Computing Midcurve by Neural Networks

Yogesh Kulkarni

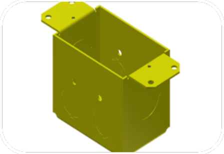
February 17, 2020

Agenda

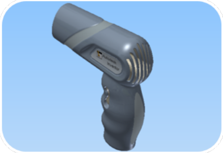
MidcurveNN : Encoder-Decoder Neural Network for Computing Midcurve of a Thin Polygon

Introduction

Aerospace

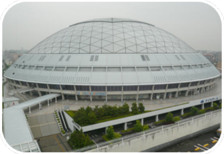


Machinery

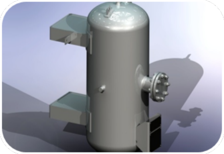


Consumer Products

Energy



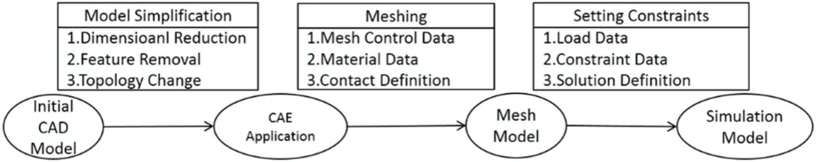
Construction



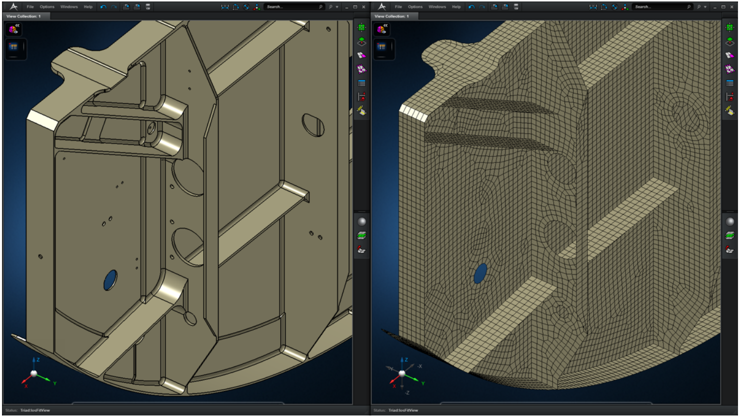
Industrial equipment

Can we use shapes directly?

- CAD : Designing Shapes
- CAE : Engineering Analysis
- CAD→CAE: Simplification for quicker results.



CAD-CAE

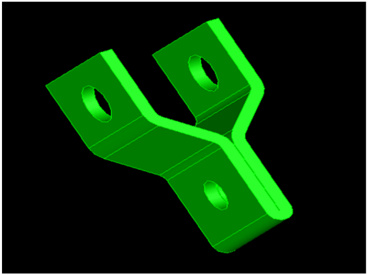


For Shapes like Sheet Metal . . .

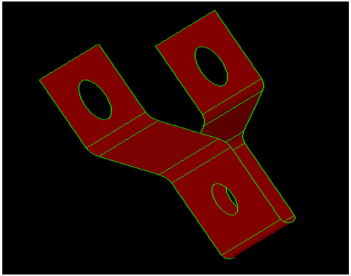
	Solid mesh	Shell+Solid mesh	Difference (%)
Element number	344,330	143,063	-58%
Node Number	694,516	75,941	-89%
Total Degrees of freedom	2,083,548	455,646	-78%
Maximum Von. Mises Stress	**418.4 MPa**	**430 MPa**	+3%
Meshing + Solving time	Out of memory	22 mins	N/A (4G RAM)
Meshing + Solving time	**30 mins**	**17 mins**	-43% (12G RAM)

Half the computation time, but similar accuracy

Midsurface is?



Input: Solid

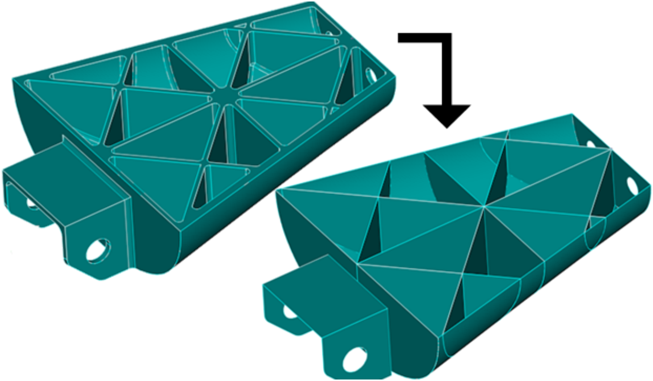


Output: Midsurface

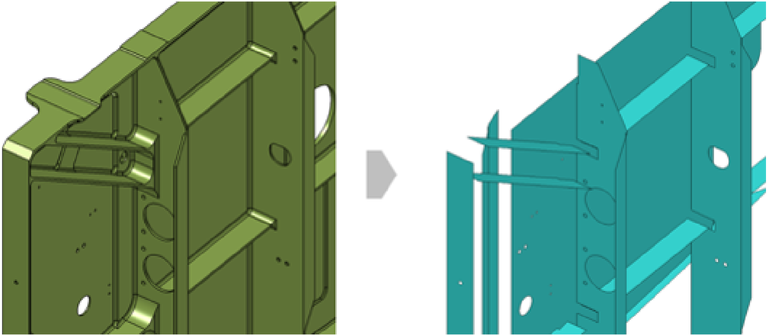
- Widely used for CAE of Thin-Walled parts
- Computation is challenging and still unsolved

Getting Midsurface

- Going on for decades . . .
- Manually by offsetting and stitching, initially
- Many CAD-CAE packages give automatic option, but . . .



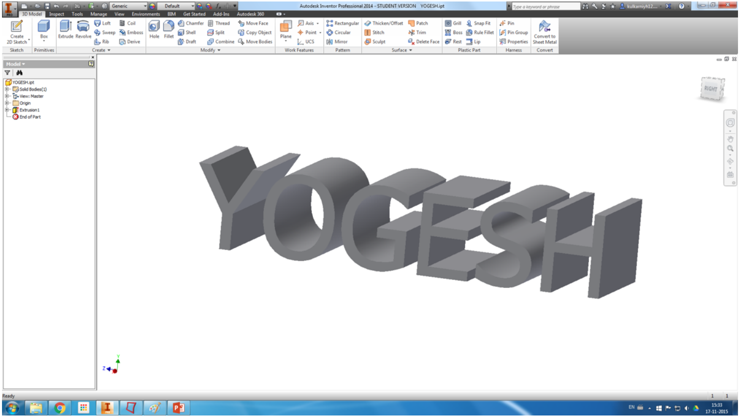
Look at the output



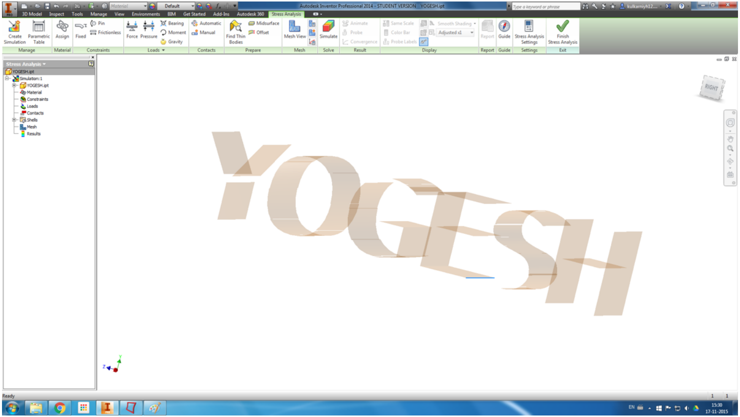
Can't tolerate gaps

- We have thickness sampling,
- To recreate-represent the original shape
- Input and output difference not desirable

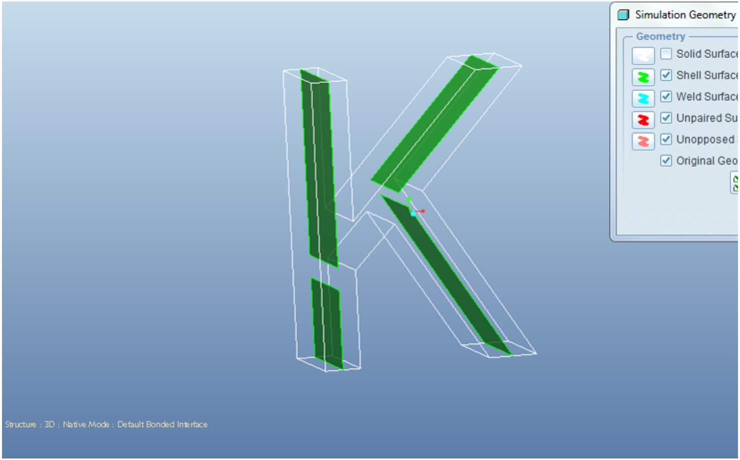
For a simple model like



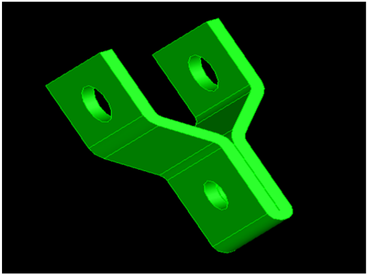
You get



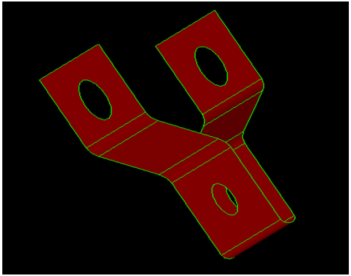
For a far simpler shape



Current Quality



Input: Solid

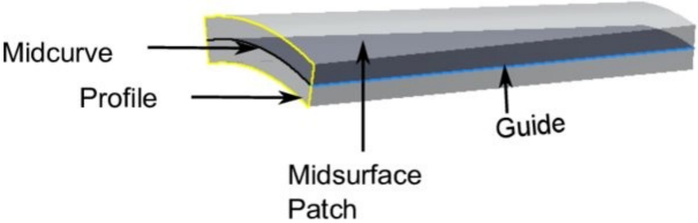


Output: Midsurface

- Errors take weeks to correct for complex parts.
- But still preferred, due to vast savings time
- From Days to hours . . .

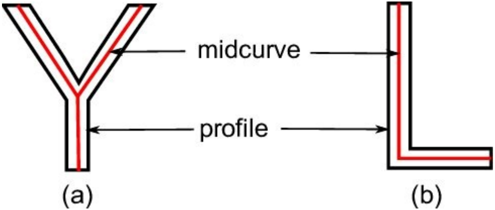
Midsurface Computation

- Midsurface of a Patch is Midcurve of its profile extruded.
- So, it boils down to computing 1D midcurve of a 2D profile



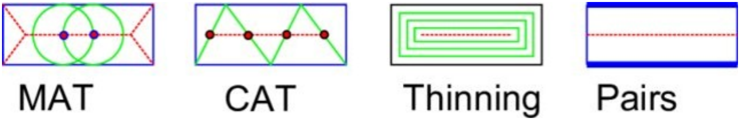
What is a Midcurve?

- Midsurface : From 3D thin Solid to 2D Surface
- Midcurve : From 2D Profile to 1D Curve

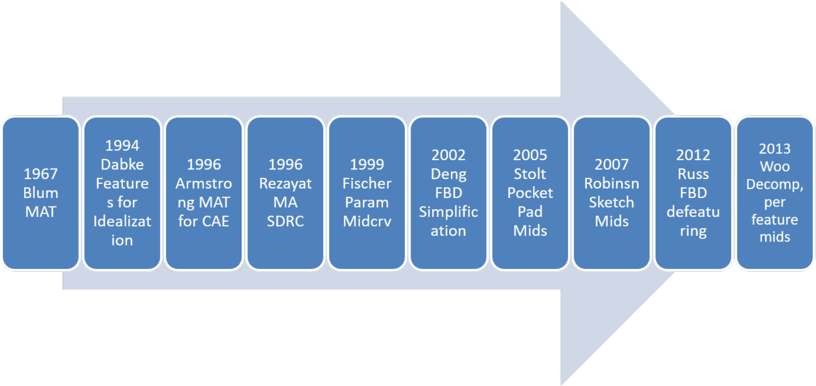


Many Approaches

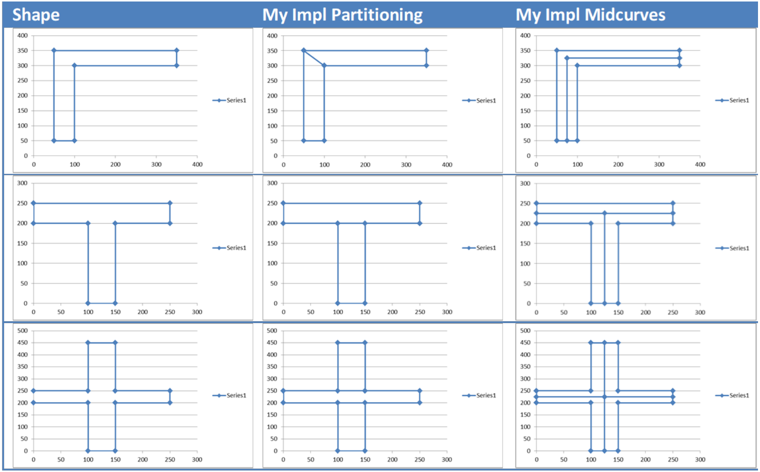
- More than 6 decades of research. . .
- Most CAD-CAE packages. . .
- Rule-based!! Heuristic!! Case-by-case basis!!



When-What?


1967 Blum MAT
1994 Dabke Features for Idealization
1996 Armstrong MAT for CAE
1996 Rezayat MA SDRC
1999 Fischer Param Midcrv
2002 Deng FBD Simplification
2005 Stolt Pocket Pad Mids
2007 Robinsn Sketch Mids
2012 Russ FBD defeaturing
2013 Woo Decomp, per feature mids

2017: My PhD Work: Rule-based



Limitations

- Fully rule-based
- Need to adjust for new shapes
- So, not scalable

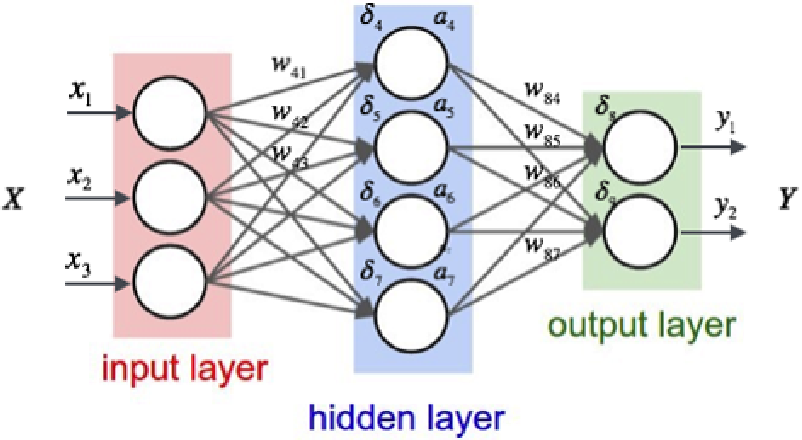
Idea

Can Neural Networks "learn" the dimension reduction transformation?

How?

- Supply lots of training data of profiles and their corresponding midcurves and train.
- Then given an unseen profile, can Neural Network compute a midcurve, mimicking the original profile shape?

Midcurve by Neural network

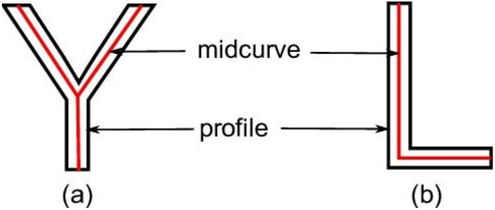


Midcurve : The Problem

- **Goal**: Given a 2D closed shape (closed polygon) find its midcurve (polyline, closed or open)
- **Input**: set of points or set of connected lines, non-intersecting, simple, convex, closed polygon
- **Output**: another set of points or set of connected lines, open/branched polygons possible

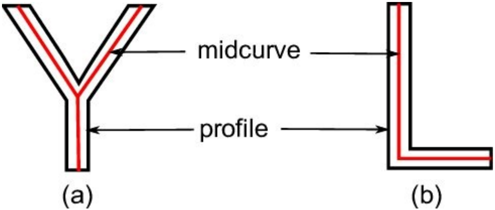
Midcurve == Dimension Reduction

- Like PCA (Principal Component Analysis), wish to find Principal curve
- That 'represents' the original profile shape



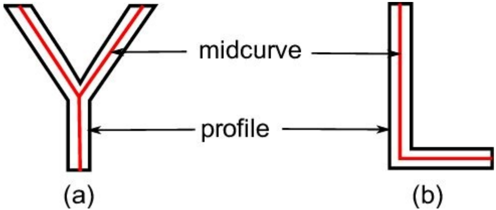
Midcurve == Translation

- Left side (input): 2D Sketch Profile
- Right Side (output): 1D Midcurve
- Sequence 2 Sequence problem



Midcurve ! = Auto-Encoder Decoder

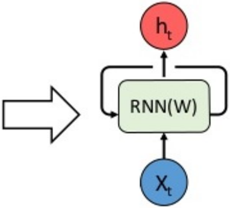
- Its not Auto-Encoder as Input and Output are different
- Its not fixed size i/o as Input and Output sizes are different



Variable Size Encoder Decoder

- Batches need fixed lengths
- Made fixed size by Padding.

Friendly	against	Scotland	at	Murray	.
Nadim	Ladki	<PAD>	<PAD>	<PAD>	<PAD>
AL-AIN	United	Arab	Emirates	<PAD>	<PAD>
ROME	1996-12	<PAD>	<PAD>	<PAD>	<PAD>
Two	goals	in	the	last	minutes



Variable Size Encoder Decoder

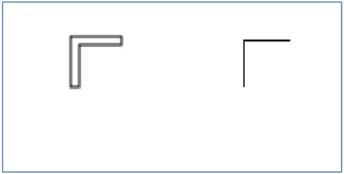
- OK for NLP, say Machine Translations, where padding values like “-1” can be added along with other words (vectors or indices)
- But in Geometry, its not OK.
- Because any value can represent a Valid Input, even though we don’t want it to be the input.

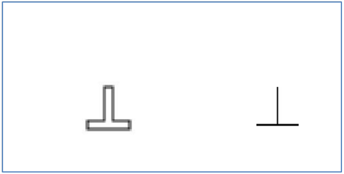
A Twist to the problem

- Till we get good variable size encoder decoder network for geometry. . .
- Decided to convert this Sequence 2 Sequence problem as Image 2 Image problem.

A Twist to the problem

- Input: Black & White Image of 2D profile
- Output: Black & White Image of 1D midcurve





Solves . . .

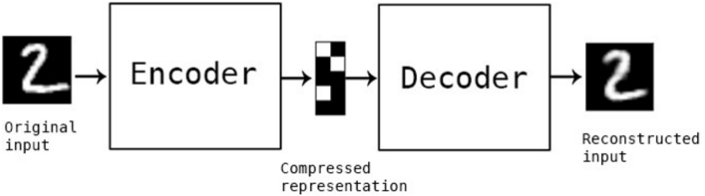
Problems of Geometric sequences

- Variable input/output sizes
- Loops need to be crossed
- Branches

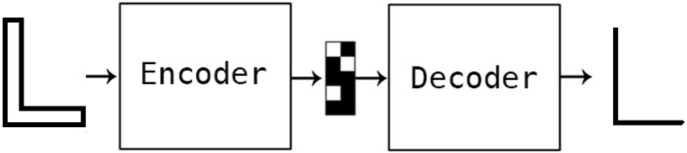
I L T X K O Y

H U S D Q W

Reuse Image Encoder Decoder



For Dimension Reduction



For Deep Learning

- Need lots of data
- Had just few input output image pairs
- How to augment/populate large variations ...

Data Preparation

Data

Original input and output are in the form of polylines, meaning a list of points, each having x,y coordinates

Profile Data	Profile Picture	Midcurve Data	Midcurve Picture
5.0 5.0		7.5 5.0	
10.0 5.0		7.5 32.5	
10.0 30.0		35.0 32.5	
35.0 30.0		7.5 32.5	
35.0 35.0			
5.0 35.0			

Data

Profile Data		Profile Picture	Midcurve Data		Midcurve Picture
0	25.0		12.5	0	
25.0	25.0		12.5	22.5	
25.0	20.0		25.0	22.5	
15.0	20.0		0	22.5	
15.0	0				
10.0	0				
10.0	20.0				
0	20.0				

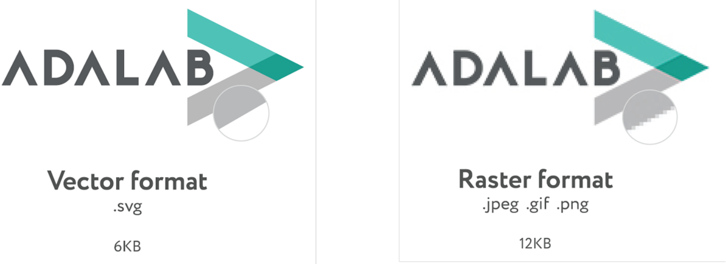
- For each shape, we have this pair of input and output. That's it.
- We need to start with these few samples only

Augmentation

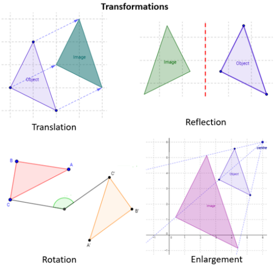
- Such few profile shapes, are just not enough for Neural Networks to train.
- Need more with as much diversity as possible.
- Will need to artificially augment data with transformations, like pan, rotate, mirror, etc.
- All needs to be automatically, programmatically

Geometry to Image

- Raw input data is in the Vector format
- Converted it to fixed size (100x100) image by rasterization of drawSVG library.

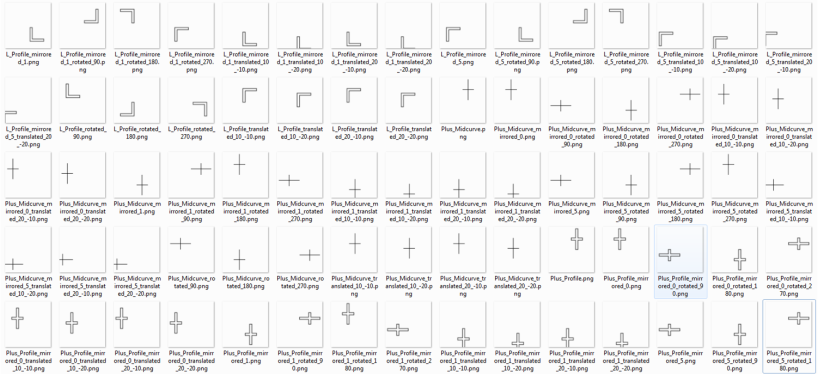


Variations



- Inputs: I, L, Plus, T
- Operations:
  - Translated
  - Rotated
  - Mirrored
  - Mirrored Translated
  - Mirrored Rotated
- Total: 896 images (still less, but not bad)

Training Data Samples



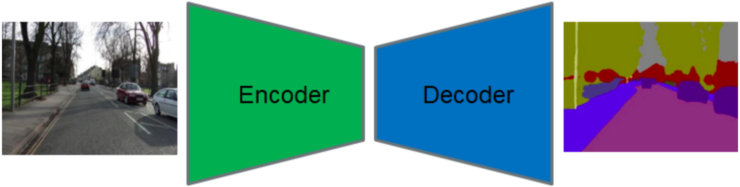
Midcurve By Neural Network

Options For Architectures

- Simple Encoder Decoder (one layer each)
- Dense Encoder Decoder
- Convolutional Encoder Decoder
- Pix2Pix
- ...

Simple Encoder Decoder

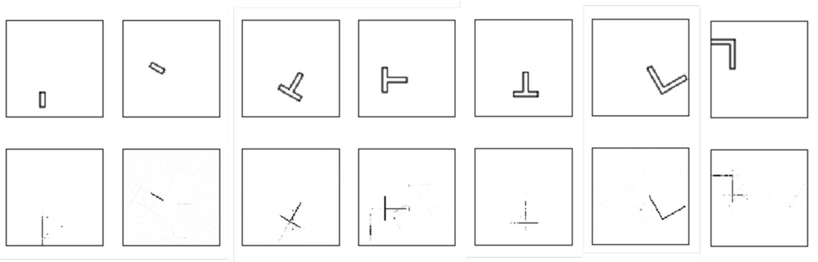
Simple Encoder Decoder



Keras Implementation

```
input_img = Input(shape=(input_dim,))

encoded = Dense(encoding_dim,
      activation='relu',activity_regularizer=regularizers.l1(10e-5))(input_img)
decoded = Dense(input_dim, activation='sigmoid')(encoded)

autoencoder = Model(input_img, decoded)

encoder = Model(input_img, encoded)
encoded_input = Input(shape=(encoding_dim,))
decoder_layer = autoencoder.layers[-1]
decoder = Model(encoded_input, decoder_layer(encoded_input))

autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')
```

Results



Results

- Not very perfect but encouraging
- NN is correct with
  - The location (bounding box)
  - Dimension Reduction is seen
- But, still some stray points and misses

What can be done?

- For the noise, use bounding boxes
- Feedback into error term: differencing with the known output expected
- Classify single pixel image as the skeleton, and rest as noise.

What Next?

- Add denoiser network after the current one
- More Network Architectures
- Sequence-to-Sequence based approaches, taking closed thin polygon as input and polyline as output
- Extending to 3D, ie Midsurface

End Notes

Summary

- Various applications need lower dimensional representation of shapes.
- Midcurve is one- dimensional(1D) representation of a two-dimensional (2D) planar shape.
- Used in animation, shape matching, retrieval, finite element analysis, etc.

Summary

- Approaches: Thinning, Medial Axis Transform (MAT), Chordal Axis Transform (CAT), Straight Skeletons, etc., all of which are rule-based.
- Proposing a novel method called MidcurveNN which uses Encoder-Decoder neural network for computing midcurve from images of 2D thin polygons in supervised learning manner.

Summary

- This dimension reduction transformation from input 2D thin polygon image to output 1D midcurve image is learnt by the neural network,
- Which can then be used to compute midcurve of an unseen 2D thin polygonal shape.

Agenda

1. Introduction to Midcurve
2. The End

References

- xxx

Thanks ... yogeshkulkarni@yahoo.com