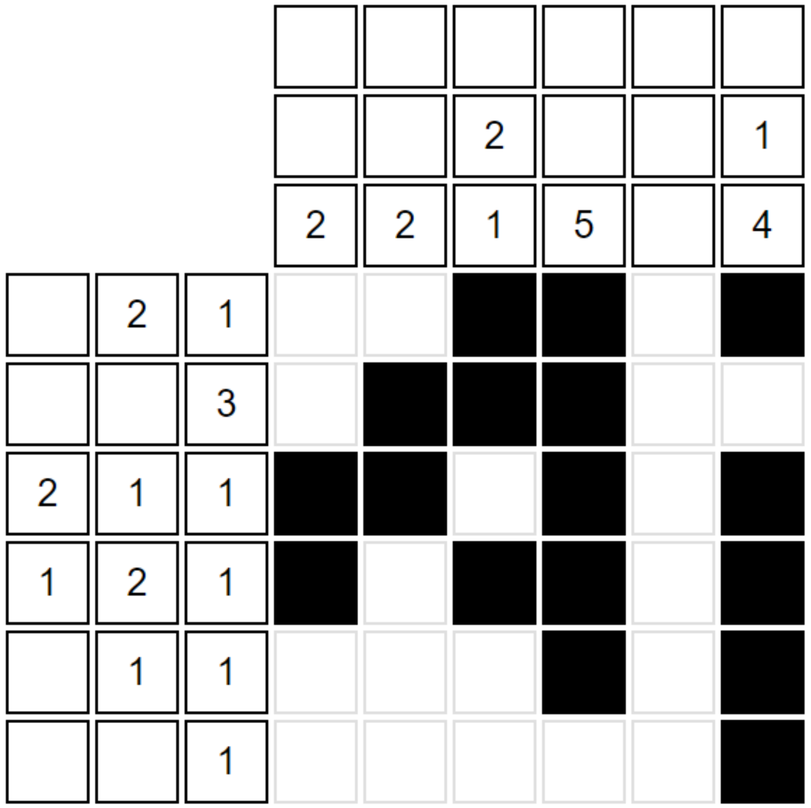
Deep Learning a gyakorlatban
Python és LUA alapon
Nagy házi feladat

Csomor Balázs Patrik

Bendicsek Márton Bendegúz

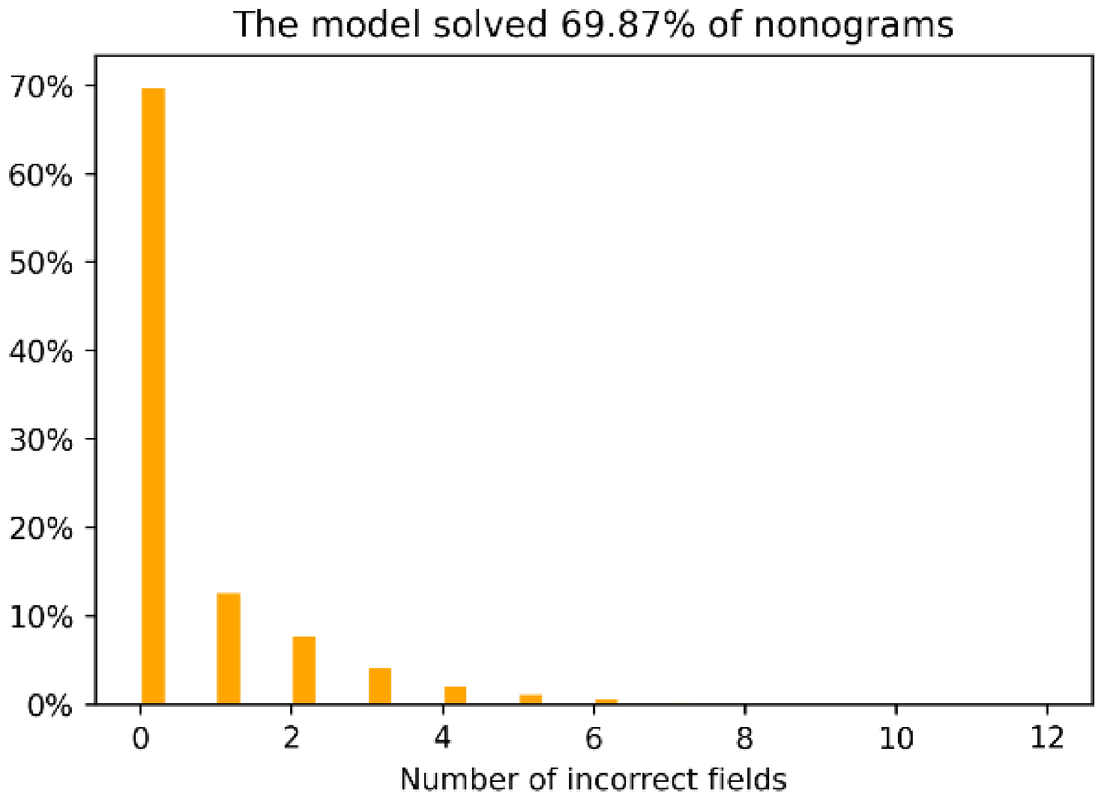
Nonogramok megoldása neurális hálókkal

Solving nonograms using neural networks

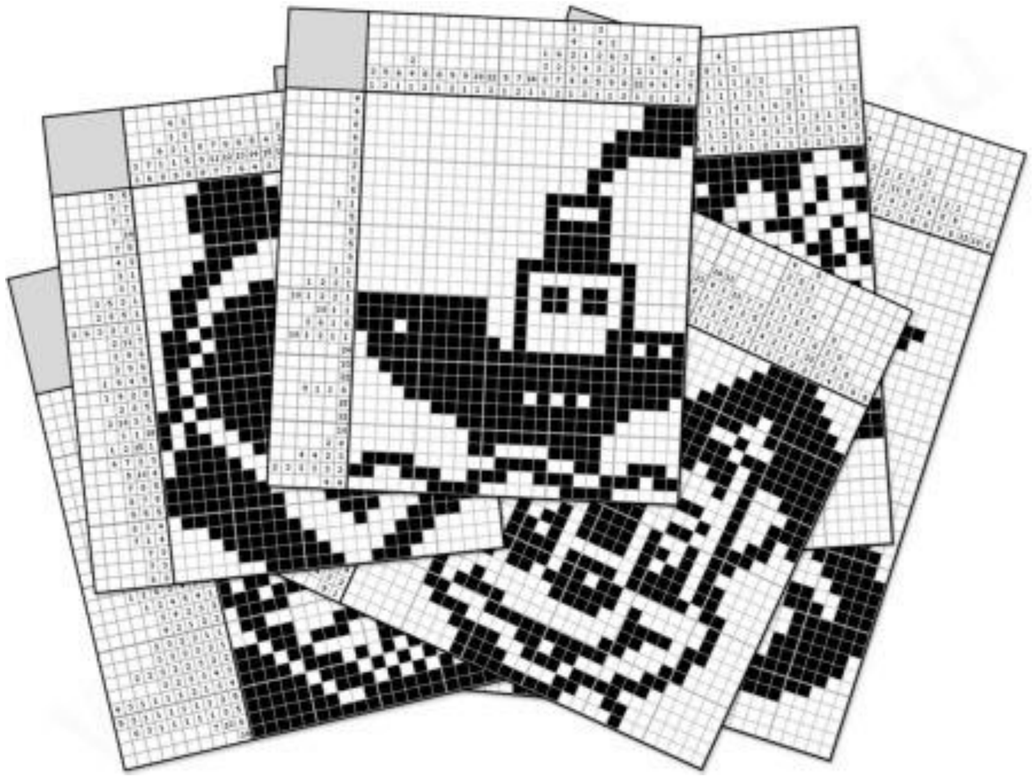


Nonogramok megoldása neurális hálókkal

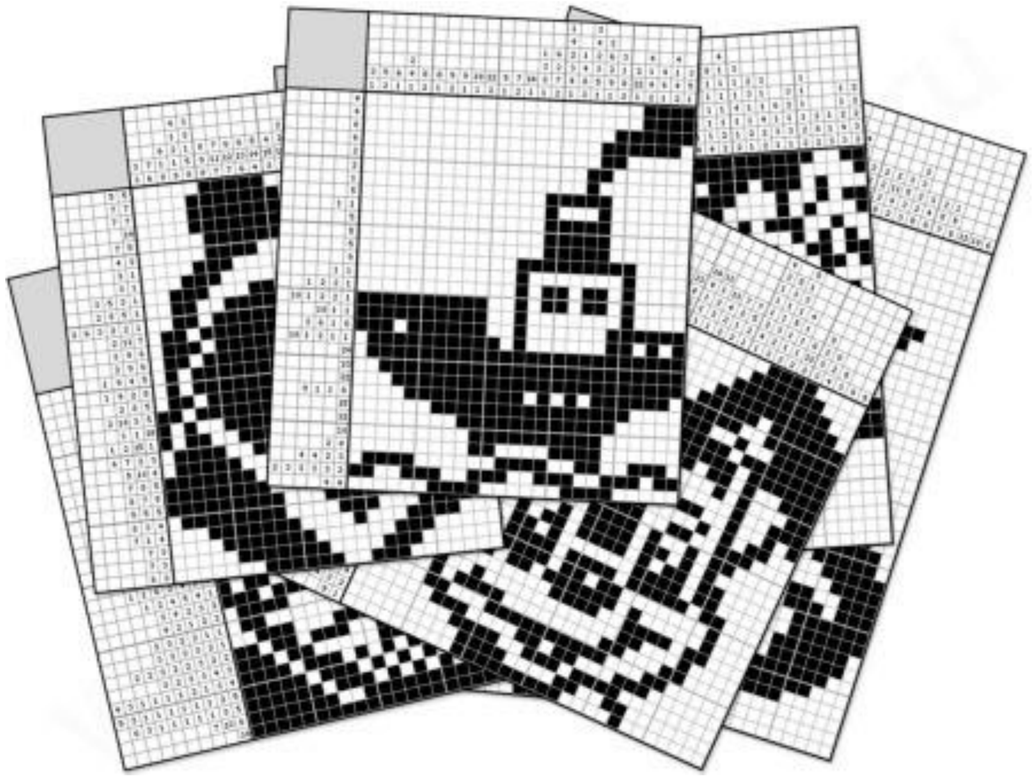
Solving nonograms using neural networks

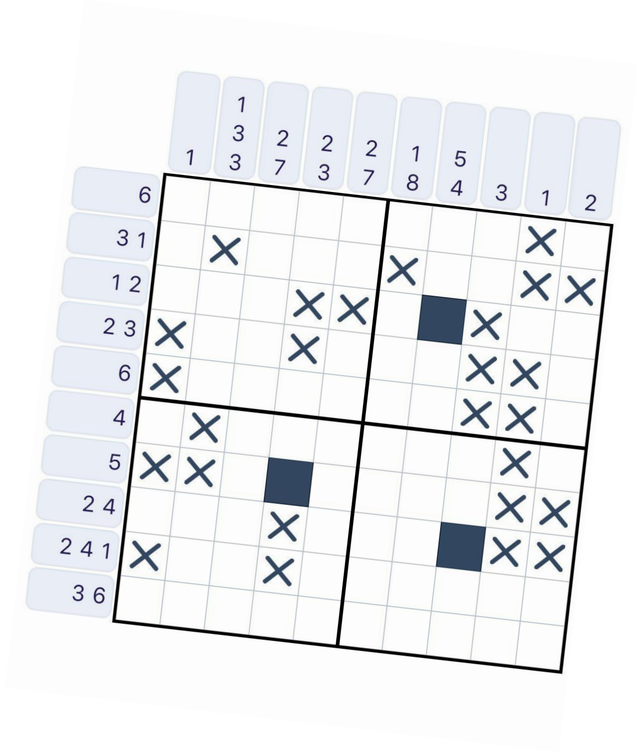


Nonogram

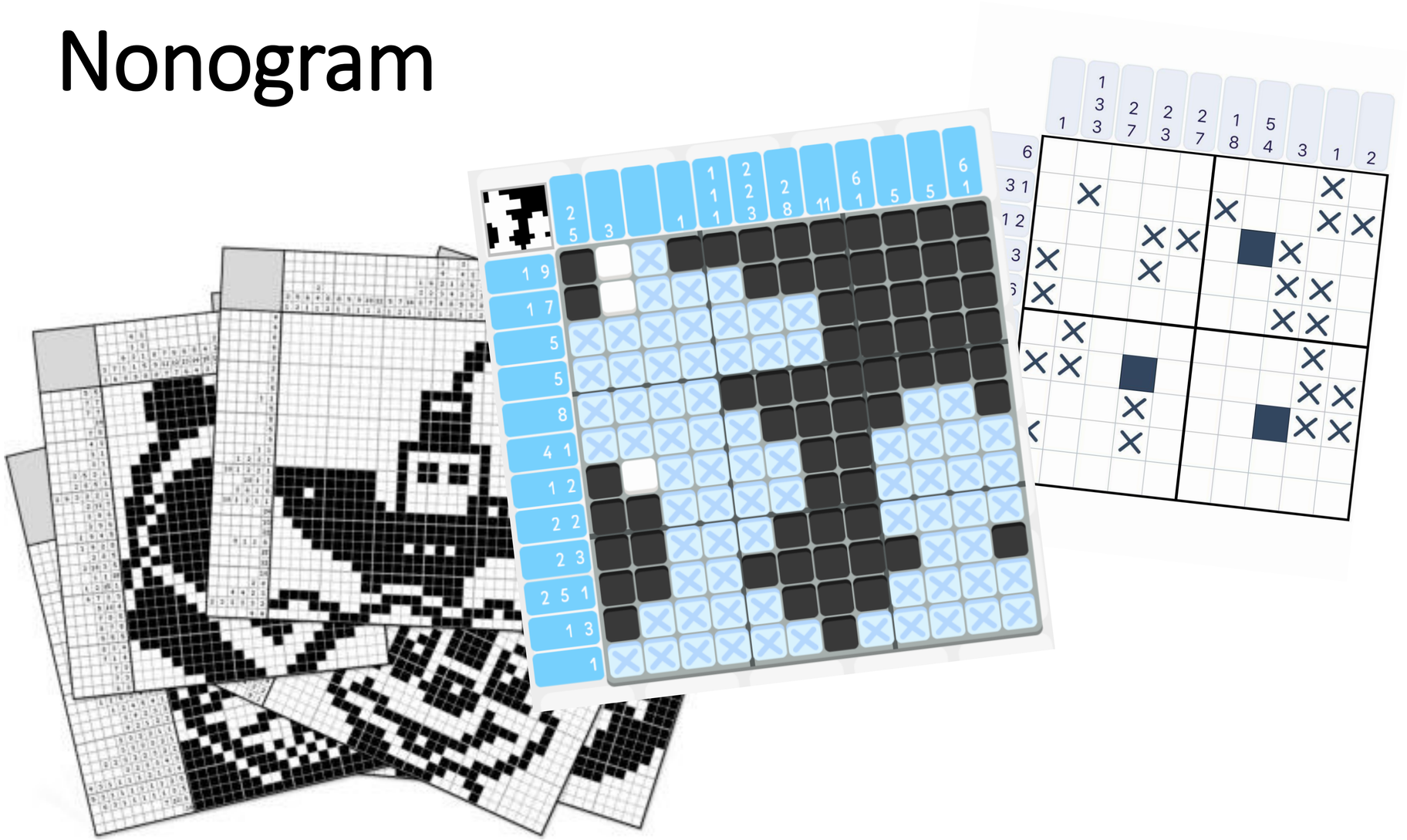


Nonogram

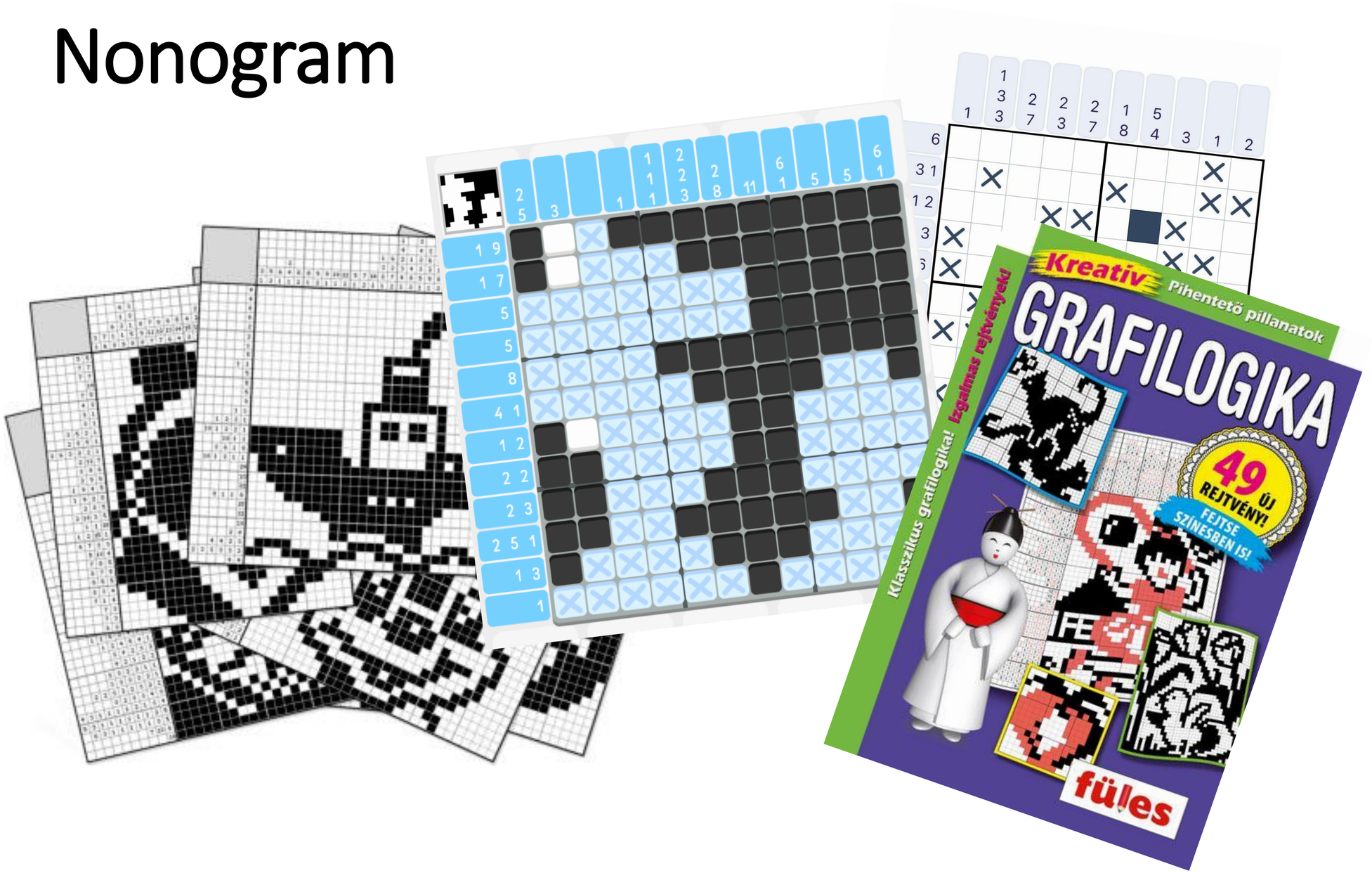




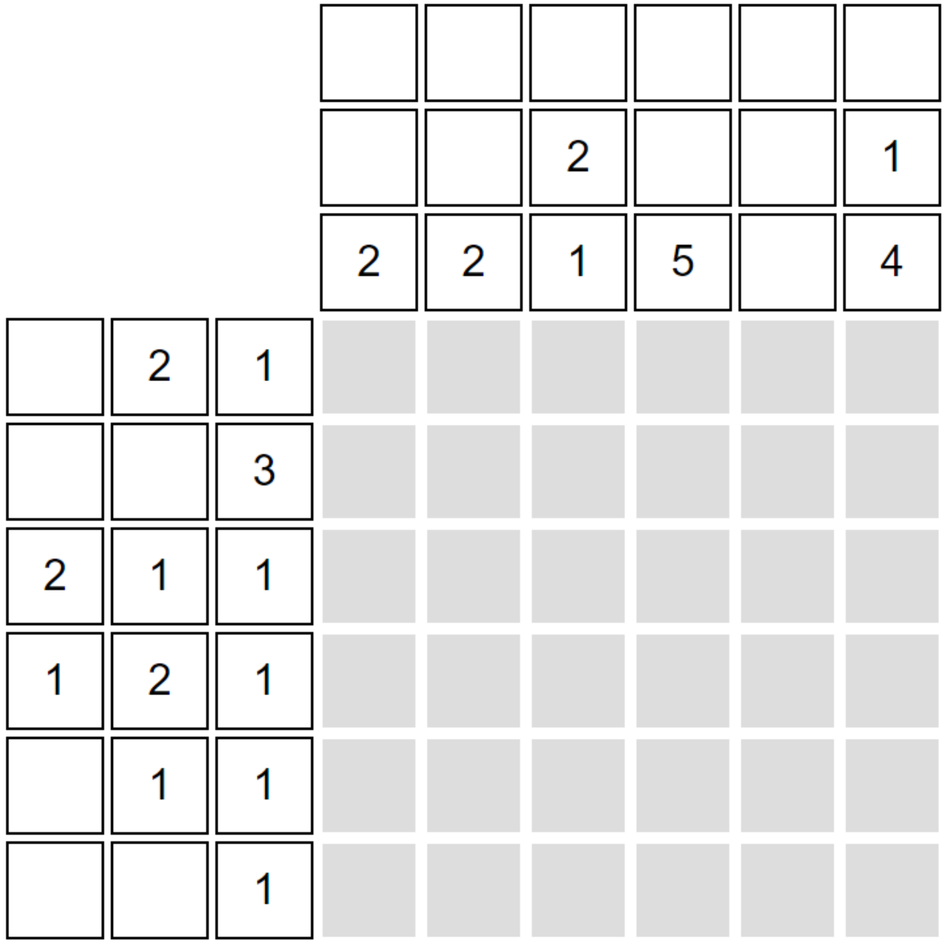
Nonogram



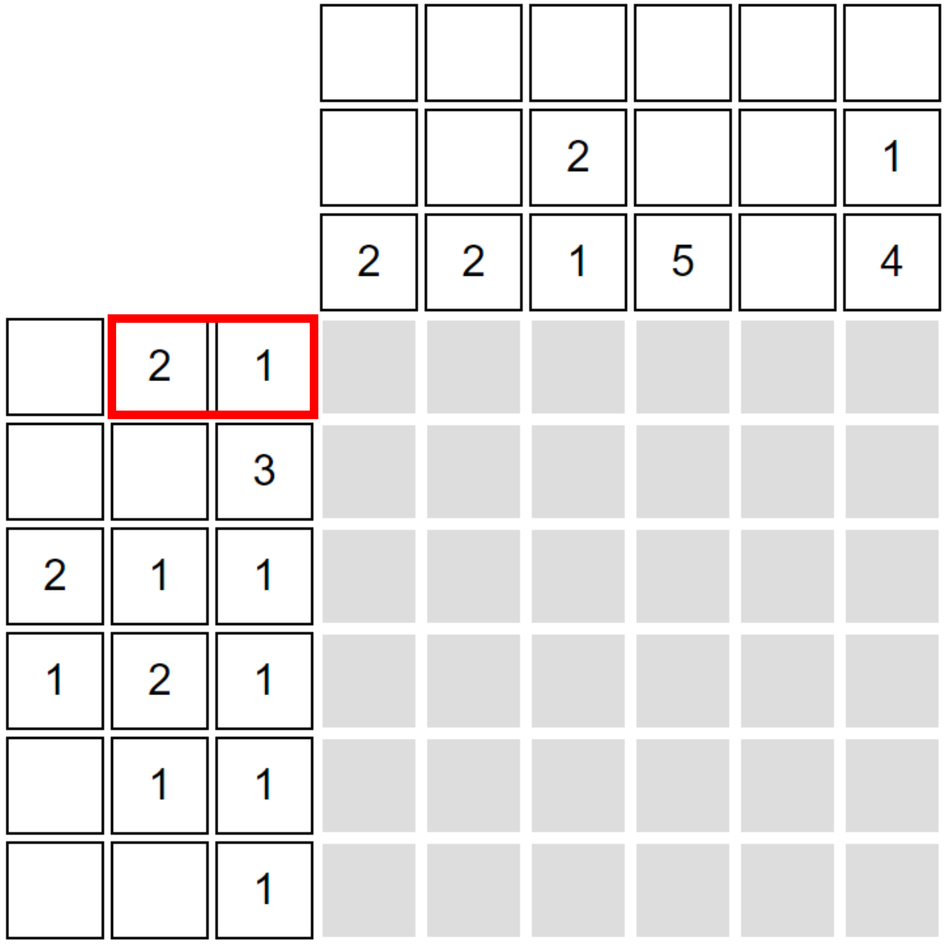
Nonogram


Kreatív
Pihentető pillanatok
GRAFILOGIKA
49 ÚJ REJTVÉNY!
FEJTSE SZÍNESBEN IS!
Klasszikus grafilogika! Izgalmas rejtvények!
füles

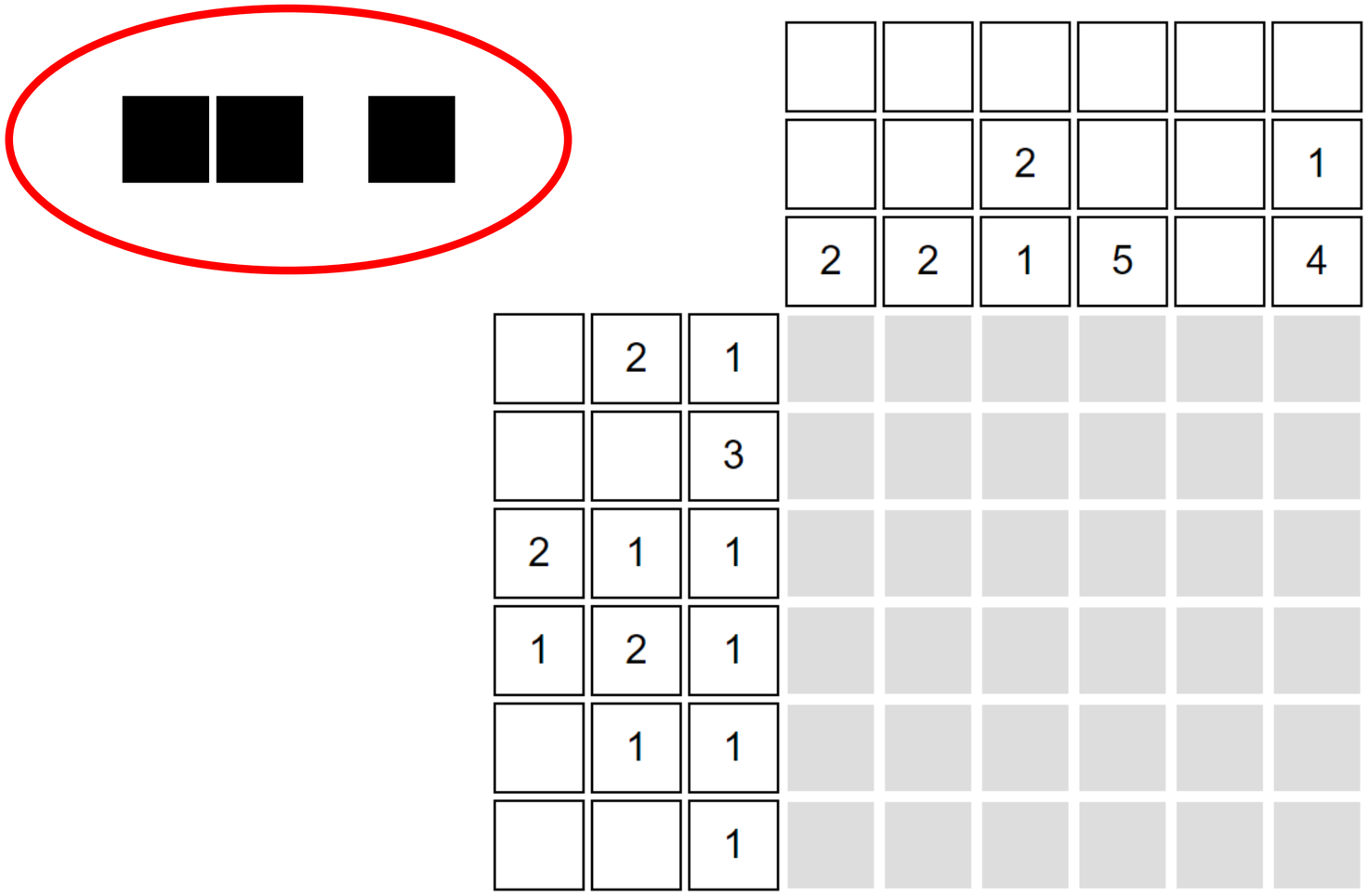
A nonogramok szabályai



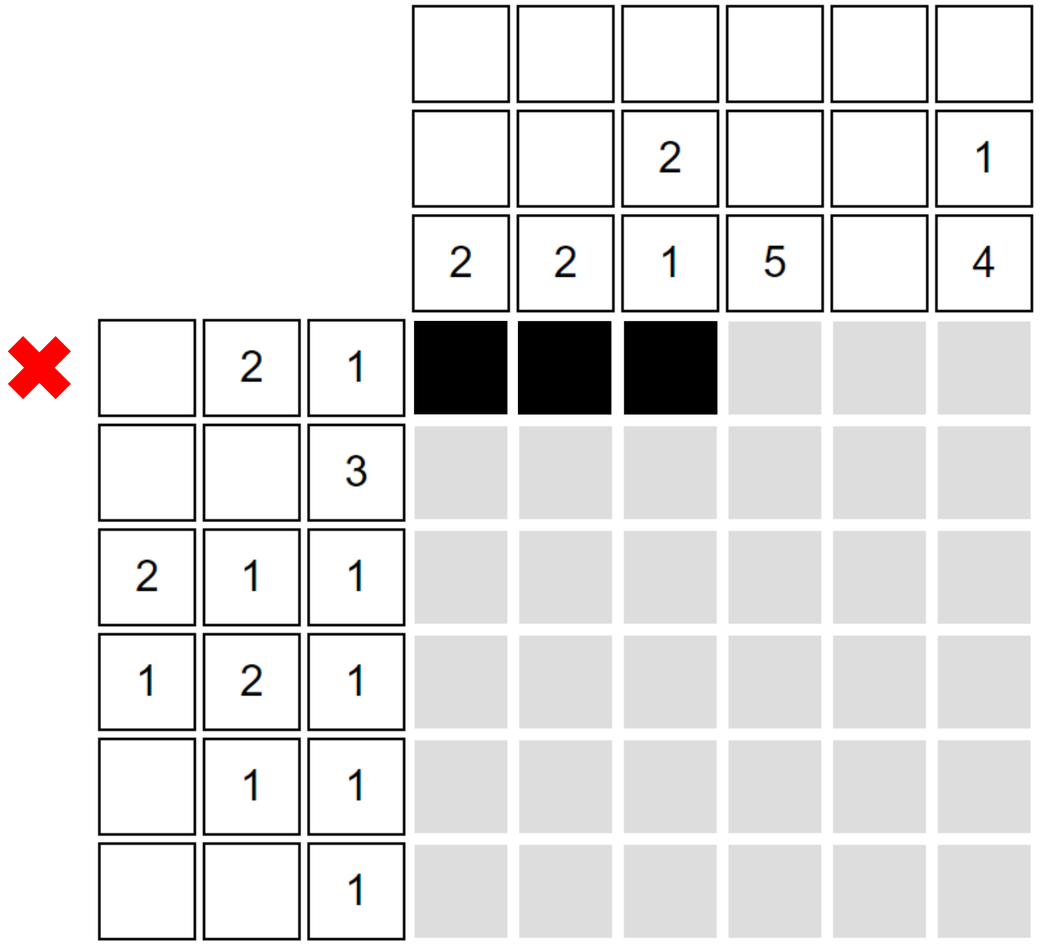
A nonogramok szabályai



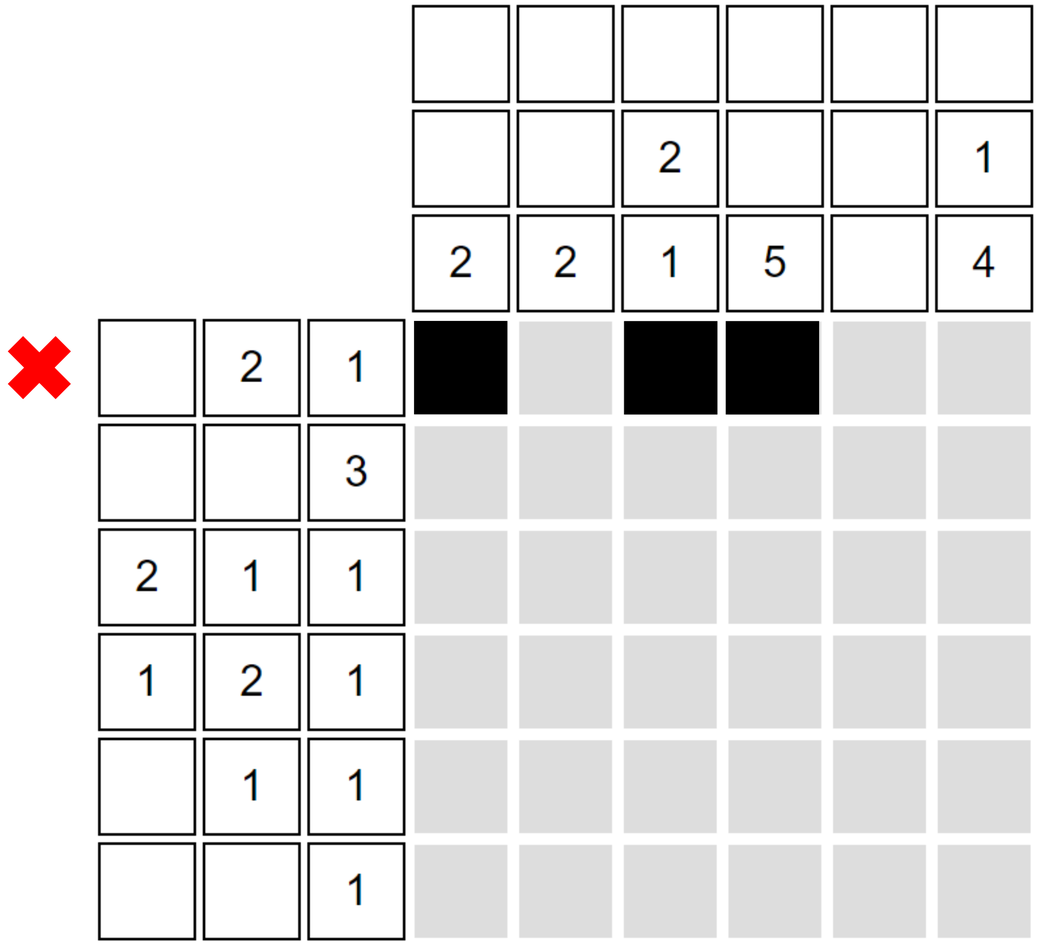
A nonogramok szabályai



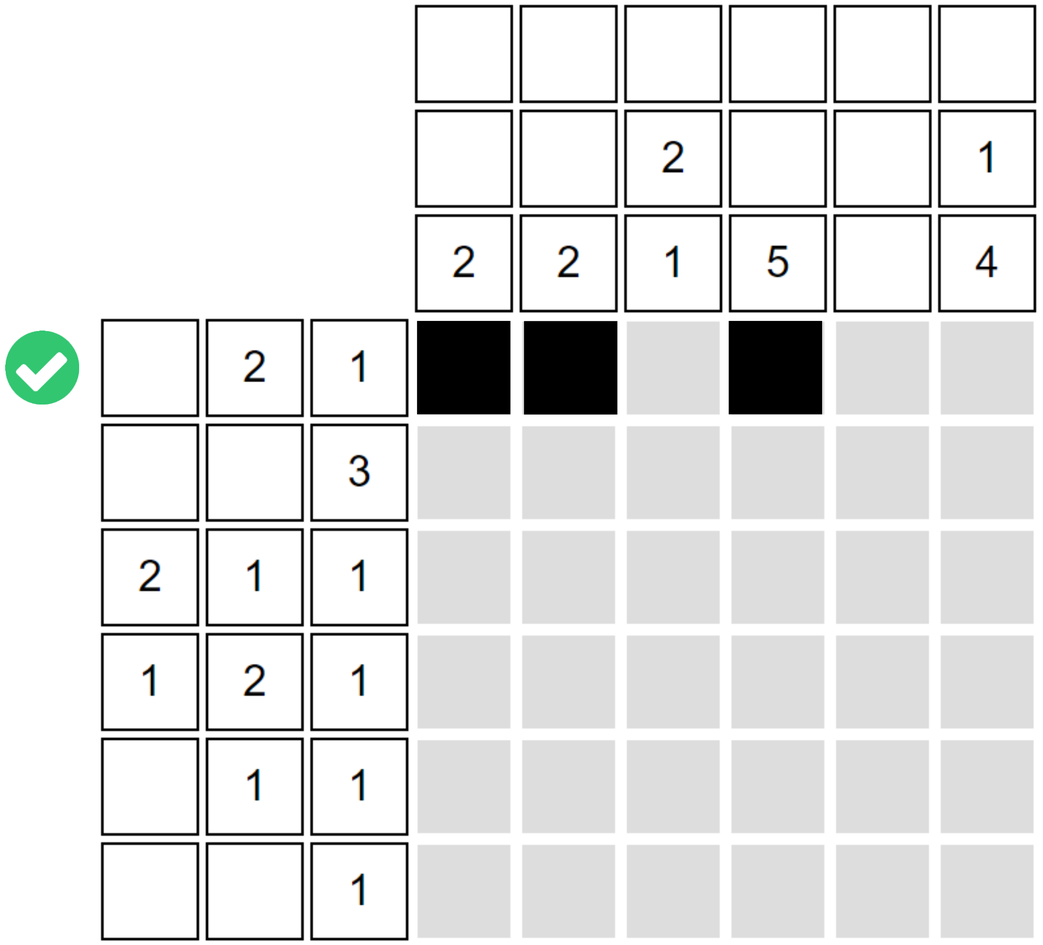
A nonogramok szabályai



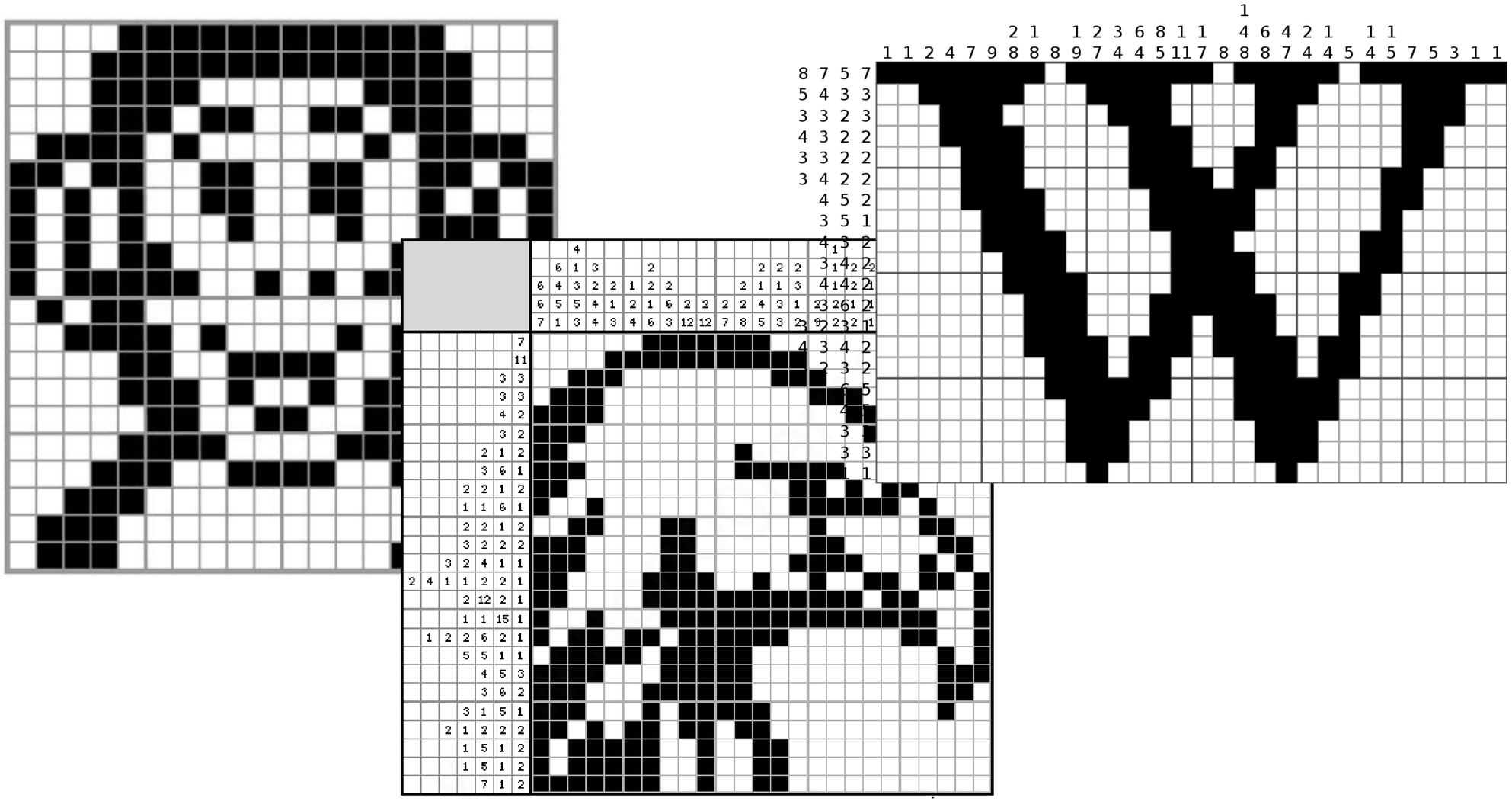
A nonogramok szabályai



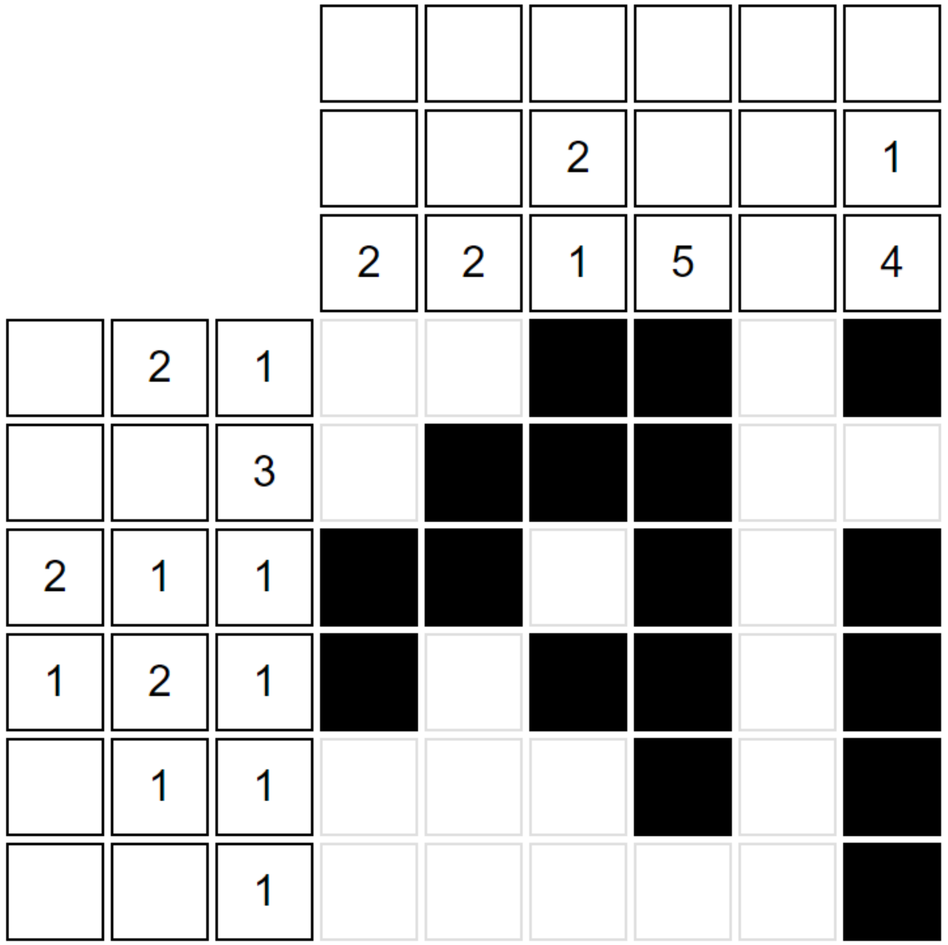
A nonogramok szabályai



A nonogramok szabályai



A nonogramok szabályai



Irodalom

Solving Nonograms by combining relaxations

K.J. Batenburg[a,*], W.A. Kosters[b]

[a]Vision Lab, Department of Physics, University of Antwerp, Universiteitsplein 1, B-2610 Wilrijk, Belgium
[b]Leiden Institute of Advance

A comparison of a genetic algorithm and a depth first search algorithm applied to Japanese nonograms

Wouter Wiggers
Faculty of EECMS, University of Twente
w.a.wigg

Constructing Simple Nonograms of Varying Difficulty

K. Joost Batenburg*,[1], Sjoerd Henstra[2], Walter A. Kosters[2], and Willem Jan Palenstijn[1]

[1]Vision Lab, Department of Physics, University of Antwerp, Belgium
[2]Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands

Improved Automatic Generation of Curved Nonograms

Mees van de Kerkhof

Master Thesis ICA-3822443
Computer Science
Utrecht University

Komplexitás

Corpus ID: 3040166

NP-completeness Results for NONOGRAM via Parsimonious Reductions

N. Ueda, T. Nagao • Published 1996 • Mathematics

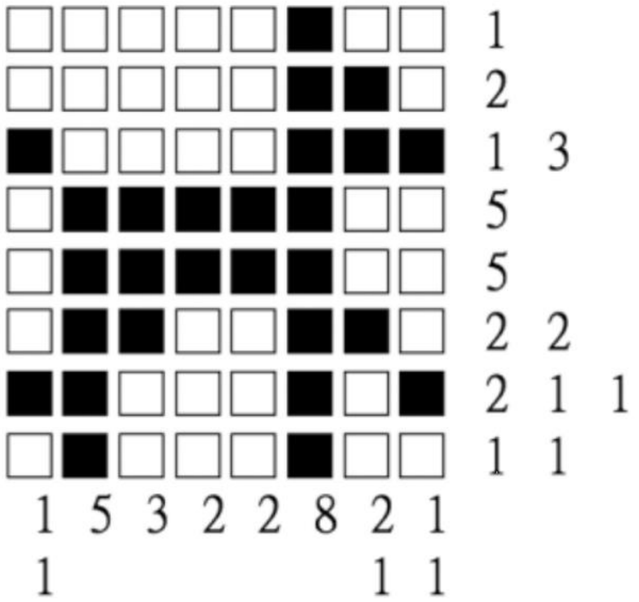
We introduce a new class of NP problems called ANOTHER SOLUTION PROBLEMs. For a given NP problem X, ANOTHER SOLUTION PROBLEM for X (ASP for X) is to ask, for a given instance I for X and its solution, whether there is another solution for I. The di culty of ASP for X may not be immediate from the di culty of X. For example, for some NP-complete problems such as 3SAT or 3DM, it is easy to show that their ASPs are NP-complete; on the other hand, ASP for, e.g., VERTEX COLORING is trivially in P... CONTINUE READING

Save to Library Create Alert Cite Launch Research Feed

Algoritmusok

An Efficient Approach to Solving Nonograms

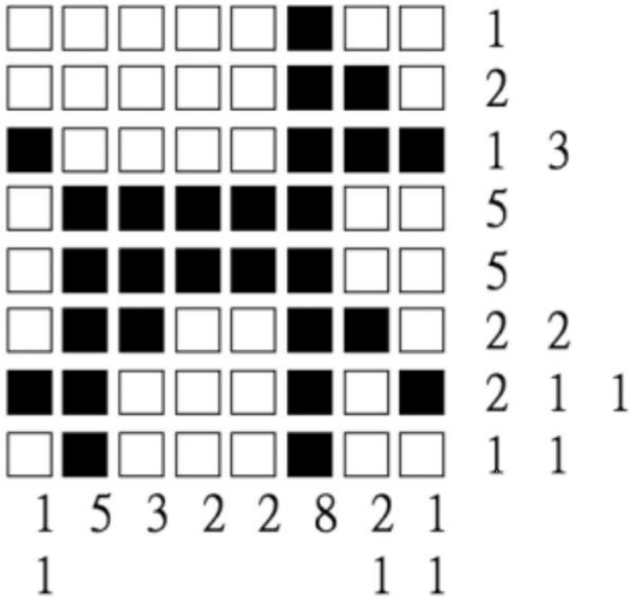
I.-Chen Wu, *Member, IEEE*, Der-Johng Sun, Lung-Ping Chen, Kan-Yueh Chen, Ching-Hua Kuo, Hao-Hua Kang, and Hung-Hsuan Lin



Algoritmusok

An Efficient Approach to Solving Nonograms

I.-Chen Wu, *Member, IEEE*, Der-Johng Sun, Lung-Ping Chen, Kan-Yueh Chen, Ching-Hua Kuo, Hao-Hua Kang, and Hung-Hsuan Lin

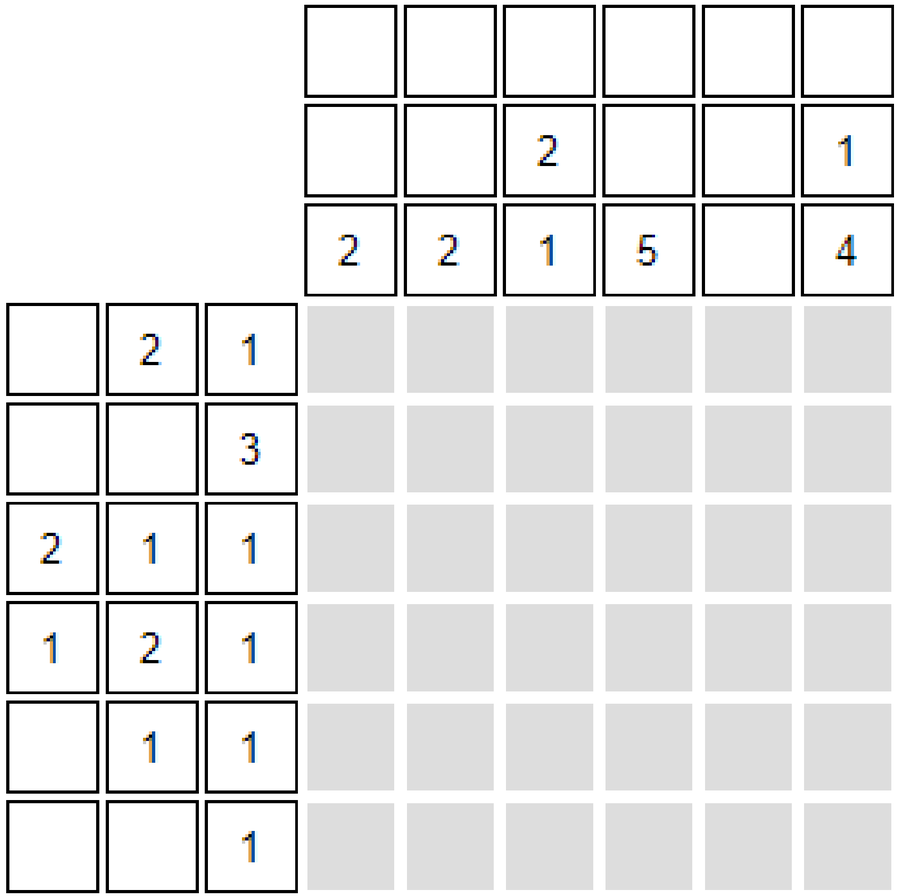


$$O(I * k)$$

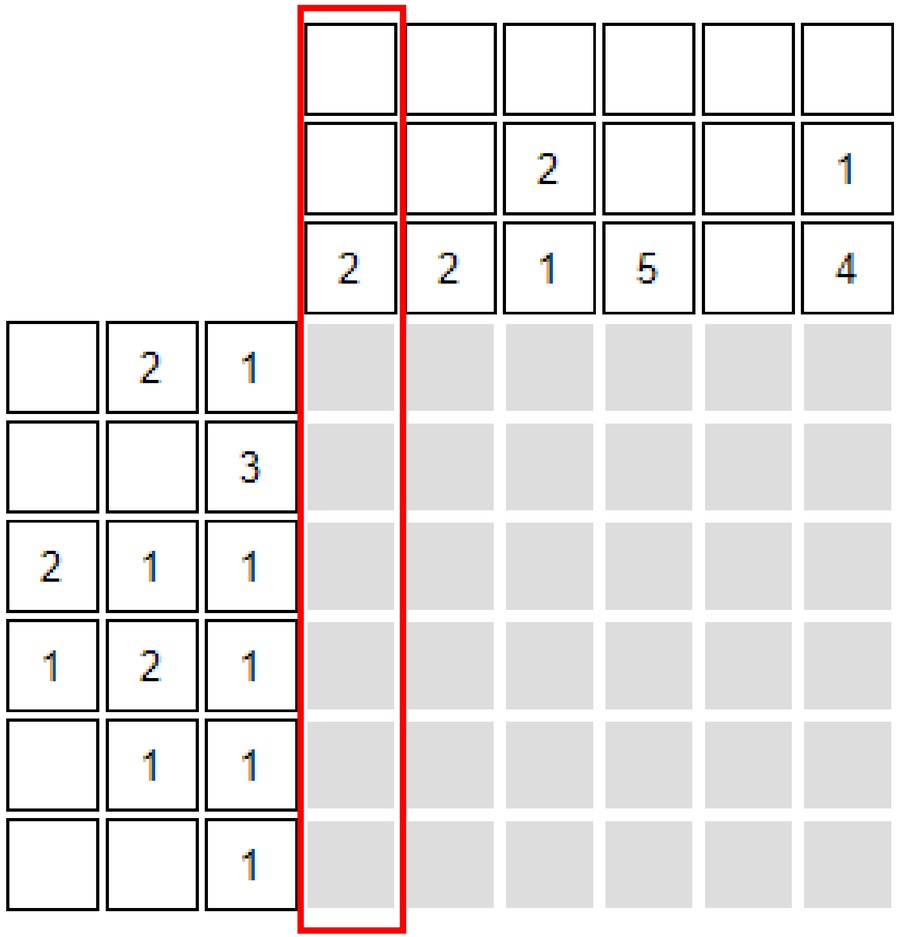
I: *grid size*

k: *average number of integers in one constraint*

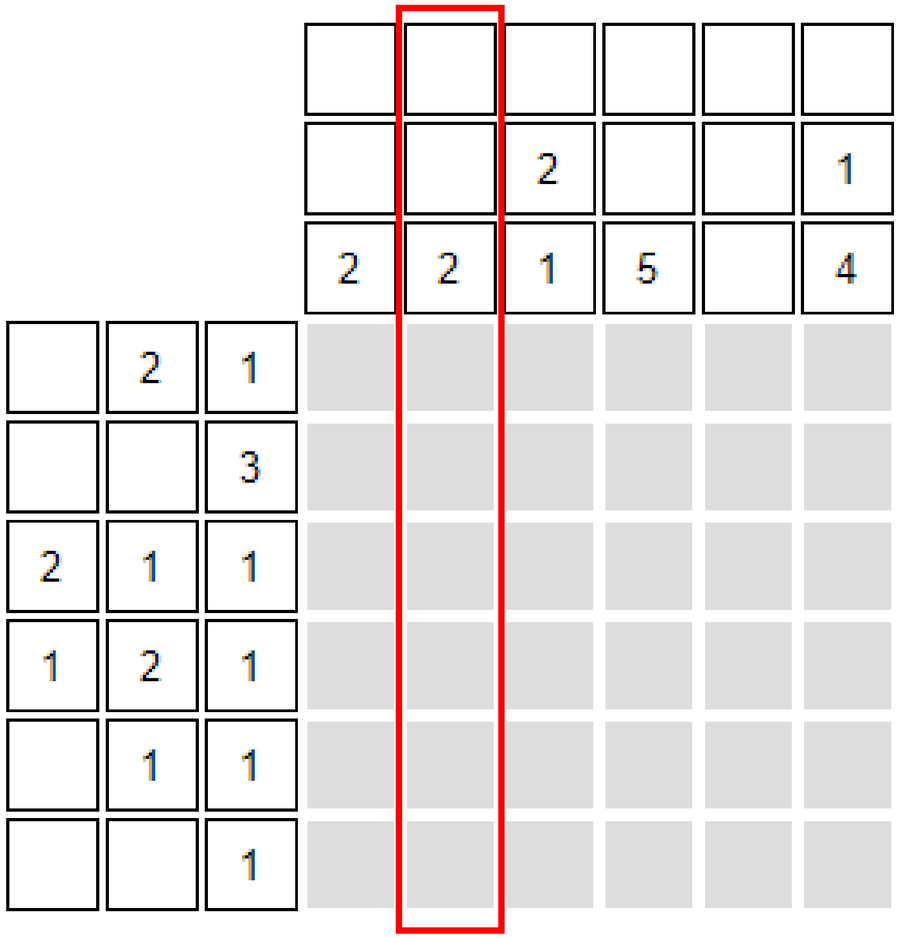
Egyszerű algoritmus



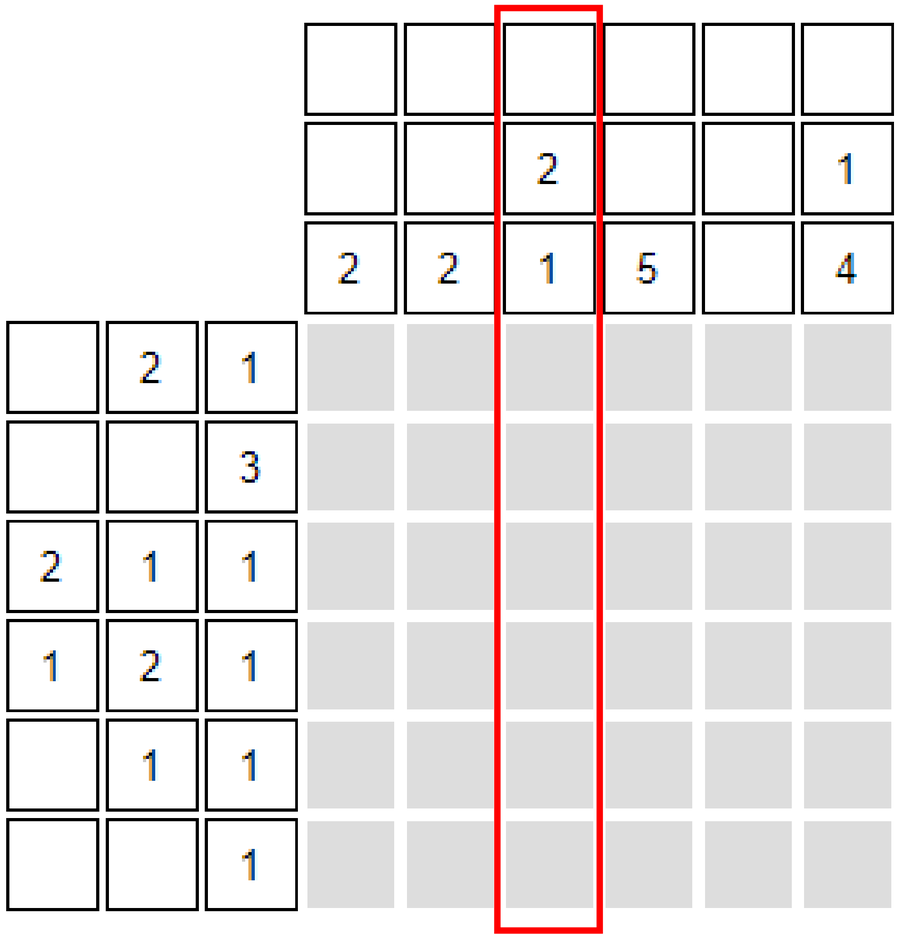
Egyszerű algoritmus



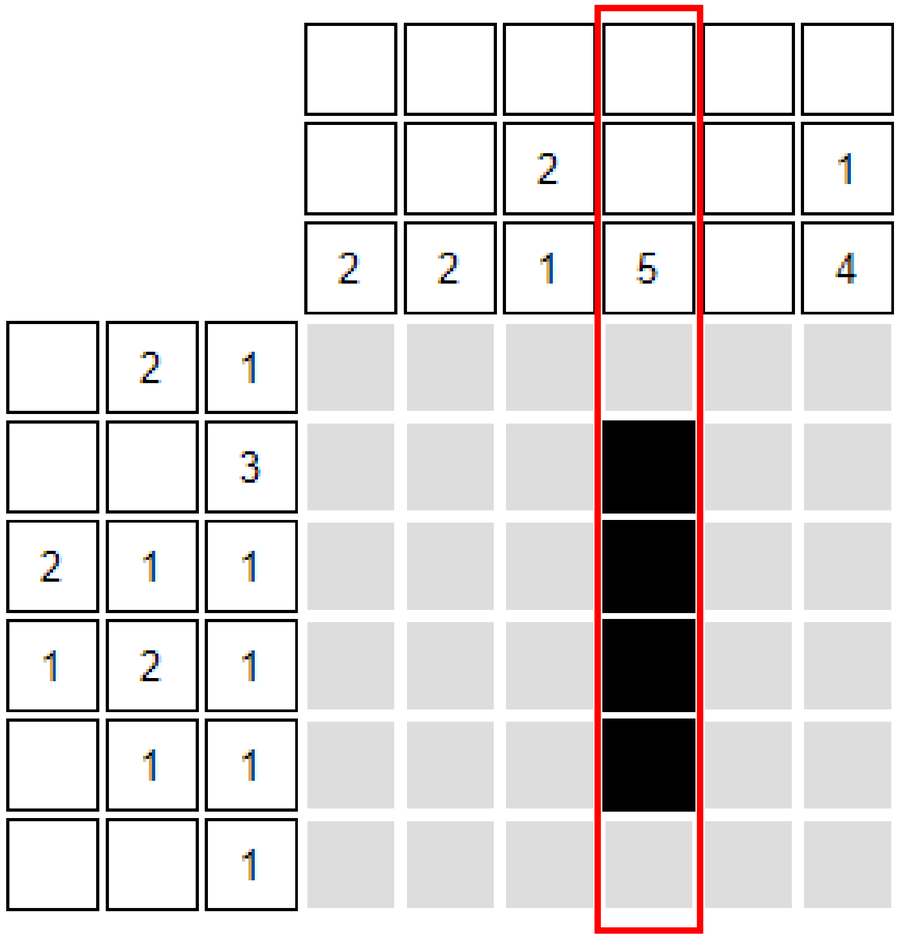
Egyszerű algoritmus



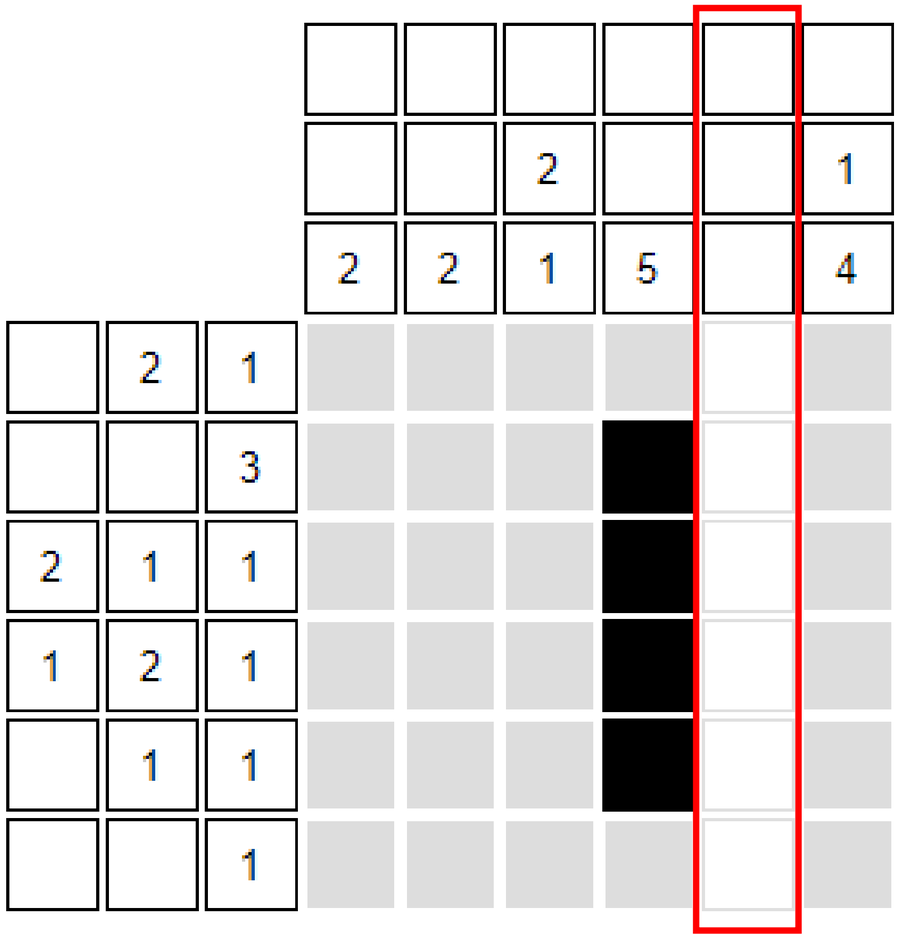
Egyszerű algoritmus



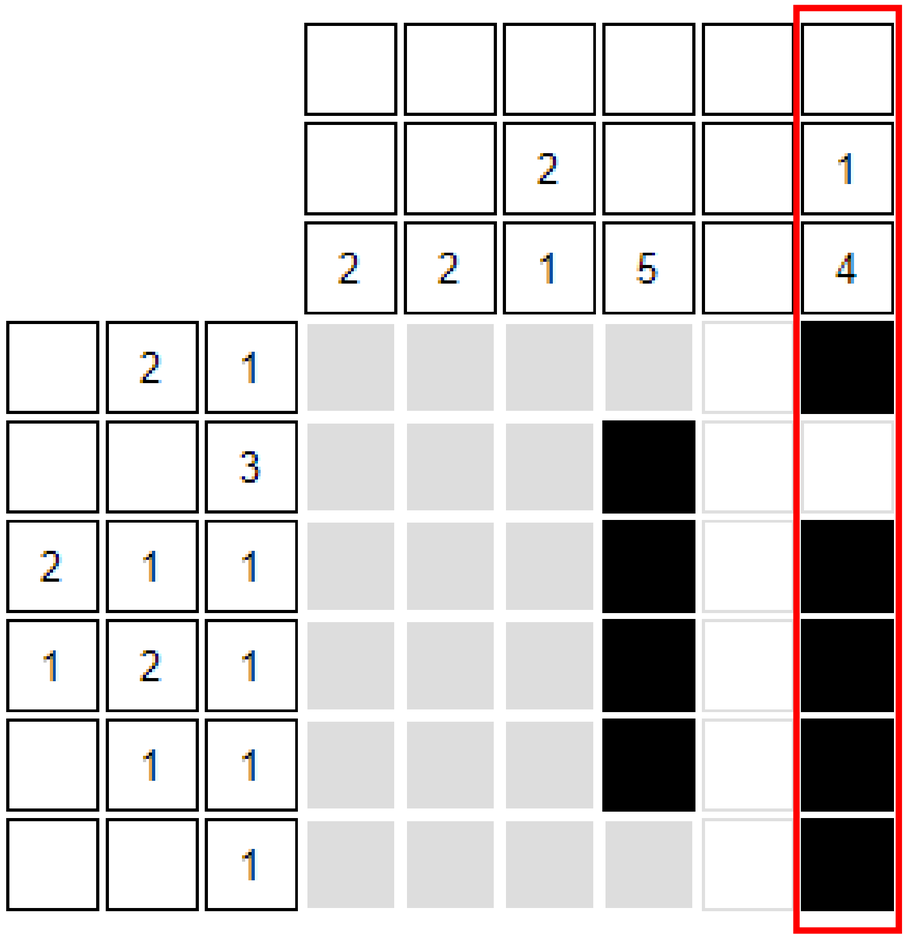
Egyszerű algoritmus



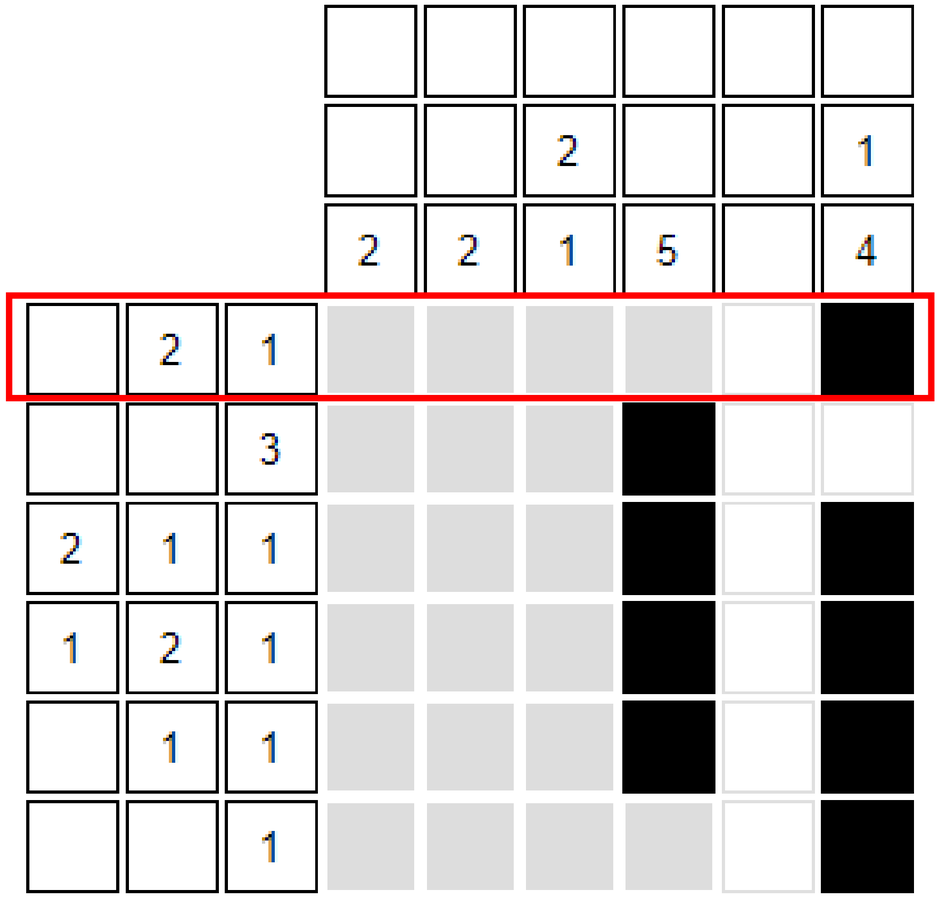
Egyszerű algoritmus



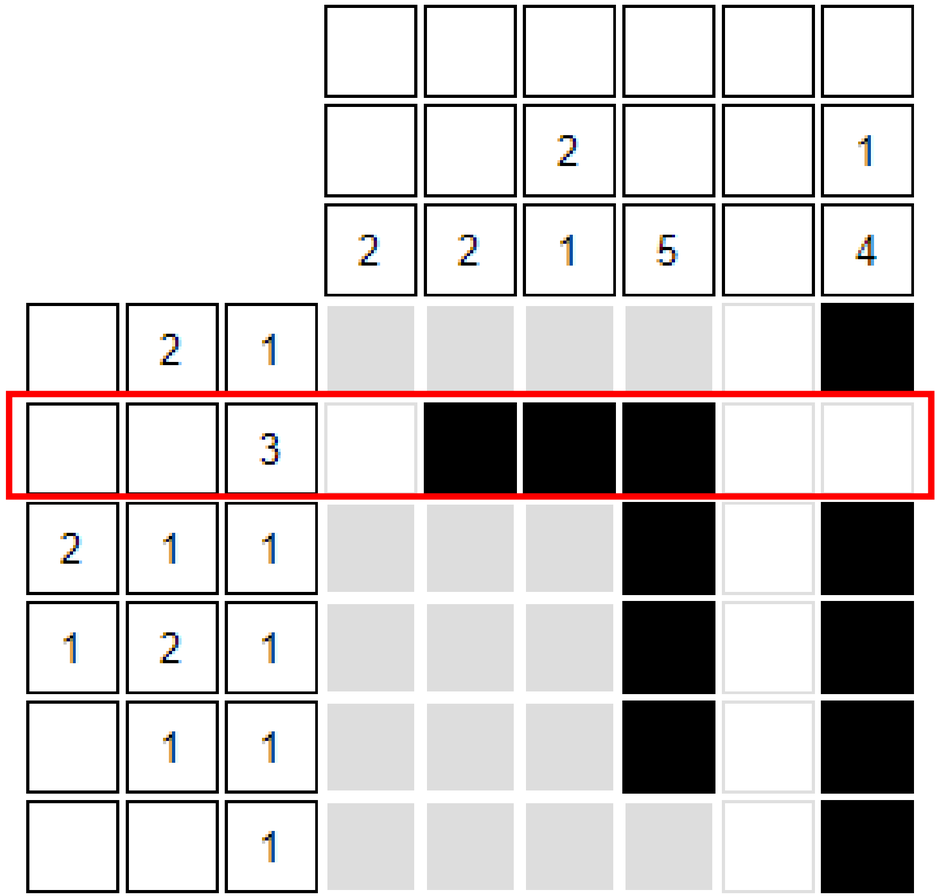
Egyszerű algoritmus



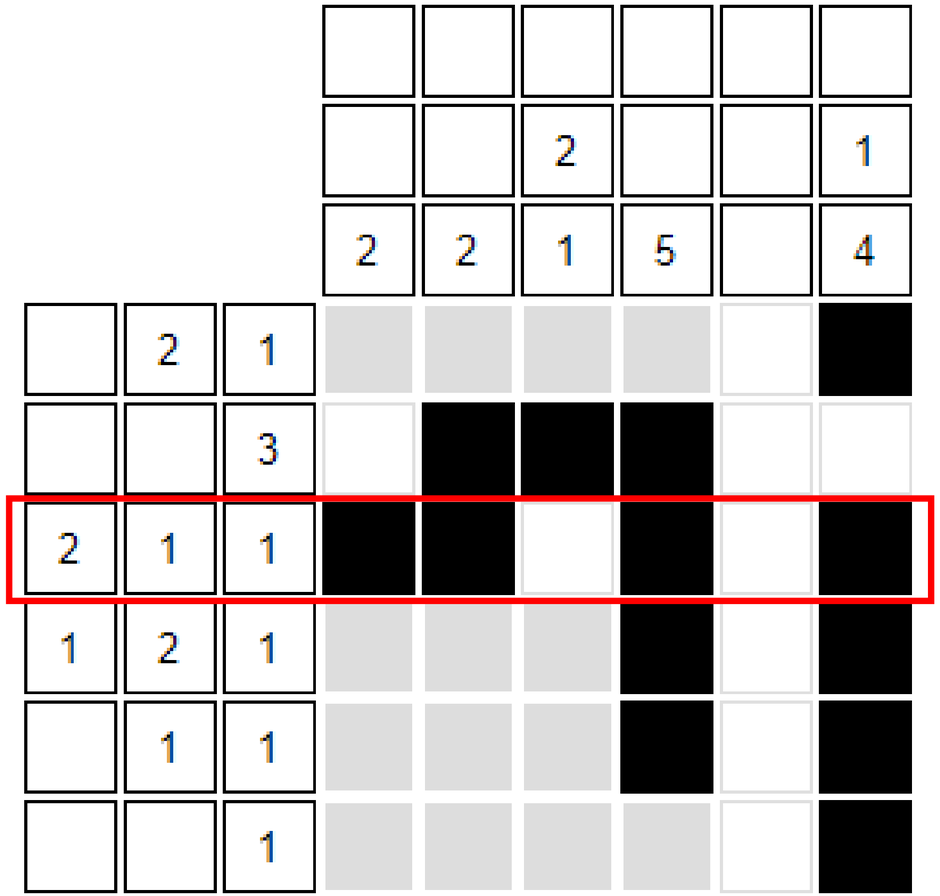
Egyszerű algoritmus



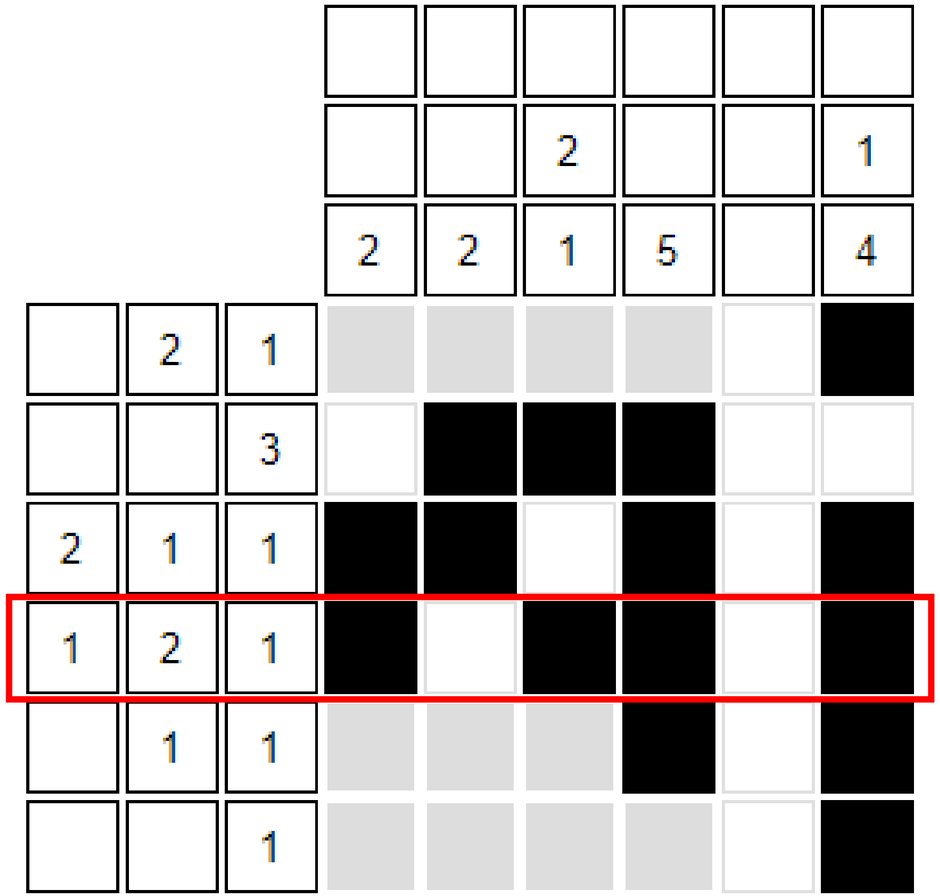
Egyszerű algoritmus



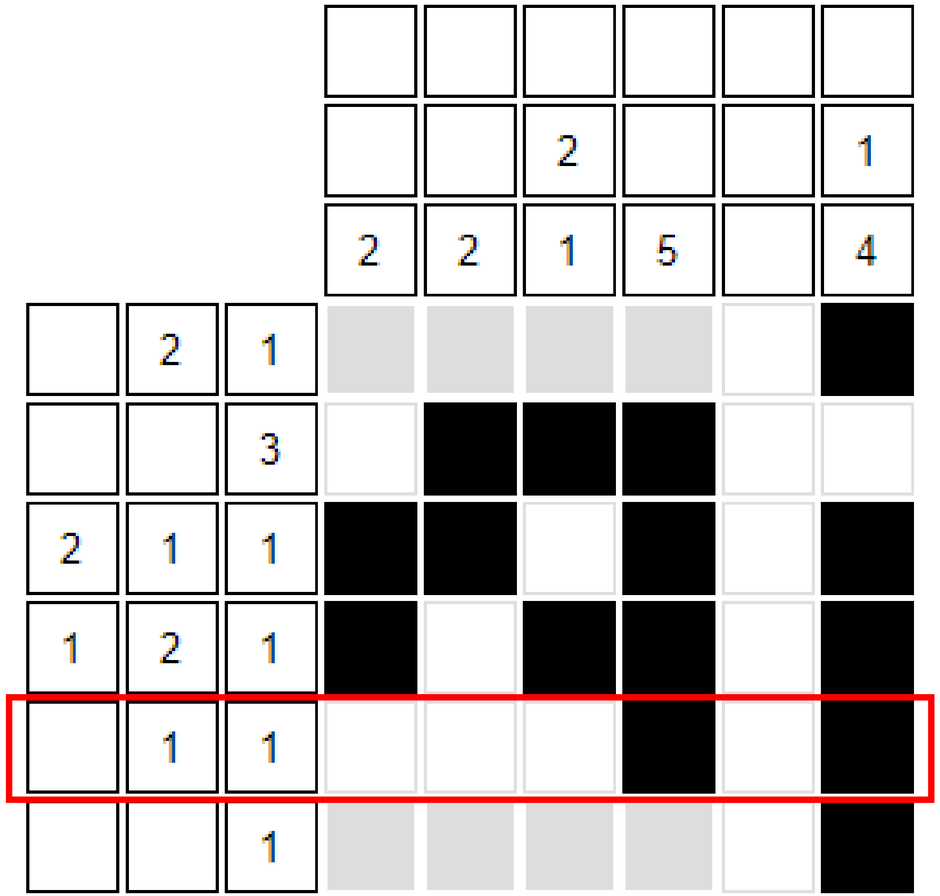
Egyszerű algoritmus



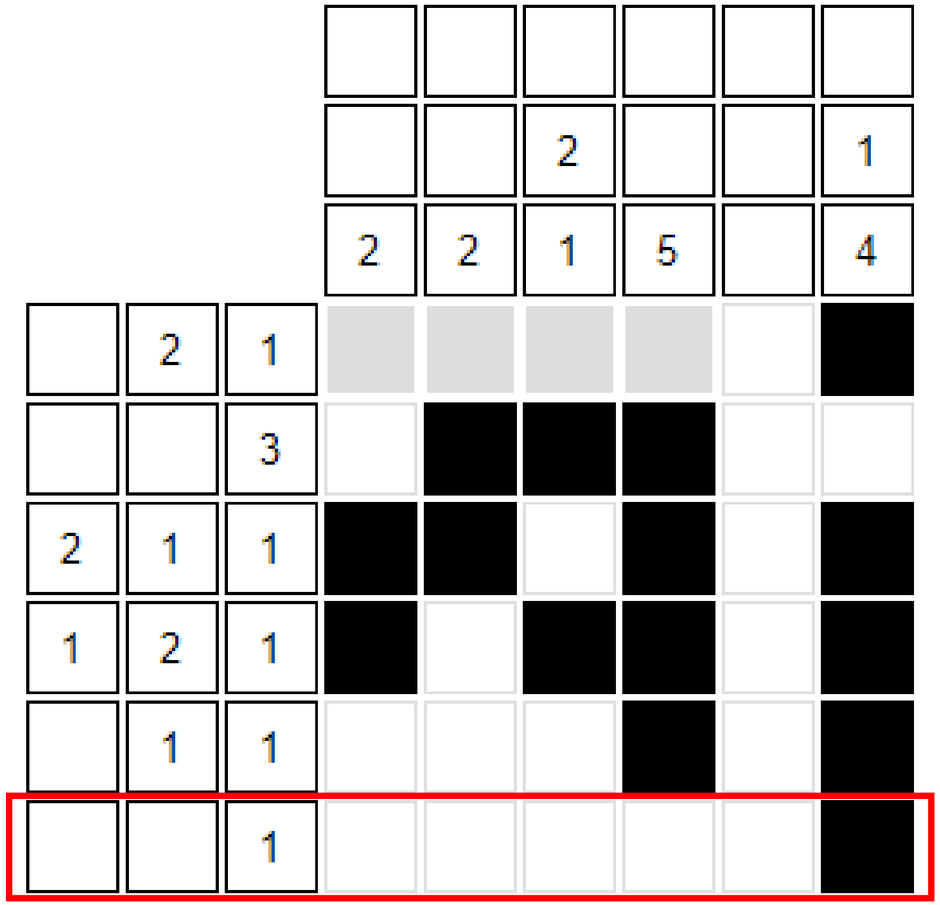
Egyszerű algoritmus



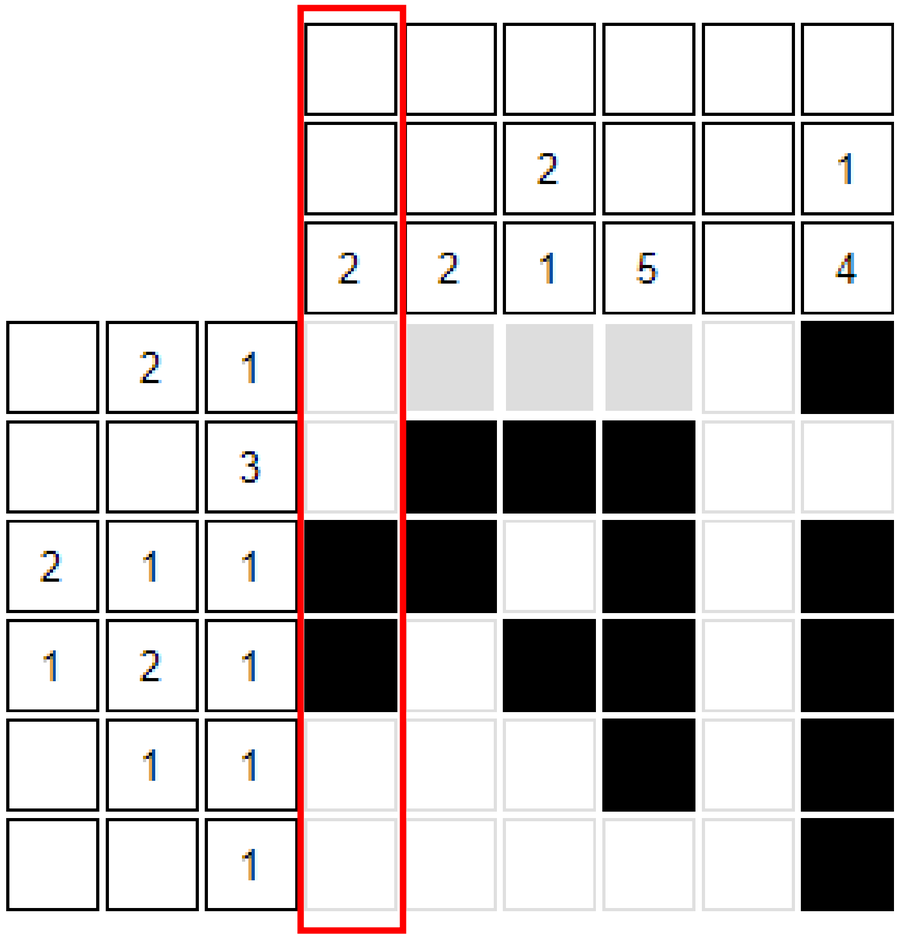
Egyszerű algoritmus



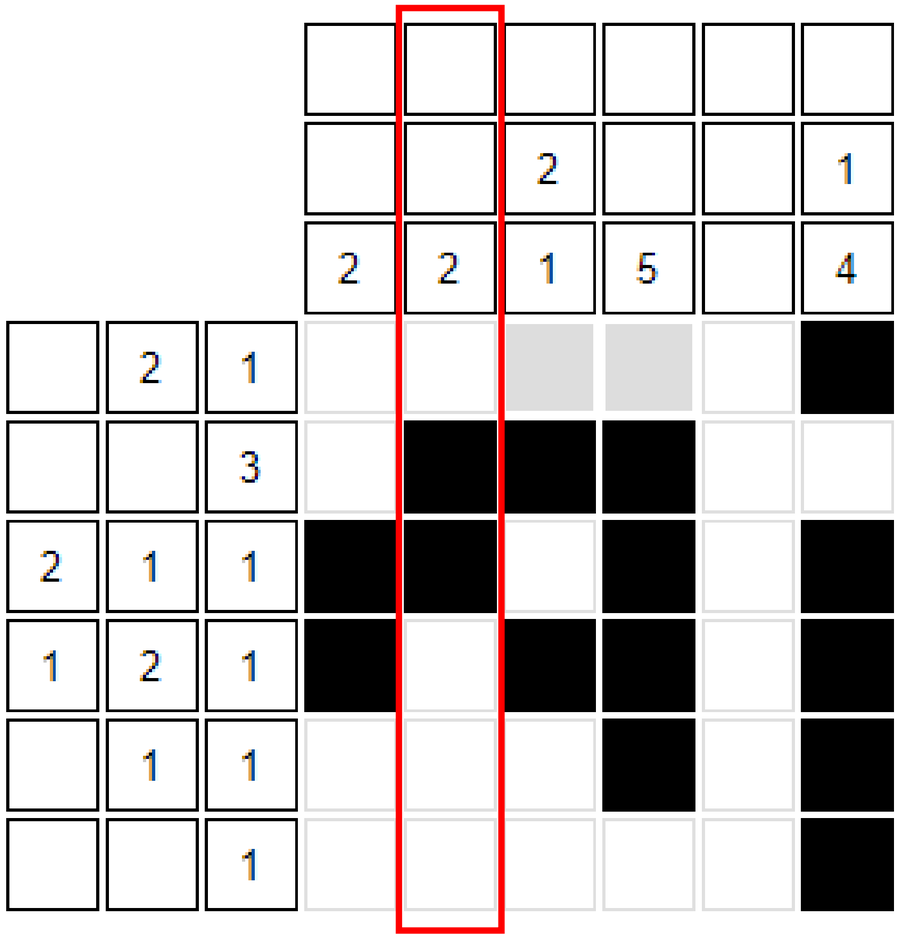
Egyszerű algoritmus



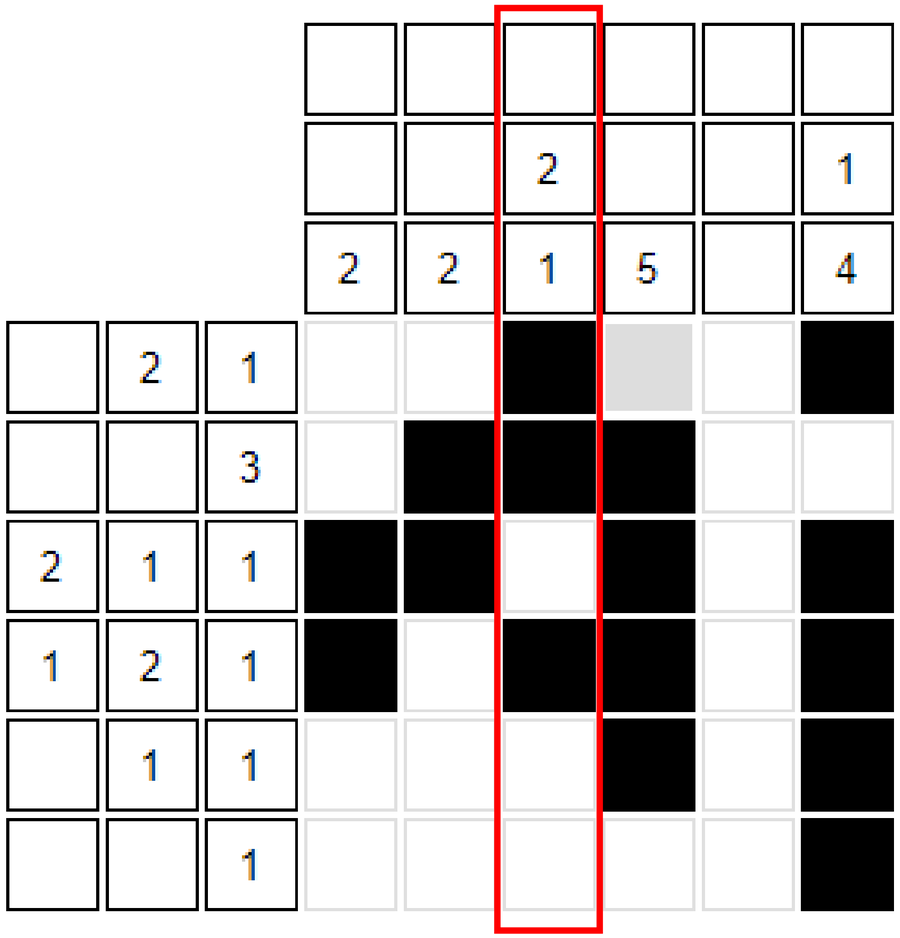
Egyszerű algoritmus



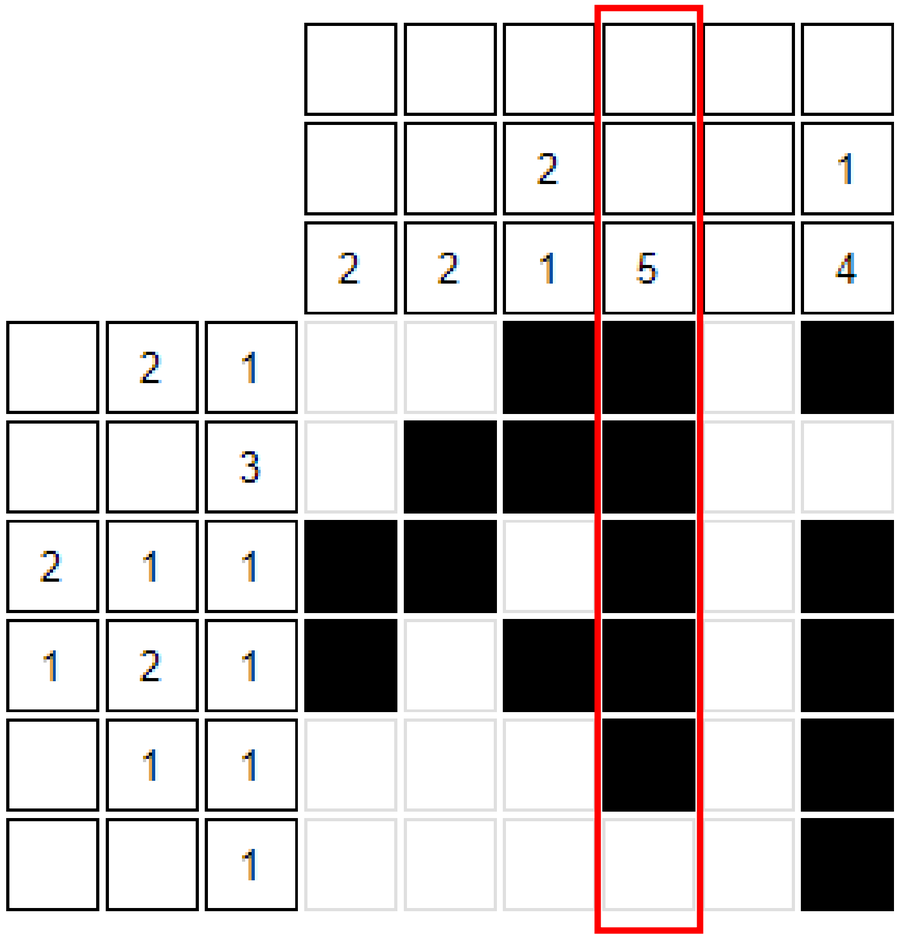
Egyszerű algoritmus



Egyszerű algoritmus



Egyszerű algoritmus



Nonogramok és gépi tanulás

Solving nonogram puzzles by reinforcement learning

Frédéric Dandurand (frederic.dandurand@gmail.com)
Department of Psychology, Université de Montréal, 90 ave. Vincent-d'Indy
Montréal, QC H2V 2S9 Canada

Denis Cousineau (denis.cousineau@uottawa.ca)
École de psychologie, Pavillon Vanier, Université d'Ottawa
136 Jean Jacques Lussier, Ottawa, Ontario, K1N 6N5, Canada

Thomas R. Shultz (thomas.shultz@mcgill.ca)
Department of Psychology and School of Computer Science, McGill University, 1205 Penfield Avenue
Montreal, QC H3A 1B1 Canada

Nonogramok és gépi tanulás

$$(n+n)!\ combinations\ of\ sequences$$

$$n = 5 \rightarrow\ 3.6\ million\ combinations$$

$$n = 6 \rightarrow\ 497\ million\ combinations$$

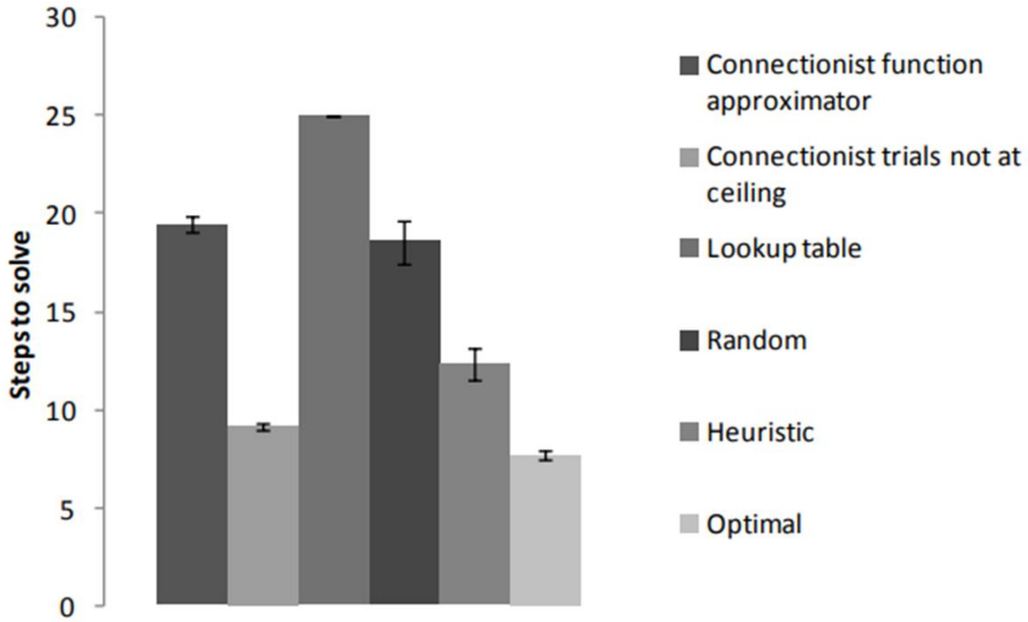
$$n = 7 \rightarrow\ 8.7 * 10^{10}\ million\ combinations$$

Nonogramok és gépi tanulás

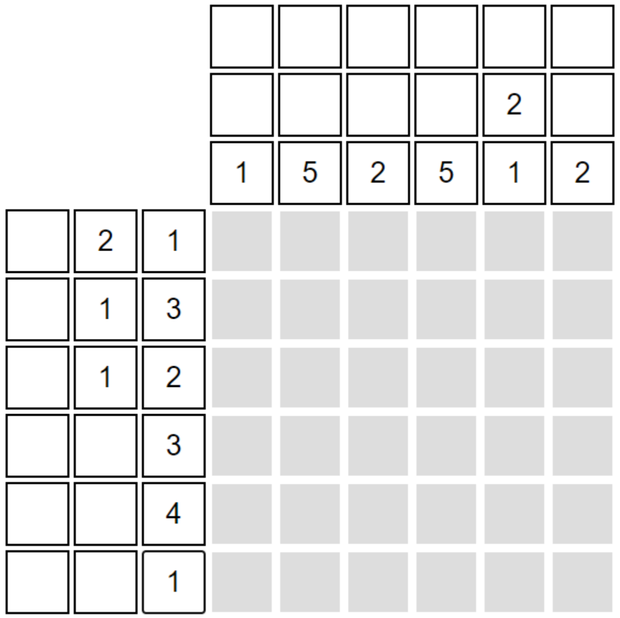
State: of the cells on the corresponding line

Action: choice of the next line to solve

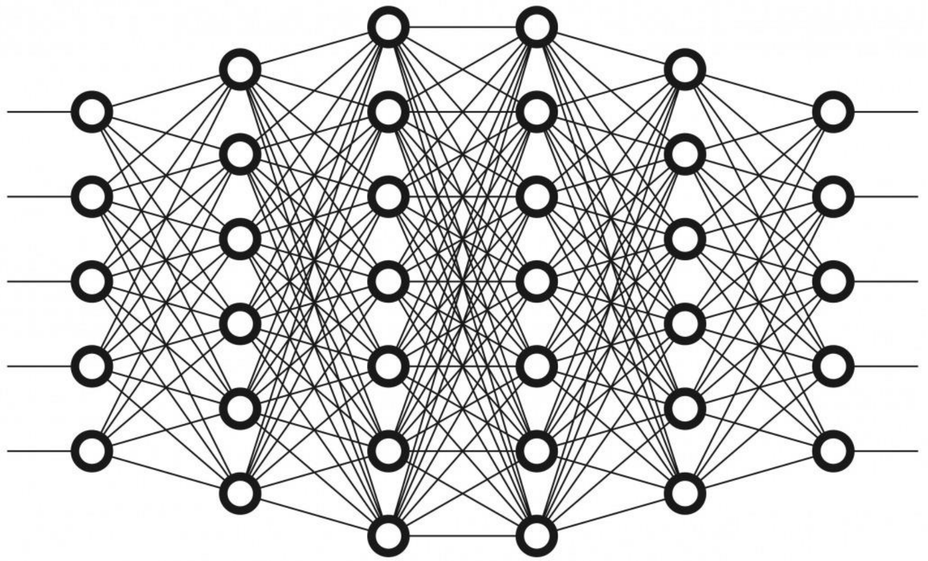
Reward: proportion of the currently unknown cells that are determined as filled or empty

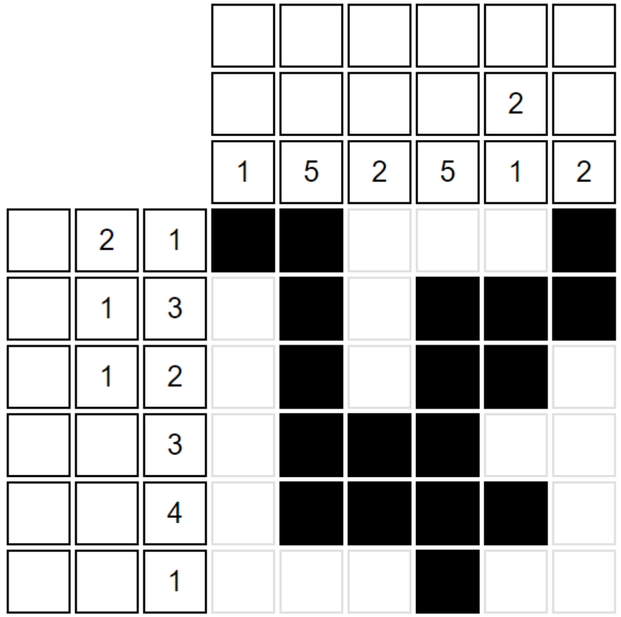


Célkitűzés

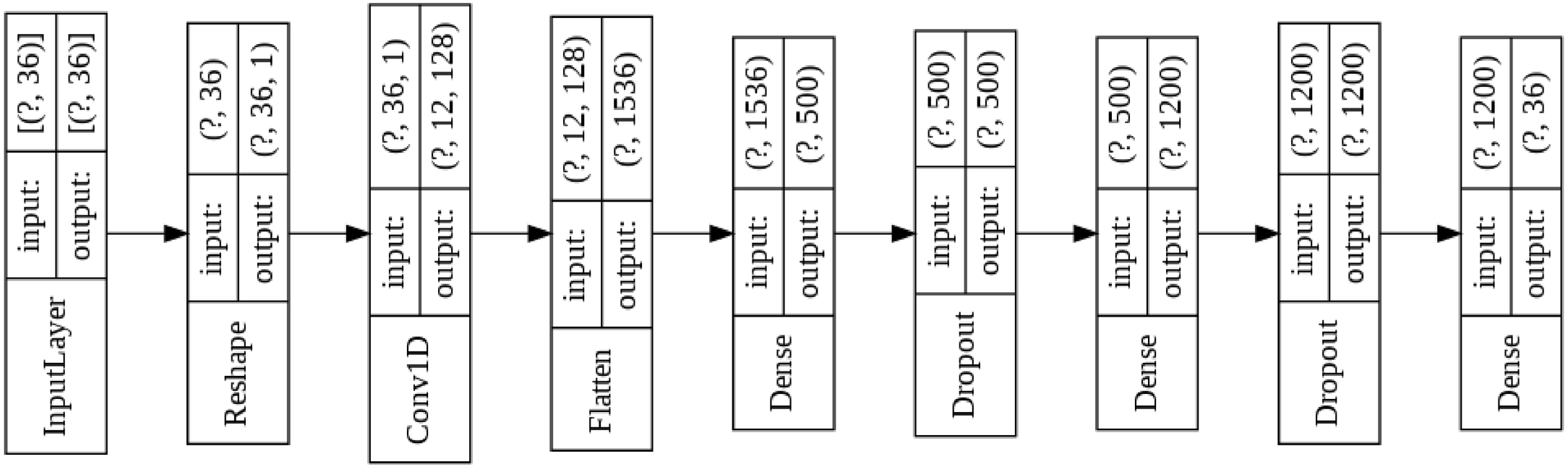


?

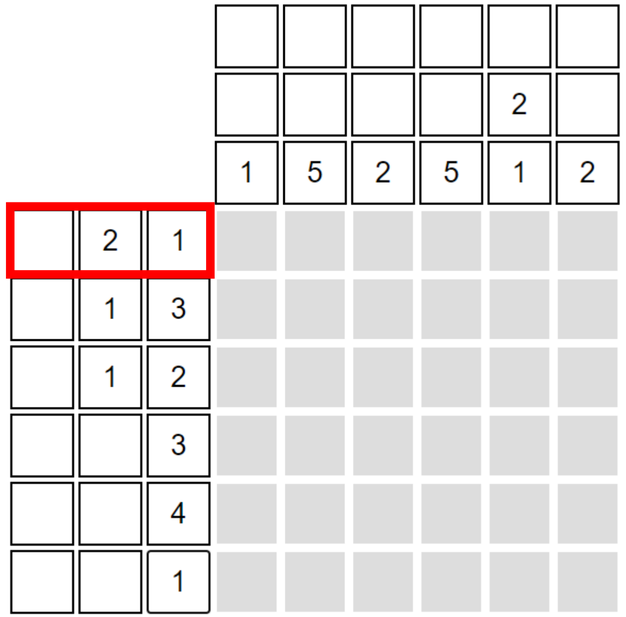


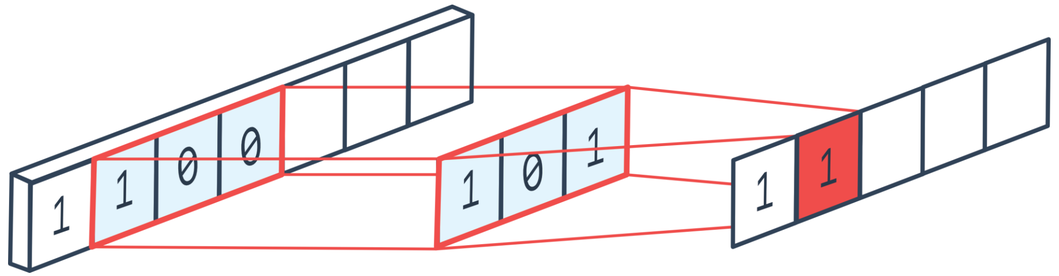


Rendszerterv



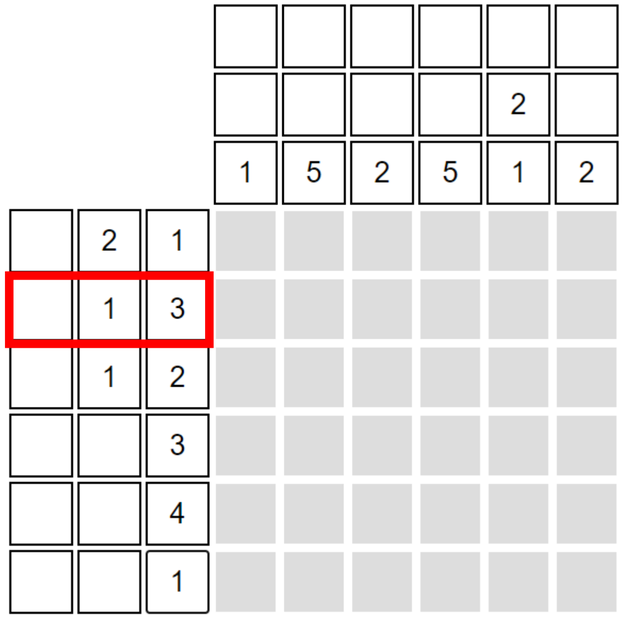
Konvolúciós réteg

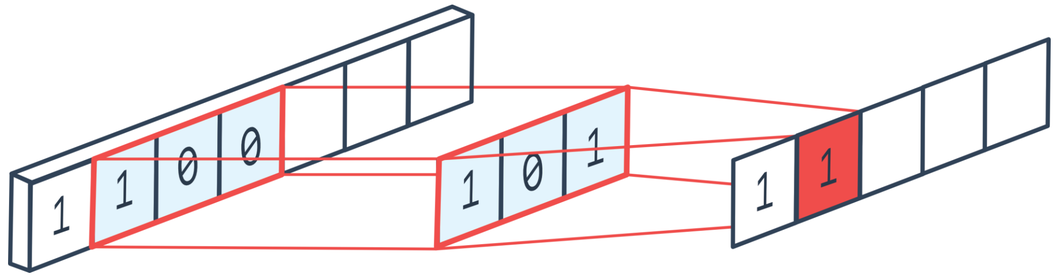




[0 2 1 0 1 3 0 1 2 0 0 3 0 0 4 0 0 1 0 0 1 0 …]

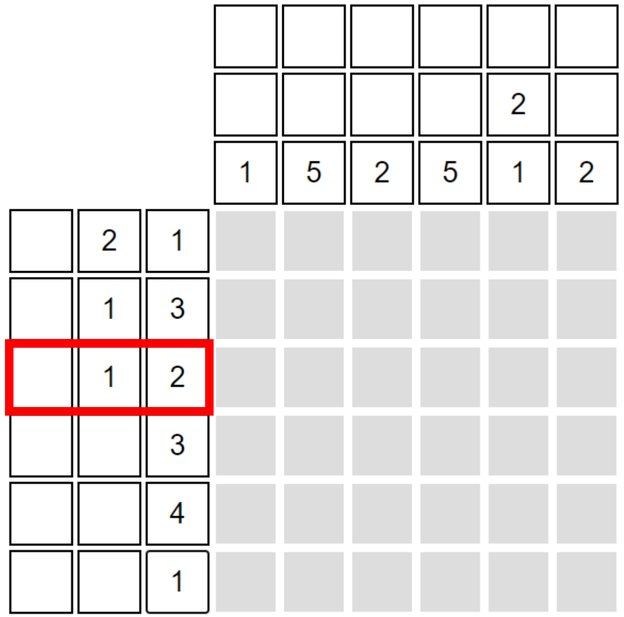
Konvolúciós réteg

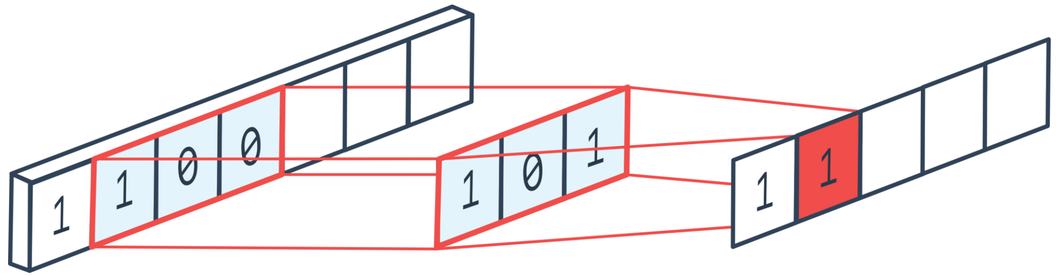




[0 2 1 0 1 3 0 1 2 0 0 3 0 0 4 0 0 1 0 0 1 0 …]

Konvolúciós réteg



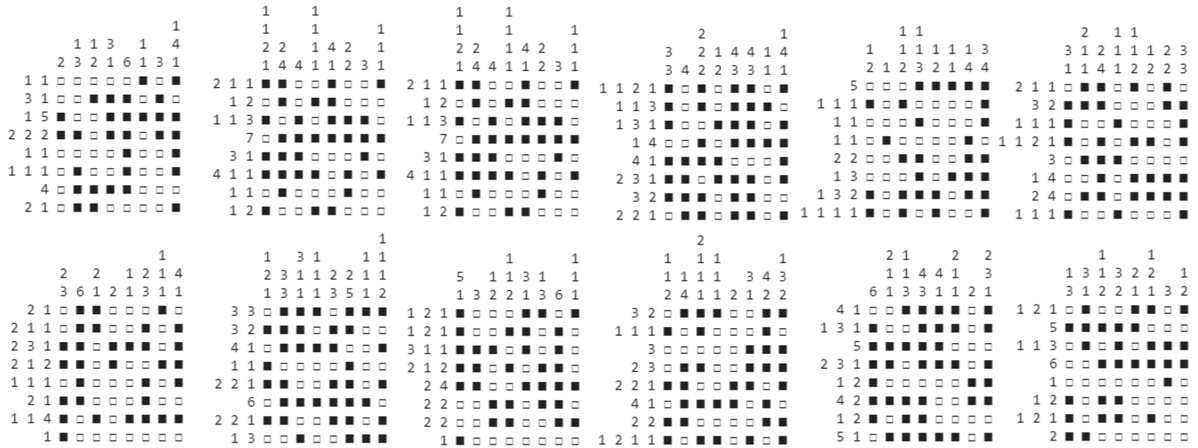


[0 2 1 0 1 3 0 1 2 0 0 3 0 0 4 0 0 1 0 0 1 0 …]

Adatgenerálás

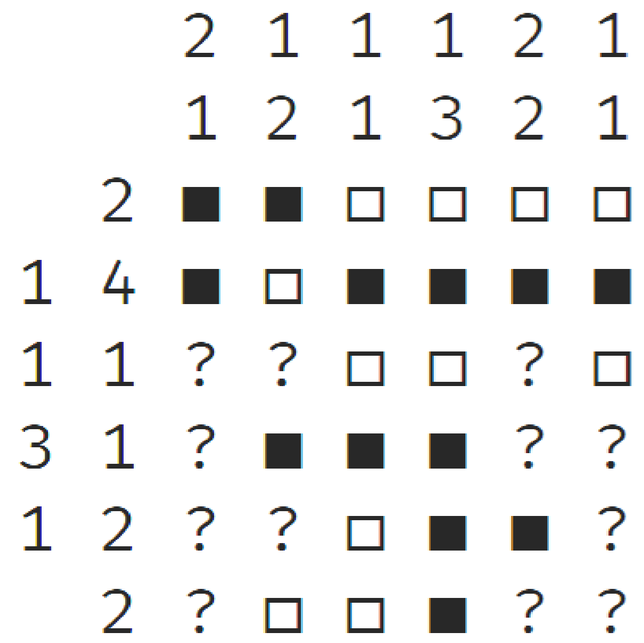
Generating data

```
datagen.py

-s, --samples Number of generated samples

-r, --rows Number of rows in a nonogram

-c, --columns Number of columns in a nonogram

-t, --train Train split [0.0, 1.0]

-v, --valid Valid split [0.0, 1.0]

-o, --output Output folder
```

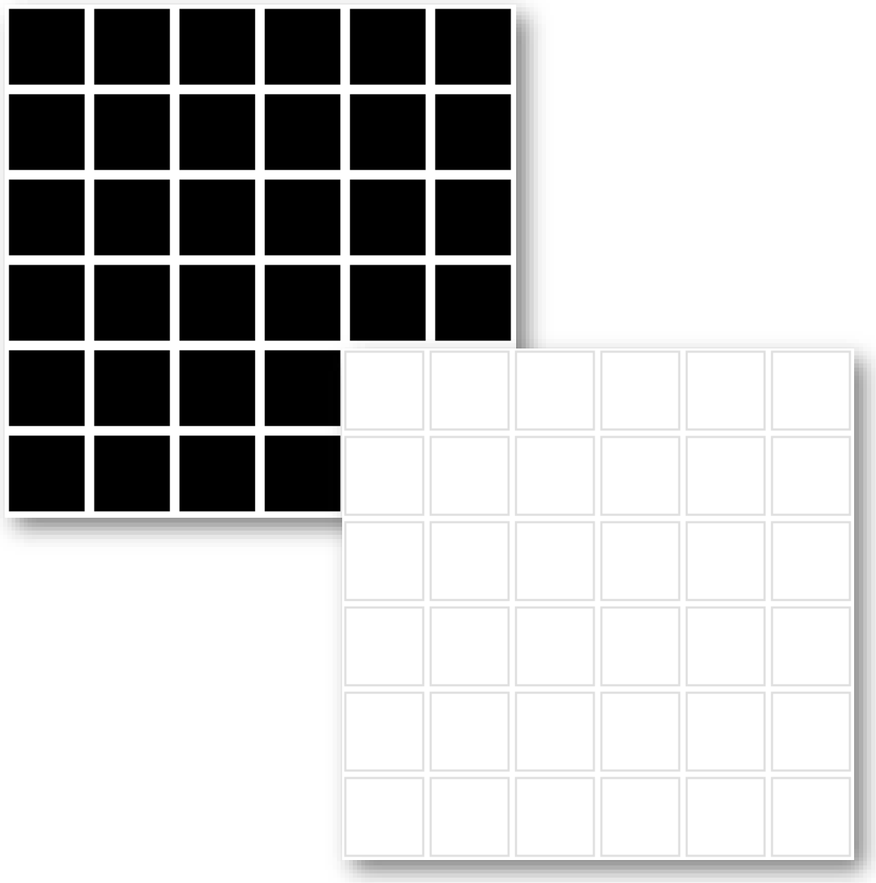


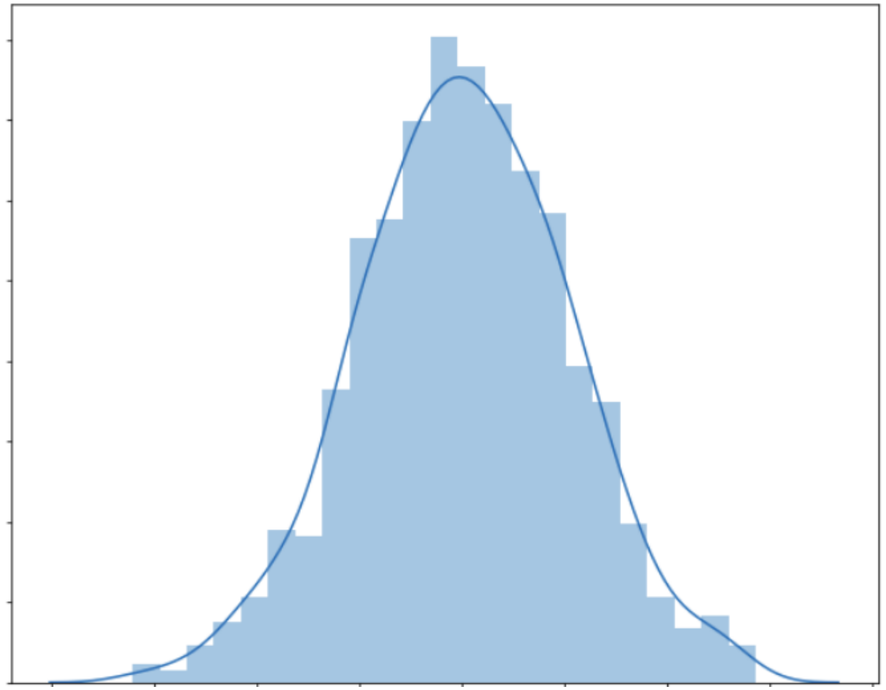
Cellák értéke

```
display_char = {
    -1: "?",
    0: "□",
    1: "■"
}
```



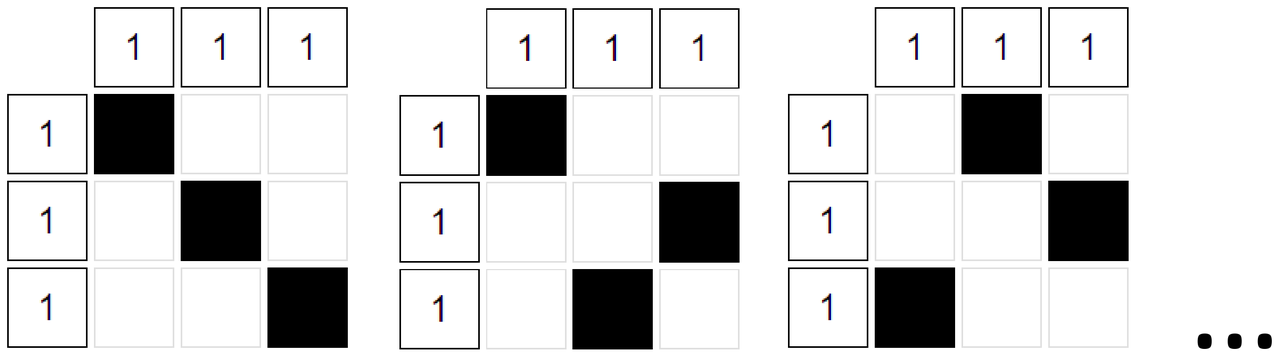
Normáleloszlás





$$\frac{(row * col)}{2}$$

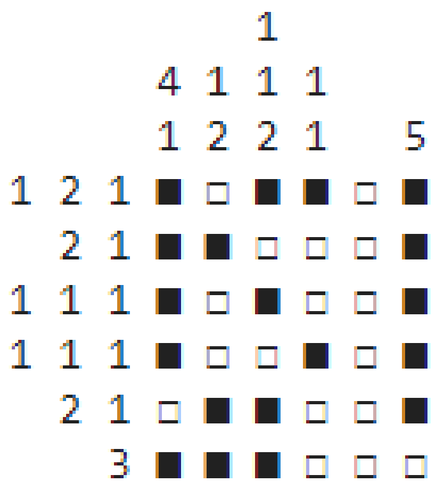
Egyértelműség



...

Tanítás

```
x = [1 2 1 0 2 1 1 1 1 1 1 1 0 2 1 0 0 3 0 4 1 0 1 2 1 1 2 0 1 1 0 0 0 0 0 5]
y = [1 0 1 1 0 1 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0]
```



Hibafüggvény

```
     real = [1 0 1 1 0 1 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0]
predicted = [1 0 1 1 0 1 0 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0]
      MAE = 1/36
```

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

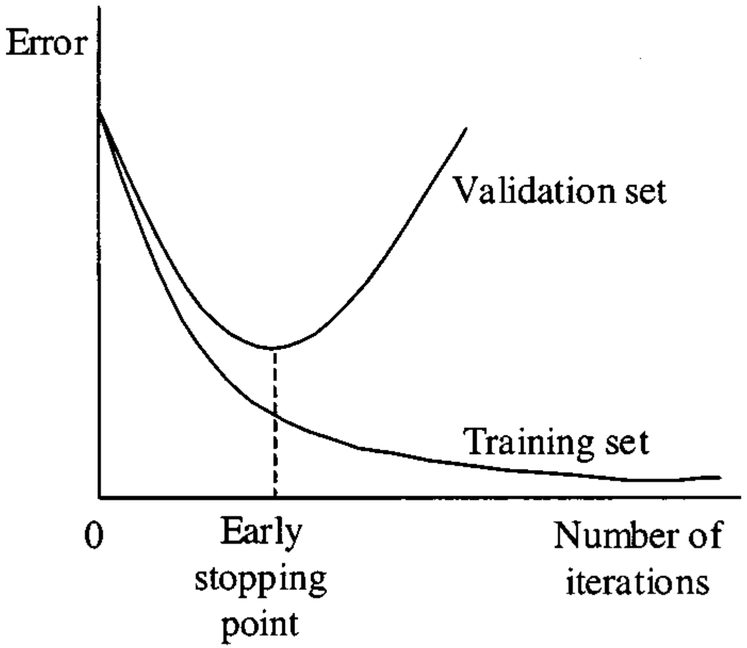
y_i = prediction

x_i = true value

n = total number of data points

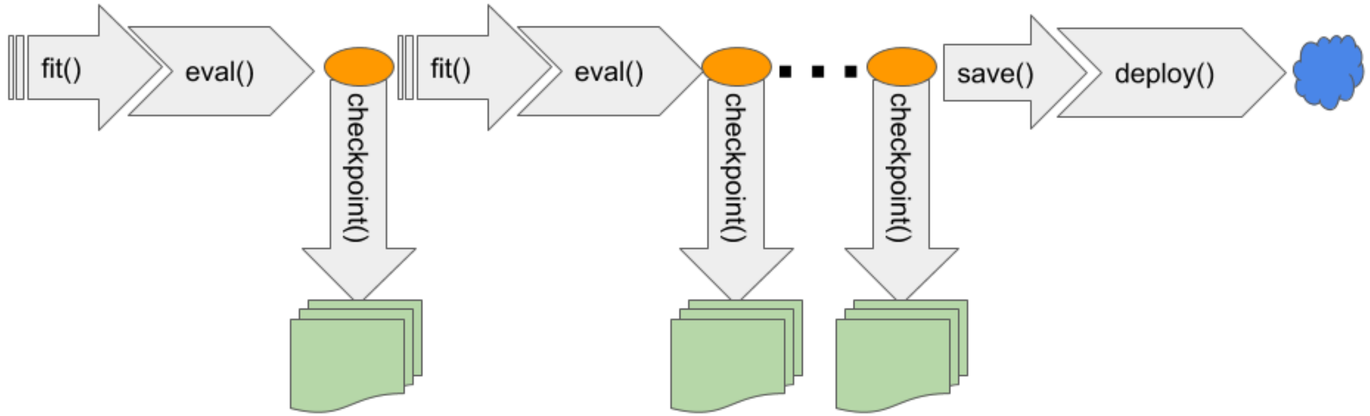
Early Stopping

```
early_stopping = EarlyStopping(patience=10, verbose=1)
```

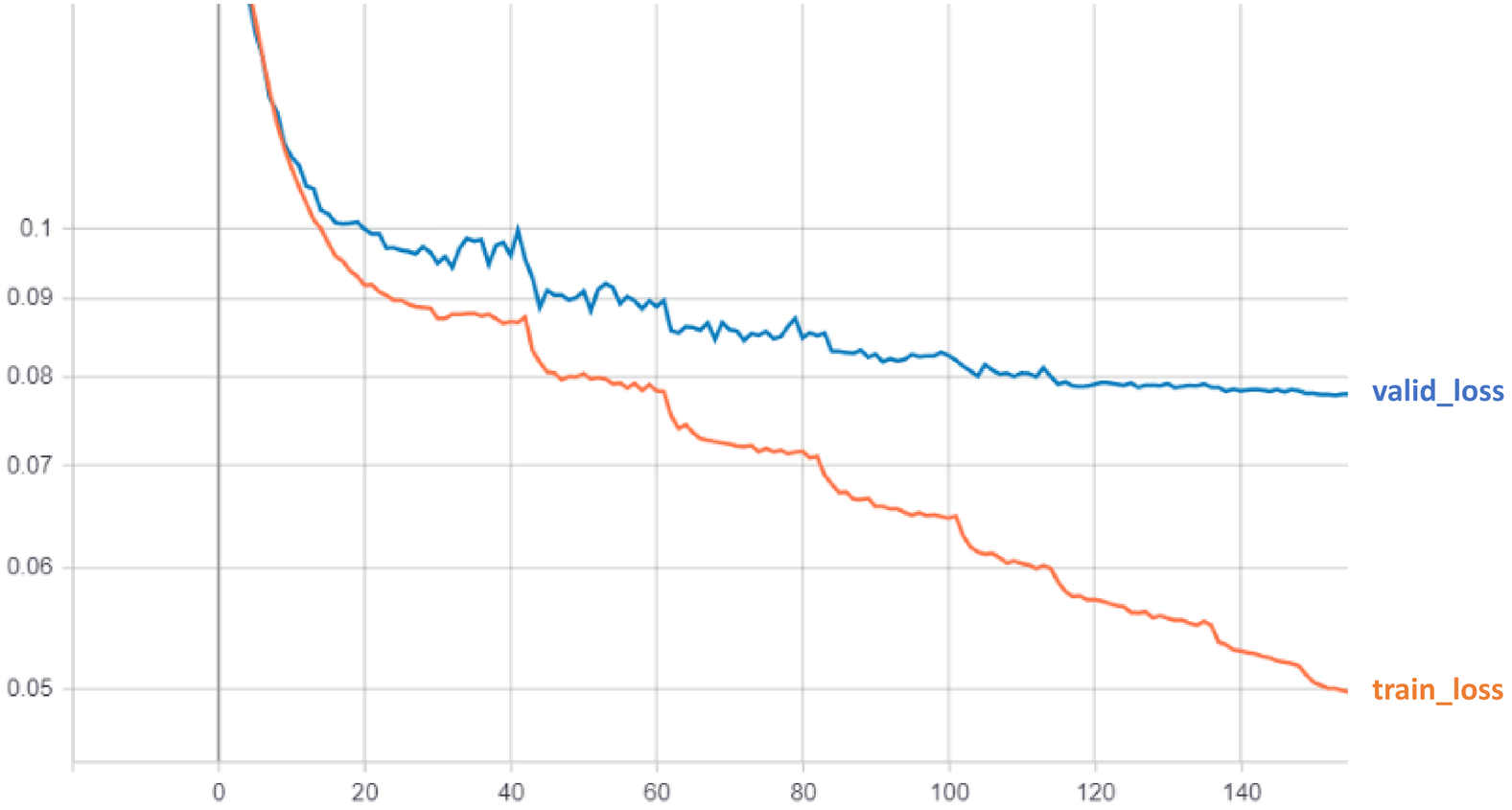


Checkpoints

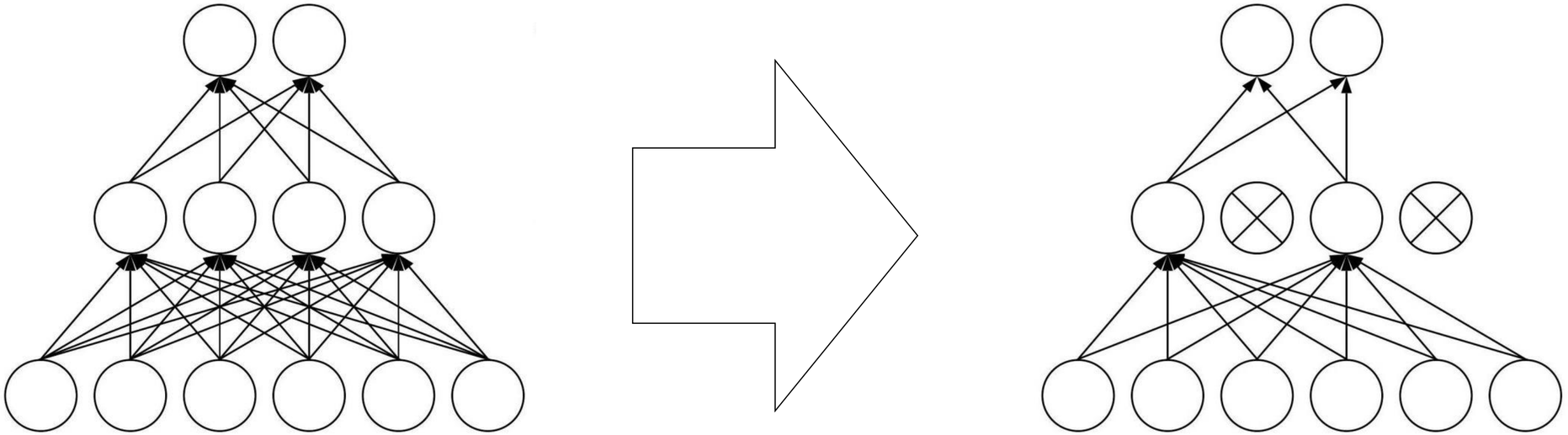
```
checkpoint = ModelCheckpoint(filepath='cnn_model_best', save_best_only=True, verbose=1)
```



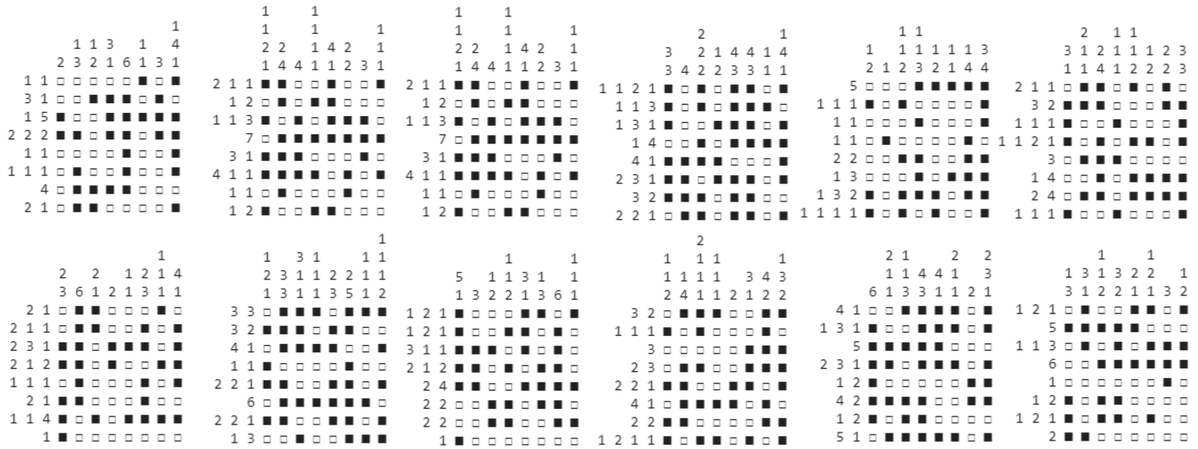
Túltanulás


0.1
0.09
0.08
0.07
0.06
0.05
0
20
40
60
80
100
120
140
valid_loss
train_loss

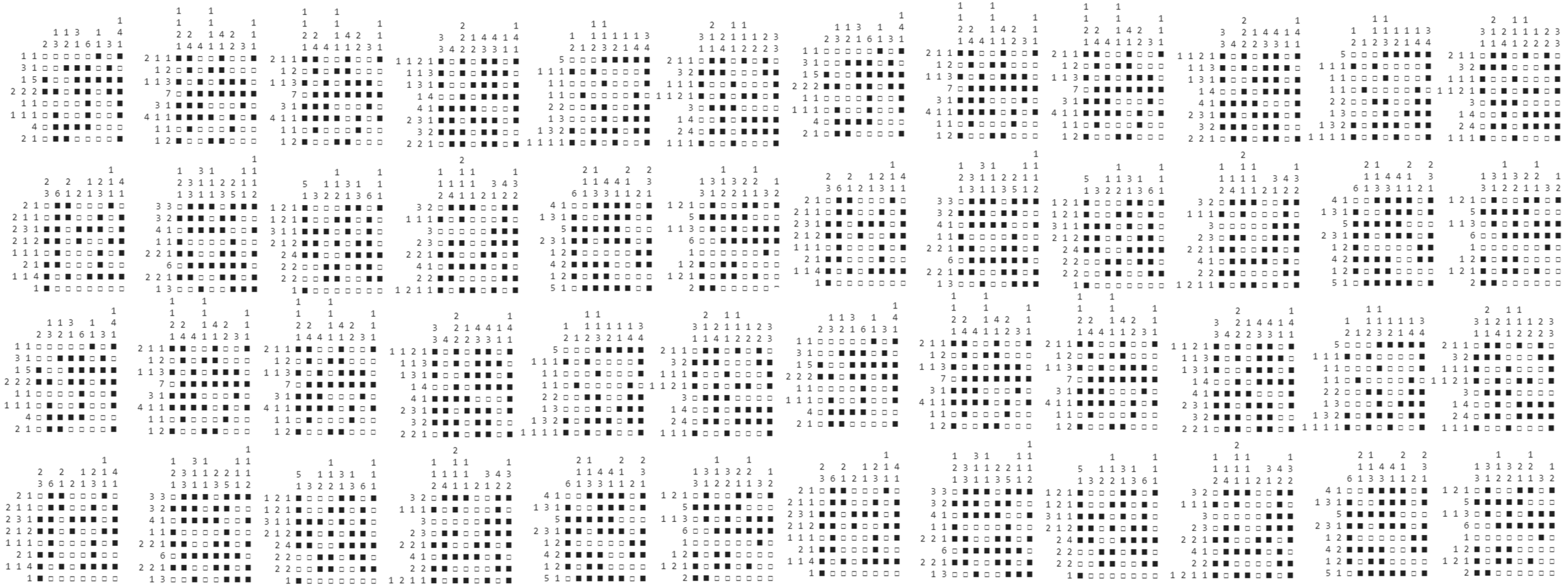
Dropout



Adatgenerálás

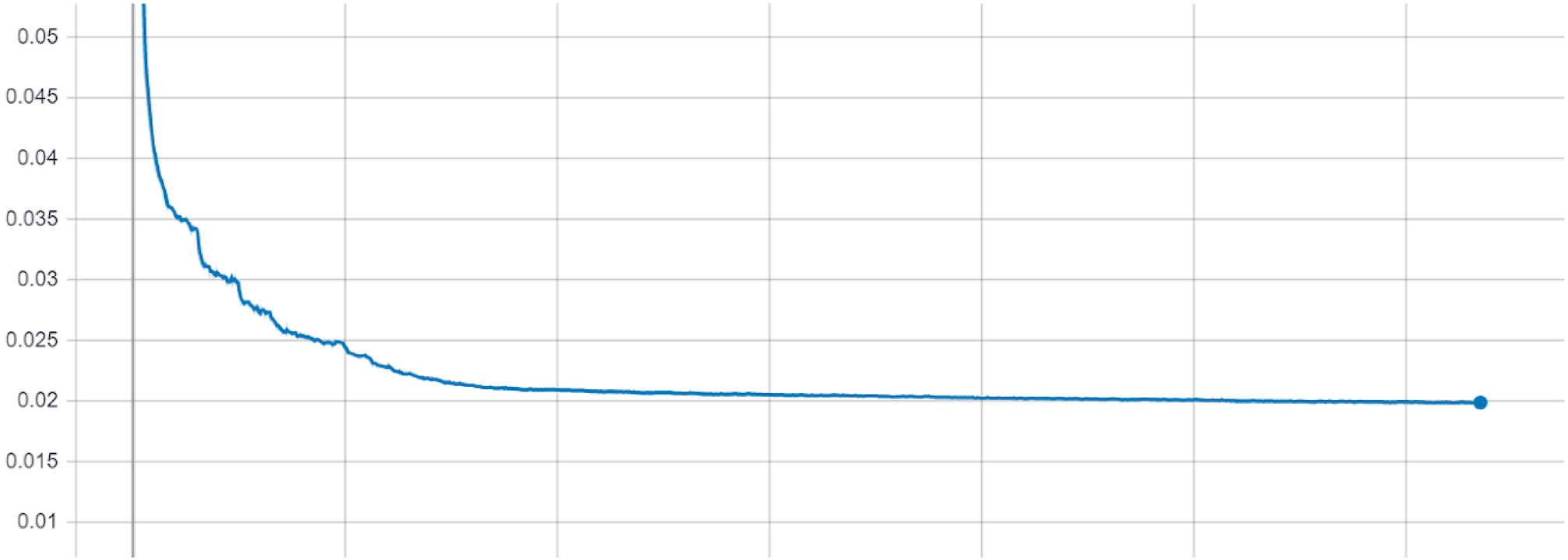


Adatgenerálás



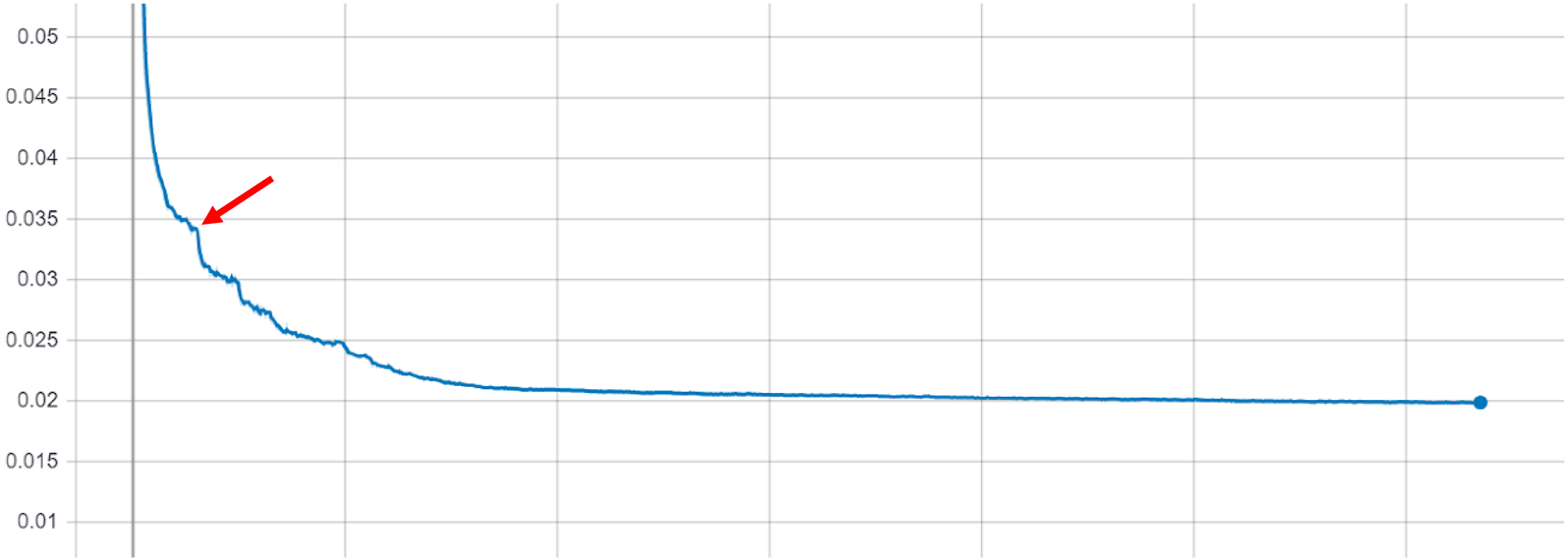
Adaptive learning rate

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.75,
                              patience=10, min_lr=0.0001, verbose=1
```



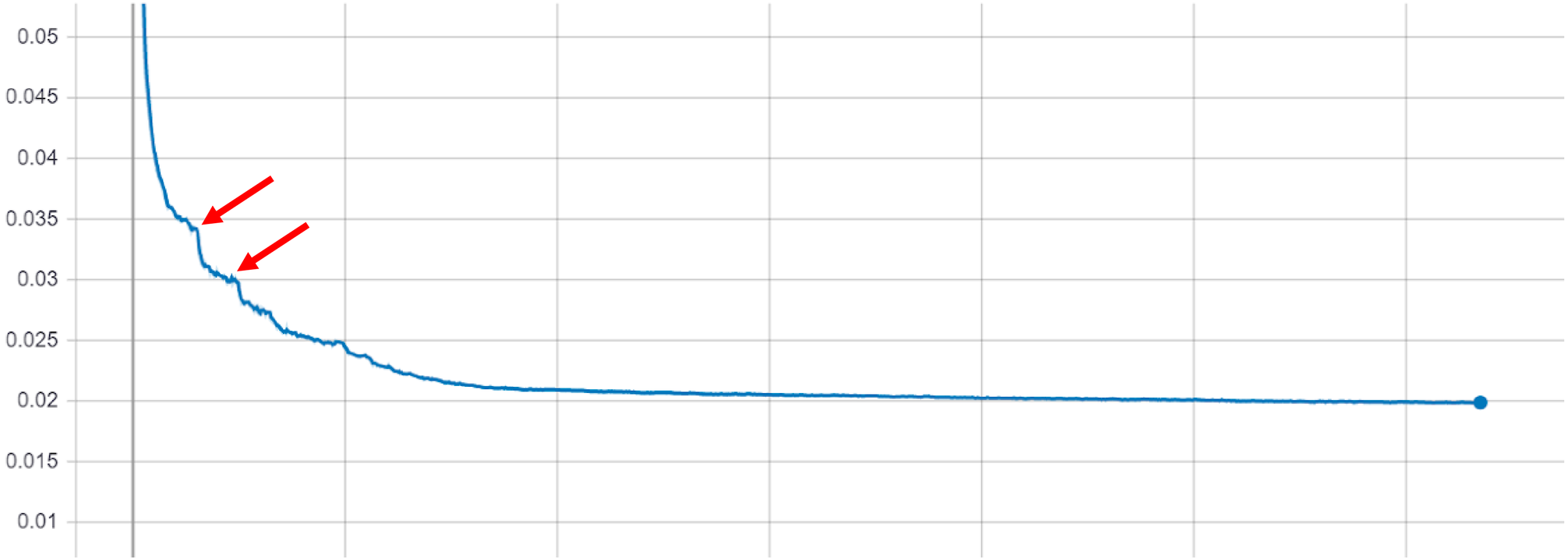
Adaptive learning rate

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.75,
                              patience=10, min_lr=0.0001, verbose=1
```



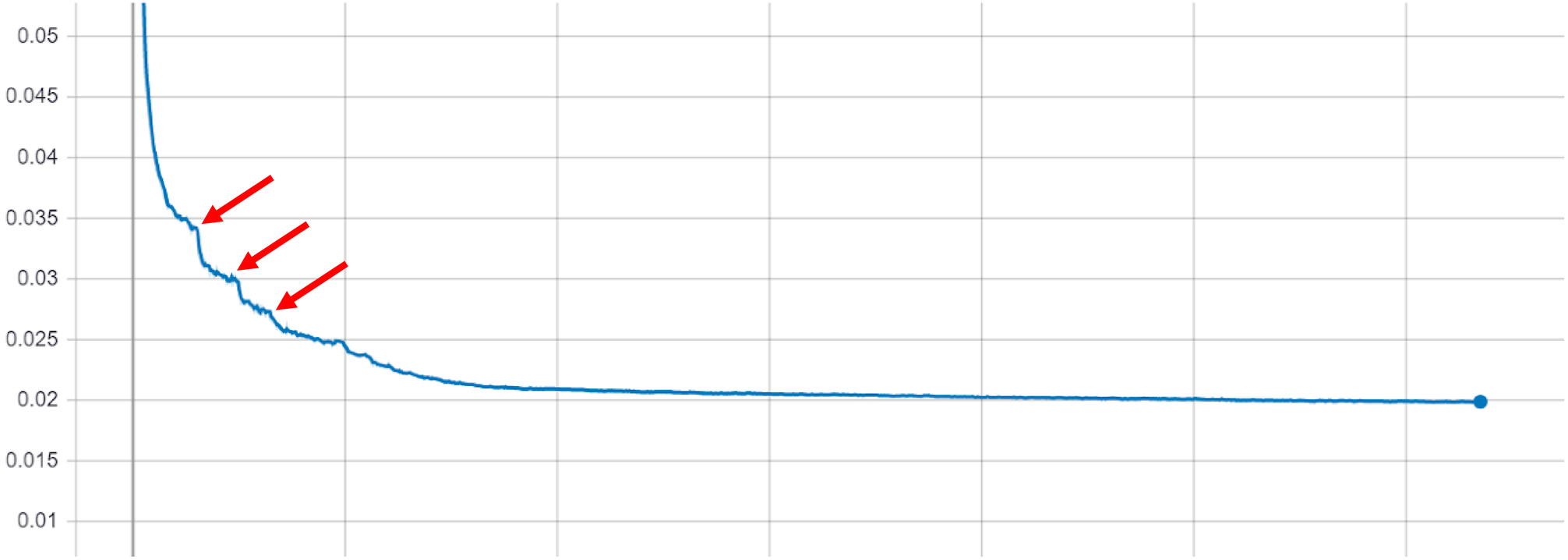
Adaptive learning rate

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.75,
                              patience=10, min_lr=0.0001, verbose=1
```



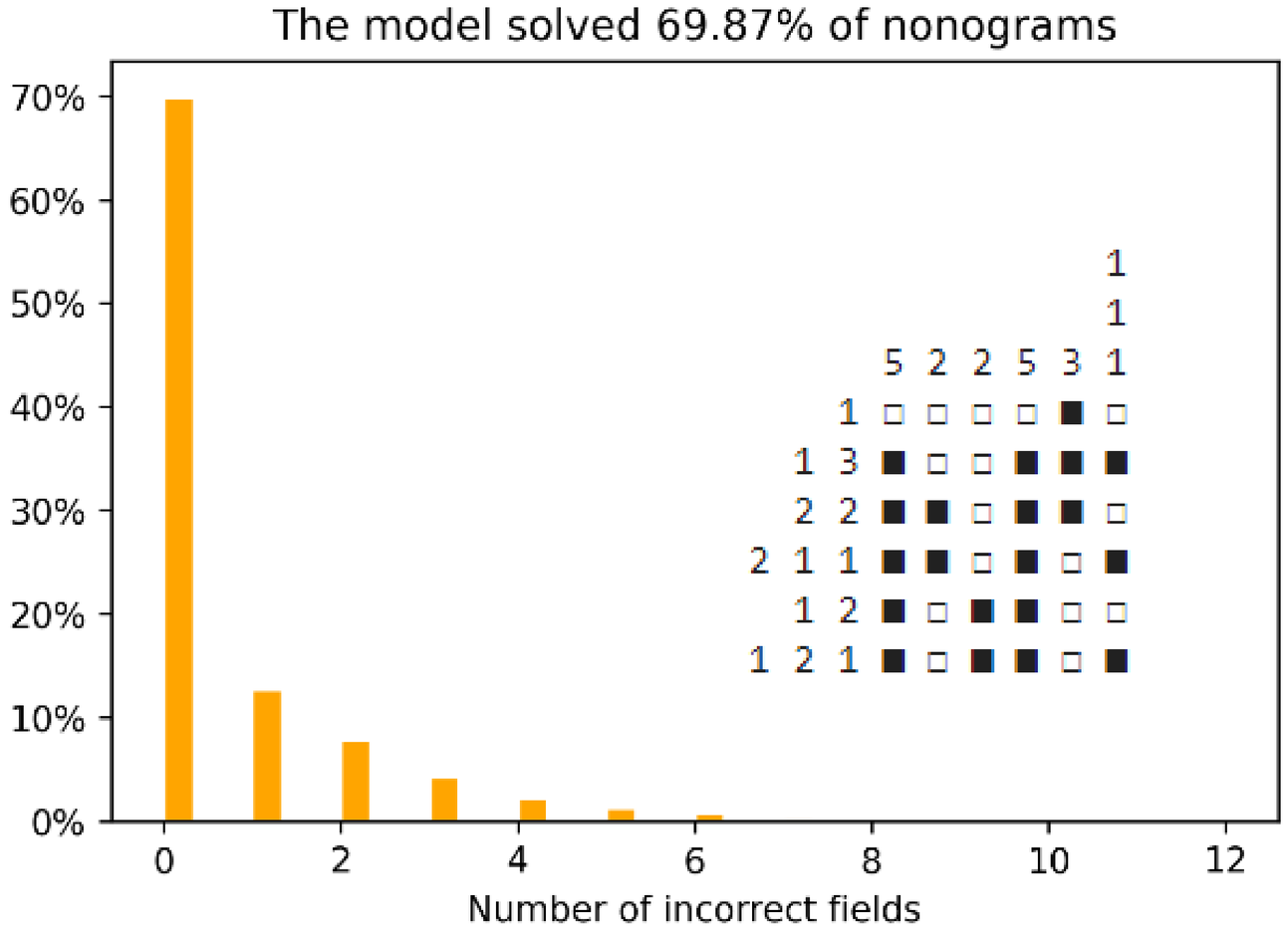
Adaptive learning rate

```
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.75,
                              patience=10, min_lr=0.0001, verbose=1
```

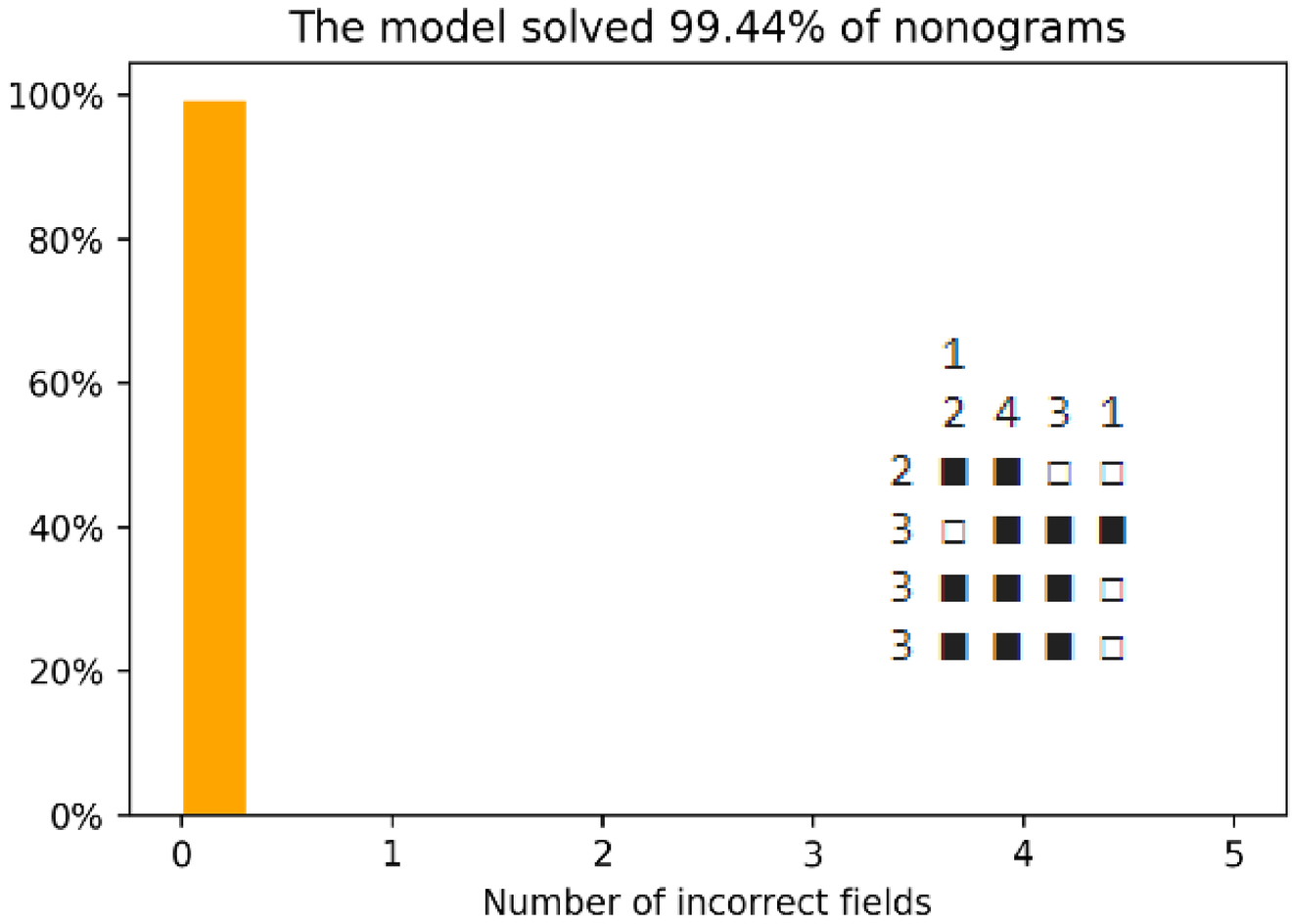


Kiértékelés

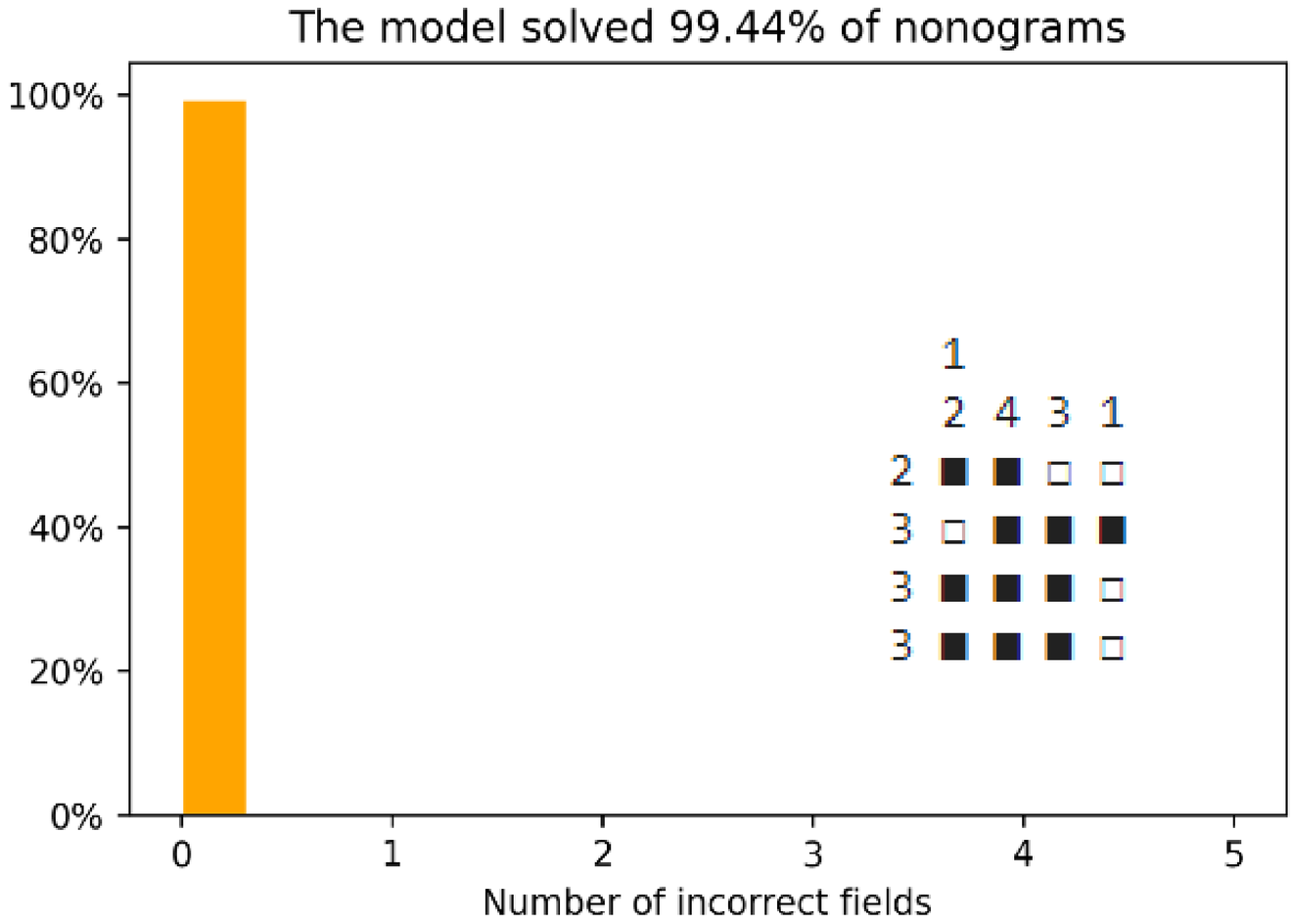
Kiértékelés



Kiértékelés



Kiértékelés



$$2^{(4*4)} = 65536$$

Deployment

TensorFlow.js is a library for machine learning in JavaScript

Develop ML models in JavaScript, and use ML directly in the browser or in Node.js.

See tutorials

Tutorials show you how to use TensorFlow.js with complete, end-to-end examples.

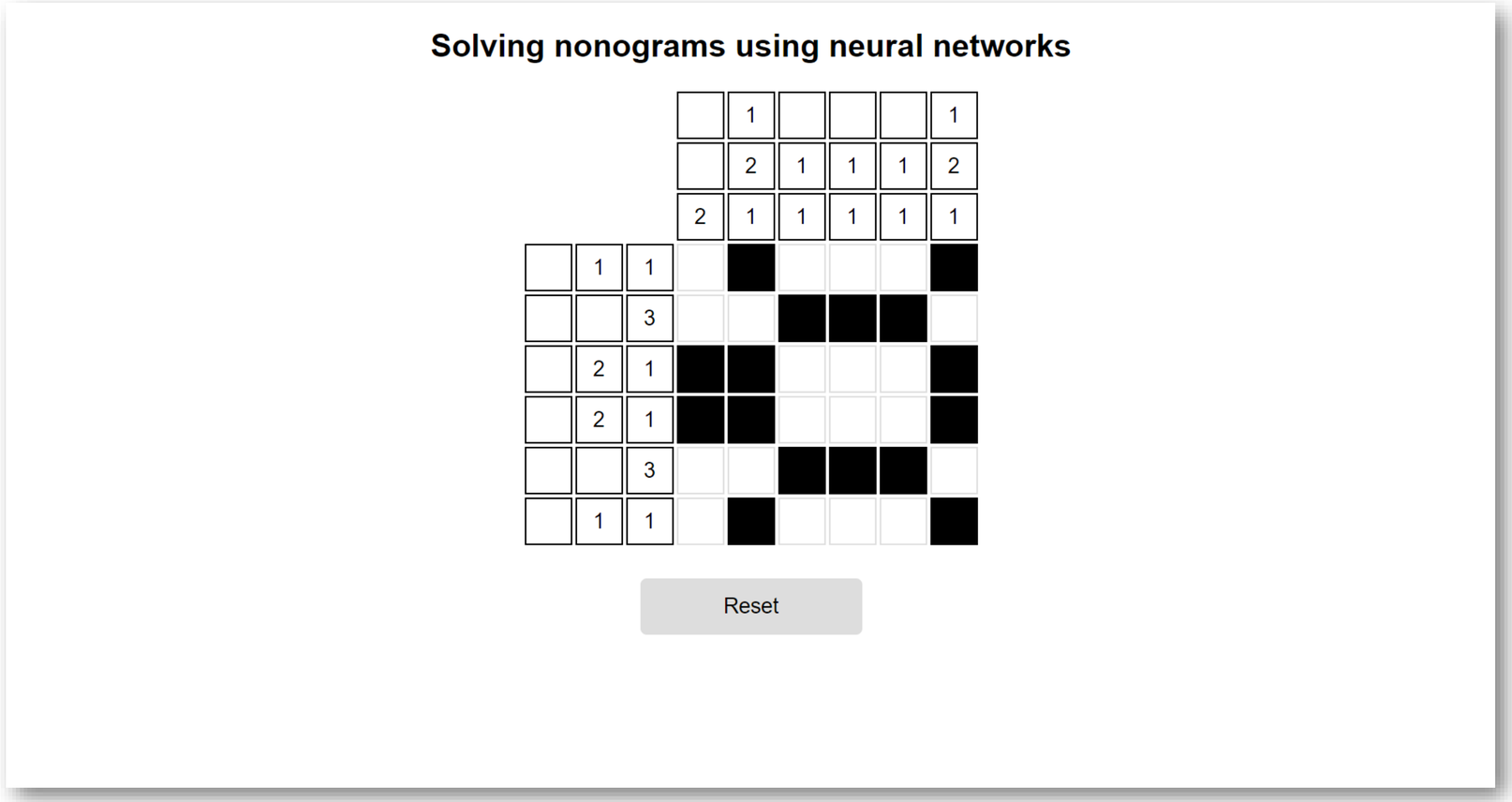
See models

Pre-trained, out-of-the-box models for common use cases.

See demos

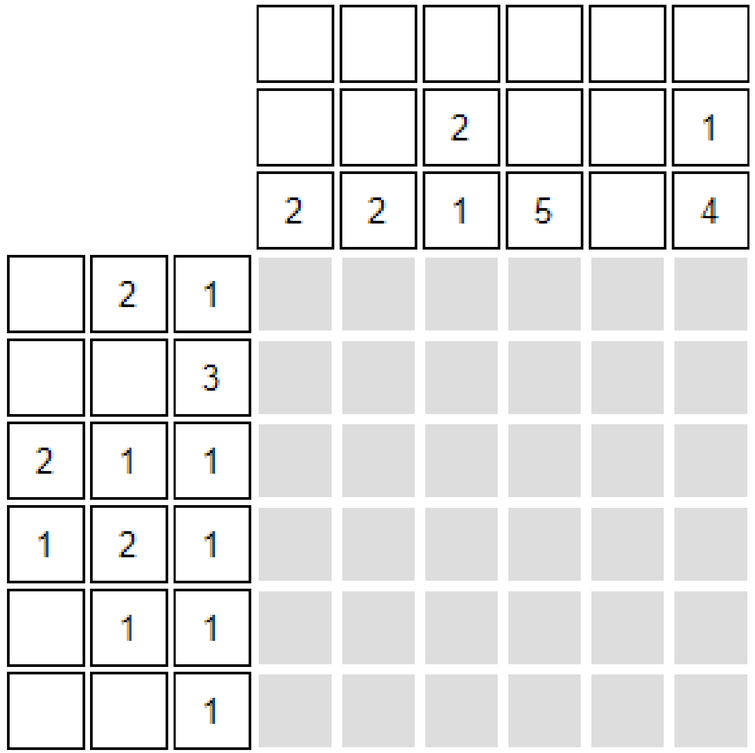
Live demos and examples run in your browser using TensorFlow.js.

Demo

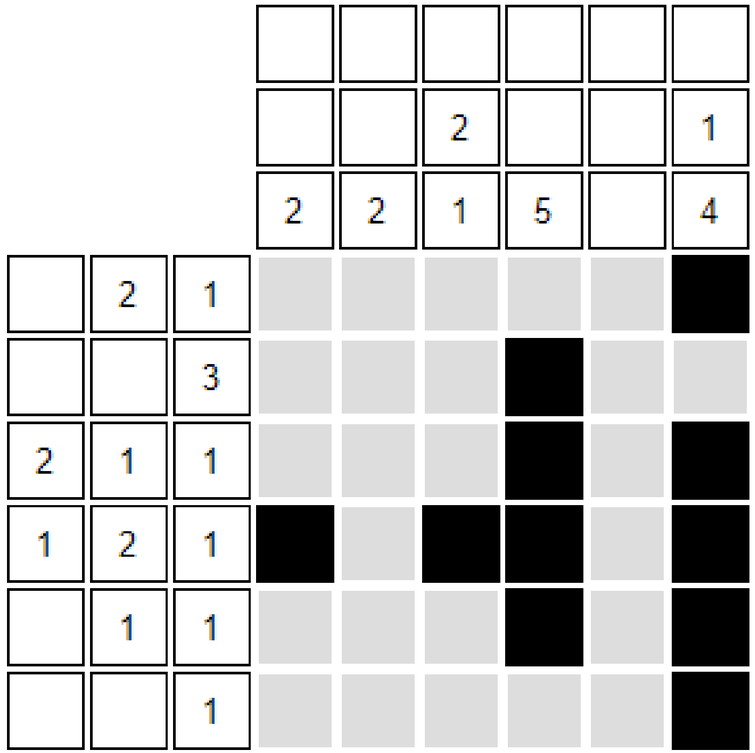


https://nonogram-solver.web.app/

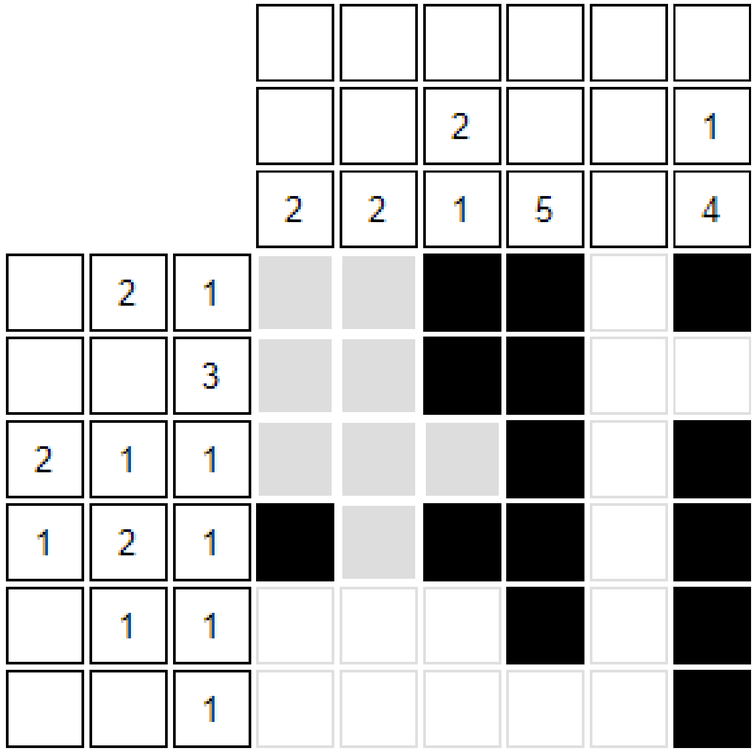
Tervek



Tervek



Tervek



Tervek