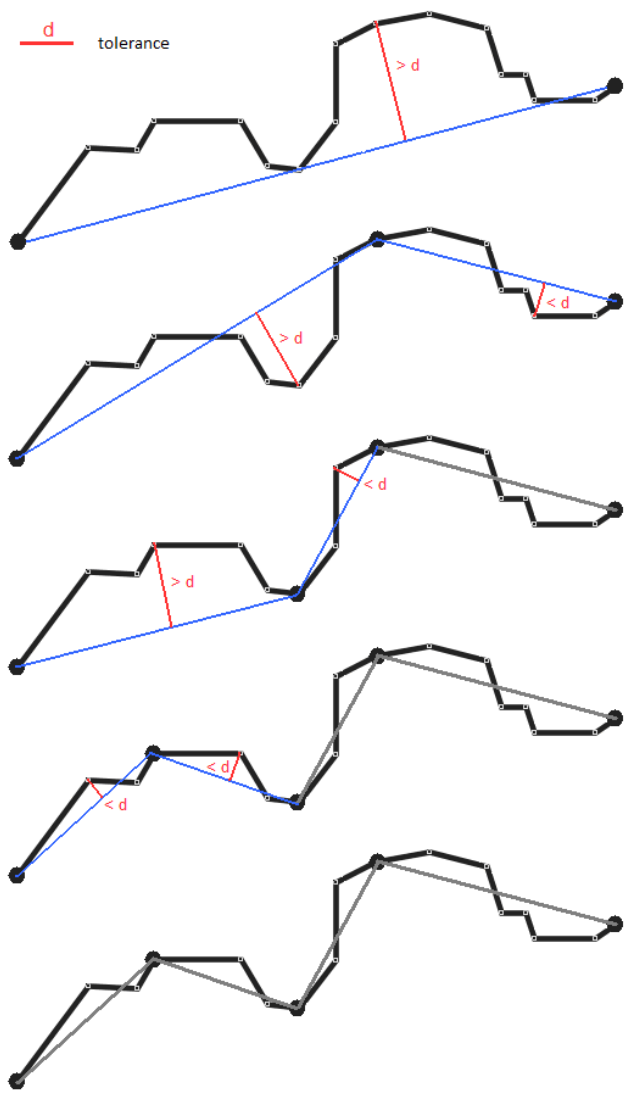
Utilisation of computational intelligence for simplification of linear objects using extended WEA algorithm

[1] Anna Fiedukowicz
Agata Pillich-Kolipińska
Robert Olszewski

[1]This work has been supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme.

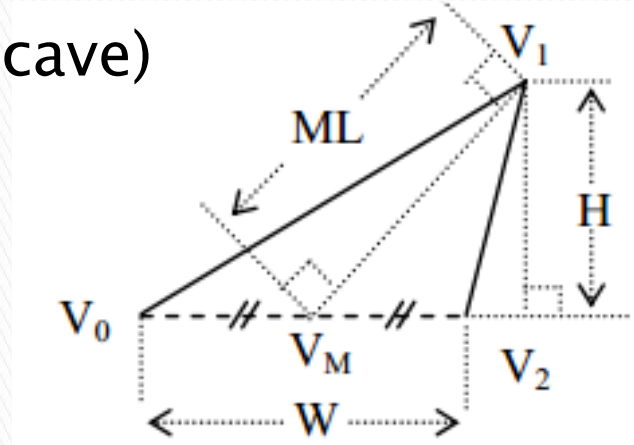
Line simplification

- Widely known and described
- Selection of „important" vertices
- Different algorithms, mostly **one-parameter** ones
  - Douglas-Peucker (1973) – distance
  - Visvalingam-Whyatt (1993) – „effective area"
  - Wang (1996) – curve radius
  - many others…



Weighted Effective Area

- „Shape-aware" algorithm
- S. Zhou, C.B. Jones (2004) – **4 parameters**:
  - „effective area" from Visvalingam-Whyatt algorithm
  - flatness (H/W)
  - skewness (H/ML)
  - convexity (↻ – convex, ↺ – concave)
- Defined filters:
  - W_{flat}
  - W_{skew}
  - W_{convex}



$$WEA = W_{flat} * W_{skew} * W_{convex} * EA$$

Modified WEA (2)

Non-deterministic approach: **knowledge base**

Vertex weight = **f**(*flatness, skewness, convexity, EA)*

Explicit
set of rules
defined by an expert

Implicit
derived from examples
provided by an expert

computational intelligence

Fuzzy Inference Systems

Artificial Neural Networks

Modified „WEA" (weight of a vertex)

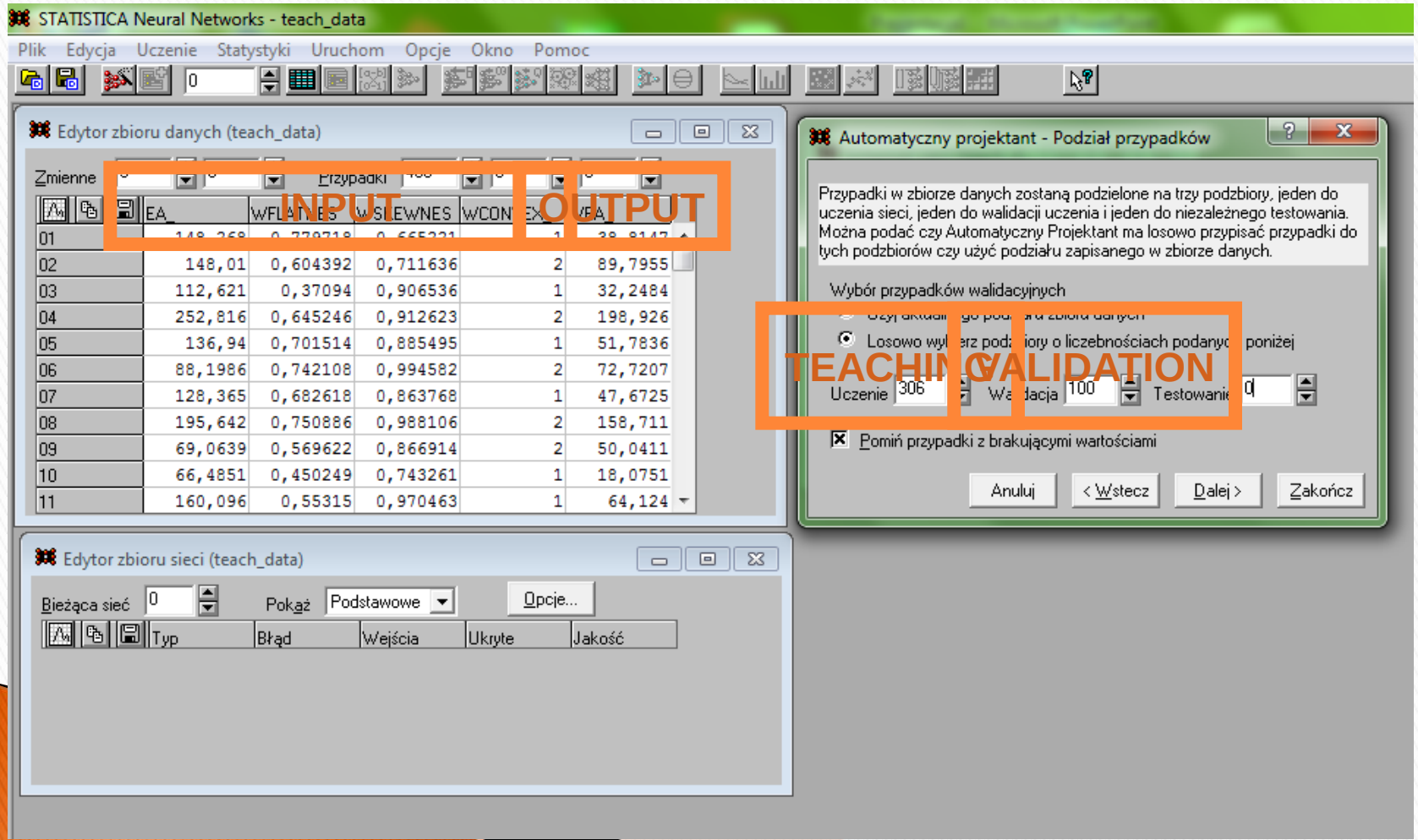
Modified WEA (2)

Vertex weight = **f**(*flatness, skewness, convexity, EA)*

rules	examples
Defined by an expert, eg. ‣ if (EA > 100) and (*flatness* > 1) and (*skewness* > 0.8) then weight = 99; *Or more fuzzy definition:* ‣ if EA is big and *flatness* is high and *skewness* is low then vertex is important	Important points derived from Topographic DBs and maps 1:10k, 1:50k, 1:250k ‣ Polish coastline (SABE 30 and surveying data) ‣ Polish rivers Vertices weights proposed by an expert

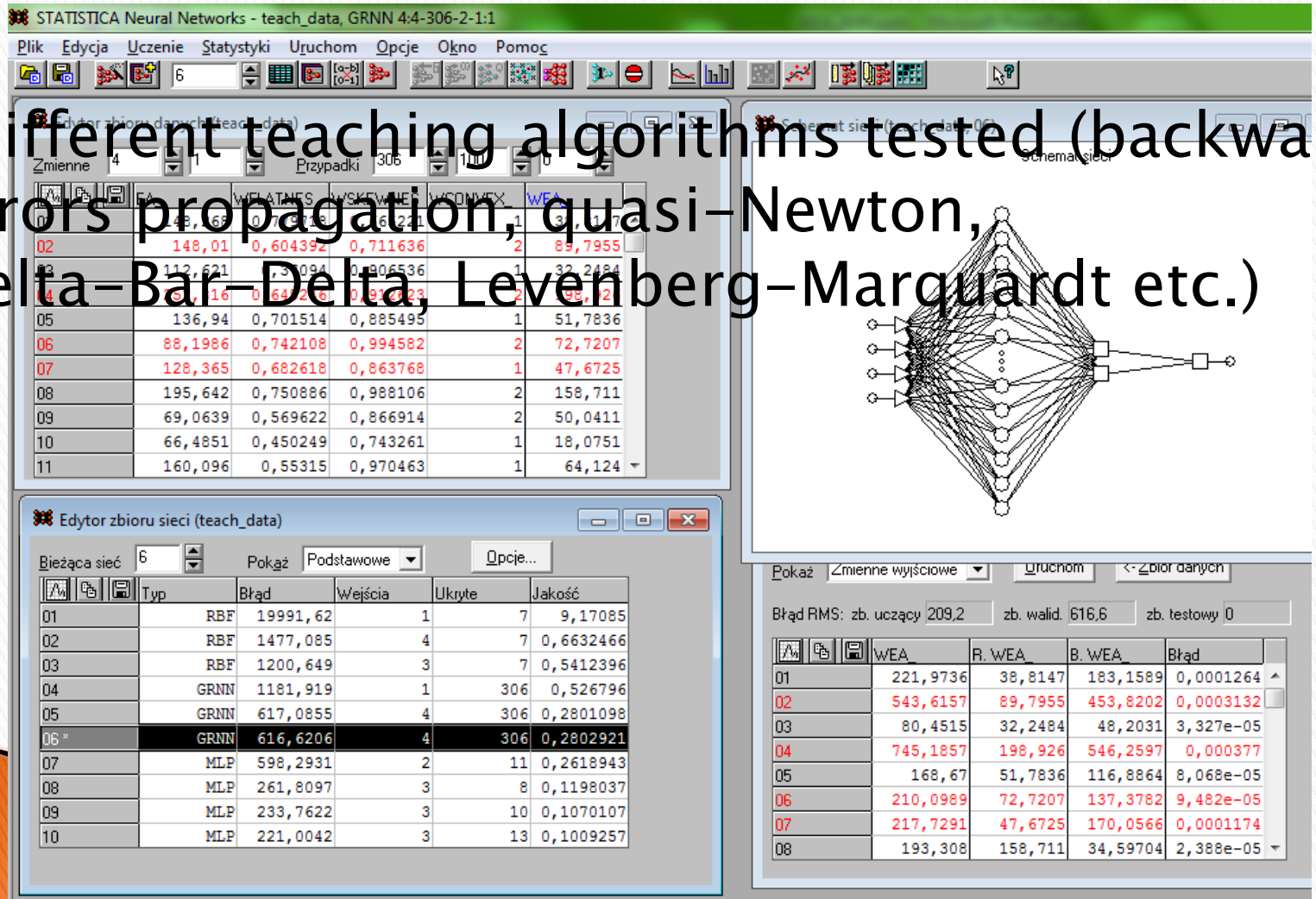
Finding f() – ANN (1)

- STATISTICA Neural Networks software



ANN (2)

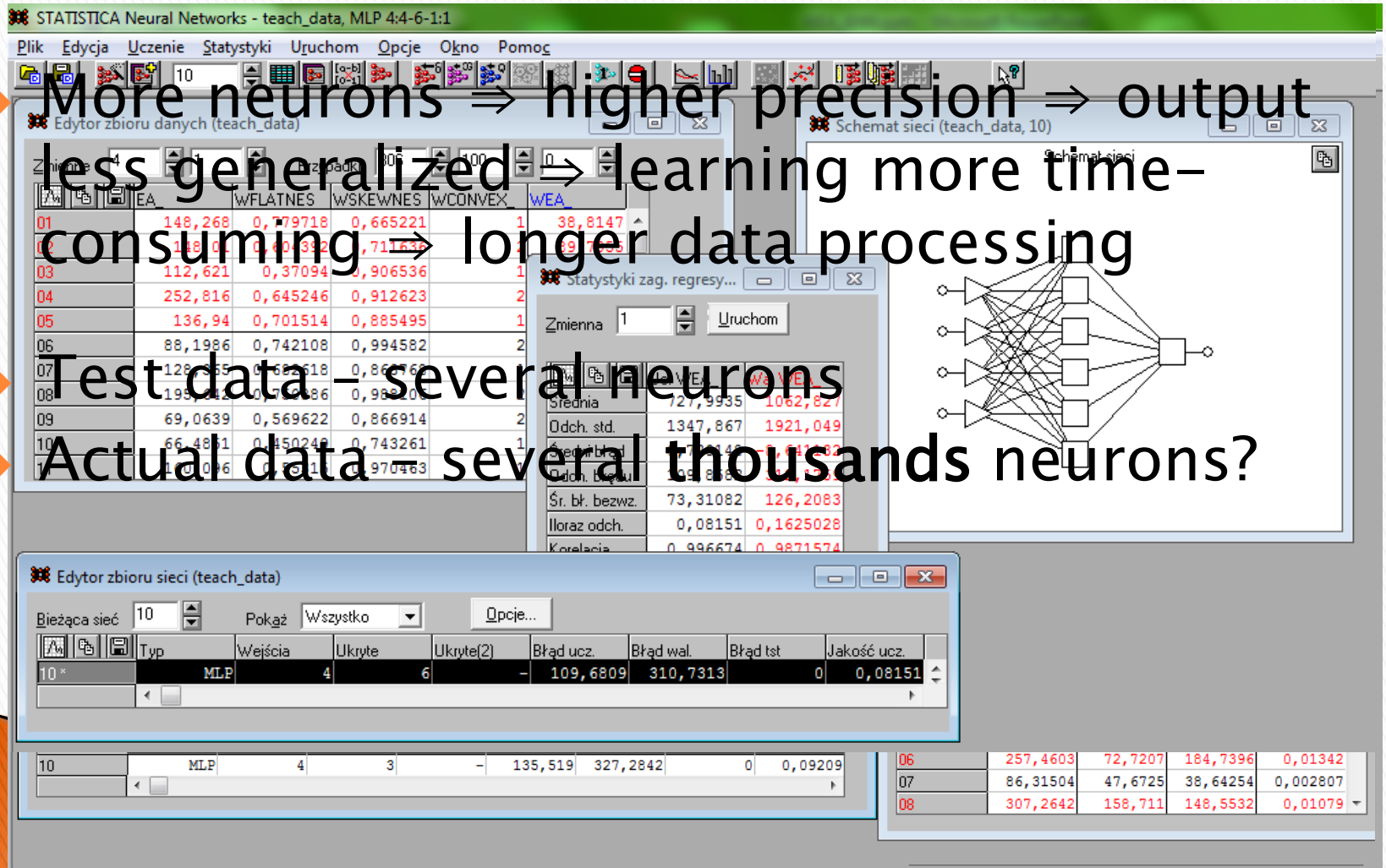
- Three types of ANNs tested



- Different teaching algorithms tested (backward errors propagation, quasi-Newton, Delta-Bar-Delta, Levenberg-Marquardt etc.)

ANN (3)

- Testing different structures of same ANN type
- More neurons ⇒ higher precision ⇒ output less generalized ⇒ learning more time-consuming ⇒ longer data processing
- Test data – several neurons
- Actual data – several **thousands** neurons?



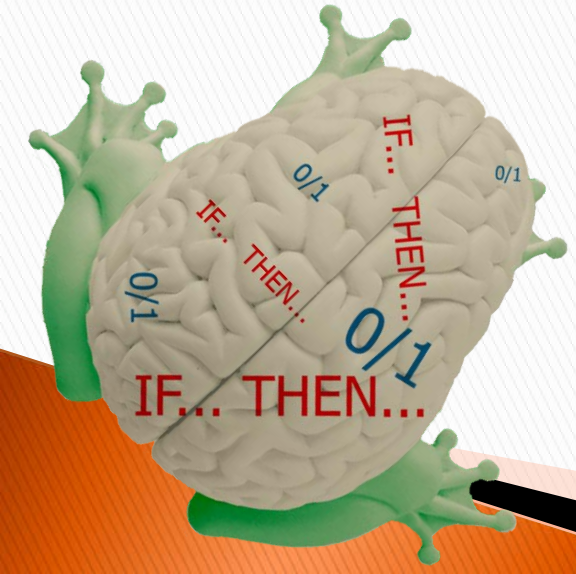
ANN – results

Rules based knowledge base

- IF building<400m² THEN y=0,05

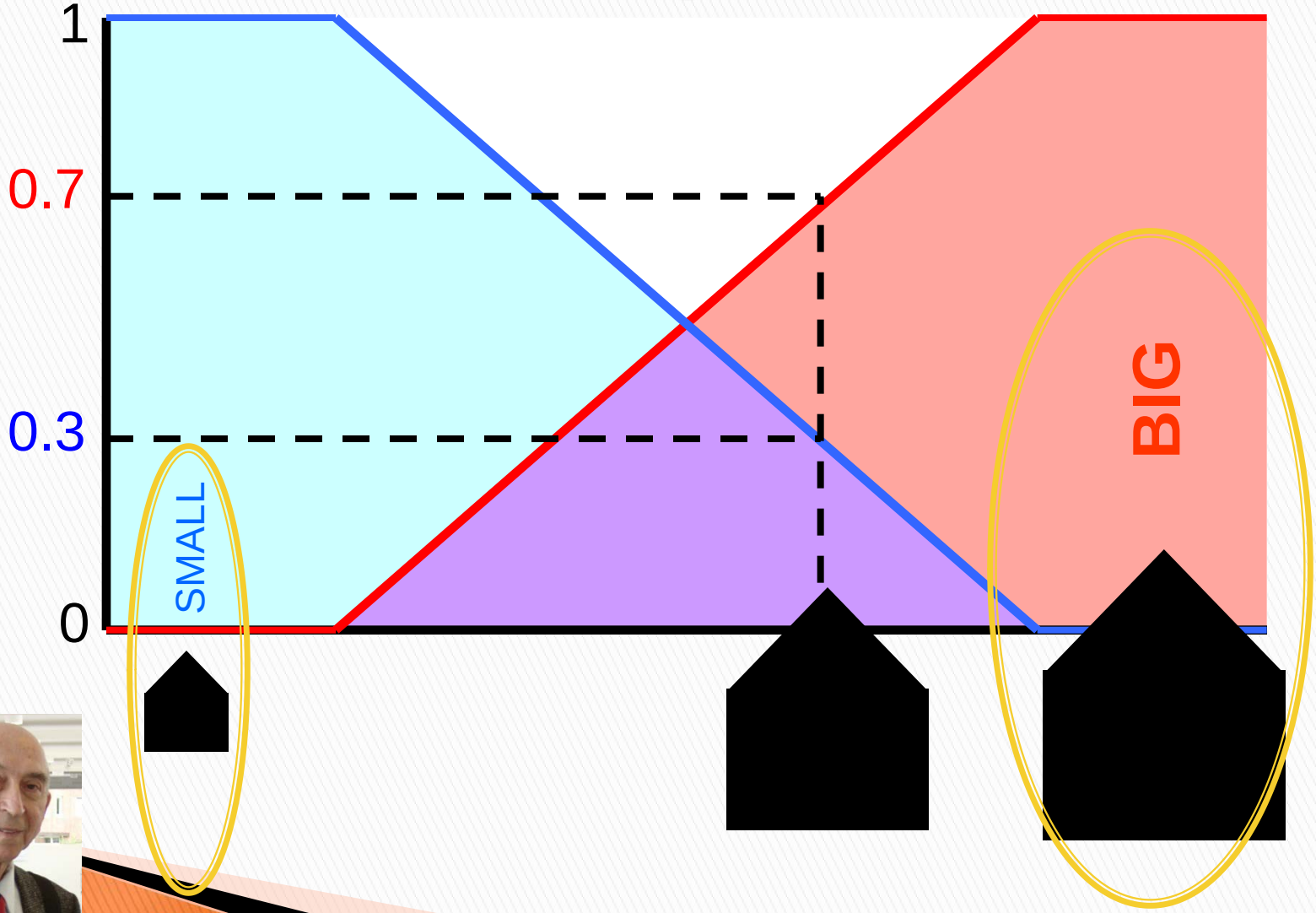
A → **B**

- IF building is small THEN simplify a bit



How you define „small"?

Membership functions
=>Linguistic variables

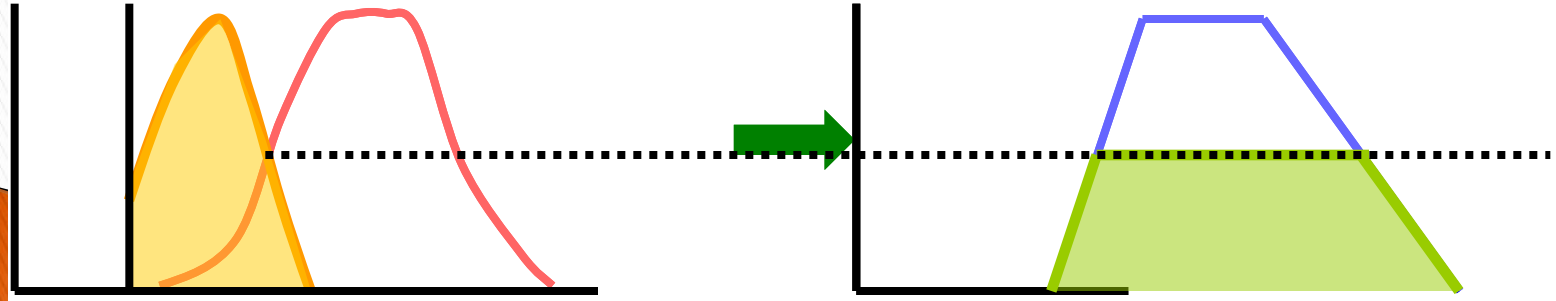


Fuzzy rules

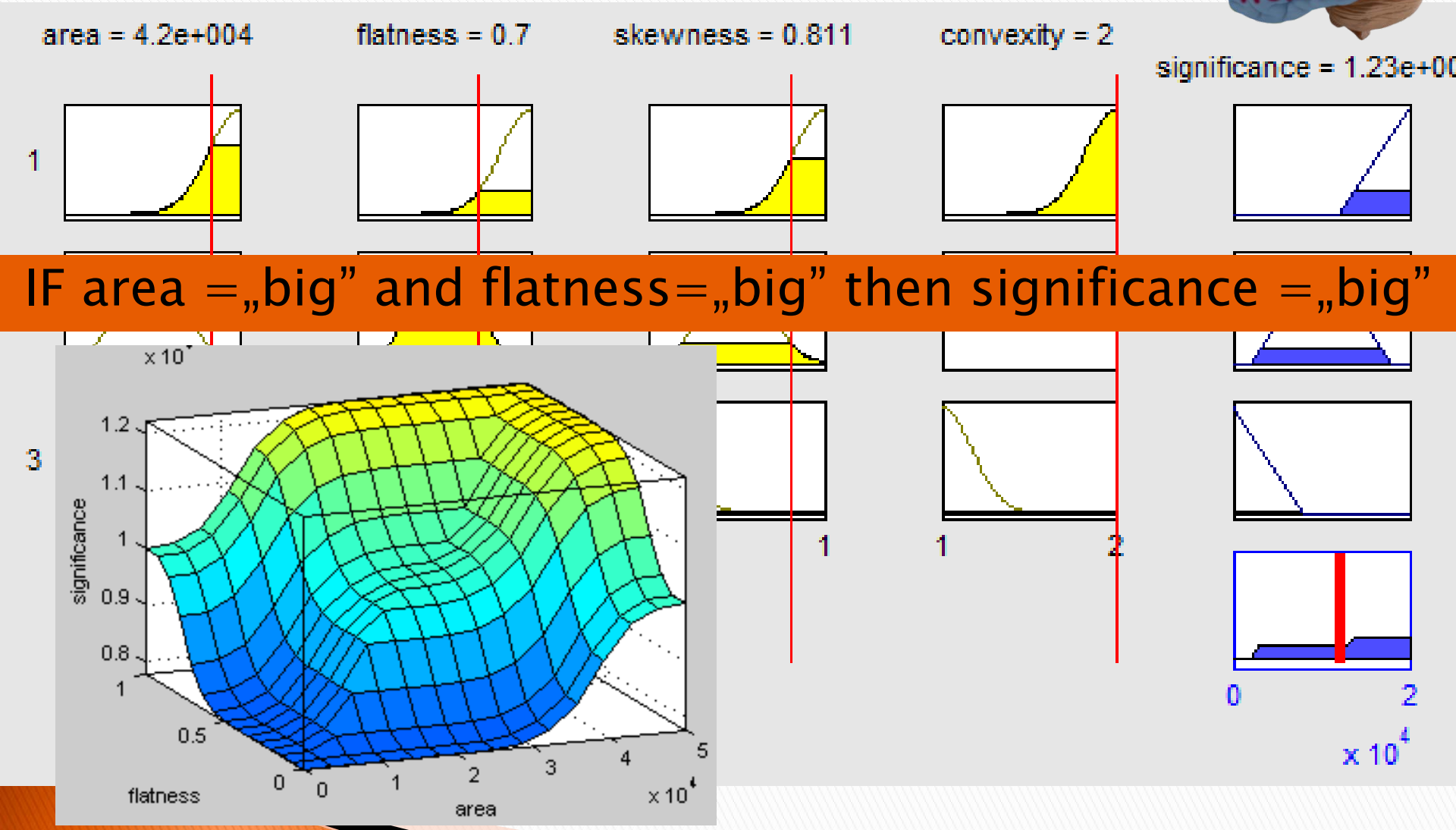
- IF x = A THEN y = B

A → **B**

- IF x = A THEN y = B
  - And what IF x = A' ? THEN y = B'



Creation of decision rules



IF area =„big” and flatness=„big” then significance =„big”

Fuzzy reasoning results

Conclusions

NEURO	FUZZY
Easy to build	Difficult to define
Black box for user	Allows process understanding
Long computation time for complex tasks	Builded once works fast

State of work

- Tools independent on GIS
- Preliminary tests on polish coast line

Up to now

- Combine fuzzy and neuro tools with GIS sofware
- Check tools on different type of data

Future plans

Thank you for your attention!

Here it comes, here comes the weekend…

r.olszewski@gik.pw.edu.pl
a.pillich@gik.pw.edu.pl
a.fiedukowicz@gik.pw.edu.pl