#NDCOslo

Deep Learning in the world of little ponies

@GaliyaWarrier

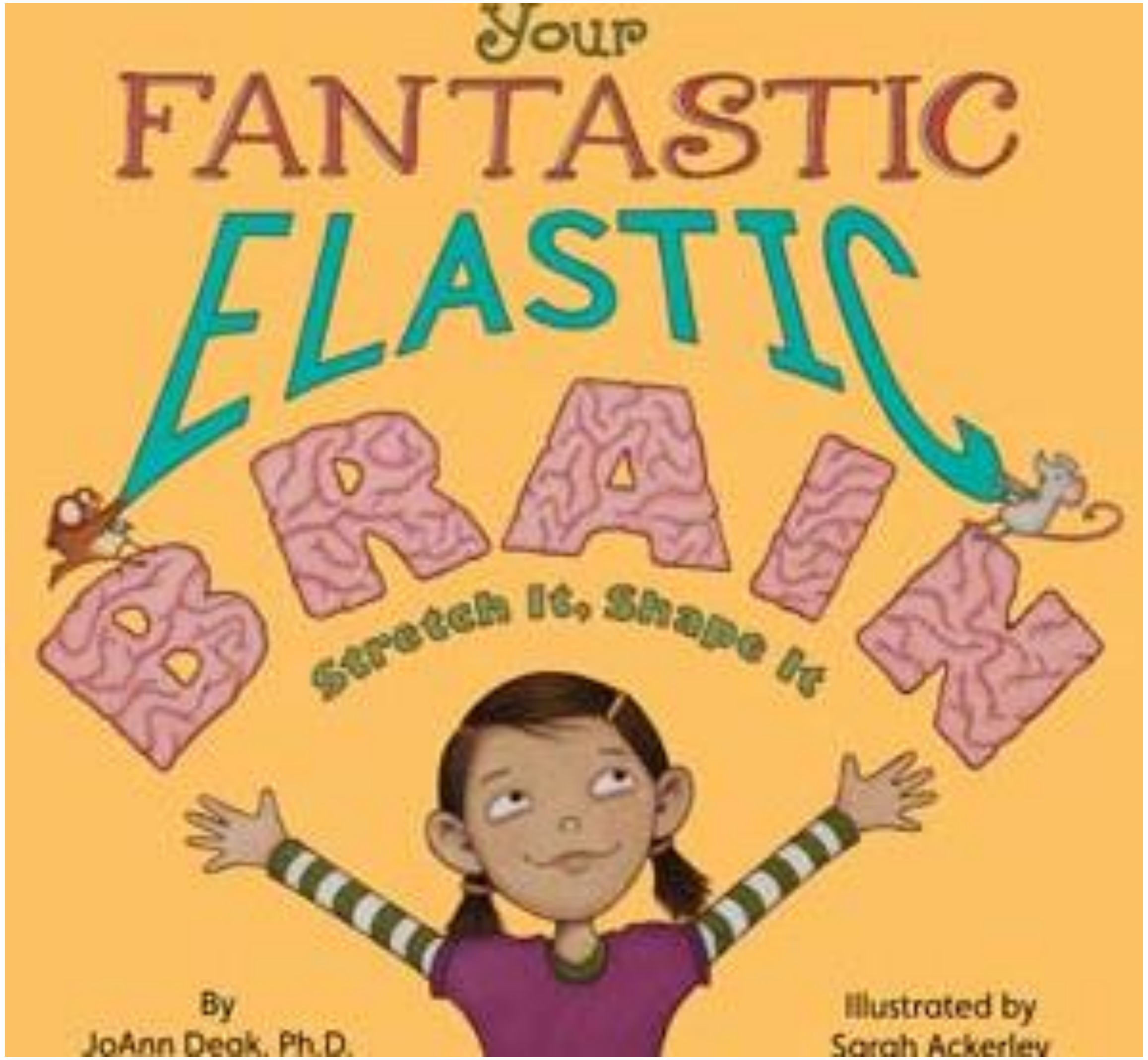
Galiya Warrier

Sr. Software Engineer,
(Commercial Software
Engineering | Health)
Microsoft

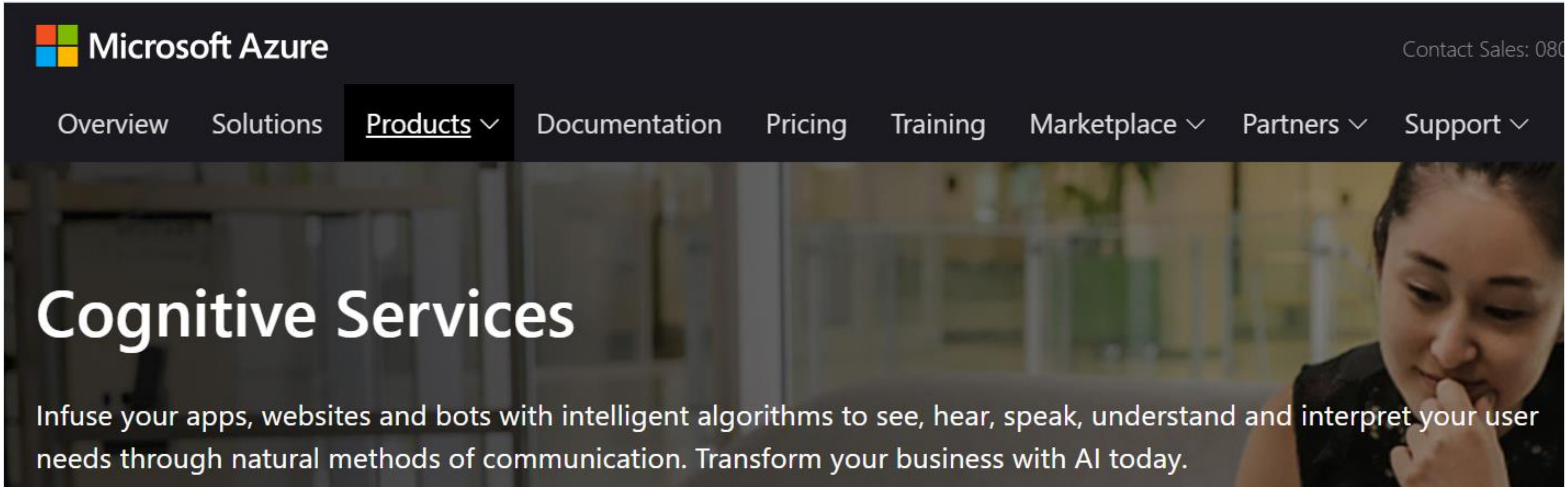
Maya Warrier

5 years old,
reception class

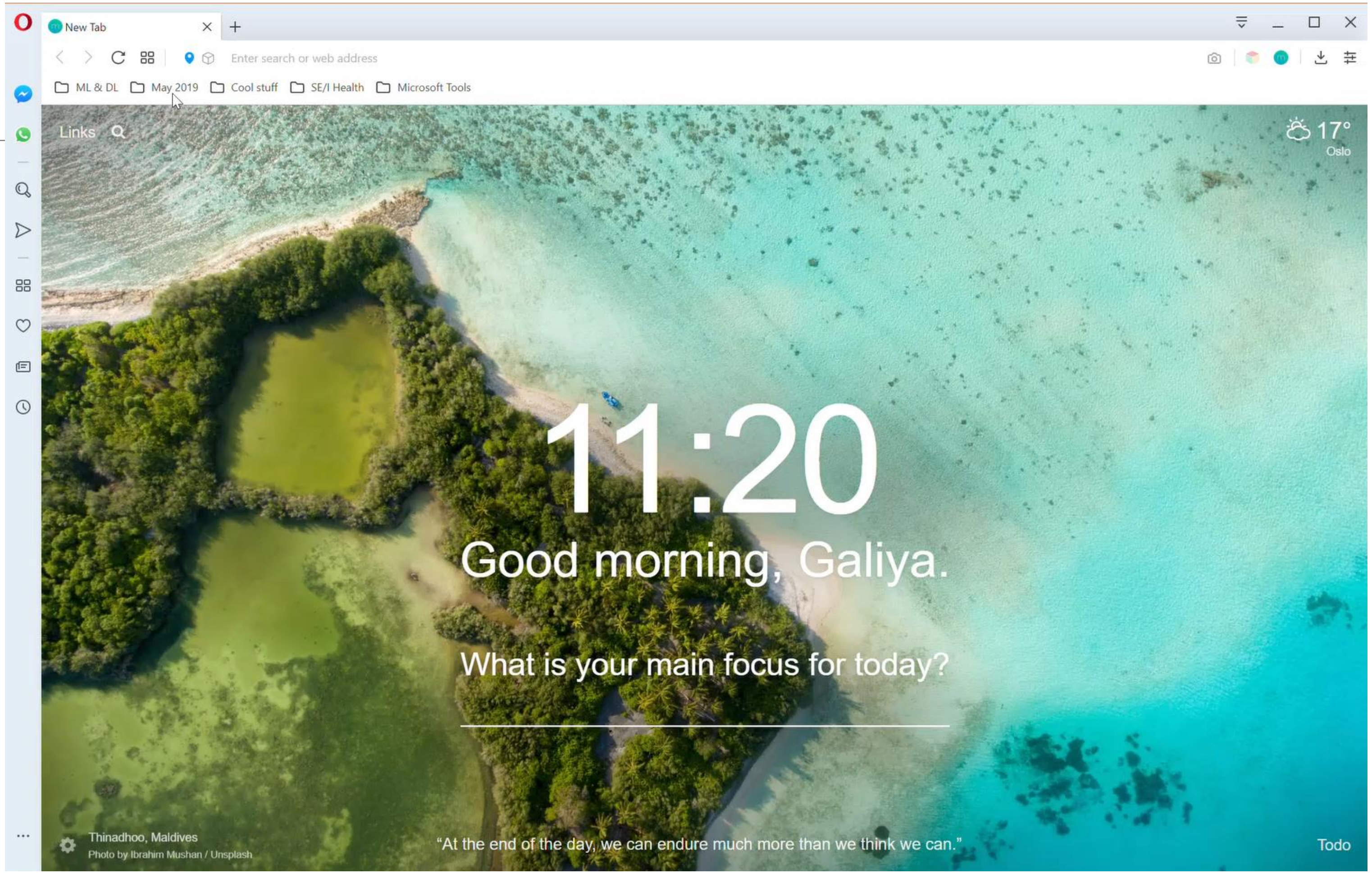
@galiyawarrier



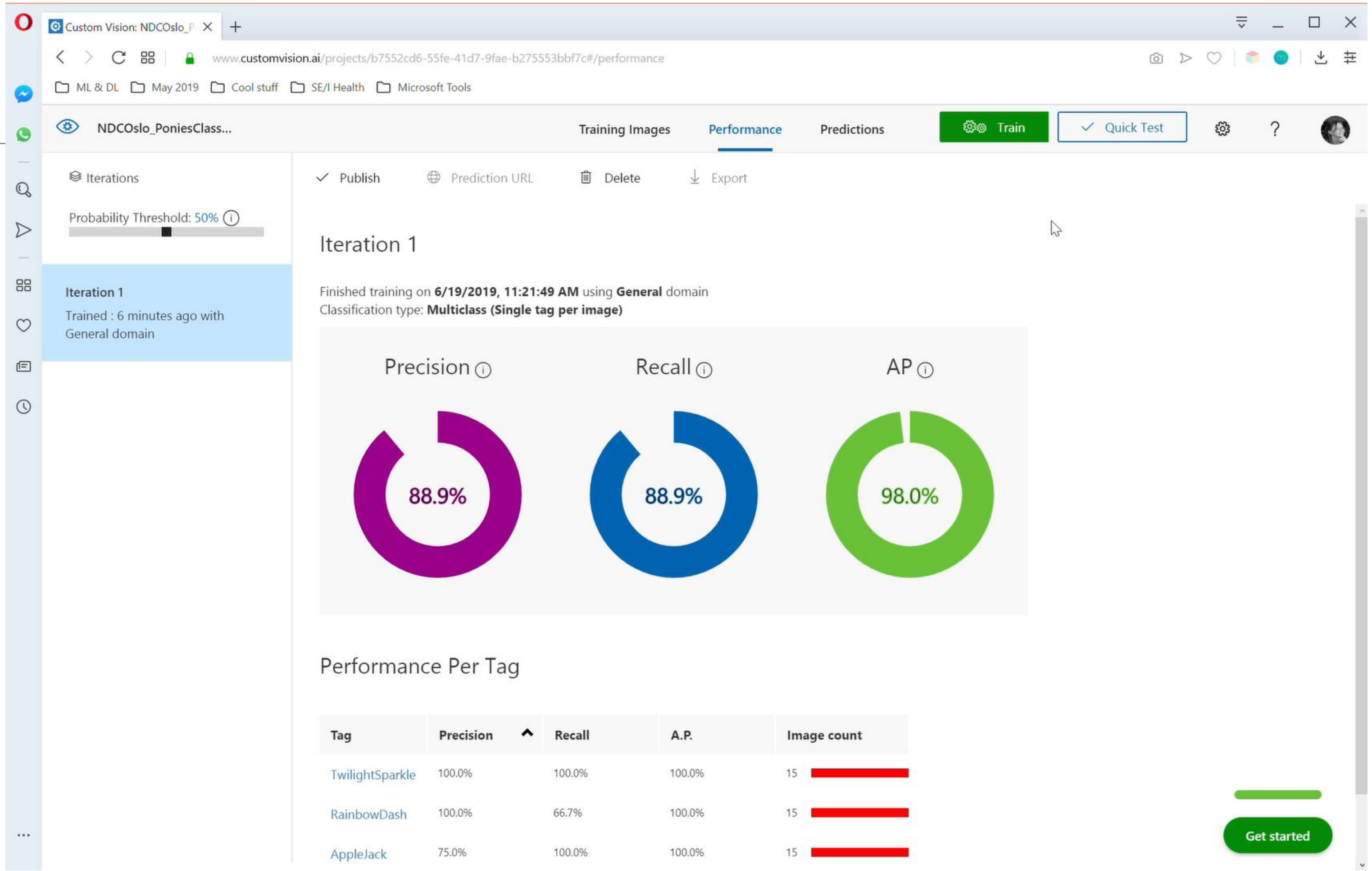
Growth mindset? ☺


Microsoft Azure
Contact Sales: 080
Overview
Solutions
Products
Documentation
Pricing
Training
Marketplace
Partners
Support
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. Transform your business with AI today.

Train model


New Tab
Enter search or web address
ML & DL
May 2019
Cool stuff
SE/I Health
Microsoft Tools
Links
17°
Oslo
11:20
Good morning, Galiya.
What is your main focus for today?
Thinadhoo, Maldives
Photo by Ibrahim Mushan / Unsplash
"At the end of the day, we can endure much more than we think we can."
Todo

Publish model




Carrier
3:26 PM
Core ML
Library
CHASE
YOUR
DREAMS
My Little Pony | Twilight - Cha... by mylittlepony
Zazzle
Hey, Galiya, it looks like ~~princess cadence~~.
FAIL!

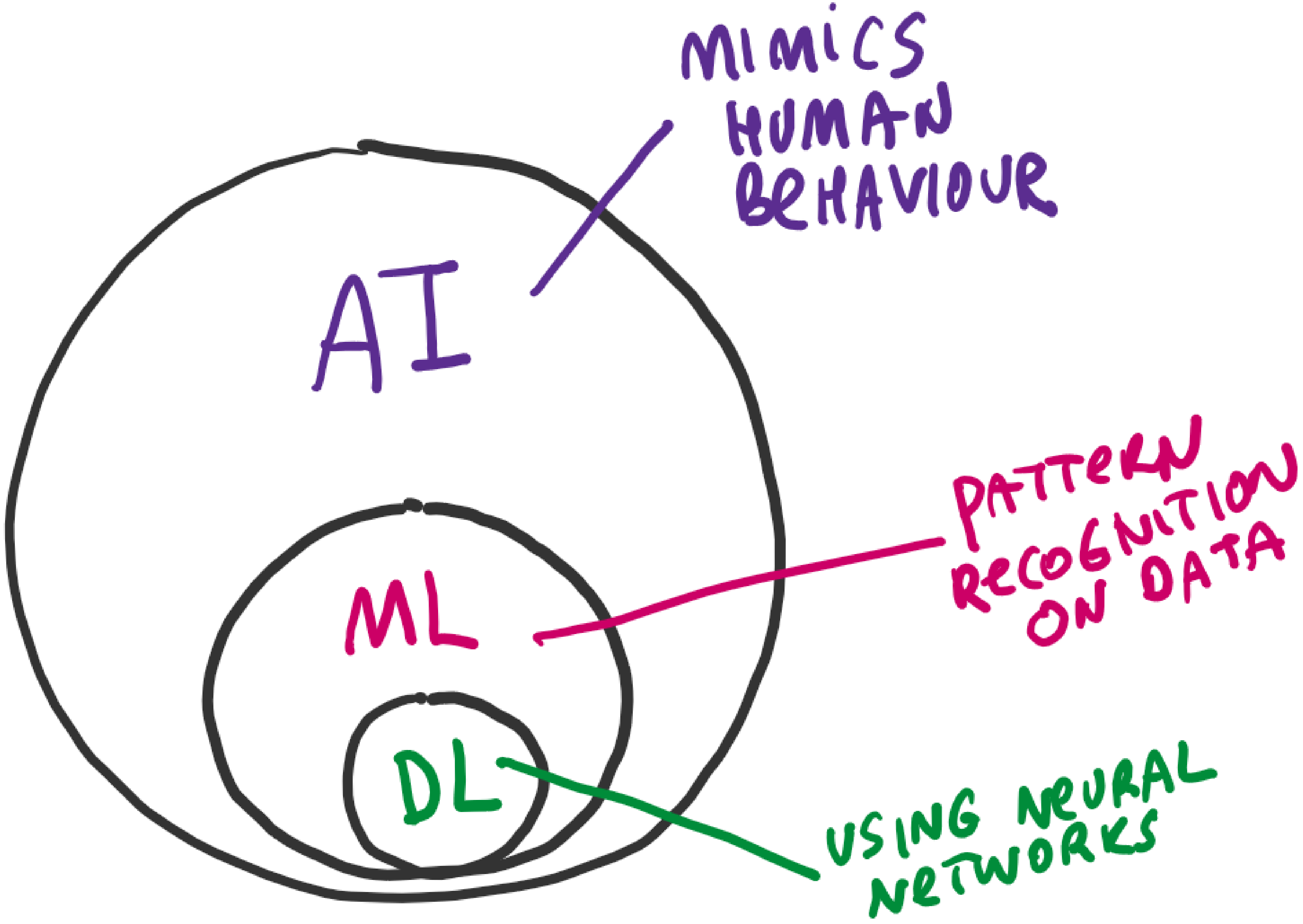

Carrier
3:22 PM
Core ML
Library
THERMOS
Hey, Galiya, it looks like princess cadence.


Carrier
3:37 PM
Core ML
Library
ADVENTURE
IS CALLING
I MUST
GO!
Hey, Galiya, it looks like rainbow dash.

Problem solved! ☺

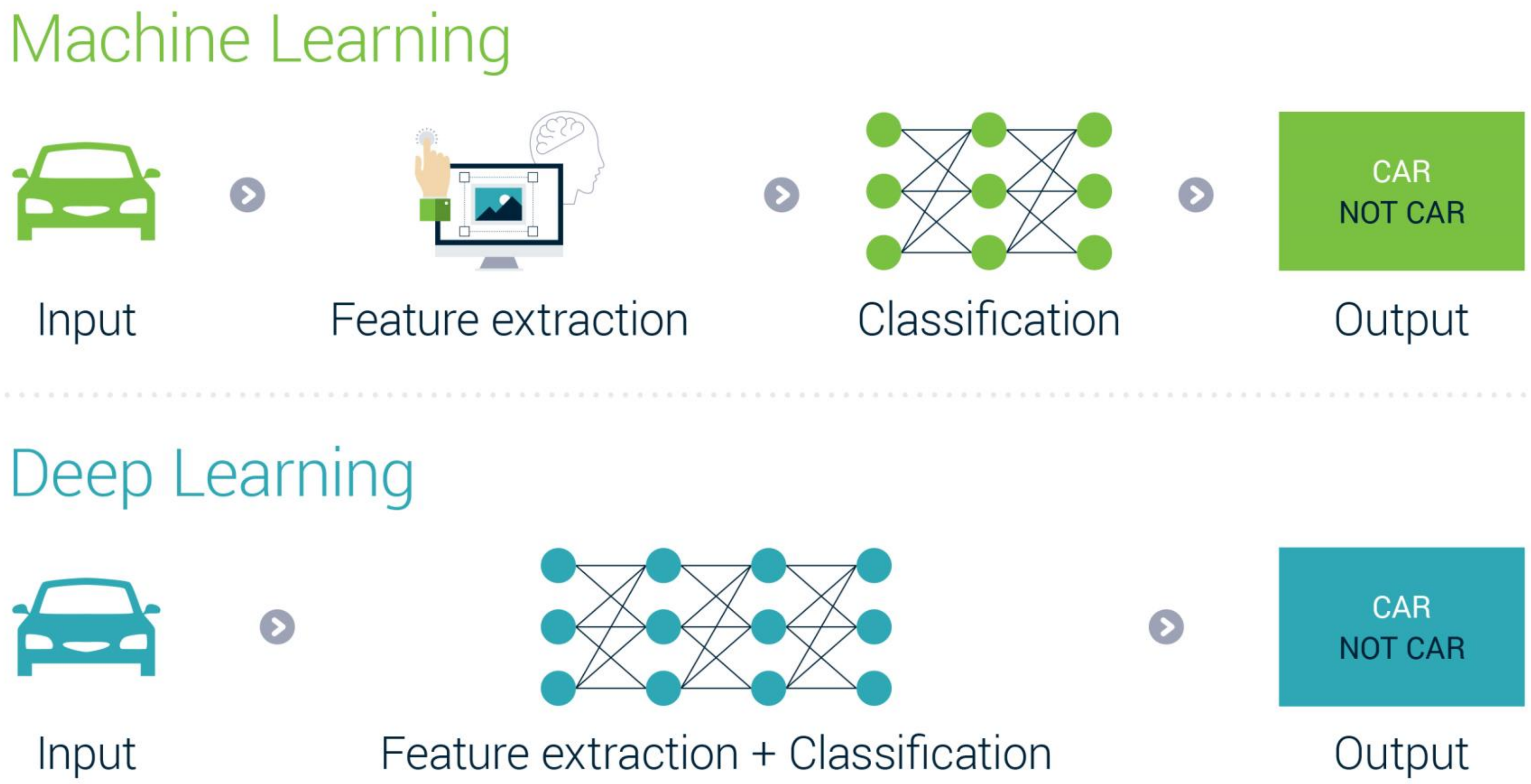
But what's behind the scenes?

Let’s review the basics!


MIMICS HUMAN BEHAVIOUR
AI
ML
PATTERN RECOGNITION ON DATA
DL
USING NEURAL NETWORKS

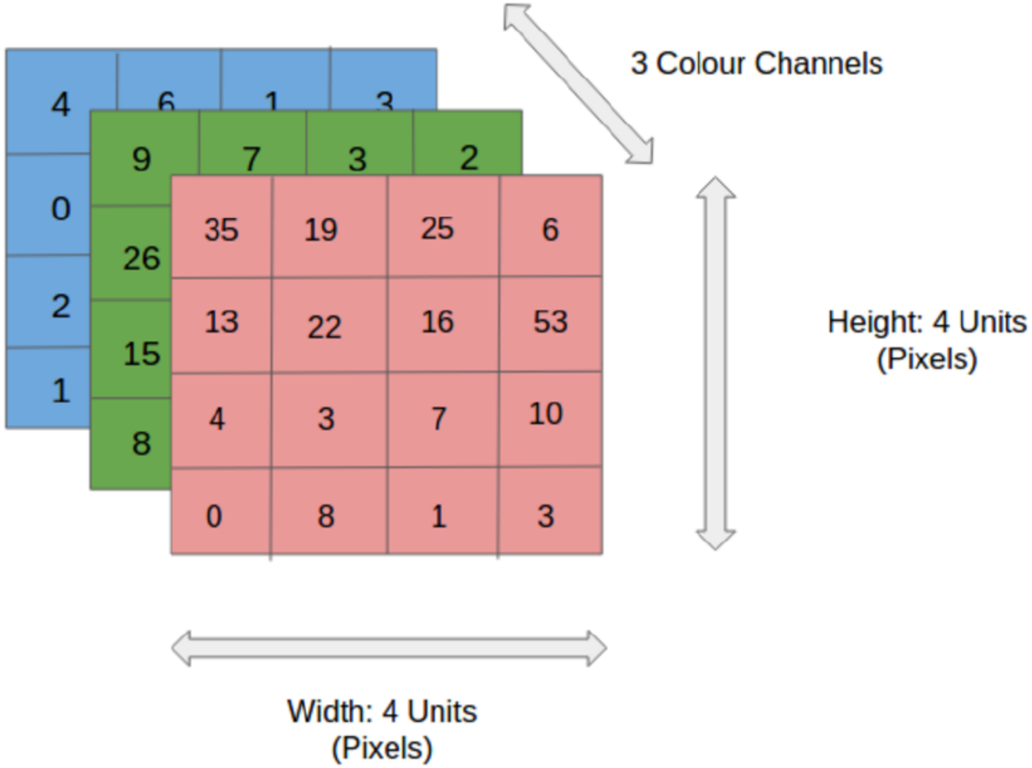
Feature engineering

“Shallow” ML vs. Deep Learning



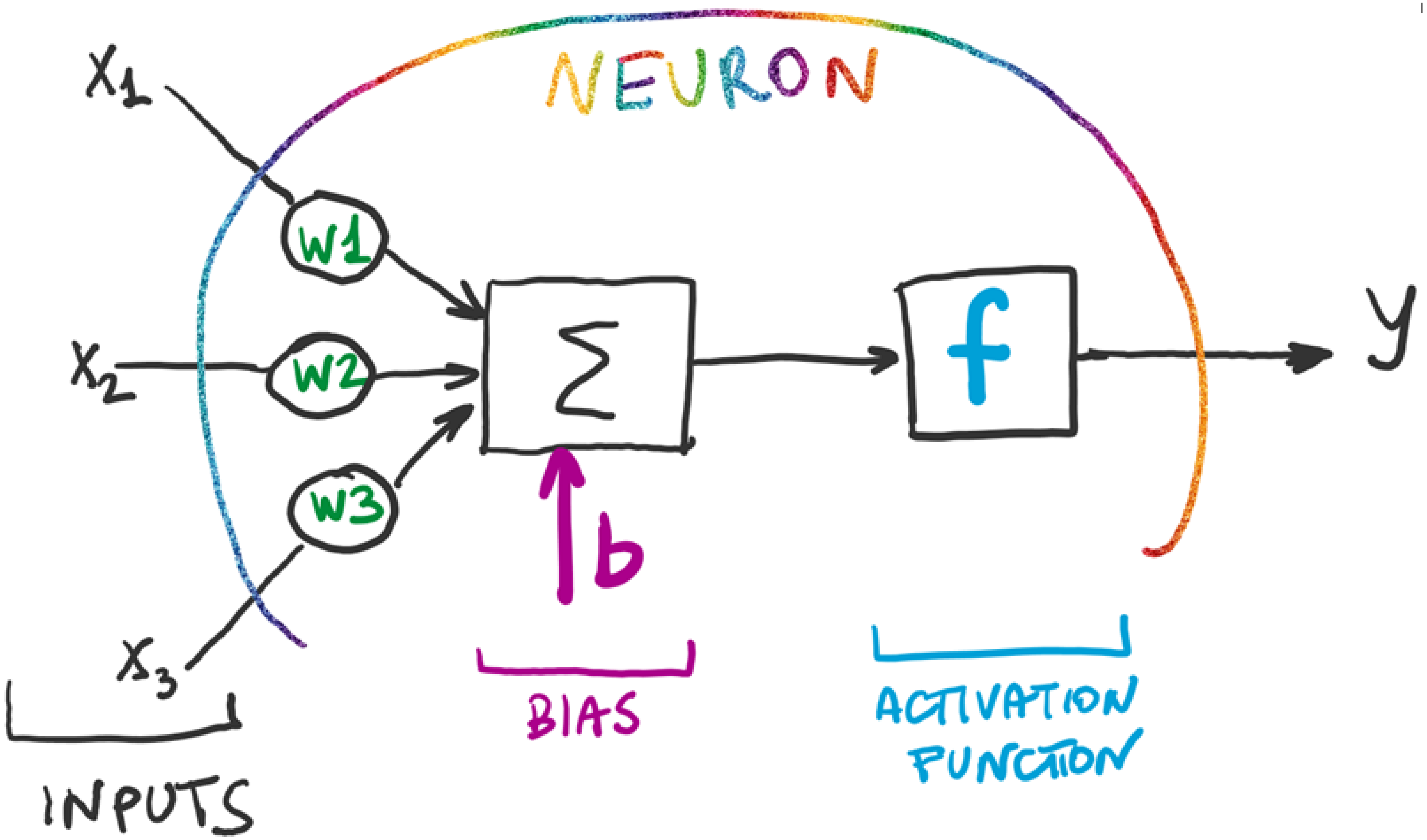
Representations

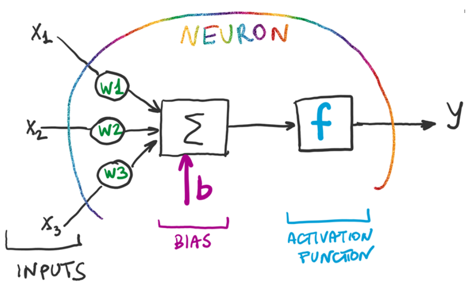
What is an image?



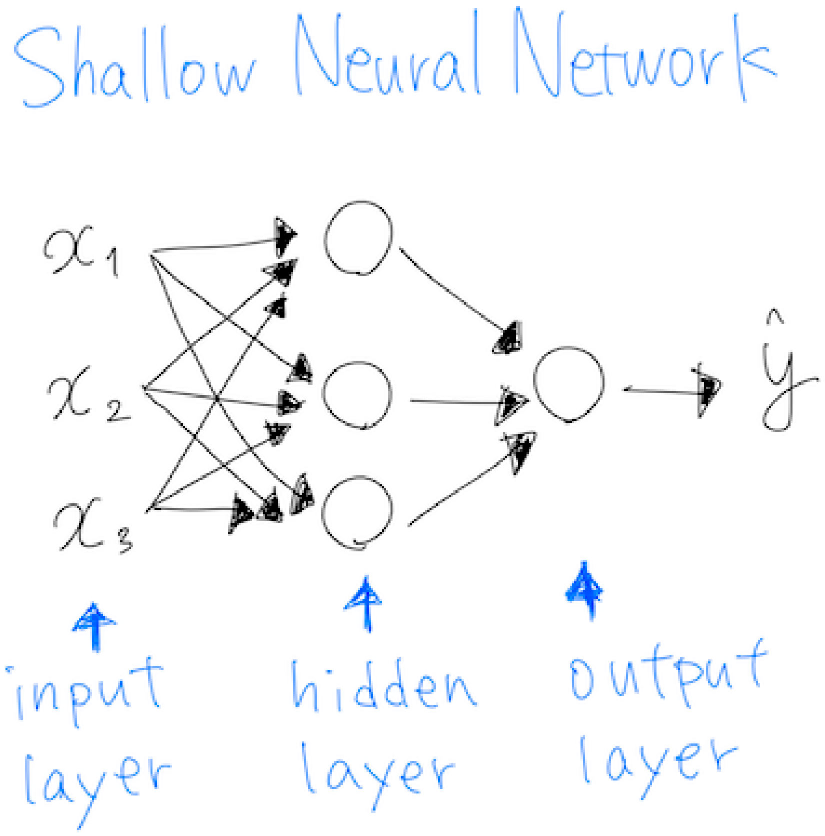
Deep-dive into neural networks

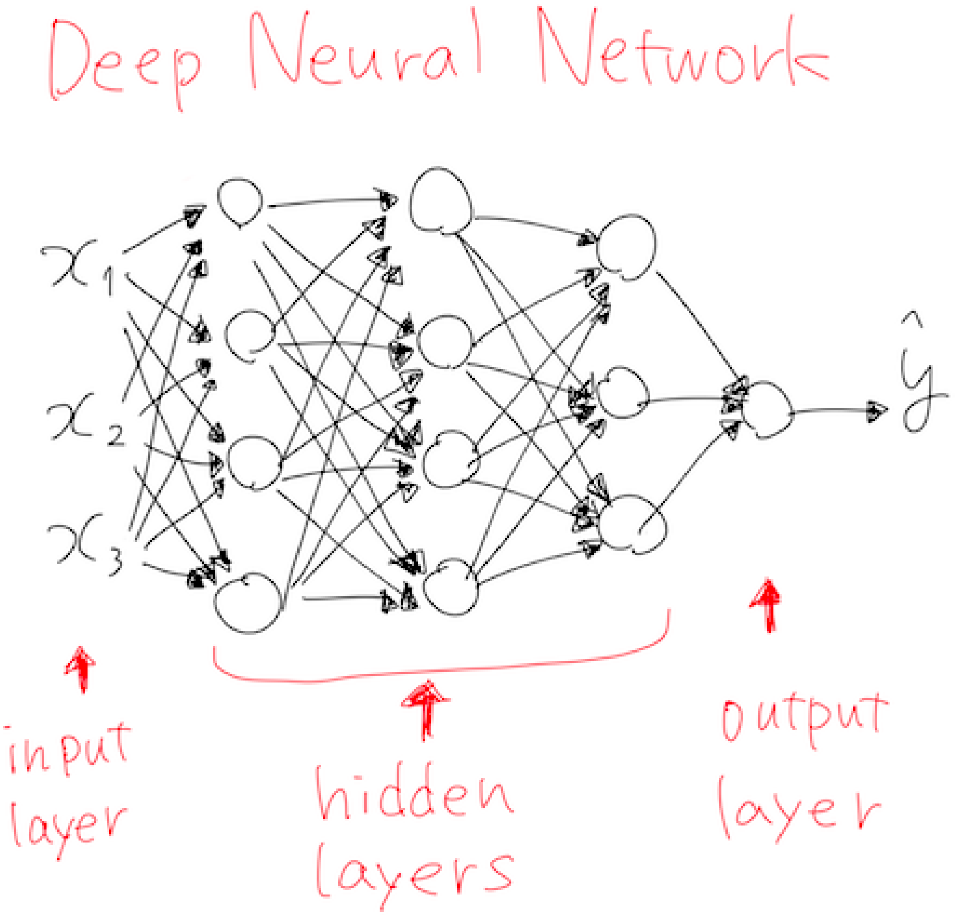
What is a Neuron?



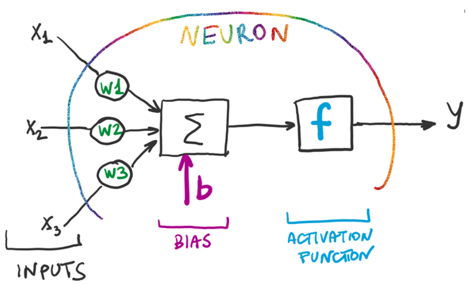


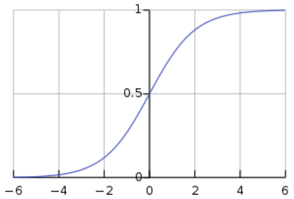
Layers in Neural Network





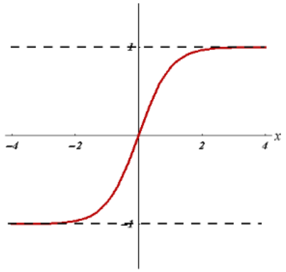
Activation functions

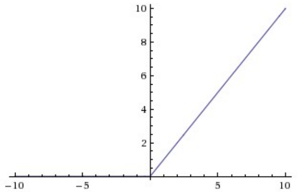




Sigmoid

Tanh





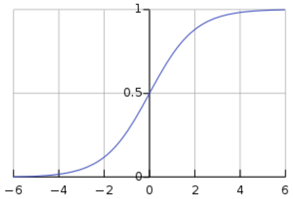
ReLU
(Rectified Linear Unit)

Hidden layer activation functions

- ✓ Continuous & infinite in domain
- ✓ Monotonous and same direction
- ✓ Functions (and their derivatives) can be computed efficiently

NONE

Predict raw data values



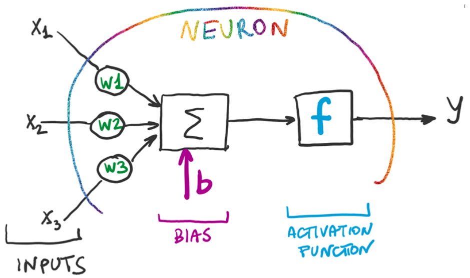
Sigmoid
(Unrelated probs)

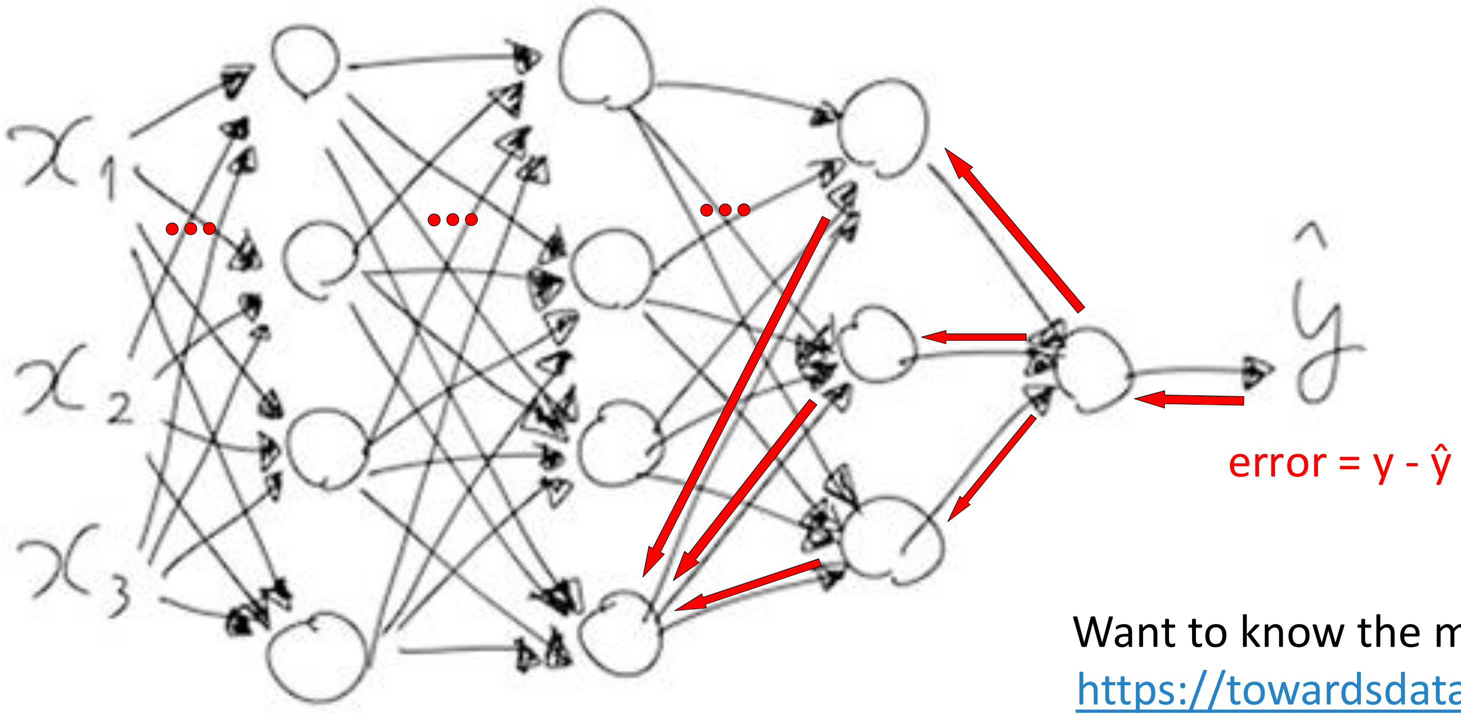
SOFTMAX

Softmax
("which one" prob)

Output layer activation functions

Backpropagation





Want to know the math behind the scene?
https://towardsdatascience.com/learning-backpropagation-from-geoffrey-hinton-619027613f0

Neural Network Architectures

Densely Connected Neural Networks

Convolutional Neural Networks
Spatial relationships (i.e. vision)

Recurrent Neural Networks
Time relationships (i.e. time series, language, speech)

Putting things into practice

TensorFlow
mxnet
GLUON
The Microsoft Cognitive Toolkit
Caffe

torch
PYTORCH
K
theano
Caffe2

And others …

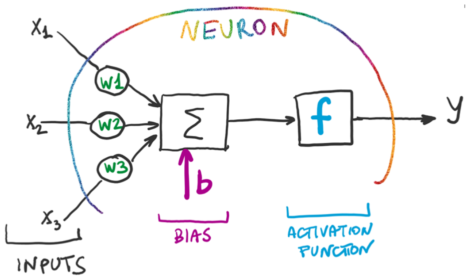
Keras: The Python Deep Learning library

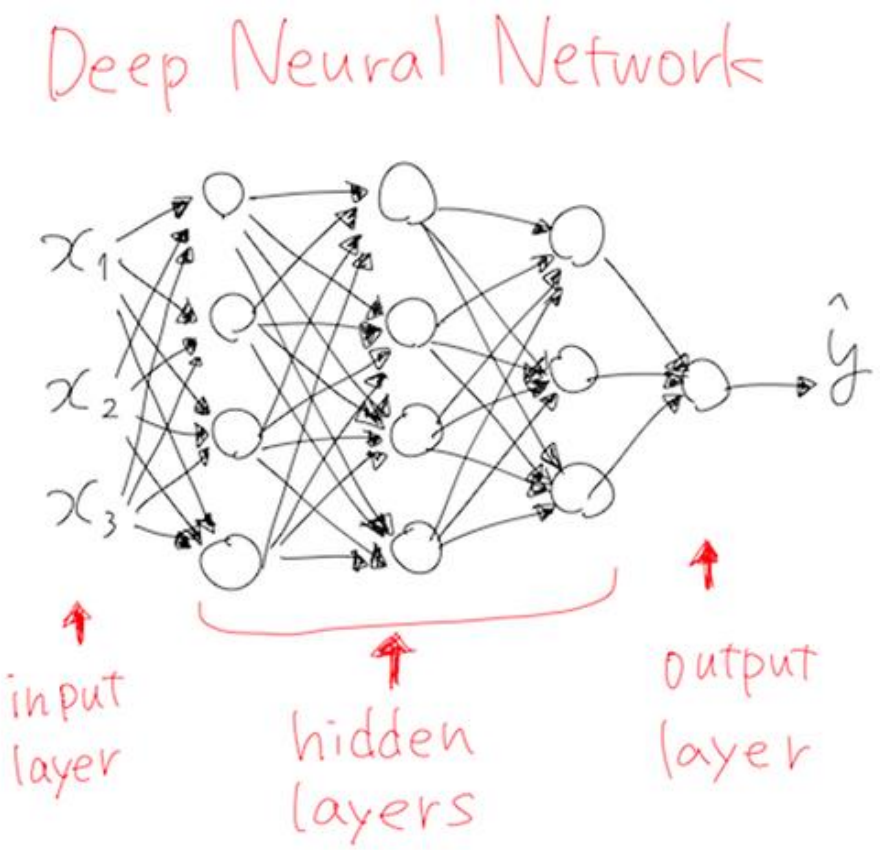
Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Densely Connected Neural Networks



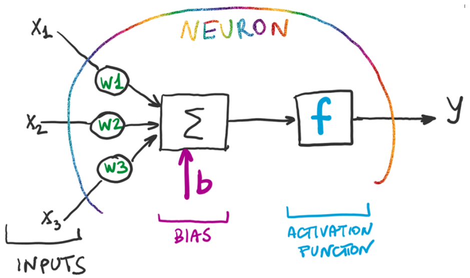


All neurons in any layer fully connected to all activations of previous layer

Number of parameters on each layer is huge

Model's capacity = number of learnable parameters

Densely Connected Neural Networks: MNIST Classification



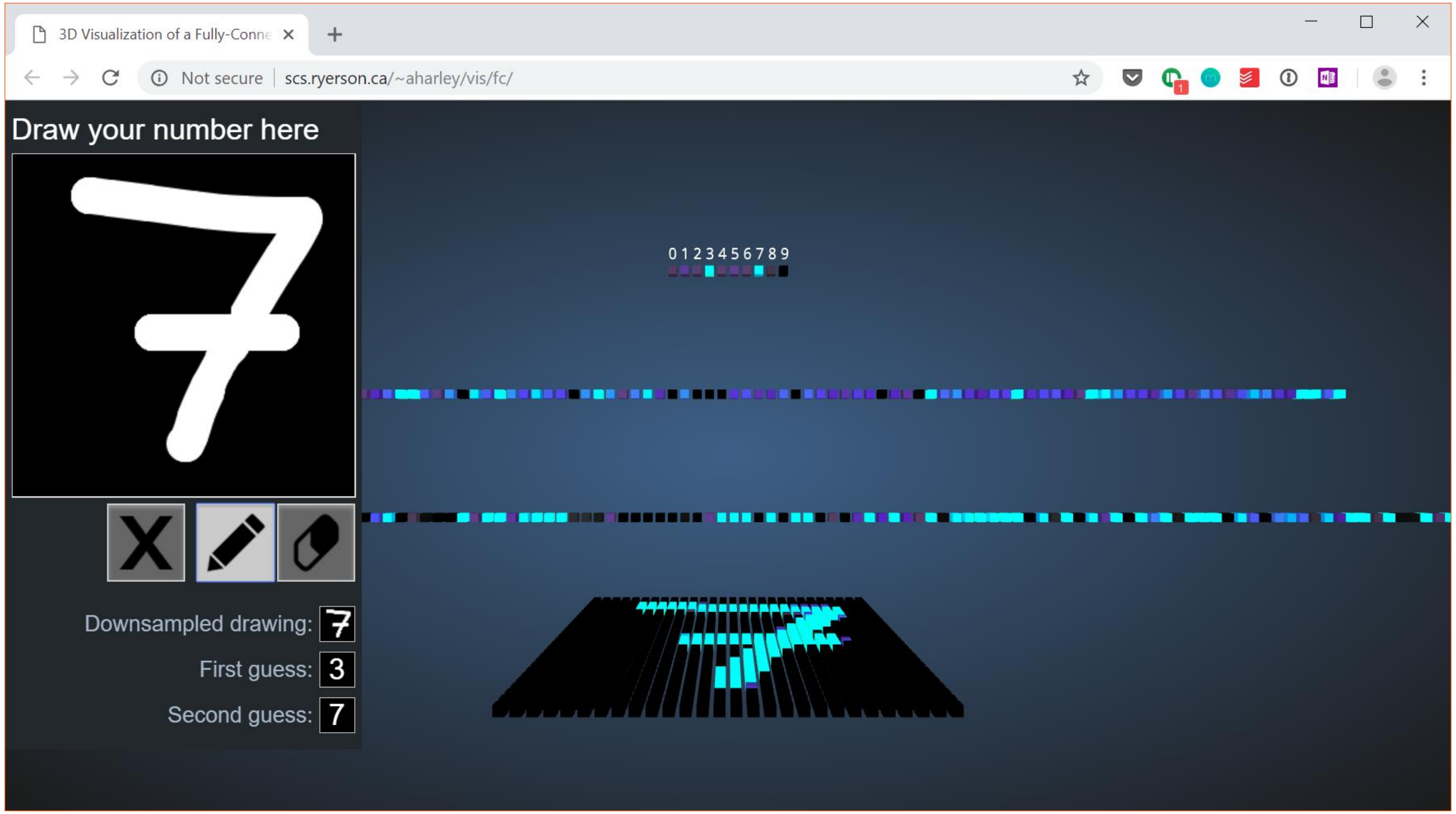
MNIST database (Modified National Institute of Standards and Technology database)

Large database of handwritten digits (28x28)

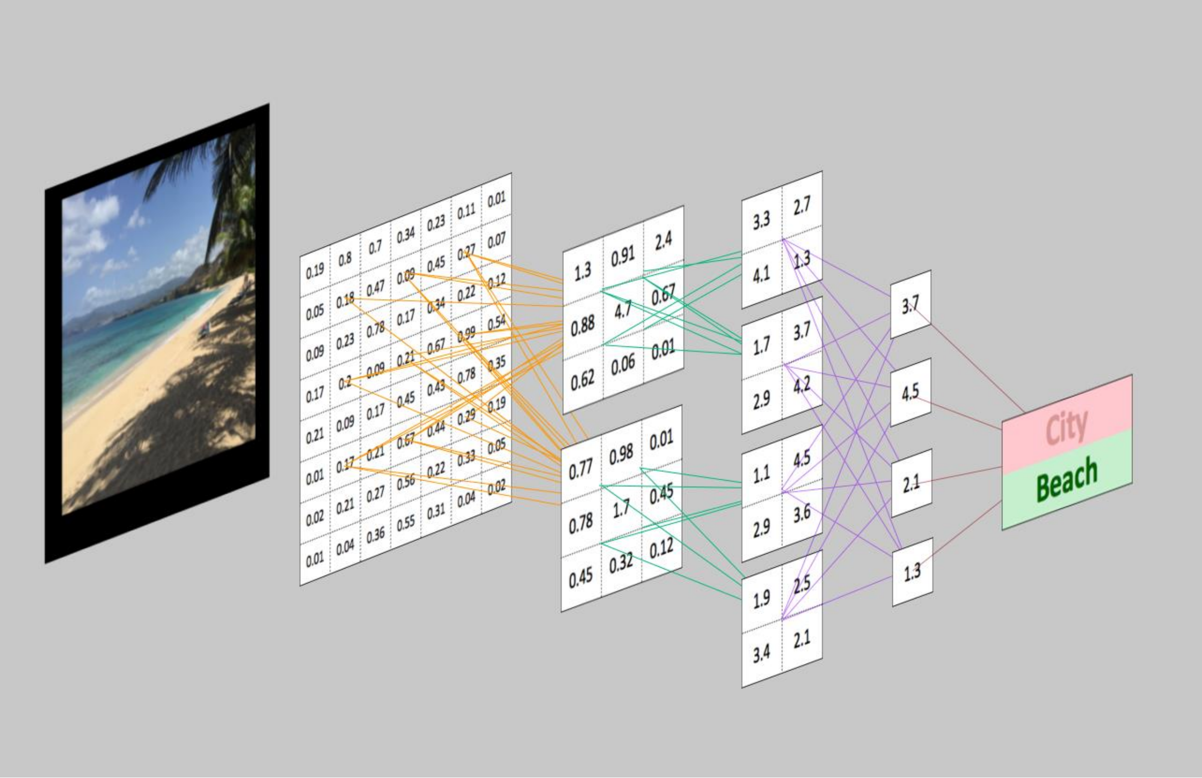
60,000 training and 10,000 testing images

“Hello World” of computer vision tasks

3D Visualization of a Dense Neural Network trained on MNIST dataset



http://scs.ryerson.ca/~aharley/vis/fc/



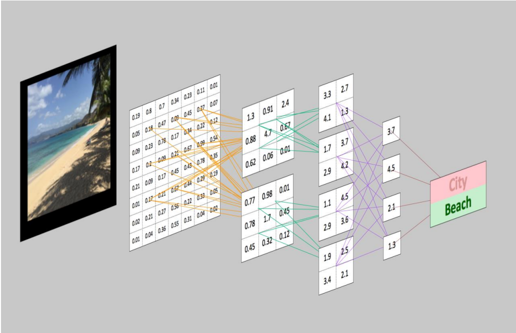
Biological inspiration
* Visual cortex in animals and humans
* Cells are sensitive to small subregions of the input

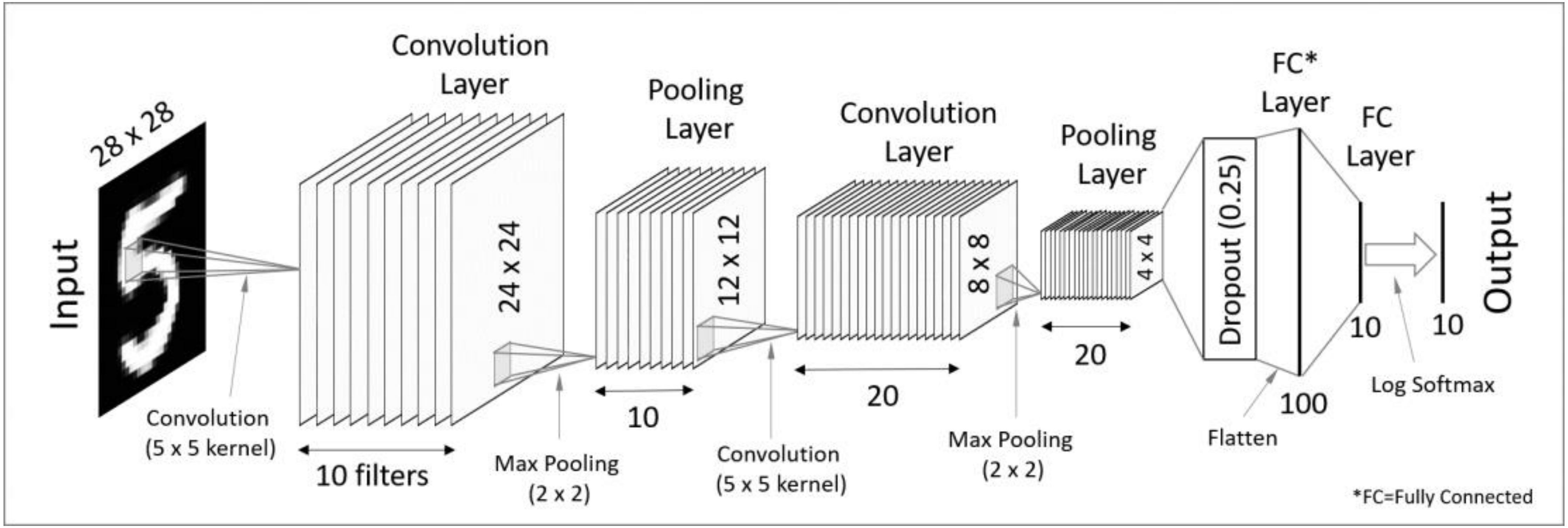
Commonly used in computer vision tasks

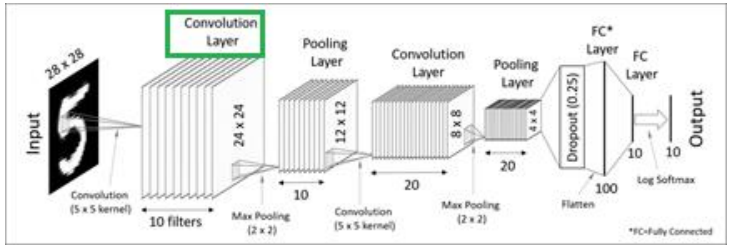
Important to have in data
* Structure to input data
* Spatial relationships
* Repeated patterns

Convolutional Neural Networks (CNNs)

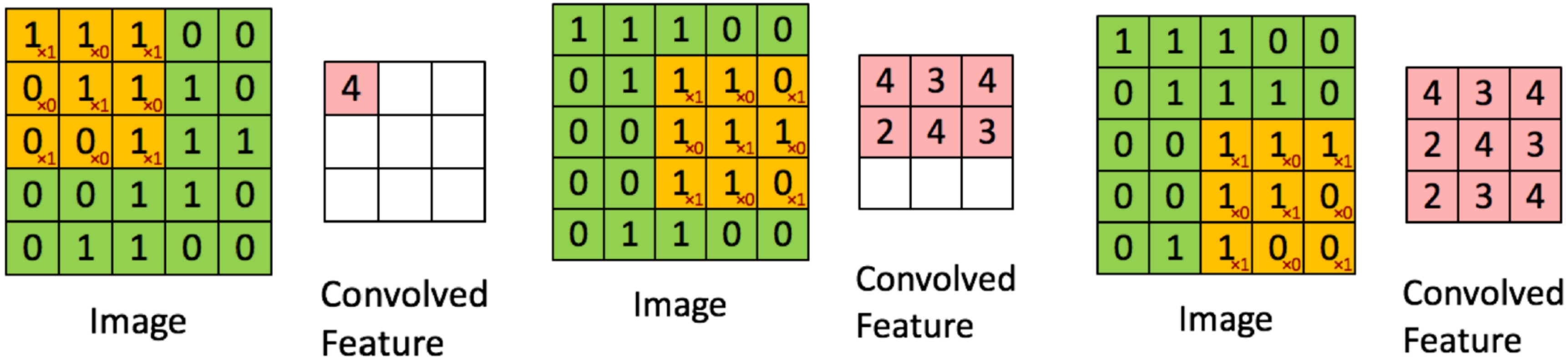
CNNs: Standard architecture layers







CNNs: Convolution Layer

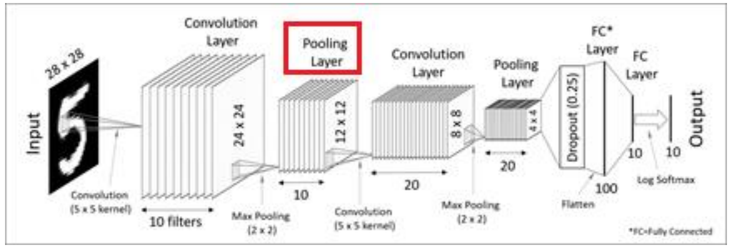


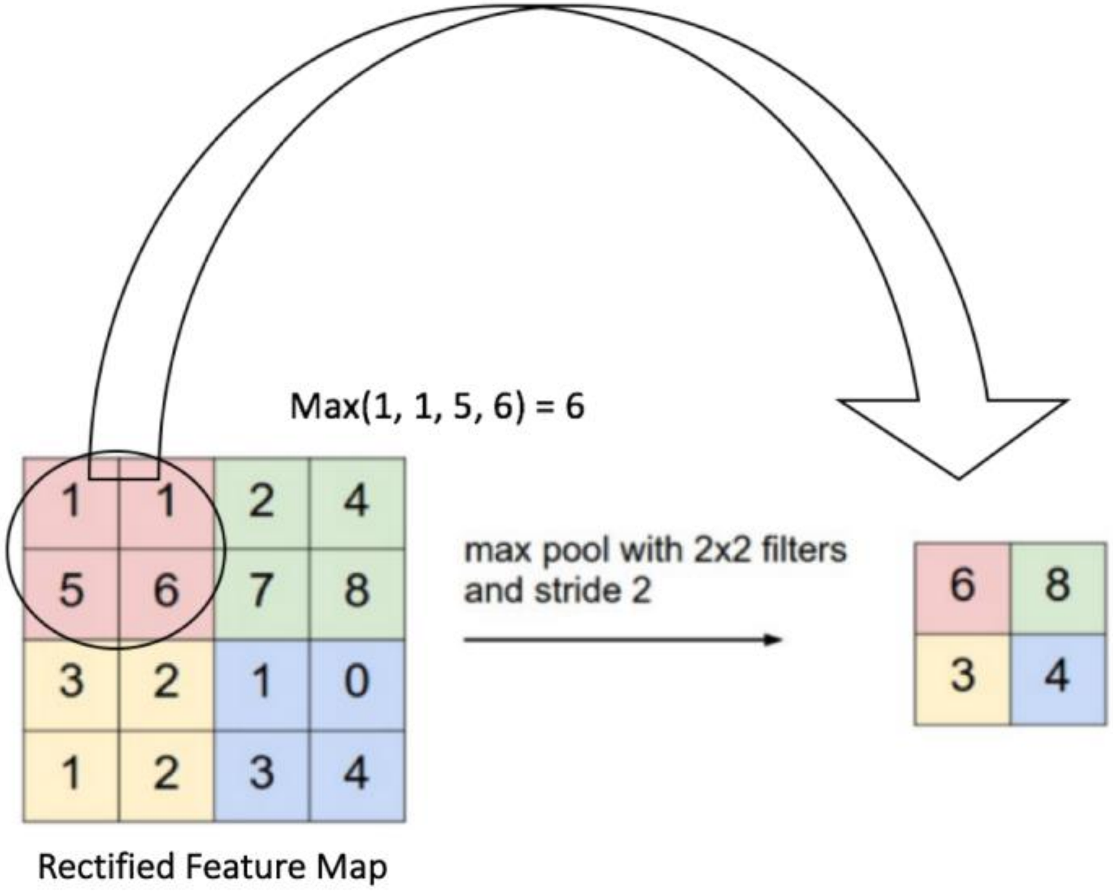
Extract features from input image

Slide filter (3x3, 5x5, etc.) over image to obtain a feature map per filter

Multiple filters

CNNs: Pooling layer

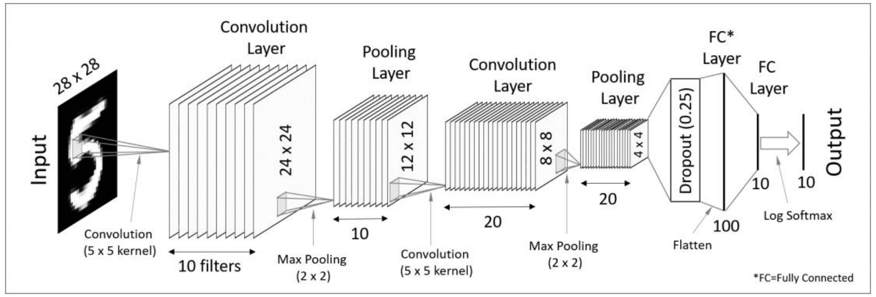




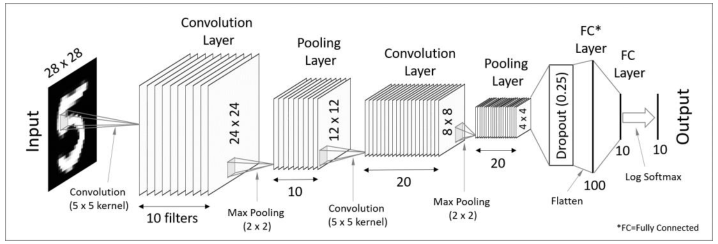
Reduce dimensionality BUT retain most important information

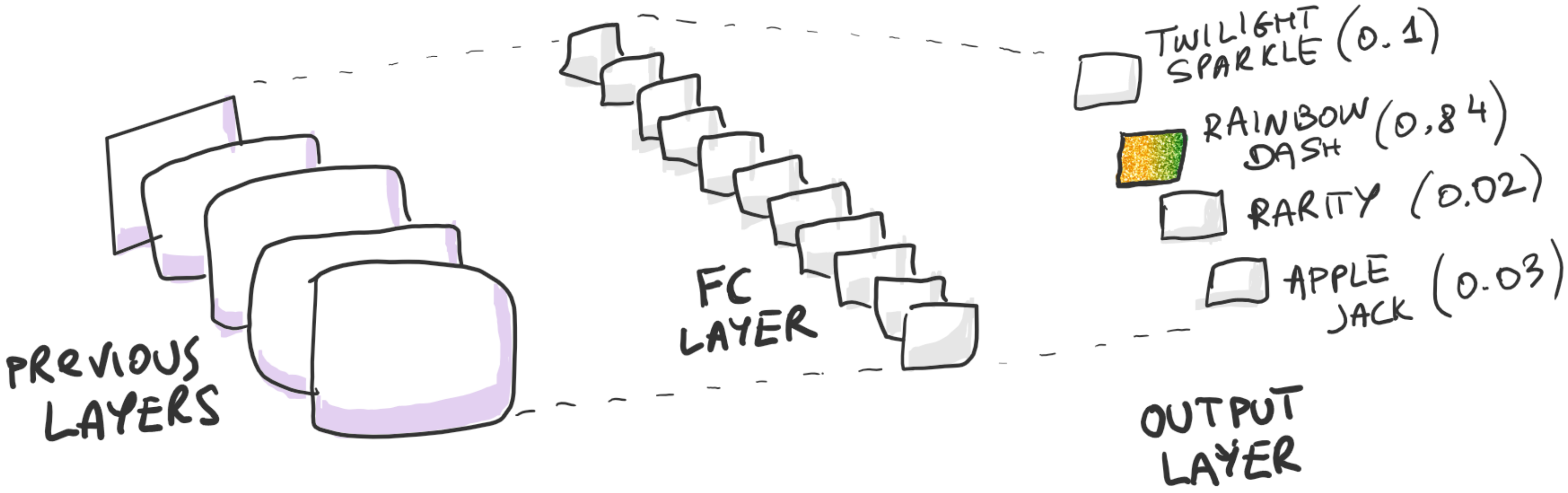
Can be of different types: Max, Avg, Sum

Information distillation

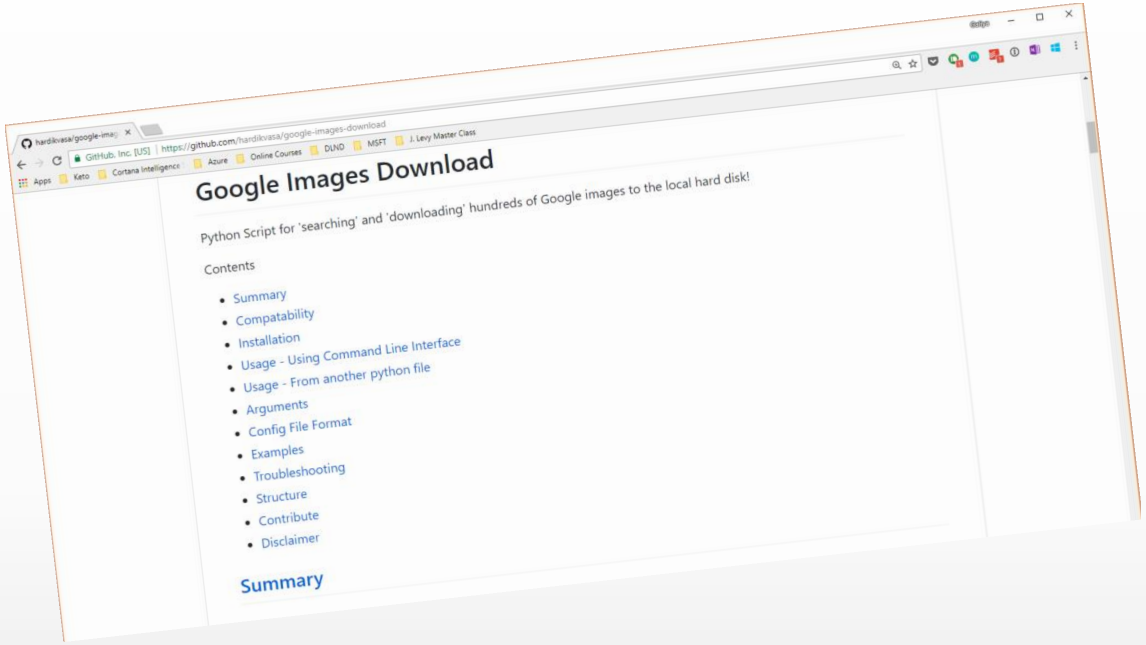


CNNs: Fully connected layer



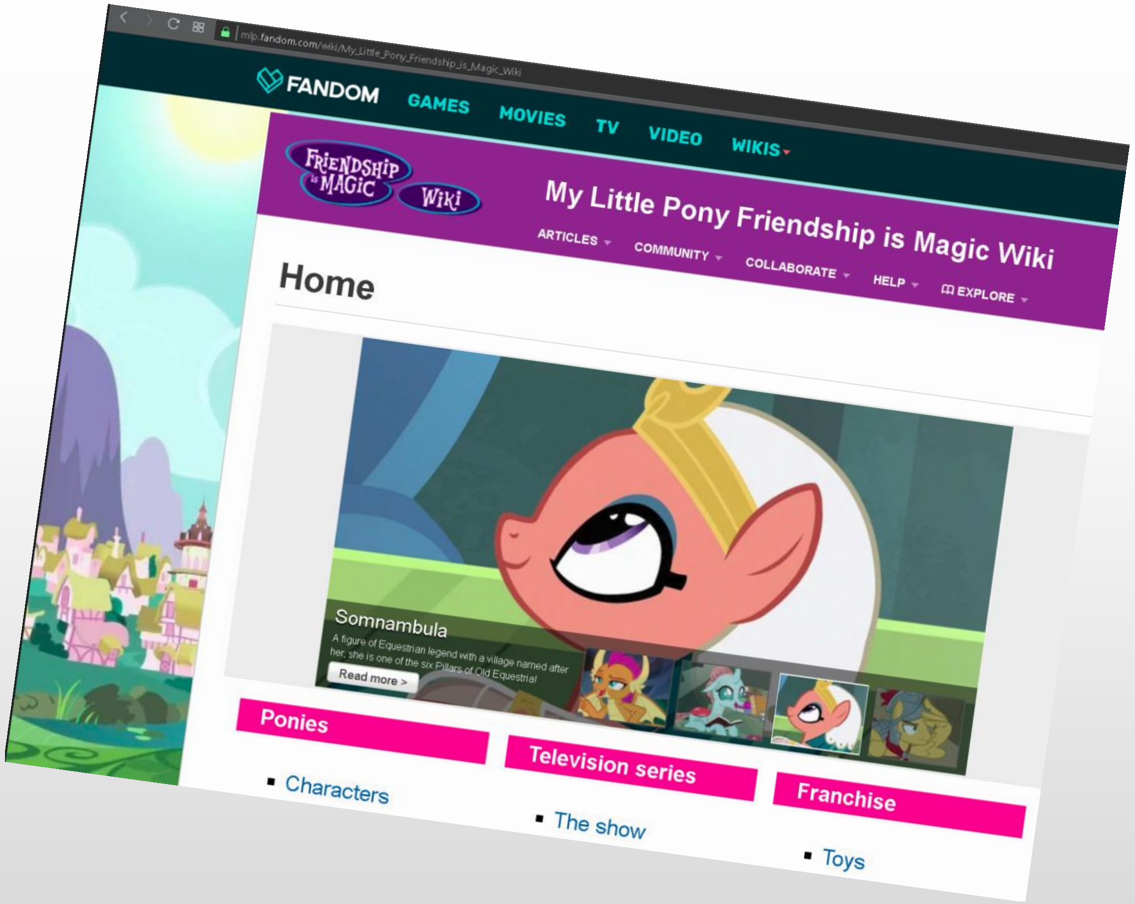


Let’s build it!

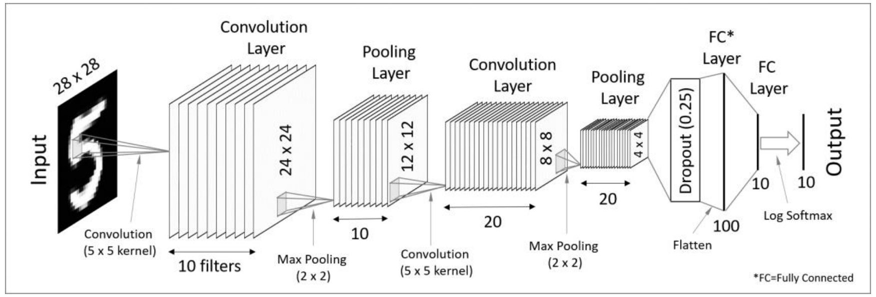


https://github.com/hardikvasa/google-images-download

https://mlp.fandom.com/wiki/My_Little_Pony_Friendship_is_Magic_Wiki



Demo: Build custom pony classifier from scratch



Input

A folder with 3 subfolders of images

150px * 150px each

Output

Number of classes: 3

Names to predict:
[Twilight Sparkle, Rainbow Dash, Apple Jack]

Typical issues in the world of DL

OVERFITTING

SMALL DATA

Overfitting

Any ML models may suffer from it

[Optimisation]
Model can adjust to get the best performance on training data

[Generalisation]
How well model performs on test data it never seen before?

More training data often helps to reduce pure "memorisation"

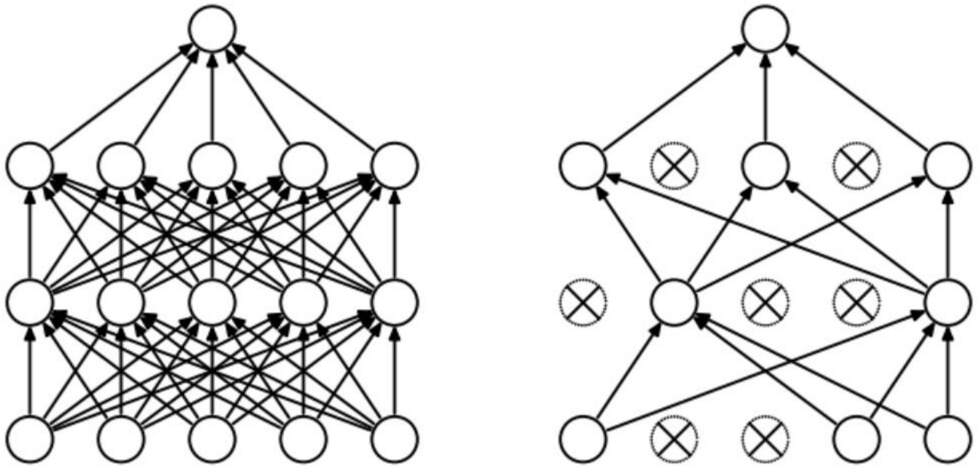
Additional techniques

Data augmentation or “squeeze more training samples from your small dataset”

Apply random transformations (“augment”) to existing training samples
to generate new training data

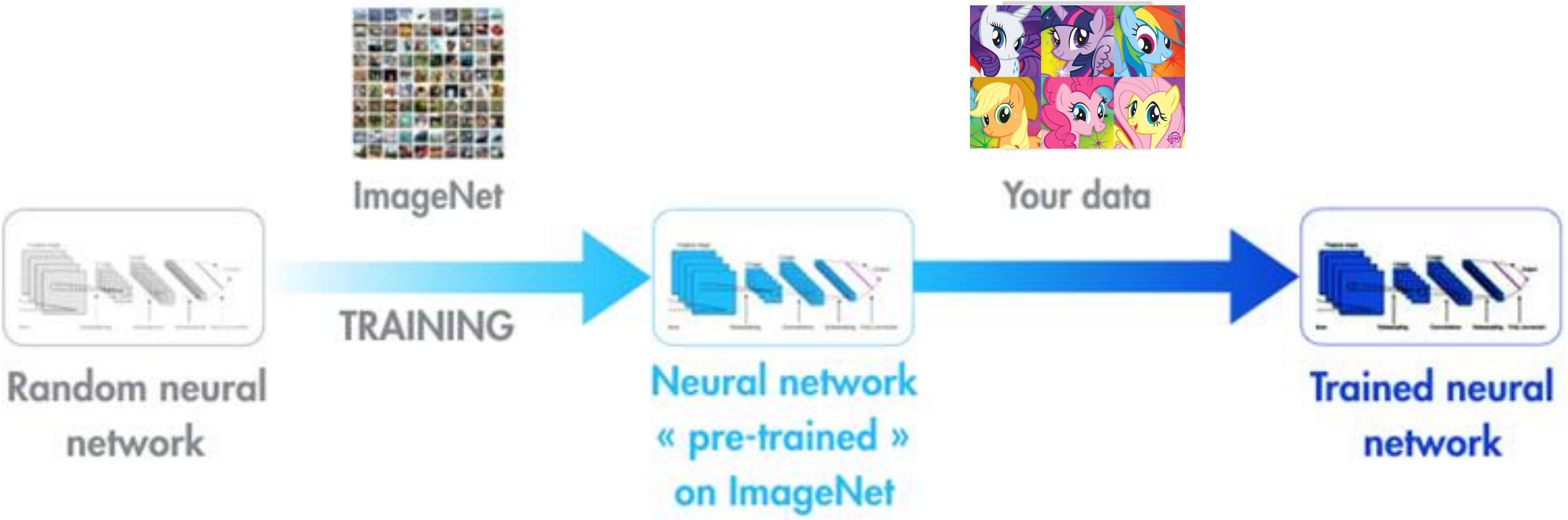
Dropout or “learn from less to learn better”

Learn more robust features

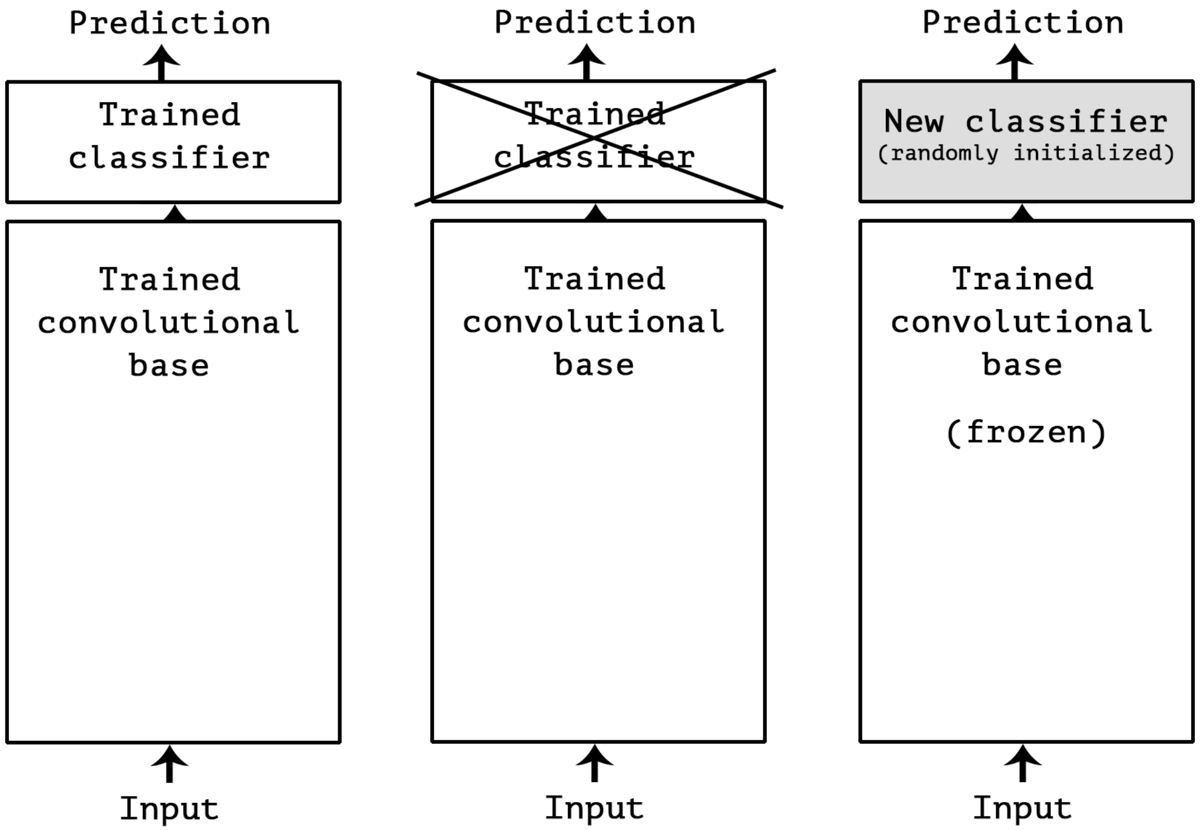


(a) Standard Neural Net (b) After applying dropout.

Transfer learning



Pre-trained models: Feature extraction

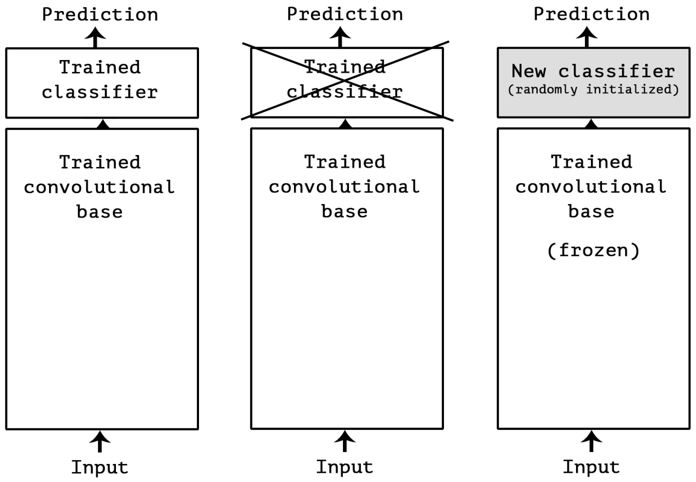


Pre-trained models: Feature extraction (Code)

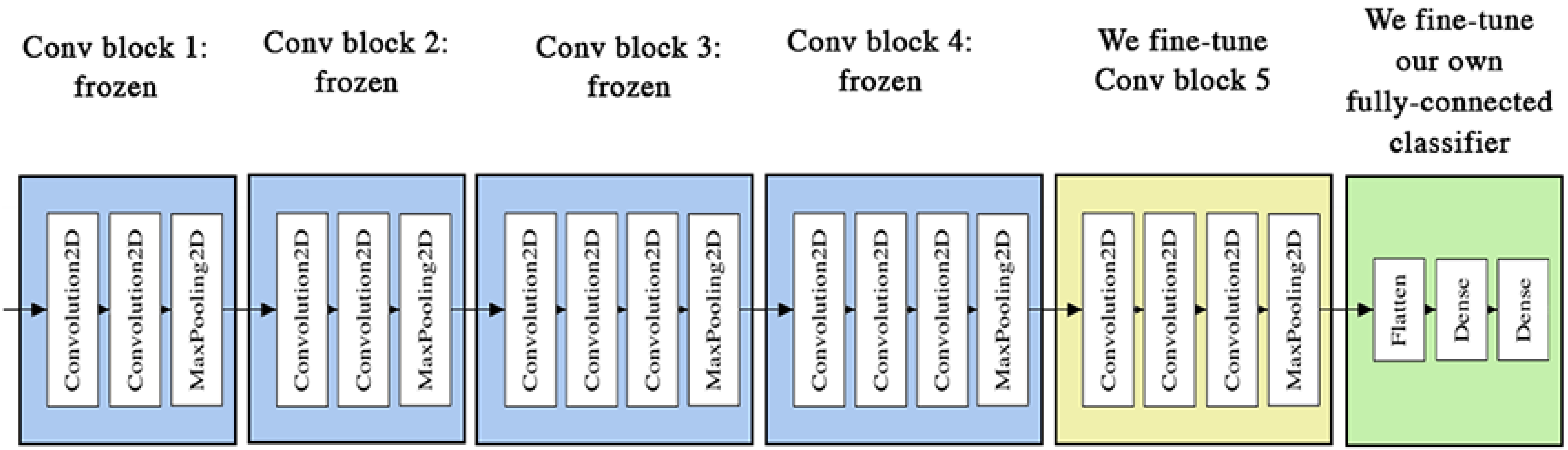
```
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input, decode_predictions, VGG16
import numpy as np
import os

conv_base = VGG16(weights = 'imagenet', include_top = False, input_shape=(150,150,3))
```

```
from keras import models
from keras import layers
conv_base.trainable = False
model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))
```



Pre-trained models: Fine-tuning



Demo: Use pre-trained models for classification task

Input

A number of personal images

Various sizes

Output

Image class (as per pre-trained model)

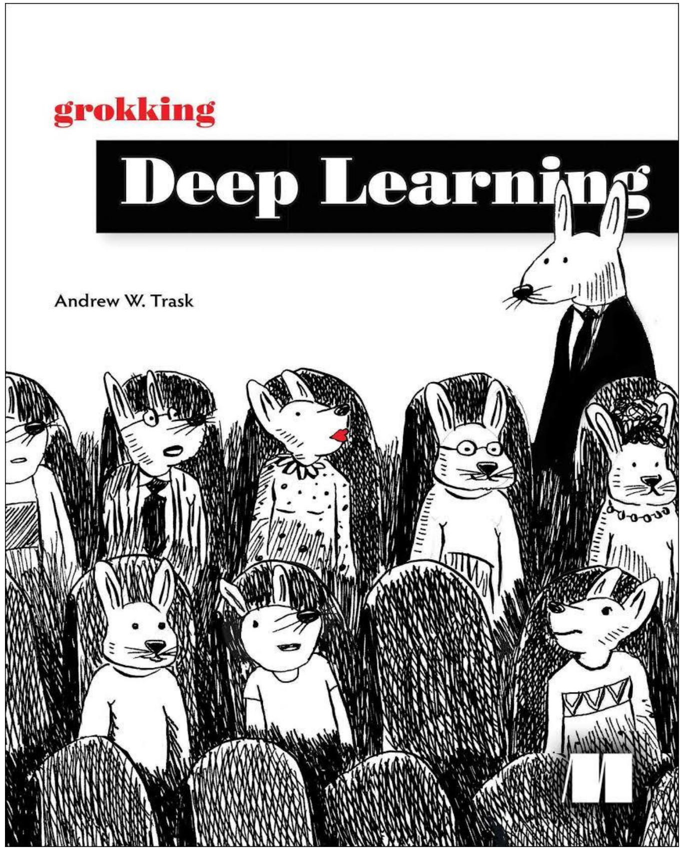
Probability

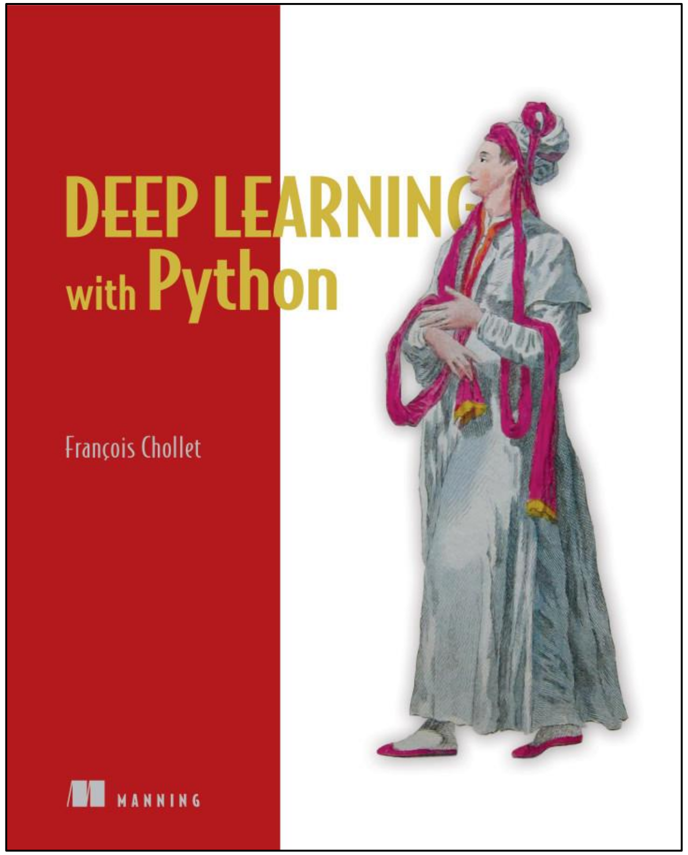
DEMO

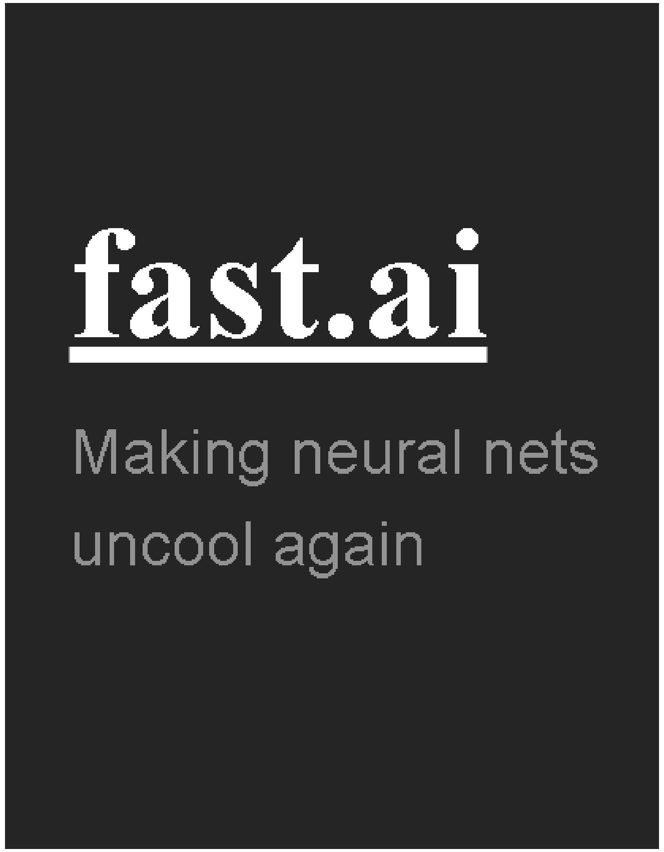
Takeaways

- Only interested in API? Look at [Custom Vision Service]
- Building your own classifiers?
  - Have good dataset to work with (with lots of samples!)
  - Start with Keras
  - You will need access to GPU (your own / on the cloud)
- Custom Convolutional Neural Networks:
  - Start with naïve approach “from scratch” or use pre-trained models
  - Apply data augmentation
  - Introduce dropout layers

Takeaways: Resources







What’s next?


POKéMON

PAW
PATROL

Thank you!

@galiyawarrier

https://github.com/galiya/DeepLearningPonies