Deep Learning

Ekapol Chuangsuwanich
Department of Computer Engineering
Chulalongkorn University

DNNs (Deep Neural Networks)

Greatly improved performance in Speech Recognition, Computer Vision, Robotics, Machine Translation, Natural Language Processing, etc.

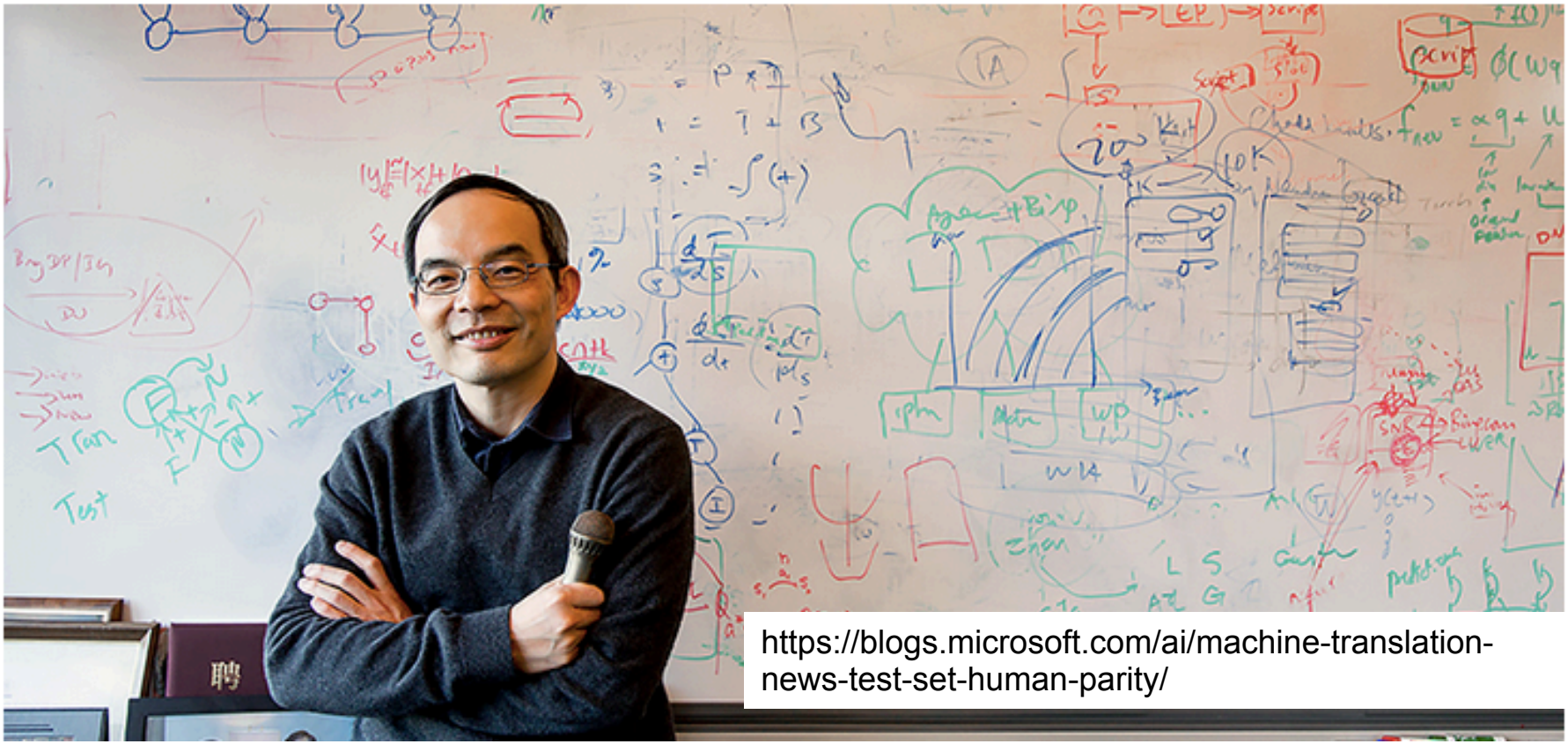
Surpassed human performance in many tasks

Task	Previous state-of-the-art	Deep learning (2012)	Deep learning (2018)
Telephone conversation	24%	16%	5%
Google voice search	16%	12%	5%
ImageNet	28%	16%	4%

Geoffrey Hinton, Deep Neural Networks for Acoustic Modeling in Speech Recognition, 2012

Microsoft reaches a historic milestone, using AI to match human performance in translating news from Chinese to English

Mar 14, 2018 | Allison Linn



https://blogs.microsoft.com/ai/machine-translation-news-test-set-human-parity/

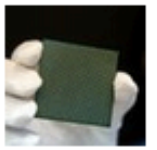
Google's AlphaGo Defeats Chinese Go Master in Win for A.I.

点击查看本文中文版

By PAUL MOZUR MAY 23, 2017

RELATED COVERAGE


A.I. I
Repla


China
FEB. 3,


THE FU
The F


Maste
Goog

https://www.nytimes.com/2017/05/23/business/google-deepmind-alphago-go-champion-defeat.html

Why now

Neural Networks has been around since 1980s

Big data – DNN can take advantage of large amounts of data better than other models

GPU – Enable training bigger models possible

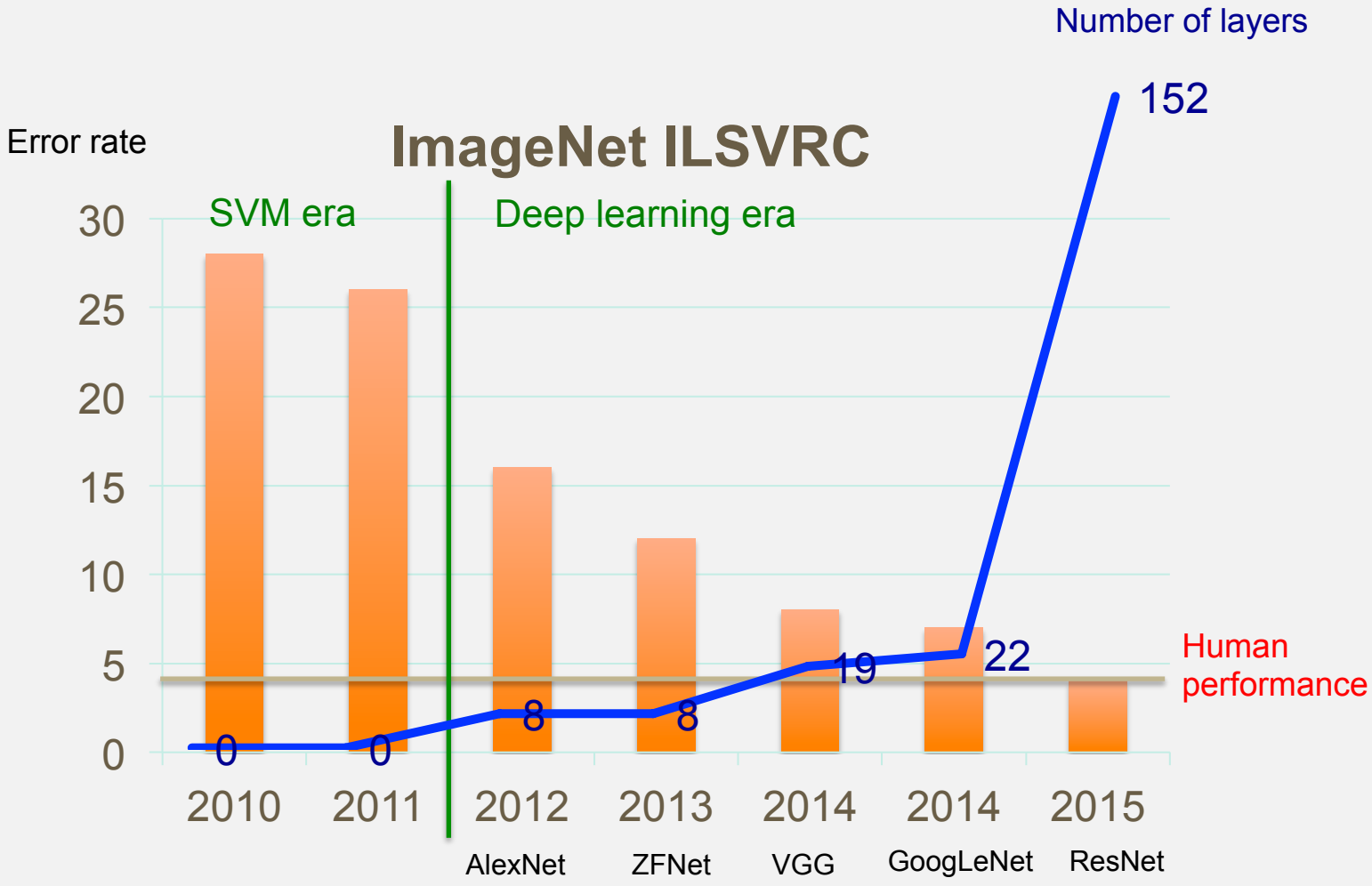
Deep – Easier to train when the model is large

ImageNet – object classification challenge

Alex, Krizhevsky, Imagenet classification with deep convolutional neural networks, 2012

Wider and deeper networks

Olga Russakovsky, ImageNet Large Scale Visual Recognition Challenge, 2014 https://arxiv.org/abs/1409.0575



Why is deep learning good

Traditional machine learning approaches need features engineering

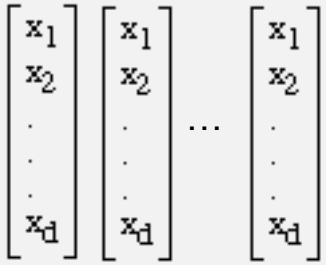
- Hard to come up with good and robust features
- Can you describe why a car looks like a car?

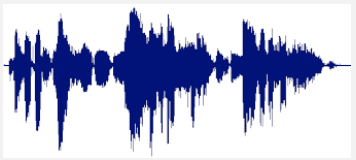
Deep learning can automatically learn the features

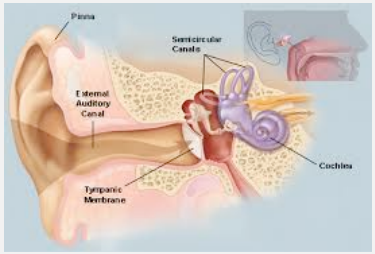
Deep learning combines features engineering and modeling into one single optimization problem

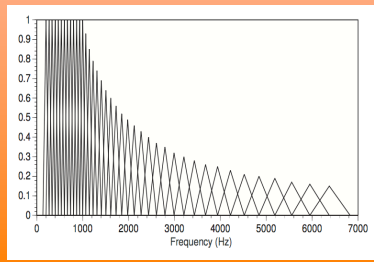
Feature engineering in Speech recognition

Features for traditional ASR









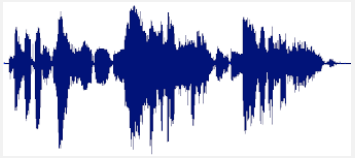
Select lower coefficients

DCT

log

|STFT|

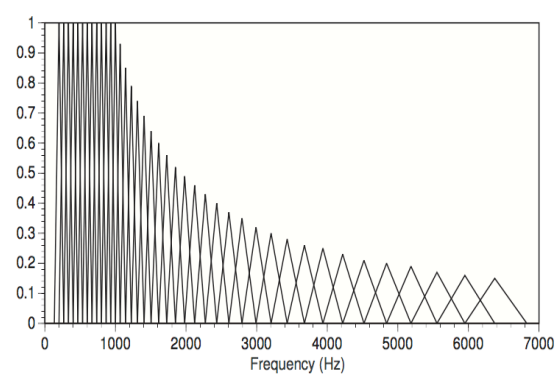
DNN features

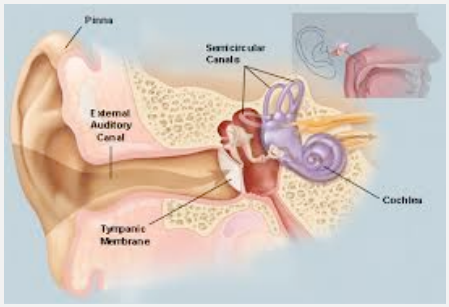


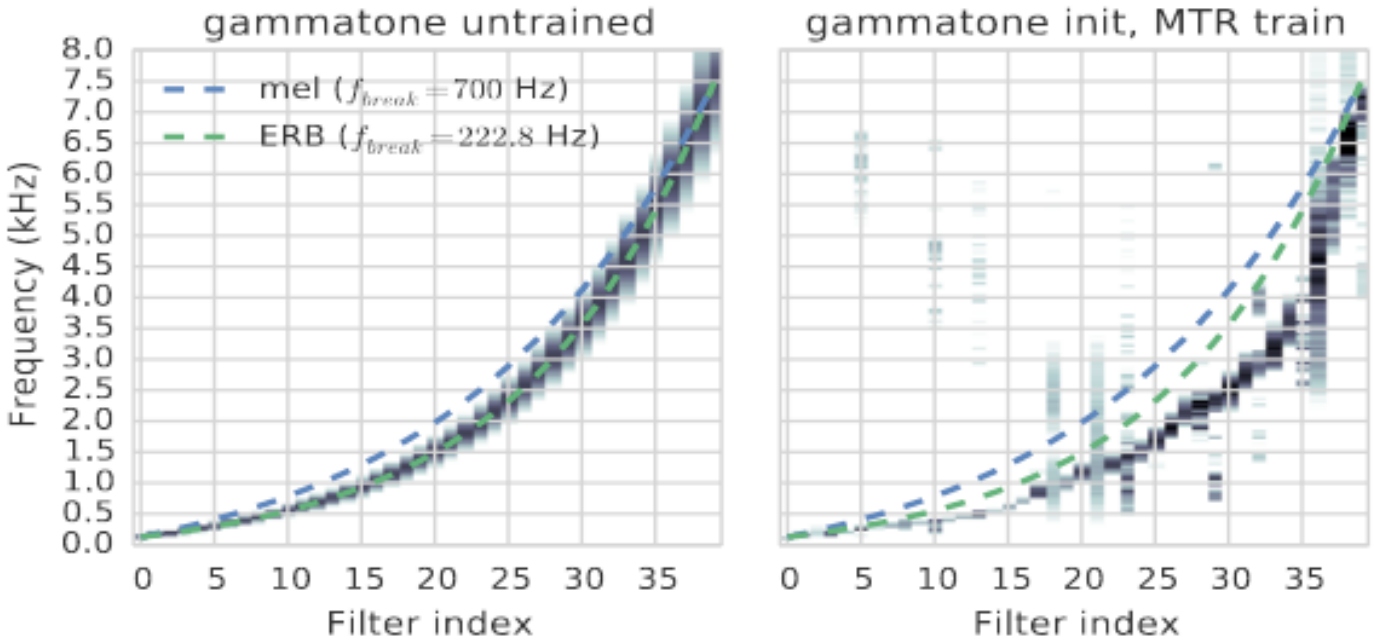
Features for ASR with DNN

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_d \end{bmatrix} \cdots \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_d \end{bmatrix}$$

Learning the ear response

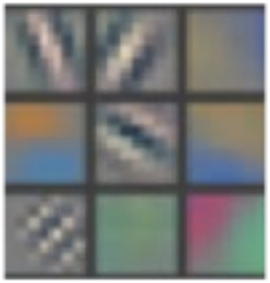




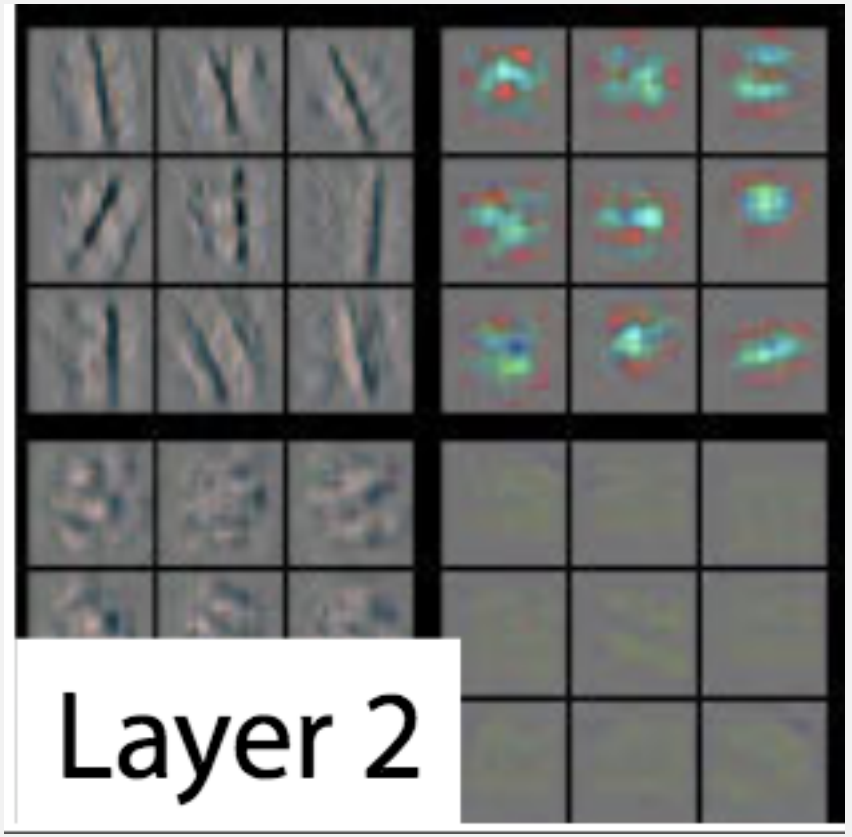


Tara Sainath, Learning the Speech Front-end with Raw Waveform CLDNNs, 2015

Computer Vision Features

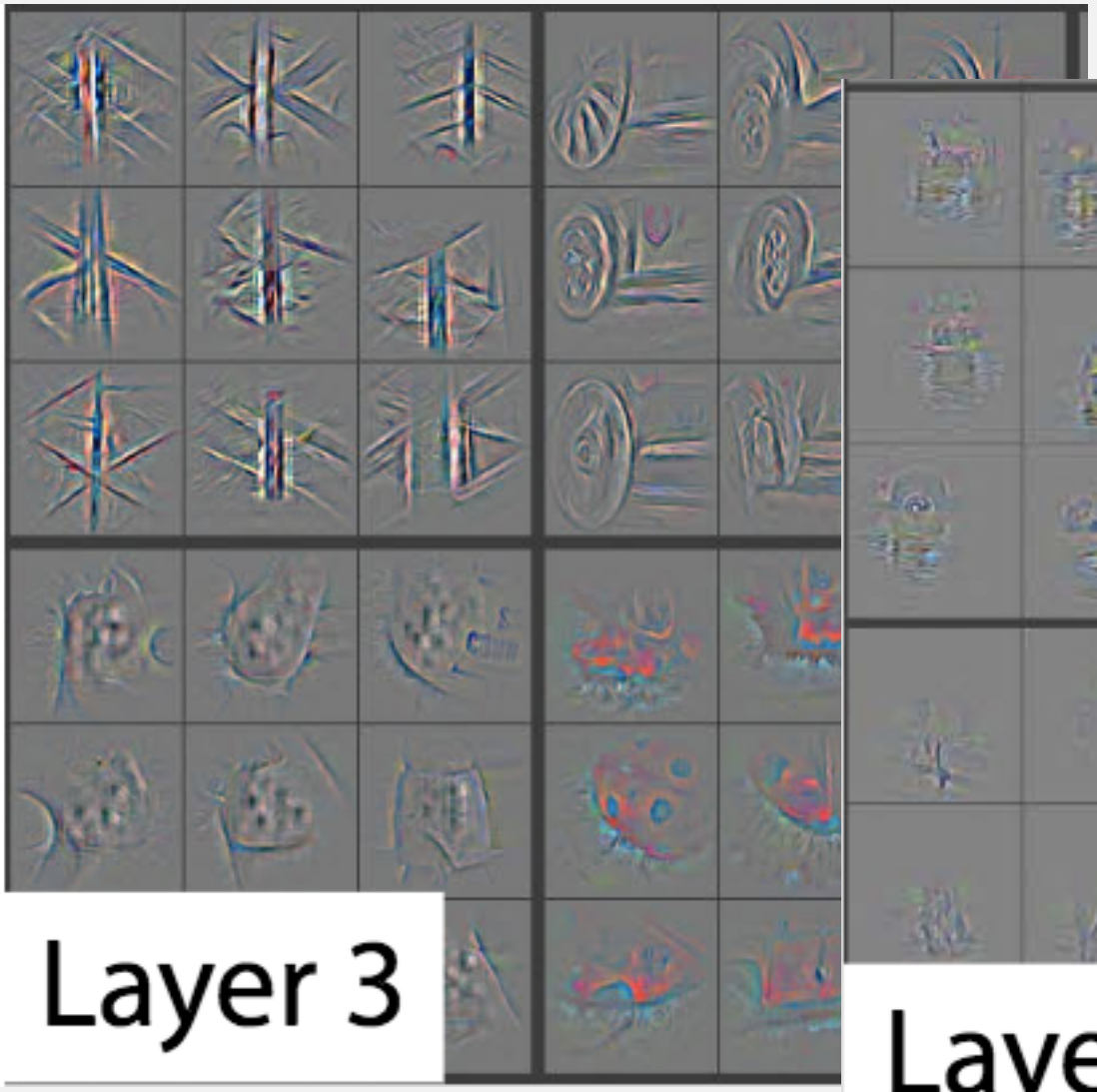


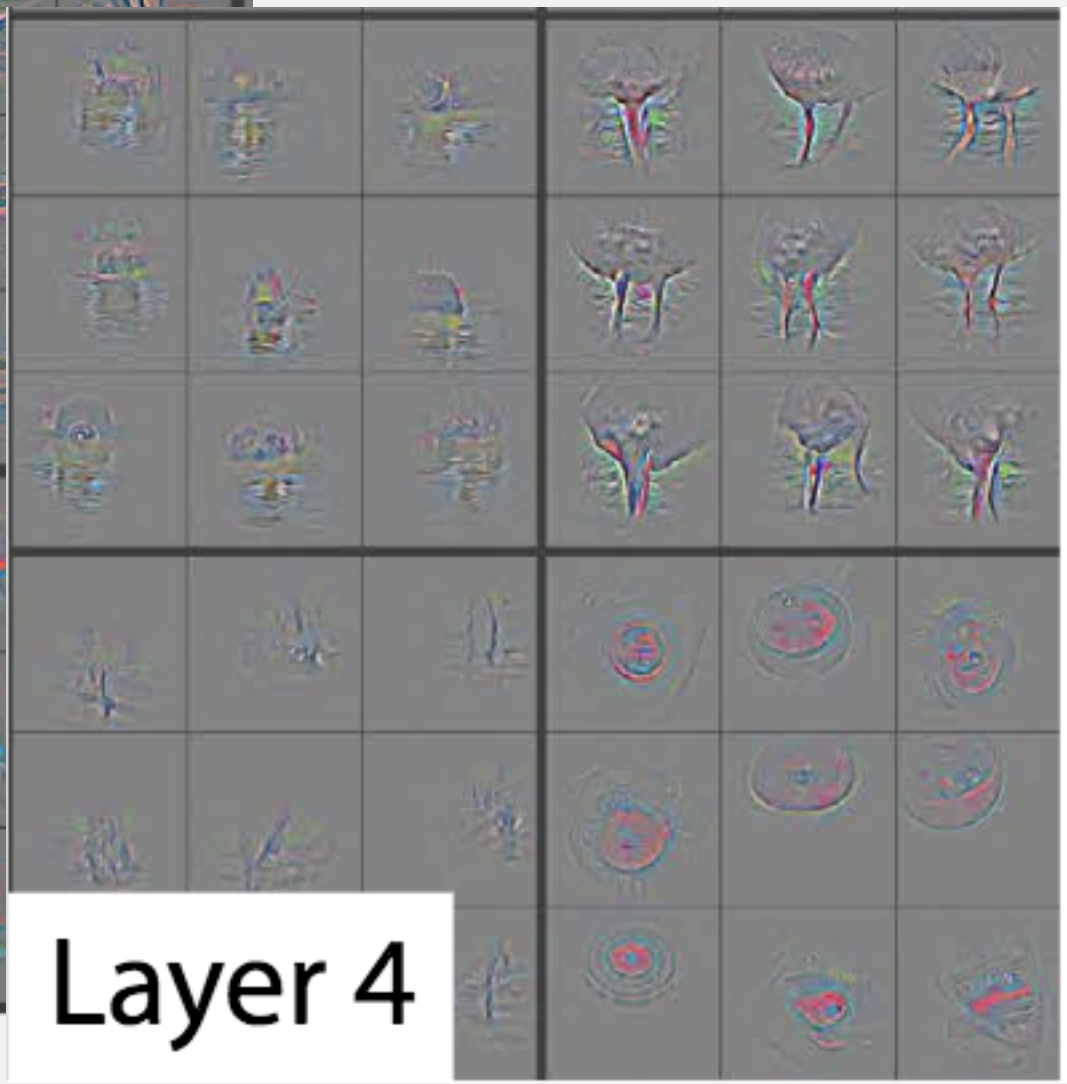
Layer 1



Layer 2

Matthew D Zeiler, Visualizing and Understanding Convolutional Networks, 2013.


Layer 3


Layer 4

Deep learning building blocks

Fully connected networks

Convolutional neural networks

Recurrent neural networks

Gated recurrent units

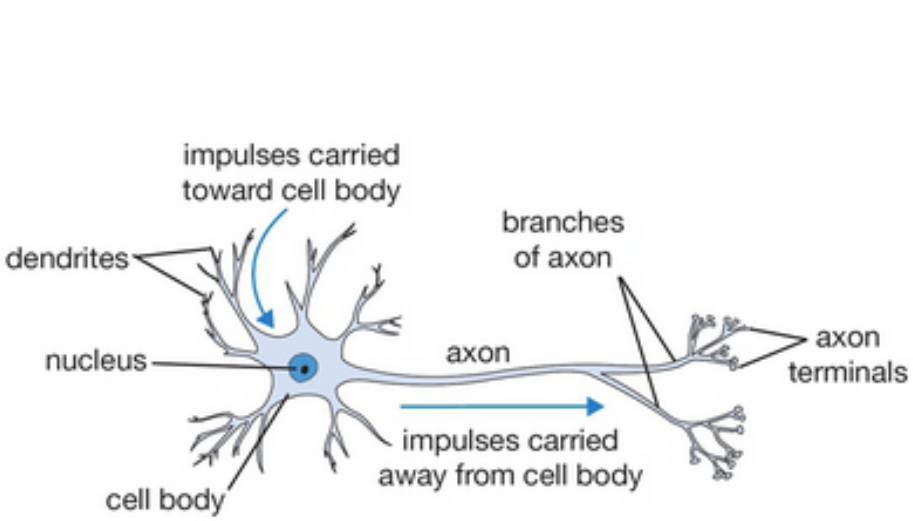
Long short-term memory networks

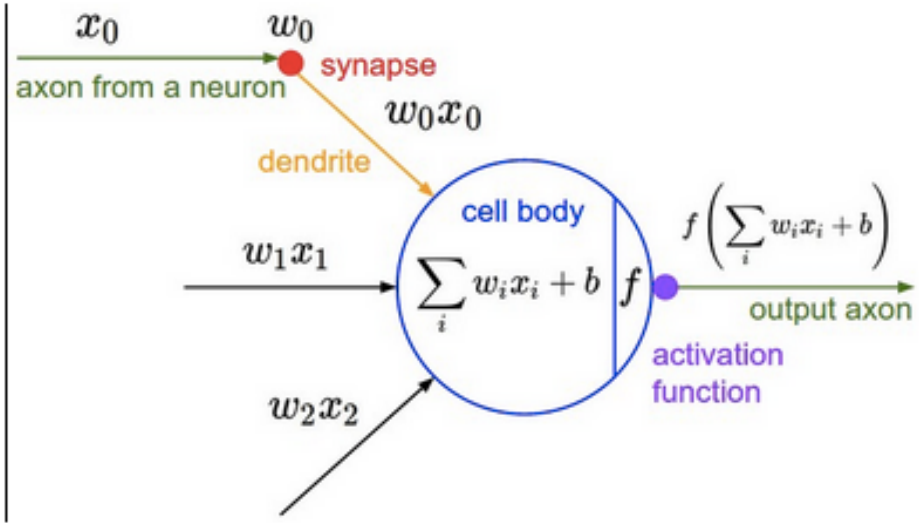
Encoder-Decoder

Fully connected networks

Many names: feed forward networks or deep neural networks or Multilayer perceptron

Composed of multiple neurons





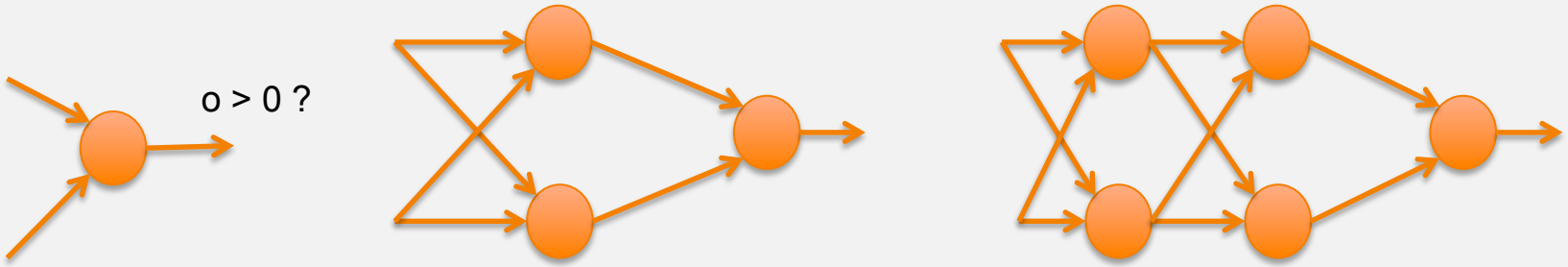
http://cs231n.github.io/neural-networks-1/

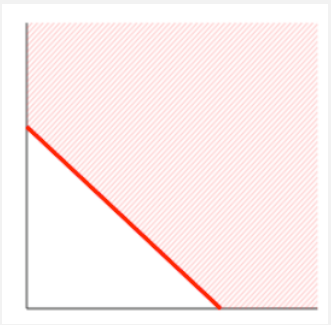
Combining neurons

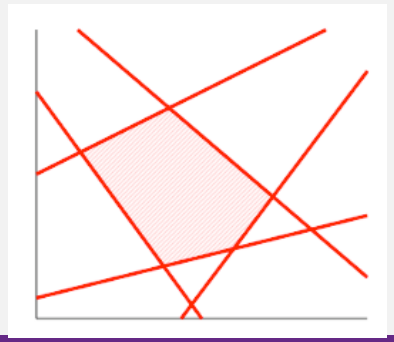
Each neuron splits the feature space with a hyperplane

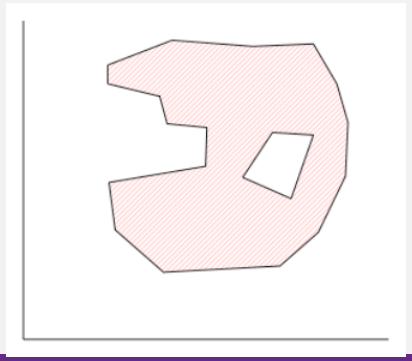
Stacking neuron creates more complicated decision boundaries

More powerful but prone to overfitting

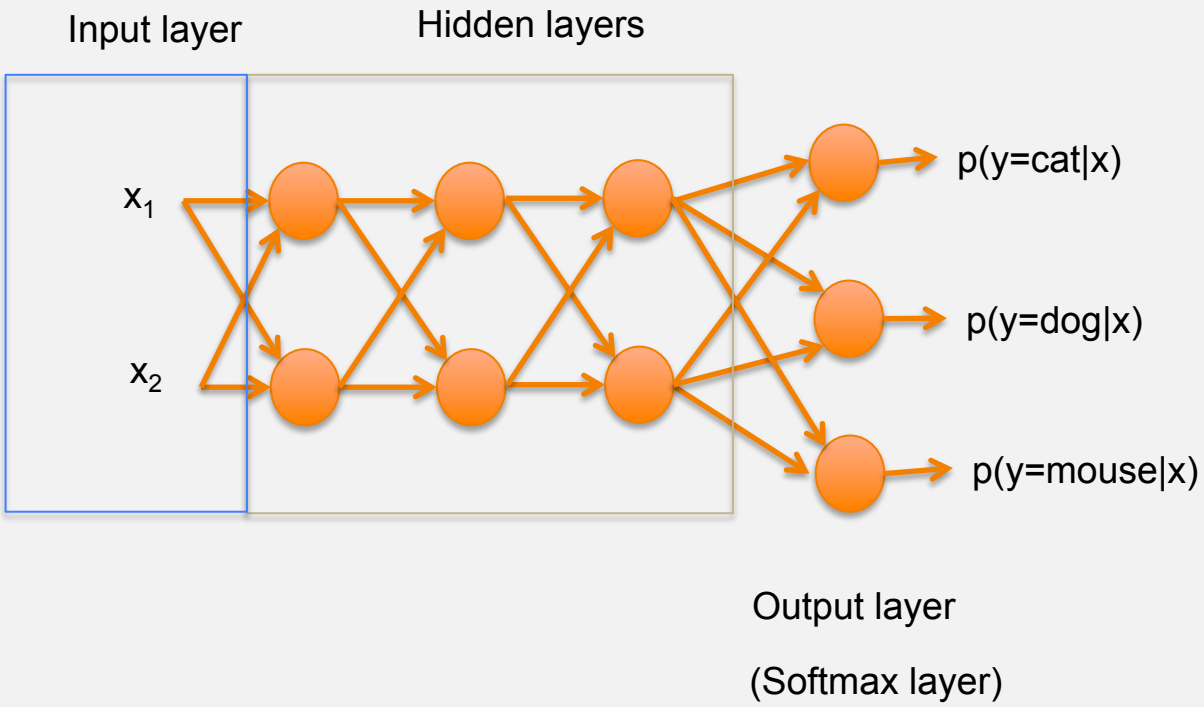




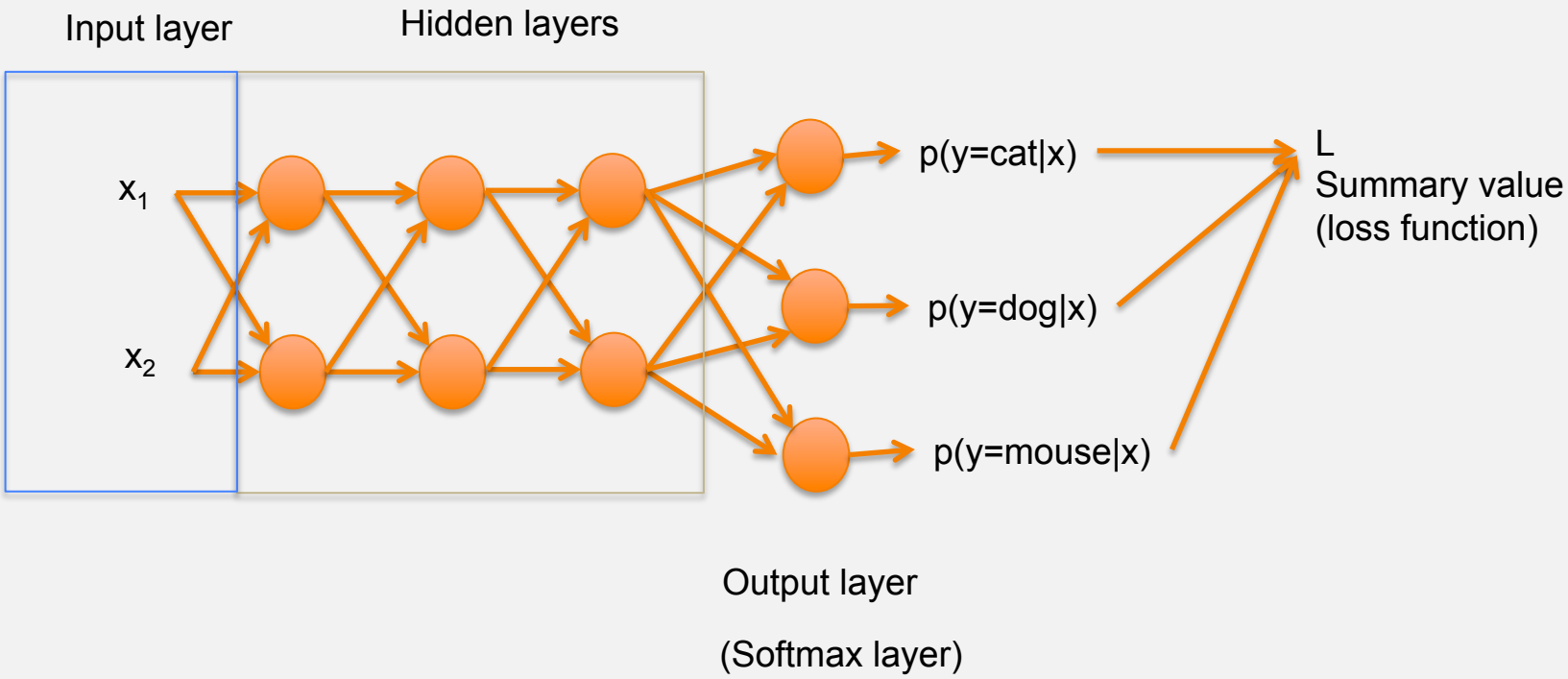




Terminology



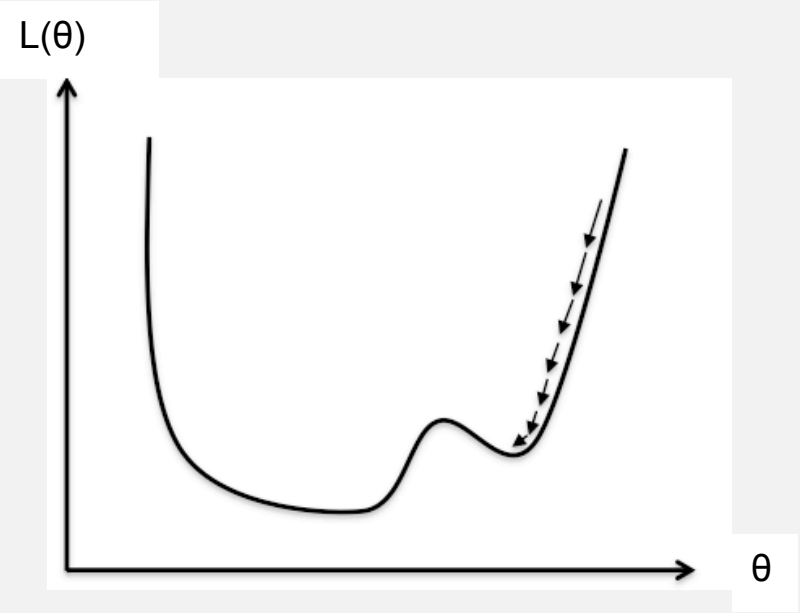
Terminology



Minimization using gradient descent

We want to minimize L with respect to θ

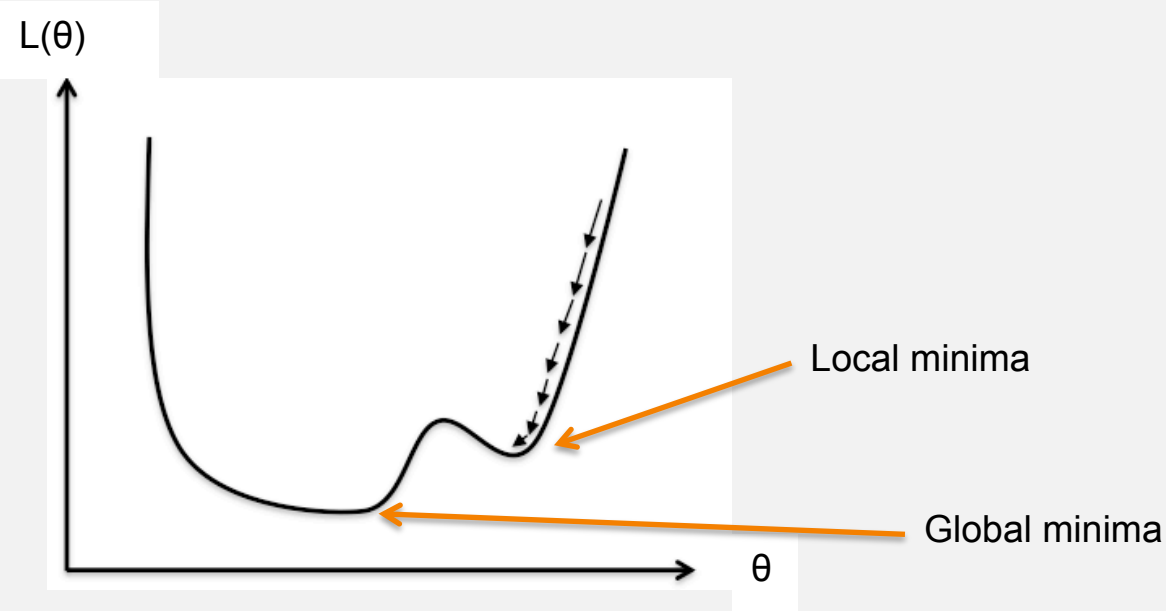
- Differentiate with respect to θ
- Gradients passes through the network by Back Propagation



Deep networks better avoid local minimas

When the network is deep, it is harder to fall into local minimas

Need to beware of overfitting instead



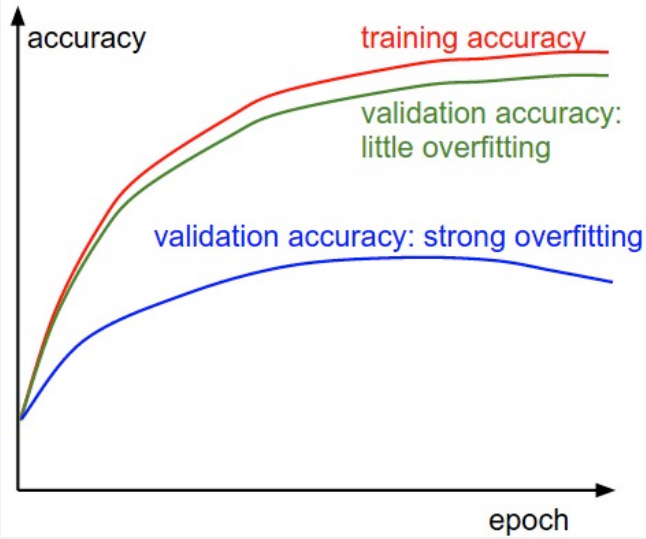
Kenji Kawaguchi,Deep Learning without Poor Local Minima, 2016 https://arxiv.org/abs/1605.07110

Overfitting in DNN

You can keep doing back propagation and overfits to the data

Need to monitor performance on a held out set

Stop or decrease learning rate when overfit happens



http://cs231n.github.io/neural-networks-3/

Ways to combat overfitting

- Dropout
- Batch normalization

Deep Learning Building Blocks

~~Fully connected networks~~

Convolutional neural networks

Recurrent neural networks

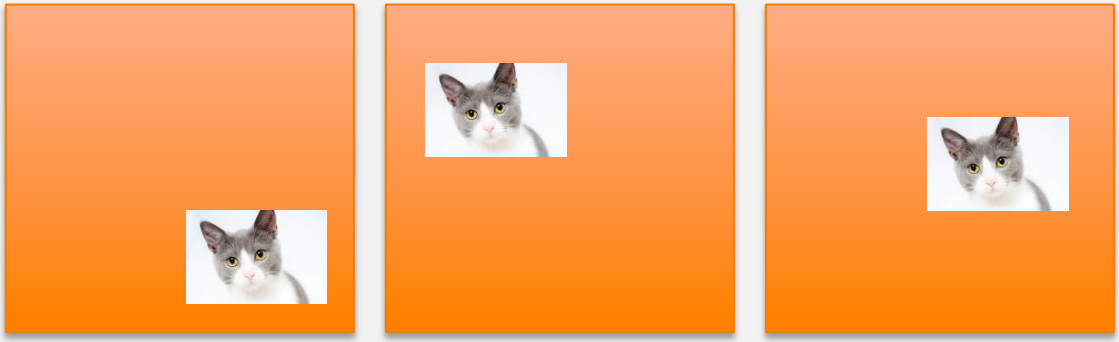
Gated recurrent units

Long short-term memory networks

Encoder-decoder

Convolutional Neural Networks (CNNs)

Consider an image of a cat. DNNs need different neurons to learn every possible location a cat can be



Can we use the same parameters to learn that a cat exists regardless of location?

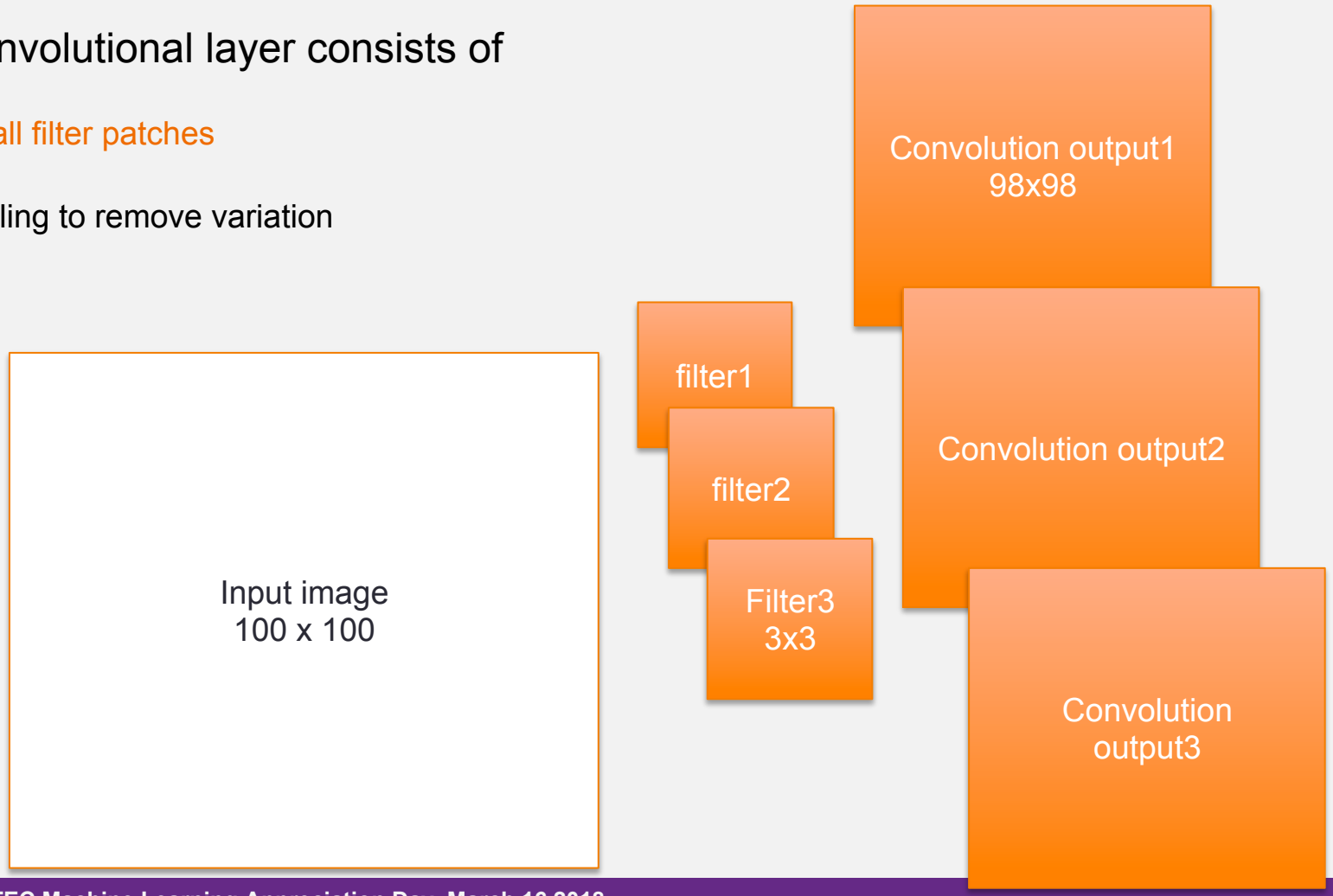
2 parts: convolutional layer and pooling layer

Convolutional filters

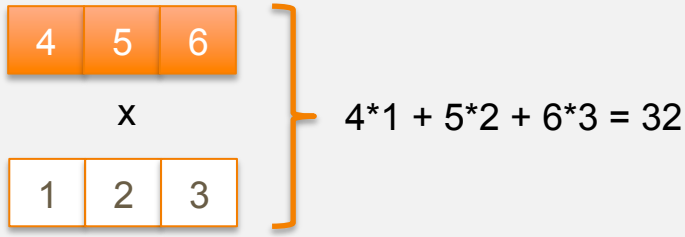
Convolutional layer consists of

Small filter patches

Pooling to remove variation



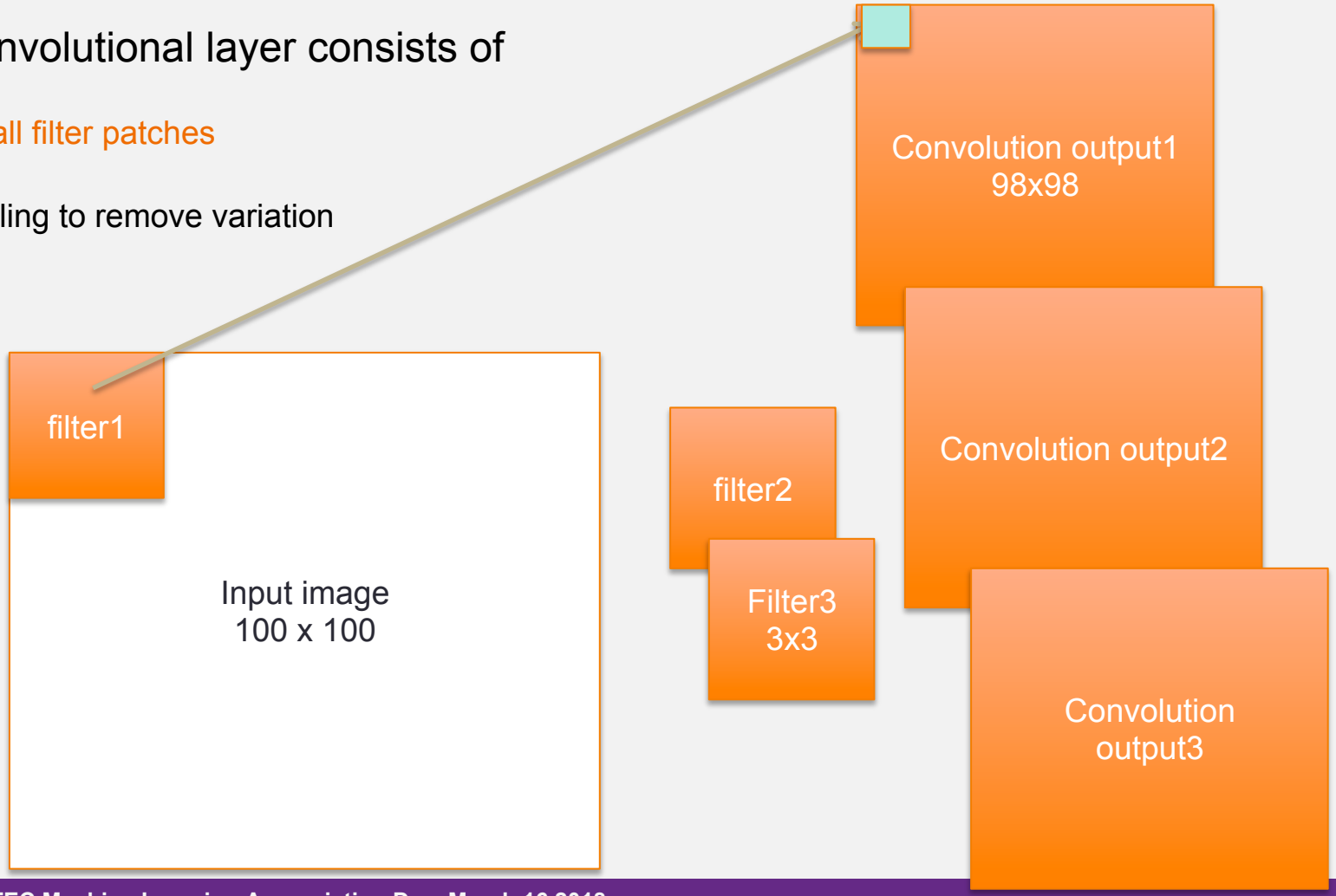
Convolutional filters



Convolutional layer consists of

Small filter patches

Pooling to remove variation

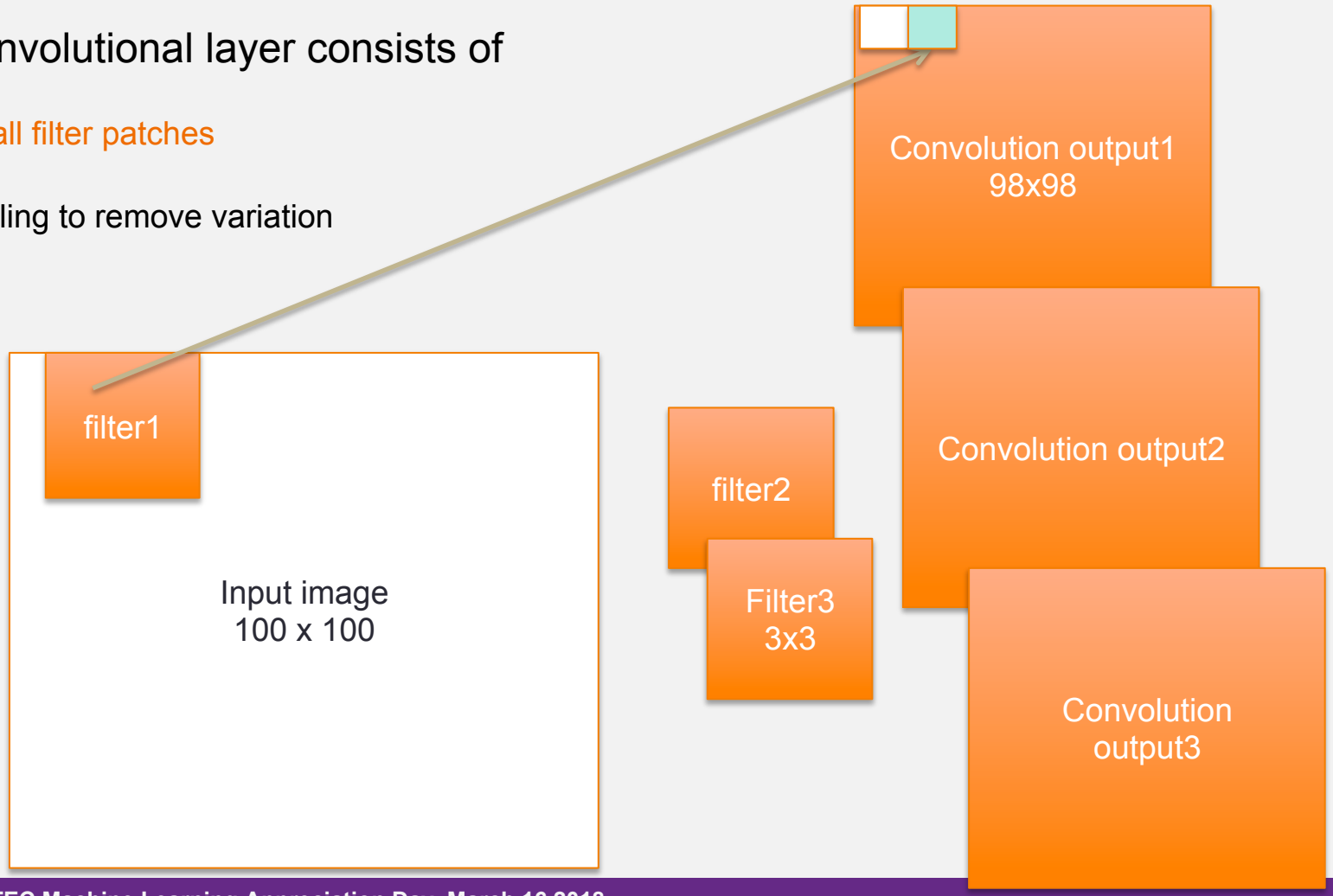


Convolutional filters

Convolutional layer consists of

Small filter patches

Pooling to remove variation

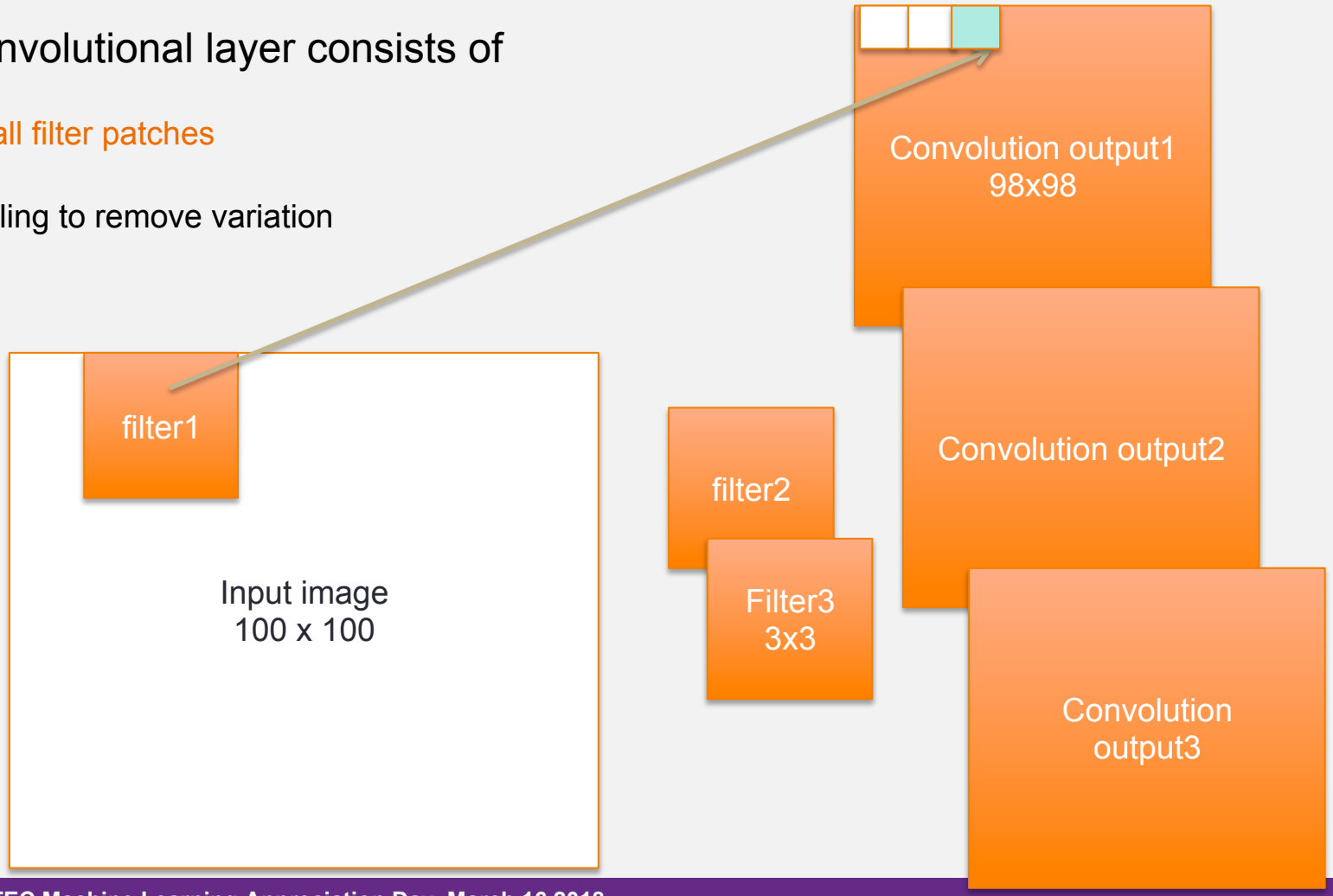


Convolutional filters

Convolutional layer consists of

Small filter patches

Pooling to remove variation

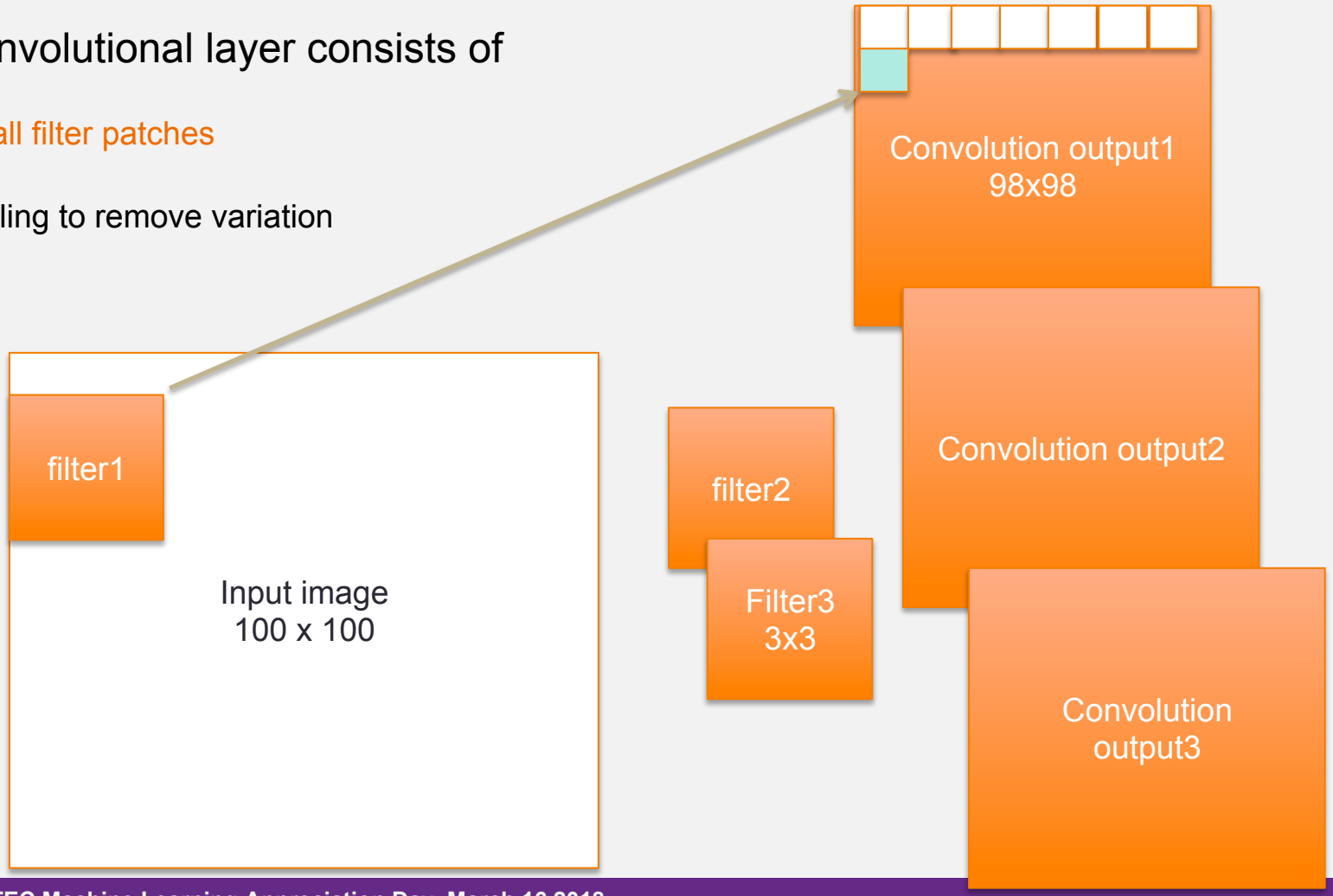


Convolutional filters

Convolutional layer consists of

Small filter patches

Pooling to remove variation

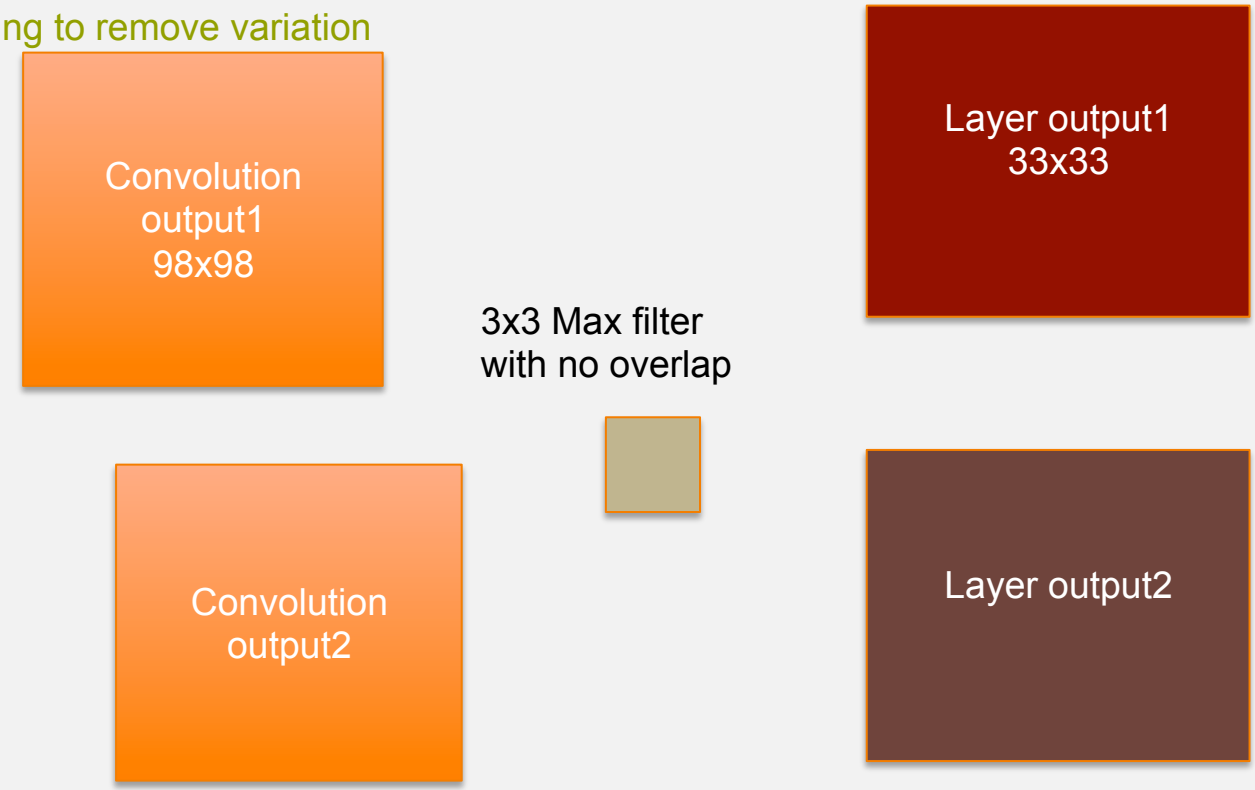


Pooling/subsampling

Convolutional layer consists of

Small filter patches

Pooling to remove variation

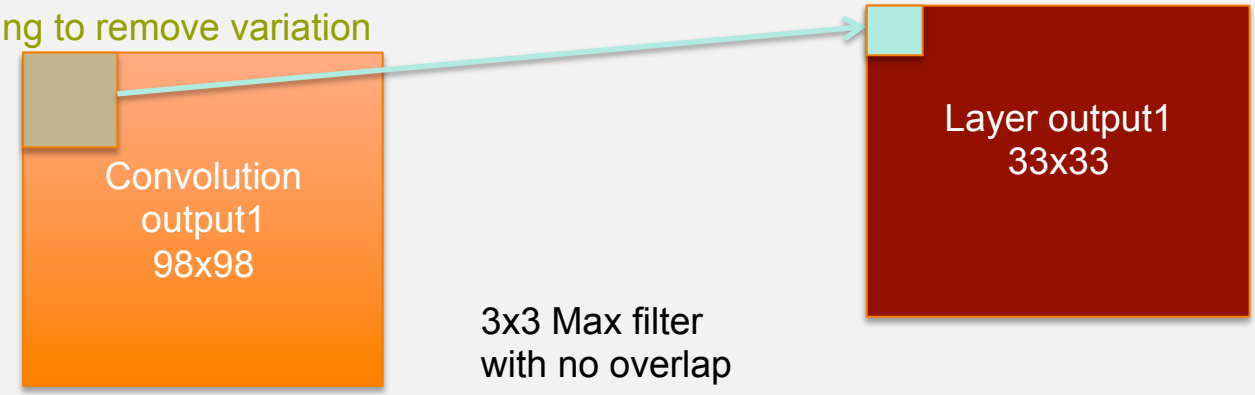


Pooling/subsampling

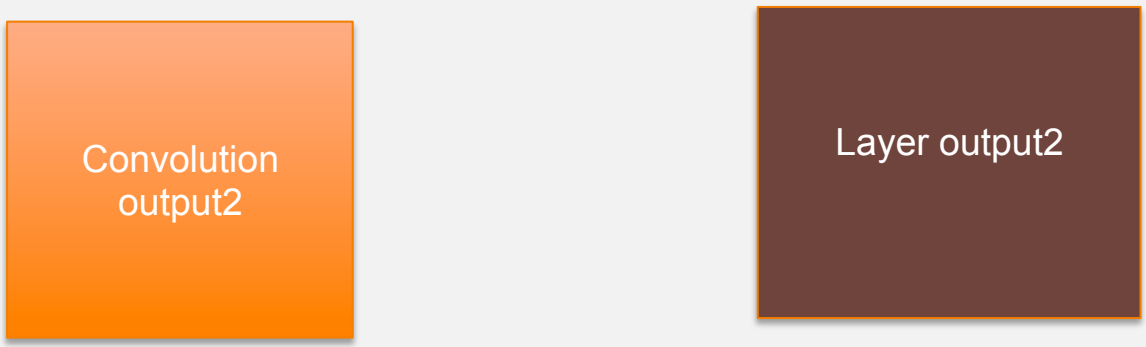
Convolutional layer consists of

Small filter patches

Pooling to remove variation



3x3 Max filter
with no overlap

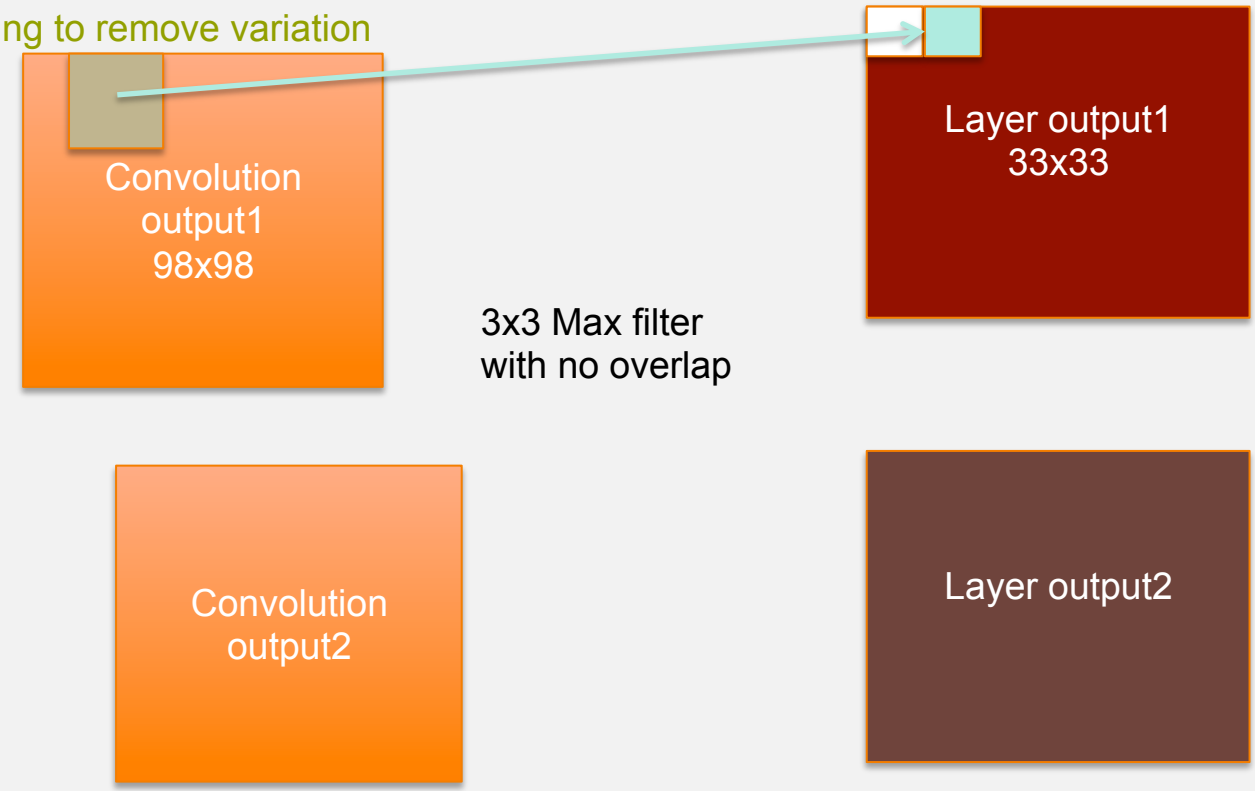


Pooling/subsampling

Convolutional layer consists of

Small filter patches

Pooling to remove variation

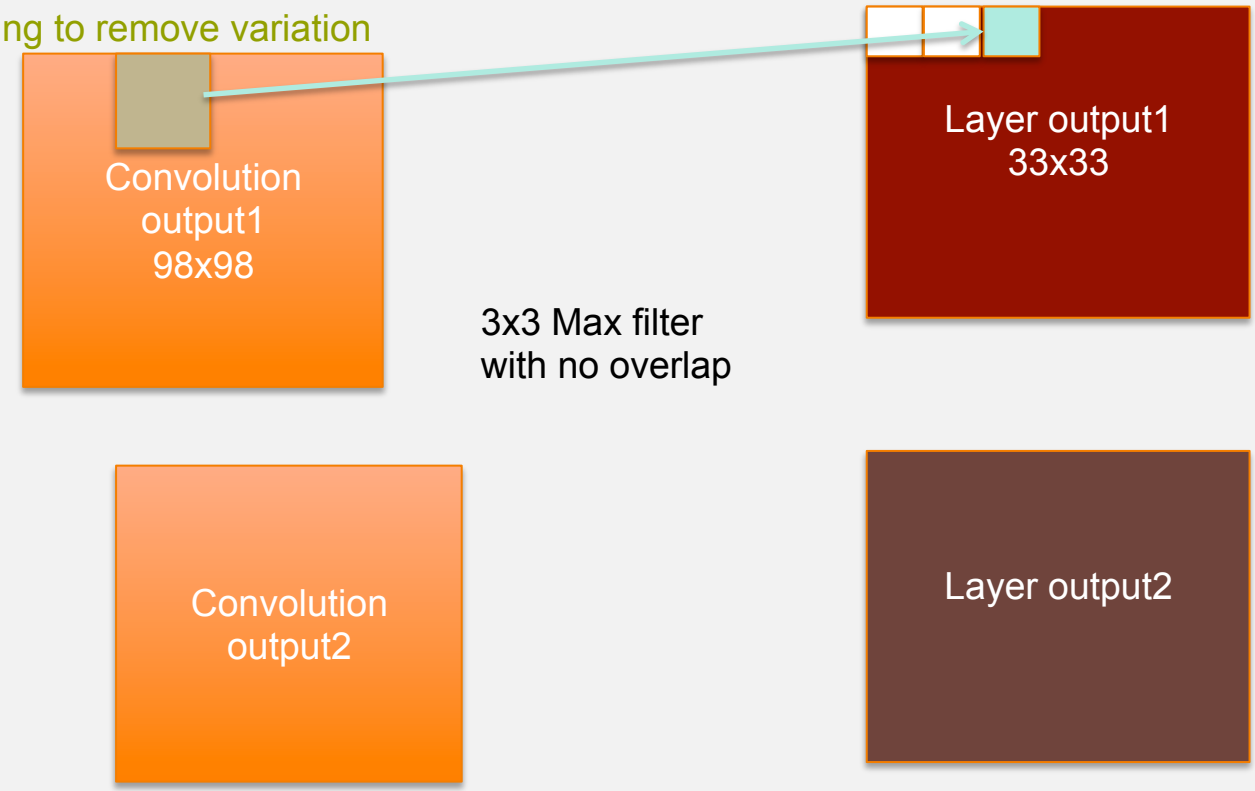


Pooling/subsampling

Convolutional layer consists of

Small filter patches

Pooling to remove variation



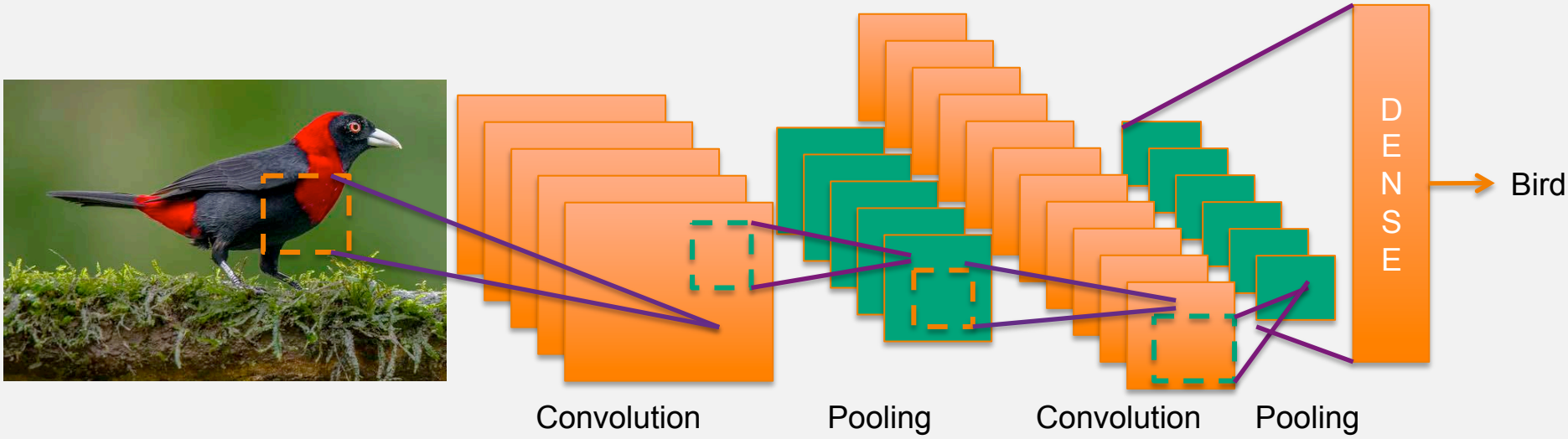
CNN

Filter size, number of filters, and pooling rate are all parameters

Usually followed by a fully connected network at the end

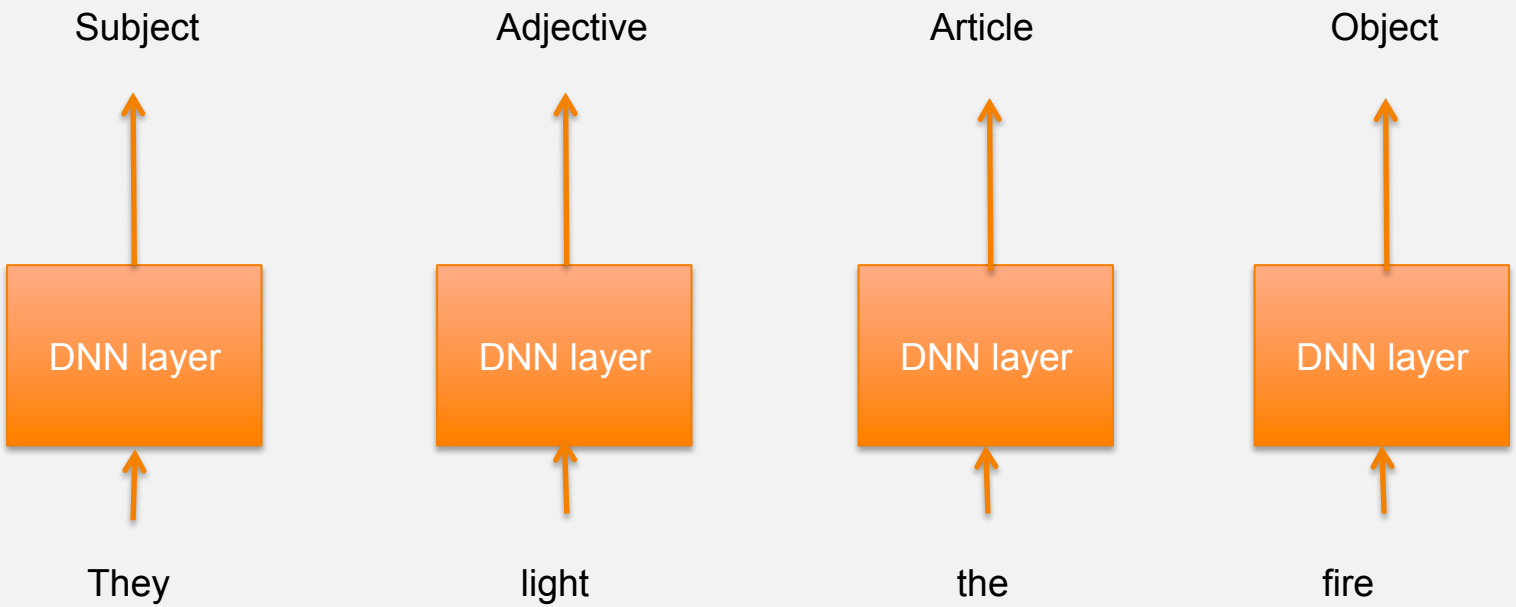
CNN is good at learning low level features

DNN combines the features into high level features and classify



Recurrent neural network (RNN)

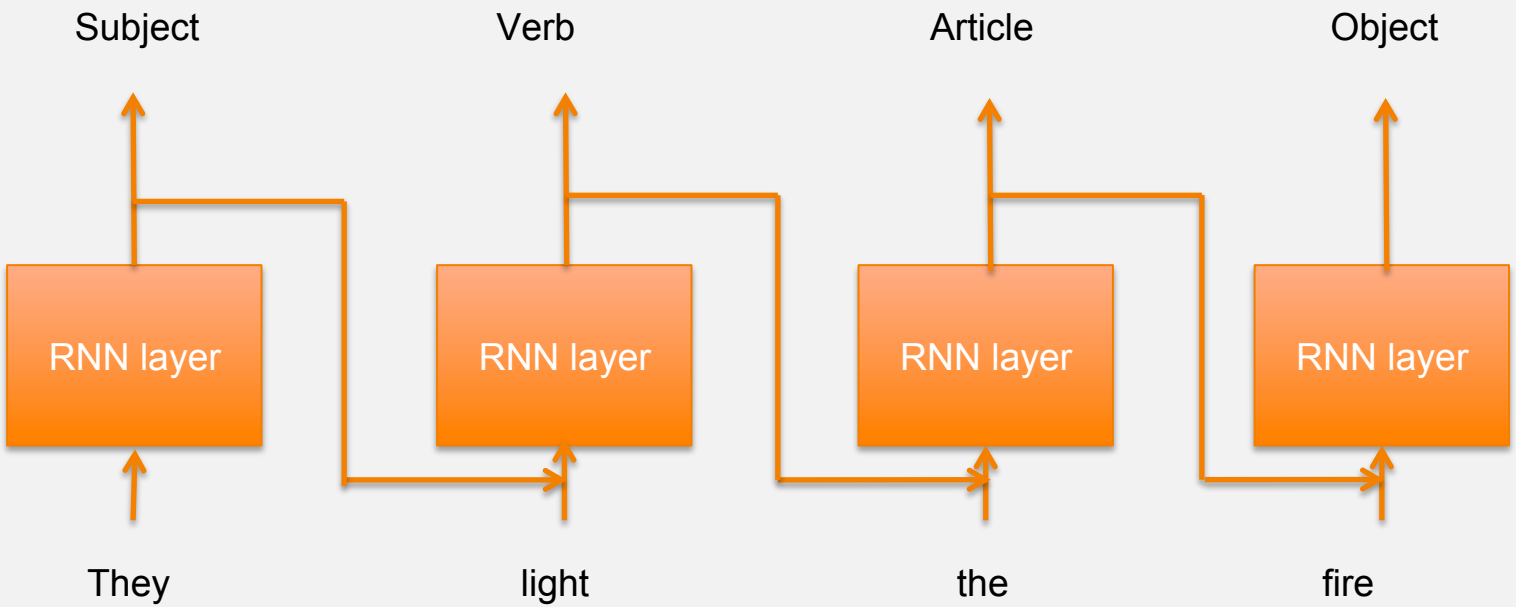
DNN framework



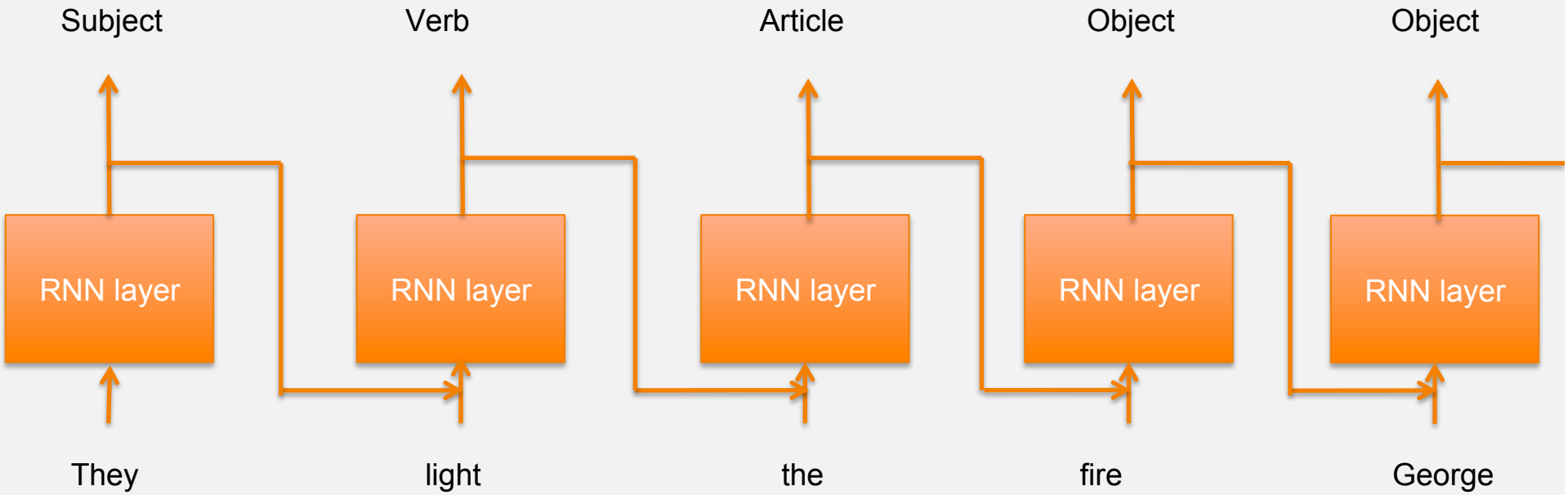
Problem: need a way to remember the past

Recurrent neural network (RNN)

RNN framework



Recurrent neural network (RNN)



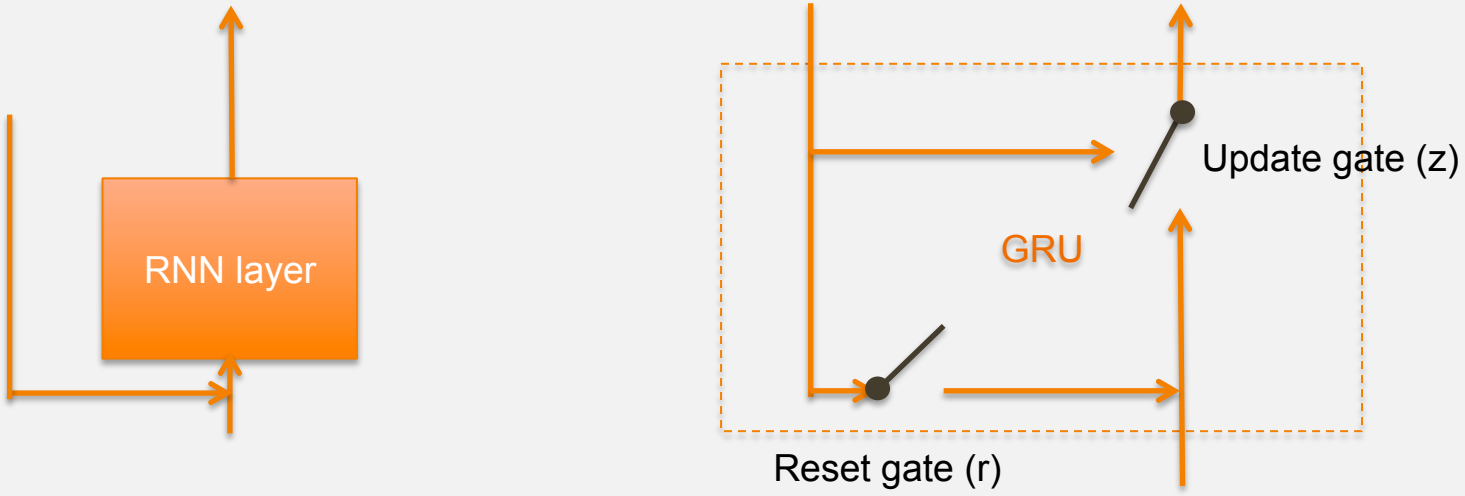
Problem2: needs a way to stop remembering

Can the network learn when to start and stop remembering things?

Gated Recurrent Unit (GRU)

Forms a Gated Recurrent Neural Networks (GRNN)

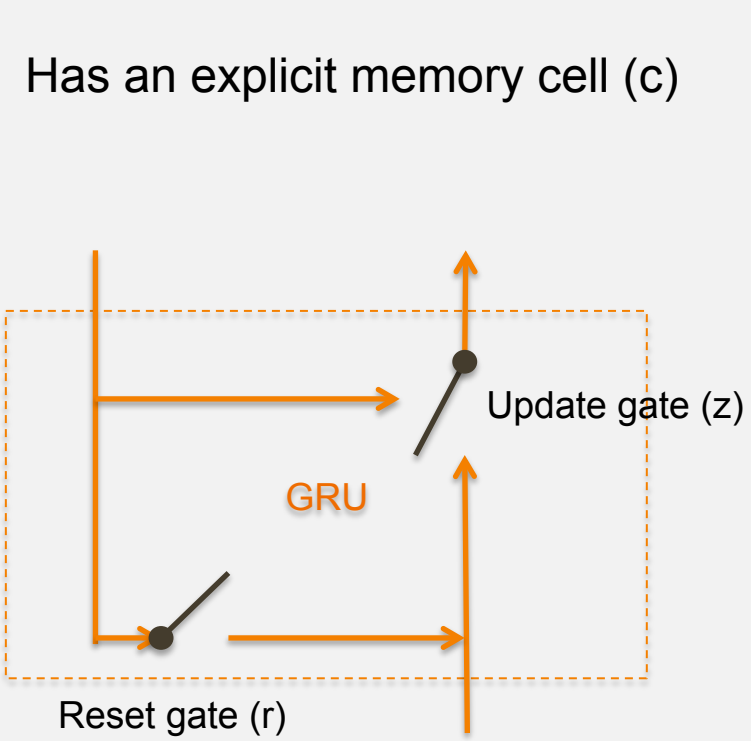
Add gates that can choose to reset (r) or update (z)

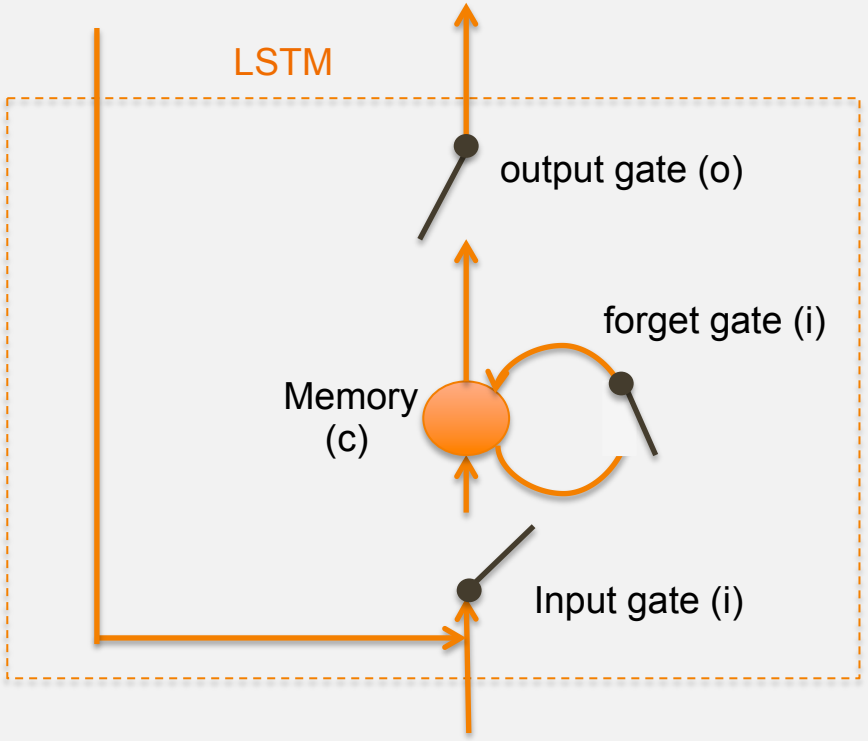


Long Short-Term Memory (LSTM)

Have 3 gates, forget (f), input (i), output (o)

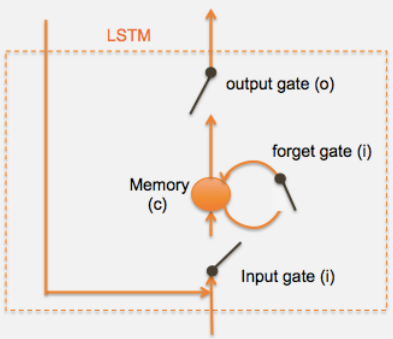
Has an explicit memory cell (c)





Both works for data with time dependency. Try GRU first

The sentiment neuron



Training a LSTM to predict the next character of Amazon reviews.

A LSTM cell that indicates sentiment pops up surprisingly.

This is one of Crichton's best books. The characters of Karen Ross, Peter Elliot, Munro, and Amy are beautifully developed and their interactions are exciting, complex, and fast-paced throughout this impressive novel. And about 99.8 percent of that got lost in the film. Seriously, the screenplay AND the directing were horrendous and clearly done by people who could not fathom what was good about the novel. I can't fault the actors because frankly, they never had a chance to make this turkey live up to Crichton's original work. I know good novels, especially those with a science fiction edge, are hard to bring to the screen in a way that lives up to the original. But this may be the absolute worst disparity in quality between novel and screen adaptation ever. The book is really, really good. The movie is just dreadful.

https://blog.openai.com/unsupervised-sentiment-neuron/

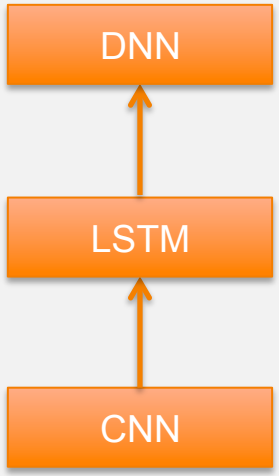
A typical deep learning architecture

Usually combined all three for real applications

DNN – good for classification

LSTM – good for time dependency

CNN – good for local feature learning



Encoder-decoder

We know neural networks can learn representations internally

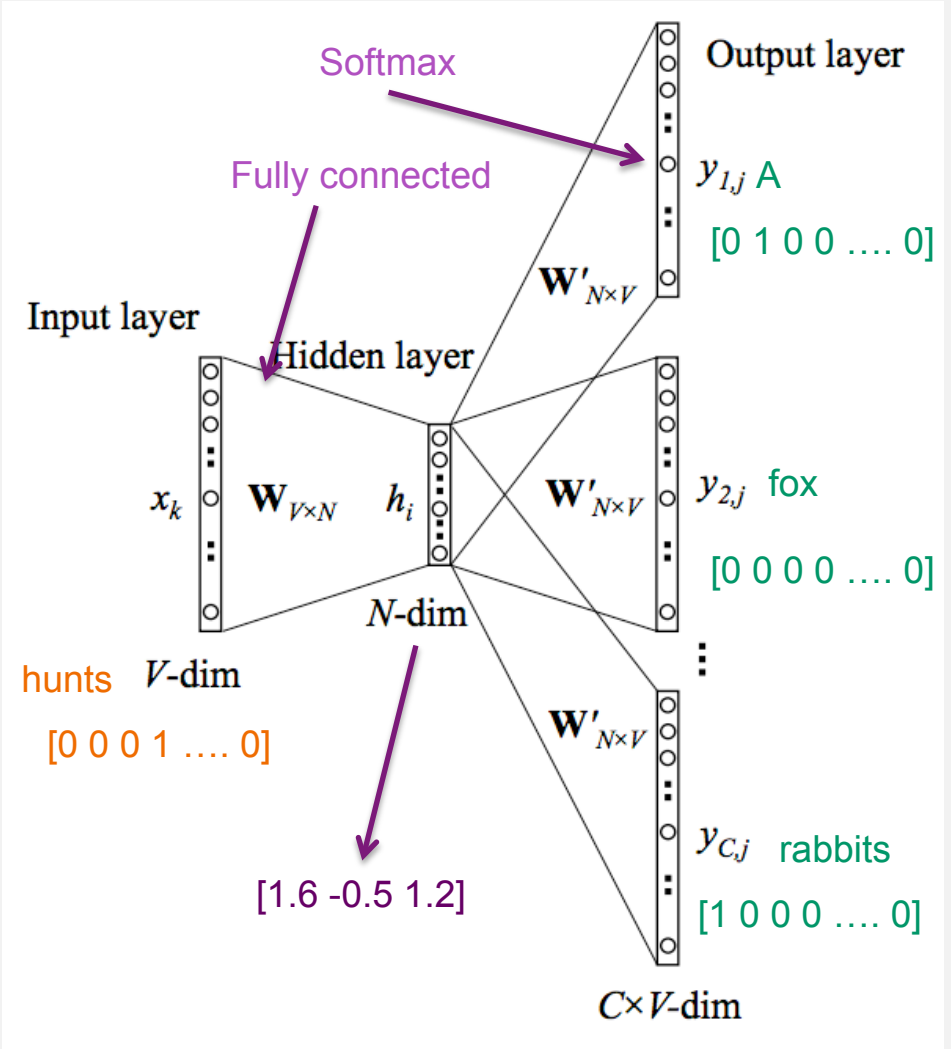
Can we use those internal representations?

Word2Vec

Maps a word to a vector representation using neural networks

A word meaning is from the context around it.

A fox hunts for rabbits

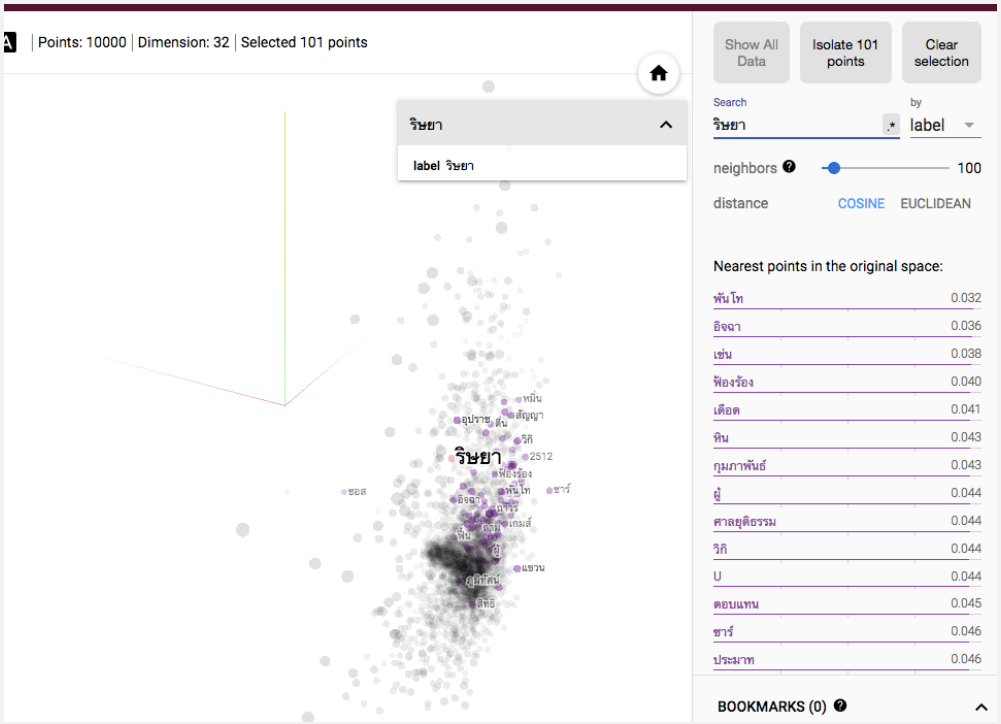


Mikolov, Linguistic Regularities in Continuous Space Word Representations, 2013

Word2Vec

Vectors from similar words are near each other.

Simple Word2Vec demo http://bit.ly/2s0SNHl

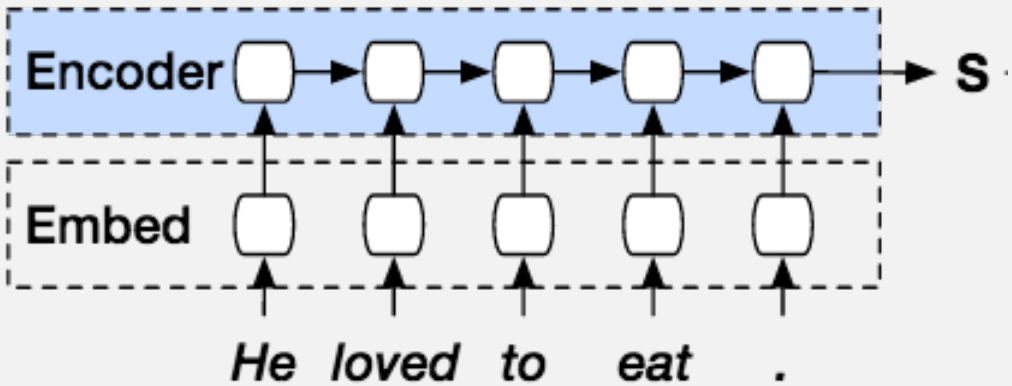


Sentence encoder

How can we represent sentences?

Similarly, use the internal states as the representation

- LSTMs/GRUs are good for sequence task

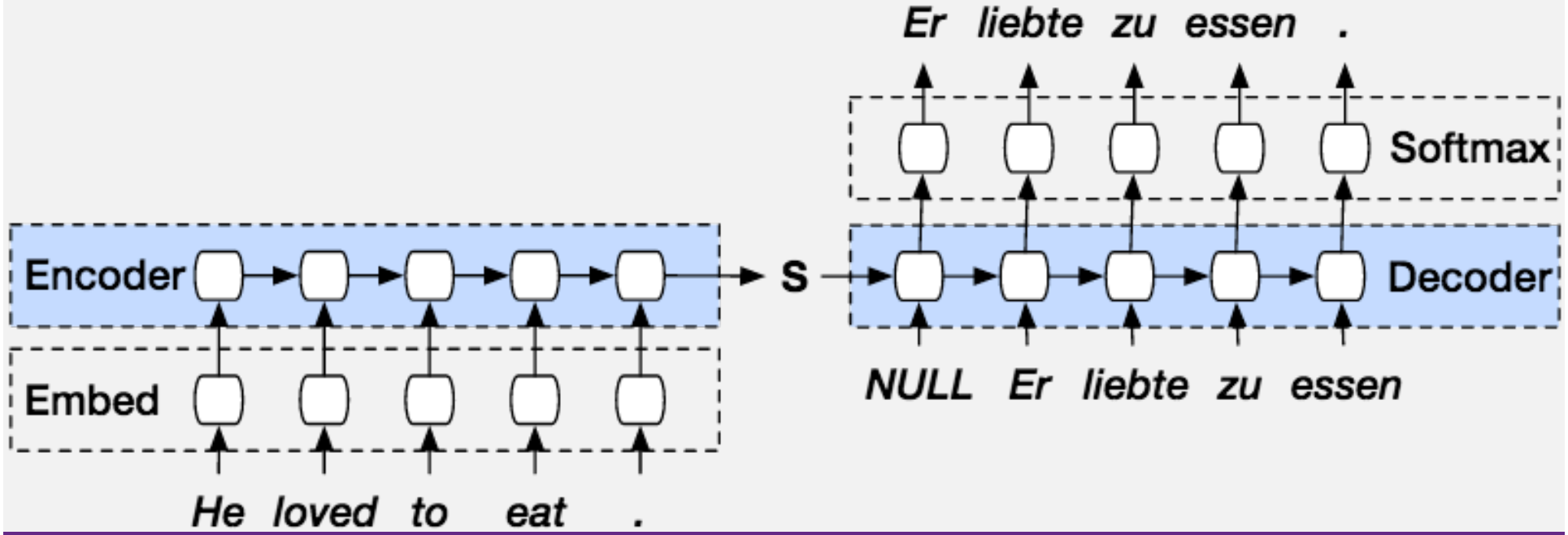


Machine translation using encoder-decoder

Use another LSTM as the decoder

Input to the LSTM is the encoded sentence and the previously output word

Ilya Sutskever, Sequence to Sequence Learning with Neural Networks, 2014, https://arxiv.org/abs/1409.3215



Deep learning building blocks

Fully connected networks

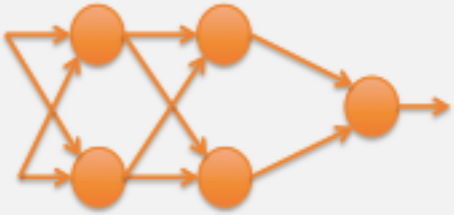
Convolutional neural networks

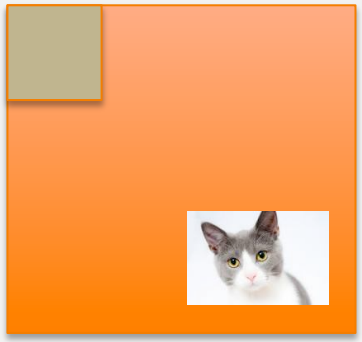
Recurrent neural networks

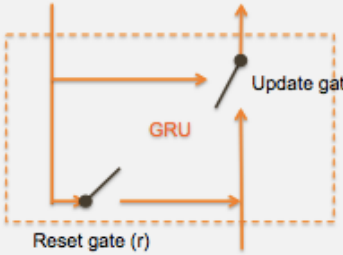
Gated recurrent units

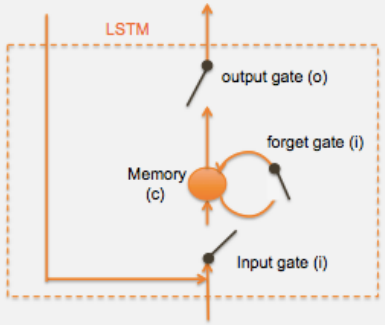
Long short-term memory networks

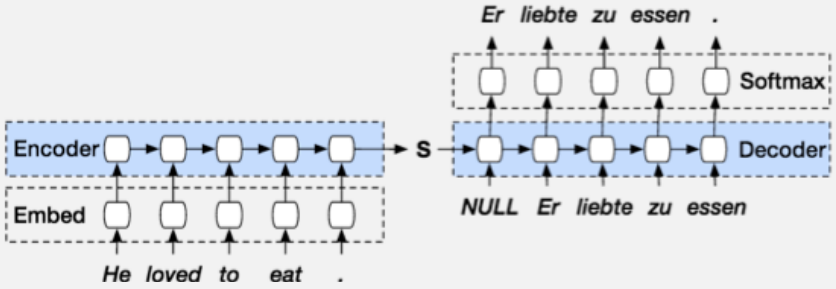
Encoder-decoder











Case study: Diagram description for the visually impaired

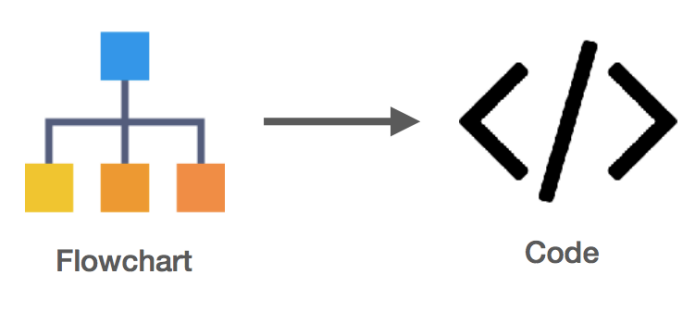
Teaching computer science to a visually impaired person is hard.

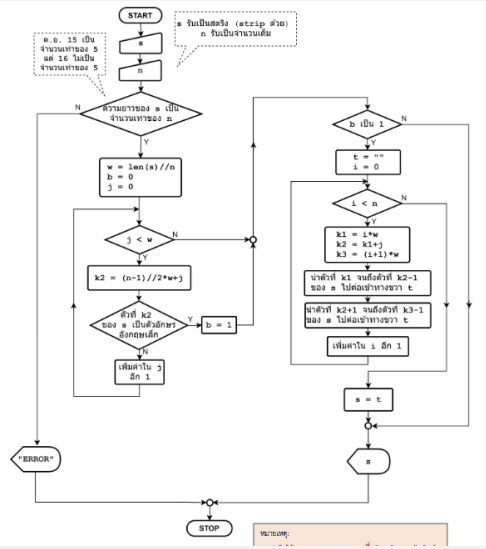
- Lots of diagrams: A complicated diagram can take up to 6 hours to explain
- Complicated diagrams cannot be printed in braille

Can we use conventional teaching material to teach visually impaired people?

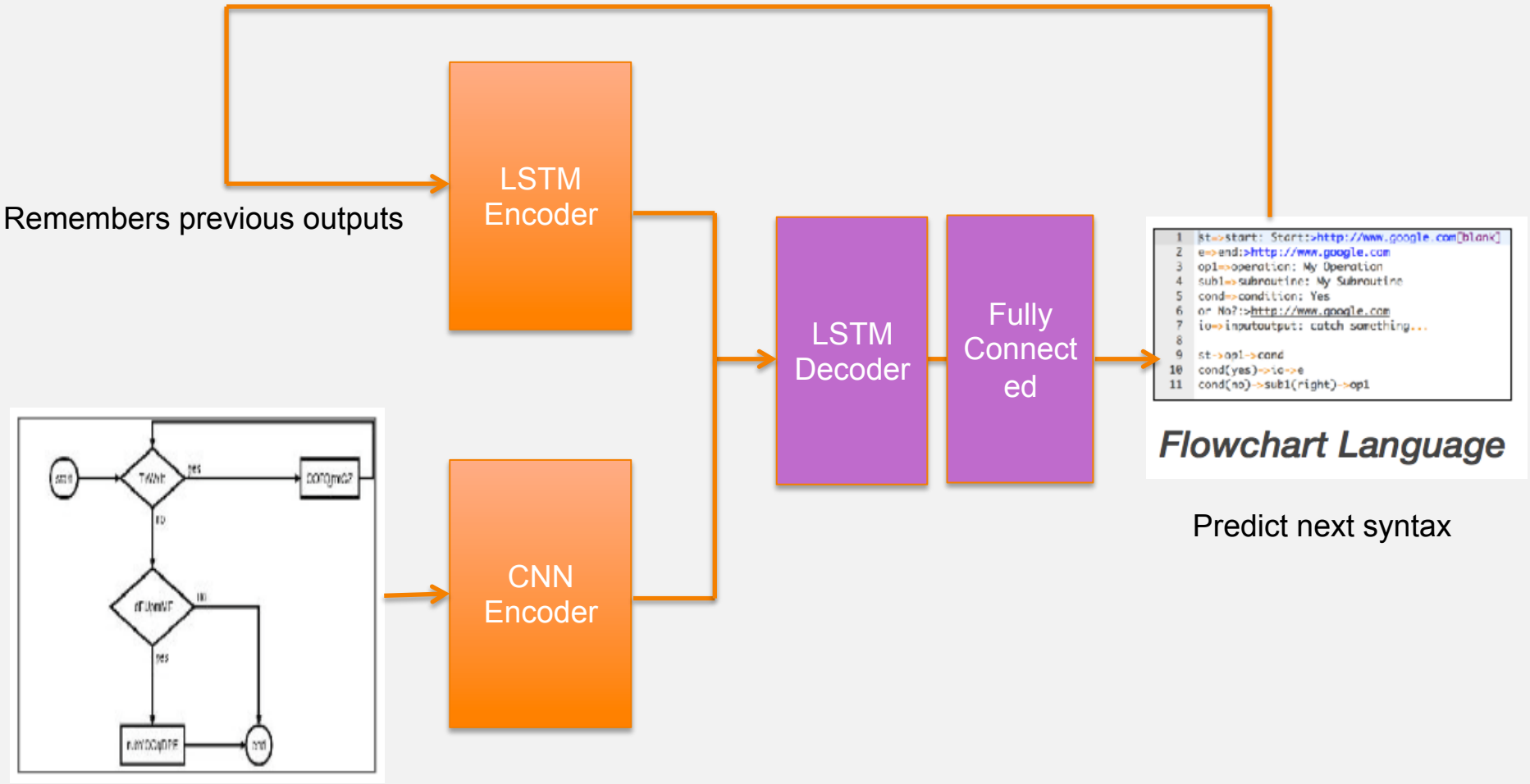
- Need to first understand pictures of diagrams and flowcharts

Change from pixels to machine interpretable representation

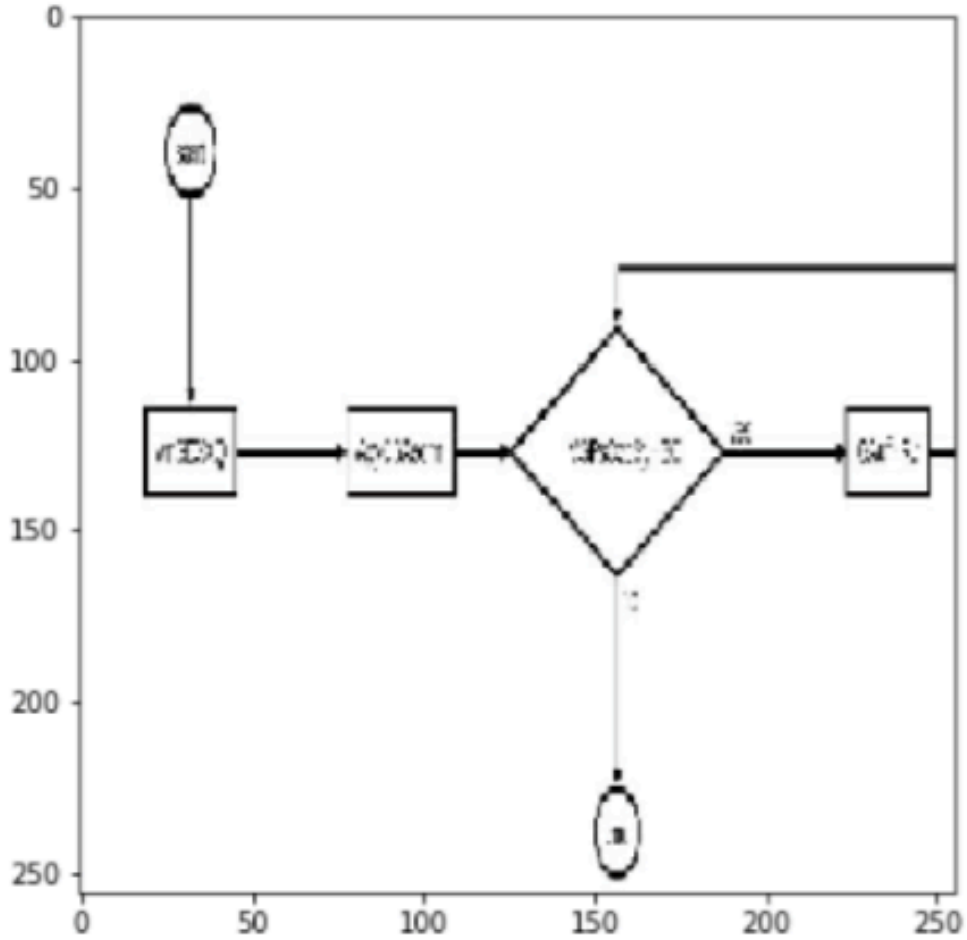




Model: Pix encoder – LSTM decoder

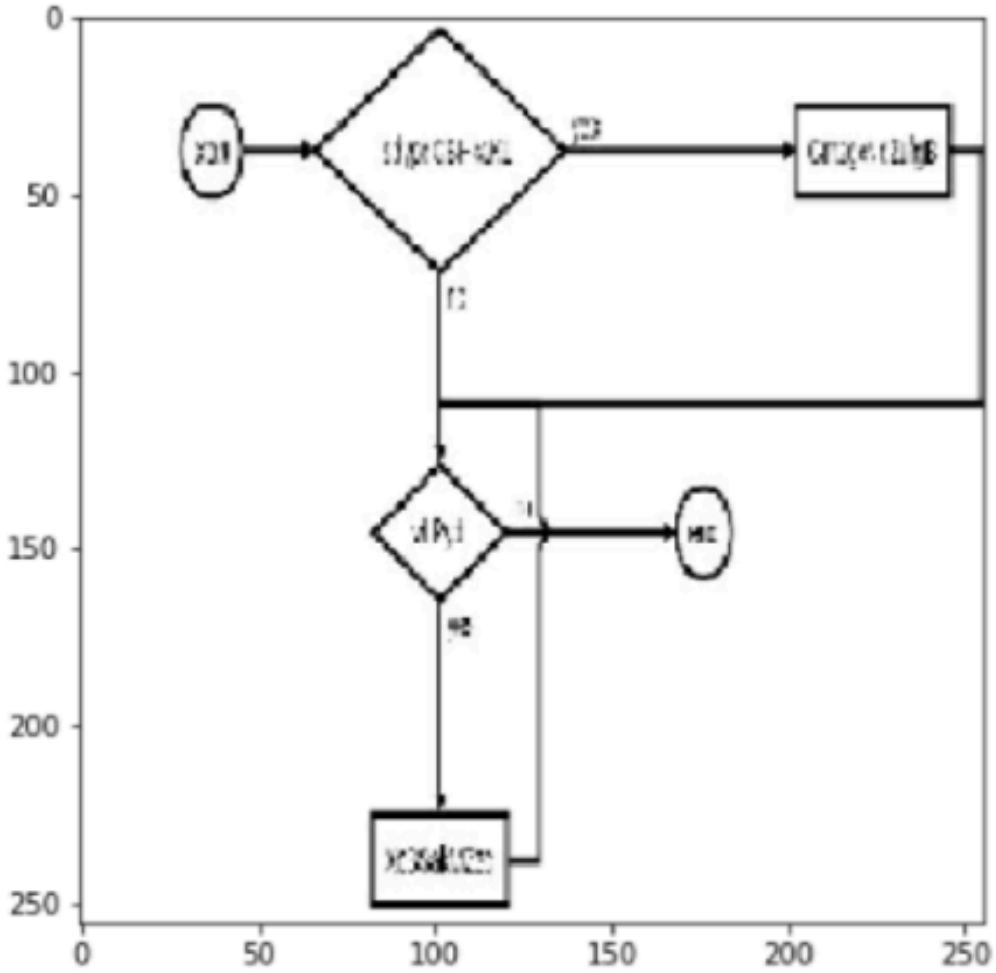


Result




```
Actual :  ['statement', 'statement', 'for', 'statement', 'end']
Predict:  ['statement', 'statement', 'for', 'statement', 'end', '<END>']
```

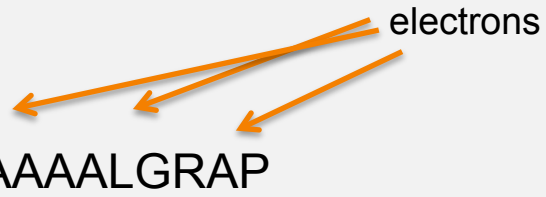
Result

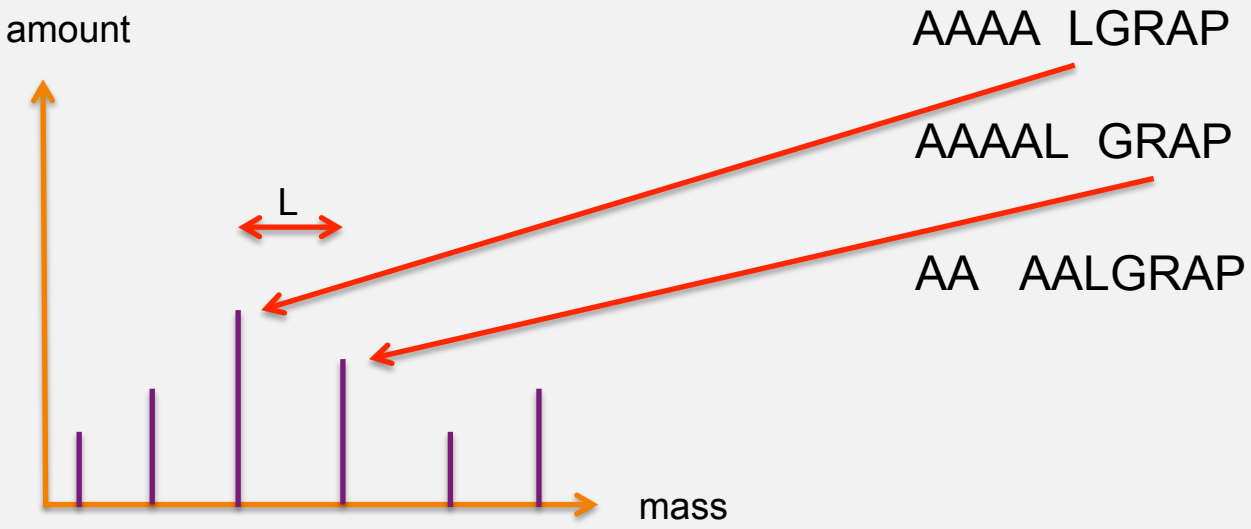



```
Actual :  ['if', 'statement', 'end', 'for', 'statement', 'end']
Predict:  ['if', 'statement', 'end', 'while', 'statement', 'end', '<END>']
```

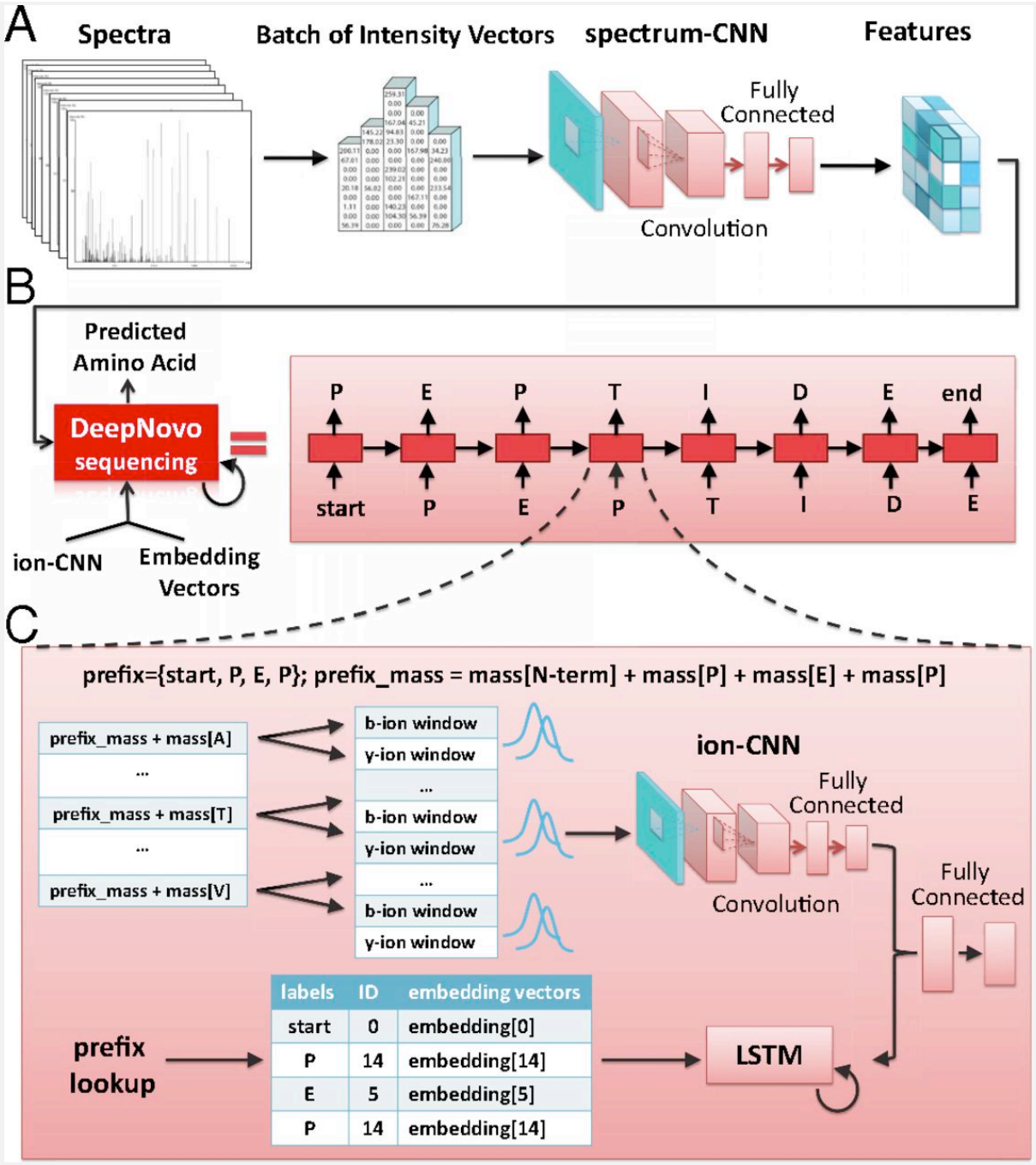
Case study II: De novo peptide sequencing

Guess the sequence of a peptide from its mass





Ngoc Hieu Tran, De novo peptide sequencing by deep learning. 2017


A
Spectra
Batch of Intensity Vectors
spectrum-CNN
Features
Fully Connected
Convolution
B
Predicted Amino Acid
DeepNovo sequencing
ion-CNN
Embedding Vectors
P E P T I D E end
start P E P T I D E
C
prefix={start, P, E, P}; prefix_mass = mass[N-term] + mass[P] + mass[E] + mass[P]
prefix_mass + mass[A]
...
prefix_mass + mass[T]
...
prefix_mass + mass[V]
b-ion window
y-ion window
...
b-ion window
y-ion window
...
b-ion window
y-ion window
ion-CNN
Fully Connected
Convolution
Fully Connected
prefix lookup
labels ID embedding vectors
start 0 embedding[0]
P 14 embedding[14]
E 5 embedding[5]
P 14 embedding[14]
LSTM

Case study III: Speeding up simulations

Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks

We present a technique for efficiently synthesizing images of atmospheric clouds using a combination of Monte Carlo integration and neural networks.

November 13, 2017
ACM SIGGRAPH Asia 2017

Authors

Thomas Müller
Disney Research

Brian McWilliams
Disney Research

Markus Gross
Disney Research

Jan Novák
Disney Research

https://www.disneyresearch.com/publication/deep-scattering/

"We used clouds to generate clouds in order to not to rely on clouds"

From expensive and slow HPC to fast and cheap GPU

Clouds

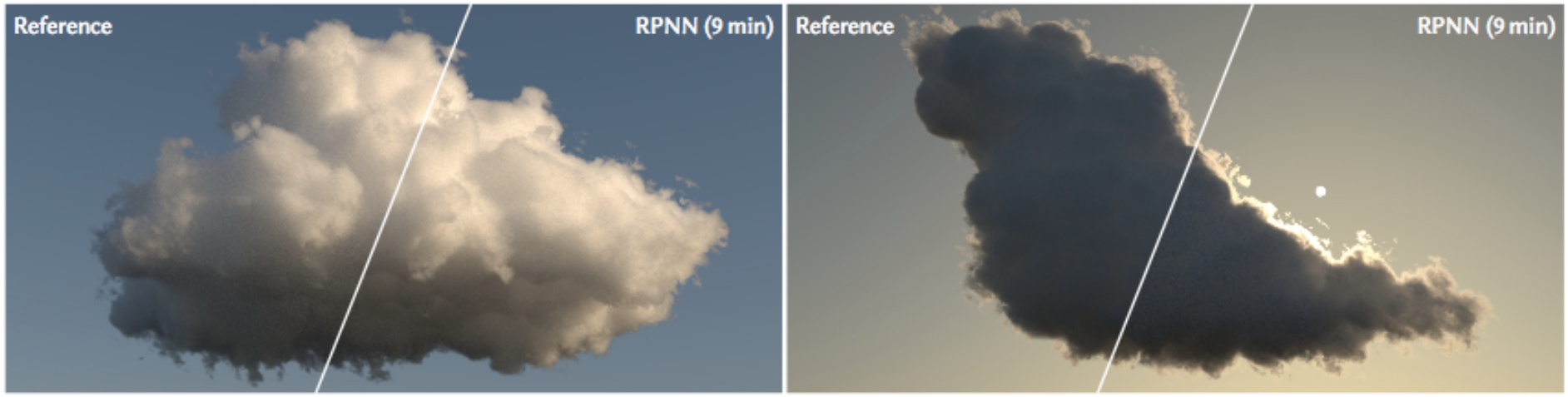
(a) 1 interaction
Render time = 0.017 h

(a) 4 interactions
Render time = 0.89 h

(a) 16 interactions
Render time = 6.5 h

(a) 64 interactions
Render time = 16 h

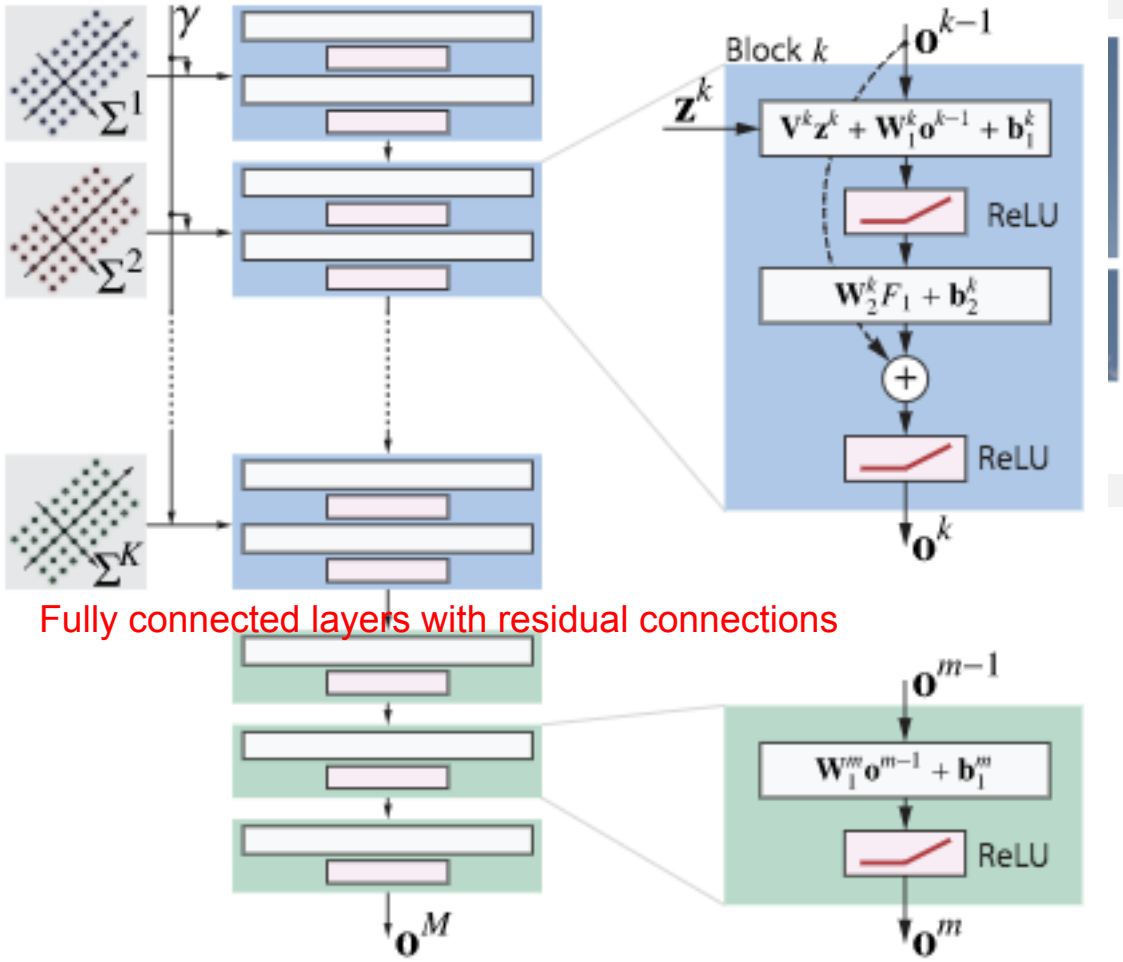
(a) All interactions
Render time = 34 h



Clouds

(a) 1 interaction
Render time = 0.017 h

(a) 4 interactions
Render time = 0.89 h



Fully connected layers with residual connections

Other notable neural networks building blocks

Attention mechanism

Variational Auto Encoder (VAE)

Deep Q network (DQN)

Generative Adversarial Network (GAN)

Learning distributions

Supervised learning tasks usually have one correct answer

Learning distributions

Supervised learning tasks usually have one correct answer

Sometimes there are more than one possibility

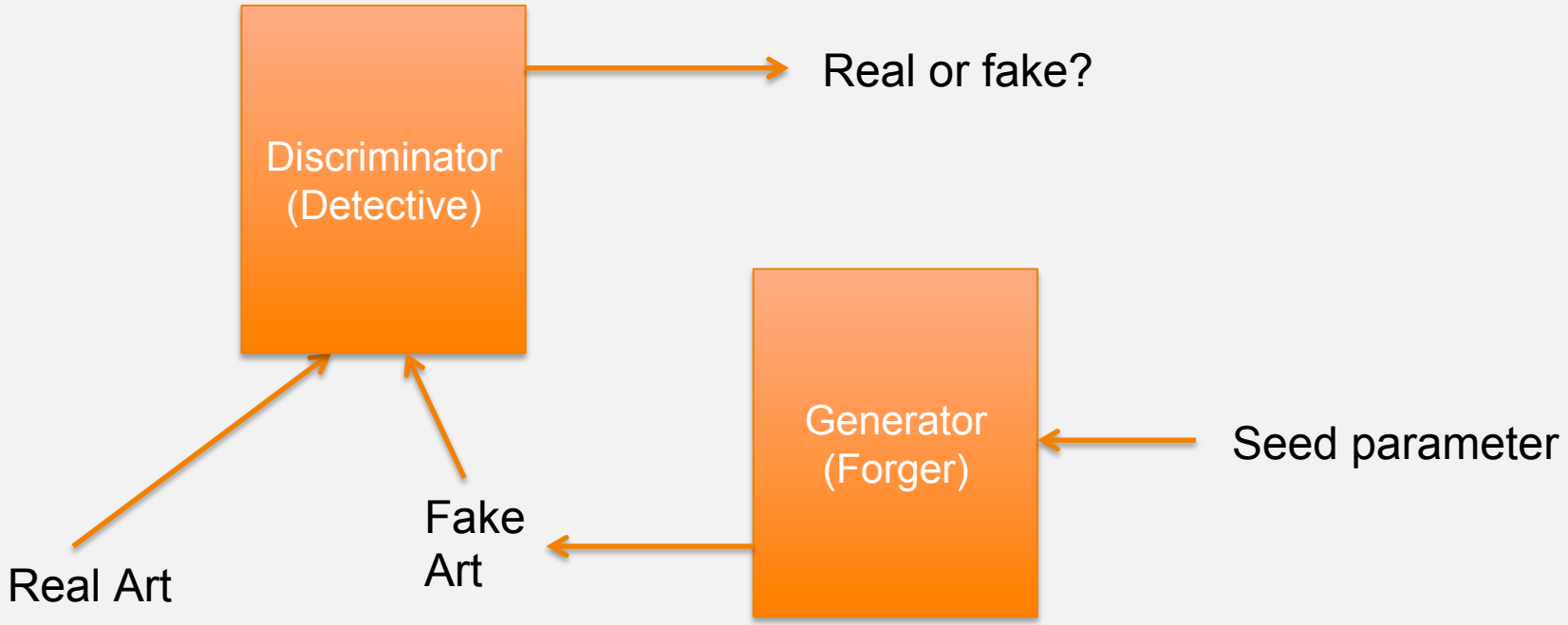
What is the next frame of a video?

What is the missing pixels in an image?

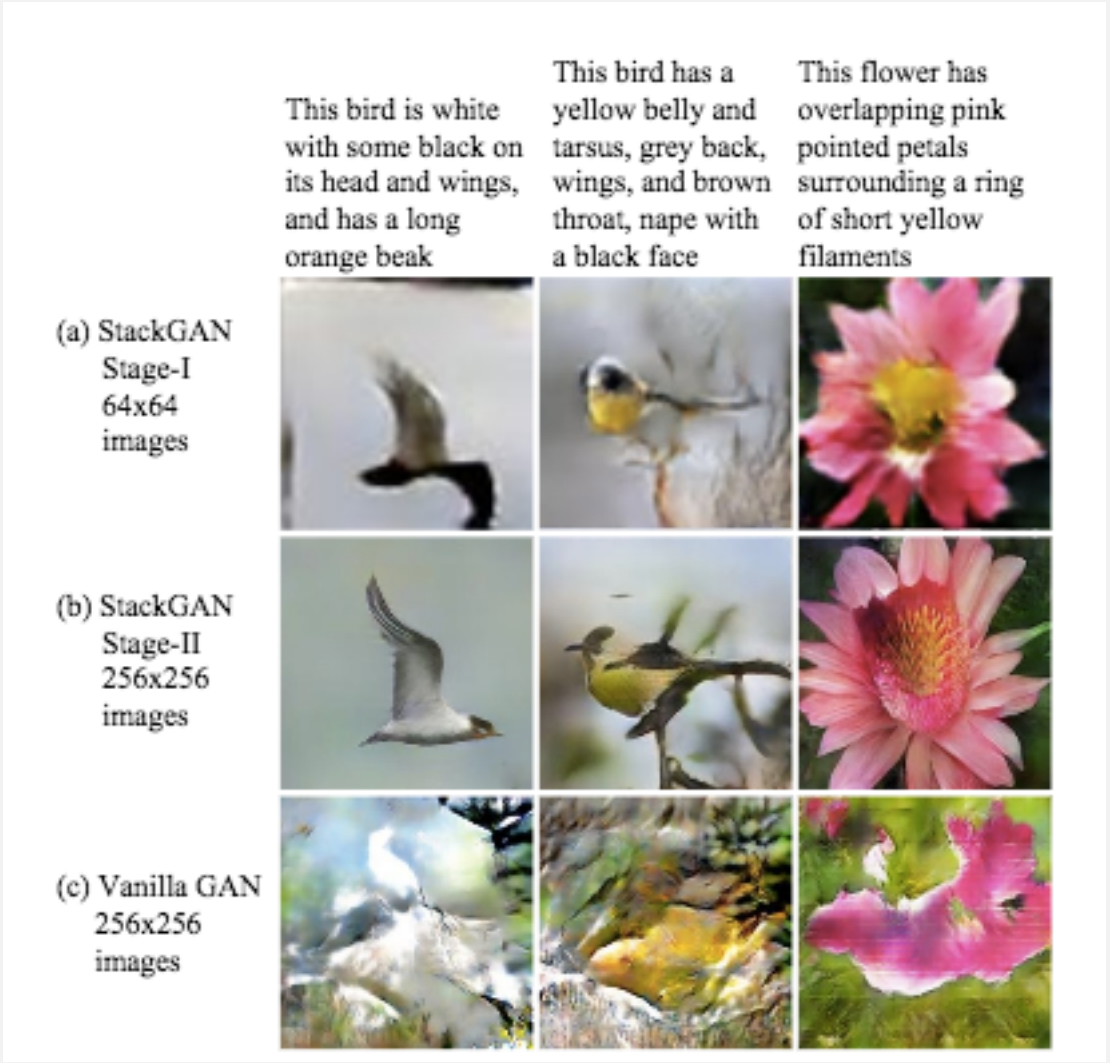
Which side will a pencil fall over?

GAN as a generative model

GAN learns distributions and we can sample from the distribution to produce a likely output



StackGAN



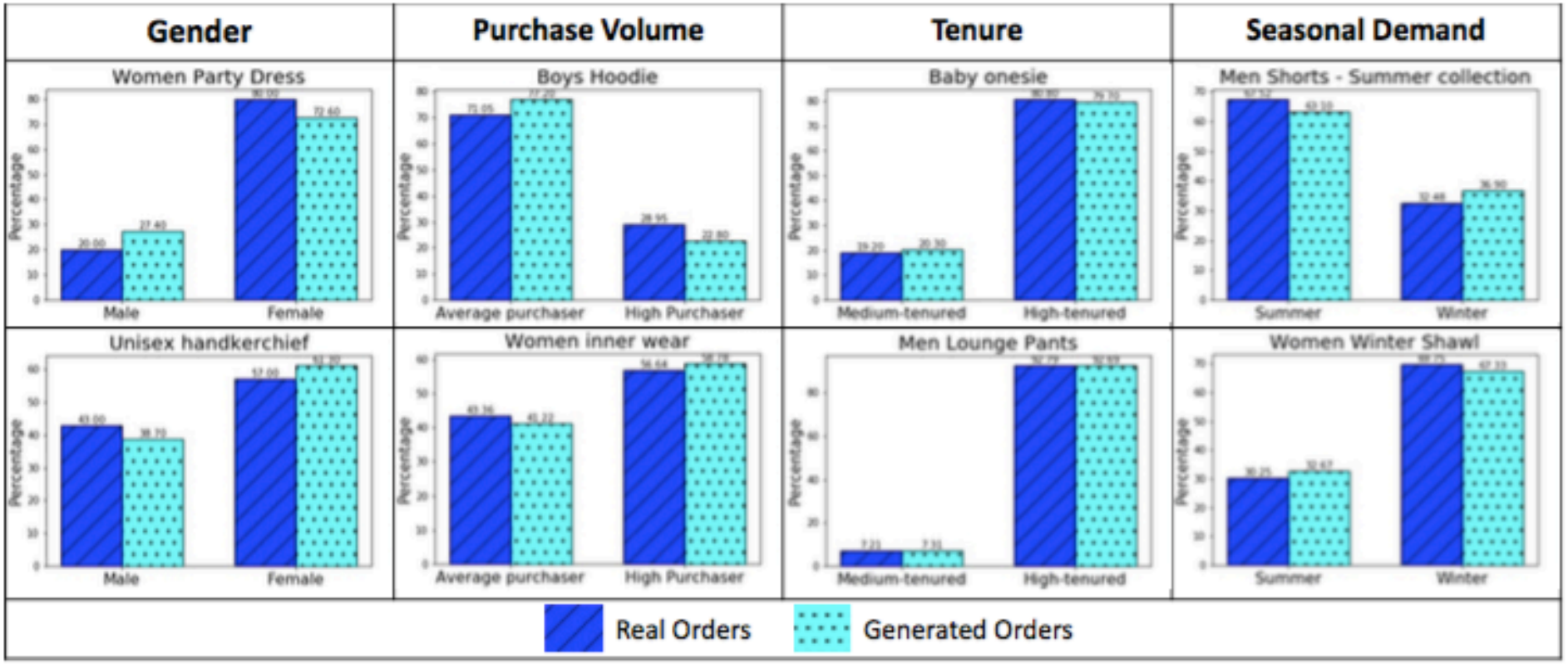
H Zhang, StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, 2017

TextureGAN

W Xian, TextureGAN: Controlling Deep Image Synthesis with Texture Patches, 2017

eCommerceGAN

Predicts sales, prices, and customers using text information for a new product



A Kumar, eCommerceGAN : A Generative Adversarial Network for E-commerce, 2018

Can neural networks help me understand X?

People usually say deep learning is a black box and we don't understand why it's giving the outputs

That's not true…

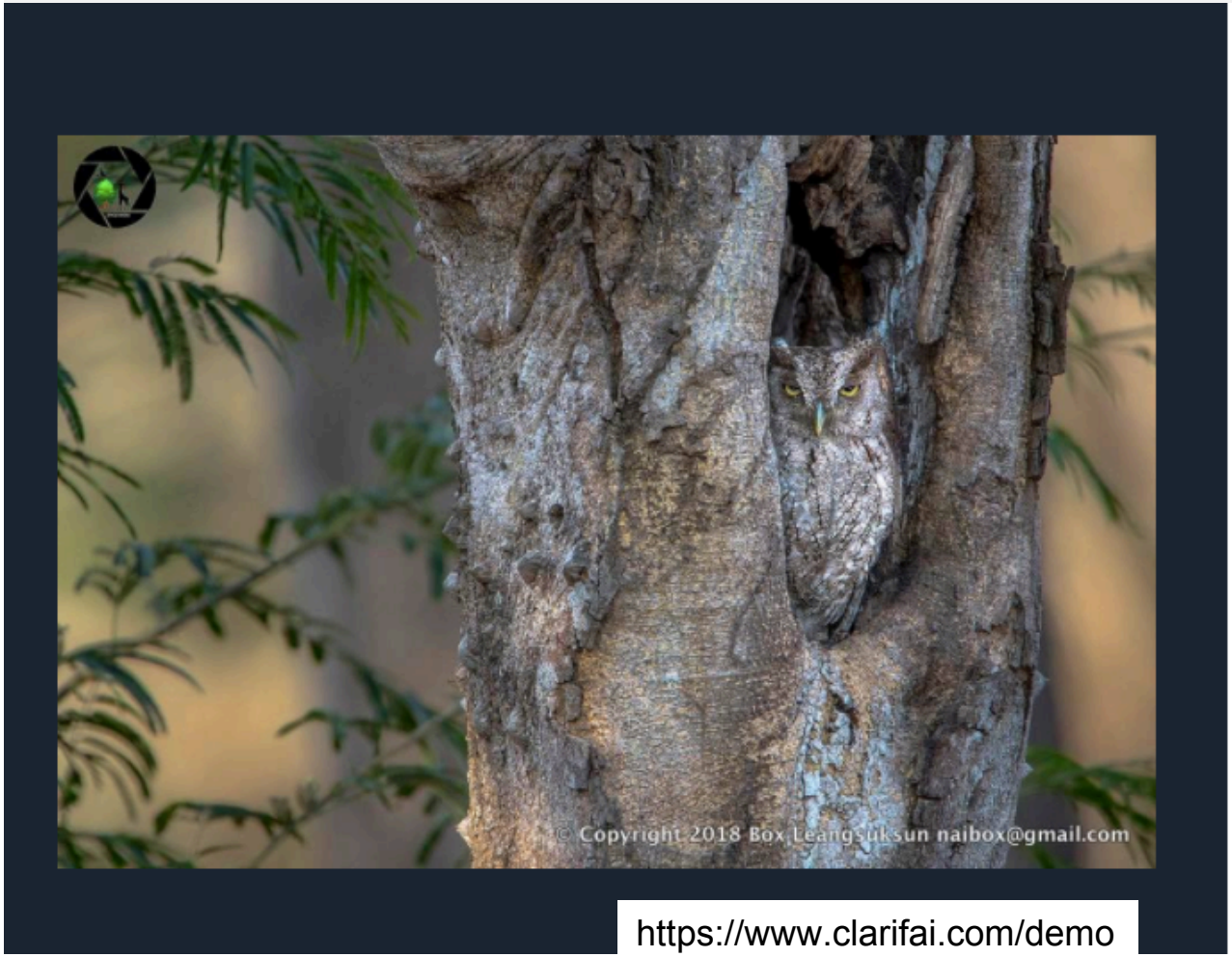
We actually know how and why it thinks that way, but most of the time it's not useful to understand why

Deep learning and humans think differently

Can a neural network recognize the owl?

Yes, easily

Convolutions are very robust in picking up things buried in noise



LANGUAGE

English (en)

PREDICTED CONCEPT	PROBABILITY
tree	0.997
no person	0.990
nature	0.988
wildlife	0.971
outdoors	0.961
wood	0.956
bird	0.942
wild	0.927
leaf	0.893

Yes, easily

Convolutions are very robust in picking up things buried in noise

https://www.clarifai.com/demo

wood	0.956
bird	0.942
wild	0.927
leaf	0.893
animal	0.891
tropical	0.875
closeup	0.863
branch	0.855
bark	0.815
environment	0.815
desktop	0.809
owl	0.806

But it fails at this

https://iotsecurity.eecs.umich.edu/#roadsigns

When can’t we use neural networks?

Short answer: when you don’t have data

Long answer: it depends

Cautionary notes

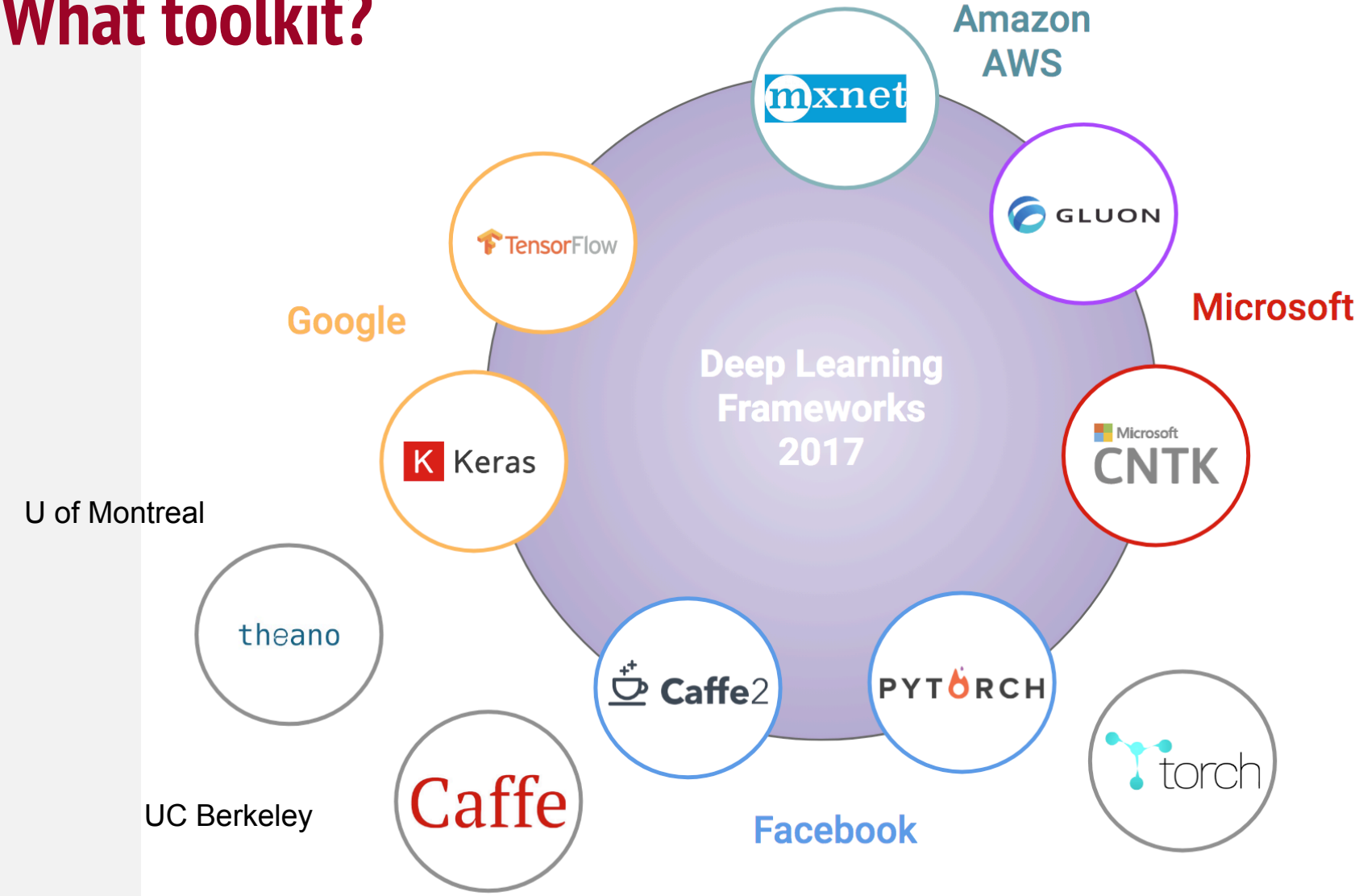
"There is no free lunch."

The **"No Free Lunch"** theorem states that there is no one model that works best for every problem. (Even humans aren't good at every tasks)

Depends on

- Nature of the task
- Nature of the data
- Amount of data

What toolkit?



https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750

Which?

Easiest to use and play with deep learning: Keras

Easiest to use and tweak: pytorch

A graph is created on the fly


```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```

Dynamic computation graph in pytorch. Good for when size of and structure of input data varies. Example parse tree.

http://pytorch.org/about/

Which?

Easiest to use and play with deep learning: Keras

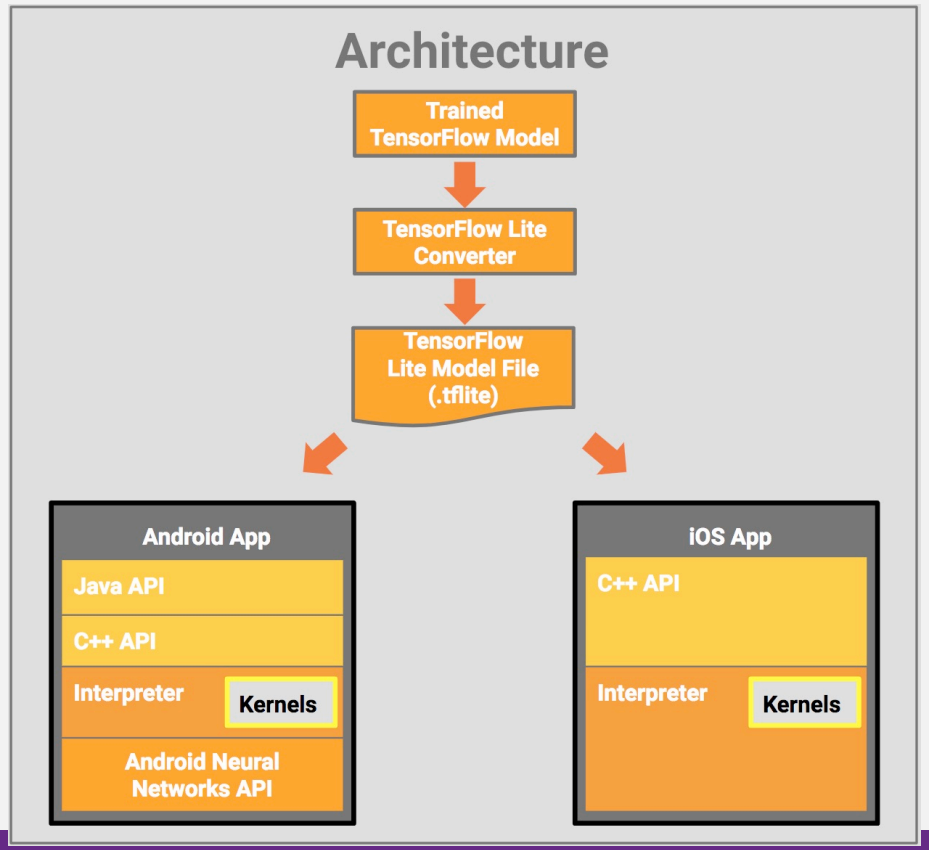
Easiest to use and tweak: pytorch

Easiest to deploy: TensorFlow

TensorFlow Lite and TensorFlow Mobile

Now support dynamic graph: TensorFlow Fold

TensorFlow Eager



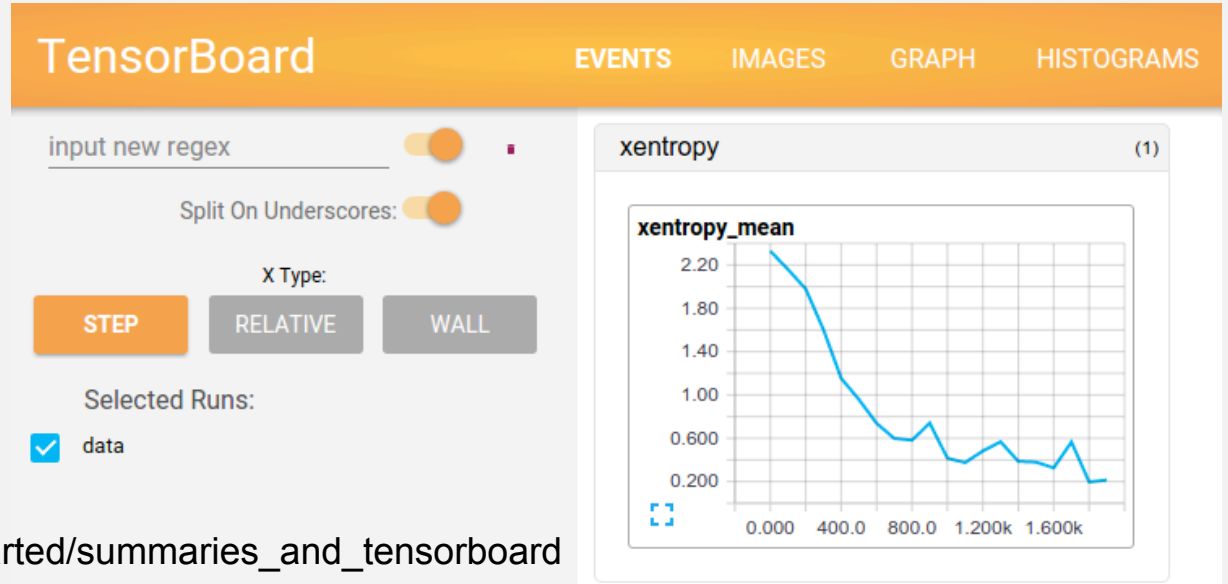
Which?

Easiest to use and play with deep learning: Keras

Easiest to use and tweak: PyTorch

Easiest to deploy: TensorFlow

Best tools: TensorFlow Tensorboard visualization



https://www.tensorflow.org/get_started/summaries_and_tensorboard

Which?

Easiest to use and play with deep learning: Keras

Easiest to use and tweak: PyTorch

Easiest to deploy: TensorFlow

Biggest community: TensorFlow

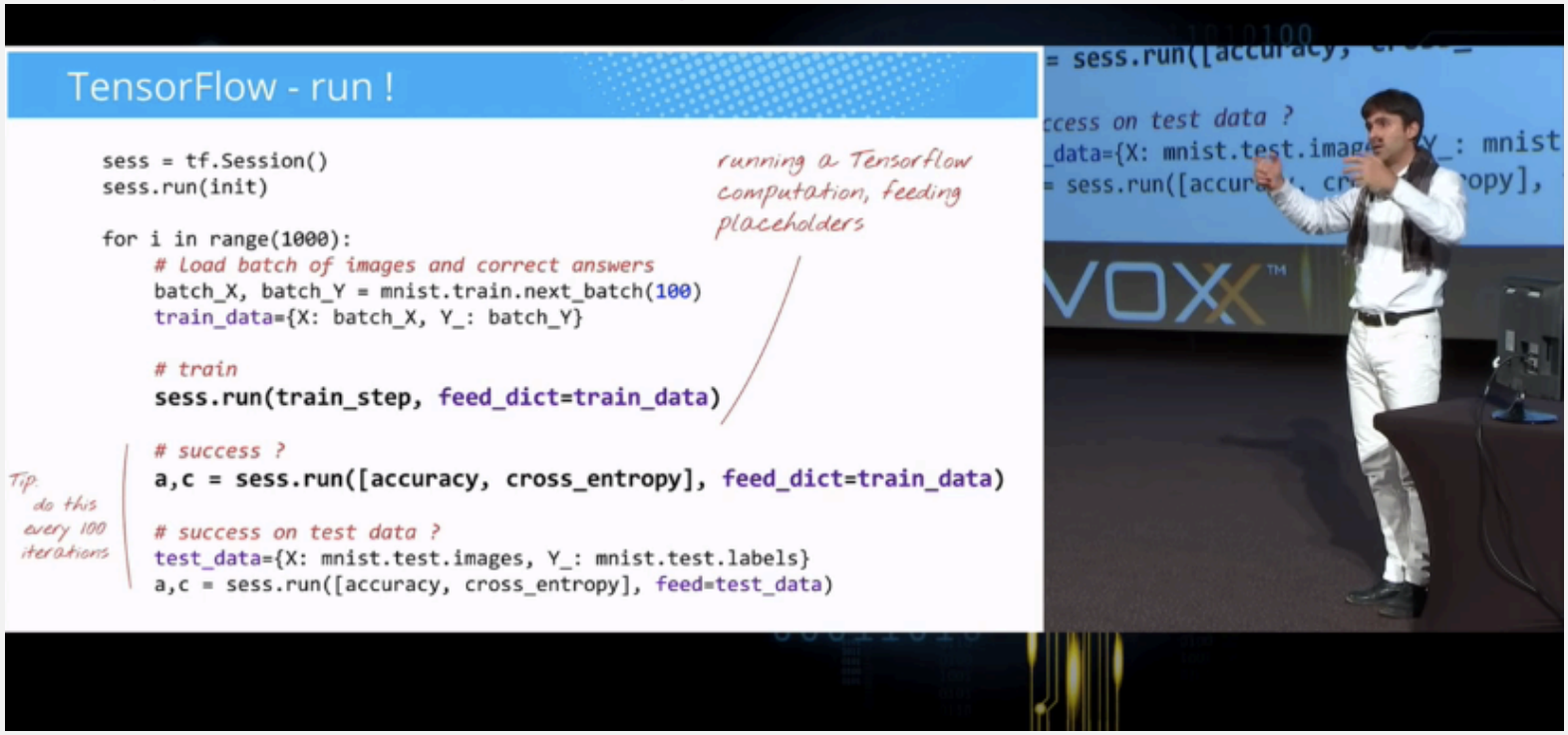
TPUs & (Quantum computers): Google(?)

Where to learn more

Time commitment 3 hours:

Tensorflow and deep learning - without a PhD by Martin Görner

https://www.youtube.com/watch?v=vq2nnJ4g6N0



Where to learn more

Fei-Fei Li and Andrew Karpathy's Computer Vision class (Stanford cs231n 2015/2016)

- Full course about deep learning (vision oriented)
- http://cs231n.stanford.edu/

My channel… Pattern Recognition course (Chula Fall 2017)

- Full ML course (half traditional, half deep learning) (Also currently teaching NLP and ASR courses)
- http://goo.gl/Si5azM

Coursera, OpenAi, Udacity, etc…

Google ML course

https://developers.google.com/machine-learning/crash-course/ml-intro

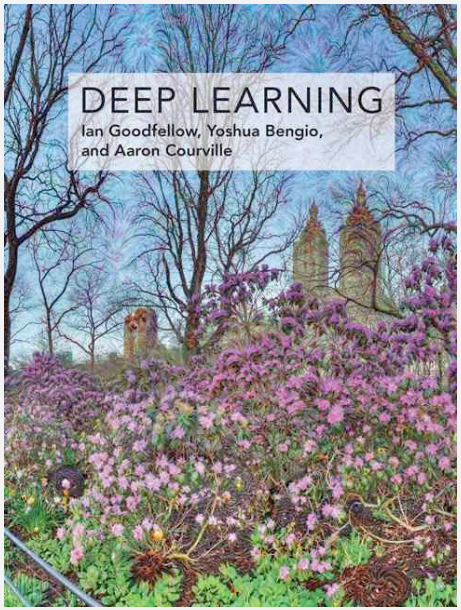
Where to learn more

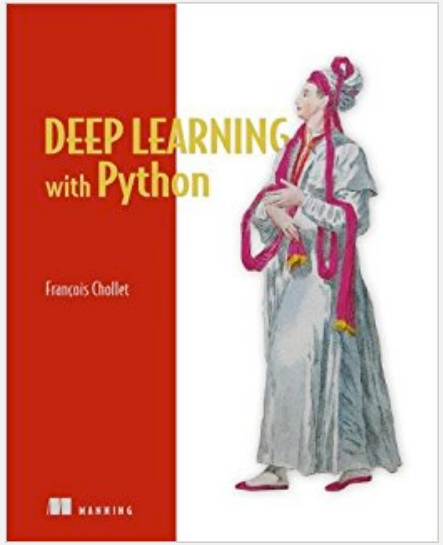
http://www.deeplearningbook.org/

- Good coverage
- more like a big review article

Deep learning with python

- Lots of Keras code examples





Deep learning building blocks

Fully connected networks

Convolutional neural networks

Recurrent neural networks

Gated recurrent units

Long short-term memory networks

Encoder-decoder