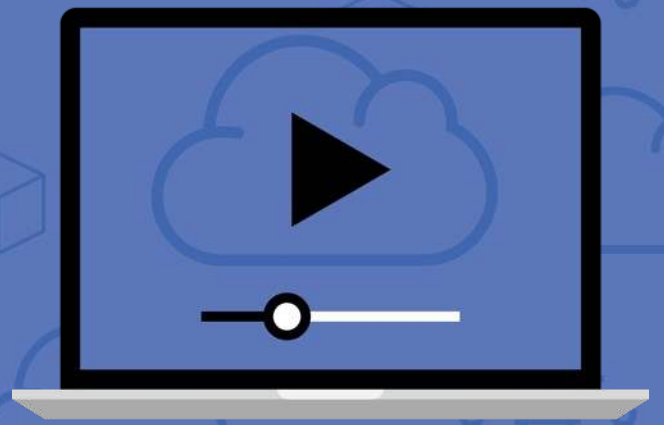
An introduction to Deep Learning



Julien Simon, AI Evangelist, EMEA
@julsimon

What to expect

- Artificial Intelligence, Machine Learning, Deep Learning
- The 5 myths of AI
- Deep Learning in action
- Basics of Deep Learning
- NVIDIA Volta V100 and AWS P3
- Q&A

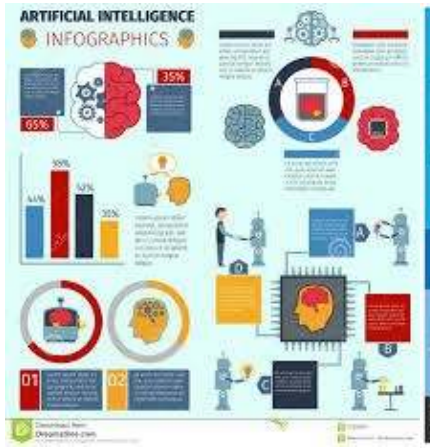
- Artificial Intelligence: design software applications which exhibit human-like behavior, e.g. speech, natural language processing, reasoning or intuition

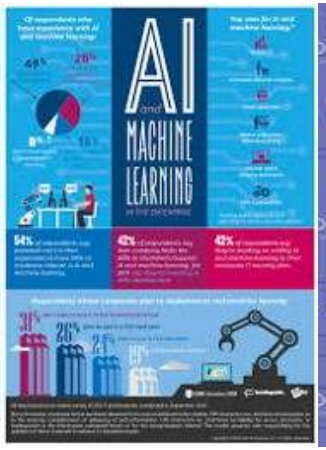
- Machine Learning: teach machines to learn without being explicitly programmed

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

The 5 Myths of AI

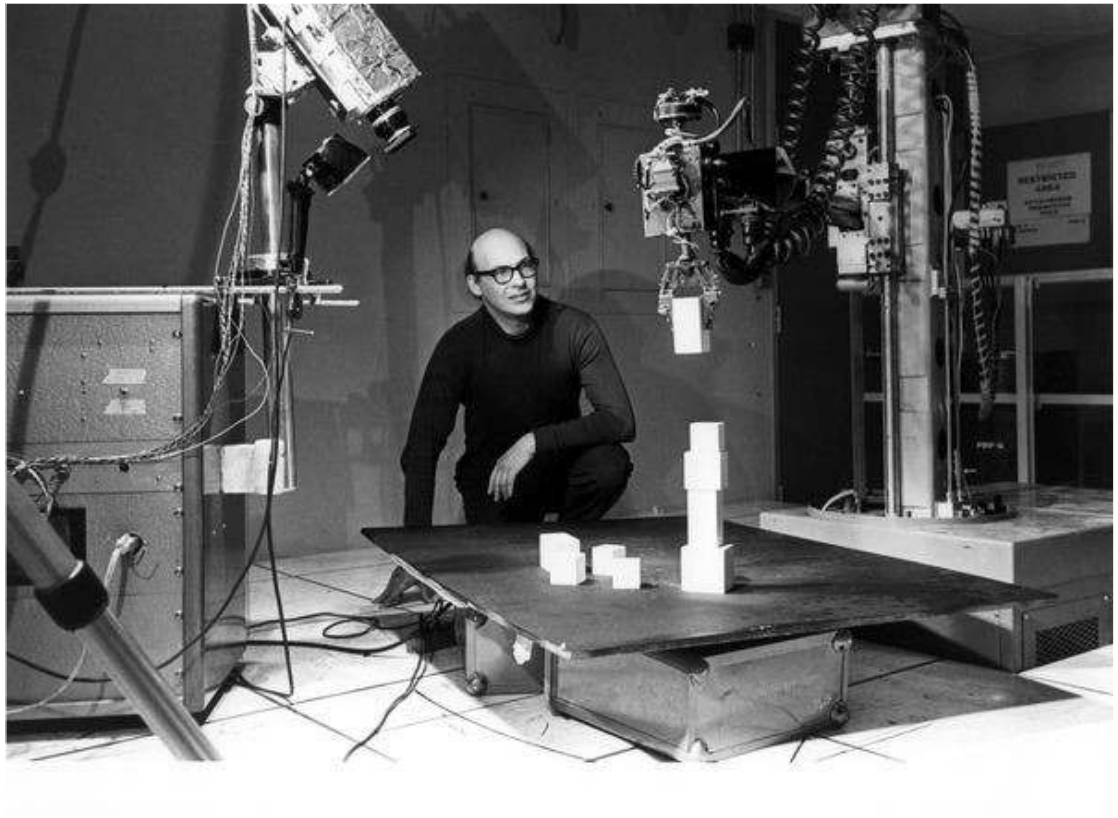
Myth #1 - AI is the flavour of the month





Fact #1 - AI is 60 years old

John McCarthy (1927-2011)
1956 - Coined the term “Artificial Intelligence”
1958 - Invented LISP
1971 - Received the Turing Award



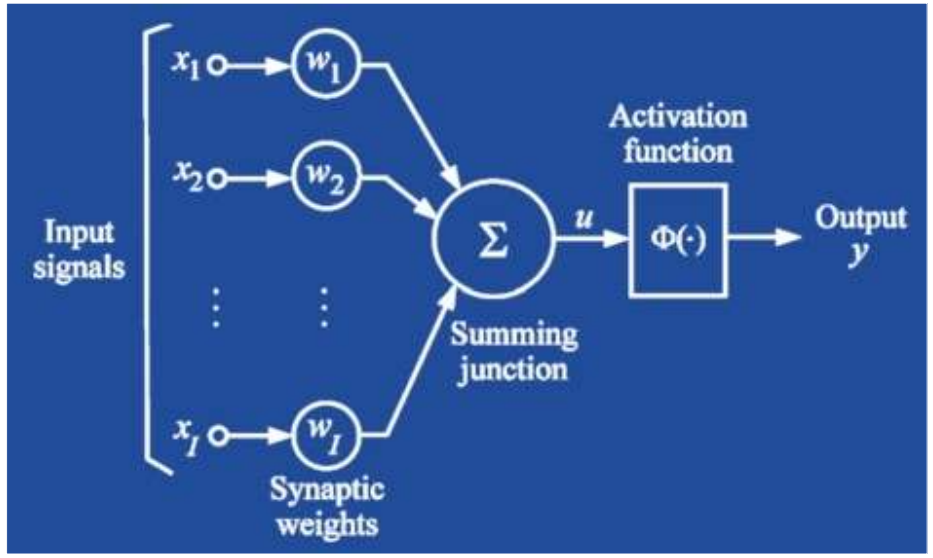
Marvin Minsky (1927-2016)
1959 - Co-founded the MIT AI Lab
1968 - Advised Kubrick on “2001: A Space Odyssey”
1969 - Received the Turing Award

Myth #2 - AI is dark magic

aka « You’re not smart enough »

Fact #2 - AI is math, code and chips

A bit of Science, a lot of Engineering



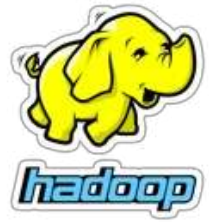

```
data = mx.symbol.Variable('data')
conv1 = mx.sym.Convolution(data=data, kernel=(5,5), num_filter=20)
relu1 = mx.sym.Activation(data=conv1, act_type="relu")
pool1 = mx.sym.Pooling(data=relu1, pool_type="max", kernel=(2,2), stride=(2,2))
conv2 = mx.sym.Convolution(data=pool1, kernel=(5,5), num_filter=50)
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')
```

Myth #3 – The “cognitive” unicorn

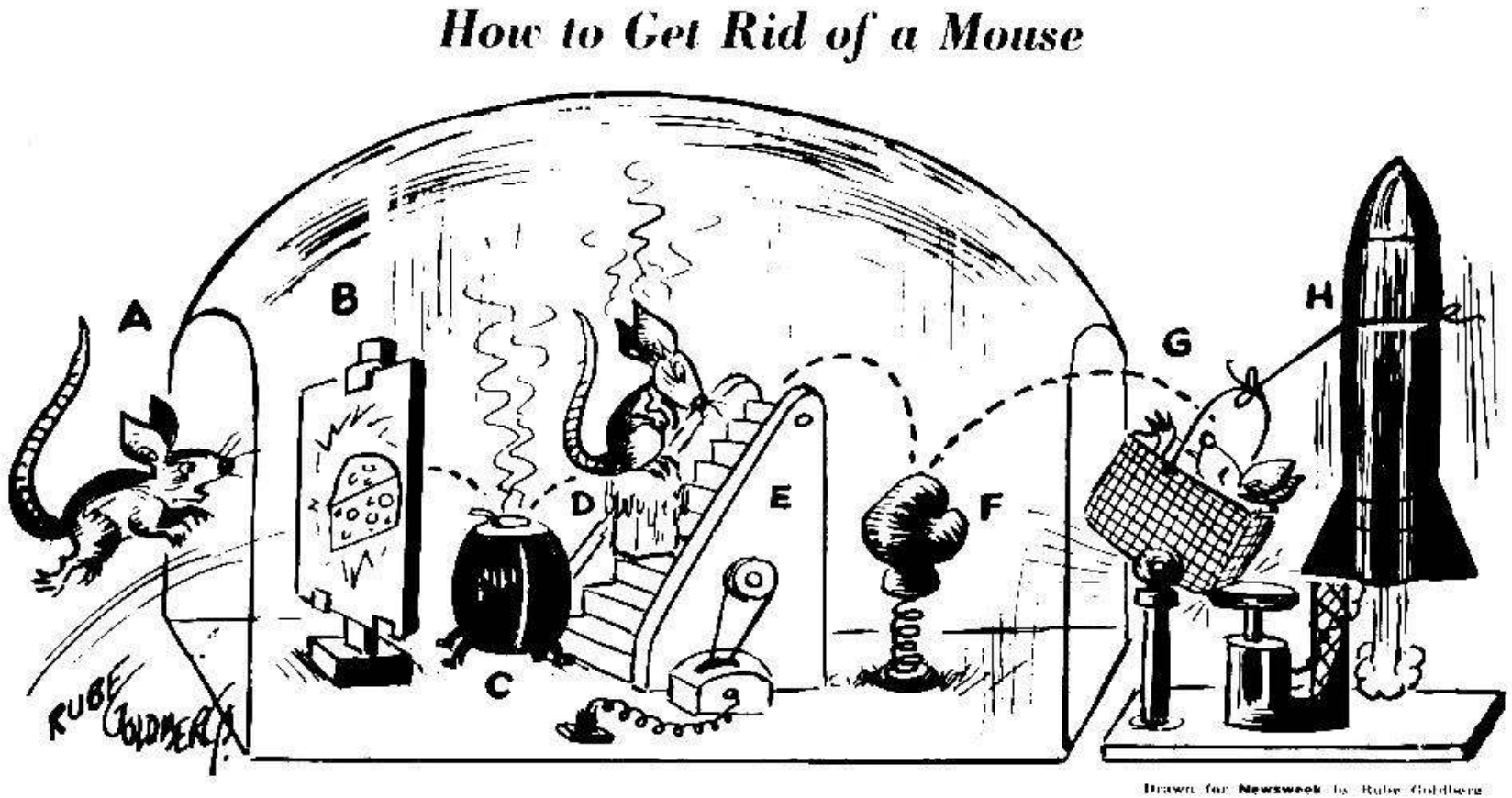
Myth #3 – The “cognitive” unicorn

Fact #3: AI is a wide range of techniques and tools

- Machine Learning
- Natural Language Processing
- Speech
- Vision
- Expert Systems
- And more



Myth #4 - AI is for esoteric use cases



The best mousetrap by Rube Goldberg: Mouse (A) dives for painting of cheese (B), goes through canvas and lands on hot stove (C). He jumps on cake of ice (D) to cool off. Moving escalator (E) drops him on boxing glove (F) which knocks him into basket (G) setting off miniature rocket (H) which takes him to the moon.

Fact #4: AI shines on intuitive problems

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 »

Conversational chatbots

Amazon Lex
Build chatbots to engage customers

Learn more »

Language Services

Amazon Comprehend
Discover insights and relationships in text

Learn more »

Amazon Translate
Fluent translation of text

Learn more »

Amazon Transcribe
Automatic speech recognition

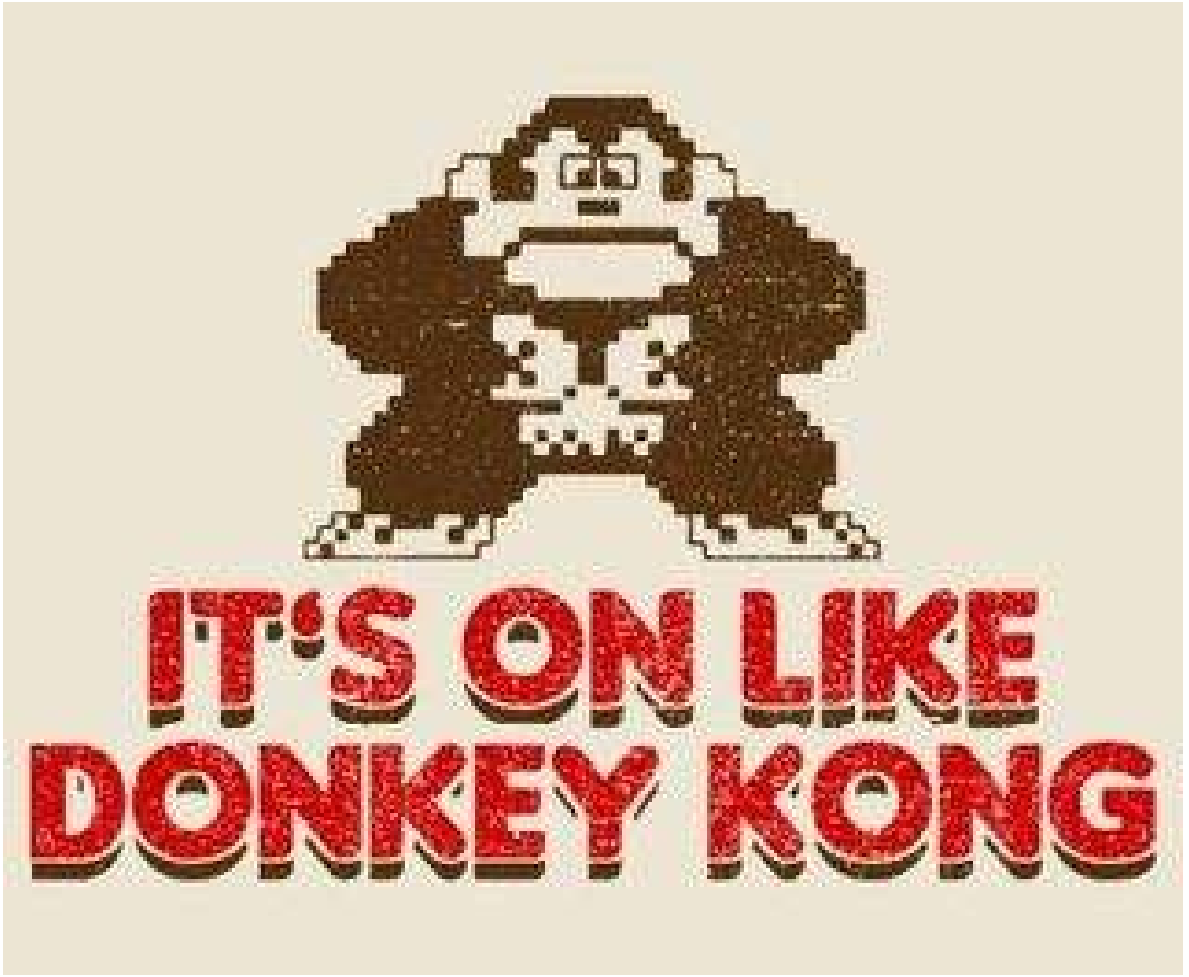
Learn more »

Amazon Polly
Natural sounding text to speech

Learn more »

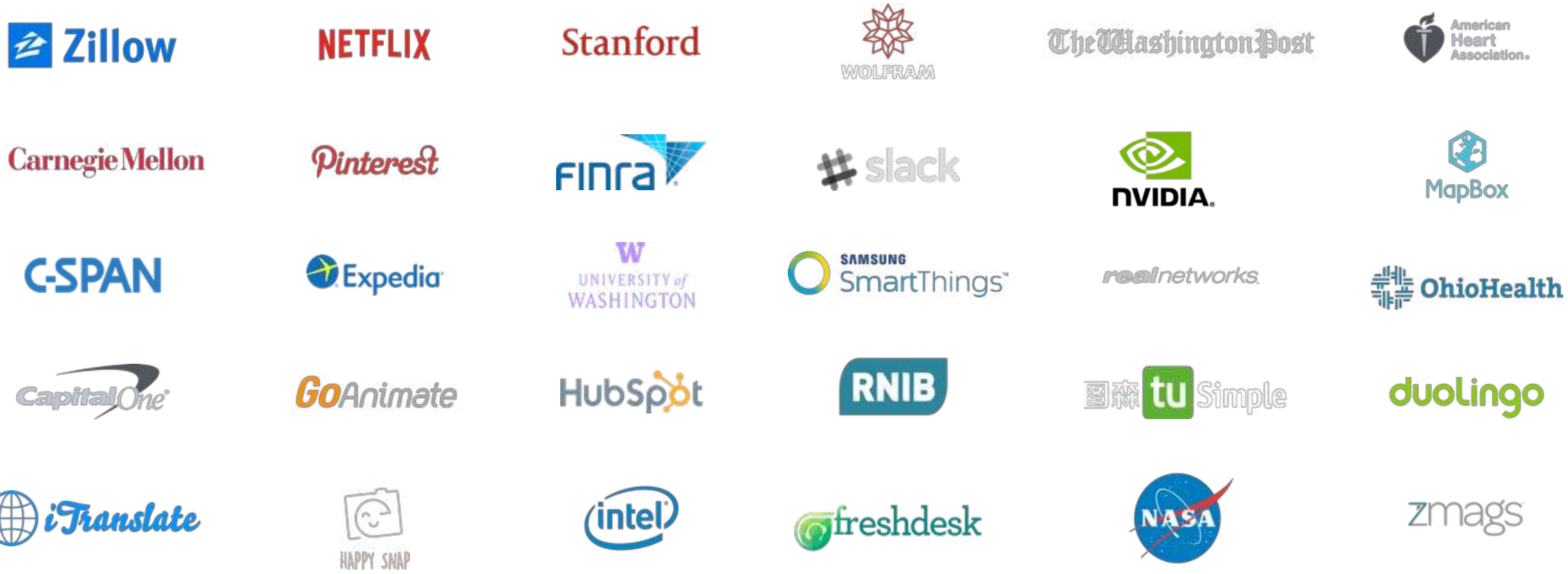
Myth #5 - AI is not production-ready

Fact #5: AI means business



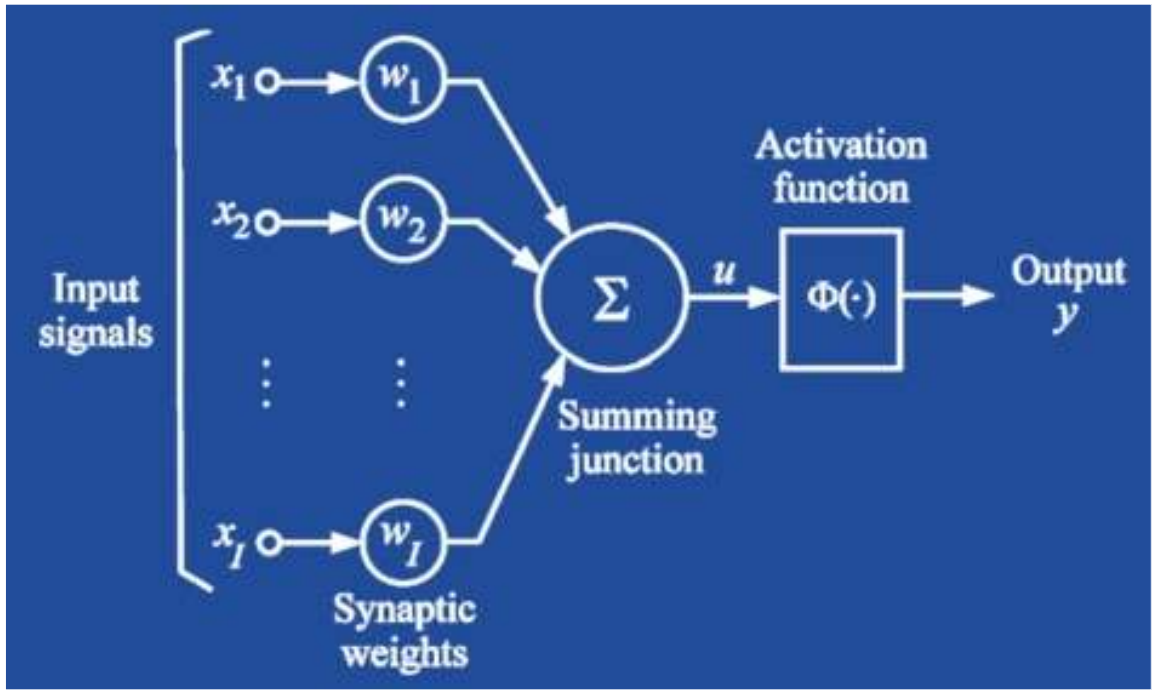
Deep Learning in action

Selected customers running AI on AWS



Basics of Deep Learning

The neuron



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

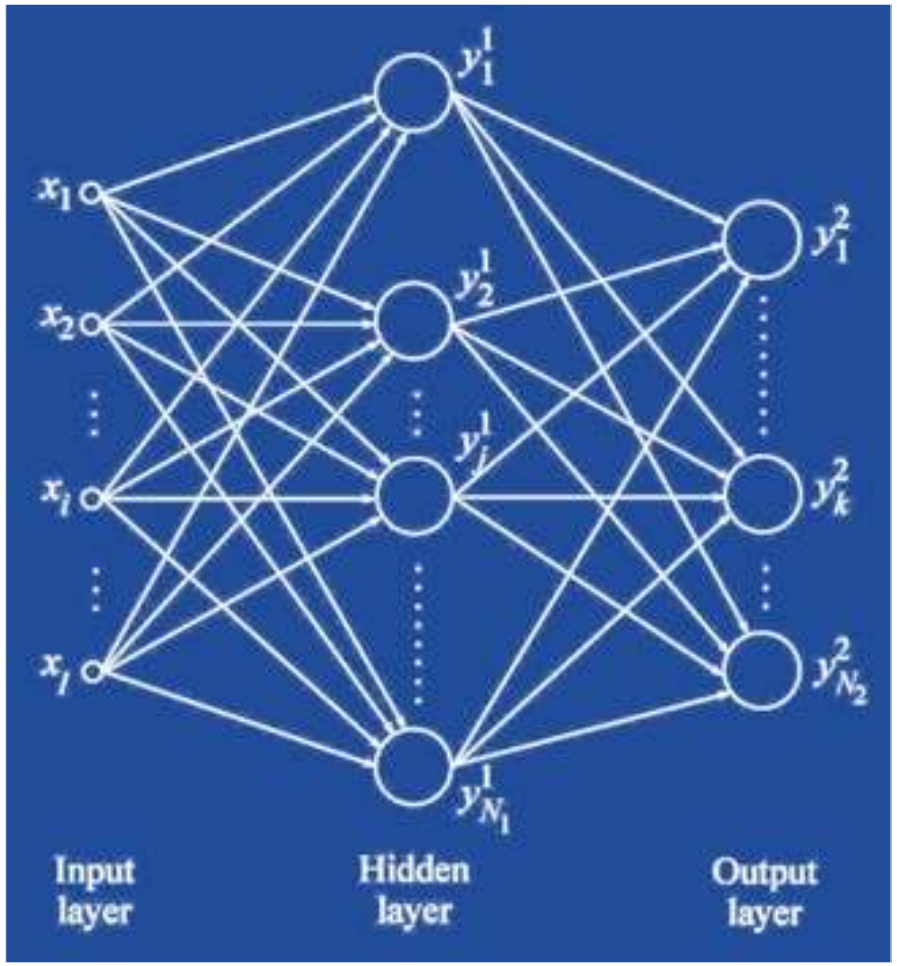
”Multiply and Accumulate”

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



x =

$x_{11}, x_{12}, \ldots x_{1l}$
$x_{21}, x_{22}, \ldots x_{2l}$
… … …
$x_{m1}, x_{m2}, \ldots x_{ml}$

l features

m samples

y =

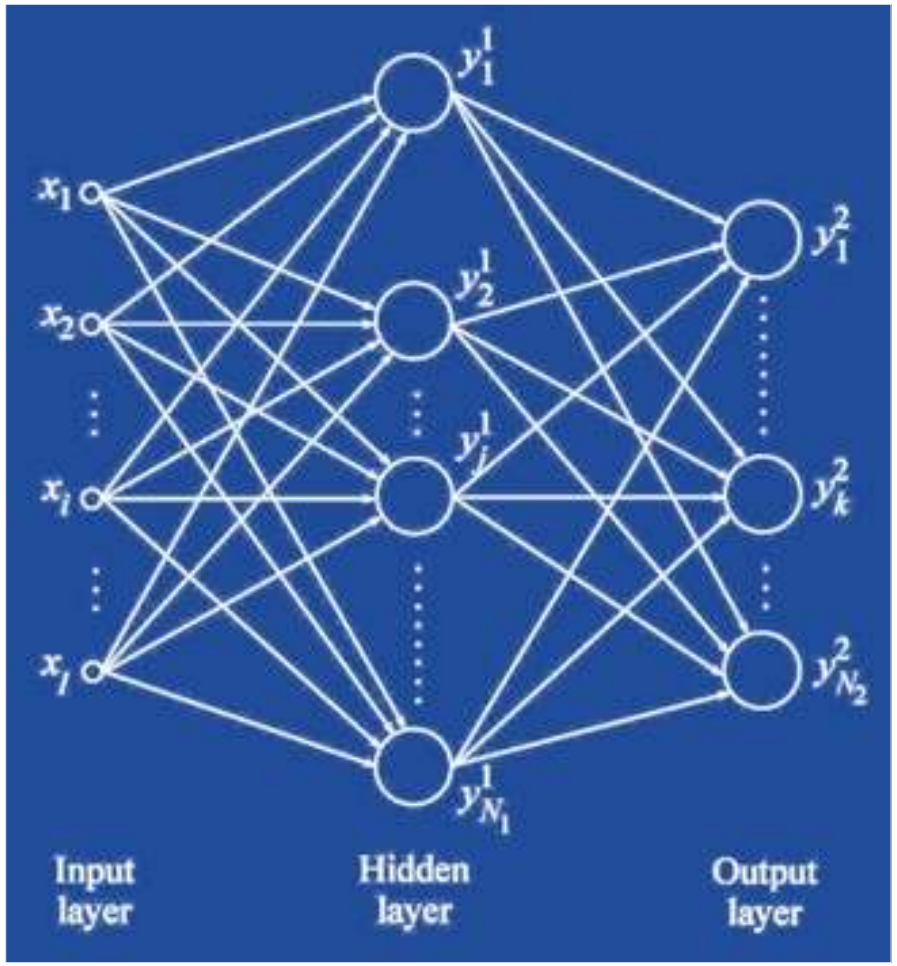
y_1
y_2
…
y_m

0,0,1,0,0,…,0
1,0,0,0,0,…,0
…
0,0,0,0,1,…,0

One-hot encoding

m labels,
N_2 outputs

Neural networks



l features

$$x = \begin{bmatrix} x_{11}, x_{12}, \ldots x_{1l} \\ x_{21}, \ldots x_{2l} \\ \ldots \\ x_{m1}, \ldots \end{bmatrix}$$

m samples

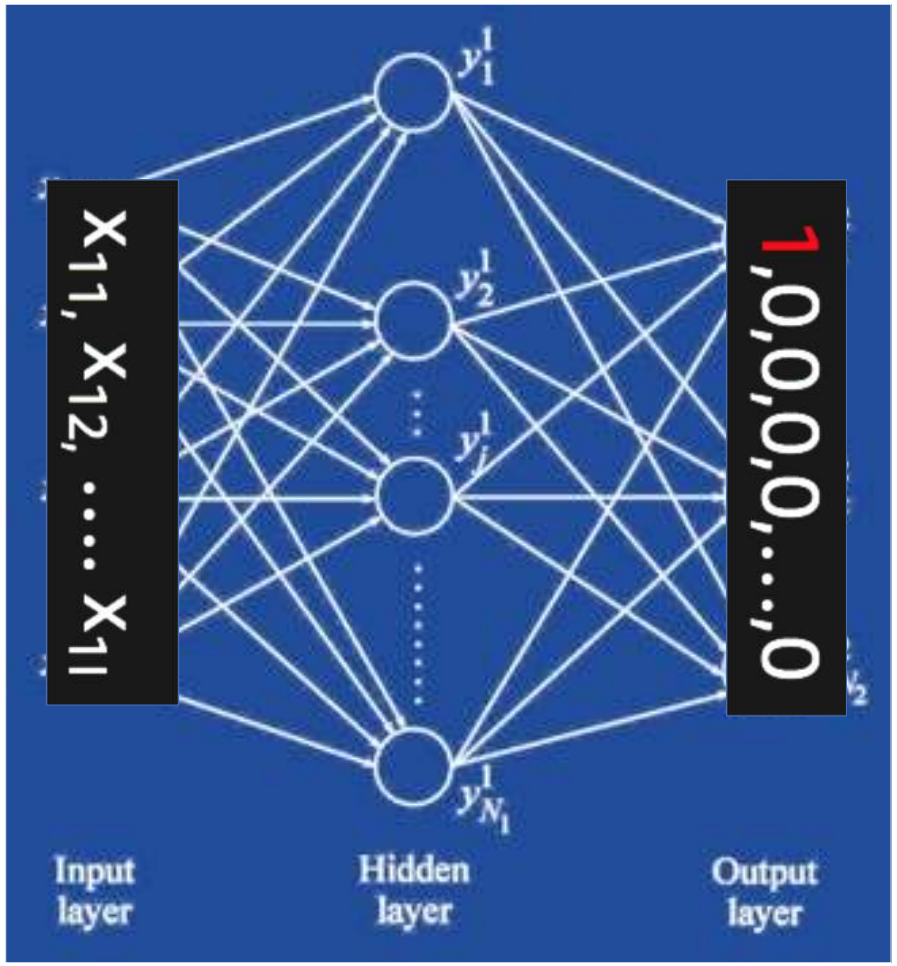
$$y = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_m \end{bmatrix} \quad \begin{bmatrix} 0,0,1,\ldots,0 \\ 1,0,0,\ldots,0 \\ \ldots \\ 0,0,0,\ldots,0 \end{bmatrix}$$

m labels,
N_2 categories

One-hot encoding

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

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

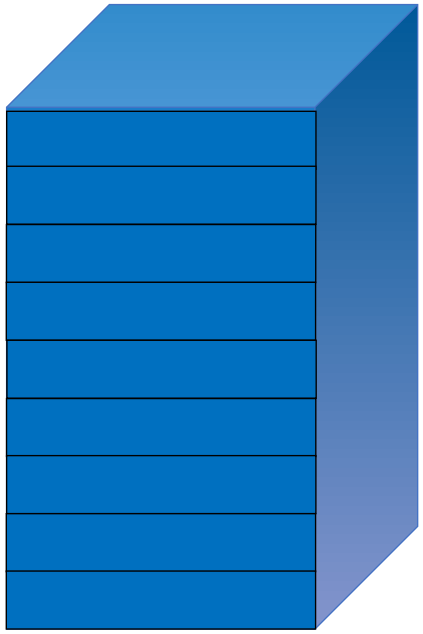
$error = loss(Y_1, Y'_1)$

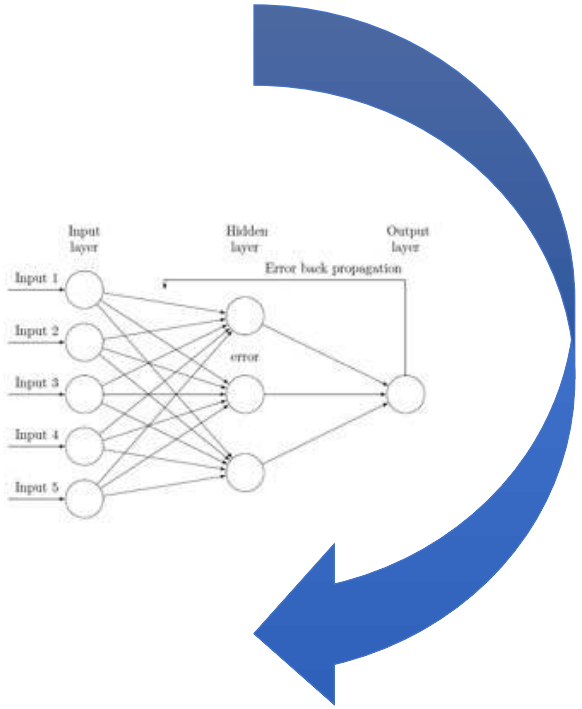
For a batch of samples:

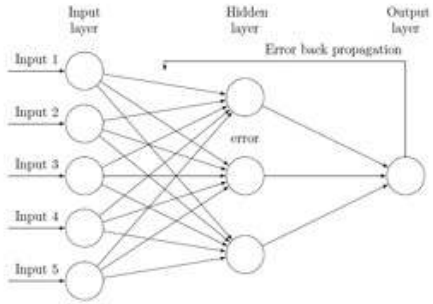
$$\sum_{i=1}^{batch\ size} loss(Y_i, Y'_i) = \text{batch error}$$

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

Training







Trained neural network

Training data set

Backpropagation

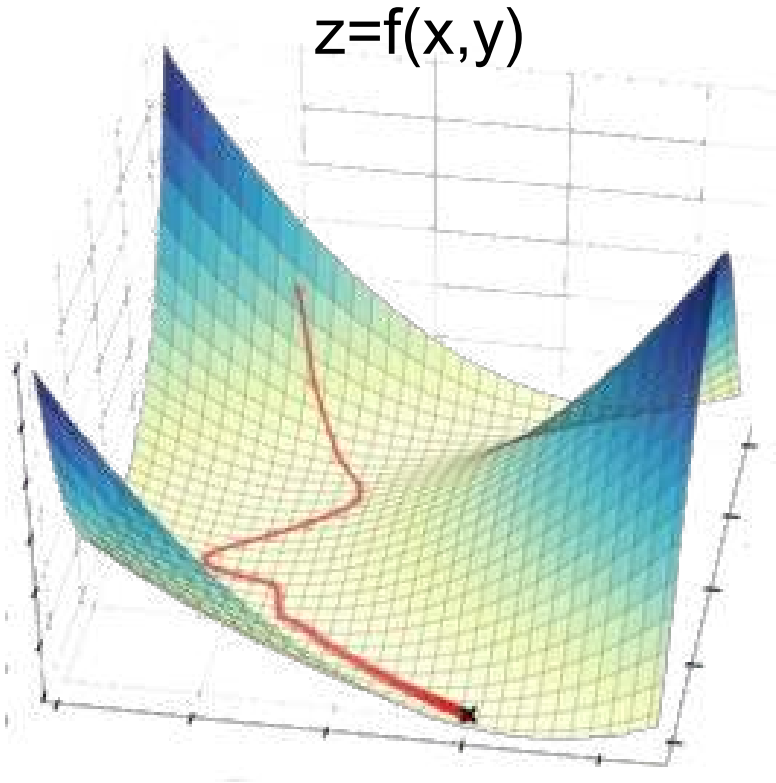
Batch size
Learning rate
Number of epochs

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.

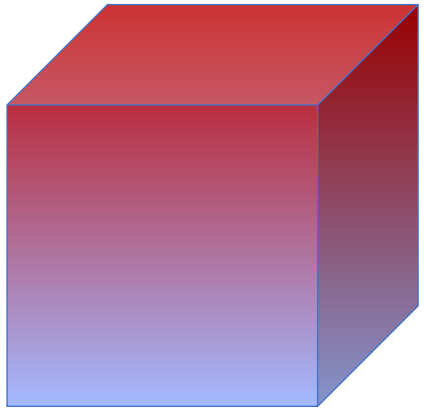
Tim Dettmers
University of Lugano
2015

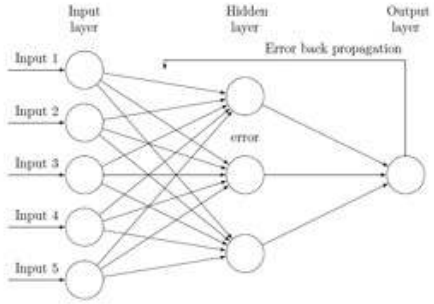


The « step size » is called the learning rate

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

Validation





Validation accuracy

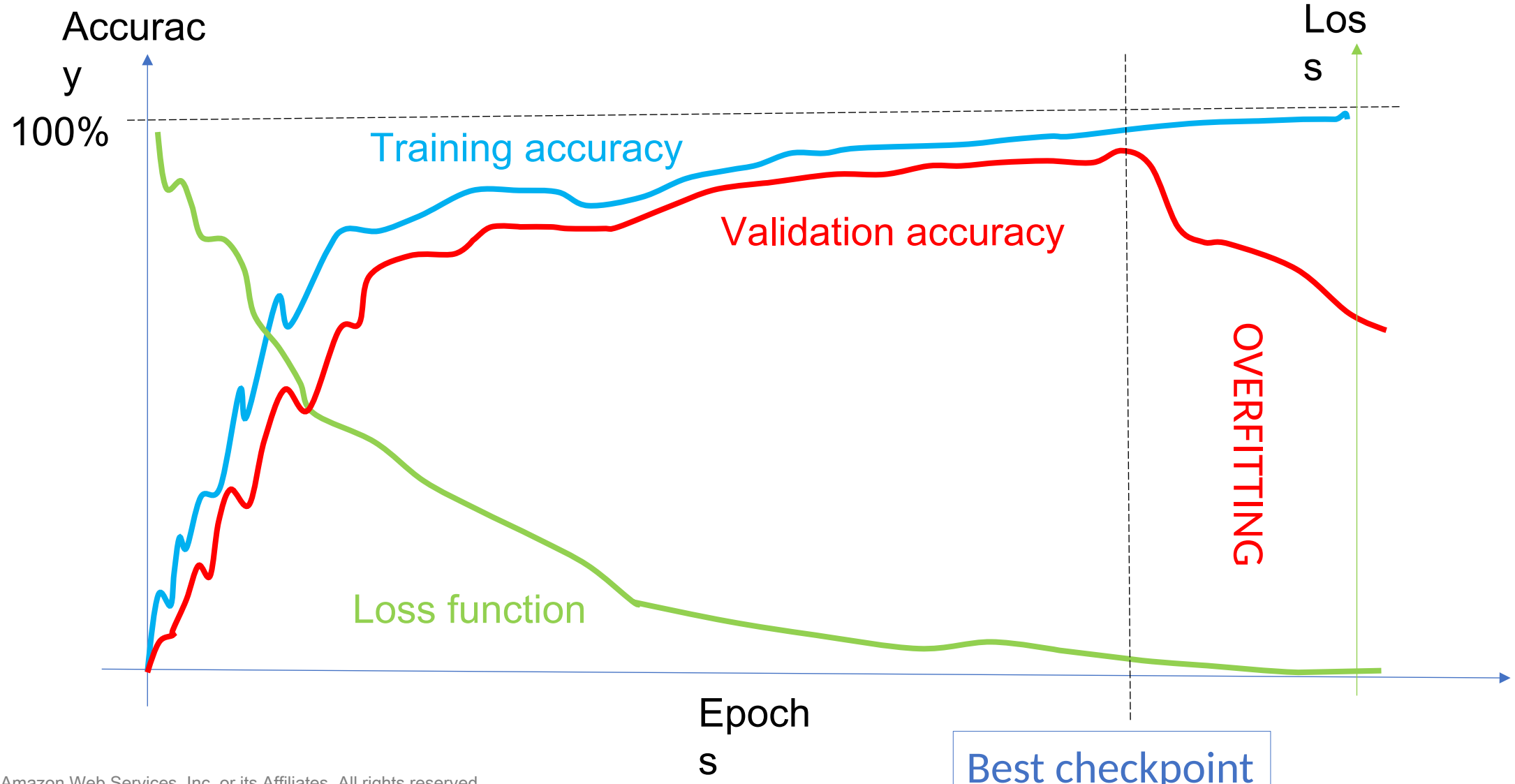
Validation data set

Trained neural network

Prediction at the end of each epoch

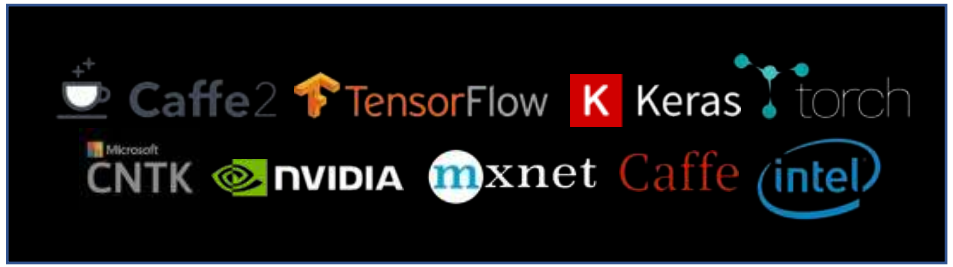
Save the model at the end of each epoch

Early stopping



Deep Learning in practice

- One-click launch
- Single node or distributed
- CPU, GPU, FPGA
- NVIDIA & Intel libraries
- Anaconda Data Science Platform
- Python w/ AI/ML/DL libraries



NVIDIA Volta and AWS P3

Resources

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

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

https://www.nvidia.com/en-us/deep-learning-ai/

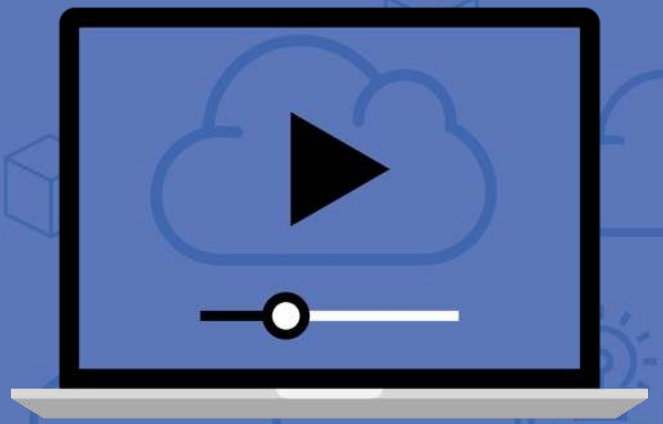
https://www.nvidia.fr/dli

https://www.nvidia.fr/data-center/volta-gpu-architecture/

https://aws.amazon.com/ec2/instance-types/p3/

https://medium.com/@julsimon

Thank you!



Julien Simon, AI Evangelist, EMEA
@julsimon