Deep Learning and the Future of Search: Objects, Apps, and Beyond

Adam M. Smith, UC Santa Cruz
amsmith@ucsc.edu

IEEE-CNSV
December 11, 2018

I found out I'd be giving this talk on Sunday night.

The 475 students in my *Intro to Programming* class are taking their final exam this week.

I'll be a bit informal.

About me

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

Assistant Professor of Computational Media
Jack Baskin School of Engineering

TERRACOM communications

Los Alamos NATIONAL LABORATORY EST. 1943

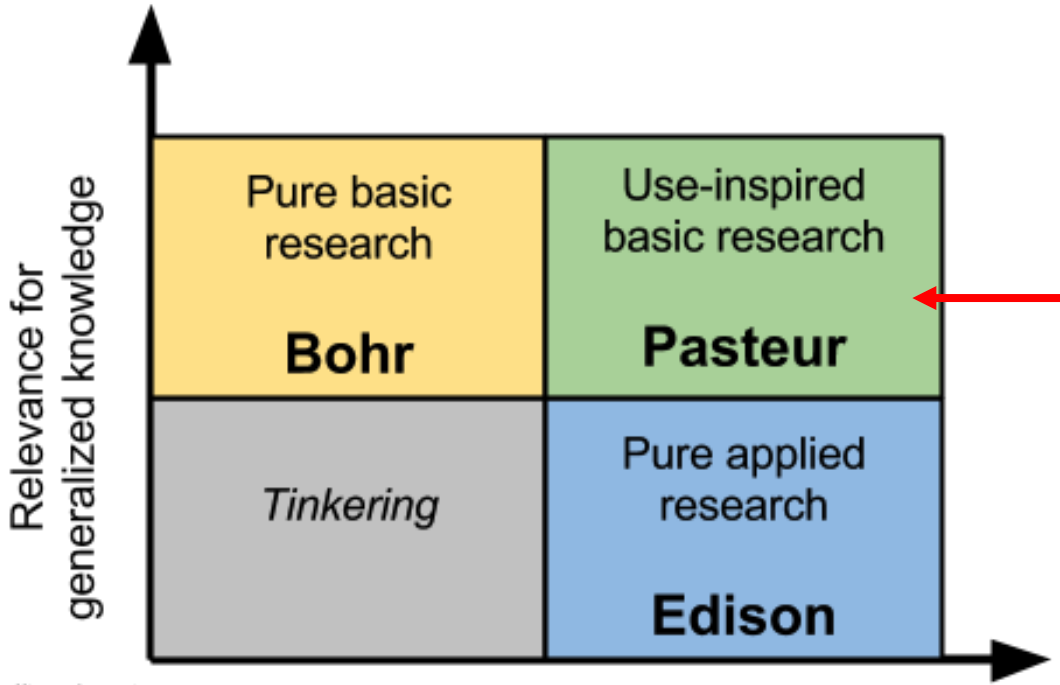
AMES RESEARCH CENTER

Enlearn

Google

Microsoft

Research philosophy



I like this quadrant.

Tonight

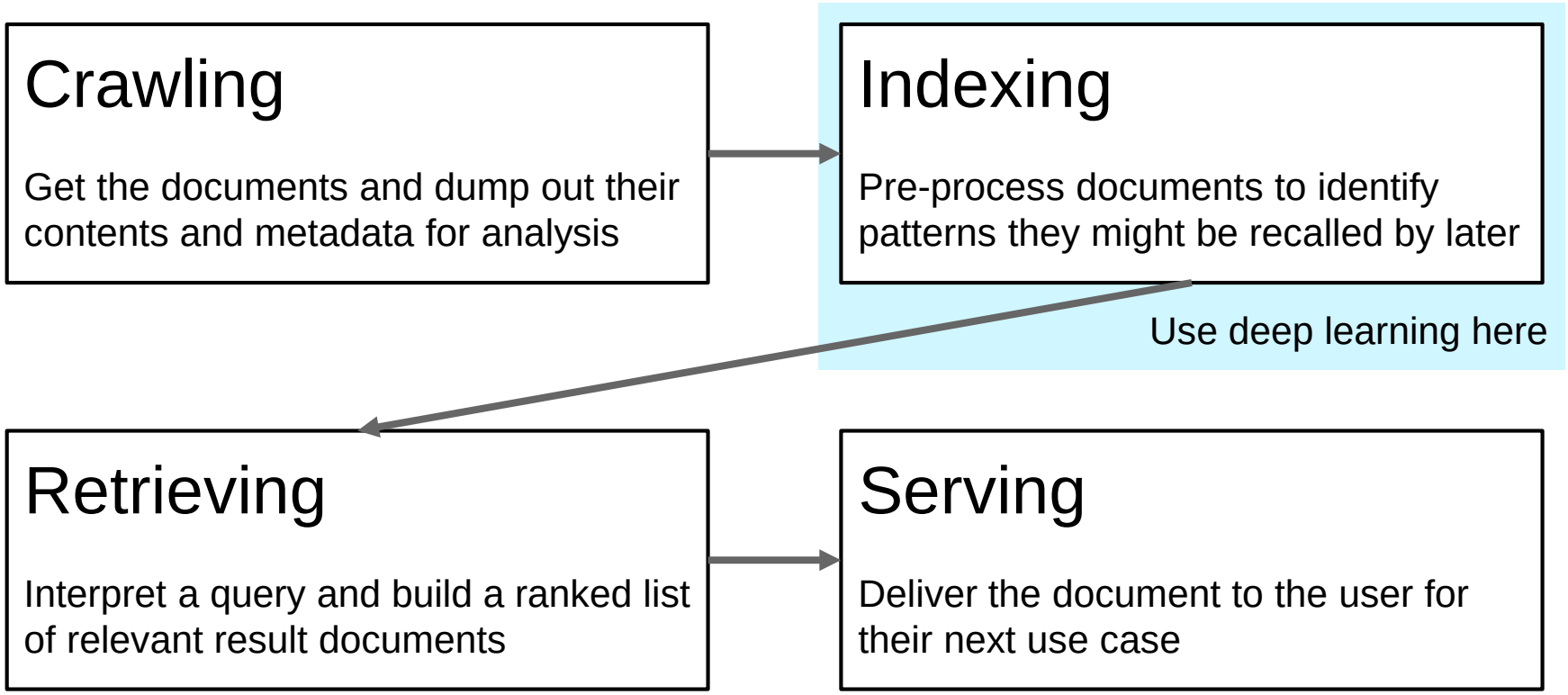
- **Technology behind a new generation of search engines**
- Elements of basic AI research (perception, action, and understanding)
- Demo videos from tangible systems with near-term deployments

Survey time!

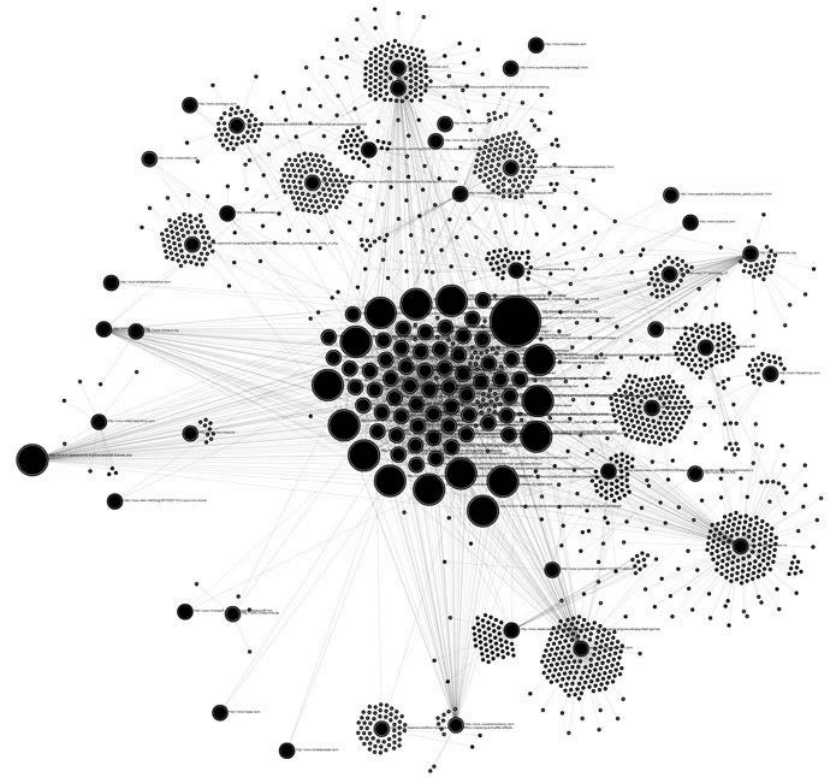
(in 1997)

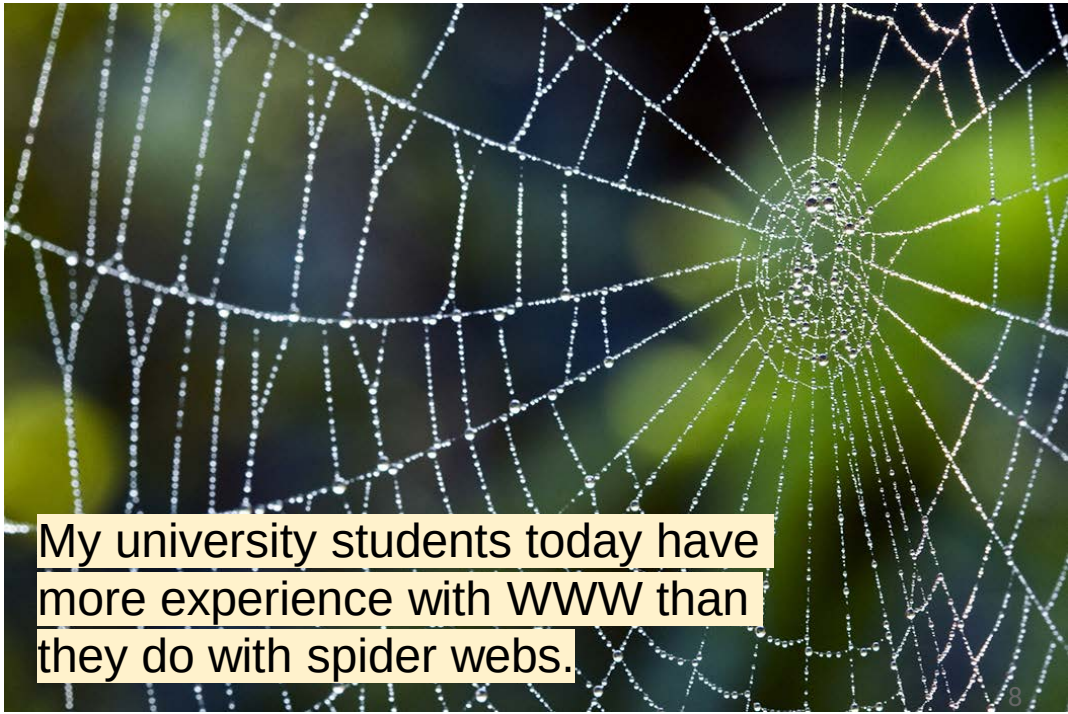
DuckDuckGo

Core processes in (web) search engines



Crawling *(skip for now; imagine content already in DB)*





Indexing *(oversimplified)*

Keyword extraction:

- Names (from a pre-existing list)
- Numbers
- Other key phrases
- (not) stop words

Automated feature extraction:

- n-grams of characters and words
  - “lol” vs “lolololololol” and “looooool”

Result:

- A bag / multi-set of keywords:
  {“cat”: 1, “food”: 1, “cat food”: 1}
- A very sparse vector:
  [0,0,0,0,0,0,0,1,0,0,00,1,0,0,0,1,0,..]
- A dense vector (after random projection):
  [-0.024,+0.282,-0.024,+0.452]

A word cloud based on my text notes for this talk (generated with https://www.wordclouds.com/)

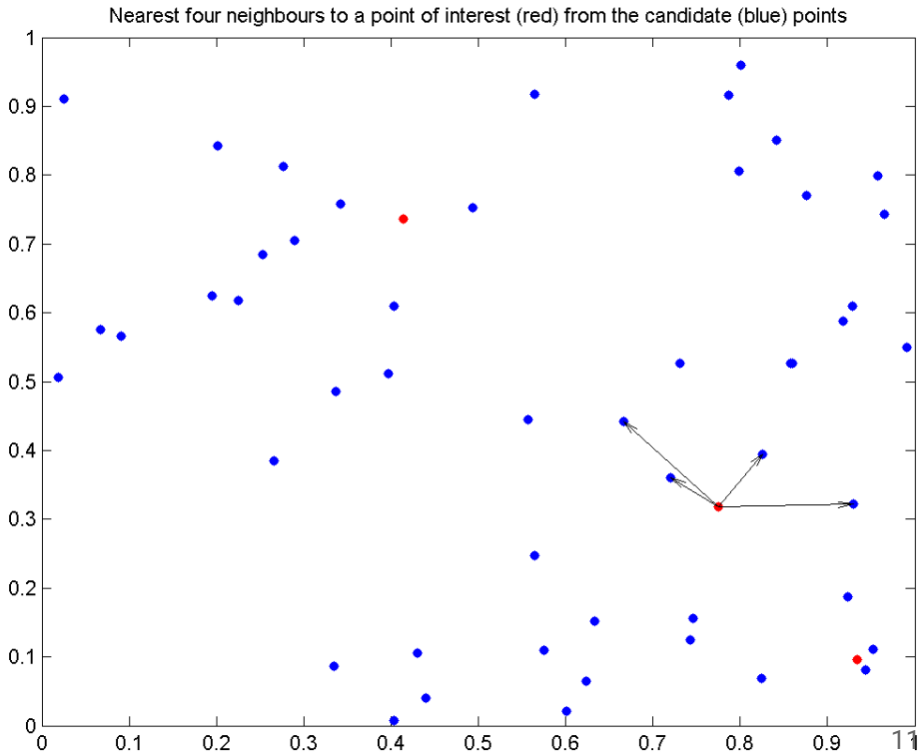
Retrieving *(vector space model)*

Query formation:

- "cat food":
  Q = termVec("CAT") + termVec("FOOD") + termVec("CAT FOOD");

Nearest-neighbor retrieval:

- Return a top-*k* list of documents, sorted by vector distance (L1 or L2 common)



Aside

Ideas like resilient distributed storage and map-reduce computation are arguably byproducts of the need to build Indexing and Retrieving systems that matched the scale of the web.

Cloud technologies are the generalization of this infrastructure.

We'll invent new ideas to power the next generation of search engines.

Manifold technologies (made-up term) will be the generalization of this infrastructure. Time-space-identity vectors will tell apps how to personalize themselves for you without requiring your data up front.

Non-text search: Audio

Shazam (https://www.shazam.com/apps)

Indexing:

- Rolling windows of audio (a few seconds each) are embedded into a vector space using DSP (https://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf)
- Vector is quantized into a codewords (~keywords that can be re-identified even if there is background noise) → *perceptual hashing*

Retrieval:

- Several windows of query audio are sampled and encoded, then sequences are matched

Non - text search: Images

Google Image Search (https://images.google.com/)

Indexing:

- Text features extracted from context of an image file found on the web
- Text from image itself extracted with OCR
- “Scene descriptor” vector is computed from appearance properties

Retrieval:

- **Conceptual**: The query image is analyzed for text features and then text retrieval follows
- **Appearance**: A few crops of the image are generated and appearance descriptor matching follows

When the text processing model falls apart

Synonymy

The document you want uses similar but distinct terminology to what you used in the query

This hurts *recall* (a metric capturing chance that a relevant document is actually returned)

“Cat food” versus “kitten dinner”

Muffled mic or distracting background noise in audio retrieval

Polysemy

Words in your query also mean something else, and results for the other concept overwhelm the others

This hurts *precision* (a metric capturing chance that a returned document is actually relevant)

“bank” (synsets on WordNet)

Some segments of music audio are highly ambiguous (kick drum intro for EDM)

Another failure mode

Adversaries

Someone carefully crafts a document so as to spoof its indexed representation

“Cat kitten kitty feline whisker paw *bitcoin* purr meow”

Google’s **PageRank** and **anchor text indexing** were fascinating historical responses to early SEO efforts.

Is nobody doing SEO for rich-media search engines yet???

Latent semantic analysis

Can we somehow **transform** the document vectors to account for **synonymy** and **polysemy**?

While we're at it, can we **compress** the vectors into a lower dimensional space (one we might even want to **look at** on the screen)?

Latent semantic analysis (LSA) is a technique that partially addresses these concerns while being very simple apply (a linear transformation).

I'll use examples from this excellent tutorial: https://technowiki.wordpress.com/2011/08/27/latent-semantic-analysis-lsa-tutorial/

The term - document matrix

i. The Neatest Little Guide to Stock Market Investing
ii. Investing For Dummies, 4th Edition
iii. The Little Book of Common Sense Investing: The Only Way to Guarantee Your Fair Share of Stock Market Returns
iv. The Little Book of Value Investing
v. Value Investing: From Graham to Buffett and Beyond
vi. Rich Dad's Guide to Investing: What the Rich Invest in, That the Poor and the Middle Class Do Not!
vii. Investing in Real Estate, 5th Edition
viii. Stock Investing For Dummies
ix. Rich Dad's Advisors: The ABC's of Real Estate Investing: The Secrets of Finding Hidden Profits Most Investors Miss

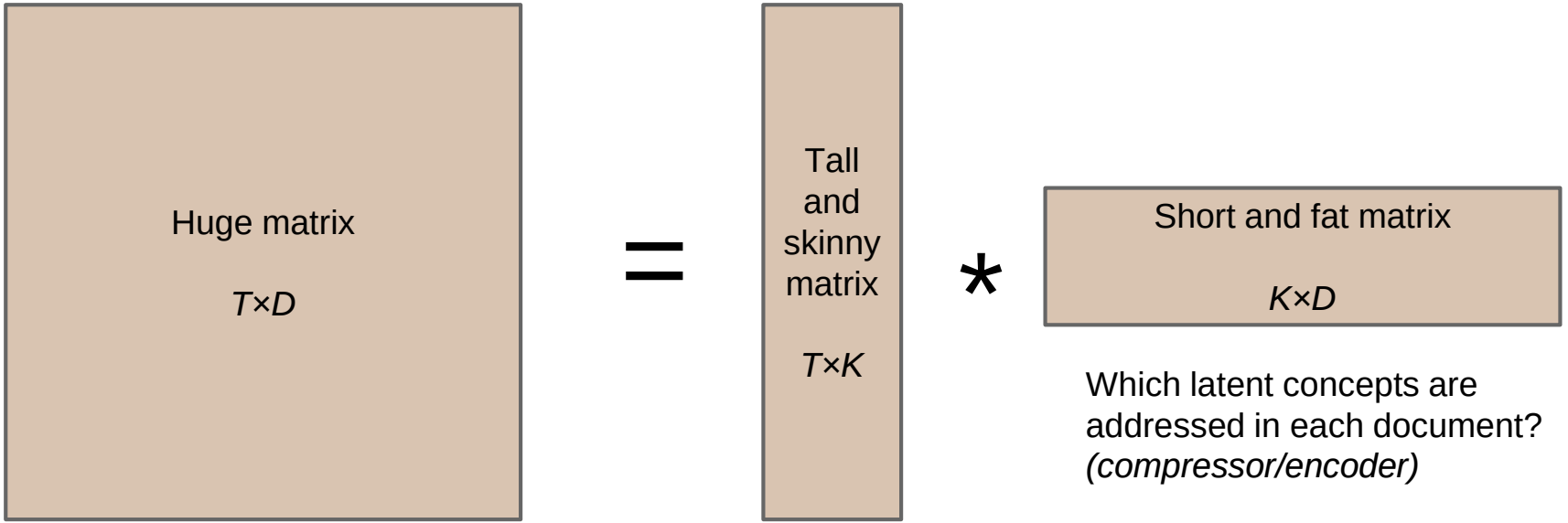
The corpus (book titles)

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

The matrix

This is the same math behind many other applications,
such as predicting movie ratings.

Truncated singular value decomposition

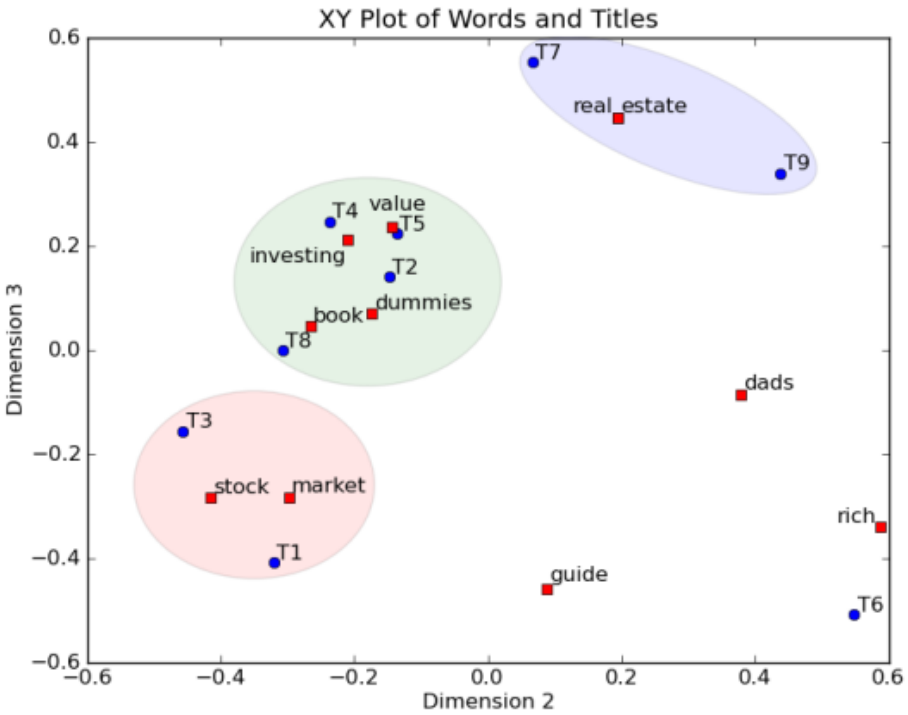


Which documents include which terms?
(training data)

Which words are used to express each latent concept?
(decompressor/decoder)

Which latent concepts are addressed in each document?
(compressor/encoder)

Embedding into latent space



The horizontal and vertical dimensions don't have any intuitive meaning. *What does it mean to have a count of -0.4 of abstract keyword #3?*

Keywords float near the documents in which they are mentioned.

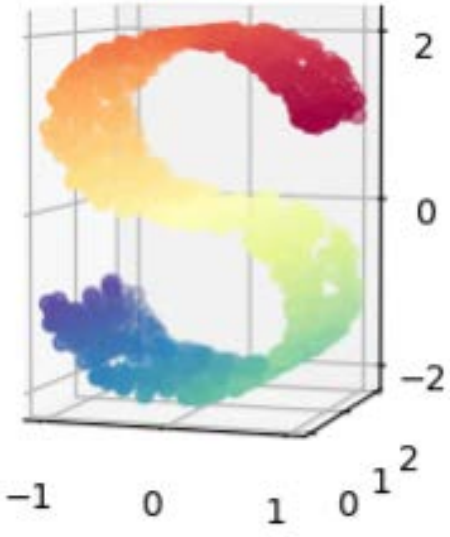
New books can be added to our 2D index by just multiplying by the term-concept matrix, without re-running LSA. Query vectors can be constructed in the same way.

A conceptual shift

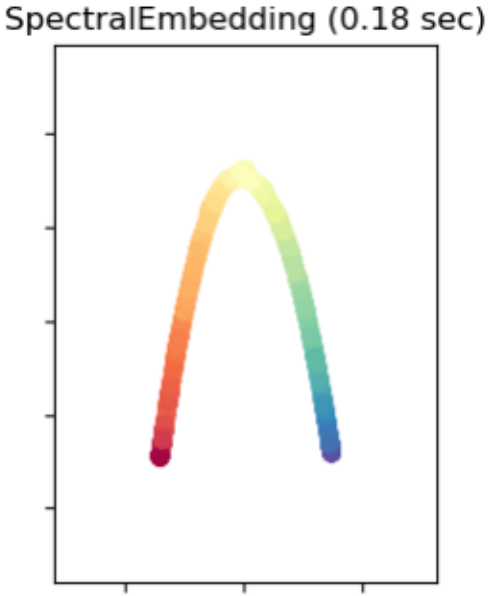
Before: Vectors are the compact, approximate version of keyword counts. They are an implementation detail that speeds up keyword matching.

After: Keyword multi-sets are a noisy, discrete expression of cleaner geometric concepts. They are quantized representation with unintended quantization error.

Alternative view of indexing: embedding



Indexing is about **mapping** from the noisy content space to the clean (and usually lower dimensional) latent **semantic** space.



https://scikit-learn.org/stable/auto_examples/manifold/plot_compare_methods.html

Word vectors

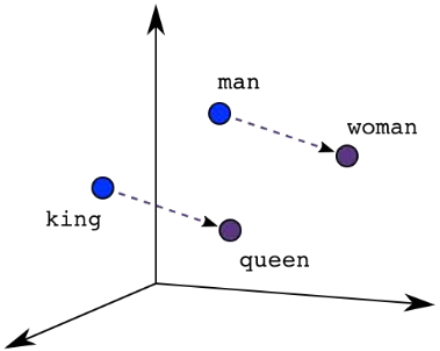
You shall know a word by the company it keeps (Firth, J. R. 1957:11)

"I saw **two tabby <u>cats</u> with dirty** paws yesterday."

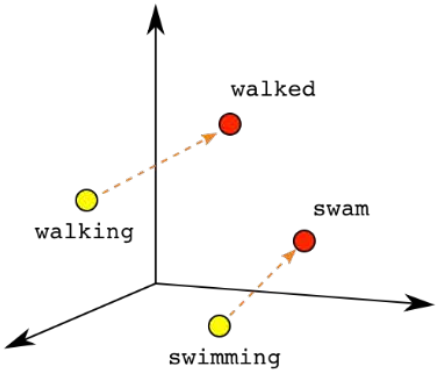
"cats" → "two" (are countable objects)
"cats" → "tabby" (can have silky textures)
"cats" → "with" (...)
"cats" → "dirty" (can become dirty)

This is *input-output* data that can be used in supervised machine learning. The intermediate vector representation associated with each word here is called a word vector.

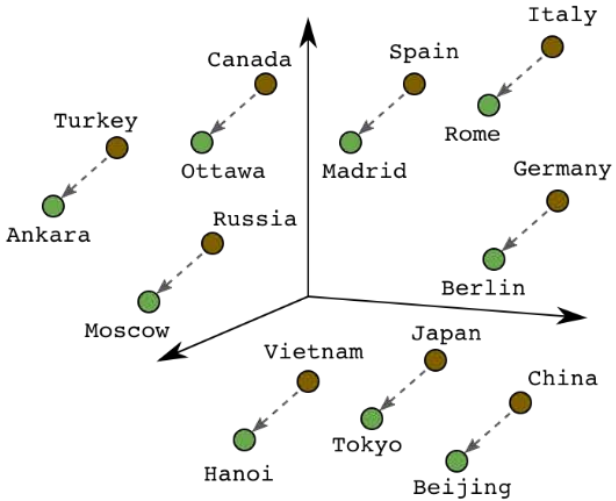
Word vectors have interesting properties



Male-Female



Verb Tense

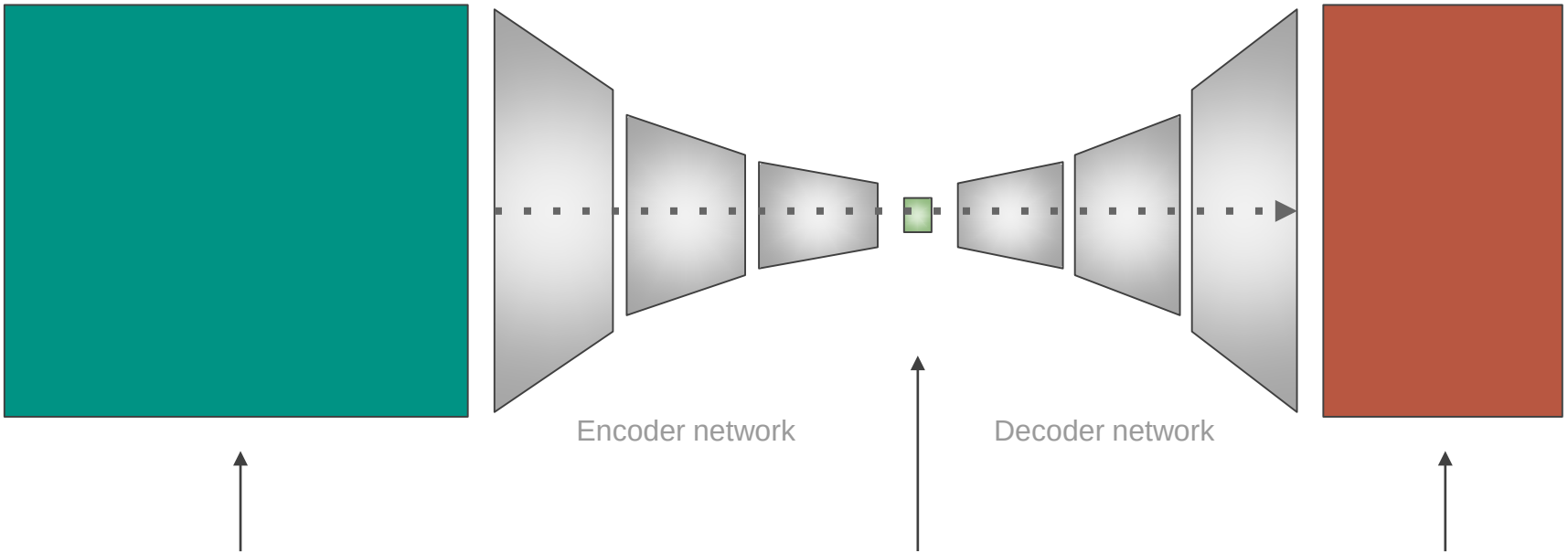


Country-Capital

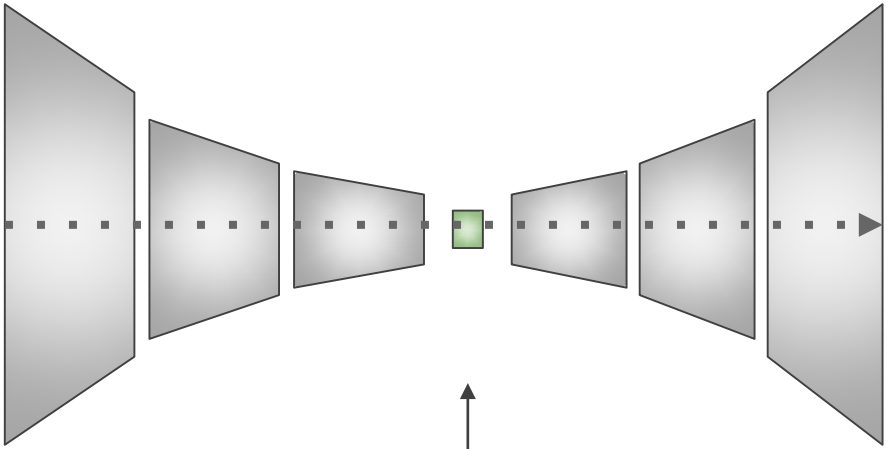
https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space

Use content to predict context

Traditional ML focused on scalar and vector data types, but now we can use tensors, sequences, trees, graphs, and and more!



Pix2Mem, a proxy prediction task



Encoder network

Decoder network

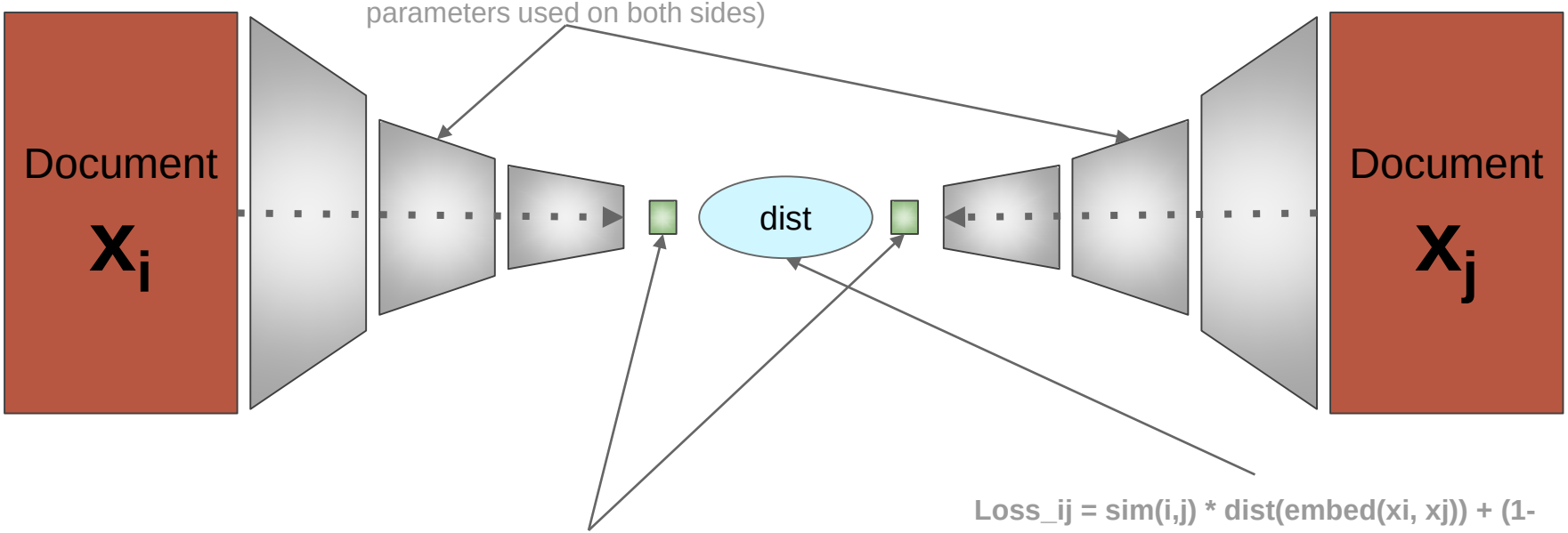
Input screenshot

256-dimensional
bottleneck representation

Output memory state

(Deep) Manifold learning

Training on proxy tasks are a heuristic for getting good embeddings. Manifold learning tries to optimize the manifold structure directly.



Let's put it all together:

AR shopping experience in IKEA Place app

Crawling: IKEA has a database of images of their own products. They know when two different pictures show the same product in different settings.

Indexing: Apply a deep convolutional neural network trained with manifold learning techniques (Siamese networks with contrastive loss).

Retrieving: Nearest-neighbors by L2 distance.

Serving: Load a 3D model of the furniture item and let people move it around in their home.

Read more about a similar app from Pinterest.

A video demo of visual search in the IKEA Place AR app (watch fullscreen for detail).

Browsing the embedding space

SpaceSheet: A spreadsheet-like interface for performing simple arithmetic on semantic vectors and seeing a preview of the content best matching the result vector

ArtsExperiments: A browseable landscape of famous paintings with emergent grouping by visual similarity.

Zooey's game moment visualizer: A 3D map of the interesting moments of Super Metroid.

Traditional search engines print the sorted list of results in a linear list intended for reading (while throwing away relative distances and context from more distant results).

2D/3D browsing interface might make better use of our visual human perception skills, our searching and gathering instincts, and help people make use of results outside the top 10.

Another application:

Finding moments in interactive media

Crawling: Run videogames in an emulator, press random buttons (new paper about improved crawling techniques). Keep screenshots and snapshots of memory bytes.

Indexing: Embedding function maps screenshots to vectors (trained on Pix2Mem task).

Retrieving: Nearest-neighbors by L1 distance.

Serving: Just show screenshot (but ideally offer VM image that you can download and continue to play).

More in Crawling, Indexing, and Retrieving Moments in Videogames.

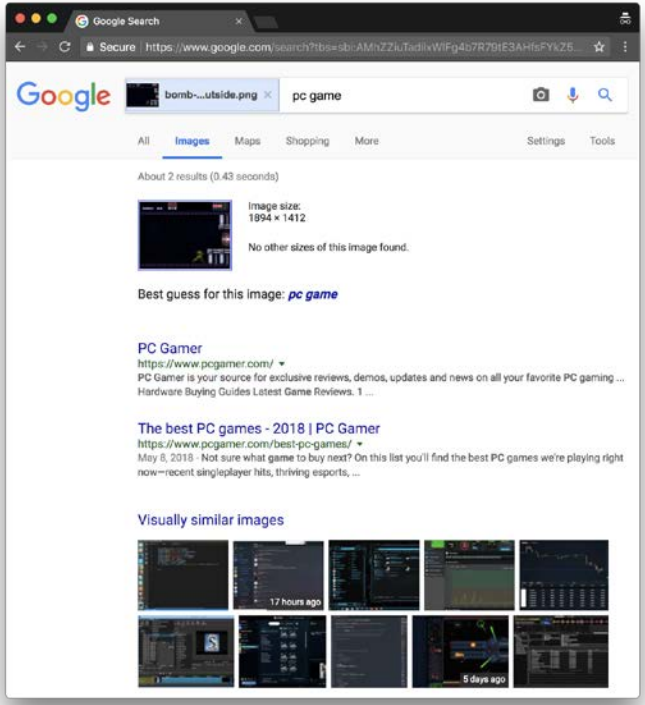
A very early tech - demo from my lab, also showing relevance feedback to improve query

Comparison with Google Image Search

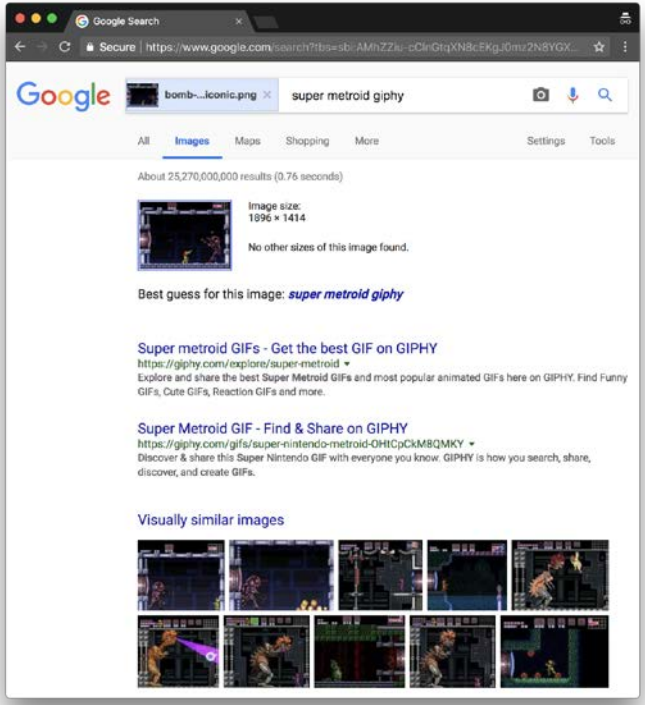
A: blurry capture of an unpopular moment

B: crisp capture of an iconic boss battle

Google’s results



A: “pc game”



B: “super metroid giphy”

Our results

A: 3/10 images of the precise room, 3 more are from immediately adjacent room (relevant!)

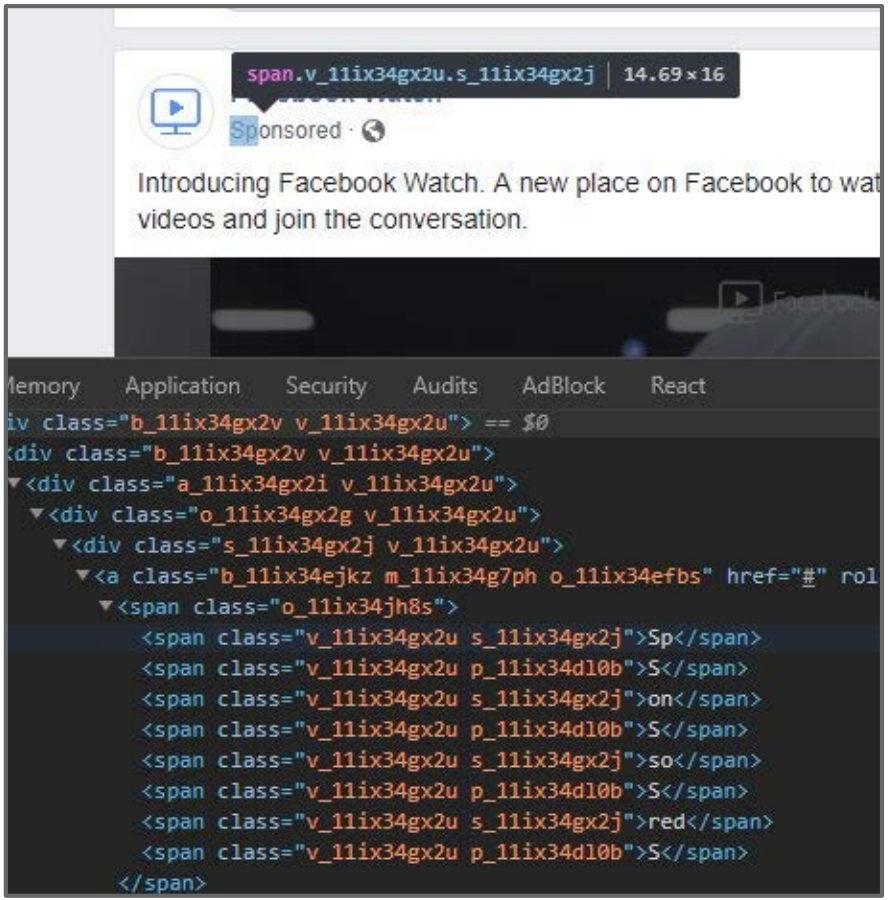
B: 9/10 are from the precise room, but none show the actual boss battle

Why index moments in SNES games?

Sequence: Atari 2600, GameBoy, SNES, N64, (then Android and Unity desktop games)

Use-inspired research: We want to be able to index apps freshly uploaded to mobile app markets.

- Can't always trust screenshots and description from developers or even comments from users (same problem faced by AltaVista)
- Connectivity between moments via interaction provides additional clues to structure/meaning (this is our BackRub/PageRank idea)
- Malware and other content might be hidden behind the gate of gameplay interaction (we saw how the arms race evolved for web pages, so we can predict what's going to happen for apps and games)



Next-gen web crawlers (and ad blockers) will need to do perceptual inference on web pages and simulate clicks to take action in order to figure out what a web page really does and how one page leads to another.

https://twitter.com/aaronkbr/status/1071214578980261888

Some notes on the future:

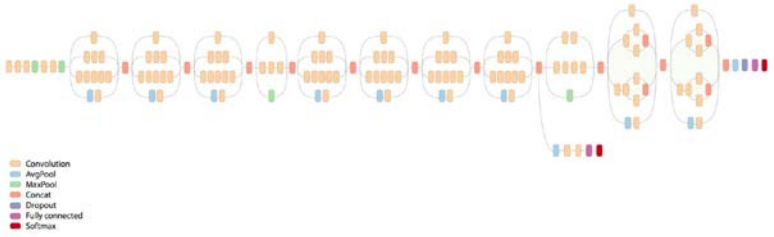
What is going to impact applications today and in the next two years?

(It’s 2am, so I’ll put fewer pictures into the next few slides…)

Representation learning

ConceptNet
Numberbatch

fastText



Pre-trained models and *pre-embedded data tables* give your system the knowledge of deep networks without requiring the expertise, time, or hardware to train them yourself!

Future DL applications might not be so data hungry.

A new balance of power between individuals, corps, and govs: It's not necessary for pictures of your friends to leave your phone to get SoTA face recognition ability in your app.

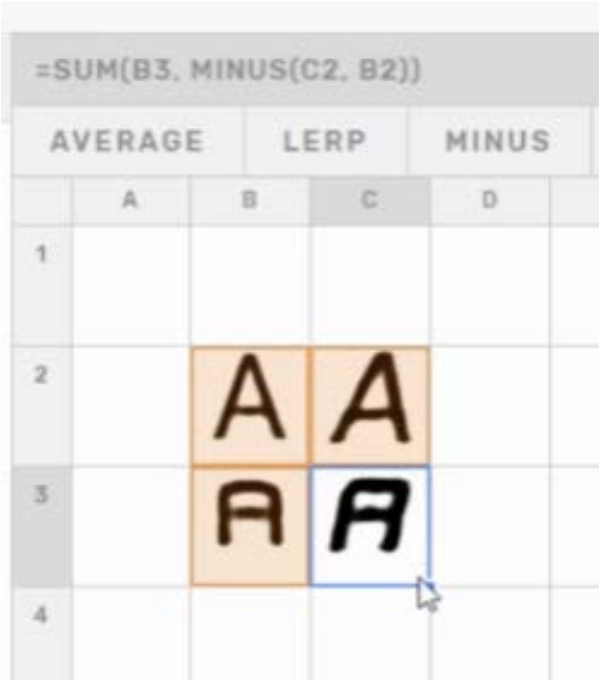
Multi - modal search

By aligning word vectors across languages, you can search with keywords not considered during original indexing (alignment can be as simple as rotating the vector space).

By learning a deeper mapping between modalities, you might search any kind of data by any kind of query

- Search for image by description in voice
- Recommend camera app settings by GPS location, compass direction, and time

Vector analogies



+bold-italic (SpaceSheet)

“A is to B what C is to {the thing I’m actually looking for}”

Point at something you can’t quite describe!

Good for mushy concepts like style, mood, or personality.

Good for navigating the subtlety of a language you don’t speak.

Vector analogies are 10+ years old, but nobody has used them for forming queries in a search engine before???

On-device and in-browser computation

Good software and hardware acceleration for both prediction (using a deep model) and training (updating it with more data) exist now.

Previously, many designs needed to offload perceptual AI tasks to the cloud. Now “distillation” (making a smaller/cheaper network behave like a big/costly one) and “quantization” (getting rid of costly float operations) are changing the balance.

GPUs and FPGAs are now integral parts of Google/Bing, and they’ll soon be a key part of local search tools that can’t or don’t want to talk to the cloud.

Some takeaways:

Crawling: Take actions to uncover new content (even navigating between web pages is starting to feel more like gameplay than reading a hypertext document)

Indexing: Embed the tings you want to find into a space where relative distances and directions are meaningful (using context prediction tasks or manifold learning techniques)

Retrieving: Form a point from the query, collect what's nearby (simple idea, hard part is typical Big Data VVV challenges)

Serving: Don't just make a list; make a space you can browse to see larger patterns

Deep Learning and the Future of Search: Objects, Apps, and Beyond

Adam M. Smith
amsmith@ucsc.edu

IEEE-CNSV
December 11, 2018