge ;
- › makes it easier to solves much harder problems.

References

No-Free-Lunch theorem:

- › Schaffer, Cullen. "A conservation law for generalization performance." Proceedings of the 11th international conference on machine learning. 1994.
- › Wolpert, David H. "The supervised learning no-free-lunch theorems." Soft computing and industry. Springer London, 2002. 25-42.
- › Wolpert, David H., and William G. Macready. "No free lunch theorems for optimization." IEEE transactions on evolutionary computation 1.1 (1997): 67-82.

References

Non-sequential network architecture examples:

- › U-network: Von Eicken, Thorsten, et al. "U-Net: A user-level network interface for parallel and distributed computing." ACM SIGOPS Operating Systems Review. Vol. 29. No. 5. ACM, 1995.
- › Ladder Network: Rasmus, Antti, et al. "Semi-supervised learning with ladder networks." Advances in Neural Information Processing Systems. 2015.
- › End-to-end memory: Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. "End-to-end memory networks." Advances in neural information processing systems. 2015.

More resources

A lot of useful links can be found in:

- › Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117.