Deep learning: an overview of datasets and algorithms
DL taskforce Caterpillar

Rob Romijnders

Eindhoven, University of Technology

RomijndersRob@gmail.com

May 26, 2017

Overview

1. Why
2. How
3. Datasets
4. What
5. Wrap up
6. Questions

Object detection

Faster R-CNN

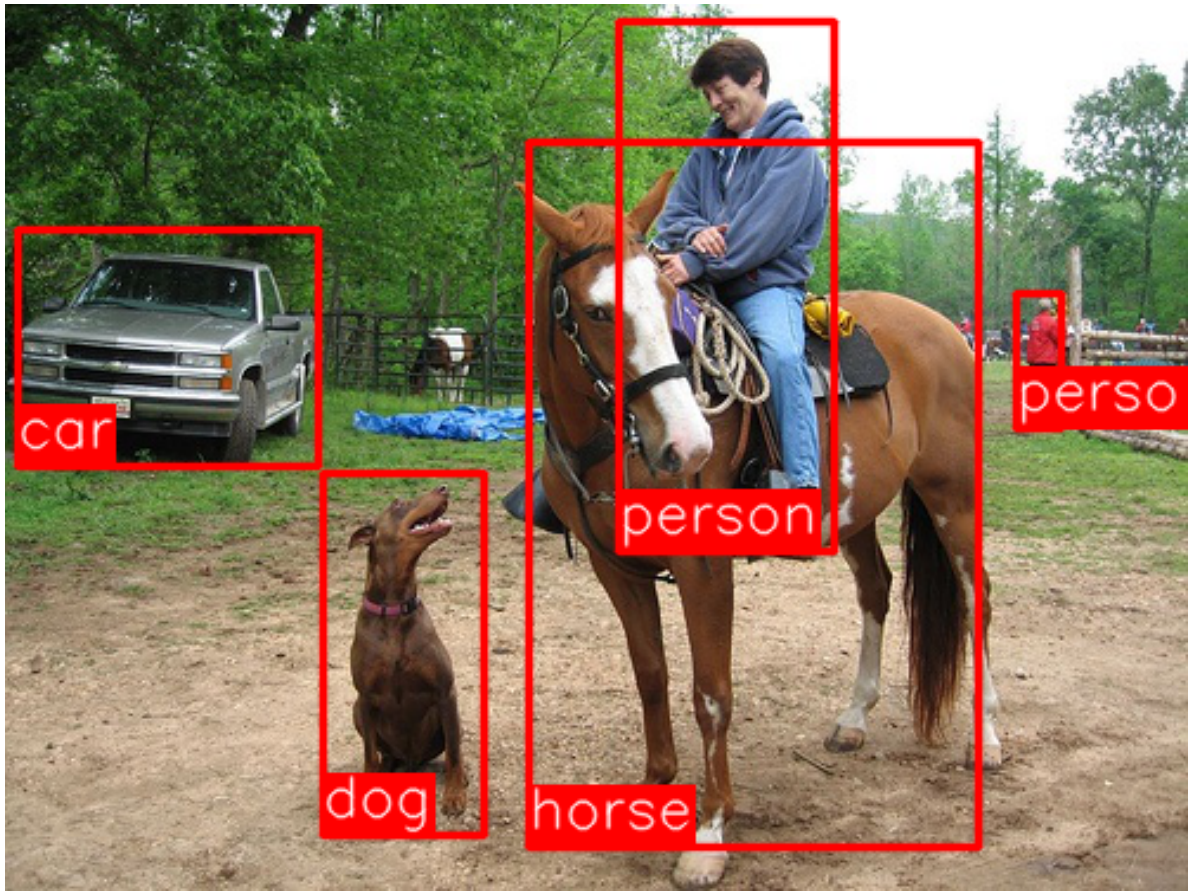


Figure: faster rcnn (image from Github *mitmul*)

Apple siri

Figure: Photo: cultofmac.com

GMail reply

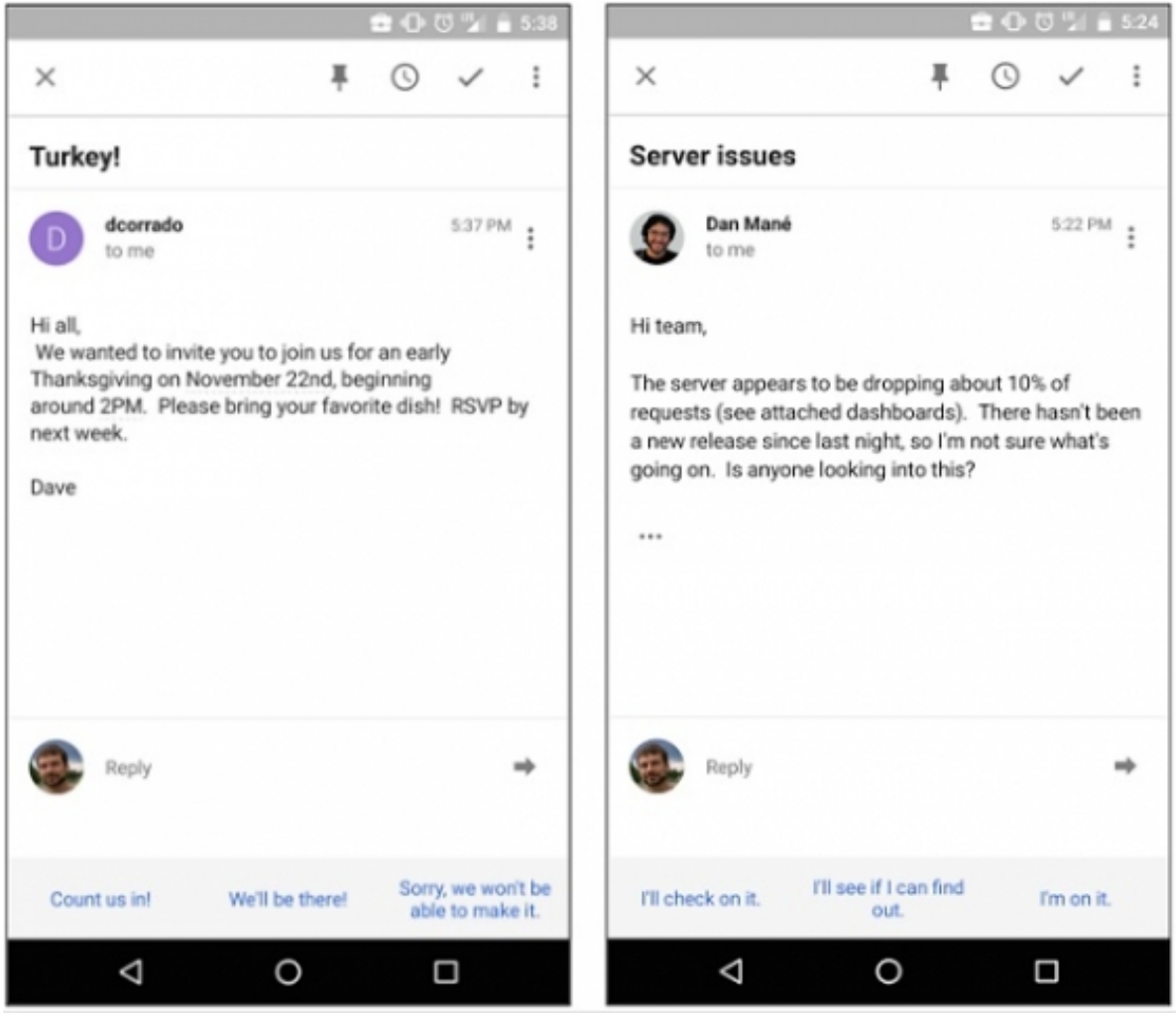


Figure: Photo: Greg Corrado, Google Research Blog

Segmentation

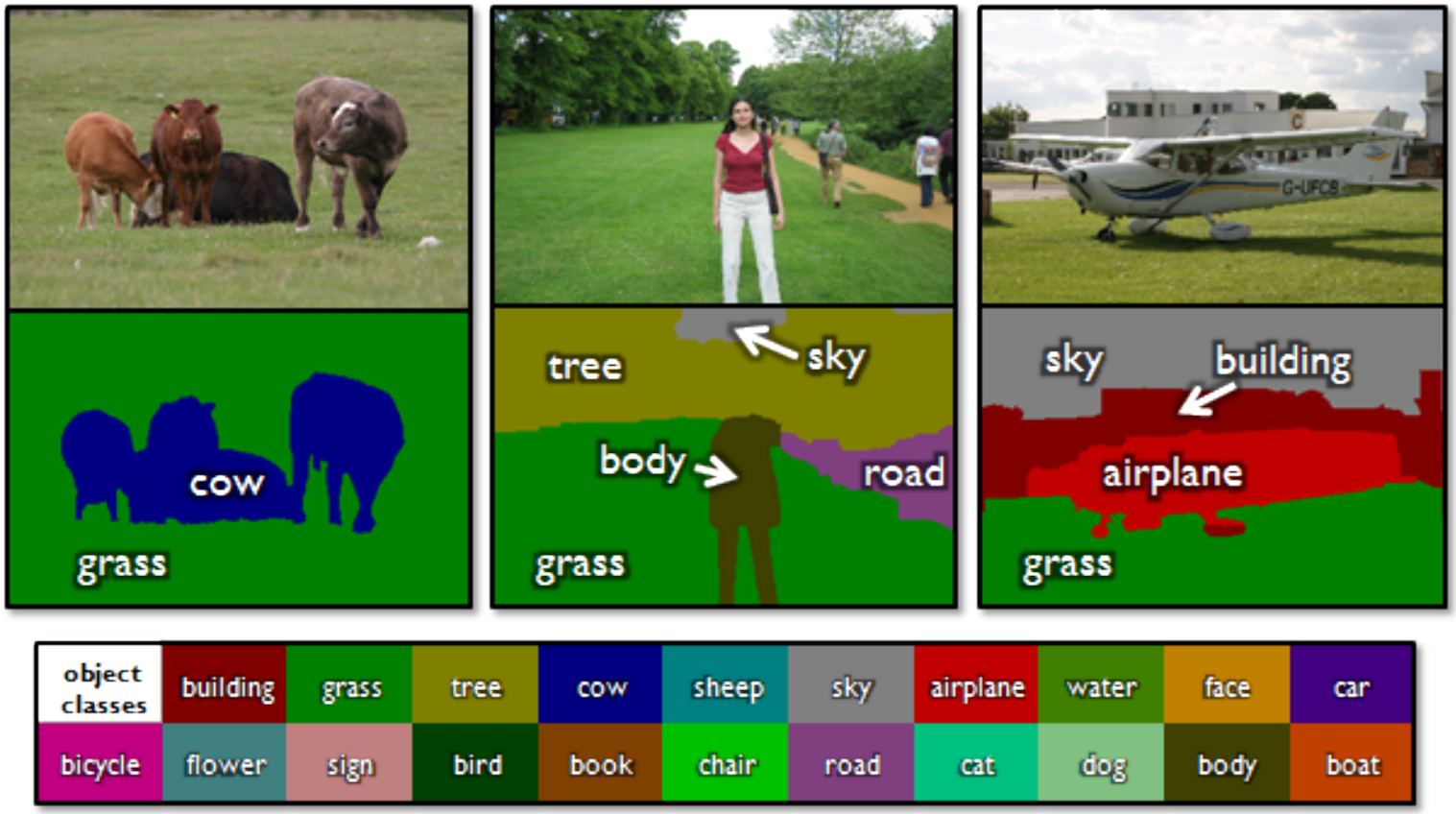
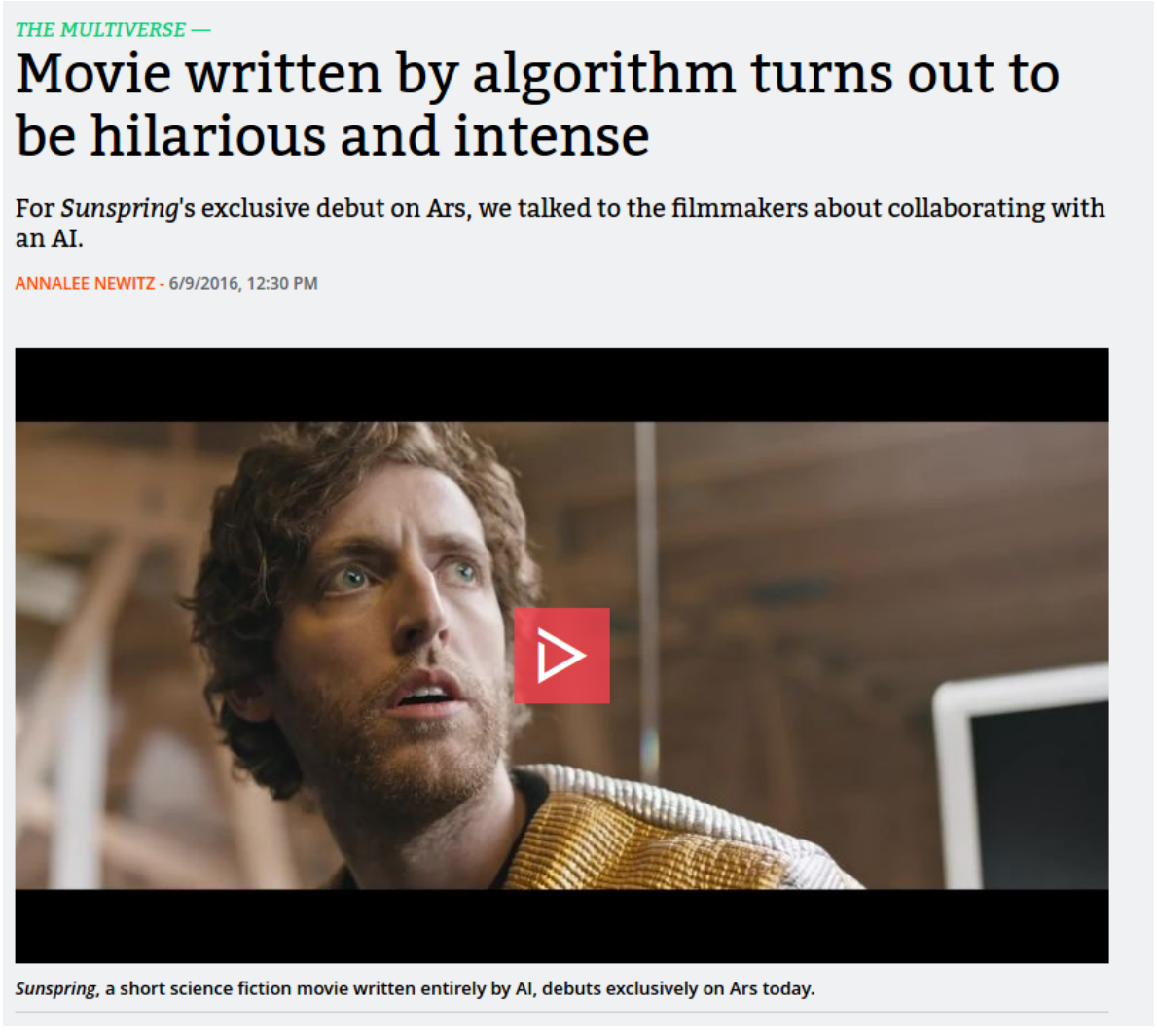


Figure: Semantic Segmentation with CNN (image from jamie.shotton.org)

Generation



THE MULTIVERSE —

Movie written by algorithm turns out to be hilarious and intense

For *Sunspring*'s exclusive debut on Ars, we talked to the filmmakers about collaborating with an AI.

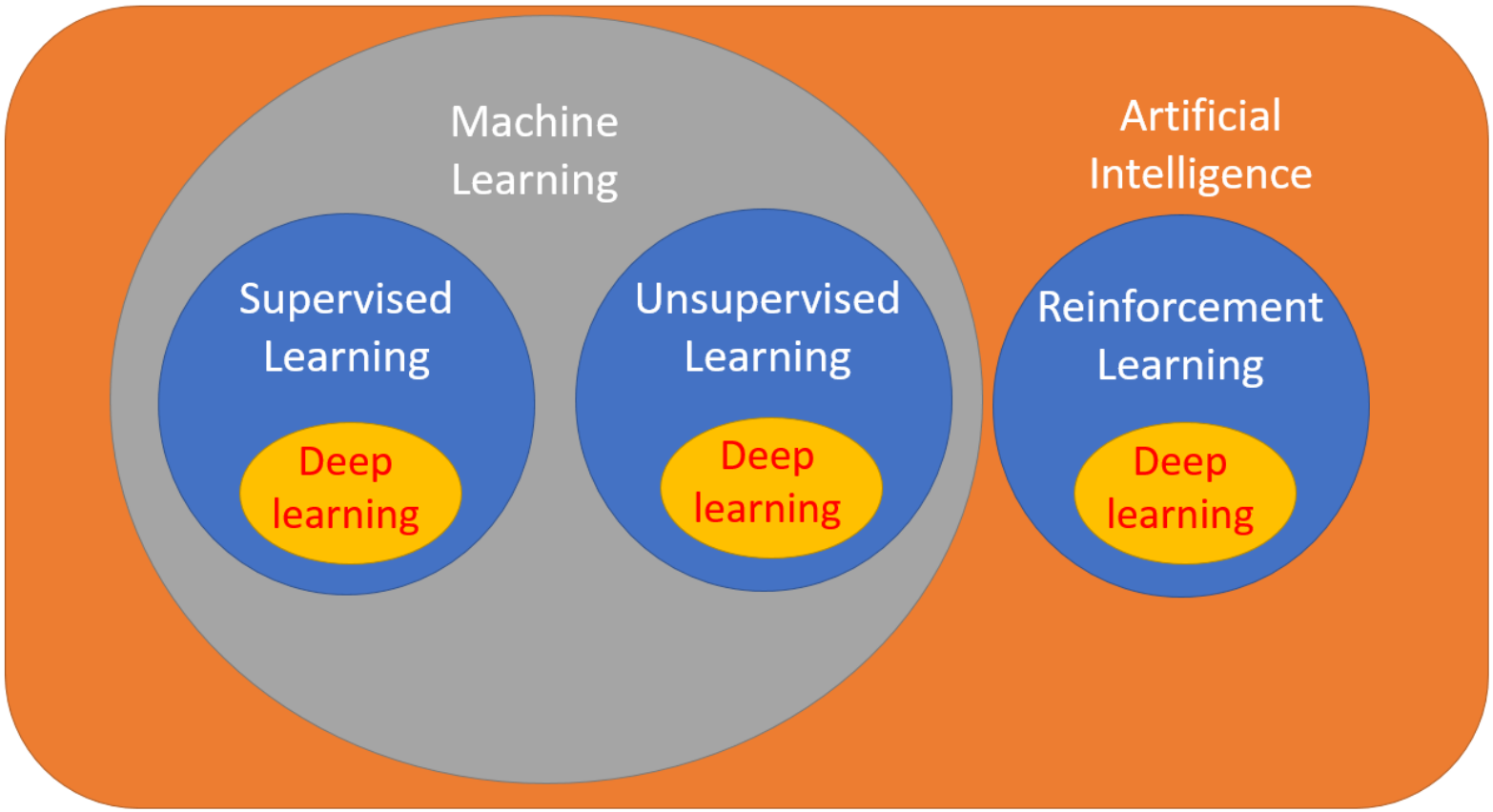
ANNALEE NEWITZ - 6/9/2016, 12:30 PM

Sunspring, a short science fiction movie written entirely by AI, debuts exclusively on Ars today.

Figure: http://arstechnica.com/the-multiverse/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/

Why not

Overview of the field



Which data not

1. YES: media type data
   1. Text, language, speech
   2. Images, video, maps
   3. Time-series, stocks, valuta
   4. Games, reward-like situations
   5. Robots, actuators
2. NO: categorical data
   1. Properties of instances
   2. Features of instances
   3. Categories of products

Neural Networks

Neural nets

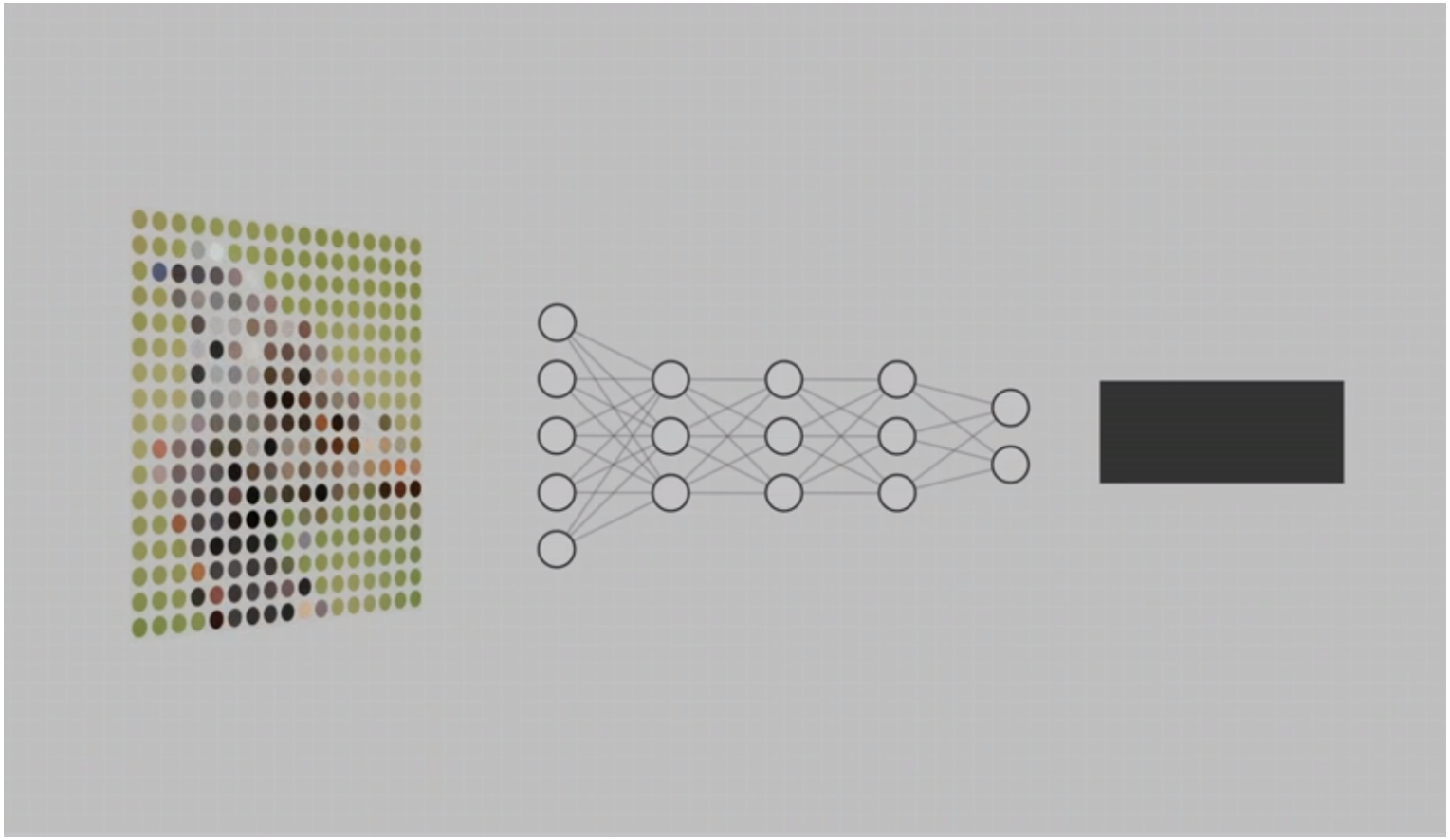


Figure: Neural network explained (credits: Blaise Aguera y Arcas)

Basic equation

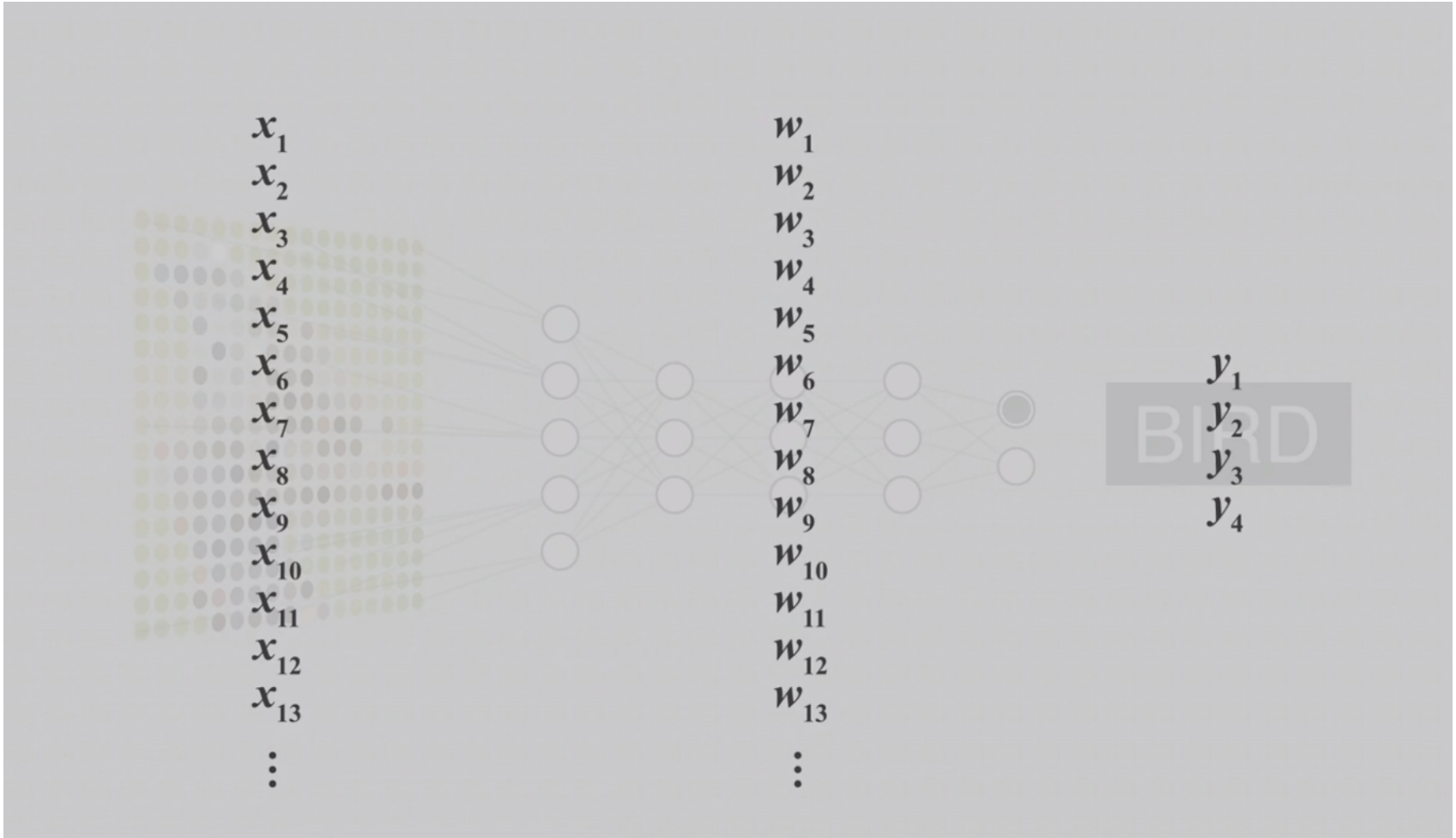


Figure: Neural net in algebraic form (credits: Blaise Aguera y Arcas)

Template equation

Figure: Template equation neural net (credits: Blaise Aguera y Arcas)

How to use

How to use them

$$w \cdot x = y$$

$$2 \cdot 3 = y$$

Forward inference

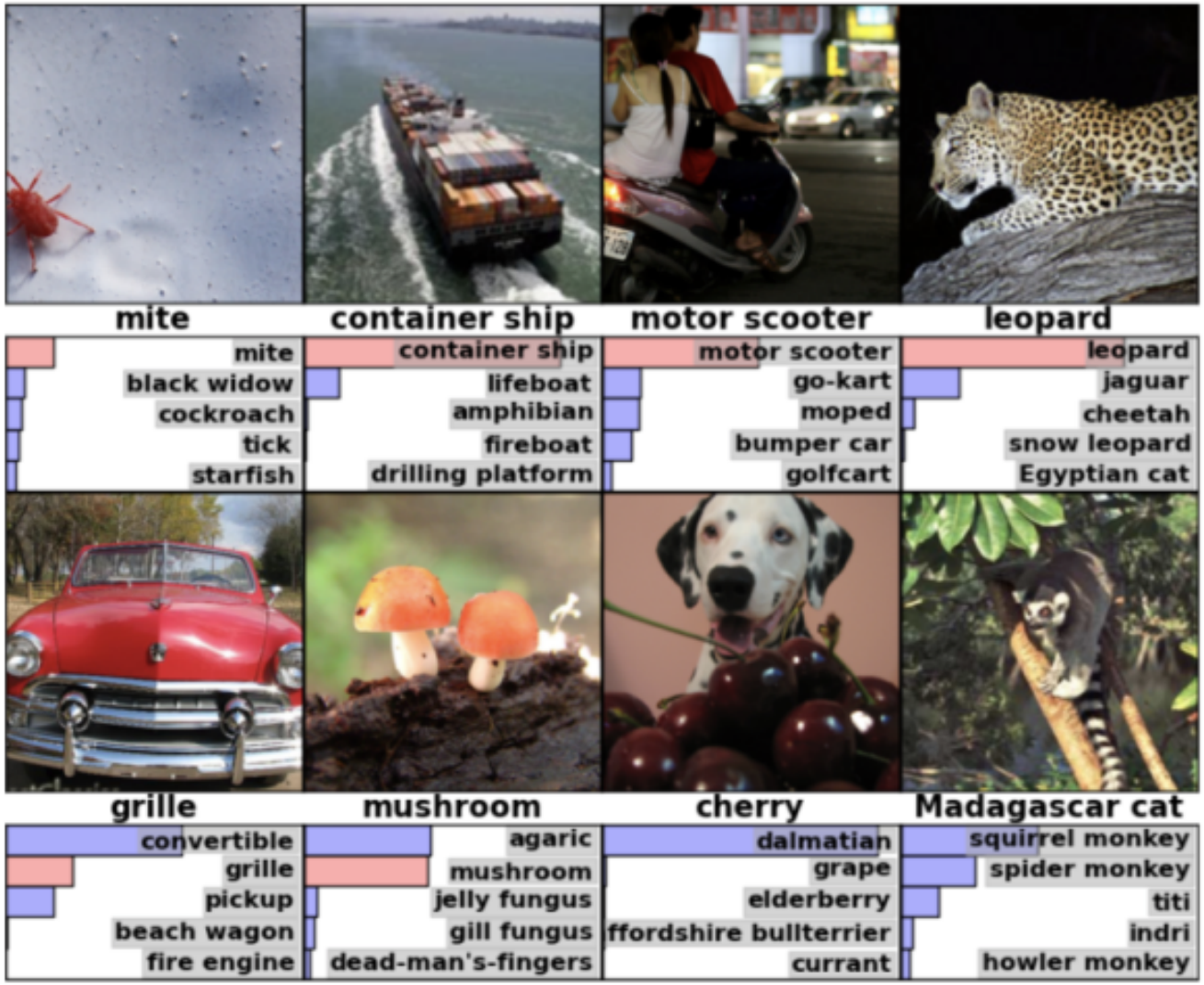


Figure: Forward inference CNN (Krizhevsky et al. 2012)

How to train

How to train them

$$w \cdot x = y$$

$$w \cdot 3 = 6$$

$$y \div x = w$$

$$0 = w \cdot x - y$$

$$0 = w \cdot 3 - 6$$

$$\text{error} = |w \cdot x - y| \Longrightarrow 0$$

Loss function

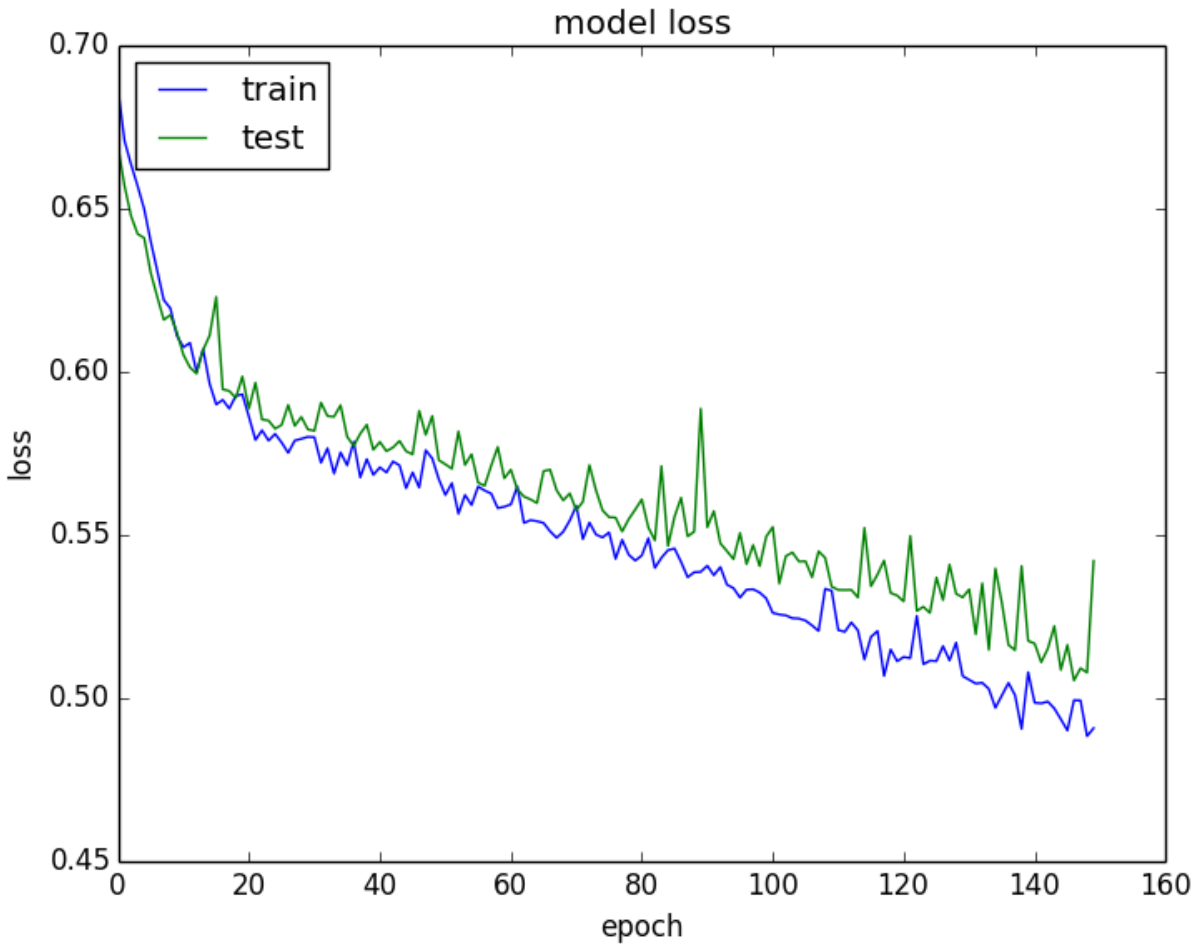


Figure: (http://machinelearningmastery.com/)

Architecture

Architecture

Architecture

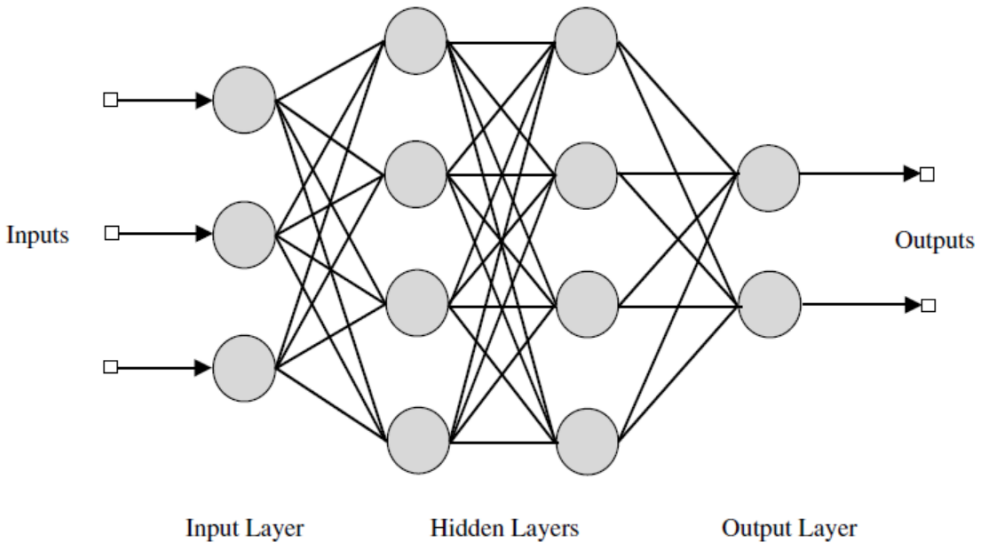


Figure: Feedforward neural network

$$y = w \cdot x$$

Two architectures

1. Convolution

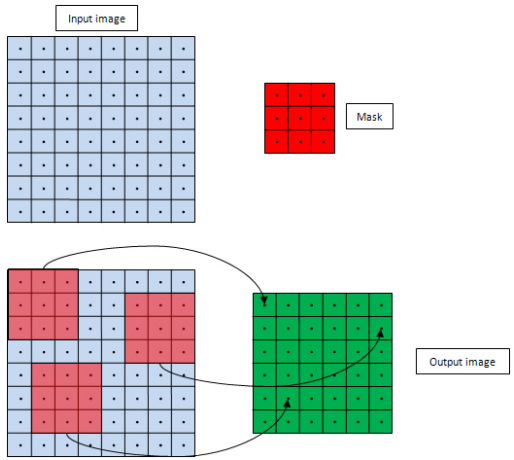


Figure: developer.amd.com

2. Recursion

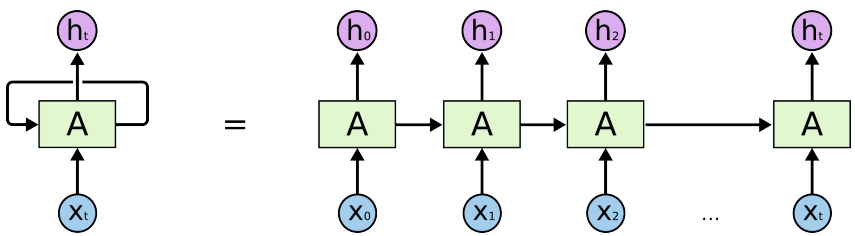


Figure: credits: colah.github.io

Convolution

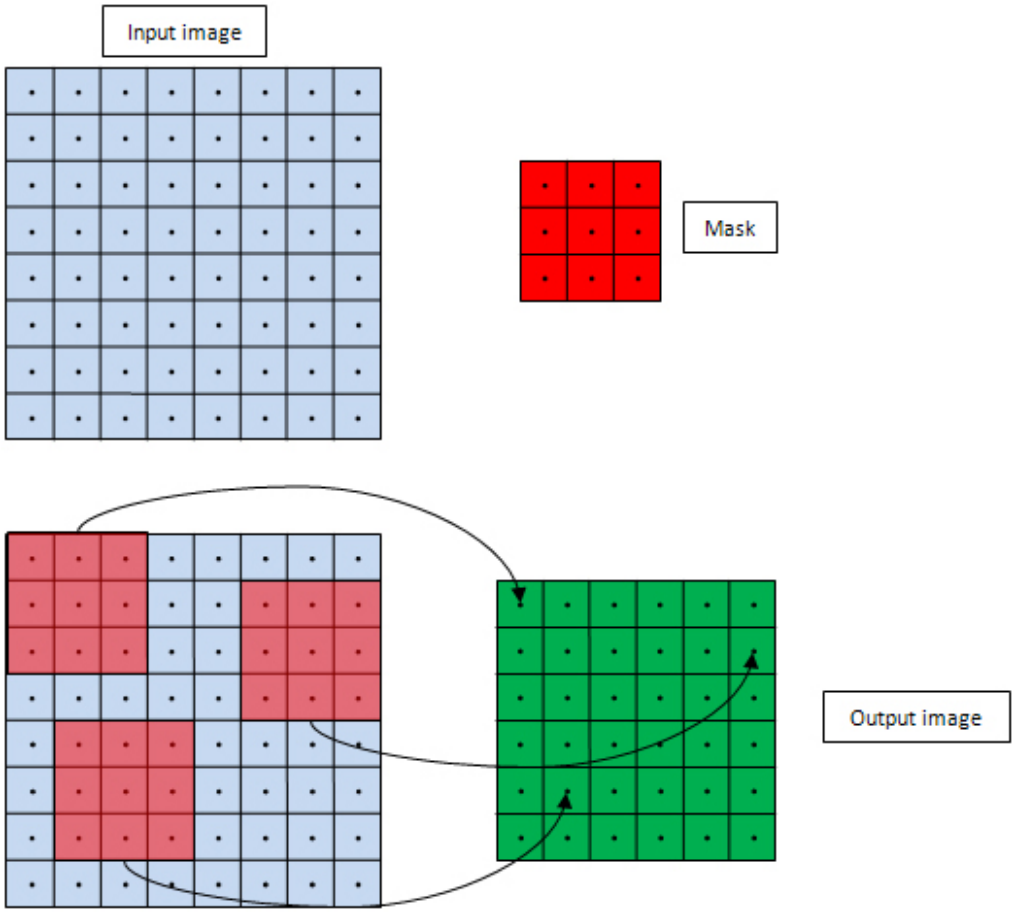


Figure: developer.amd.com

CNN

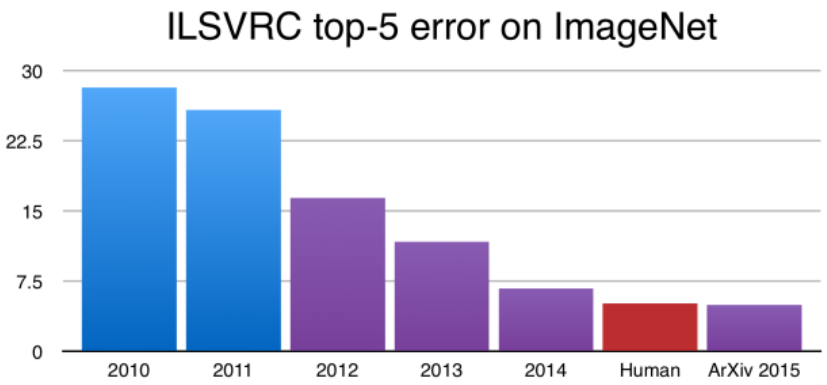


Figure: ImageNet results (credits jackkelly.github.io/)

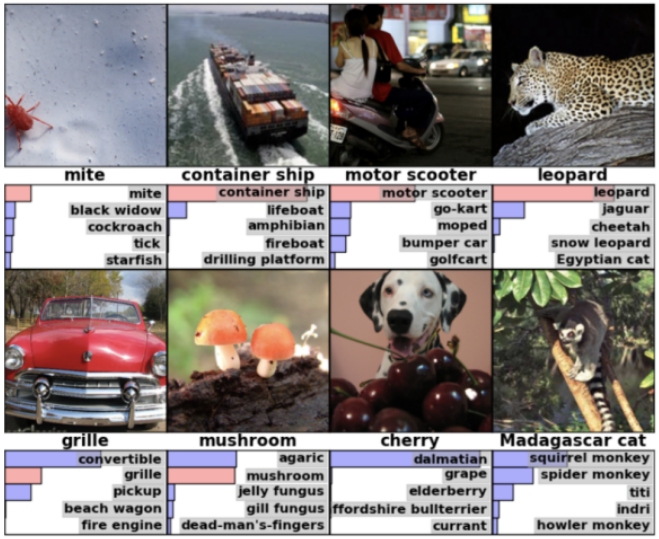


Figure: Forward inference CNN (Krizhevsky et al. 2012)

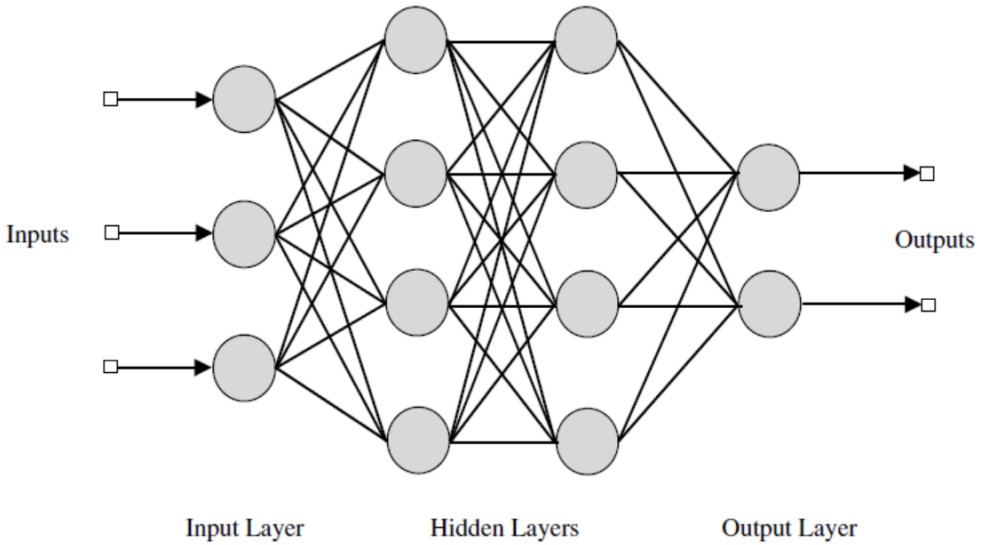


Figure: Feedforward neural netword

$$32 \cdot 32 \cdot 100 = 102400$$

$$5 \cdot 5 \cdot 5 \cdot 5 = 625$$

Recursion

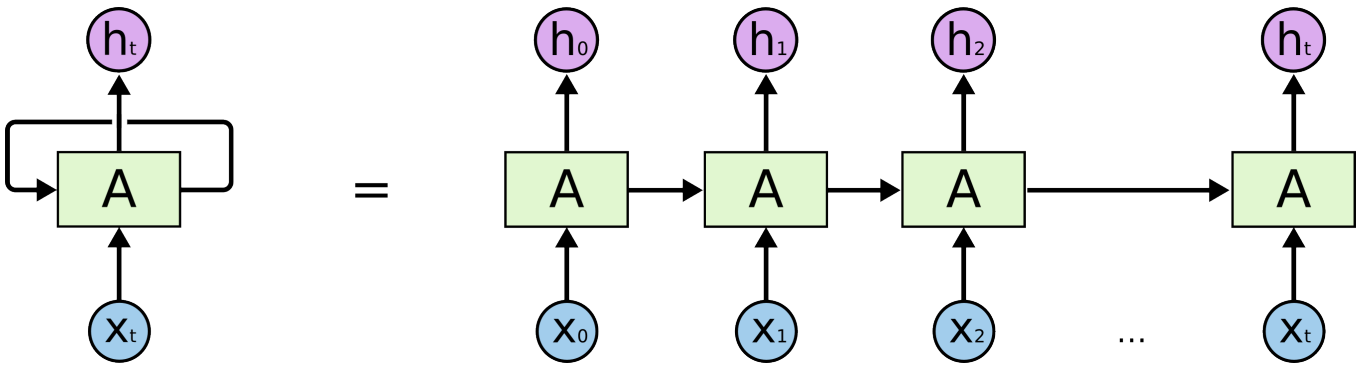
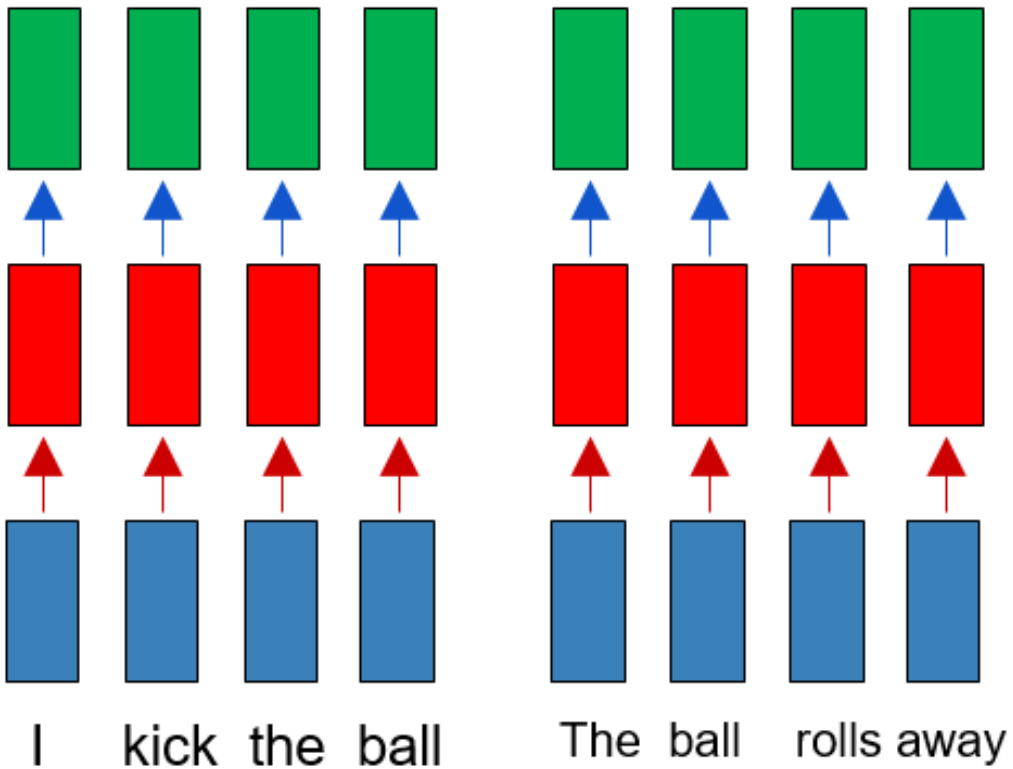


Figure: credits: colah.github.io

Word tagging



Connect over time

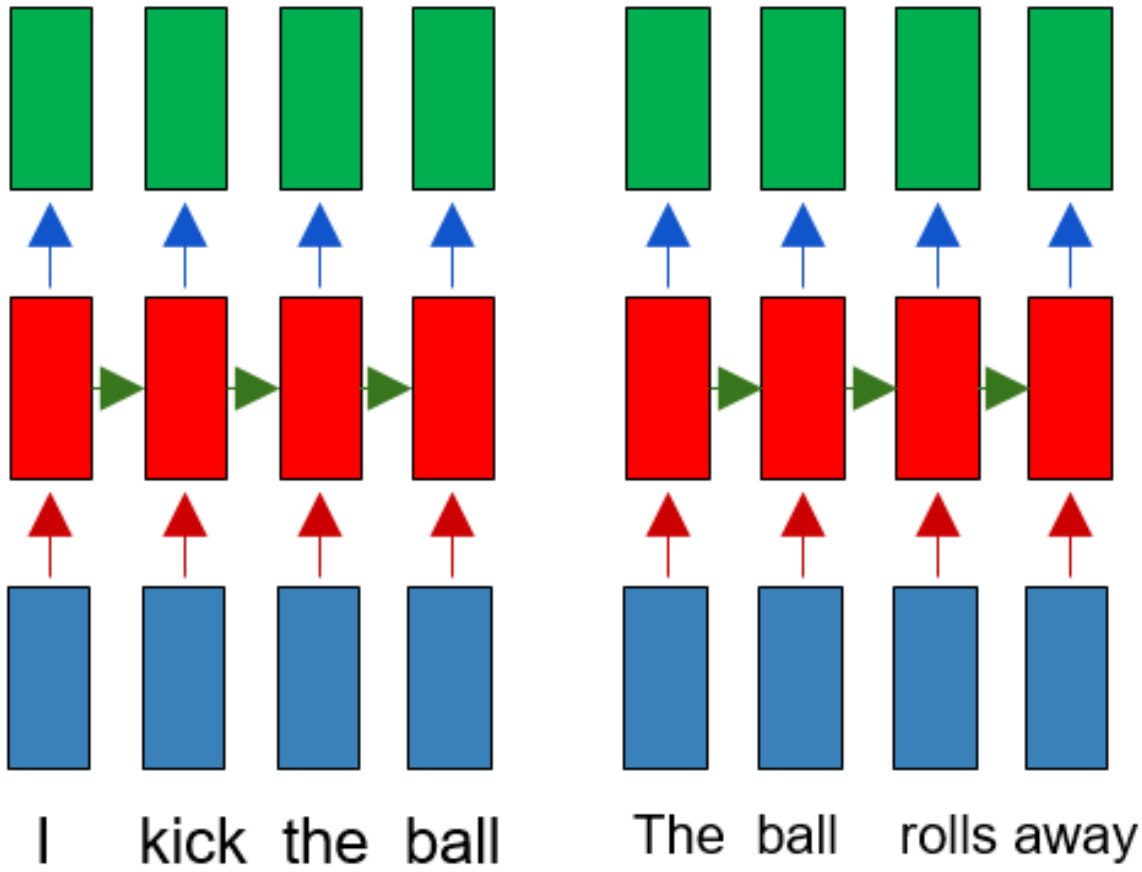


Figure: RNN in 112 lines

Python code

```
#FF
def step(x):
    h = np.tanh(np.dot(W1, x) + b1)
    y = np.dot(W2, h) + b2
    return y

y = step(x)
```

Python code

```
# RNN
rnn = RNN()
y = rnn.step(x)

class RNN:
  def step(self, x):
    self.h = np.tanh(np.dot(W1, self.h)
             + np.dot(W2,x))
    y = np.dot(W3, self.h)
    return y
```

Python code

```
#FF
def step(x):
    h = np.tanh(np.dot(W1, x) + b1)
    y = np.dot(W2, h) + b2
    return y
#RNN
class RNN:
    def step(self, x):
        self.h = np.tanh(np.dot(W1, self.h)
                + np.dot(W2,x))
        y = np.dot(W3, self.h)
        return y
```

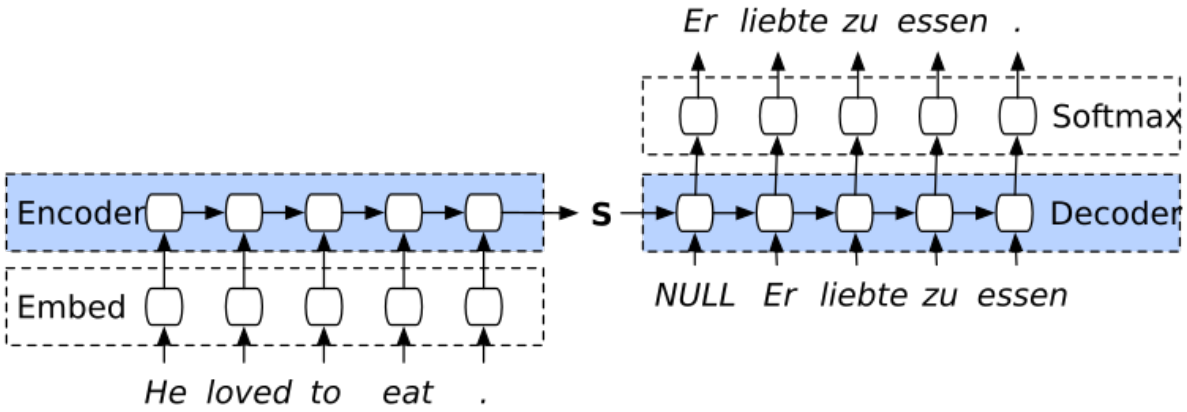


Figure: RNN for Machine Translation (credits: smerity.com)

Example

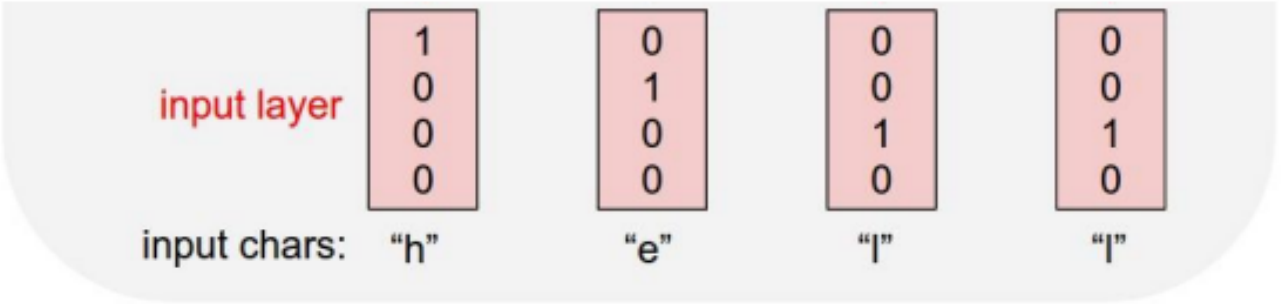


Figure: cs231n, Andrej Karpathy

Example

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

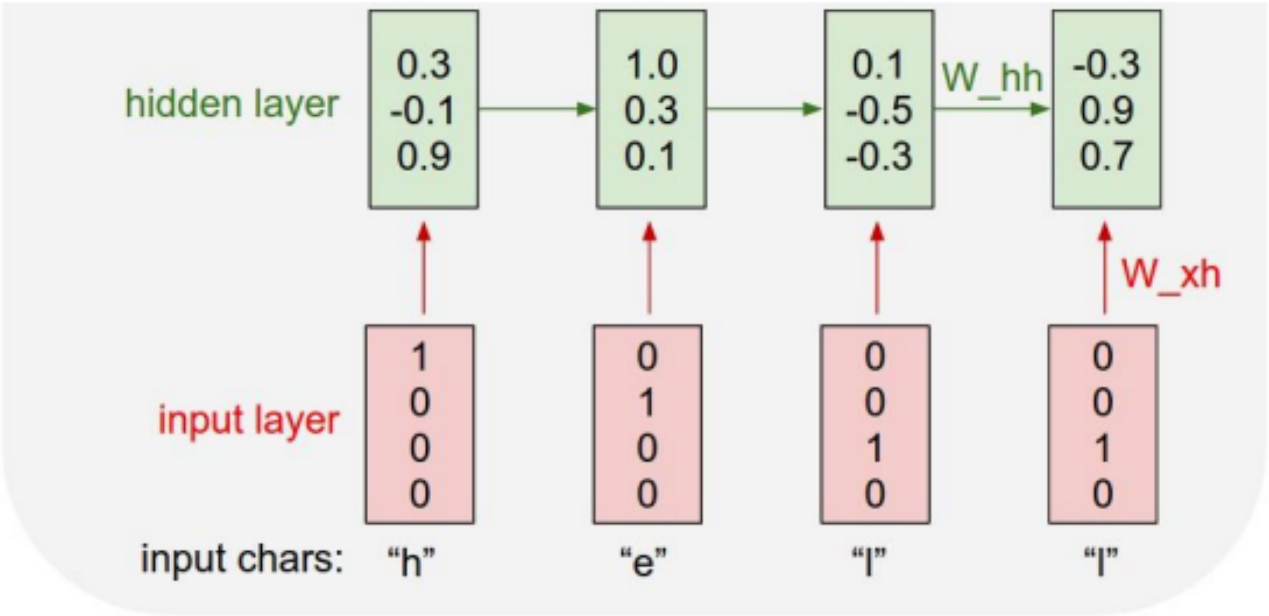


Figure: cs231n, Andrej Karpathy

Example

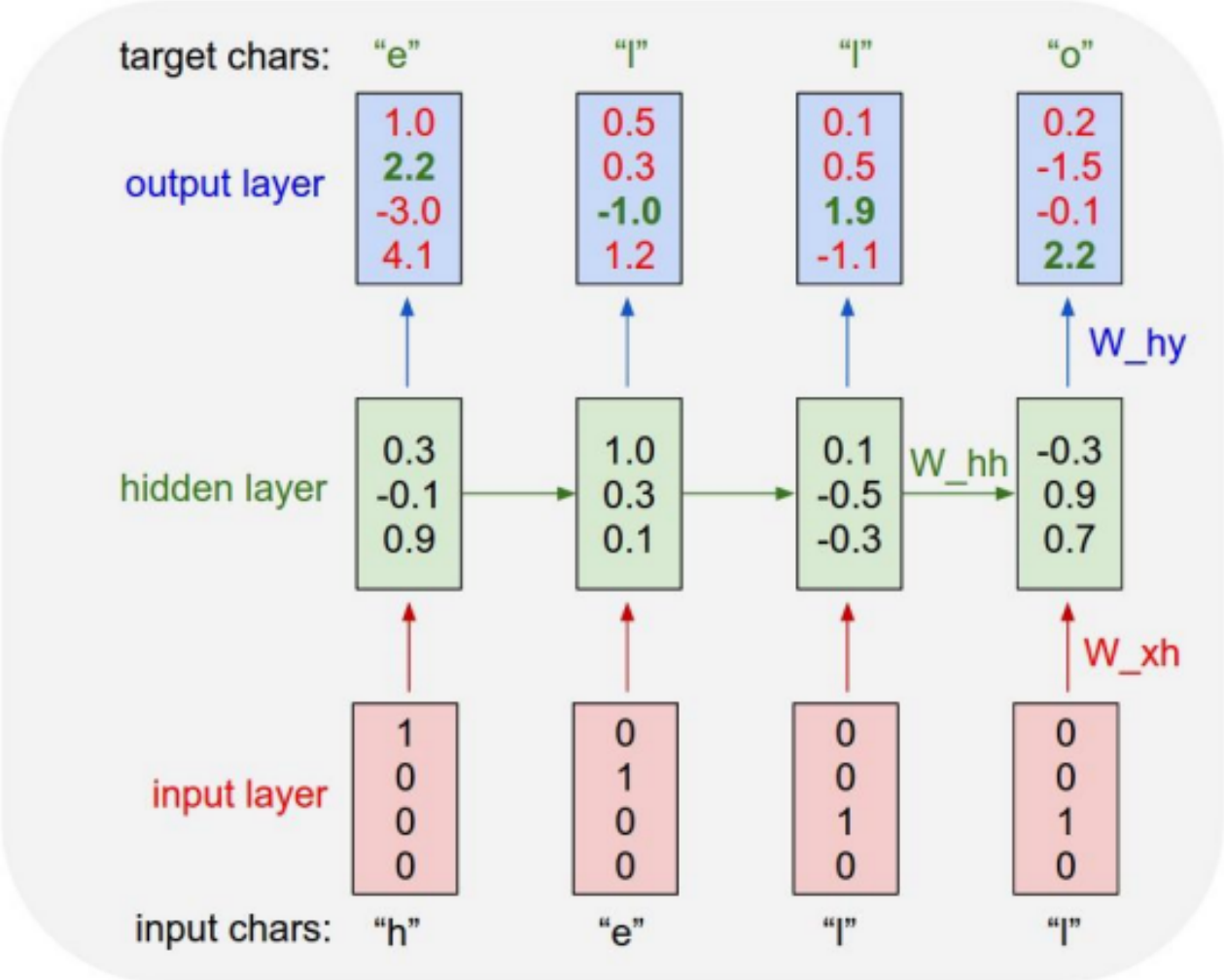


Figure: cs231n, Andrej Karpathy

Long Short-term memory

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)
\end{aligned}
$$

$$
\begin{aligned}
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
h_t &= o_t \ tanh(c_t)
\end{aligned}
$$

LSTM

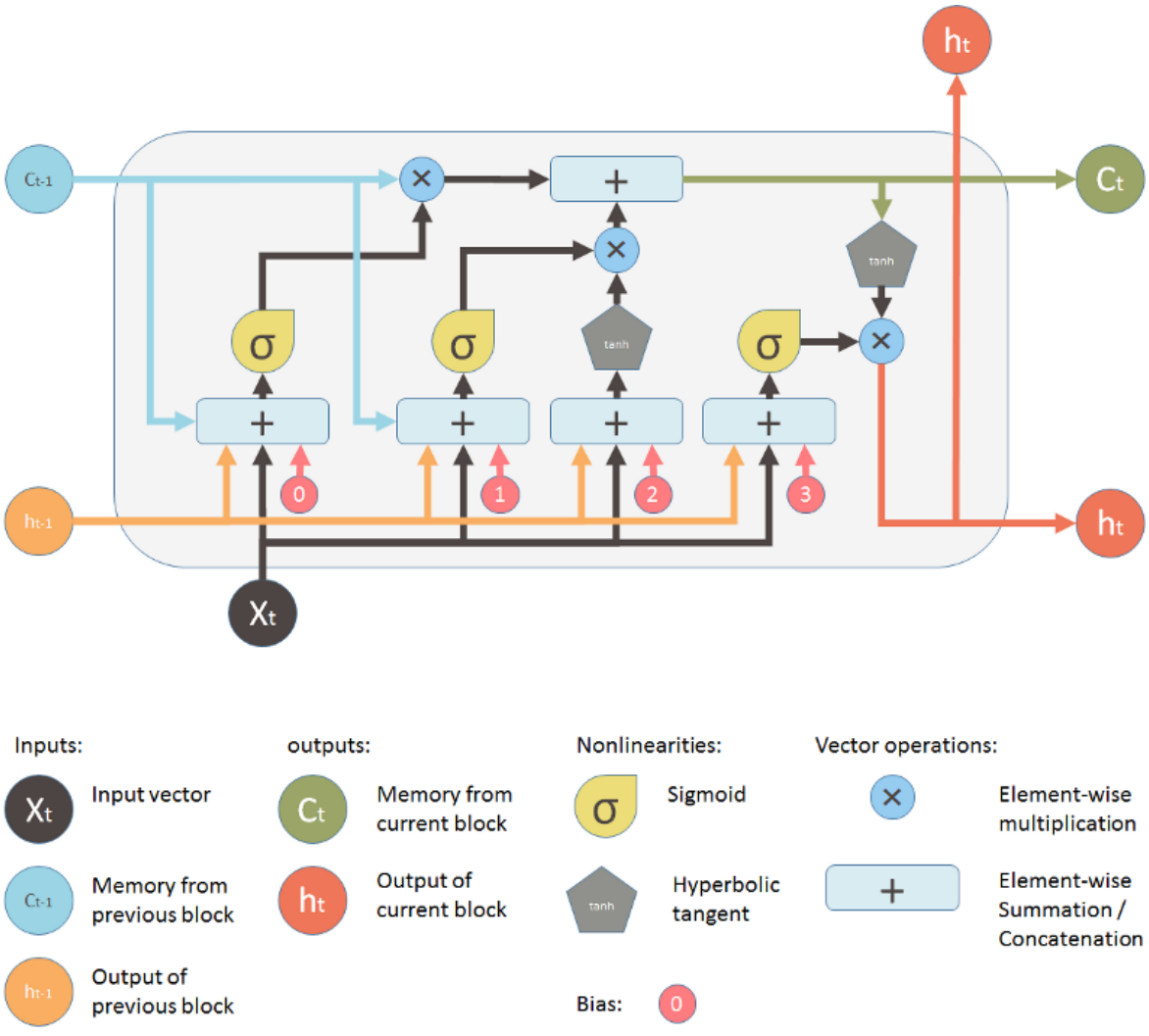


Figure: Diagram depicting LSTM block
Author: Shi Yan. Source: medium.com/@shiyan/

Overview			
	FF	RNN	CNN
- Basic computation	Matix multiplies	Recurrent multiplications	Convolutions
- Typical data	Embedding vectors	Sequences	Images
- Weights ties	No	Sequential	Spatial
- Effective span	Across input	Long term	Local receptive field
- Stationary		Seldom events	Frequent patterns
- How to train	Backprop	Backprop	Backprop
- Performance	No guarantee	No guarantee	No guarantee
- Typical examples	None	Machine translation, speech tagging, text generation	segmentation, image processing, object detection

Span

1. CNN

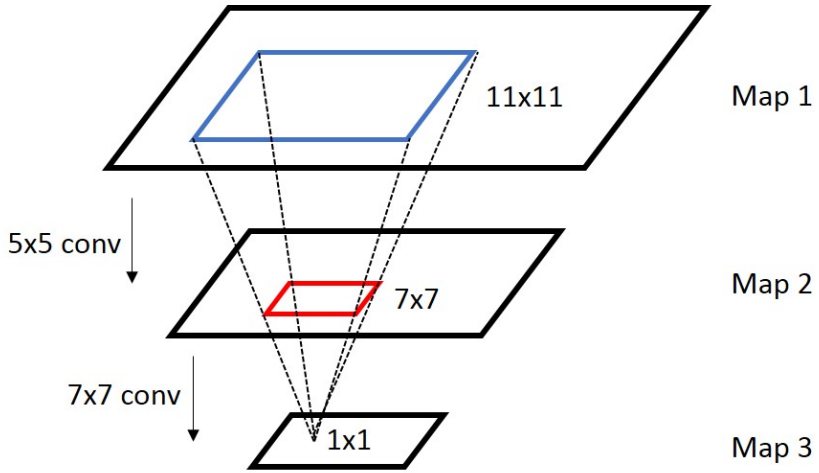


Figure: credits: cvmarcher.com/

2. RNN

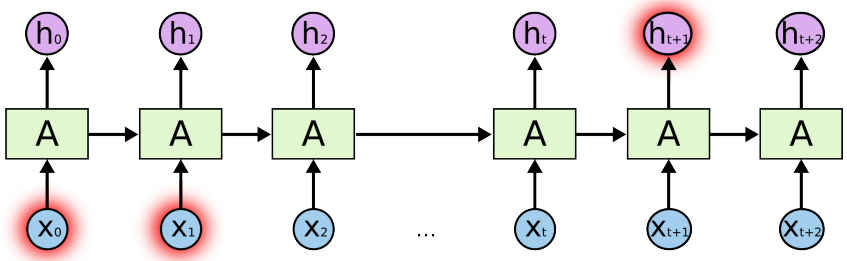


Figure: credits colah.github.io

Datasets and algorithms

Tagging of clinical events

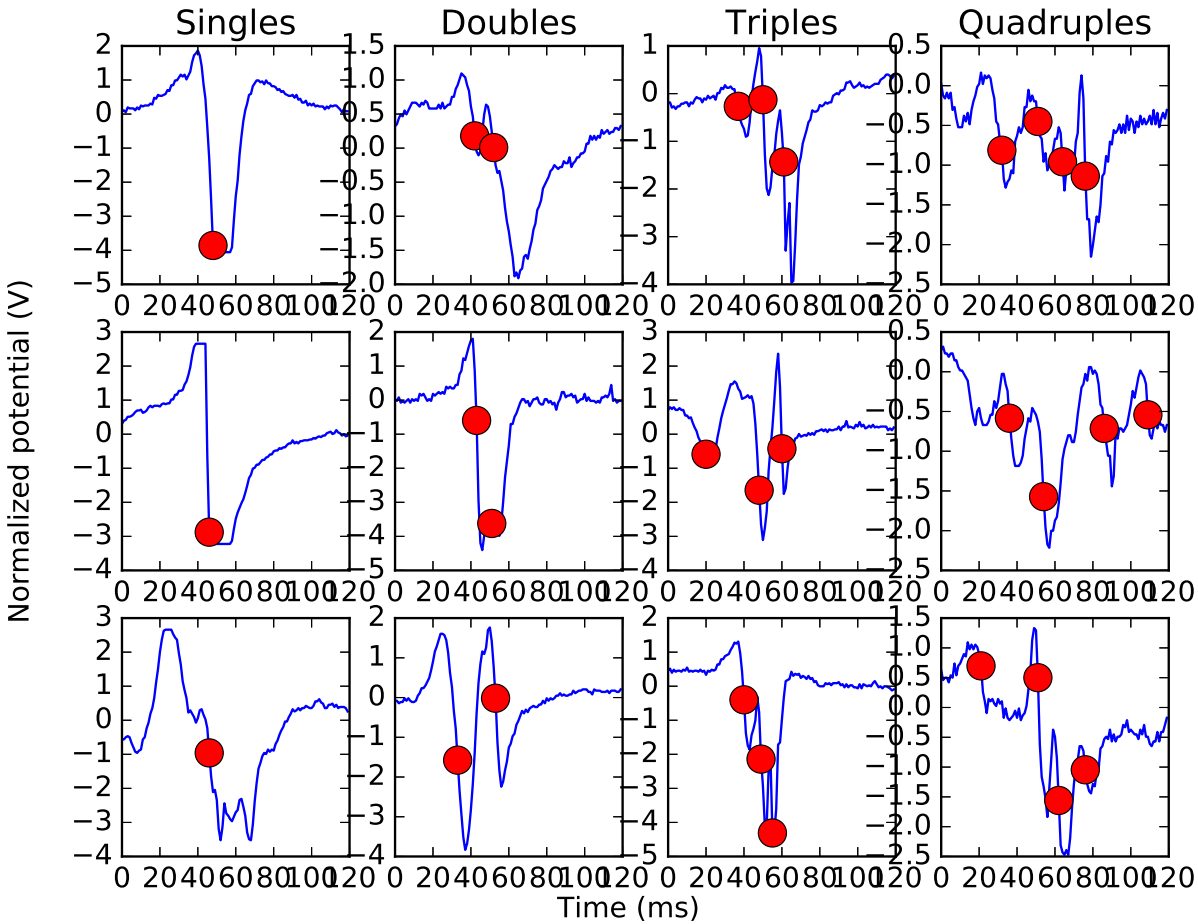


Figure: Classification of fractionated electrograms in epicardial mappings using a recurrent neural network, R Romijnders et al.

Use LSTM's, because the deflections occur as events

Classifying music

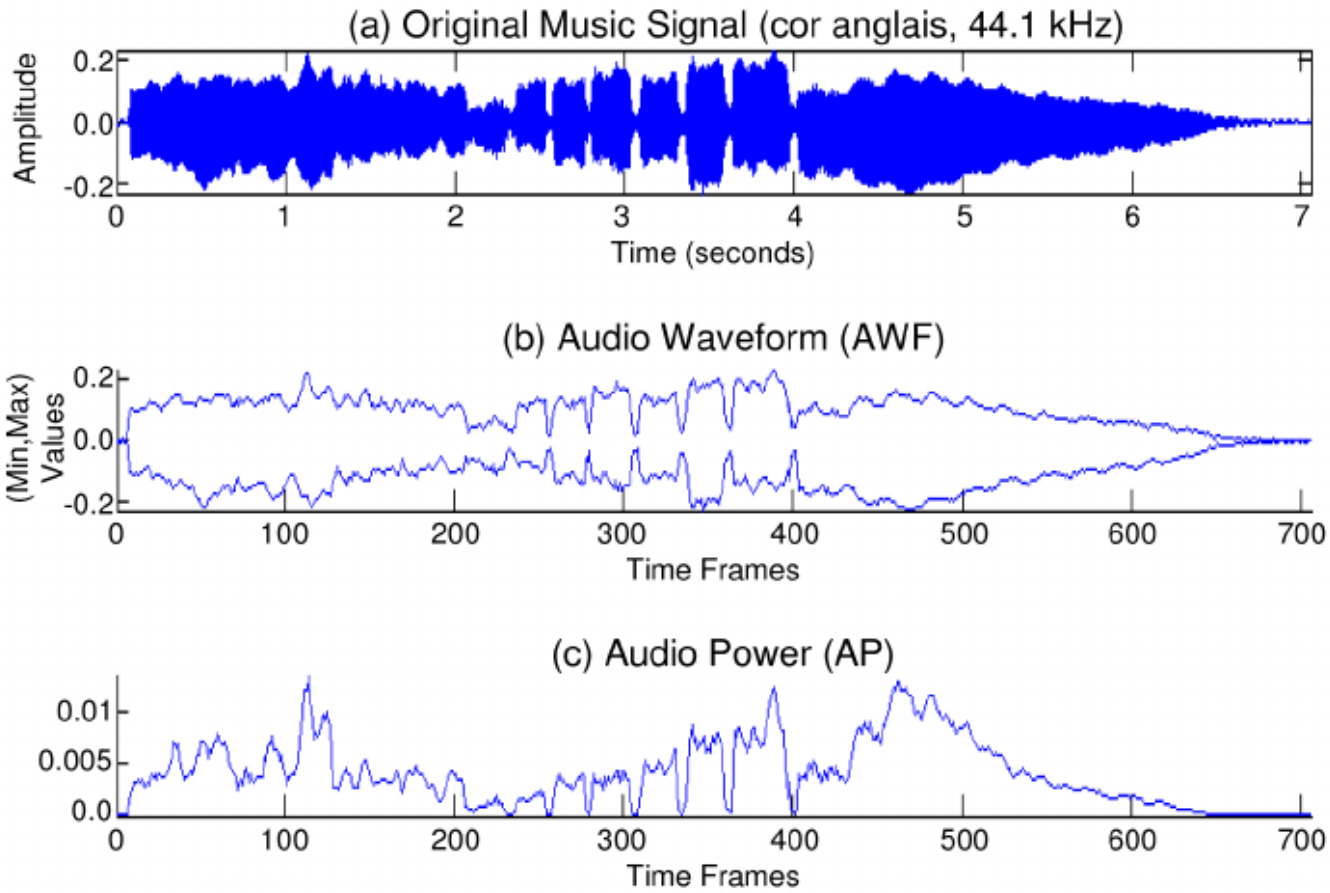


Figure: ML for music

Use CNN's, because music genre follows a certain pattern

1. *The film did n't move me one way or the other , but it was an honest effort and if you want to see a flick about telemarketers this one will due .*
2. *For those of us who respond more strongly to storytelling than computer-generated effects , the new Star Wars installment has n't escaped the rut dug by the last one .*

IMDB sentiment challenge, Kaggle

Use LSTM, because a single word could change the sentiment. Compare *I do like this* and *I do not like this*

Architectures for RNN

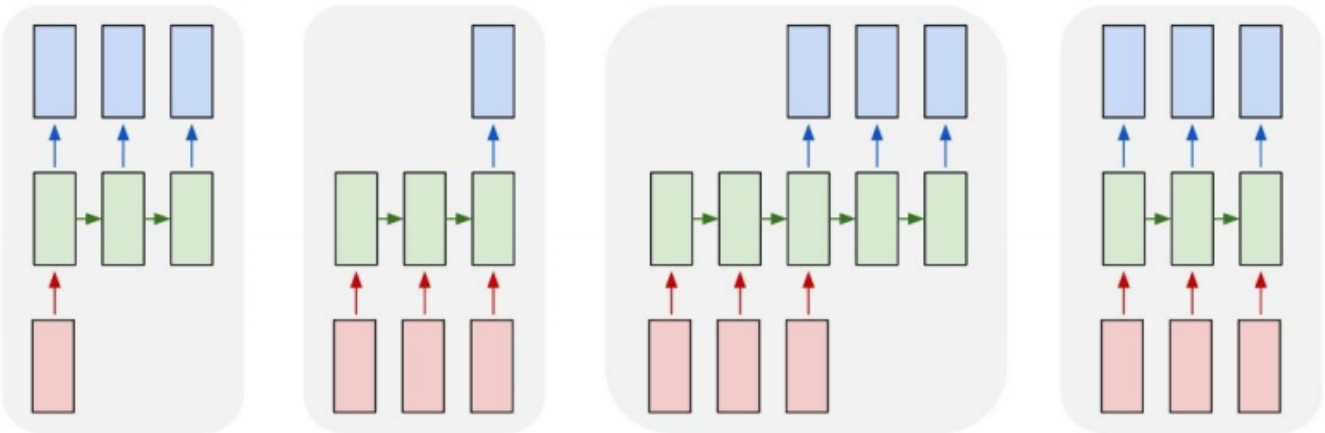


Figure: credits: Andrej Karpathy

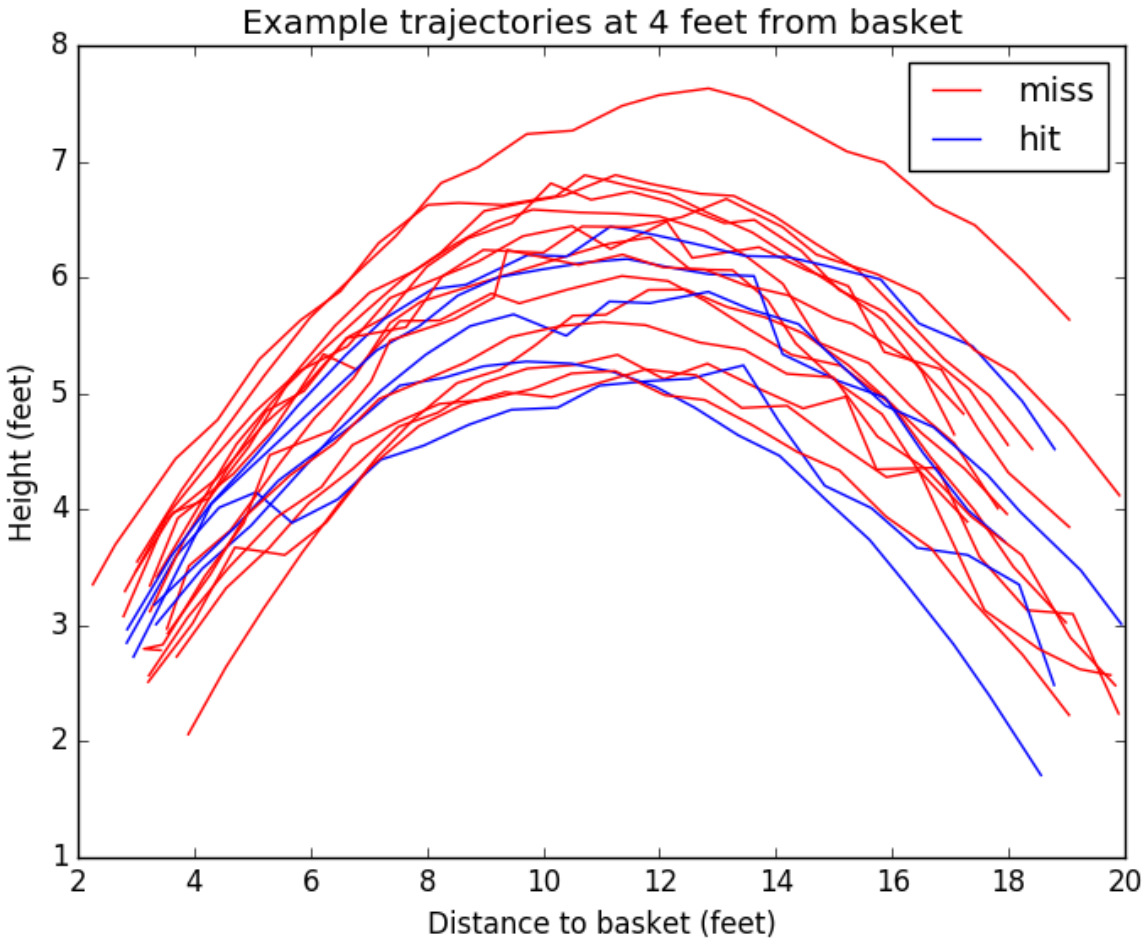


Figure: DL for basketball, Rajiv and Rob

Works with both CNN and LSTM

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Figure: credits: Andrej Karpathy

Use LSTM's, because it predicts a vector (word) for each time step

Tabular

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Figure: Titanic KAGGLE

Do NOT use deep learning. Go for Random Forests or SVM

Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with legos toy."

"boy is doing backflip on wakeboard."

Figure: Link to page

Use CNN for pattern extraction from image. Use RNN to generate caption.

DIY

DIY

1. Gather data
2. Choose architecture
3. Code up neural net
4. Train
5. Add bells and whistles
6. Post processing

Gather data

1. SMALL:
   1. MNIST: 60.000 samples
   2. TIMIT: 630 speakers, 10 sentences
2. NORMAL:
   1. Imagenet: 3.2 million images
   2. FLICKR: 1 million images
   3. Word vectors on all of Wikipedia

Amazon Mechanical Turk

1 Convolution

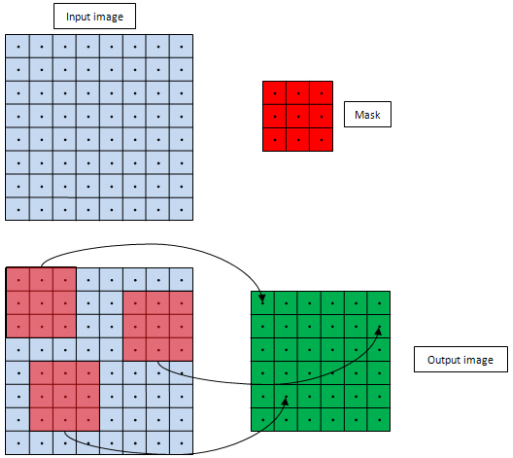


Figure: developer.amd.com

2 Recursion

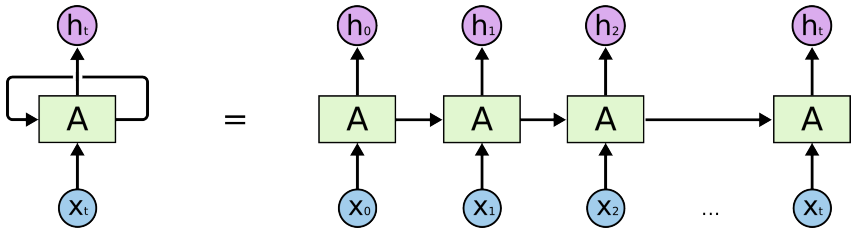


Figure: credits: colah.github.io

Code

1. Low level
    1. Tensorflow
    2. Torch, pyTorch
2. High level
    1. Keras
    2. Scikit flow, pretty tensor, tfslim, ...

Train

1. No bells and whistles
2. Small sub dataset
3. See if you can overfit

Bells and whistles

1. Batch normalization, layer normalization
2. Dropout
3. Add more data: semi supervised learning
4. Residual connections
5. etcetera, etcetera

Activation functions

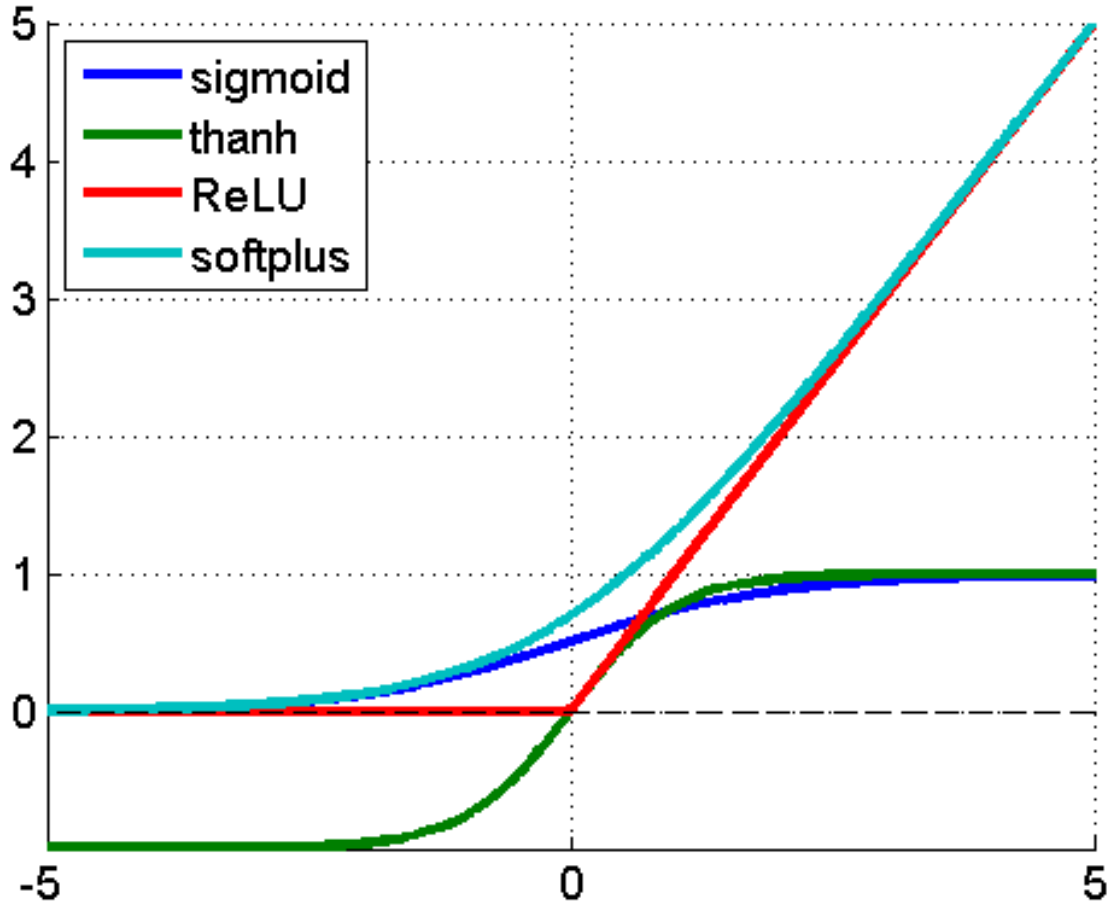


Figure: Common activation functions (credits Vanessa Imiloa)

MNIST

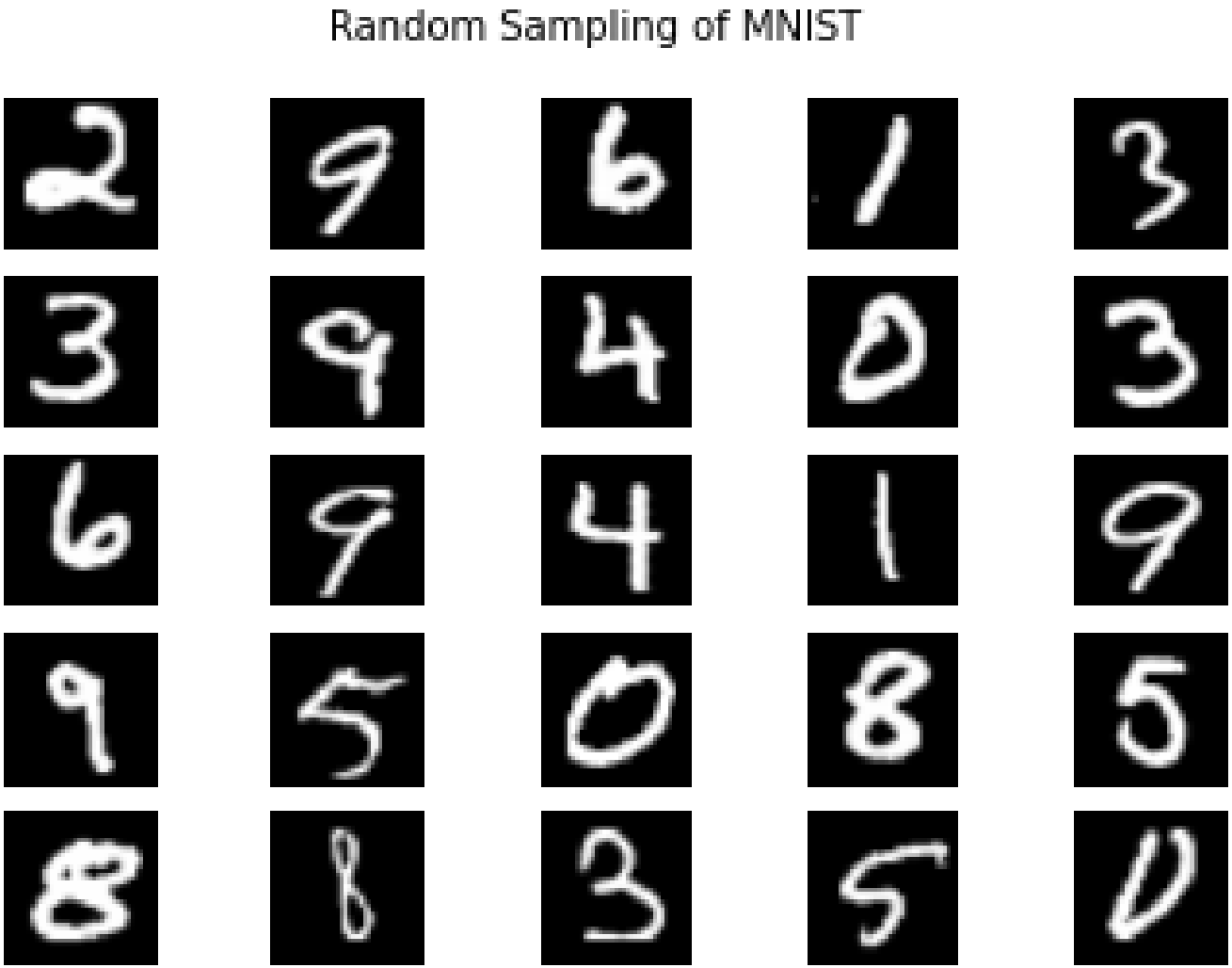


Figure: Caption

Post processing

1. Plot low confidence / high uncertainty
2. Plot confusion matrix
3. Inspect neurons

Regularization

Figure: Example for regularization

Figure: Detecting tanks

Two architectures

1. Convolution

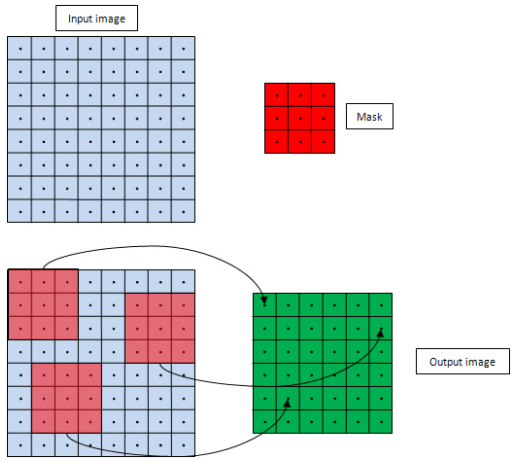


Figure: developer.amd.com

2. Recursion

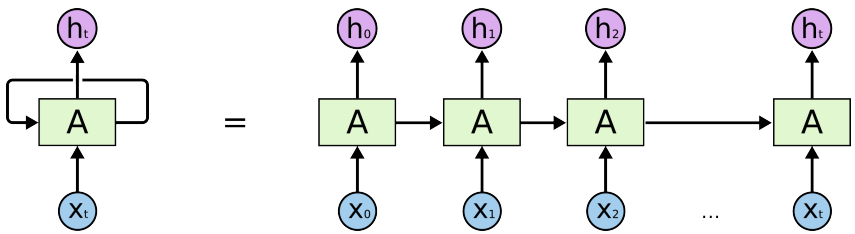


Figure: credits: colah.github.io

Questions

Further reading

1. RNN
    1. The Unreasonable Effectiveness of Recurrent Neural Networks, Andrej Karpathy
    2. Supervised sequence labelling with RNN, Alex Graves
    3. Chris Olah's blog (colah.github.io)
2. CNN
    1. Course on CNN: cs231n (cs231n.stanford.edu/syllabus.html)
    2. Udacity on Deep learning (udacity.com/course/deep-learning–ud730)
3. My page (robromijnders.github.io)