Resource-efficient Machine Learning

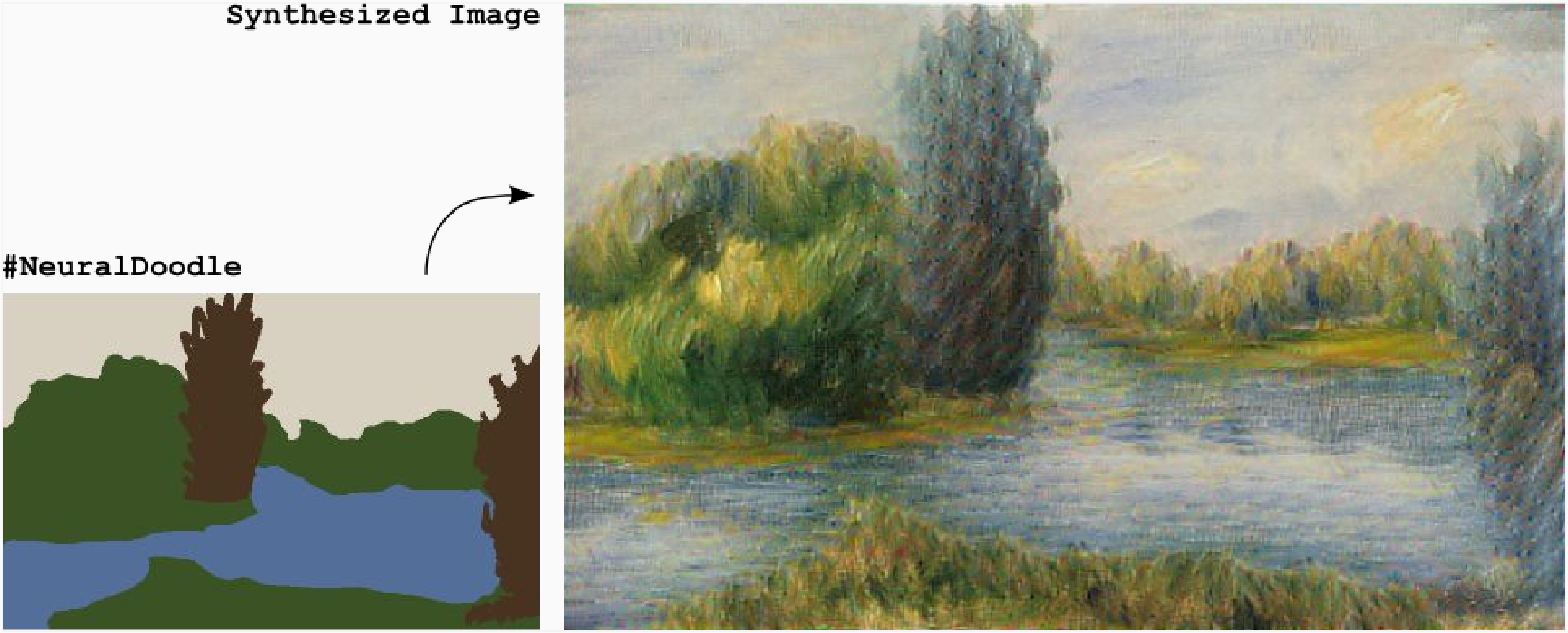
March 16th, 2016

Theodore Vasiloudis, SICS/KTH

What can we do with machine learning these days?

What can we do with machine learning these days?

- We can paint pictures!



Source: Champandard (2016)

NeuralDoodle: Turning Two-Bit Doodles into Fine Artwork

What can we do with machine learning these days?

- We can paint pictures!
- We can beat top-ranked players at Go!

sam byford
@345triangle

Follow

lee se-dol leaves the match room bathed in camera flashes after historic defeat to deepmind theverge.com/2016/3/9/11184 …

AlphaGo beats Lee Se-dol in first of five matches

What can we do with machine learning these days?

- We can paint pictures!
- We can beat professionals at Go!

What can we do with machine learning these days?

- We can paint pictures!
  - Optimization problems
  - i.e. we can approximate unknown functions

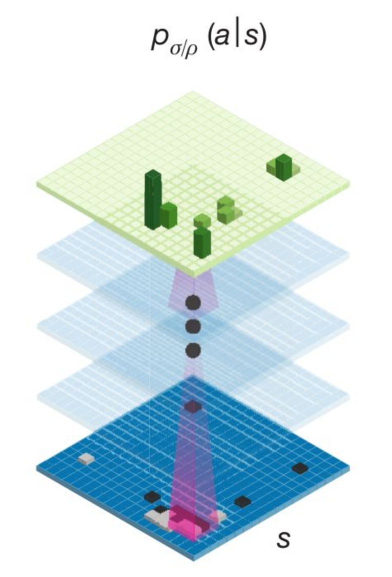
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

SWEDISH ICT SICS

What can we do with machine learning these days?

- We can paint pictures!
  - Optimization problems
  - i.e. we can approximate unknown functions
- We can beat professionals at Go!
  - Probabilistic problems
  - i.e. we can also approximate unknown distributions*

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$



*definition abuse warning, see Silver et al. (2016) for details

SWEDISH ICT SICS

How can we do “resource-efficient” Machine Learning?

How can we do “resource-efficient” Machine Learning?

- Two ways to look at the problem:

How can we do “resource-efficient” Machine Learning?

- Two ways to look at the problem:
    - Algorithms

How can we do “resource-efficient” Machine Learning?

- Two ways to look at the problem:
  - Algorithms
    - Create smarter algorithms and use math tricks to reduce computations

How can we do “resource-efficient” Machine Learning?

- Two ways to look at the problem:
  - Algorithms
    - Create smarter algorithms and use math tricks to reduce computations
  - Systems

How can we do “resource-efficient” Machine Learning?

- Two ways to look at the problem:
  - Algorithms
    - Create smarter algorithms and use math tricks to reduce computations
  - Systems
    - Ensure that computations are efficient and minimize communication

Resource efficient algorithms

Resource-efficient algorithms

- “Sum all numbers from 1 to 100”
  - 1 + 2 + 3 + … = ?

Resource-efficient algorithms

- “Sum all numbers from 1 to 100”
  - 1 + 2 + 3 + … = ?
  - 1+100=101, 2+99=101, 3+98=101, ..., 50+51=101.
    - 50 × 101 = 5050

Resource-efficient algorithms

- "Sum all numbers from 1 to 100"
  - 1 + 2 + 3 + … = ?
  - 1+100=101, 2+99=101, 3+98=101, ..., 50+51=101.
    - 50 × 101 = 5050
  - sum(1...n) = n(n+1)/2

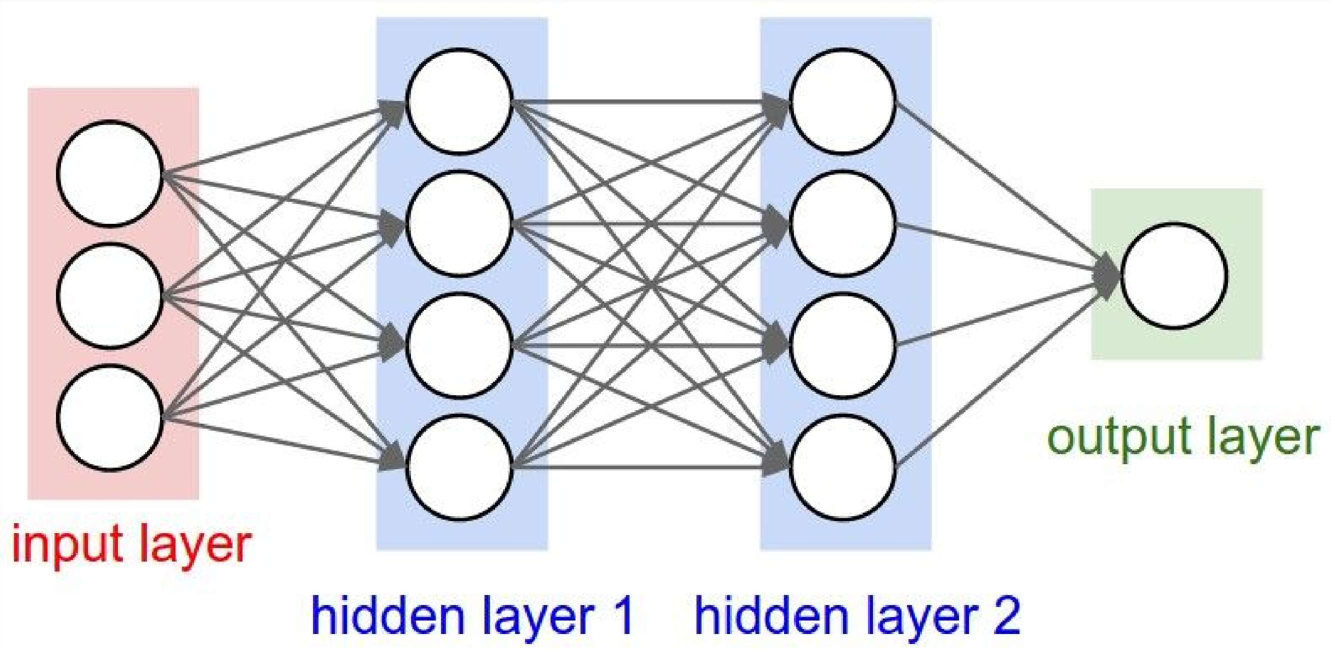
Resource-efficient deep learning

- *“Structured Transforms for Small-footprint Deep Learning”*, Sindhwani et al., NIPS 2015

Resource-efficient deep learning

Resource-efficient deep learning

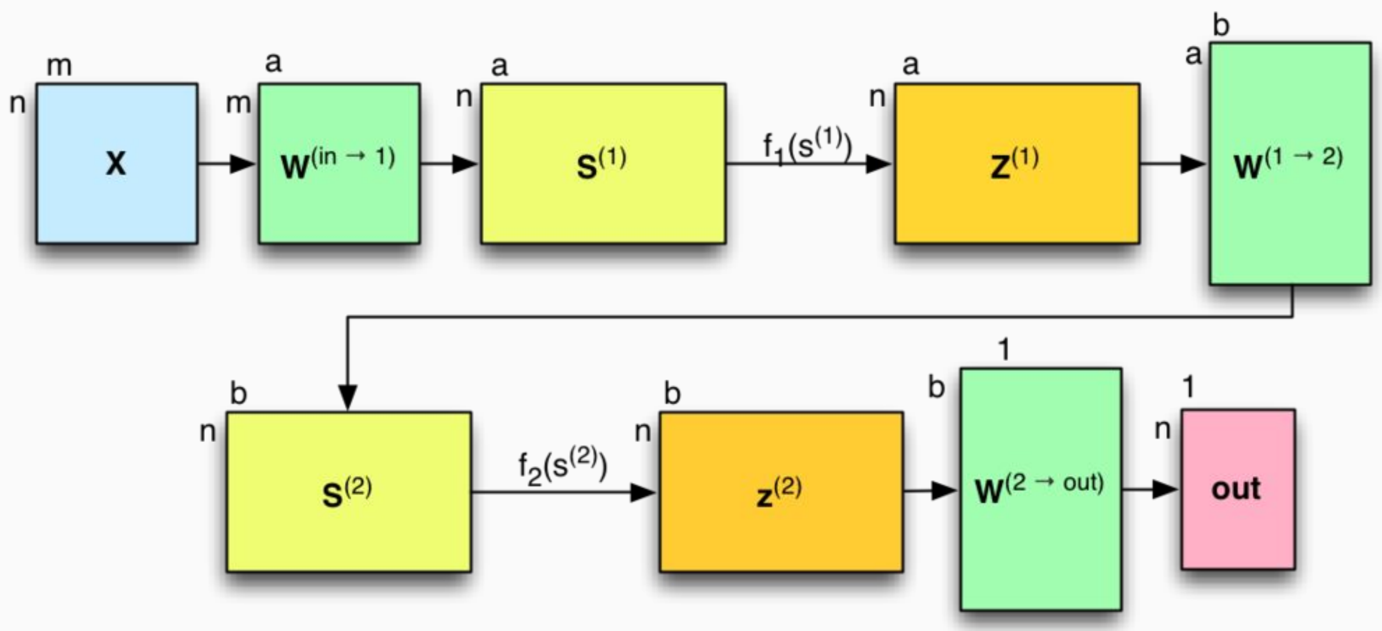
- Deep learning = Neural Networks (NN) = Matrix math (Linear algebra)



Source: Karpathy

Resource-efficient deep learning

- Deep learning = Neural Networks (NN) = Matrix math (Linear algebra)



Source: Dolhansky

Resource-efficient deep learning

- *Structured matrices*: Matrices whose elements exhibit a common structure, e.g in a Toeplitz matrix each diagonal is constant:

$$\begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}$$

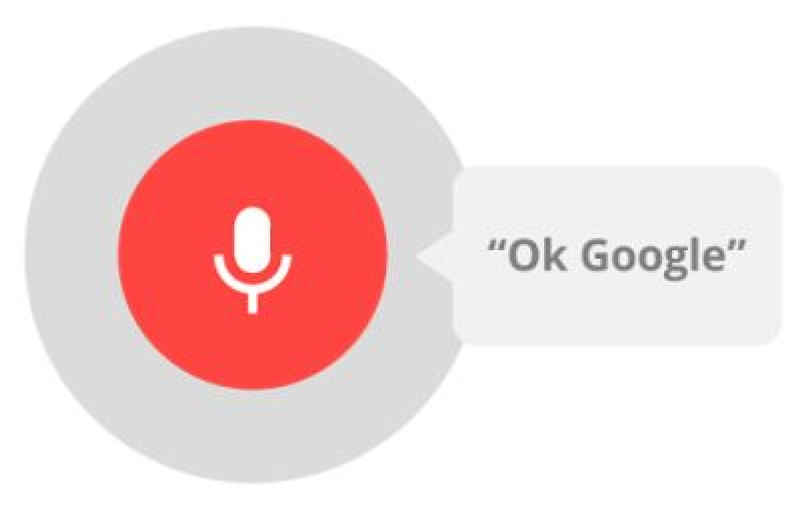
Resource-efficient deep learning

- Idea: Represent NN matrices as combinations of Toeplitz matrices, allowing us to do “superfast” linear algebra

$$\begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-(n-1)} \\ t_1 & t_0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}$$

Resource-efficient deep learning

- Results: Networks 80 times smaller than original, with ~99.8% of the performance.



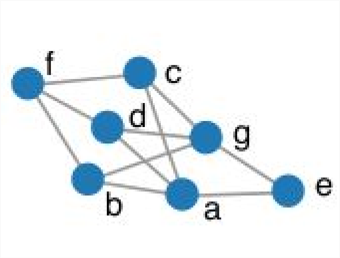
Resource-efficient similarity calculation

Resource-efficient similarity calculation

- Similarity between objects
  - Websites for search
  - Users for recommendations
  - Proteins for disease study

Resource-efficient similarity calculation

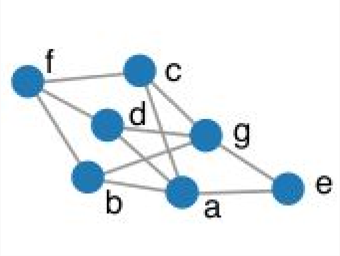
- Similarity between objects
  - Websites for search
  - Users for recommendations
  - Proteins for disease study
- Generality: Model object and relations in a graph



Görnerup (2015)

Resource-efficient similarity calculation

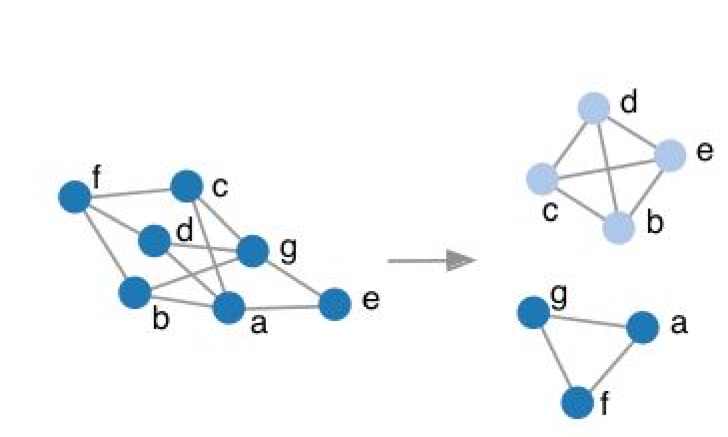
- Similarity between objects
  - Websites for search
  - Users for recommendations
  - Proteins for disease study
- Generality: Model object and relations in a graph
- Problems
  - Too many nodes and connections!
  - Current approaches don't scale!



Görnerup (2015)

Resource-efficient systems

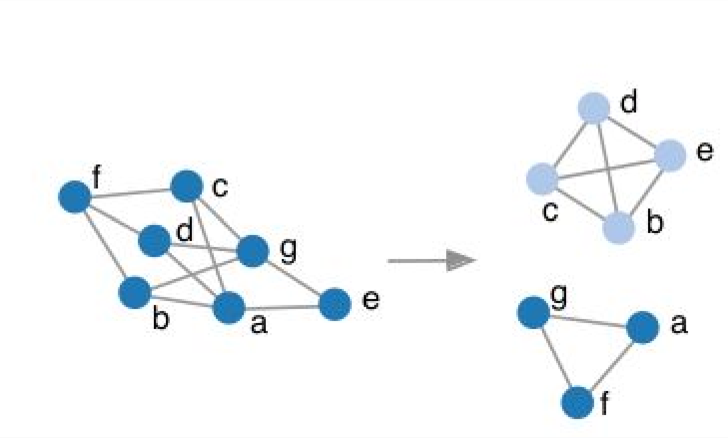
- Idea: Don’t calculate the similarity between Taylor Swift and Beethoven!



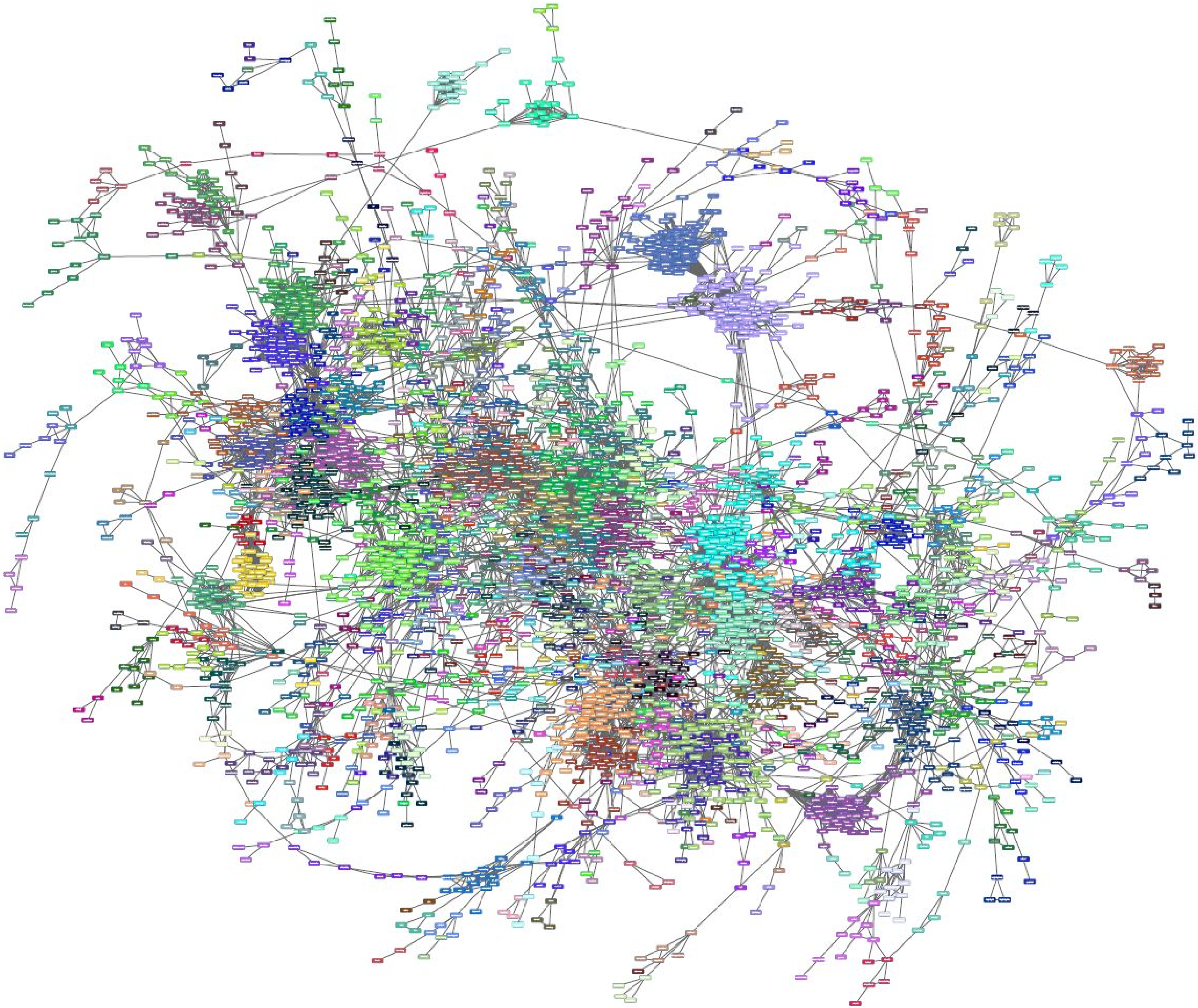
Görnerup (2015)

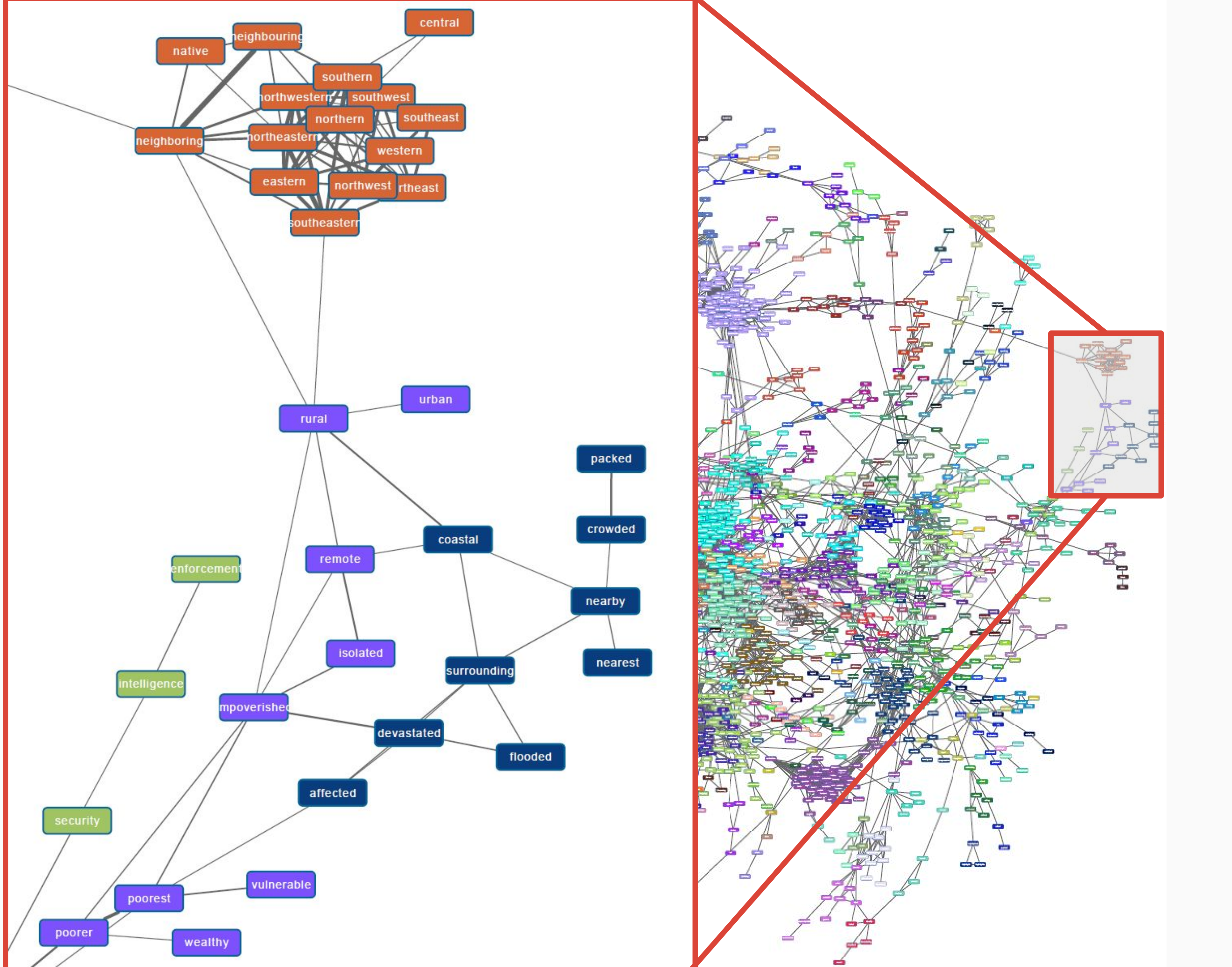
Resource-efficient systems

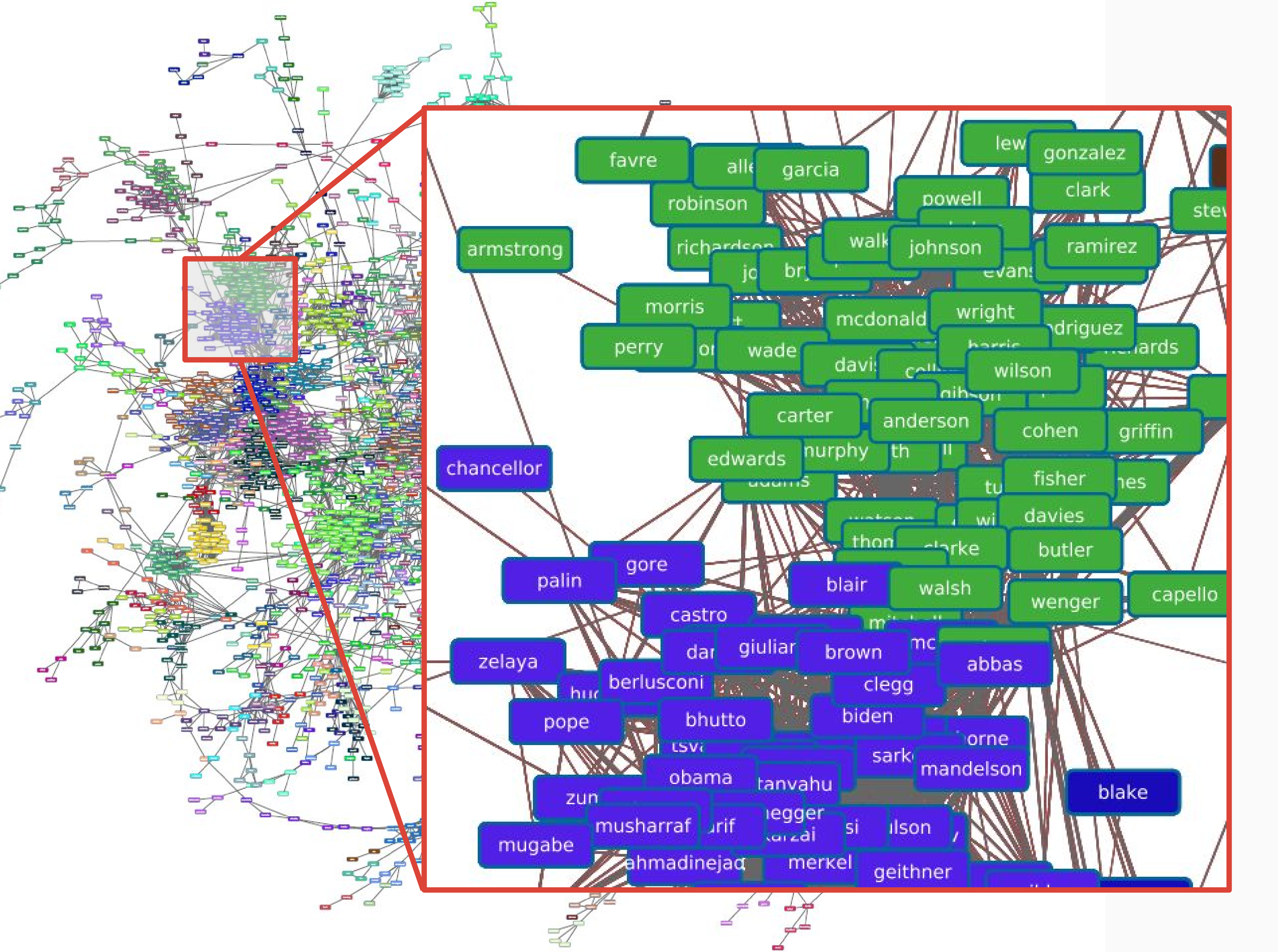
- Idea: Don't calculate the similarity between Taylor Swift and Beethoven!
- Allows us to tackle orders of magnitude larger graphs!



Görnerup (2015)




central
neighbouring
native
southern
northwestern
southwest
northern
southeast
neighboring
northeastern
western
eastern
northwest
rtheast
southeastern
urban
rural
packed
crowded
coastal
remote
enforcement
nearby
isolated
nearest
intelligence
surrounding
mpoverished
devastated
flooded
affected
security
vulnerable
poorest
poorer
wealthy

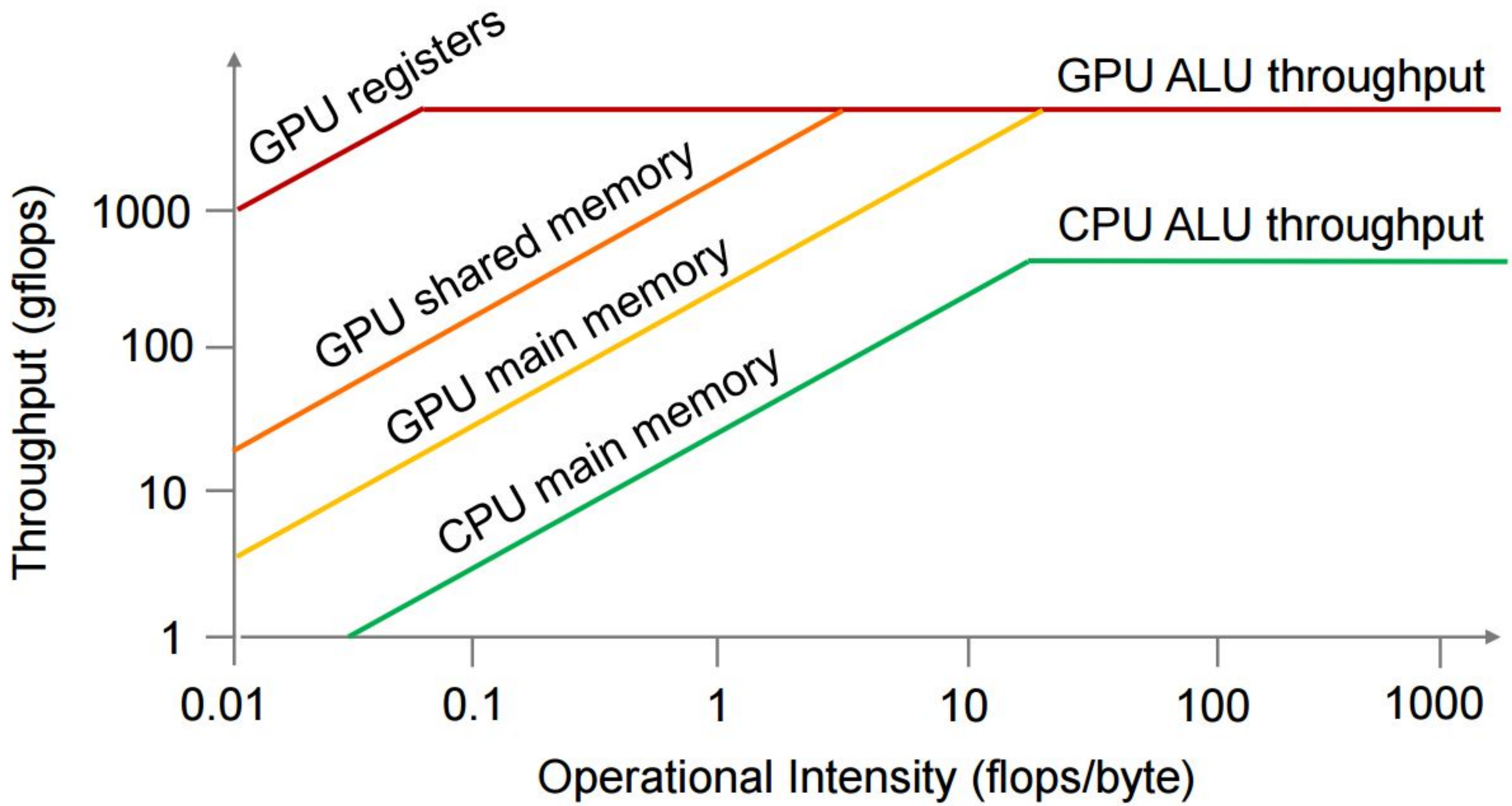

favre
garcia
gonzalez
robinson
powell
clark
armstrong
johnson
ramirez
morris
mcdonald
wright
perry
wade
wilson
carter
anderson
cohen
griffin
chancellor
edwards
fisher
davies
butler
gore
palin
blair
walsh
wenger
capello
castro
zelaya
brown
abbas
berlusconi
clegg
pope
bhutto
biden
mandelson
obama
blake
musharraf
mugabe
merkel
geithner

Resource efficient systems

BID Data Toolkit

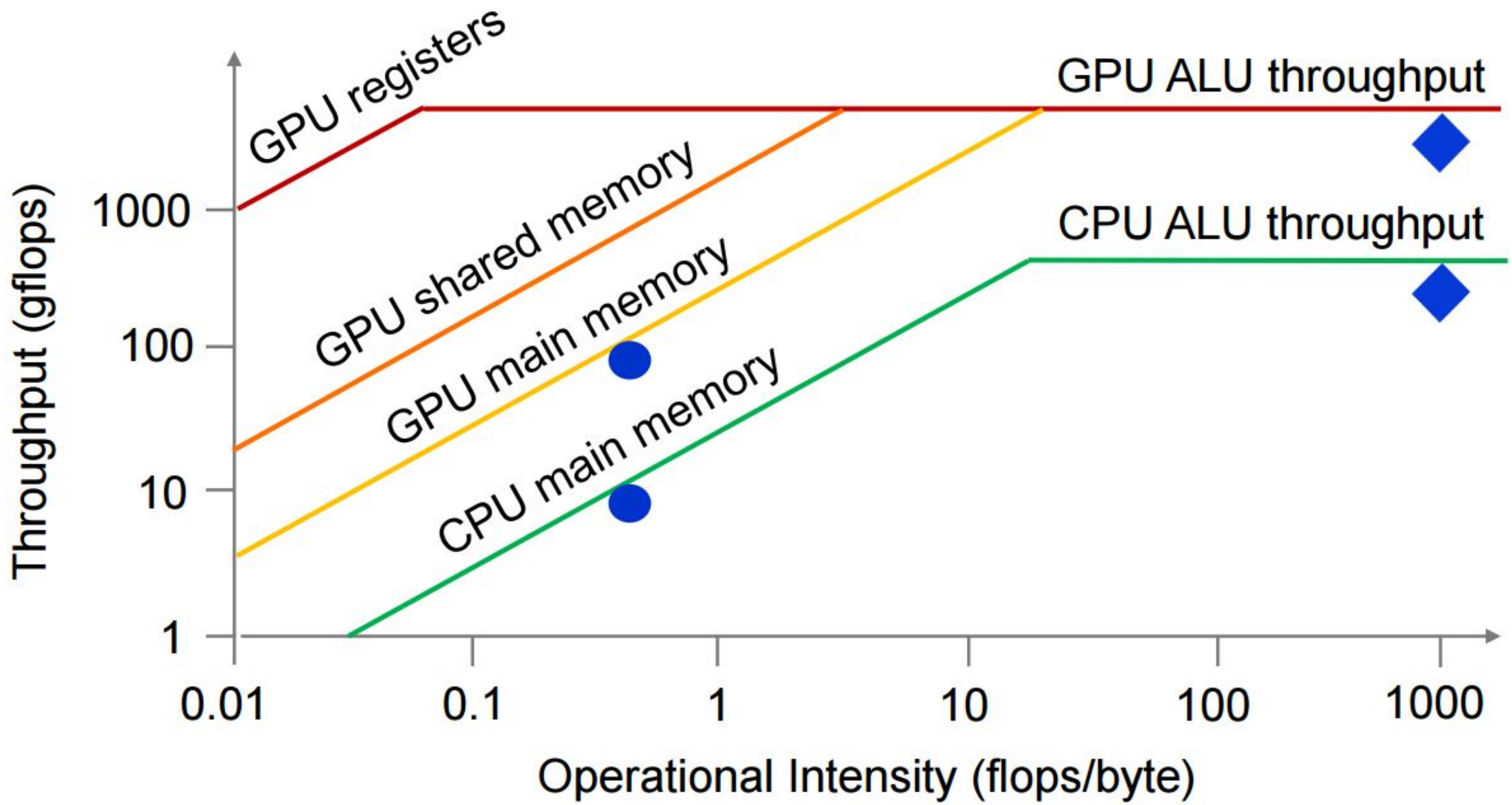
- Canny et al.: *"Big Data Analytics with Small Footprint: Squaring the Cloud"*, KDD 2013
- Canny et al.: *"BIDMach: Large-scale Learning with Zero Memory Allocation"*, NIPS 2013 BigLearn workshop

- Roofline design establishes fundamental performance limits for a computational kernel.



Roofline design

- Dense matrix multiply ◆
- Sparse matrix multiply ●



Roofline design

System	Nodes/cores	Dim	Error	Time (s)	Cost	Energy (KJ)
Graphlab	18/576	100		376	$3.50	10,000
Spark	32/128	100	0.82	146	$0.40	1000
BIDMach	**1**	**100**	**0.83**	**90**	**$0.015**	**20**
Spark	32/128	200	0.82	544	$1.45	3500
BIDMach	**1**	**200**	**0.83**	**129**	**$0.02**	**30**
BIDMach	**1**	**500**	**0.83**	**600**	**$0.10**	**150**

Matrix factorization on the complete Netflix dataset

System	nodes /cores	nclust	Error	Time (s)	Cost	Energy(KJ)
Spark	32/128	256	1.5e13	180	$0.45	1150
BIDMach	**1**	**256**	**1.44e13**	**320**	**$0.06**	**90**
Sk-Learn	1/8	256		3200x4 *	$1.0	10
Spark	96/384	4096	1.05e13	1100	$9.00	22000
BIDMach	**1**	**4096**	**0.995e13**	**735**	**$0.12**	**140**

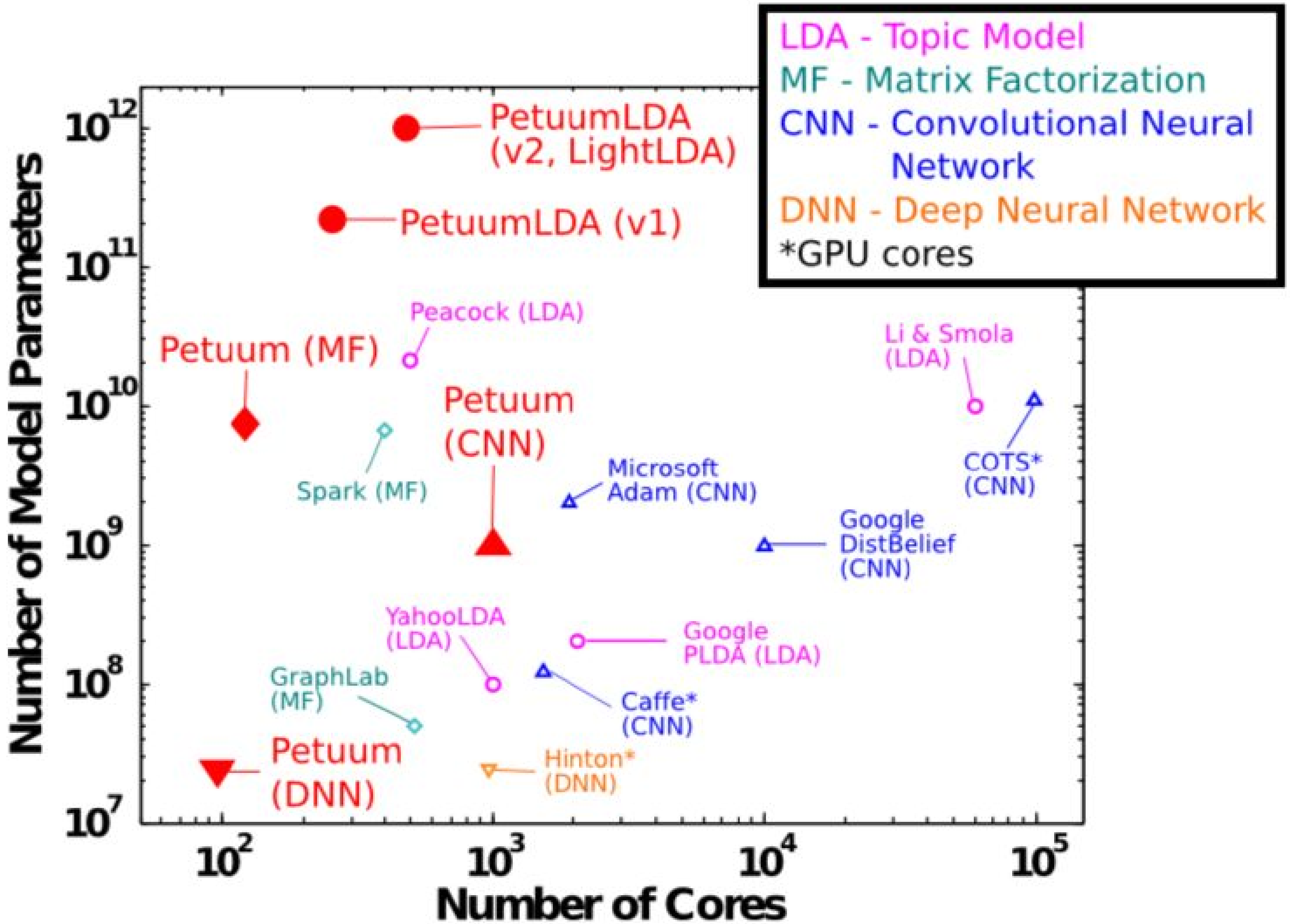
Kmeans on MNIST-8M

Petuum

- Xing et al.: *“Petuum: A New Platform for Distributed Machine Learning on Big Data”*, KDD 2015

Petuum

- Distributed machine learning
- Exploit common properties of ML algorithms to achieve efficient implementation



Source: Xing (2015)

Petuum vs. state of the art (2015)

Takeaway

- Use smart systems and algorithms for large-scale ML
- Don’t need a cluster to do most things

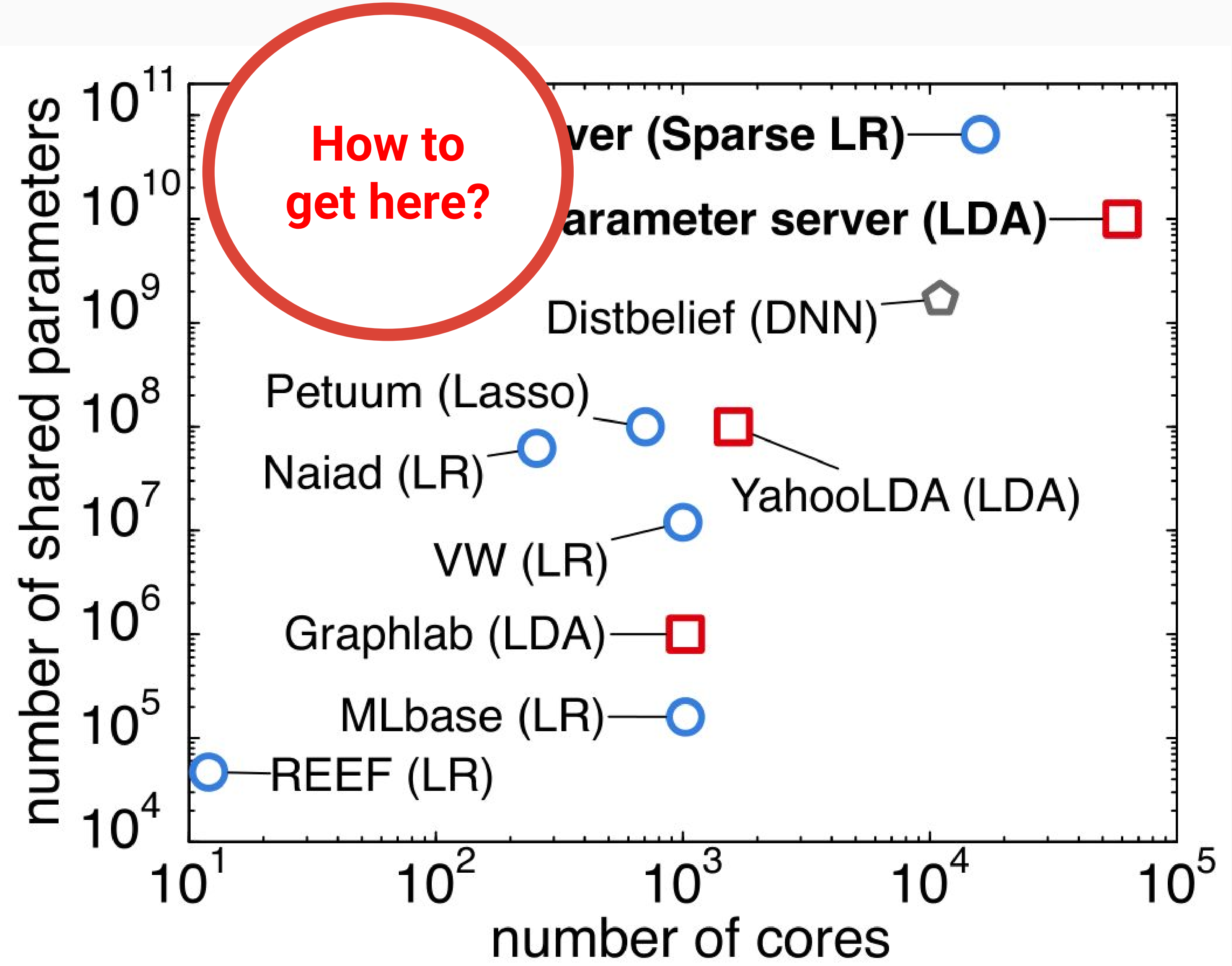
Challenges and research issues

Current issues

- Communication efficient algorithms and systems

Current issues

- Communication efficient algorithms and systems
- Resource efficient algorithms and systems



Active research area

Future issues

- Integrated systems and ML research: “Symbiotic” systems and algorithms
    - How can we further take advantage of ML program properties to build better systems?
    - How can we reconcile view of “ML people” and “systems people” to achieve progress on both sides?

Future issues

- “Symbiotic” systems and algorithms research
- Streaming/online learning

Future issues

- “Symbiotic” systems and algorithms research
- Streaming/online learning
- Exascale ML

Thank You.

@thvasilo
tvas@sics.se

References

- Silver (2016): Mastering the game of Go with deep neural networks and tree search
- Karpathy: CS231n Course material
- Dolhansky: Artificial Neural Networks Blog post series
- Champandard (2016): Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork
- Görnerup (2015): Knowing an Object by the Company it Keeps: A Domain-Agnostic Scheme for Similarity Discovery
- BID Data Toolkit: BID Data Project Website
- CMU Petuum: Petuum Project

Other references found in the text.