AI for every developer

Gian Paolo Santopaolo

IBV Solutions - Switzerland

Microsoft Regional Director

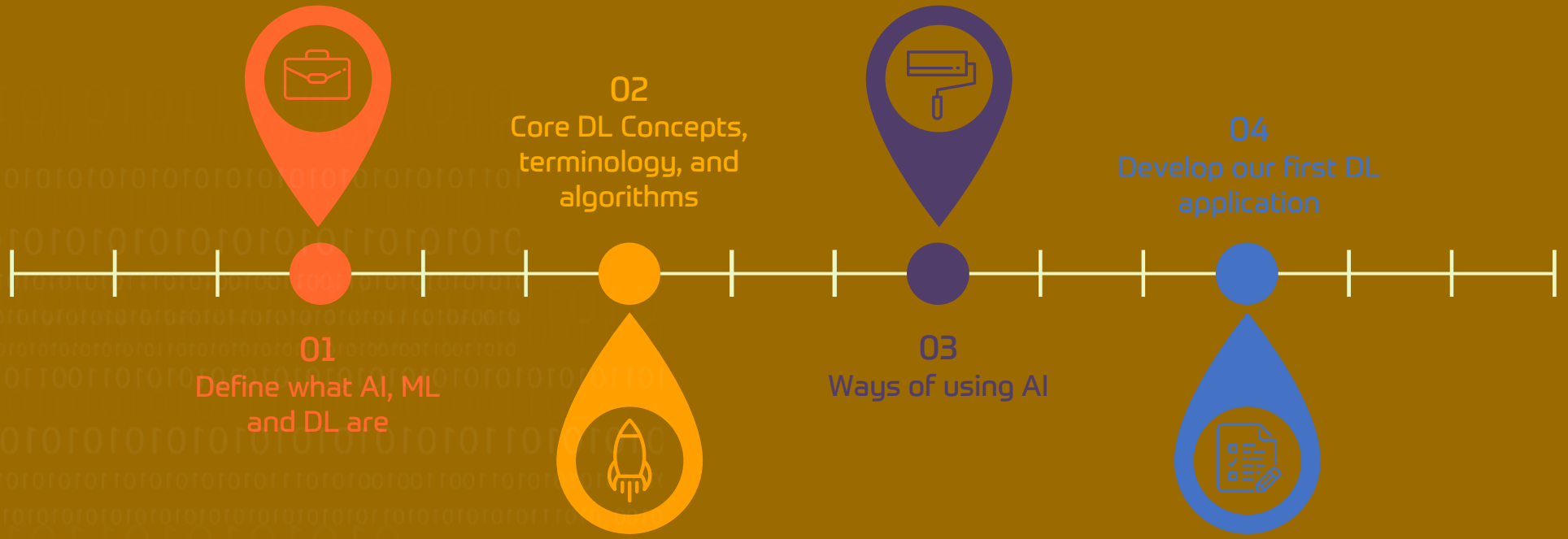
Microsoft Windows Dev MVP

@gsantopaolo

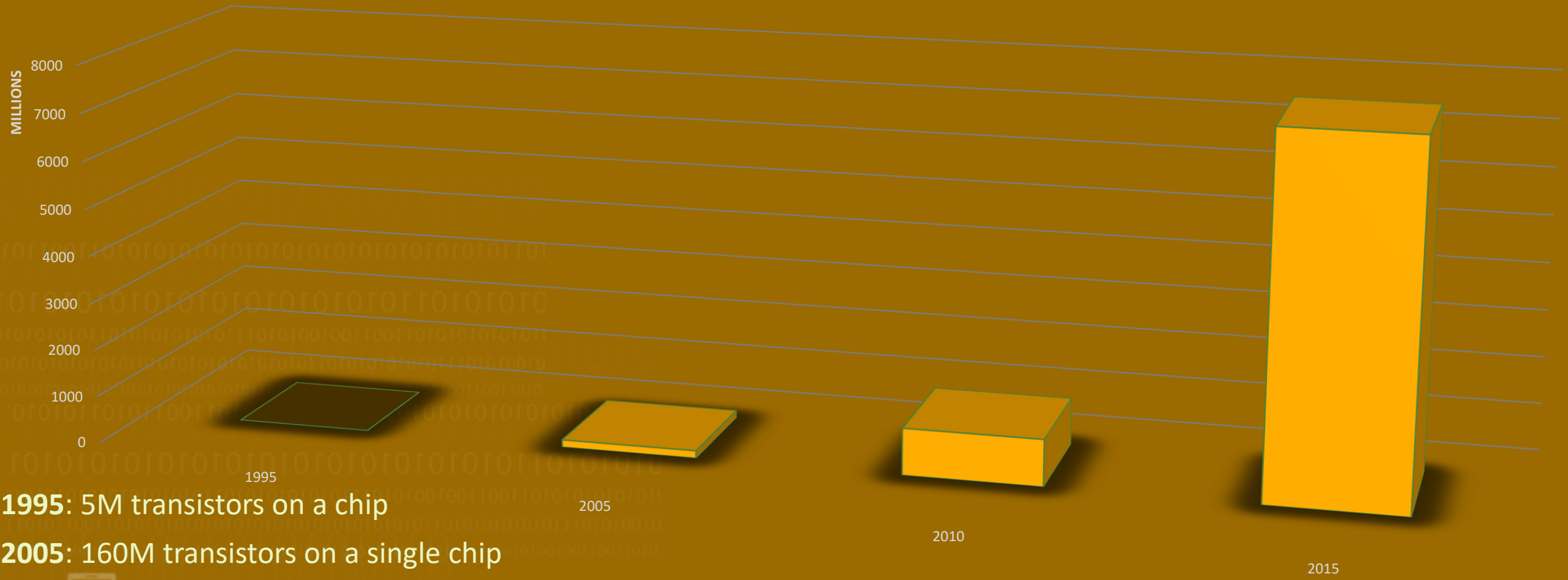
update

Today's Goals

What we are going to learn



A note on the computational power



1995: 5M transistors on a chip

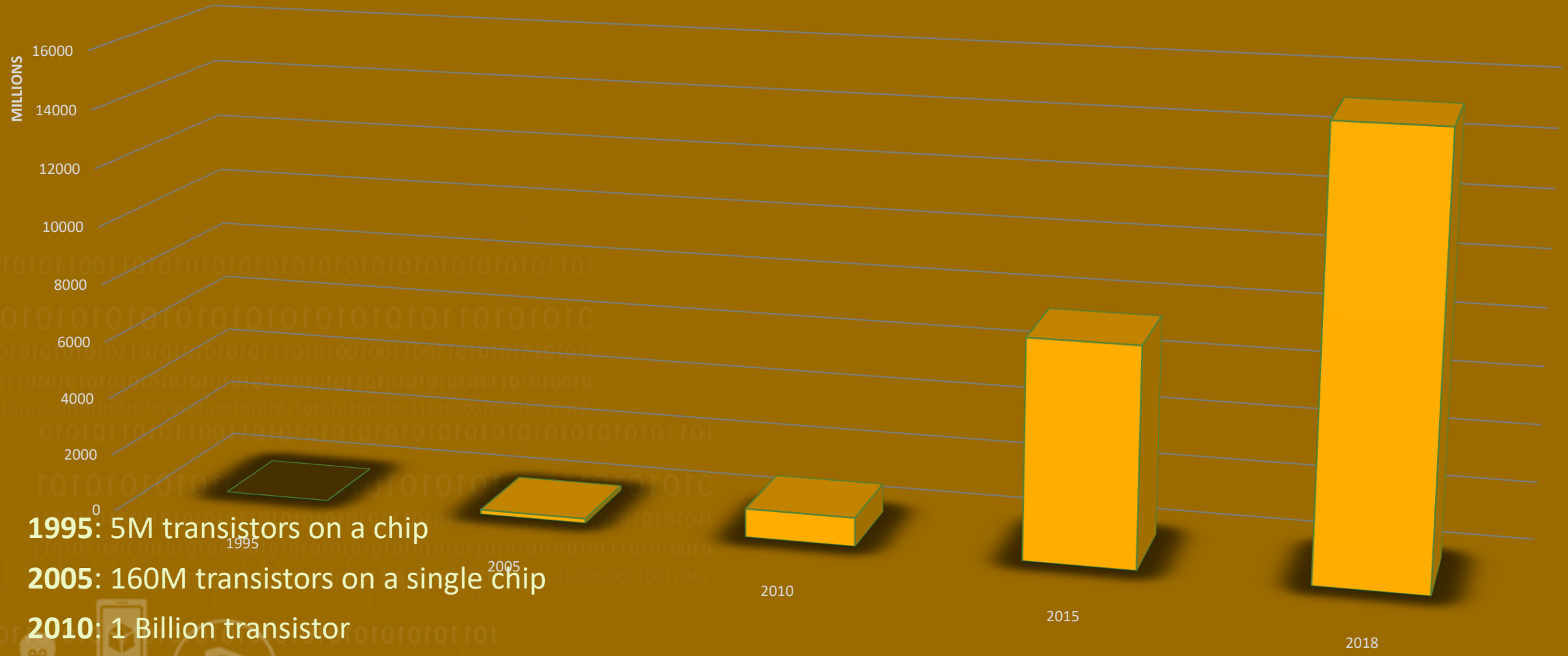
2005: 160M transistors on a single chip

2010: 1 Billion transistor

2015: 7.6 Billions transistors on a chip

2017: 15.2 Billions transistors on a chip

A note on the computational power



1995: 5M transistors on a chip

2005: 160M transistors on a single chip

2010: 1 Billion transistor

2015: 7.6 Billions transistors on a chip

2017: 15.2 Billions transistors on a chip

A note on the computational power

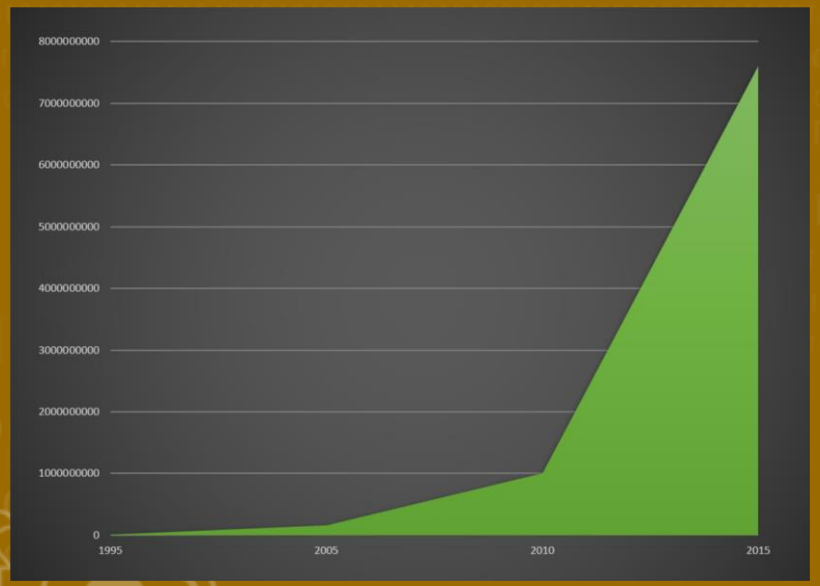
1995: 5M transistors on a chip

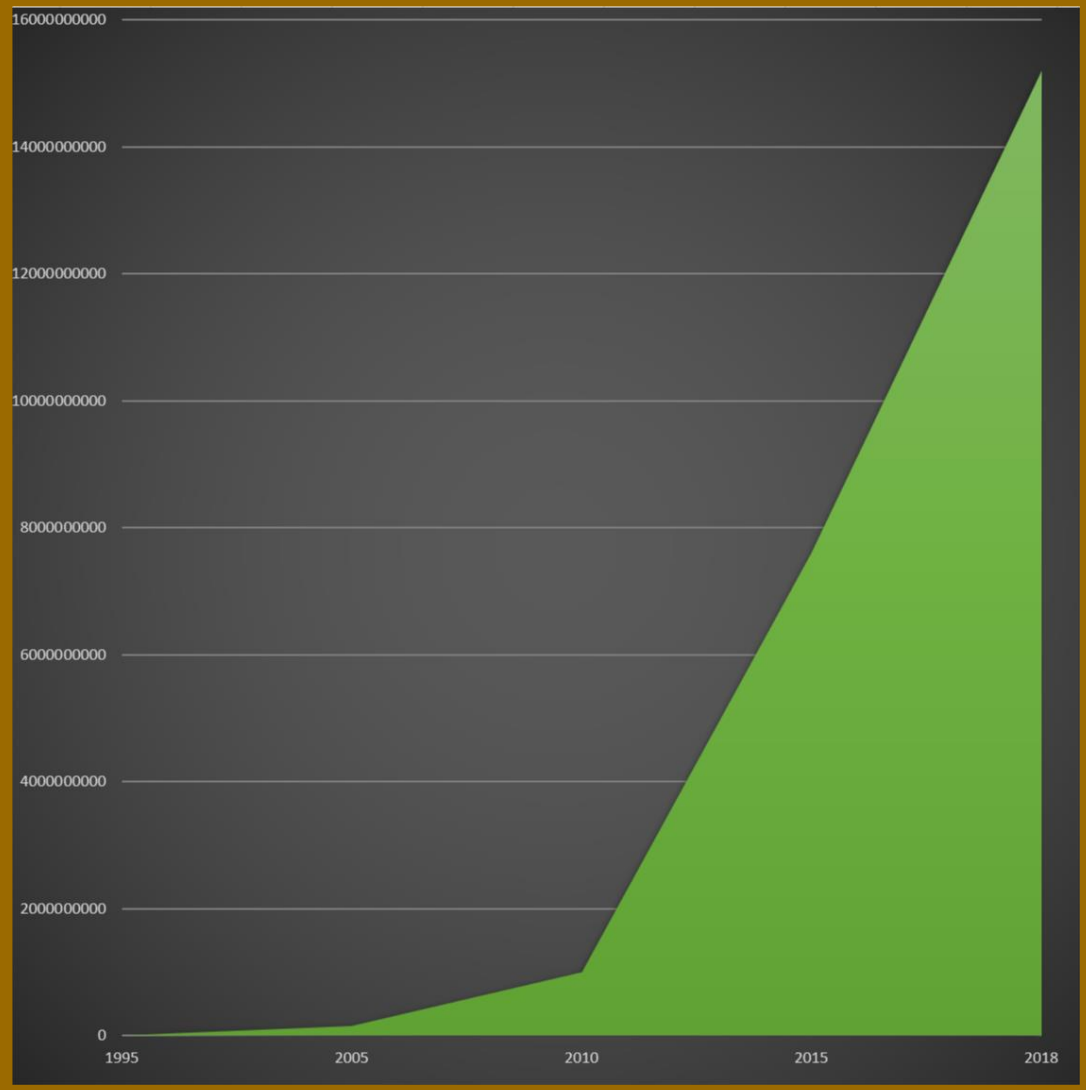
2005: 160M transistors on a single chip

2010: 1 Billion transistor

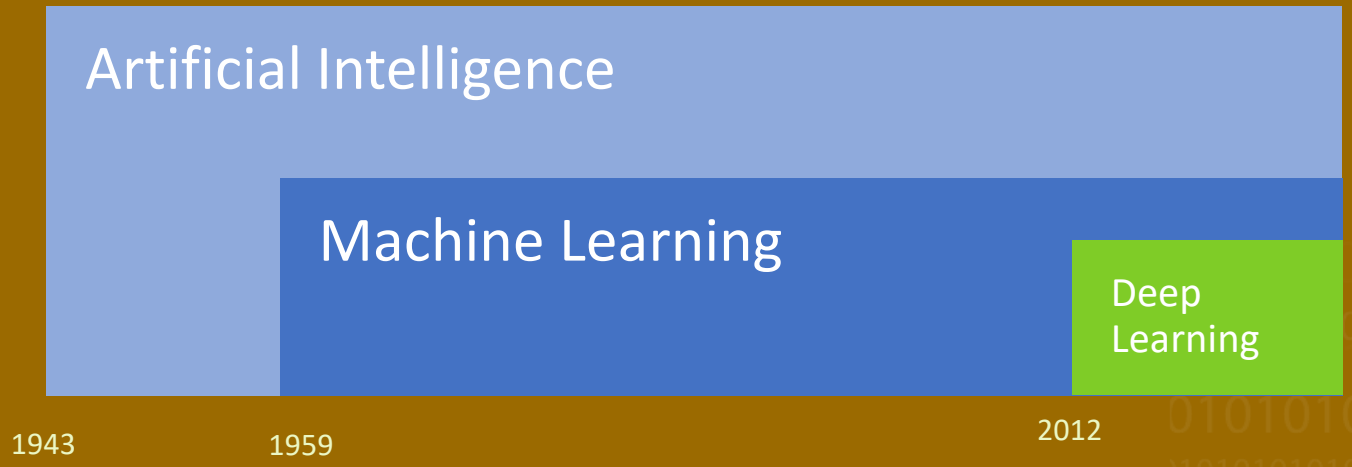
2015: 7.6 Billions transistors on a chip

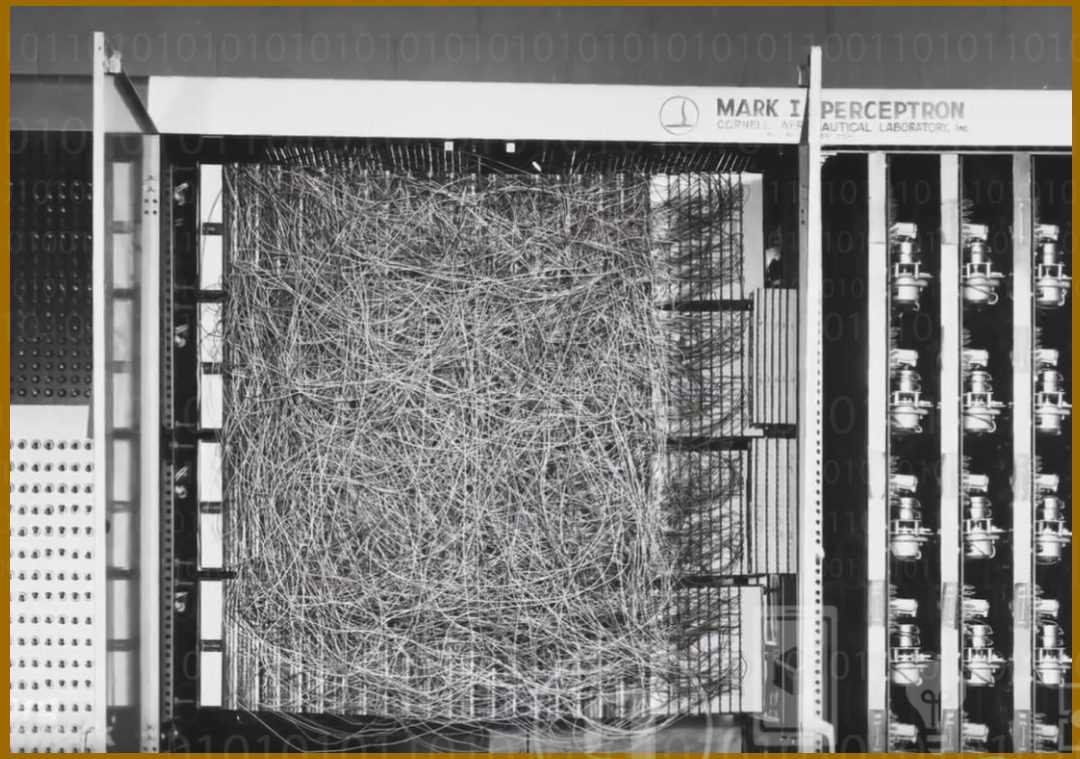
2017: 15.2 Billions transistors on a chip





01 Define what AI, ML and DL are





1943 - the first work that is now generally recognized as AI was McCullouch and Pitts' formal design for Turing-complete "artificial neurons".

1959 - The name machine learning was coined by Arthur Samuel. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data

2012 – A group of scientist built a network of 16,000 computer processors with one billion connectors and let it browse YouTube looking for cats. The "brain" simulation was exposed to 10 million randomly selected YouTube video thumbnails over the course of three days and, after being presented with a list of 20,000 different items, it began to recognize pictures of cats using a "deep learning" algorithm

Artificial Intelligence (AI) is the science of making things smart. Can be defined as:

> *Human Intelligence exhibited by machines*

A board term of getting computers to perform human tasks. The scope of AI is disputed and constantly changing over time.

Artificial Intelligence – the state of the art

System Implemented today are a form of narrow AI

A system that can do just one (or a few) defined things as well or better than humans

Like recognize object / detect a CC fraud in real time

update

Machine Learning

Machine Learning (ML) can be generally defined:

“An approach to achieve artificial intelligence through systems that can get better the more you run them”

ML involves teaching a computer to recognize patterns by example,
rather than programming it with specific rules.

Patterns can be found within data.

ML is about creating algorithms (or a set of rules) that learn complex function (or pattern) from data and make prediction on it

update

How ML works

ML is about predicting stuff and it is intelligent because:

- It takes some data (to train the system)
- Learns patterns from this data
- Classifies new data it has not seen before for a best guess
- Classification is based on knowledge gained in the learning process

Traditional programming

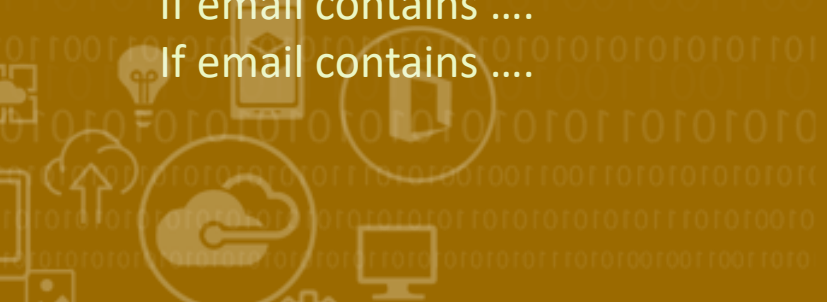
Write a computer program with explicit rules to follow

If email contains “Viagra”
then mark as spam
If email contains ….
If email contains ….

Machine Learning Programs

Write a computer program that learn from examples

Try to classify some emails (training)
Apply learned model to classify



update

Deep Learning

Deep Learning (DL) can be generally defined:

“A technique for implementing ML”

A DL technique is Deep Neural Networks (DNNs)

DL is the context of DNNs
DNNs is where the code you write is arranged in layers that loosely mimic the brain

update

AI Common Use Case

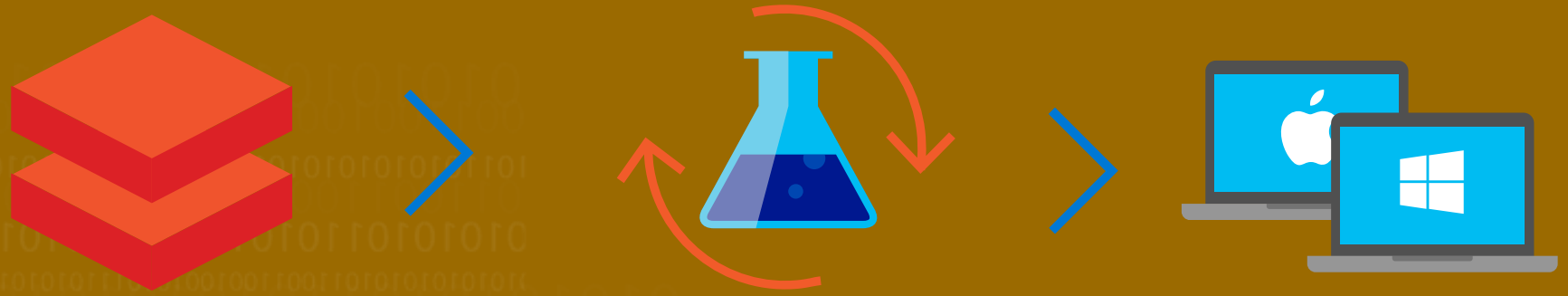
Object recognition
Speech recognition
Natural Language Processing
Creative (eg style transfer)
Prediction
Data Security
Financial (trading / fraud detection)

Healthcare
Marketing (custom ads/recommendations)
Online search
Smart Cars

update

02. Core DL Concepts, Terminology and Algorithms

AI development lifecycle



Prepare Data

Build & Train

Using…

Visual Studio 2017 with AI Tools

Visual Studio 2017 and ML.Net

Train on Azure

Deploy and use your model on client and server

update

Features and Attributes

Identify **attributes** of the things you are trying to classify
If I'm trying to classify **fruit** maybe is **color** and **weight**

Dimension usually refers to the number of attributes

Feature (or Attribute) is one particular "type of data" in your **datapoints**

Each **datapoint** (like fruit) contains many different attributes (like color, weight, shape)

update

The data challenge

Once you know features to use then the **challenge** is to find **enough data** to train the model

Imagine you need to **recognize cats**. You may need 10K sample images of cats if you want to get a good result

Data is any sort of features you need to classify your problem: it can be database rows, sound sample, video samples, text and so on

update

ML knowledge

An ML system cannot predict stuff it does not know about

Let's image you train an ML system with a dataset containing features: animal name, number of legs, color, weight :

- Dog, 4, black, 10Kg
- Chicken, 2, orange, 5Kg

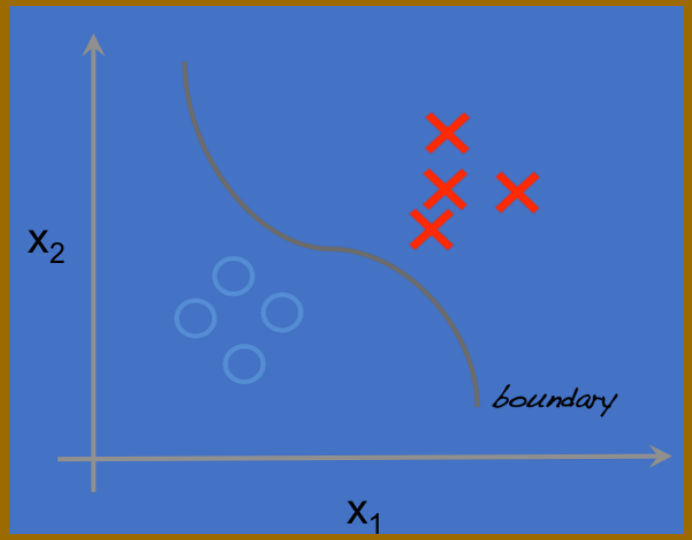
If you now ask the model to evaluate the row:

- Cow, 4, black, 200Kg

The model will predict "Dog" because it only knows about dogs and chickens and it got the nearest match

Ways to train a model – Supervised Learning

When the ML is trained with data, and those data are **labelled**



LABEL, number of legs, color, weight :

- Dog, 4, black, 10Kg
- Chicken, 2, orange, 5Kg

Give three inputs (number of legs, color, weight) we are telling the system what output label we do expect

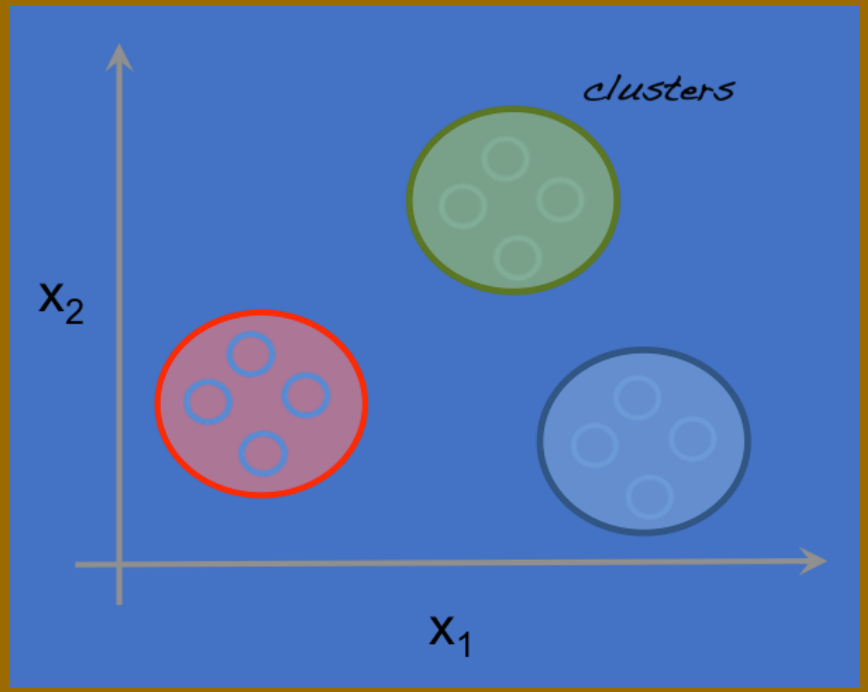
ML uses this data to predict future unseen data

Ways to train a model – Unsupervised Learning

ML learns from an unlabeled data set

Imagine we have some points on a graph representing 3 different things

ML must realize itself there are 3 distinct clusters and categorize them



This is tricky, because the number of clusters may not be known in advance, so it has to take a best guess.
Sometimes the clusters are not as clear as the ones shown here

Ways to train a model – Reinforcement Learning

ML learns by playing the game millions of times.

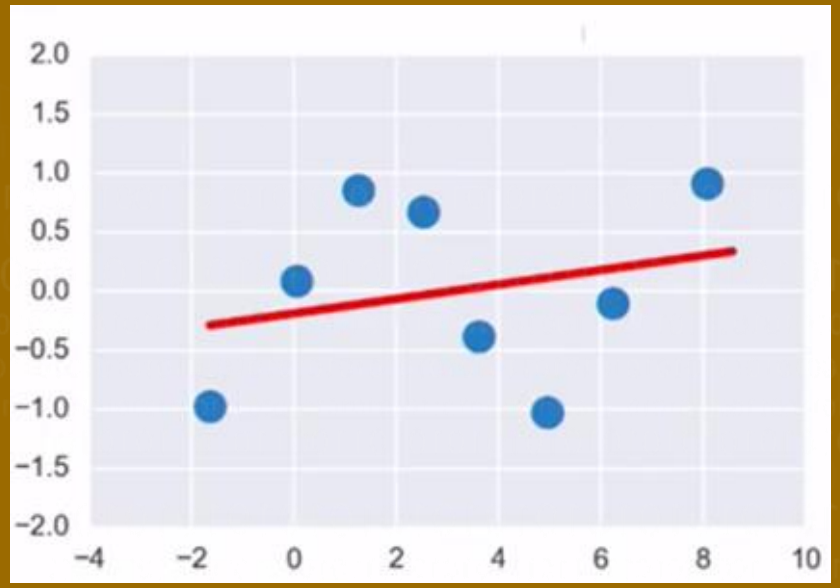
ML is rewarded when it makes a good move

When it loses, we give him no (or negative reward)

Overtime ML learns to maximize reward without the human explicitly telling the rules. It can lead to better than human performance when it finds paths that no one thought of doing before.

How the model training works

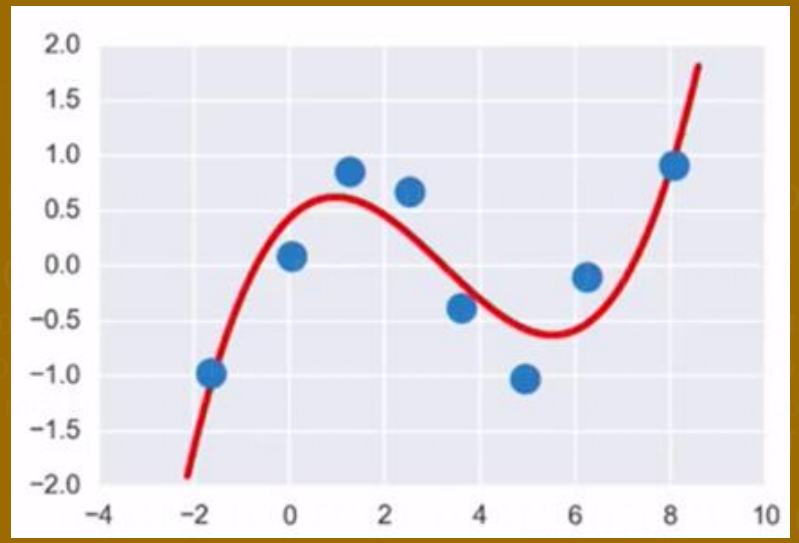
There are many ways a ML can learn pattern to classify data



In this example we use a line to divide two clusters.

We can predict future data saying anything above the line is owned by one cluster and anything below the line is owned by the other cluster

How the model training works



Cubic curve

Instead of a straight line we may use a cubic curve to identify clusters.

That way the ML will predict future points it has not seen before by understanding in witch area the point belongs

Neural Networks

Our brain consists of **86 billion** or so interconnected **neurons**.

Each neuron responds to certain stimuli and passes output to another.

There may be a bunch of them dedicated to recognize cats (some for fur, eyes, whiskers, etc) **each having a different weighting** (based on how important the feature is) to the overall contribution. If all of those fire, your brain tells you saw a cat

In ML, Artificial Neural Networks (loosely modelled to the brain) are used to calculate the probabilities for features they are trained to look for.

Artificial Neuron

The 3 arrows correspond to the 3 inputs coming into the network

Values [0.7,0.6,1.4] are the weights assigned to the corresponding input

Inputs get multiplied with their respective weights and their sum is taken.

Consider 3 inputs as x1,x2,x3.

Consider 3 weights be w1,w2,w3

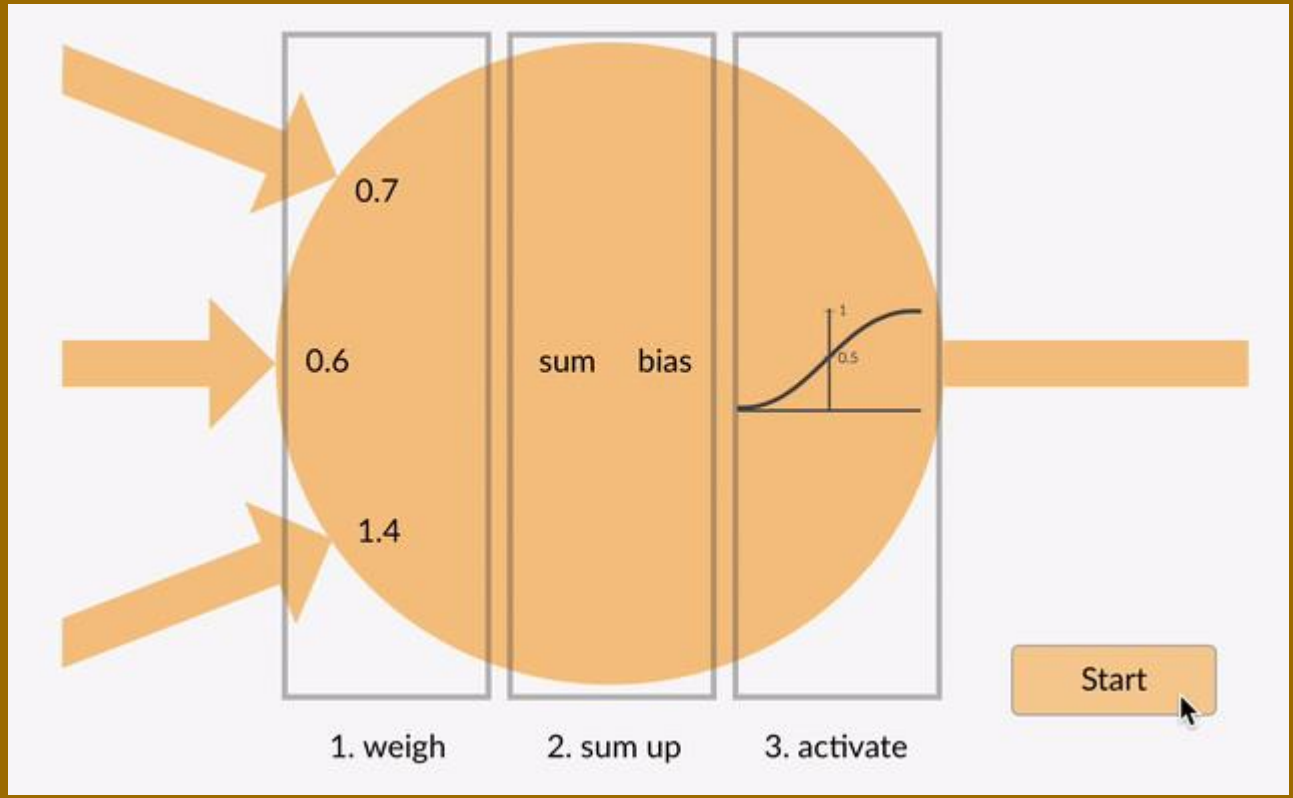
Sum = x1w1 + x2w2+x3w3.

i.e. Sum=x1(0.7) + x2(0.6) + x3(1.4)

After summing we add bias to the sum obtained.

This Bias is just a constant number say 1 which is added for scaling purposes.

NewSum=x1(0.7) + x2(0.6) + x3(1.4) + bias



Animation and definition courtesy of becominghuman.ai

It's not necessary to add bias but it is a good practice as it speeds up the process.

After adding bias, we reach at threshold step. If the new sum calculated is above the threshold value the neuron gets excited and it passes out the output.

If it doesn't get excited it won't pass on the output

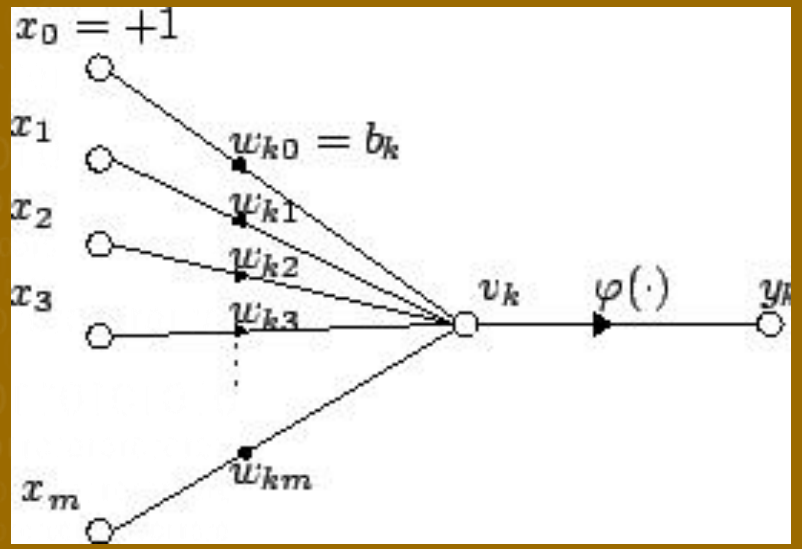
Artificial Neuron – the math behind

For a given artificial neuron, let there be $m + 1$ inputs with signals x_0 through x_m and weights w_0 through w_m. Usually, the x_0 input is assigned the value +1, which makes it a *bias* input with $w_{k0} = b_k$. This leaves only m actual inputs to the neuron: from x_1 to x_m

The output of the kth neuron is:

$$y_k = \varphi\left(\sum_{j=0}^{m} w_{kj} x_j\right)$$

Where (phi) is the transfer function

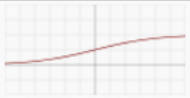
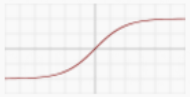
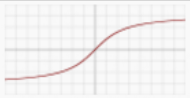


The output is analogous to the axon of a biological neuron, and its value propagates to the input of the next layer, through a synapse. It may also exit the system, possibly as part of an output vector.

It has no learning process as such. Its transfer function weights are calculated and threshold value are predetermined.

Source Wikipedia

Artificial Neuron - Activation Functions

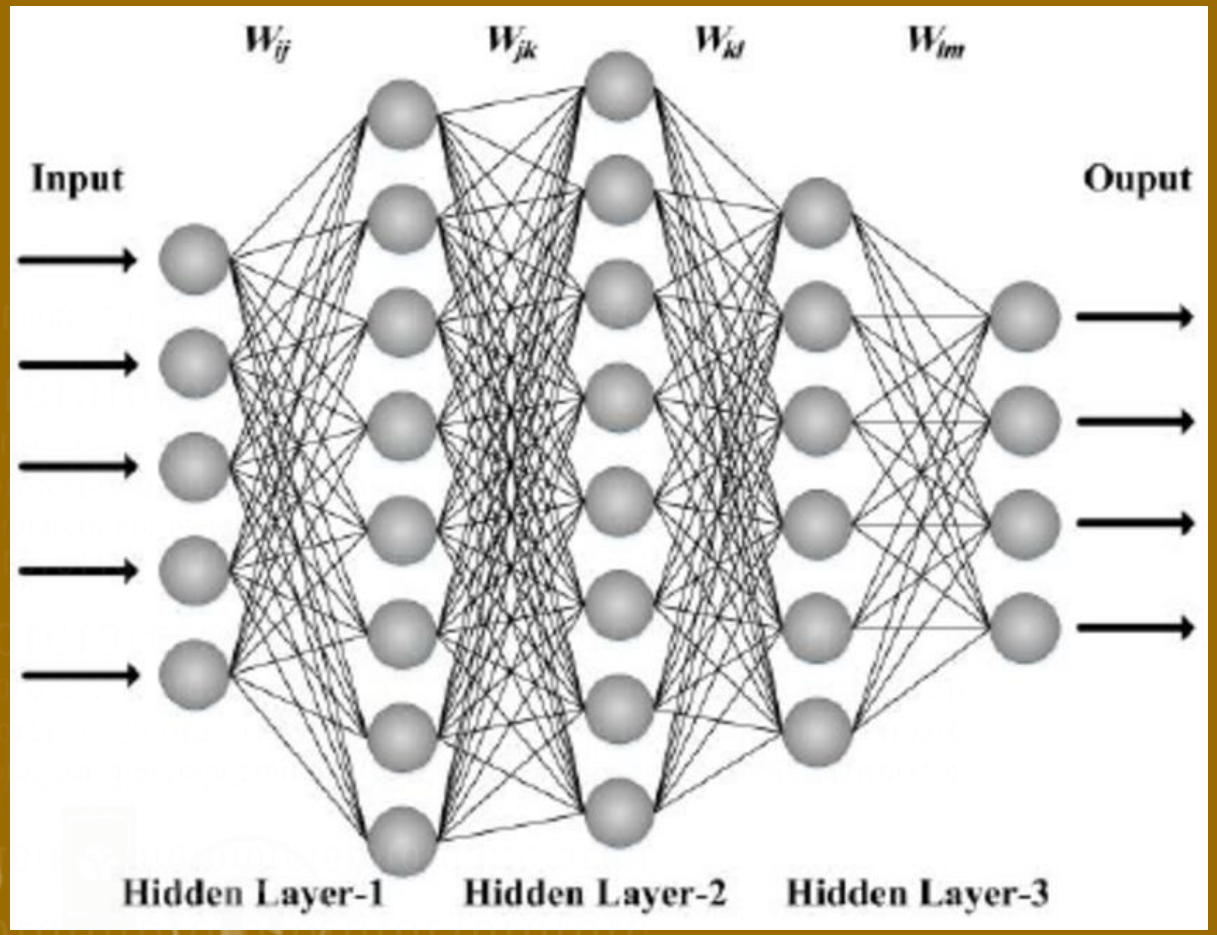
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

ReLu is the most used one

sigmoid/tanh rarely

update

Multi layer neuron



Stacking up many Artificial Neuron help us to create Intelligent Systems

update

Deep Neural Network

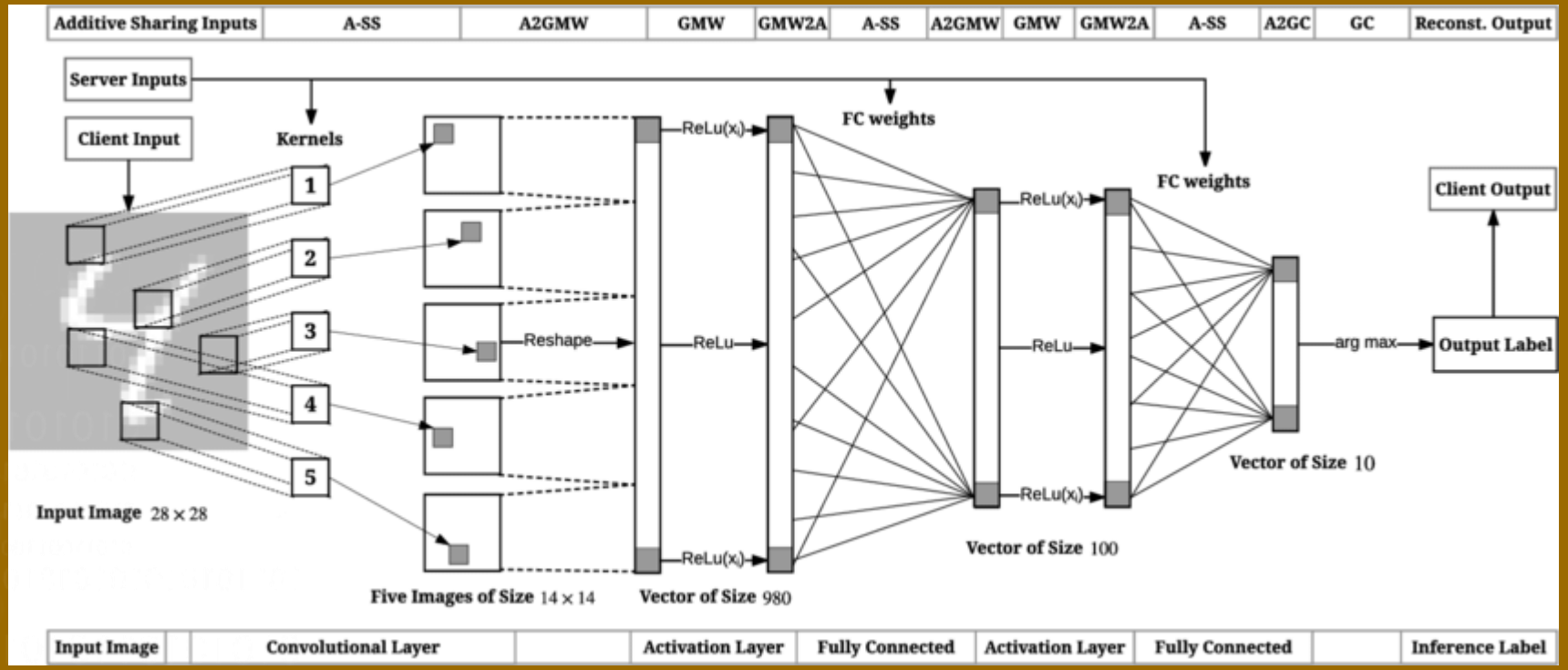
A Deep Neural Network (DNN) simply consists of many “hidden layers” between the input and the output.

Each layer can learn from the one before.

Hidden layer are usually of a lower dimension so the can generalize better and not overfit input data.

Middle layers can learn features of for features.

For example bunches of edges can lead to “face parts” witch leads to “faces”



DN comes in two very popular flavors

CNN

Convolutional Neural Networks
Vision

RNN

Recurrent Neural Networks
Language

The two most popular types of Deep Networks

Common ML Output Types

Regression

Predict numerical values
(e.g. price of house)

Classification

On of n labels..
Cat, dog, human

Clustering

Most similar other examples
(e.g. related products on Amazon)

Sequence Prediction

What comes next?
Given 1,2,3,4,5
Predict 6

04. Develop our first DL application

Tools

Visual Studio Tools For AI

Getting Started with Visual Studio Tools for AI:
https://channel9.msdn.com/events/Build/2018/THR3129
Sample repo: https://github.com/Microsoft/samples-for-ai

Win.ML

Introducing ML.Net: https://channel9.msdn.com/Events/Build/2018/BRK3203
Samples repo: https://github.com/dotnet/machinelearning-samples

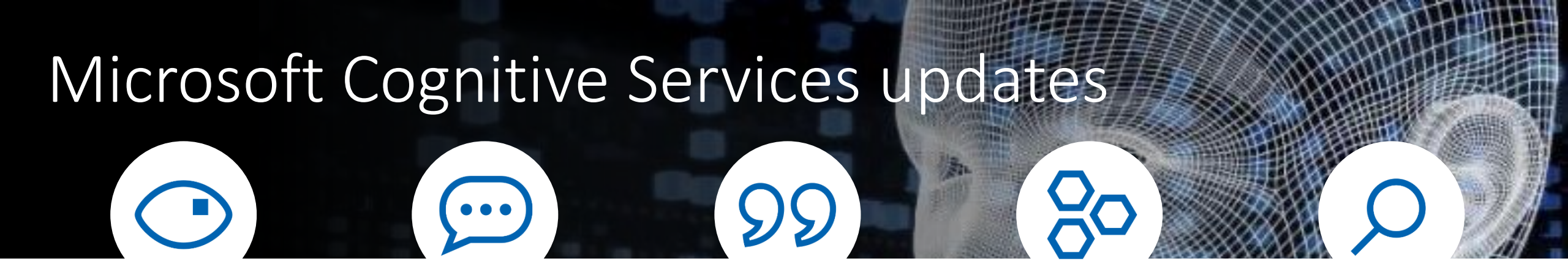
03. Three ways of using AI

Option 1 – Easy

cognitive services

Infuse your apps, websites and bots with intelligent algorithms to see, hear, speak, understand and interpret your user needs through natural methods of communication

Microsoft Cognitive Services updates



Vision	Speech	Language	Knowledge	Search
Video Indexer	Speaker Recognition	Text Analytics	QnA Maker	Bing Entity Search
Computer Vision	Bing Speech	Bing Spell Check		Bing Autosuggest
Face	Translator Speech	Translator Text		Bing Search
Emotion	Unified Speech			Bing Statistics add-in
Content Moderator				Bing Visual Search

Pre-built ML Models (Azure Cognitive Services)

Consume (C#, VB, F#)

e.g. Sentiment Analysis using Azure Cognitive Services

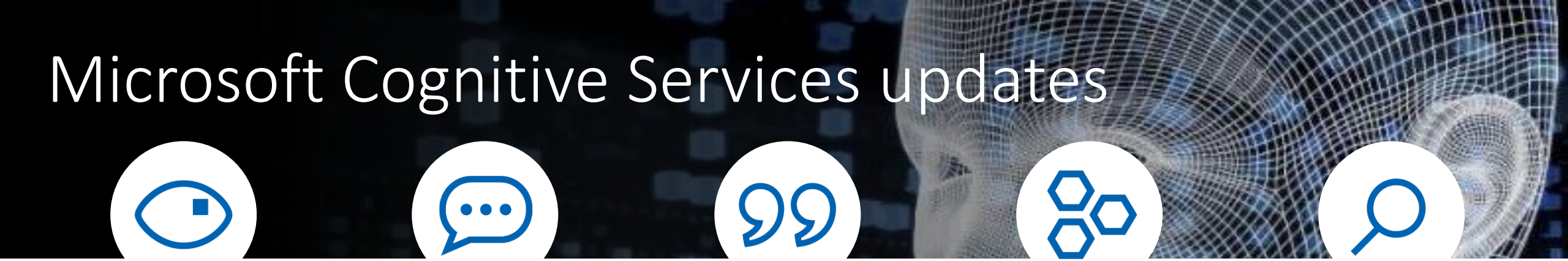
```
TextAnalyticsAPI client = new TextAnalyticsAPI();
client.AzureRegion = AzureRegions.Westus;
client.SubscriptionKey = "1bf33391DeadFish";

client.Sentiment(
new MultiLanguageBatchInput(
new List<MultiLanguageInput>()
{
 new MultiLanguageInput("en","0",
     "This is a great movie")
}));
```

96% positive

03. Three ways of using AI

Option 2 – Still easy, but more work

Microsoft Cognitive Services updates

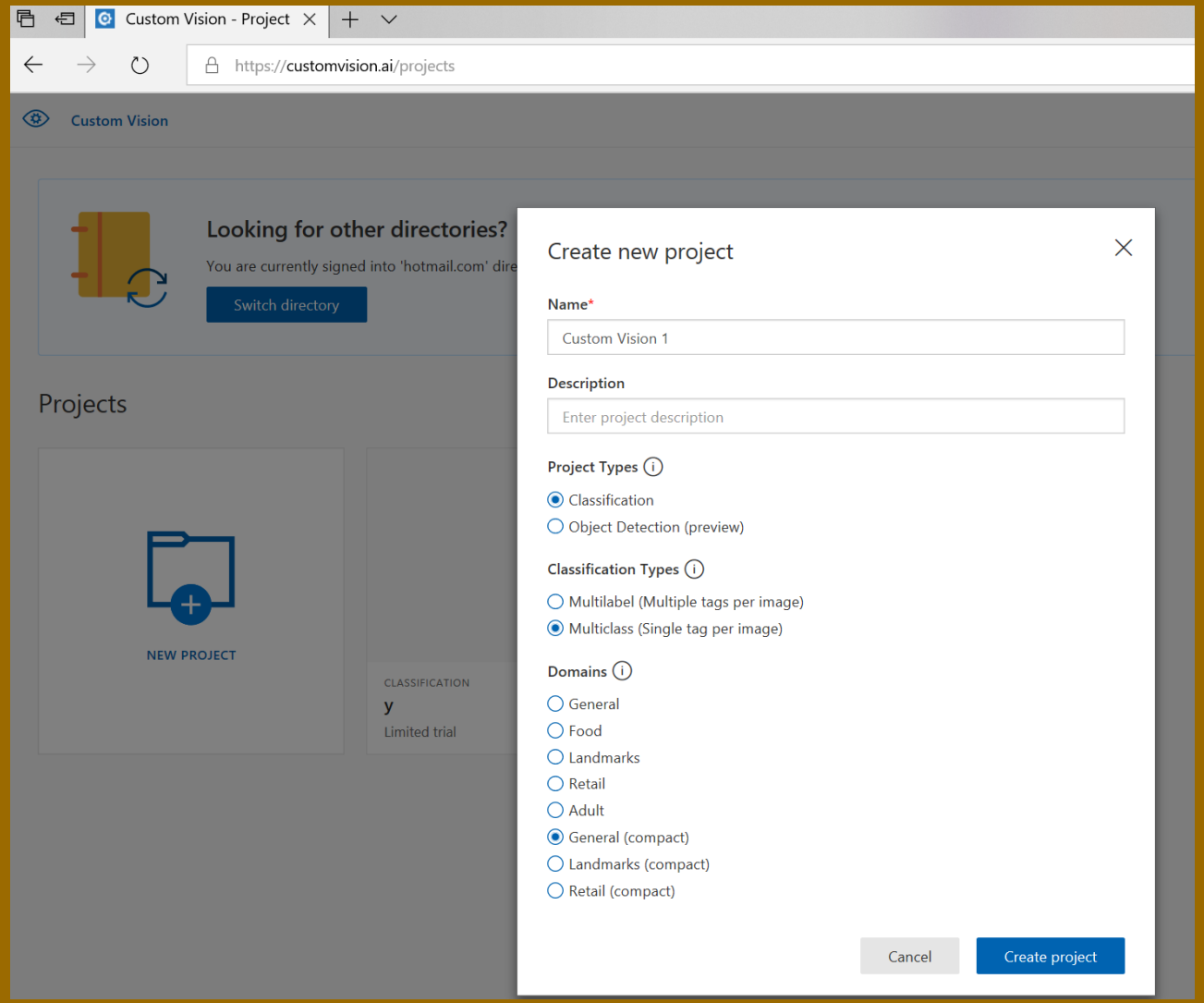


Vision	Speech	Language	Knowledge	Search
Video Indexer	Speaker Recognition	Text Analytics	QnA Maker	Bing Entity Search
Computer Vision	Bing Speech	Bing Spell Check		Bing Autosuggest
Face	Translator Speech	Translator Text		Bing Search
Emotion	Unified Speech			Bing Statistics add-in
Content Moderator				Bing Visual Search

Customization

Custom Vision	Custom Speech	Language Understanding (LUIS)	Custom Decision	Bing Custom Search

Using Custom Cognitive Services

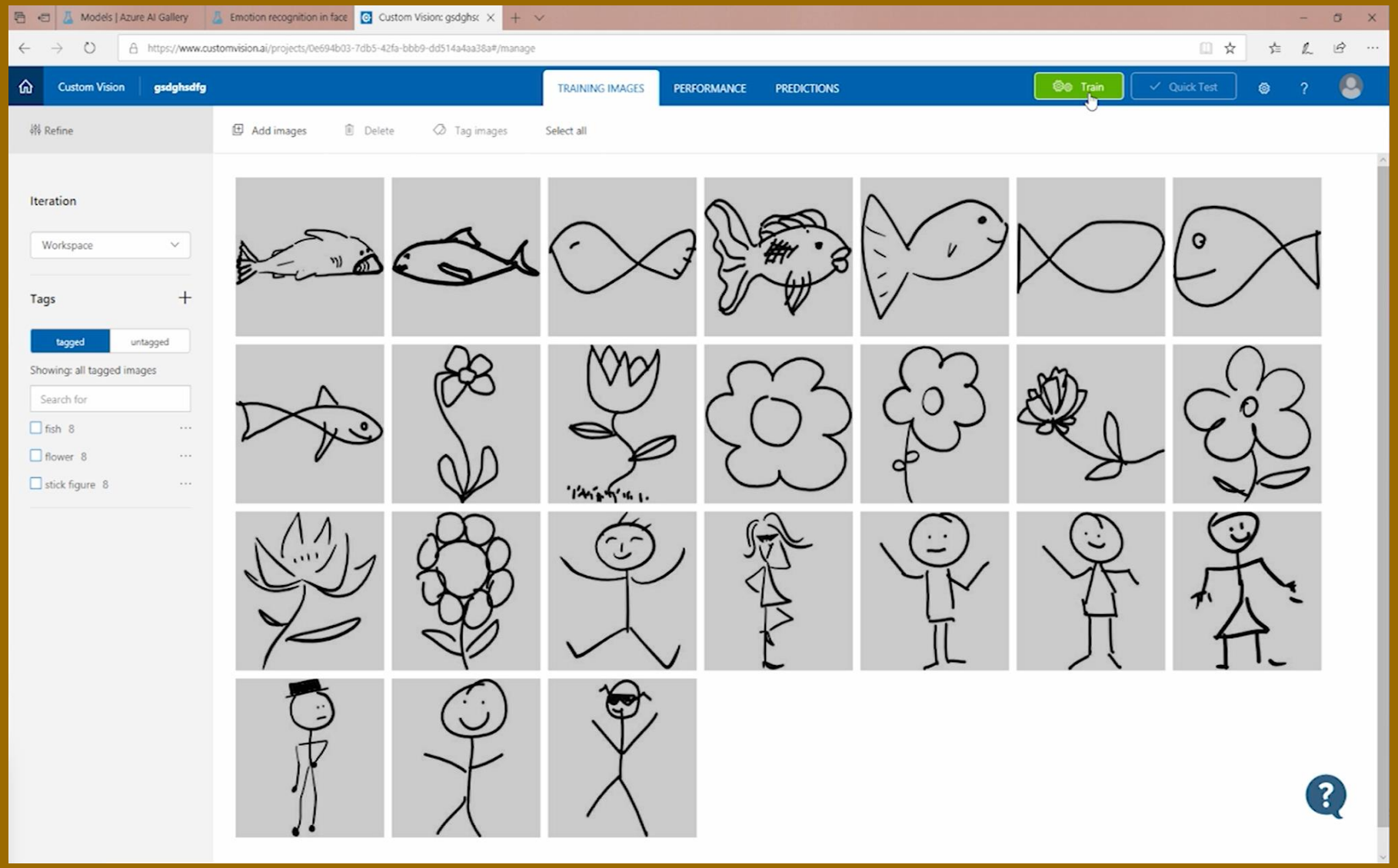


Customvision.ai

Create new project

update

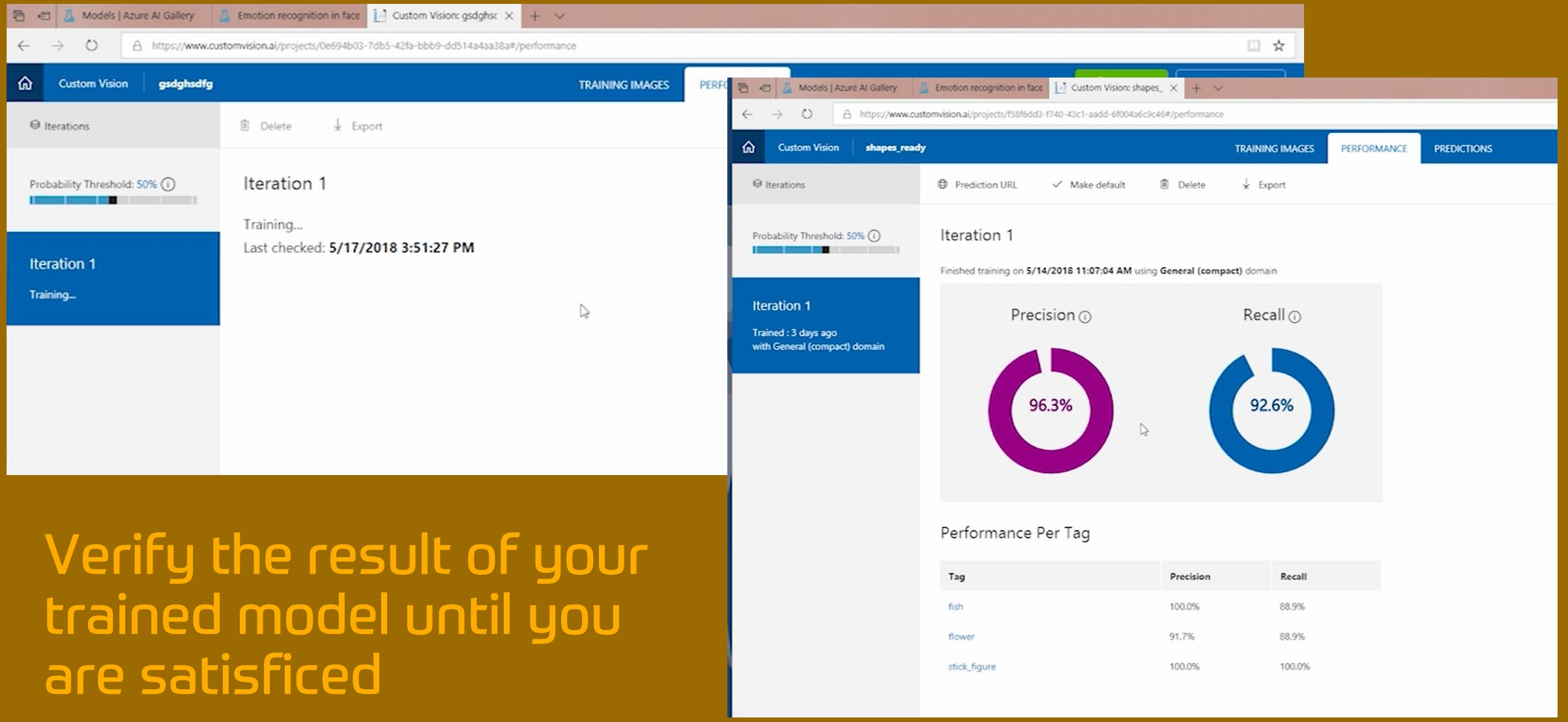
Using Custom Cognitive Services



Add your sample data

Train the model

Using Custom Cognitive Services



Verify the result of your trained model until you are satisficed

Using Custom Cognitive Services

```
protected async override void OnNavigatedTo(NavigationEventArgs e)
{
    // load Model
    var file = await StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-appx:///inkshapes.onnx"));
    model = await Model.CreateModel(file);

    setup InkCanvs, sound and timers
}
```

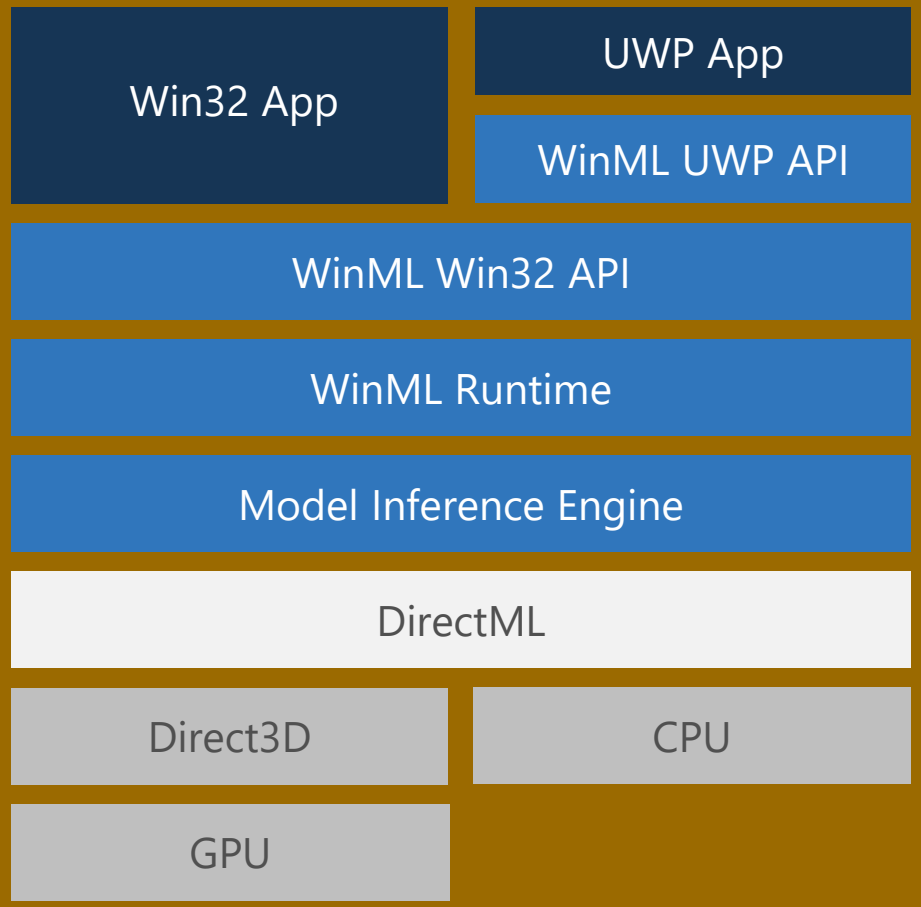
Import the model in your project and load it

```
private async void InkPresenter_StrokesCollectedAsync(InkPresenter sender, InkStrokesCollectedEventArgs args)
{
    var bitmap = Inker.GetCropedSoftwareBitmap(newWidth: 227, newHeight: 227, keepRelativeSize: true);
    var frame = VideoFrame.CreateWithSoftwareBitmap(bitmap);
    var input = new ModelInput() { data = frame };

    var output = await model.EvaluateAsync(input);

    var guessedTag = output.classLabel.First();
    var guessedPercentage = output.loss.OrderByDescending(kv => kv.Value).First().Value;

    if (guessedPercentage < 0.9)
    {
        SubText.Text = $"draw {currentShape} to snooze - don't know what that is";
    }
    else if (guessedTag != currentShape)
```

Evaluate against your data

Windows ML solves three problems for you

1. Developers can focus on their data and their scenarios, using Windows ML for model evaluation
2. Enables using ML models trained with a diverse set of toolkits
3. Hardware acceleration gets fast evaluation results across the diversity of the entire Windows device ecosystem.



ONNX

OPEN NEURAL NETWORK EXCHANGE FORMAT

The new open ecosystem for interchangeable AI models

MAY 2, 2018 ONNX EXPANSION SPEEDS AI DEVELOPMENT READ MORE

aws Facebook Open Source Microsoft

What is ONNX?

ONNX is a open format to represent deep learning models. With ONNX, AI developers can more easily move models between state-of-the-art tools and choose the combination that is best for them. ONNX is developed and supported by a community of partners.

AMD arm NVIDIA HUAWEI IBM intel AI MEDIATEK Preferred Networks Qualcomm SOPHON skymizer Baidu 百度

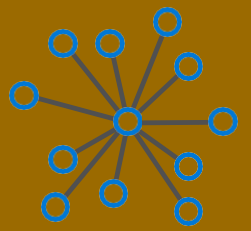
How do I get ONNX models to use in my application?

- Azure Machine Learning Services gives you an end-to-end solution to prepare data, and train your model in the Cloud.
- WinMLTools converts existing models from CoreML, scikit-learn, LIBSVM, and XGBoost
- Azure Custom Vision makes it easy to create your own image models - https://customvision.ai/

- Azure AI Gallery curates models for use with Windows ML - https://gallery.azure.ai/models

03. Three ways of using AI

Option 3 – Hands will be dirtied

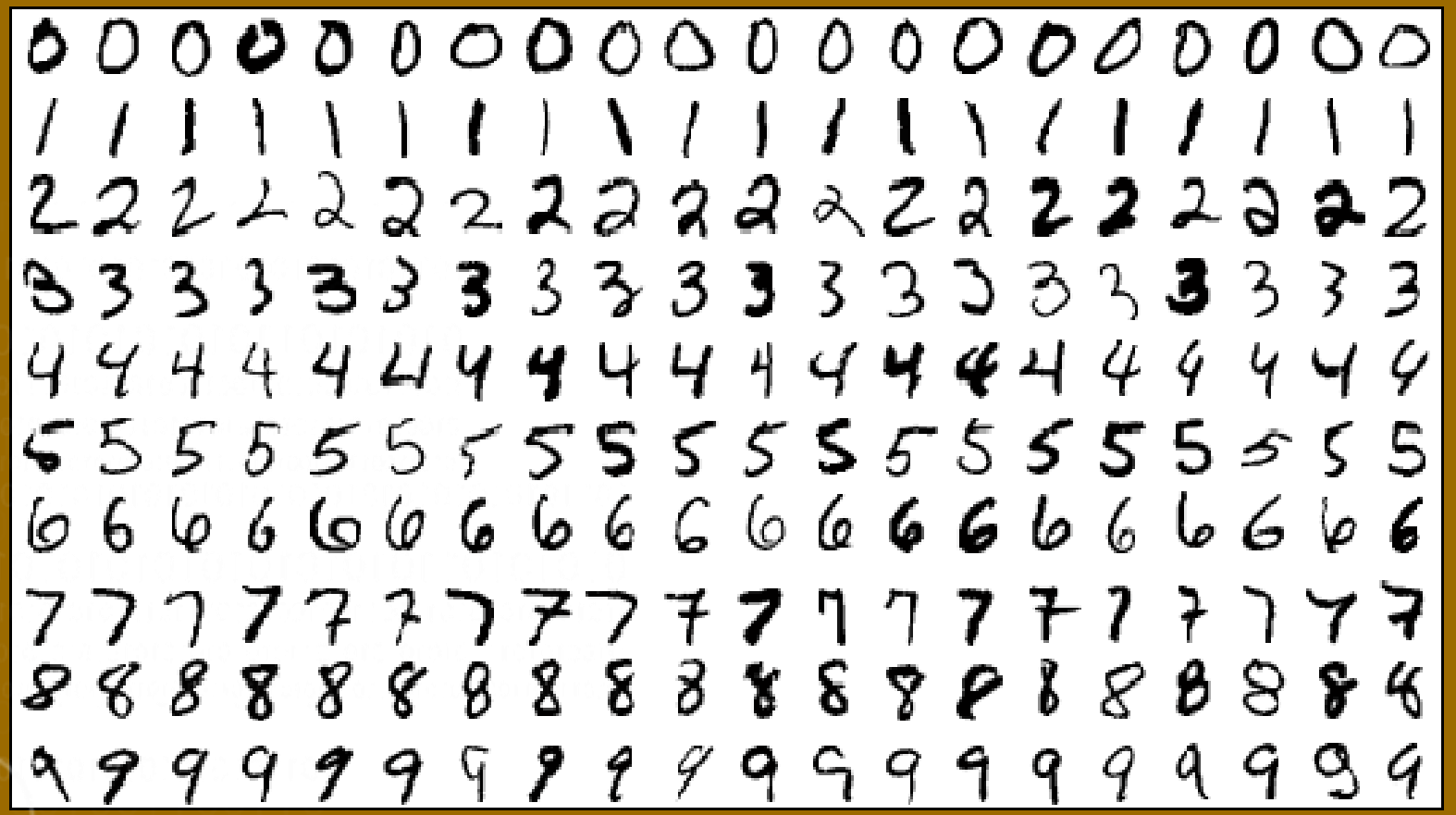


Coding your AI

Datasets

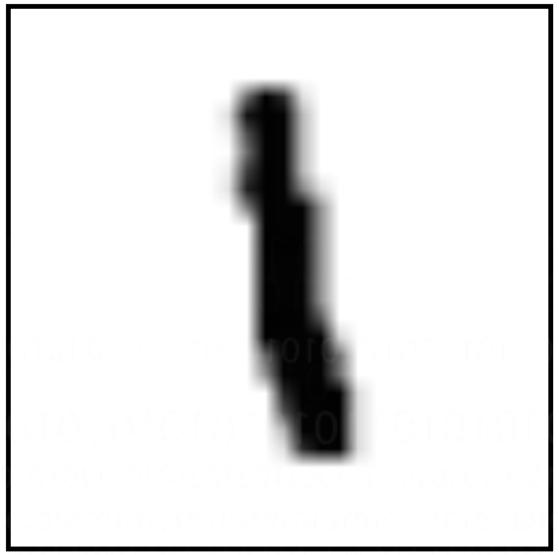
MNIST database of hand-written digits

(Modified) National Institute of Standards and Technology



http://yann.lecun.com/exdb/mnist/

28



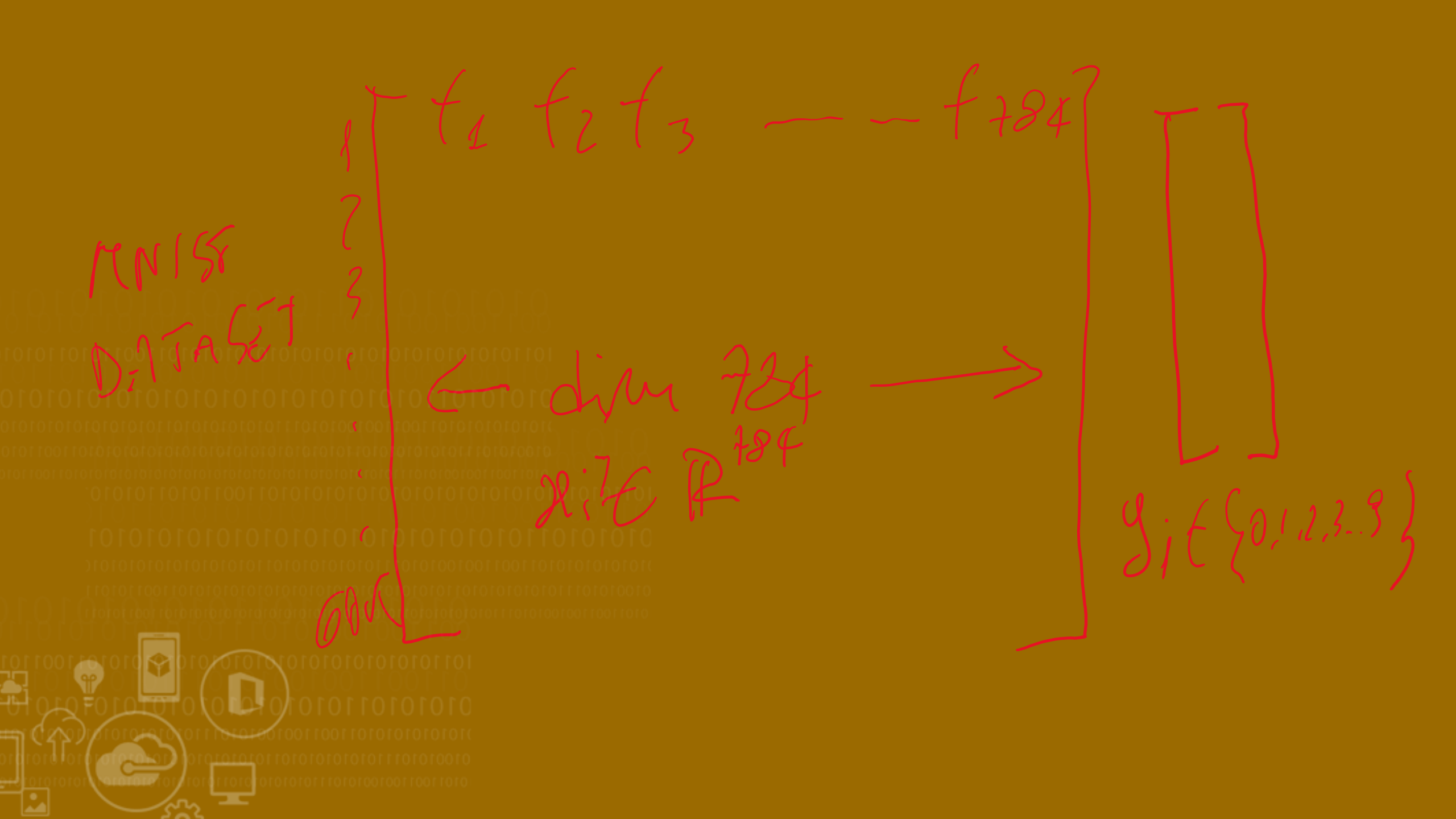
28

≈

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	.6	.8	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.7	1	0	0	0	0	0	0
0	0	0	0	0	0	.5	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.4	0	0	0	0	0
0	0	0	0	0	0	0	1	.7	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	.9	1	.1	0	0	0	0
0	0	0	0	0	0	0	.3	1	.1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

DP = R_1 = image = [ ] 28x28 —> FLATTENING ROW/COLUM

[ ] 784 x 1


MNIST
DATASET
f_1 f_2 f_3 — — f_{784}
dim 784
$x_i \in \mathbb{R}^{784}$
$y_i \in \{0,1,2,3..9\}$

04. Develop our first DL application

Resources

Make Your Own Neural Network - Rashid, Tariq
Deep Learning with Python - Chollet, Francois
Yaser S. Abu-Mostafa – Learning from data https://www.amazon.com/Learning-Data-Yaser-S-Abu-Mostafa/dp/1600490069 (inside the book instruction to access whole semester course gave by prof. Yaser S. Abu-Mostafa at California Institute of Technology)
Andrew Glassner – Deep Learning: A crash Course https://www.youtube.com/watch?v=r0Ogt-q956I
MIT - Massachusetts Institute of Technology – Professional Certificate Program in Machine Learning & Artificial Intelligence

Microsoft's official AI site for devs azure.com/ai
Microsoft's AI School https://aischool.microsoft.com/en-us/learning-paths
Rick Barraza – AI for Everyone! https://www.youtube.com/watch?v=xhp1m9thnbE
Jason Mayers – Machine Learning in 45 Minutes https://www.youtube.com/watch?v=X4I9QmcSEYo
Chris Lauren – Seth Juarez - Demystifying Machine and Deep Learning for Developers https://www.youtube.com/watch?v=cU7Wq5k8u-U
Introducing ML.NET https://www.youtube.com/watch?v=OhCysVU5RDA
ONNX https://onnx.ai/
ONNX Model Zoo https://github.com/onnx/models
ONNX models for WinML https://docs.microsoft.com/en-us/windows/ai/get-onnx-model
Azure AI Gallery curates models for use with Windows ML - https://gallery.azure.ai/models
Ml.Net Samples https://github.com/dotnet/machinelearning-samples
CH9 – AI Show https://channel9.msdn.com/Shows/AI-Show
The MNIST Database http://yann.lecun.com/exdb/mnist/
Kaggle Datasets database https://www.kaggle.com/datasets
Getting started with TensorFlow https://tf.wiki/en/basic.html
Getting Started with Visual Studio Tools for AI https://www.youtube.com/watch?v=LLW_0MQoQ1g
Visual Studio Tools For AI https://docs.microsoft.com/en-us/visualstudio/ai/?view=vs-2017
Visual Studio Tools For AI GitHub Sample repo https://github.com/Microsoft/samples-for-ai
CNTK Examples https://docs.microsoft.com/en-us/cognitive-toolkit/Examples
TensorFlow model zoo https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
Cafee model zoo https://github.com/BVLC/caffe/wiki/Model-Zoo
Christopher Olah's blog about DL http://colah.github.io/

Credits

Special thanks to the following people and groups that contributed inspiration, ideas or support for this ever-evolving talk / workshop:

https://twitter.com/jason_mayes
https://twitter.com/rickbarraza
https://twitter.com/sethjuarez
https://twitter.com/ankitasthana86
https://twitter.com/ch402
http://www.asimovinstitute.org/

update

Thank You

Got questions?

Gian Paolo Santopaolo
IBV Solutions
gp@ibv.ch
@gsantopaolo