Ranks of elliptic curves and deep neural networks

DANGER 3, London Institute for Mathematical Sciences
August 24th, 2023.

Matija Kazalicki[1], Domagoj Vlah[2]

[1]University of Zagreb, Faculty of Science, Department of Mathematics
[2]University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Applied Mathematics

Ranks of elliptic curves

Diophant from Alexandria


DIOPHANTI
ALEXANDRINI
ARITHMETICORVM
LIBRI SEX,
ET DE NVMERIS MVLTANGVLIS
LIBER VNVS.
Nunc primùm Græcè & Latinè editi, atque absolutissimis Commentariis illustrati.
AVCTORE CLAVDIO GASPARE BACHETO
MEZIRIACO SEBVSIANO, V.C.
LVTETIAE PARISIORVM,
Sumptibus SEBASTIANI CRAMOISY, via Iacobæa, sub Ciconiis.
M. DC. XXI.
CVM PRIVILEGIO REGIS.

Figure 1: Edition from 1621.

Diophant from Alexandria


DIOPHANTI
ALEXANDRINI
ARITHMETICORVM
LIBRI SEX,
ET DE NVMERIS MVLTANGVLIS
LIBER VNVS.
Nunc primùm Græcè & Latinè editi, atque absolutissimis
Commentariis illustrati.
AVCTORE CLAVDIO GASPARE BACHETO
MEZIRIACO SEBVSIANO V.C.
LVTETIAE PARISIORVM,
Sumptibus SEBASTIANI CRAMOISY, via
Iacobæa, sub Ciconiis.
M. DC. XXI.
CVM PRIVILEGIO REGIS.

Figure 1: Edition from 1621.

Arithmetic (Book IV, Problem 24): Find (positive and rational) x i y such that

$$y(6 - y) = x^3 - x.$$

Example of elliptic curve

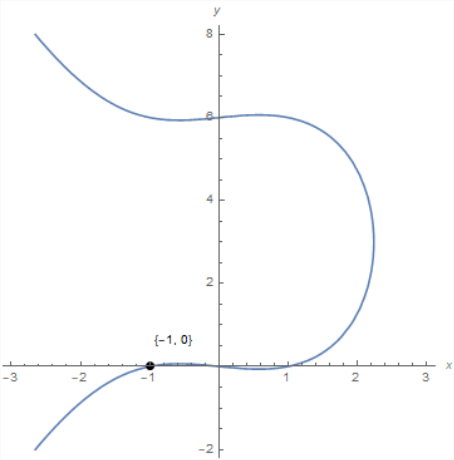
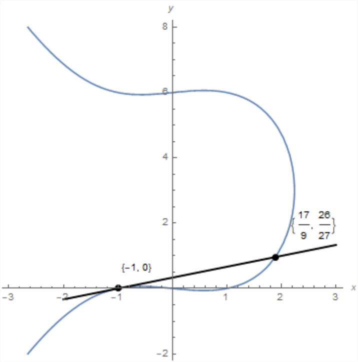


Figure 2: $C : y(6-y) = x^3 - x$

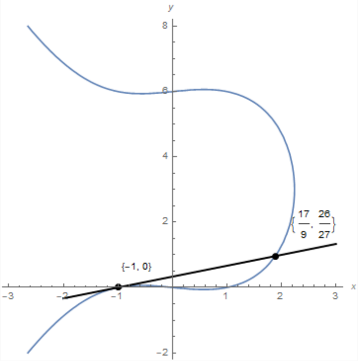
He sketched curve C and noticed a rational point $(-1, 0)$.

A new point from old



Intersection of tangent line through point $(-1, 0)$ and elliptic curve C gives one solution $(17/9, 26/27)$ to the problem!

A new point from old



Intersection of tangent line through point $(-1, 0)$ and elliptic curve C gives one solution $(17/9, 26/27)$ to the problem!

Today we know that with this method (tangent chord process), starting from the points $(-1, 0)$ and $(0, 0)$ (we call them generators), we can obtain all rational points on C.

Rational points on elliptic curve

Definition
For $a_1, a_2, a_3, a_4, a_6 \in \mathbb{Q}$

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

is called **elliptic curve** over $\mathbb{Q}$ (provided that the discriminant $\Delta(a_1, a_2, a_3, a_4, a_6)$ is nonzero).

Rational points on elliptic curve

Definition
For $a_1, a_2, a_3, a_4, a_6 \in \mathbb{Q}$

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

is called **elliptic curve** over $\mathbb{Q}$ (provided that the discriminant $\Delta(a_1, a_2, a_3, a_4, a_6)$ is nonzero).

We denote the set of rational points on E by $E(\mathbb{Q})$. It is an abelian group with respect to the operation previously mentioned (point at infinity is neutral element).

Rational points on elliptic curve

Definition
For $a_1, a_2, a_3, a_4, a_6 \in \mathbb{Q}$

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

is called **elliptic curve** over $\mathbb{Q}$ (provided that the discriminant $\Delta(a_1, a_2, a_3, a_4, a_6)$ is nonzero).

We denote the set of rational points on E by $E(\mathbb{Q})$. It is an abelian group with respect to the operation previously mentioned (point at infinity is neutral element). We denote by N the conductor of elliptic curve.

Rank of elliptic curve

Theorem (Mordell)
$E(\mathbb{Q})$ is a **finitely generated** *abelian group, i.e.*

$$E(\mathbb{Q}) \cong E_{tors}(\mathbb{Q}) \times \mathbb{Z}^r,$$

where $E_{tors}(\mathbb{Q})$ is a finite group of elements of finite order and r is non-negative integer called (algebraic) **rank** *of elliptic curve.*

Rank of elliptic curve

Theorem (Mordell)
$E(\mathbb{Q})$ is a **finitely generated** *abelian group, i.e.*

$$E(\mathbb{Q}) \cong E_{tors}(\mathbb{Q}) \times \mathbb{Z}^r,$$

where $E_{tors}(\mathbb{Q})$ is a finite group of elements of finite order and r is non-negative integer called (algebraic) **rank** *of elliptic curve.*

Example
Rank of $C(\mathbb{Q})$ is two.

Computing rank of elliptic curve

Elliptic curves of high rank

- Rank is a mysterious quantity. Is the rank unbounded?
  Current record is 28 (Elkies).
- Montgomery (1987) - elliptic curves of high rank can speed up elliptic-curve factorization method (ECM)

Rank 28 example

$$y^2 + xy + y = x^3 - x^2 - 20067762415575526585033208209338542750930230312178956502x + 34481611795030556467032985690390720374855944359319180361266008296291939448732243429$$

Rank 28 example

$$y^2 + xy + y = x^3 - x^2 - 20067762415575526585033208209338542750930230312178956502x + 3448161179503055646703298569039072037485594435931918036126600829629193944873224342 9$$

P1 = [-2124150091254381073292137463, 259854492051899599030515511070780628911531]
P2 = [2334509866034701756884754537, 18872004195494469180868316552803627931531]
P3 = [-1671736054062369063879038663, 251709377261144287808506947241319126049131]
P4 = [2139130260139156666492982137, 36639509171439729202421459692941297527531]
P5 = [1534706764467120723885477337, 85429585346017694289021032862781072799531]
P6 = [-2731079487875677033341575063, 262521815484332191641284072623902143387531]
P7 = [2775726266844571649705458537, 12845755474014060248869487699082640369931]
P8 = [1494385729327188957541833817, 88486605527733405986116494514049233411451]
P9 = [1868438228620887358509065257, 59237403214437708712725140393059358589131]
P10 = [2008945108825743774866542537, 47690677880125552882151750781541424711531]
P11 = [2348360540918025169651632937, 17492930006200557857340332476448804363531]
P12 = [-1472084007090481174470008663, 246643450653503714199947441549759798469131]
P13 = [2924128607708061213363288937, 28350264431488878501488356474767375899531]
P14 = [5374993891066061893293934537, 286188908427263386451175031916479893731531]
P15 = [1709690768233354523334008557, 71898834974686089466159700529215980921631]
P16 = [2450954011353593144072595187, 44452281735326343570492625506107147365531]
P17 = [2969254709273559167464674937, 32766893075366270801333682543160469687531]
P18 = [2711914934941692601332882937, 20684366127783816986504139815065906135531]
P19 = [20078586077996854528778328937, 27796085411378066046560517256246240300915311]

⋮

P28 = [2230868289773576023778678737, 28558760030597485663387020600768640028531]

Computing rank is hard

Determining the rank of elliptic curve is computationally expensive task mainly because finding rational points on elliptic curves is a difficult problem

Computing rank is hard

Determining the rank of elliptic curve is computationally expensive task mainly because finding rational points on elliptic curves is a difficult problem

A descent algorithms that are commonly used eventually reduce to a naive point search on some auxiliary curves. Also, there is no algorithm known to correctly compute the rank in all cases.

Computing rank is hard

Determining the rank of elliptic curve is computationally expensive task mainly because finding rational points on elliptic curves is a difficult problem

A descent algorithms that are commonly used eventually reduce to a naive point search on some auxiliary curves. Also, there is no algorithm known to correctly compute the rank in all cases.

Is it possible to determine rank without finding explicit generators?

Birch and Swinnerton-Dyer conjecture

For every prime p, define $a_p = p + 1 - \#E(\mathbb{F}_p)$.

Birch and Swinnerton-Dyer conjecture

For every prime p, define $a_p = p + 1 - \#E(\mathbb{F}_p)$.

For $\Re(s) > 3/2$ we define **Hasse-Weil** L-function by absolutely convergent infinite product

$$L(E, s) = \prod_{p \text{ prime}} \left(1 - a_p p^{-s} + p^{1-2s}\right)^{-1}.$$

By the Modularity theorem $L(E, s)$ extends to an entire function.

Birch and Swinnerton-Dyer conjecture

For every prime p, define $a_p = p + 1 - \#E(\mathbb{F}_p)$.

For $\Re(s) > 3/2$ we define **Hasse-Weil** L-function by absolutely convergent infinite product

$$L(E, s) = \prod_{p \text{ prime}} \left(1 - a_p p^{-s} + p^{1-2s}\right)^{-1}.$$

By the Modularity theorem $L(E, s)$ extends to an entire function.

Conjecture (BSD)
Order of vanishing of $L(E, s)$ at $s = 1$ (the quantity known as analytic rank) is equal to the rank of elliptic curve E.

Heuristics for computing rank

As an alternative approach to descent algorithms, one can use rank heuristics that are inspired by BSD conjecture. These heuristics (we will call them Mestre-Nagao sums) help in identifying probable candidates for elliptic curves of high rank.

Mestre-Nagao sums

For example, one of these sums

$$\tilde{S}_5(B) = \sum_{\substack{p<B,\\ \text{good reduction}}} \log\left(\frac{p+1-a_p}{p}\right)$$

has a property that $\exp(-\tilde{S}_5(B))$ is the partial product of $L_E(s)$

$$\prod_{\substack{p<B,\\ \text{good reduction}}} \left(1 - a_p p^{-s} + p^{1-2s}\right)^{-1}, \qquad (1)$$

evaluated at $s = 1$ (ignoring the primes of bad reduction). One expects that $\tilde{S}_5(B)$ should be large if E has a large rank since then the partial product should rapidly approach zero.

For example, one of these sums

$$\tilde{S}_5(B) = \sum_{\substack{p<B, \\ \text{good reduction}}} \log\left(\frac{p+1-a_p}{p}\right)$$

has a property that $\exp(-\tilde{S}_5(B))$ is the partial product of $L_E(s)$

$$\prod_{\substack{p<B, \\ \text{good reduction}}} \left(1 - a_p p^{-s} + p^{1-2s}\right)^{-1}, \tag{1}$$

evaluated at $s = 1$ (ignoring the primes of bad reduction). One expects that $\tilde{S}_5(B)$ should be large if E has a large rank since then the partial product should rapidly approach zero. This sum was used Elkies and Klagsbrun as a first step in finding rank-record breaking curves with fixed cyclic torsion $\mathbb{Z}/n\mathbb{Z}$ for $n = 2, 3, \ldots 7$.

Approach by convolutional neural networks

In our work, we investigate a deep learning algorithm for rank classification based on convolutional neural networks (CNN).

Approach by convolutional neural networks

In our work, we investigate a deep learning algorithm for rank classification based on convolutional neural networks (CNN).

These networks take as an input the conductor of the elliptic curve together with the sequence of normalized a_p-s (i.e. $a_p/\sqrt{p}$) for p in a fixed range ($p < 10^k$ za $k = 3, 4, 5$) and output the rank of the elliptic curve.

Approach by convolutional neural networks

In our work, we investigate a deep learning algorithm for rank classification based on convolutional neural networks (CNN).

These networks take as an input the conductor of the elliptic curve together with the sequence of normalized a_p-s (i.e. $a_p/\sqrt{p}$) for p in a fixed range ($p < 10^k$ za $k = 3, 4, 5$) and output the rank of the elliptic curve.

This project was inspired by the paper **He, Lee, Oliver: Machine learning invariants of arithmetic curves. J. Symb. Comput. 115, 478–491 (2023)** where the authors, among other things, successfully used logistic regression for classifying elliptic curves of rank zero and one.

Mestre-Nagao sums and a fully connected neural networks

We compared the performance of our CNN algorithm to that of the Mestre-Nagao sums ($S_0, S_1, \cdots S_6$ and Ω).

Mestre-Nagao sums and a fully connected neural networks

We compared the performance of our CNN algorithm to that of the Mestre-Nagao sums ($S_0, S_1, \cdots S_6$ and Ω).

A priori, it is not clear how to decide on the rank of the elliptic curve based on the value of its Mestre-Nagao sum, so we train a simple fully connected neural network to do that task for us. Since the answer critically depends on the conductor of the elliptic curve, these networks, besides the Mestre-Nagao sum, take the conductor of the elliptic curve as an input. Training these networks revealed the optimal cutoff of the specific Mestre-Nagao sum for rank classification.

List of Mestre-Nagao sums

These are some of the Mestre - Nagao sums we considered

$$S_0(B) = \frac{1}{\log B} \sum_{\substack{p<B, \\ \text{good reduction}}} \frac{a_p(E)\log p}{p},$$

$$S_3(B) = \sum_{\substack{p<B, \\ \text{good reduction}}} \frac{-a_p(E)+2}{p+1-a_p(E)} \log p,$$

$$S_4(B) = \frac{1}{B} \sum_{\substack{p<B, \\ \text{good reduction}}} -a_p(E)\log p,$$

List of Mestre-Nagao sums

These are some of the Mestre - Nagao sums we considered

$$S_0(B) = \frac{1}{\log B} \sum_{\substack{p<B, \\ \text{good reduction}}} \frac{a_p(E)\log p}{p},$$

$$S_3(B) = \sum_{\substack{p<B, \\ \text{good reduction}}} \frac{-a_p(E)+2}{p+1-a_p(E)} \log p,$$

$$S_4(B) = \frac{1}{B} \sum_{\substack{p<B, \\ \text{good reduction}}} -a_p(E)\log p,$$

Ω is a neural network that takes as an input conductor of an elliptic curve together with all seven sums sums $(S_0, S_1, \cdots, S_6)$.

Training process

Two datasets

For training, we used two datasets

- LMFDB - contains $3,824,372$ elliptic curves defined over $\mathbb{Q}$, distributed in $2,917,287$ isogeny classes. It contains three different datasets: all curves of conductor less than $500,000$, all curves whose conductor is 7-smooth, and all curves of prime conductor $p \leq 300,000,000$). Curves have rank between 0 and 5

Two datasets

For training, we used two datasets

- LMFDB - contains $3,824,372$ elliptic curves defined over $\mathbb{Q}$, distributed in $2,917,287$ isogeny classes. It contains three different datasets: all curves of conductor less than $500,000$, all curves whose conductor is 7-smooth, and all curves of prime conductor $p \leq 300,000,000$). Curves have rank between 0 and 5
- Custom made dataset - contains $2,074,863$ elliptic curves defined over $\mathbb{Q}$ with trivial torsion and conductor less than 10^{30}. These curves have rank between 0 and 10.

Construction of custom dataset

- curves with random Weierstrass coefficients for rank 0 and 1

Construction of custom dataset

- curves with random Weierstrass coefficients for rank 0 and 1
- curves of higher rank were obtained as random specializations of pencils of cubics through randomly selected k rational points in the plane, for $k = 2, 3, \ldots, 8$

Construction of custom dataset

- curves with random Weierstrass coefficients for rank 0 and 1
- curves of higher rank were obtained as random specializations of pencils of cubics through randomly selected k rational points in the plane, for $k = 2, 3, \ldots, 8$
- using PARI/GP ellrank we tried to compute ranks of all previously generated curves (assuming the Parity conjecture) and discarded curves for which PARI/GP couldn't find the rank (e.g. all curves for which $Sha(E)[4]$ is nontrivial)

Bias of custom dataset

Dataset suffers from bias as it is constructed by sampling the rational points of small height in the pencil of cubics. Consequently, it contains many elliptic curves with small canonical height generators and hence small regulator.

Bias of custom dataset

Dataset suffers from bias as it is constructed by sampling the rational points of small height in the pencil of cubics. Consequently, it contains many elliptic curves with small canonical height generators and hence small regulator.

This presents a problem, particularly for curves of small rank and large conductor, since the regulator is typically expected to be large under standard conjectures.

$\tilde{S}_5$ and the original BSD

The size of the regulator is significant since the Mestre-Nagao sums ultimately approximate a term whose size depends on both the rank and the regulator.

$\tilde{S}_5$ and the original BSD

The size of the regulator is significant since the Mestre-Nagao sums ultimately approximate a term whose size depends on both the rank and the regulator.

For instance, from the original Birch and Swinnerton-Dyer conjecture,

$$\prod_{\substack{p<B, \\ \text{good reduction}}} \frac{p+1-a_p}{p} \approx A \log(B)^r,$$

it follows (by taking logarithms) that $\tilde{S}_5(B) = \log A + r \log\log B + o(1)$, where A is a constant that conjecturally depends on the regulator of E (Goldfeld).

Experiments

For each neural network (the CNN or one of the Mestre-Nagao sums, in total 9) we have performed 24 tests by varying

a) dataset - LMFDB or custom,

Experiments

For each neural network (the CNN or one of the Mestre-Nagao sums, in total 9) we have performed 24 tests by varying

a) dataset - LMFDB or custom,
b) range of a_p-s - we considered $(a_p)_{p \leq B}$ for $B = 10^3, 10^4, 10^5$,

Experiments

For each neural network (the CNN or one of the Mestre-Nagao sums, in total 9) we have performed 24 tests by varying

a) dataset - LMFDB or custom,
b) range of a_p-s - we considered $(a_p)_{p \leq B}$ for $B = 10^3, 10^4, 10^5$,
c) test curves - uniformly selected (20% from the dataset) or all curves in the top conductor range (which is $[10^8, 10^9]$ for the LMFDB and $[10^{29}, 10^{30}]$ for the custom dataset),

Experiments

For each neural network (the CNN or one of the Mestre-Nagao sums, in total 9) we have performed 24 tests by varying

a) dataset - LMFDB or custom,
b) range of a_p-s - we considered $(a_p)_{p \leq B}$ for $B = 10^3, 10^4, 10^5$,
c) test curves - uniformly selected (20% from the dataset) or all curves in the top conductor range (which is $[10^8, 10^9]$ for the LMFDB and $[10^{29}, 10^{30}]$ for the custom dataset),
d) type of classification - binary or all ranks (for the LMFDB the rank range is from 0 to 5, and for the custom dataset from 0 to 10).

Experiments

For each neural network (the CNN or one of the Mestre-Nagao sums, in total 9) we have performed 24 tests by varying

a) dataset - LMFDB or custom,
b) range of a_p-s - we considered $(a_p)_{p \leq B}$ for $B = 10^3, 10^4, 10^5$,
c) test curves - uniformly selected (20% from the dataset) or all curves in the top conductor range (which is $[10^8, 10^9]$ for the LMFDB and $[10^{29}, 10^{30}]$ for the custom dataset),
d) type of classification - binary or all ranks (for the LMFDB the rank range is from 0 to 5, and for the custom dataset from 0 to 10).

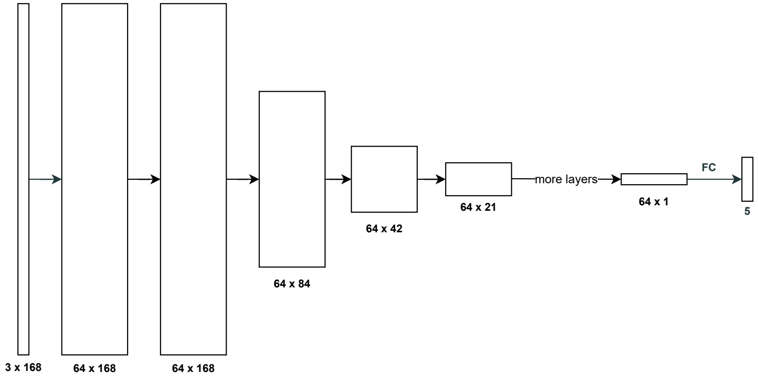
In binary classification, curves are labeled as either of low or high rank. For the LMFDB high rank means 4 (we did not consider 19 rank 5 curves), while for custom dataset high rank is $8, 9$ or 10.

Modeling rank classifier using deep neural network

- architecture: a sequence of convolutional neural network layers + fully connected classification layer in the end
- activation function: ReLU - pointwise $f(x) := \max(x, 0)$
- loss function:
  - weighted cross entropy loss
  - weights reflect relative size of classes - different ranks
- gradient descent optimizer: Adam (a variant of SGD)
- autograd using Pytorch library
- input normalization
- train / validation / test set split
- quality of classification was decided using Matthews correlation coefficient (or phi coefficient)

Architecture of CNN's

- Case $B = 1000$ - first 168 primes
- kernel size 33, ReLU activation function
- cca. 773,000 parameters



Matthews correlation coefficient

- balanced measure of classification quality
  - even for classes of very different sizes
- e.g. for binary classification, MCC is computed using:

$$\mathrm{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + EP) \cdot (TN + FN)}}$$

  - TP - number of true positives
  - FN - number of false negatives
  - TN - number of true negatives
  - FP - number of false positives
- MCC lies in the segment $[-1, 1]$
- $\mathrm{MCC} = 1$ only in the case of perfect classification

Results

Cutoffs for S_5

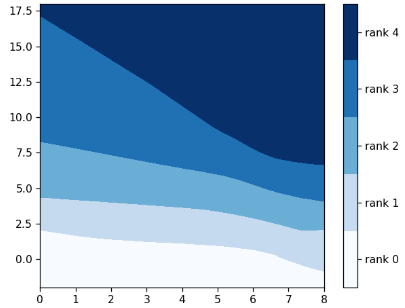
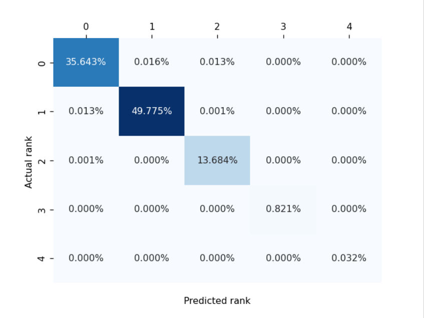


Figure 3: Rank cutoffs of the classifier S_5 as a function of a conductor, trained on the LMFDB with the uniform test set and $p < 10^4$. On the x-axis are $\log_{10}$ values of conductors and on the y-axis are values of the sum S_5. The unexpected shape of the cutoff between ranks 3 and 4 is the consequence of a small number of rank 4 curves with a small conductor, which are present in the dataset.

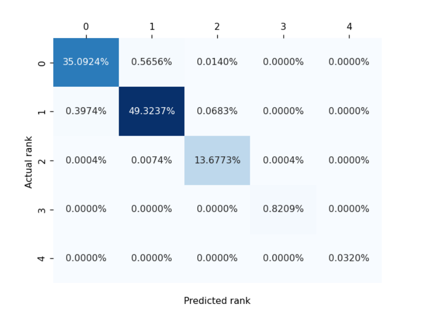
Comparison of different classifiers for LMFDB dataset with uniform test set

Type of classifier	Number of a_p-s used		
	$p < 10^3$	$p < 10^4$	$p < 10^5$
CNN	**0.9507**	**0.9958**	**0.9992**
S_0	0.6823	0.8435	0.9068
S_1	0.6848	0.8507	0.9301
S_2	**0.7277**	**0.8697**	0.9359
S_3	0.6933	0.8499	0.9142
S_4	0.2678	0.3015	0.1525
S_5	0.6132	0.7774	0.8463
S_6	0.6969	0.8647	**0.9381**
Ω	**0.8685**	**0.9602**	**0.9826**

Confusion matrices of CNN and Ω for $p < 10^5$ and uniform test dataset



CNN MCC $= 0.9992$

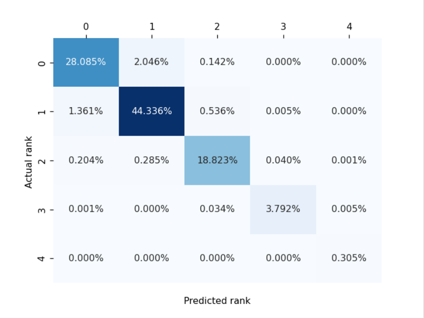


Ω MCC $= 0.9826$

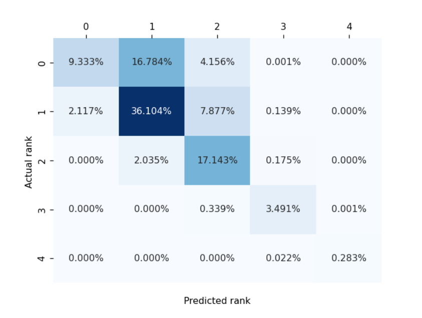
Comparison of different classifiers for LMFDB dataset with top conductor range test set

Type of classifier	Number of a_p-s used		
	$p < 10^3$	$p < 10^4$	$p < 10^5$
CNN	**0.5669**	**0.9289**	**0.9846**
S_0	0.2880	**0.5057**	0.6545
S_1	0.2791	0.4883	0.6658
S_2	0.2790	0.4968	**0.6730**
S_3	0.2897	0.5030	0.6574
S_4	0.1352	0.1424	0.1850
S_5	**0.2960**	0.3913	0.5261
S_6	0.2632	0.4542	0.6416
Ω	**0.4433**	**0.7013**	**0.8530**

Confusion matrices of CNN and S_0 for $p < 10^4$ and top conductor range test dataset



CNN MCC $= 0.9289$



S_0 MCC $= 0.5057$

LMFDB dataset highlights

- for all rank classification in uniform test set and $p < 10000$, CNN (MCC=0.9958) misclassfied only 0.25% of the curves
- the best Mestre-Nagao sum in the same mode, S_2 (MCC=0.8697), missclassified 8.1% of curves
- for all rank classification and top conductor rank MCC of CNN is 0.9289 while MCC of the best Mestre-Nagao sum S_6 is 0.5057!

Comparison of different classifiers for custom dataset with uniform test set

Type of classifier	Number of a_p-s used		
	$p < 10^3$	$p < 10^4$	$p < 10^5$
CNN	**0.6129**	**0.7218**	**0.7958**
S_0	0.5738	0.6782	0.7462
S_1	**0.5780**	**0.6890**	**0.7592**
S_2	0.5649	0.6761	0.7521
S_3	0.5551	0.6616	0.7361
S_4	0.2893	0.2472	0.2251
S_5	0.4987	0.5990	0.6696
S_6	0.5230	0.6509	0.7361
Ω	**0.5999**	**0.7069**	**0.7807**

Comparison of different classifiers for custom dataset with top conductor range test set

Type of classifier	Number of a_p-s used		
	$p < 10^3$	$p < 10^4$	$p < 10^5$
CNN	0.2147	0.3019	0.3655
S_0	0.2533	**0.3233**	**0.3719**
S_1	**0.2573**	**0.3291**	**0.3834**
S_2	0.2340	0.3118	0.3688
S_3	**0.2556**	0.3189	0.3645
S_4	0.1234	0.1228	0.1024
S_5	0.2081	0.2858	0.3380
S_6	0.1803	0.2757	0.3527
Ω	**0.2622**	**0.3246**	**0.3905**

the custom dataset highlights

- a classification is much more challenging
- in all ranks classification with $p < 10000$ and uniform range, MCC of CNN is 0.7218 and it misclassified 23% of curves, while for 3% of the curves prediction missed true rank for more than 1
- in the same mode, the best Mestre-Nagao sum S_1 with $MCC = 0.6890$ misclassified 26% of the curves
- in the top conductor range with $p < 10000$, MCC of CNN is 0.3019 and it misclassified 61% of the curves while for 12% of the curves prediction missed true rank more than 1
- the best Mestre-Nagao sum in this mode, S_2, has $MCC = 0.3291$

How does N and B influence the quality of classification?

How does N and B influence the quality of classification?

As a toy model for detecting rank-0 curves, we can numerically evaluate $L(E,1)$ using the approximate functional equation given by

$$L(E,1) = 2\sum_{n=1}^{\infty} \frac{a_n}{n} e^{-2\pi n/\sqrt{N}}.$$

How does N and B influence the quality of classification?

As a toy model for detecting rank-0 curves, we can numerically evaluate $L(E, 1)$ using the approximate functional equation given by

$$L(E, 1) = 2 \sum_{n=1}^{\infty} \frac{a_n}{n} e^{-2\pi n/\sqrt{N}}.$$

To ensure a small error of the approximation, we require $B = \sqrt{N}$ at a minimum.

How does N and B influence the quality of classification?

As a toy model for detecting rank-0 curves, we can numerically evaluate $L(E,1)$ using the approximate functional equation given by

$$L(E,1) = 2\sum_{n=1}^{\infty} \frac{a_n}{n} e^{-2\pi n/\sqrt{N}}.$$

To ensure a small error of the approximation, we require $B = \sqrt{N}$ at a minimum.

Therefore, it make sense to investigate the dependence of the quality of classification of various models, as measured by the MCC, on the quantity $B/\sqrt{N}$.

The quality of CNN for different B's

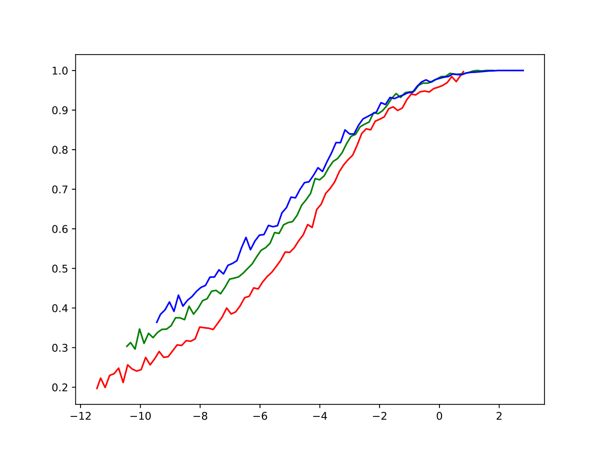


Figure 4: MCC of the CNN as a function of $\log_{10}(B/\sqrt{N})$, for $B = 10^3, 10^4, 10^5$ in red, green, and blue, respectively.

Comparison of different models

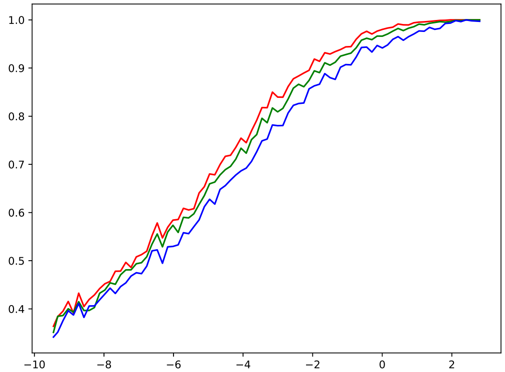


Figure 5: MCC as a function of $\log_{10}(B/\sqrt{N})$, for $B = 10^5$ and three different models: the CNN, Ω, and S_1 in red, green, and blue, respectively.

Classification in families of elliptic curves

Consider the K3 elliptic surface with discriminant -163

$$y^2 = x^3 + (65536t^4 - 17472t^3 - 10176t^2 + 18672t - 3535)x^2 + 1024(t+1)^2(15t-8)^2(31t-7)^2 x,$$

of Mordell-Weil rank 4 (over $\mathbb{Q}(t)$) with $\mathbb{Z}/4\mathbb{Z}$ torsion subgroup.

Classification in families of elliptic curves

Consider the K3 elliptic surface with discriminant -163

$$y^2 = x^3 + (65536t^4 - 17472t^3 - 10176t^2 + 18672t - 3535)x^2 \\ + 1024(t+1)^2(15t-8)^2(31t-7)^2 x,$$

of Mordell-Weil rank 4 (over $\mathbb{Q}(t)$) with $\mathbb{Z}/4\mathbb{Z}$ torsion subgroup.

- sampled 808 curves with $N < 10^{29}$ by substituting $t \in \{-2000, \ldots, 2000\}$
- discarded 37 curves because we were unable to compute their rank.

Confusion matrix of the CNN

The MCC of the CNN is equal to 0.2492.

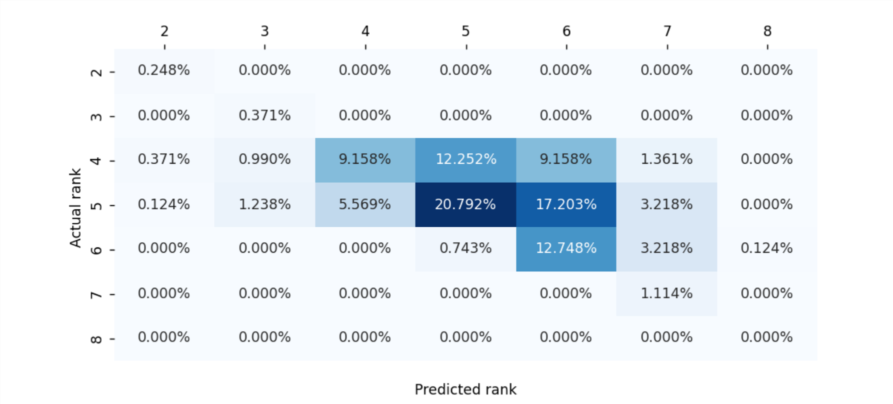


Figure 6: Confusion matrix of the CNN for the K3 elliptic family and $p < 10^5$.

Concluding remarks and the future work

- It is unclear why the CNN works much better than all of the other Mestre-Nagao sum-based models on the curves from the LMFDB. Did the CNN discover some new mathematics?

Concluding remarks and the future work

- It is unclear why the CNN works much better than all of the other Mestre-Nagao sum-based models on the curves from the LMFDB. Did the CNN discover some new mathematics?
- How would the other invariants of elliptic curves (such as root number or the size of Tate-Shafarevich group), if provided as an input to the neural network, effect the quality of classification?

Concluding remarks and the future work

- It is unclear why the CNN works much better than all of the other Mestre-Nagao sum-based models on the curves from the LMFDB. Did the CNN discover some new mathematics?
- How would the other invariants of elliptic curves (such as root number or the size of Tate-Shafarevich group), if provided as an input to the neural network, effect the quality of classification?
- What if we try to classify rank using the values of the Mestre-Nagao sums for two different B's?

Concluding remarks and the future work

- It is unclear why the CNN works much better than all of the other Mestre-Nagao sum-based models on the curves from the LMFDB. Did the CNN discover some new mathematics?
- How would the other invariants of elliptic curves (such as root number or the size of Tate-Shafarevich group), if provided as an input to the neural network, effect the quality of classification?
- What if we try to classify rank using the values of the Mestre-Nagao sums for two different B's?
- Can we construct new improved Mestre-Nagao sums?

Thank you for your attention!

Thank you for your attention!

More details:

Kazalicki, Vlah: Ranks of elliptic curves and deep neural networks, Res. number theory 9, 53 (2023)

https://doi.org/10.1007/s40993-023-00462-w

https://github.com/domagojvlah/deepellrank