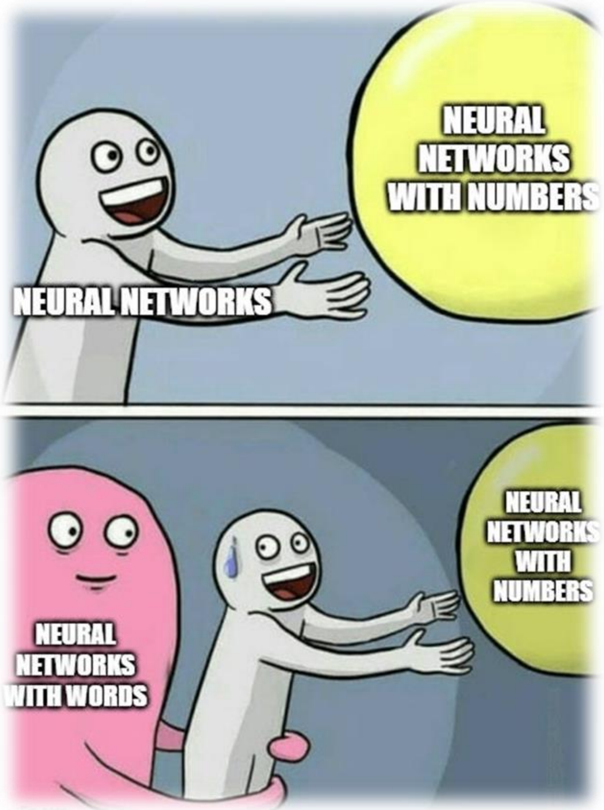
Embeddings

Alberto TONDA, Ph.D. (Senior permanent researcher, DR)

UMR 518 MIA-PS, INRAE, AgroParisTech, Université Paris-Saclay
UAR 3611, Institut des Systèmes Complexes de Paris Île-de-France

Outline

- Embeddings
- Building embeddings
- Embeddings from trained deep networks
- Embeddings with Autoencoders
- Embeddings of Vocabularies

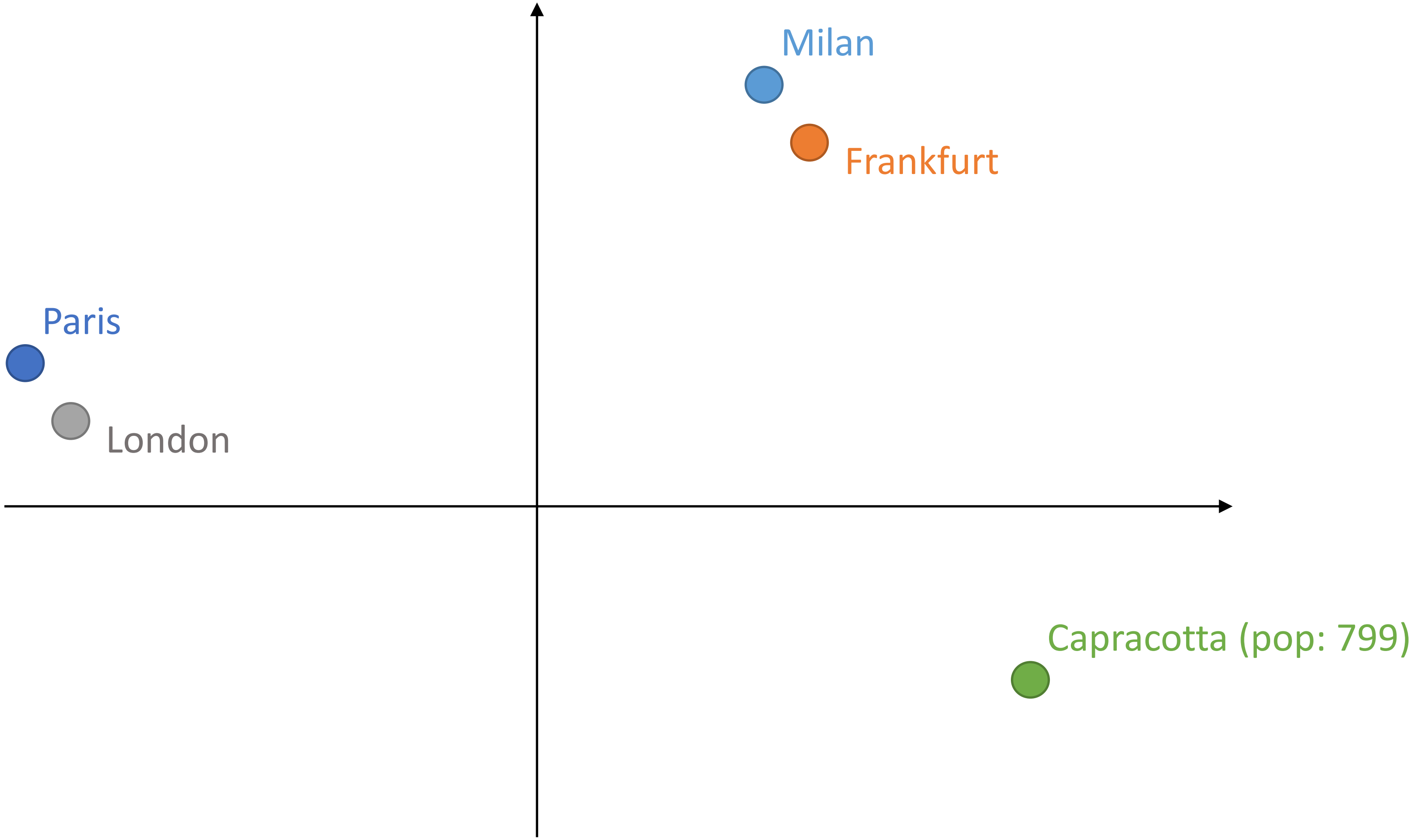


Embeddings

- Curiously, an embedding is not necessarily related to DL
- An embedding is a **vectorial space**, $\boldsymbol{x} \in \mathbb{R}^d$ with semantics
  - Distances and relative positions between points have a *meaning*
  - And displacements/transformations might also have meaning
  - Vocabulary clash, “vector”/“point”

- Let’s build an embedding! Meaningful dimensions for **cities**?

Embedding for cities

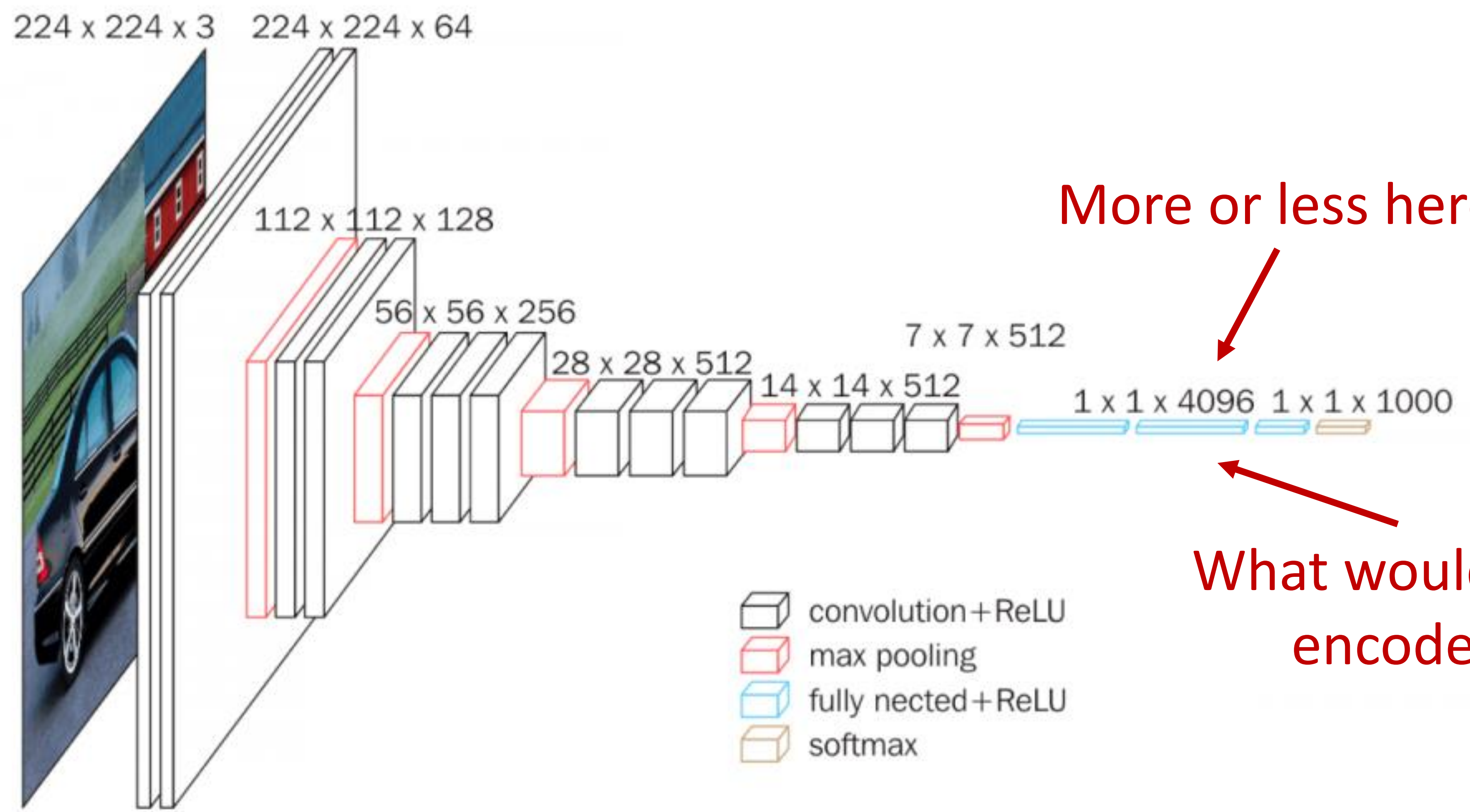


Building embeddings

- Building embeddings is pretty *hard*
  - Already when we have meaningful features
  - But for **relational data**? Pixels, words, videos, sound, …?

- Supervised task in DL architecture *might* create embedding
  - Output of a (deep) module can be interpreted as vectorial space
  - Encoding **meaning related to supervised task**

Embeddings from trained deep networks



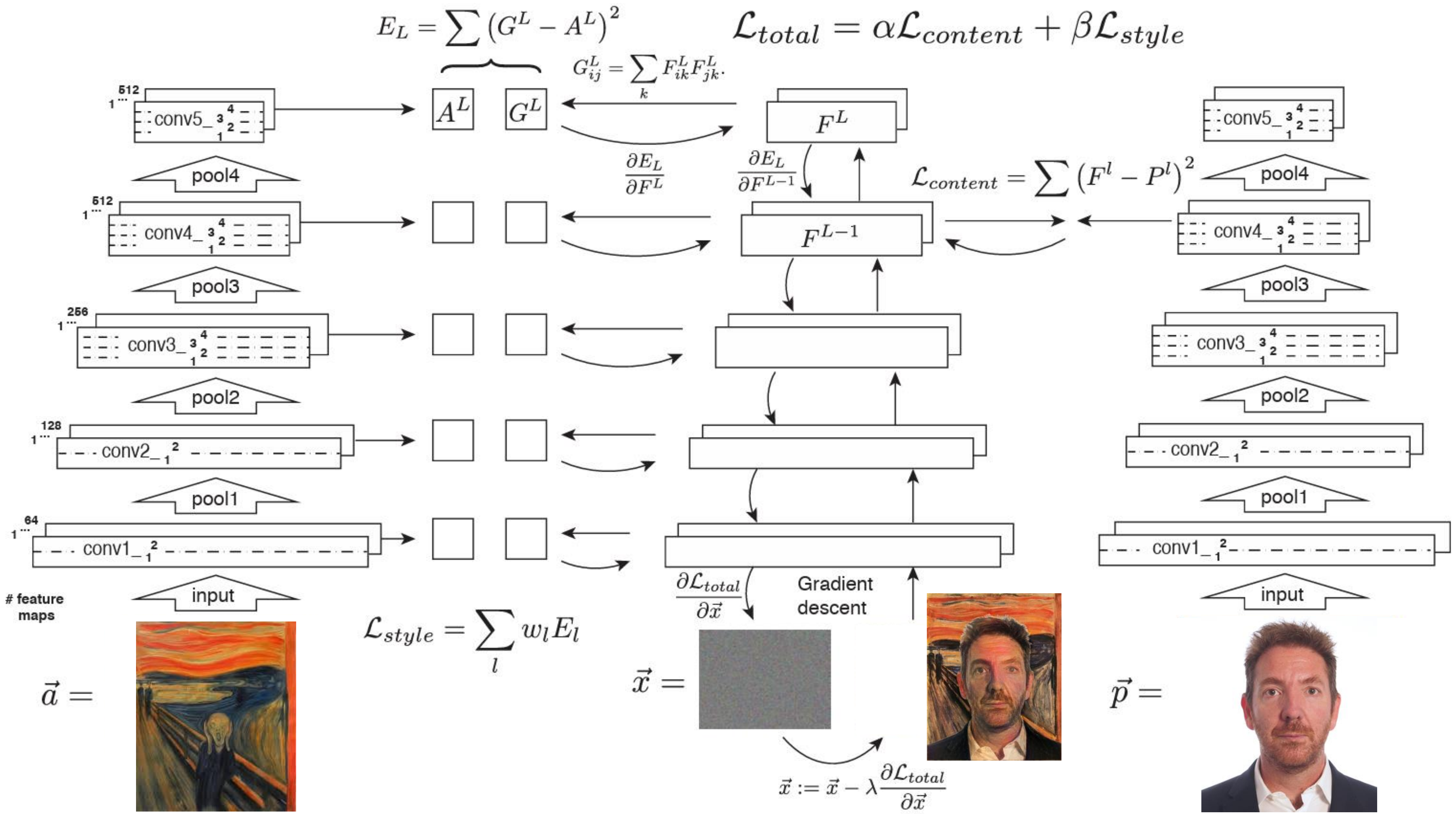
More or less here

What would this encode?

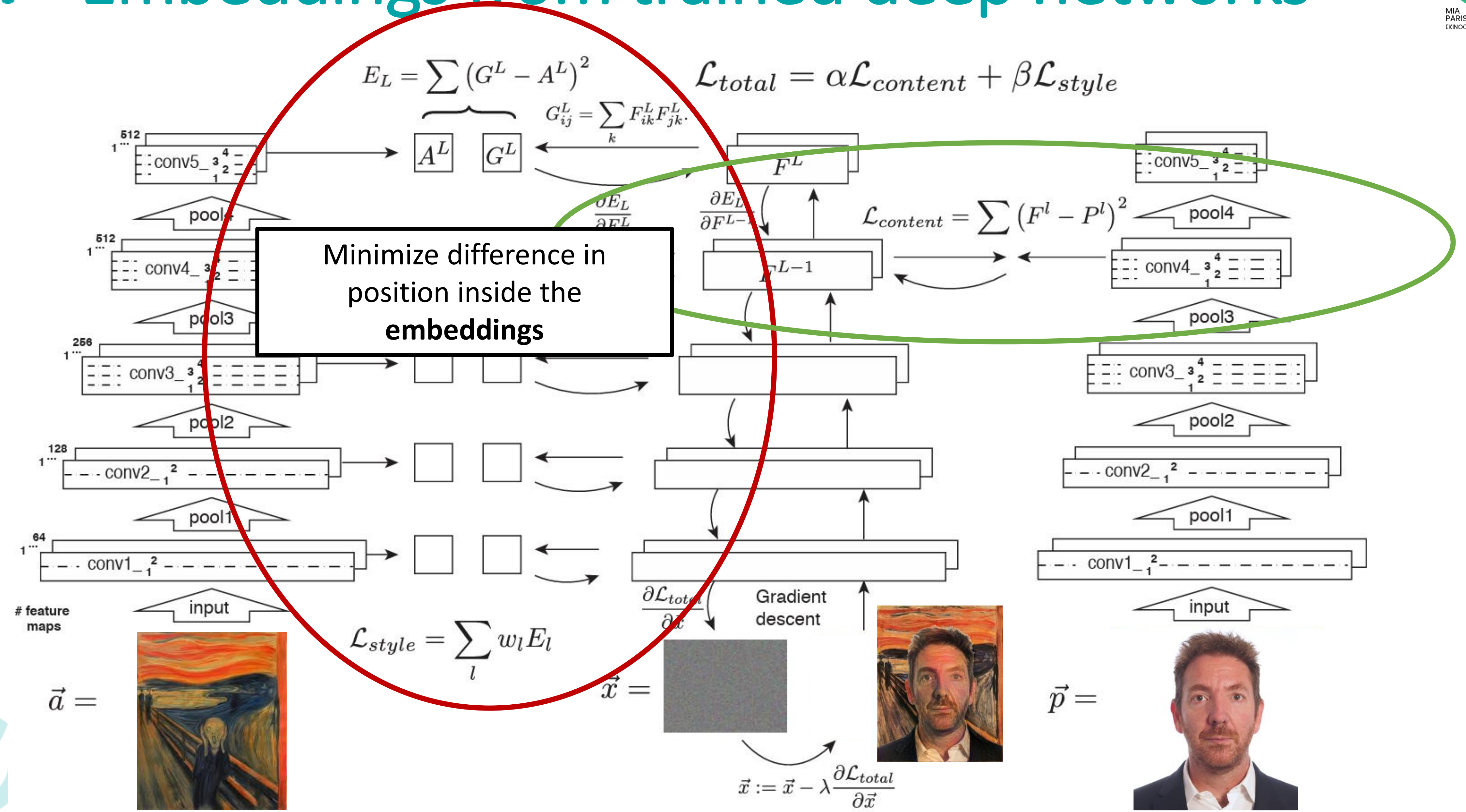
Embeddings from trained deep networks

- Taking VGG-16, output of deep modules
  - Might encode samples of same class close together
  - If we are lucky, samples of *visually similar classes* close together
  - However, encoded meaning related to **visual aspects**, only
  - Photo of bear and tiger very different, both ferocious predators

Embeddings from trained deep networks



Embeddings from trained deep networks



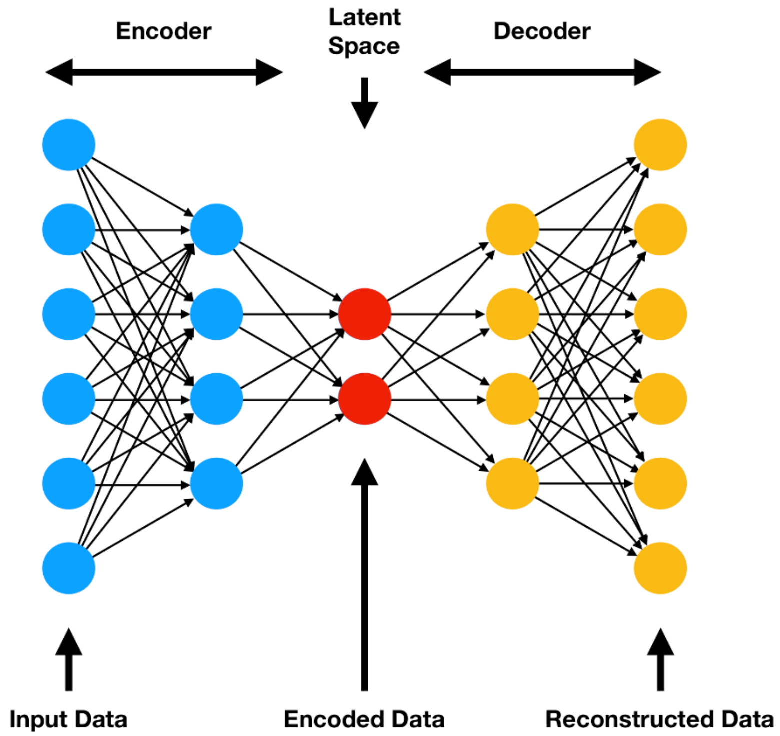
Minimize difference in position inside the **embeddings**

Embeddings, no tasks?

- What if do not have a task, or a ground truth?

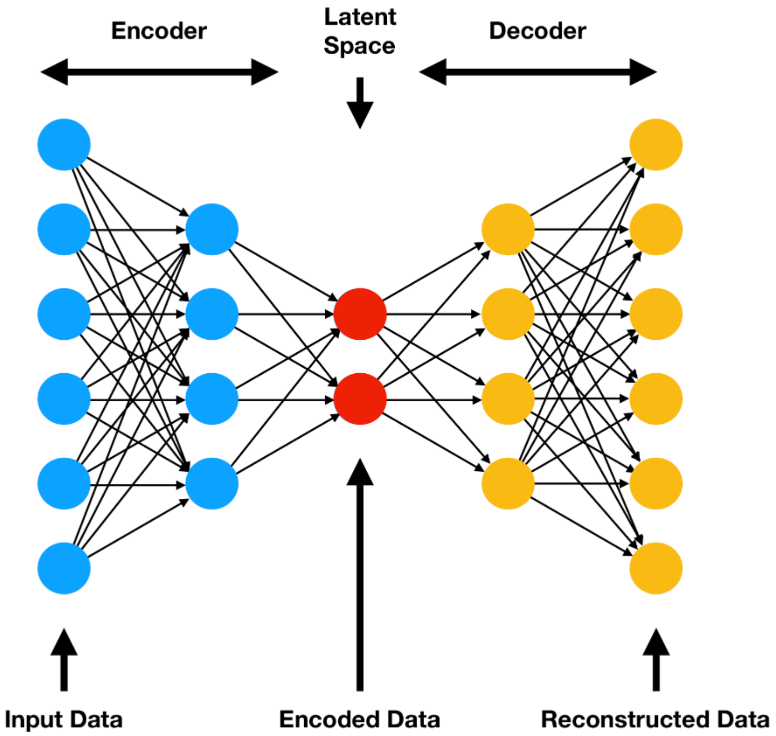
Embeddings with Autoencoders

- Deep learning architecture
  - Series of modules with less and less output dimensions
  - A bit like the "funnel" in CNNs
  - Module in the middle, lowest dimension
  - Second part, "funnel out"
  - Back to the original dimension of input
- Loss function: difference input-output
  - What does it mean?

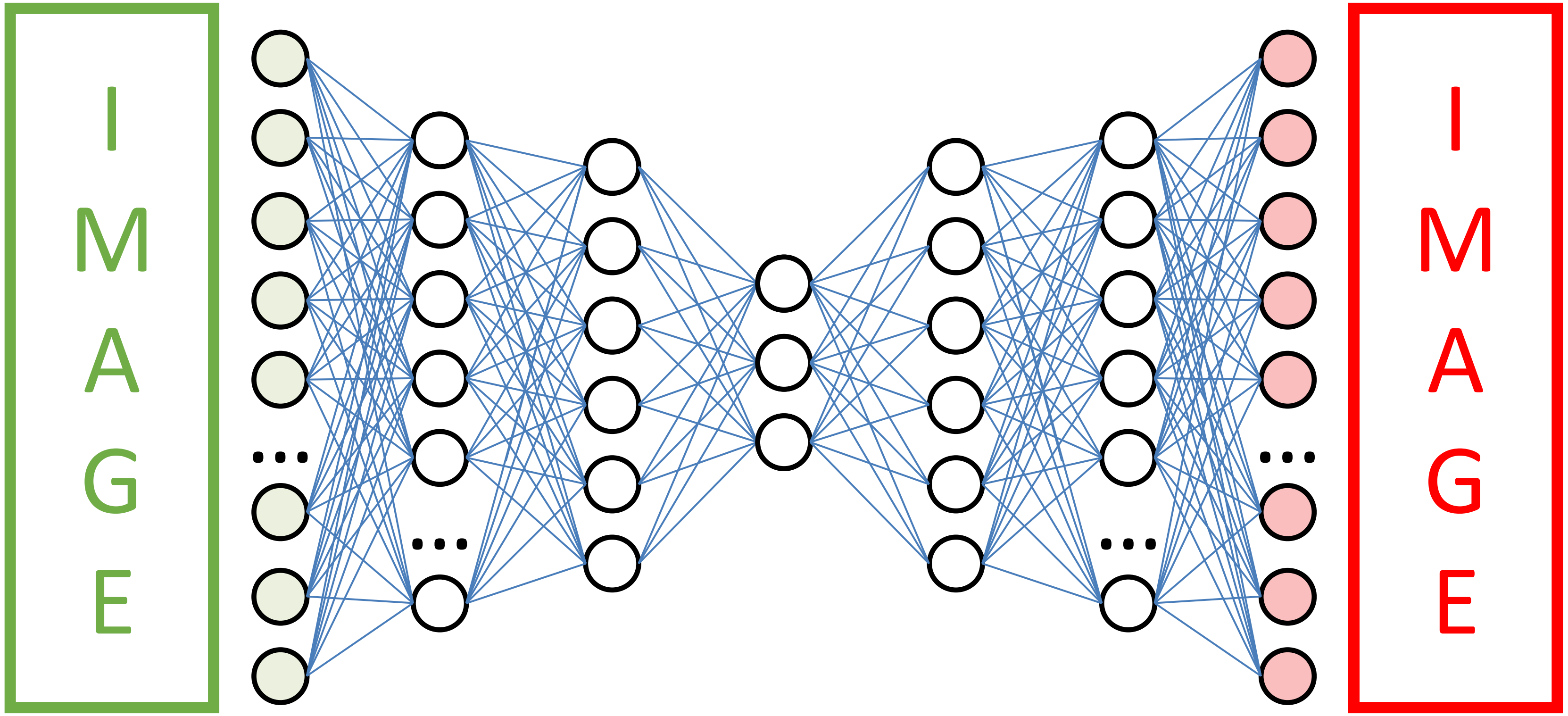


Embeddings with Autoencoders

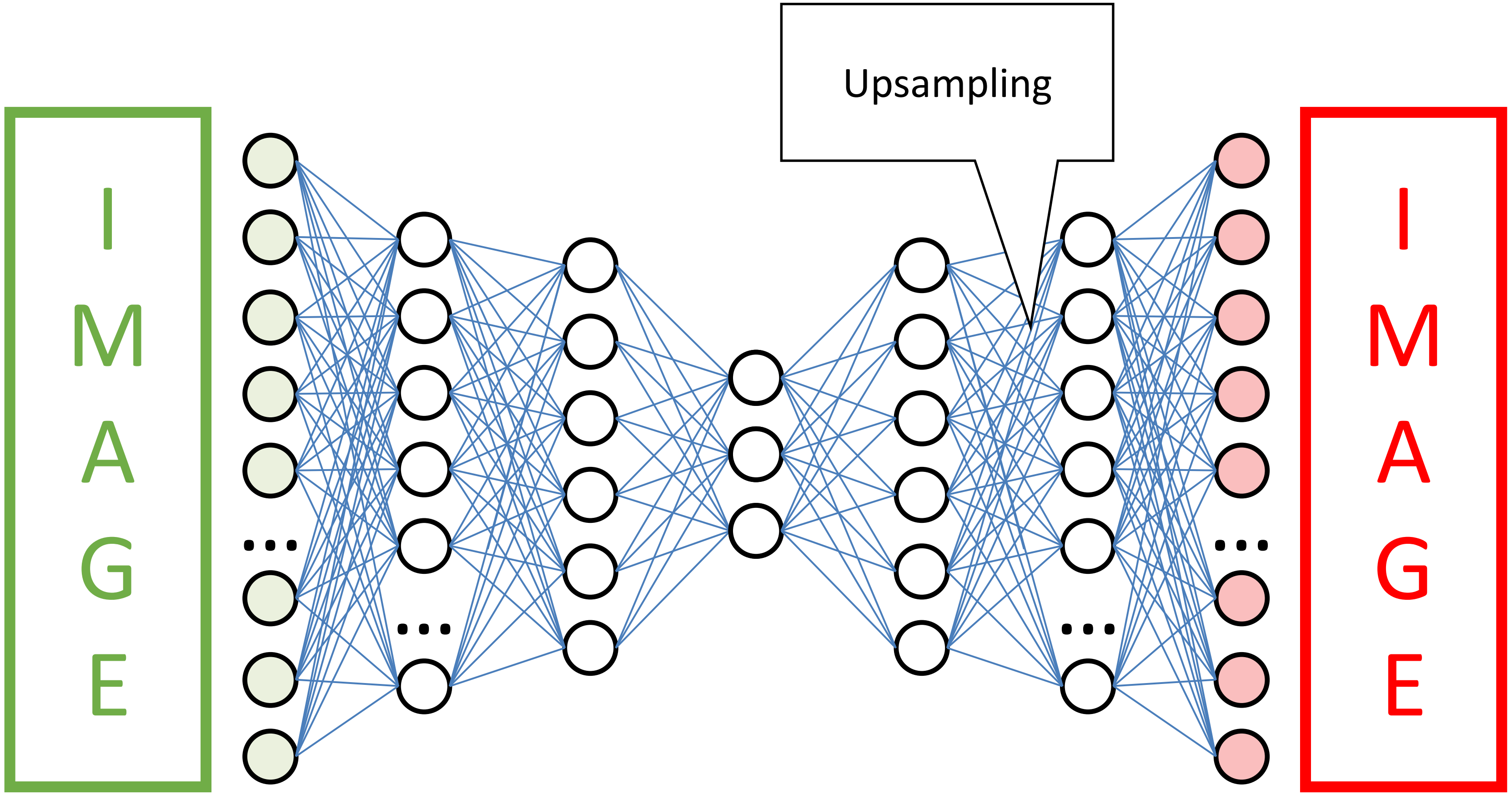
- Deep learning architecture
  - Series of modules with less and less output dimensions
  - A bit like the “funnel” in CNNs
  - Module in the middle, lowest dimension
  - Second part, “funnel out”
  - Back to the original dimension of input
- Loss function: difference input-output
  - Dimensionality reduction
  - **Meaningful**, to be able to reconstruct



Embeddings with Autoencoders

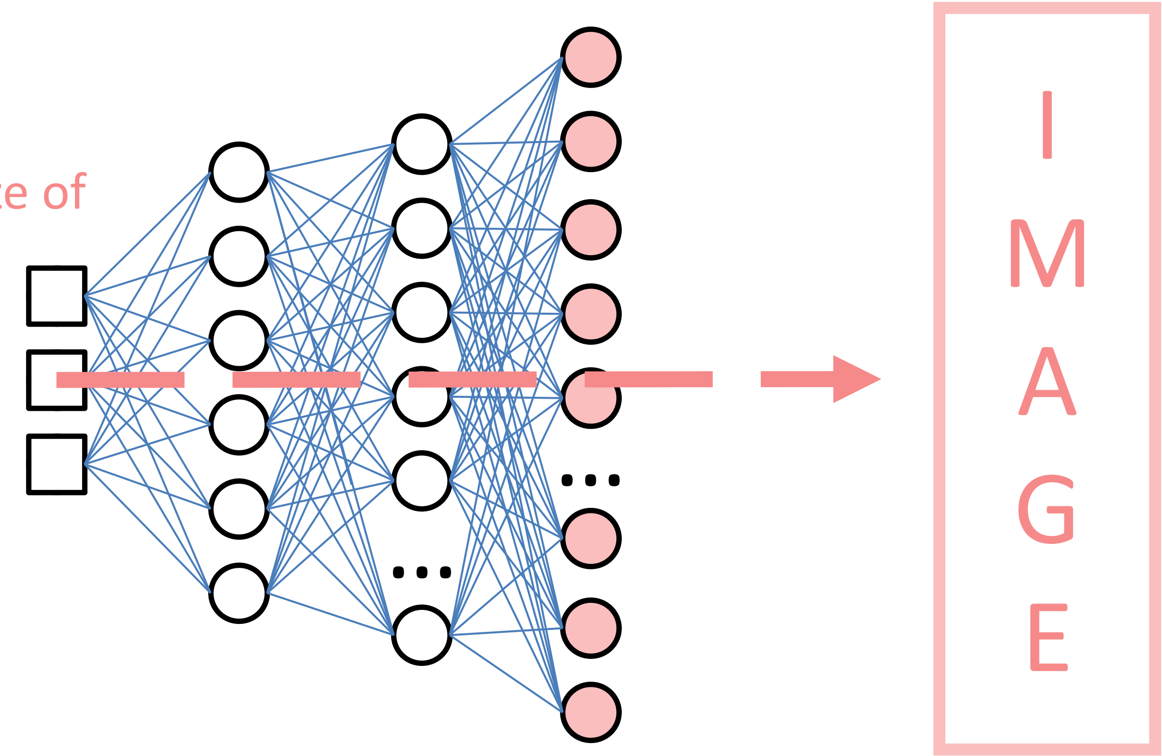


Embeddings with Autoencoders



Embeddings with Autoencoders

Random tensor of same size of the module output



Embeddings with Autoencoders

Embeddings with Autoencoders

- Two different points (vectors) in the same embedding…

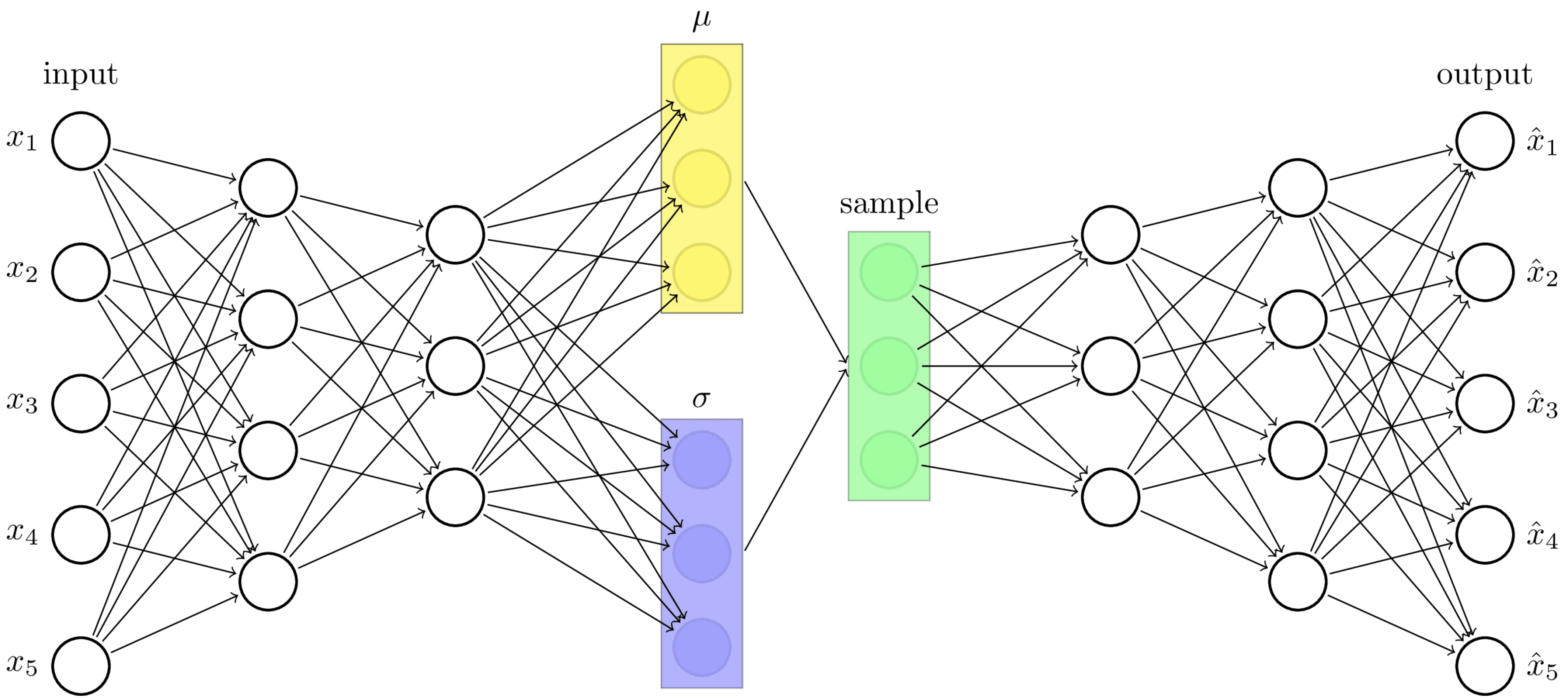
Embeddings with Autoencoders

- The latent space is brittle, full of “holes”
- Can we force this latent space to behave better?
- And if so, what parts of the NN would **you** act on?

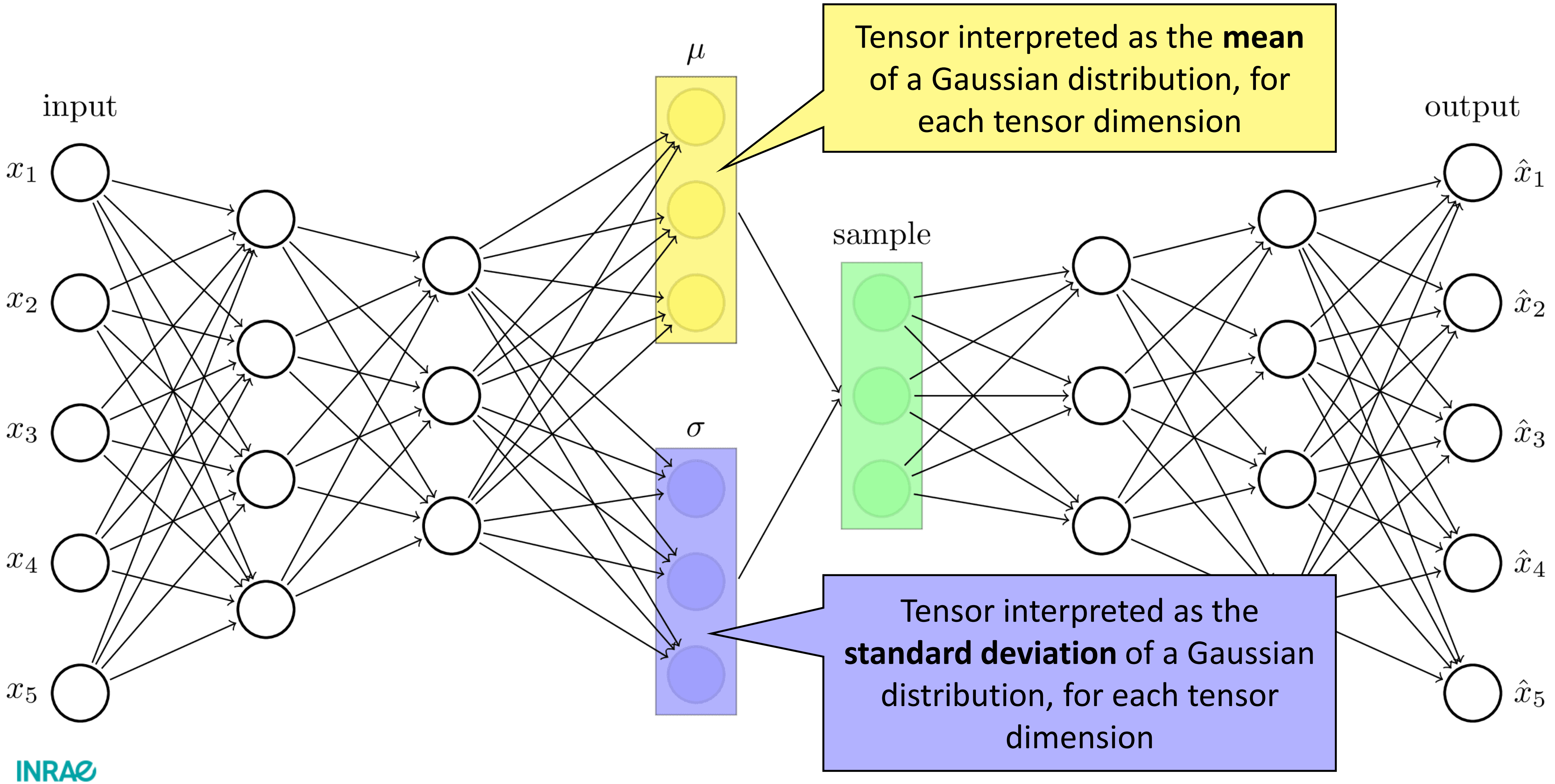
Variational Autoencoders

- Each image does **not** correspond to a single point
- But rather to a (Gaussian) distribution around a point
- This makes it possible to
  - Express *uncertainty* (small or large variance)
  - **Enforce structure** of the latent space (less brittle)

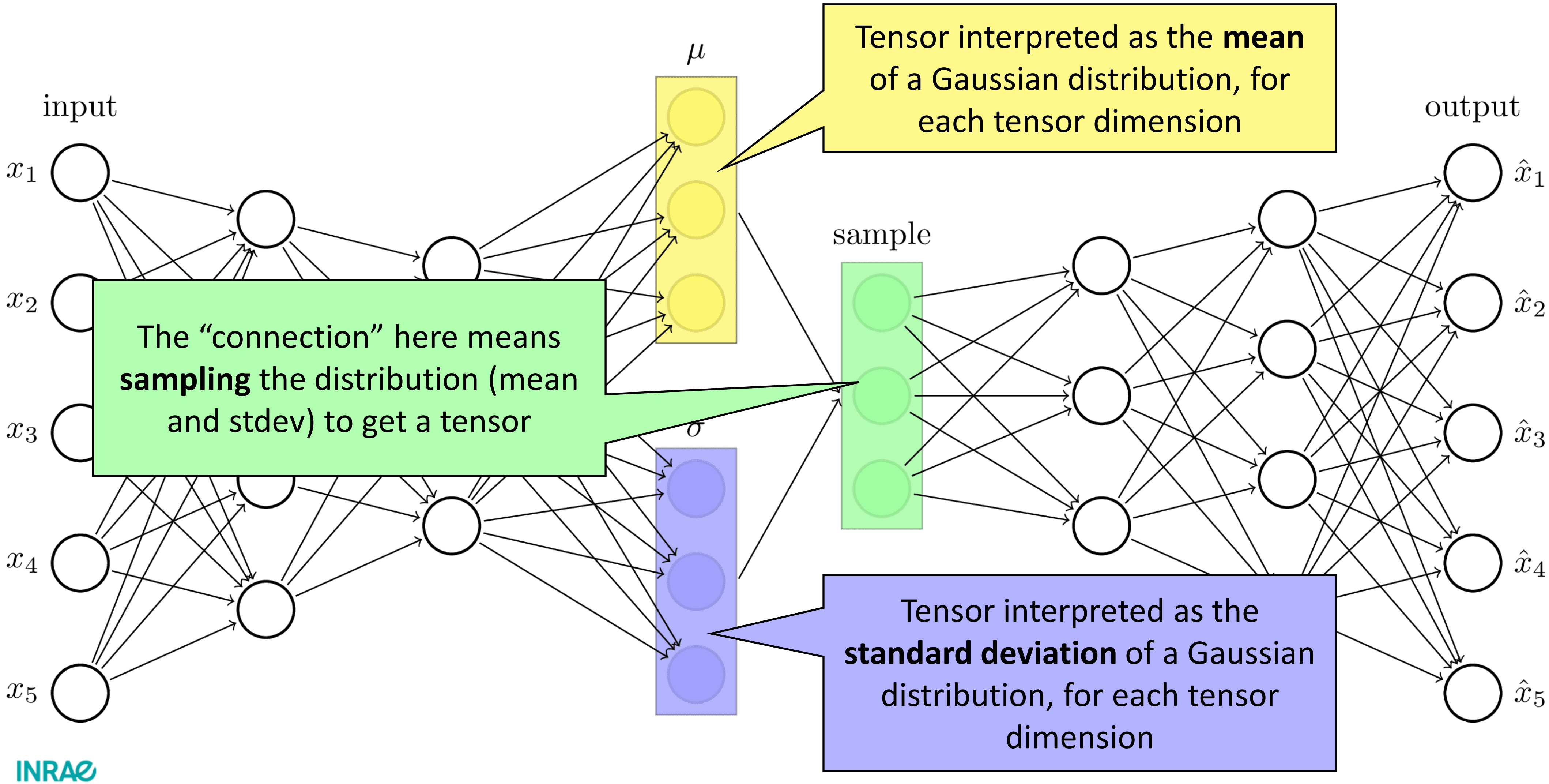
Variational Autoencoders



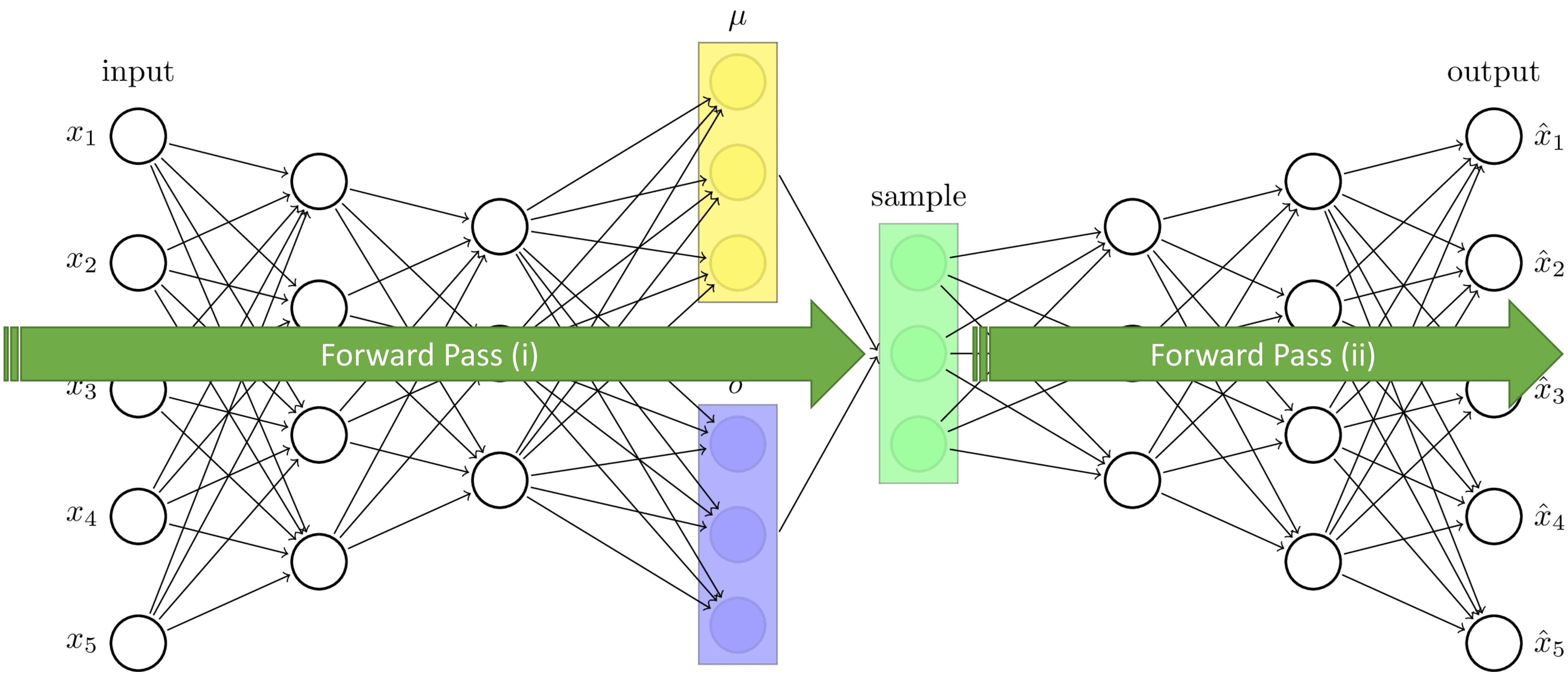
Variational Autoencoders



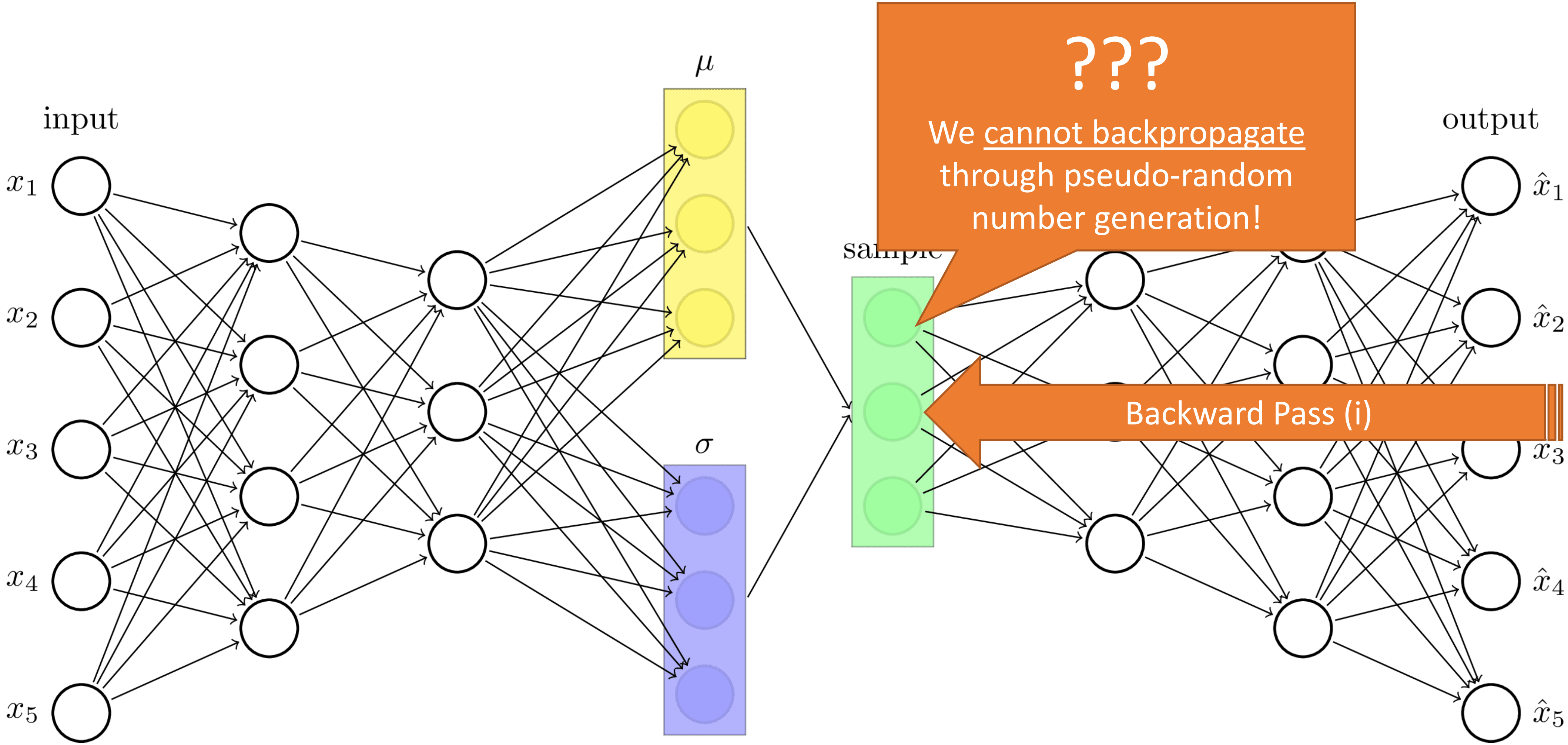
Variational Autoencoders



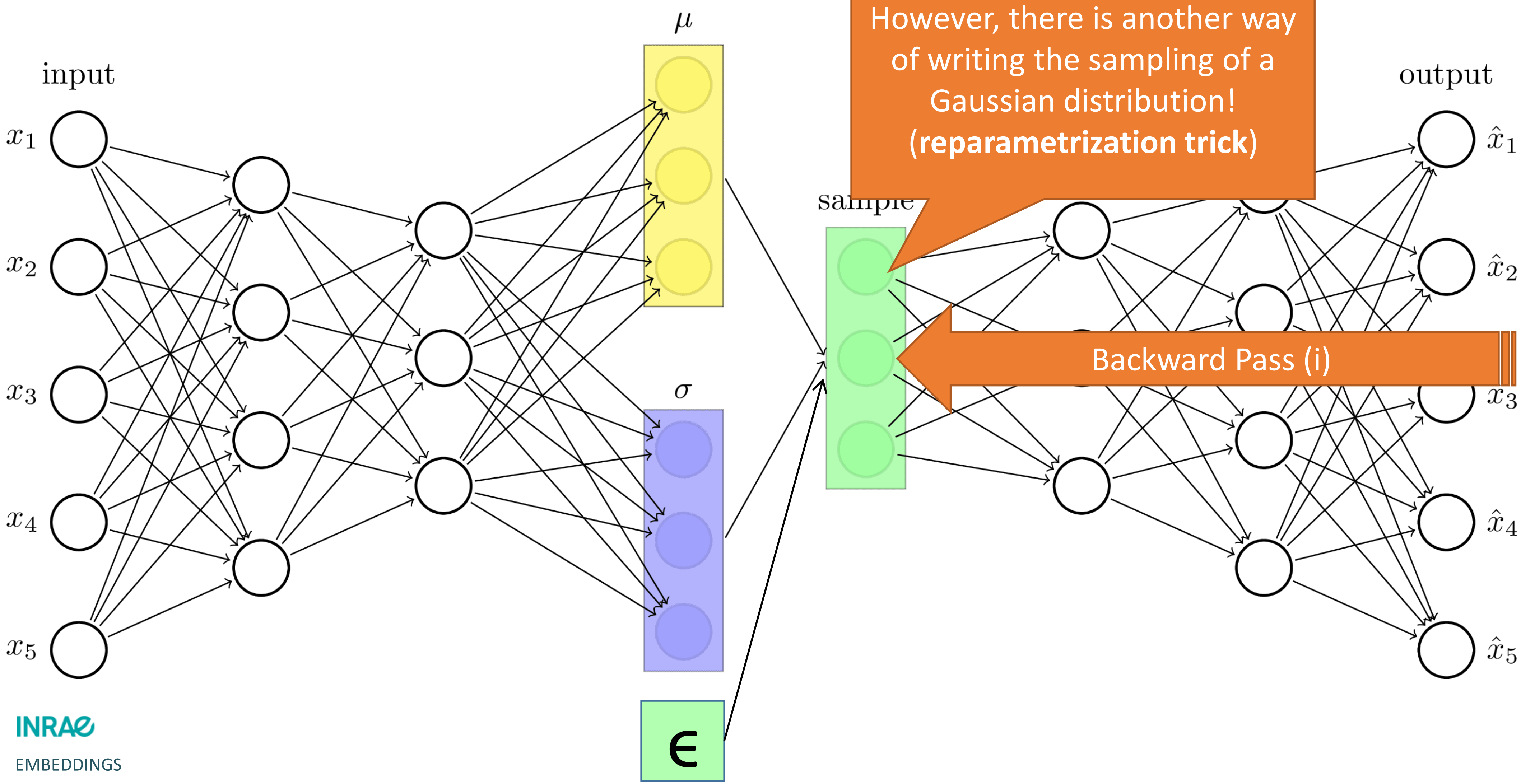
Variational Autoencoders



Variational Autoencoders

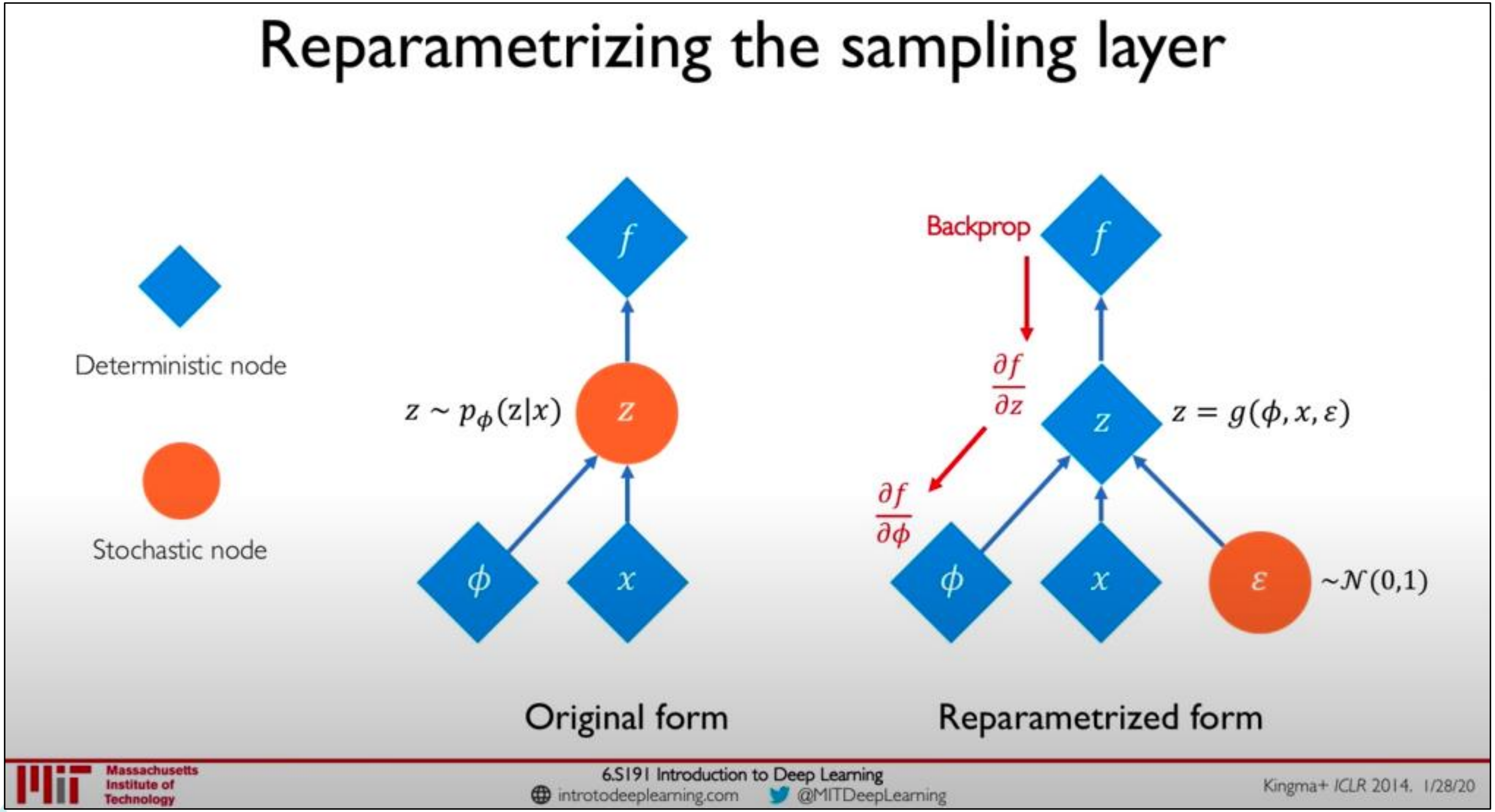


$$z \in \mathcal{N}(\mu, \sigma^2) \longrightarrow z = \mu + \sigma \odot \epsilon, \text{ where } \epsilon \in \mathcal{N}(0,1)$$

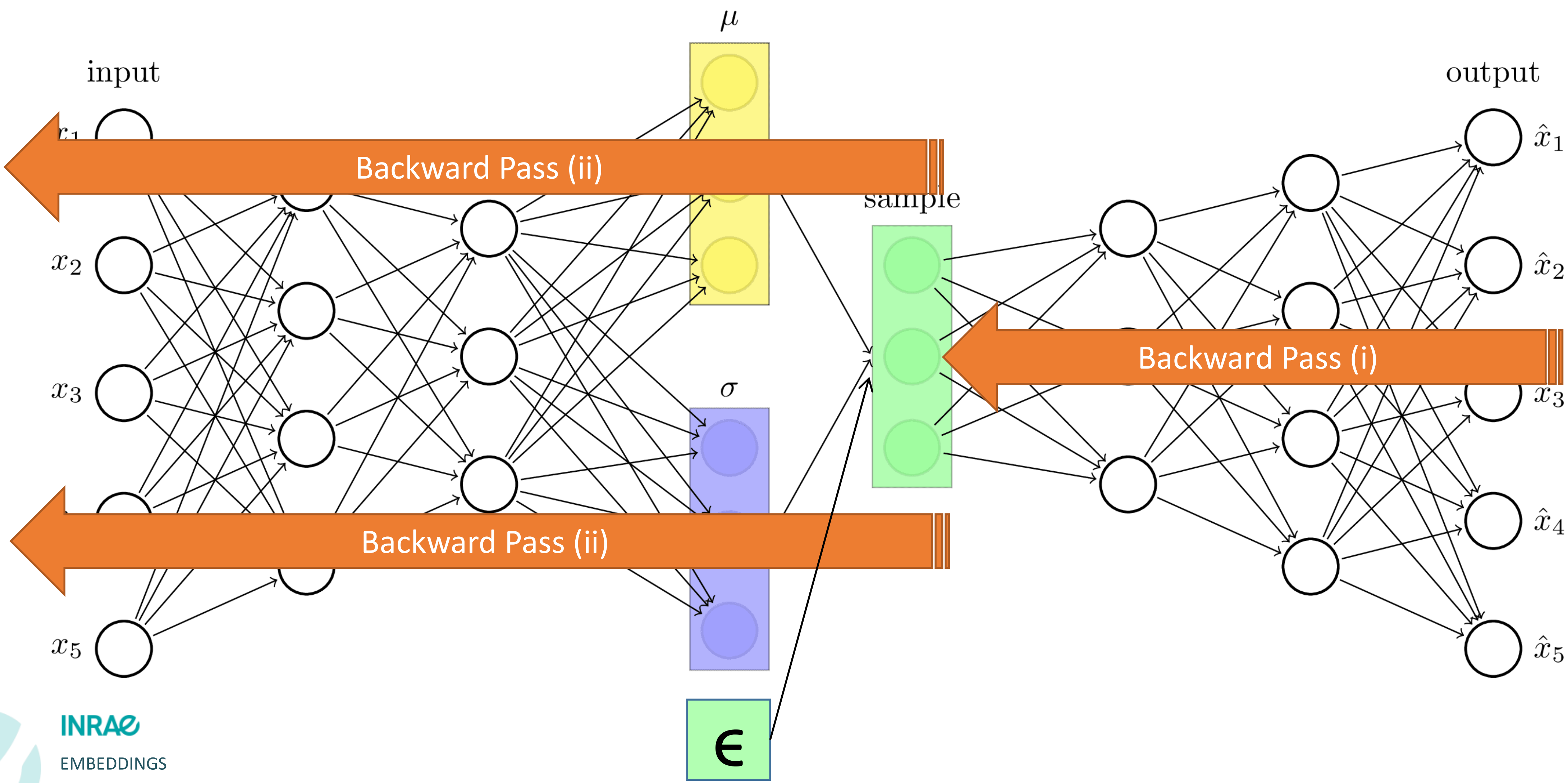


However, there is another way of writing the sampling of a Gaussian distribution! (**reparametrization trick**)

Variational Autoencoders



Variational Autoencoders



Variational Autoencoders

- Loss function in two parts
- The first one checks correctness of the output
  - For example, pixel-wise MSE or binary cross-entropy (BCE)
  - BCE assumes that pixels are "on" or "off" {0,1}
- The second one tries to keep distribution close to $\mathcal{N}(0, I)$
  - In other words, for each dimension Gaussian has $\mu = 0$
  - And covariance matrix is an identity matrix I
  - Which means that all latent dimensions are independent
- They can be weighted with a hyperparameter (β-VAE)

Variational Autoencoders

- Evaluating the difference between two distributions is hard
- Kullback–Leibler divergence

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$

- However, if both distributions are Gaussians, becomes easier

Embeddings of Vocabularies

- Vocabulary
  - Set of finite size, discrete elements (element: “token”)
  - Classic example: words, characters, bitstrings
  - But also, e.g. possible chessboard configurations in chess

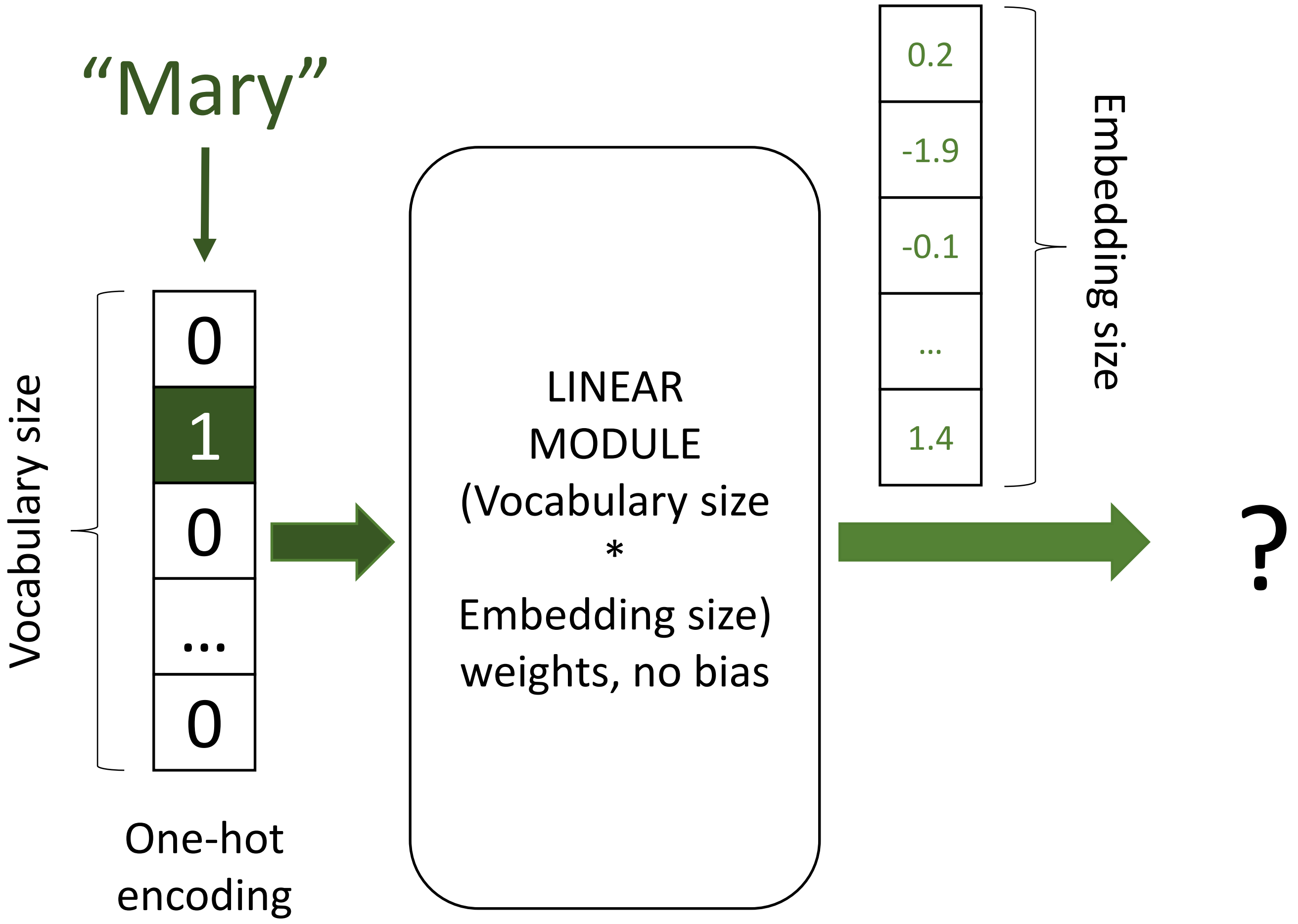
- **Words**, in particular, are difficult to manipulate for AI
  - Map to symbols, but not 1:1
  - “Chicken” vs “chicken”! “Mole” vs “mole” vs “mole”! *Context*?!?
  - Discrete: an integer per word? Not really expressive…

Vocabulary embedding: Word2Vec

- Revolutionary idea (grandfather of ChatGPT)
  - Arbitrary number of dimensions for the embedding
  - (n_vocabulary, n_embedding)

- Simple neural network architecture
  - Input: one-hot encoding of a vocabulary
  - Output/Loss: task connected to meaning
  - Use **learned weights** as **word embeddings**

Vocabulary embedding: Word2Vec



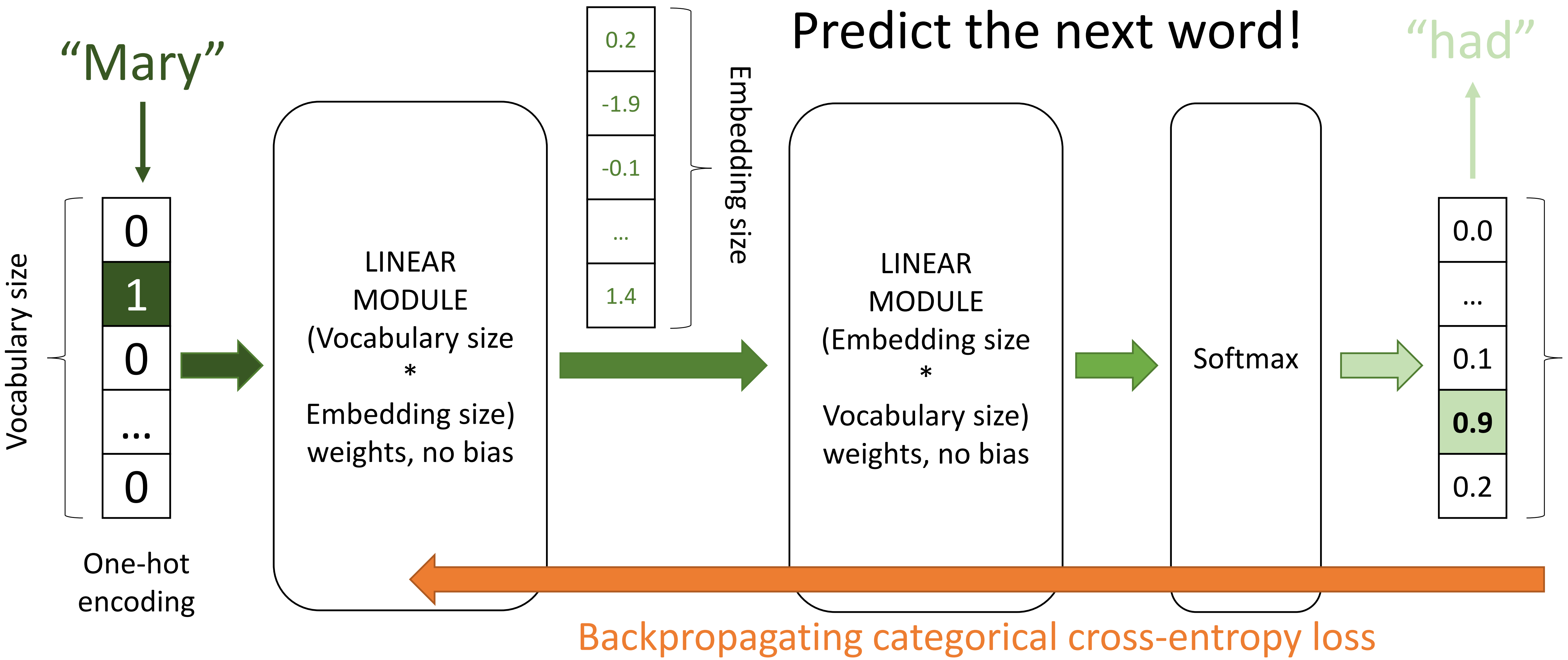
We don't have a task! How do we know if this representation is meaningful?

Vocabulary embedding: Word2Vec

- We already have lots of organized **relational data**
  - Text! Words are not randomly organized: sentences, paragraphs
  - Use text to generate meaningful training samples
  - Classic **unsupervised learning** tricks

- Slide a window on the text
  - For example, predict the next word
  - “Mary had a little lamb”
  - (Mary, had), (had, a), (a, little), (little, lamb)

Vocabulary embedding: Word2Vec



Vocabulary embeddings: Word2Vec

- Continuous Bag of Words (CBoW)
  - Predict the word between two other words
  - “Mary had a little lamb”
  - (Mary, had), (a, had); (had, a), (little, a); (a, little), (lamb, little)

Vocabulary embeddings: Word2Vec

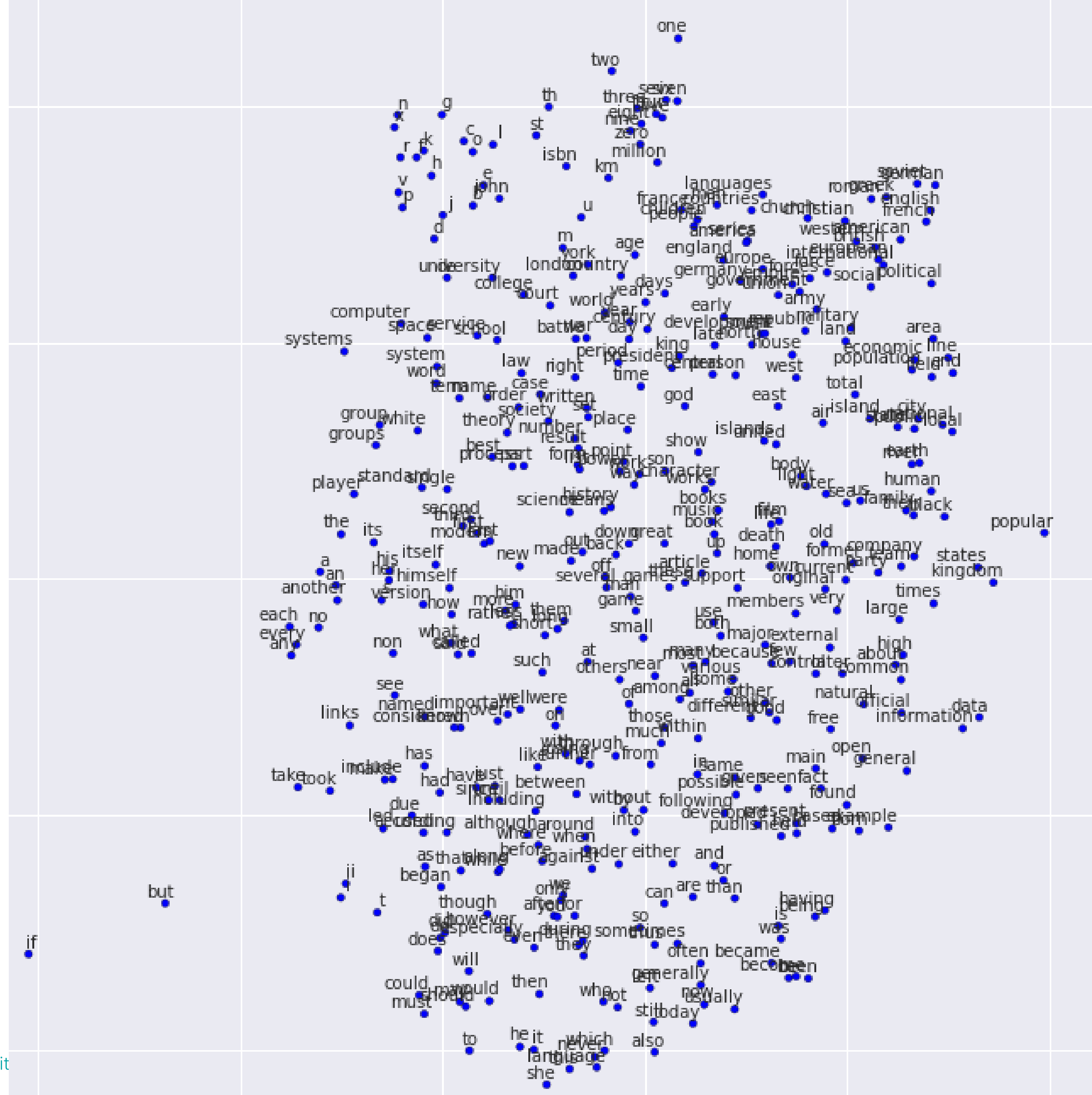
- Another possibility, skip-gram
  - "Context window" centered on a word, with n words on each side
  - Create training samples from central word to each side word

The [wide road **shimmered** in the] hot sun.

shimmered, wide
shimmered, road
shimmered, in
shimmered, the

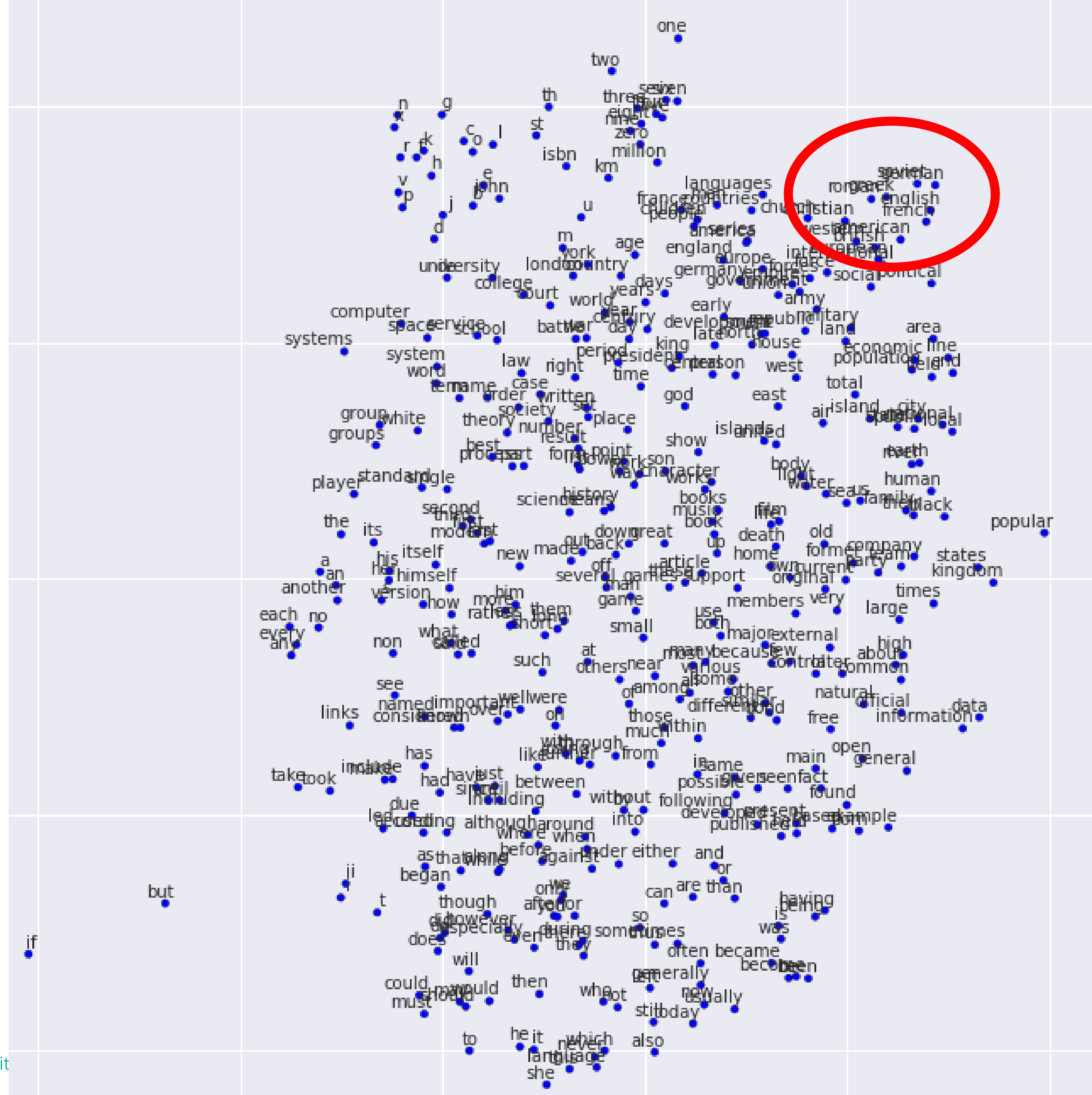
- Skip-gram also uses **negative sampling**
  - For each positive sample (shimmered, wide)
  - Get *random words* that **don't appear** in context window
  - Loss function to get softmax outputs as close as possible to zero

Word2Vec



one
two
isbn
million
km
languages
english
french
university
college
computer
systems
system
law
right
time
god
east
west
area
population
total
player
popular
death
home
states
kingdom
times
large
small
links
data
information
general
take
took
but
if
began
though
will
could
must
then
also
she

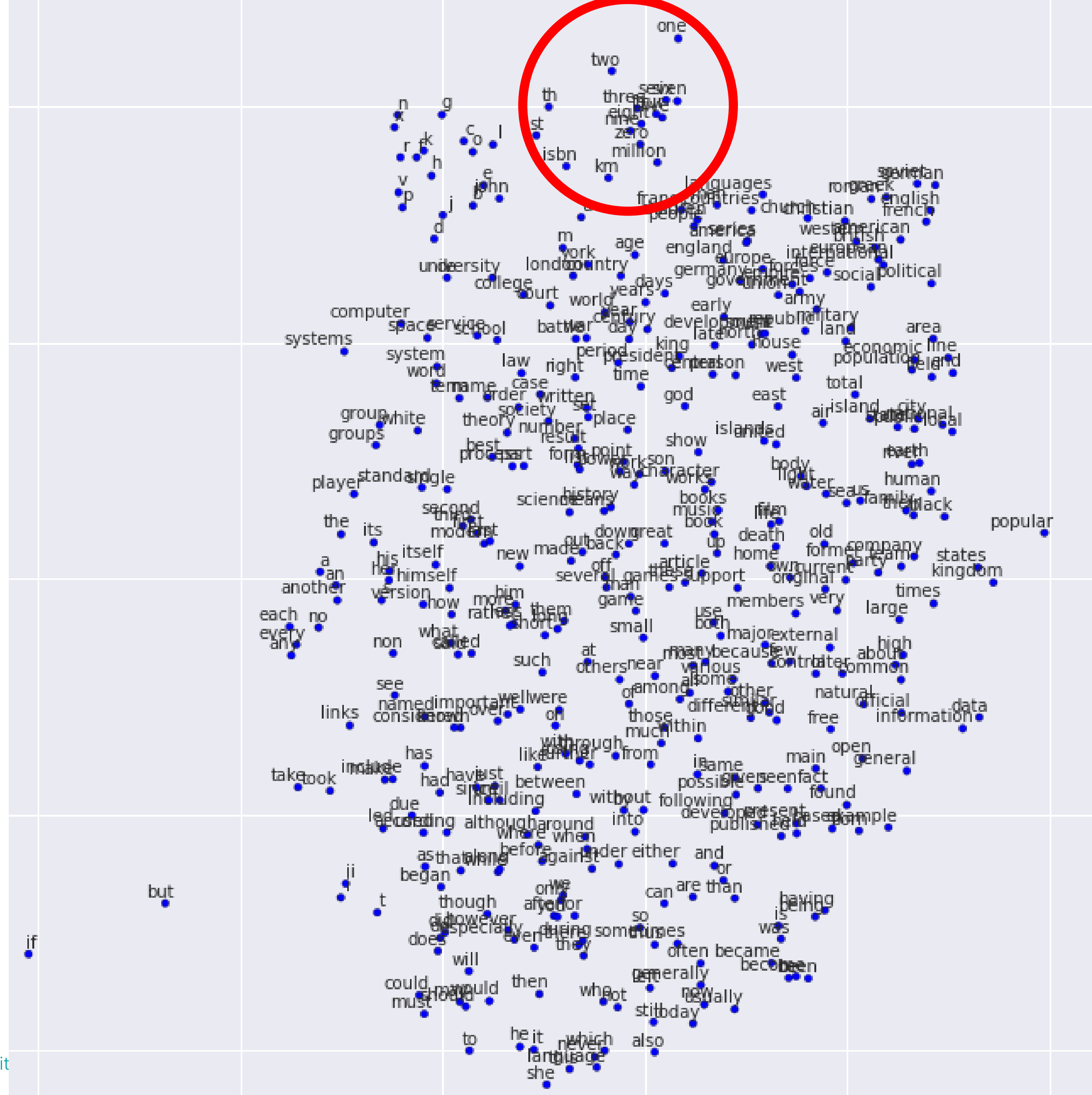
Word2Vec


one
two
th
st
isbn
million
km
languages
english
french
american
england
europe
germany
london
country
university
college
computer
systems
system
word
school
law
right
time
god
east
west
total
area
island
city
show
body
human
popular
player
standard
single
history
books
music
book
death
old
home
company
states
kingdom
the
its
itself
himself
another
version
new
made
several
members
very
large
times
each
no
every
any
non
what
such
small
major
external
high
common
see
links
important
those
much
natural
official
free
information
data
take
took
has
had
between
from
main
open
general
found
due
although
around
when
into
example
and
or
began
but
if
though
can
are
than
so
was
often
became
been
will
could
must
then
generally
usually
today
to
he
it
she
which
also

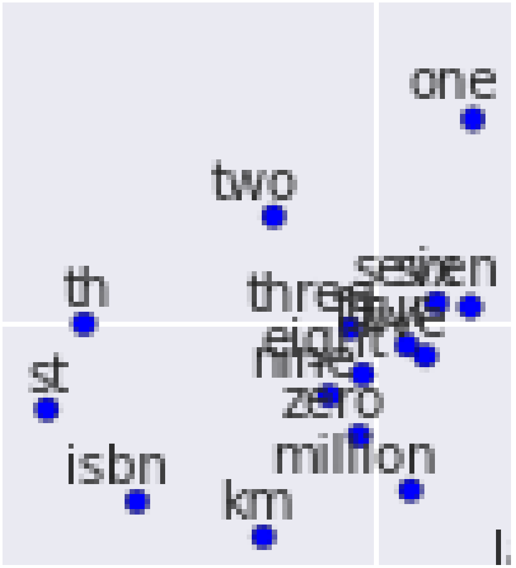
Word2Vec

- “French”, “British”, “American”…
  - Adjectives for nationality!
  - Nearby, you have “languages”, “countries”
  - Also, “England”, “Europe”, “International”, …

Word2Vec


one
two
th
st
isbn
km
nine
zero
million
languages
england
europe
age
university
college
computer
systems
system
law
right
time
god
east
west
area
total
island
show
body
human
popular
states
kingdom
the
its
itself
himself
another
each
every
any
no
non
such
links
see
take
took
has
had
between
from
main
open
general
found
but
if
began
though
will
could
must
then
to
she
also
which
often
became
generally
usually
today

Word2Vec

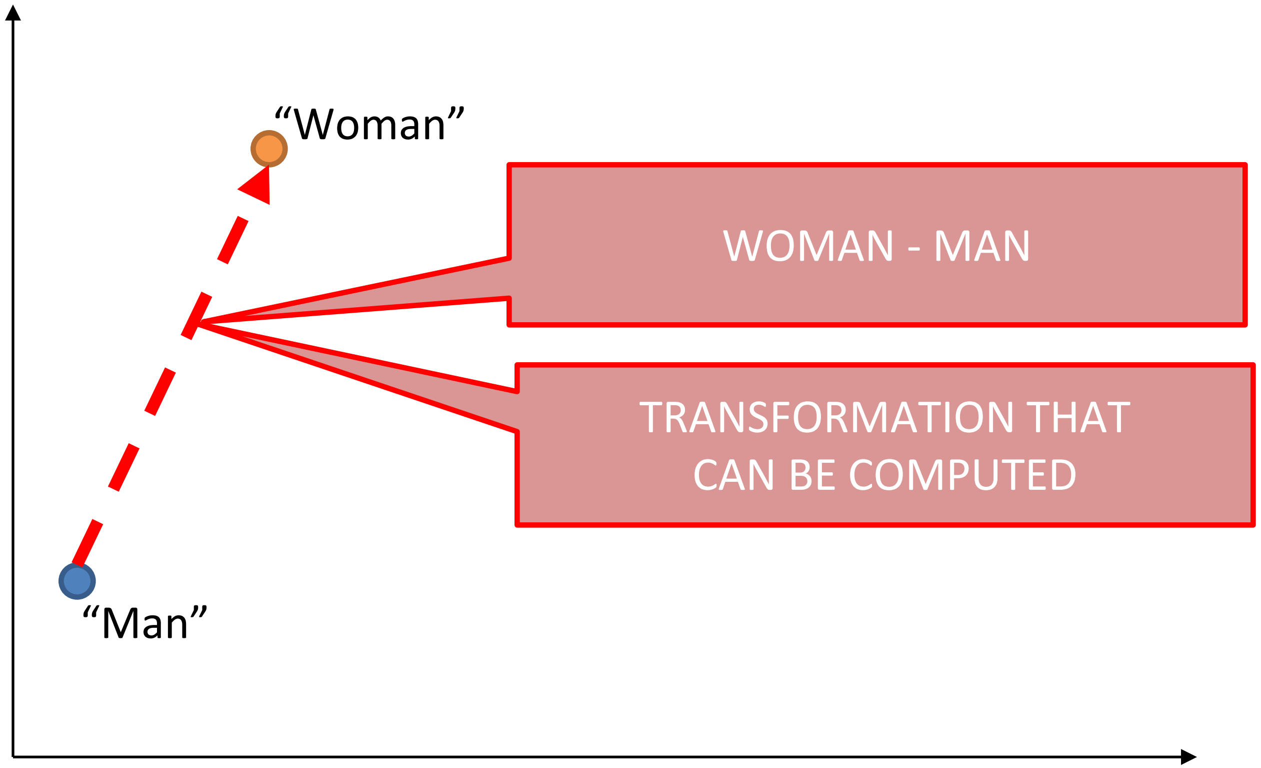


- “one”, “two”, “zero”, “seven”, “million”…
  - Numbers, quantities; nearby, some units of measurement
  - Also “th”, and “st”, as in 9-th, 1-st
  - ISBN (guess usually appears nearby numbers!)

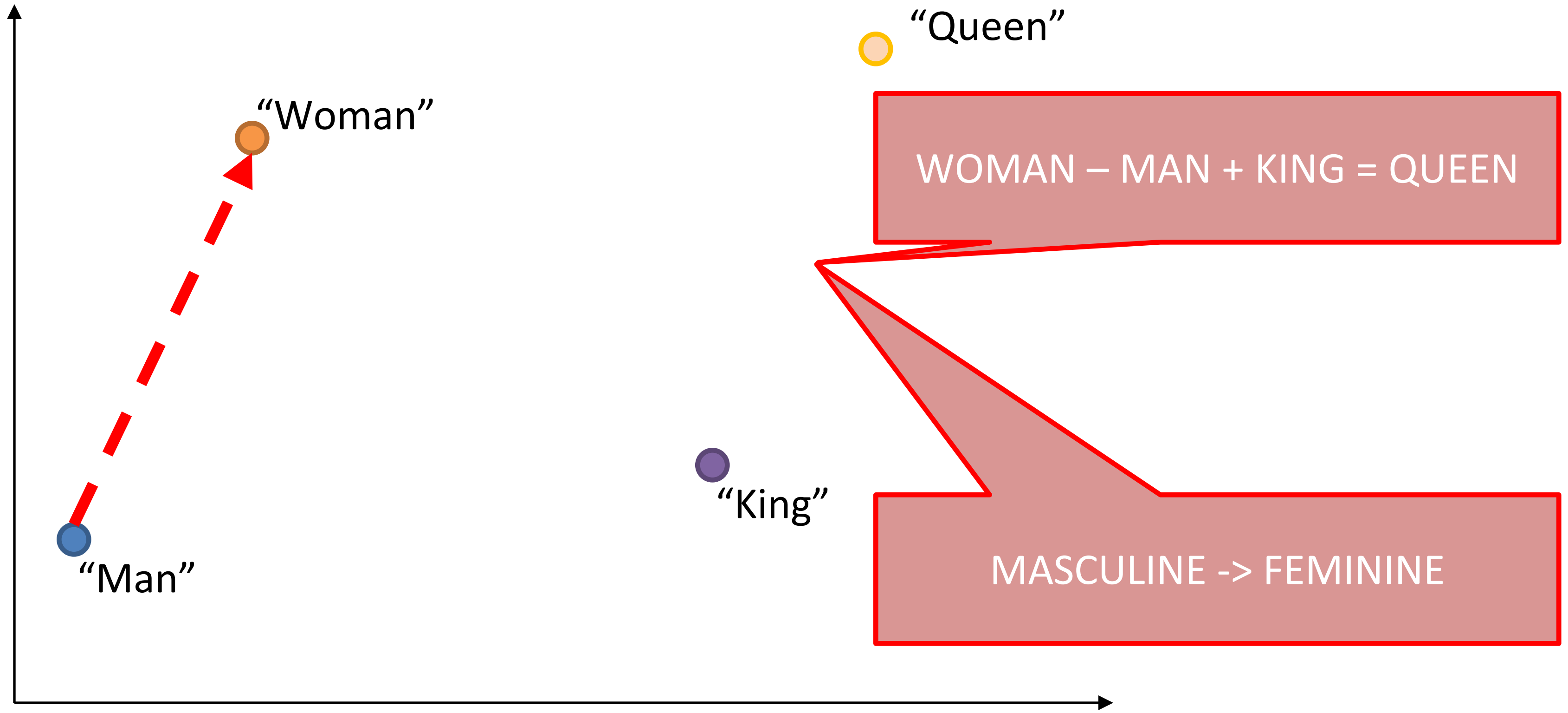
Word2Vec

- What is happening here?
  - Algorithm has **no semantic** info (**no meaning**)
  - But words with **similar meaning** are **close**
- Just by looking at the position of words in text
  - Similar *use* -> same *positions* with respect to other words
  - Word2Vec captures *some* aspects of meaning
- Can we do something else with Word2Vec?

Word2Vec

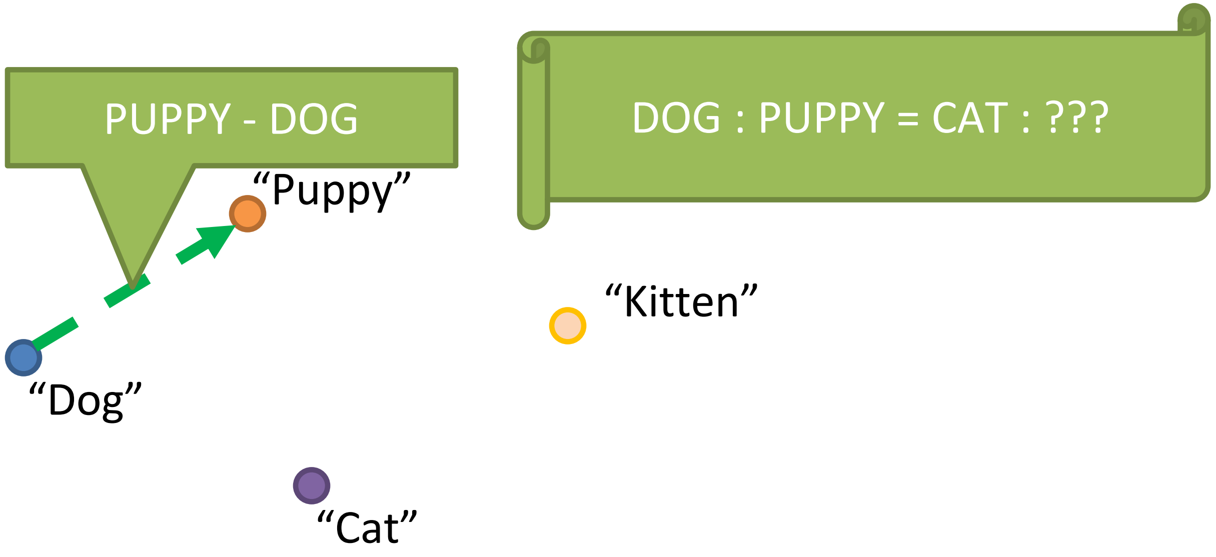


Word2Vec


"Queen"
"Woman"
WOMAN – MAN + KING = QUEEN
"King"
"Man"
MASCULINE -> FEMININE

Word2Vec

- Let's think about this for a second
  - No meaning inserted into the algorithm
  - Words with similar meaning cluster together
  - *Mathematical* transformations provide *meaningful* results



Word2Vec

- What does all this mean?
  - We are not really sure
  - Obtain *semantics* just from word positions
  - Maybe semantics is not as hard of a problem as it looks
- And Word2Vec is an incredibly *simple* architecture!
  - More complex architectures might get even better results
  - And they did (Transformers, ChatGPT, etc.)

Creating a Vocabulary

- Creating a Vocabulary is not straightforward
  - Elements in Vocabulary are called “tokens”
  - “Tokenization” is the process to create good tokens

- It’s another optimization problem
  - “What is the set of tokens that gives me the best result?”
  - E.g., discard or keep less frequent tokens?

Creating a Vocabulary

- Example: text from natural language?

Creating a Vocabulary

- Example: text from natural language?
- Seems pretty intuitive, 1 word = 1 token; but nuances
  - Special tokens, <SOS> (Start of Sequence) and <EOS> (End)
  - Actually more efficient to use **parts of a word**
    - “Cod” might be better than “Code”: “Cod-es”, “Cod-ing”, “En-cod-ing”
    - Kind like mapping to word roots, even if not exactly
- What happens if input is “**Grzegorz Brzęczyszczykiewicz**”?

Creating a Vocabulary

- In practice, solved by declaring **character-level tokens**
  - Tokenization attempts on largest possible matching tokens
  - If it fails, fall back on smaller and smaller tokens

- Test GPT-4 tokenization, platform.openai.com/tokenizer

Creating a Vocabulary

Questions?

Bibliography

- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781.