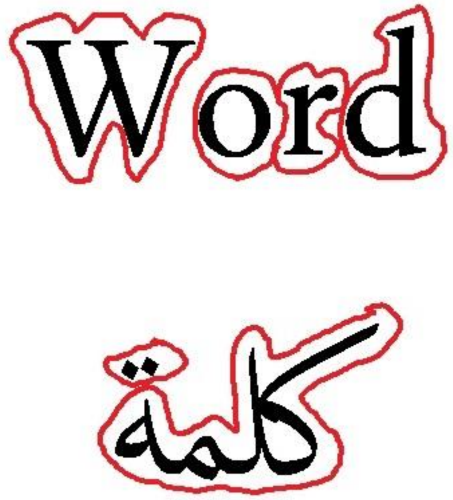
TU Clausthal

An approach using neural networks on curves to determine the font of Arabic calligraphy art works

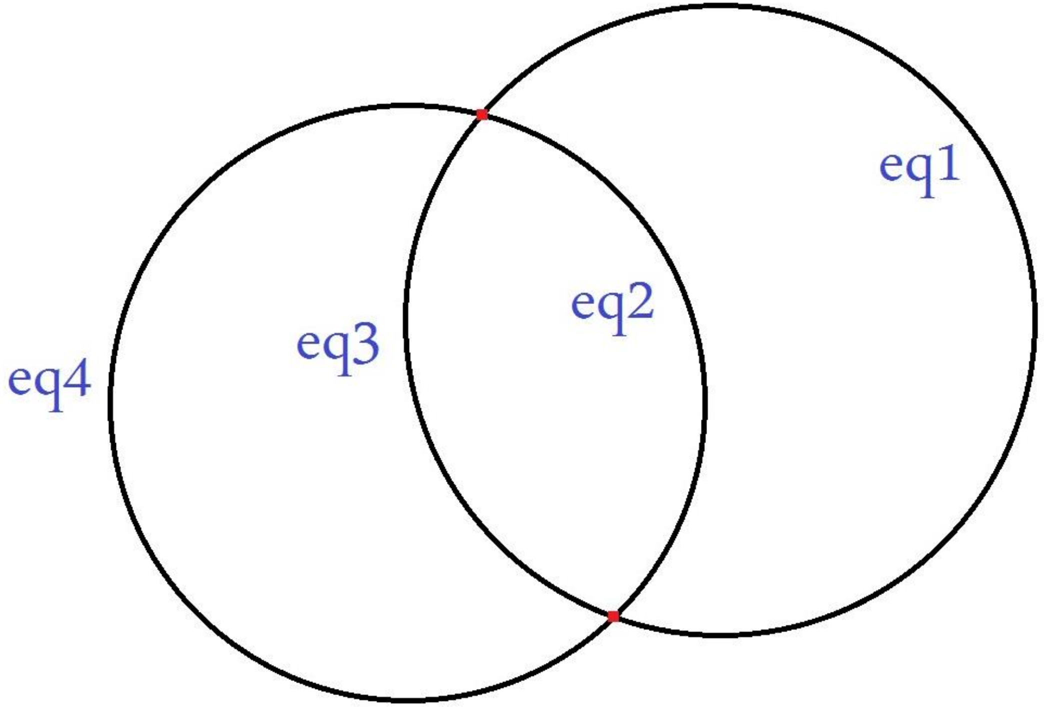
Hayyan Helal

Datum: 13.03.2018

State of Art & Problem with Arabic

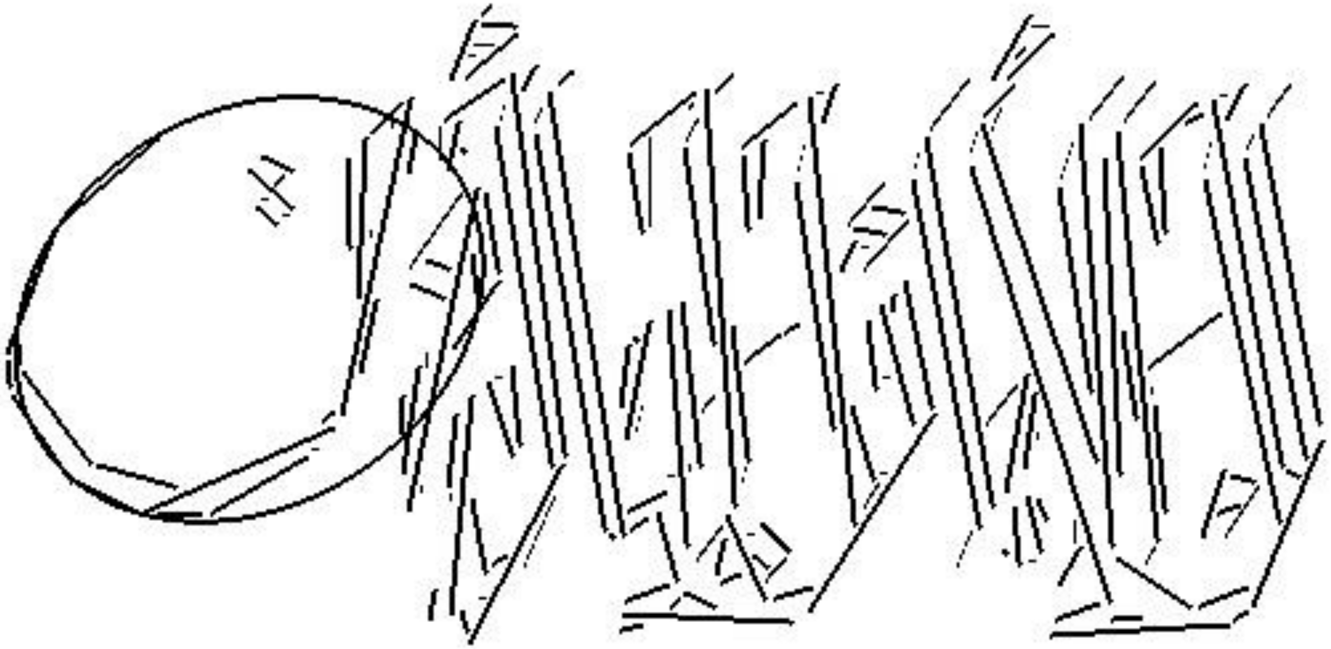


Proposed Solution: Curves



TU Clausthal

Original Idea: Homemade Vectorization



Modified Idea: SVG, Potrace

- SVG: Scalable vector graphics.
- Potrace: Auto-vectorizer for .bmp-images.
- Result:
  - Lines $\rightsquigarrow$ two points (`l`).
  - Curves $\rightsquigarrow$ control points of the Bézier-curves (`c`).
  - Connected black points $\rightsquigarrow$ path (`path d="…"`).

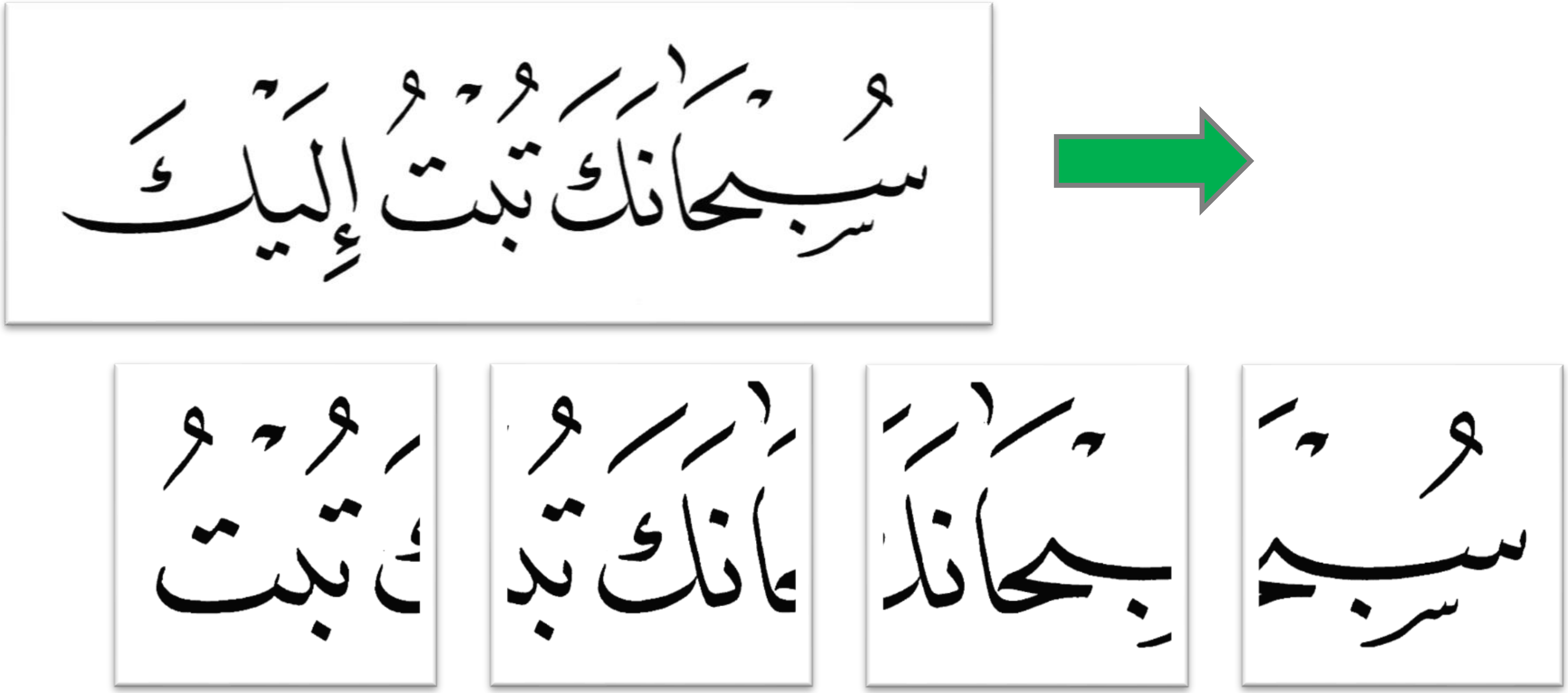
Steps

- I. From photo to curves.
- II. From curves to font.

TU Clausthal

I. From Photo to Curves

- Convert: .jpg to .bmp.
- Black and White.
- Cut in lines & pieces: due to long curves.
- Vectorize: Potrace.
- Simplify: due to long curves.
- To List of Integers: input of the NN.


سُبحانَكَ تُبتُ إِلَيكَ

SVG-Structure

-
  ```
  <path d="M437 1813 c-3 -5 -8 -42
  -11 -83 l-7 -75 56 -48 c67 -58 108
  -133 108 l-197 1 -25 -3 -84 -7 -132
  -7 -74 -5 -88 7 -88 16 0 40 76 62
  198 25 147 -19 277 -128 378 -57 53
  -72 61 -80 47z"/>
  ```

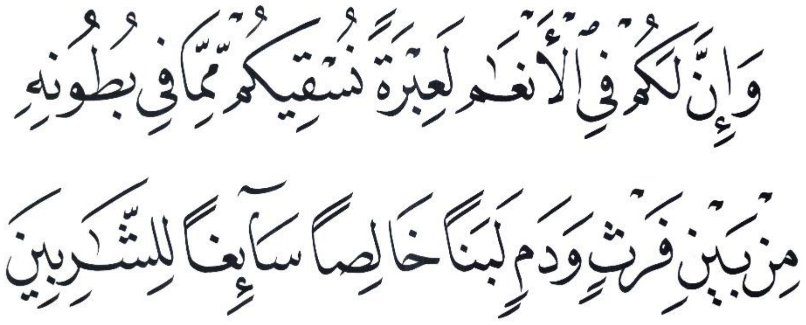
II. From Curves to Font

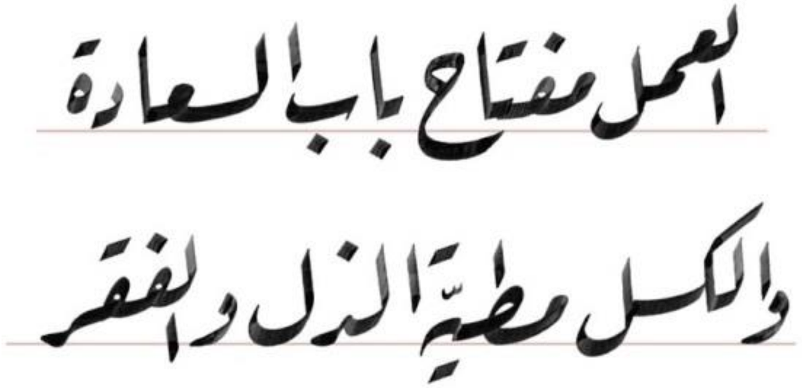
- Fit lists into input: simplify or add 0’s.
- Train neural network with curves …

It’s not that easy …

IMPROVEMENTS ON THE NEURAL NETWORK

Example


وَإِنَّ لَكُمْ فِي ٱلْأَنْعَٰمِ لَعِبْرَةً نُّسْقِيكُم مِّمَّا فِي بُطُونِهِۦ
مِنۢ بَيْنِ فَرْثٍ وَدَمٍ لَّبَنًا خَالِصًا سَآئِغًا لِّلشَّٰرِبِينَ


العمل مفتاح باب السعادة
والكسل مطيّة الذل والفقر

Naskhi (Copying) font

Ruqa’i (Sheet) font

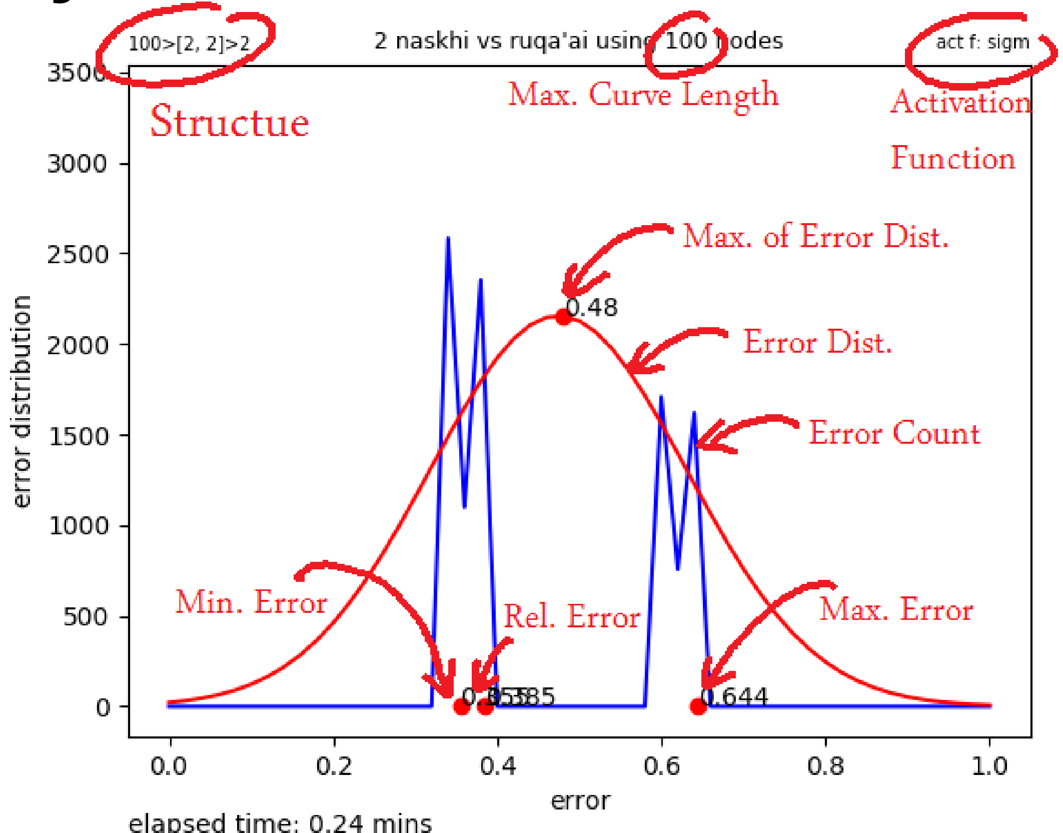
TU Clausthal

Example: Naskhi vs. Ruqa’i

- Input: Bézier-curves (100 control points).
- Output: (x, y)
  - x: Possibility of being in Naskhi.
  - y: Possibility of being in Ruqa’i.
- Error:
  - Expected output: $(1,0) \rightsquigarrow \max\{|x-1|, |y-0|\}$.
  - Expected output: $(0,1) \rightsquigarrow \max\{|x-0|, |y-1|\}$.

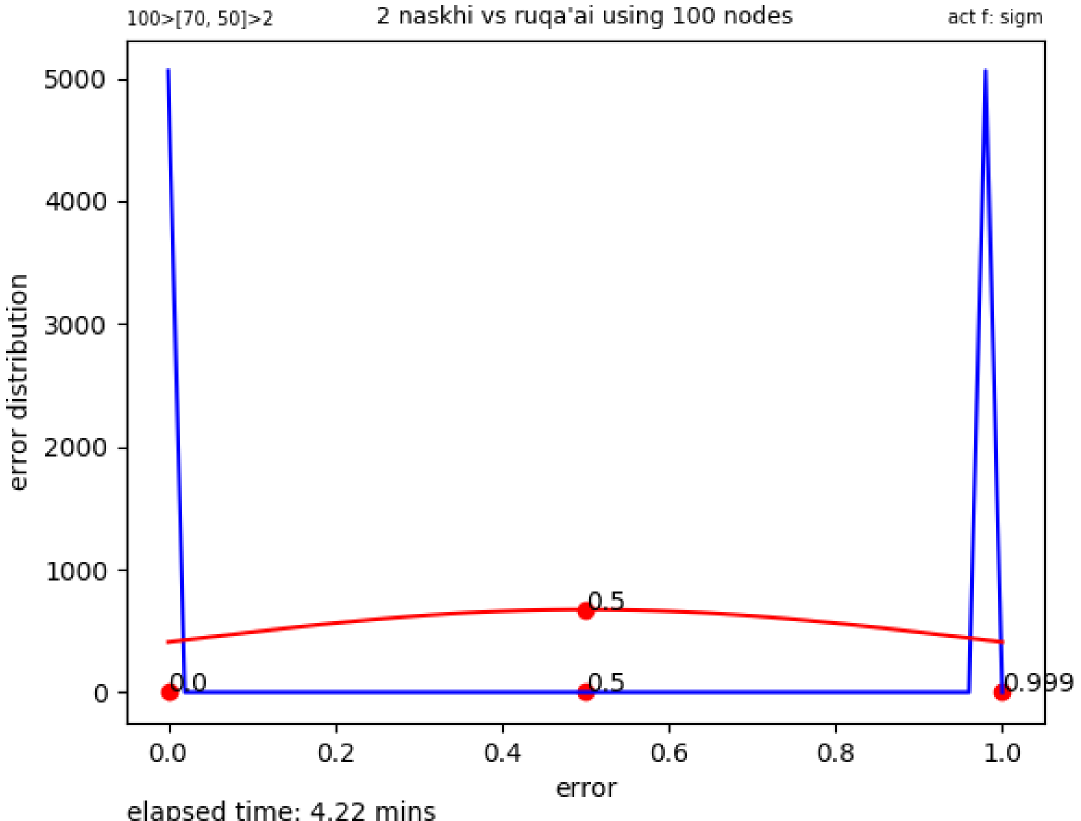
TU Clausthal

First Try

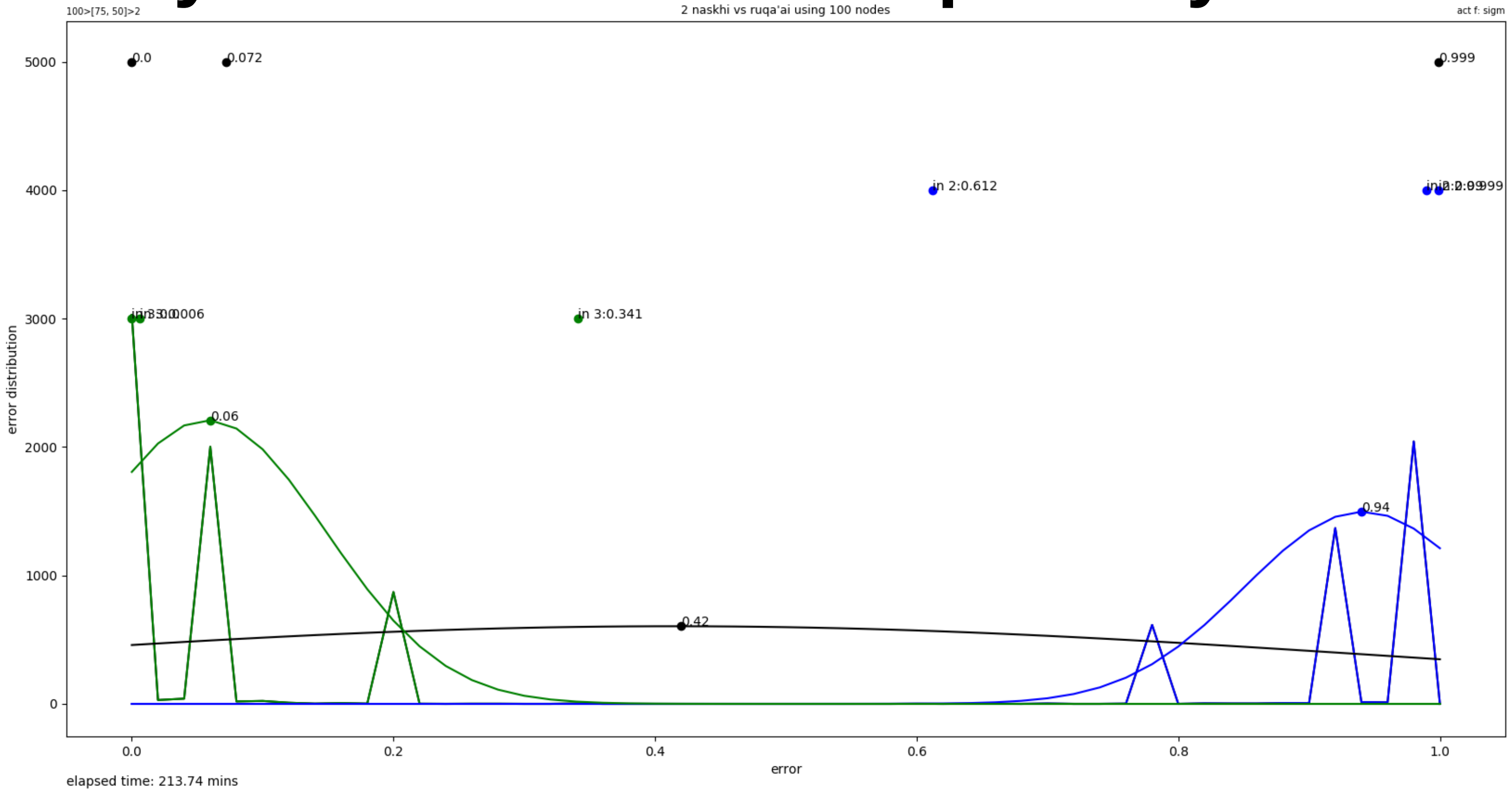


elapsed time: 0.24 mins

Change the Structure of Hidden Layers

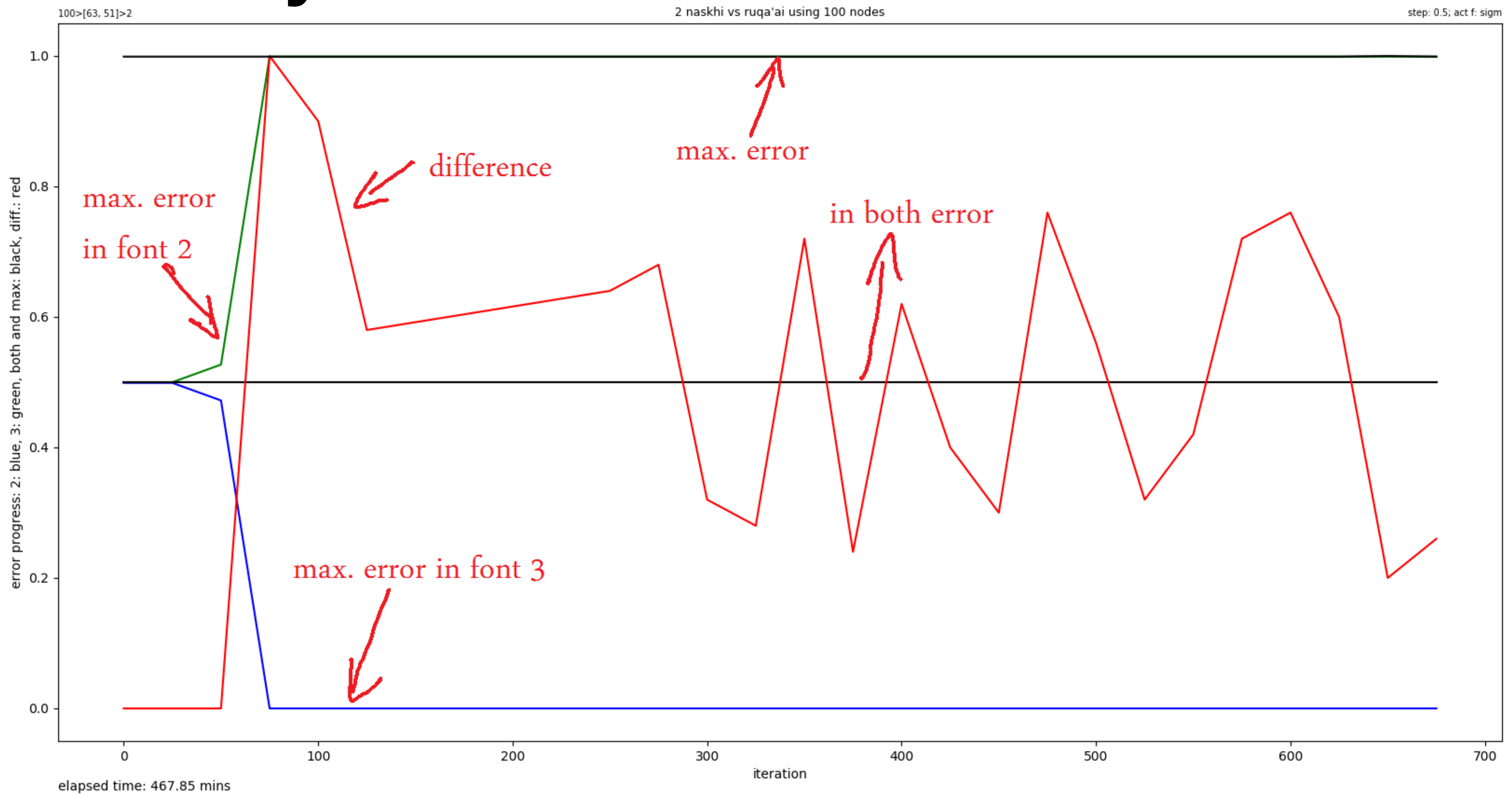


Analyse Error in Fonts Separately



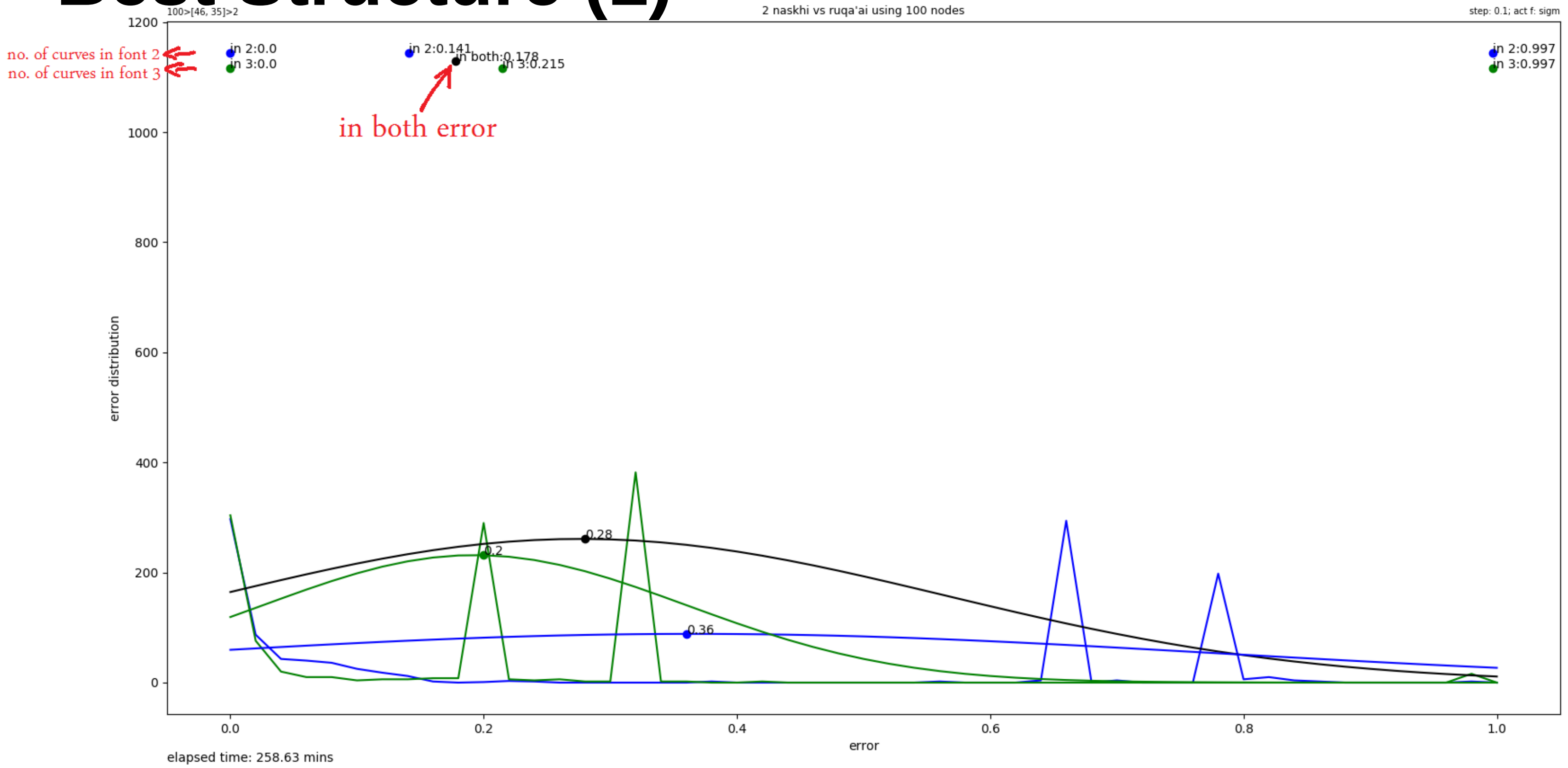
TU Clausthal

Learn by Heart



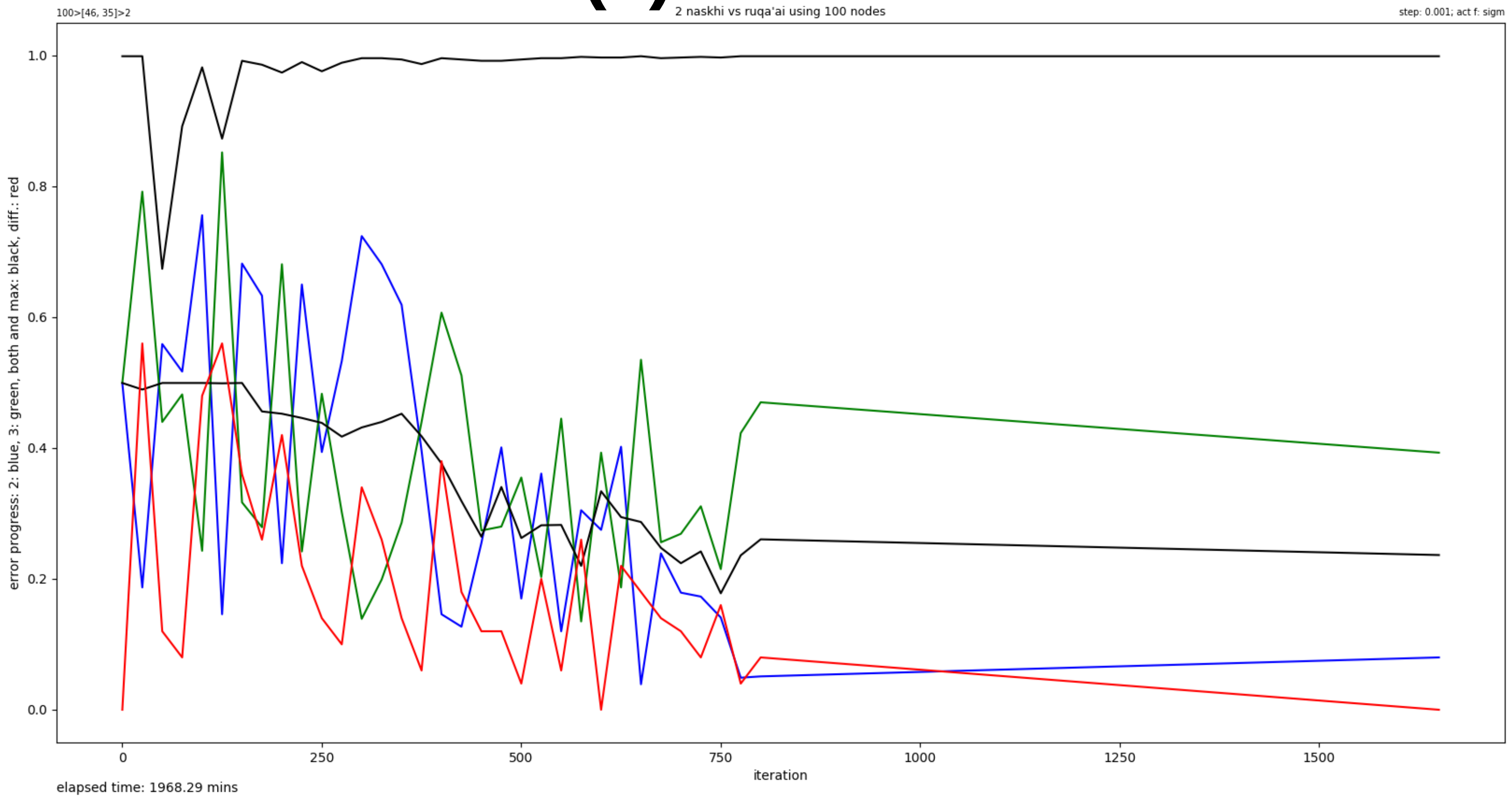
TU Clausthal

Best Structure (1)



TU Clausthal

Best Structure (2)



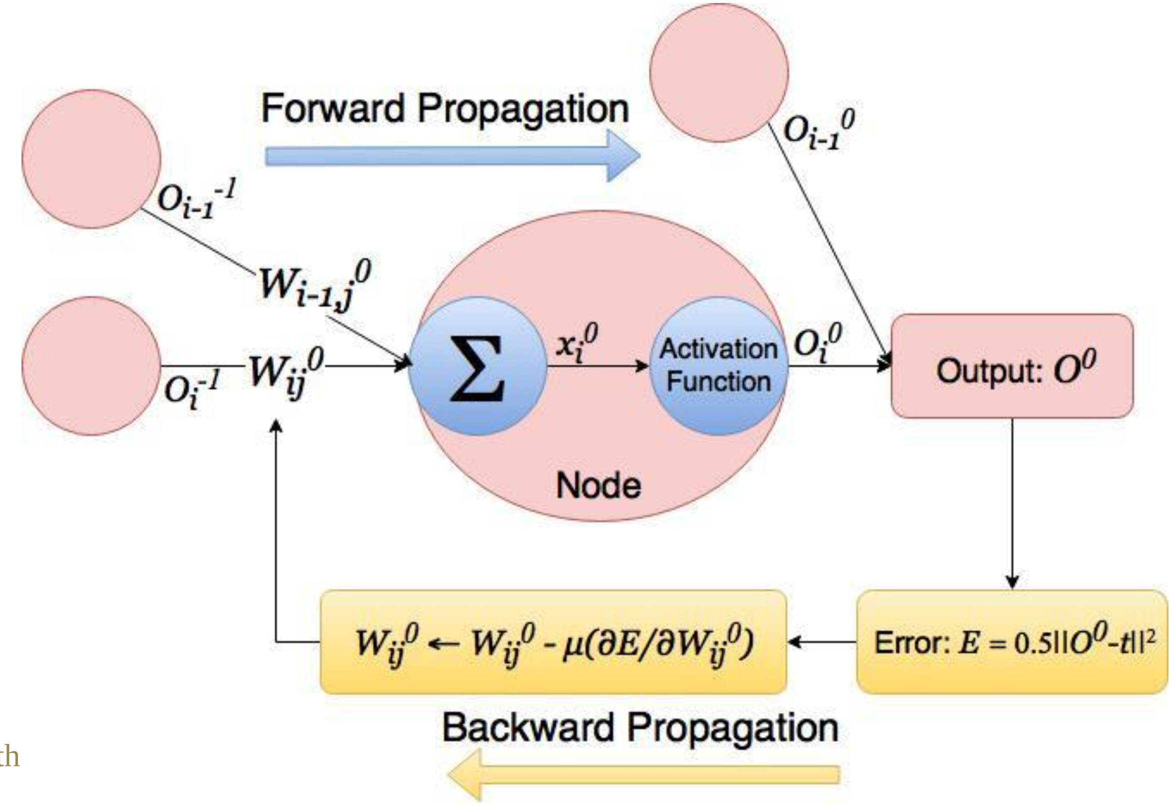
Results About Structure

- Best structures: 2 layers, $[n, \frac{3}{4}n]$.
- Big structures (>70 neurons) $\rightsquigarrow$ learn by heart.
- Small structures (<10 neurons) $\rightsquigarrow$ no evolution.

Change Dimensions

- 125 control points (simplify less):
  - Slower, same accuracy.
- 75 control points (simplify more):
  - Faster, worse accuracy.
- +16 line points (respect lines):
  - Slower, worse accuracy.
- So: 100 control points, only curves.

Change Step (1)



Change Step (2)

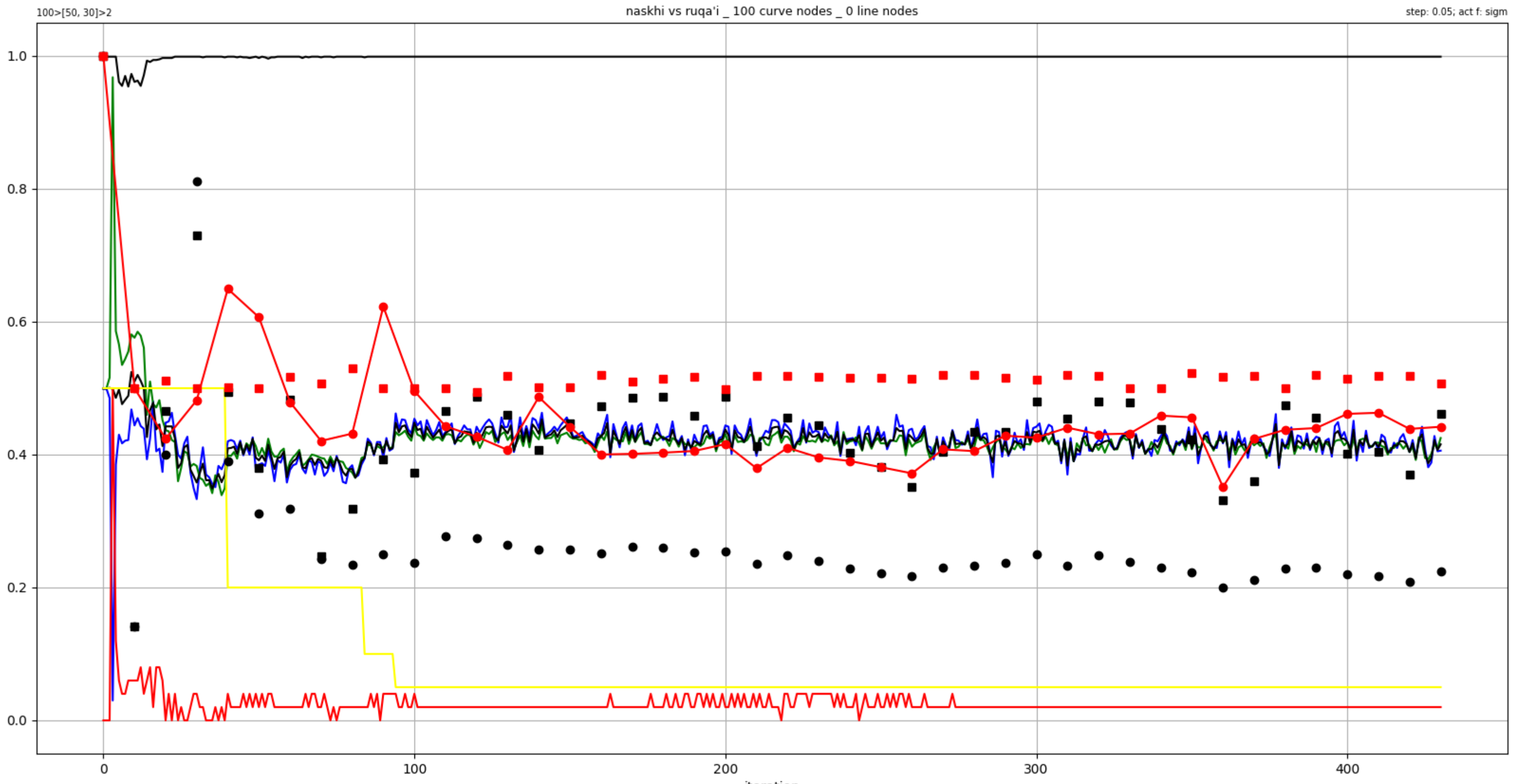
- Big step $\rightsquigarrow$ Fluctuation.
- Small step $\rightsquigarrow$ No evolution.
- So:
  - Fluctuation $\rightsquigarrow$ Make step smaller.
  - No evolution $\rightsquigarrow$ Make step bigger.

Interpretation of the Result

- Full Result of a photo p:
  - $\mathrm{mean}\{\mathrm{NN}(\mathrm{c}) : \mathrm{c} \in Cu(p)\} = (x, y)$.
- Full Error:
  - Expected output: $(1{,}0) \;\leadsto\; \max\{|x - 1|, |y - 0|\}$.
  - Expected output: $(0{,}1) \;\leadsto\; \max\{|x - 0|, |y - 1|\}$.

TU Clausthal

Add Testing Photos



Representativeness

- Order on curves.
- For $\mathrm{c}_1, \mathrm{c}_2 \in Cu(p)$ in a photo p :
  - Let $\mathrm{NN}(\mathrm{c}_1) =: (x_1, y_1)$, $\mathrm{NN}(\mathrm{c}_2) =: (x_2, y_2)$
  - $\mathrm{c}_1 \prec_{\mathbf{repr.}} \mathrm{c}_2 \;:\Leftrightarrow |x_1 - y_1| < |x_2 - y_2|$

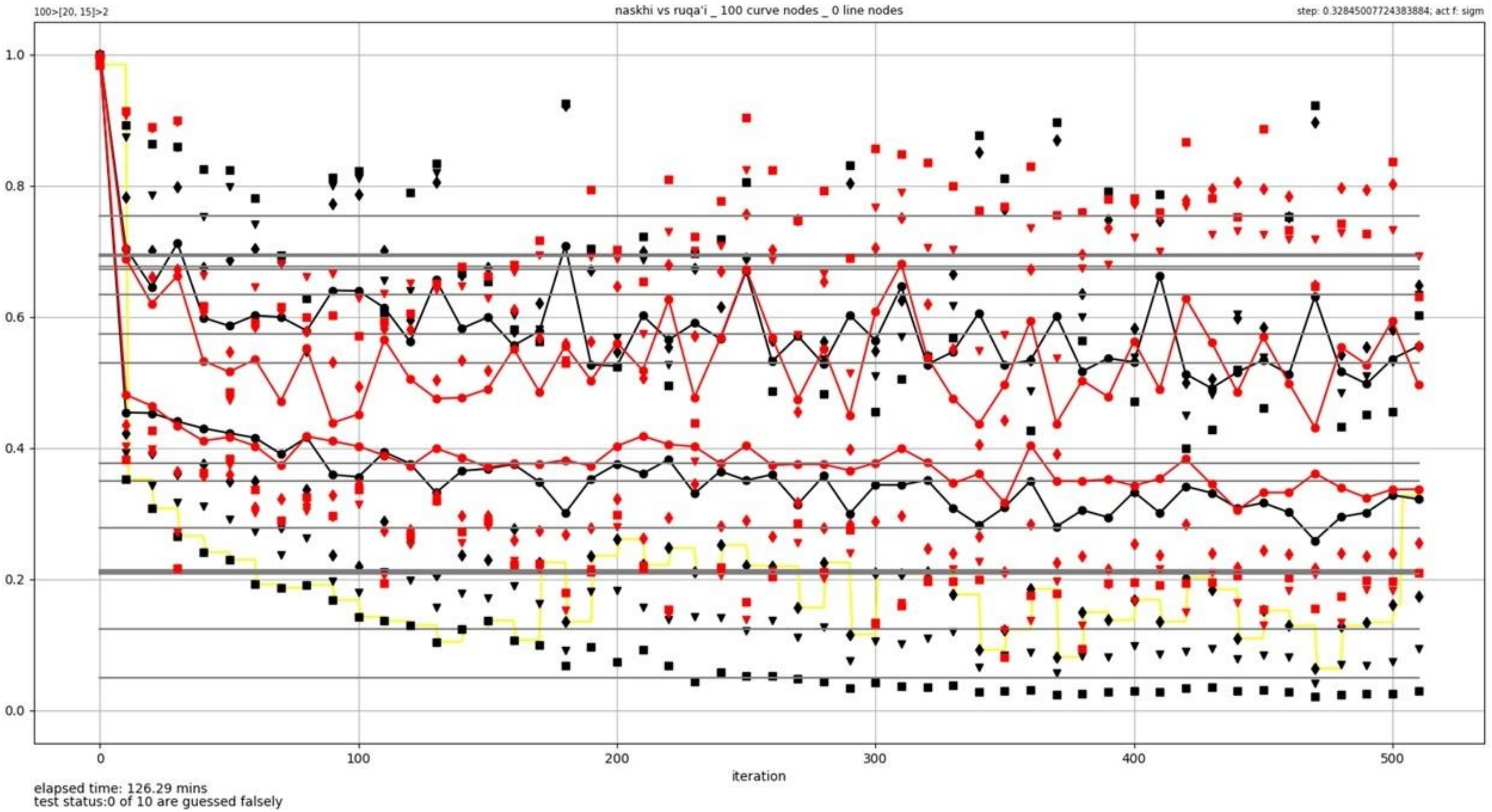
$(\boldsymbol{n})$Best Full Result

1. Order $Cu(p)$ with $\prec_{\mathbf{repr.}}$.
2. Take the first $^1/_n\, th$ curves $\leadsto A$.
3. $\mathrm{mean}\{\mathrm{NN}(\mathrm{c}) : \mathrm{c} \in A\} = (x, y)$.

- (n)Best Full Error:
  - Expected output: $(1,0) \leadsto \max\{|x-1|, |y-0|\}$.
  - Expected output: $(0,1) \leadsto \max\{|x-0|, |y-1|\}$.

TU Clausthal

Different Interpretations of the Result



elapsed time: 126.29 mins
test status:0 of 10 are guessed falsely

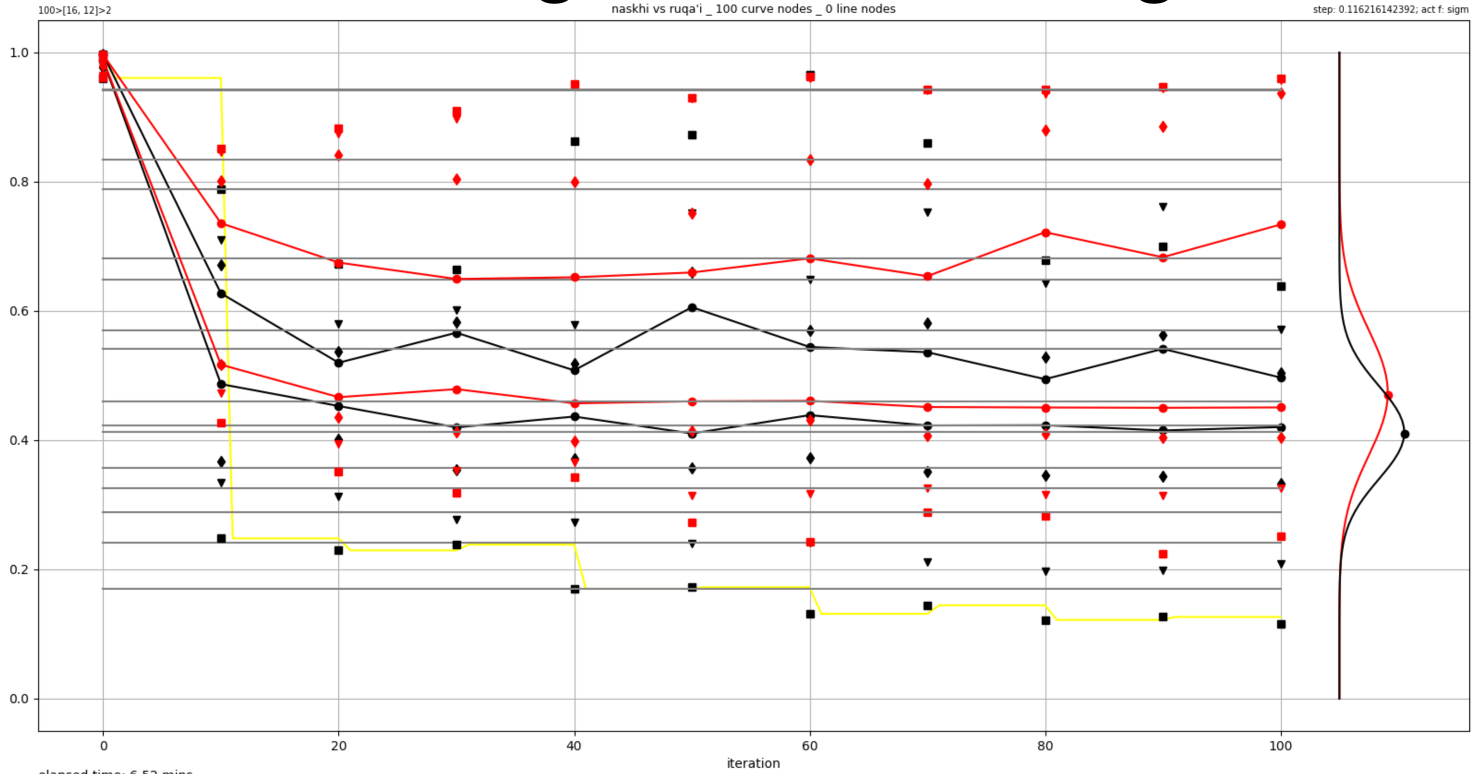
$(\boldsymbol{x}, \boldsymbol{y}) \leadsto (\mathbf{1}, \mathbf{0})$

- (1,0) if $x > 0.5$ and $y < 0.5$
- (0,1) if $x < 0.5$ and $y > 0.5$
- So: error $< 0.5 \leadsto$ Font is correctly recognized!
- Recall error:
  - Expected output: $(1,0) \leadsto \max\{|x-1|, |y-0|\}$.
  - Expected output: $(0,1) \leadsto \max\{|x-0|, |y-1|\}$.

Ratio: Training Photos:Testing Photos (1)

- More training photos $\leadsto$ easier.
- More testing photos $\leadsto$ harder.
- Before: 89:10 $\simeq$ 9:1
- Let’s try 64:32 $\simeq$ 2:1 …

Ratio: Training Photos:Testing Photos (2)



```
elapsed time: 6.52 mins
test status 1:9 of 32 are guessed falsely
test status 2:9 of 32 are guessed falsely
test status 3:8 of 32 are guessed falsely
test status 4:9 of 32 are guessed falsely
```

More?

- Optimization?
  - Simplify and make curves longer in another way.
  - Read off equations from SVG and use these as input.
- Further?
  - Recognize characters using curves $\rightsquigarrow$ read text of a calligraphy art work (connected, tangled).

TU Clausthal

THANKS FOR YOUR ATTENTION

Faculty of Mathematics, Computer Science and Engineering