Introduction to Deep Learning

Lecture 1

Subir Varma

Office Hours

- Lectures: Mondays and Wednesdays: 7:35-9:10 PM

- Office: Lucas 221S
- Office Hours: On Demand, Please Email or Text to setup Time
  Tuesday and Thursday Afternoons work best for me

- Contact Information: svarma2@scu.edu
- Phone: (408) 420 1518

Books for the Course

- Main Text Books:
  - "Introduction to Deep Learning" by Das and Varma
    https://srdas.github.io/DLBook2/
  - "Deep Learning with Python, Second Edition" by Francois Chollet

- Supplementary Reading:
  - "Deep Learning" by Goodfellow, Bengio and Courville
    http://www.deeplearningbook.org/

Pre-Requisites

- Knowledge of:
  - Introductory Machine Learning
  - Multi-Variable Calculus (mostly Partial Differentiation)
  - Python (NumPy) Programming
- Covered in Lecture 2:
  - Basic Probability Theory
  - Basic Linear Algebra (Matrix Multiplication) and Tensor Algebra

Software Packages

- Keras: keras.io
- Tensor Flow: https://www.tensorflow.org
- Anaconda (Scientific Python Distribution): https://www.tensorflow.org/install/install_mac#installing_with_anaconda
- Google Colab: Run Jupyter Notebooks on the cloud, has access to fast GPUs and TPUs

- Python Numpy Tutorials :

  https://sites.engineering.ucsb.edu/~shell/che210d/numpy.pdf
  http://cs231n.github.io/python-numpy-tutorial/

  Please Review these Tutorials

Homeworks, Exams etc.

The course grade will be distributed as follows:

- Homework: 30%

  Group Assignments: Please form groups of 2
- Mid-Term Exam: 40%
- Course Project: 30%

  Project Groups of 2

What is Deep Learning?

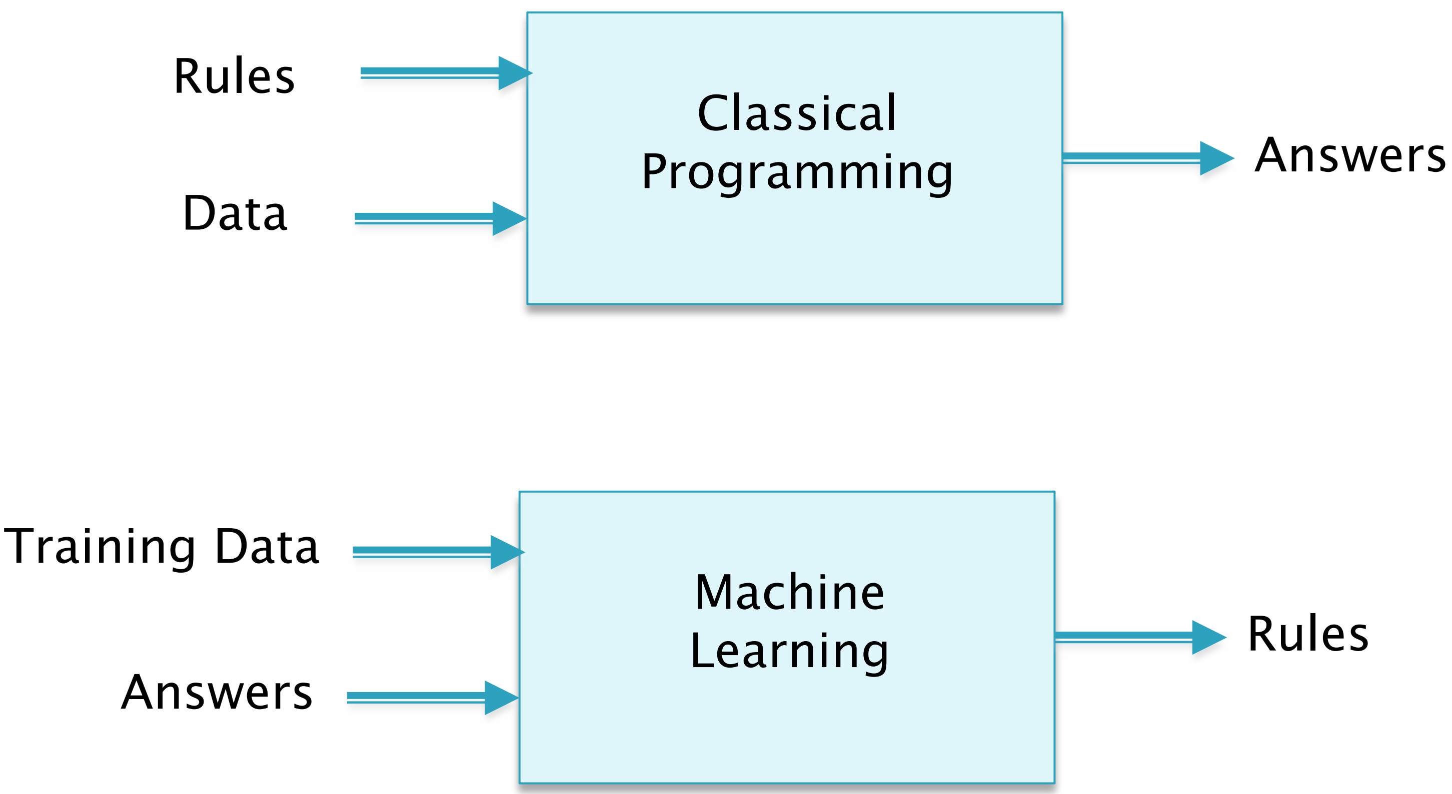
What is Deep Learning?

- An important subset of the field of Machine Learning
- What is Machine Learning?
  - The science of designing systems that can learn from experience
  - Instead of explicitly programming a task, can the computer learn the rules by looking at data?

  - Use a portion of the data (experience) to build a model (also called training)
  - Once trained, the system is able to work effectively even for input data that are not part of the training set

A way to solve complex problems by using models that can be learnt from data

Machine Learning



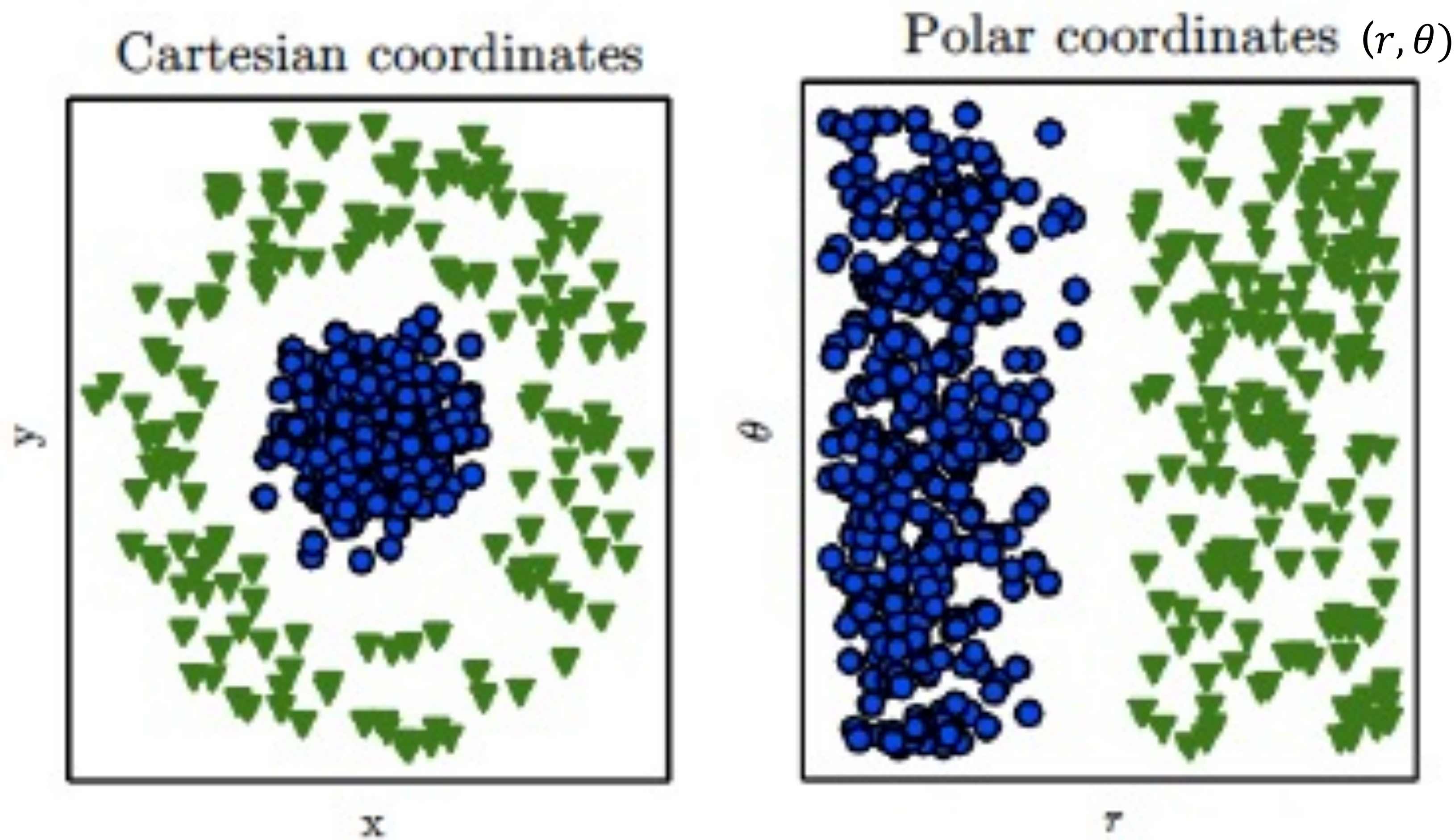
Difference between ML and DL: Data Representations

- How is DL different from ML?
  - ML requires that we supply the model with a good representation for the data
  - DL creates higher level representations of data as part of the learning process

- Data can be represented in different ways, and this has an enormous influence on the performance of ML/DL algorithms.
  Example: Roman Numerals vs Arabic Numerals

- We would like to map the raw data into some other representation in a way that makes the relationships between different things more explicit

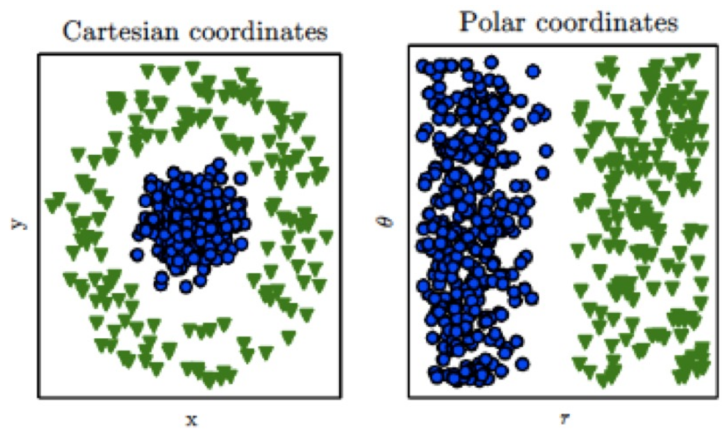
The Importance of Representations



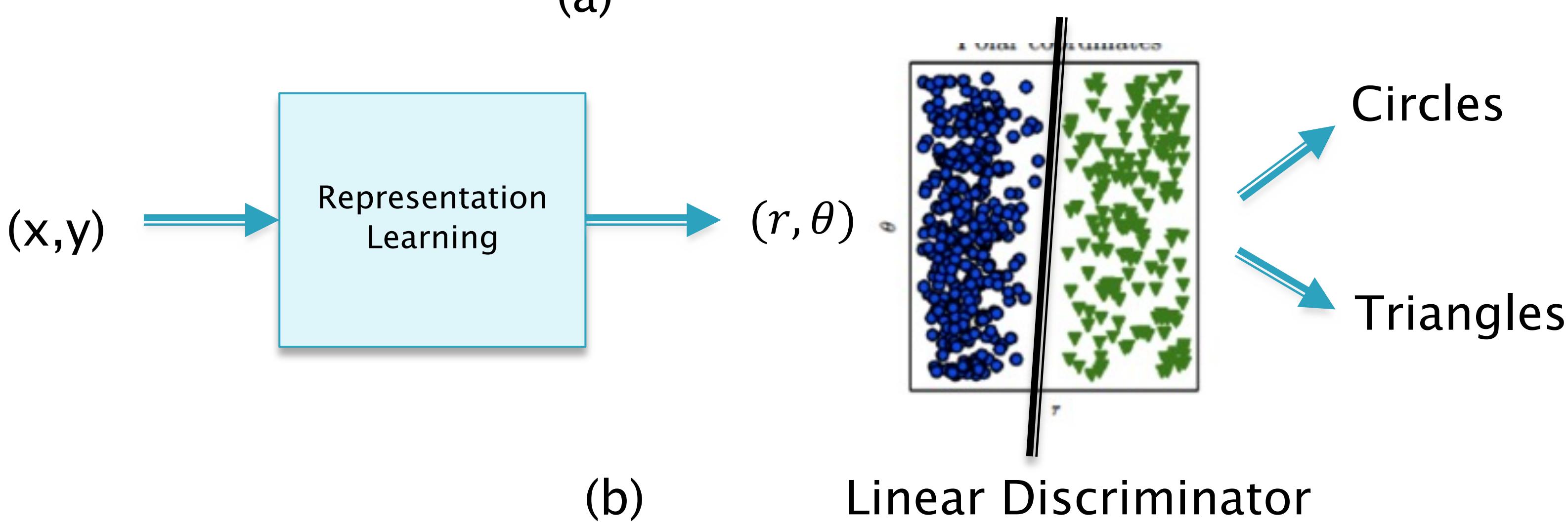
Representation Learning

- Simple Machine Learning is good at doing Linear Discrimination
- Before the advent of Deep Learning,
  - Choosing a data representation appropriate for the problem, which could then be fed into a simple Machine Learning system, was a manual time consuming process
  - With many problems it was difficult to know what features should be extracted
- With Deep Learning:
  - The system discovers the best representation itself, which can then be fed into a Linear Discriminator – This is called Representation Learning
  - Leads to better performance compared to hand design representations, and allows the system to adapt to newer tasks with minimal human intervention.

Classification using DL



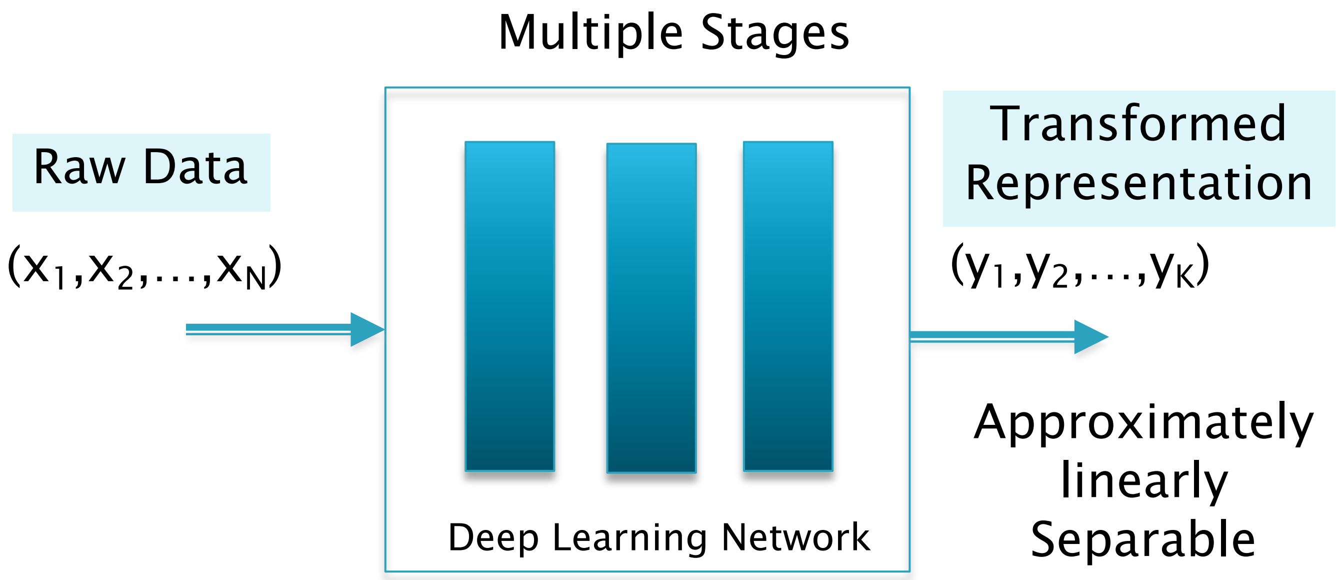
(a)



(b)

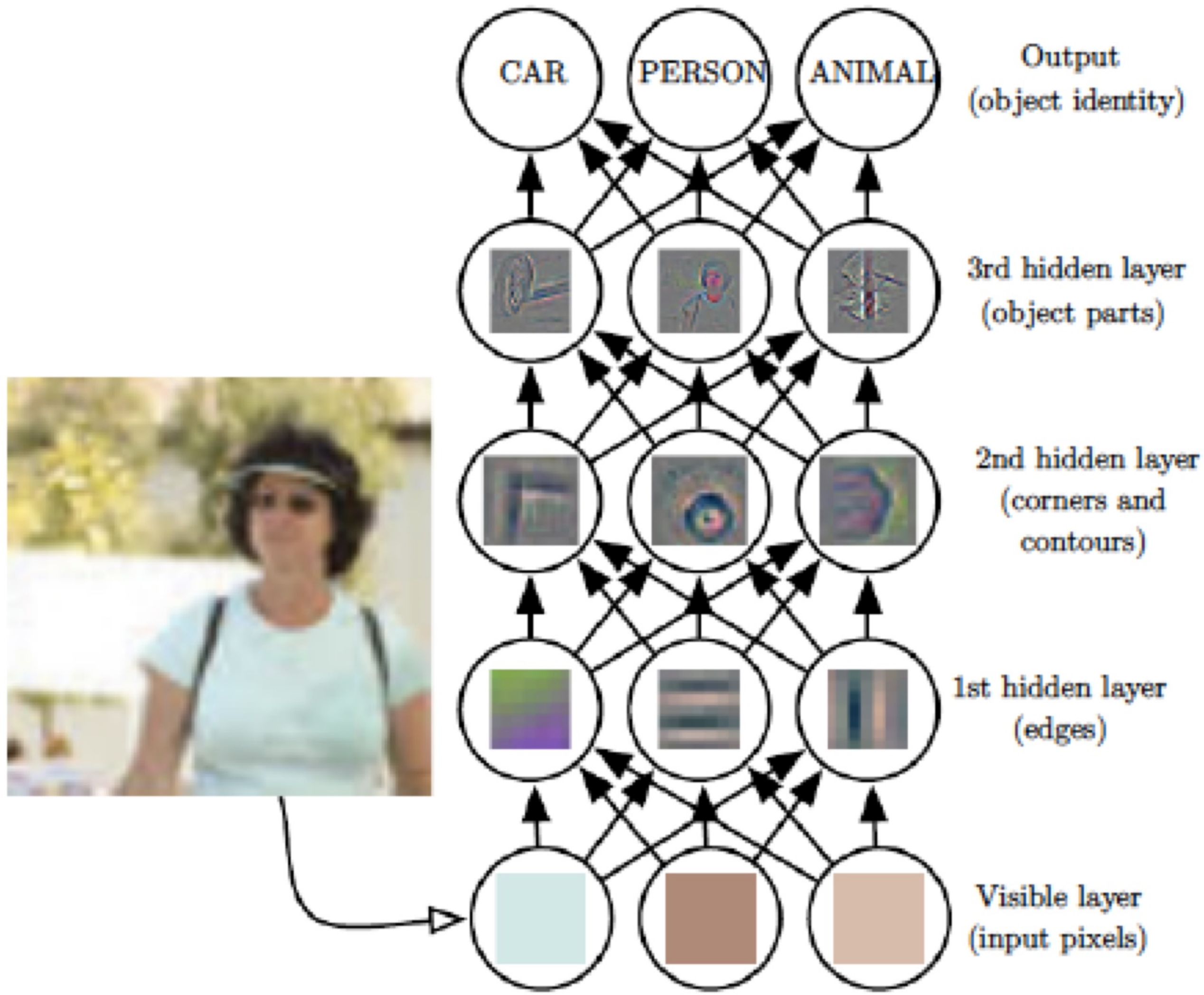
Look up Chollet, Chapter 1, Fig 1.4 for another example

The "Deep" in Deep Learning



The more "mixed" up the data is, the more stages required to "separate" it

How Deep Learning Creates Image Representations



How Does DL Learn Representations?

Deep Learning solves the problem of Representation Learning by using

Using Multiple Nodes per Layer

1. <u>Compositions</u>
   - Process of assembling a more complex representation from simpler object representations
2. <u>Hierarchies</u>
   - Process of building higher level representations by combining simpler ones

Using Multiple Layers

Image Representations
Word Representations

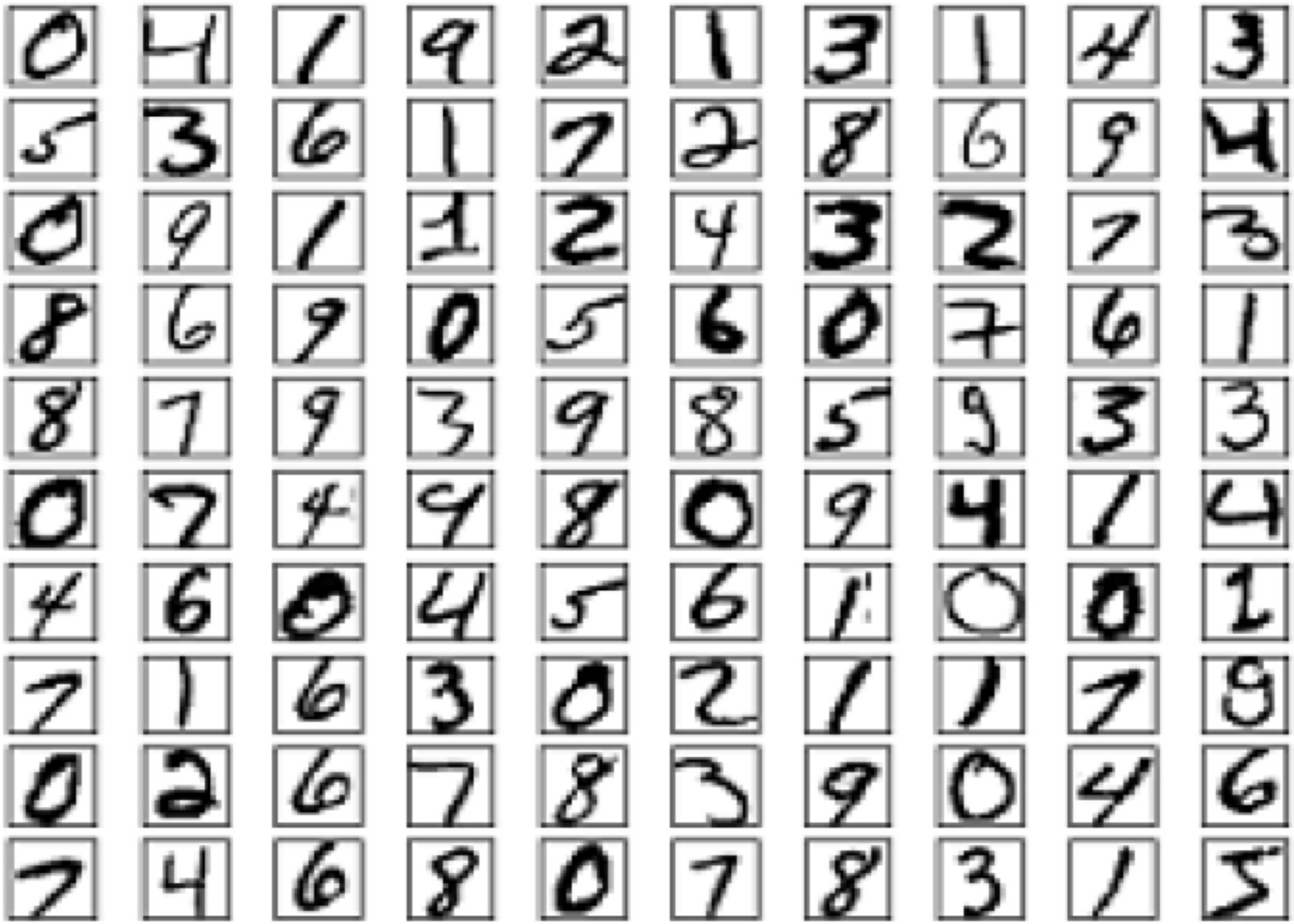
Image Representations

- Deep Learning: Image represented as the output of a Neural Network
  - Enables us to do operations that require a deeper (semantic) understanding of the image, such as:
    - Detect the main objects in the image and classify them
    - Provide a verbal description of the image
    - Generate similar images

Previously

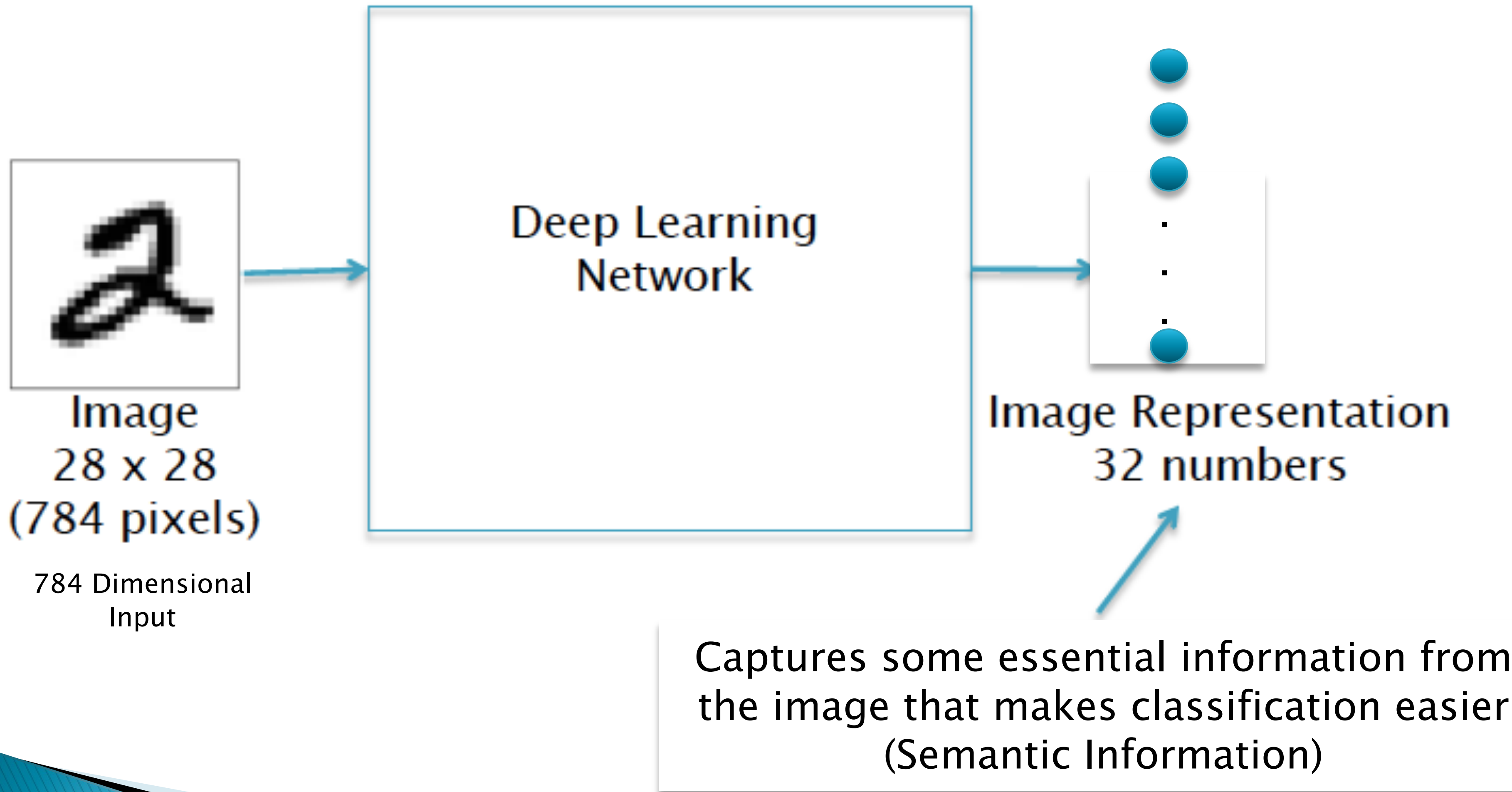
- Image represented as chemicals on a photographic film:
  - Good for certain operations, such as film development; difficult to manipulate or transmit image
- Image represented as digital bits
  - Makes possible all kinds of image manipulations, compression and enables easy image transmissions

The MNIST Handwritten Digit Data Set



Classification

DL based Image Representations



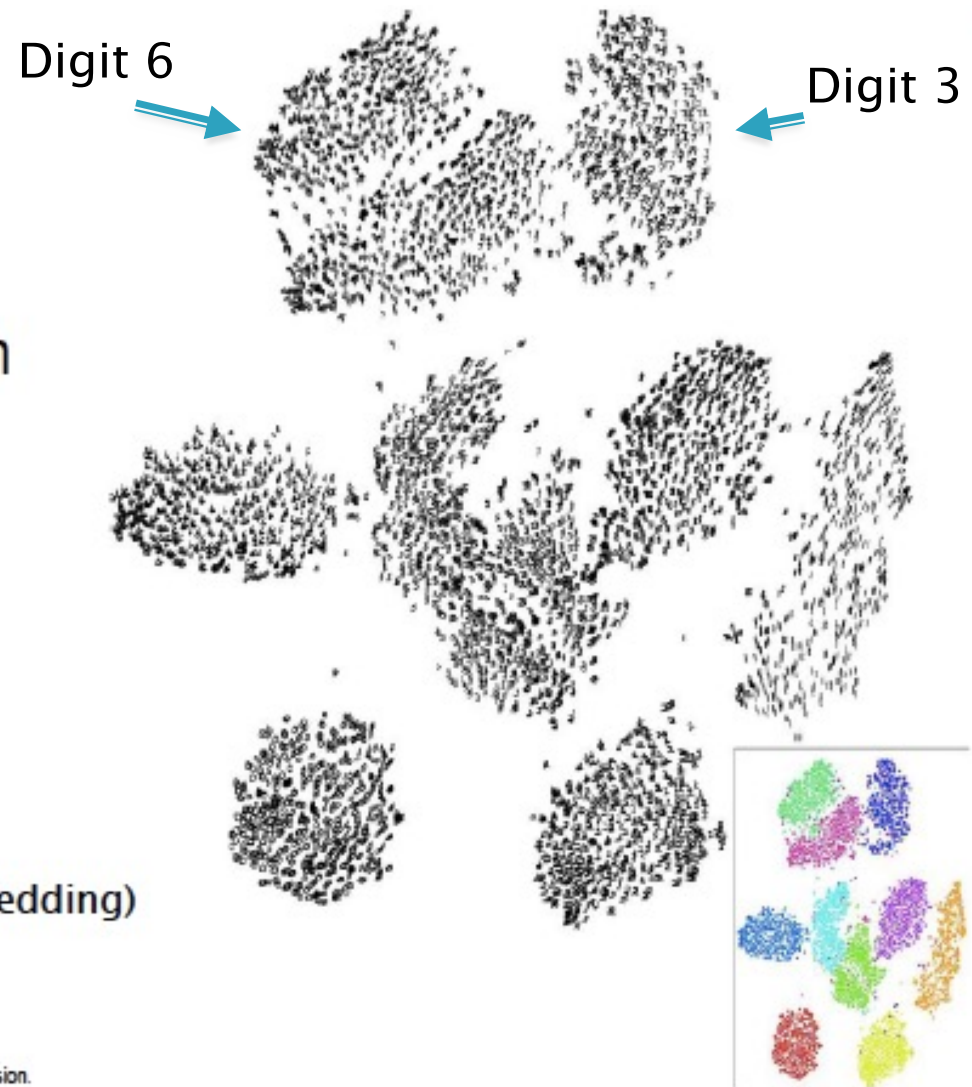
Visualizing the Representation

Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 32 to 2 dimensions

Simple algorithm: Principle Component Analysis (PCA)

More complex: **t-SNE**

T-Distributed Stochastic Neighbor Embedding)



Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008

Image Representations

Image
224x224x3 pixels
(150K)

From the
Imagenet Database

Deep Learning
Network

Image Representation
4096 numbers

Captures Semantic Information

Clustering of Similar Images

Projection of
4096 dimensions
into 2 dimensions

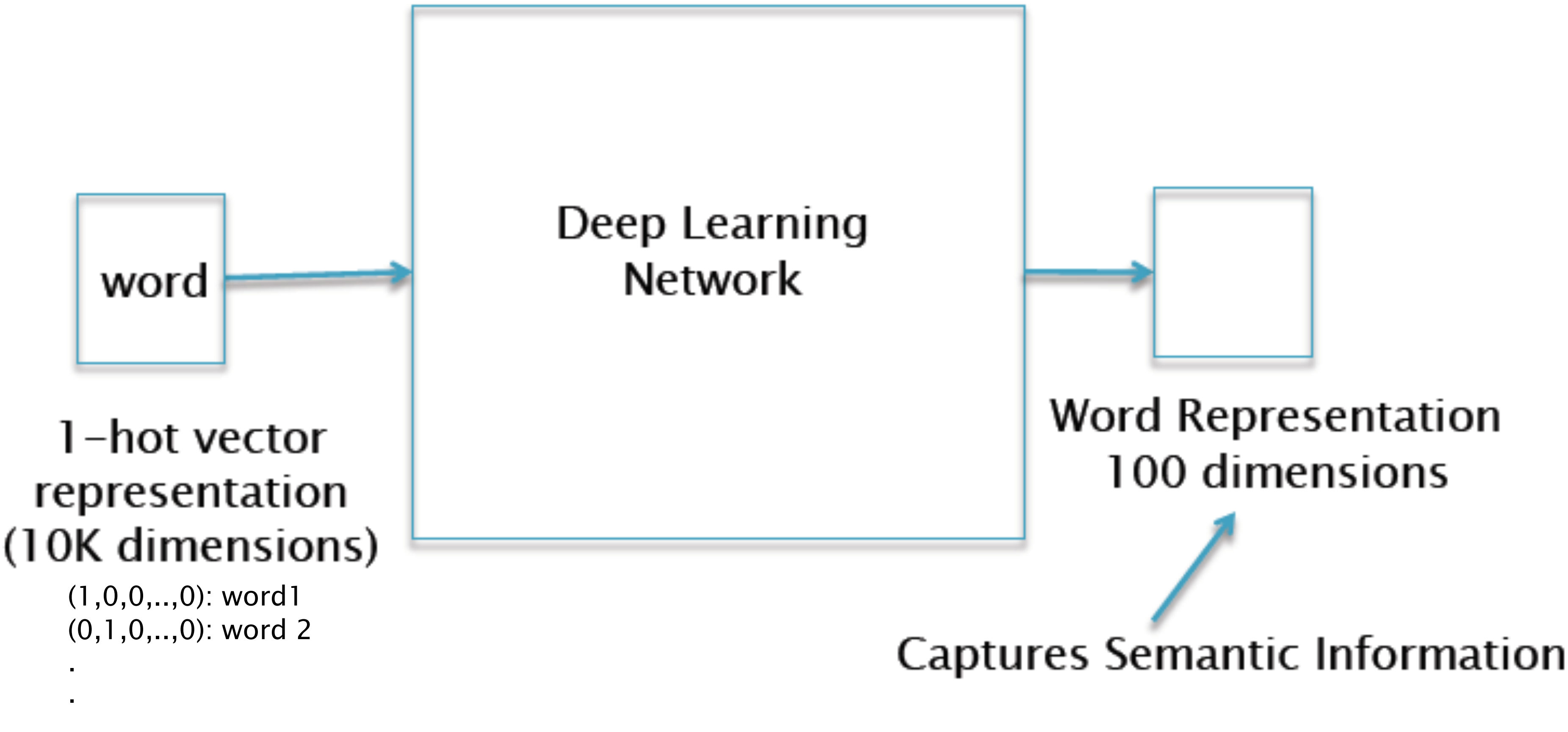
Flowers

Dogs

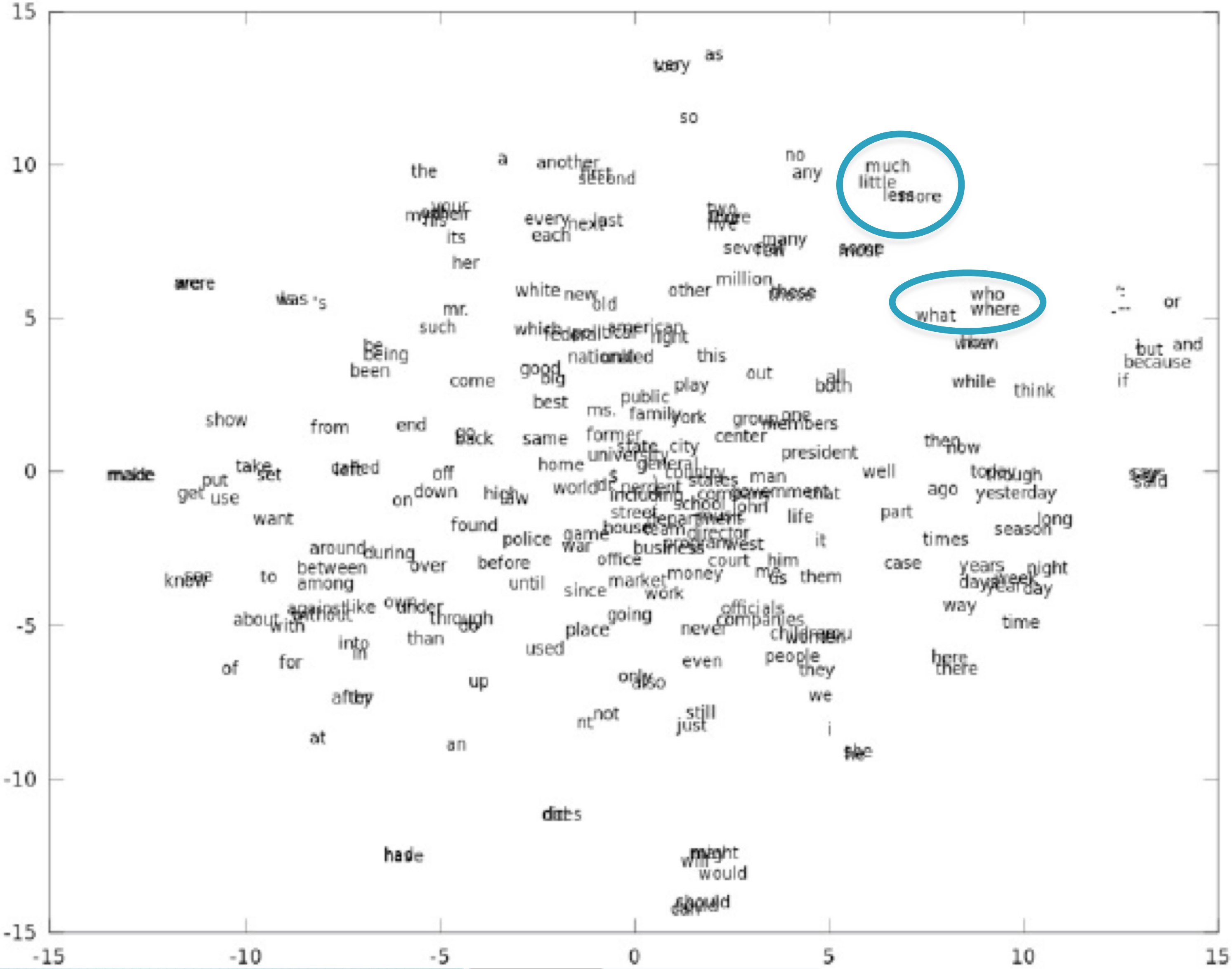
Representations in Natural Language Processing

- In traditional Computer Science, words/documents are represented using data structures such as arrays, dictionaries etc.
- These representations are good enough to answer questions such as the number of times a particular word occurs in the document
- But what about Higher Level semantic queries, such as:
  - Translate this book into German
  - Did the reviewer like this book
  - Text Summarization
  - Text Classification

Word Representations



Representing Words as Vectors



What Deep Learning Has Achieved

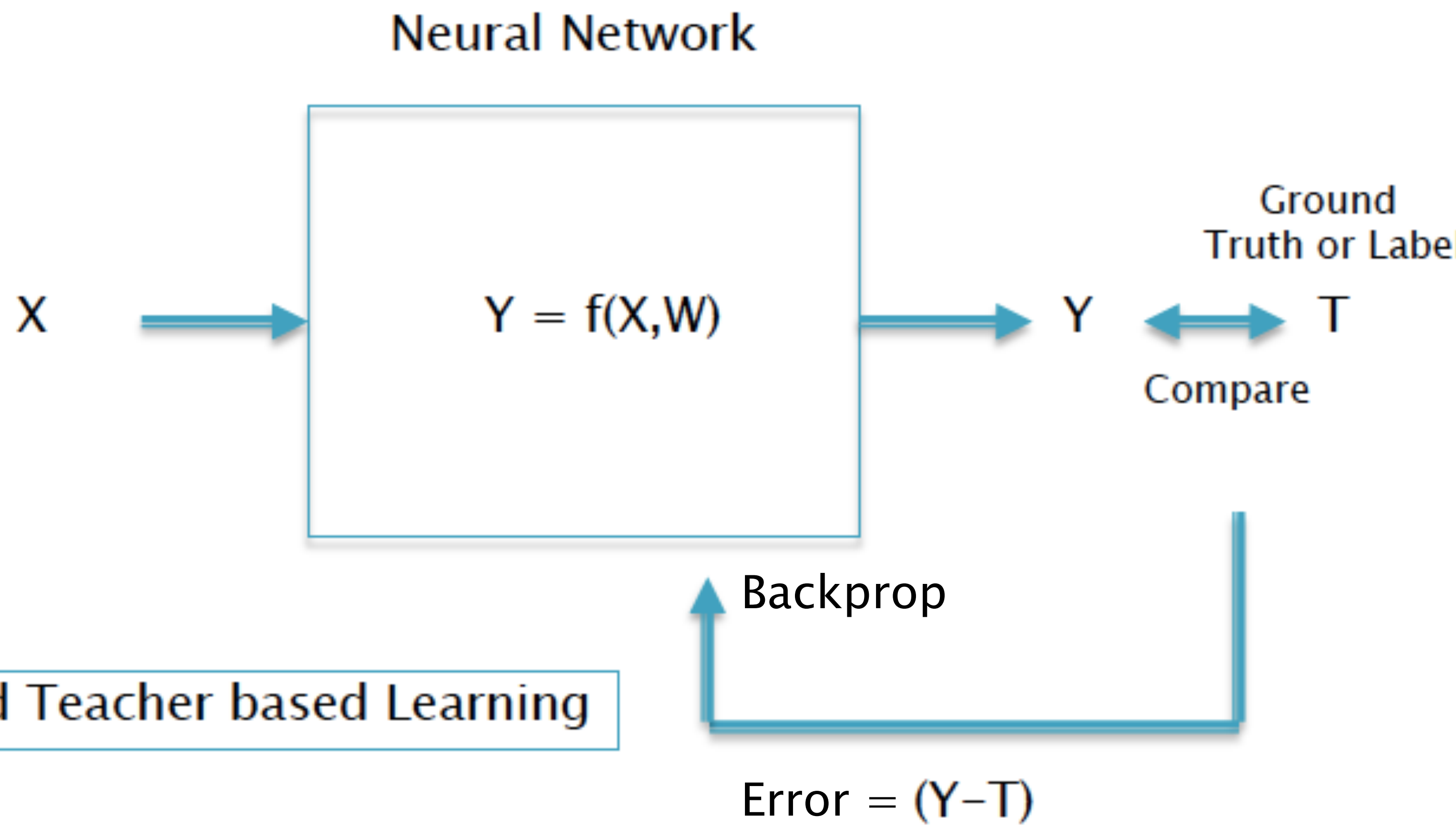
- Near-human-level image classification
- Near-human-level speech recognition
- Near-human-level handwriting transcription
- Improved machine translation
- Improved text-to-speech conversion
- Digital assistants such as Google Now and Amazon Alexa
- Improved Ad targeting, as used by Google, FB etc
- Improved search results on the web
- Ability to answer natural language questions
- Superhuman Game Playing (Go, Chess etc)
- Image and Text Generation

Types of Deep Learning

- Supervised Learning: Learn from Labeled Examples of the Correct Output
    - Self-Supervised Learning: Labels are automatically generated from the data
- Unsupervised Learning: There are no labeled examples – Look for interesting patterns, find representations
- Reinforcement Learning: Instead of being told the correct output, the system is given rewards instead

Supervised Learning

Supervised Learning



Also called Teacher based Learning

Choose parameters W to
Minimize Difference with Label

Examples of Supervised Learning

1. Image Processing
2. Natural Language Processing

Image Processing

Image Classification: The MNIST Dataset

10 Classes

70K Labeled images

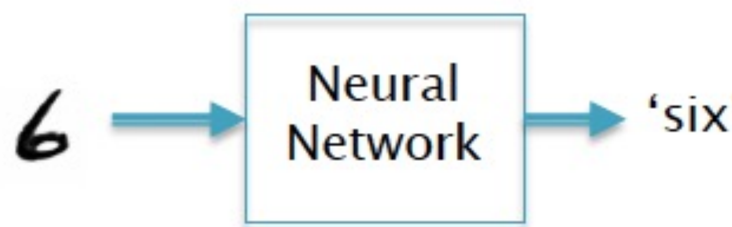
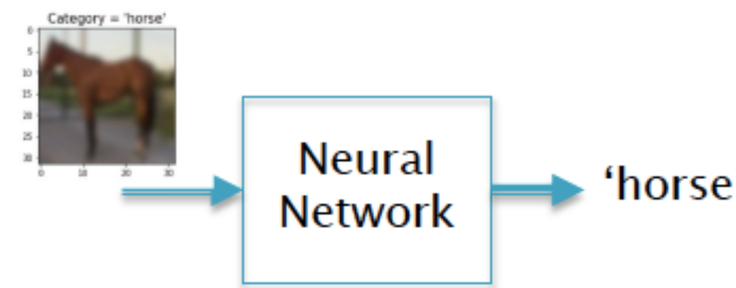


Image Classification: CIFAR-10 Image Dataset

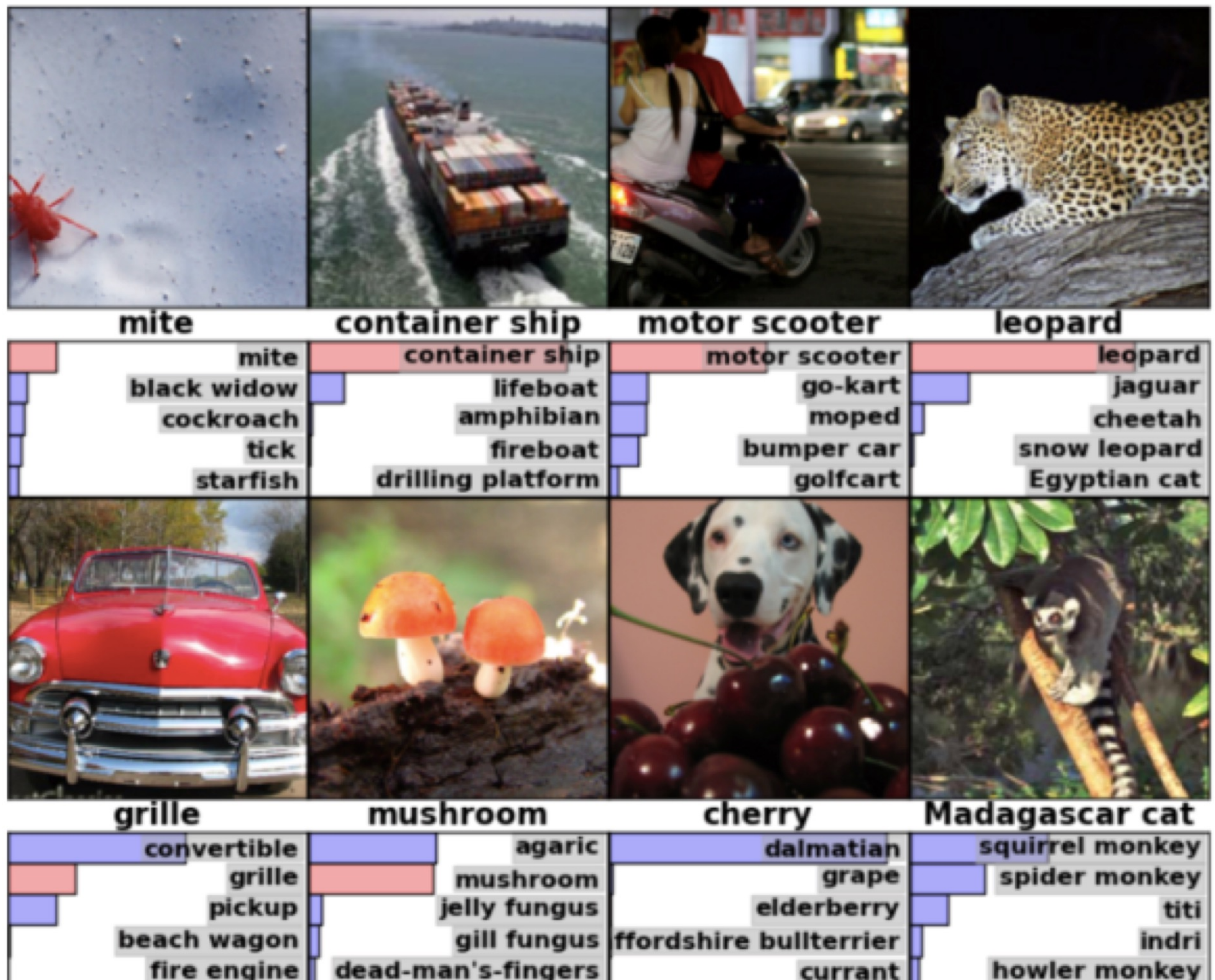
10 classes
50,000 training images
10,000 testing images



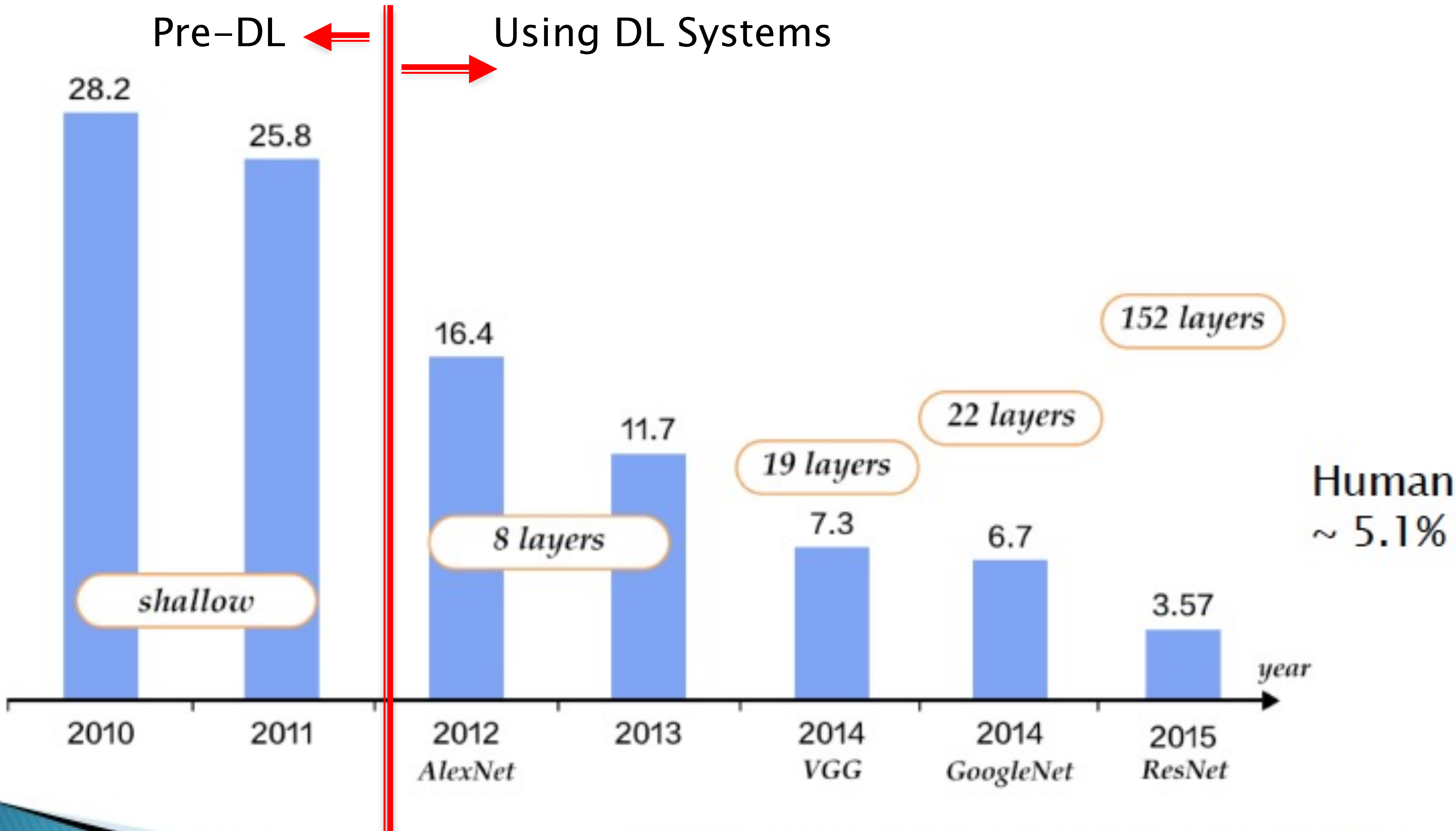
Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.

Image Classification: ImageNet Dataset

-1.4M Images
-1000 Categories

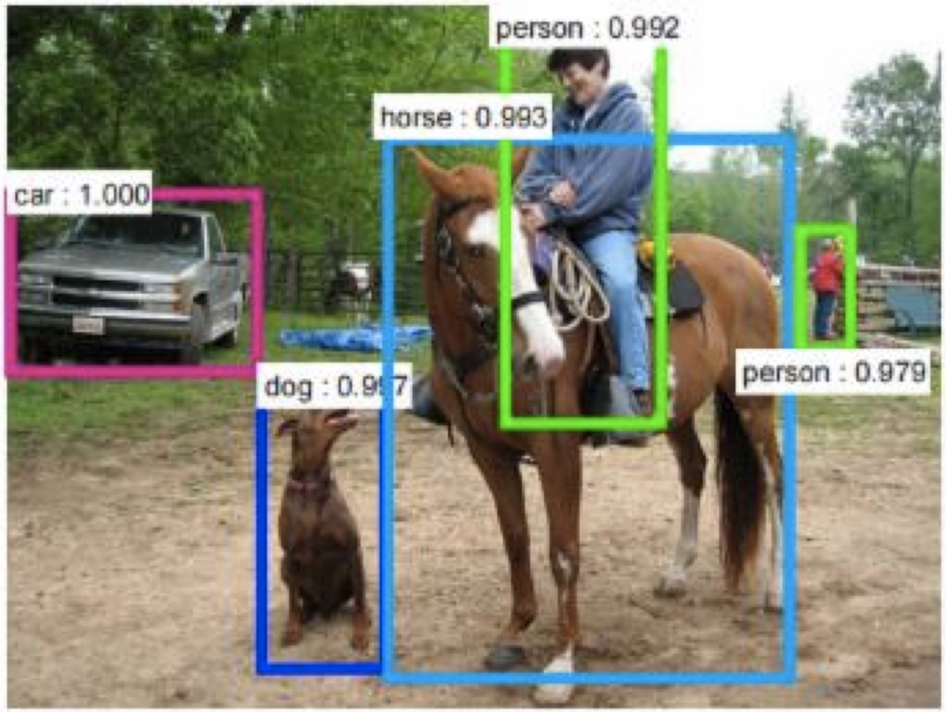


Progress in ImageNet Classification



ILSVRC: 1.5M Images in 1000 categories

Image Detection



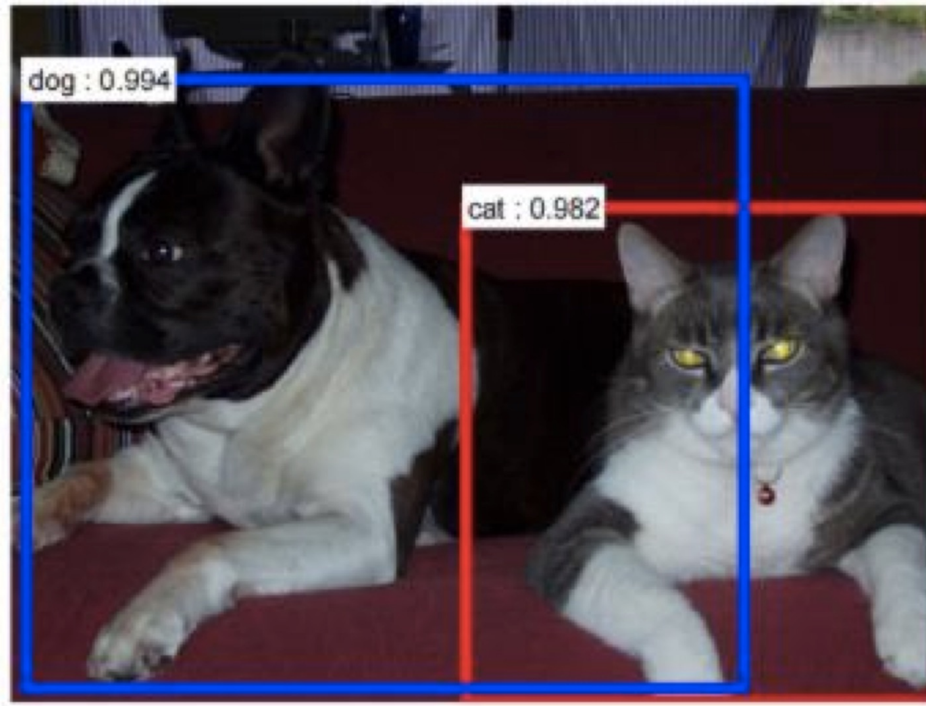


Image Segmentation

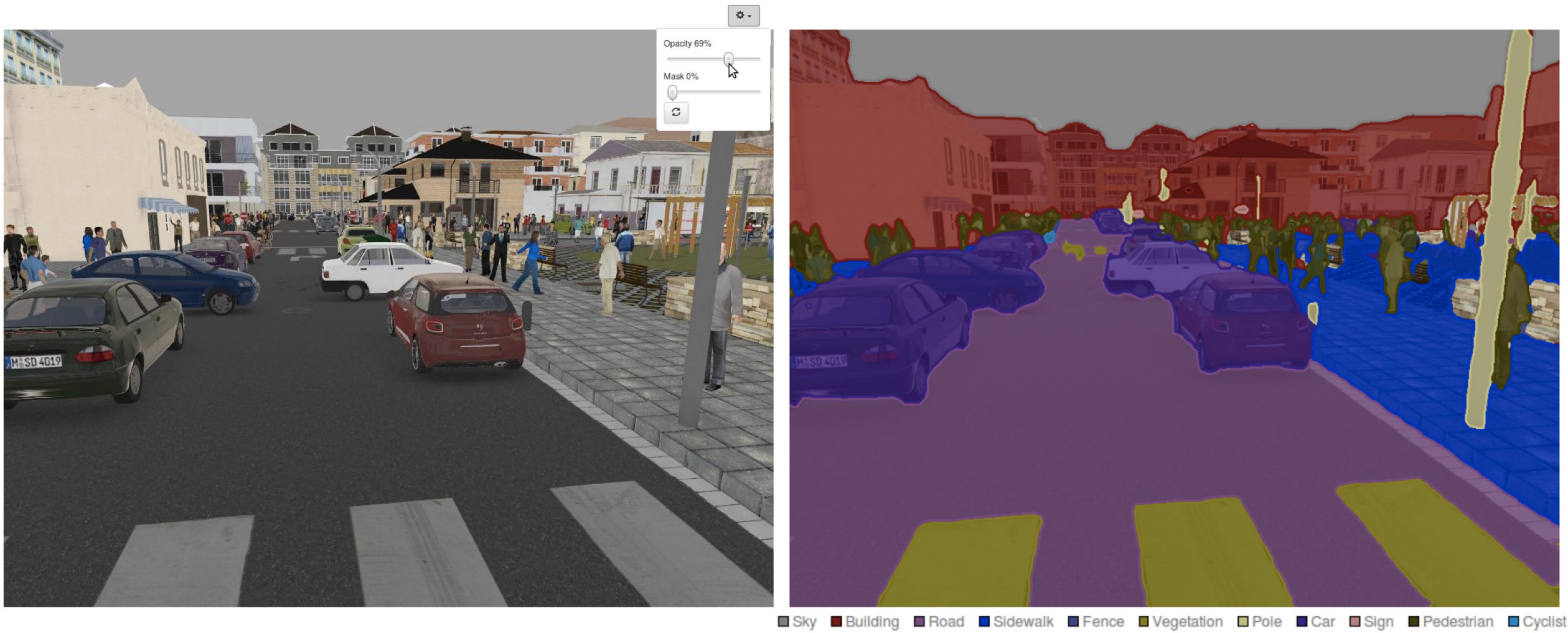


Image Generation

Deep Learning Network

“elephant”

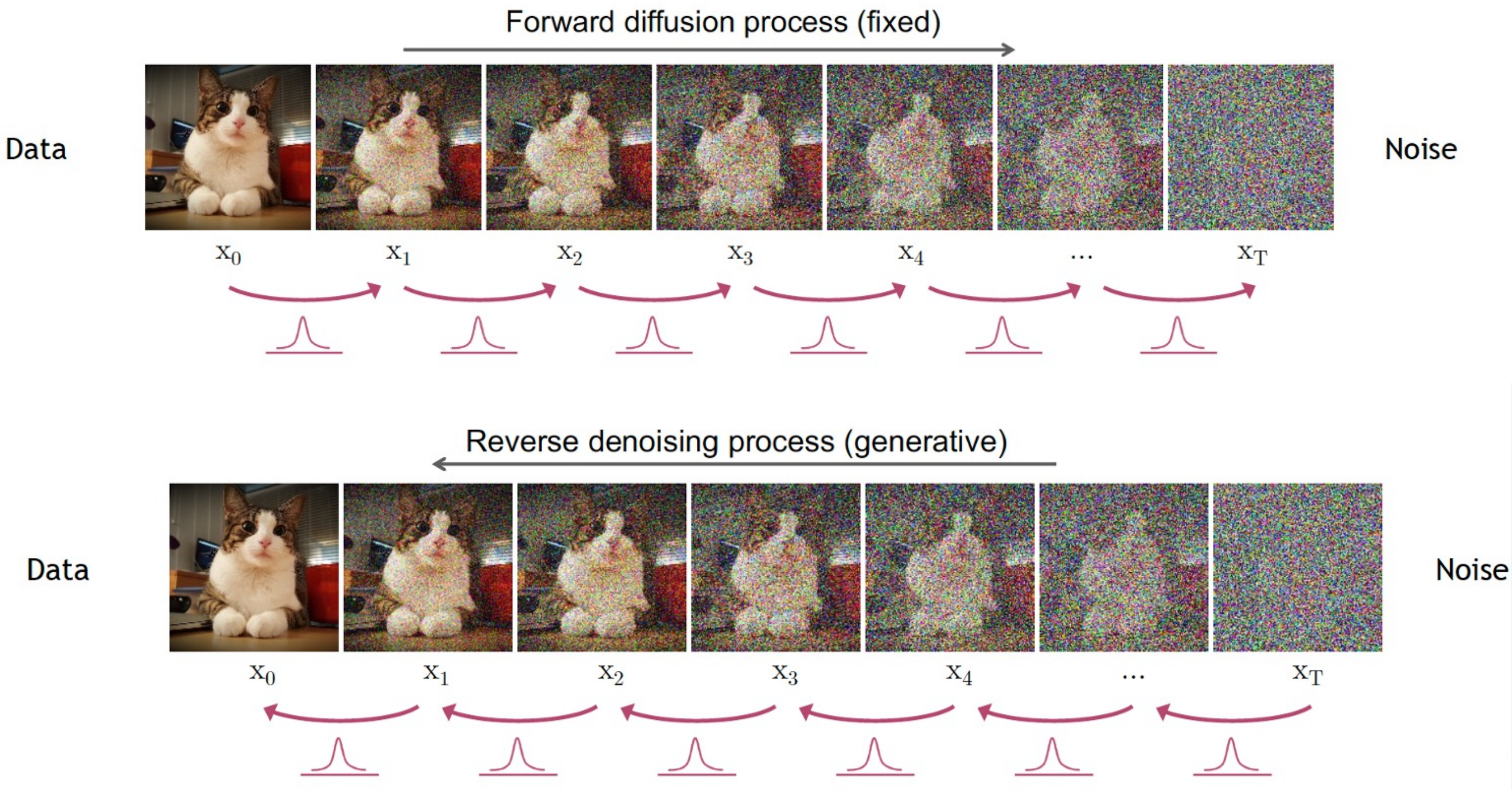
Can we run the DL network in the “reverse” direction and generate images?

Deep Learning Network

“elephant”

Generative Models!

Image Generation Using Diffusion Models

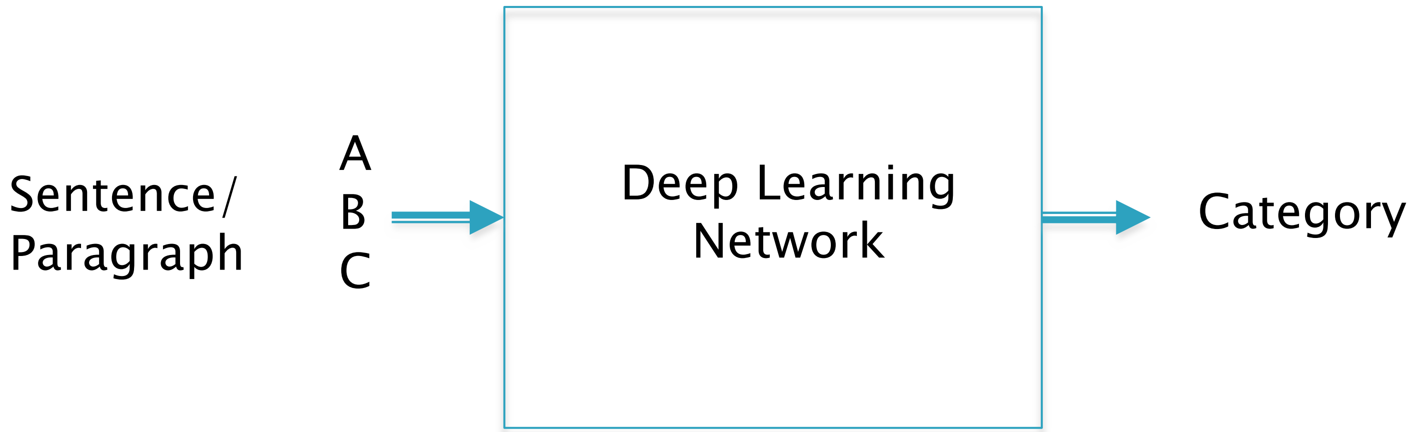


Natural Language Processing

Text Classification

The order of the input sequence matters!

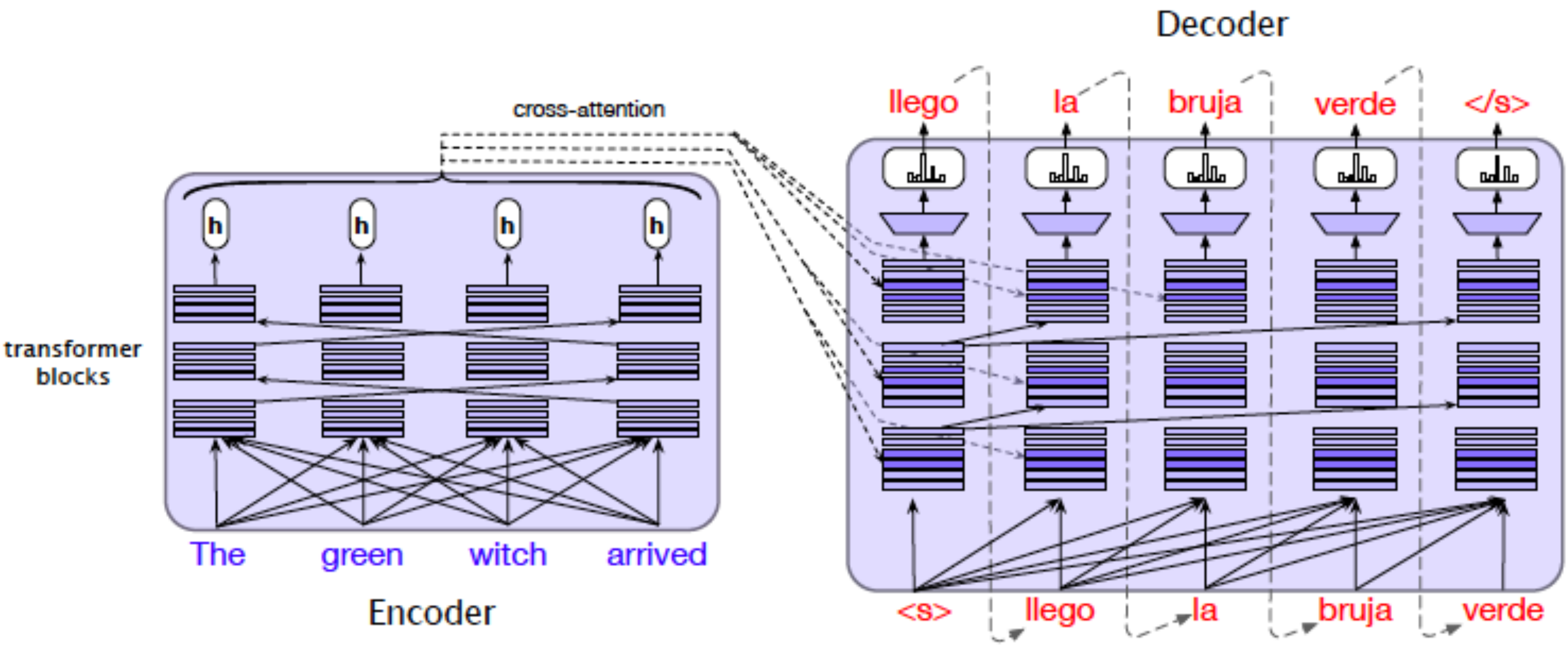
How to represent text?



NLP is done using Recurrent Neural Networks (RNNs)

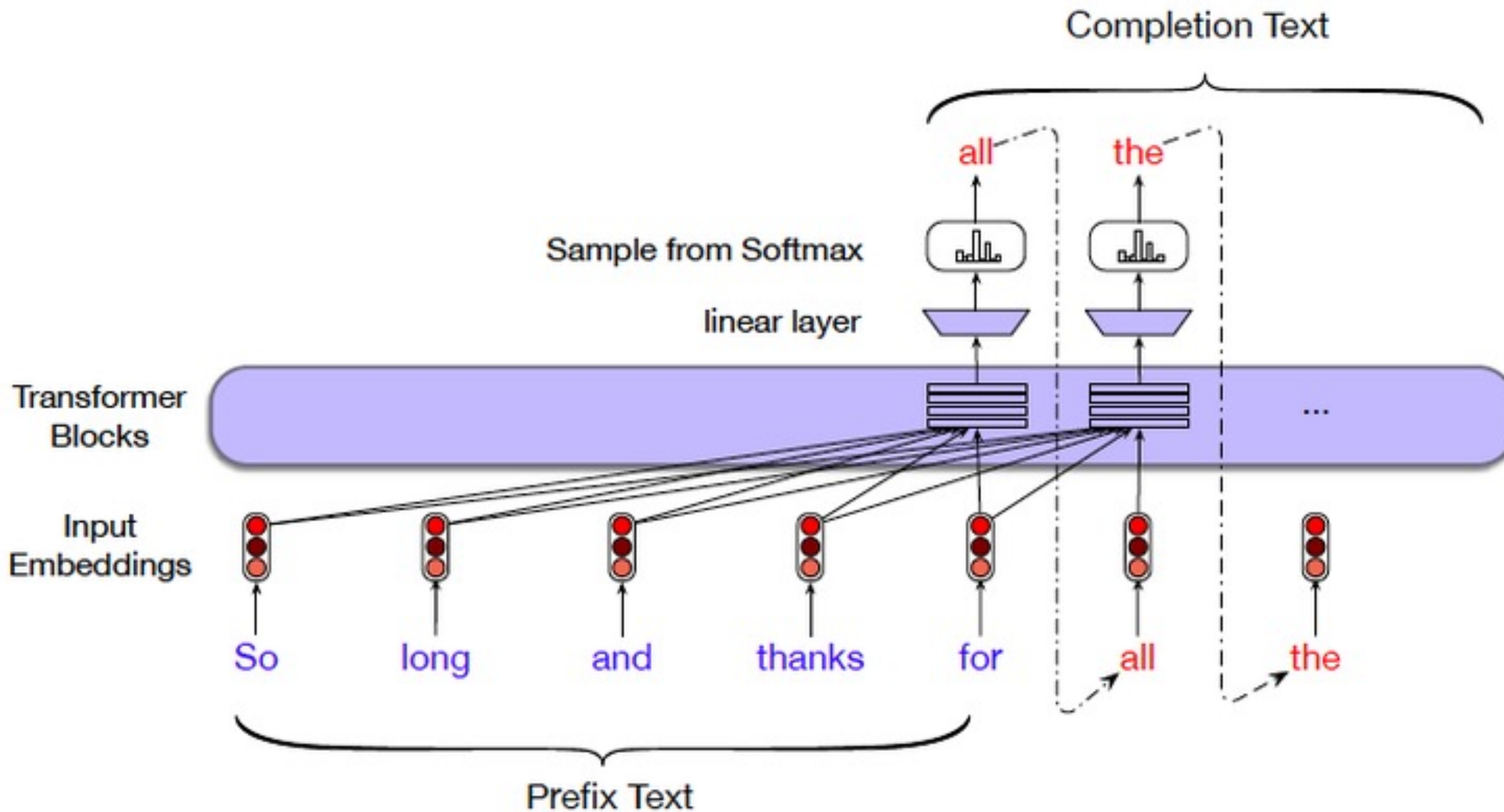
Transformers are a newer Model for solving NLP problems

Language Translation

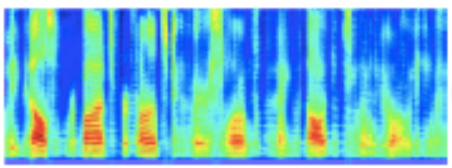


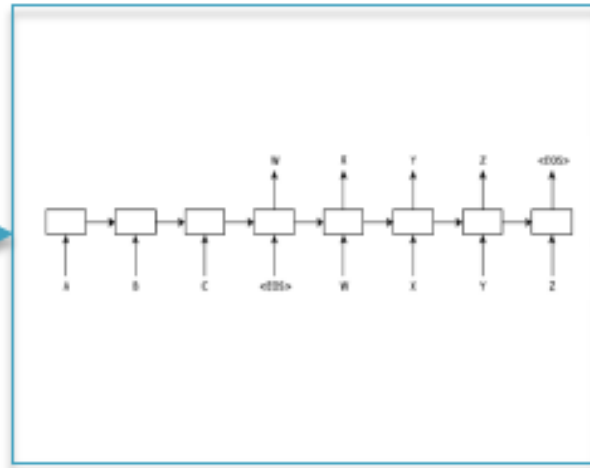
- The Queries come from the previous decoder layer
- The Memory keys and the values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the input sequence

Text Generation



Speech Recognition





W
X
Y
Z

Input: Audio Speech Pattern

Output: Sentence

Self Supervised Learning

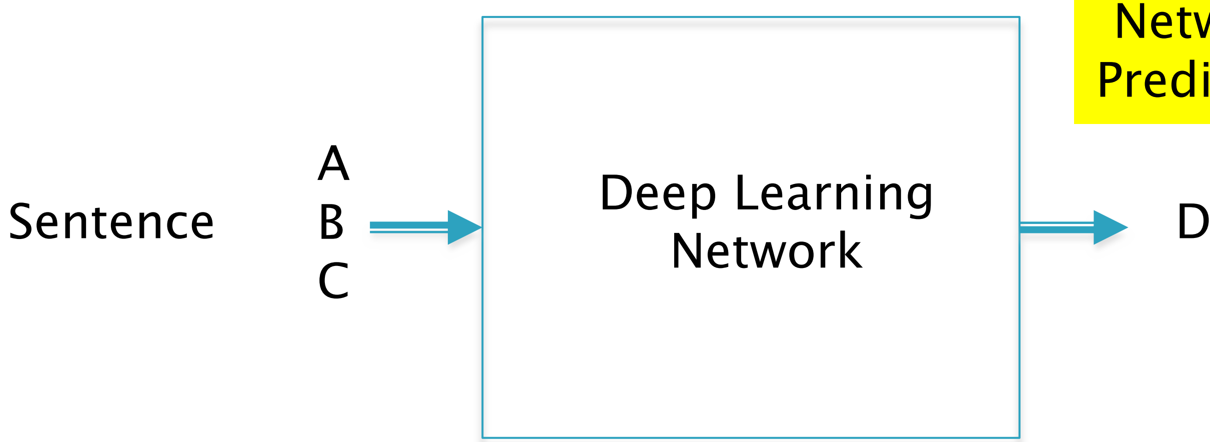
Self Supervised Learning

- Humans learn without using labels, how do they do it?
- Train models using prediction → Labels are generated automatically
- Example: NLP models trained by trying to predict the next word

Language Models

Labels are Auto Generated

Network is trained to Predict the Next Word!

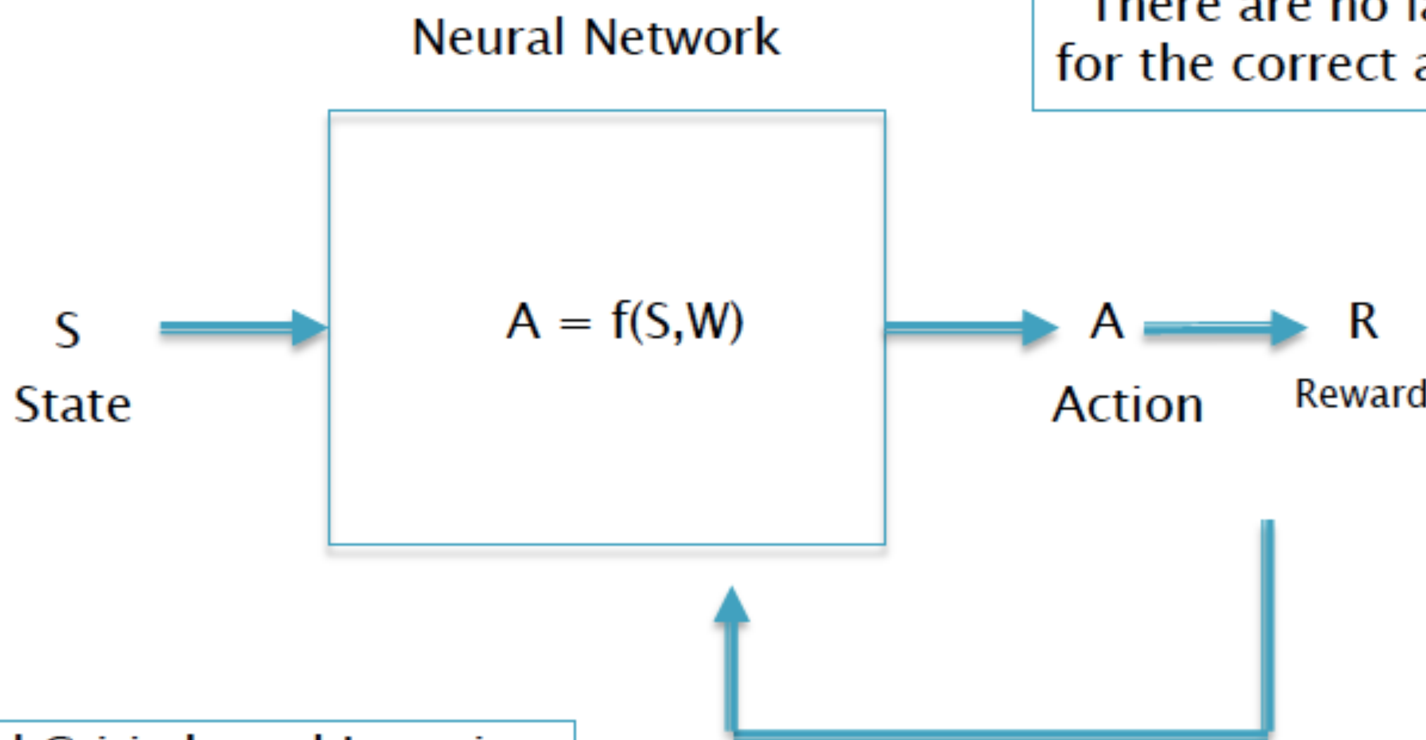


Once the DLN is trained, it can be modified to do other NLP tasks such as classification or Translation, using Supervised Learning with a much smaller training dataset

Reinforcement Learning

Reinforcement Learning

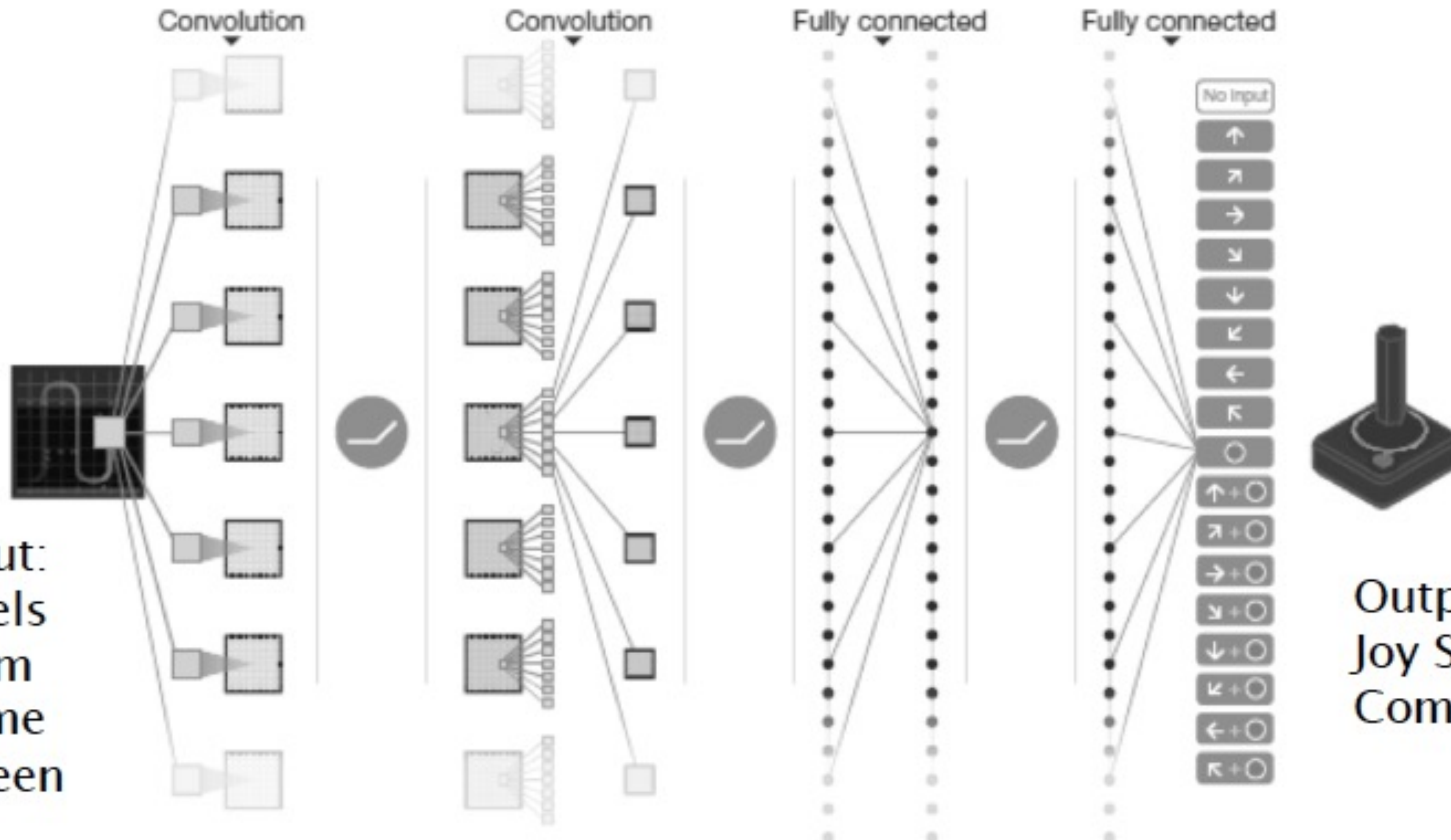
There are no labels for the correct action!



Also called Critic based Learning

Choose parameters W (ie Actions) to Maximize Total Rewards

Atari Game Player



Input:
Pixels
From
Game
Screen

Output:
Joy Stick
Commands

From “Human-level control through deep reinforcement learning”
By Mnih et.al.

Course Overview

General Tools and Algorithms

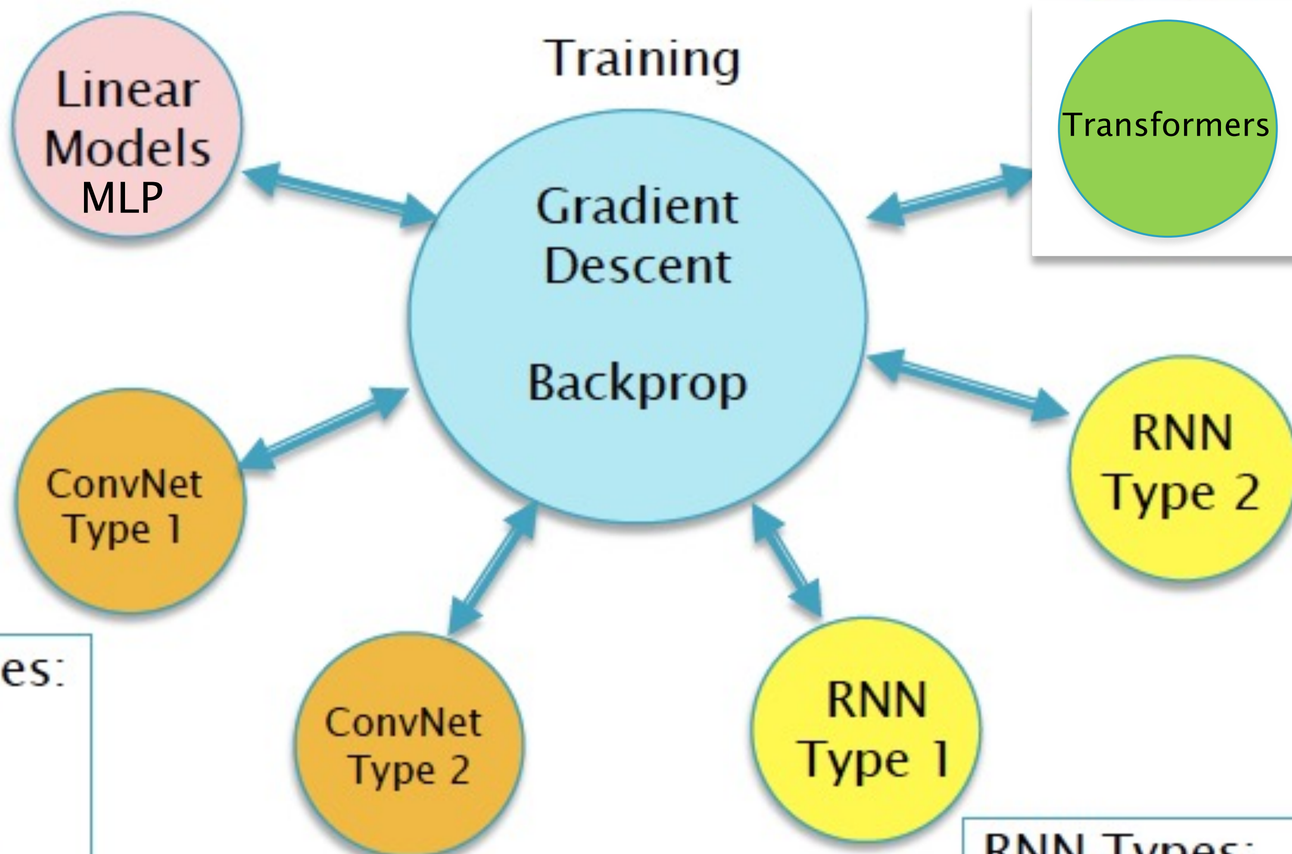
- Learning Models: Supervised, Self-Supervised
- Training DLNs: Gradient Descent, Backprop
- Improving the Training Process and Models
- Generalization Ability
- DLN Tools: Keras

Specialized Architectures

- Dense Feed Forward Networks
- Convolutional Neural Networks (ConvNets)/Image Processing
- Recurrent Neural Networks (RNNs)/Natural Language Processing
- Transformers
- Diffusion Models
- Reinforcement Learning

One Algorithm – Multiple Architectures

Transformer Types:
GPT
BERT
T5



ConvNet Types:
AlexNet
ResNet
InceptionNet

RNN Types:
Encoder Decoder RNNs
LSTMs
GRUs

Combo RNNs+ConvNets

Lecture Schedule

- **Lecture 1 – Introduction**: Introduction to Deep Learning and discussion of important applications, Introduction to Types of Deep Learning Systems: Supervised Learning, Reinforcement Learning, Unsupervised Learning, Self-Supervised Learning, An historical overview of Deep Learning
- **Lecture 2 – Mathematical Preliminaries**: An overview of Probability Theory, Bayes Rule, Random Variables, Random Sequences, Markov Chains, Maximum Likelihood Estimation, Basics of Linear Algebra, Matrices, Tensors
- **Lecture 3 – Linear Models**: The Classification and Regression Problems, Solving these using Supervised Learning, Binary Classification, Linear Models (Logistic Regression), Loss Functions, Introducing Gradient Descent, K-ary Classification, Using Keras to solve linear models
- **Lecture 4 – Dense Feedforward Models, Backprop**: Interpreting the Linear Classifier, Dense Feedforward Networks, The Backprop Algorithm, Forward and Backward Passes, Gradient Flow Algebra, Derivation of the Backprop Algorithm, Dense Feedforward Networks using Keras
- **Lecture 5 – Tools and Techniques**: Training Process, Some Common Training Datasets: MNIST, CIFAR-10, ILSVRC, IMDB etc, Getting Deeper into Keras, Ingesting Data into Keras Models: Image, Text and Tabular
- **Lecture 6 – The Backprop Algorithm**: Gradient Flow Calculus, Forward and Backward Passes in Backprop, Derivation of Backprop
- **Lecture 7 – Training Part 1**: Vanishing Gradient Problem, Activation and Loss Functions, Techniques to Improve Stochastic Gradient Descent, Illustration of algorithms using Keras, Instructions for doing Term Project
- **Lecture 8 – Training Part 2**: Weight Initialization, Data Pre-Processing, Batch Normalization, Model Under-fitting and Over-fitting problems, Illustration of Algorithms using Keras
- **Lecture 9 – Training Part 3**: Regularization Techniques – L2, L1 and Dropout, Dataset Augmentation, Hyper-Parameter Selection – Manual and Automated Tuning, Model Ensembles, Illustration of Algorithms using Keras
- **Lecture 10 – Convolutional Neural Networks (ConvNets) Part 1**: History and Applications of ConvNets, ConvNet Architecture, 2D Convolutions, 1D Convolutions, Sizing ConvNets, Modeling ConvNets with Keras

Lecture Schedule

- **Lecture 11 – Convolutional Neural Networks (ConvNets) Part 2**: Pooling and Padding in ConvNets, Trends in ConvNet Design: Small Filters, Global Max Pooling, Depthwise Separable Convolutions, Some Historically significant ConvNet Architectures – LeNet5, AlexNet, ZFNet, VGGNet
- **Lecture 12 – Convolutional Neural Networks (ConvNets) Part 3**: ConvNet Architectures (cont): InceptionNet, ResNet. DenseNet, SqueezeNet, Transfer Learning using Keras, Text and Tabular Data Processing using 1D Convolutions
- **Lecture 13 – Convolutional Neural Networks (ConvNets) Part 4**: Solution of Image processing problems such as Localization, Detection and Segmentation using ConvNets, Visualization in ConvNets: Inverse convolutions, Generating Images using Gradient Ascent, the Deep Dream algorithm, Texture synthesis using Gram Matrices, Neural Style Transfer
- **Lectures 14 – Recurrent Neural Networks (RNNs) Part 1**: RNN Architectures – One to One, Many to One, Many to Many; Contrasting RNNs with ConvNets, Deep and Bi-Directional RNNs, Combination of RNNs and ConvNets
- **Lectures 15 – Recurrent Neural Networks (RNNs) Part 2**: Difficulties in Training RNNs and how to solve them, Back Propagation through Time (BPTT) Algorithm, LSTMs, GRUs, Word Embeddings and the Word2Vec algorithm, Modeling RNNs with Keras
- **Lectures 16 – Natural Language Processing (NLP) Part 1**: Application of RNNs to Natural Language Processing, Probabilistic Language Models, Beam Search, Softmax Temperature, Text Classification, Machine Translation, Attention Mechanism in RNNs.
- **Lectures 17 – Natural Language Processing (NLP) Part 2**: Image Captioning, Question Answering Systems, Reading Comprehension, Information Retrieval Systems, Speech Transcription
- **Lecture 18 – Transformers:** Self Attention, Transformer Encoder and Decoder models, OpenAI Transformer (GPT), BERT
- **Lecture 19 – Diffusion Models**: Latent Variables, ELBO Bound, Forward and Backward Diffusion Processes, DDPM and DDIM Algorithms
- **Lectures 20 – Reinforcement Learning (RL):** Introduction, Components of a RL System: Agents, Rewards, Actions, Deep RL, Playing Pong with Policy Gradients, Playing Go with Supervised Learning and Policy Gradients, Imitation Learning

Further Reading

- Chapter 1 of Das and Varma
  https://srdas.github.io/DLBook2/Introduction.html
- Chapter 1 of Chollet

- Python Numpy Tutorials :
  https://sites.engineering.ucsb.edu/~shell/che210d/numpy.pdf
  http://cs231n.github.io/python-numpy-tutorial/