Large-scale language and vision models of concept representation

University of Pavia

Feb 3-6 2023

Dr. Fritz Günther

www.fritzguenther.de

Modelling mental representations

Introduction

- What is a TOWER?

Introduction

- What is a TOWER?
- Where did you get this information from?

What are "representations"?

- We can't store/manipulate things "in our head", just representations of them

What are "representations"?

- We can't store/manipulate things "in our head", just representations of them
- Representation as *"an encoding of some information, which an individual can construct, retain in memory, access, and use in various ways"* (Smith, 1998)

What are “representations”?

- What is our representation of
  - `LION`
  - `WORLD`
  - `FAITH`
  - `DIFFERENCE`

What are “representations”?

- What is our representation of
  - `LION`
  - `WORLD`
  - `FAITH`
  - `DIFFERENCE`
- Can we measure representations?

What are “representations”?

- What is our representation of
  - `LION`
  - `WORLD`
  - `FAITH`
  - `DIFFERENCE`
- Can we measure representations?
- We can’t observe these representations directly, but have to infer them

Modelling representations

- How do we go scientific about representations?

Modelling representations

- How do we go scientific about representations?
- We need objective, quantitative models:

Modelling representations

- How do we go scientific about representations?
- We need objective, quantitative models:

  "The problem of hand-coded representations is the most serious problem facing computational modeling as a scientific enterprise. All models are sensitive to their representation, so the choice of representation is among the most powerful wildcards at the modeler's disposal." (Hummel & Holyoak, 2003)

Inferring representations from behavioral data/ "the outcome level"

- Popular method
  (e.g., de Deyne et al., 2016; Kenett et al., 2017; Hebart et al., 2020)
  - Collect behavioral data (for example, word similarity ratings or free associations)

Inferring representations from behavioral data/ "the outcome level"

- Popular method

  (e.g., de Deyne et al., 2016; Kenett et al., 2017; Hebart et al., 2020)

  - Collect behavioral data (for example, word similarity ratings or free associations)
  - Estimate the representational system that most likely produced these data

Inferring representations from behavioral data/ "the outcome level"

- Popular method

  (e.g., de Deyne et al., 2016; Kenett et al., 2017; Hebart et al., 2020)

  - Collect behavioral data (for example, word similarity ratings or free associations)
  - Estimate the representational system that most likely produced these data

Westbury, 2016

Inferring representations from the outcome level

- Problem: May be a good description, but does not offer an explanation.

  "One issue with all three of these classic models is that none ever did actually learn anything." (Jones, Willits, & Dennis, 2015)

Inferring representations from the outcome level

- Problem: May be a good description, but does not offer an explanation.

  "One issue with all three of these classic models is that none ever did actually learn anything." (Jones, Willits, & Dennis, 2015)

- What exactly makes people think that LION is more similar to TIGER than BATHTUB?

Inferring representations from the outcome level

- Problem: May be a good description, but does not offer an explanation.

  "One issue with all three of these classic models is that none ever did actually learn anything." (Jones, Willits, & Dennis, 2015)

- What exactly makes people think that LION is more similar to TIGER than BATHTUB?
- Assumption: This is a function of our *experience*

Building from experience: Starting from the input level

- Aim: Build representations as a function of the input experienced by the system

Building from experience: Starting from the input level

- Aim: Build representations as a function of the input experienced by the system
- "When the system, over the course of its life, encounters this data, it will develop these representations"

Building from experience: Starting from the input level

- Aim: Build representations as a function of the input experienced by the system
- “When the system, over the course of its life, encounters this data, it will develop these representations”
- We need a lot of data to approximate this experience!

Building representations from language experience

Learning meaning from language experience: A demonstration

What is a CATAPHRACT?

Learning meaning from language experience: A demonstration

What is a CATAPHRACT?

D1 In my opinion, the main goal of CATAPHRACTS was to fight heavy infantries.

Learning meaning from language experience: A demonstration

What is a CATAPHRACT?

D1 In my opinion, the main goal of CATAPHRACTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective CATAPHRACT clad in thick iron plates which covered their entire body.

Learning meaning from language experience: A demonstration

What is a CATAPHRACT?

D1 In my opinion, the main goal of CATAPHRACTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective CATAPHRACT clad in thick iron plates which covered their entire body.

D3 These CATAPHRACTS specialised in forming a wedge formation and penetrating enemy formations to create gaps, enabling lighter troops to make a breakthrough.

Learning meaning from language experience: A demonstration

What is a CATAPHRACT?

D1 In my opinion, the main goal of CATAPHRACTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective CATAPHRACT clad in thick iron plates which covered their entire body.

D3 These CATAPHRACTS specialised in forming a wedge formation and penetrating enemy formations to create gaps, enabling lighter troops to make a breakthrough.

D4 Nations in the East occasionally fielded CATAPHRACTS mounted on camels rather than on horses .

Learning from language experience: A demonstration

KNIGHT can be used in the same contexts as CATAPHRACT!

Learning from language experience: A demonstration

KNIGHT can be used in the same contexts as CATAPHRACT!

D1 In my opinion, the main goal of KNIGHTS was to fight heavy infantries.

Learning from language experience: A demonstration

KNIGHT can be used in the same contexts as CATAPHRACT!

D1 In my opinion, the main goal of KNIGHTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective KNIGHTS clad in thick iron plates which covered their entire body.

Learning from language experience: A demonstration

KNIGHT can be used in the same contexts as CATAPHRACT!

D1 In my opinion, the main goal of KNIGHTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective KNIGHTS clad in thick iron plates which covered their entire body.

D3 These KNIGHTS specialised in forming a wedge formation and penetrating enemy formations to create gaps, enabling lighter troops to make a breakthrough

Learning from language experience: A demonstration

KNIGHT can be used in the same contexts as CATAPHRACT!

D1 In my opinion, the main goal of KNIGHTS was to fight heavy infantries.

D2 Shapur II further reformed the army by adopting heavier and more effective KNIGHTS clad in thick iron plates which covered their entire body.

D3 These KNIGHTS specialised in forming a wedge formation and penetrating enemy formations to create gaps, enabling lighter troops to make a breakthrough

D4 Nations in the East occasionally fielded KNIGHTS mounted on camels rather than on horses

The distributional hypothesis

- Words with similar meanings occur in similar contexts
- *You shall know a word by the company it keeps*

Firth, 1957; Harris, 1954; Sahlgren, 2008

Distributional semantic models

- The passionate nurse treats patients in the hospital
- The passionate doctor treats patients in the hospital
- The doctor saved her patient
- A whale travels the ocean

Landauer & Dumais, 1997; Lund & Burgess, 1996

Distributional semantic models

- The passionate nurse treats patients in the hospital
- The passionate doctor treats patients in the hospital
- The doctor saved her patient
- A whale travels the ocean

Distributional vectors:

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

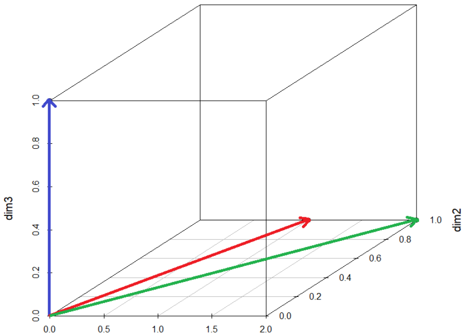
Landauer & Dumais, 1997; Lund & Burgess, 1996

Excursus: Vector algebra

Excursus: Vector algebra

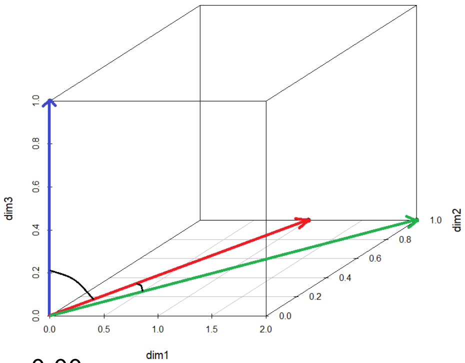
Vector space

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1



Excursus: Vector algebra

Cosine similarity



- $\cos(90^\circ) = 0.00$
- $\cos(00^\circ) = 1.00$
- more similar distribution
  $\Longrightarrow$ smaller angle $\Longrightarrow$ larger cosine similarity

- We define the *n nearest neighbors* of a word as those *n* other words (from a given lexicon) with the highest cosine similarity to that word

Excursus: Getting familiar with R

Excursus: Getting familiar with R

- When it comes to computational models, it usually makes sense (and is more fun) to actually *use* them, not just *hear* about them
- In addition to some web tools, we will also learn to use them in the statistical computing environment R
- First, let's get a basic understanding of R

Excursus: Getting familiar with R

Installing R

- Go to: https://cran.rproject.org/
- Select the version of R according to your OS.
- Install. Default options will fit in most cases. I suggest you to discard 32 bit files (if your OS is 64 bit) and message translations (annoying when you try googling them).

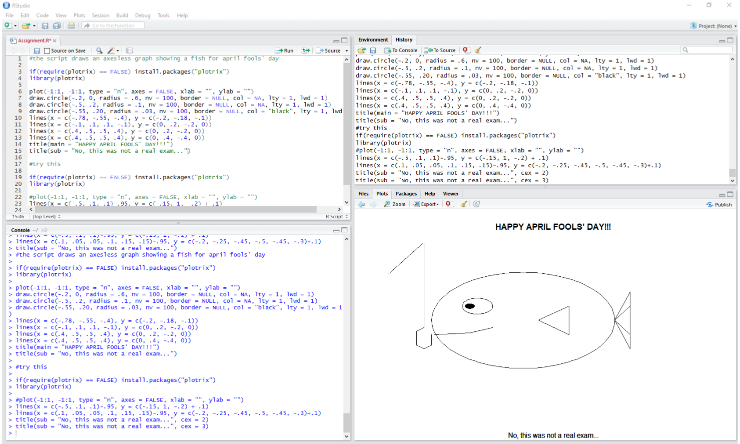
Excursus: Getting familiar with R

Installing Rstudio

- User interface that makes working with R easier
- Go to: `https://www.rstudio.com/`
- DO NOT go to: `http://www.r-studio.com/` (this looks like an expensive data recovery tool, unrelated to R)
- Products > Rstudio > Desktop > Download

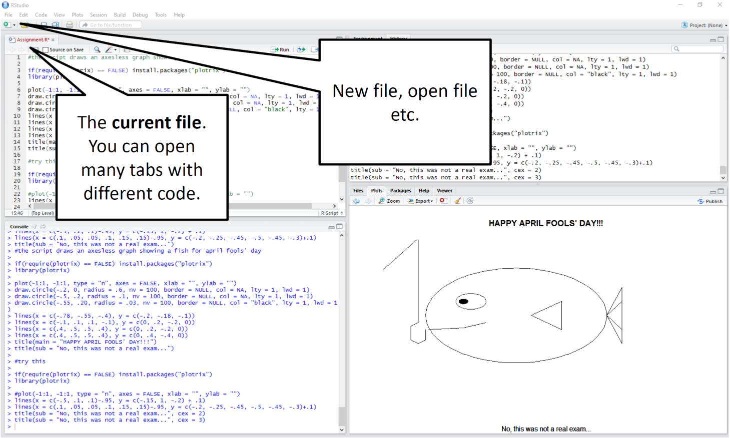
Excursus: Getting familiar with R

Rstudio and its windows



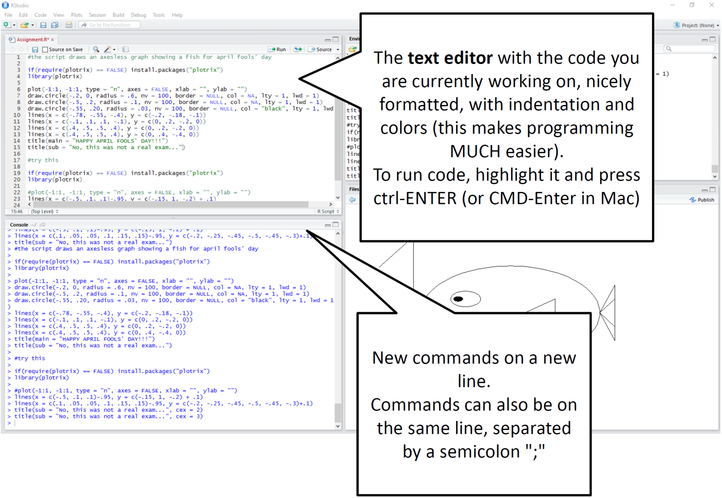
Excursus: Getting familiar with R

Rstudio and its windows



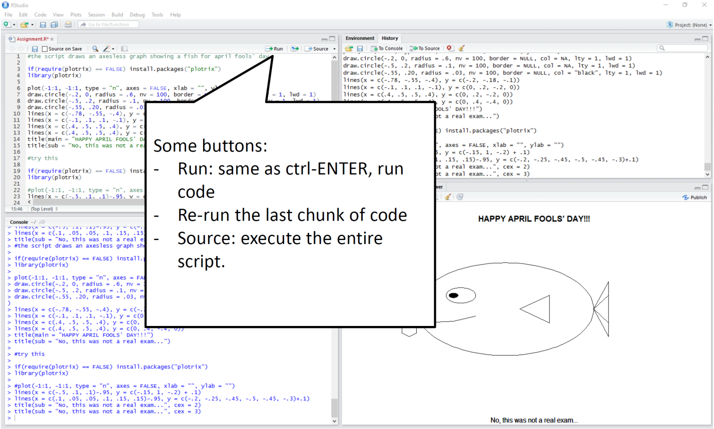
Excursus: Getting familiar with R

Rstudio and its windows



Excursus: Getting familiar with R

Rstudio and its windows

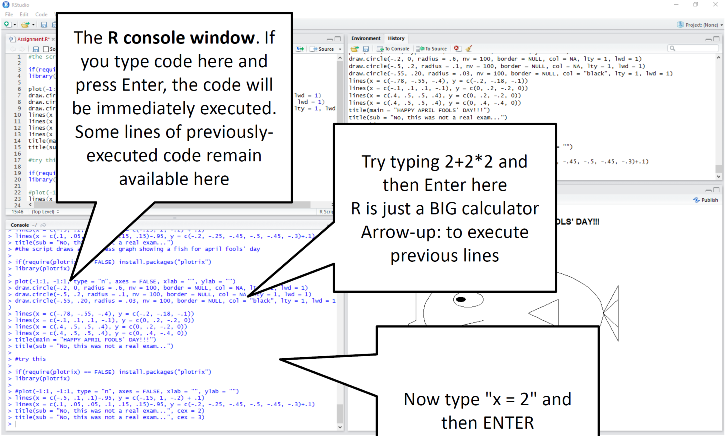


Some buttons:

- Run: same as ctrl-ENTER, run code
- Re-run the last chunk of code
- Source: execute the entire script.

Excursus: Getting familiar with R

Rstudio and its windows



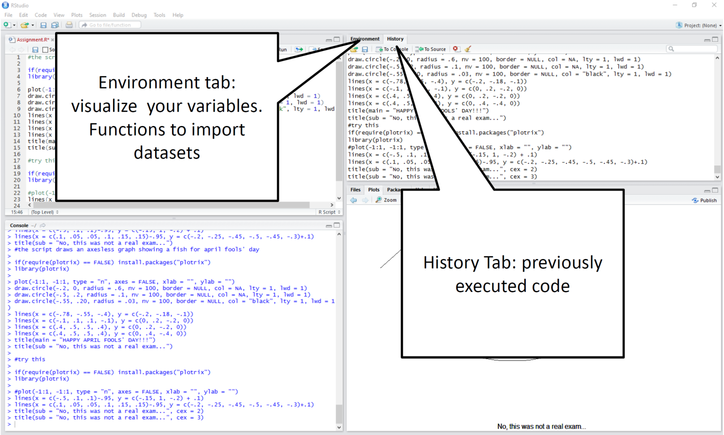
The **R console window**. If you type code here and press Enter, the code will be immediately executed. Some lines of previously-executed code remain available here

Try typing 2+2*2 and then Enter here
R is just a BIG calculator
Arrow-up: to execute previous lines

Now type "x = 2" and then ENTER

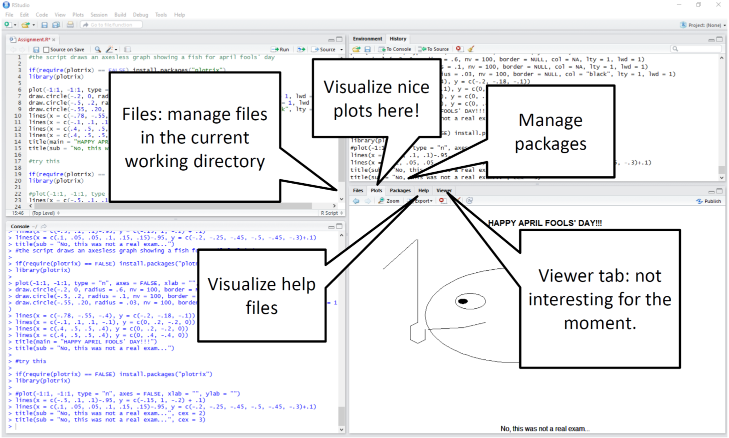
Excursus: Getting familiar with R

Rstudio and its windows



Excursus: Getting familiar with R

Rstudio and its windows



Excursus: Getting familiar with R

Using the R console

- Type `2 + 2*2` into the console and press ENTER
- Type `2^2` into the console and press ENTER
- Type `sqrt(9)` into the console and press ENTER

Excursus: Getting familiar with R

Using R scripts

- Click on "File > New File > R Script"
- Save it somewhere where you can find it, using "File > Save As"
- In this file in the text editor
  - Type `2 + 2*2`
  - Type `2^2`
  - Type `sqrt(9)`
- Select everything (Ctrl+A) anc click on "Run" (the green arrow) or use Ctrl+ENTER

- Scripts are extremely useful: You can later open them again and run the same analysis without having to type anything in the console

Excursus: Getting familiar with R

Variables

- You can assign values to variables using
  `x <- 2`

Excursus: Getting familiar with R

Variables

- You can assign values to variables using
  `x <- 2`
- You can pretty much use any variable name you like
  `topolino <- 2`

Excursus: Getting familiar with R

Variables

- You can assign values to variables using
  `x <- 2`
- You can pretty much use any variable name you like
  `topolino <- 2`
- To see the value of a variable, simply write it and press ENTER
  `topolino`
- You can perform computations with variables
  `topolino + 4`

Excursus: Getting familiar with R

Functions

- Functions take an input (so-called arguments) and return an output
- It's all about the functions
- Simple case with one argument: `sqrt(9)`
- Here, `sqrt()` is the function and `9` is the argument
- You can chain functions:
  `sqrt(exp(9))`
- You can save the output of a function as a variable
  `x <- sqrt(exp(9))`

- Functions can have multiple arguments that do different things:
- Type
  `seq(from = 1, to = 10, by = 1)`
- To see how a function works (incl. which arguments it takes, which output it returns etc), type `?name_of_function`, like `?seq`

- Many functions are not included in the base version of R, but are provided in packages (written by other users)
- To install a package (here, the `lsa` package), simply type and run:
  `install.packages("lsa")`
  (needs to be done only once)
- To access the functions of this package in an R session, use
  `library("lsa")`

Excursus: Getting familiar with R

Vector algebra

- To create a vector (ordered list of numbers), simply use the `c()` function:
  `vec <- c(1,4,9)`

Excursus: Getting familiar with R

Vector algebra

Now you!

- Create a vector called `nurse`: [1, 1, 0]
- Create a vector called `doctor`: [2, 1, 0]
- Compute their cosine similarity using the `cosine()` function in the `lsa` package

Distributional Semantic Models

Distributional semantic models

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- Real DSMs are not built from a few sentences with a few words

Distributional semantic models

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- Real DSMs are not built from a few sentences with a few words
- Rather, built from large-scale language corpora that serve as proxies for our language experience

Distributional semantic models

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- Real DSMs are not built from a few sentences with a few words
- Rather, built from large-scale language corpora that serve as proxies for our language experience
- Real DSMs start with tens of thousands of rows and columns $\implies$ many high-dimensional vectors

Distributional semantic models

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- These are “raw count” vectors

Distributional semantic models

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- These are "raw count" vectors
- These raw counts are typically transformed further:
  - weighting
  - dimensionality reduction

Distributional semantic models

Weighting

- Problem: Cell entries are usually high for frequent words, even if they are not informative
- Example:

	patient	hospital	medicine	ocean	the
nurse	45	22	37	0	1887
doctor	34	45	51	2	2003
whale	1	0	0	112	1654

- These vectors are all very similar (cos $>$ .99!), just because one entry is very large

- Counter-measure: Weighting of cell entries

- Counter-measure: Weighting of cell entries
- Possible option: Pointwise mutual information

$$\mathrm{PMI} = \frac{P(a \wedge b)}{P(a) \cdot P(b)}$$

Distributional semantic models

Weighting

- Counter-measure: Weighting of cell entries
- Possible option: Pointwise mutual information

$$\mathrm{PMI} = \frac{P(a \wedge b)}{P(a) \cdot P(b)}$$

- PMI is
  - lower for higher base frequencies/probabilities of a and b alone
  - higher the more often a and b occur *together*

- Problem: With large language corpora, the vectors become very long (many columns)

Distributional semantic models

Dimensionality reduction

- Problem: With large language corpora, the vectors become very long (many columns)
- Many columns will provide redundant information (= be very similar to other columns) – think of columns such as `doctor`, `physician`, `nurse`, ...

Distributional semantic models

Dimensionality reduction

- Problem: With large language corpora, the vectors become very long (many columns)
- Many columns will provide redundant information (= be very similar to other columns) – think of columns such as `doctor`, `physician`, `nurse`, ...
- Many cell entries will be zero

Distributional semantic models

Dimensionality reduction

- Problem: With large language corpora, the vectors become very long (many columns)
- Many columns will provide redundant information (= be very similar to other columns) – think of columns such as `doctor`, `physician`, `nurse`, ...
- Many cell entries will be zero
- Counter-measure: Dimensionality reduction

Excursus: Dimensionality reduction

Excursus: Dimensionality reduction

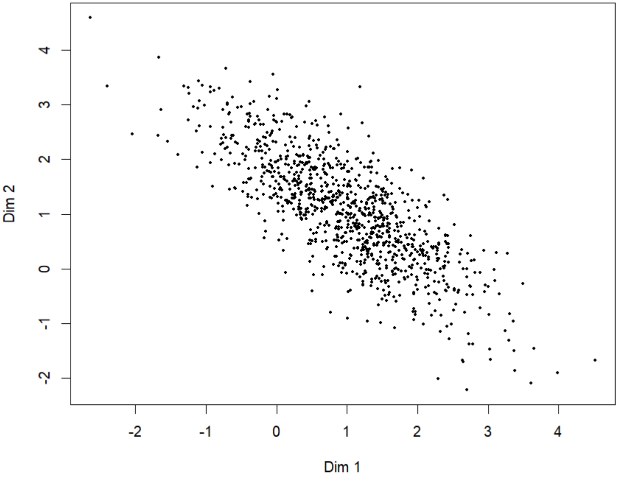
- Dimensionality reduction is an important topic also in other areas in psychology, such as diagnostics/personality psychology (for example Big Five model)

Excursus: Dimensionality reduction

- Dimensionality reduction is an important topic also in other areas in psychology, such as diagnostics/personality psychology (for example Big Five model)
- Assume you have a number of questions, do you get independent information from each question, or are there redundancies and fewer “underlying dimensions”?

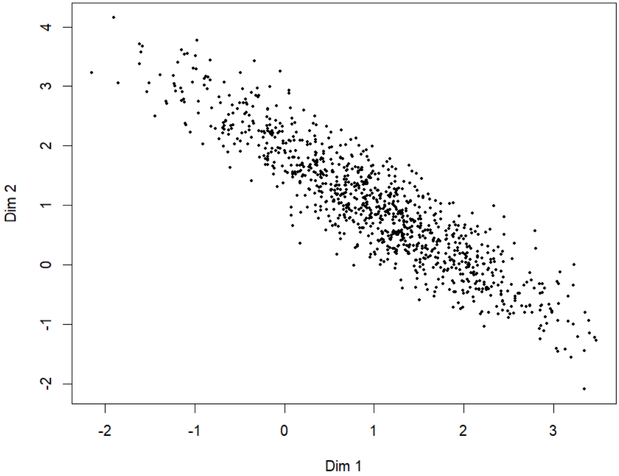
Excursus: Dimensionality reduction

- Do you need two dimensions to describe this pattern?



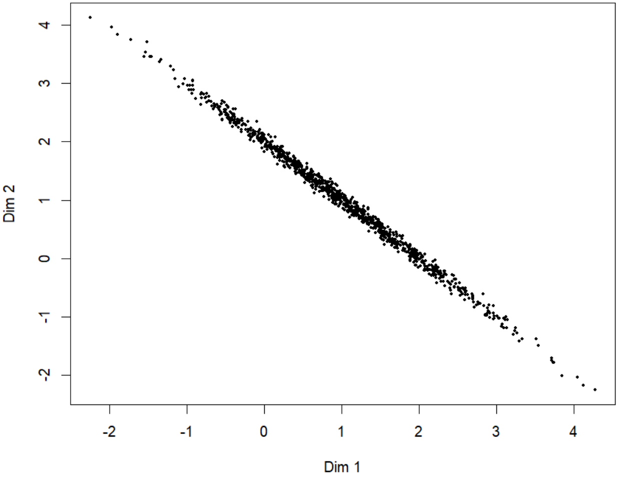
Excursus: Dimensionality reduction

- Do you need two dimensions to describe this pattern?



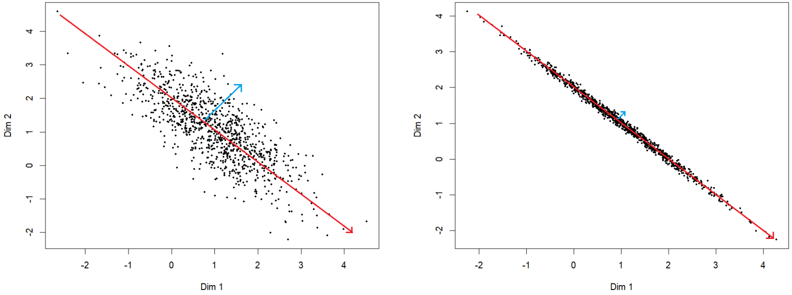
Excursus: Dimensionality reduction

- Do you need two dimensions to describe this pattern?



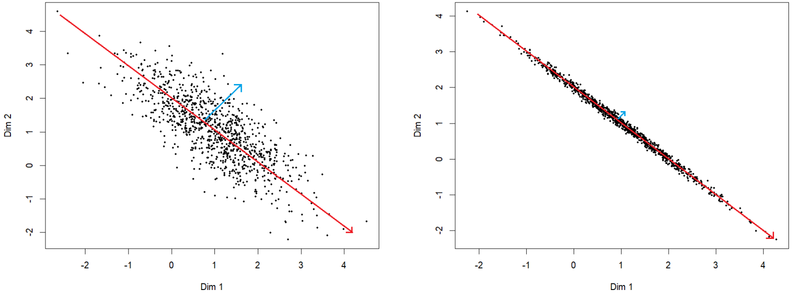
Excursus: Dimensionality reduction

Prinicipal component analysis (PCA)



Excursus: Dimensionality reduction

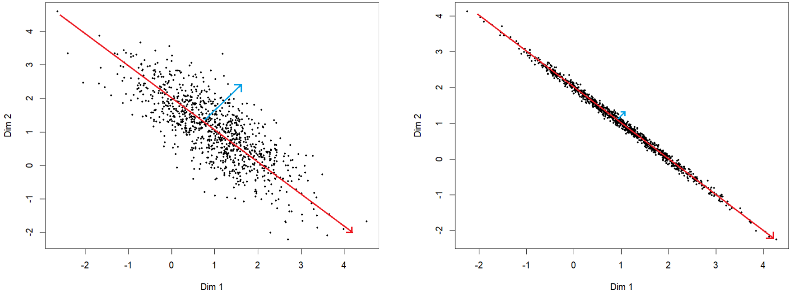
Prinicipal component analysis (PCA)



- Identify the (orthogonal) principal components – mathematical algorithm to find new “axes” for the data

Excursus: Dimensionality reduction

Prinicipal component analysis (PCA)



- Identify the (orthogonal) principal components – mathematical algorithm to find new "axes" for the data
- Keep only the principal components that you need

Excursus: Dimensionality reduction

- Selecting the number of dimensions either by
  - Keeping those that explain substantial variance (“internal criterion”)
  - Checking how many dimensions you need to explain other data such as similarity judgments (“external criterion”)

Excursus: Dimensionality reduction

- Selecting the number of dimensions either by
  - Keeping those that explain substantial variance ("internal criterion")
  - Checking how many dimensions you need to explain other data such as similarity judgments ("external criterion")
- In DSMs, we typically end up with 300 - 400 dimensions

Excursus: Dimensionality reduction

- Selecting the number of dimensions either by
  - Keeping those that explain substantial variance ("internal criterion")
  - Checking how many dimensions you need to explain other data such as similarity judgments ("external criterion")
- In DSMs, we typically end up with 300 - 400 dimensions
- Also called "latent semantic dimensions"

- Dimensionality reduction as transition from **“episodic” memory** (= experience with concrete instances) to **“semantic” memory** (= abstracted knowledge of concepts)

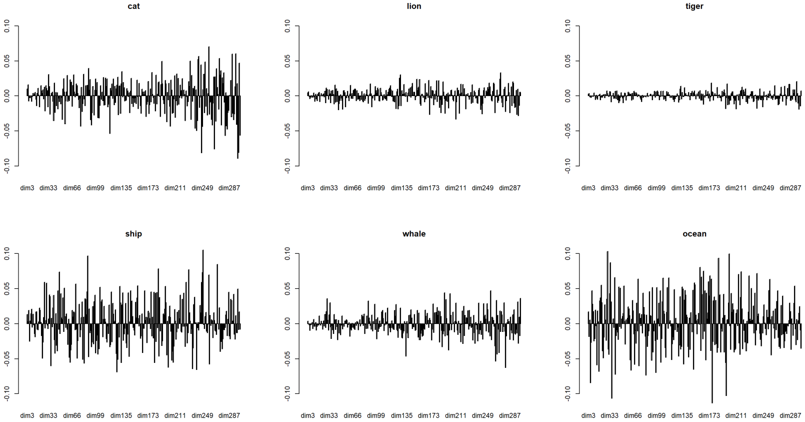
- Dimensionality reduction allows model to identify not only **first-order relations** (= words that co-occur/occur in the same context), but also **higher-order relations** (= words that appear with other words [etc. ...] that appear in the same contexts)

Landauer & Dumais, 1997

Distributional Semantic Models

Distributional semantic models

- Let's have a look at some distributional vectors for words



A note on nomenclature

- "Distributional semantic models" are known under different names, most commonly
  - Distributional semantics
  - Vector space models of meaning
  - Word embeddings
- Some people (mostly psychologists) will also just call them LSA (Latent Semantic Analysis) – we will later see why

The LSA model

Using DSMs I: The LSA homepage

Dennis, 2007

- Go to http://wordvec.colorado.edu/
- Using the LSA "General reading up to 1st year college" model
  - Find the 20 nearest neighbors of "cat" and visualize them
  - Calculate the cosine similarity between
    - mouse – dog
    - cat – rodent
    - tea – tree
  - Compute all pairwise similarities between
    - mouse – rat – keyboard – cat

Using DSMs I: The LSA homepage

Dennis, 2007

- Why the low similarity for “mouse - keyboard”?

Using DSMs I: The LSA homepage

Dennis, 2007

- Why the low similarity for “mouse - keyboard”?

  $\rightarrow$ the TASA corpus from which this specific instance of the model was created from 1990s textbooks!

- **Important:** Note that *corpus* (= training data) and *model* (= algorithm that derives representations from training data) are two different things!!

The LSA model

- LSA (Latent Semantic Analysis) is one particular DSM
- Became very popular in psychology in the late 90s (Landauer & Dumais, 1997), so that for many psychologists "distributional semantics" and "LSA" are synonymous
- Let's look at how it works

The LSA model

Step 1: The raw count data

- LSA starts from a term($=$ word)-by-document matrix

D1 The passionate nurse treats patients in the hospital

D2 The passionate doctor treats patients in the hospital

D3 The doctor saved her patient

The LSA model

Step 1: The raw count data

- LSA starts from a term(= word)-by-document matrix

D1 The passionate nurse treats patients in the hospital

D2 The passionate doctor treats patients in the hospital

D3 The doctor saved her patient

	D1	D2	D3
nurse	1	0	0
doctor	0	1	1
patient	1	1	1
hospital	1	1	0

The LSA model

Step 2: Weighting

- LSA applies a co-called “log-entropy” weighting on the raw counts

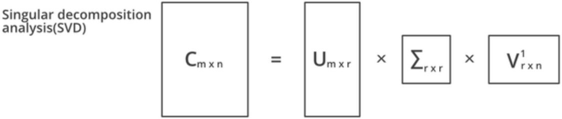
The LSA model

Step 2: Weighting

- LSA applies a co-called “log-entropy” weighting on the raw counts
- Same purpose as PMI weighting discussed earlier:
  - Reduce impact of very frequent words
  - Focus on informative relations instead of raw co-occurrence

The LSA model

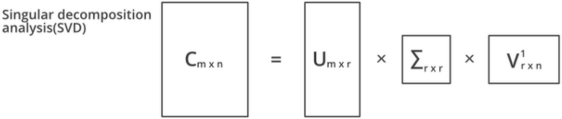
Step 3: Dimensionality reduction via Singular Value Decomposition (SVD)



- See previous discussion on dimensionality reduction (SVD is very similar to Principal Component Analysis, but for non-quadratic matrices)

The LSA model

Step 3: Dimensionality reduction via Singular Value Decomposition (SVD)



- See previous discussion on dimensionality reduction (SVD is very similar to Principal Component Analysis, but for non-quadratic matrices)
- Special property of SVD on term-by-document-matrix:
  We get *term vectors* and *document matrix* with the same number of dimensions
- All of those can be compared to one another using cosine similarity!!

The LSA model

- The possibility to compare single words and whole documents, or to compare two documents, makes LSA very interesting for some purposes
- One can show that a vector of a document is the sum of all its term vectors, so
$$\overrightarrow{w_1 w_2 ... w_n} = \overrightarrow{w_1} + \overrightarrow{w_2} + ... + \overrightarrow{w_n}$$
- We can also use this to get a representation for *any new* document (phrase, sentence, ...) consisting of several words

Using DSMs I: The LSA homepage

- Go to http://wordvec.colorado.edu/
- Using the LSA "General reading up to 1st year college" model
  - Calculate the cosine similarity between
    - A small black cat is sitting on my balcony

      The mouse ate all my cheese
    - cat

      The mouse ate all my cheese

The LSA model

Some applications

LSA has been successfully used for

- Document retrieval by query Manning, Raghavan, & Schütze (2008)
- Question answering Tellex, Katz, Lin, Fernandes, & Marton (2003)
- Sentiment analysis of documents Pang, Lee, & Vaithyanathan (2002)
- Assigning reviewers to academic papers Dumais & Nielsen (1992)
- Automatic essay grading
  Foltz, Laham, & Landauer (1999); Lenhard, Baier, Hoffmann, & Schneider (2007)

The LSA model

Empirical evaluation as a cognitive model

LSA can predict a range of empirical phenomena

- Passes synonym tests at the same level as human second-language speakers Landauer & Dumais (1997)
- Word categorization Laham (1997); Louwerse & Zwaan (2009)
- Lexical priming effects

  Jones, Kintsch, & Mewhort (2006), Günther, Dudschig, & Kaup (2016a, 2016b)

LSA and priming

An example study

- Aim: Control LSA cosine similarities as an independent variable
- Model: LSA space from German corpus, $\sim$ 880 mio. words
- Item generation procedure:
  1. Select the target words (medium frequency nouns)

Günther, Dudschig, & Kaup, 2016a, 2016b

LSA and priming

An example study

- Aim: Control LSA cosine similarities as an independent variable
- Model: LSA space from German corpus, $\sim$ 880 mio. words
- Item generation procedure:
  1. Select the target words (medium frequency nouns)
  2. Assign each target to a similarity range (.00 - .10, or .19 - .20, etc.)

Günther, Dudschig, & Kaup, 2016a, 2016b

LSA and priming

An example study

- Aim: Control LSA cosine similarities as an independent variable
- Model: LSA space from German corpus, $\sim$ 880 mio. words
- Item generation procedure:
  1. Select the target words (medium frequency nouns)
  2. Assign each target to a similarity range (.00 - .10, or .19 - .20, etc.)
  3. Sample a prime word from that similarity range (also medium frequency nouns)

Günther, Dudschig, & Kaup, 2016a, 2016b

LSA and priming

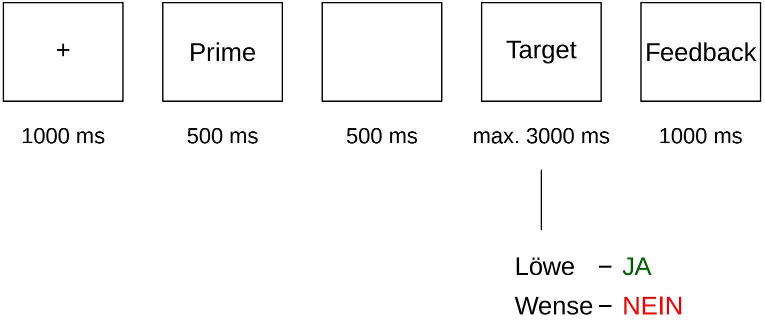
A study example

Item examples

prime		target		cosine
Butter	(butter)	Hochhaus	(skyscraper)	.08
Wirsing	(savoy cabbage)	Wanne	(tub)	.22
Hexen	(witches)	Tempel	(temple)	.47
Elster	(magpie)	Eule	(owl)	.72
Posaune	(trombone)	Flöte	(flute)	.91

Günther, Dudschig, & Kaup, 2016a, 2016b

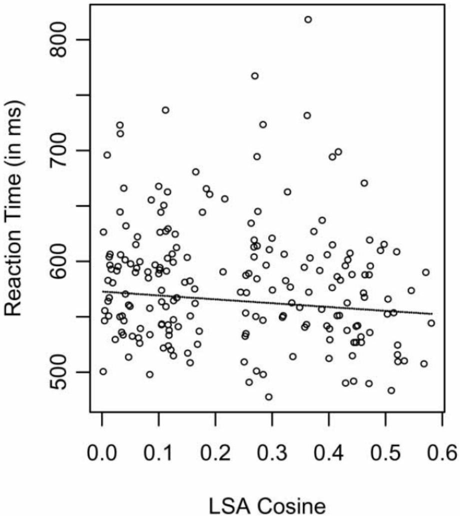
LSA and priming



Günther, Dudschig, & Kaup, 2016a, 2016b

LSA and priming

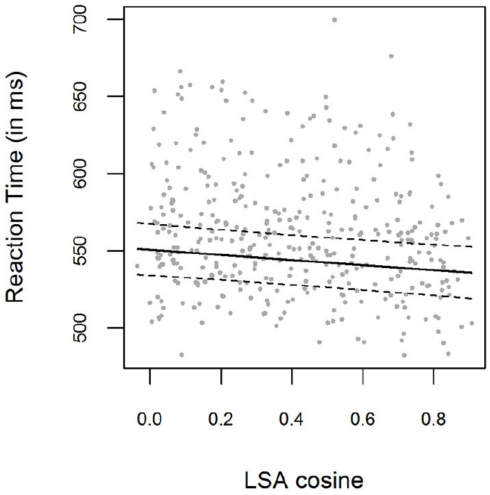
A study example



Günther, Dudschig, & Kaup, 2016a

LSA and priming

A study example



Günther, Dudschig, & Kaup, 2016b

Word-by-document versus word-by-word models

Word-by-document versus word-by-word models

	Paradigmatic relations **Selections: "x or y or..."**			
Syntagmatic relations **Combinations:** **"x and y and..."**	she	adores	green	paint
	he	likes	blue	dye
	they	love	red	colour

Sahlgren, 2008

Word-by-document versus word-by-word models

We have now encountered two types of model:

- Those that start from word-by-document data (such as LSA)

	D1	D2	D3
nurse	1	0	0
doctor	0	1	1
patient	1	1	1
hospital	1	1	0

- These tend to get similar representations for words that appear together in the same documents
- *Syntagmatic* or *Associative relations* such as ROAD − CAR

Sahlgren, 2008

Word-by-document versus word-by-word models

We have now encountered two types of model:

- Those that start from word-by-word data

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

- These tend to get similar representations for words that are surrounded by the same words = are interchangeable by one another
- *Paradigmatic* or *Semantic relations* such as ROAD – STREET

Sahlgren, 2008

Word-by-document versus word-by-word models

- First famous word-by-word model: HAL (Hyperspace Analogue to Language)

Lund & Burgess, 1996

Word-by-document versus word-by-word models

- First famous word-by-word model: HAL (Hyperspace Analogue to Language)
- Word counts as "appearing in the context of another word" if it's within the *n* (content) words before or after this word (*moving window*)

Lund & Burgess, 1996

Word-by-document versus word-by-word models

- First famous word-by-word model: HAL (Hyperspace Analogue to Language)
- Word counts as "appearing in the context of another word" if it's within the n (content) words before or after this word (*moving window*)
- Example: n = 2

  In addition to providing care and support, nurses educate the public, and promote health and wellness.

Lund & Burgess, 1996

Word-by-document versus word-by-word models

- First famous word-by-word model: HAL (Hyperspace Analogue to Language)
- Word counts as "appearing in the context of another word" if it's within the *n* (content) words before or after this word (*moving window*)
- Example: n = 2

  In addition to providing care and support, nurses educate the public, and promote health and wellness.

	addition	care	support	educate	public	health
nurse	0	1	1	1	1	0

Lund & Burgess, 1996

Word-by-document versus word-by-word models

- The literature on lexical priming reports
  - Purely associative priming (ROAD – CAR)
  - Purely semantic priming (ROAD – STREET)

Word-by-document versus word-by-word models

- The literature on lexical priming reports
  - Purely associative priming (ROAD – CAR)
  - Purely semantic priming (ROAD – STREET)
- Jones et al. (2006):
  - LSA (word-by-document) captures *associative* priming effects
  - HAL (word-by-word) captures *semantic* priming effects

Jones, Kintsch, & Mewhort, 2006

Word-by-document versus word-by-word models

- The literature on lexical priming reports
  - Purely associative priming (ROAD – CAR)
  - Purely semantic priming (ROAD – STREET)
- Jones et al. (2006):
  - LSA (word-by-document) captures *associative* priming effects
  - HAL (word-by-word) captures *semantic* priming effects
- With large corpora and general stimulus pairs, the models however become very similar:
  In Günther et al. (2016b), we find $r = .89$ between LSA and HAL cosine similarities

Jones, Kintsch, & Mewhort, 2006

Word-by-document versus word-by-word models

- With large corpora and general stimulus pairs (i.e., not specifically selected for associative/semantic relations), the models however become very similar:

Günther et al., 2016b

Word-by-document versus word-by-word models

- With large corpora and general stimulus pairs (i.e., not specifically selected for associative/semantic relations), the models however become very similar:
- In Günther et al. (2016b), we find $r = .89$ and $r = .91$ between LSA and HAL cosine similarities for our 200 pseudo-randomly sampled pairs

Günther et al., 2016b

Count versus predict models

Count versus predict models

- So far, we looked at models that start from *counting* how often a word appears in a given context

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017

Count versus predict models

- So far, we looked at models that start from *counting* how often a word appears in a given context
- However, these are unrealisitic learning models:

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017

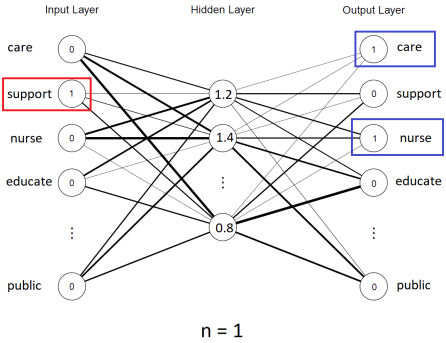
Count versus predict models

- So far, we looked at models that start from *counting* how often a word appears in a given context
- However, these are unrealisitic learning models:
- All raw "episodic" data has to be stored and transformed with each new language input

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017

Count versus predict models

- Prediction models based on neural networks as incremental learners of word representations

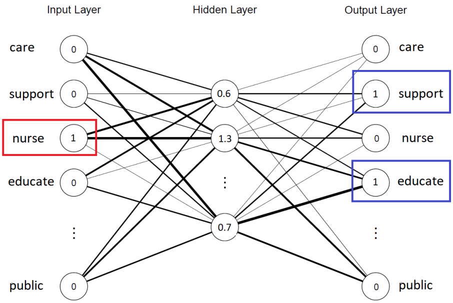


In addition to providing care and support, nurses educate the public, and promote health and wellness.

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

Count versus predict models

- Prediction models based on neural networks as incremental learners of word representations



n = 1

In addition to providing care and support, nurses educate the public, and promote health and wellness.

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

Excursus: Neural Networks

Excursus: Neural Networks

- In modern AI and machine learning, neural networks essentially do *everything*
- Examples:
  - https://www.deeparteffects.com/
  - https://www.craiyon.com/
  - https://www.deepl.com
  - ... and many more

Excursus: Neural Networks

Overview: Lanham, M. (2021). *Generating a New Reality.* Apress.

Excursus: Neural Networks

Interviewer:	Why should we hire you?
Applicant:	I am an expert in machine learning.
Interviewer:	So you're good ad maths? What is $16 + 3$?
Applicant:	4
Interviewer:	That's not even close, it's 19!
Applicant:	13
Interviewer:	Still too far, it's 19!
Applicant:	18
Interviewer:	No, 19!
Applicant:	19
Interviewer:	You're hired!

Excursus: Neural Networks

- Nice overview about implementing neural networks in R can be found here:
  https://selbydavid.com/2018/01/09/neural-network/

Excursus: Neural Networks

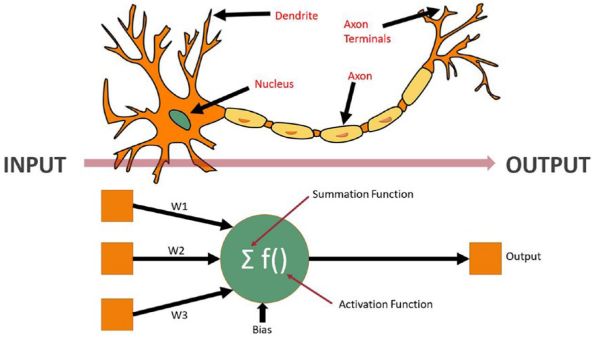
- In their essence, neural networks are regression models:
- Their aim is to predict the velues of certain output variables from certain input variables

Excursus: Neural Networks

- The basics: The perceptron

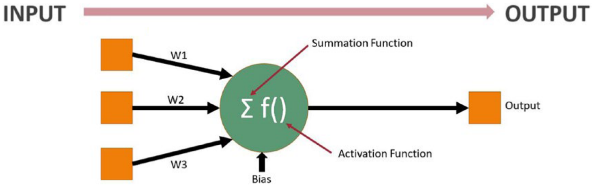
Excursus: Neural Networks

- The basics: The perceptron
- Several input values, one output value



Lanham, 2021

Excursus: Neural Networks

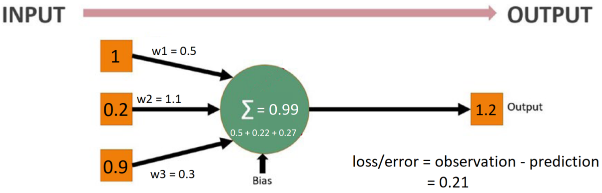


- Computing the output:
  $y = \sum_1 f(w_i \cdot x_i) + bias$

Excursus: Neural Networks

Simple case:

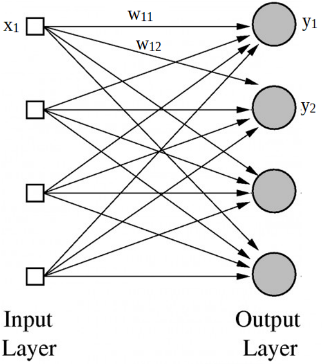
- Activation function is identity ($f(x) = x$)
- Bias is zero
- so $y = \sum_1 w_i \cdot x_i$



Lanham, 2021

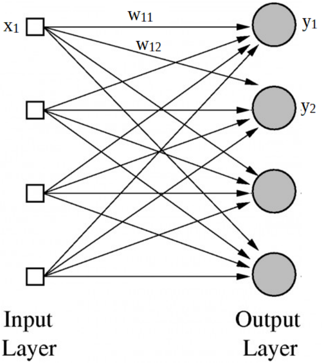
Excursus: Neural Networks

- Neural networks can also predict multiple outcomes at the same time from a set of predictors



Lanham, 2021

Excursus: Neural Networks

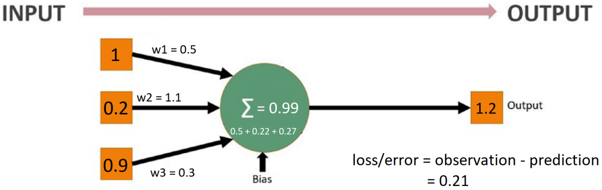


- In that case, we have

$$y_j = \sum_1 f(w_i j \cdot x_i) + bias$$

Lanham, 2021

Excursus: Neural Networks



Aim:

- Predict observed data as accurately as possible
- $\Longrightarrow$ Reduce loss/error to minimum
- Achieved by changing the weights

Lanham, 2021

Excursus: Neural Networks

Basic procedure

- Start with random weights
- Training data consisting of complete input-output pairs is presented in training cycles (in a stepwise manner, piece by piece)

Excursus: Neural Networks

Basic procedure

- Start with random weights
- Training data consisting of complete input-output pairs is presented in training cycles (in a stepwise manner, piece by piece)
- Compute error/loss

Excursus: Neural Networks

Basic procedure

- Start with random weights
- Training data consisting of complete input-output pairs is presented in training cycles (in a stepwise manner, piece by piece)
- Compute error/loss

Excursus: Neural Networks

Basic procedure

- Start with random weights
- Training data consisting of complete input-output pairs is presented in training cycles (in a stepwise manner, piece by piece)
- Compute error/loss
- In each cycle, weights are changed as a function of the loss/error: larger adjustments for larger errors

Excursus: Neural Networks

Basic procedure

- Start with random weights
- Training data consisting of complete input-output pairs is presented in training cycles (in a stepwise manner, piece by piece)
- Compute error/loss
- In each cycle, weights are changed as a function of the loss/error: larger adjustments for larger errors
- Repeat for *n* cycles (repeatedly through the entire training material) or until weights no longer change substantially between the cycles

Excursus: Neural Networks

Updating the weights: Backpropagation of errors

- Often used: The Delta Rule (similar to Rescorla-Wagner learning rule)

- Often used: The Delta Rule (similar to Rescorla-Wagner learning rule)
- (1) Compute difference between predicted and actual output:

$$(t_j - y_j)$$

- Often used: The Delta Rule (similar to Rescorla-Wagner learning rule)
- (1) Compute difference between predicted and actual output:

$$(t_j - y_j)$$

- Adjust by a *learning parameter* α (fixed parameter for the network):

$$\alpha(t_j - y_j)$$

Excursus: Neural Networks

Updating the weights: Backpropagation of errors

- Often used: The Delta Rule (similar to Rescorla-Wagner learning rule)
- (1) Compute difference between predicted and actual output:

$$(t_j - y_j)$$

- Adjust by a *learning parameter* α (fixed parameter for the network):

$$\alpha(t_j - y_j)$$

- Larger learning rate: Higher impact of error on change in weights in each cycle

- (3) Change in weight linking input x_i to y_j is this product multiplied by input activation

$$\Delta w_{ij} = \alpha(t_j - y_j) \cdot x_j$$

- (3) Change in weight linking input x_i to y_j is this product multiplied by input activation

$$\Delta w_{ij} = \alpha(t_j - y_j) \cdot x_j$$

- This is the delta rule for linear activation functions; the general case is a bit more complicated

- Training continues until the changes in weights $\Delta w_i j$ no longer exceed a threshold value t. Every training cylce uses all training items.

- Training continues until the changes in weights Δw_{ij} no longer exceed a threshold value t. Every training cylce uses all training items.

Excursus: Neural Networks

Training neural networks yourselves

- If you want to get serious about using neural networks (which are a great asset!), you should probably move to python:
  - pyTorch
  - tensorflow

Tutorials:

- The book by Lanham (2011) includes examples for every chapter:
  Lanham, M. (2021). *Generating a New Reality.* Apress.

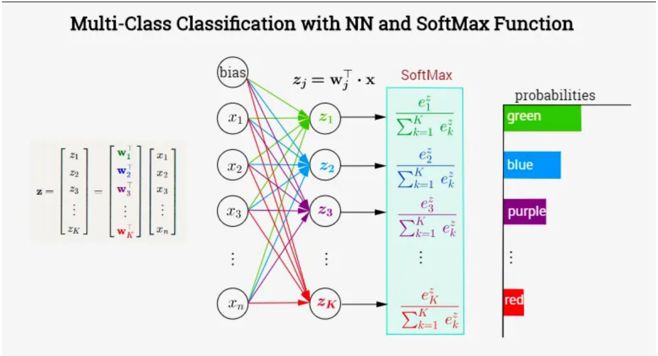
- Building a neural network to classify colors from their RGB code:
  https://medium.com/analytics-vidhya/building-rgb-color-classifier-part-1-af58e3bcfef7

- So far, we have looked at neural networks predicting numbers (continuous variable as output)
- More often than not, neural networks are used as *classifiers*: To predict categorical variables (image labels, words in a corpus, ...)

Excursus: Neural Networks

Categorical outcomes (classifiers)

- In the final layer, you have one neuron for each possible outcome
- Values in the final layer: Probability of each possible outcome
- Values are usually converted into probabilities using *softmax* (dividing by the sum of all values in the final layer)



Excursus: Neural Networks

Categorical outcomes (classifiers)

Let's use neural network classifiers!

- https://playground.tensorflow.org/
- Aim: Predict class (blue vs orange) from X1 and X2 values
- Settings:
  - Ratio of training data: 80 %
  - Noise: 0
  - Batch size: 10
  - Enable "Show test data"

Excursus: Neural Networks

Categorical outcomes (classifiers)

- Set the “hidden layers” to zero (we will talk about those in a second)
- Use the third type of data (“Gaussian”, the two separate point clouds)
- Use only X1 and X2 as features (i.e., input)
- How good does the performance get (in terms of loss?)

- Now use the second data type (“Exclusive or”)
- How good does the performance get?

Excursus: Neural Networks

Categorical outcomes (classifiers)

- Now use the second data type (“Exclusive or”)
- How good does the performance get?
- What’s the problem here?

Excursus: Neural Networks

Categorical outcomes (classifiers)

- Now use the second data type (“Exclusive or”)
- How good does the performance get?
- What’s the problem here?
- How can you improve performance?

Excursus: Neural Networks

Categorical outcomes (classifiers)

- Now use the second data type (“Exclusive or”)
- How good does the performance get?
- What’s the problem here?
- How can you improve performance?
- Does that also work for data type “Spiral”?

Excursus: Neural Networks

Hidden layers

- The neural networks we discussed so far predict the output directly from the input

Excursus: Neural Networks

Hidden layers

- The neural networks we discussed so far predict the output directly from the input
- Remember that the influence of each input neuron is just the activation in this neuron multiplied with a weight

Excursus: Neural Networks

Hidden layers

- The neural networks we discussed so far predict the output directly from the input
- Remember that the influence of each input neuron is just the activation in this neuron multiplied with a weight
- This means we can only get a linear influence of each input neuron

Excursus: Neural Networks

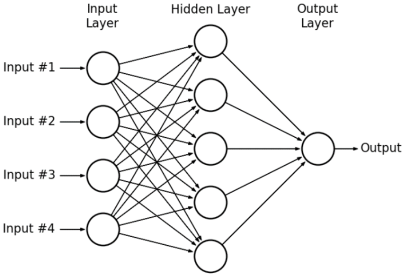
Hidden layers

- The neural networks we discussed so far predict the output directly from the input
- Remember that the influence of each input neuron is just the activation in this neuron multiplied with a weight
- This means we can only get a linear influence of each input neuron
- Which means we can't capture non-linear relationships (like in the "Spiral" data)

Excursus: Neural Networks

Hidden layers

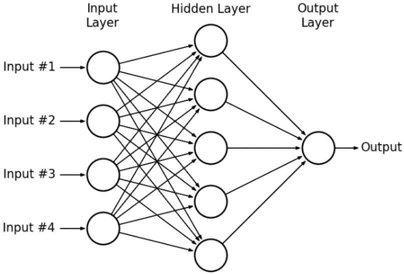
- A neural network can include *hidden layers* between input and output
- These take input from a set of neurons of the previous layer (often all of them),
  and give output to a set of neurons in the next layer (often all of them)



Lanham, 2021

Excursus: Neural Networks

Hidden layers



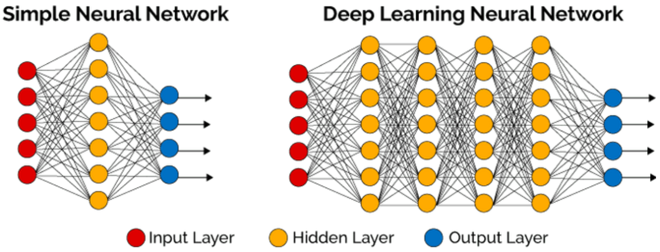
- Hidden layers allow the network to capture very complex (non-linear) relations between input and output

Lanham, 2021

Excursus: Neural Networks

Hidden layers

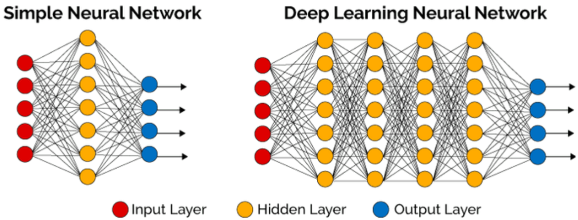
- If you have a number of hidden layers, you can call the network a “deep learning” network



Lanham, 2021; LeCun et al., 2015

Excursus: Neural Networks

Hidden layers



- There are a lot of options here:
  - Which neurons are linked
  - Different activation functions in the different layers
  - ...

Lanham, 2021; LeCun et al., 2015

Excursus: Neural Networks

Hidden layers

Now you!

- Go back to https://playground.tensorflow.org/
- Design a network that only takes X1 and X2 as Input and can accurately predict the "Exclusive or" data
- Design a network that can accurately predict the "Spiral" data
- You can add hidden layers, and change the number of neurons in each hidden layer

Lanham, 2021; LeCun et al., 2015

- With a few deep layers, RGB color classification reaches an accuracy of around .89: https://medium.com/analytics-vidhya/building-rgb-color-classifier-part-1-af58e3bcfef7

A note on parameters and parsimony

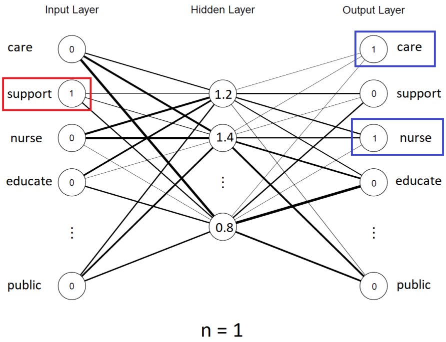
- With hidden layers, you very quickly add a lot of parameters to the model
- Occam's razor: Try to explain things with as few parameters as possible
- Does more hidden layers always mean more parameters?

Lanham, 2021; LeCun et al., 2015

Back to distributional semantic models

Predict models: word2vec

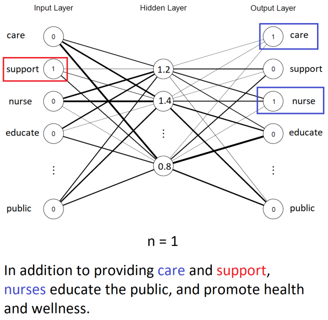
- Let's have a look at predict models again: Mikolov's *word2vec* model



In addition to providing care and support, nurses educate the public, and promote health and wellness.

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

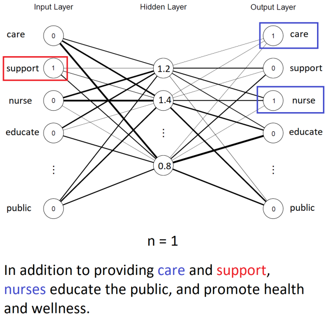
Predict models: word2vec



- Input and output layer contain as many neurons as words (with frequency $> n$) in the corpus, one neuron for each word
- One-hot encoding: Target and context words are 1, everything else 0

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

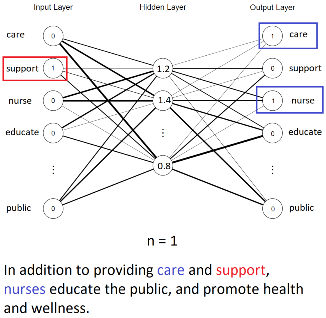
Predict models: word2vec



- One hidden layer, 300 neurons
- Go through the corpus word by word to train the network

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

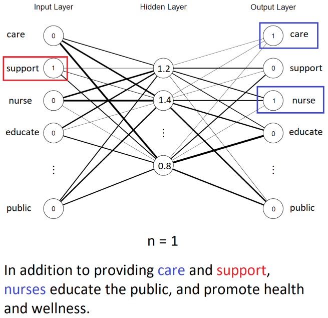
Predict models: word2vec



- Once the model is trained, the activation of the hidden layer for a given word input is the 300-dimensional distributional vector for this word

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

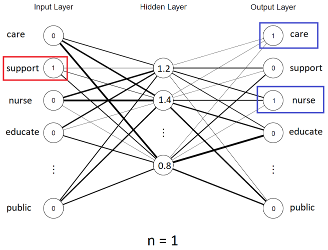
Predict models: word2vec



- Once the model is trained, the activation of the hidden layer for a given word input is the 300-dimensional distributional vector for this word

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

Predict models: word2vec



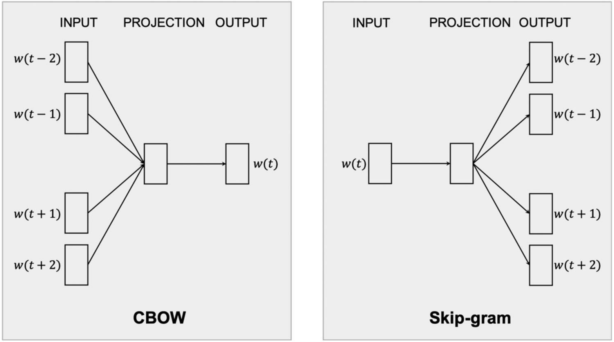
In addition to providing care and support, nurses educate the public, and promote health and wellness.

- Once the model is trained, the activation of the hidden layer for a given word input is the 300-dimensional distributional vector for this word

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

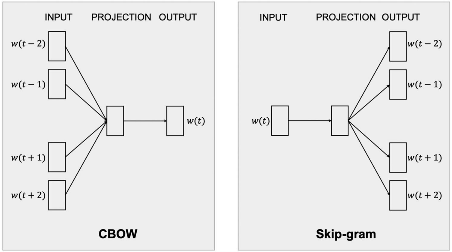
Predict models: word2vec

- word2vec comes in two variants:
  - *cbow*: predicting target from context
  - *skip-gram*: predicting context from target



Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

Predict models: word2vec

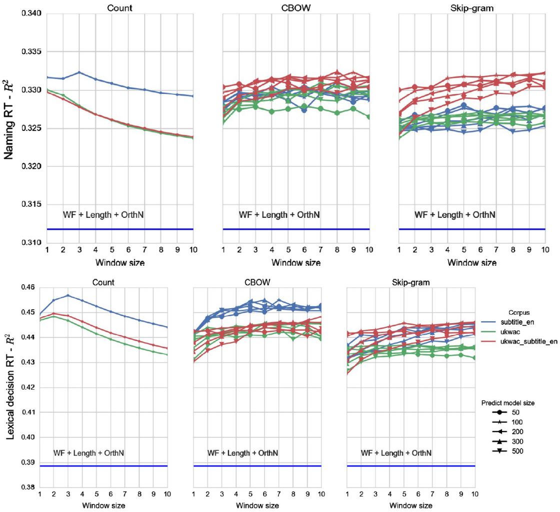


- For predicting behavioral data, cbow appears to be better Baroni et al., 2014; Mandera et al., 2017
- Also more in line with psychological learning theories Hollis, 2017; Mandera et al., 2017

Hollis, 2017; Mandera, Keuleers, & Brysbaert, 2017; Mikolov et al., 2013

Predict models: word2vec

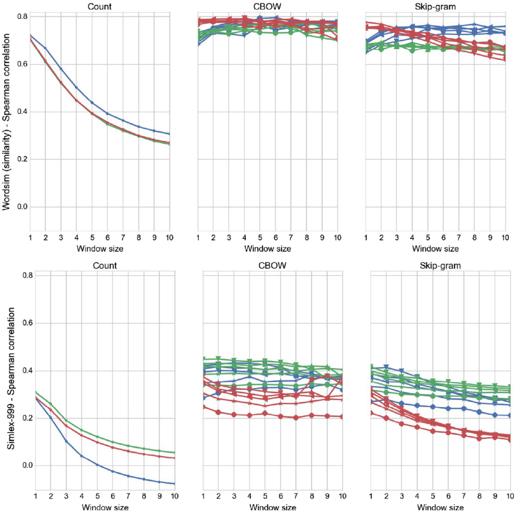
Priming effects



Mandera, Keuleers, & Brysbaert, 2017

Predict models: word2vec

Word similarity ratings



Mandera, Keuleers, & Brysbaert, 2017

Predict models: word2vec

Comparing count vs predict

name	task	measure	source	soa
rg	relatedness	Pearson	Rubenstein and Goodenough (1965)	Hassan and Mihalcea (2011)
ws	relatedness	Spearman	Finkelstein et al. (2002)	Halawi et al. (2012)
wss	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
wsr	relatedness	Spearman	Agirre et al. (2009)	Agirre et al. (2009)
men	relatedness	Spearman	Bruni et al. (2013)	Bruni et al. (2013)
toefl	synonyms	accuracy	Landauer and Dumais (1997)	Bullinaria and Levy (2012)
ap	categorization	purity	Almuhareb (2006)	Rothenhäusler and Schütze (2009)
esslli	categorization	purity	Baroni et al. (2008)	Katrenko and Adriaans (2008)
battig	categorization	purity	Baroni et al. (2010)	Baroni and Lenci (2010)
up	sel pref	Spearman	Padó (2007)	Herdağdelen and Baroni (2009)
mcrae	sel pref	Spearman	McRae et al. (1998)	Baroni and Lenci (2010)
an	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)
ansyn	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013a)
ansem	analogy	accuracy	Mikolov et al. (2013a)	Mikolov et al. (2013c)

Baroni, Dinu, & Kruszewski, 2014

Predict models: word2vec

Comparing count vs predict

count

window	weight	compress	dim.	mean rank
2	PMI	no	300K	35
5	PMI	no	300K	38
2	PMI	SVD	500	42
2	PMI	SVD	400	46
5	PMI	SVD	500	47
2	PMI	SVD	300	50
5	PMI	SVD	400	51
2	PMI	NMF	300	52
2	PMI	NMF	400	53
5	PMI	SVD	300	53

predict

win.	hier. softm.	neg. samp.	subsamp.	dim	mean rank
5	no	10	yes	400	10
2	no	10	yes	300	13
5	no	5	yes	400	13
5	no	5	yes	300	13
5	no	10	yes	300	13
2	no	10	yes	400	13
2	no	5	yes	400	15
5	no	10	yes	200	15
2	no	10	yes	500	15
2	no	5	yes	300	16

Baroni, Dinu, & Kruszewski, 2014

Predict models: word2vec

Comparing count vs predict

A word of caution:

- word2vec seems to fail for small corpora Altszyler et al., 2017
- In absolute terms, differences in performance are not dramatic
- Lenci et al. (2022), large-scale evaluation of distributional semantic models with many different tasks: “the alleged superiority of predict based models is more apparent than real, and surely not ubiquitous”

Altszyler et al., 2017; Lenci et al., 2022

Using DSMs I: The LSA homepage

Dennis, 2007

- Go to http://wordvec.colorado.edu/
- Using the **word2vec** model
  - Find the 20 nearest neighbors of "cat" and visualize them
  - Calculate the cosine similarity between
    - mouse – dog
    - cat – rodent
    - tea – tree
  - Compute all pairwise similarities between
    - mouse – rat – keyboard – cat

Using DSMs I: The LSA homepage

Dennis, 2007

- Are the results different than when using the LSA space?

Comparing DSMs

Whenever comparing two models, keep in mind that different components can differ:

- The training corpus (was kept identical in these studies)
  - incl. corpus preprocessing
- The general algorithm
  - incl. its parameter values

Using DSMs I: The LSA homepage

Dennis, 2007

- How easily can you use the results of this homepage in your data analysis?

Using DSMs II: The SNAUT website

Mandera et al., 2017

- Go to http://meshugga.ugent.be/snaut//
- Using the English cbow space
  - Find the 20 nearest neighbors of "cat"
  - Calculate the cosine similarity between
    - mouse – dog
    - cat – rodent
    - tea – tree
  - Compute all pairwise similarities between
    - mouse – rat – keyboard – cat

Using DSMs II: The SNAUT website

- How easily can you use these results in your data analysis?

Using DSMs III: The R package LSAfun

Günther, Dudschig, & Kaup, 2015

- Install the package in R, using `install.packages("LSAfun")`
- Load it using `library("LSAfun")`
- For an overview over the package, use `help(package="LSAfun")`
- There is a video tutorial at www.fritzguenther.de/software-resources/video_tutorials
- Download a semantic space from `www.fritzguenther.de/software-resources/semantic_spaces`
  Save it somewhere you can find it again!

Using DSMs III: The R package LSAfun

Günther, Dudschig, & Kaup, 2015

- Load the semantic space into the R workspace using either

  ```
  setwd("PATH")
  load("NAMEOFSPACE")
  ```

  or

  ```
  load("PATH/NAMEOFSPACE")
  ```

- `"PATH"` is where you saved the file
- `"NAMEOFSPACE"` is the name of the file

Using DSMs III: The R package LSAfun

Günther, Dudschig, & Kaup, 2015

- Using the LSAfun package and the semantic space you downloaded
  - Find the 20 nearest neighbors of "cat", using `neighbors()`
  - Visualize this neighborhood using `plot_neighbors()`
  - Using `pairwise()` , calculate the cosine similarity between
    - mouse – dog
    - cat – rodent
    - tea – tree
  - Using `multicos()`, compute all pairwise similarities between
    - mouse – rat – keyboard – cat
  - Use `?function` for info on how to use a function

Using DSMs III: The R package LSAfun

Günther, Dudschig, & Kaup, 2015

- How easily can you use these results in your data analysis?

Using DSMs

Analyzing real data

- MEN dataset: Similarity scores for word pairs (Bruni et al., 2014)
- Download the dataset from URL
  Save it somewhere you can find it!
- Load the dataset into the R workspace:
  - Option 1: Read the file using RStudio's "Import dataset", naming it `men`
  - Option 2: Read the file using
    ```
    men <- read.csv("PATH TO DATA/men.csv")
    ```
  - Option 3: Read the file using
    ```
    setwd("PATH TO DATA")
    men <- read.csv("men.csv")
    ```

Inspect the data file!

- Option 1: Click on it in the "Environment" panel
- Option 2: `View(men)`
- Option 3: `head(men)`

Excursus: Some additional R

Inspecting data

- Get a summary of the data structure with
  `summary(men)`
  `str(men)`

- To get a specific row of the data frame, you can use commands like

```
men[1,]
men[c(1,3,5),]
men[1:10,]
```

- Same for a specific column of the data frame
  `men[,1]`
  `men[,c(1,3)]`

- To get a column by name, use one of the following:
  `men[,"rt"]`
  `men$rt`

- To add a column to the dataframe, you can also use the $ operator: `men$number <- 1:nrow(men)`

- **Compute the rank correlation between the MEN similarity ratings and DSM cosine similarities, using any method, source, and model of your choice**
- To compute a rank correlation in R, use `cor(x,y,method="spearman",use="pairwise.complete.obs")`
- (For this exercise, you can ignore missing values)

The content of DSMs

What are we measuring?

- The *type* of relation: Semantic vs. associative Jones, Kintsch, & Mewhort, 2007
- However, most actual models measure both relations to some degree

The content of DSMs

What are we measuring?

- DSM cosine similarities are **not** simply co-occurrence probabilities
- Words can have very similar vectors even if they never occur together

- The passionate nurse treats patients in the hospital
- The passionate doctor treats patients in the hospital
- The doctor saved her patient
- A whale travels the ocean

	patient	hospital	ocean
nurse	1	1	0
doctor	2	1	0
whale	0	0	1

Günther, Rinaldi, & Marelli, 2019

The content of DSMs

What are we measuring?

- DSMs **are** estimated only from text corpora
- They thus only have access to information that is communicated via language (linguistic experience)
- The training corpus will have a huge result on the resulting representations
  - A DSM trained on biology textbooks will probably not have a good representation of Middle Eastern geopolitics
  - To model human semantic memory, we want corpora that are representative for an average speaker's language experience
  - Algorithms tend to work better on larger corpora (less noise)

Günther, Rinaldi, & Marelli, 2019

The content of DSMs

What are we measuring?

- DSMs **are** estimated only from text corpora
- However, via the training corpora they still have access to world knowledge (contingent facts about the world we live in, which is typically not considered part of the "semantics" of a word)
- Example: Who is US president at a time, who is his wife, ...
- After all, speakers produce language to talk about the world they live in

The content of DSMs

What are we measuring?

- Since text corpora are generated by human speakers, DSMs also learn the biases of these human speakers
- For example, standard DSMs show gender and racial biases
  Caliskan et al., 2017; Bhatia, 2017

The content of DSMs

What are we measuring?

- DSMs only learn from text and have no access to perceptual experience
- Only models of lexical semantics, not human concepts?
  Glenberg & Robertson, 2000

Günther, Rinaldi, & Marelli, 2019

The content of DSMs

What are we measuring?

- DSMs only learn from text and have no access to perceptual experience
- Only models of lexical semantics, not human concepts?
  Glenberg & Robertson, 2000
- People use language to talk about the world: Language data encode perceptual/sensorimotor aspects of our world Louwerse, 2011
- To some degree, DSMs capture these aspects
- Analogy: Congenitally blind person's representation of visual information

Günther, Rinaldi, & Marelli, 2019

The content of DSMs

What are we measuring?

Now you!

- Using one of the methods we have discussed so far (LSA homepage, SNAUT, LSAfun), explore the representations of a DSM of your choice (using similarities, neighborhoods, ...)
  - Use examples you find interesting
  - Do you find something intuitive/counter-intuitive?

Vision-based representations

Vision-based representations

- We have more experience than just language

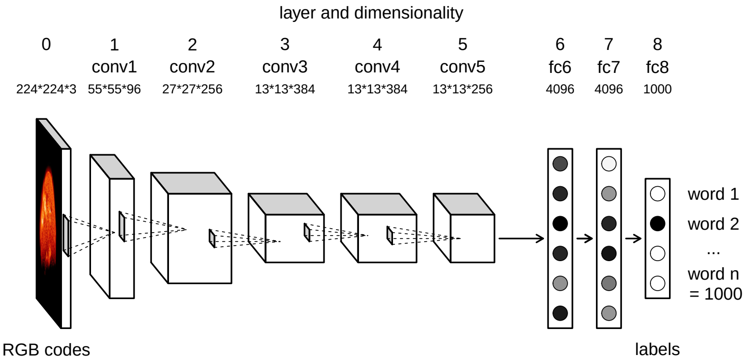
Vision-based representations

- We have more experience than just language
- Sensorimotor experience, especially vision, is very important as well

Vision-based representations

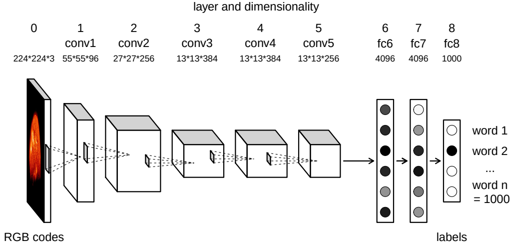
- We have more experience than just language
- Sensorimotor experience, especially vision, is very important as well
- How do we build vision-based representations?

A deep convolutional neural network (DCNN)



Chatfield et al., 2014

A deep convolutional neural network (DCNN)

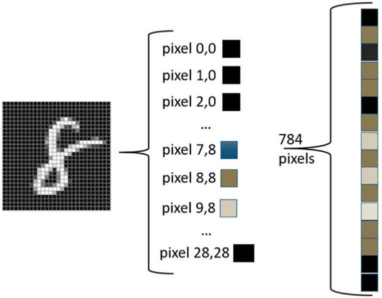


- Output: Classification into 1,000 different image classes (by label)
  We have discussed classifiers before
- Let's have a look at the input

Chatfield et al., 2014

Excursus: Convolutional networks

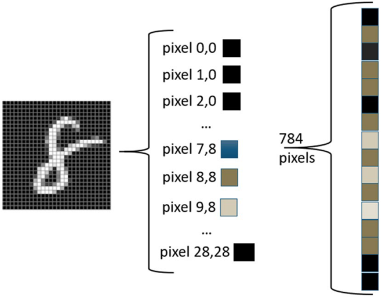
- "Up until 2012 image analysis with neural networks was done by flattening an image to a single one-dimensional (1D) vector" p. 43



- Values in this vector could be brightness of each pixel (i.e., position on a gray scale for black-and-white images)

Lanham, 2021

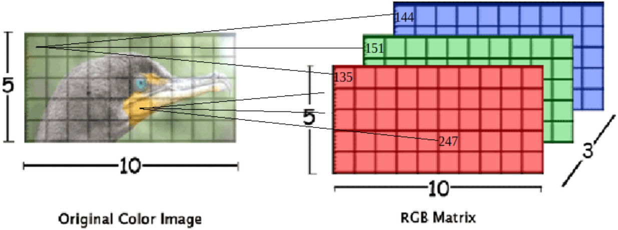
Excursus: Convolutional networks



- “often missed obvious image features.” p. 43

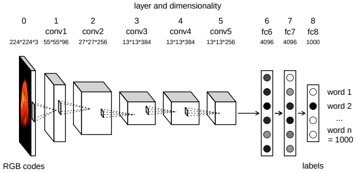
Excursus: Convolutional networks

- New approach: No flattening; taking spatial information seriously Krizhevsky et al., 2012
- Encode image as a matrix instead of a vector
  For colored images, use a tensor ("3D-Matrix") encoding the RGB code of each pixel



Lanham, 2021

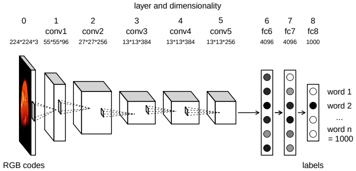
Excursus: Convolutional networks



- Layers 6– 8 are fully connected:
- Each neuron receives input from each neuron in the previous layer
- These are the “standard” hidden layers we discussed before

Chatfield et al., 2014

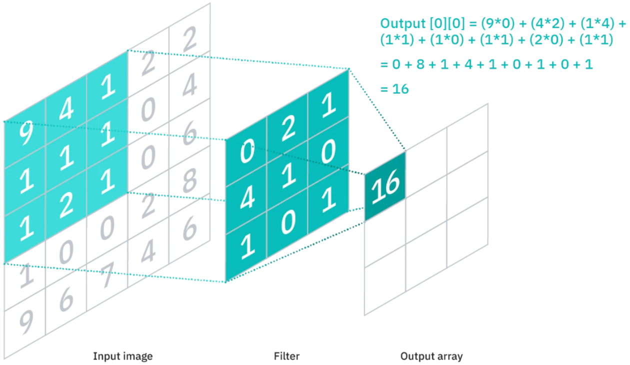
Excursus: Convolutional networks



- Layers 1–5 are convolutional layers
- These only receive input from *some* neurons in the previous layer; more specifically, only from a certain area
- Let's see what that means

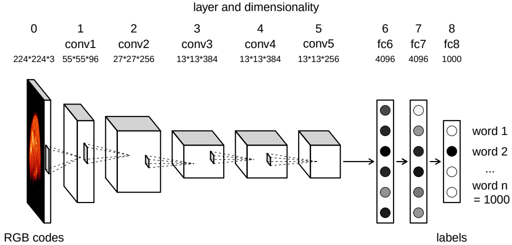
Chatfield et al., 2014

Excursus: Convolutional networks



Lanham, 2021

Excursus: Convolutional networks

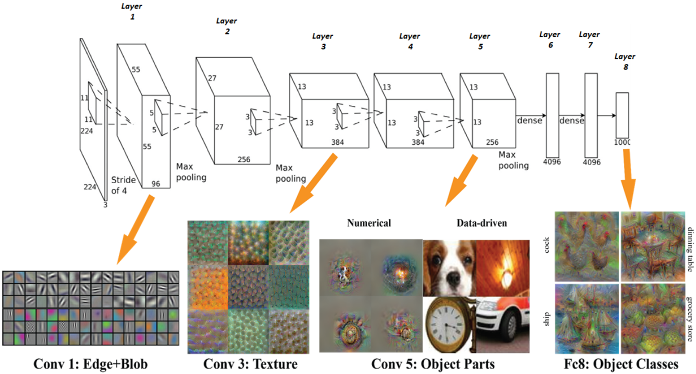


- Is this structure similar to the (neural) human visual system?
- Yes and no

  Cichy & Kaiser, 2019; Jacobs & Bates, 2019; Kriegeskorte, 2015; Lindsay, 2021; Majaj & Pelli, 2018; Serre, 2019

Chatfield et al., 2014

The VGG-F model



https://donglaiw.github.io/page/mneuron

Chatfield et al., 2014

The VGG-F model

- These networks are called *deep convolutional neural networks* (DCNNs)
- Trained on large image databases (ImageNet); very good classification performance
- DCNNs for images come in many, many shapes and sizes (the one shown here is just one example, VGG-F)

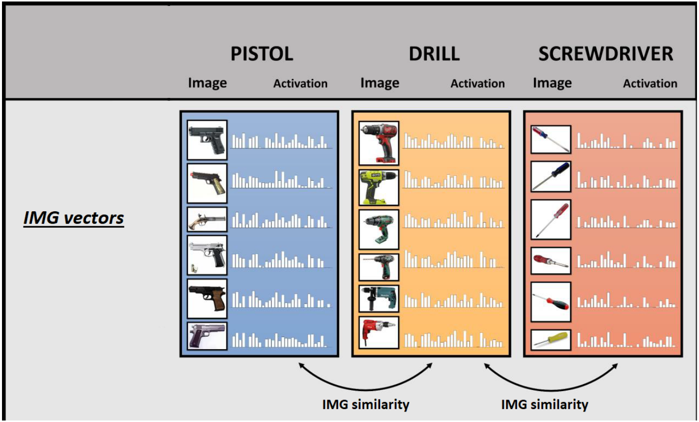
The VGG-F model

Image representations

- Once the network model is trained, we have a fixed set of weights
- For any image we use as input (also images outside the training set!), this will produce a representation in each layer of the network
- For predicting human behavioral data, the deeper layers (6-7) of the VGG-F network show the best performance (4,096-dimensional vector representations) Günther et al., 2012

The VGG-F model

Image representations



The VGG-F model

Now you!

- Go to http://vispa.fritzguenther.de
- The site is modelled after SNAUT
- With the aid of the picture picker on the right, calculate the similarity between three image pairs of your choice
- Find the 10 most similar images to an image of your choice

The VGG-F model

Now you!

- Repeat the same calculations in R using the LSAfun package
- Download the IMG space from www.fritzguenther.de/software-resources/vispa
- Again, there is a video tutorial at
- There is a video tutorial at www.fritzguenther.de/software-resources/video_tutorials
- This works exactly like with DSMs; you only use image names instead of words

The VGG-F model

Image representations

- Is this useful for psychology?
- Can we predict human behavioral data?

Visual similarity ratings for images

3,000 image pairs; 480 participants

Which look the most similar? Which look the least similar? Hollis, 2018

Günther, Marelli, Tureski, & Petilli, 2021

Visual similarity ratings for images

3,000 image pairs; 480 participants

rating value = .821

[BERRY]

[RASPBERRY]

rating value = .818

[SURGEON]

[SURGERY]

Günther, Marelli, Tureski, & Petilli, 2021

Visual similarity ratings for images

3,000 image pairs; 480 participants

rating value = .204

[CHEETAH]

[PHONE]

rating value = .205

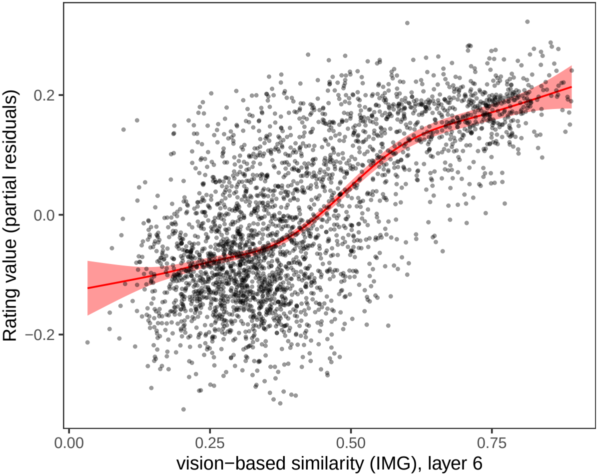
[ROD]

[NOVICE]

Günther, Marelli, Tureski, & Petilli, 2021

Visual similarity ratings for images

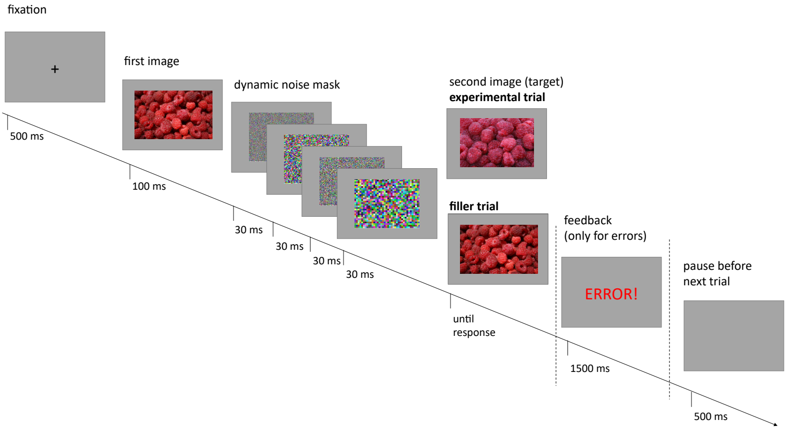
3,000 image pairs; 480 participants



Günther, Marelli, Tureski, & Petilli, 2021

Discrimination task for images

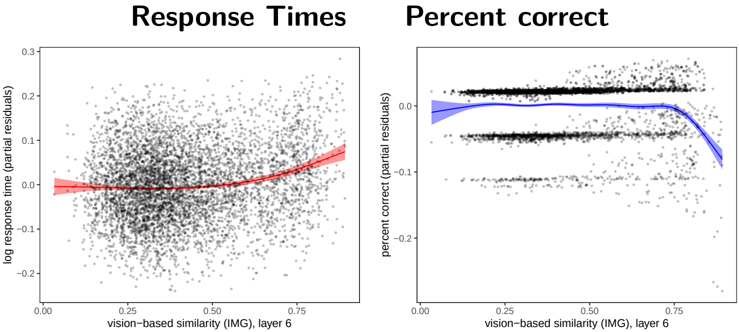
3,000 image pairs (from Study 2); 750 participants



Günther, Marelli, Tureski, & Petilli, 2021

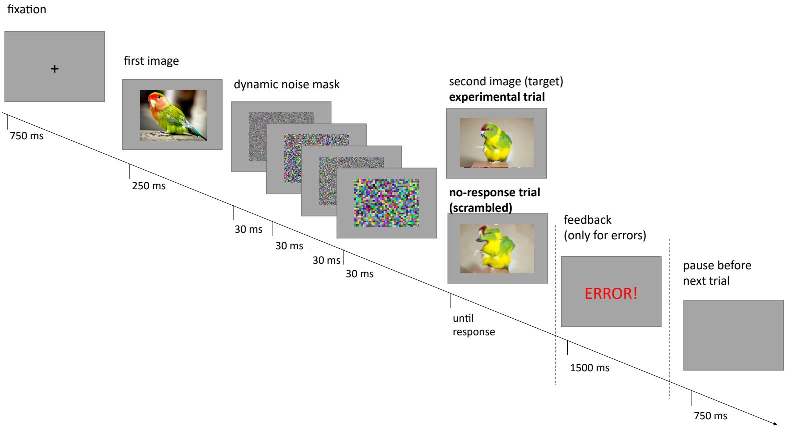
Discrimination task for images

3,000 image pairs (from Study 2); 750 participants



Priming (visual decision task) for images

3,000 image pairs (from Study 2 and 4); 750 participants



Günther, Marelli, Tureski, & Petilli, 2021

Priming (visual decision task) for images

3,000 image pairs (from Study 2 and 4); 750 participants

Diffeomorphic scrambling of images (Stojanoski & Cusack, 2014)

original

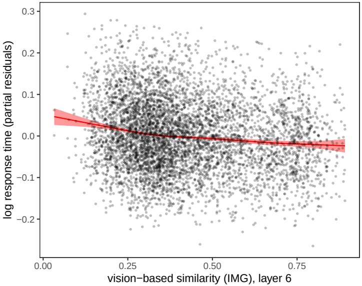
scrambled

Günther, Marelli, Tureski, & Petilli, 2021

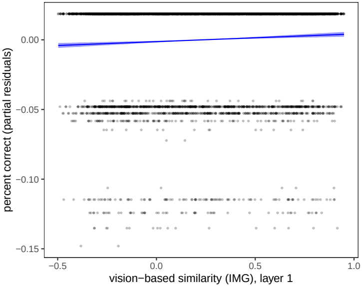
Priming (visual decision task) for images

3,000 image pairs (from Study 2 and 4); 750 participants

Response Times



Percent correct



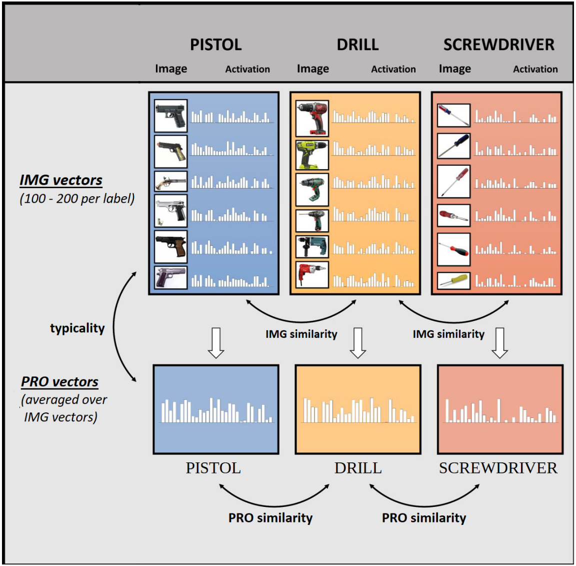
Vision-based concept representations

- We have reasonable representations for individual images

- Can we use this to build vision-based concept representations?

ALPACA

Vision-based concept representations



There are other options; see Battleday et al. (2020)

The VGG-F model

Now you!

- With a method of your choice, calculate the **visual** similarity (DCNN) and the **semantic** similarity (DSM) between any three word pairs you like

Visual similarity ratings for words/concepts

3,000 word pairs; 480 participants

Which look the **most** similar?*

- ○ spear - arm
- ○ apple - peach
- ○ salt - stallion
- ○ pebble - diamond

Which look the **least** similar?*

- ○ spear - arm
- ○ apple - peach
- ○ salt - stallion
- ○ pebble - diamond

Continue

Günther, Marelli, Tureski, & Petilli, 2021

Visual similarity ratings for words/concepts

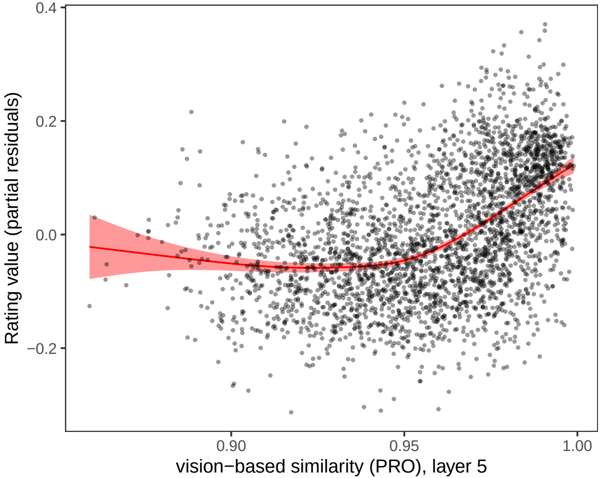
3,000 word pairs; 480 participants

highest scores		*lowest scores*	
item	**value**	**item**	**value**
feline – kitty	.821	inn – jellyfish	.204
coke – pepsi	.819	salt – teacher	.206
cream – milk	.817	uphill – gravy	.212
tangerine – orange	.816	giraffe – jelly	.212
chimpanzee – ape	.814	flamingo – office	.212

Günther, Marelli, Tureski, & Petilli, 2021

Visual similarity ratings for words/concepts

3,000 word pairs; 480 participants



Günther, Marelli, Tureski, & Petilli, 2021

Visual effects on semantic similarity ratings

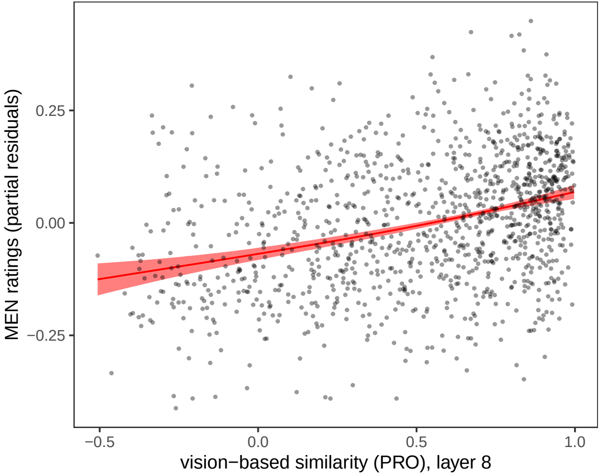
Now you!

- Compute the visual similarity for the word pairs in the MEN dataset
- Compute the Spearman correlation between these similarities and the MEN similarity ratings
- What would you expect to happen?

Visual effects on semantic similarity ratings

1,167 items from the MEN dataset (Bruni et al., 2015)

$r_S = .79$ with visual similarity ratings



Visual effects in lexical priming

1,128 word pairs; Semantic Priming Project, Hutchison et al., 2013

- Vision-based prototype similarity predicts priming effects in lexical decision (i.e., for *word pairs*),

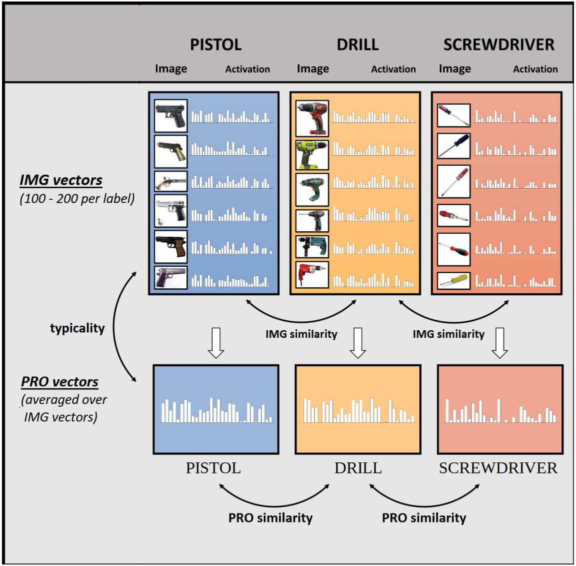
Visual effects in lexical priming

1,128 word pairs; Semantic Priming Project, Hutchison et al., 2013

- Vision-based prototype similarity predicts priming effects in lexical decision (i.e., for *word pairs*),
- even after controlling for DSM similarities

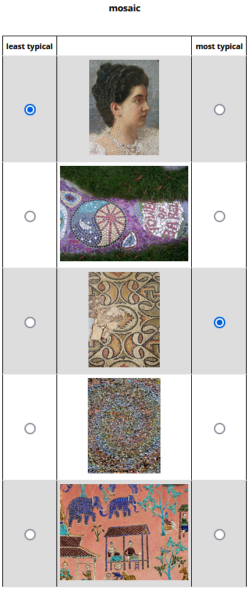
Petilli, Günther, Vergallito, Ciapparelli, & Marelli, 2021

Typicality


PISTOL
DRILL
SCREWDRIVER
Image
Activation
Image
Activation
Image
Activation
IMG vectors
(100 - 200 per label)
typicality
IMG similarity
IMG similarity
PRO vectors
(averaged over
IMG vectors)
PISTOL
DRILL
SCREWDRIVER
PRO similarity
PRO similarity

Typicality ratings for word-image pairs

1,500 words à 5 images; 900 participants



Typicality ratings for word-image pairs

1,500 words à 5 images; 900 participants

lemon

.308

.356

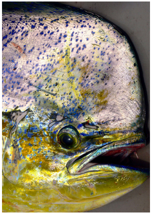
.561

.599

.903

dolphin

.096



.382

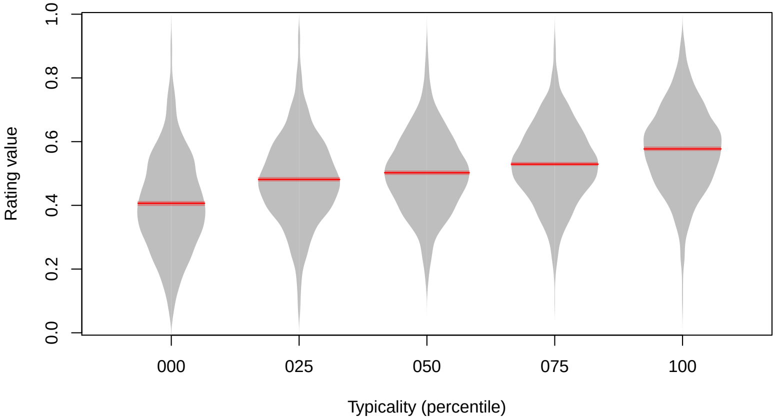
.568

.607

.619

Typicality ratings for word-image pairs

1,500 words à 5 images; 900 participants



Günther, Marelli, Tureski, & Petilli, 2021

Typicality

Now you!

- With a method of your choice, find two images that are very typical of their category, and two that are very atypical
- Why might these particular images be atypical?

Linking language and vision

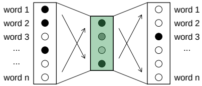
Linking language and vision

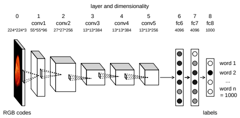
- Before, we have seen the argument that language is used to talk about the world and thus encodes perceptual information
- Can we predict how things look like when we just have language information?

Günther, Petilli, Vergallito, & Marelli, 2020

Linking language and vision

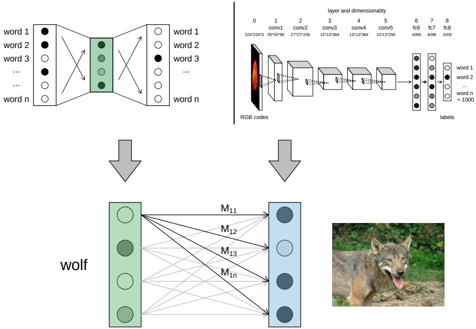
Training set: 7,801 words





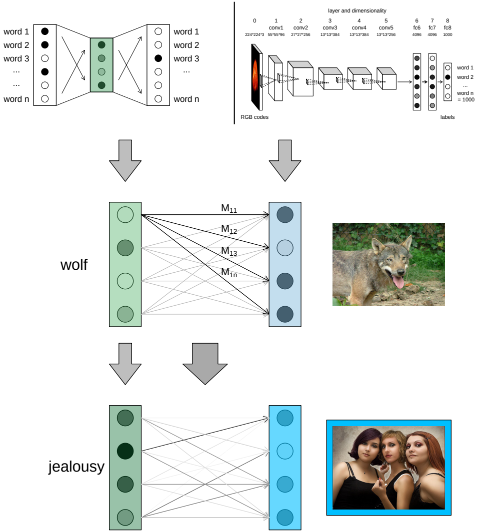
Linking language and vision

Training set: 7,801 words



Linking language and vision

Training set: 7,801 words



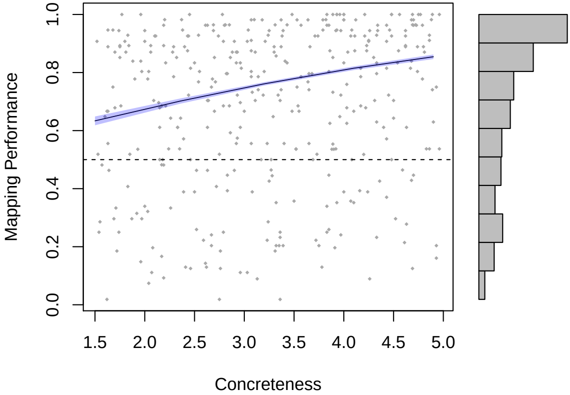
Linking language and vision

word type	**word**	**model prediction**	**random image**
concrete	stallion		
concrete	scout		
concrete	aspirin		
abstract	childhood		
abstract	jealousy		

Günther, Petilli, Vergallito, & Marelli, 2020

Linking language and vision

- 371 items outside the training set
- How often did participants choose the model prediction (2AFC)?



Günther, Petilli, Vergallito, & Marelli, 2020

Linking language and vision

- Language encodes (at least some) perceptual information

Linking language and vision

- Language encodes (at least some) perceptual information
- This information can be de-coded even with simple linear regression

Günther, Petilli, Vergallito, & Marelli, 2020

Linking language and vision

Now you!

- There are far more powerful text-to-vision models in AI research
- Go to https://www.craiyon.com/ and try out some things
- (To fully explain what's going on here, we'd need a few additional concepts like generative networks and modern language models, which we will not cover here)

Summary

Summary

- Recent research in computer science/AI (fields like natural language processing and computer vision) has provided us with powerful representation models
- We can use these to approximate human mental representations
- Empirical evidence so far is promising

Room for Discussion