MidcurveNN

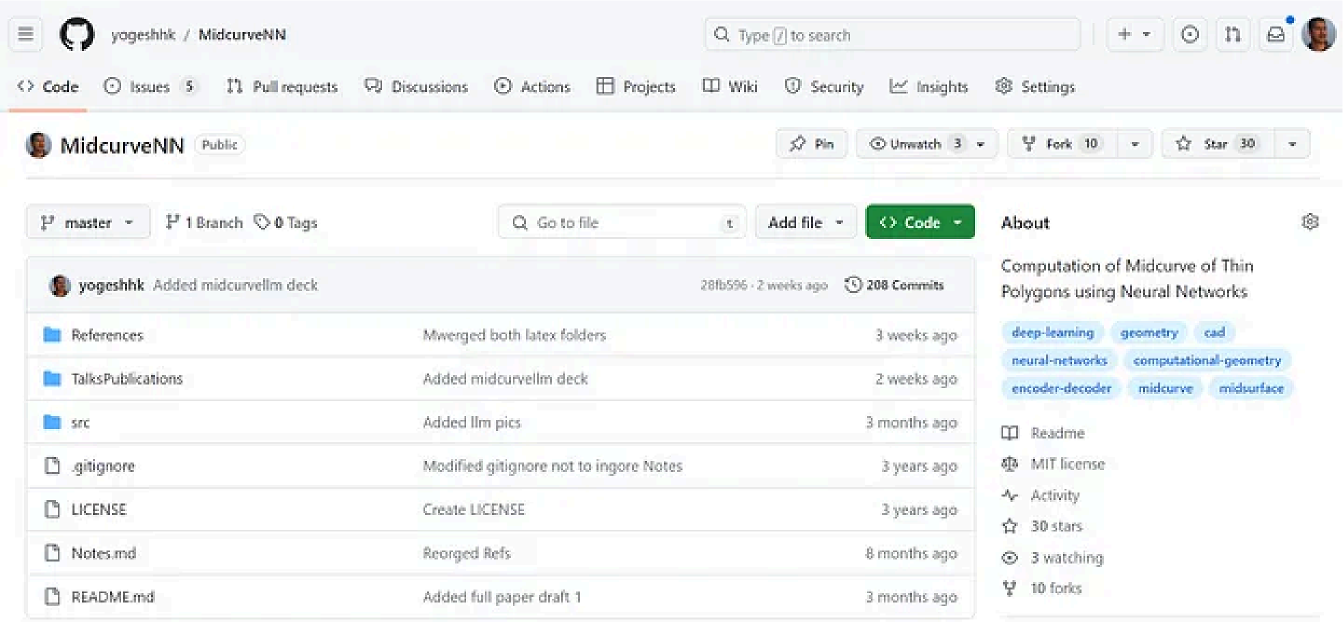
An open-source repository for generating Midcurve by Neural Networks

Yogesh Haribhau Kulkarni (PhD)

Published in Technology Hits

6 min read · 20 hours ago

Listen Share More



yogeshhk Added midcurvellm deck 28fb596 · 2 weeks ago 208 Commits

References	Mwerged both latex folders	3 weeks ago
TalksPublications	Added midcurvellm deck	2 weeks ago
src	Added llm pics	3 months ago
.gitignore	Modified gitignore not to ingore Notes	3 years ago
LICENSE	Create LICENSE	3 years ago
Notes.md	Reorged Refs	8 months ago
README.md	Added full paper draft 1	3 months ago

Computation of Midcurve of Thin Polygons using Neural Networks

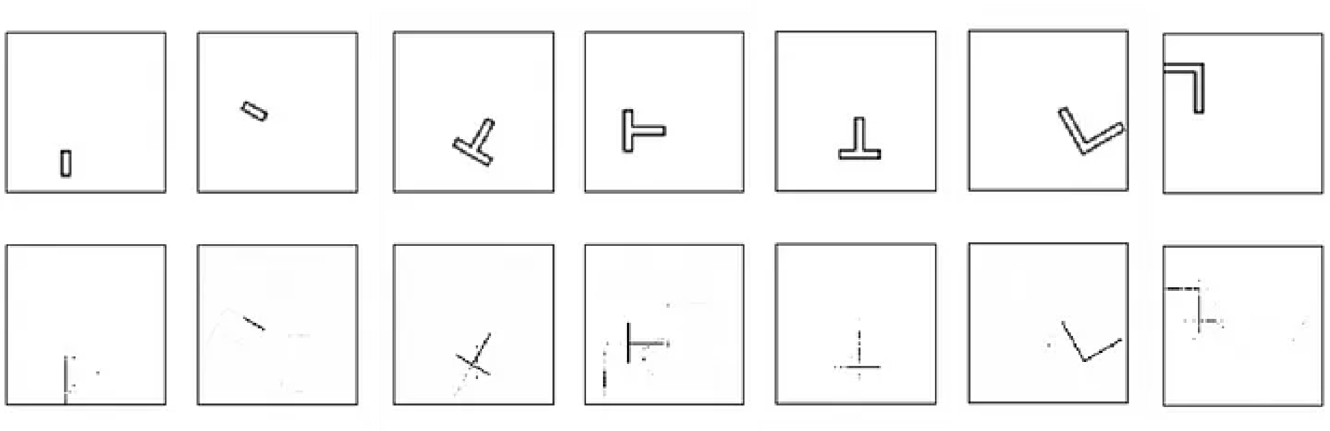
Screenshot of the repository

MidcurveNN is a project aimed at solving the challenging problem of finding the midcurve of a 2D closed shape using neural networks. The primary goal is to transform a closed polygon, represented by a set of points or connected lines, into another set of points or connected lines, allowing for the possibility of open or branched polygons in the output.

Instructions to Run

```
cd src
conda env create -f environment.yml
```

```
activate midcurvenn
python simpleencoderdecoder\main_simple_encoderdecoder.py
```



Results of Simple Encoder Decoder Neural Network

Thoughts

Graph Summarization/Dimension-Reduction/Compression: Reducing a large graph to a smaller graph preserving its underlying structure, similar to text summarization, which attempts to keep the essence.

Representation Issue

- **Shapes cannot be modeled as sequences:** Although a polygon shape may appear as a sequence of points, it is not truly sequential.
- **Not all shapes can be drawn without lifting a pencil:** Some shapes, like the letter “Y” or concentric circles, cannot be accurately represented as sequences. Therefore, modeling the Midcurve transformation as a Sequence-to-Sequence network is not feasible.
- **Challenges in representing geometric figures:** How can we represent a geometric figure numerically for use in machine/deep learning models? Especially when dealing with 2D linear profile shapes, converting them to vectors is nontrivial.
- **Graphs vs. geometric structures:** While graphs are a natural choice for representing connectivity, they lack spatial position information. Graph Neural Networks, which convolve neighbors around nodes and pool outputs, are not ideal for this task.
- **Research and development needed:** We require innovative approaches to generate geometric-graph embeddings that consider both node coordinates and

arc geometry. The challenge lies in formulating pooling mechanisms that aggregate information from incident curves and node coordinates into a meaningful representation.

Variable Lengths Issue

This is a dimension-reduction problem wherein, in 2D, the input constitutes the sketch profile (parametrically 2D), while the output represents the midcurve (parametrically 1D). Input points are ordered, mostly forming a closed loop known as a manifold. Conversely, output points may lack a defined order and can exhibit branches, referred to as non-manifold structures.

It poses a variable input and variable output challenge, given that the number of points and lines differs in the input and output.

This presents a network 2 network problem (not Sequence to Sequence) with variable-sized inputs and outputs. For Encoder-Decoder networks like those in Tensorflow, fixed-length inputs are necessary. Introducing variable lengths poses a significant hurdle as padding with a distinct unused value, such as 0,0, is not feasible, considering it could represent a valid point.

The limitation with Seq2Seq models is notable; the input polygons and output branched midcurves are not linearly connected. They may exhibit loops or branches, necessitating further consideration for a more suitable solution. (More details on limitations below)

Instead of adopting a point list as input/output, let’s consider the well-worked format of images. These images are standardized to a constant size, say 64x64 pixels. The sketch profile is represented as a color profile in a bitmap (black and white only), and similarly, the midcurve appears in the output bitmap. This image-based format allows for the application of LSTM encoder-decoder Seq2Seq models. To diversify training data, one can introduce variations by shifting, rotating, and scaling both input and output. The current focus is on 2D sketch profiles, limited to linear segments within a single, simple polygon without holes.

The vectorization process involves representing each point as 2 floats/ints. Consequently, the total input vector for a polygon with ‘m’ points is 2m floats/ints. For closed polygons, the first point is repeated as the last one. The output is a vector of 2n points, repeating the last point in the case of a closed figure. Preparing training data involves using data files from MIDAS. For scalability, input and output can be scaled with different factors, and entries can be randomly shuffled. Determining the maximum number of points in a profile establishes a fixed length for both input and output.

Resources and further exploration for Seq2Seq models can be found at Tensorflow Seq2Seq Tutorial, Seq2Seq Video Tutorial, A Neural Representation of Sketch Drawings, and Sketch RNN GitHub Repository.

Additionally, consider incorporating plotting capabilities to visualize polygons and their midcurves, facilitating easy debugging and testing of unseen figures.

Dilution to Images

Addressing both representation and variable-size issues, the dilution to images introduces a crucial aspect. However, a notable limitation is the discrepancy between true geometric shapes, akin to vector images, and the raster-type images utilized in this context. Approximation becomes inevitable.

Post-modeling, the predicted output requires post-processing to align with the geometric form, presenting a challenging task. Consequently, the project is bifurcated into two distinct phases:

Phase I: Image to Image Transformation Learning

- Img2Img: Input/output fixed-size 100x100 bitmaps.
- Populate numerous instances by scaling, rotating, and translating both input and output shapes within the fixed size.
- Utilize Encoder-Decoder structures like Semantic Segmentation or Pix2Pix of Images to learn dimension reduction.

Phase II: Geometry to Geometry Transformation Learning

- Construct both input and output polyline graphs with (x, y) coordinates as node features and edges with node ID pairs.
- For poly-lines (lines without curves), no need to store geometric intermediate points as features; for curves, store sampled fixed 'n' points.
- Develop an Image-Segmentation-like Encoder-Decoder network, employing Graph Convolution Layers from DGL instead of the typical Image-based 2D

convolution layer, within the PyTorch encoder-decoder model.

- Generate a variety of input-output polyline pairs using geometric transformations (in contrast to image transformations in Phase I).
- Investigate the potential contributions of Variational Graph Auto-Encoders (VGAE), as demonstrated in DGL's PyTorch examples.

While Phase I has been implemented in a simplistic manner, Phase II aims to identify an optimal representation for storing geometry, graphs, and networks as text. This facilitates the application of Natural Language Processing (NLP) techniques. A geometry representation akin to that found in 3D B-rep (Boundary representation) but in 2D is leveraged, as demonstrated below:

```
{
        'ShapeName': 'I',
        'Profile': [(5.0, 5.0), (10.0, 5.0), (10.0, 20.0), (5.0, 20.0)],
        'Midcurve': [(7.5, 5.0), (7.5, 20.0)],
        'Profile_brep': {
                'Points': [(5.0, 5.0), (10.0, 5.0), (10.0, 20.0), (5.0, 20.0)]
                'Lines': [[0, 1], [1, 2], [2, 3], [3, 0]],
                'Segments': [[0, 1, 2, 3]]
        },
        'Midcurve_brep': {
                'Points': [(7.5, 5.0), (7.5, 20.0)],
                'Lines': [[0, 1]],
                'Segments': [[0]]
        },
}
```

The structure provides detailed information about the shape, profile, midcurve, and their boundary representation (Brep). The use of text-based methods over image-based methods is advantageous, particularly in precise point representation, facilitating tasks such as line removal. For a broader discussion on the application of Deep Learning in Computer-Aided Design (CAD), refer to Notes.

Conclusion

MidcurveNN presents a novel approach to solving the problem of finding midcurves in 2D closed shapes using neural networks. While the project is still in development and faces various challenges in representation and variable lengths, it provides valuable insights into the intersection of geometry and machine learning. Future work will focus on refining the methodologies proposed and addressing the limitations outlined, with the aim of achieving more accurate and efficient midcurve generation. Contributions, suggestions, and improvements from the community are highly encouraged and welcomed.

Publications/Talks

Vixra paper MidcurveNN: Encoder-Decoder Neural Network for Computing Midcurve of a Thin Polygon, viXra.org e-Print archive, viXra:1904.0429

ODSC proposal

CAD & Applications 2022 Journal paper 19(6)

Google Developers Dev Library

Google Dev Library | What will you build?

Edit description

devlibrary.withgoogle.com

Medium story Geometry, Graphs and GPT talks about using LLMs (Large Language Models) to see if geometry serialized as line-list can predict the midcurve. An extended abstract on the same topic.

Disclaimer

The author (firstnamelastname at yahoo dot com) gives no guarantee of the results of the program. It is just a fun script. Lot of improvements are still to be made. So, don’t depend on it at all.

Midcurve Artificial Intelligence Generative Ai Tools Open Source Geometry

Edit profile

Written by Yogesh Haribhau Kulkarni (PhD)

1.3K Followers · Editor for Technology Hits

PhD in Geometric Modeling | Google Developer Expert (Machine Learning) | Top Writer 3x (Medium) | More at https://www.linkedin.com/in/yogeshkulkarni/

More from Yogesh Haribhau Kulkarni (PhD) and Technology Hits

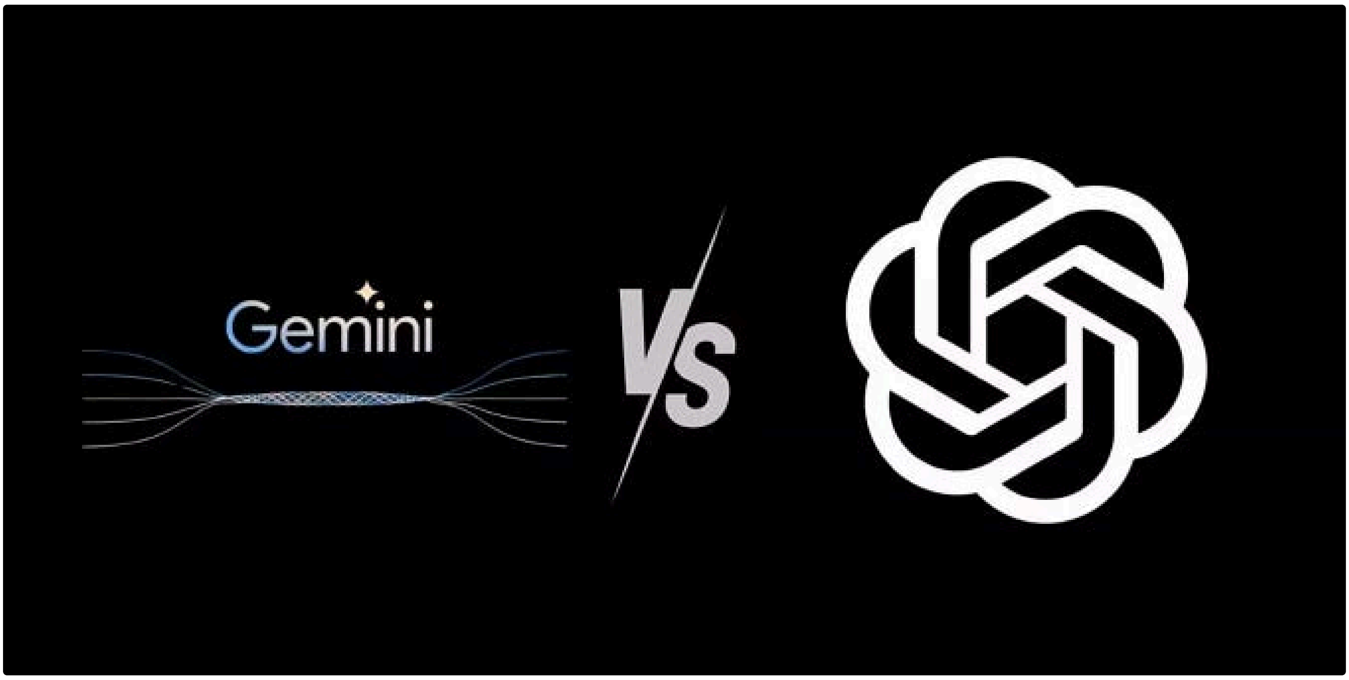
Yogesh Haribhau Kulkarni (PhD) in Technology Hits

Unveiling Manifold learning

What a neural network is really doing?

5 min read · Feb 23, 2024

275 1



Hammad Hassan in Technology Hits

ChatGPT Vs Gemini: Which One I Like As A Writer

As a writer, I know the struggle of digging through mountains of information.

✦ · 4 min read · Feb 8, 2024

1.8K 44

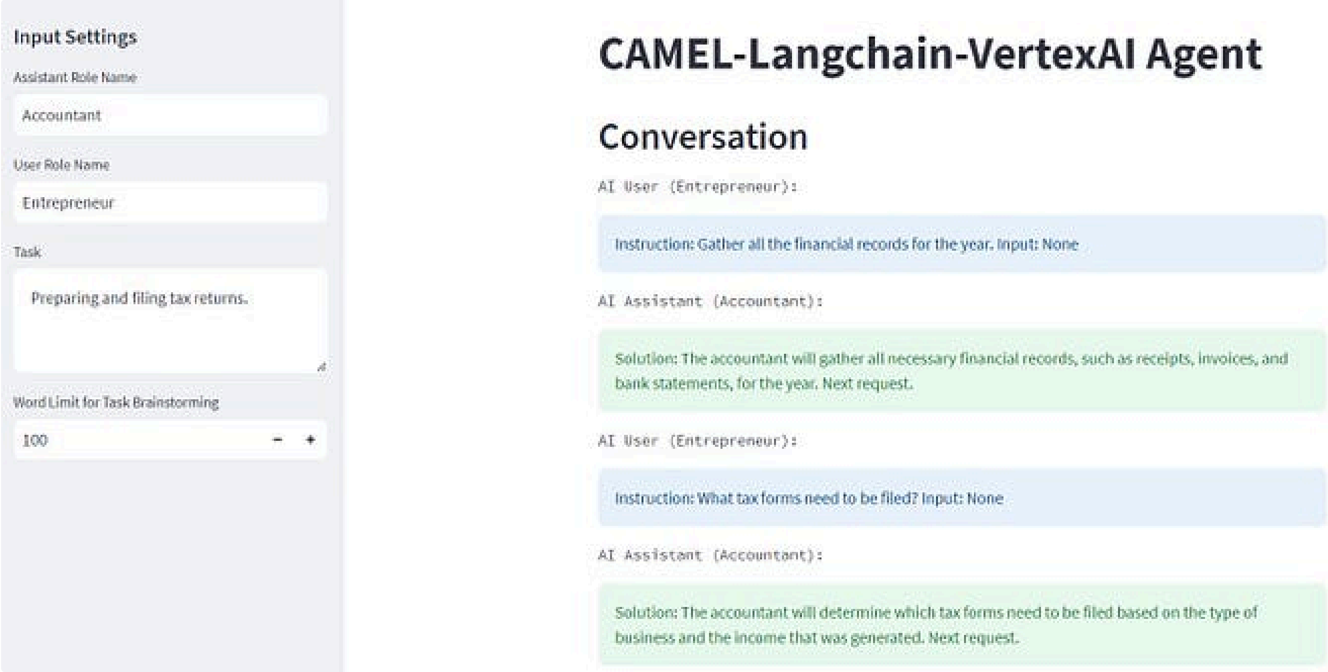
Nuno Campos in Technology Hits

Obsidian Plugins Review — 44

Elevate Your Obsidian Experience: Discover the Best New Plugins for Enhanced Note-Taking and Boost Productivity

✦ · 5 min read · Feb 10, 2024

434 4



Input Settings

Assistant Role Name

Accountant

User Role Name

Entrepreneur

Task

Preparing and filing tax returns.

Word Limit for Task Brainstorming

100

CAMEL-Langchain-VertexAI Agent

Conversation

```
AI User (Entrepreneur):
```

Instruction: Gather all the financial records for the year. Input: None

```
AI Assistant (Accountant):
```

Solution: The accountant will gather all necessary financial records, such as receipts, invoices, and bank statements, for the year. Next request.

```
AI User (Entrepreneur):
```

Instruction: What tax forms need to be filed? Input: None

```
AI Assistant (Accountant):
```

Solution: The accountant will determine which tax forms need to be filed based on the type of business and the income that was generated. Next request.

Yogesh Haribhau Kulkarni (PhD) in Google Developer Experts

A CAMEL ride

A Story of AI Role-Playing using CAMEL, Langchain and VertexAI

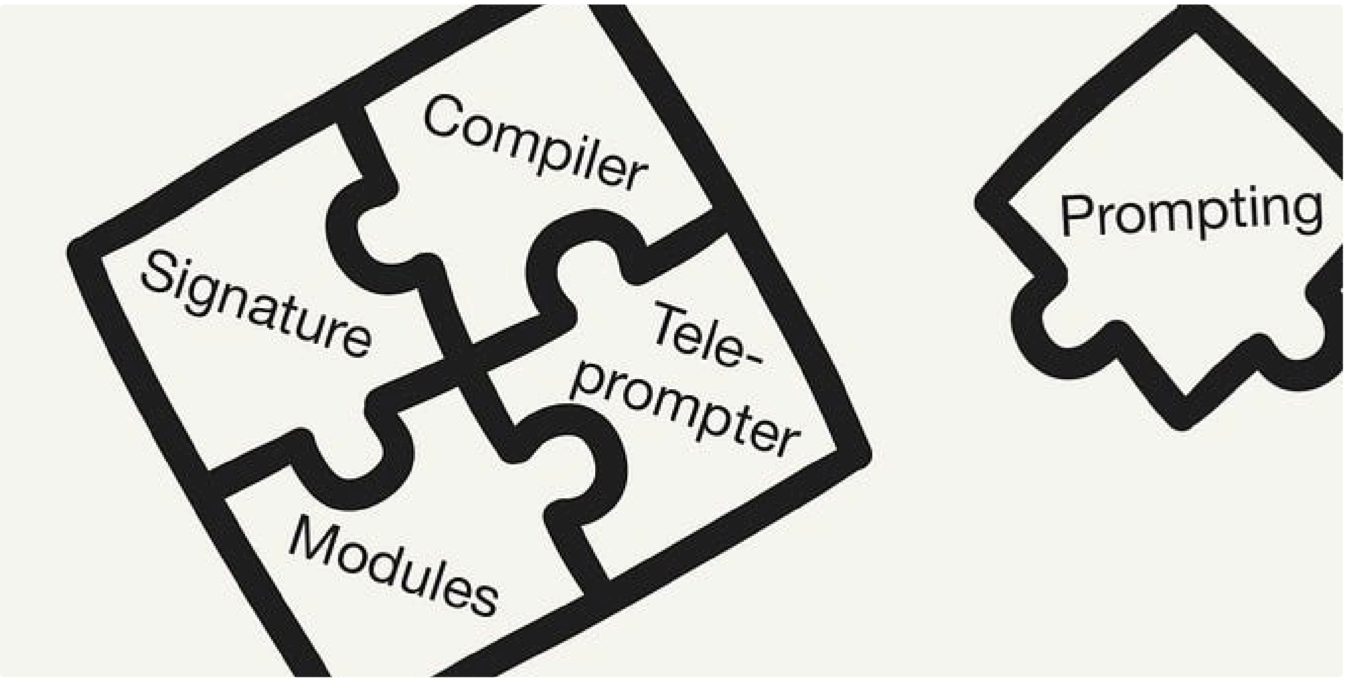
8 min read · Oct 2, 2023

53

See all from Yogesh Haribhau Kulkarni (PhD)

See all from Technology Hits

Recommended from Medium



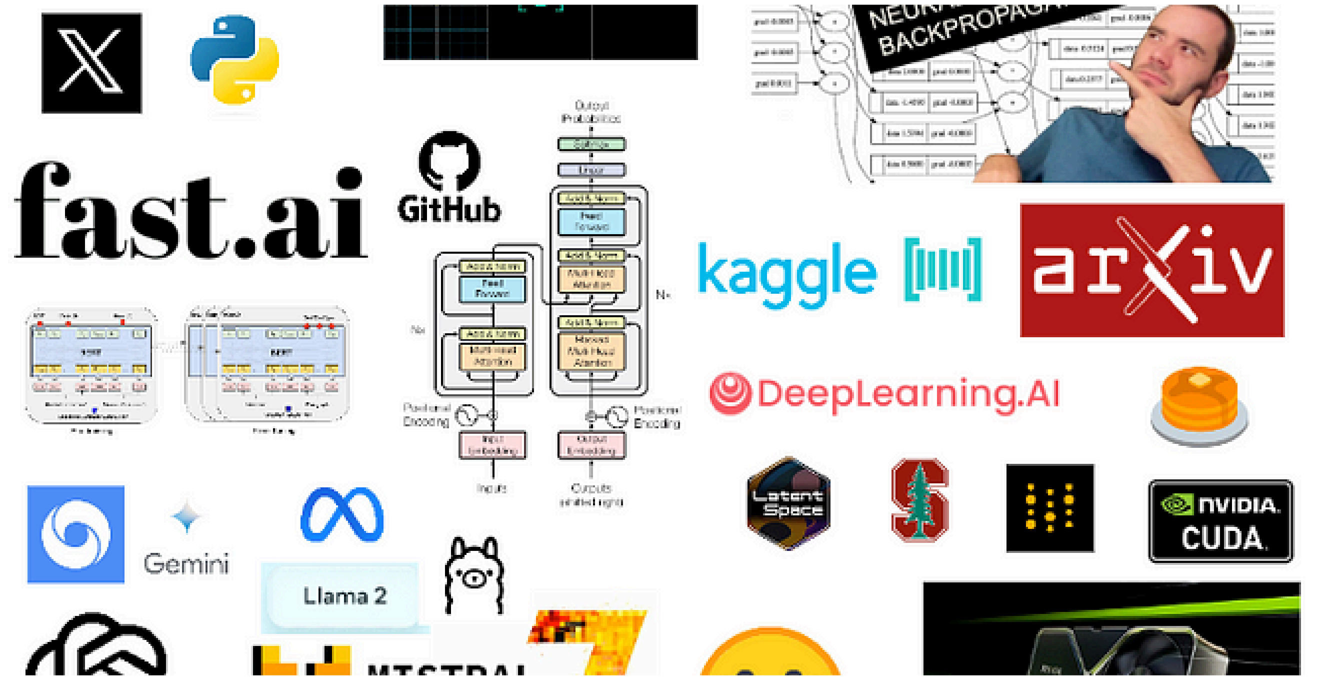
Leonie Monigatti in Towards Data Science

Intro to DSPy: Goodbye Prompting, Hello Programming!

How the DSPy framework solves the fragility problem in LLM-based applications by replacing prompting with programming and compiling

✦ · 13 min read · 6 days ago

1.8K 8



Benedict Neo in bitgrit Data Science Publication

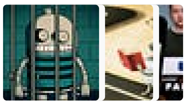
Roadmap to Learn AI in 2024

A free curriculum for hackers and programmers to learn AI

11 min read · Feb 21, 2024

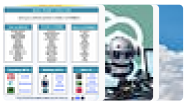
3.9K 45

Lists



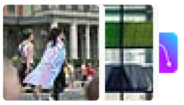
AI Regulation

6 stories · 349 saves



ChatGPT

21 stories · 499 saves



Generative AI Recommended Reading

52 stories · 783 saves

Natural Language Processing

1253 stories · 733 saves

Rajesh K in Towards AI

LLM Quantization Techniques- GPTQ

Recent advances in neural network technology have dramatically increased the scale of the model, resulting in greater sophistication and…

11 min read · Feb 18, 2024

115

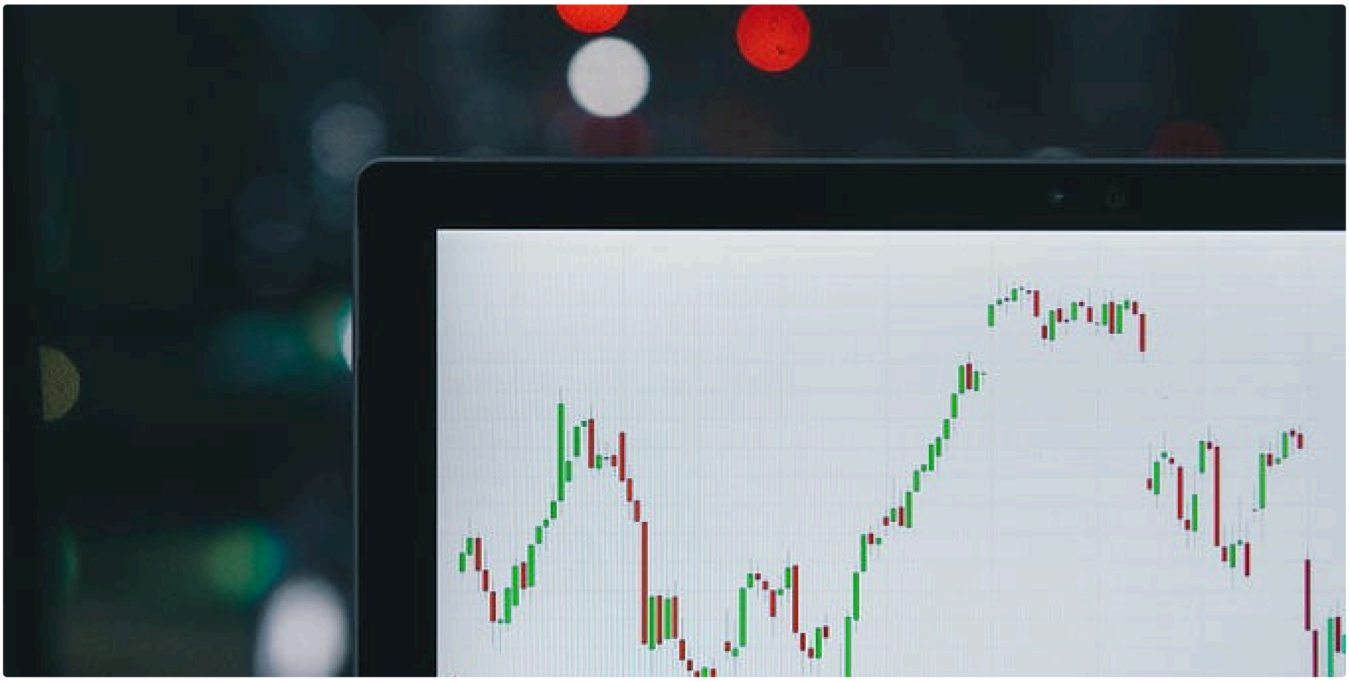
Gene Bernardin

A New Easy Desktop UI for Running Open-Source LLMs: Jan

The newest and easiest way to run open source and free large language models!

12 min read · Jan 22, 2024

335 1



Yanli Liu in Level Up Coding

The Future of Financial AI: Stock Sentiment Analysis with SLIM Models

How to perform complex multi-step analytics using LLMWare on CPU

✦ · 8 min read · 6 days ago

326 1

Andrew Zuo

Gemini 1.5 Pro Is Insane

I recently hooked up Google’s Gemini Pro. And right after I did OpenAI put out this press release:

✦ · 5 min read · Feb 24, 2024

982 16

See more recommendations