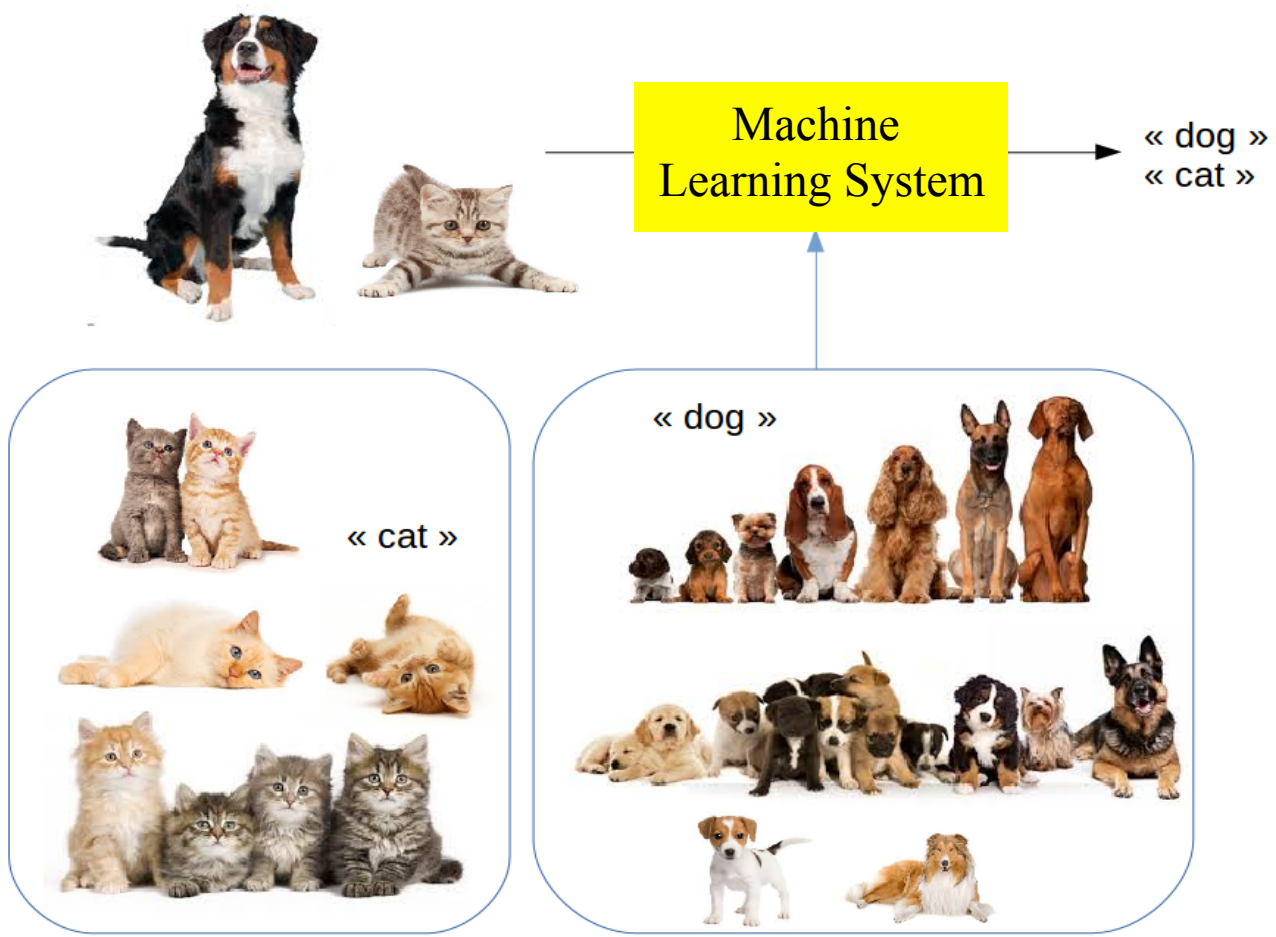
Recent Advances in Machine Learning

Deep Learning

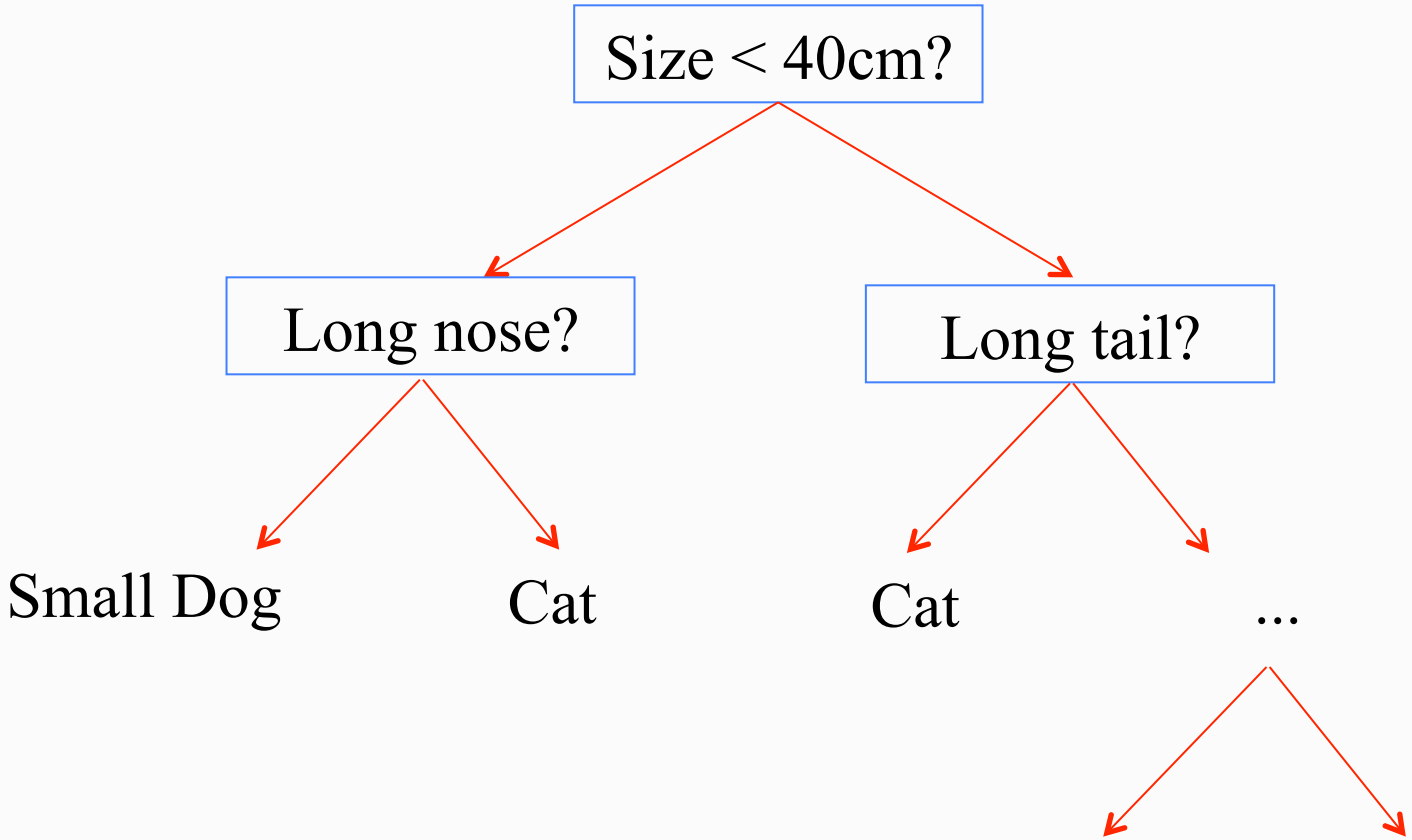
Berrin Yanıkoğlu
berrin@sabanciuniv.edu

Thanks to 0wikipedia, Andrew Ng and others for many beautiful images.

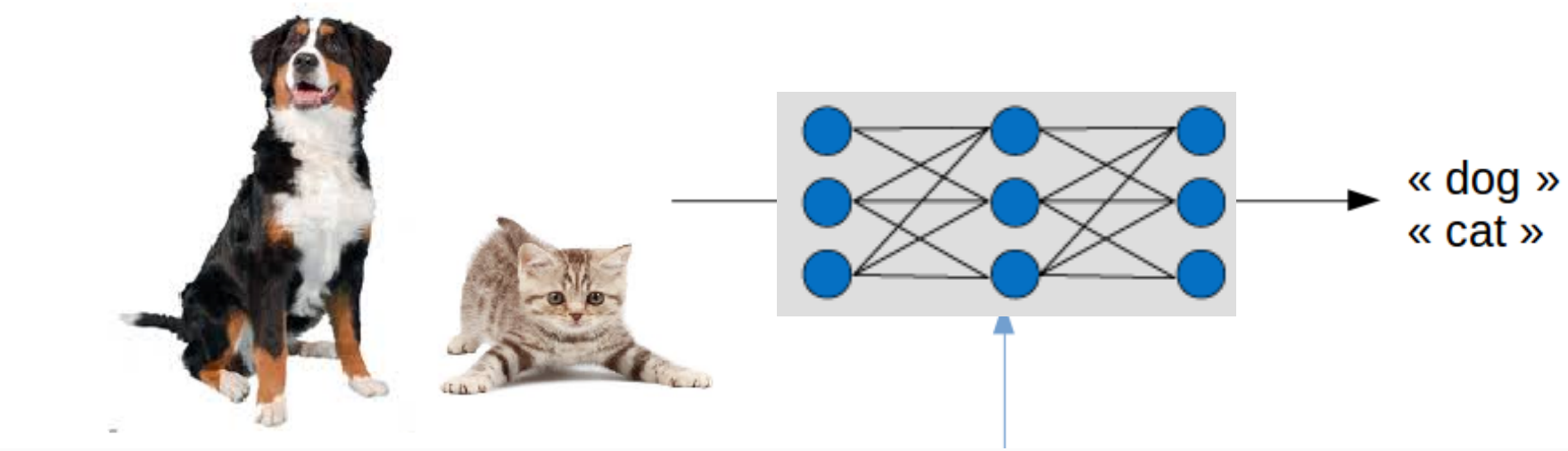
- Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed.



Decision Trees



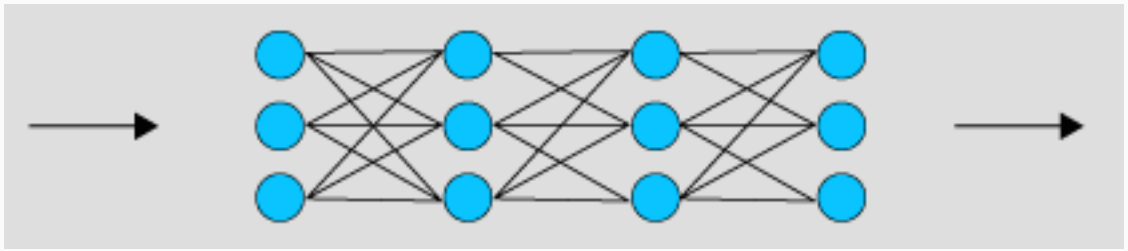
Neural Networks



Supervised Learning

- Learning the underlying mapping from given input-output pairs.
  - **Classification**: Given a set of previously observed good/bad applicants, learn a model to classify good applicants from bad ones.
  - **Regression**: Given the data of a previous applicants, learn a model to predict the applicant ranking score.

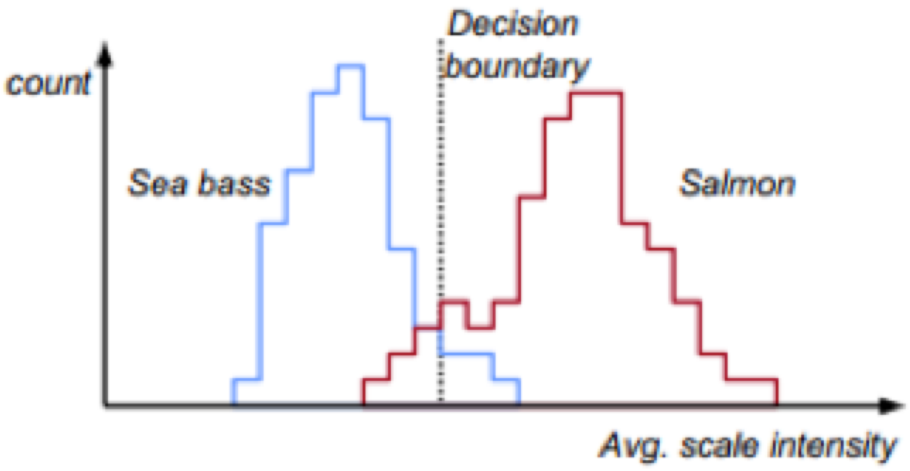
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{bmatrix}$$

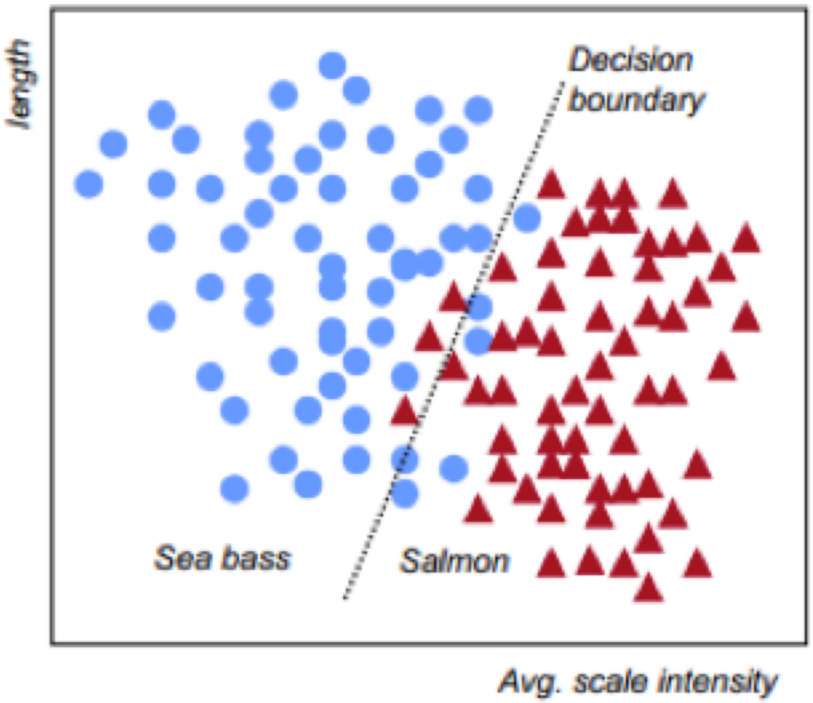


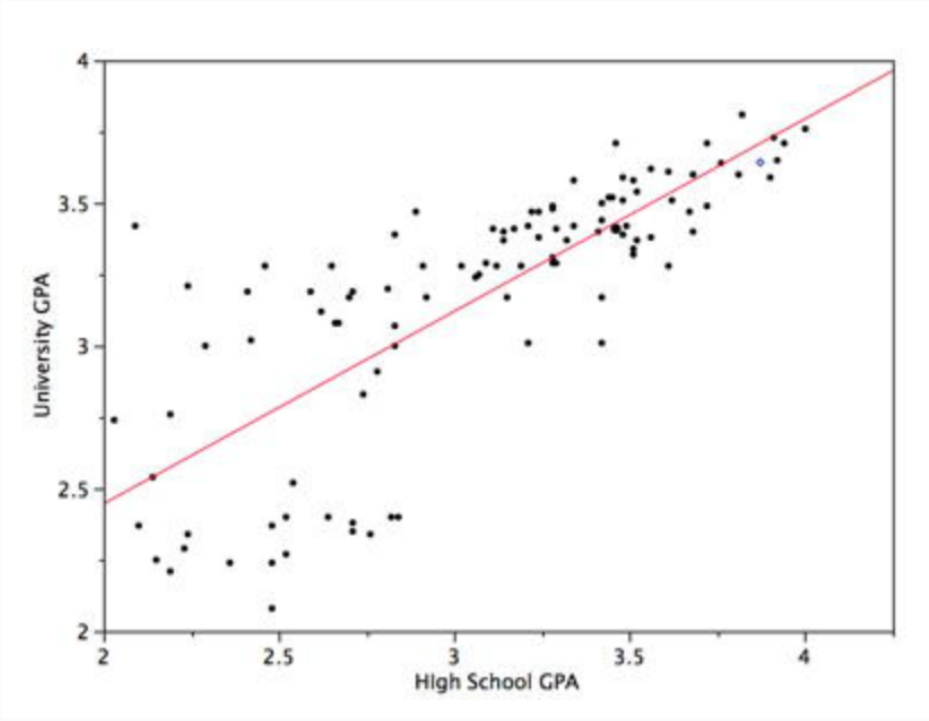
y

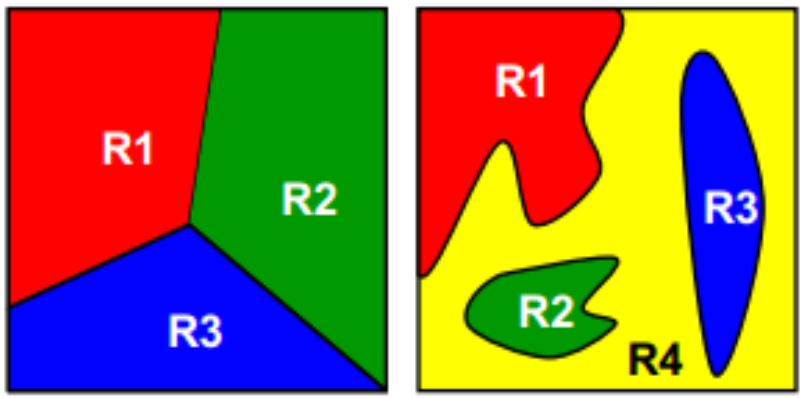
Training set: $\left\{\left(\mathbf{x}^t, y^t\right)\right\}$

Mapping to learn: $f : \mathbf{x} \longrightarrow y$


count
Decision boundary
Sea bass
Salmon
Avg. scale intensity


length
Decision boundary
Sea bass
Salmon
Avg. scale intensity


University GPA
High School GPA
2
2.5
3
3.5
4


R1
R2
R3
R1
R3
R2
R4

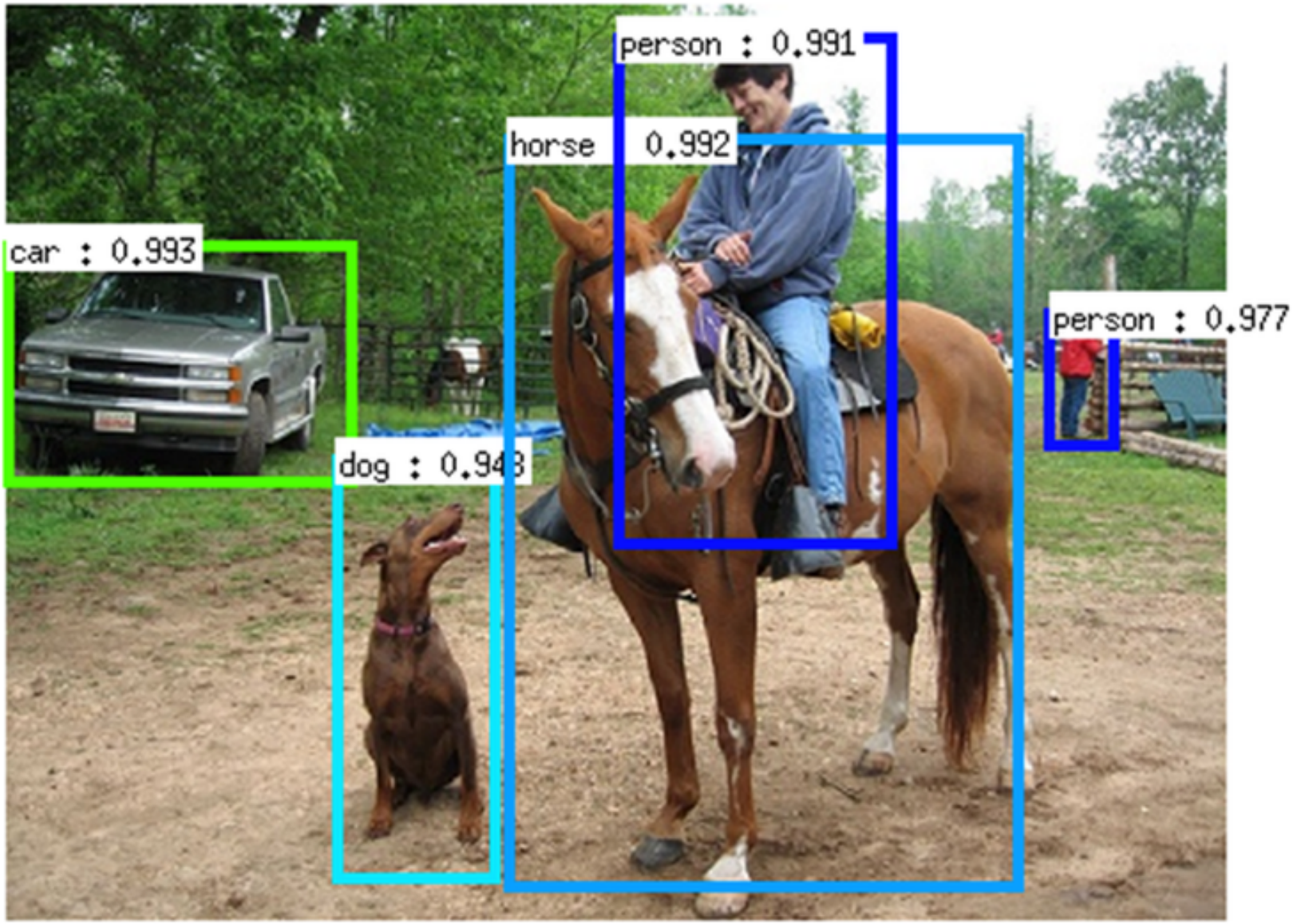
Current Machine Learning Problems

Object Classification

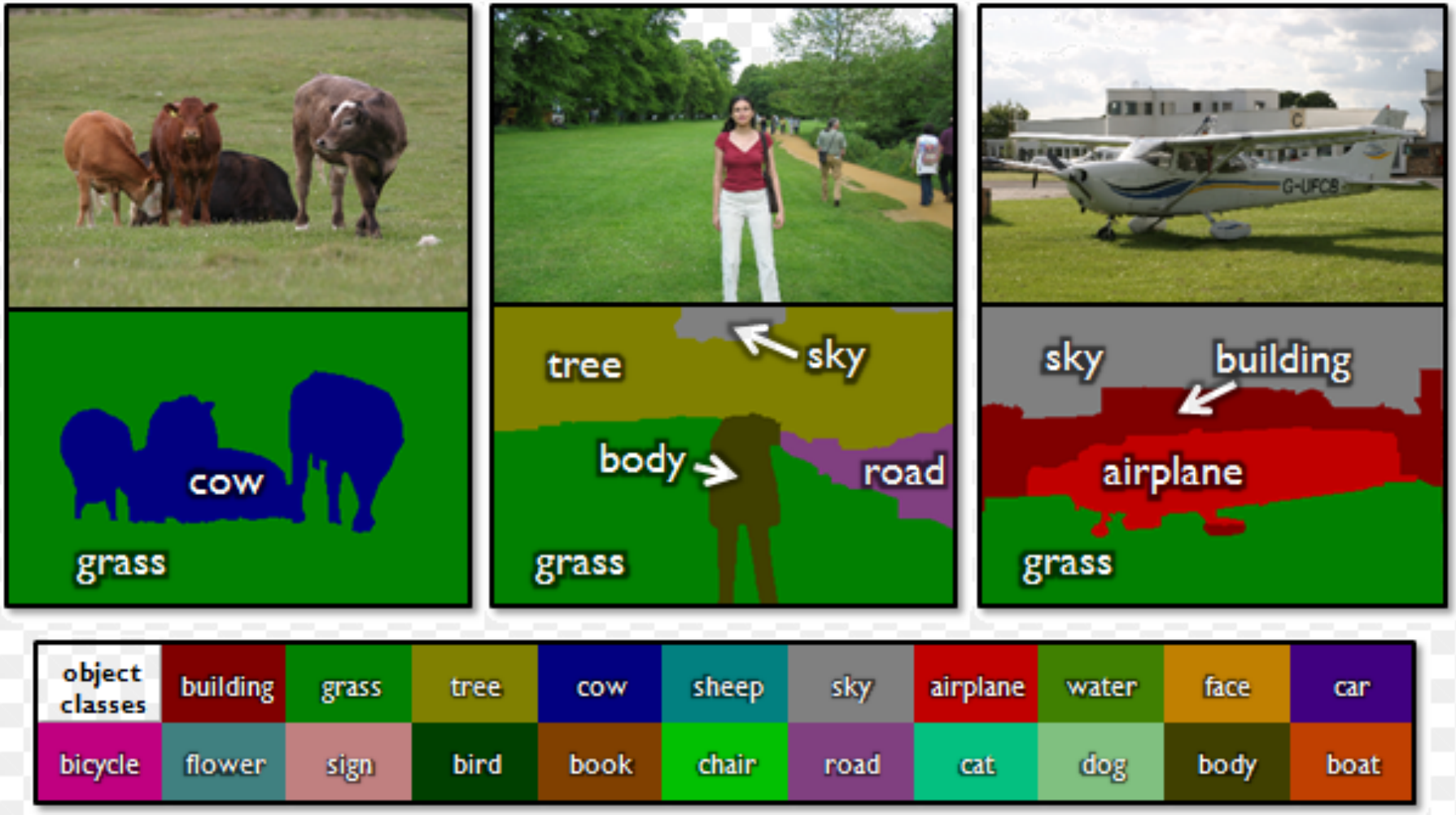
Dog: 94%
Cat: 31%
Bird: 2%
Boat: 0%

Dog: 37%
Cat: 91%
Bird: 21%
Boat: 1%

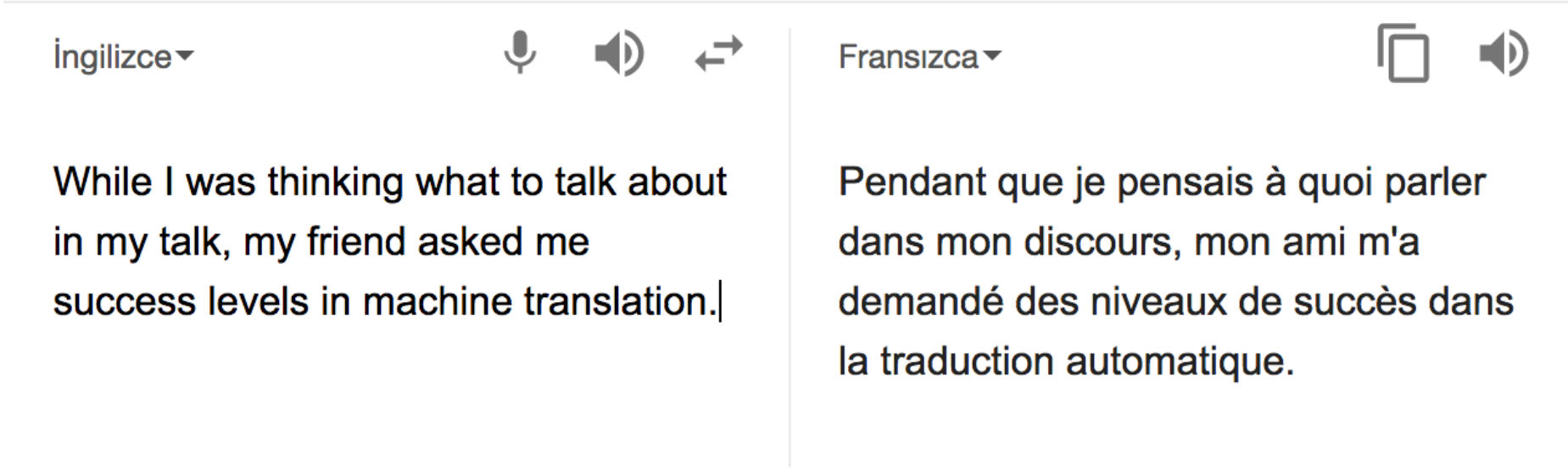
Object Detection



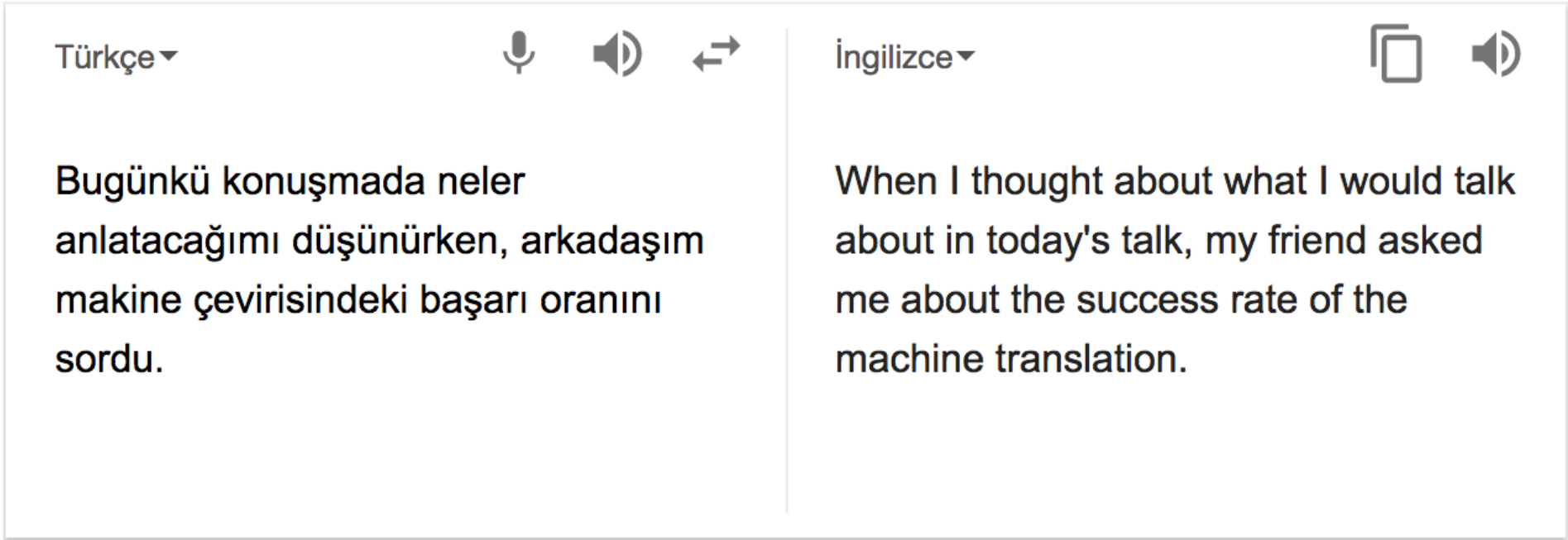
Semantic Segmentation



Machine Translation



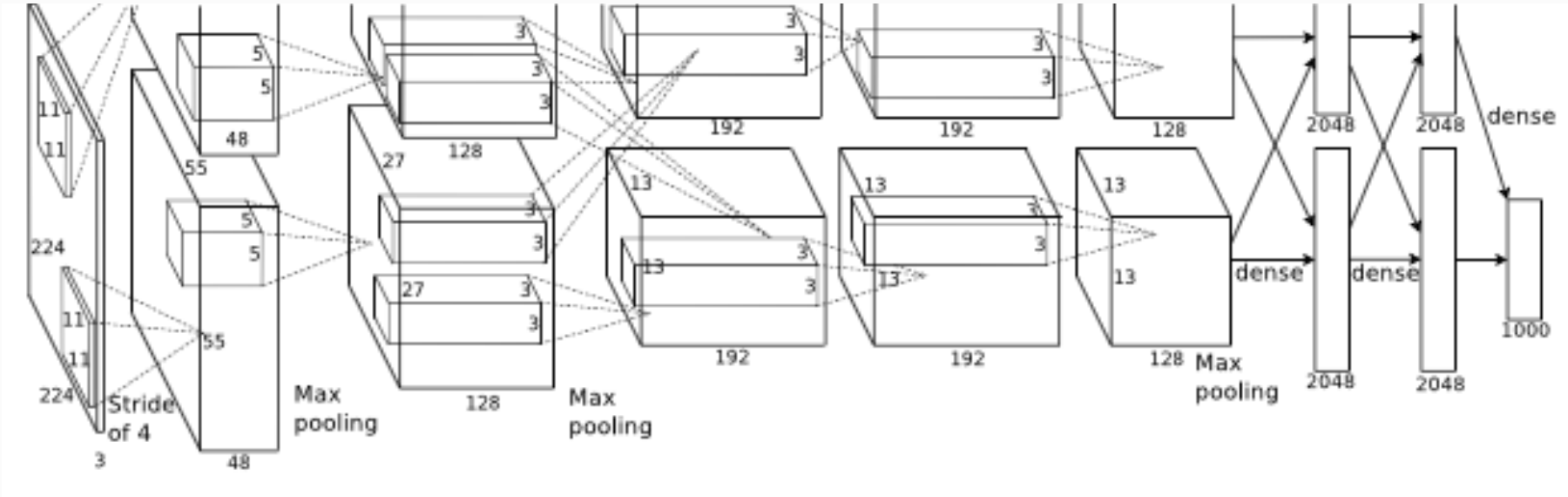
Machine Translation



Bugünkü konuşmada neler anlatacağımı düşünürken arkadaşım otomatik çeviride başarı nasıl diye sordu

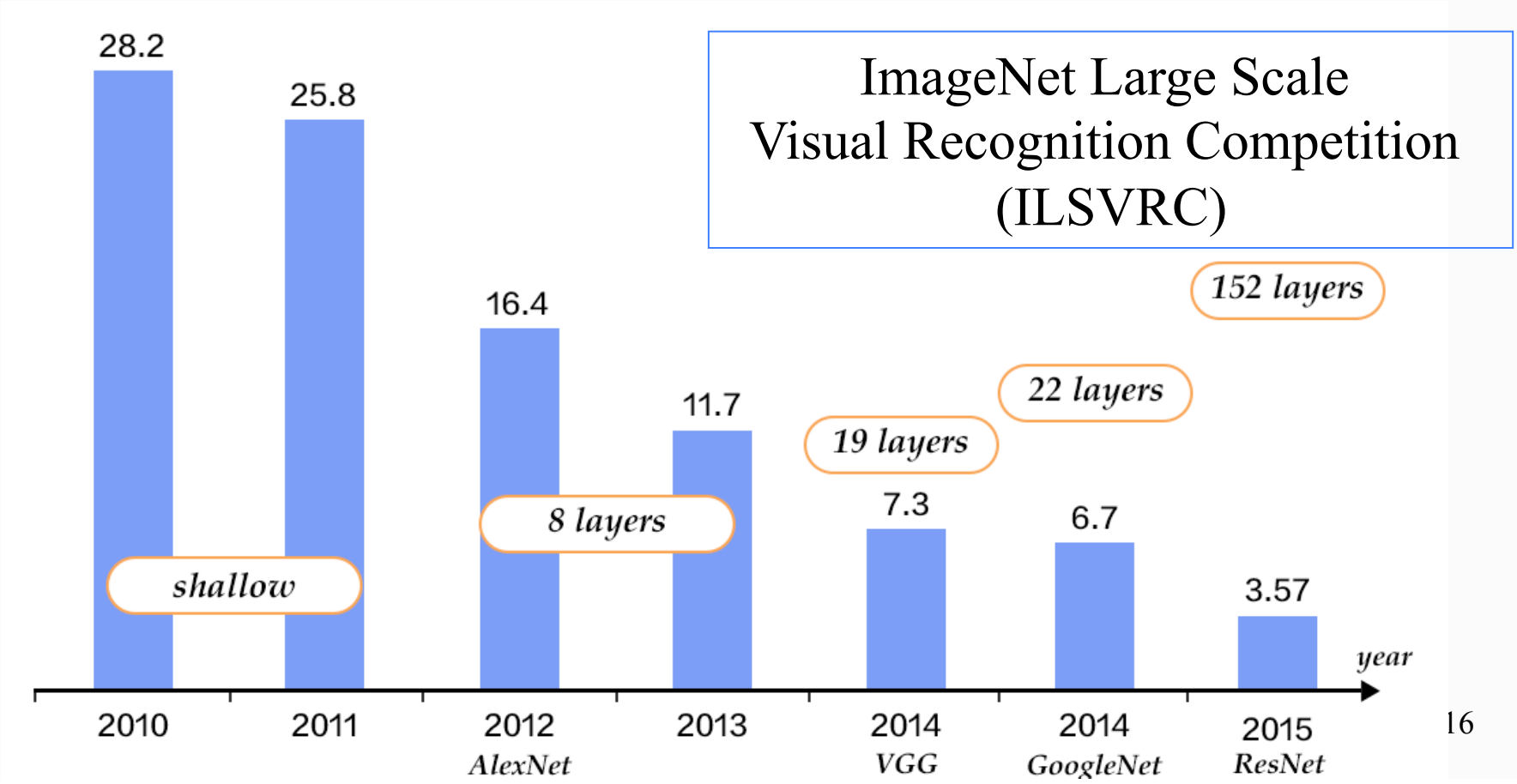
When I thought about what I would talk about in today's talk, my friend asked me how to succeed in automatic conversation

Deep Learning

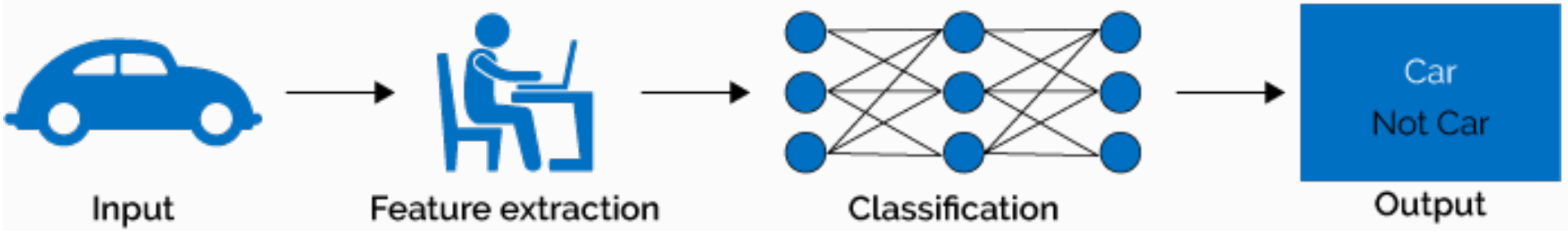


650,000 neurons and 60 Million parameters!

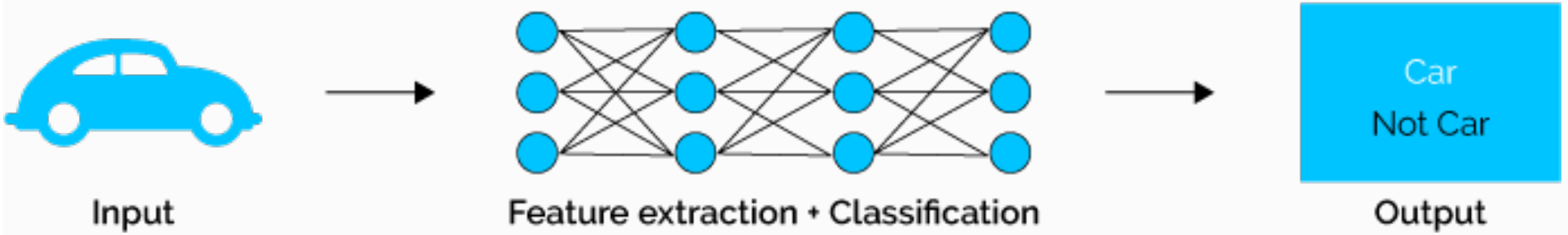
- Deep neural networks have shown major breakthroughs in many machine learning, computer vision, NLP, speech understanding... problems.

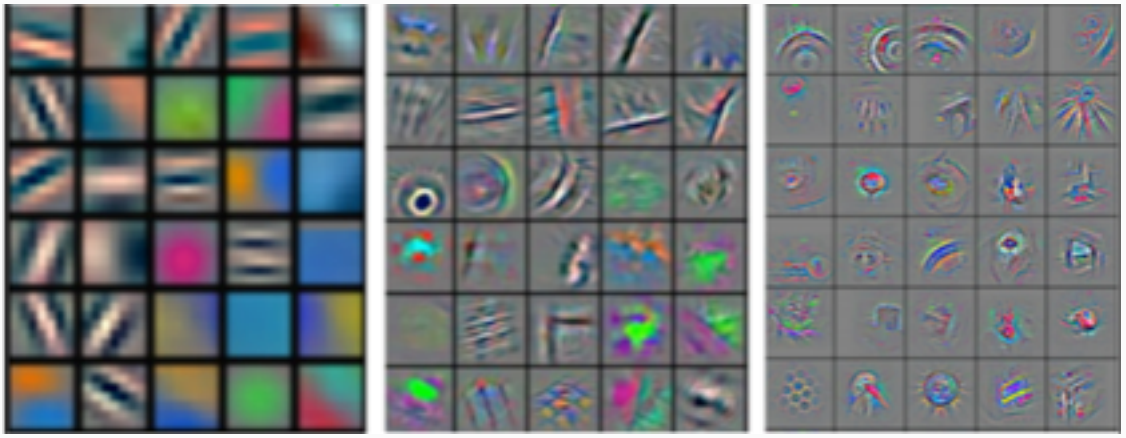


Machine Learning

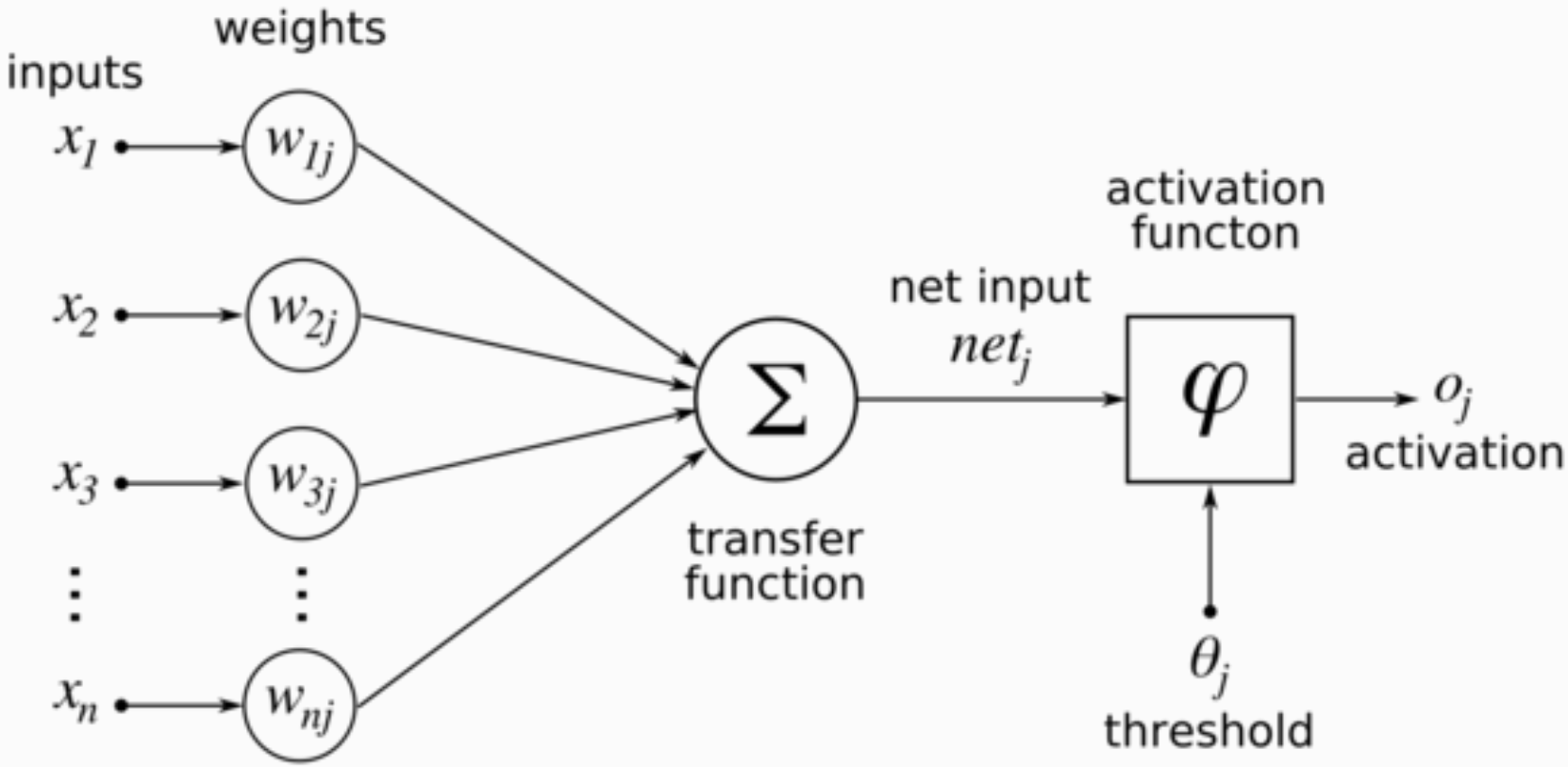


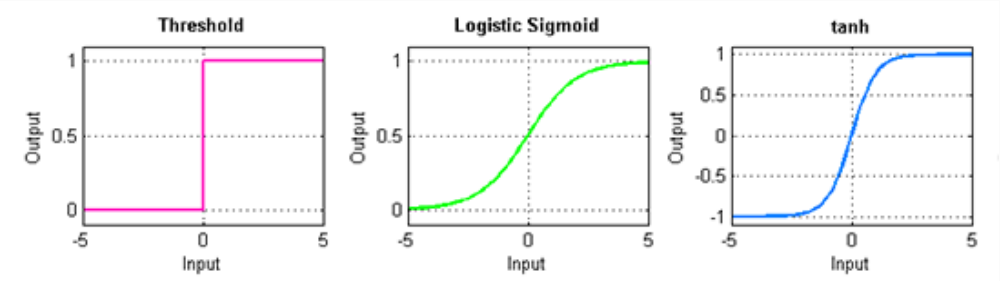
Deep Learning



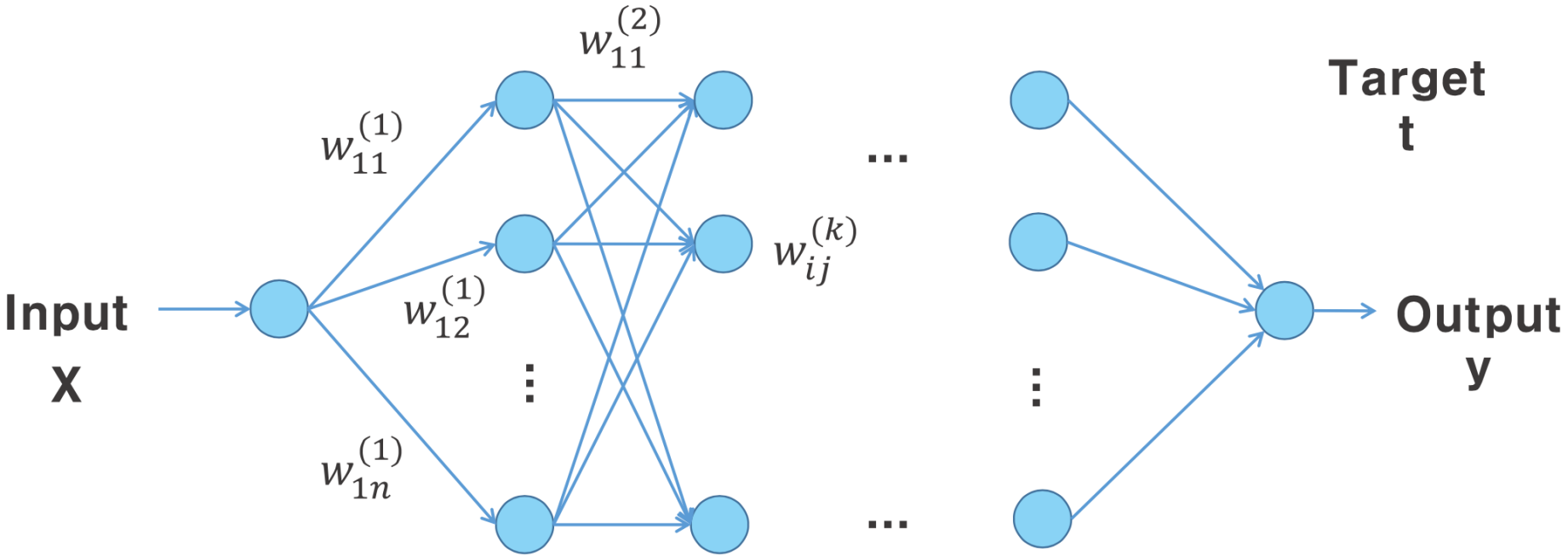


Artificial Neuron





Training a Neural Network



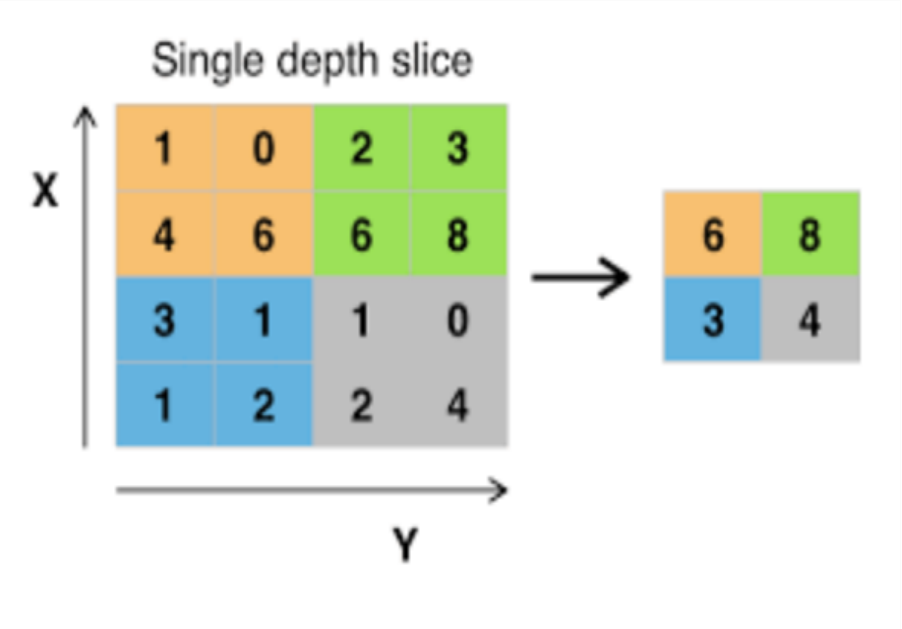
Given training set $\{(x_i, t_i)\}$,

Find **W** that minimizes $E = \sum ||t_i - y_i||^2$

Iteratively update **W** along error gradient

-> **gradient descent**

- Data!
- Advances in algorithms and theory
  - Weight sharing through convolutions
  - Relu
  - Max pool
  - Dropout
  - Data augmentation
  - ...
- GPU programming

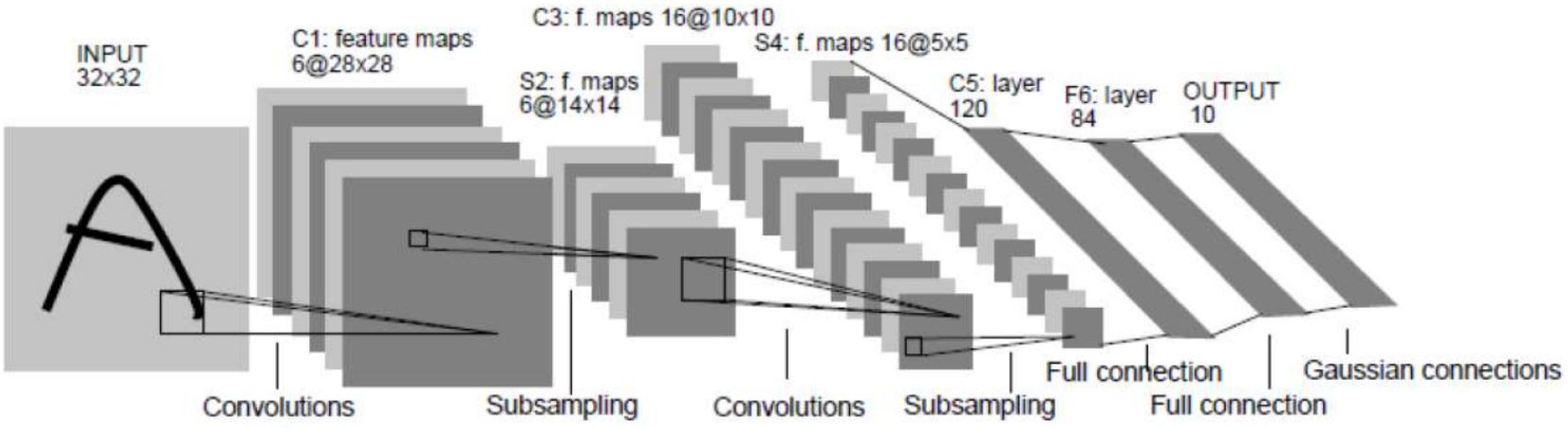


Convolutional Neural Network(CNN)

- **Handwritten digit recognition [LeCun 98]**
- **(Convolution-Subsampling) × N + (Full connection) × M**

Features extraction **Classification**

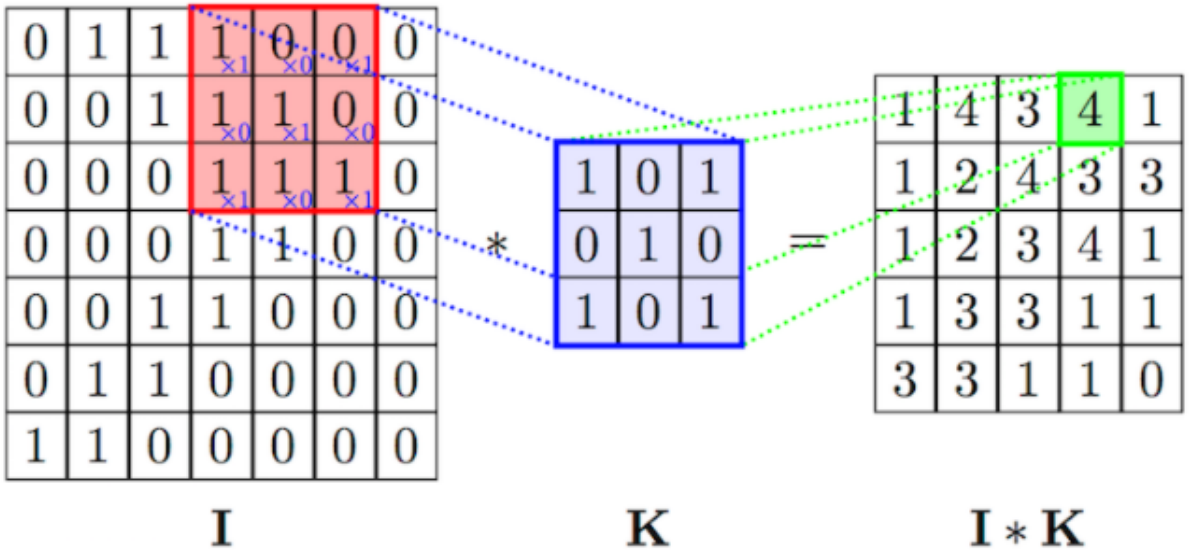
- **Neural network that makes use of prior knowledge about images**



Convolution

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

Dot product between pixels in the receptive field and the weights

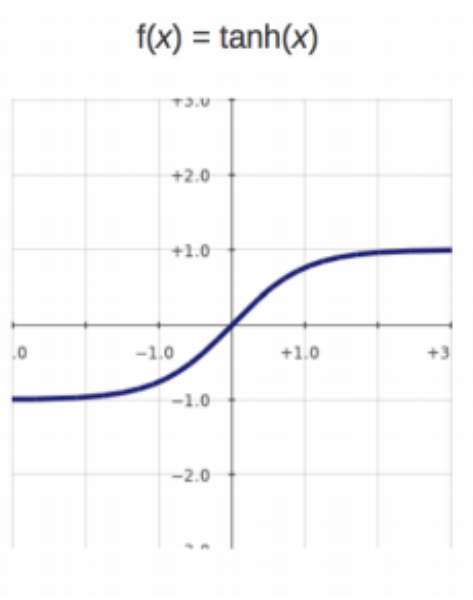


RELU Nonlinearity

- Standard way to model a neuron

  $f(x) = tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$

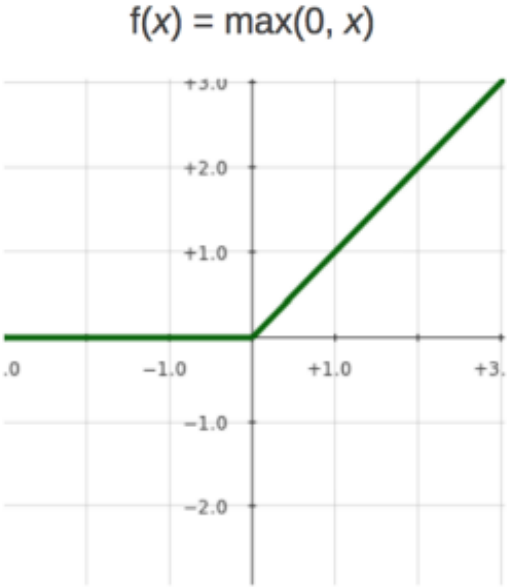
  Very slow to train



- Non-saturating nonlinearity (RELU)

  $f(x) = max(0, x)$

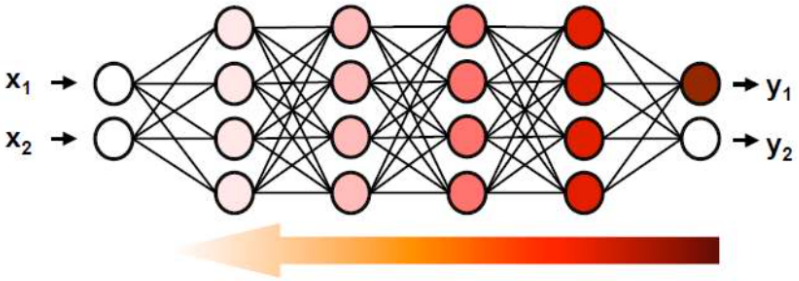
  Quick to train



Pre-training

- Backpropagation may not work well with deep network
  - vanishing gradient problem
  - lower layers may not learn much about the task.

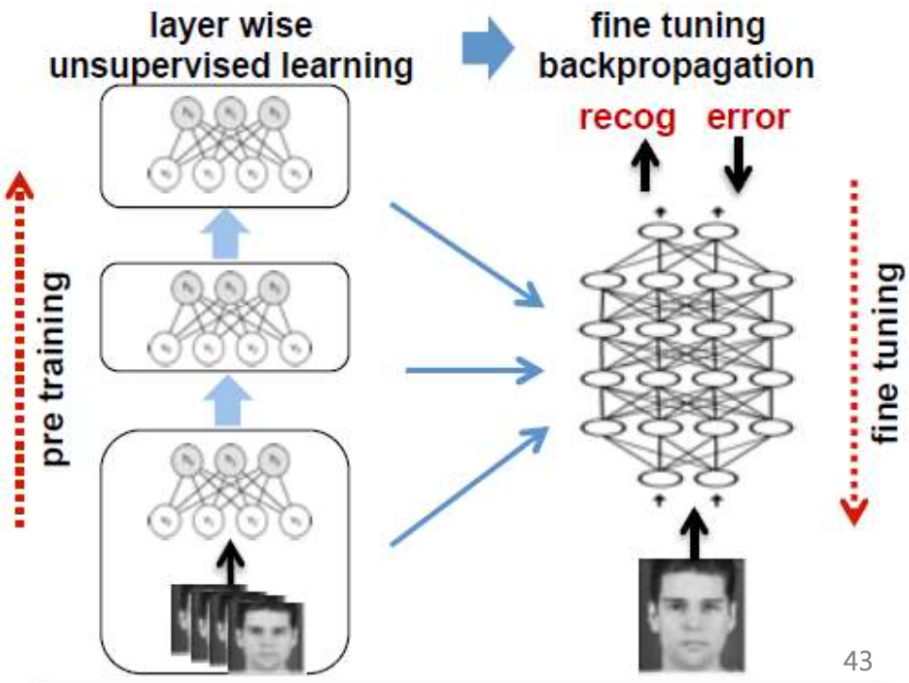
vanishing gradient



Backward error information vanishing

good initialization is crucial

Pre-training

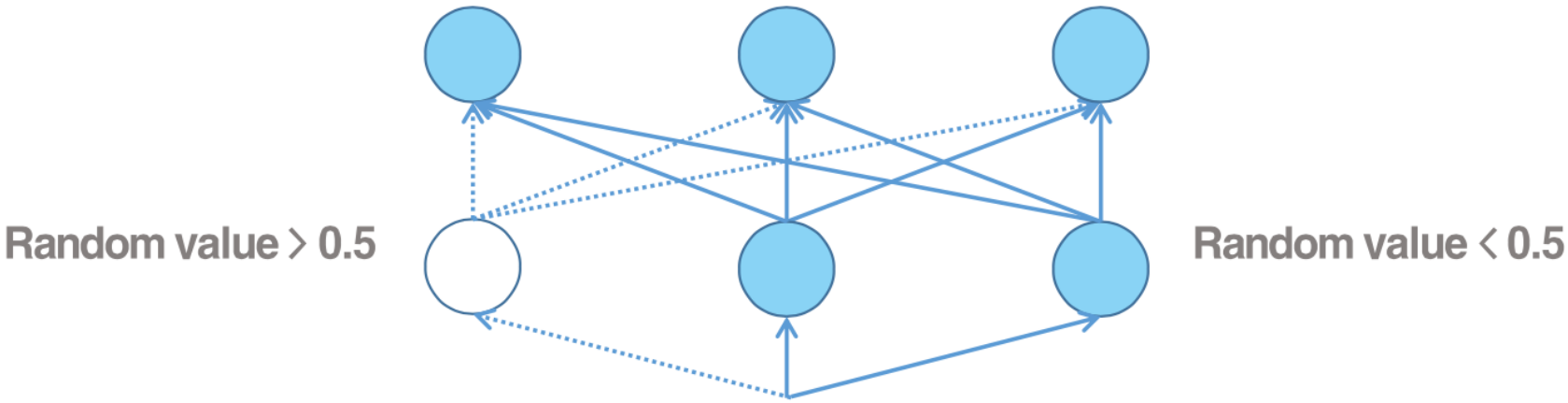


Dropout

An Efficient way to average many large neural nets.

- Consider a neural net with one hidden layer
- Each time we present a training example,
  randomly omit each hidden unit with probability 0.5
- Randomly sampling from 2^H different architectures.

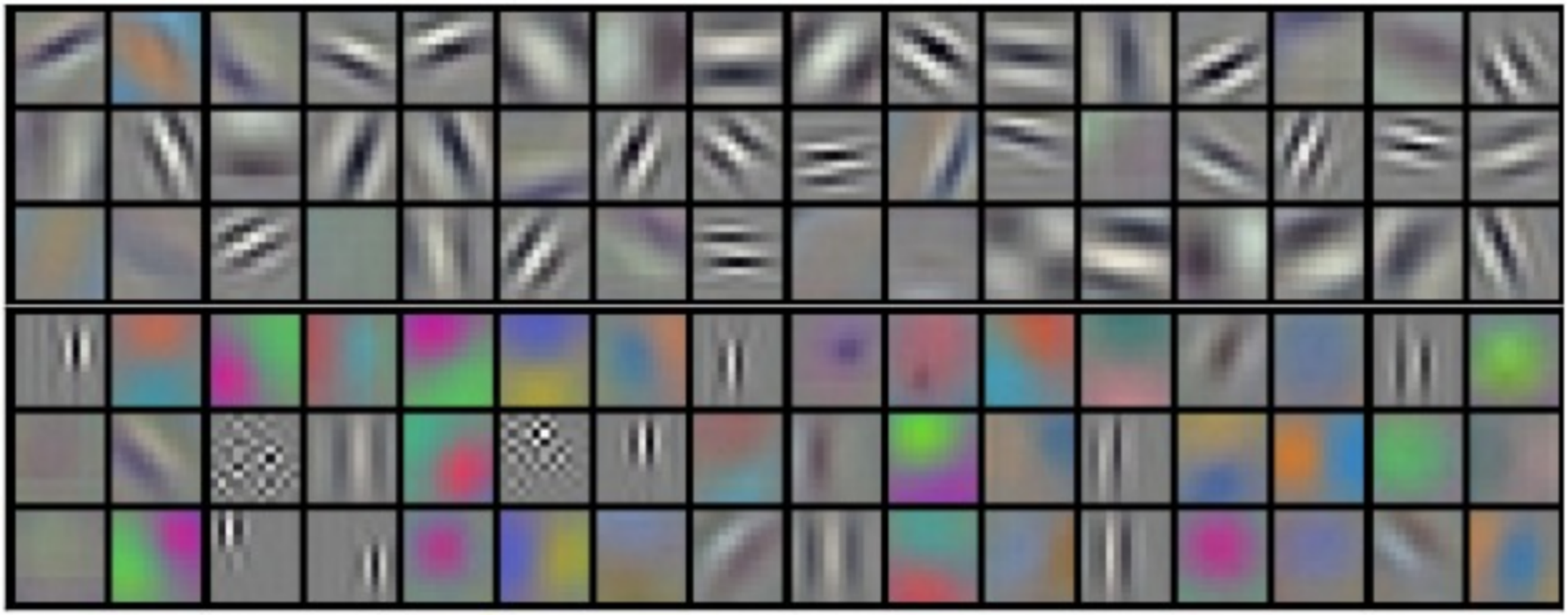
All architectures share weights



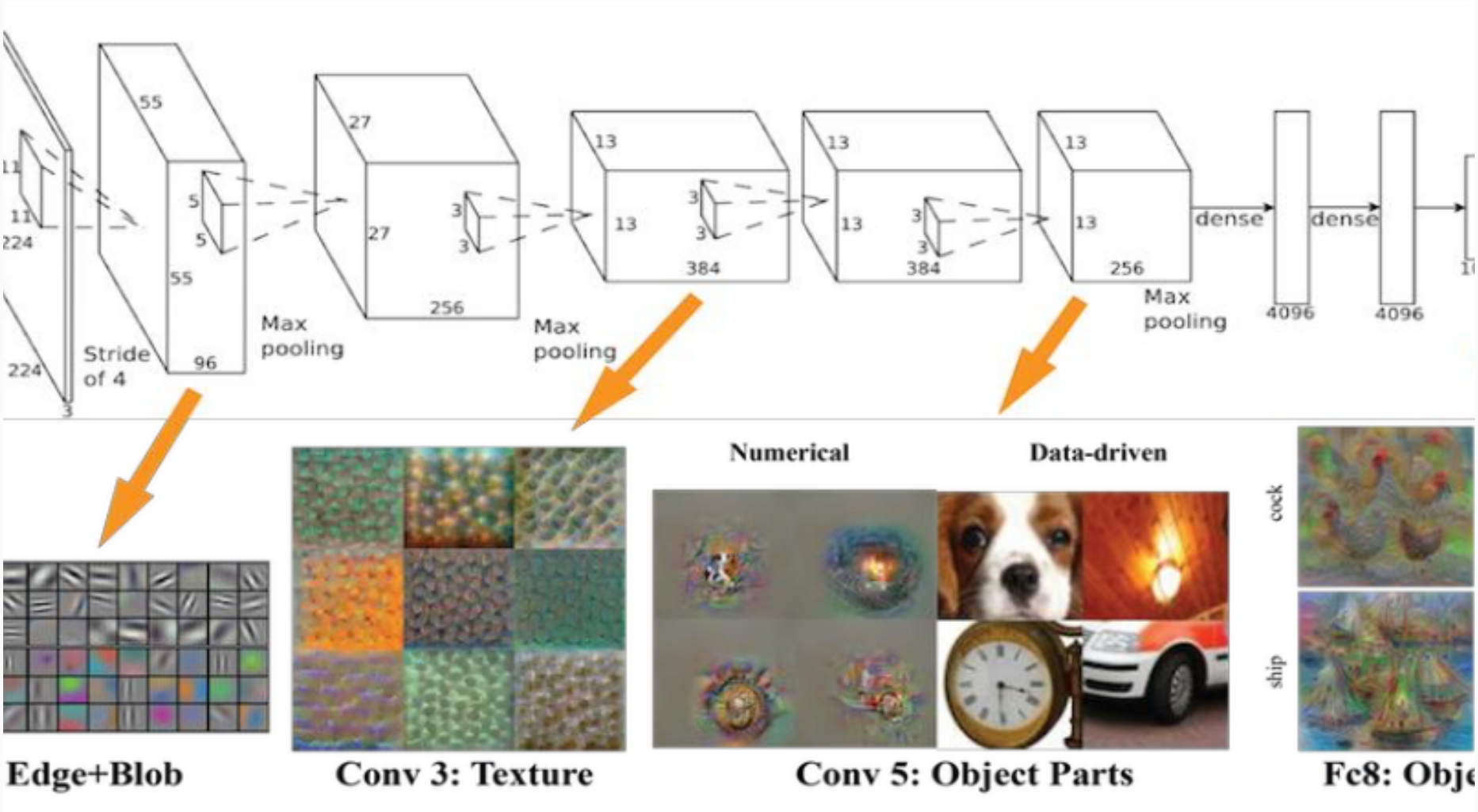
Visualization

Weight Visualization

from AlexNet



The 96 convolutional kernels of size 11x11x3 learned by the first convolutional layer on the 224x224x3 input images.


55
27
13
13
13
11
11
224
5
5
55
27
3
3
13
13
3
3
384
13
13
3
3
384
13
13
256
dense
dense
Max
pooling
4096
4096
224
Stride
of 4
96
Max
pooling
Max
pooling
3
Numerical
Data-driven
cock
ship
Edge+Blob
Conv 3: Texture
Conv 5: Object Parts
Fc8: Obje

Learned Features

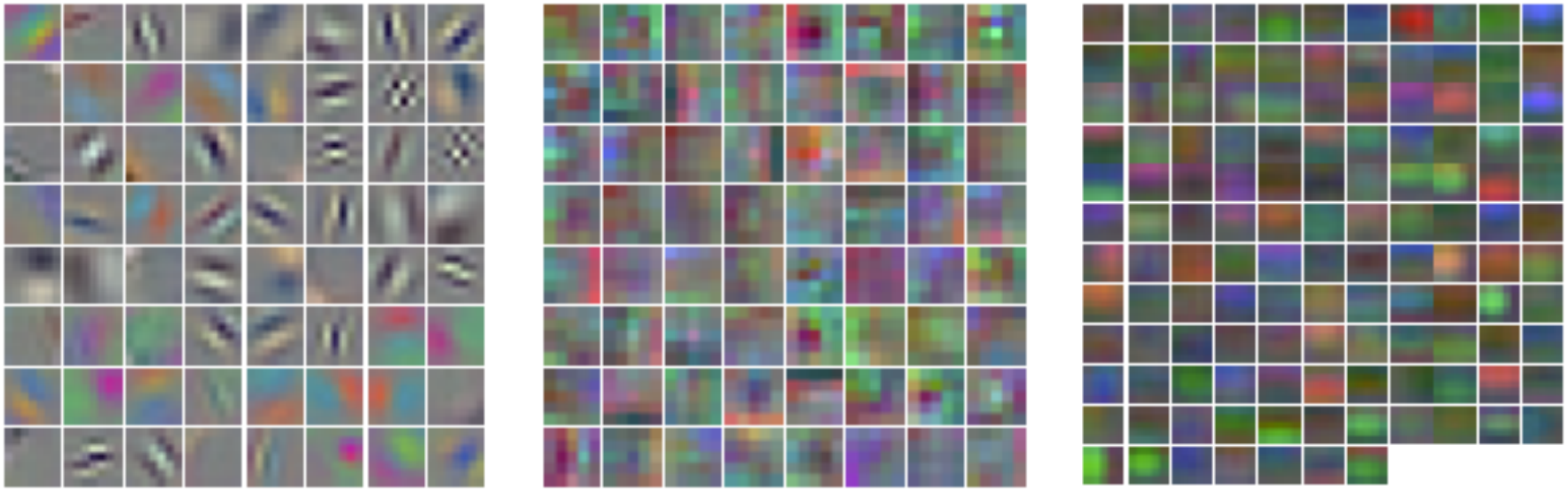
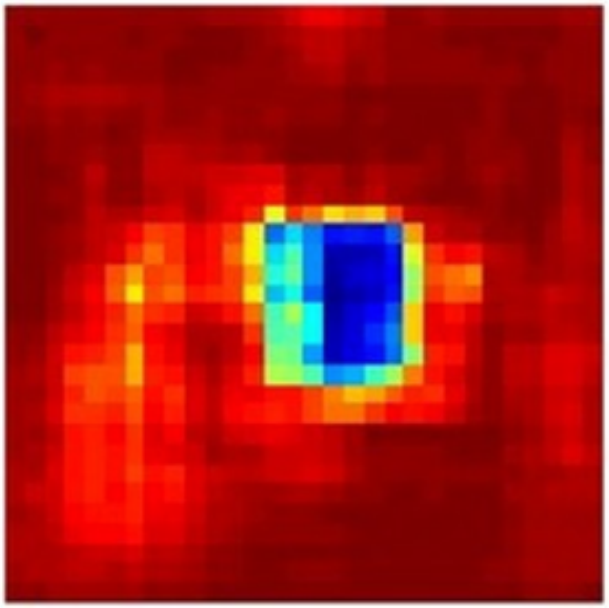


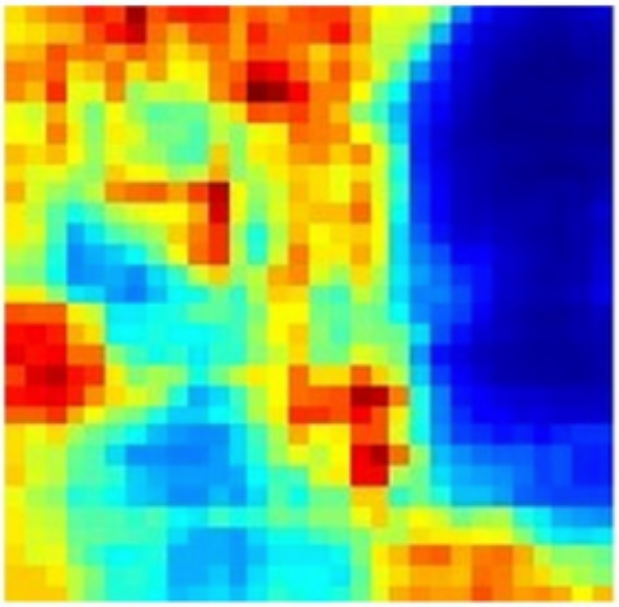
Figure 3: The network filter banks of the first (*Left*), middle (*Center*), and last (*Right*) layers, learned by fine-tuning GoogLeNet using the LifeCLEF 2015 plant dataset.

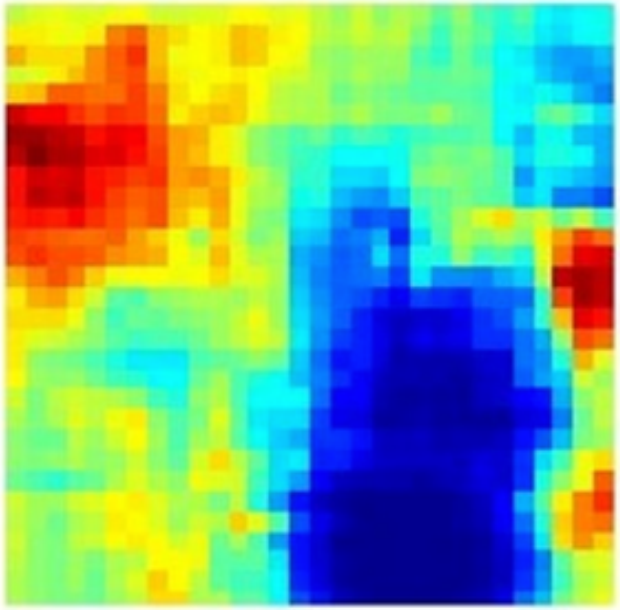

True Label: Pomeranian


DEKRA
8
DAC
True Label: Car Wheel


True Label: Afghan Hound



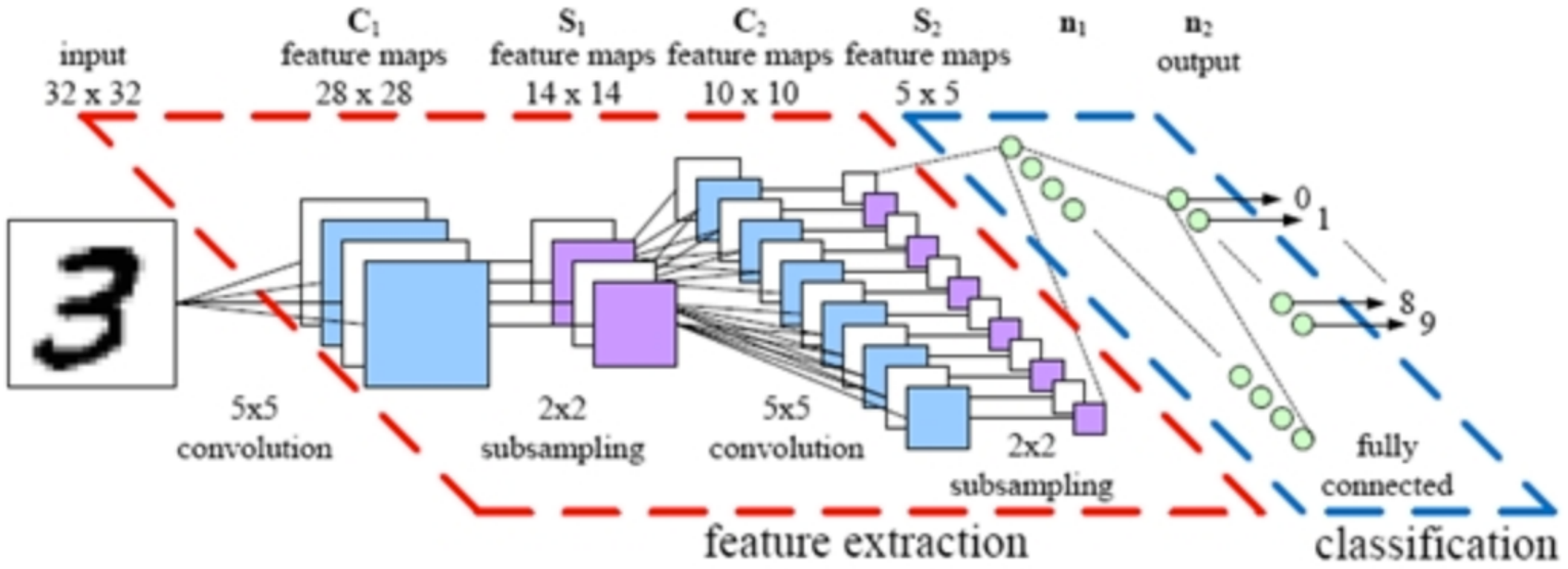




Easily Building
Deep learning Systems

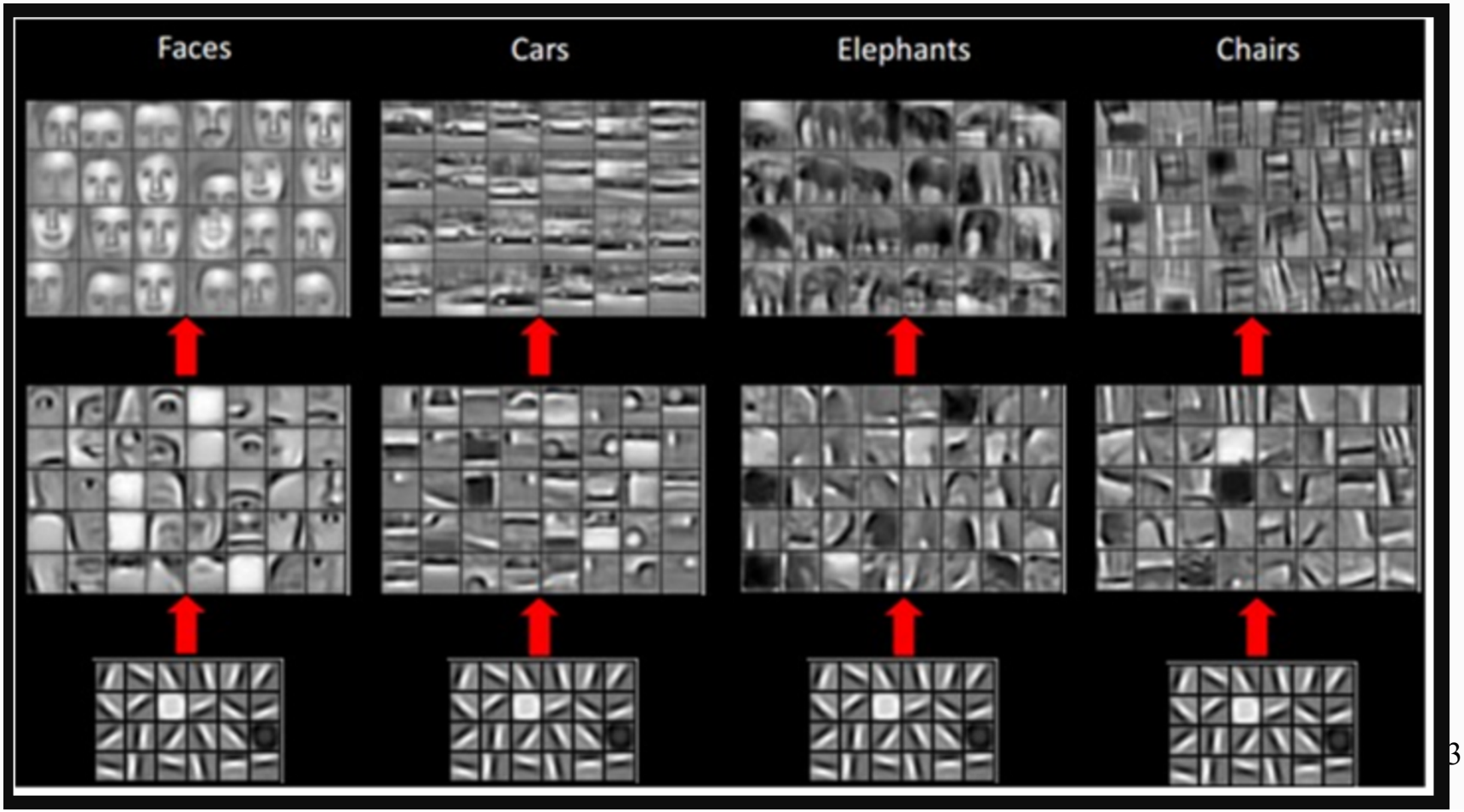
Training Deep Networks

- Train deep neural networks from scratch: requires A LOT of data and resources
- Adapt a pre-trained deep network from a similar task domain – use as feature detector: provides reasonable performance even with modest data amounts and training times ☺
- Fine-tune: Change the last layer as needed and keep finetune some of the earlier layers' weights.

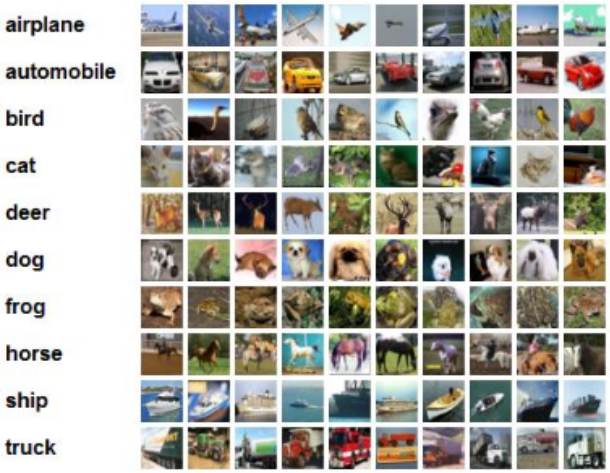


Transfer learning

- Instead of training a deep network from scratch, you can take a network trained on a different domain (e.g. ImageNet) for a different source task, and adapt it for your domain (e.g. plant recognition) and your target task



ImageNet:
1.2M training images
(~ 256 x 256px)
50k validation images
1000 classes



CIFAR-10:
50k training images
(32x32 px)
10k validation images
10 classes

Choosing a net to fine-tune..

- GoogLeNet
  - Winner of ILSVRC 2014, 6.8 million parameters

  Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015)

- VGGNet
  - Runner-up in ILSVRC 2014, 144 million parameters

  Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computing Research Repository (CoRR) (2014) arXiv: 1409.1556.

- Also frequently used: ALEXNet, now Squeeze Network, ...

- Implementations in CAFFE, TensorFlow, Torch, Matlab, ...

How to fine-tune?

-> **How many iterations**: as many as you can afford.

-> **Data augmentation**: how many more samples are to be generated per input image?

-> **Batch size**: the number of input samples used during gradient calculation for weight update

+ learning rate, **weight decay**, **momentum** (typically initialized to default values)

The first 3 affect training time linearly!

Data Augmentation

- Training data is augmented with artificial samples, through translation, rotation, scale, reflection, elastic deformations, intensity variations..., in order to obtain more robust systems.
  - Data augmentation is done internally within CAFFE and can be supplemented as well.
  - Data augmentation is done to increase train data size by 10xfold or more and is very effective in reducing overfitting.

Applications

ImageNet (ILSVRC)

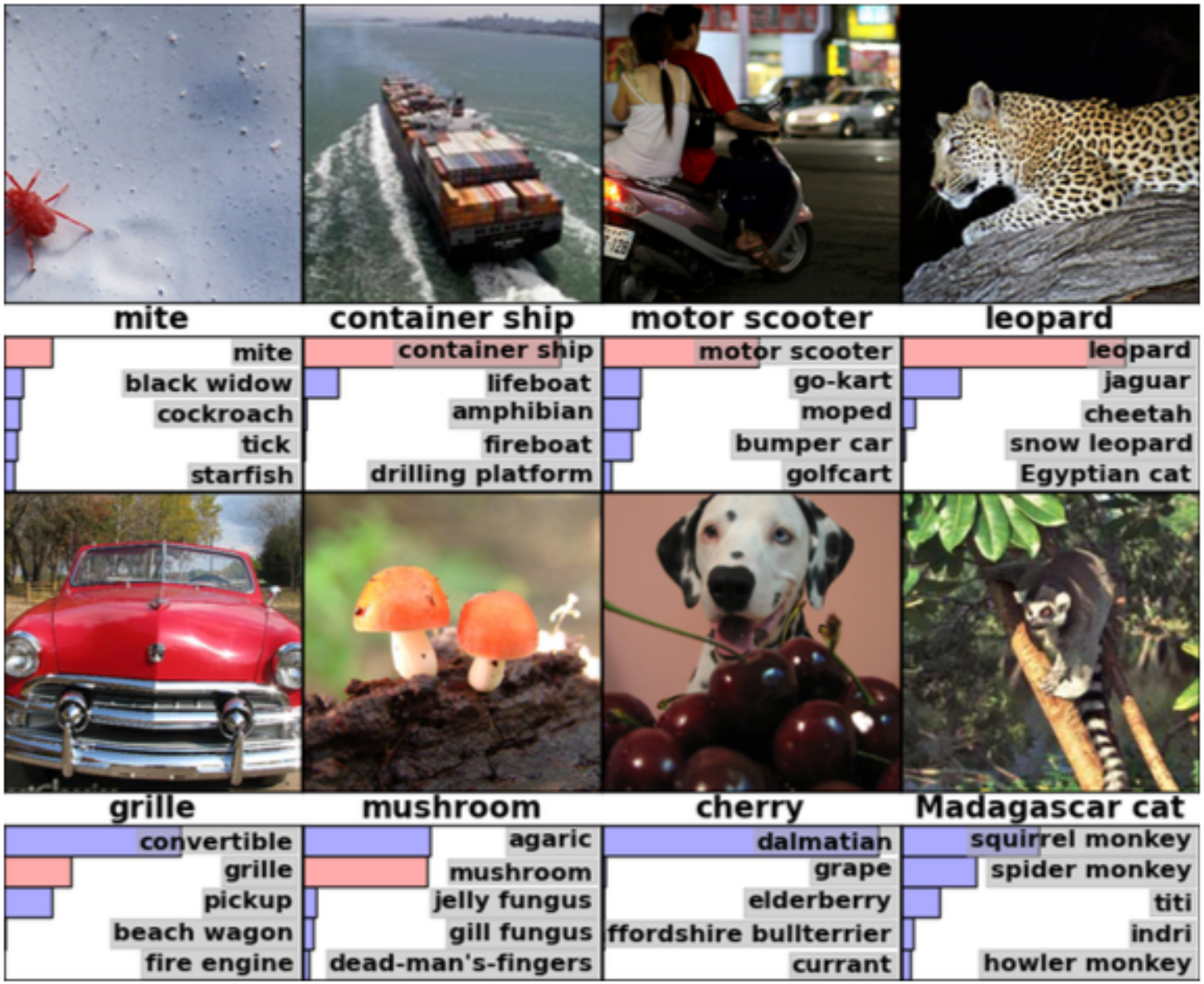
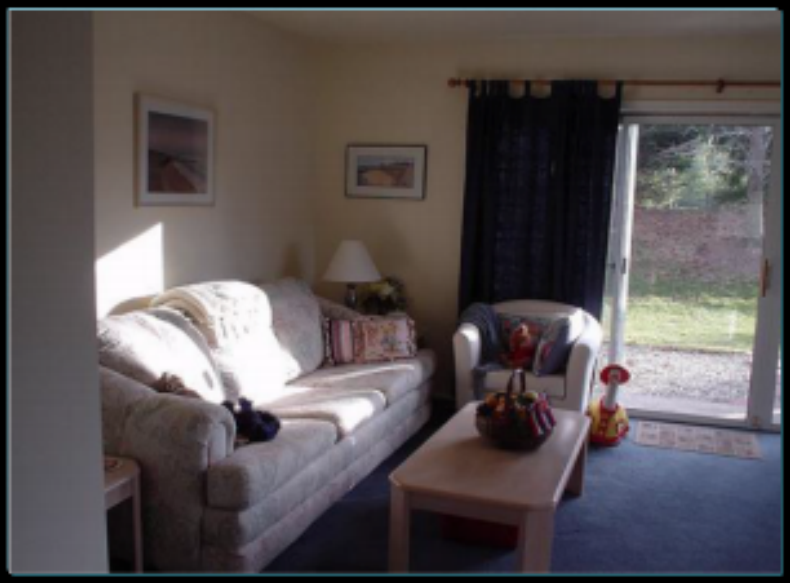


Image Captioning

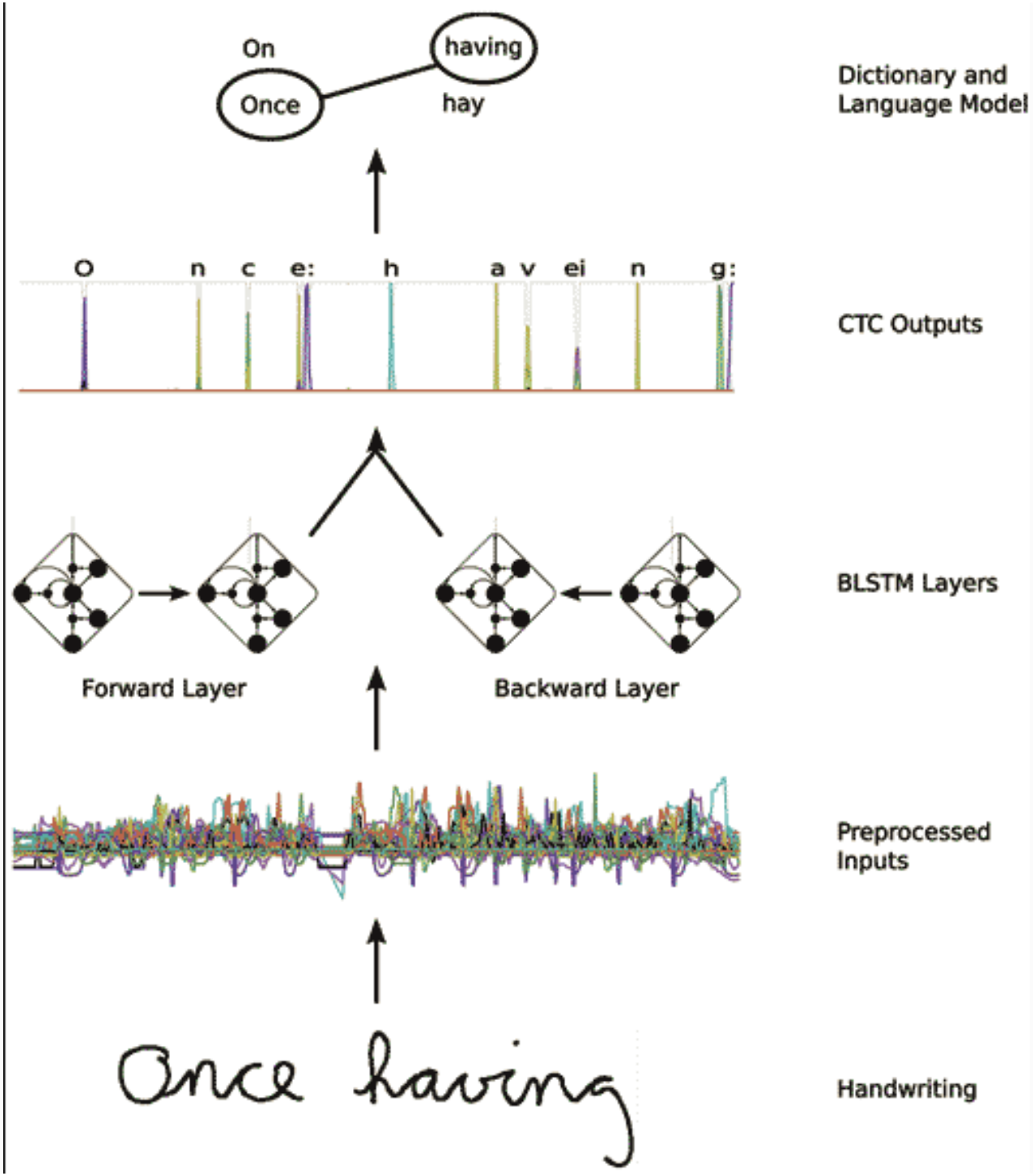
A yellow bus driving down a road with green trees and green grass in the background.



Living room with white couch and blue carpeting. The room in the apartment gets some afternoon sun.

Many images thanks to Andrew Ng.

Face Verification


On
having
Once
hay
Dictionary and
Language Model
O n c e: h a v ei n g:
CTC Outputs
Forward Layer
Backward Layer
BLSTM Layers
Preprocessed
Inputs
Once having
Handwriting

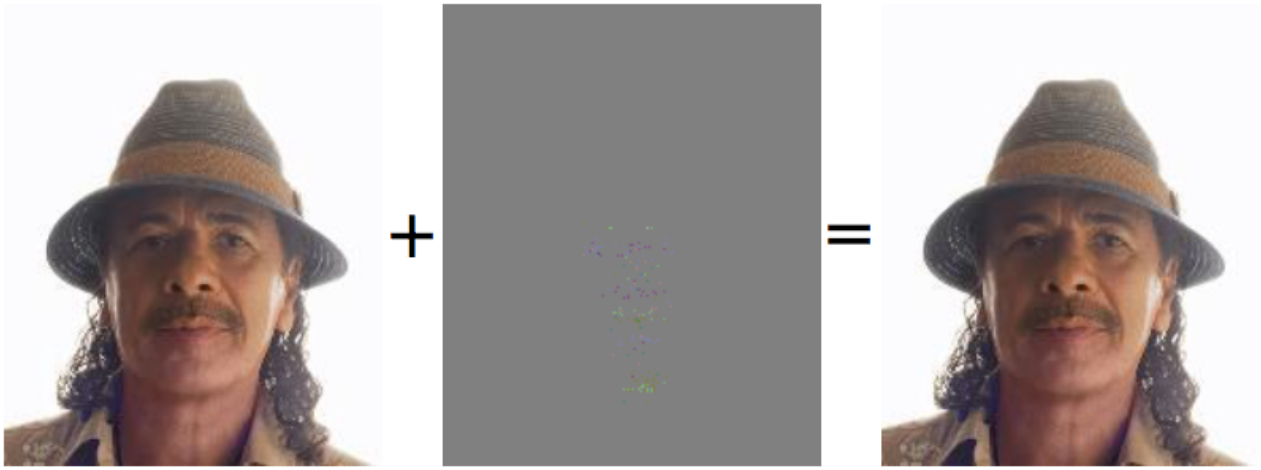
Shortcomings of Deep Learning SYSTEMS

- Are they really learning what we think they are learning (concepts) or are they just catching on some discriminative aspects?

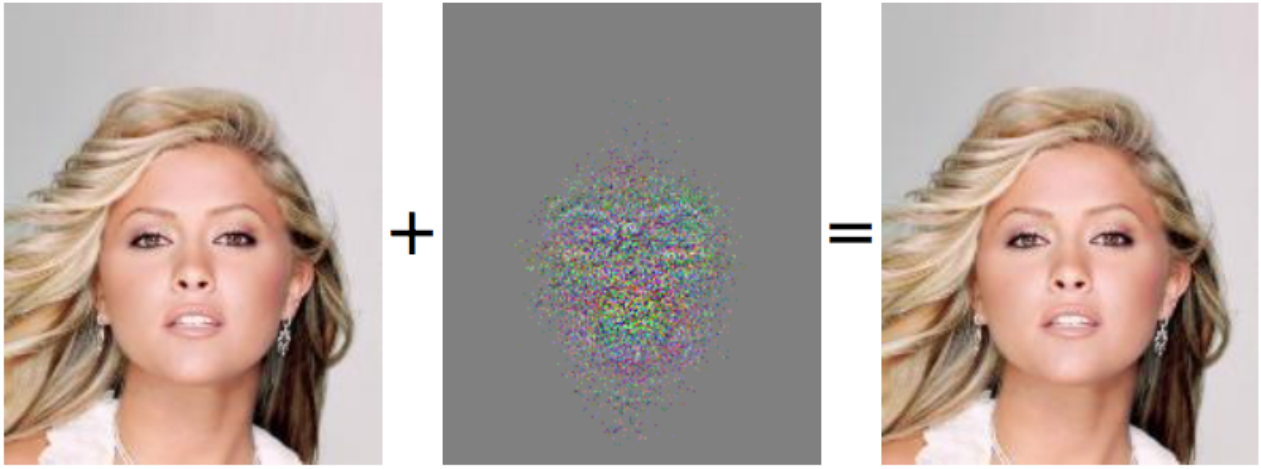
- Need for data!

- Humans can learn from a few examples. What is the next machine learning breakthrough?

Adversarial Examples



(a) Fixing a Natural Adversarial on Gender: from *female* to *male*



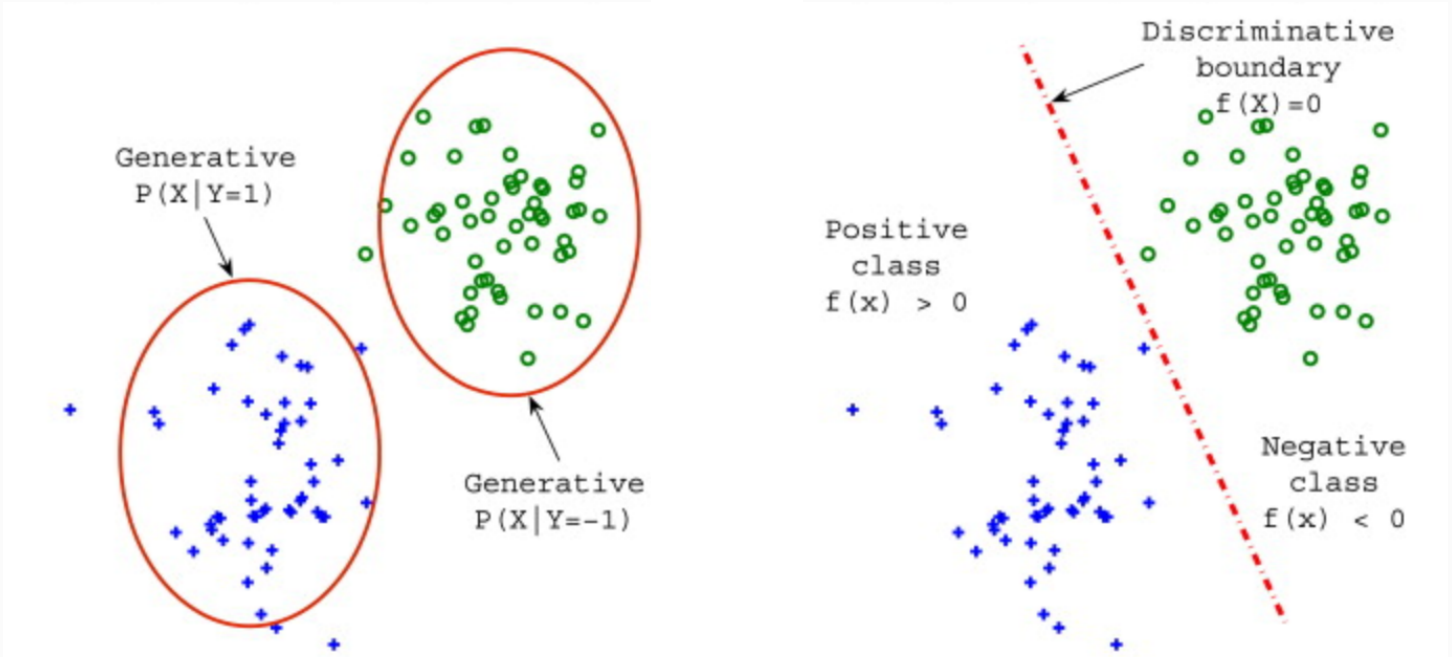
(b) Flipping Age: from *young* to *old*

Imperceptable perturbations made to an image can fool the system!

Generative Adversarial Networks (GANs)

Generative versus Discriminative Classifiers

- Goal: Wish to learn f: x ⟶ y
  - Discriminative classifiers (e.g. logistic regression, most NNs): Estimate P(Y|X) directly
  - Generative classifiers (e.g. Bayesian classifiers): Estimate P(X|Y) and P(Y) from data

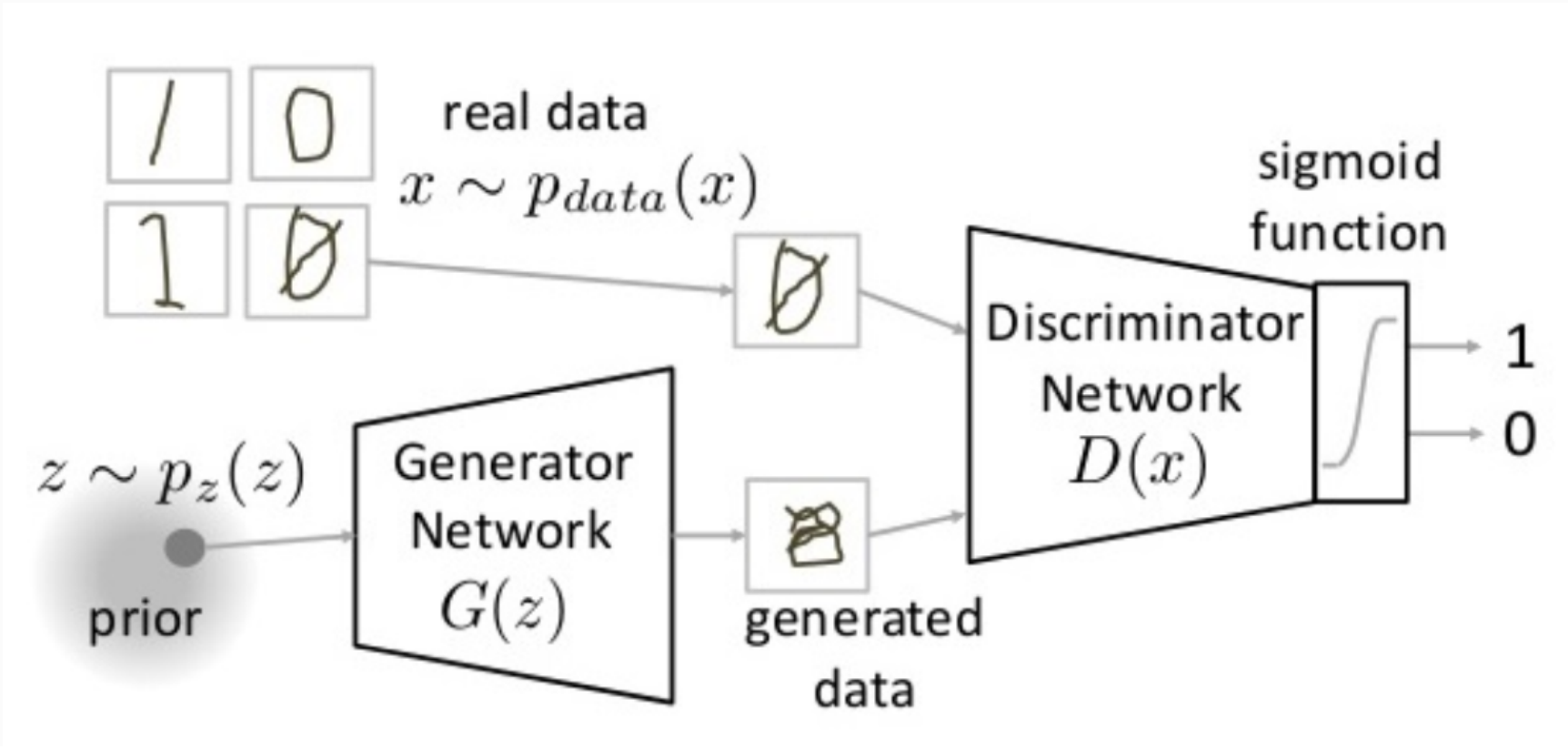


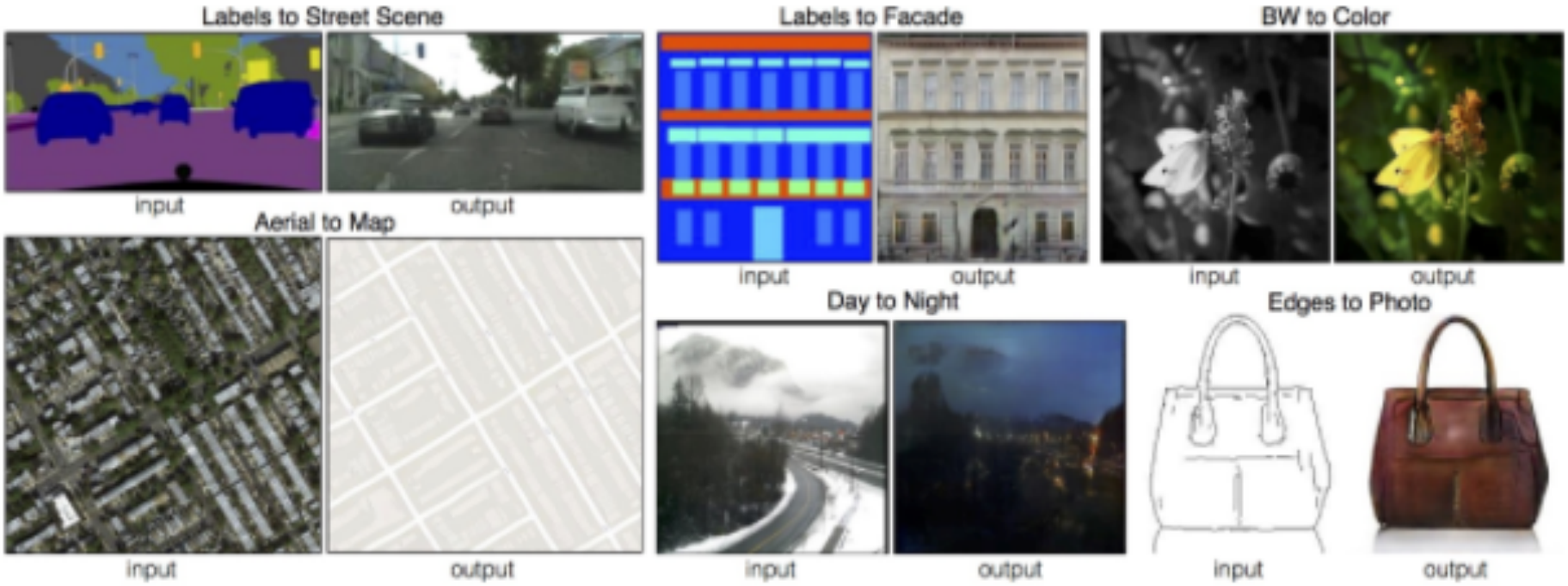
Generating Pokémon

By Yota Ishida

Generative Adversarial Networks

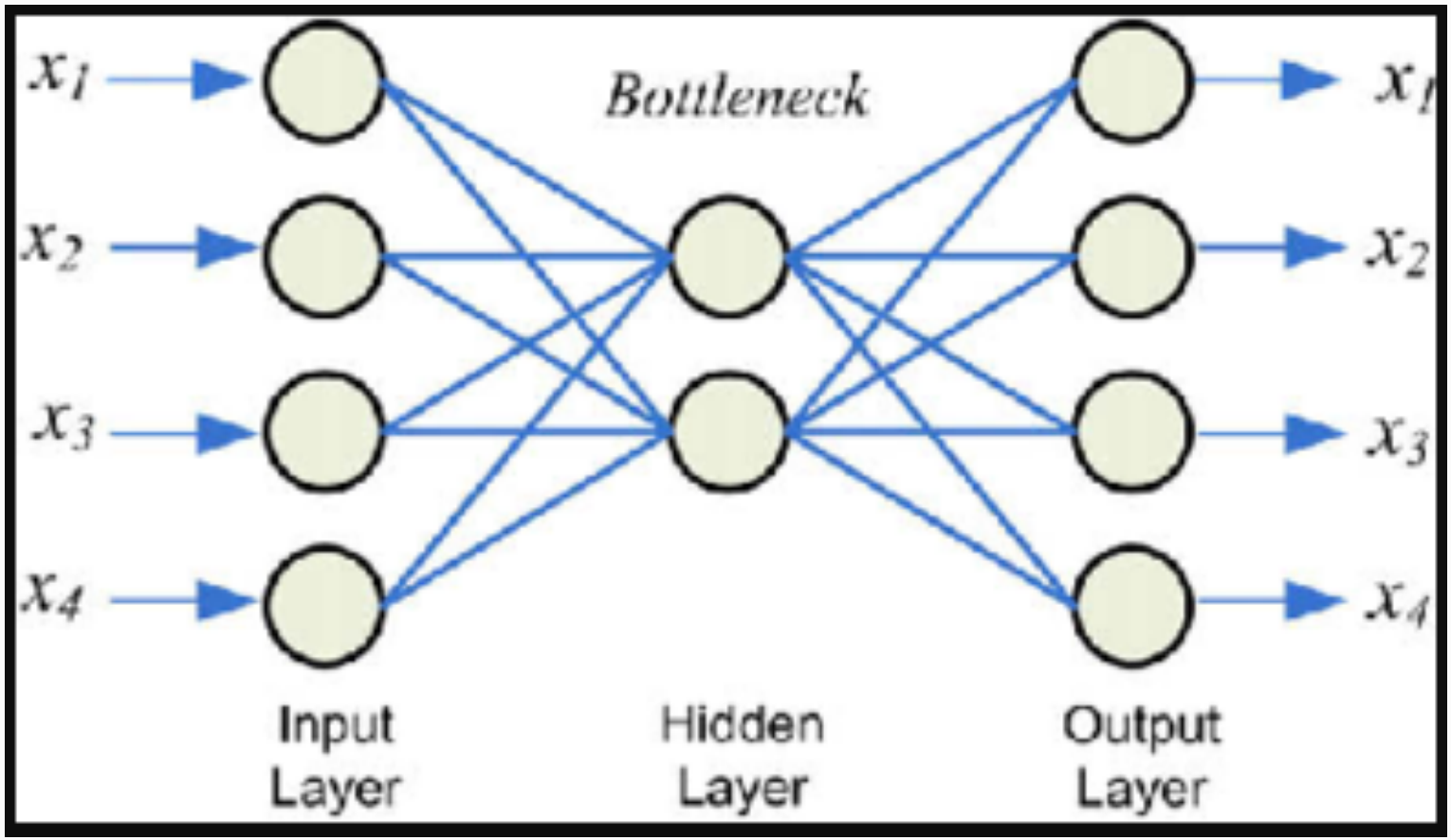
A generative model is learned with the ‘help’ of a discriminator that calls the generated samples as Fake/Genuine.




Labels to Street Scene
input
output
Aerial to Map
input
output
Labels to Facade
input
output
Day to Night
input
output
BW to Color
input
output
Edges to Photo
input
output

Other Exciting Work

AutoEncoders

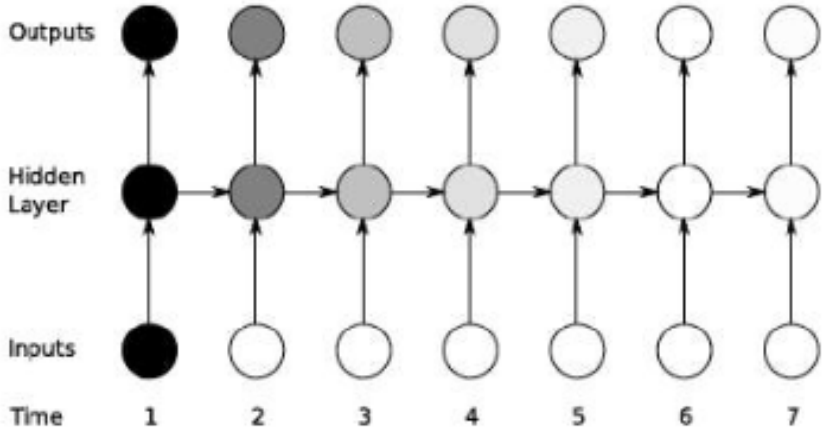


Recurrent NNs for Sequence Learning

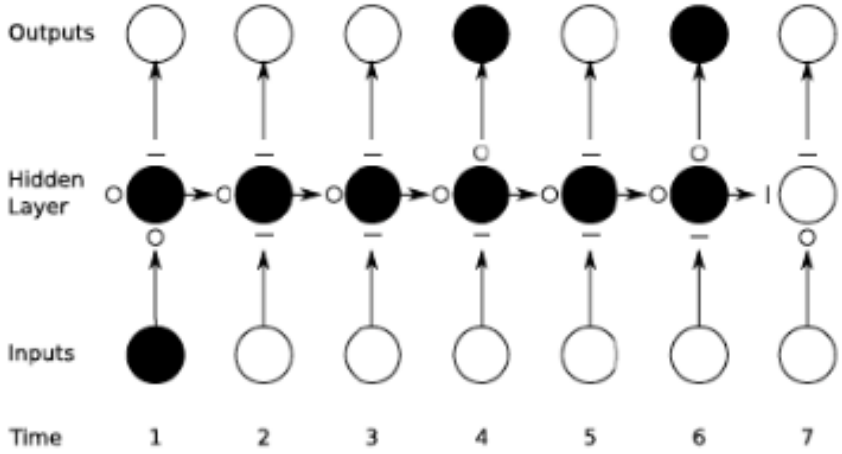
- Speech, handwriting recognition
- Natural language understanding

- New architectures address the problem of vanishing gradients in long term dependencies by explicitly introducing keep/forget gates.

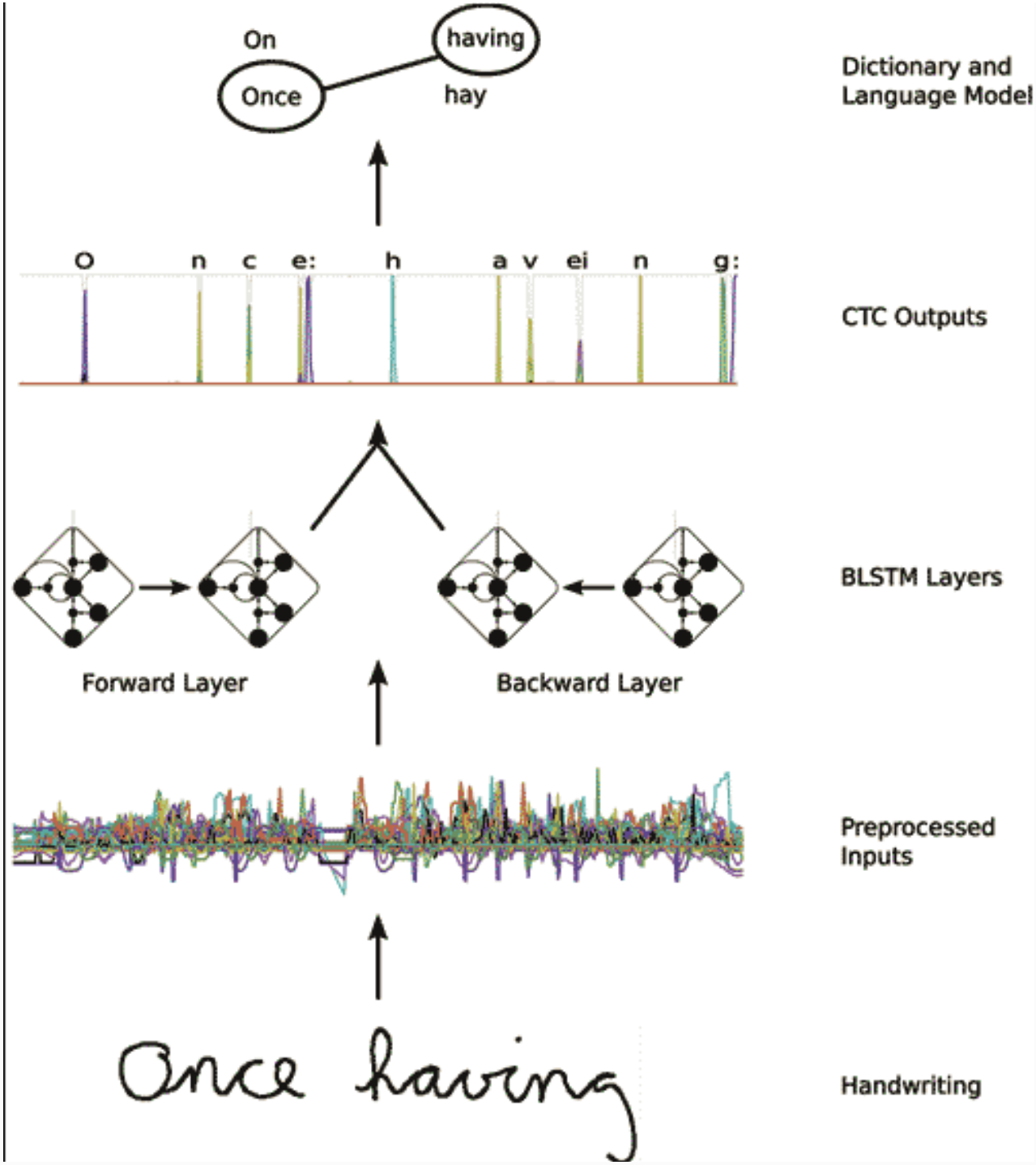
RNN



LSTM

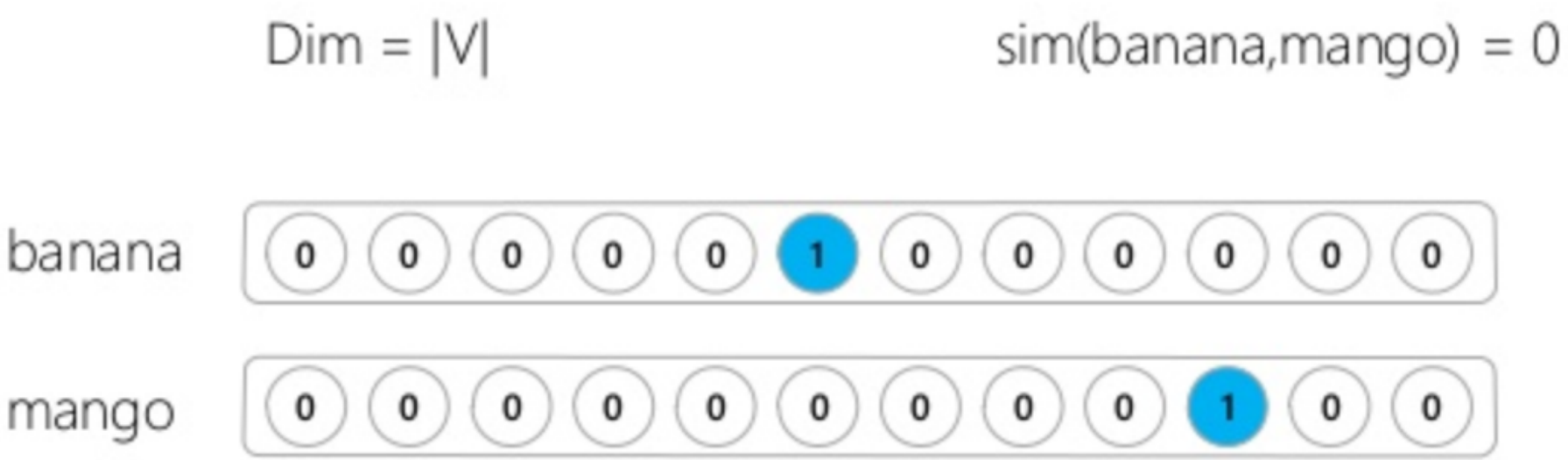


(Graves 2012)


On
having
Once
hay
Dictionary and
Language Model
O n c e: h a v ei n g:
CTC Outputs
Forward Layer
Backward Layer
BLSTM Layers
Preprocessed
Inputs
Once having
Handwriting

Word Embeddings

- In text understanding, NLP applications, words used to be represented as one-hot vectors
  - No relation between similar words,...



Word Embedding

Language Understanding(semantic)

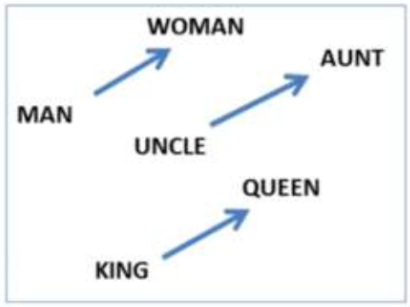
- **Word embedding $W: words \rightarrow \mathbb{R}^n$ function mapping to high-dimensional vectors**

one hot vector representation

dog	0 1 0 0 0 0 0 0 0 0
cat	0 0 1 0 0 0 0 0 0 0

Word Embedding

dog	0.3 0.2 0.1 0.5 0.7
cat	0.2 0.8 0.3 0.1 0.9



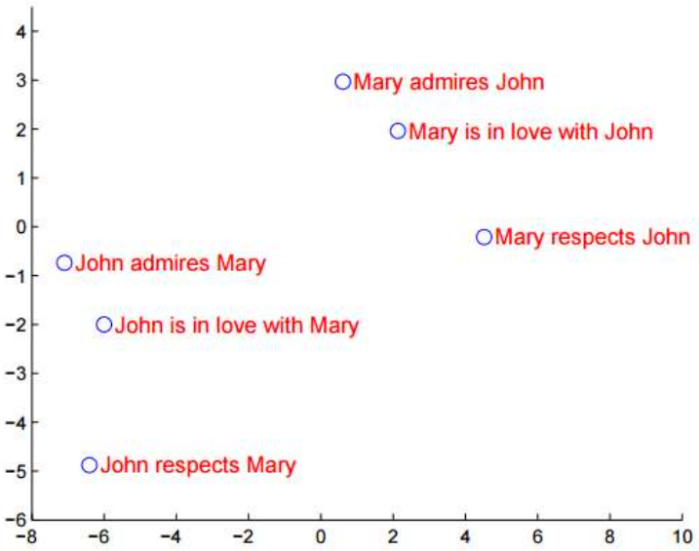
Word	Neighbors
car	van, cab, suv, vehicule, jeep
boy	toddler, gentleman, daughter, son
street	road, streets, highway, freeway
horse	pony, donkey, pig, goat, mule
computer	computers, pc, crt, chip, compute

Nearest neighbors a few words

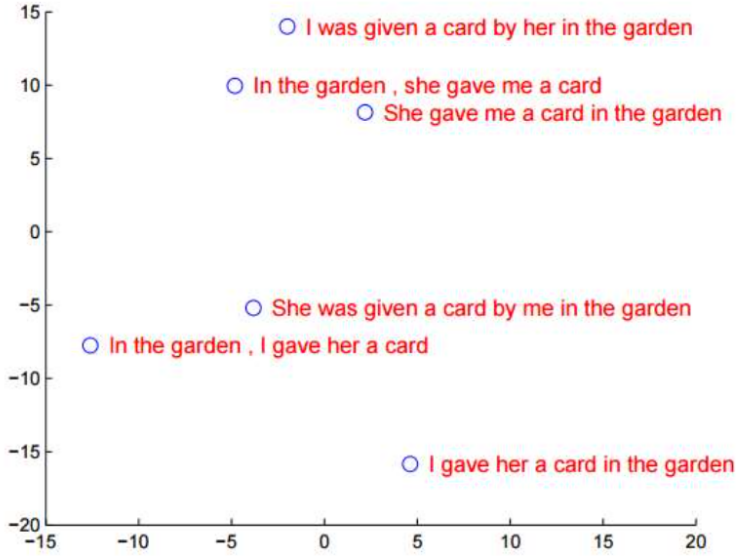
[Vinyals 14]

65

- **Sequence convert to sequence learning**
- **Sequence representation 1000D -> PCA 2D**



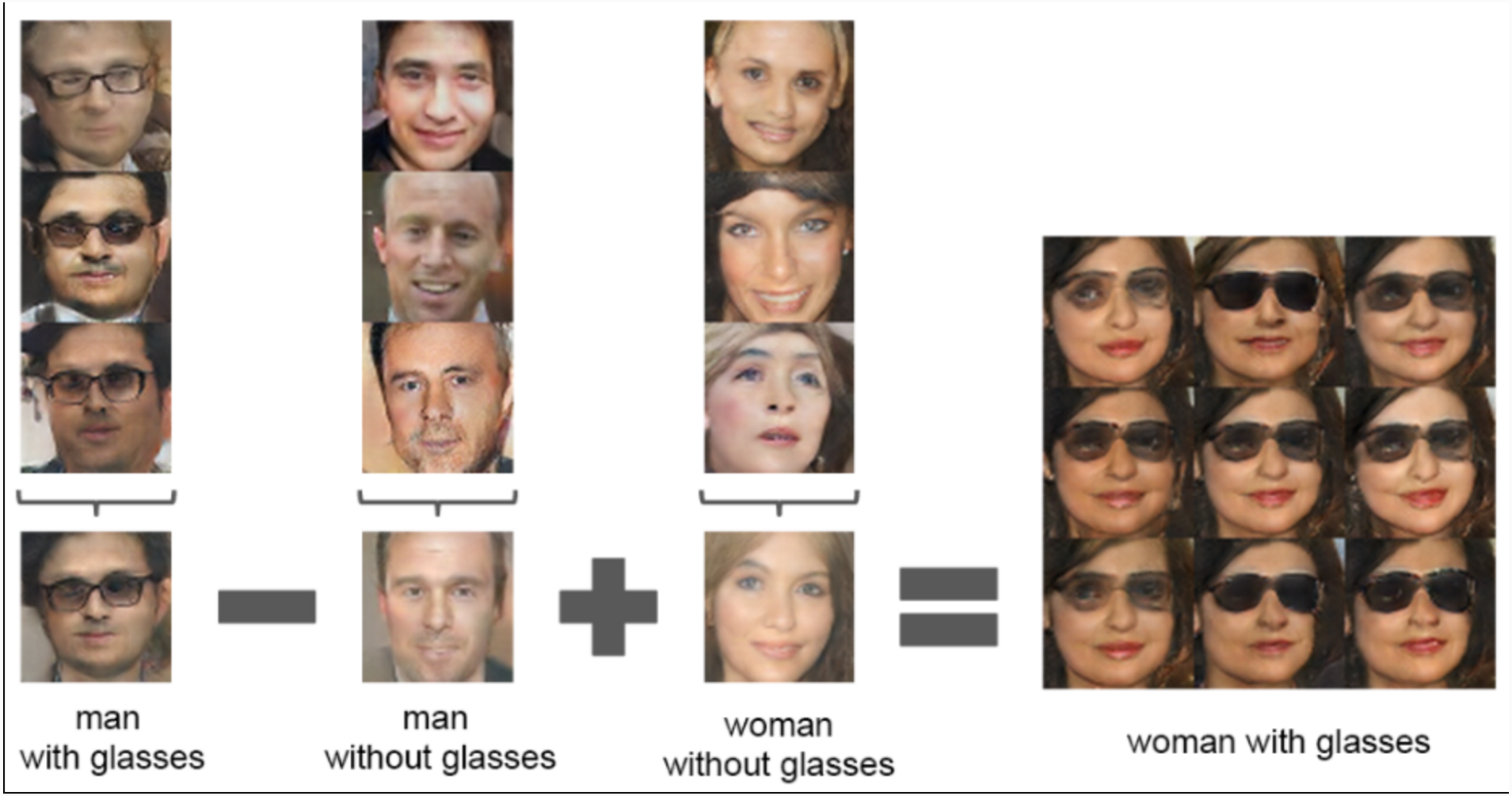
sensitive to word order



Invariant to active voice and passive voice

[Sutskever 14]

Multimodal Embeddings



AlphaGo - DeepMind

- **Further reading** (where some of the content and images are taken from):

- deeplearning.net
- http://cs231n.github.io/convolutional-networks/
- Papers by Hinton, Krizhevsky et al.
- http://alex.smola.org/drafts/thebook.pdf