Deep Learning for Developers

16/12/2017

Julien Simon, AI Evangelist, EMEA

@julsimon

What to expect

- AI ?
- An introduction to Deep Learning
- Common neural network architectures and use cases
- An introduction to Apache MXNet
- Demos using Jupyter notebooks on Amazon SageMaker
- Resources

- Artificial Intelligence: design software applications which exhibit human-like behavior, e.g. speech, natural language processing, reasoning or intuition

- Machine Learning: teach machines to learn without being explicitly programmed

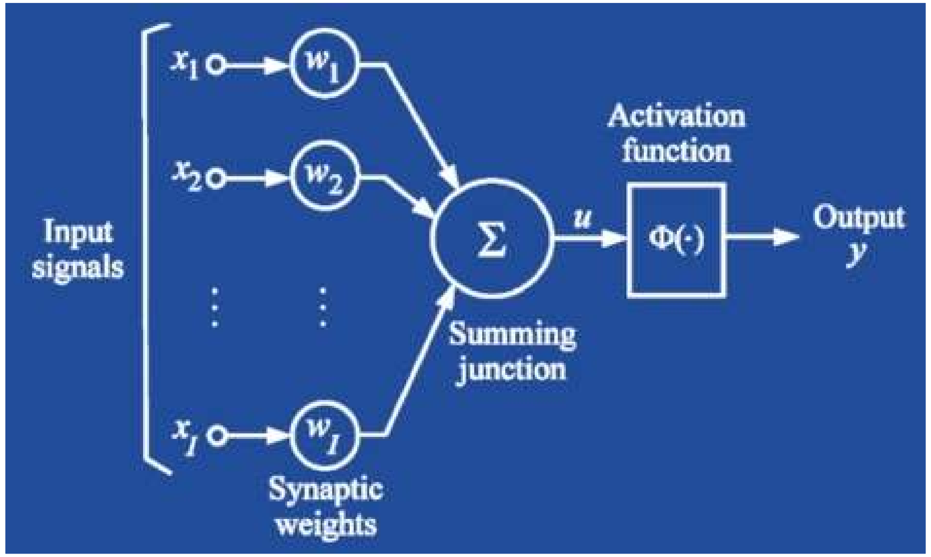
- Deep Learning: using neural networks, teach machines to learn from complex data where features cannot be explicitly expressed

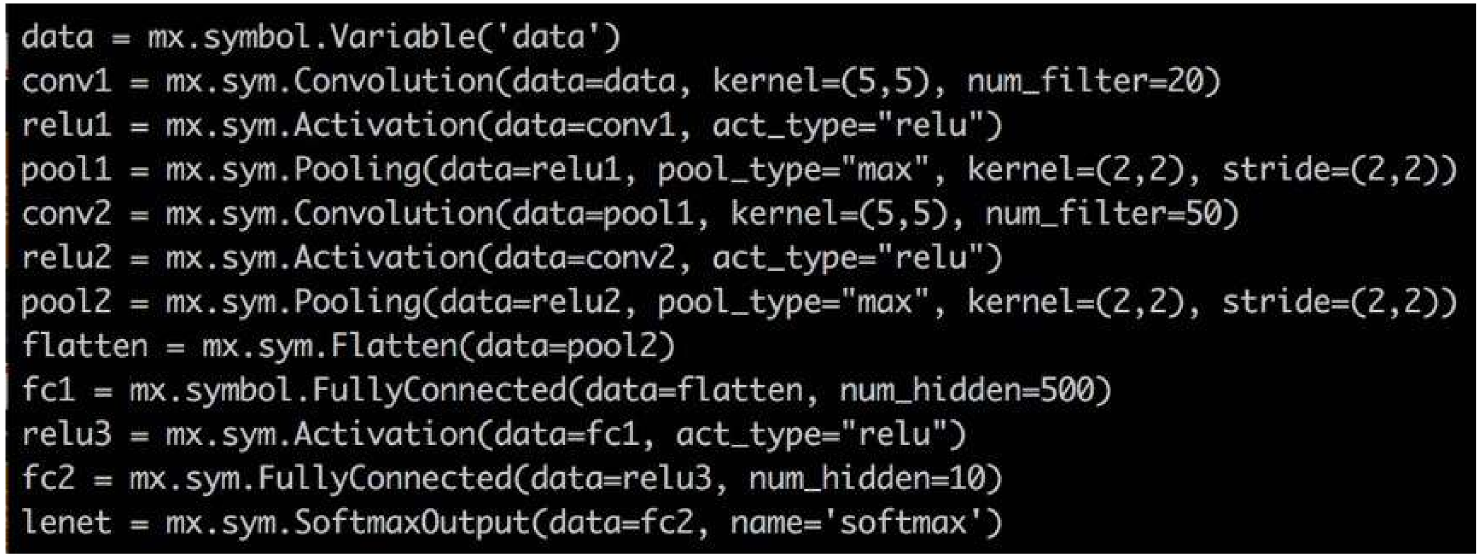
Myth: AI is dark magic

aka « You're not smart enough »

Fact: AI is math, code and chips

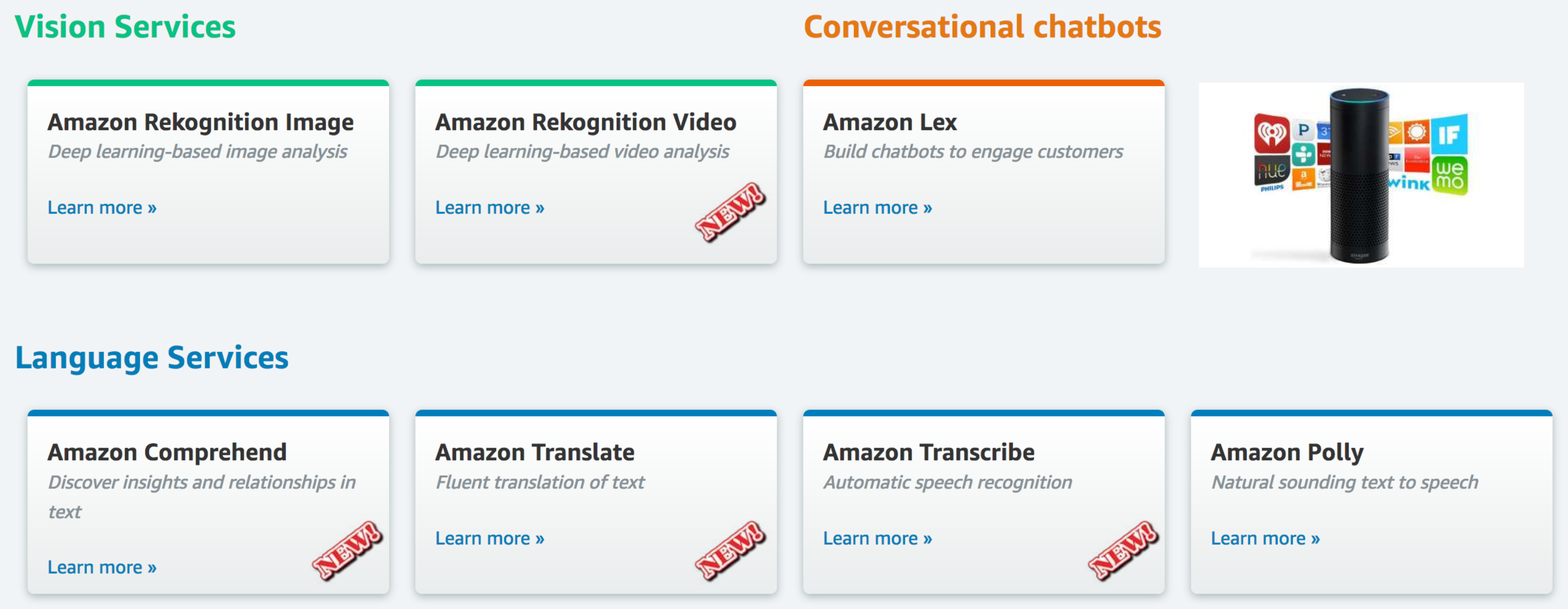
A bit of Science, a lot of Engineering





```
data = mx.symbol.Variable('data')
conv1 = mx.sym.Convolution(data=data, kernel=(5,5), num_filter=20)
relu1 = mx.sym.Activation(data=conv1, act_type="relu")
pool1 = mx.sym.Pooling(data=relu1, pool_type="max", kernel=(2,2), stride=(2,2))
conv2 = mx.sym.Convolution(data=pool1, kernel=(5,5), num_filter=50)
relu2 = mx.sym.Activation(data=conv2, act_type="relu")
pool2 = mx.sym.Pooling(data=relu2, pool_type="max", kernel=(2,2), stride=(2,2))
flatten = mx.sym.Flatten(data=pool2)
fc1 = mx.symbol.FullyConnected(data=flatten, num_hidden=500)
relu3 = mx.sym.Activation(data=fc1, act_type="relu")
fc2 = mx.sym.FullyConnected(data=relu3, num_hidden=10)
lenet = mx.sym.SoftmaxOutput(data=fc2, name='softmax')
```

Amazon AI is based on Deep Learning



Vision Services

Amazon Rekognition Image
Deep learning-based image analysis

Learn more »

Amazon Rekognition Video
Deep learning-based video analysis

Learn more »

NEW!

Conversational chatbots

Amazon Lex
Build chatbots to engage customers

Learn more »

Language Services

Amazon Comprehend
Discover insights and relationships in text

Learn more »

NEW!

Amazon Translate
Fluent translation of text

Learn more »

NEW!

Amazon Transcribe
Automatic speech recognition

Learn more »

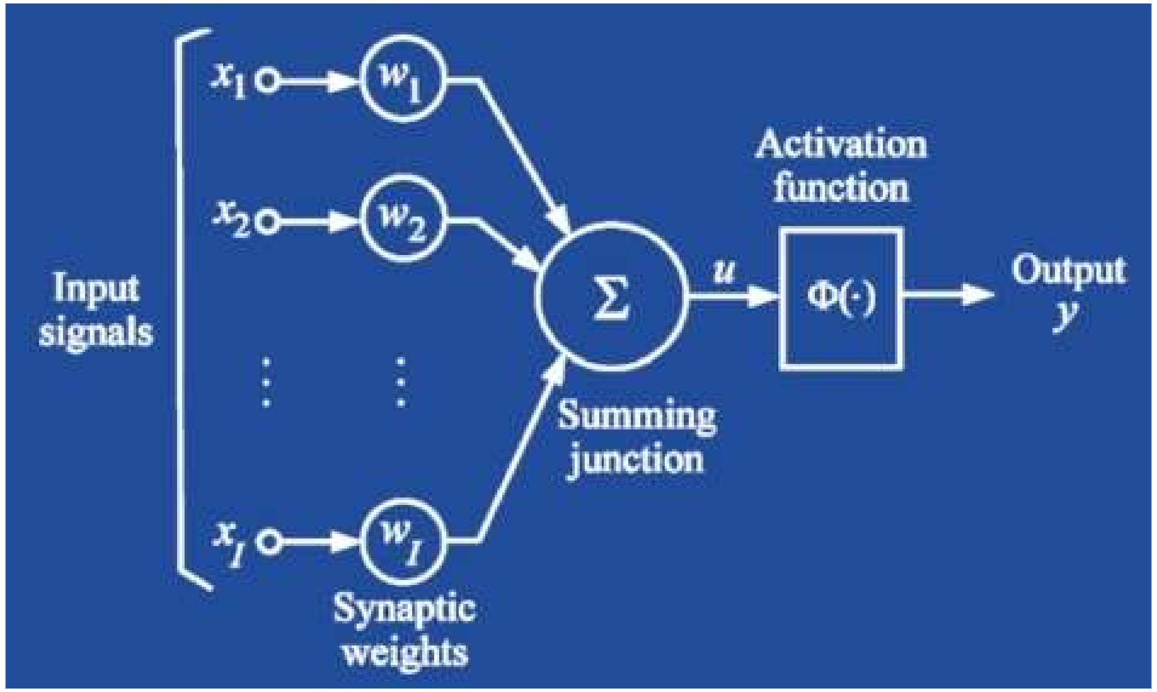
NEW!

Amazon Polly
Natural sounding text to speech

Learn more »

An introduction to Deep Learning

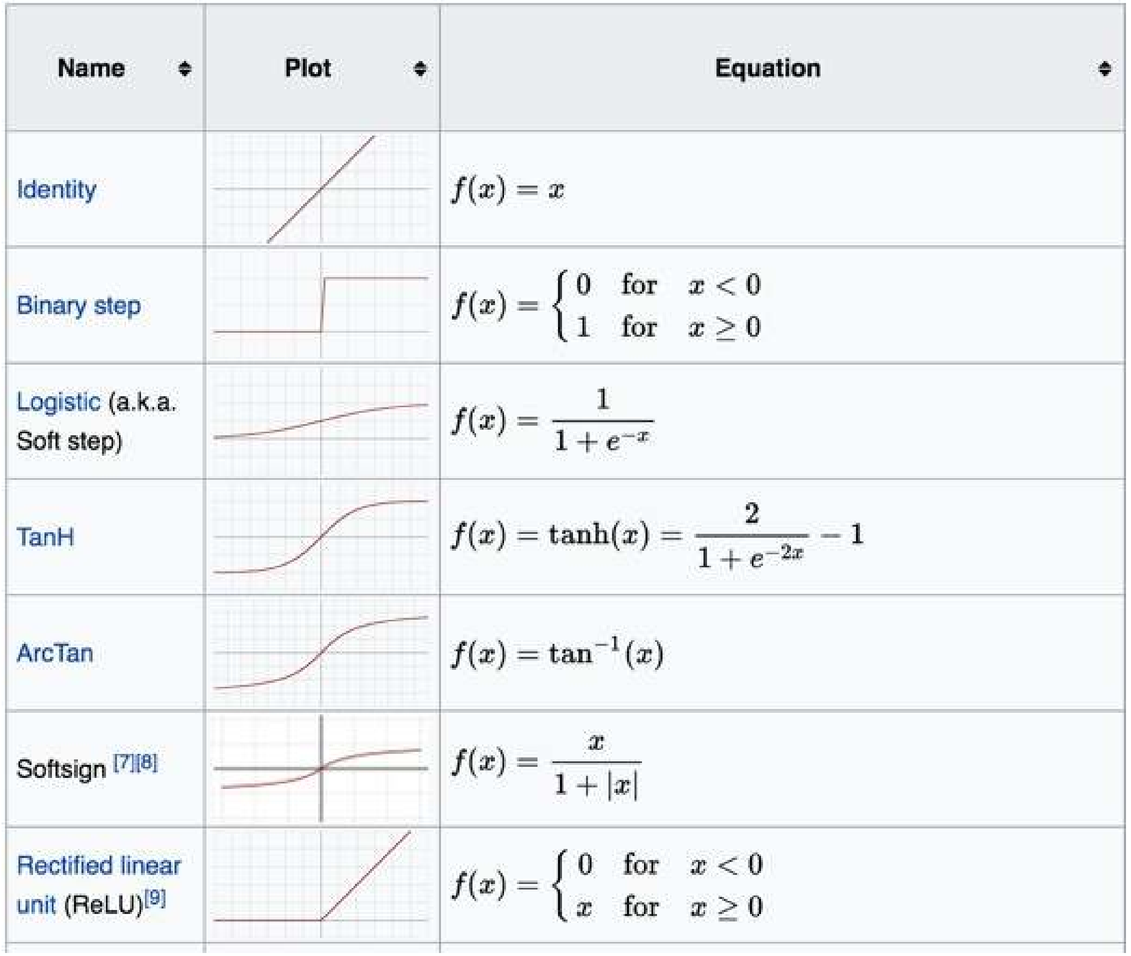
The neuron



$$\sum_{i=1}^{l} x_i * w_i = u$$

”Multiply and Accumulate”

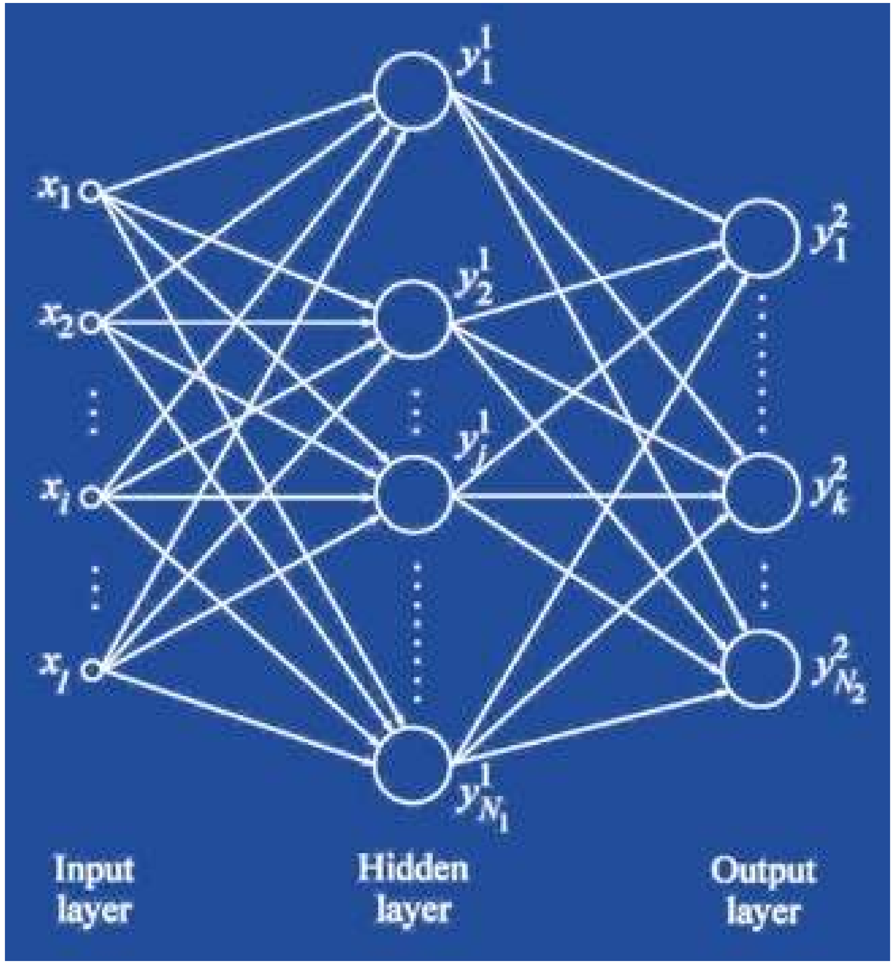
Activation functions



Name	Plot	Equation
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for} & x < 0 \\ 1 & \text{for} & x \geq 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Softsign [7][8]		$f(x) = \frac{x}{1 + \lvert x \rvert}$
Rectified linear unit (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for} & x < 0 \\ x & \text{for} & x \geq 0 \end{cases}$

Source: Wikipedia

Neural networks



l features

$$x = \begin{bmatrix} x_{11}, x_{12}, \dots x_{1l} \\ x_{21}, x_{22}, \dots x_{2l} \\ \dots \dots \dots \\ x_{m1}, x_{m2}, \dots x_{ml} \end{bmatrix}$$

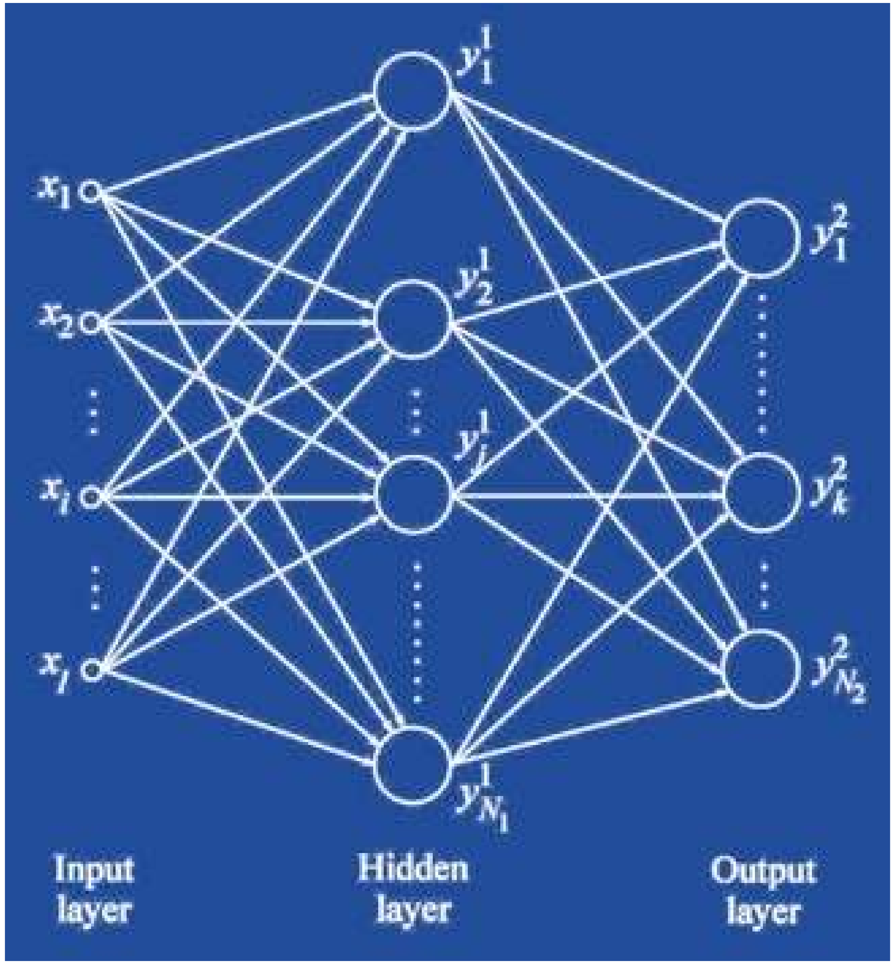
m samples

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix} \quad \begin{bmatrix} 0,0,1,0,0,\dots,0 \\ 1,0,0,0,0,\dots,0 \\ \dots \\ 0,0,0,0,1,\dots,0 \end{bmatrix}$$

One-hot encoding

m labels, N_2 outputs

Neural networks



l features

$$x = \begin{bmatrix} x_{11}, x & \dots & x_{1l} \\ x_{21}, x & \dots & x_{2l} \\ \dots & \dots & \\ x_{m1}, x & \dots & \end{bmatrix}$$

m samples

$x_{11}, x_{12}, \dots, x_{1l}$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix} \quad \begin{bmatrix} 0,0,1, \dots,0 \\ 1,0,0, \dots,0 \\ \dots \\ 0,0,0, \dots,0 \end{bmatrix}$$

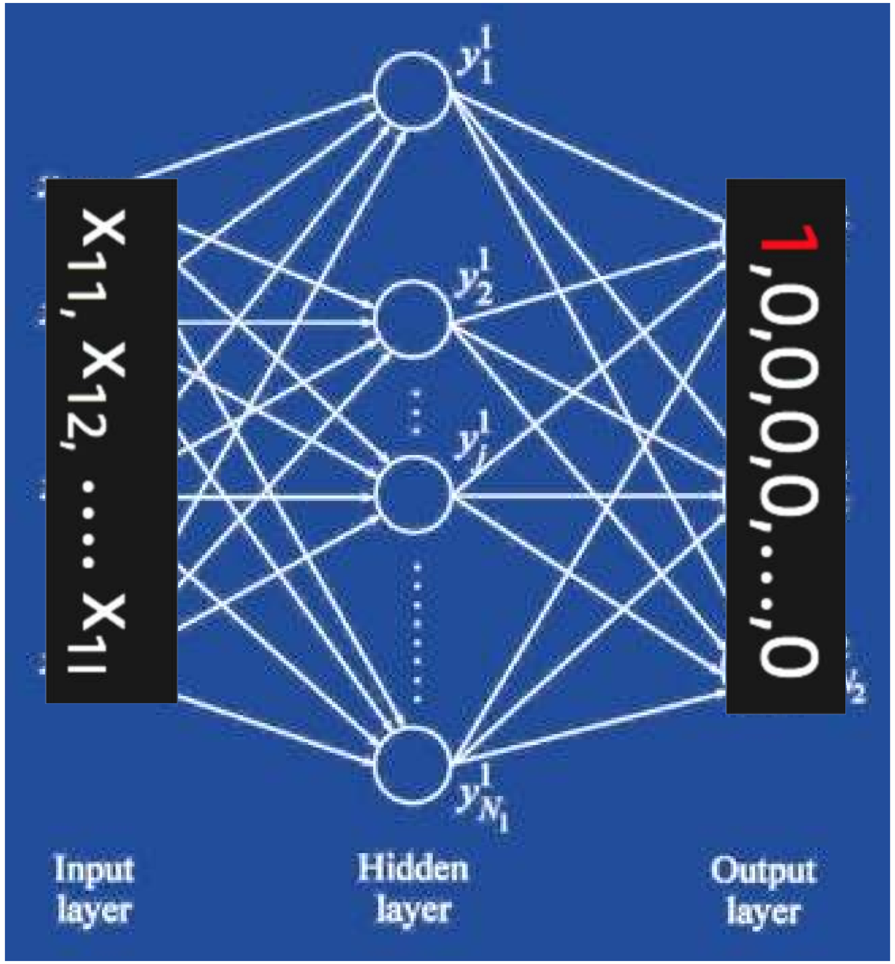
0,0,1,0,0,…,0

One-hot encoding

m labels, N_2 categories

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Neural networks



Initially, the network will not predict correctly

$f(X_1) = Y'_1$

A loss function measures the difference between the real label Y_1 and the predicted label Y'_1

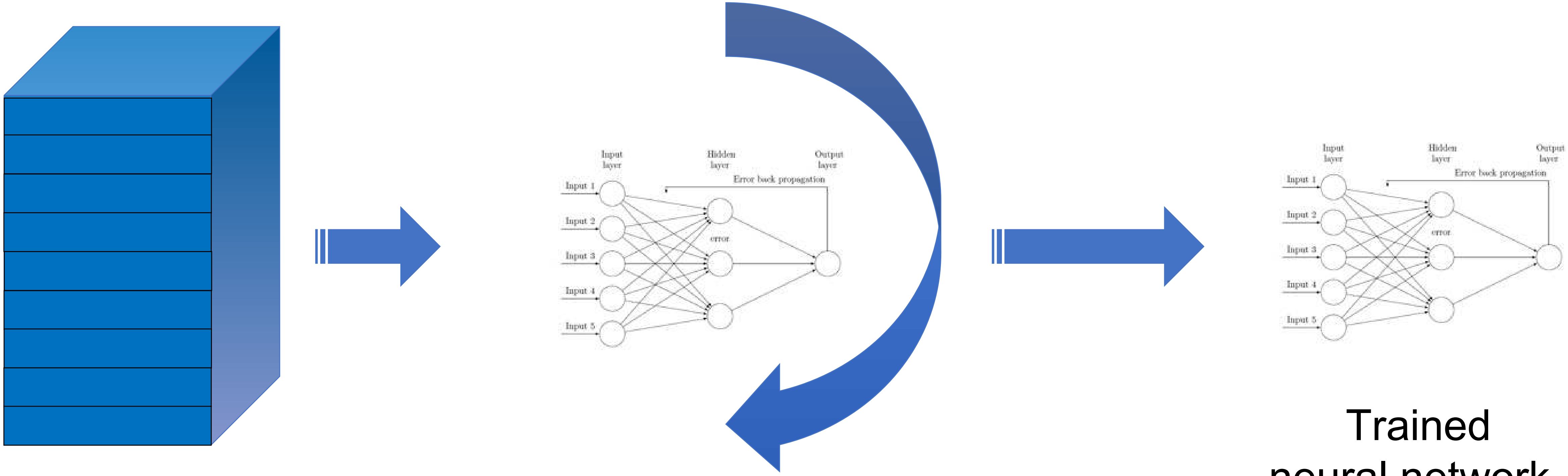
$error = loss(Y_1, Y'_1)$

For a batch of samples:

$$\sum_{i=1}^{batch\ size} loss(Y_i, Y'_i) = \text{batch error}$$

The purpose of the training process is to minimize error by gradually adjusting weights

Training



Training data set

Backpropagation

Trained
neural network

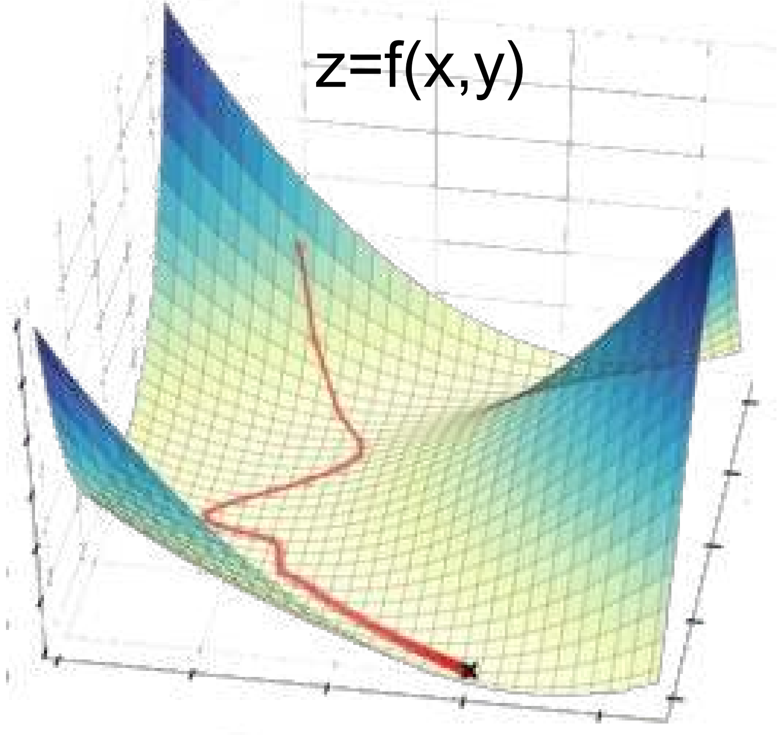
Batch size
Learning rate
Number of epochs

Hyper parameters

Stochastic Gradient Descent (SGD)

Imagine you stand on top of a mountain with skis strapped to your feet. You want to get down to the valley as quickly as possible, but there is fog and you can only see your immediate surroundings. How can you get down the mountain as quickly as possible? You look around and identify the steepest path down, go down that path for a bit, again look around and find the new steepest path, go down that path, and repeat—this is exactly what gradient descent does.

Tim Dettmers
University of Lugano
2015

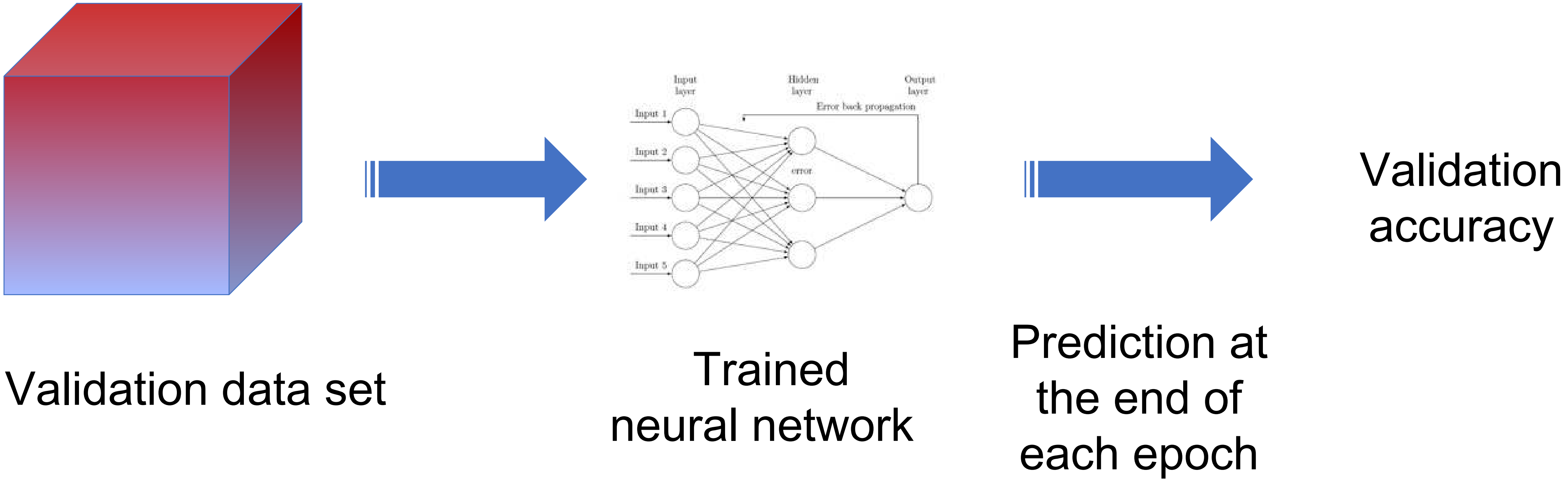


The « step size » is called the learning rate

Alternatives to SGD: rmsprod, adagrad, adadelta, adam, etc.

https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/

Validation



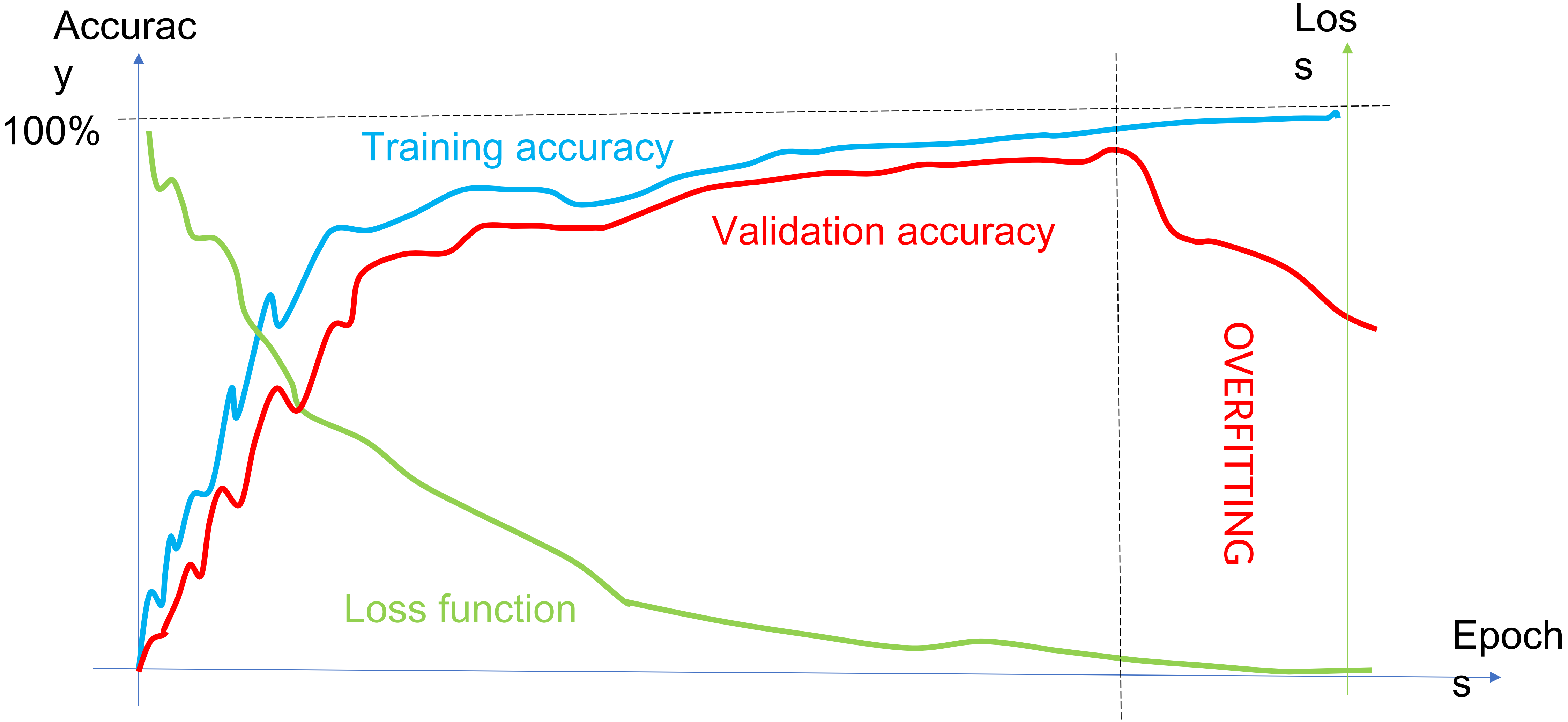
Validation data set

Trained neural network

Prediction at the end of each epoch

Validation accuracy

Early stopping



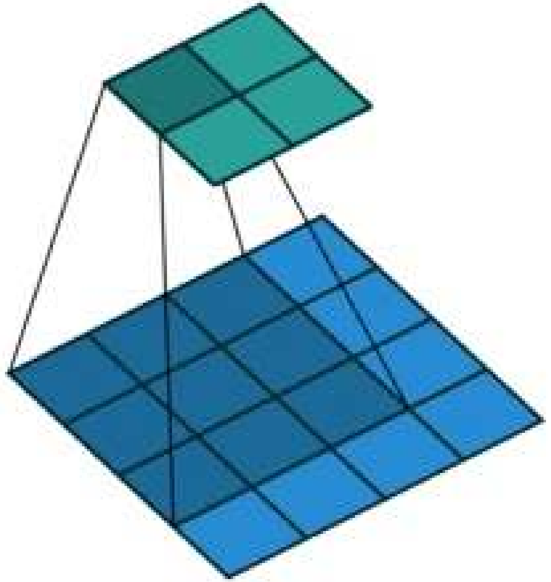
Save the model at the end of each epoch ☺

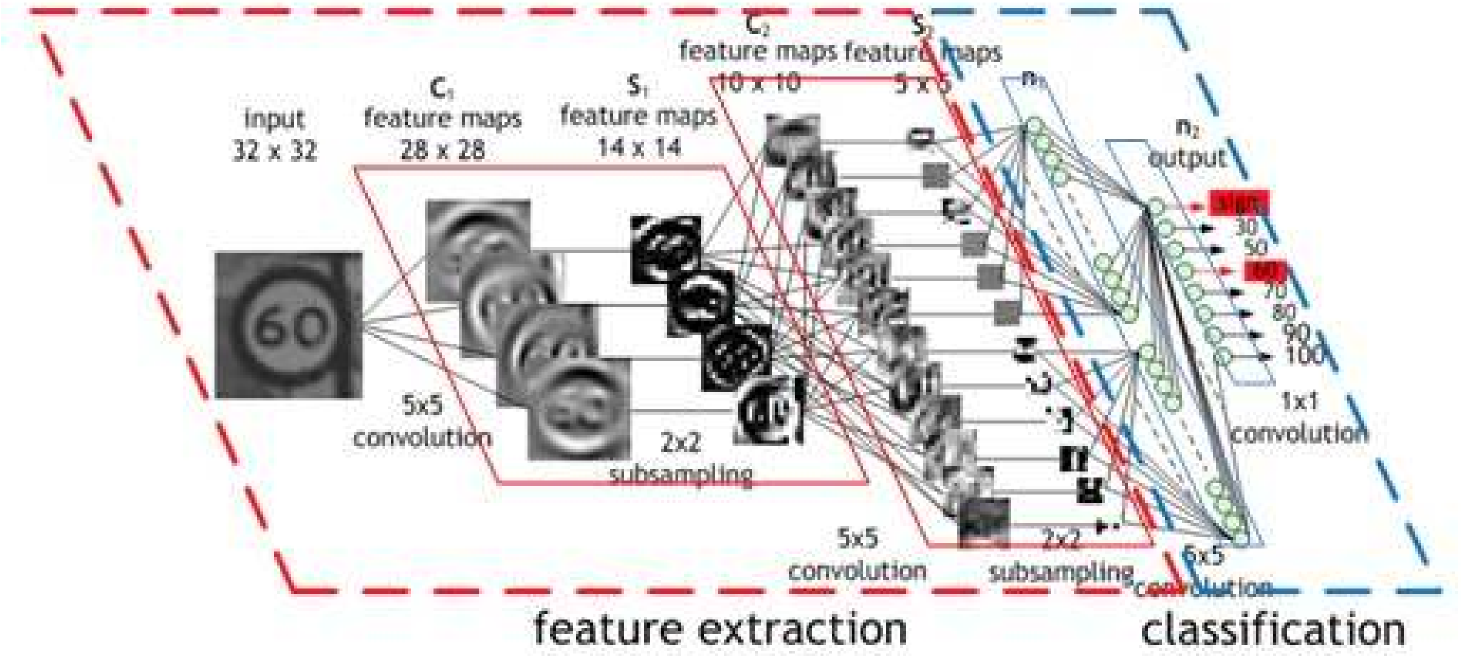
Common network architectures and use cases

Convolutional Neural Networks (CNN)

Le Cun, 1998: handwritten digit recognition, 32x32 pixels

Convolution and pooling reduce dimensionality



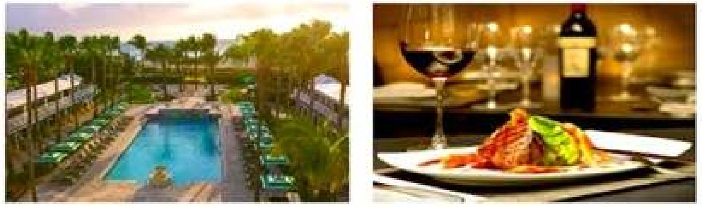


https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

CNN: Object Classification

- Expedia have over 10 million images from 300,000 hotels
- Using great images boosts conversion
- Using Keras and EC2 GPU instances, they fine-tuned a pre-trained Convolutional Neural Network using 100,000 images
- Hotel descriptions now automatically feature the best available images

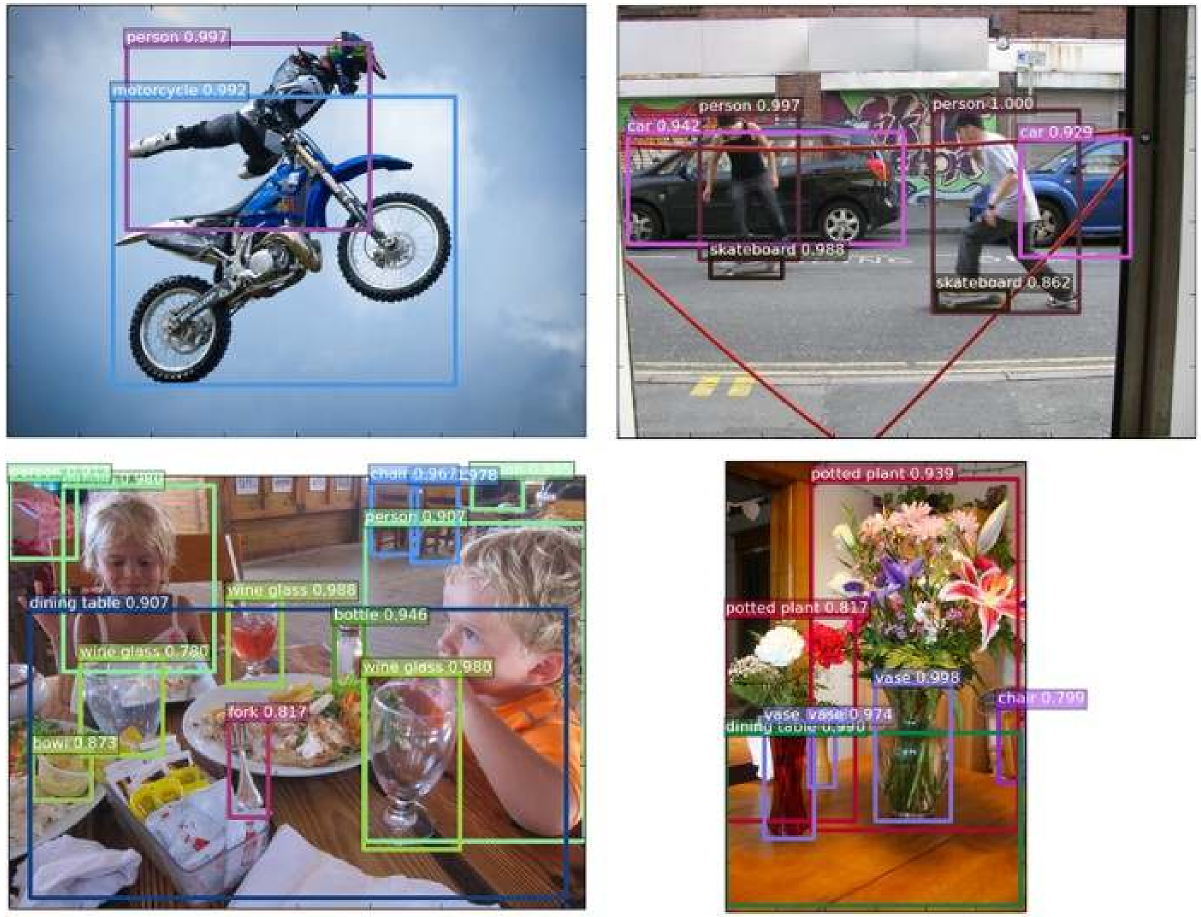
Some images are really good



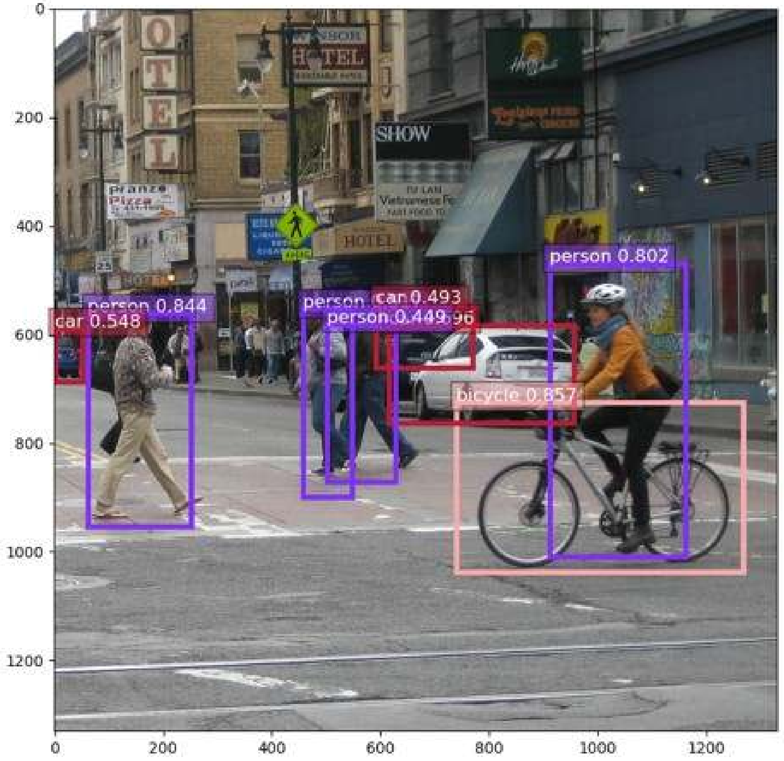
Others not so much

https://news.developer.nvidia.com/expedia-ranking-hotel-images-with-deep-learning/

CNN: Object Detection



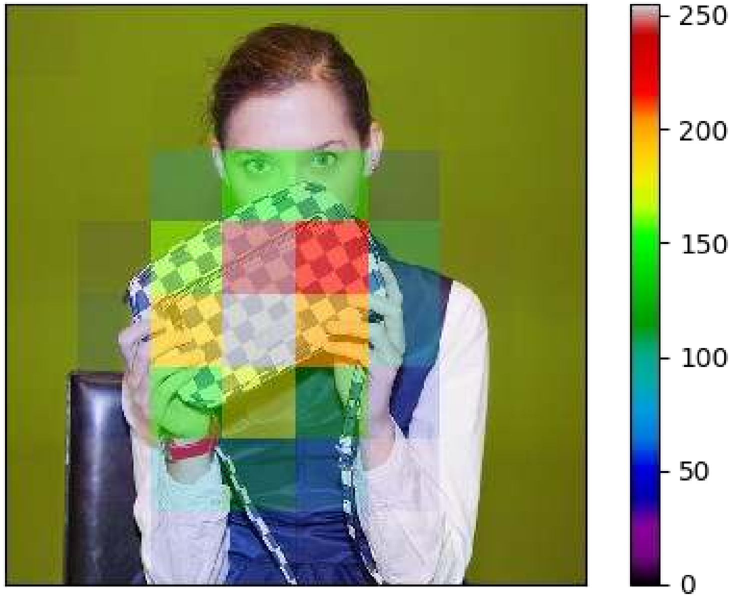
https://github.com/precedenceguo/mx-rcnn

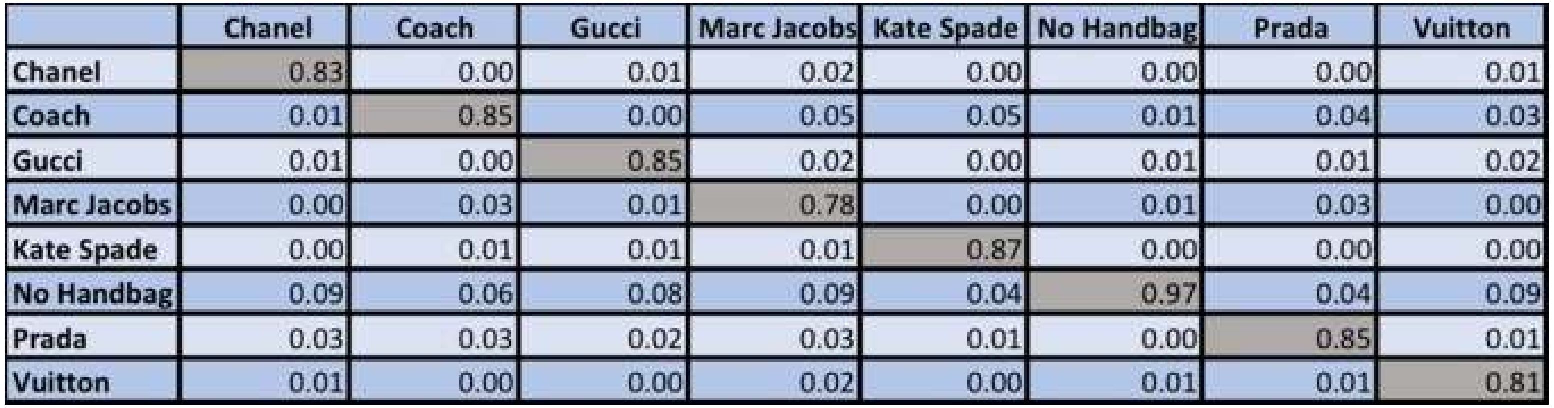


https://github.com/zhreshold/mxnet-yolo

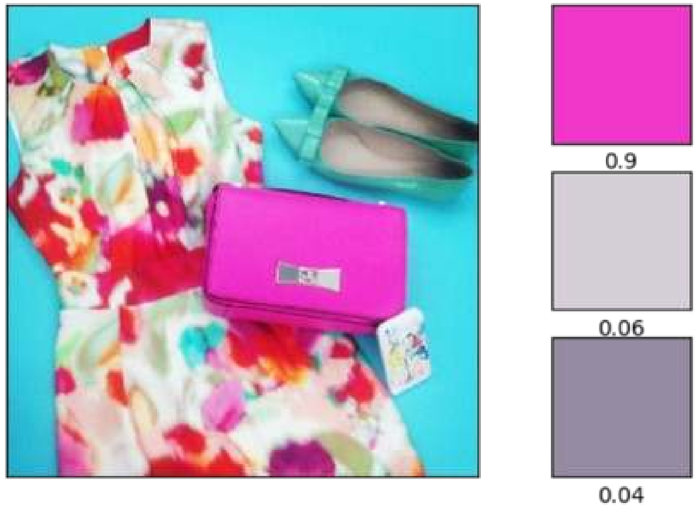
CNN: Object Detection

- 17,000 images from Instagram
- 7 brands
- Deep Learning model pre-trained on ImageNet
- Fine-tuning with TensorFlow and EC2 GPU instances
- Additional work on color extraction



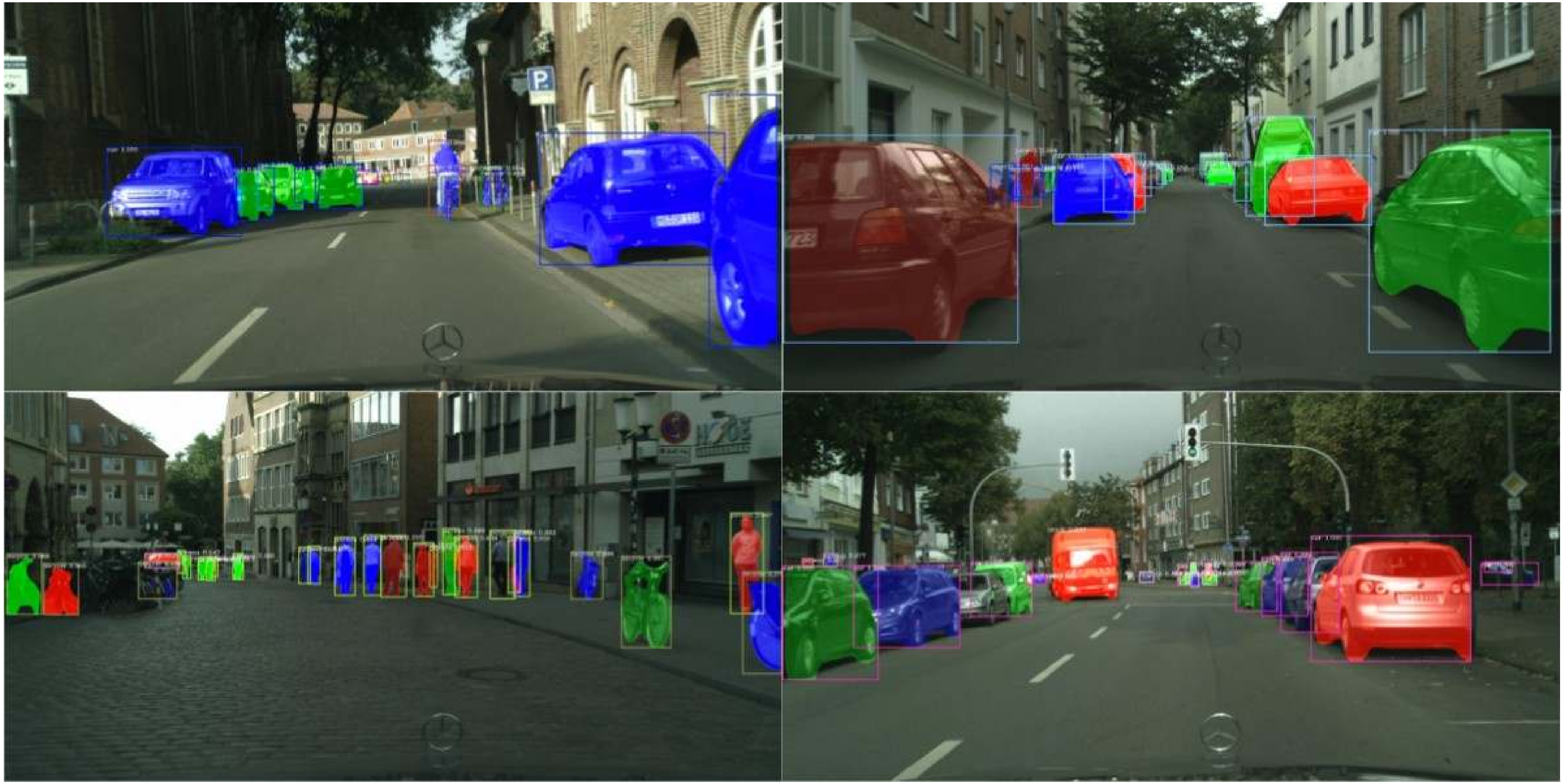


	Chanel	Coach	Gucci	Marc Jacobs	Kate Spade	No Handbag	Prada	Vuitton
Chanel	0.83	0.00	0.01	0.02	0.00	0.00	0.00	0.01
Coach	0.01	0.85	0.00	0.05	0.05	0.01	0.04	0.03
Gucci	0.01	0.00	0.85	0.02	0.00	0.01	0.01	0.02
Marc Jacobs	0.00	0.03	0.01	0.78	0.00	0.01	0.03	0.00
Kate Spade	0.00	0.01	0.01	0.01	0.87	0.00	0.00	0.00
No Handbag	0.09	0.06	0.08	0.09	0.04	0.97	0.04	0.09
Prada	0.03	0.03	0.02	0.03	0.01	0.00	0.85	0.01
Vuitton	0.01	0.00	0.00	0.02	0.00	0.01	0.01	0.81



https://technology.condenast.com/story/handbag-brand-and-color-detection

CNN: Object Segmentation



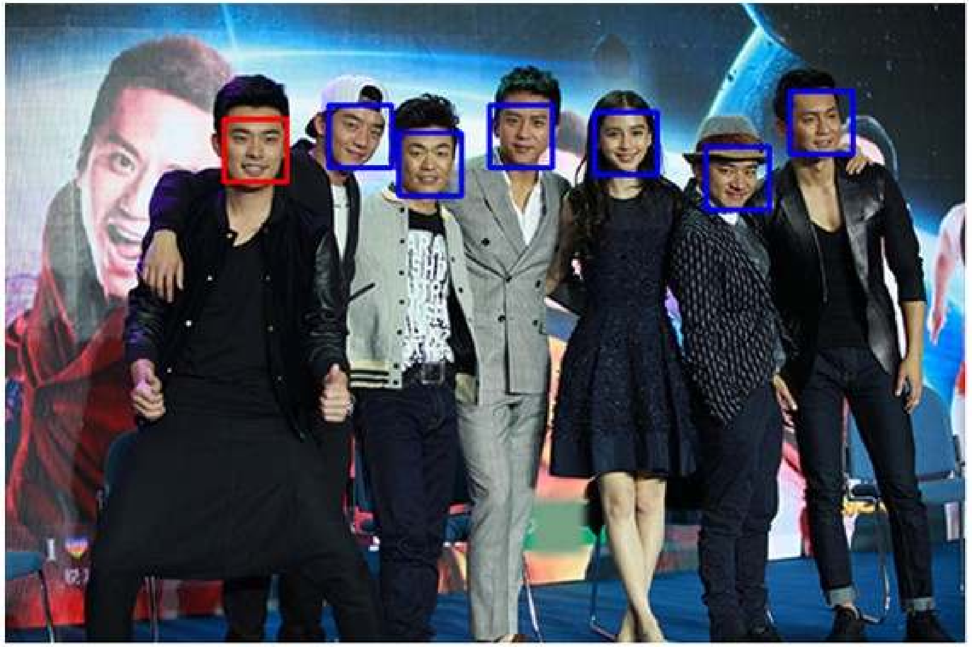
https://github.com/TuSimple/mx-maskrcnn

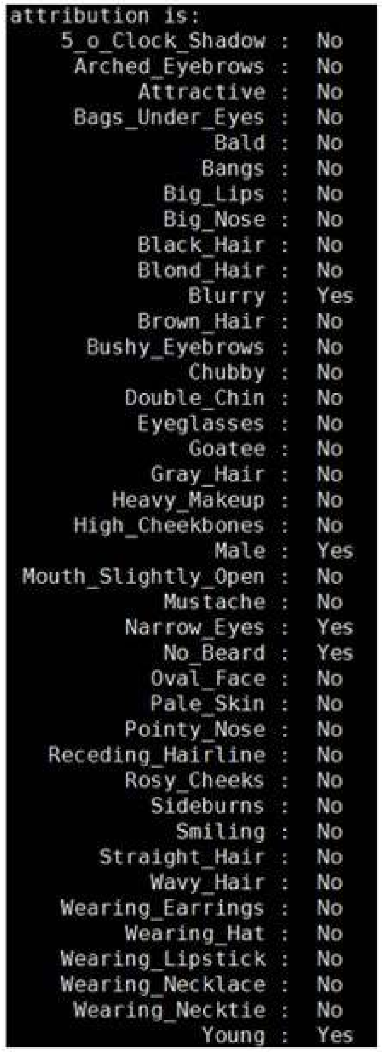
https://www.oreilly.com/ideas/self-driving-trucks-enter-the-fast-lane-using-deep-learn

图森 tu Simple

Last June, tuSimple drove an autonomous truck
for 200 miles from Yuma, AZ to San Diego, CA

CNN: Face Detection

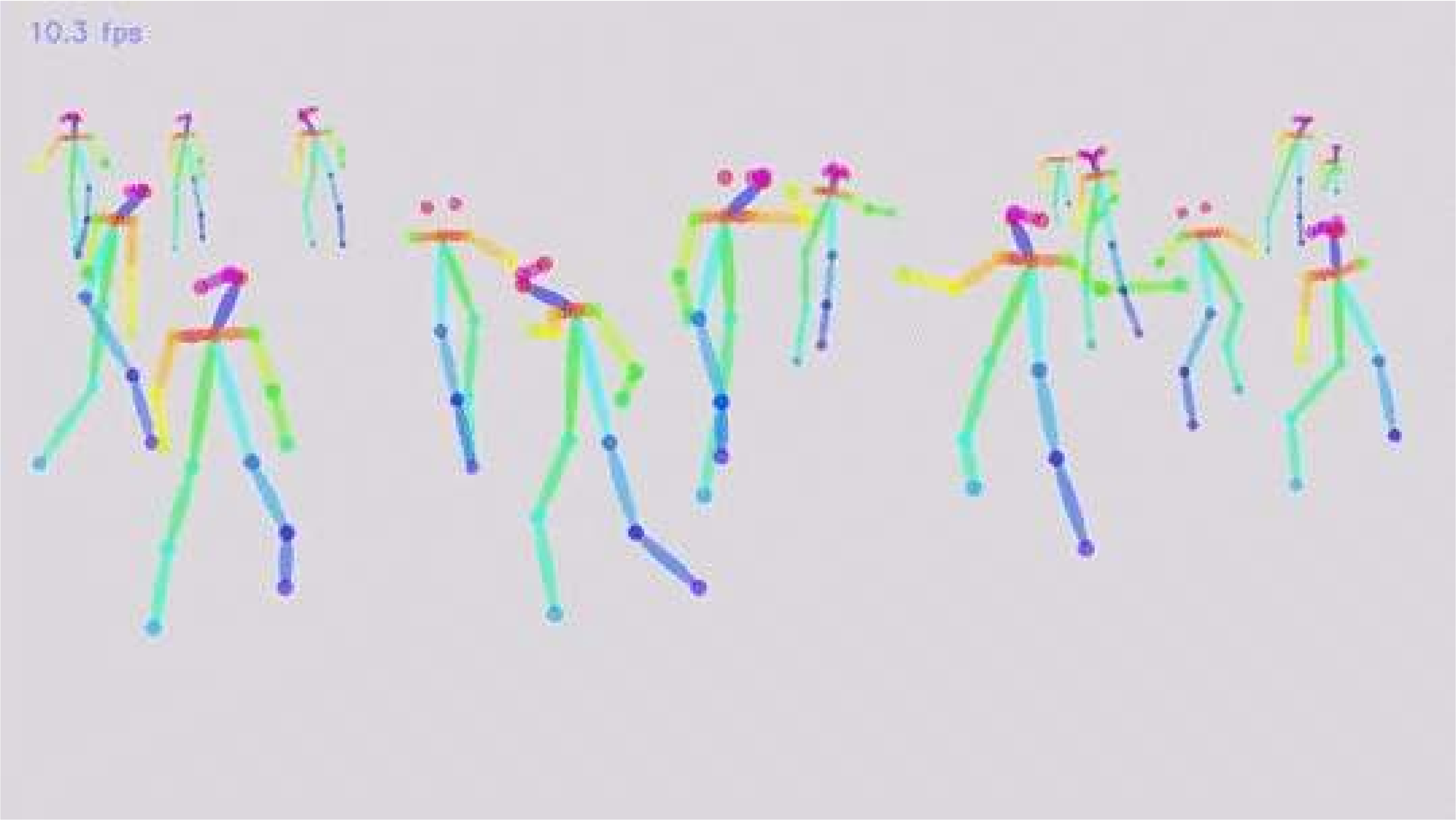




```
attribution is:
       5_o_Clock_Shadow :  No
        Arched_Eyebrows :  No
             Attractive :  No
        Bags_Under_Eyes :  No
                   Bald :  No
                  Bangs :  No
               Big_Lips :  No
               Big_Nose :  No
             Black_Hair :  No
             Blond_Hair :  No
                 Blurry :  Yes
             Brown_Hair :  No
         Bushy_Eyebrows :  No
                 Chubby :  No
            Double_Chin :  No
             Eyeglasses :  No
                 Goatee :  No
              Gray_Hair :  No
           Heavy_Makeup :  No
        High_Cheekbones :  No
                   Male :  Yes
    Mouth_Slightly_Open :  No
               Mustache :  No
            Narrow_Eyes :  Yes
               No_Beard :  Yes
              Oval_Face :  No
              Pale_Skin :  No
            Pointy_Nose :  No
       Receding_Hairline :  No
            Rosy_Cheeks :  No
              Sideburns :  No
                Smiling :  No
          Straight_Hair :  No
              Wavy_Hair :  No
       Wearing_Earrings :  No
            Wearing_Hat :  No
       Wearing_Lipstick :  No
       Wearing_Necklace :  No
        Wearing_Necktie :  No
                  Young :  Yes
```

https://github.com/tornadomeet/mxnet-face

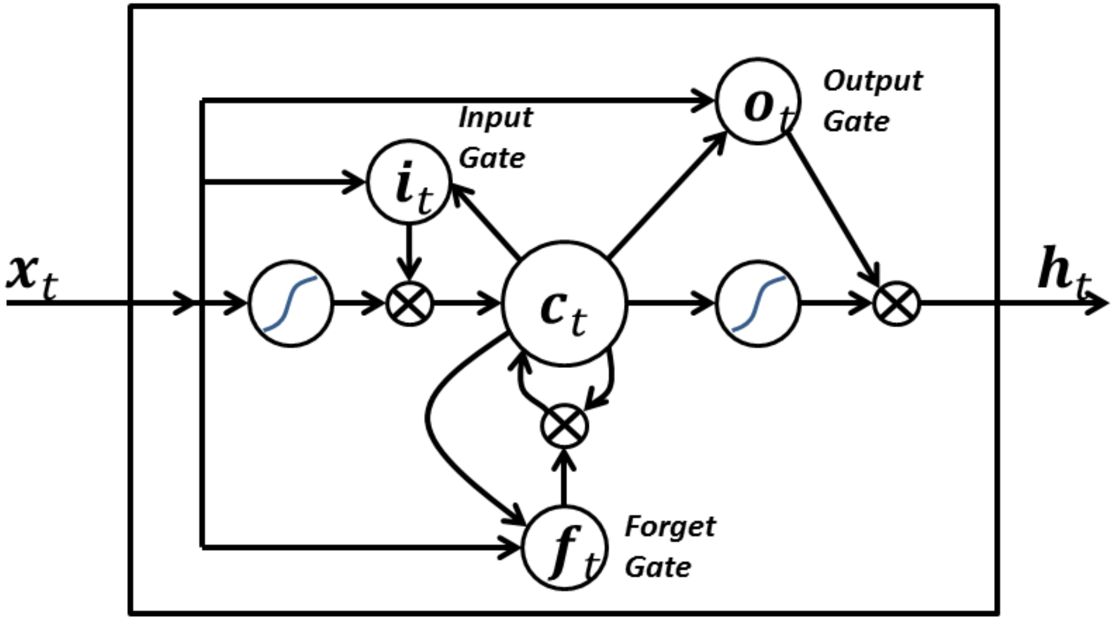
Real-Time Pose Estimation



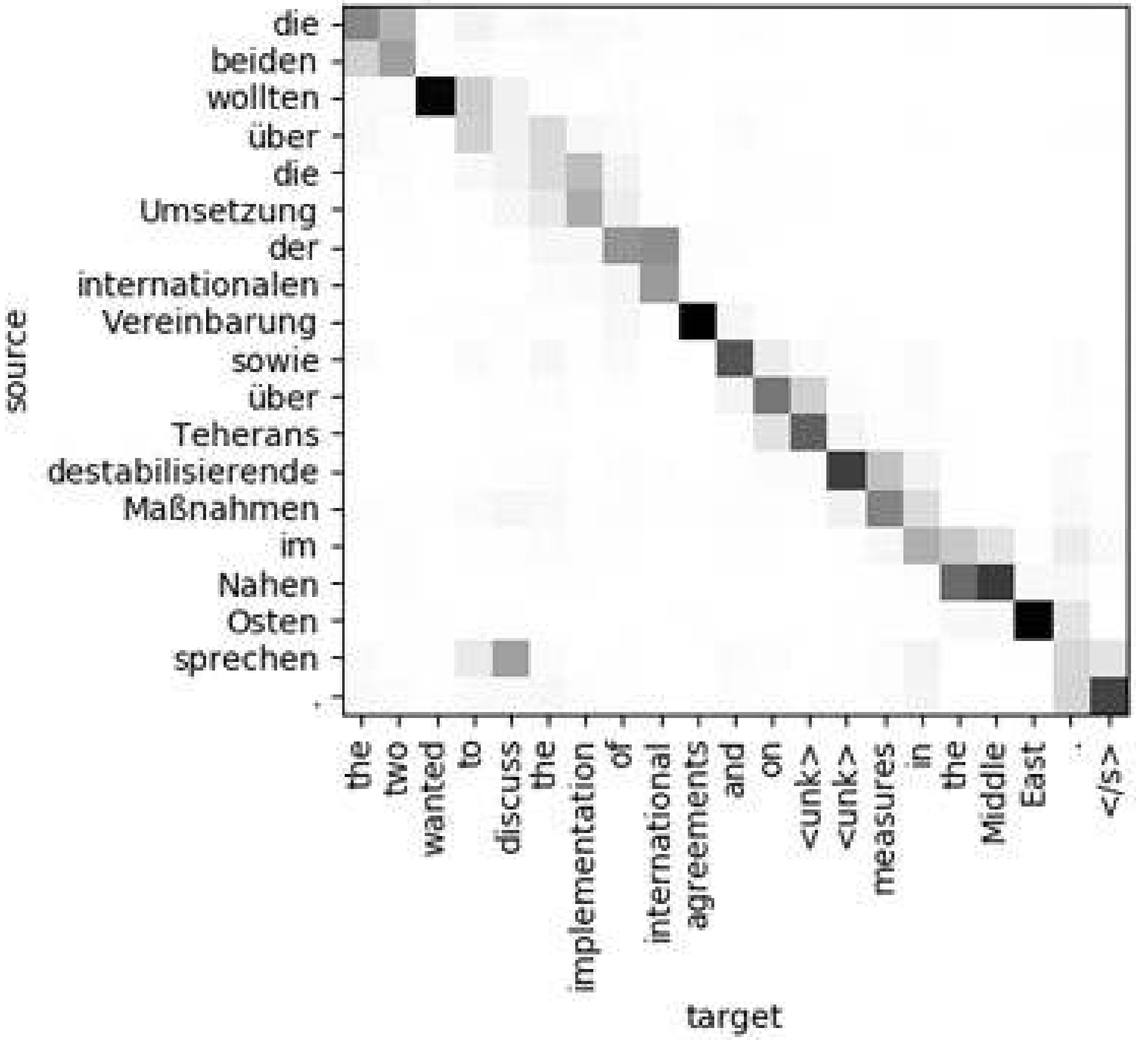
https://github.com/dragonfly90/mxnet_Realtime_Multi-Person_Pose_Estimation

Long Short Term Memory Networks (LSTM)

- A LSTM neuron computes the output based on the input and a previous state
- LSTM networks have memory
- They're great at predicting sequences, e.g. machine translation



LSTM: Machine Translation

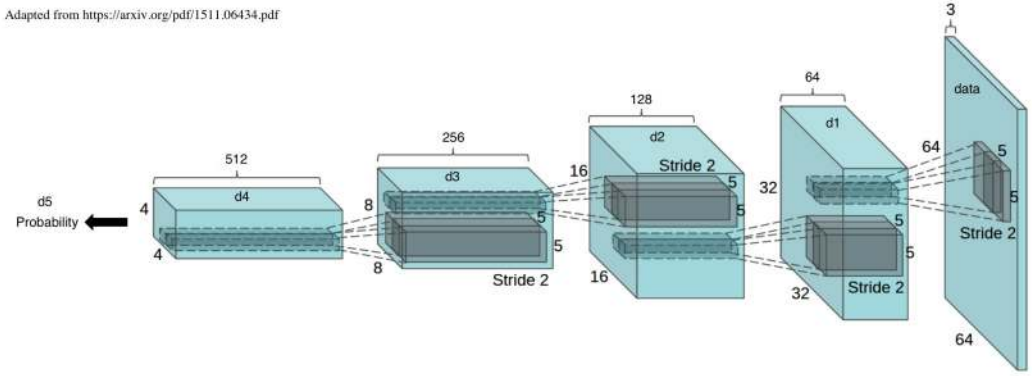


https://github.com/awslabs/sockeye

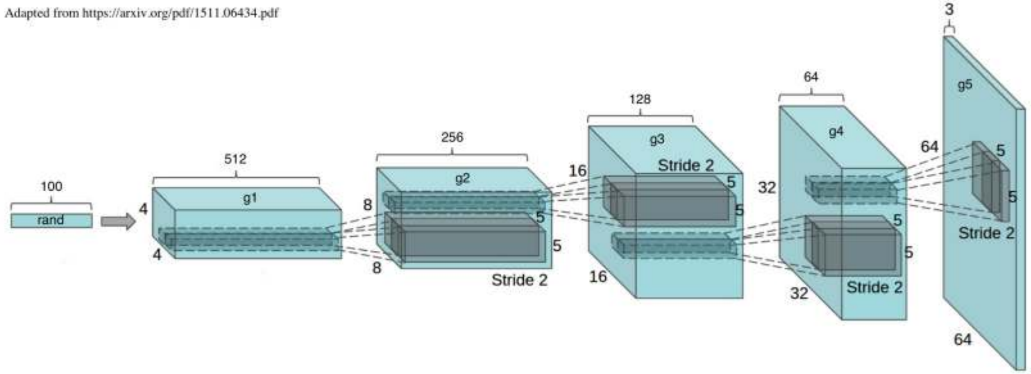
Generative Adversarial Networks (GAN)

- Goodfellow, 2014
- Dual-network architecture

- Detector network
  - Sees real samples
  - Learn how to detect real samples from fake ones created by the Generator network
- Generator network
  - Doesn't see real samples
  - Creates images from random data
  - Applies the same weight updates as the Generator
  - Learns gradually how to generate better fake samples



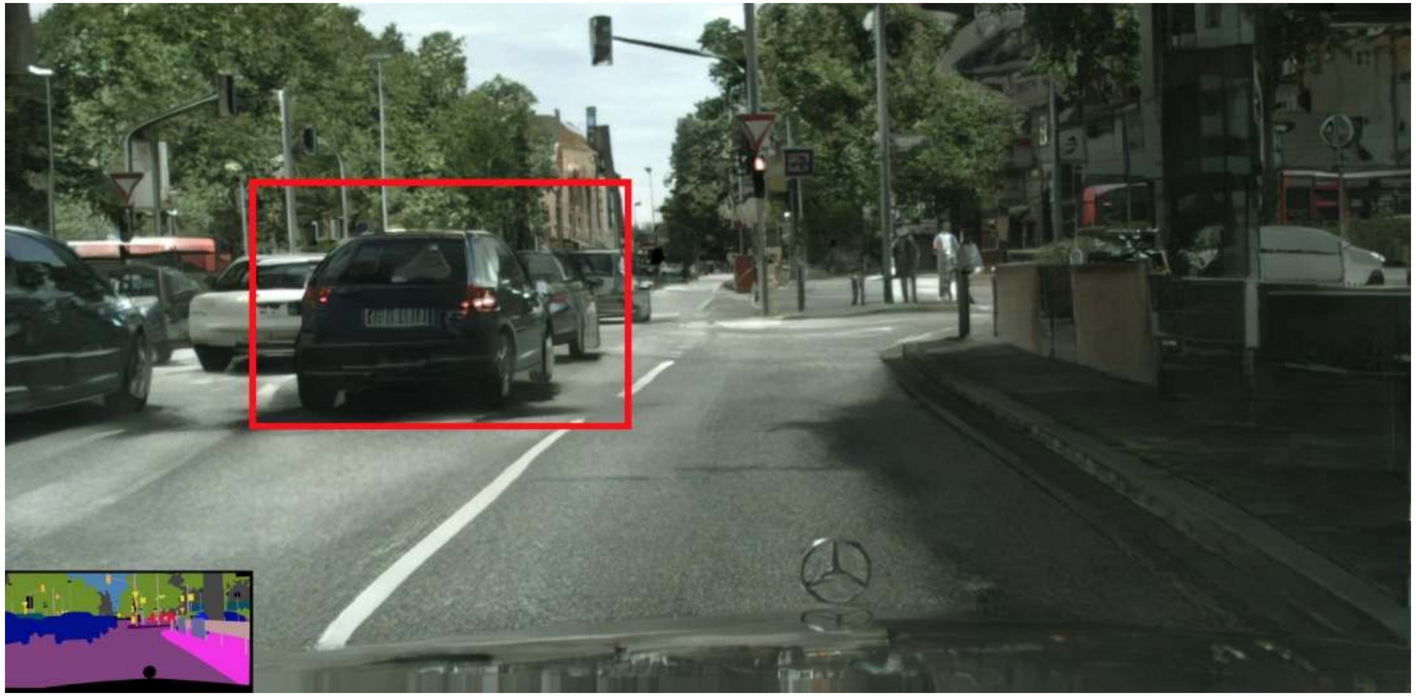
Adapted from https://arxiv.org/pdf/1511.06434.pdf



Adapted from https://arxiv.org/pdf/1511.06434.pdf

GAN: Welcome to the (un)real world, Neo

Generating new ”celebrity” faceshttps://github.com/tkarras/progressive_growing_of_gans



From semantic map to 2048x1024 picture
https://tcwang0509.github.io/pix2pixHD/

Apache MXNet

Apache MXNet: Open Source library for Deep Learning

Programmable

Simple syntax,
multiple
languages

Portable

Highly efficient
models for
mobile
and IoT

High Performance

Near linear scaling
across hundreds of
GPUs

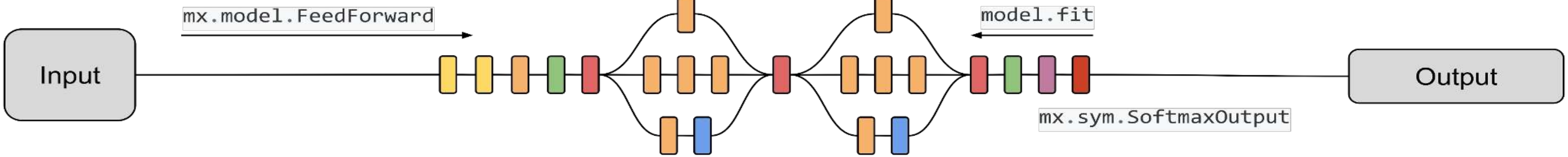
Most Open

Accepted into the
Apache Incubator

Best On AWS

Optimized for
Deep Learning on AWS

NEW! MXNet 1.0 released on December 4th



```
mx.model.FeedForward
```

```
model.fit
```

```
mx.sym.SoftmaxOutput
```

Image

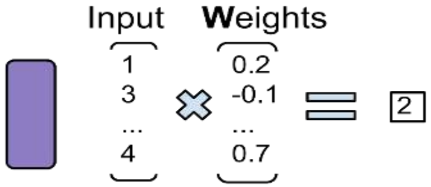
Video

Speech

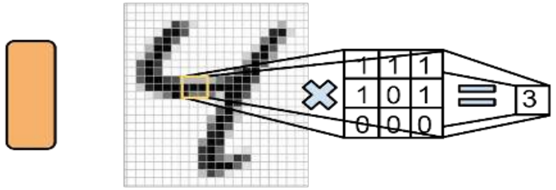
Events

“People Riding Bikes”

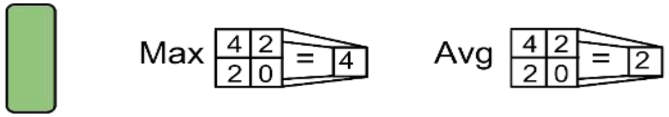
Text



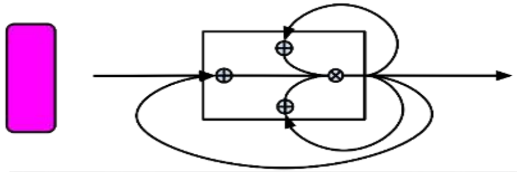
```
mx.sym.FullyConnected(data, num_hidden=128)
```



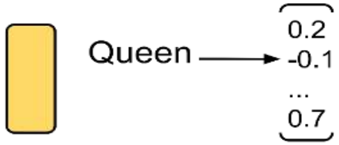
```
mx.sym.Convolution(data, kernel=(5,5), num_filter=20)
```



```
mx.sym.Pooling(data, pool_type="max", kernel=(2,2),
stride=(2,2)
```



```
lstm.lstm_unroll(num_lstm_layer, seq_len, len, num_hidden, num_embed)
```



$$\cos(w, \boldsymbol{queen}) = \cos(w, \boldsymbol{king}) - \cos(w, \boldsymbol{man}) + \cos(w, \boldsymbol{woman})$$

```
mx.symbol.Embedding(data, input_dim, output_dim = k)
```

```
mx.sym.Activation(data, act_type="xxxx")
```

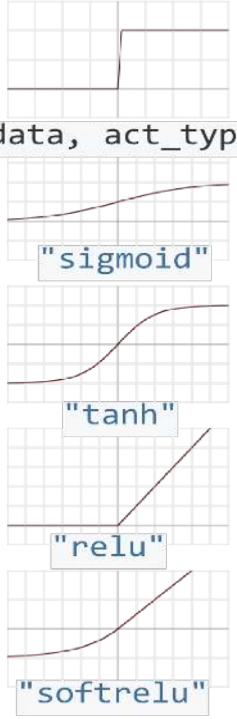


Image Segmentation

Face Search

Neural Art

“People Riding Bikes”

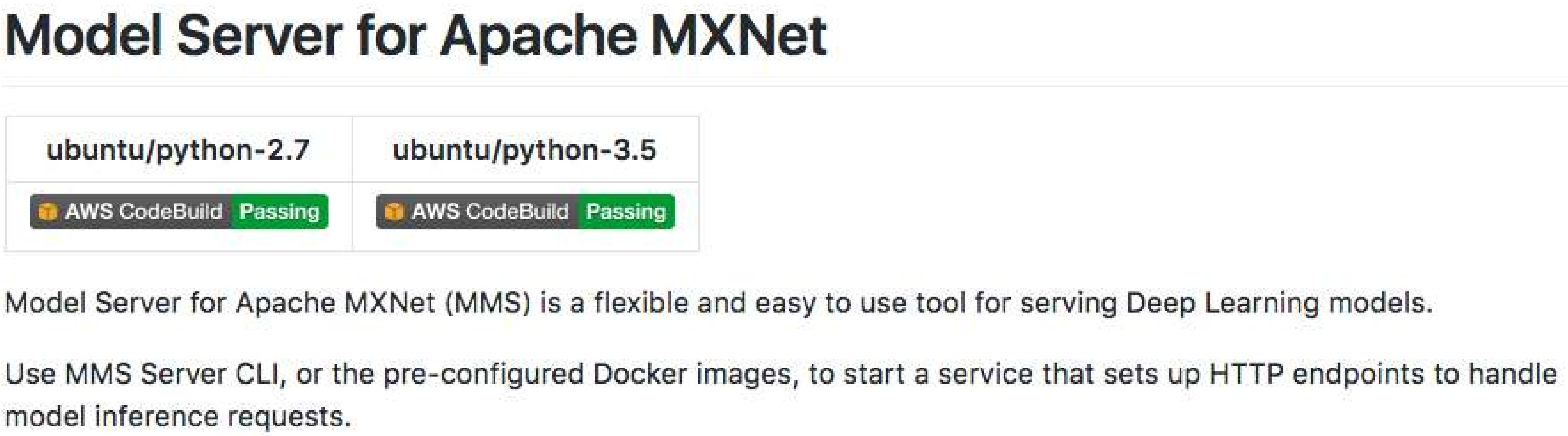
Image Caption

Bicycle, People, Road, Sport

Image Labels

“Οι άνθρωποι ιππασίας ποδήλατα”

Machine Translation



Model Server for Apache MXNet

ubuntu/python-2.7	ubuntu/python-3.5
AWS CodeBuild Passing	AWS CodeBuild Passing

Model Server for Apache MXNet (MMS) is a flexible and easy to use tool for serving Deep Learning models.

Use MMS Server CLI, or the pre-configured Docker images, to start a service that sets up HTTP endpoints to handle model inference requests.

https://github.com/awslabs/mxnet-model-server/

ONNX
OPEN NEURAL NETWORK EXCHANGE FORMAT
The new open ecosystem for interchangeable AI models

https://aws.amazon.com/blogs/ai/announcing-onnx-support-for-apache-mxnet/

The Apache MXNet API

- Storing and accessing data in multi-dimensional arrays
  →*NDArray* API

- Building models (layers, weights, activation functions)
  → *Symbol* API

- Serving data during training and validation
  → Iterators

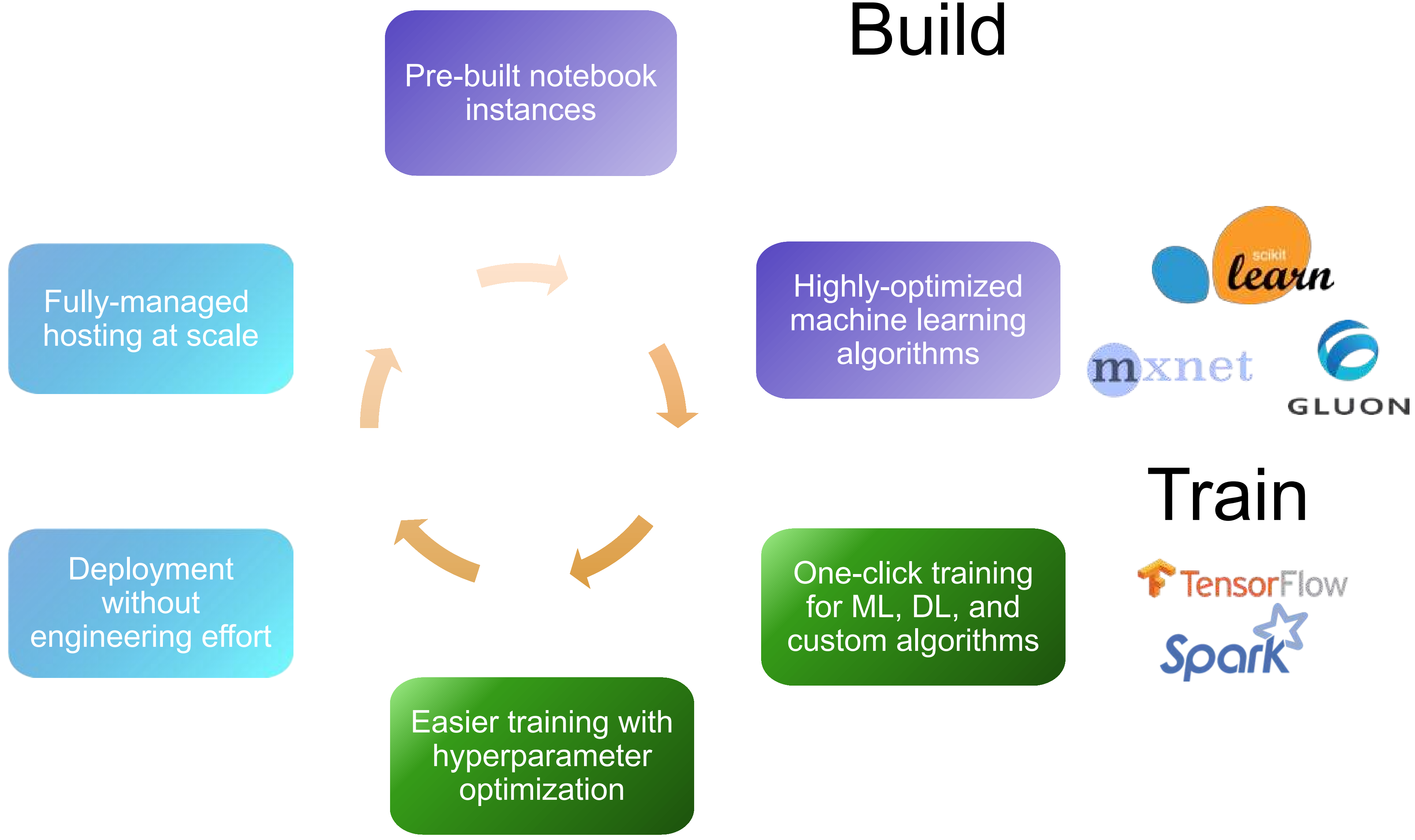
- Training and using models
  → *Module* API

Demos

https://github.com/juliensimon/dlnotebooks

1) Hello World: learn a synthetic data set
2) Classify images with pre-trained models
3) Learn and predict MNIST with a Multi-Layer Perceptron and the LeNet CNN
4) Train, host and deploy a model with Amazon SageMaker
5) Generate MNIST samples with a GAN

Amazon SageMaker



Build

Pre-built notebook instances

Highly-optimized machine learning algorithms

Train

One-click training for ML, DL, and custom algorithms

Easier training with hyperparameter optimization

Deploy

Deployment without engineering effort

Fully-managed hosting at scale

Resources

https://aws.amazon.com/machine-learning

https://aws.amazon.com/sagemaker

https://aws.amazon.com/blogs/ai

https://mxnet.incubator.apache.org

https://github.com/apache/incubator-mxnet

https://github.com/gluon-api

An overview of Amazon SageMaker https://www.youtube.com/watch?v=ym7NEYEx9x4

https://medium.com/@julsimon

Thank you!

Julien Simon, AI Evangelist, EMEA
@julsimon