ICR The Institute of Cancer Research

Introduction to Machine Learning and Deep Learning

25 April 2018

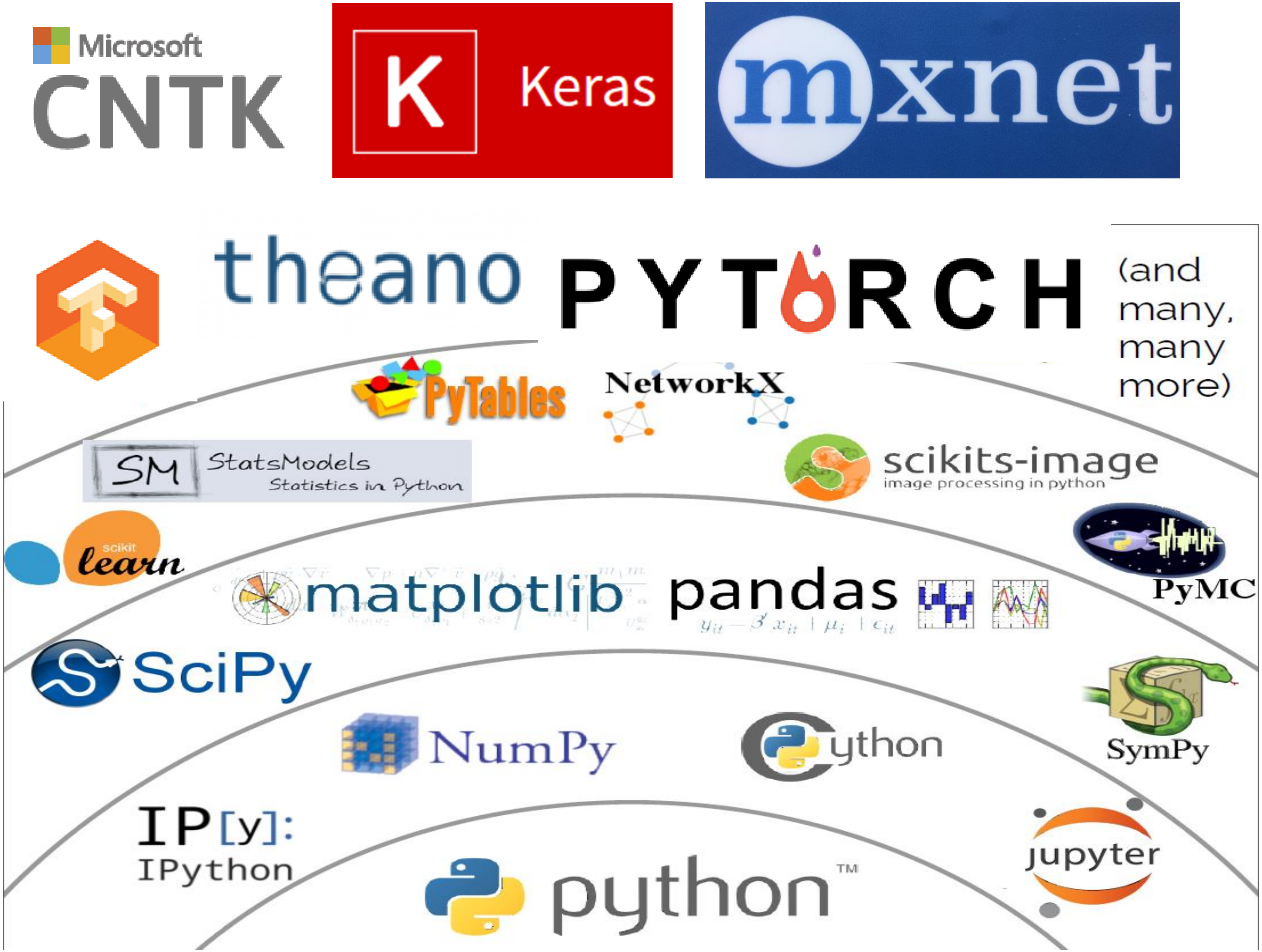
Plan

- 25/04 Introduction
- 02/05 Calculus and Linear Algebra
  Linear models and logistic regression
- 09/05 SVM; k-fold cross-validation and boosting
- 16/05 CNNs; Backprop; Representation Learning; Regularisation; SGD
- 23/05 Image classification using Deep Learning models;
  Keras, Tensorflow and TF-tensorboard

Feedback is welcome

Expect changes and corrections

Please be aware it is not the same room all the times



(and many, many more)

Python

- Numpy, scipy
- Scikit-learn
- Scikit-image, opencv, pillow
- Pandas
- Tensorflow, theano, keras, pytorch, mxnet, cntk
- Matplotlib, bokeh

IDE: Jupyter notebook, Spyder, PyCharm

DL framework requirements

- Need to work with tensors
- Neural Network primitives
- Symbolic derivatives to do efficient backprop
- Offloading to GPUs
- Support for parallelism (multi-device)

Frameworks

- **Caffe, Caffe2 -** Berkeley, FB; model Zoo
- **Theano -** U. Montreal; Python, symbolic derivatives
- **TensorFlow -** Google; Python, symbolic derivatives
- **Pytorch -** FB; Python, Dynamic Neural Networks
- **Apache Mxnet –** Amazon (+ Intel, Baidu and others); Python
- **Microsoft Cognitive Toolkit (CNTK) -** Microsoft
- **Keras -** François Chollet, Python, runs atop of TF, Theano
- **Lasagne -** Python, runs on Theano

Setting yourself up for hands-on experiments

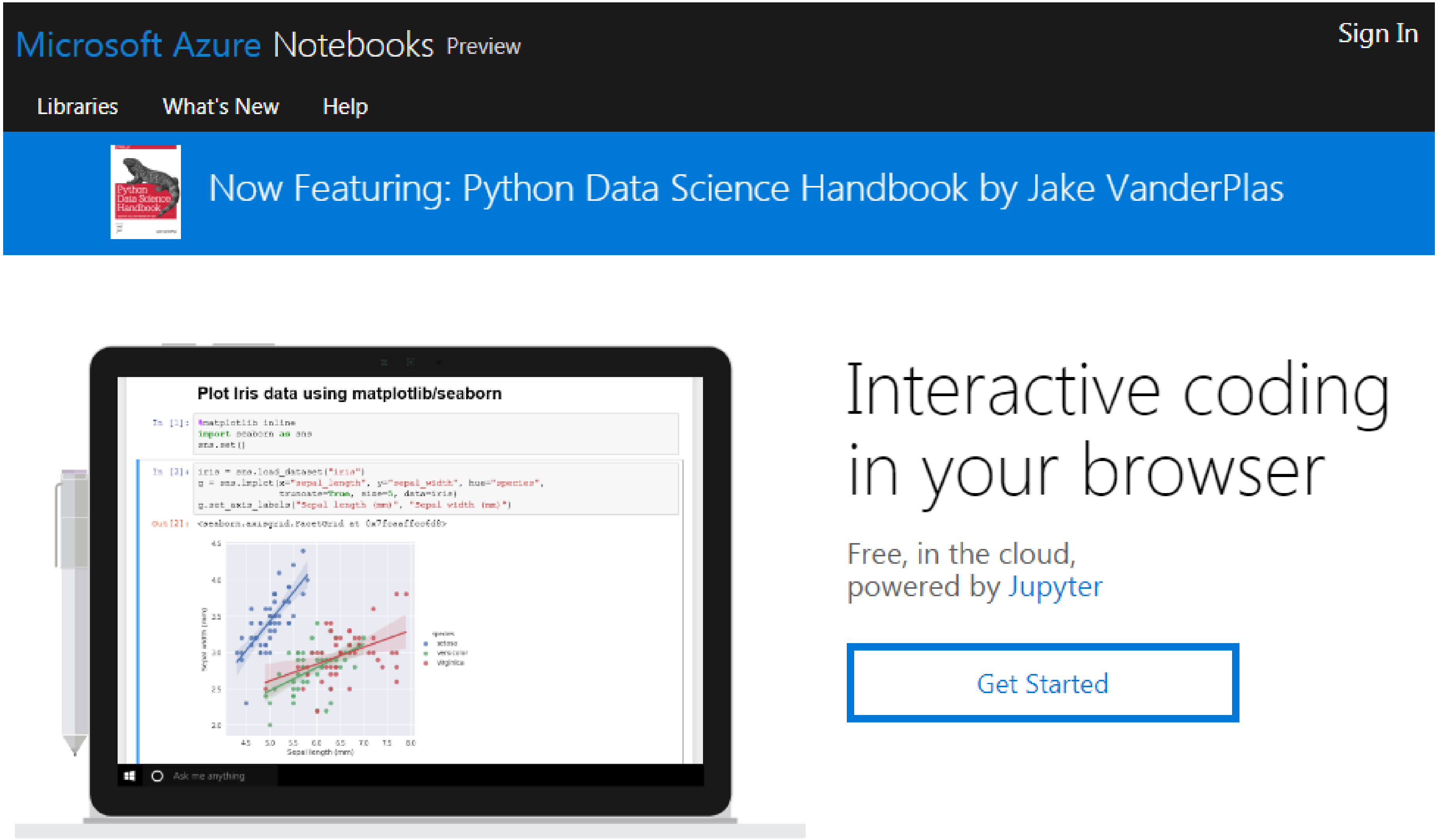
- When using laptop we suggest to install Anacoda or miniconda

  https://conda.io/miniconda.html

  https://www.anaconda.com/download/
- You can also use Davros and setup your environment there
- Further details on setting up conda environment
  https://shadow.icr.ac.uk/Wiki/conda
- One easy option is MS Notebook service on Azure which is free but you only get two vCPUs
- Remember, serious training requires one or more GPUs

Running ipython notebooks on MS Azure:

1) Go to https://notebooks.azure.com/



Microsoft Azure Notebooks Preview

Now Featuring: Python Data Science Handbook by Jake VanderPlas

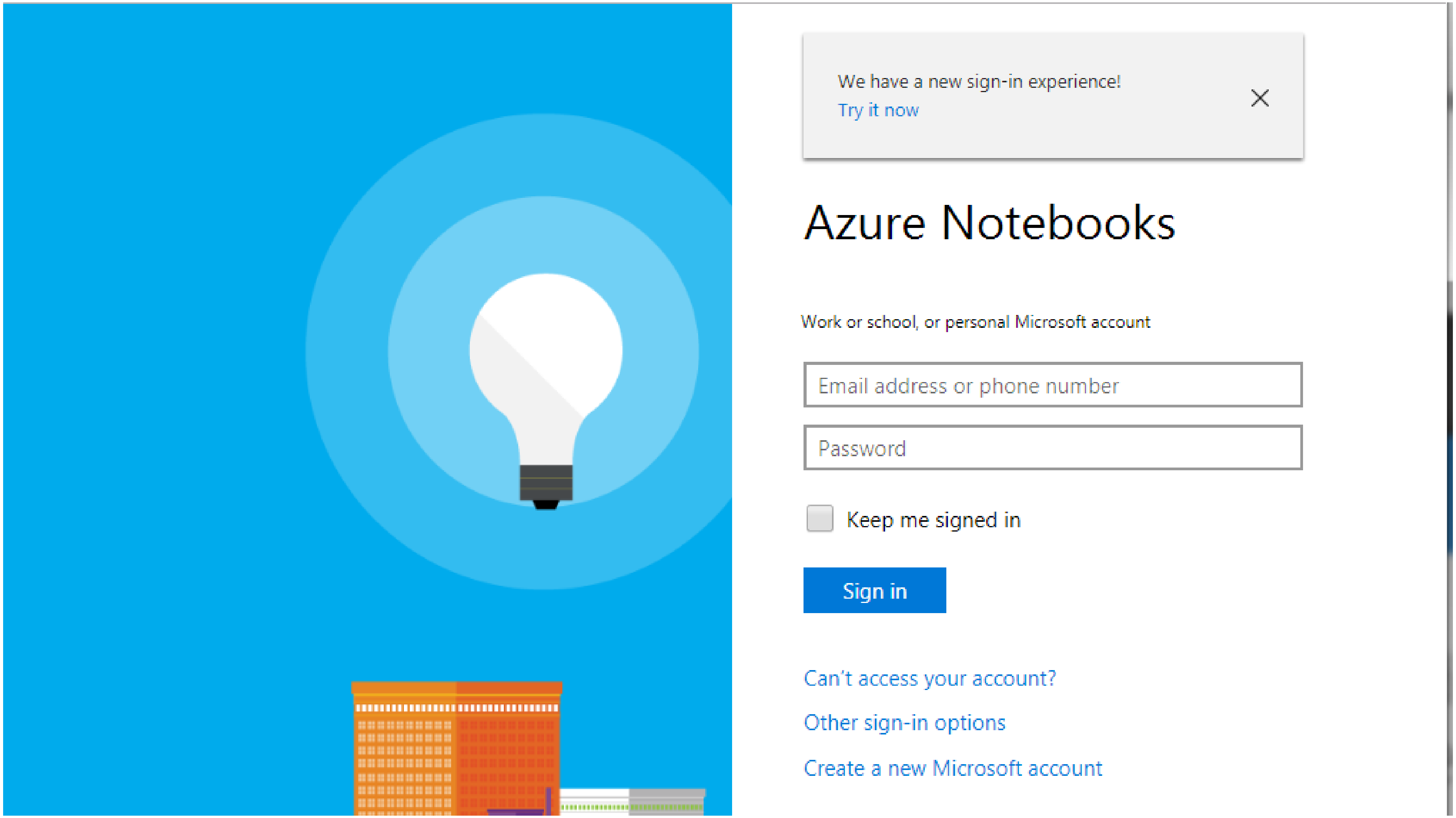
Plot Iris data using matplotlib/seaborn

```
In [1]: %matplotlib inline
        import seaborn as sns
        sns.set()

In [2]: iris = sns.load_dataset("iris")
        g = sns.lmplot(x="sepal_length", y="sepal_width", hue="species",
                       truncate=True, size=5, data=iris)
        g.set_axis_labels("Sepal length (mm)", "Sepal width (mm)")

Out[2]: <seaborn.axisgrid.FacetGrid at 0x7fcaaffcc6d0>
```

Interactive coding in your browser

Free, in the cloud, powered by Jupyter

2) Sign in with any MS account



We have a new sign-in experience!
Try it now

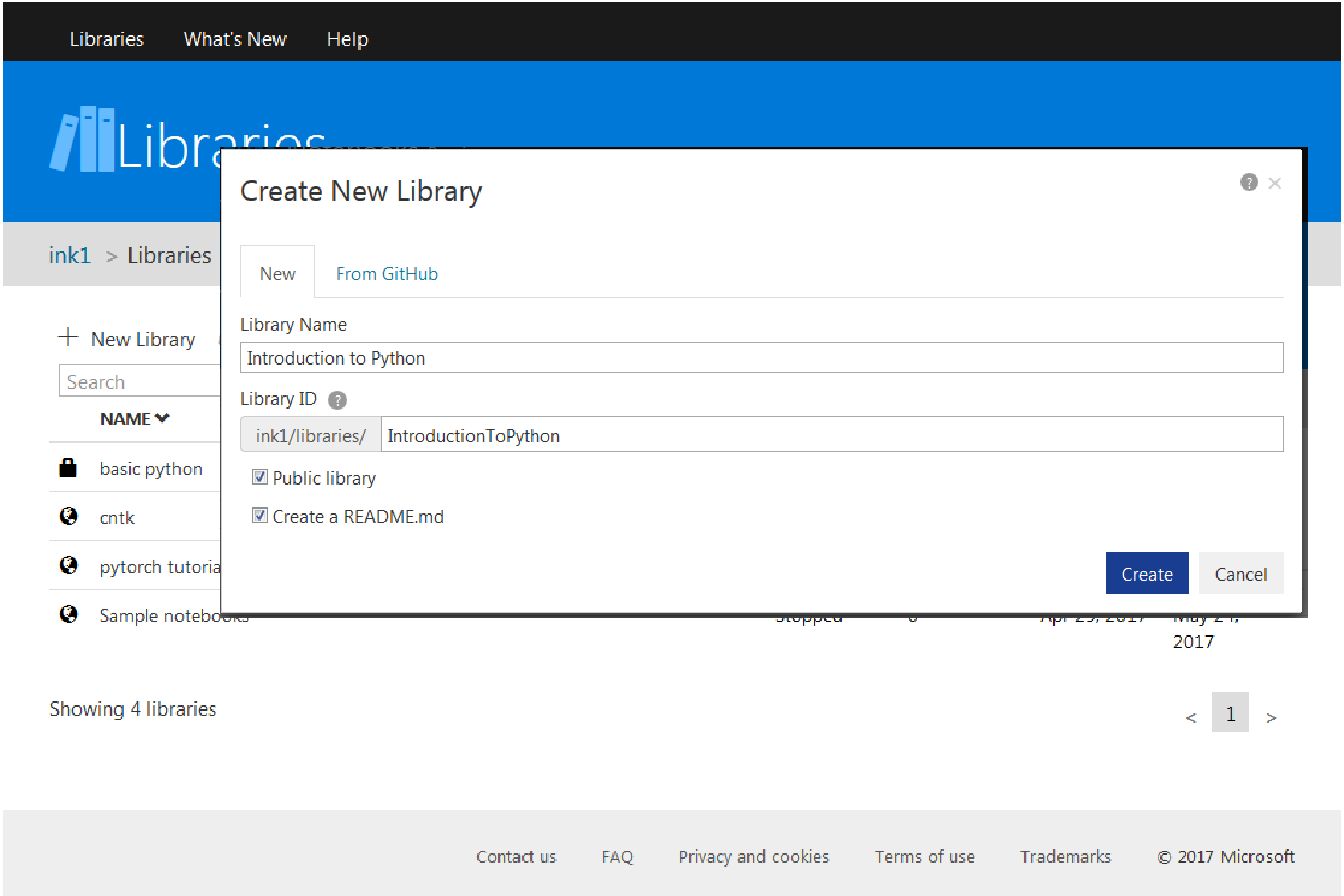
Azure Notebooks

Work or school, or personal Microsoft account

Email address or phone number
Password
□ Keep me signed in
Sign in
Can't access your account?
Other sign-in options
Create a new Microsoft account

3) Add new library "Introduction to Python":

+ New Library



Create New Library

New | From GitHub

Library Name: Introduction to Python

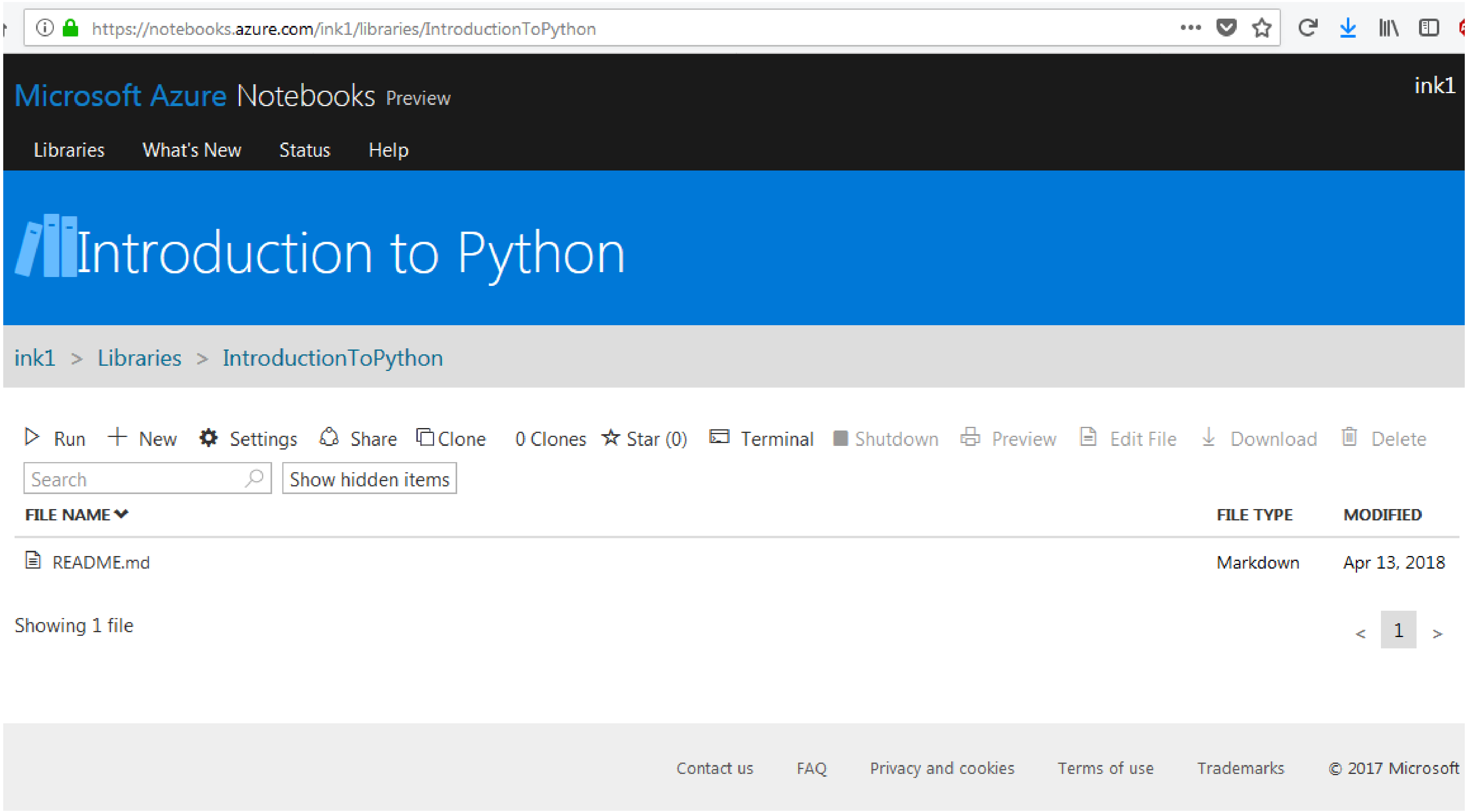
Library ID: ink1/libraries/ IntroductionToPython

☑ Public library

☑ Create a README.md

4) Start Python notebook:

+ New



https://notebooks.azure.com/ink1/libraries/IntroductionToPython

Microsoft Azure Notebooks Preview

ink1

Libraries What's New Status Help

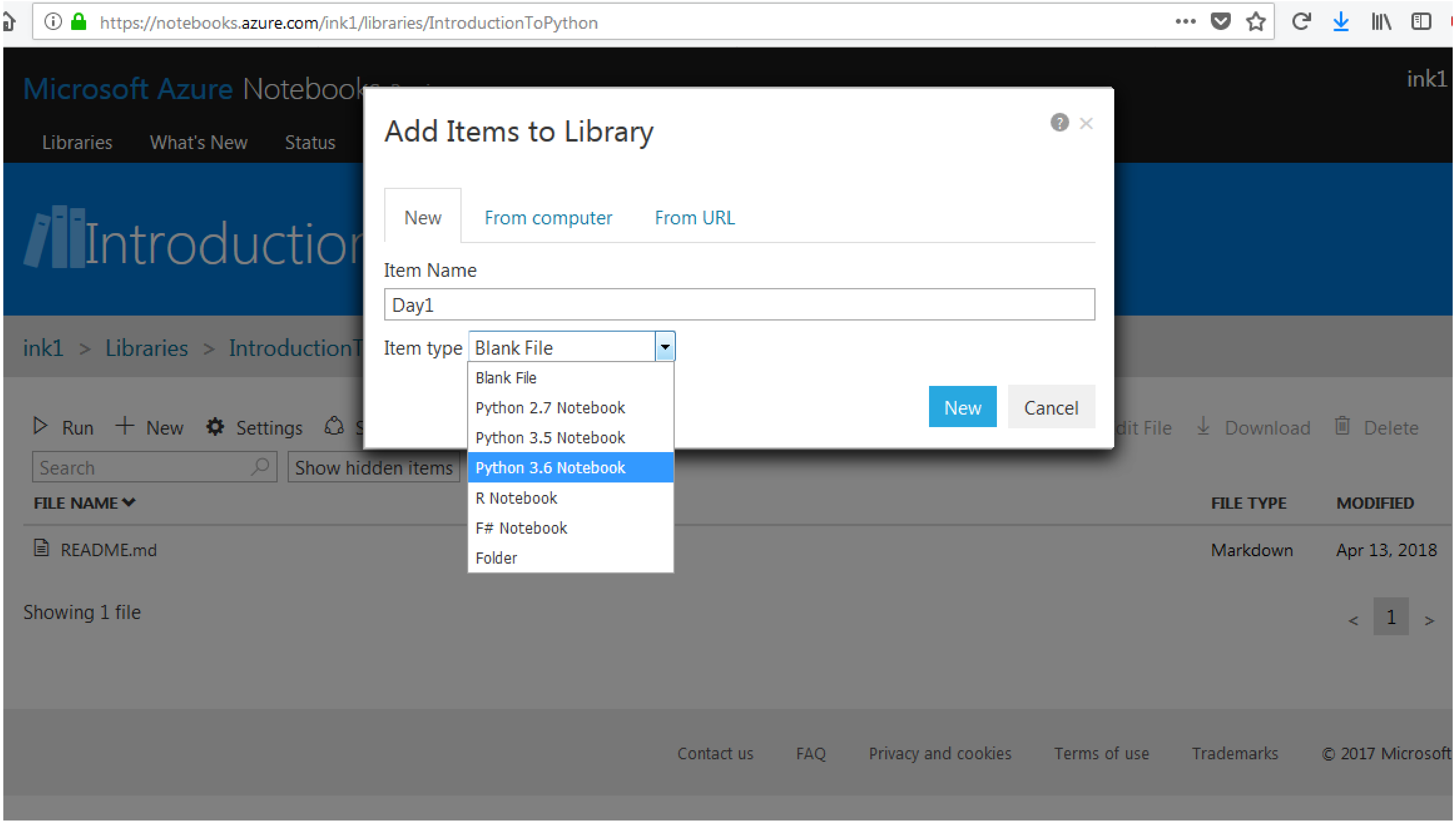
Introduction to Python

ink1 > Libraries > IntroductionToPython

FILE NAME	FILE TYPE	MODIFIED
README.md	Markdown	Apr 13, 2018

Showing 1 file

5) Start Python notebook:
select Python version



https://notebooks.azure.com/ink1/libraries/IntroductionToPython

Add Items to Library

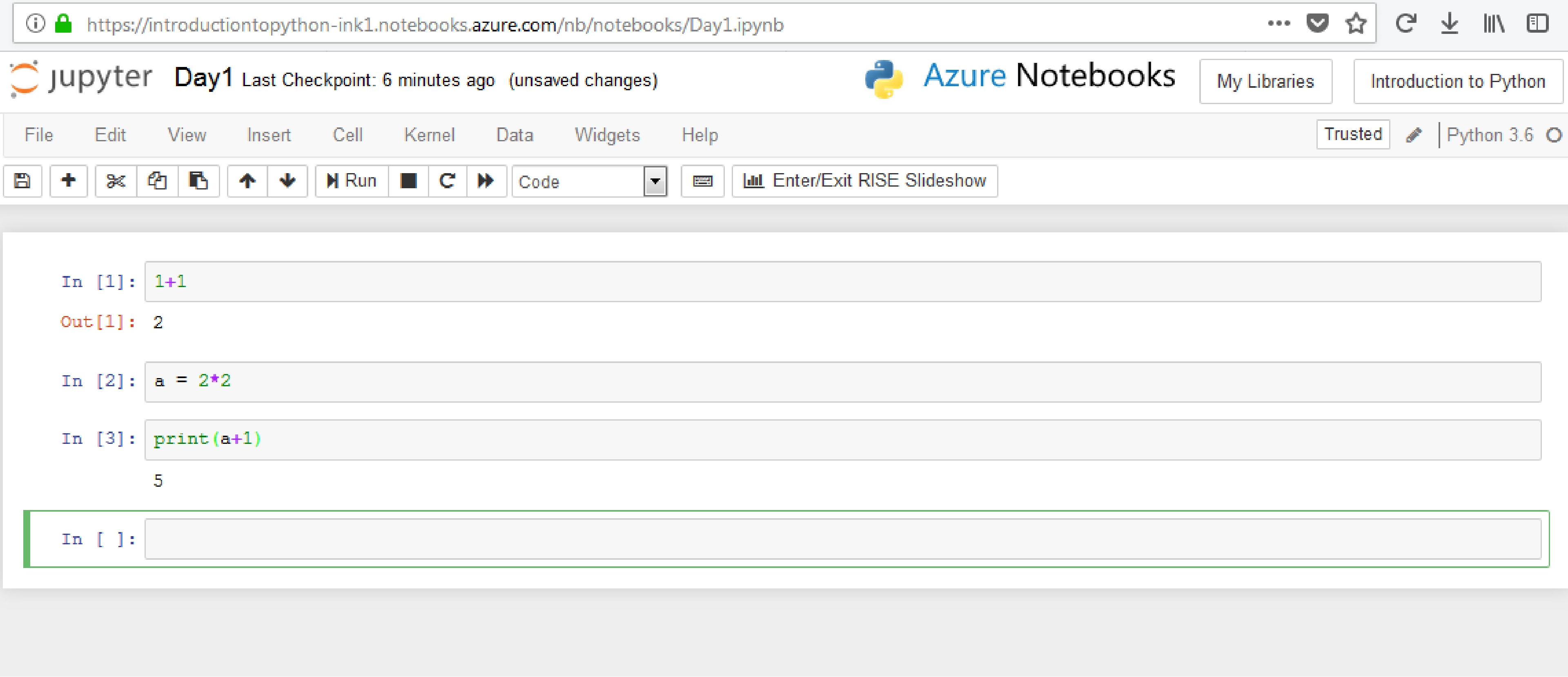
New | From computer | From URL

Item Name: Day1

Item type: Blank File

- Blank File
- Python 2.7 Notebook
- Python 3.5 Notebook
- Python 3.6 Notebook
- R Notebook
- F# Notebook
- Folder

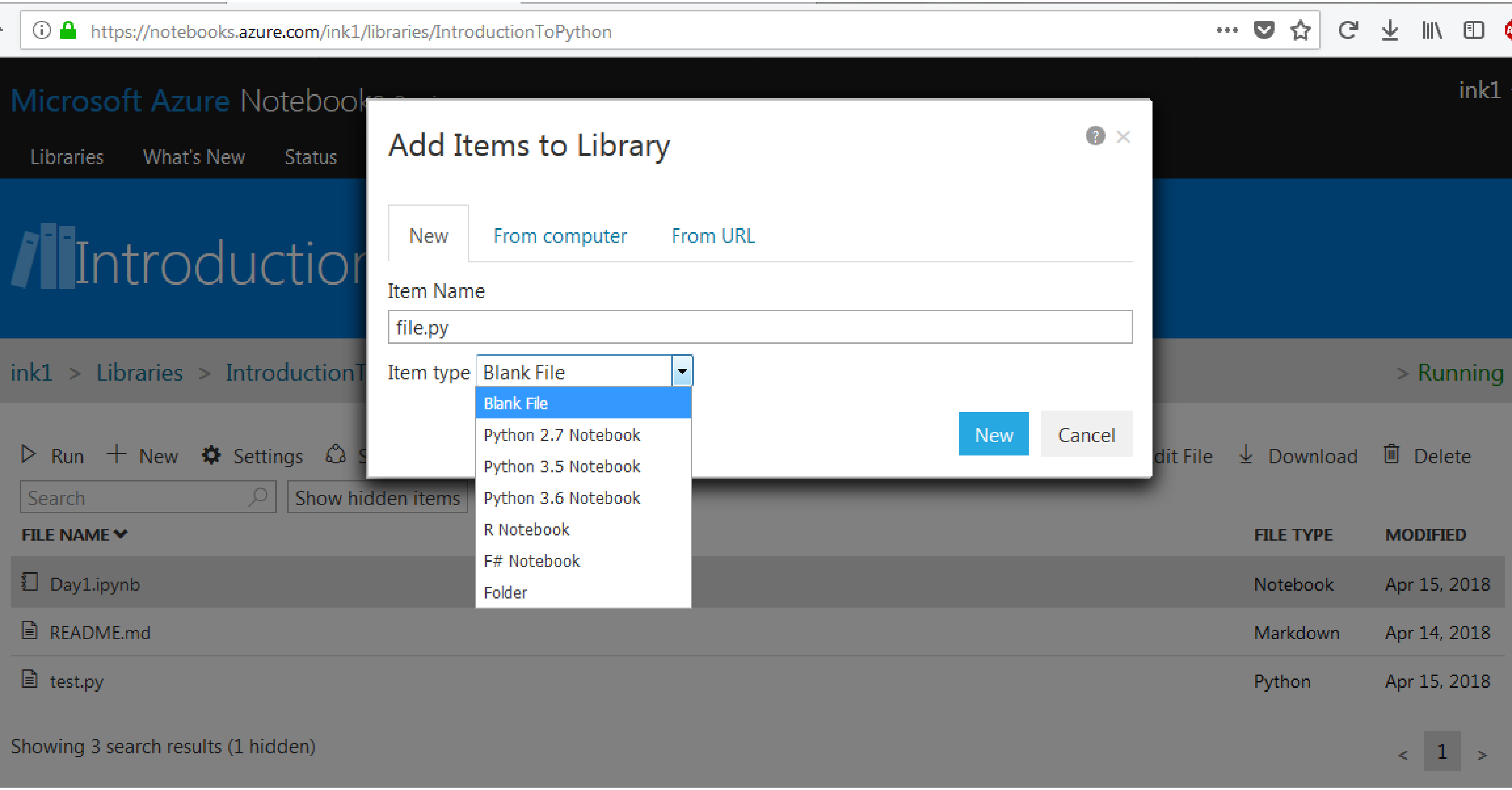
6) start using your notebook



https://introductiontopython-ink1.notebooks.azure.com/nb/notebooks/Day1.ipynb
jupyter Day1 Last Checkpoint: 6 minutes ago (unsaved changes)
Azure Notebooks
My Libraries
Introduction to Python
File Edit View Insert Cell Kernel Data Widgets Help
Trusted
Python 3.6
Run
Code
Enter/Exit RISE Slideshow

```
In [1]: 1+1
Out[1]: 2

In [2]: a = 2*2

In [3]: print(a+1)
        5

In [ ]:
```

7) adding a file to your library



https://notebooks.azure.com/ink1/libraries/IntroductionToPython

Add Items to Library

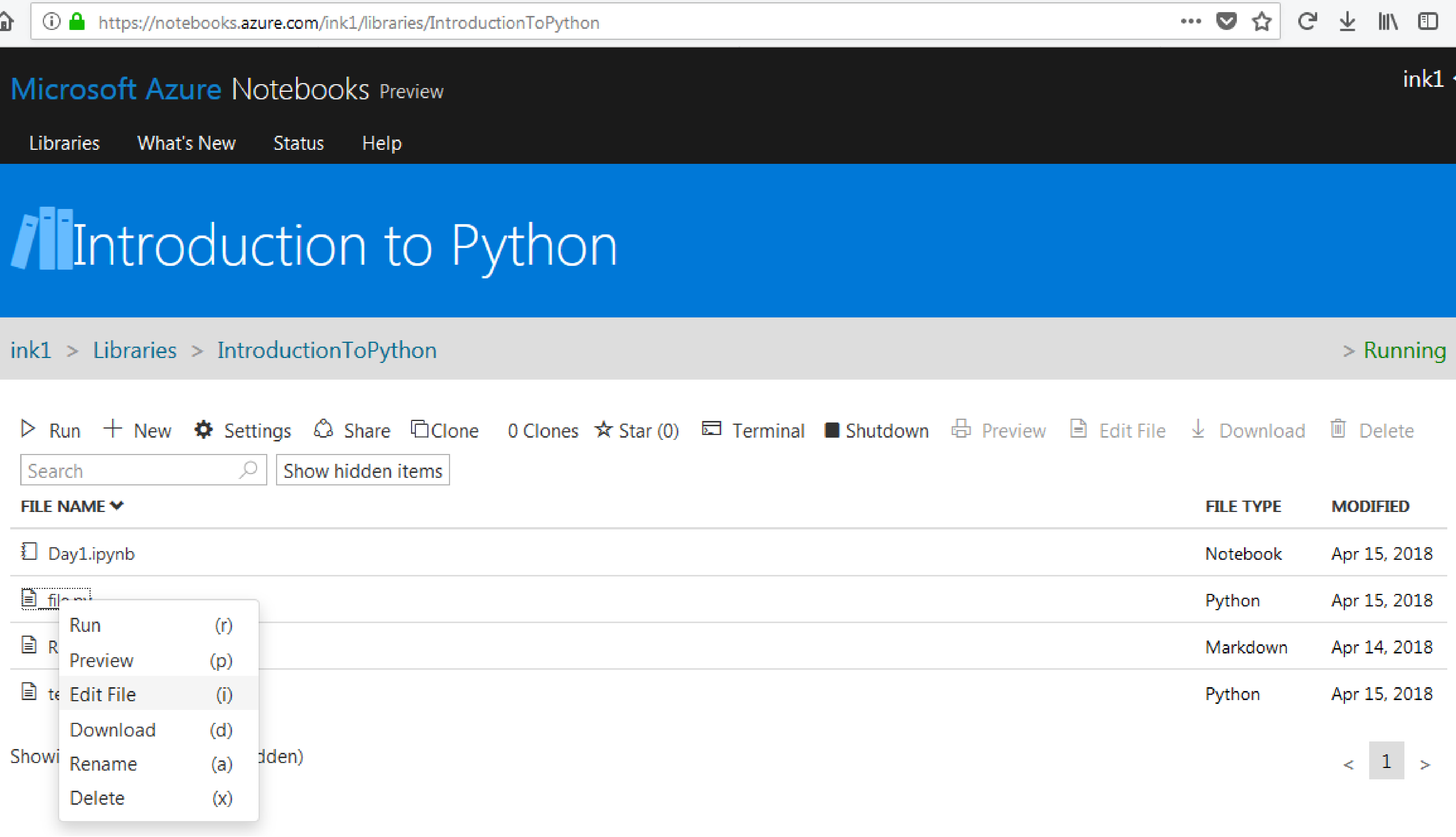
Item Name: file.py

Item type: Blank File

FILE NAME	FILE TYPE	MODIFIED
Day1.ipynb	Notebook	Apr 15, 2018
README.md	Markdown	Apr 14, 2018
test.py	Python	Apr 15, 2018

Showing 3 search results (1 hidden)

8) editing a file - right click - Edit File



https://notebooks.azure.com/ink1/libraries/IntroductionToPython

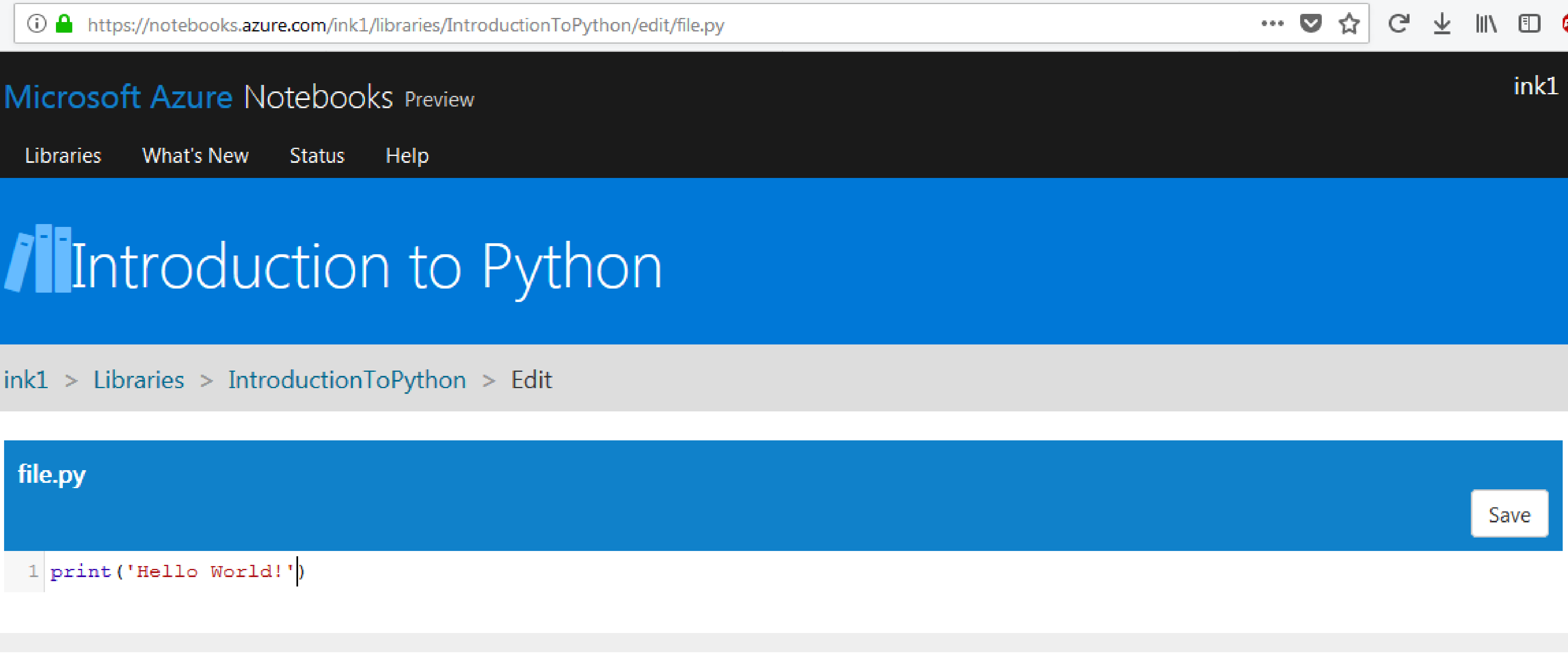
FILE NAME	FILE TYPE	MODIFIED
Day1.ipynb	Notebook	Apr 15, 2018
file.py	Python	Apr 15, 2018
R	Markdown	Apr 14, 2018
te	Python	Apr 15, 2018

Showi dden)

9) add one line

print('Hello World!')

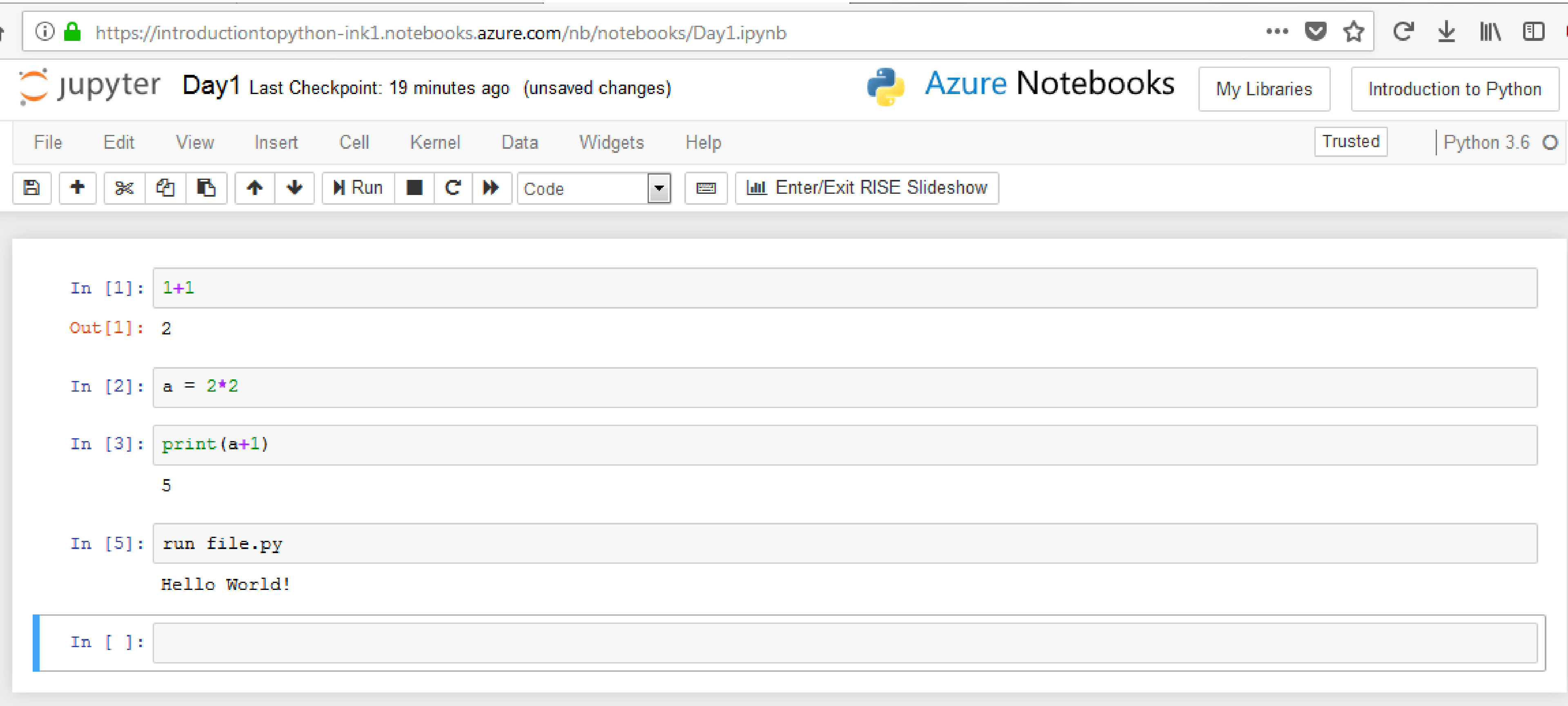
and save the file



https://notebooks.azure.com/ink1/libraries/IntroductionToPython/edit/file.py
Microsoft Azure Notebooks Preview
ink1
Libraries
What's New
Status
Help
Introduction to Python
ink1 > Libraries > IntroductionToPython > Edit
file.py
Save

```
print('Hello World!')
```

10) run the file


https://introductiontopython-ink1.notebooks.azure.com/nb/notebooks/Day1.ipynb
jupyter Day1 Last Checkpoint: 19 minutes ago (unsaved changes)
Azure Notebooks
My Libraries
Introduction to Python
File
Edit
View
Insert
Cell
Kernel
Data
Widgets
Help
Trusted
Python 3.6
Run
Code
Enter/Exit RISE Slideshow

```
In [1]: 1+1
Out[1]: 2

In [2]: a = 2*2

In [3]: print(a+1)
        5

In [5]: run file.py
        Hello World!

In [ ]:
```

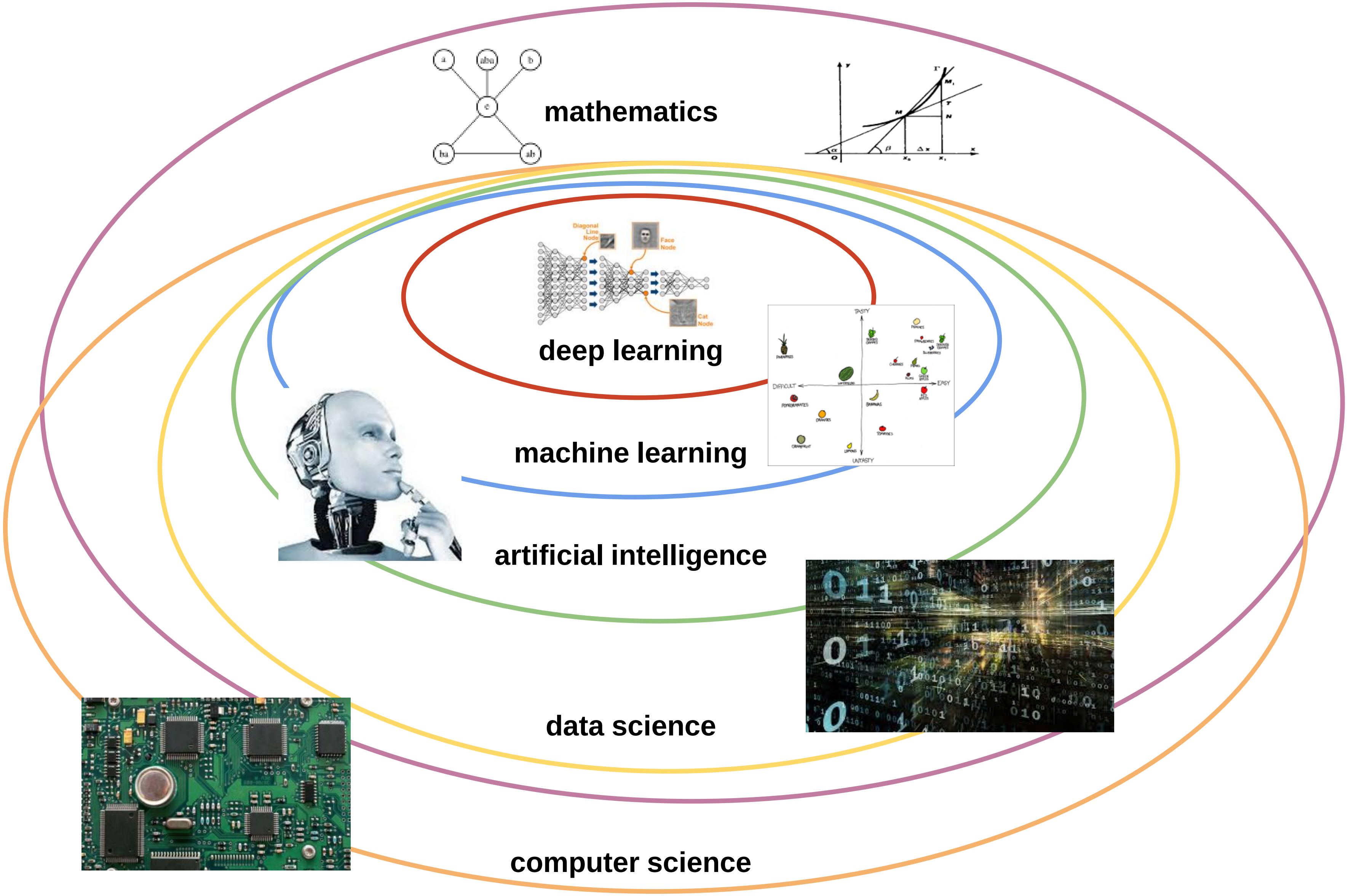
Conda package management

```
# open terminal through Azure notebook
# this gives you access to the VM itself

nbuser@nbserver:~$ . anaconda3_501/etc/profile.d/conda.sh
nbuser@nbserver:~$ conda list
...
nbuser@nbserver:~$ which python
/usr/bin/python
nbuser@nbserver:~$ conda activate base
(base) nbuser@nbserver:~$ which python
/home/nbuser/anaconda3_501/bin/python
(base) nbuser@nbserver:~$ conda install -c damianavila82 rise
```

Conda package management

```
# originally an alternative Python & env manager
# available as Conda or Anaconda (conda + ~200 packages)
# https://www.continuum.io/downloads

~> conda create --name myenv keras
~> conda activate myenv
(myenv) ~> conda list --export > package-list.txt
(myenv) ~> conda deactivate
~> conda info -e       # list environments
```

Deep Learning Universe



mathematics

deep learning

machine learning

artificial intelligence

data science

computer science

Definitions

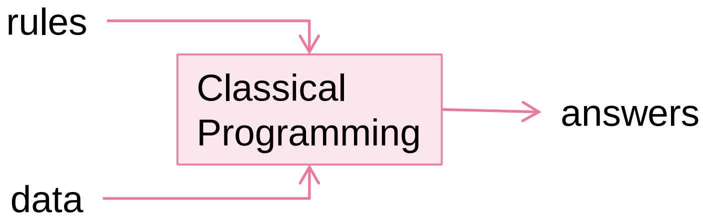
Artificial Intelligence (AI) born in 1950s, the effort to automate intellectual tasks normally performed by humans.

Symbolic AI, expert systems of 1980s

Explicit rules, logical problems such as playing chess

Machine Learning (ML) is learning from data as an alternative to manual coding for specific response, features etc.

Deep Learning (DL) is a branch of ML employing multi-layered Neural Networks (NNs). The word “deep” is used to differentiate from “shallow” NNs employing a few layers only.

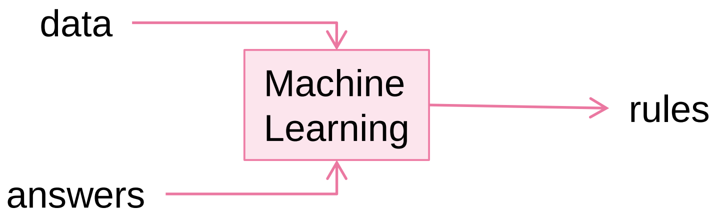


rules

Classical Programming

answers

data



data

Machine Learning

rules

answers

Machine Learning

- Model is trained rather than programmed
- ML finds statistical structure in data that allows the system to come up with rules for automating the task
- ML is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions

Tasks solvable by ML

- Classification $\mathcal{F}\colon \mathbb{R}^n \longrightarrow \{1, \dots k\}$
  - Classification with missing input
- Regression $\mathcal{F}\colon \mathbb{R}^n \longrightarrow \mathbb{R}$
- Structured output
  - Transcription
  - Translation
  - Image segmentation
- Anomaly detection
- Imputation of missing values $\mathcal{F}\colon \mathbb{R}^n(with\,missing\,values) \longrightarrow \mathbb{R}^n$
- Denoising $\mathcal{F}\colon \mathbb{R}^n \longrightarrow \mathbb{R}^n$

Types of ML Algorithms

Supervised Learning

Learn from training data with labels

Unsupervised Learning

Learn from unlabelled data
(categorize unlabelled data based on discovered similarities)

Reinforcement Learning

Learn from interactions and rewards from the world
(train desired behaviour within a specific context)

Supervised learning in general

- Input data $X = \{X_1, \dots, X_N\}, X_i \in \mathbb{R}^n$
- Output data $Y = \{Y_1, \dots, Y_N\}, Y_i \in \mathbb{R}^m$
- Find mapping $\mathcal{F}(X_i; \omega) = Y_i{}'$
- such that the loss function $\mathcal{G}(\omega) = \Sigma_i G(Y_i, Y_i{}'; \omega)$
- is minimal over the parameter space $\omega \in \mathbb{R}^k$

How to select the right $\mathcal{F}(X;\omega)$?

- Model selection (functional space) – restrictions on the class of functions
- Parameter optimisation – find the best point $\omega \in \ \mathbb{R}^k$
- Criteria for being the best are far from obvious

Machine Learning

$Y = \{black, red\}, X \in \mathbb{R}^2$

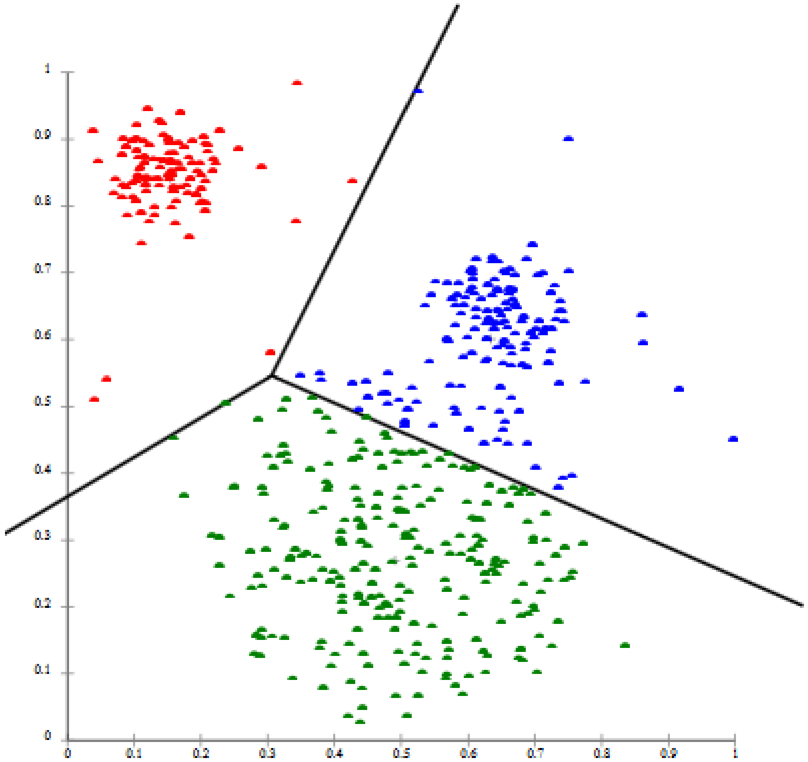
Linear transformation in 2D plane allows easy selection of red dots by choosing $x_1' > 0$.

What does the learning do?

Given input data and expected output, compute a representation which can be used to predict the output.

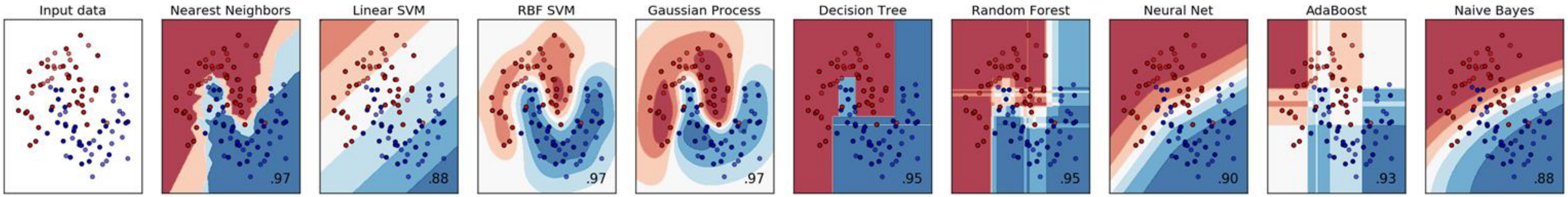
An example of a representation can be HSV (Hue Saturation Value) which is another (exact) representation of RGB format.

The power of a representation is related to model capacity and will in general incur some data loss.



What does the learning do?

- All ML algorithms consist of automatically finding such series of transformations that turn input data into more useful representation for a given task. These can be linear transformations, projections, non-linear operations and so on but the hypothesis space is fixed.
- If such a transformation cannot be found immediately the process can proceed iteratively using a feedback from minimisation of a loss function which characterises how close (or far) the current output is from the desired output.



Consider 1D regression

Statistics refresh

Discrete random variable X takes values from a set, e.g. coin flip {0,1}.

Probability of an event $P(A)$.

If $X \in \{0, 1\}$ then $P(X = 0) + P(X = 1) = 1$.

Union of events

$$P(A \vee B) = P(A) + P(B)$$

Joint probability

$$P(A, B) = P(A \wedge B) = P(A|B)P(B)$$

Conditional probability (provided that $P(B) > 0$)

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

Bayes rule

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

where $P(B)$ may be computed as $P(B) = \Sigma_a P(B|a)P(a)$

Mean value of probability distribution over all possible events

$$\mathbb{E}[X] = \Sigma_a \, a \, P(a)$$

Statistics refresh

Continuous random variable X can take values from $\mathbb{R}$.

Probability density function $p(x)$.

Probability is $P(a < X < b) = \int_a^b p(x)\, dx$.

Mean, or expected value μ

$$\mathbb{E}[X] = \int x\, p(x) dx$$

Variance, measure of "spread" of a distribution, σ^2

$$var[X] = \mathbb{E}[(X - \mu)^2] = \int (x - \mu)^2 p(x) dx = \mathbb{E}[X^2] - \mu^2$$

Standard deviation

$$std[X] = \sqrt{var[X]}$$

Normal distribution

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

Statistics refresh

Self-information of an event $X = x$

$$I(X = x) = -logP(X = x)$$

Shannon entropy

$$H(X) = -\Sigma_{k=1}^{K} P(X = k)\ log\ P(X = k) = \mathbb{E}_{X\sim P}[I(X)]$$

Cross entropy of two probability distributions P and Q over the same random variable X

$$H(P, Q) = -\mathbb{E}_{X\sim P}[logQ] = -\Sigma_{k=1}^{K} P\ log\ Q$$

Kullback-Leibler (KL) divergence or relative entropy is a measure of dissimilarity of two probability distributions P and Q over the same random variable X

$$\mathrm{KL}(P||Q) = -H(P) + H(P, Q)$$

Note that generally $\mathrm{KL}(P||Q)\ \neq\ \mathrm{KL}(Q||P)$

$\mathrm{KL}(P||Q) = 0$ if and only if $P = Q$

Frequentist Statistics

- Randomness is objective
- Probability is the frequency of past events; in this way it’s objective and doesn’t depend on one’s beliefs
- Data X is random but model parameters ω are fixed (but unknown)
- **Bias** of ω' is the difference between the expectation value of ω' and the true value of ω; the expectation is over the data X seen as samples from a random variable
- **Variance** of ω' is a measure of how the estimate of ω would change if we resample the dataset from the underlying data generating process
- Ideally we want both bias and variance of our estimate be small and $\omega' \rightarrow \omega$ as $N \rightarrow \infty$
- Find a point estimate ω' using maximum likelihood: $\omega' = \arg\max_{\omega} P(X|\omega) = \arg\max_{\omega} \log P(X\,|\omega)$

Bayesian Statistics

- Randomness is subjective
- Probability is a measure of our belief so that probability is subjective and refers to the future
- Data X is fixed (given) but model parameters ω are random; search for probability distribution which generates (random) parameters
- Compute posterior

$$P(\omega|X) = \frac{\overset{\text{likelihood}}{P(X|\omega)}\,\overset{\text{prior}}{P(\omega)}}{\underset{\text{evidence}}{P(X)}}$$

- However most interesting models are intractable, hence point estimates
- Maximum A Posteriori (MAP) point estimate
  $\omega_{MAP} = \arg\max_{\omega} P(\omega|X) = \arg\max_{\omega} \log P(X|\omega) + \log P(\omega)$

Coin tossing example (after Geoffrey Hinton)

- Suppose we only know each tossing event produces a head with some unknown probability P and a tail with probability $1 - P$
- This means that our model has one parameter - P
- Suppose we observed 100 tosses and 53 of them gave heads
- What is P?

Coin tossing example Frequentist answer (aka maximum likelihood)

Pick the value of p (probability of head) that makes the observation A of 53 heads and 47 tails most probable.

$$P(A) = p^{53}\,(1-p)^{47}$$

Differentiate over p

$$\frac{dP}{dp} = 53p^{52}(1-p)^{47} - 47p^{53}(1-p)^{46} = \left(\frac{53}{p} - \frac{47}{1-p}\right)\,[p^{53}(1-p)^{47}] = 0$$

$$\frac{53}{p} = \frac{47}{1-p}$$

$$53 - 53p = 47p$$

$$p = 0.53$$

Coin tossing example
Bayesian answer

- Start with a prior distribution over p:
$$P(p) = 1$$
- Step 1a: we get a head after the first toss, multiply by the probability of observing a head given that value
- Step 1b: rescale probability density to 1

- Step 2a: we get a tail after the second toss, multiply by the probability of observing a tail given that value
- Step 2b: rescale probability density to 1

Coin tossing example
Bayesian answer

- Start with a *prior* distribution over p:

$$P(p) = 1$$

- Step 1a: we get a head after the first toss, multiply by the probability of observing a head given that value
- Step 1b: rescale probability density to 1

- Step 2a: we get a tail after the second toss, multiply by the probability of observing a tail given that value
- Step 2b: rescale probability density to 1
- …
- After 100 steps we get a *posterior* probability distribution

ML models

- Probabilistic modelling such as Naïve Bayes algorithm which ML classifier based on applying Bayes’ theorem while assuming that the features in the input are all independent
- Kernel methods such support vector machines (SVM)
- Decision trees, random forests an gradient boosting machines
- K-nearest neighbours

Parametric vs Non-parametric models

Parametric model has a fixed number of parameters

E.g., linear regression

Non-parametric model has a number of parameters which grows with the amount of training data.

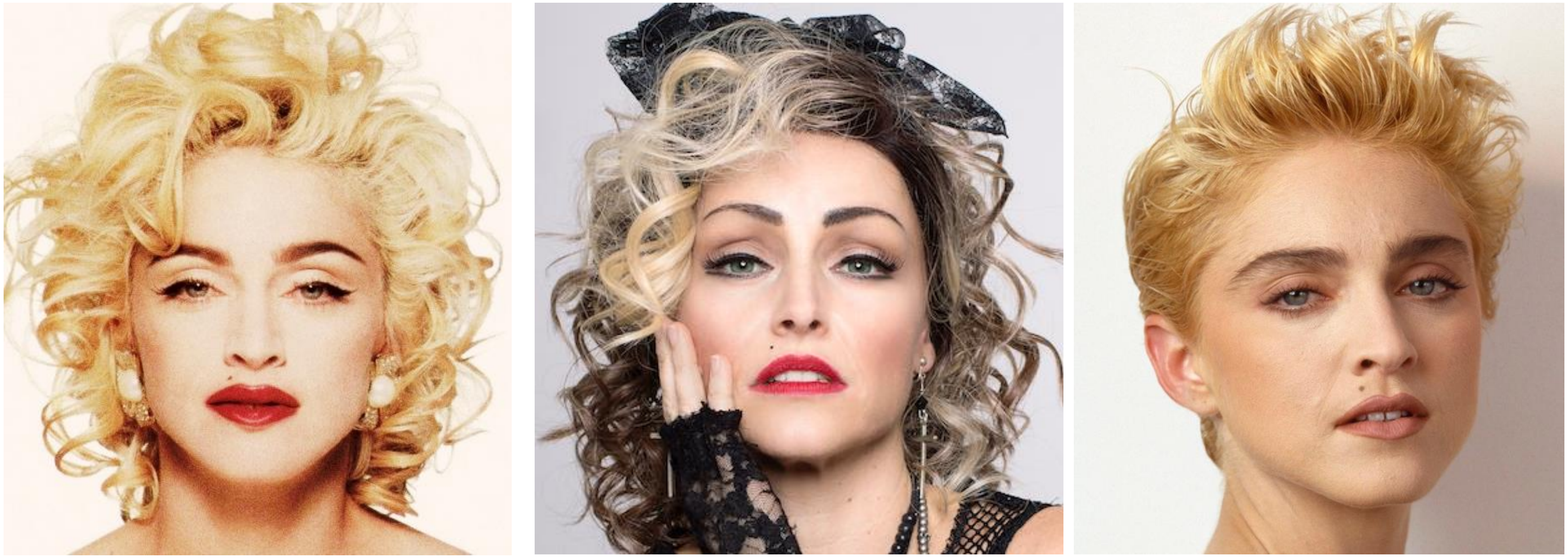
E.g. K-nearest neighbour (KNN)

Deep Learning

- DL is strongly related to mathematical statistics
- but the statistical apparatus has to be much simplified when faced with computational complexity and the amount of data (e.g. millions of images)
- Therefore DL exhibits relatively little mathematical theory and is engineering oriented
- Often ideas are proposed based on (sometimes limited) empirical evidence
- Nevertheless there has been incredibly good progress in the last couple of years

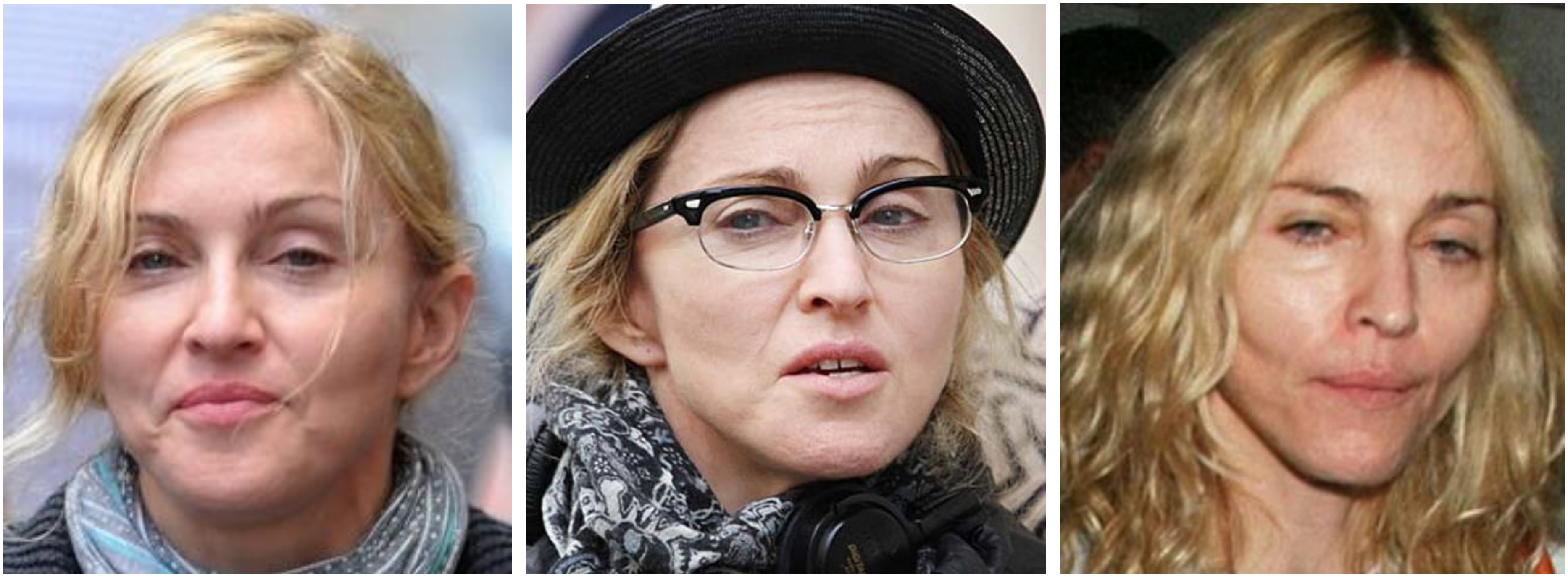
Learning process

Here is Madonna.
Assume you didn’t know who she is



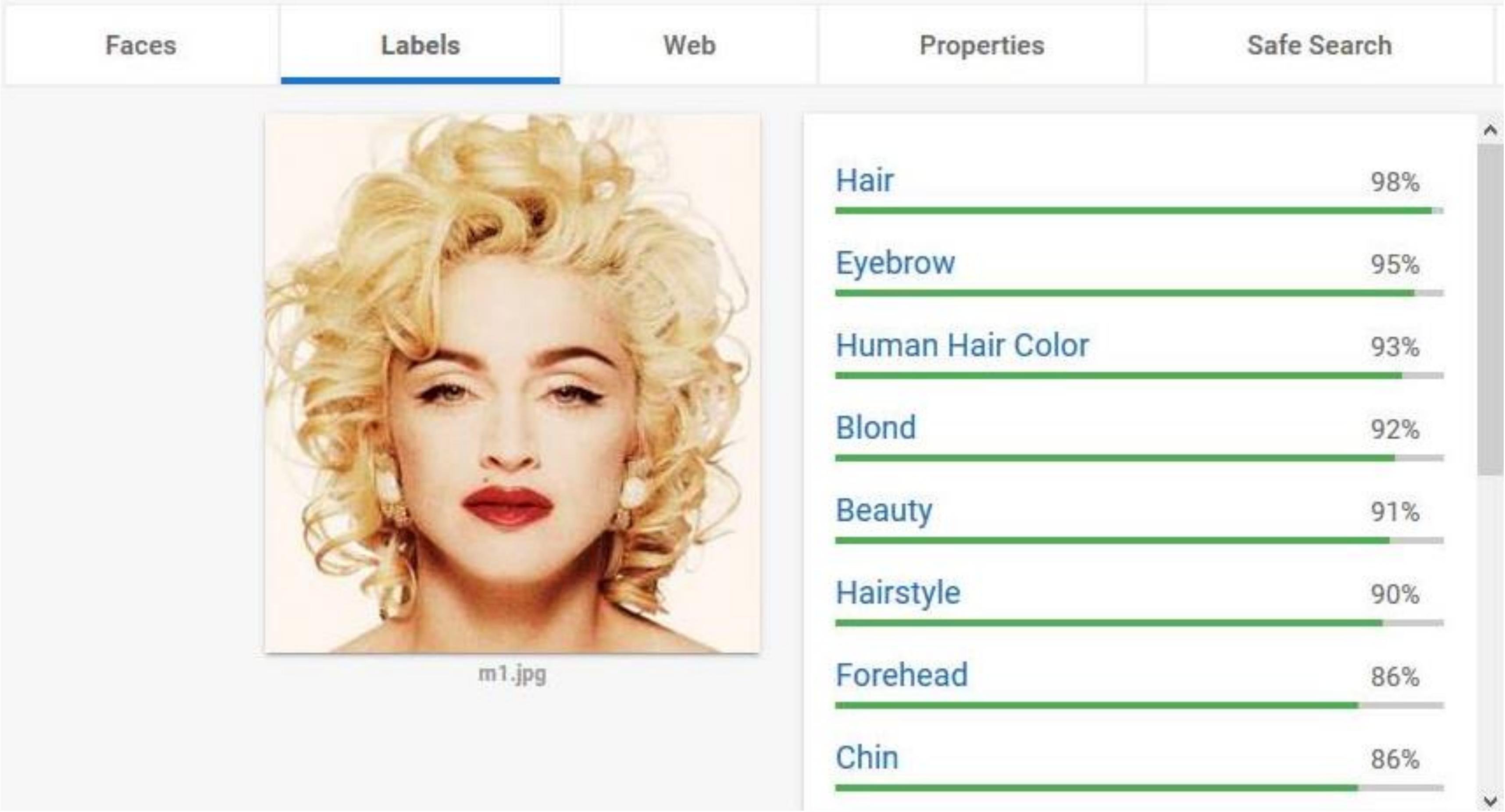
This is your training set.

Now, do you recognise these people?



This is your test set.

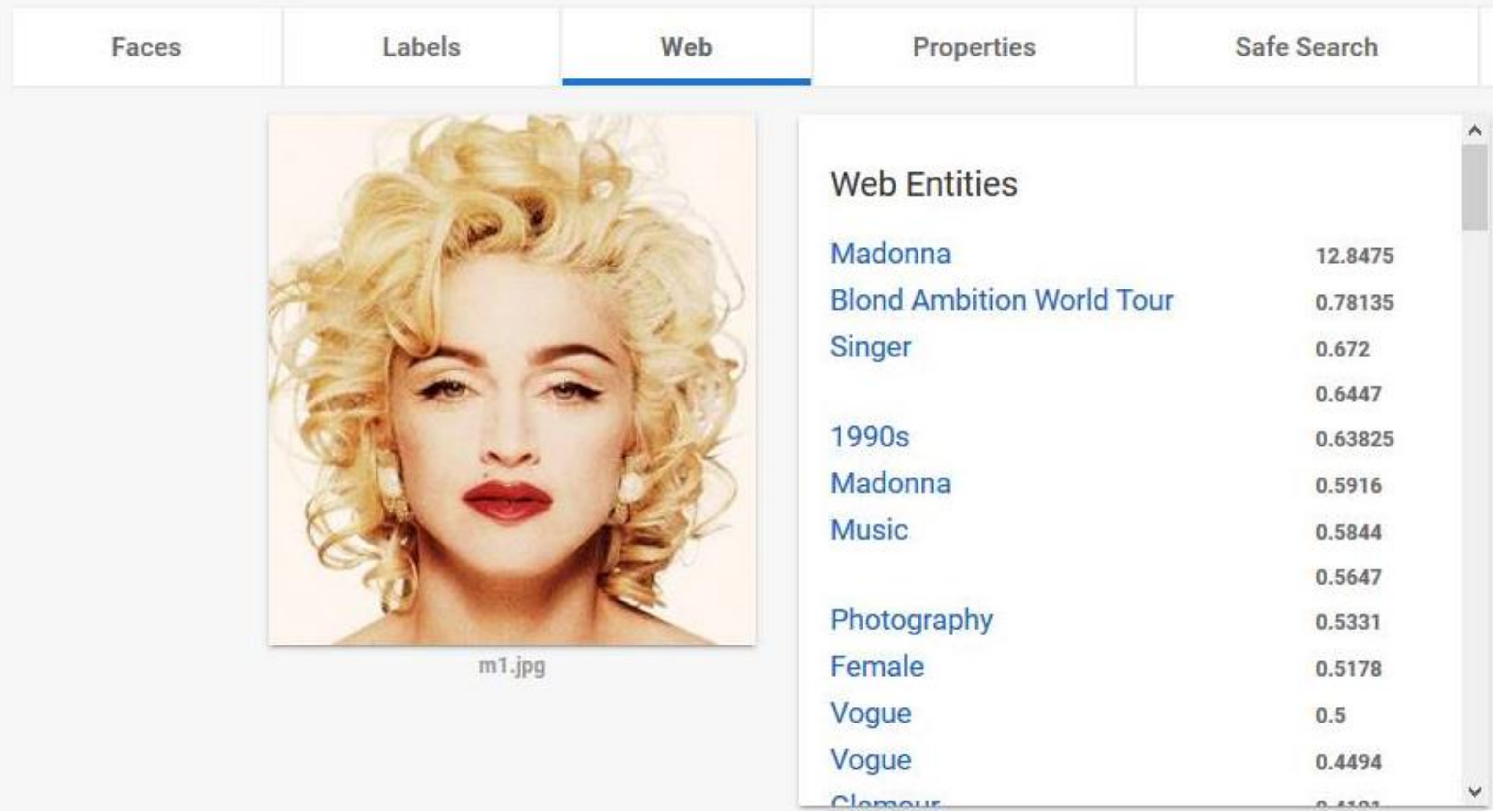
What about ML? Can a machine do it? https://cloud.google.com/vision/



Hair	98%
Eyebrow	95%
Human Hair Color	93%
Blond	92%
Beauty	91%
Hairstyle	90%
Forehead	86%
Chin	86%

Not quite what we wanted to see

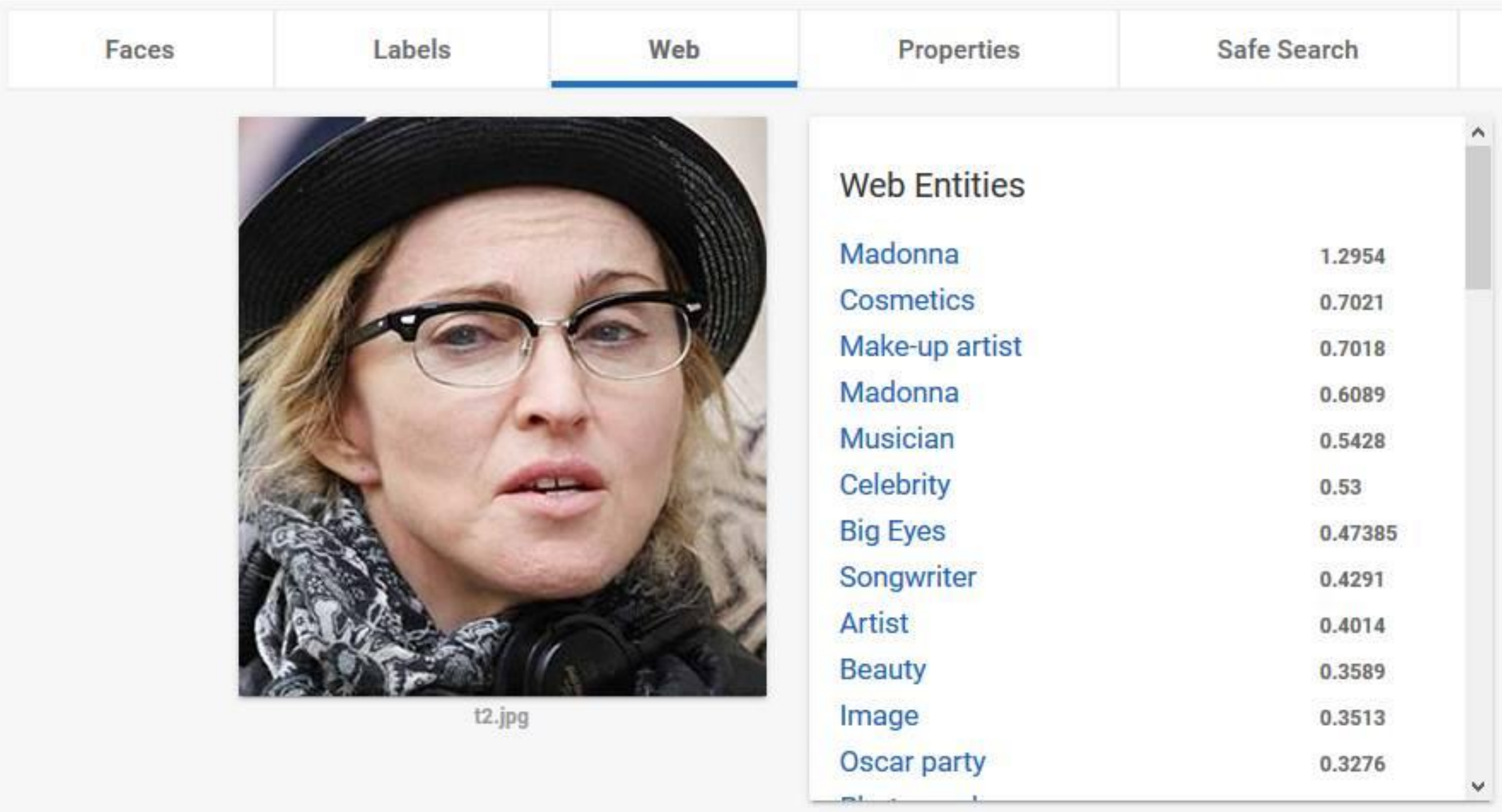
Can a machine recognise image? https://cloud.google.com/vision/



Web Entities

Madonna	12.8475
Blond Ambition World Tour	0.78135
Singer	0.672
	0.6447
1990s	0.63825
Madonna	0.5916
Music	0.5844
	0.5647
Photography	0.5331
Female	0.5178
Vogue	0.5
Vogue	0.4494
Glamour	0.4191

Can a machine recognise image? https://cloud.google.com/vision/

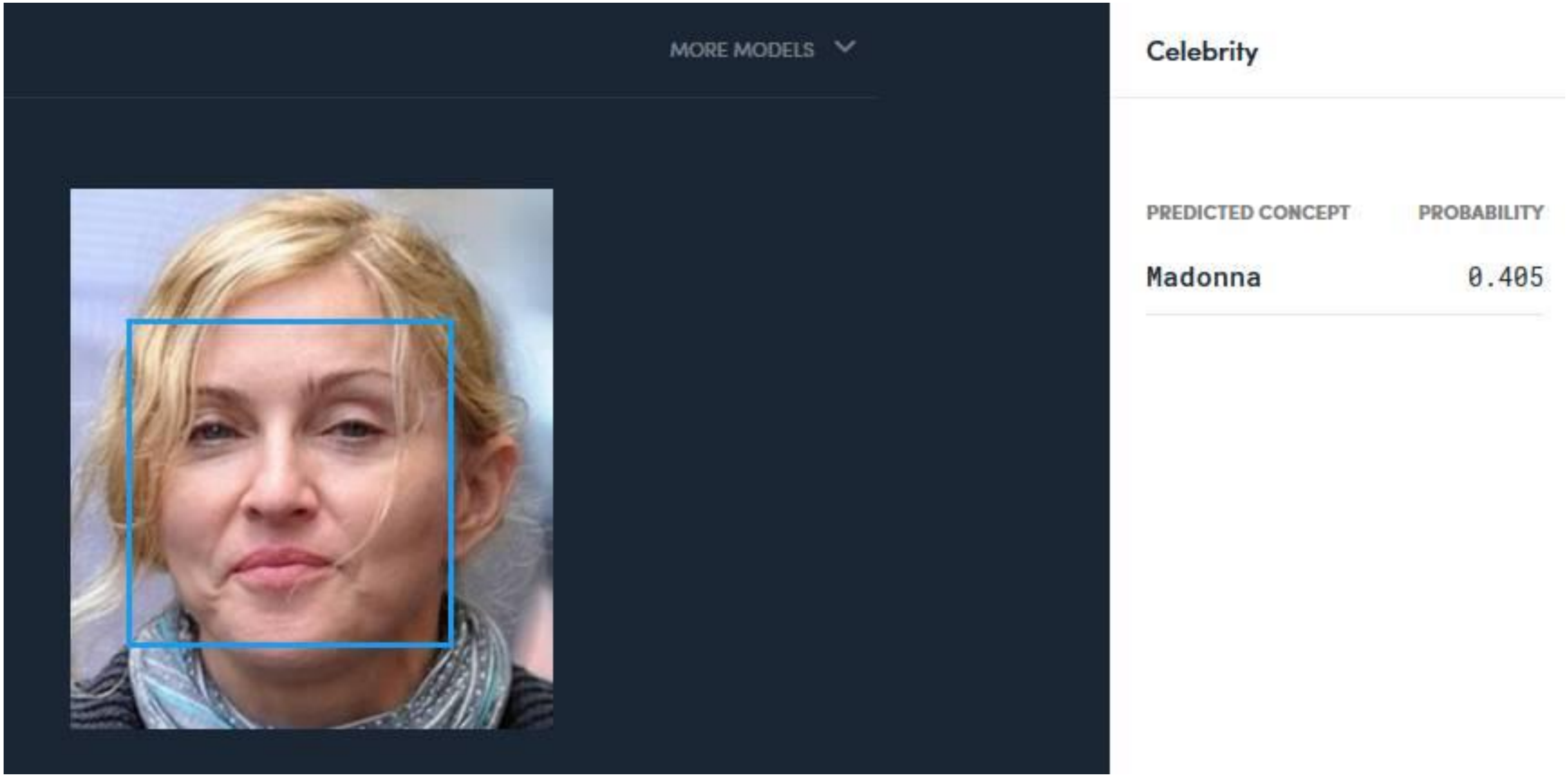


Web Entities

Madonna	1.2954
Cosmetics	0.7021
Make-up artist	0.7018
Madonna	0.6089
Musician	0.5428
Celebrity	0.53
Big Eyes	0.47385
Songwriter	0.4291
Artist	0.4014
Beauty	0.3589
Image	0.3513
Oscar party	0.3276

If we use the first web entity as a predictor then Cloud Vision API identified all the test images correctly. However this should not come as surprise because the images were collected through Google search in the first place.

Can a machine recognise image? https://clarifai.com/demo

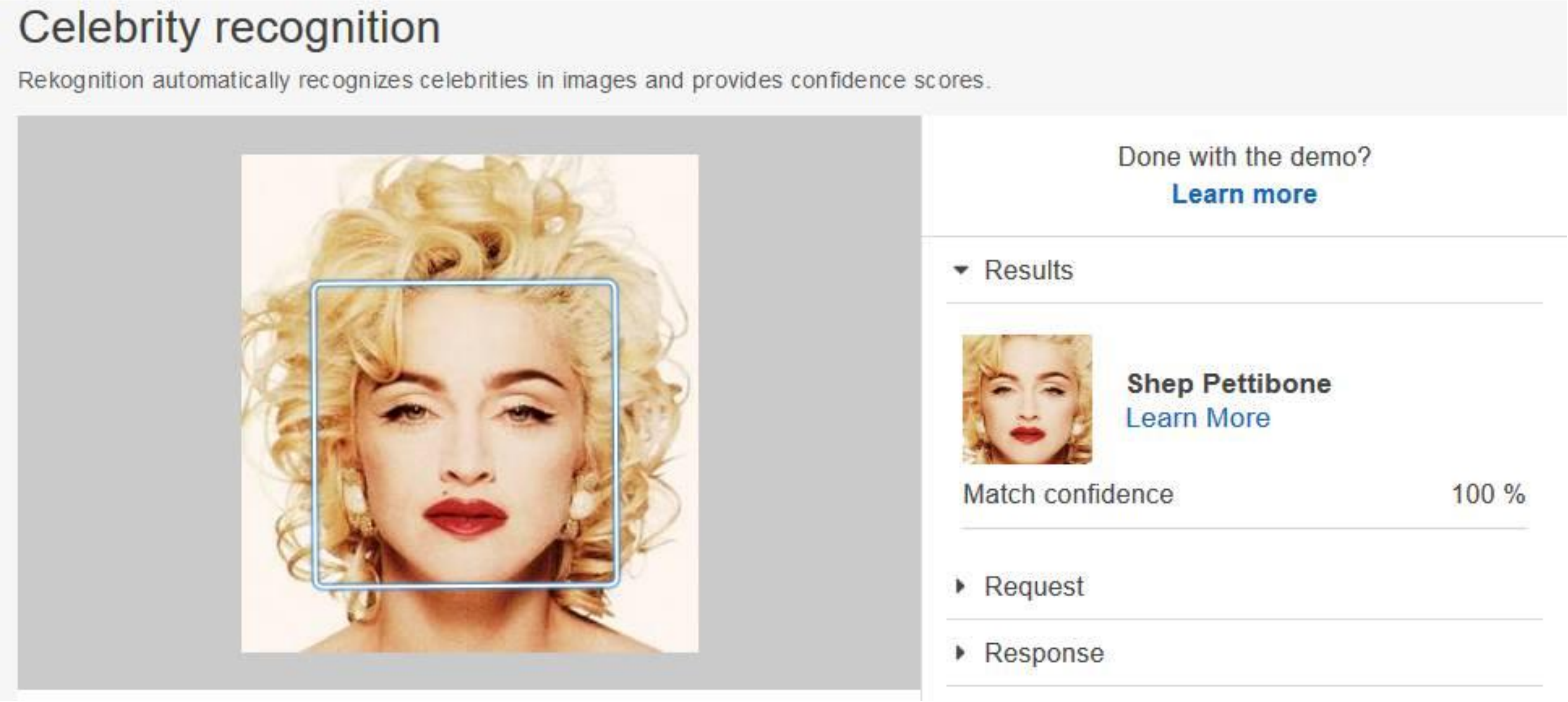


Celebrity

PREDICTED CONCEPT	PROBABILITY
Madonna	0.405

Clarifai was only able to guess the first test image with P = 0.4. It did not recognise the second “training” image (P(m1) = 0.989 and P(m3) = 0.543).

Can a machine recognise image? Amazon Rekognition



Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores.

Done with the demo?
Learn more

▾ Results

Shep Pettibone
Learn More

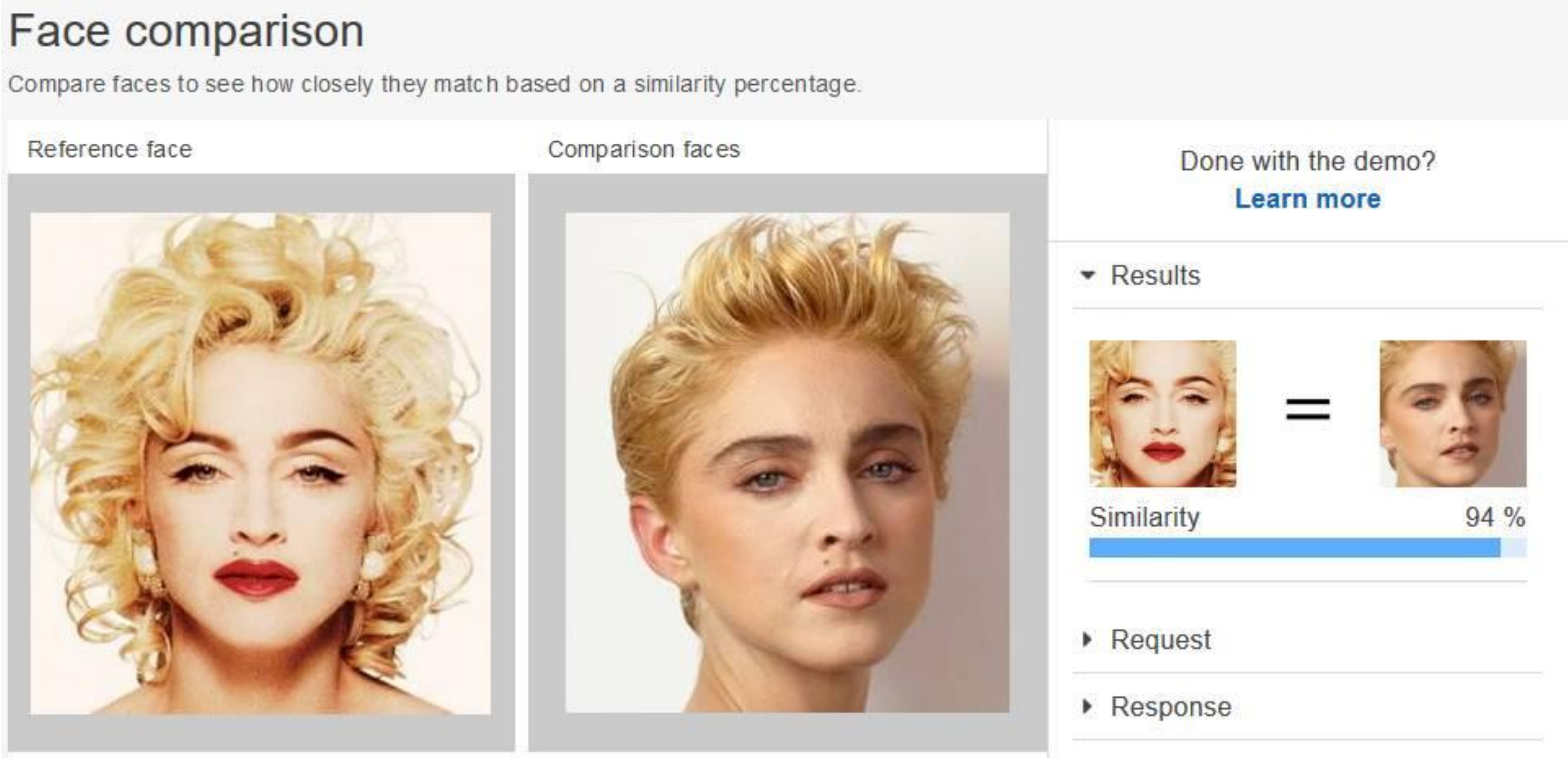
Match confidence 100 %

▸ Request

▸ Response

Amazon has special Celebrity recognition mode. Unfortunately it screwed up badly (P(m1) = 1, P(m3) = 0.96 and nothing else at all).

Can a machine compare images? Amazon Rekognition



Face comparison

Compare faces to see how closely they match based on a similarity percentage.

Reference face

Comparison faces

Done with the demo?
Learn more

Results

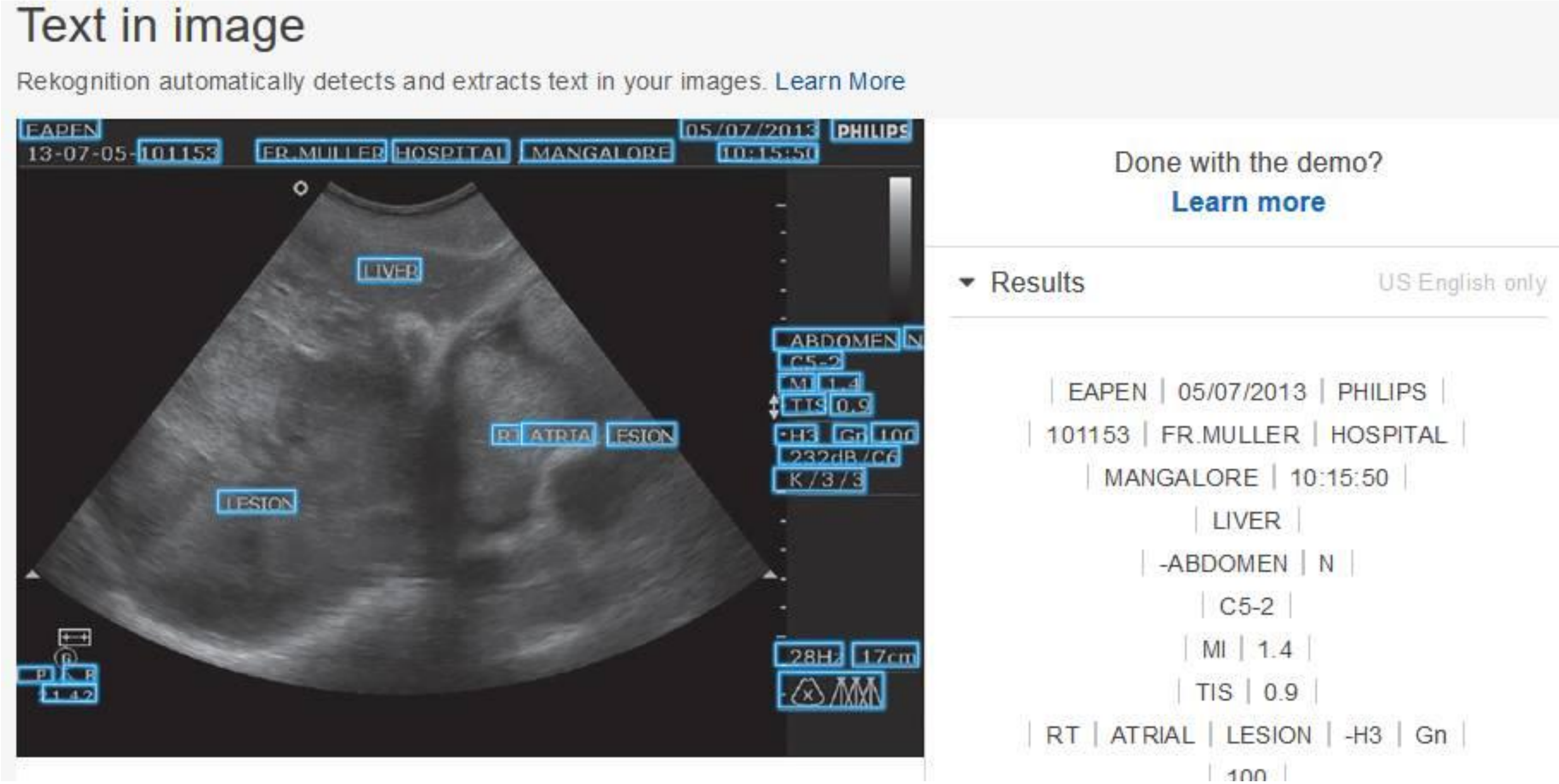
Similarity 94 %

Request

Response

Rekognition also has face comparison mode and at least it is self-consistent. But that’s pretty much it – none other pair of images is comparable.

Just to show that Amazon Rekognition may work quite well



Text in image

Rekognition automatically detects and extracts text in your images. Learn More

Done with the demo?
Learn more

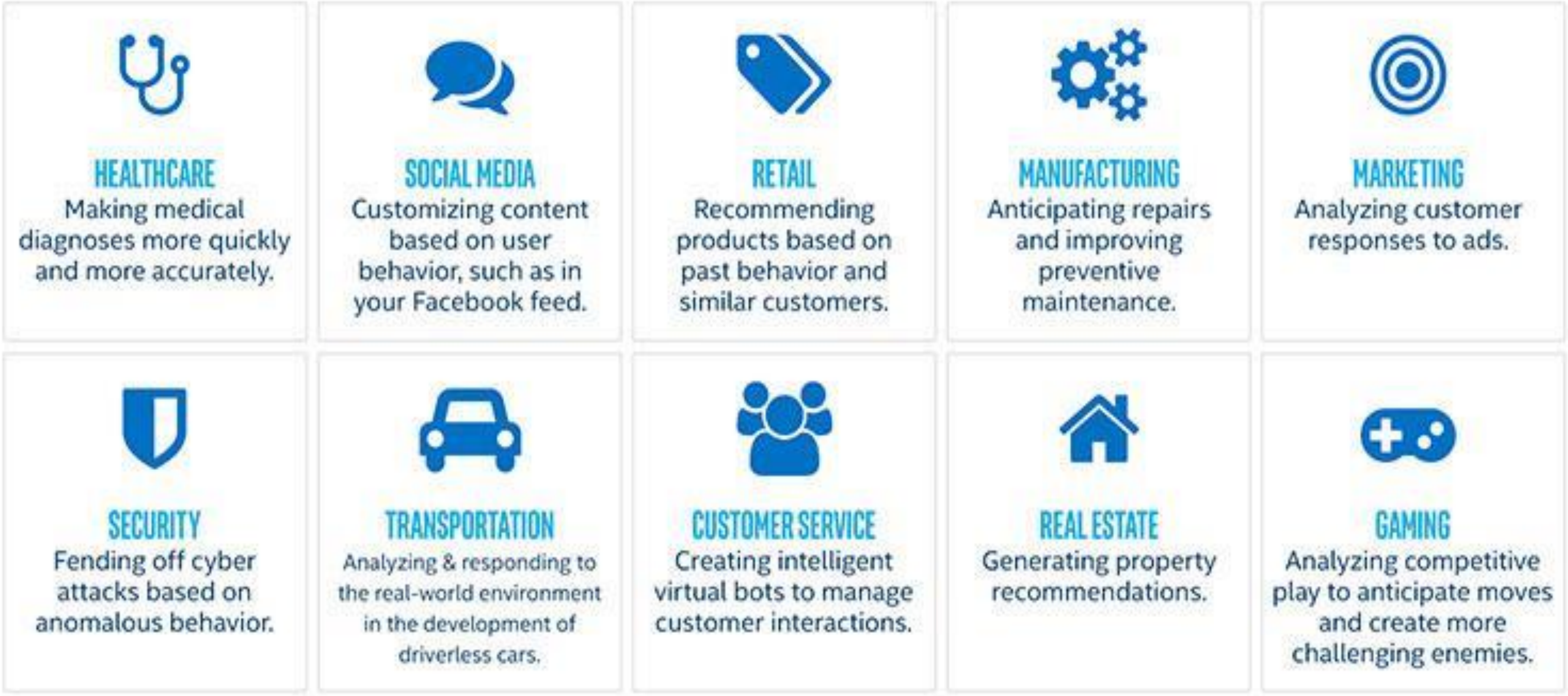
Results US English only

EAPEN	05/07/2013	PHILIPS
101153	FR.MULLER	HOSPITAL
MANGALORE	10:15:50
LIVER
-ABDOMEN	N
C5-2
MI	1.4
TIS	0.9
RT	ATRIAL	LESION	-H3	Gn
100

Amazon Rekognition in text detection mode.

Note, that examples above were Demos out-of-the-box, without adjustments or tuning.

Applications of ML by industry



HEALTHCARE
Making medical diagnoses more quickly and more accurately.

SOCIAL MEDIA
Customizing content based on user behavior, such as in your Facebook feed.

RETAIL
Recommending products based on past behavior and similar customers.

MANUFACTURING
Anticipating repairs and improving preventive maintenance.

MARKETING
Analyzing customer responses to ads.

SECURITY
Fending off cyber attacks based on anomalous behavior.

TRANSPORTATION
Analyzing & responding to the real-world environment in the development of driverless cars.

CUSTOMER SERVICE
Creating intelligent virtual bots to manage customer interactions.

REAL ESTATE
Generating property recommendations.

GAMING
Analyzing competitive play to anticipate moves and create more challenging enemies.

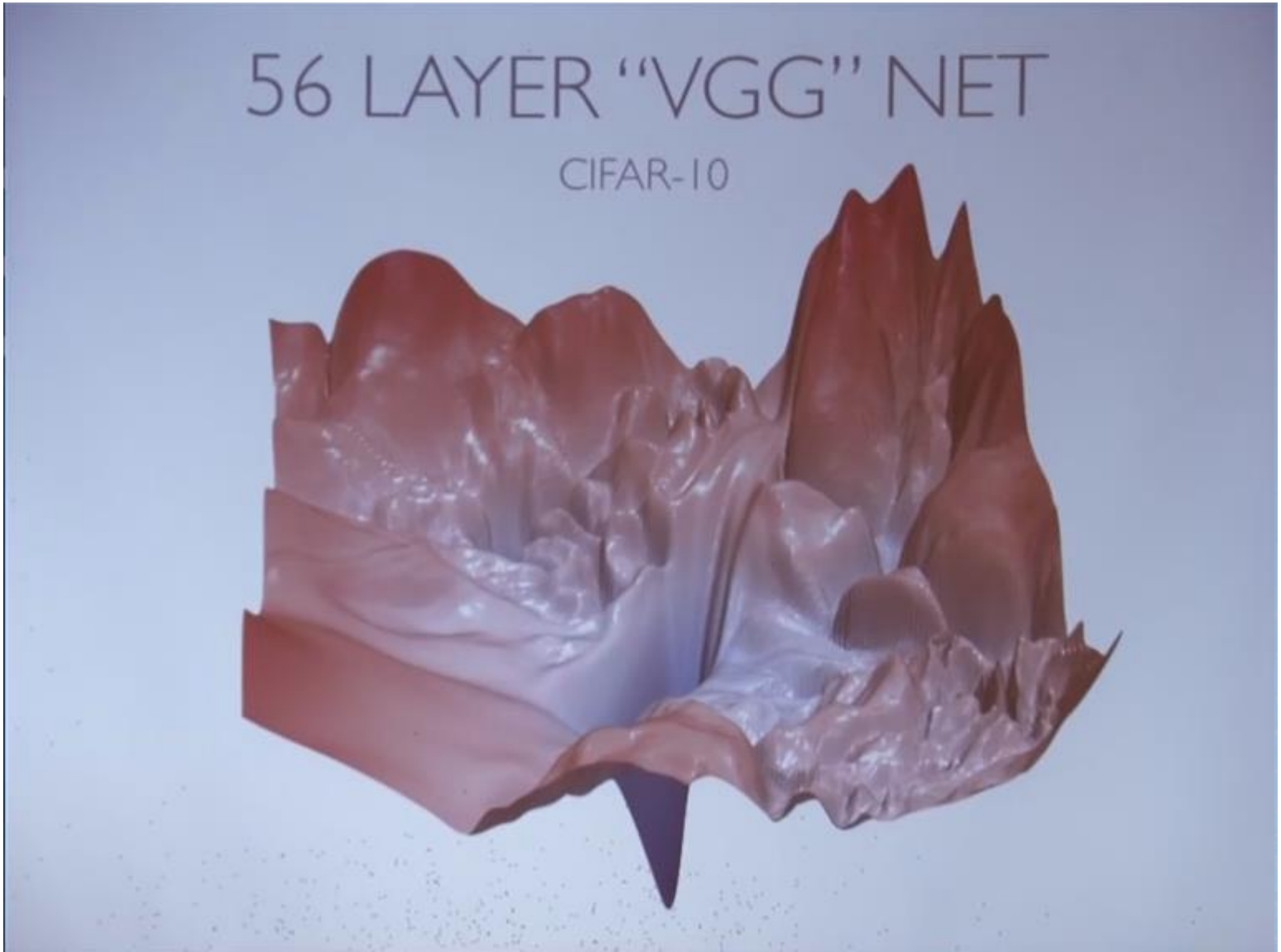
ML may be not perfect but it is coming

- The logic behind the new revolution is that this time it’s prediction which is going to become cheap
- Before the breakthroughs were computing, then internet search and now prediction
- The challenge is to build (reliable) systems
- The foundation has been laid out (or so we are lead to believe)
- Just as it is very useful to know how search or computing work, it is good to know how predictions work even if you are not planning to create ML models yourself
- Think of the OCF/IBM ML demo which analysed a medical dataset without asking or telling what the model was
- We are likely to see more tools with ML engines inside in the coming years

Why Deep Learning works?

- The world is **hierarchical**
- **Manifold** hypothesis - most input configurations are unlikely, real examples concentrate near a (much) lower dimensional manifold in feature space
- Although a single layer can be as powerful as **many layers**, deep (nested) layers need fewer neurons to create powerful representations
- Layer action is mostly linear (kernels are simple computationally)
- Which helps finding derivatives used in **backpropagation**
- Increased computational power (clusters of GPUs, TPUs)
- Availability of training data
- Technological advances such as batch normalisation, dropout etc
- Stochastic Gradient Descent (SGD); global minimum is not essential; optimisation process is never fully converged
- Human perception is fast: ~ 0.1 sec.
  Neurons fire at most 100 times a sec.
  Hence our neurons fire 10 times at most.
  Therefore 10 layers can solve any perception task

Tom Goldstein –
What do neural loss surfaces look like?



56 LAYER "VGG" NET

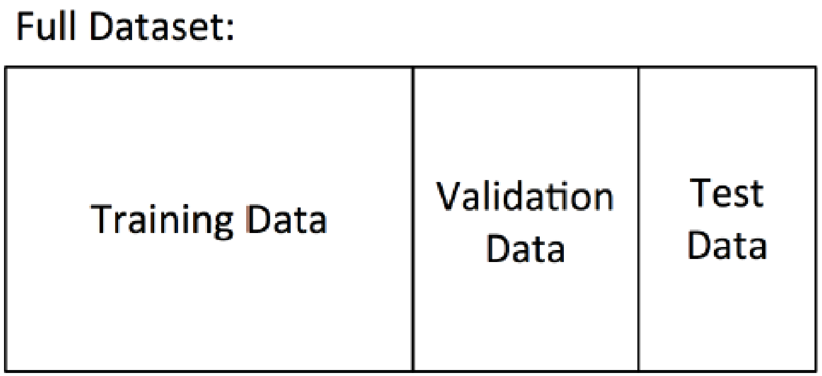
CIFAR-10

Typical Machine Learning Workflows

Traditional Machine Learning: Given data, **design feature set** and use a suitable ML algorithm to train a model, test on test data set.

Typical Deep Learning: Given data, **design DL architecture** and use it to train a model. Feature sets are created automatically (trained).

1. train (iteratively if necessary) on training set,
2. validate on validation set (change hyper parameters if necessary and repeat),
3. test on test set (this is a test of the power of your model: its accuracy and generalisation)



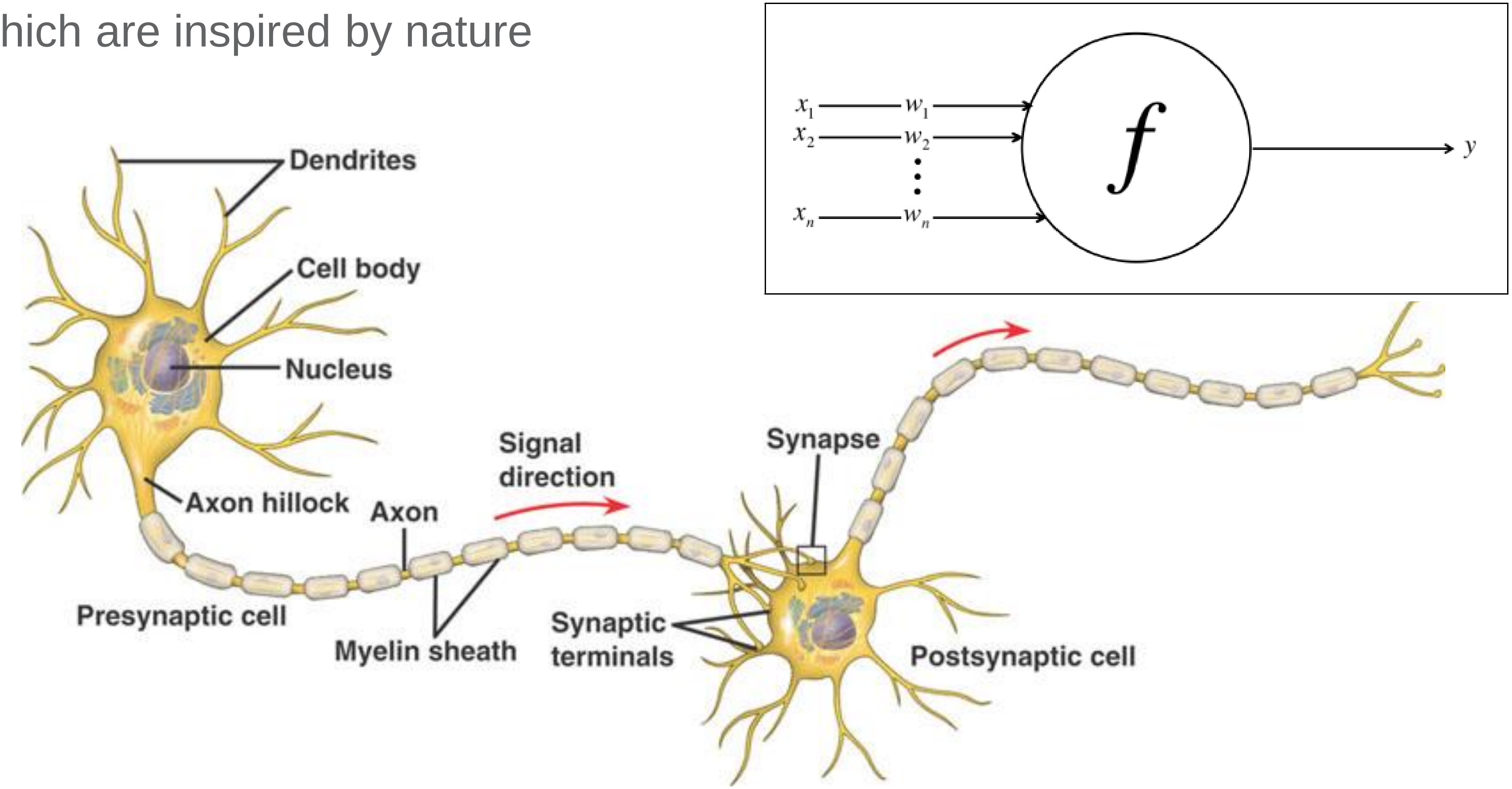
Full Dataset:

Training Data	Validation Data	Test Data

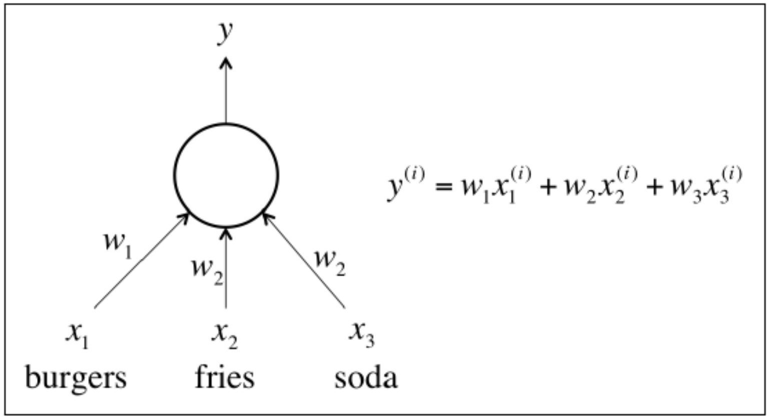
Deep Learning uses

Neural Networks

which are inspired by nature



Simple Three Layer Neural Network

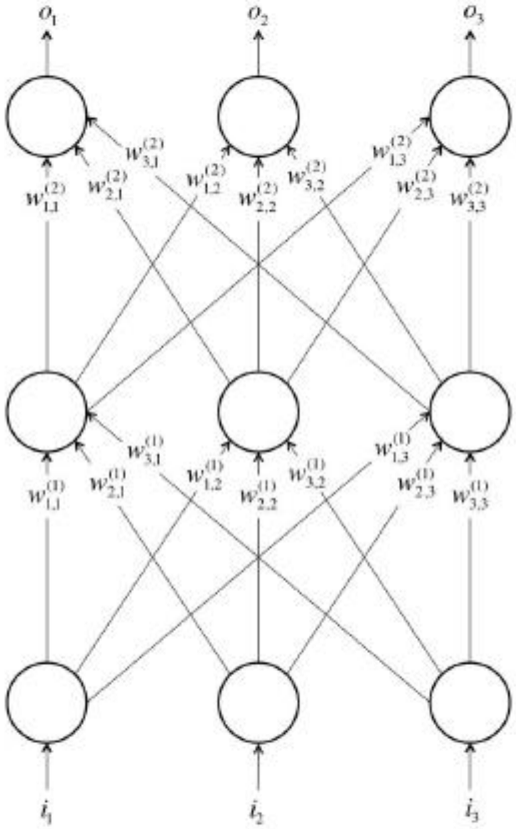


$$y^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + w_3 x_3^{(i)}$$

Output layer - layer 3

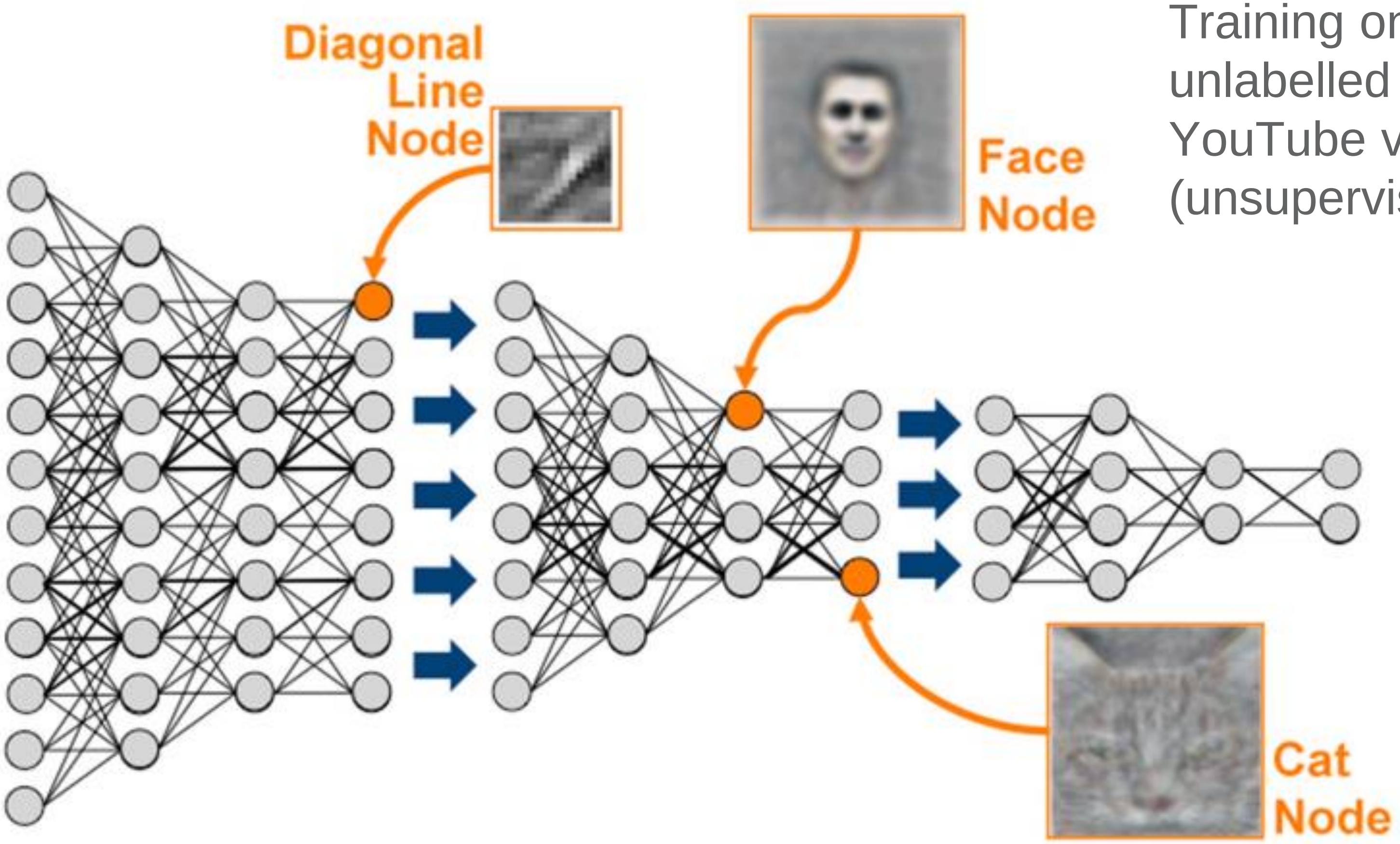
Hidden layer - layer 2

Input layer - layer 1



This is a flat network. Deep networks have more layers or network hops.

Hierarchical multi-layer network



Training on 10 M unlabelled frames from YouTube videos (unsupervised learning).

Hierarchy of Representations

Instead of traditional approach using feature design, thanks to emergence of large data sets, features are trained directly from data (which is why number of parameters in models is quite large – tens of millions is common). Input is pre-processed (e.g. average subtracted, signal normalised)

From concrete to abstract:

- vision: pixel, motif, part, object
- text: character, word, clause, sentence
- speech: audio, band, phoneme, word

MNIST - one of the first successes

‘Hello World’ of Deep Learning

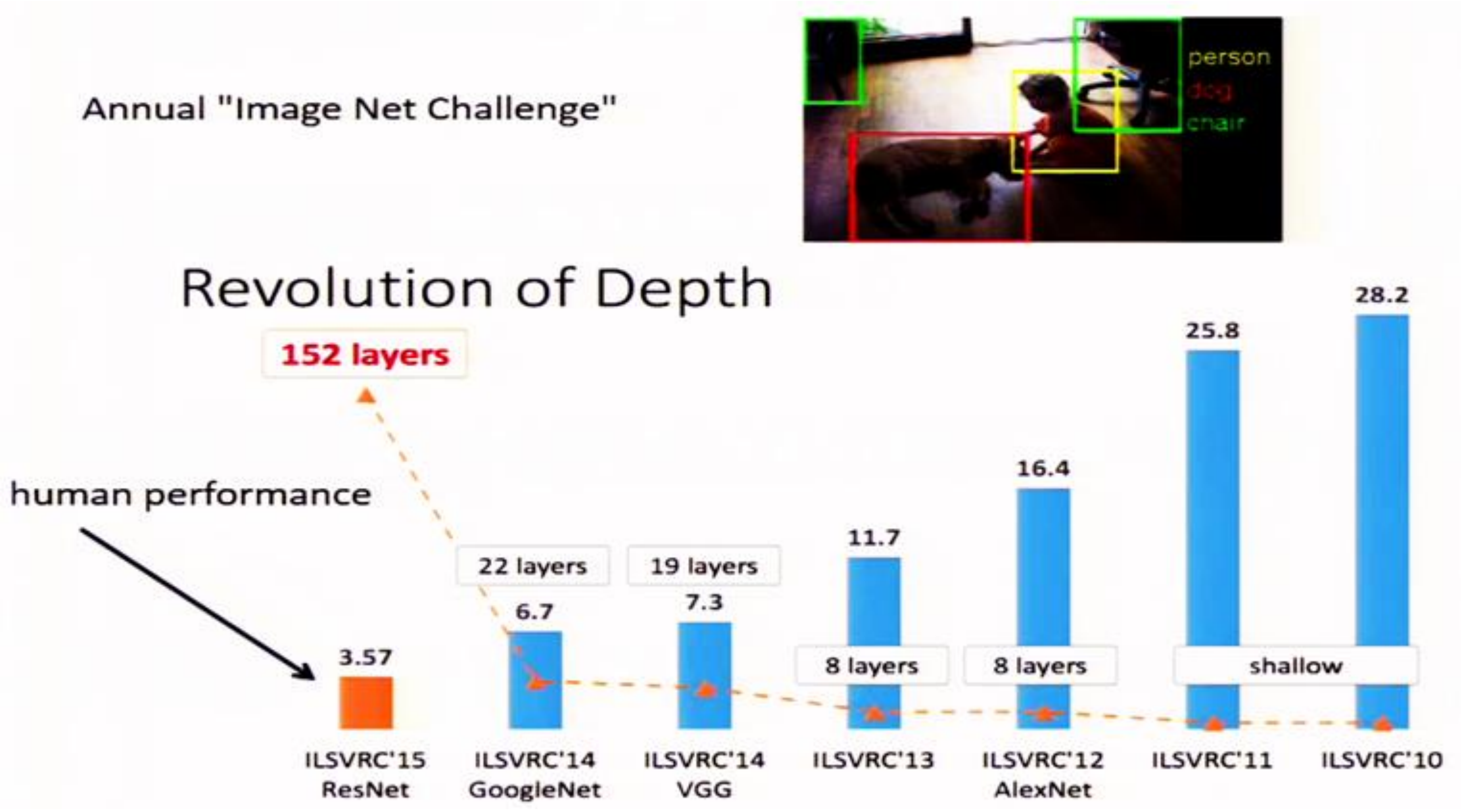
- created by Yann LeCun et al
- database of handwritten digits
- a training set of 60,000 and a test set of 10,000 labelled 28x28 images
- end-of-life as a useful benchmark

Type	Error Rate %
Linear Classifier	7.6
K-Nearest Neighbours	0.52
Support vector machine	0.56
Convolutional neural network	0.23

Table based on https://en.wikipedia.org/wiki/MNIST_database

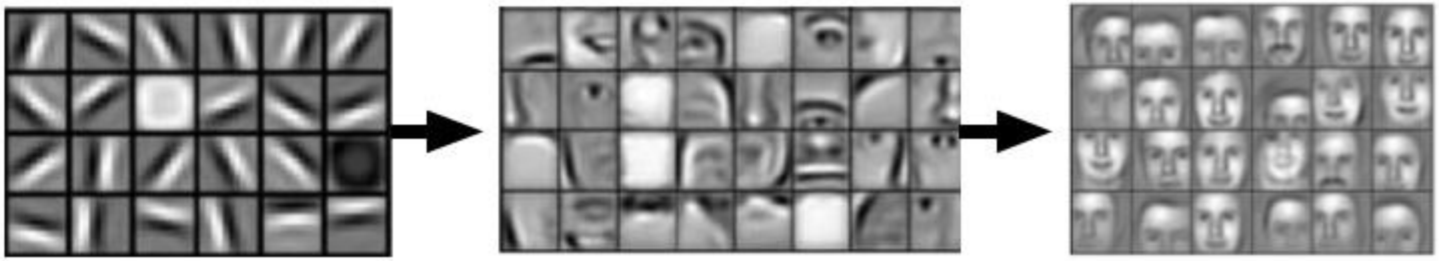
Progress in recent years -

Visual Object Classification



Convolutional Networks

- Features are discovered automatically; layer 1 features resemble Gabor filters; layers represent hierarchy of features

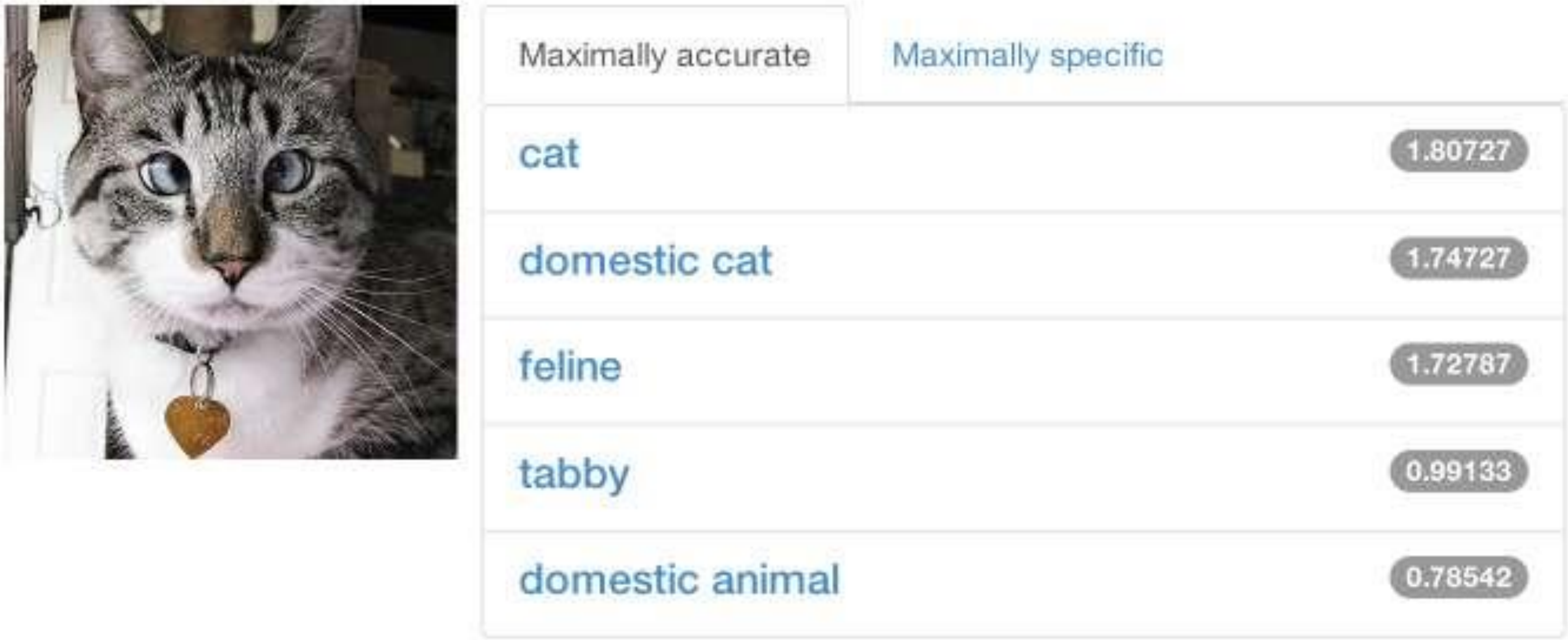


- Training is done iteratively (forward pass is followed by backprop) and may take a long time

- Once the training is done the inference (the forward pass) is usually quick making real time tagging (identification of live stream) possible.

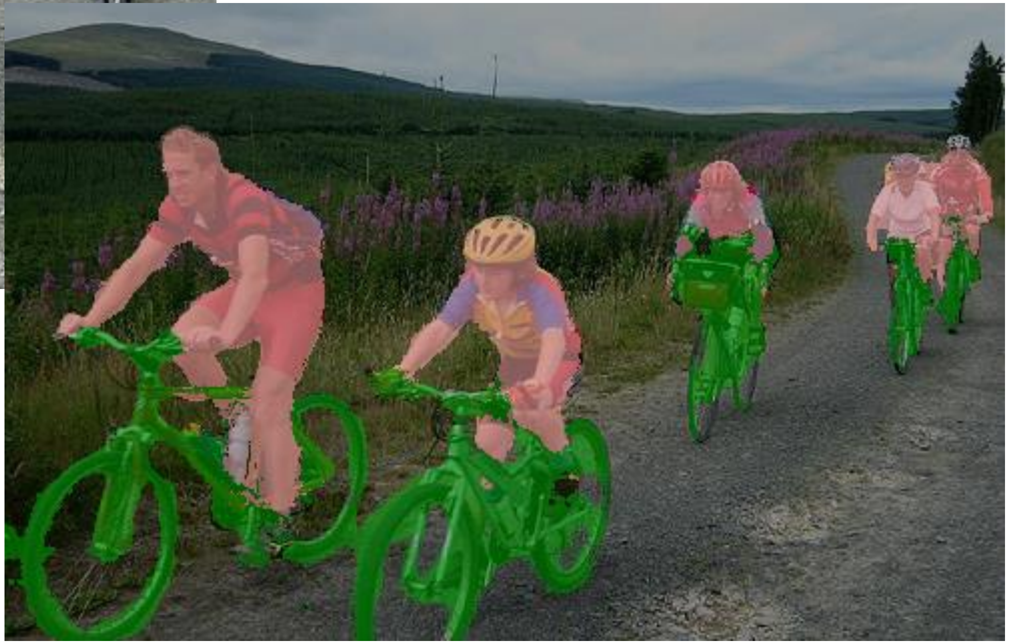
ConvNet

Find probabilities (or scores) of labels



cat	1.80727
domestic cat	1.74727
feline	1.72787
tabby	0.99133
domestic animal	0.78542

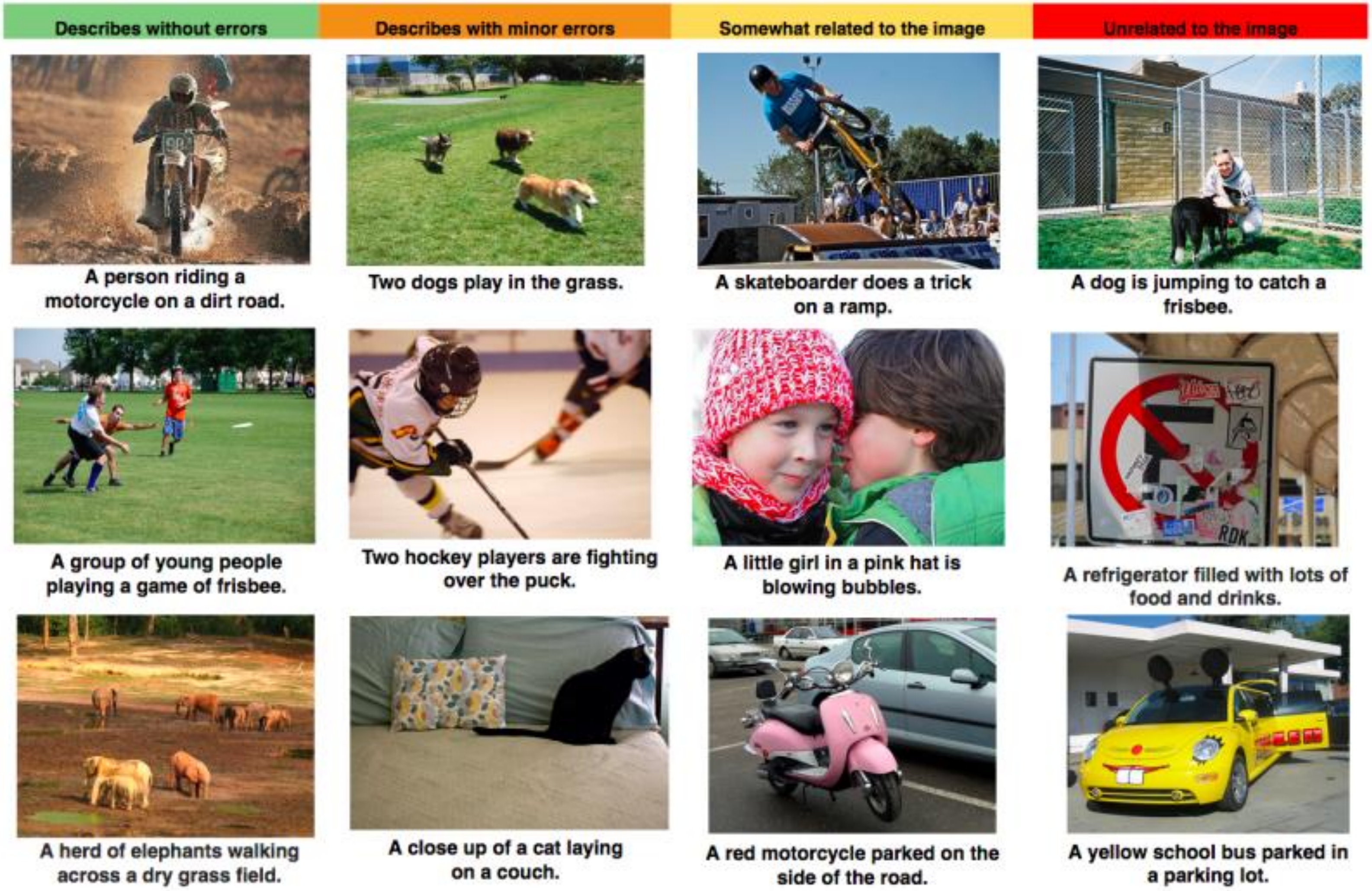
Semantic Image Segmentation



Objects appearing in the image:
Bicycle,
Person

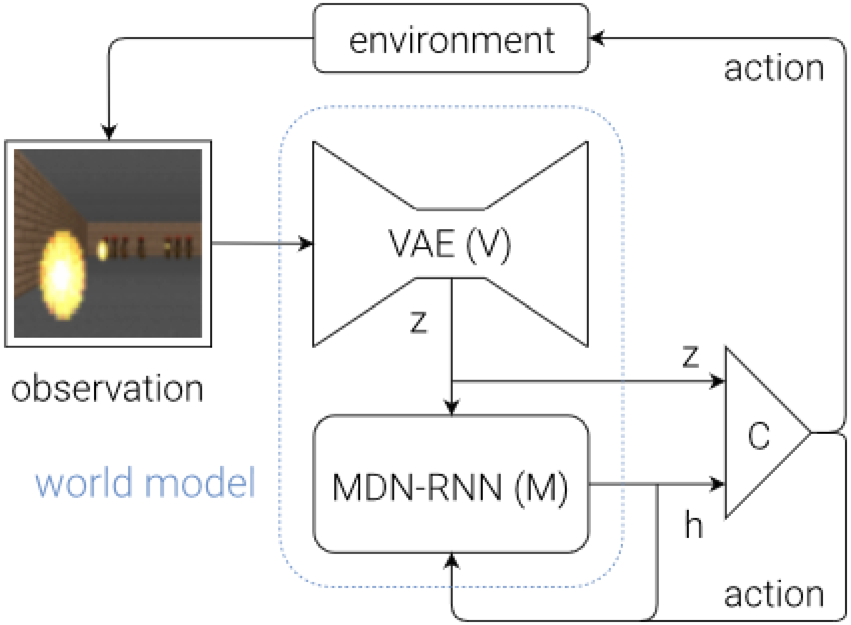
http://www.robots.ox.ac.uk/~szheng/crfasrnndemo

Image captions



Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image
A person riding a motorcycle on a dirt road.	Two dogs play in the grass.	A skateboarder does a trick on a ramp.	A dog is jumping to catch a frisbee.
A group of young people playing a game of frisbee.	Two hockey players are fighting over the puck.	A little girl in a pink hat is blowing bubbles.	A refrigerator filled with lots of food and drinks.
A herd of elephants walking across a dry grass field.	A close up of a cat laying on a couch.	A red motorcycle parked on the side of the road.	A yellow school bus parked in a parking lot.

Recent headlines



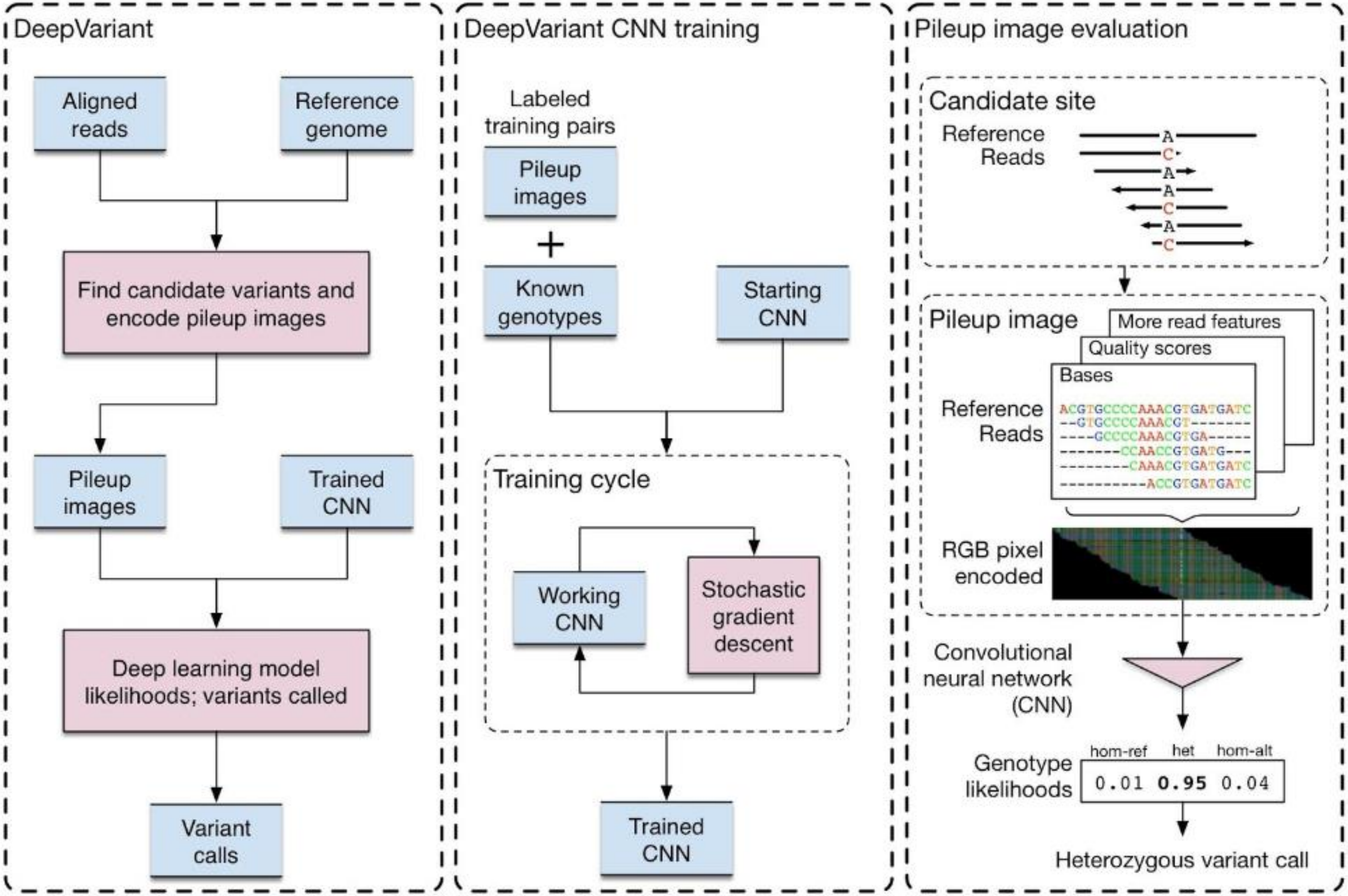
World model to play Doom (reinforcement learning)

Pogressive Growing of GANs

https://github.com/tkarras/progressive_growing_of_gans

From AlphaGo to AlphaZero (DeepMind)

DeepVariant



DeepVariant

Aligned reads

Reference genome

Find candidate variants and encode pileup images

Pileup images

Trained CNN

Deep learning model likelihoods; variants called

Variant calls

DeepVariant CNN training

Labeled training pairs

Pileup images

+

Known genotypes

Starting CNN

Training cycle

Working CNN

Stochastic gradient descent

Trained CNN

Pileup image evaluation

Candidate site

Reference

Reads

Pileup image

More read features

Quality scores

Bases

Reference

Reads

RGB pixel encoded

Convolutional neural network (CNN)

Genotype likelihoods

hom-ref	het	hom-alt
0.01	**0.95**	0.04

Heterozygous variant call

References and homework

- François Chollet, *Deep Learning with Python*, Manning 2017
  Chapter 1: What is Deep Learning
  https://www.manning.com/books/deep-learning-with-python

- Ian Goodfellow and Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press 2016
  Chapter 5: Machine Learning Basics
  http://www.deeplearningbook.org/contents/ml.html

- Homework to refresh Numpy
  https://notebooks.azure.com/ink1/libraries/IntroductionToPython/html/numpy_intro.ipynb

- Let us know if you would like to join Deep Learning group at ICR
  https://nexus.icr.ac.uk/teams/Deep%20Learning/

Next week Sebastian Poelsterl will cover

1. Linear Algebra
   - Scalar, Vector, Matrix and Tensor
   - Transpose
   - Dot product
   - Identity matrix
   - Linear system of equations
   - Matrix inverse
   - Matrix and vector norms
   - Symmetric and orthogonal matrix

2. Calculus
   - Gradient
   - Hessian / curvature
   - Gradient descent / iterative optimisation
   - Minimum / Maximum / Saddle Point

3. Least squares

4. Logistic regression

For refresher, see https://www.khanacademy.org/math/linear-algebra

ICR