

Machine Learning and Data Mining

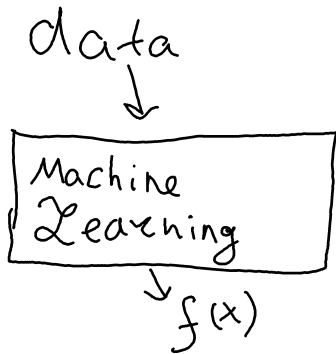
Maxim Borisyak

National Research University Higher School of Economics (HSE)

February 4, 2019

Machine Learning

Machine Learning



- data comes in;
- an algorithm (decision function) comes out.

Typical learning algorithm structure

- model :

$$\text{model} = \{f_{\theta} : \text{inputs} \rightarrow \text{predictions} \mid \theta \in \text{parameters}\};$$

- solver :

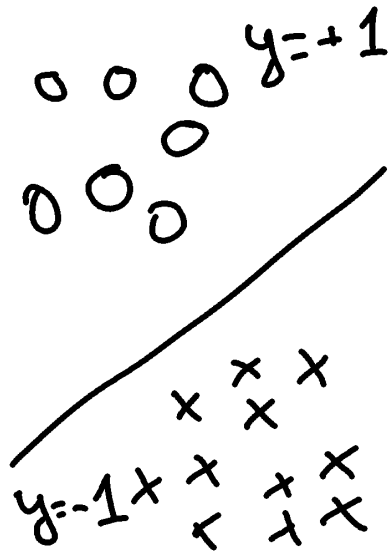
$$\text{solver} : \text{data} \rightarrow \text{model};$$

- loss function:

$$\mathcal{L}(f, \text{data}) = \sum_{x,y \in \text{data}} \text{error}(f(x), y);$$

- optimizer: gradient descent, genetic algorithms etc.
- quality metric: shows how good algorithm is.

Linear models



$$f(x) = w \cdot x + b;$$

$$w \in \mathbb{R}^2, b \in \mathbb{R}$$

$$\mathcal{L}(f) = \sum_i \log(1 + \exp(y_i f(x_i)))$$

Non-linear models



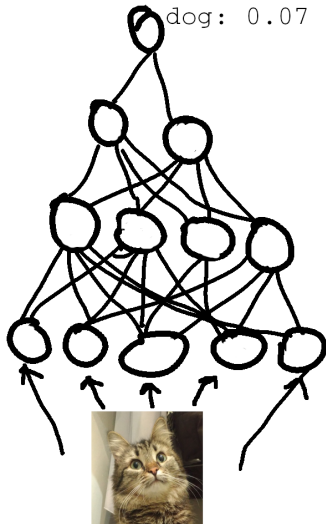
$Y = \text{CAT}$



$Y = \text{DOG}$

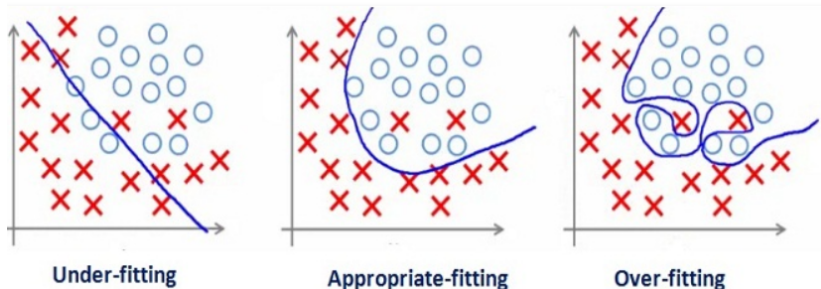
cat: 0.93

dog: 0.07



Over/under-fitting

Over/under-fitting



Put yourself into network shoes.

It is a Diplodocus:



Put yourself into network shoes.

It is a Diplodocus:



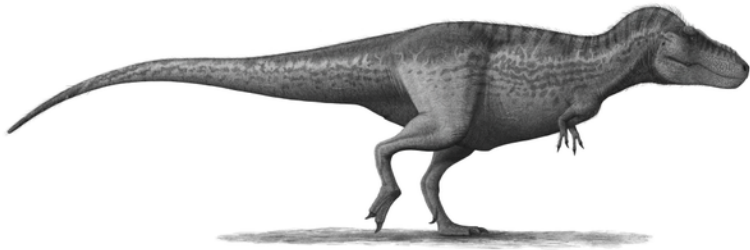
Put yourself into network shoes.

It is not a Diplodocus:



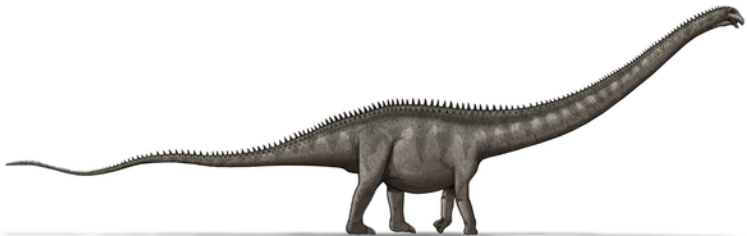
Put yourself into network shoes.

It is not a Diplodocus:



Put yourself into network shoes.

Is it a Diplodocus?



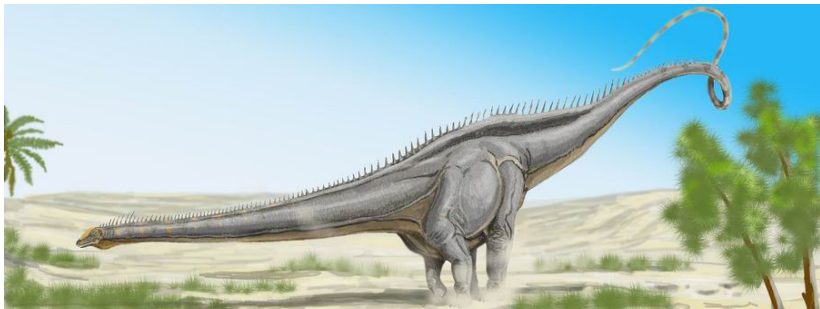
Put yourself into network shoes.

Is it a Diplodocus?



Put yourself into network shoes.

Is it a Diplodocus?



How to detect

- split dataset into two:
 - training set — for selecting decision function;
 - validation set — for **independent** quality metric evaluation.
- $Q_{\text{validation}} \approx Q_{\text{train}}$ and both low — probably underfitting;
- $Q_{\text{validation}} \approx Q_{\text{train}}$ and both high — just right;
- $Q_{\text{validation}} < Q_{\text{train}}$ — overfitting;

Which ML algorithms are the best?

No Free Lunch theorem

Given:

- binary classification;
- metric: off-training set accuracy;
- **uniform prior over problems.**

Any two learning algorithms **on average**
perform equally.

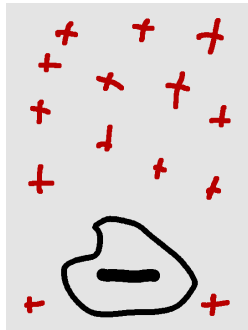
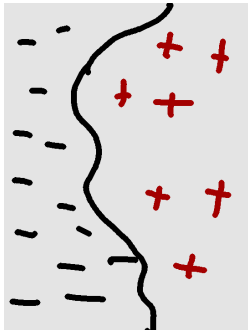
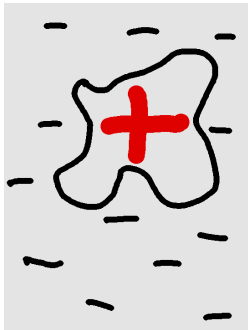
No Free Lunch theorem

Given:

- binary classification;
- metric: off-training set accuracy;

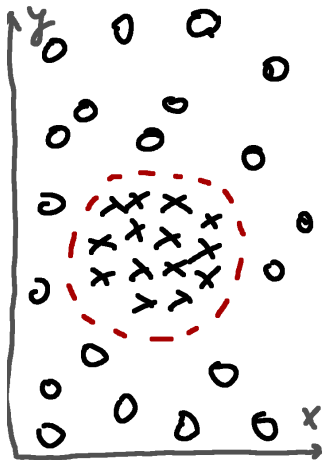
Any increase in performance on one set of problems **must** be accompanied by equivalent decrease on another.

Example

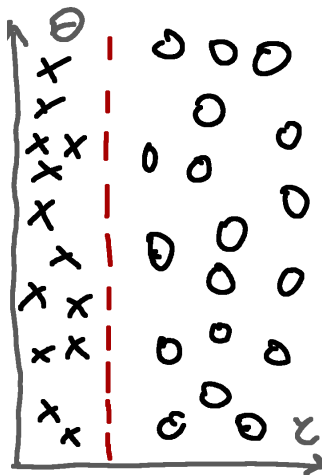


Example

Cartesian

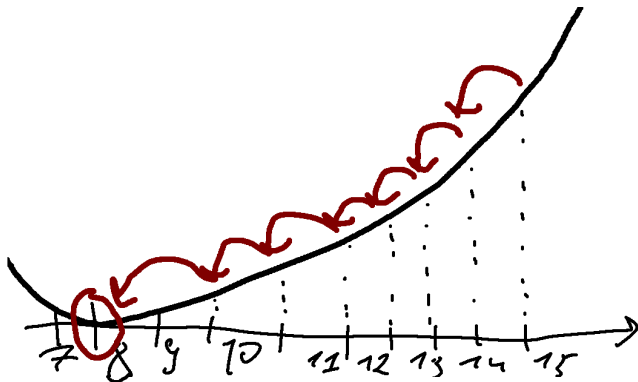


Polar



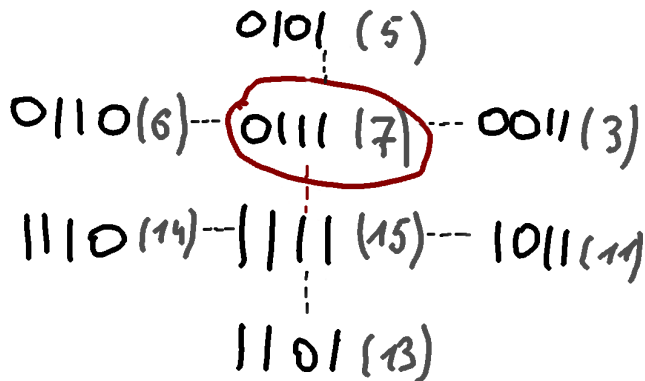
Example

$$\min_{x \in \{0, \dots, 15\}} (x - 8)^2$$



Example

$$\min_{x \in \{0, \dots, 15\}} (x - 8)^2$$



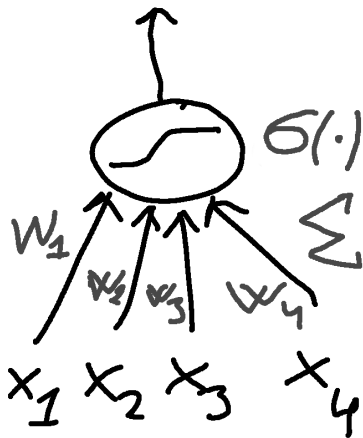
Neural Networks

One learning algorithm can not be better than others¹.

Family of algorithms can.

¹assuming uniform prior over problems

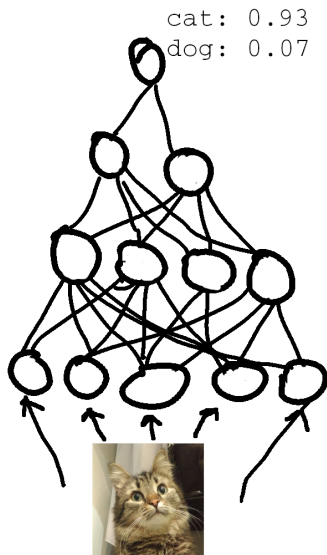
"Neuron"



$$\text{output} = \sigma\left(b + \sum_i w_i x_i\right)$$

- sum of all inputs with weights;
- non-linearity.

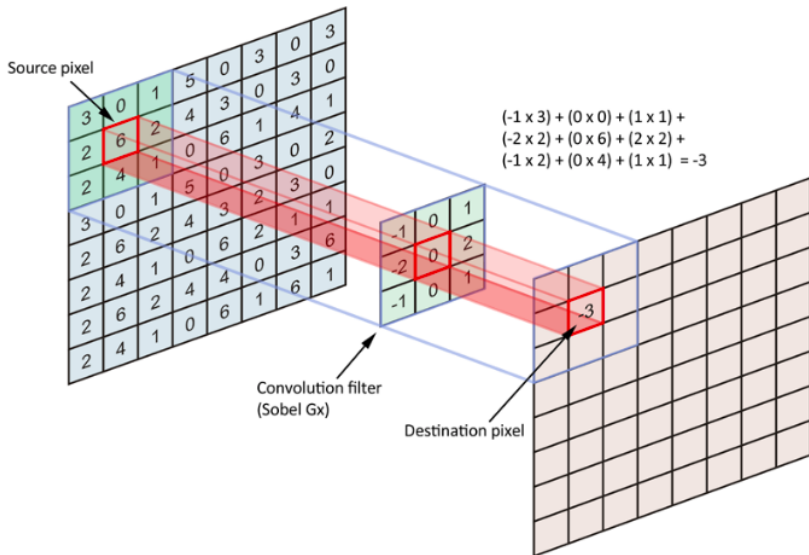
Deep learning



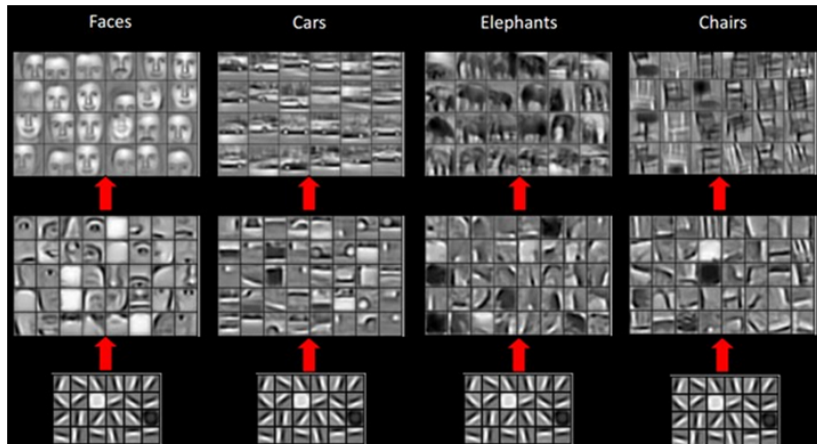
- neurons are organized into layer;
- layer are typically connected sequentially.

Convolutional Networks

Convolutional Networks

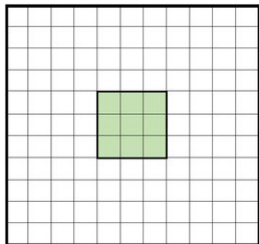


Convolutional Networks

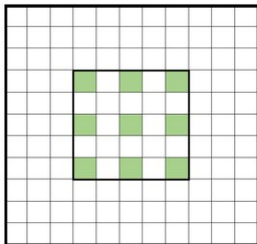


Types of convolution

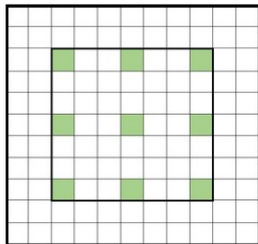
- ordinary / atrous / strided;
- size of the window: 1x1 / 3x3 / 5x5;
- ordinary / depthwise / separable / ...



Kernel 3 x 3
Rate = 1



Kernel 3 x 3
Rate = 2

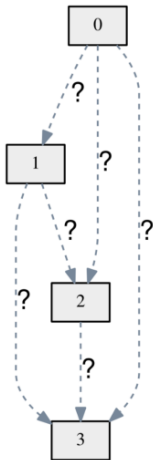


Kernel 3 x 3
Rate = 3

Which one to choose?

Which one to choose?

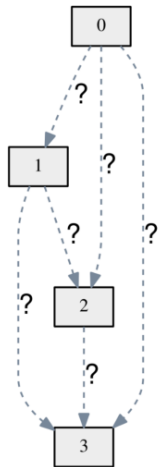
- people are bad at fine tuning;
 - even data scientists;
- checking all possible combinations:
 - 5 layer network with 3 options for each layer:
 - 243 options (~ 1 year).
- evolutionary algorithms;
- Bayesian Optimization;



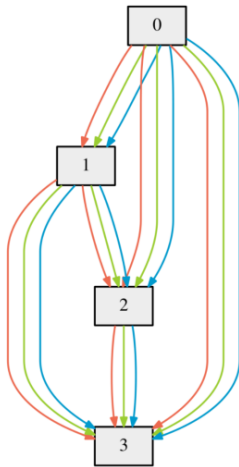
(a)

- $O_i(x)$ — i -th candidate operation:
 - e.g. $O_1(x)$ - convolution 1x1, $O_2(x)$ - convolution 3x3, etc

$$O(x) = \sum_i \frac{\exp(\alpha_i)}{\sum_k \exp(\alpha_k)} O_i(x)$$



(a)



(b)

- X_{train} — data for training;
- $X_{\text{validation}}$ — data for validation;

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{\text{val}}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \arg \min_w \mathcal{L}_{\text{train}}(w, \alpha) \end{aligned}$$

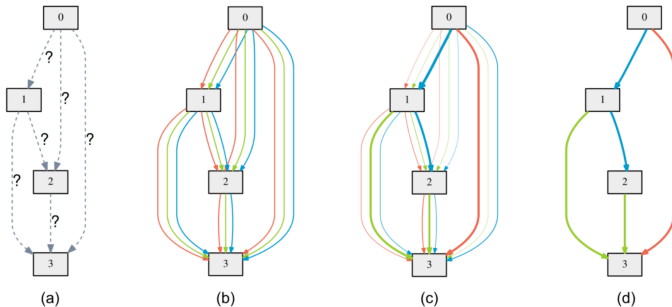


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

Machine Learning and Data Mining

Machine Learning and Data Mining

1. a little bit of theory:
 - No Free Lunch theorem;
 - bias-variance decomposition;
2. meta-algorithms:
 - boosting;
 - bagging;
 - stacking;
3. optimization:
 - gradient optimization;
 - black-box optimization (incl. Bayesian Optimization);
4. Deep Learning:
 - overview, methods and tricks;
 - generative models (incl. RBM, VAE, GAN);
5. Meta Learning:
 - model selection (incl. DARTS);
 - learning to learn; concept learning.

References

- Wolpert DH. The supervised learning no-free-lunch theorems. In Soft computing and industry 2002 (pp. 25-42). Springer, London.
- Liu H, Simonyan K, Yang Y. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055. 2018 Jun 24.
- Sermanet P, Chintala S, LeCun Y. Convolutional neural networks applied to house numbers digit classification. In Pattern Recognition (ICPR), 2012 21st International Conference on 2012 Nov 11 (pp. 3288-3291). IEEE.
- Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report. 1989 Sep;826:1989.