

Deep Learning: Frameworks and Applications

Jerome Nilmeier

IBM: Developer Advocate

Center for Open-source Data and AI Technologies

Naval Postgraduate School

Emerging and Disruptive Technologies

08 Nov 2019



About Me

Jerome Nilmeier

2015- Present:

Data Scientist and Developer Advocate,

2015-2017: IBM Spark Technology Center

2017-Present: IBM CODAIT

Research Background

2002: B.S. Chemical Engineering

UC Berkeley

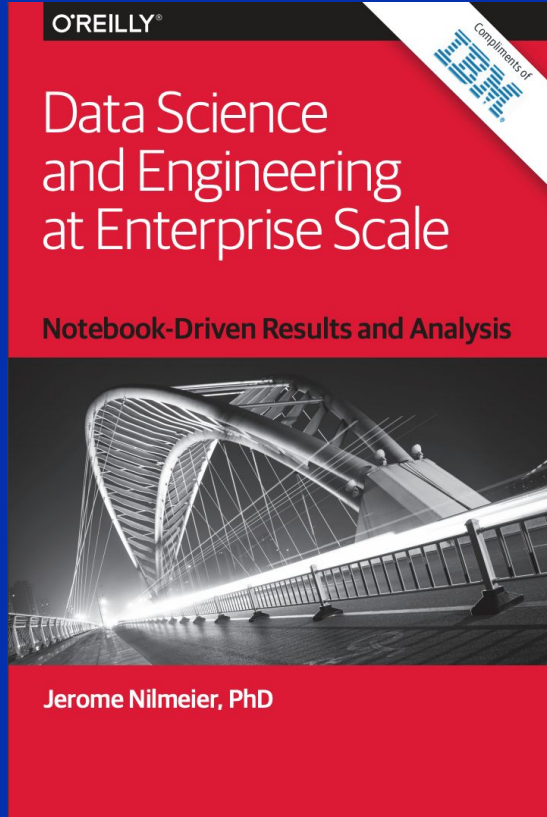
2008: Ph.D. (Computational Biophysics), UC San Francisco

2009-2015: Postdoctoral Appointments:

- UC Berkeley/ Lawrence Berkeley National Lab
- Lawrence Livermore National Lab
- Stanford OpenMM Fellow
- Insight Data Engineering Fellow



About Me

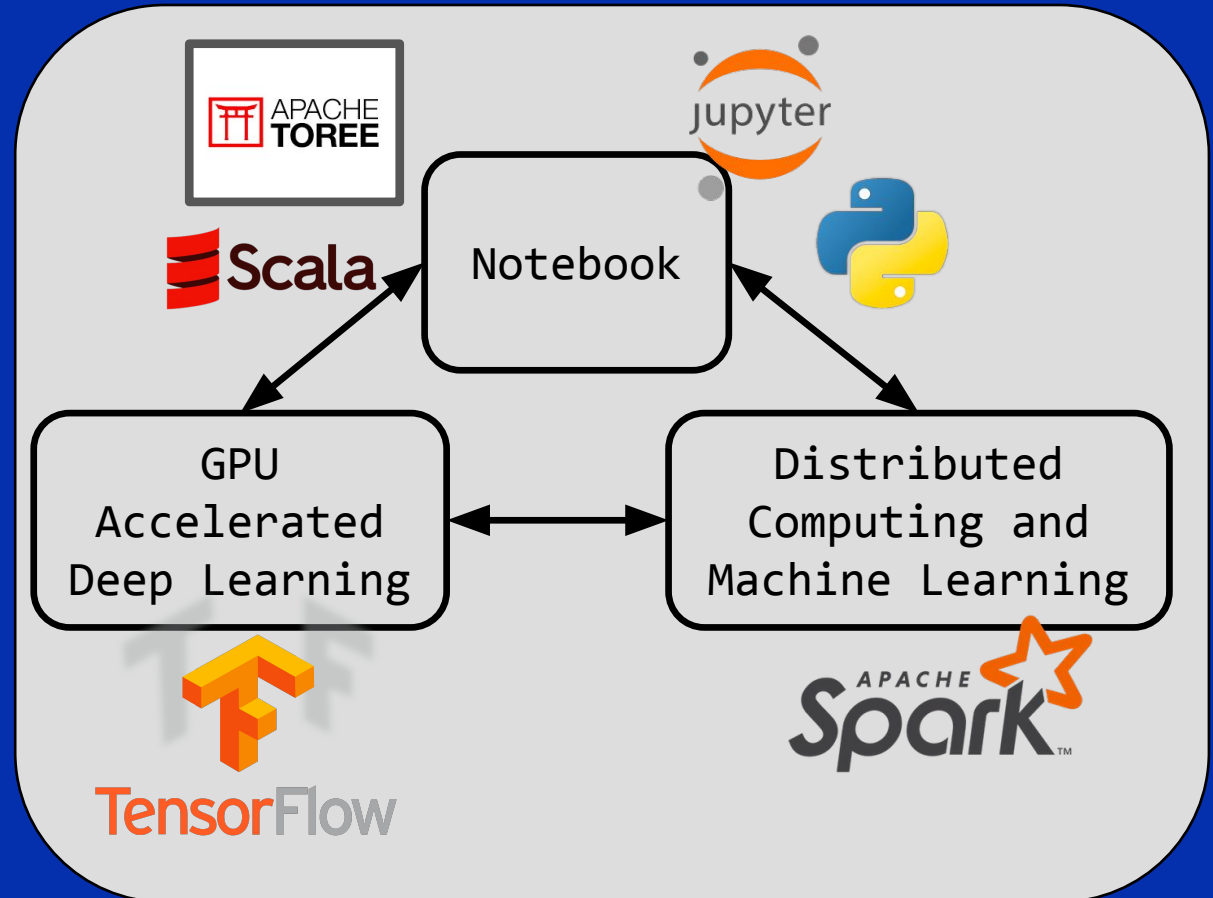
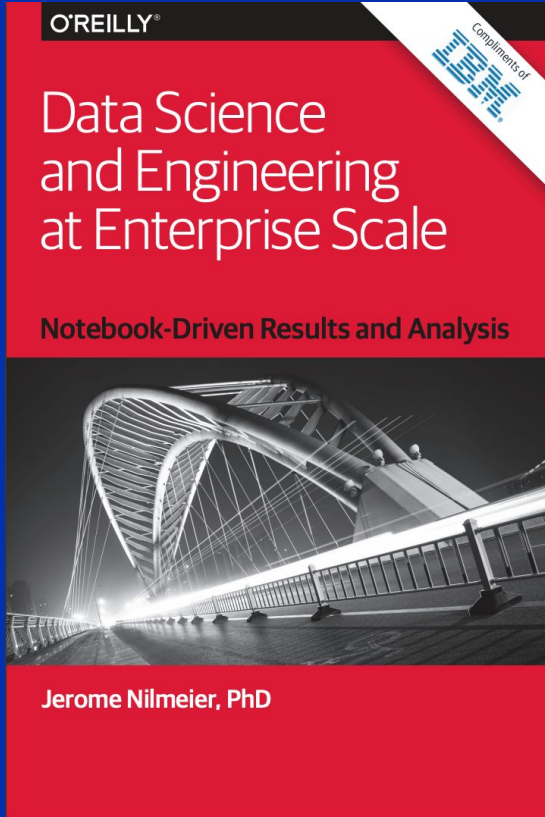


Free Book!
bit.ly/DSatES-pdf

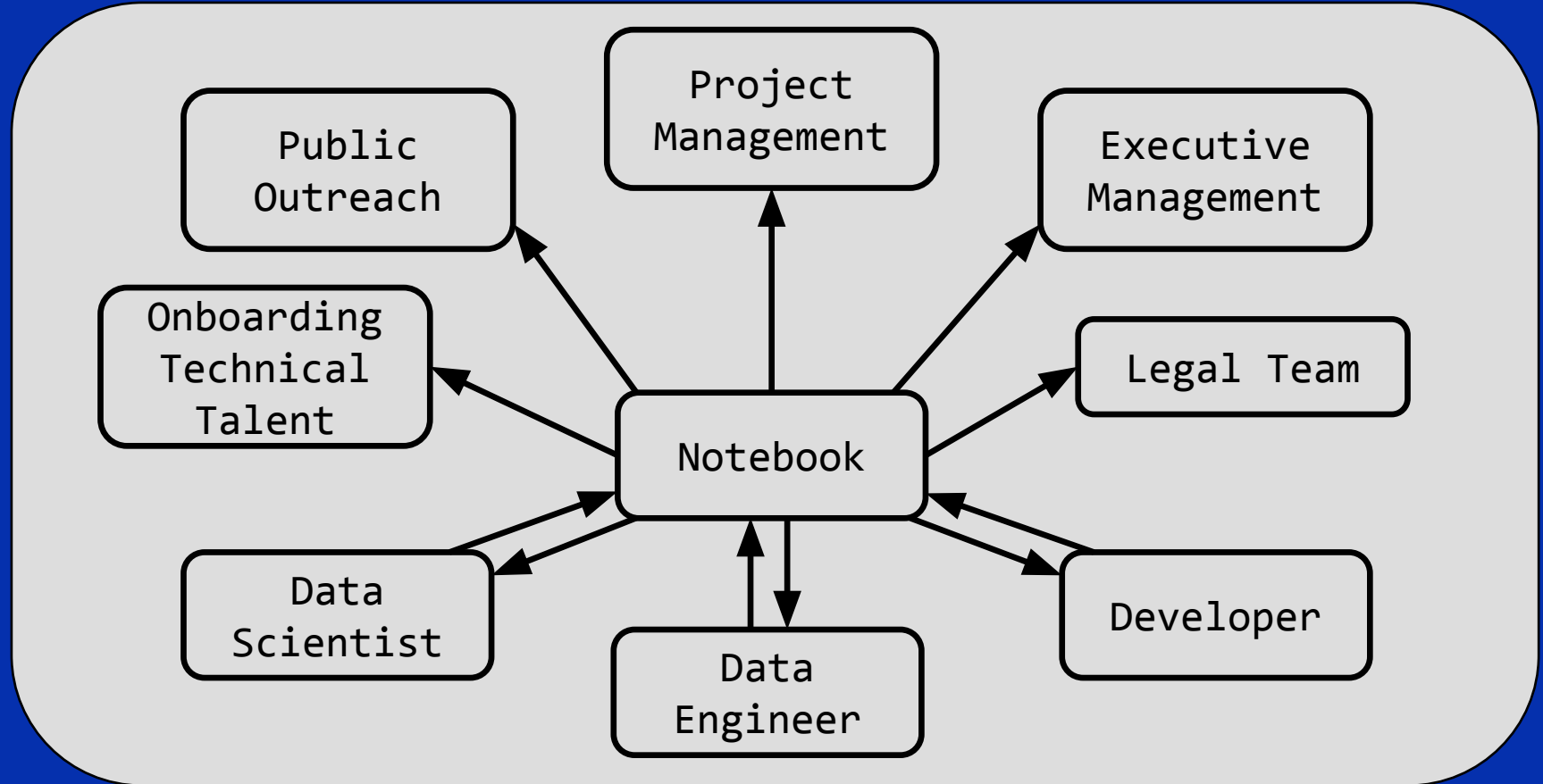
<https://github.com/nilmeier/DSatEnterpriseScale>

TensorFlow 2.0 Examples
<http://ibm.biz/TF2p0-Notebooks-NPS>

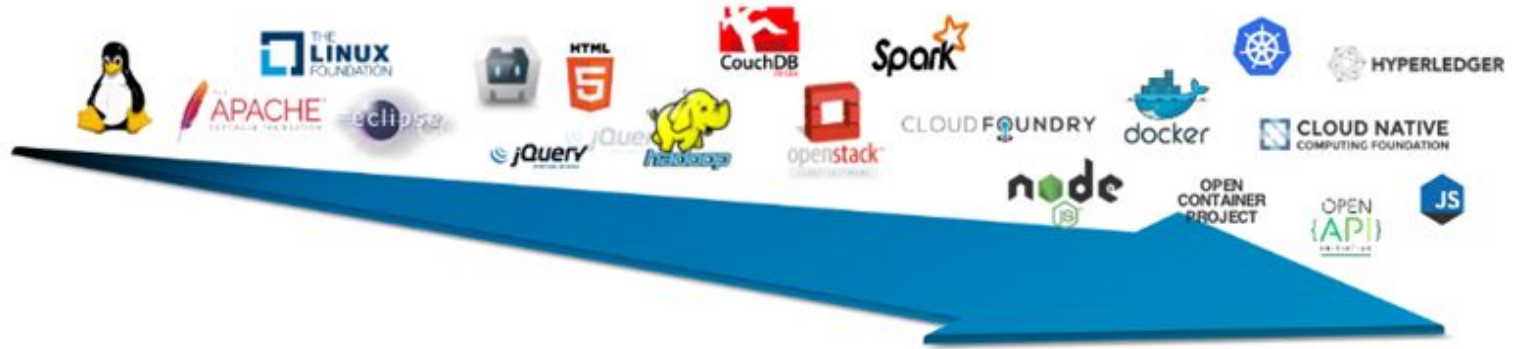
Educational Material: Courses, Workshops, Textbooks



Educational Material: Courses, Workshops, Textbooks



IBM's Long History with Open Source and open standards



Center for Open Source Data and AI Technologies

CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise

Relaunch of the Spark Technology Center (STC) to reflect expanded mission



CODAIT

codait.org

codait (French)
= coder/coded

<https://m.interglot.com/fr/en/codait>



IBM Developer Model Asset eXchange

Free, open-s
Wide variety
Multiple dee
Vetted and t
Build and de
seconds.
Start training
(FIDL) or Wa
minutes.

IBM Code Model Asset Exchange

A place for developers to find and use free and open source deep learning models.

The screenshot displays the IBM Developer Model Asset Exchange interface. At the top, there's a navigation bar with 'IBM Developer', 'Topics', 'Community', and 'More open source at IBM'. A search bar is on the right. The main content area features a teal header for the 'MODEL' section, followed by the title 'Facial Age Estimator' and a description: 'Recognize faces in an image and estimate the age of each face.' Below this are two buttons: 'Get this model' and 'Try the API'. The left sidebar contains a navigation menu with categories like 'Artificial Intelligence', 'CODE', 'CONTENT', 'COMMUNITY', and 'RELATED'. The right sidebar shows social media links and a 'CONTENTS' list. At the bottom, there are three model cards: 'Generate English-language text similar to the text in the Yelp® review data set.', 'Categorize sports videos according to which sport the video depicts.', and 'Protect communications with adversarial neural cryptography.' Each card has a 'Get this model' button.

IBM Developer Topics Community More open source at IBM Search

Artificial Intelligence

MODEL

Facial Age Estimator

Recognize faces in an image and estimate the age of each face.

Get this model Try the API

By IBM Developer Staff | Last updated September 25, 2018

Artificial Intelligence Vision

Overview

This repository contains code to instantiate and deploy a facial age estimation model.

SOCIAL

CONTENTS

- Overview
- Model Metadata
- References
- Licenses

Options available for deploying

Generate English-language text similar to the text in the Yelp® review data set. Get this model

Categorize sports videos according to which sport the video depicts. Get this model

Protect communications with adversarial neural cryptography. Get this model

Sample Generator

Short audio clips of speech commands and lo-fi samples. Get this model

Caption Generator

Captions that describe the contents of images. Get this model

Serial Cryptography

IBM Developer Model Asset eXchange

Free, open-source deep learning models.

Wide variety of domains.

Multiple deep learning frameworks.

Vetted and tested code and IP.

Build and deploy a [web service](#) in 30 seconds.

Start training on [Fabric for Deep Learning \(FfDL\)](#) or [Watson Machine Learning](#) in minutes.

IBM Code Model Asset Exchange

A place for developers to find and use free and open source deep learning models.

Featured models



Facial Age Estimator

Recognize faces in an image and estimate the age of each face.

[Get this model](#)



Weather Forecaster

Predict hourly weather features given historical data for a specific location.

[Get this model](#)



Audio Sample Generator

Generate short audio clips of speech commands and lo-fi instrumental samples.

[Get this model](#)

All models



Inception-ResNet-v2

Identify objects in images using a third-generation deep residual network.

[Get this model](#)



Scene Classifier

Classify images according to the place/location labels in the Places365 data set.

[Get this model](#)



Image Caption Generator

Generate captions that describe the contents of images.

[Get this model](#)



Review Text Generator

Generate English-language text similar to the text in the Yelp® review data set.

[Get this model](#)



Sports Video Classifier

Categorize sports videos according to which sport the video depicts.

[Get this model](#)



Adversarial Cryptography

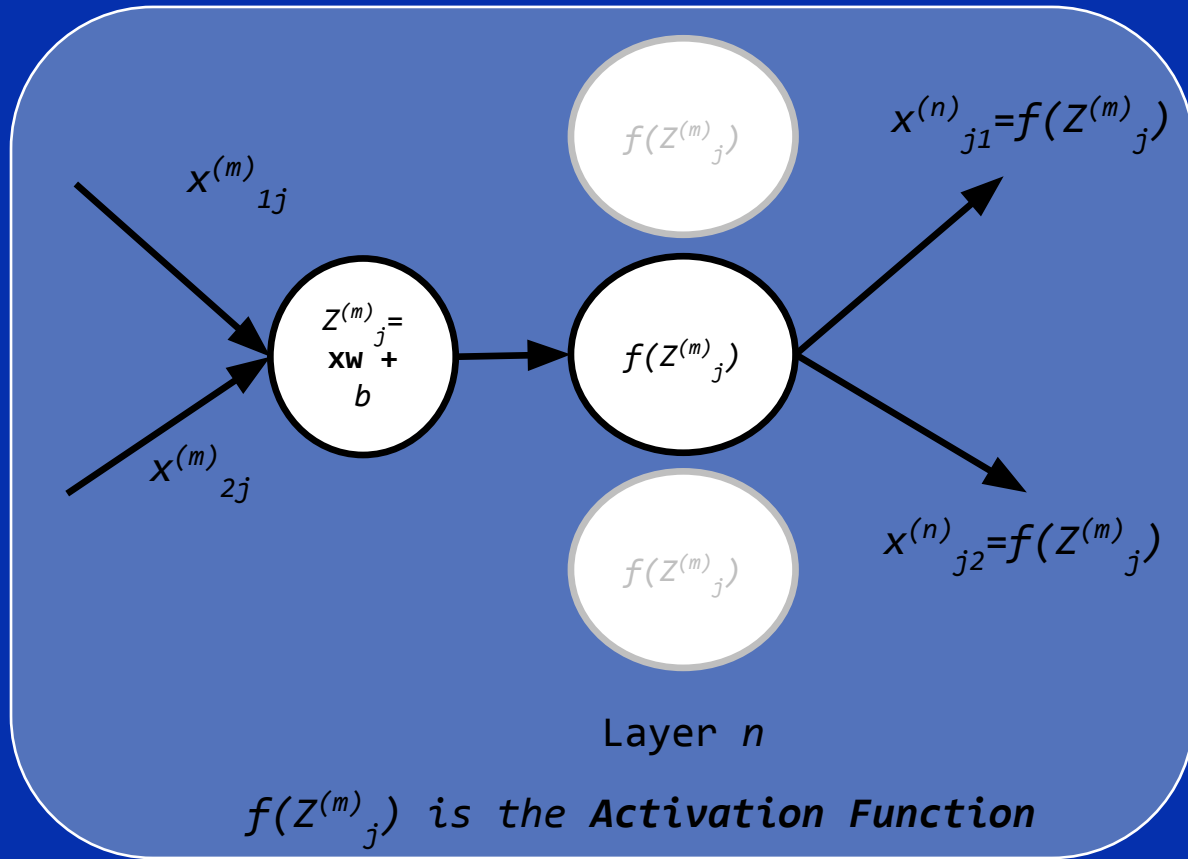
Protect communications with adversarial neural cryptography.

[Get this model](#)

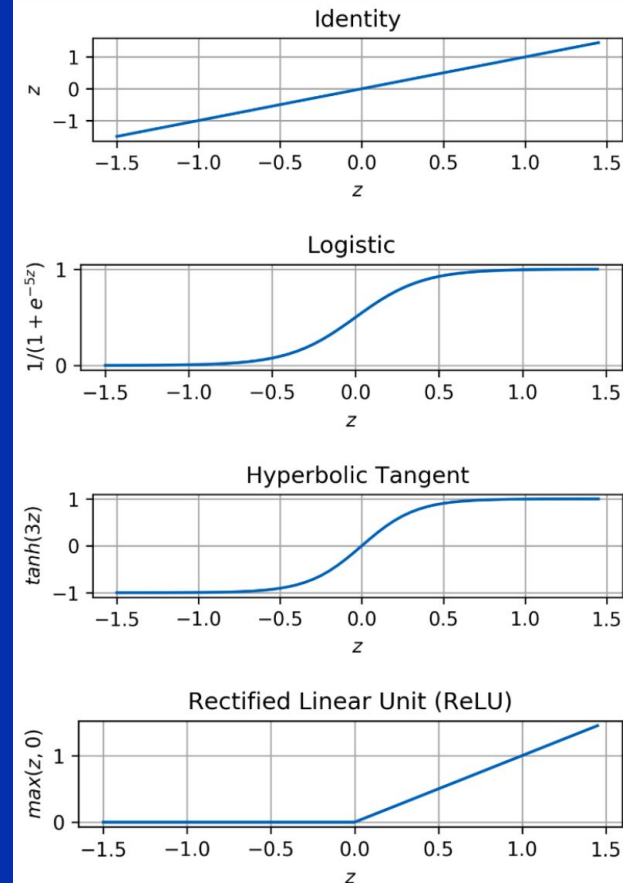
Neural Network Architectures Primer



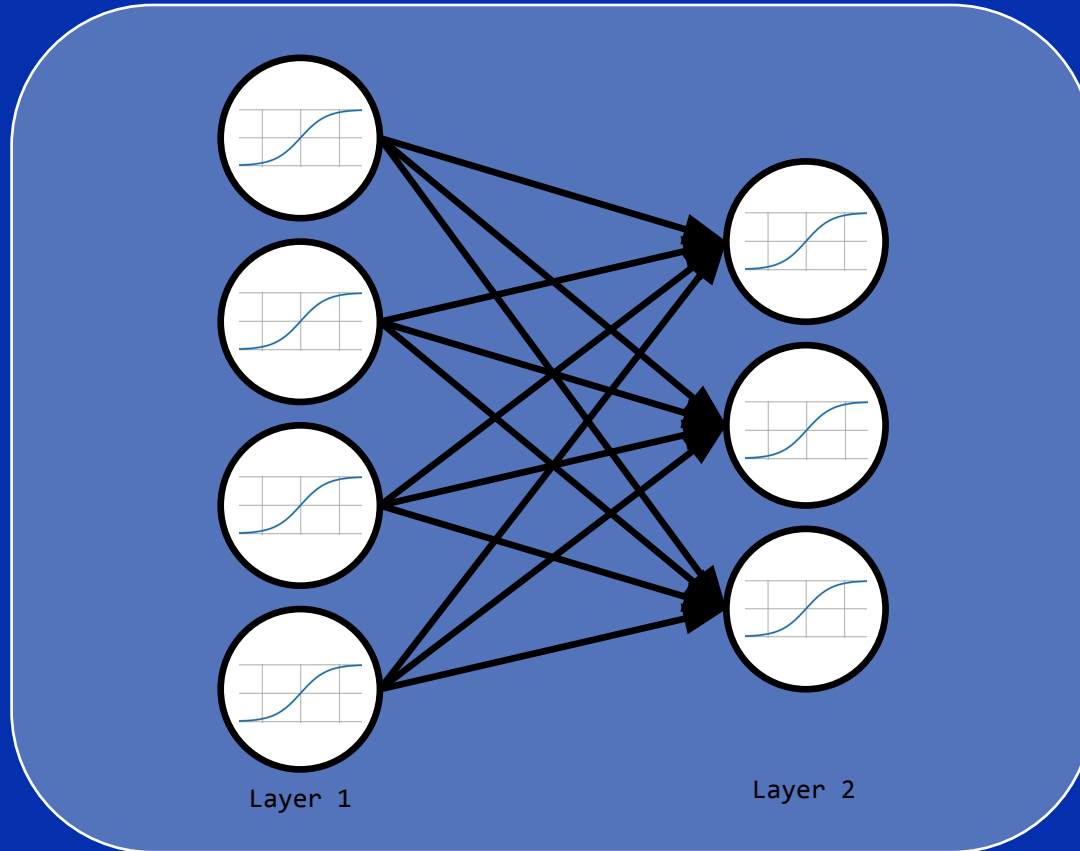
Neurons: The Basic Unit of a Neural Network



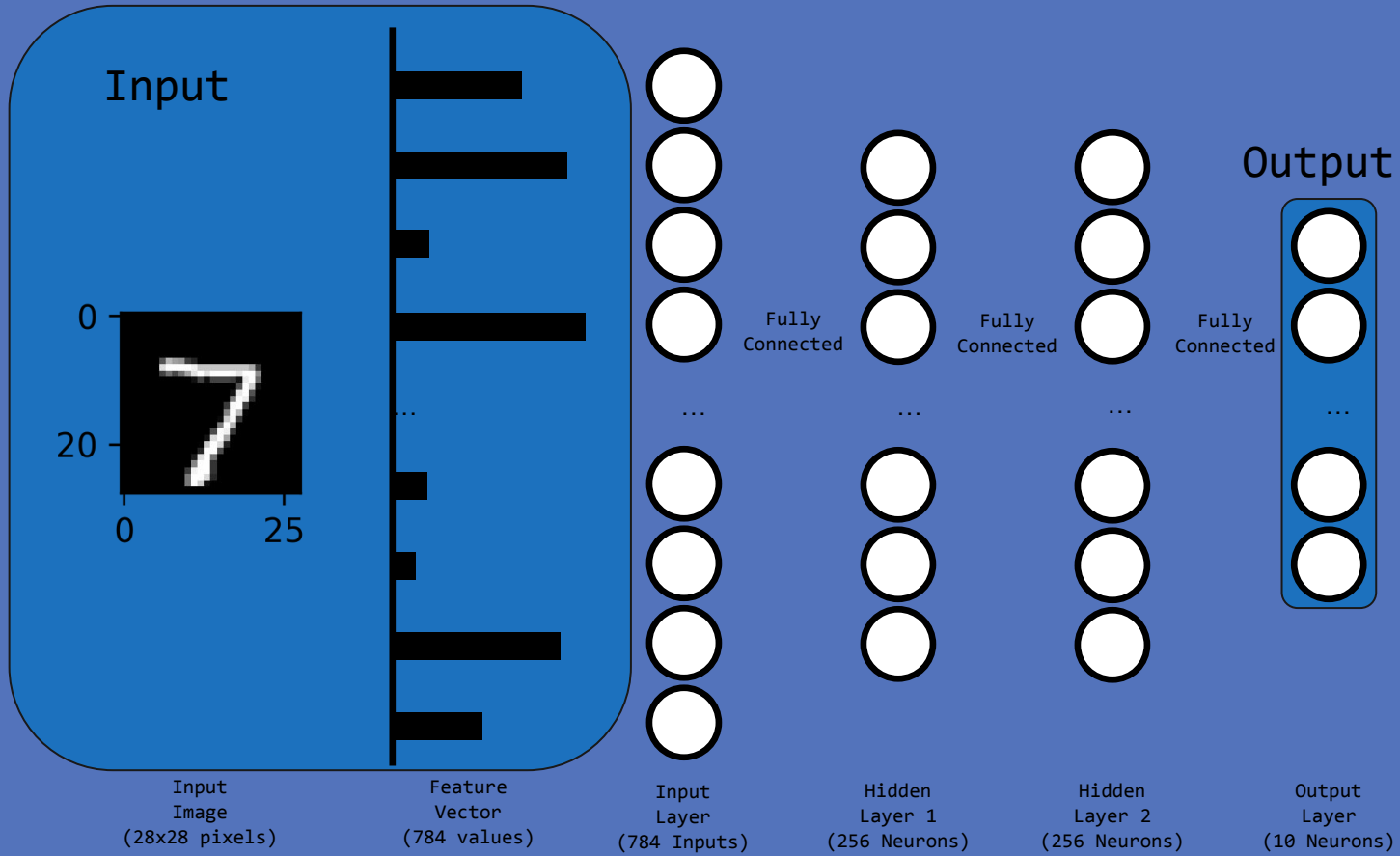
Activation Functions



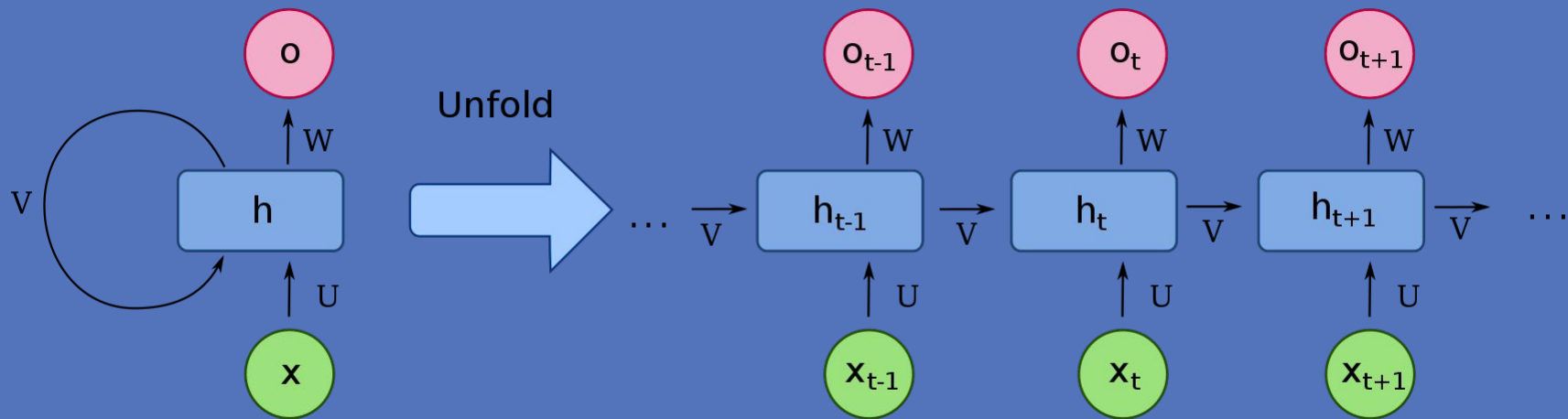
Neural Network Layers: The Dense Layer



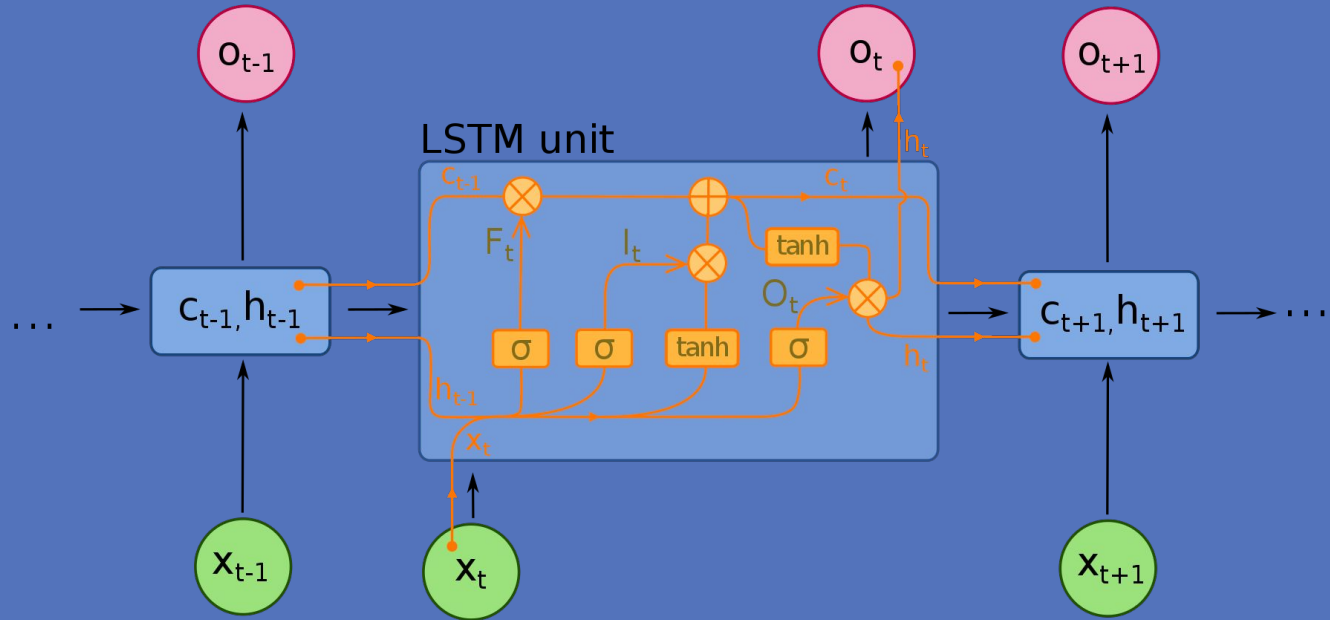
A Simple Neural Network for Identifying Numbers



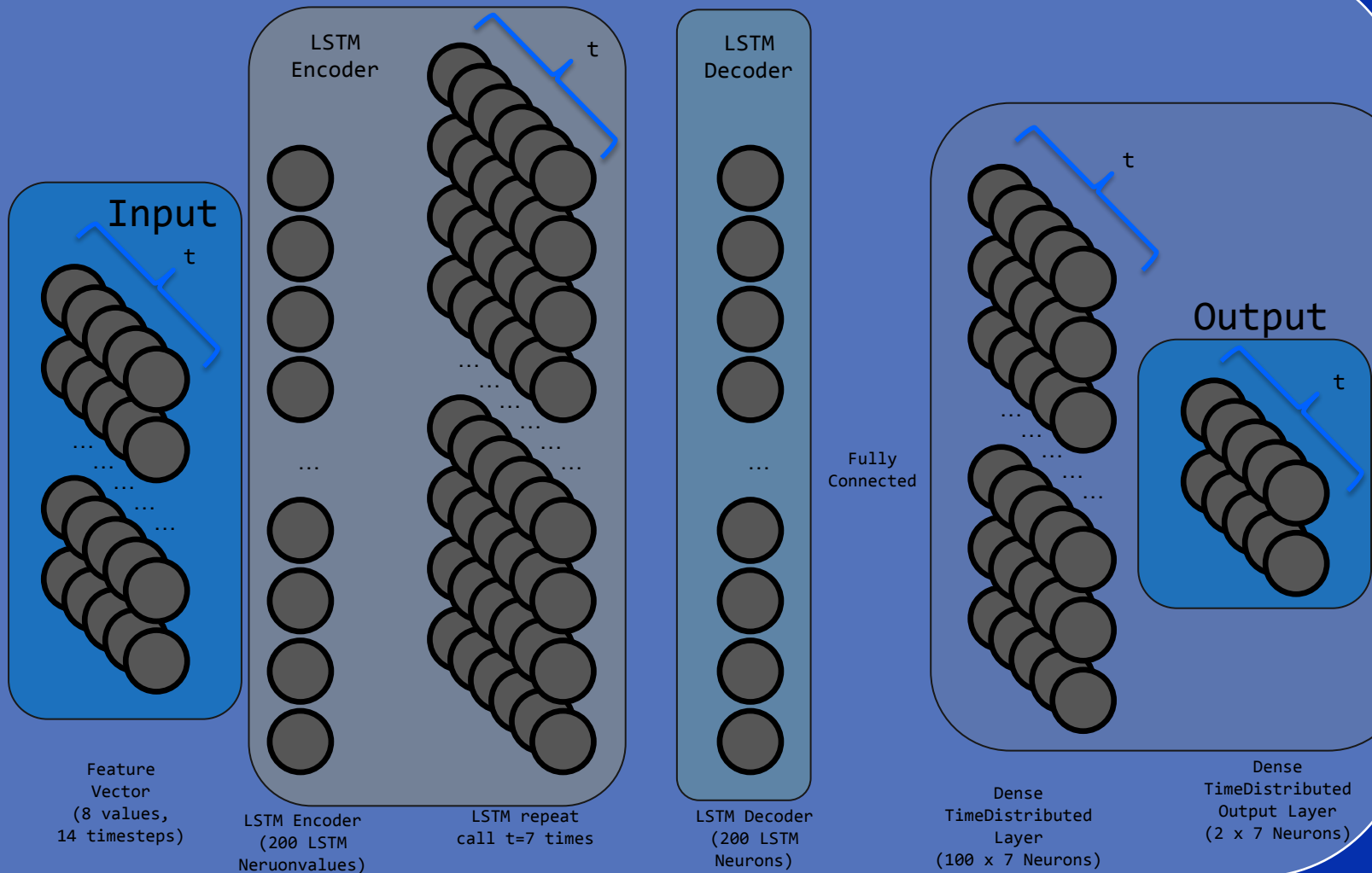
Advanced Topics: Recurrent Neural Networks



Recurrent Neural Networks



LSTM Encoder-Decoder: Multivariate Time Series

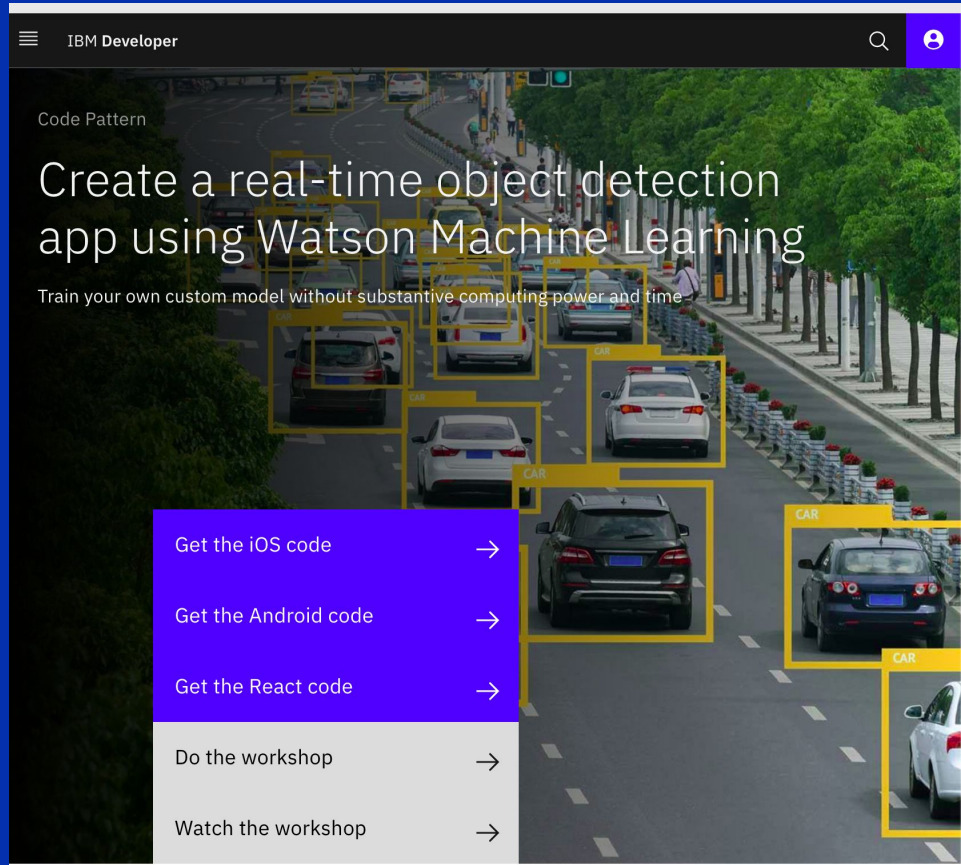


Code Pattern

Create a real-time object detection app using Watson Machine Learning

Train your own custom model without substantive computing power and time

- [Get the iOS code](#) →
- [Get the Android code](#) →
- [Get the React code](#) →
- [Do the workshop](#) →
- [Watch the workshop](#) →



Model | Deployable, Trainable

Object Detector

Localize and identify multiple objects in a single image.

Get this model



Try the API



Try the web app



Try in a Node-RED flow



By IBM Developer Staff

Updated September 21, 2018 | Published March 20, 2018



Model | Deployable, Trainable

Object Detector

Localize and identify multiple objects in a single image.

<https://cloudannotations.ai/>

Get this model



Try the API



Try the web app



Try in a Node-RED flow



By IBM Developer Staff

Updated September 21, 2018 | Published March 20, 2018



Train Your Own Object Detector

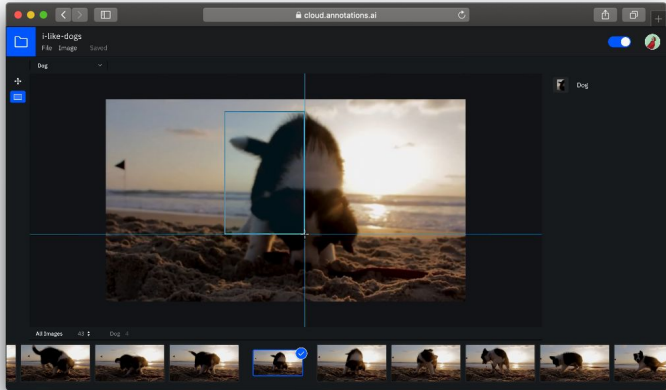
<https://cloud.annotations.ai/>

IBM Cloud Annotations Docs GitHub Log in

Cloud Annotations

A fast, easy and collaborative open source image annotation tool for teams and individuals.

[Continue with IBM Cloud](#) [Documentation](#)



What is TensorFlow Anyways?



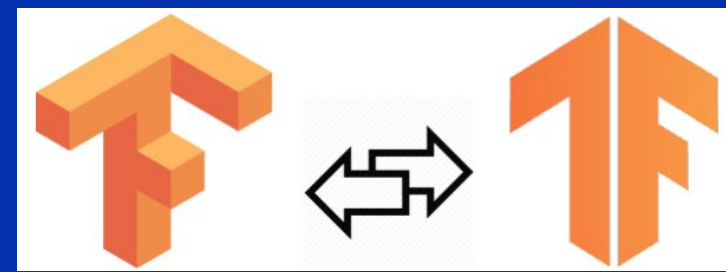
TensorFlow

Open Sourced in 2017
41 Million Downloads
1,800 Contributors

1. "Simple" Python API for running GPU Accelerated Tensor Calculations.
 1. Not just Neural Networks either!
2. Keras API for Neural Networks is even easier to use.
3. Distributed calculations are available for cluster computing.
4. Edge Computing Capabilities:
 - a) In-Browser Model training and serving (tensorflow.js)
 - b) Mobile Deployment with TensorFlow Lite
 - c) Raspberry Pi (pip3 install tensorflow)
 - i. Full installation
 - ii. TensorFlow Lite
 - iii. Node-red hosting
 - iv. Tensorflow.js

TensorFlow 2.0: The New Standard in Deep Learning

What's New in TF2?



1. **tf.data**. API for easy input management
2. Tighter integration with Keras using **tf.keras**.
3. Transfer Learning, Pre-trained models available on TensorFlow Hub
4. Run and debug with eager execution, then use **tf.function** for the benefits of graphs.
5. Distributed Strategies for parallelization of training.
6. Export to SavedModel.
7. TensorFlow will standardize on SavedModel
8. TensorFlow Lite, TensorFlow.js, TensorFlow Hub, and more.

Example Notebooks

`ibm.biz/TF2-Notebooks`



Overview.ipynb

The Sequential Model

NumPy Data Example

tf.data.Dataset Example

Functional API/subclassing

Eager Execution (all Keras Models)

```
# Configure a model for mean-squared error regression.
model.compile(optimizer=tf.keras.optimizers.Adam(0.01),
              loss='mse', # mean squared error
              metrics=['mae']) # mean absolute error

# Configure a model for categorical classification.
model.compile(optimizer=tf.keras.optimizers.RMSprop(0.01),
              loss=tf.keras.losses.CategoricalCrossentropy(),
              metrics=[tf.keras.metrics.CategoricalAccuracy()])
```

```
model.summary()
```

Model: "sequential_2"

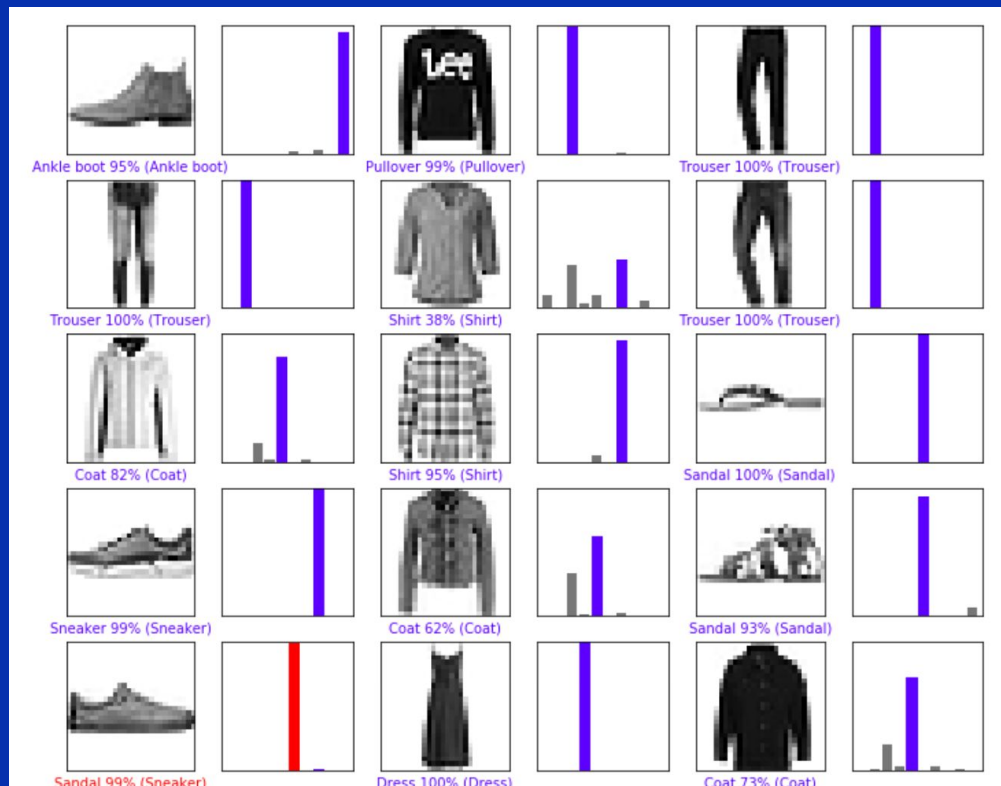
Layer (type)	Output Shape	Param #
dense_17 (Dense)	(None, 64)	2112
dense_18 (Dense)	(None, 64)	4160
dense_19 (Dense)	(None, 10)	650
Total params: 6,922		
Trainable params: 6,922		
Non-trainable params: 0		


```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

basic_classification.ipynb

Walking through the Fashion
MNIST dataset



Basic_text_classification.ipynb

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 16)	160000
global_average_pooling1d (Gl	(None, 16)	0
dense (Dense)	(None, 16)	272
dense_1 (Dense)	(None, 1)	17

Total params: 160,289

Trainable params: 160,289

Non-trainable params: 0

feature_columns.ipynb

One hot encoding, bucketized vectors, and embedding examples

```
feature_columns = []

# numeric cols
for header in ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'slope', 'ca']:
    feature_columns.append(feature_column.numeric_column(header))

# bucketized cols
age_buckets = feature_column.bucketized_column(age, boundaries=[18, 25, 30, 35, 40, 45, 50, 55, 60, 65])
feature_columns.append(age_buckets)

# indicator cols
thal = feature_column.categorical_column_with_vocabulary_list(
    'thal', ['fixed', 'normal', 'reversible'])
thal_one_hot = feature_column.indicator_column(thal)
feature_columns.append(thal_one_hot)

# embedding cols
thal_embedding = feature_column.embedding_column(thal, dimension=8)
feature_columns.append(thal_embedding)

# crossed cols
crossed_feature = feature_column.crossed_column([age_buckets, thal], hash_bucket_size=1000)
crossed_feature = feature_column.indicator_column(crossed_feature)
feature_columns.append(crossed_feature)
```

Basic_regression.ipynb

```
layers.Dense(64, activation='relu', input_shape=[len(train_dataset.keys())]),  
layers.Dense(64, activation='relu'),  
layers.Dense(1)
```

Uses a Deep Neural Network to Perform a Regression

Save_and_restore_models.ipynb

saving_and_serializing.ipynb

```
# Save JSON config to disk
json_config = model.to_json()
with open('model_config.json', 'w') as json_file:
    json_file.write(json_config)
# Save weights to disk
model.save_weights('path_to_my_weights.h5')

# Reload the model from the 2 files we saved
with open('model_config.json') as json_file:
    json_config = json_file.read()
new_model = keras.models.model_from_json(json_config)
new_model.load_weights('path_to_my_weights.h5')

# Check that the state is preserved
new_predictions = new_model.predict(x_test)
np.testing.assert_allclose(predictions, new_predictions, atol=1e-6)

# Note that the optimizer was not preserved.
```

But remember that the simplest, recommended way is just this:

```
model.save('path_to_my_model.h5')
del model
model = keras.models.load_model('path_to_my_model.h5')
```

Tensorboard_in_notebooks.ipynb

```
In [11]: 1 %tensorboard --logdir logs
```

Reusing TensorBoard on port 6006 (pid 74078), started 0:00:11 ago. (Use '!kill 74078' to kill it.)

TensorBoard

SCALARS GRAPHS DISTRIBUTIONS HISTOGRAMS INACTIVE

Show data download links
 Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing: 0.6
Horizontal Axis: STEP (selected), RELATIVE, WALL
Runs: Write a regex to filter runs
TOGGLE ALL RUNS
logs

epoch_accuracy

1


Epoch	Red	Blue	Green	Cyan
0	0.820	0.850	0.840	0.845
1	0.845	0.855	0.850	0.855
2	0.860	0.865	0.860	0.865
3	0.870	0.870	0.865	0.870
4	0.875	0.870	0.865	0.860

epoch_loss

1

Epoch	Red
0	0.510
1	0.505
2	0.500
3	0.495
4	0.490

Adversarial Attacks in TF 2.0

Jupyter adversarial_fgsm (autosaved)  Logout


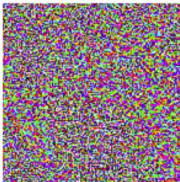

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

as described in [Explaining and Harnessing Adversarial Examples](#) by Goodfellow *et al.* This was one of the first and most popular attacks to fool a neural network.

What is an adversarial example? ¶

Adversarial examples are specialised inputs created with the purpose of confusing a neural network, resulting in the misclassification of a given input. These notorious inputs are indistinguishable to the human eye, but cause the network to fail to identify the contents of the image. There are several types of such attacks, however, here the focus is on the fast gradient sign method attack, which is a *white box* attack whose goal is to ensure misclassification. A white box attack is where the attacker has complete access to the model being attacked. One of the most famous examples of an adversarial image shown below is taken from the aforementioned paper.

	+ .007 ×		=	
x "panda" 57.7% confidence		$\text{sign}(\nabla_x J(\theta, x, y))$ "nematode" 8.2% confidence		$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$ "gibbon" 99.3% confidence

Trusted AI Lifecycle through Open Source

Pillars of trust, woven into the lifecycle of an AI application

Did anyone
tamper with it?



ROBUSTNESS

Adversarial
Robustness 360

↳ (ART)

github.com/IBM/adversarial-robustness-toolbox

art-demo.mybluemix.net

Is it fair?



FAIRNESS

AI Fairness
360

↳ (AIF360)

github.com/IBM/AIF360

aif360.mybluemix.net

Is it easy to
understand?



EXPLAINABILITY

AI Explainability
360

↳ (AIX360)

github.com/IBM/AIX360

aix360.mybluemix.net

Is it accountable?

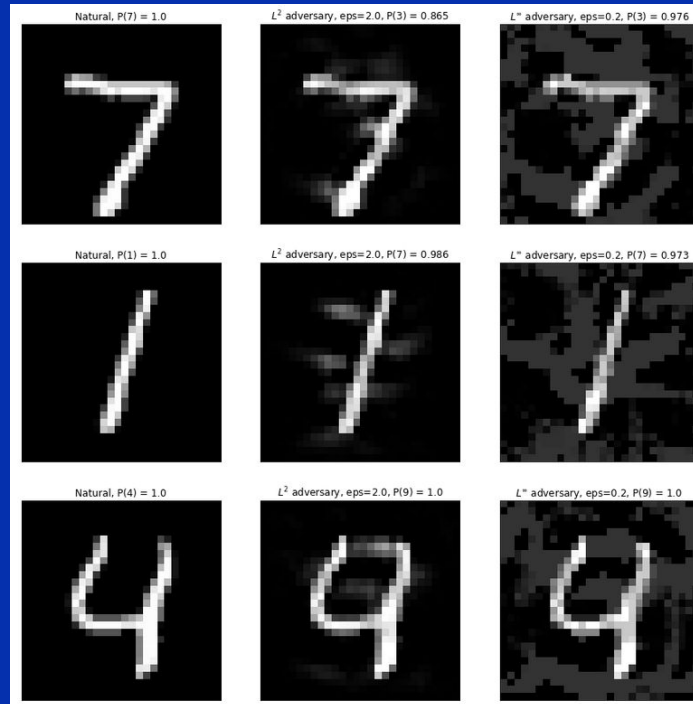


LINEAGE

↳ KubeFlow, WML

Simple AI Example: Adversarial MNIST Examples

Adversarial machine learning can be used to “trick” machine learning models into providing incorrect predictions, often with devastating consequences e.g. self driving vehicles



Adversarial machine learning: Gaming Our Algorithms

Robust AI Example: Self Driving Vehicles

Adversarial machine learning can be used to “trick” machine learning models into providing incorrect predictions, often with devastating consequences e.g. self driving vehicles

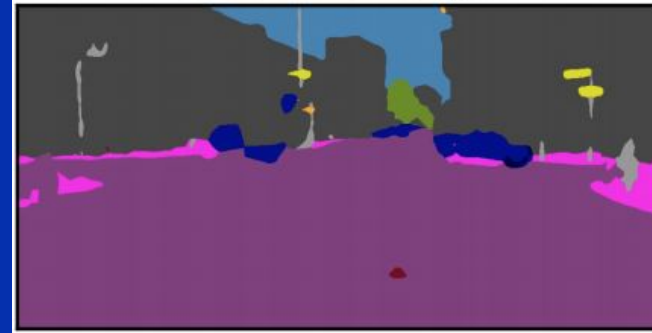
Original image



Network prediction



Adversarial target



<https://art-demo.mybluemix.net/>



AI Research

Research areas ▾

Publications

Experiments ▾

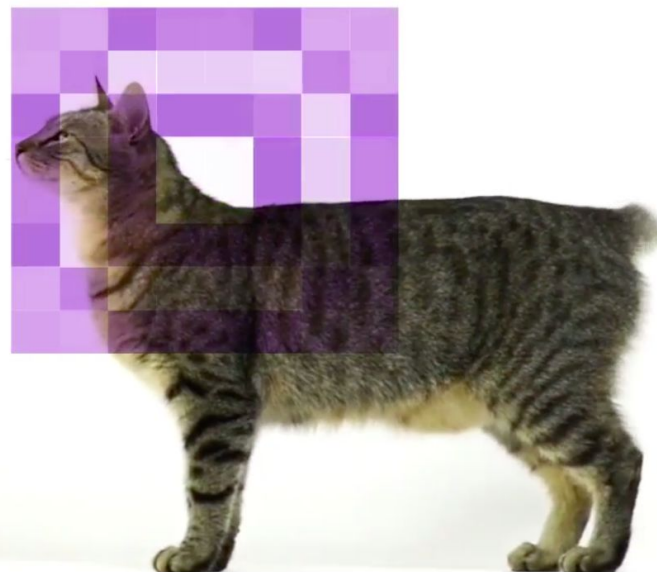
Careers

Blog

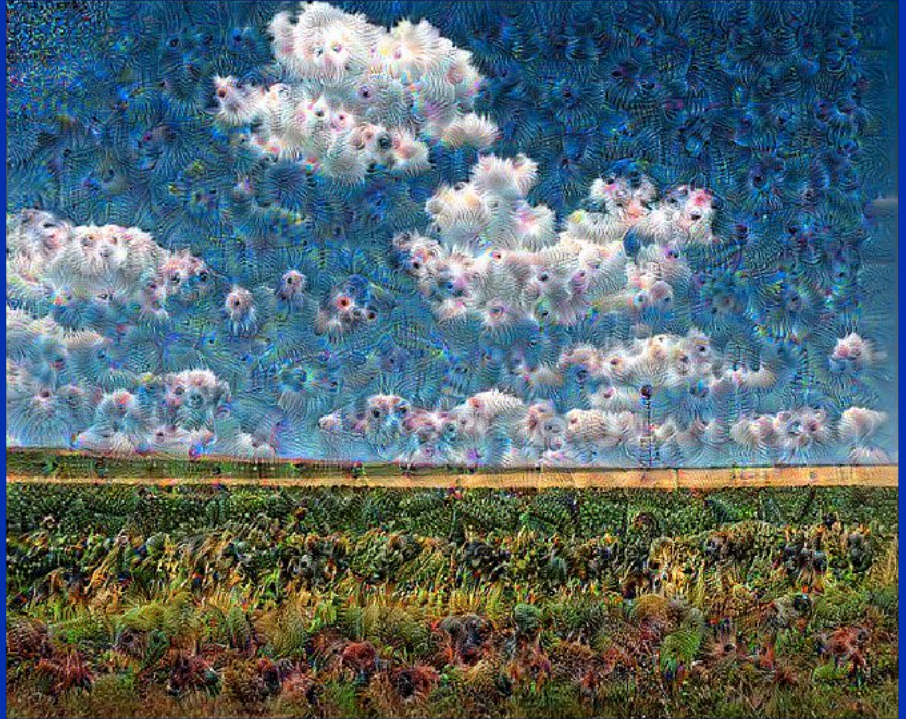


Your AI model might be telling you this is not a cat

Defend your AI model against attacks. Our open-source software library supports both researchers and developers in making AI systems more secure. Create and simulate attacks and different defense methods for machine learning models in this demo.



Deep Dream Example



Neural Style Transfer Example



TensorFlow at the Edge

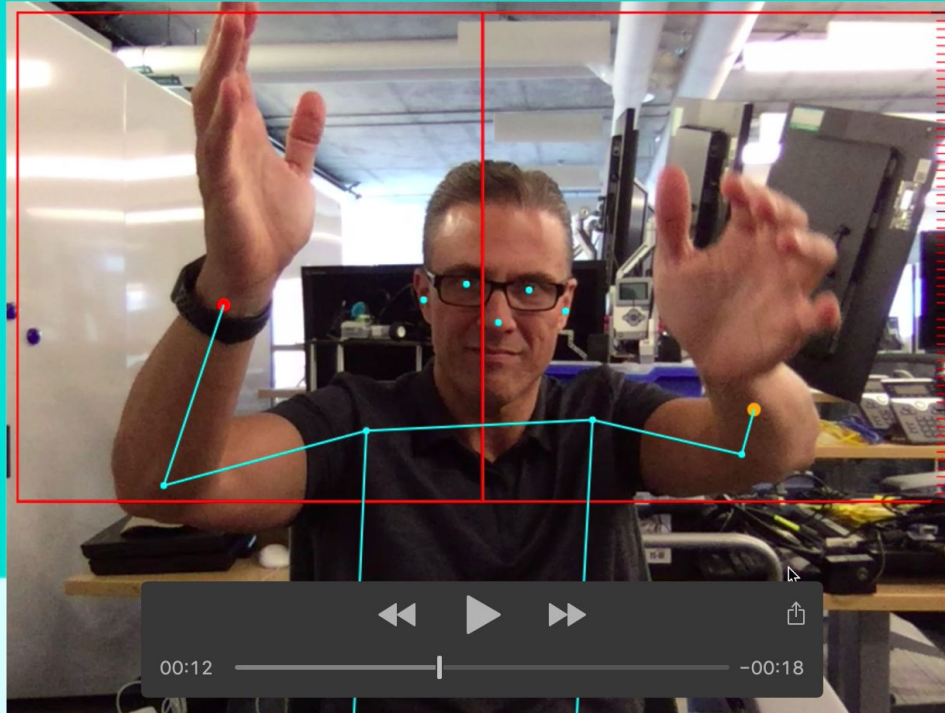


tensorflow.js: Running TensorFlow in the Browser

veremin

a video theremin based on PoseNet

<https://github.com/vabarbosa/veremin.git>
<https://veremin.mybluemix.net/>



TensorFlow Lite:
Running
TensorFlow on
iOS, Android,
Rpi 4,
CoralBoard,
Arduino, DSPs (TF
Micro) and
beyond!

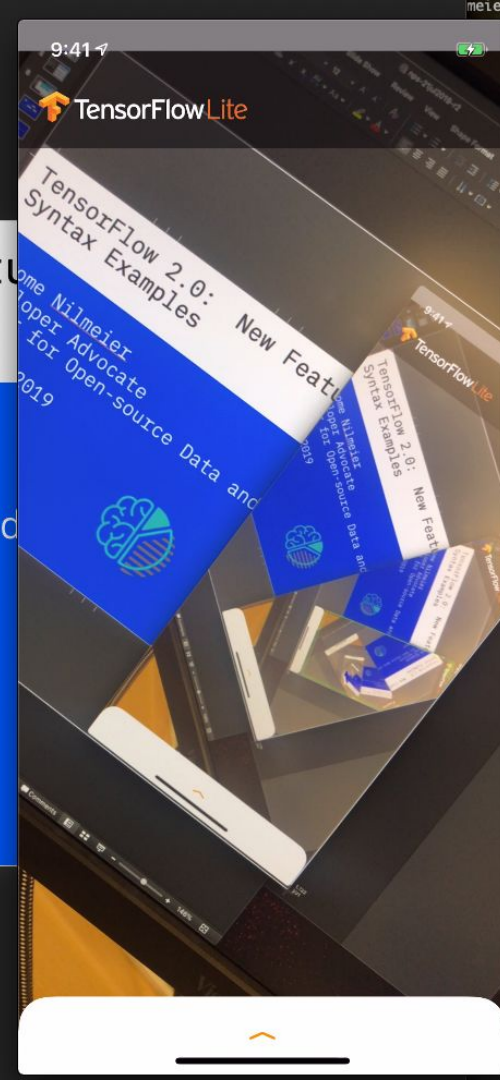
TensorFlow 2.0: New Features Syntax Examples

Jerome Nilmeier
Developer Advocate
Center for Open-source Data and

May 22, 2019



IBM Developer



Conclusions

TensorFlow 2.0 has arrived! It will be the standard for deep learning, with Keras leading the way.

Some of the most exciting new directions involve edge computing. TensorFlow is also the standard for edge computing applications.

