WHAT IS DEEP LEARNING?

WHAT IS DEEP LEARNING? [1]

- Deep Learning can be defined as the layering of simple algorithms called artificial neurons into networks several layers deep.
- Deep Learning involves a network in which artificial neurons – typically thousands, millions or many more of them – are stacked at least several layers deep.
- The artificial neurons in the first layer pass information to the second, the second to the third, and so on, until the final layer outputs some value.

WHAT IS DEEP LEARNING? [2]

- *"Deep learning algorithms seek to*
  - *exploit the unknown structure in the input distribution*
  - *in order to discover good representations,*
  - *often at multiple levels,*
  - *with higher-level learned features defined in terms of lower-level features."* ***Yoshua Bengio***

WHAT IS DEEP LEARNING? [3]

- *"Deep learning methods aim at learning feature hierarchies*
- *with features from higher levels of the hierarchy formed by the composition of lower level features.*
- *Automatically learning features at multiple levels of abstraction allow a system to*
  - *learn complex functions mapping the input to the output directly from data,*
  - *without depending completely on human-crafted features."*

Yoshua Bengio.

WHAT IS DEEP LEARNING? [4]

- *The hierarchy of concepts allows the computer to learn complicated concepts by building them out of simpler ones.*
- *If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers.*
- *For this reason, we call this approach to AI deep learning. – Deep Learning, Yoshua Bengio, Ian Goodfellow and Aaron Courville*

DEEP LEARNING AND NLP

- How is deep learning incorporated into human language applications
  - How can deep learning automatically learn features that represent the meaning of words
- *“The meaning of a word is its use in language.”*
- *“One cannot guess how a word functions. One has to look at its use and learn from that.”* Ludwig Wittgenstein, Philosophical Investigations.
  - Words on their own have no meaning.
  - Words derive their meaning by their use within the larger context of language
  - Natural Language Processing with Deep Learning relies heavily on this premise – word2vec technique

LEARNING REPRESENTATIONS AUTOMATICALLY [1]

Traditional ML techniques:

- Use clever, human-designed code that transforms raw data (images, audio, text, etc.) into input features for ML algorithms (regression, random forests or support vector machines)
- Adept at weighting features
- Not particularly good at learning features from data directly
- Manual creation of features is often a highly specialized activity

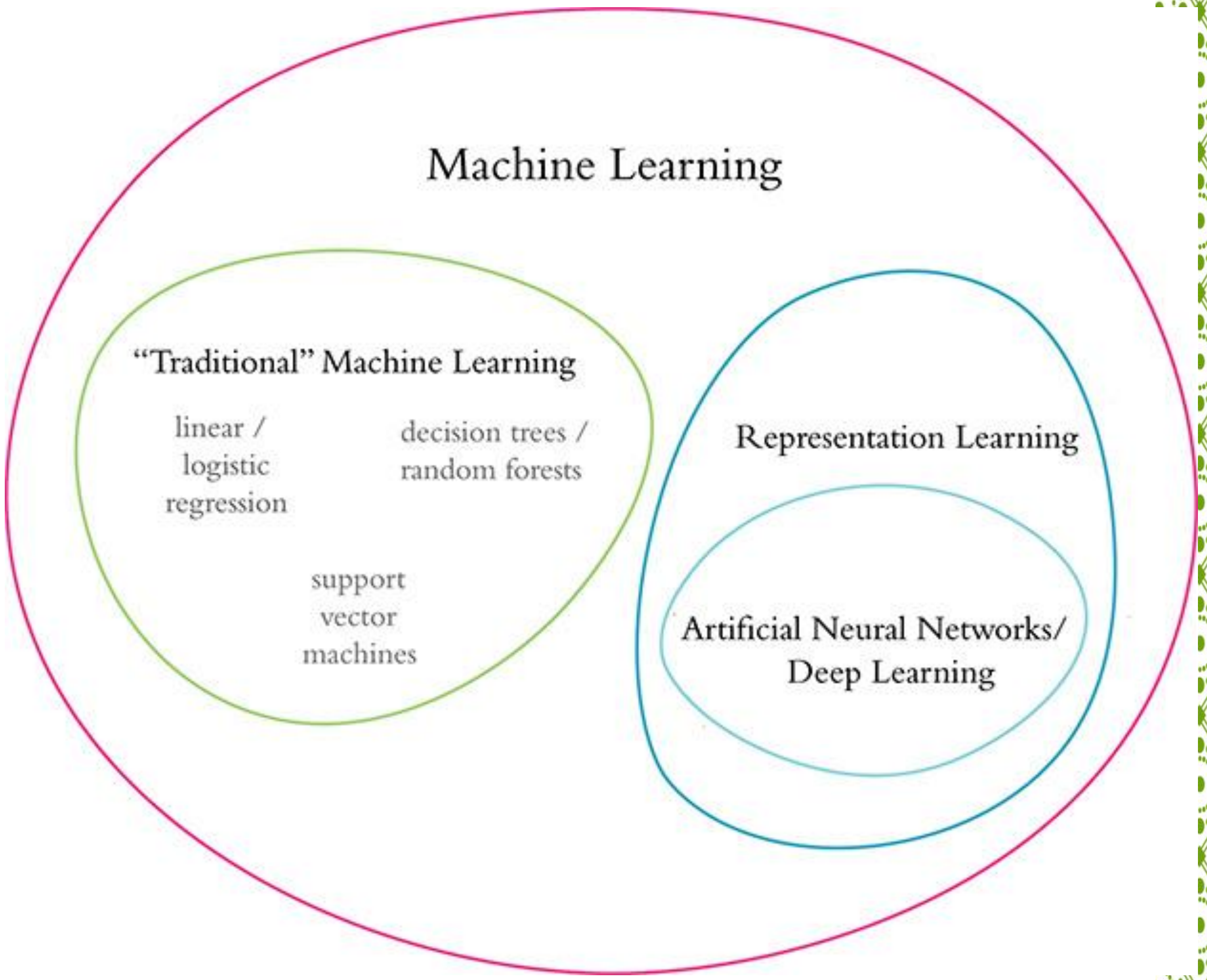
Features engineered by humans tend to be:

- less comprehensive
- Excessively specific to application and context
- Involve lengthy ongoing loops of feature ideation, design and validation (that could stretch years)

LEARNING REPRESENTATIONS AUTOMATICALLY [2]

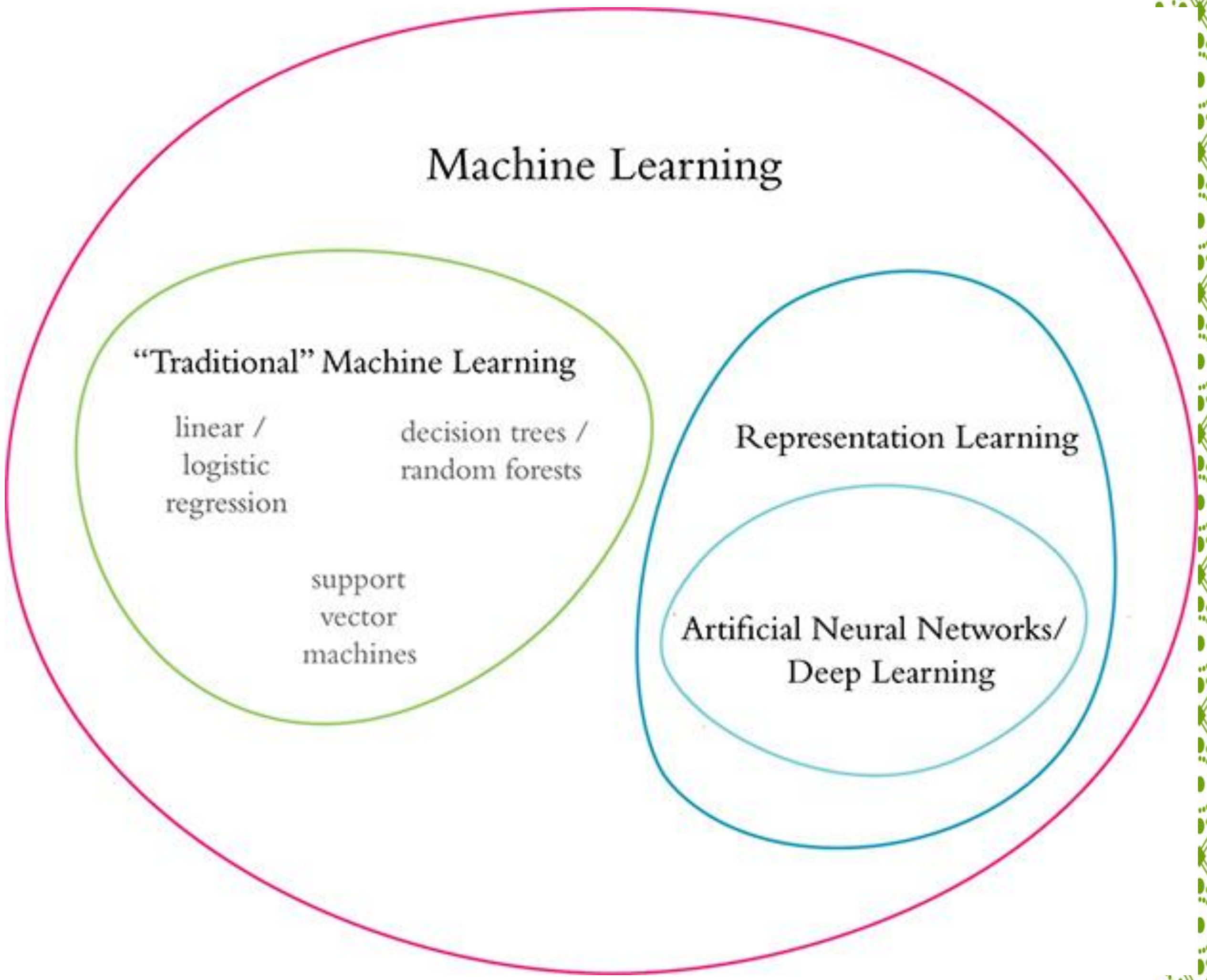
Representation learning refers to the class of techniques that learn features from data automatically.

- Feature learning and Representation learning are used interchangeably.



LEARNING REPRESENTATIONS AUTOMATICALLY [3]

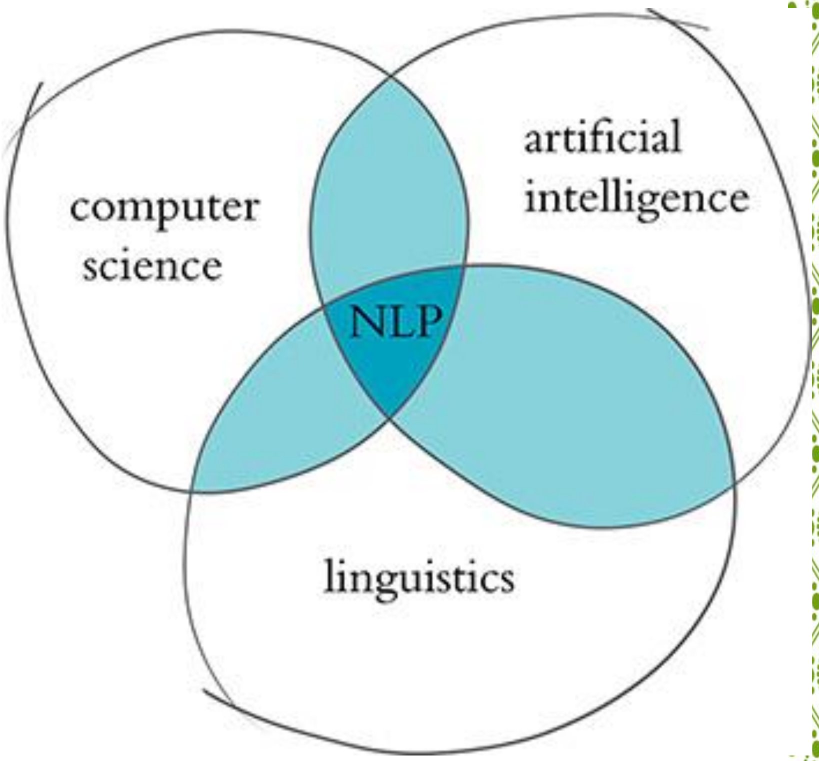
- Representation learning models
  - Generate features quickly (typically over hours or days of model training)
  - Adapt straightforwardly to changes in the data (e.g. new words, meanings or ways of using language)
  - Adapt automatically to shifts in the problem being solved



NATURAL LANGUAGE PROCESSING

Natural language processing (NLP)

- is a subfield of linguistics, computer science, and artificial intelligence
- concerned with the interactions between computers and human language
- how to program computers to process and analyze large amounts of natural language data.



Goals:

- "understanding" the contents of documents
- the contextual nuances of the language within them.
- Accurately extract information and insights contained in the documents
- Categorize and organize the documents themselves.

DEEP LEARNING & NATURAL LANGUAGE PROCESSING

NATURAL LANGUAGE PROCESSING (EXAMPLES)

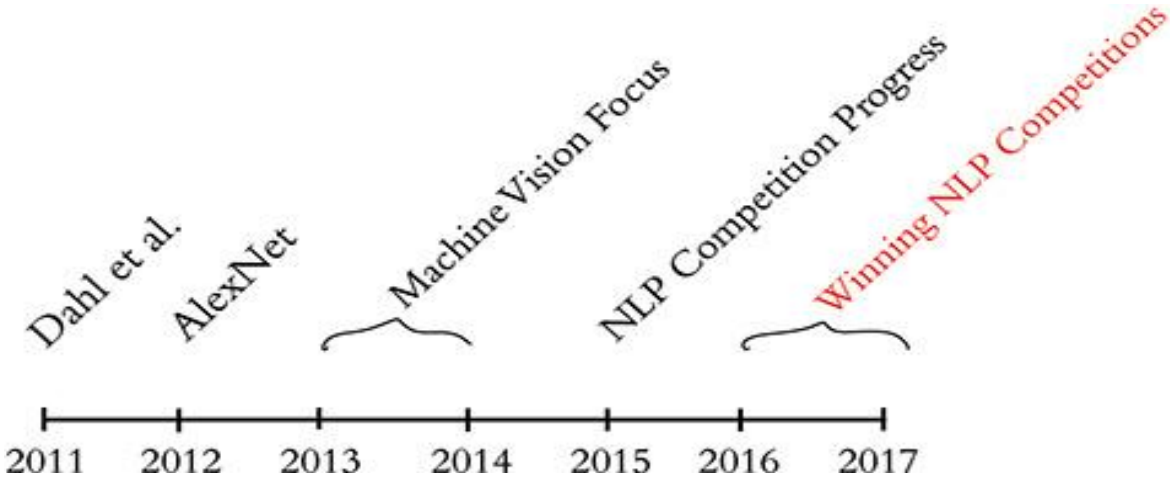
- Classifying documents: using the language within a document (e.g., an email, a Tweet, or a review of a film) to classify it into a particular category (e.g., high urgency, positive sentiment, or predicted direction of the price of a company's stock).
- Machine translation: assisting language-translation firms with machine-generated suggestions from a source language (e.g., English) to a target language (e.g., German or Mandarin); increasingly, fully automatic—though not always perfect—translations between languages.
- Search engines: autocompleting users' searches and predicting what information or website they're seeking.
- Speech recognition: interpreting voice commands to provide information or take action, as with virtual assistants like Amazon's Alexa, Apple's Siri, or Microsoft's Cortana.
- Chatbots: carrying out a natural conversation for an extended period of time;
  - Currently, the conversations are not very convincing
  - Helpful for relatively linear conversations on narrow topics
  - Routine components of a firm's customer-service phone calls.

NLP APPLICATIONS [1]

- We can classify NLP applications in terms of the difficulty in building them.
- Easiest to build: spell checkers, synonym suggesters and keyword-search querying tools
- Solved fairly straightforwardly with deterministic, rules-based code using, say, reference dictionaries or thesauruses.
- Deep learning models are not needed
- Intermediate-complexity NLP tasks
- School-grade reading level assignment to a document
- Most likely next word prediction while making a query in a search engine
- Document classification
- Information extraction from documents or websites (like prices or named entities)
- Well suited to be solved with deep learning models.

NLP APPLICATIONS [2]

- Most sophisticated NLP implementations are required for:
- machine translation
- automated question-answering
- chatbots.
- These are tricky because:
- handle ***application-critical nuance*** (example: transient humor)
- Response to a question can depend on the ***intermediate responses*** to previous questions
- Meaning can be conveyed over the course of a lengthy passage of text consisting of many sentences.
- Complex NLP tasks like these are use deep learning architectures



- 2011: Dahl et al first applied a deep learning algorithm to a large dataset – recognize a substantial vocabulary of words from audio recordings of human speeches
- 2012-2015: more focus on Machine Vision
- Deep learning based models approached the precision and accuracy of traditional machine learning models with
  - Less development time
  - Lower computational complexity
- Microsoft was able to integrate real-time machine translation software onto mobile phone processors

COMPUTATIONAL REPRESENTATION OF LANGUAGE

COMPUTATIONAL REPRESENTATION OF LANGUAGE

- To process language, it has to be modeled in an appropriate way.
- Most commonly used quantitative representation a two-dimensional matrix of numerical values.
- Two popular methods for converting text into numbers:
  - One-hot encoding
  - Word vectors

ONE-HOT REPRESENTATION OF WORDS

- Traditional Approach
- The words of natural language in a sentence ("the", "cat", "sat", etc.) are represented as columns of a matrix.
- Each row represents a unique word
- Cells – binary – indicate the position of a word (row) at a particular position (column) within the corpus
- Simplicity and sparsity – limiting factors

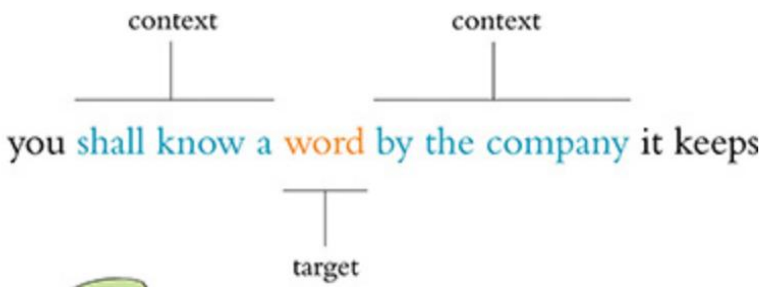
The bat sat on the cat.

words						
the	1	0	0	0	1	0
bat	0	1	0	0	0	0
on	0	0	0	1	0	0
⋮						
n_{unique_words}						

WORD VECTORS [1]

- Vector representation of words are the information-dense alternative to one-hot encoding of words
- One-hot encoding – location of words
- Word vectors – information about word meaning and location
- Word embeddings or vector-space embeddings
- Enable NLP models to automatically learn linguistic features
- Assign each word within a corpus to a particular, meaningful location within a multidimensional space called the vector space
  - Assign a word to a random location
  - Consider the words that tend to be used around a given word
  - Shift the word to locations that represent its meaning

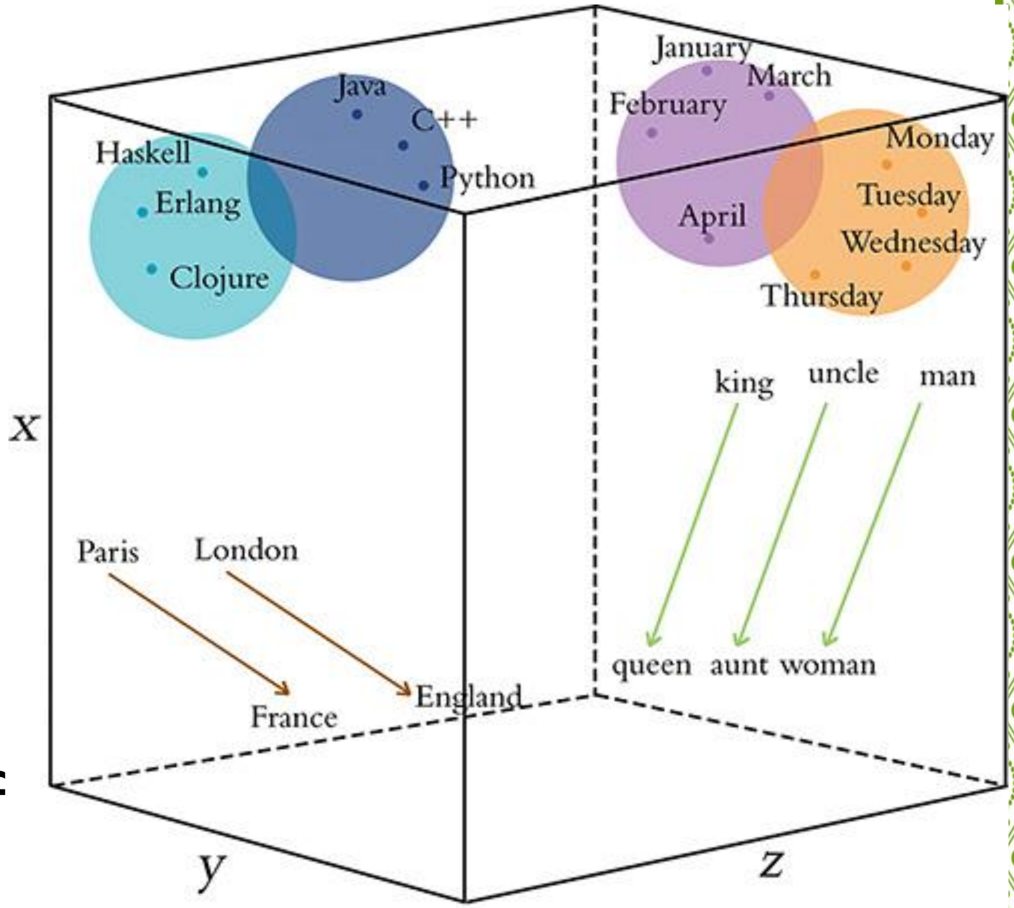
WORD VECTORS [2]



- Start with the first word in the corpus
- Consider each word as the "target" word
- Move to the right one word at a time until the final word is reached
- Consider the target word relative to the words around it – its context words
  - Context-word window size – 3 words
  - Word – (shall, know, a), (by, the, company)
- As you consider the next word, the context also shifts along with it
- **Word2vec** and **GloVe** are the two most popular techniques for converting natural language to word vectors
- Considering a **target** word, accurately **predict** the target word, given its **context** words
- With a huge corpus, words that tend to appear in similar contexts are assigned to similar locations in vector space.

VECTOR SPACE

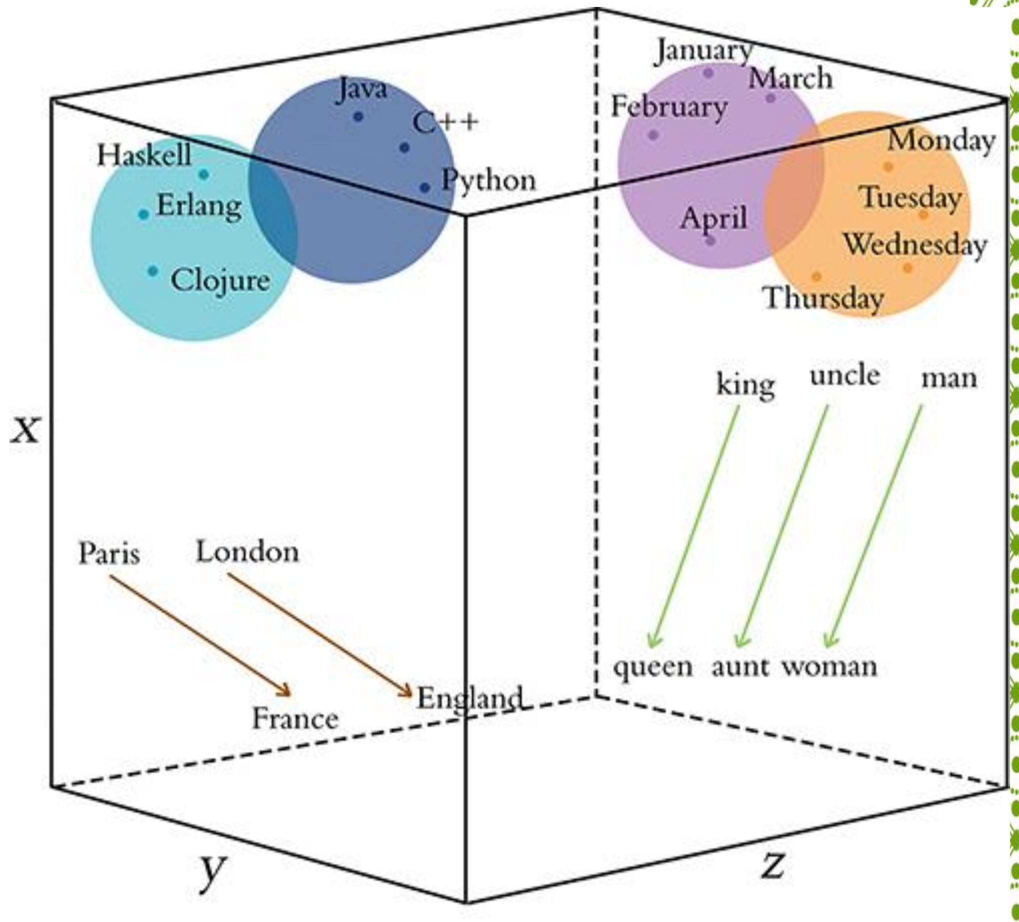
- There may be 100s of dimensions.
- Each word (say, king) is specified by a vector V_{king} that consists of these 100 components (number)
- Distance between two words gives their similarity
- Closer two words are within a vector space, the closer their meaning, as determined by the similarity of the context words
  - Synonyms and commonly mis-spellings of words
  - Words used in similar contexts – time
  - Words that convey a specific meaning
    - Created, Developed, Built



n - dimensional space

WORD-VECTOR ARITHMETIC

- Particular movements across vector space are an efficient way for relevant word information to be stored in the vector space
  - These movements then represent relative particular meanings between words
  - Countries and Capitals
- Word-vector arithmetic is the by-product of being able to trace the vectors of meaning from one word to another in the vector space
- $V_{king} = (-0.9, 1.9, 2.2)$; $V_{man} = (-1.1, 2.4, 3.0)$; $V_{woman} = (-3.2, 2.5, 2.6)$



n – dimensional space

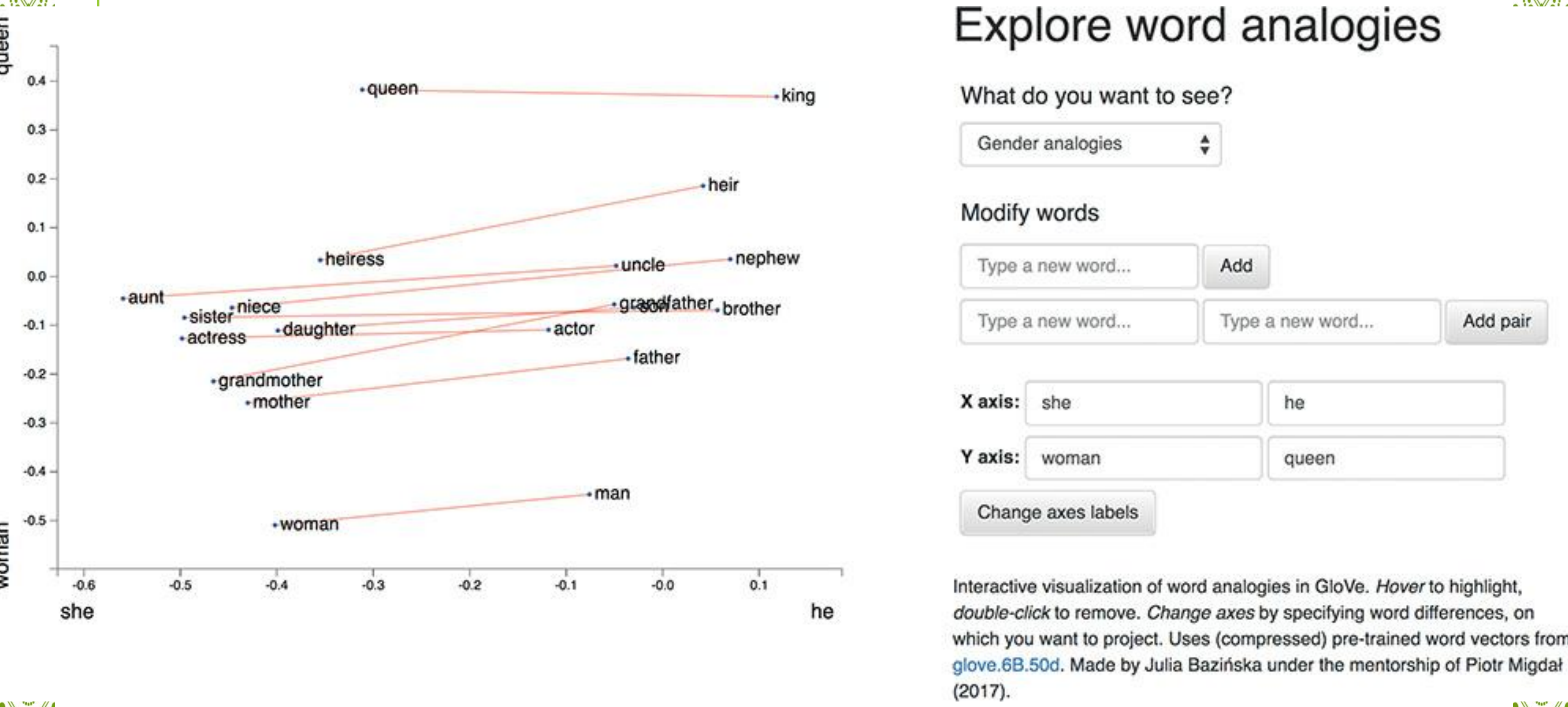
$V_{king} - V_{man} + V_{woman} = V_{queen}$

$V_{bezos} - V_{amazon} + V_{tesla} = V_{musk}$

$V_{windows} - V_{microsoft} + V_{google} = V_{android}$

$$
\begin{aligned}
x_{queen} &= x_{king} - x_{man} + x_{woman} &= -0.9 + 1.1 - 3.2 = -3.0 \\
y_{queen} &= y_{king} - y_{man} + y_{woman} &= 1.9 - 2.4 + 2.5 = 2.0 \\
z_{queen} &= z_{king} - z_{man} + z_{woman} &= 2.2 - 3.0 + 2.6 = 1.8
\end{aligned}
$$

URL: BIT.LY/WORD2VIZ OR HTTPS://LAMYIOWCE.GITHUB.IO/WORD2VIZ/



LOCALIST VERSUS DISTRIBUTED REPRESENTATIONS [1]

- Word vectors store the meaning of words in a ***distributed*** representation across *n*-dimensional space.
  - with word vectors, word meaning is distributed gradually as we move from location to location through vector space.
- One-hot representations are *localist*.
  - Information on a given word is stored discretely, within a single row of a typically extremely sparse matrix.
- Nuance
  - One-hot representations lack nuance as they are simple binary flags.
  - Vector-based representations are extremely nuanced as information about words is distributed throughout a continuous, quantitative space.
  - Infinite possibilities for capturing the relationships between words.

LOCALIST VERSUS DISTRIBUTED REPRESENTATIONS [2]

- Labour-intensive representations
  - In practice, the use of one-hot representations often requires labor-intensive, manually curated taxonomies.
    - dictionaries and other specialized reference language databases.
  - External references are unnecessary for vector-based representations, which are fully automatic with natural language data alone.

LOCALIST VERSUS DISTRIBUTED REPRESENTATIONS [3]

Handling new words

- One-hot representations don't handle new words well.
  - A newly introduced word requires a new row in the matrix
  - reanalysis relative to the existing rows of the corpus
  - Code changes—perhaps via reference to external information sources.
- With vector-based representations
  - New words can be incorporated by training the vector space on natural language that includes examples of the new words in their natural context.
  - Each new word gets its own new *n*-dimensional vector.
  - Initial inaccuracy with positioning of new word in n-dimensional space
    - Lack of training examples
  - Positions of all existing words remain the same and the model will not fail to function.
  - Over time, with additional training instances, the accuracy of the vector-space coordinates of the new word improves

LOCALIST VERSUS DISTRIBUTED REPRESENTATIONS [4]

- Interpretation of meaning
  - The use of one-hot representations often involves ***subjective interpretations*** of the meaning of language.
    - Often require coded rules or reference databases
    - These are designed by (relatively small groups of) developers
  - The meaning of language in vector-based representations is data driven from the corpus of text.
- Word Similarity
  - One-hot representations natively ignore word similarity
    - For example, similar words, such as couch and sofa, are represented no differently than unrelated words, such as couch and cat.
  - Vector-based representations innately handle word similarity
    - The more similar two words are, the closer they are in vector space.

INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS

Artificial Neurons

Threshold Units

Gradient Descent

Multilayer Networks

Back Propagation Algorithm

Hidden Layer Representations

Examples

CONNECTIONIST MODELS

Consider humans:

- Neuron switching time ~ .001 second
- Number of neurons ~ 10^{10}
- Connections per neuron ~ 10^{4-5}
- Scene recognition time ~ .1 second
- 100 inference steps doesn't seem like enough

$\rightarrow$ much parallel computation

Properties of artificial neural nets (ANN's):

- Many neuron-like threshold switching units
- Many weighted interconnections among units
- Highly parallel, distributed process
- Emphasis on tuning weights automatically

WHEN TO CONSIDER NEURAL NETWORKS

Input is high-dimensional discrete or real-valued

Output is discrete or real-valued

Output is a vector of values

Possibly noisy data

Form of target function is unknown

Human readability of output is unimportant

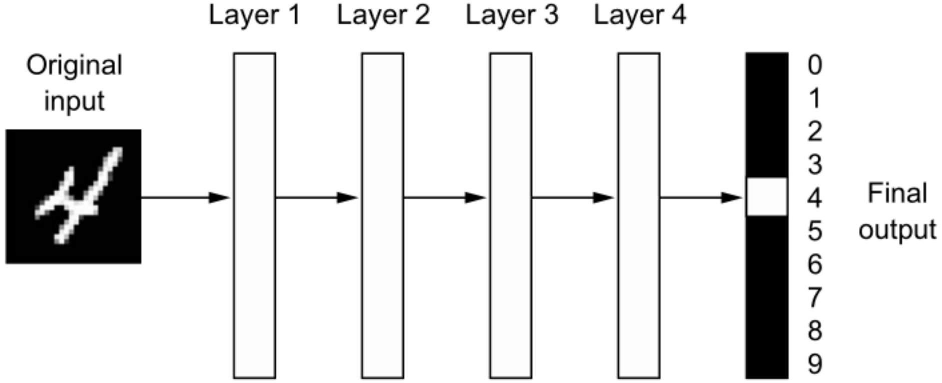
Examples:

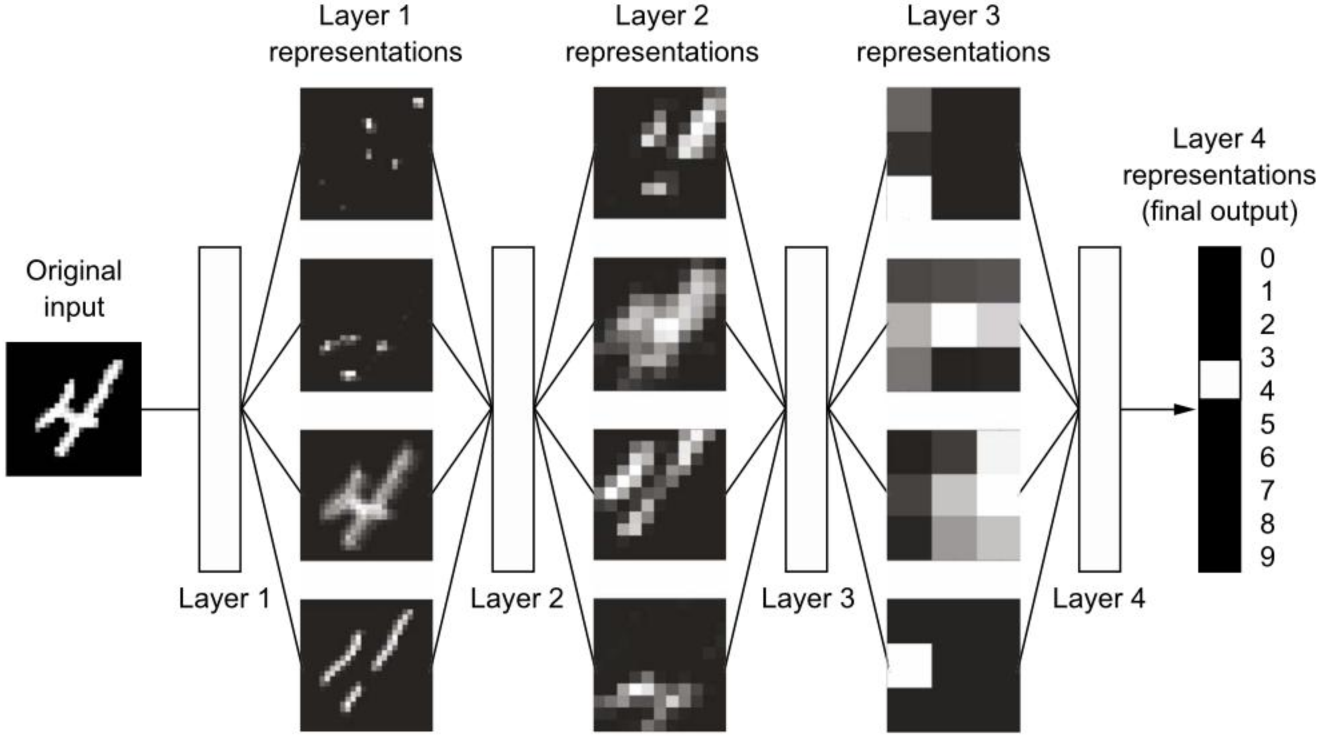
Speech Recognition

Image Classification

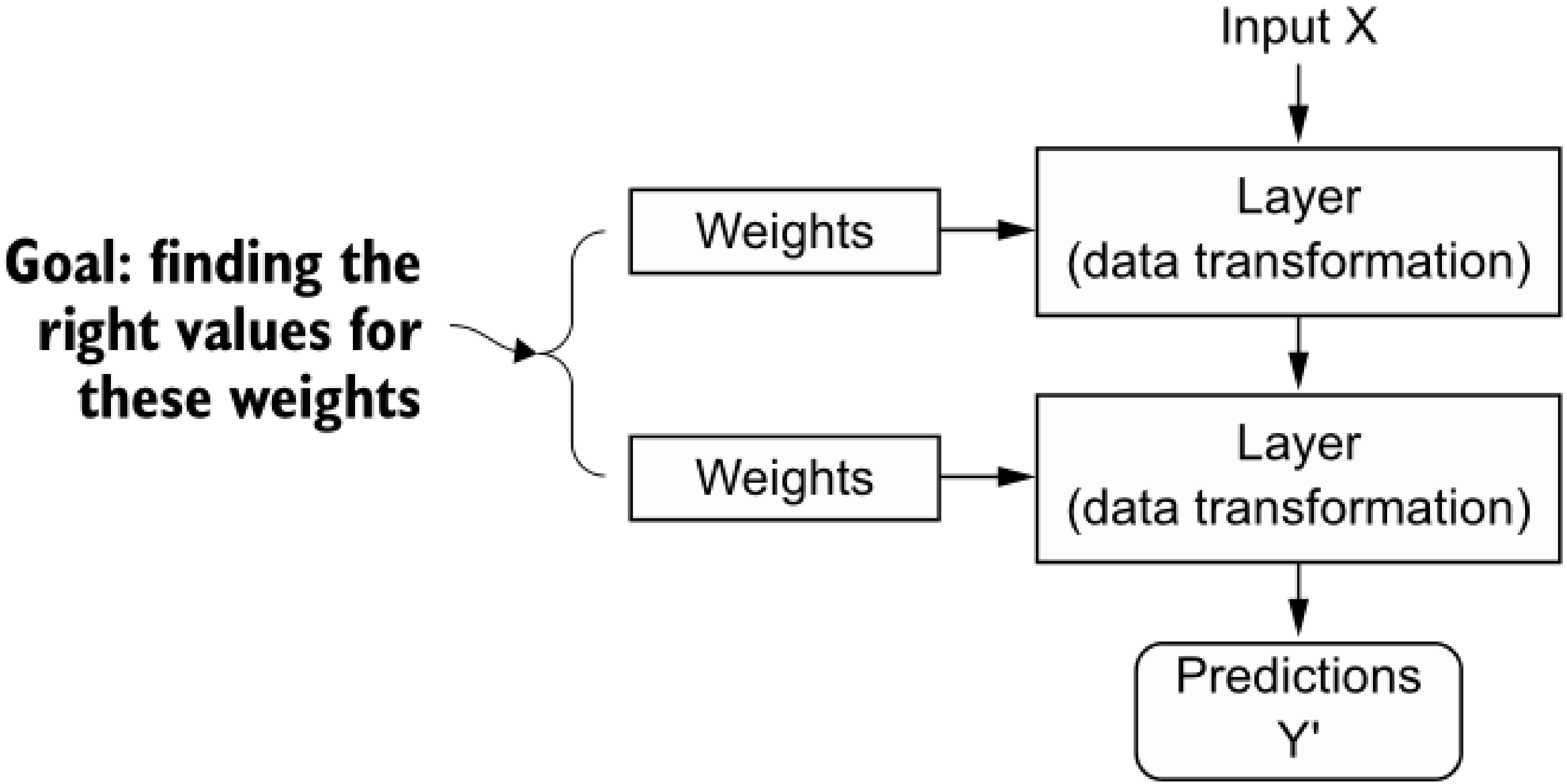
Financial Prediction

A DEEP NEURAL NETWORK FOR DIGIT CLASSIFICATION

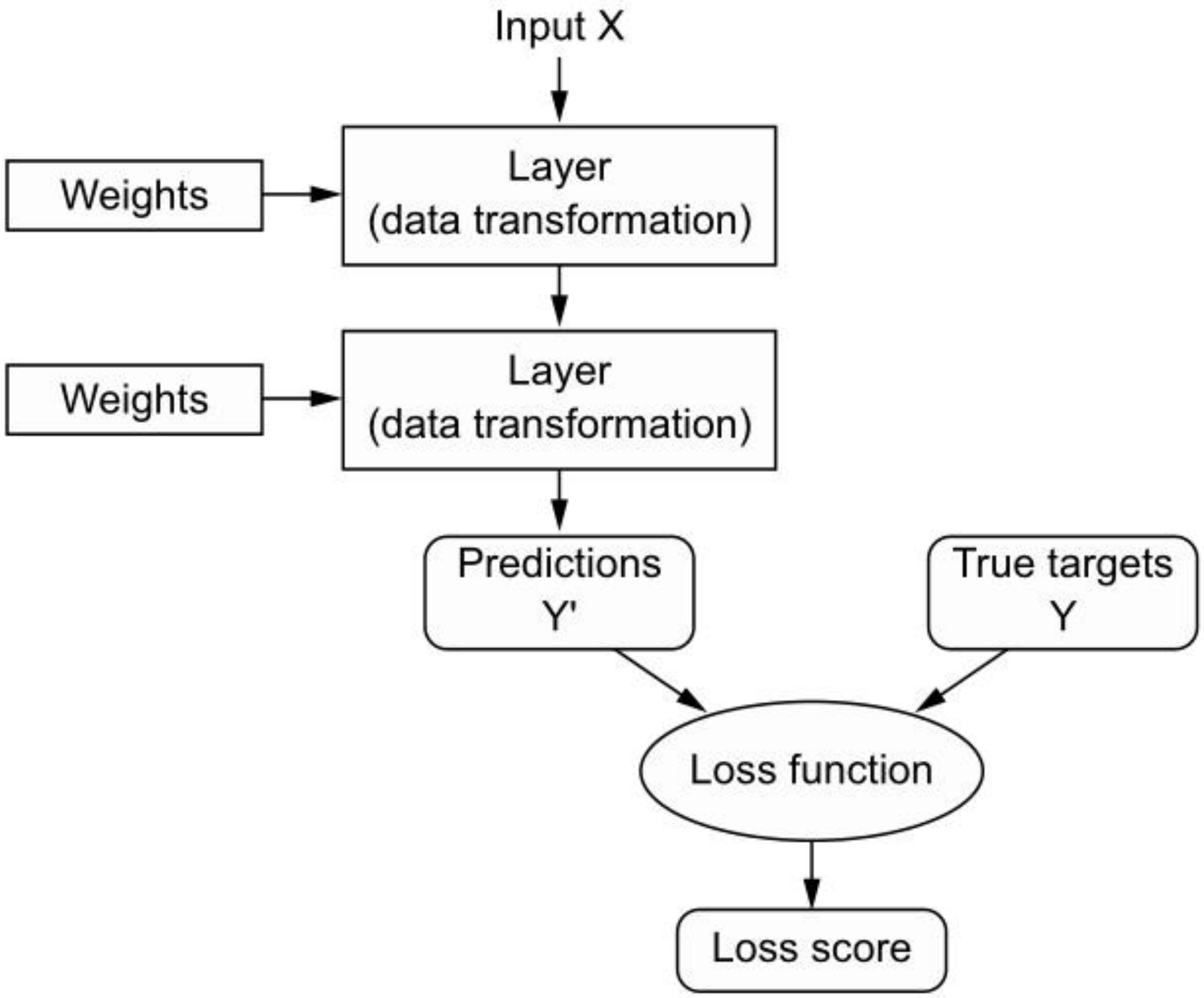




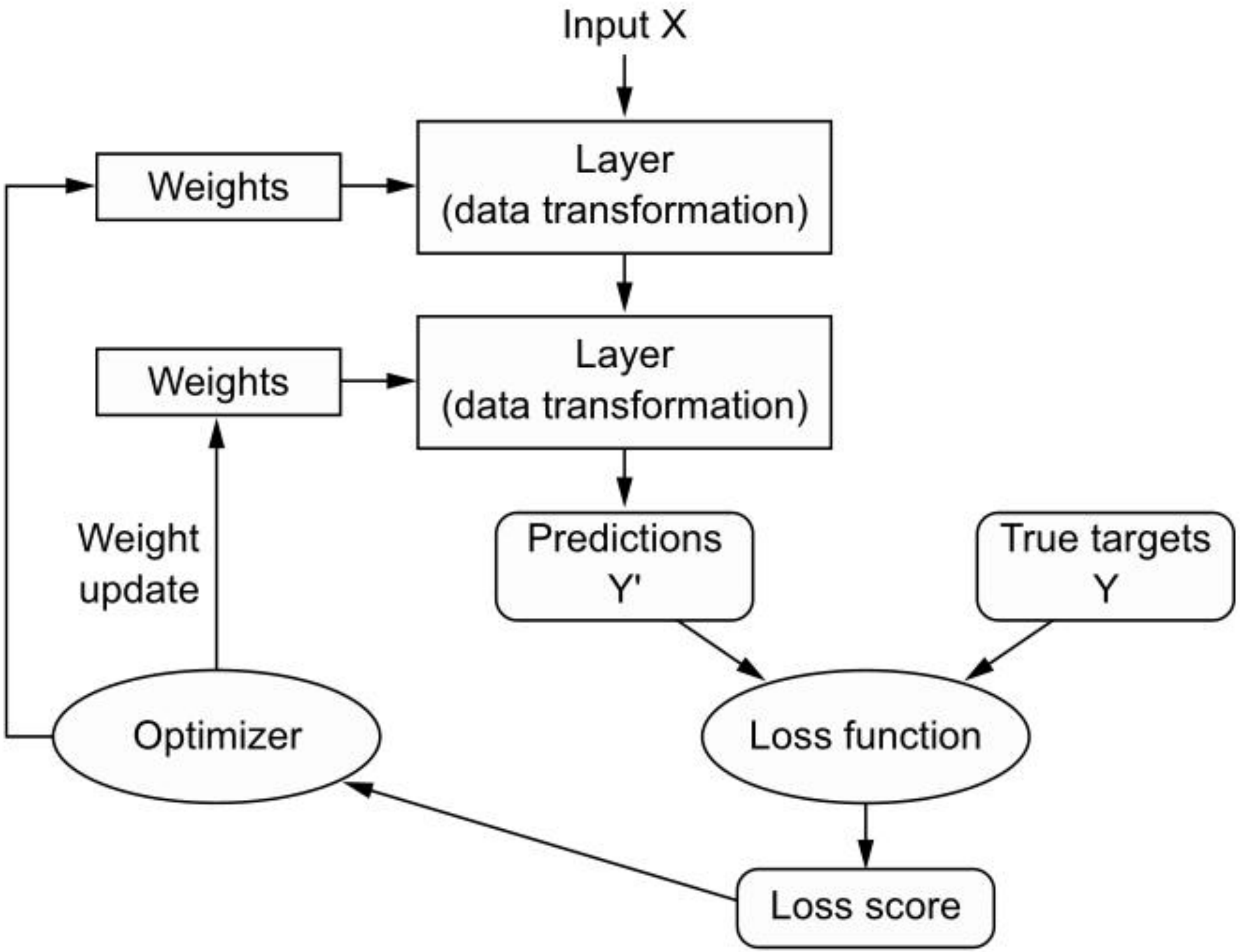
A NEURAL NETWORK IS PARAMETERIZED BY ITS WEIGHTS



A LOSS FUNCTION MEASURES THE QUALITY OF THE NETWORK'S OUTPUT.

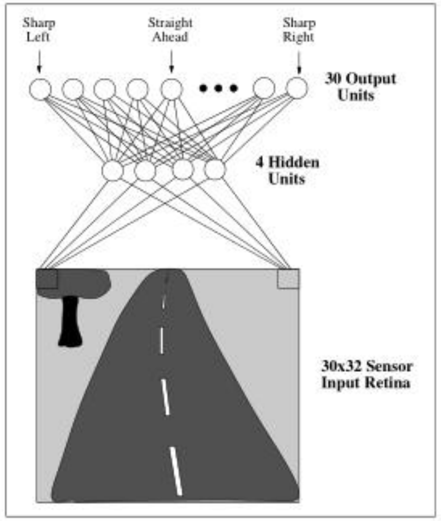


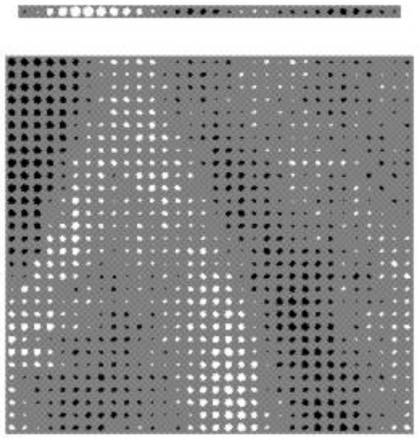
THE LOSS SCORE IS USED AS A FEEDBACK SIGNAL TO ADJUST THE WEIGHTS.



- ALVINN (Pomerleau, 1993) uses **backpropagation** to learn to steer an autonomous vehicle on highways at speeds upto 70 mph. Image of a forward mapped camera
- Mapped to 960 neural network inputs
  - 30 x 32 pixels from the image
- Fed forward to 4 hidden units
- Connected to 30 output units
  - Network outputs encode the commanded steering direction
- Right Side Figure
  - Weight values for one of the hidden units in the network
  - 30 x 32 weights in the hidden unit
  - Positive weights – white
  - Negative weights – black
- Top Strip Figure in the Right
  - Weights from hidden units to the 30 output units
- Activation of this particular hidden unit encourages a turn to the left

ALVINN drives 70 mph on highways





THANKS