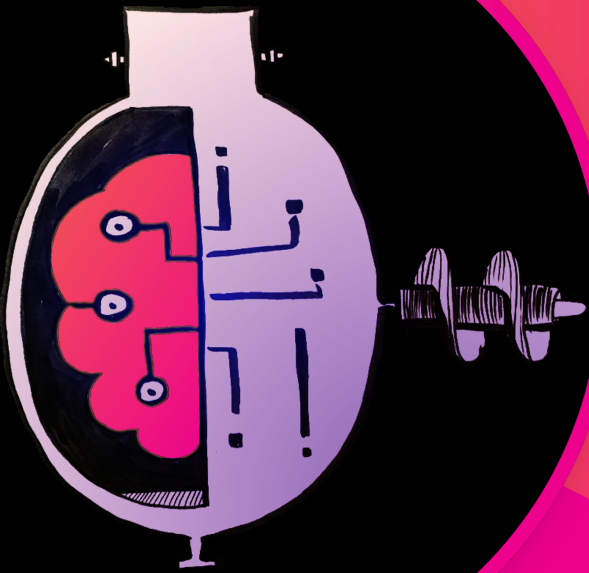


L
e
D
a
T
e
r
o
p
i
n
K
g
i
t



Announcing the Deep Learning Toolkit for Splunk with TensorFlow 2.0, PyTorch, NLP and Jupyter Lab Notebooks

Philipp Drieger

Staff Machine Learning Architect, Splunk

splunk>

.conf19

.conf19
splunk>



Philipp Drieger

Staff Machine Learning Architect | Splunk



Anthony Tellez

Staff Data Scientist | Splunk

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.

Agenda



▶ About us

▶ Intro to AI | ML | DL

▶ MLTK Container

▶ Use Cases

▶ Wrap up

About Us

Philipp Drieger | Staff Machine Learning Architect, Splunk

Anthony Tellez | Staff Data Scientist, Splunk

.conf19

splunk>



Anthony Tellez | アンソニー テレズ

CISSP, CEH, CNDA, Sec+

- | where_time@Splunk > 5y
- Previous:
 - U.S. Gov Contractor, Geospatial Analyst
 - Splunk Federal, PS Architect
- Specializations
 - Cryptography
 - Information Security – Red Team
 - Open Source Network Security: Suricata, Zeek
- Field Data Scientist
 - Security & Fraud Analytics
 - Data Visualization & Statistics
- Responsible for the relationship between emerging technologies and global field organization: US, CAN, LATAM, APAC
- AI Evangelist for Splunk, presenting at various security & industry events
- <https://github.com/anthonygtellez/>
- Fact: Spends 80% of the year on a plane traveling globally.



Phoenix, AZ

Philipp Drieger

- | where _time @Splunk > 4.5y
- Previous:
 - +15y in research, software development, visual arts
 - +3y SE across portfolio & domains in CEMEA & EE
- Specializations
 - Anomaly Detection, Data Mining, NLP, Advanced Analytics and Visualizations
 - Applied Data Science, Machine Learning, Graph Theory and Network Science
 - GPU Computing, Deep Learning
- Role @ Splunk
 - Staff Machine Learning Architect (Central EMEA)
 - Author of [DGA App for Splunk](#)
 - Author of [MLTK Container for Splunk](#)
 - Author of [Deep Learning Toolkit for Splunk](#)
 - Blog posts, conf talks, hackathons etc.
 - **Ensure Customer Success with ML**



Munich, Germany

Intro AI | ML | DL

.conf19
splunk>

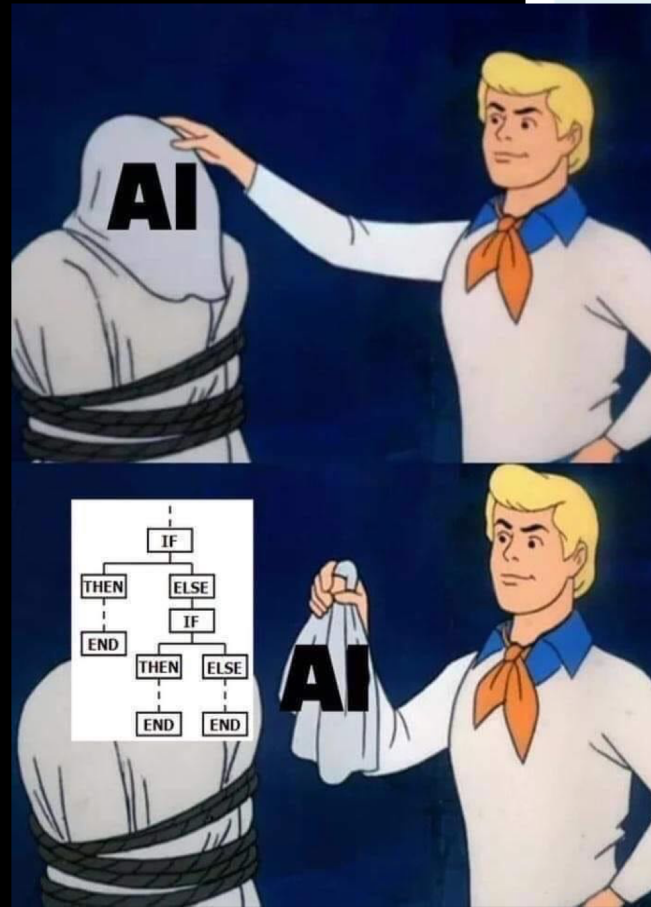
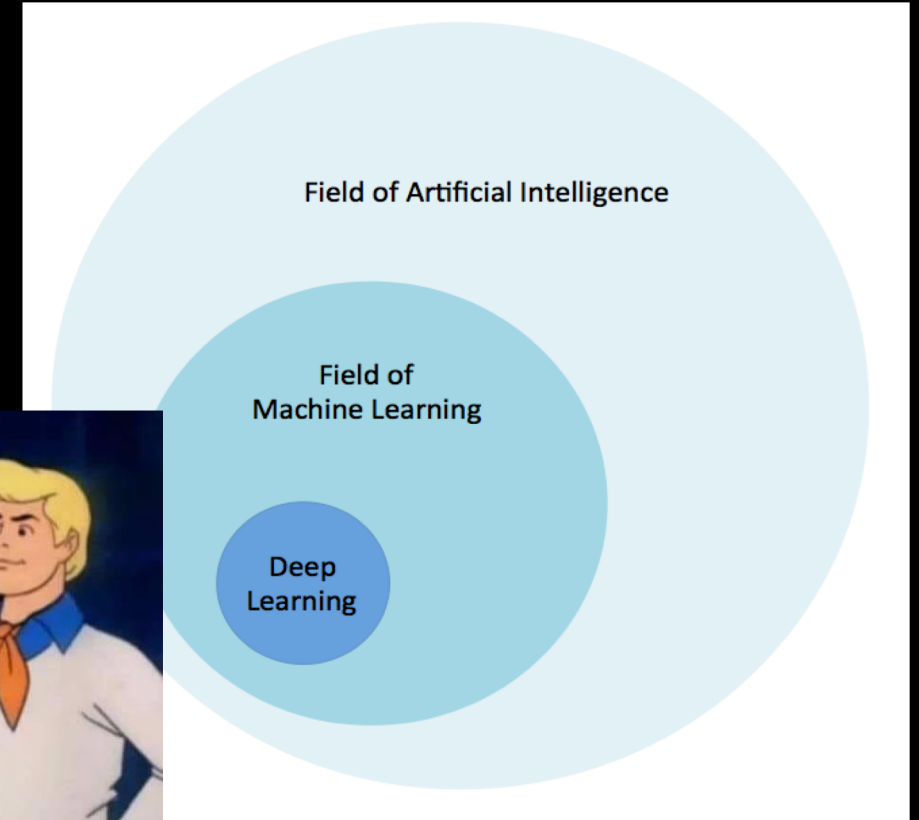
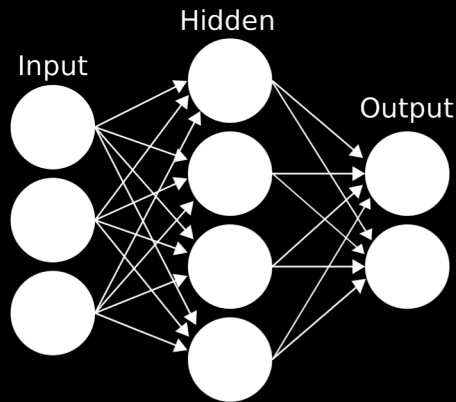


AI v. Machine Learning

- ▶ “A Function that maps features to an output” = **AI**
- ▶ “A Function that learns patterns in your data without being explicitly programmed” = **ML**

Types of ML

- Supervised
- Unsupervised
- Reinforcement





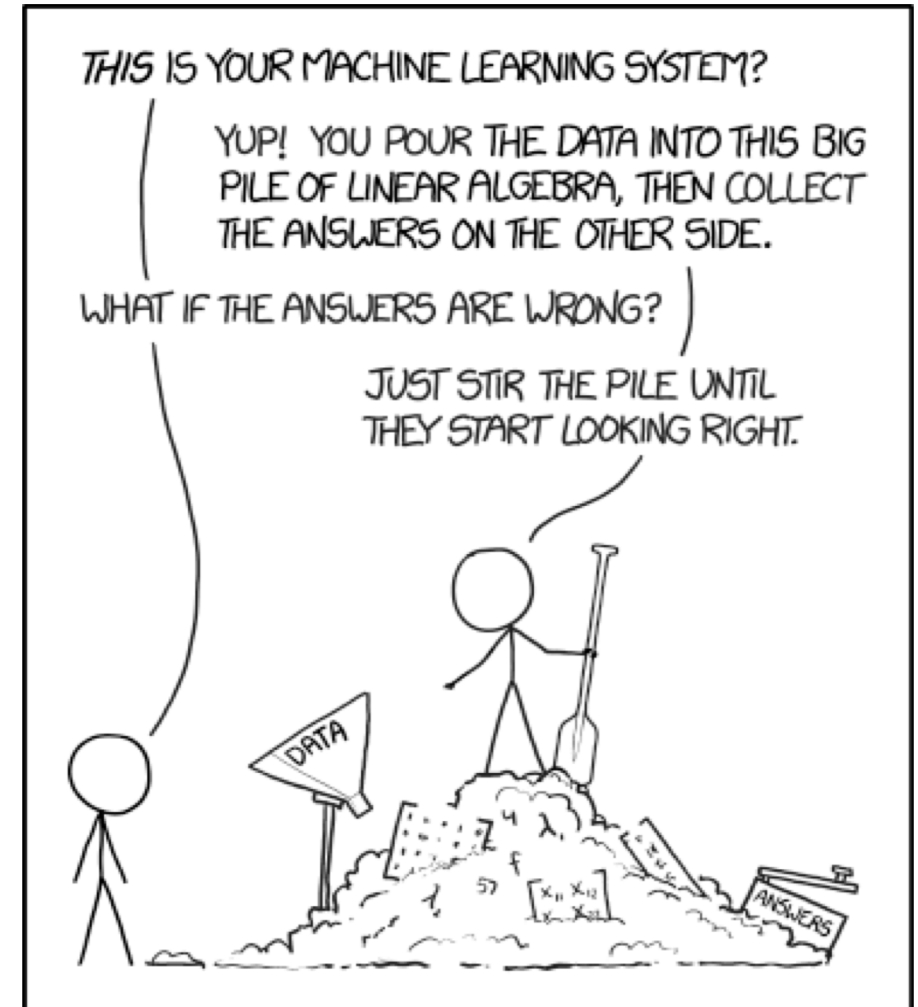
What ML & AI are not

Machine Learning is not Magic

Garbage Data = Useless Predictions

- Data Scientists spend **80% of their time** cleaning, munging and collecting data
- Throwing more data at an algorithm will not result in solving all of your SOC issues
- Machine Learning requires a solid understanding of statistics and the scientific method

ML & AI require you to understand the fundamental business problem you want to solve.





What ML & AI are not

Machine Learning is not Magic

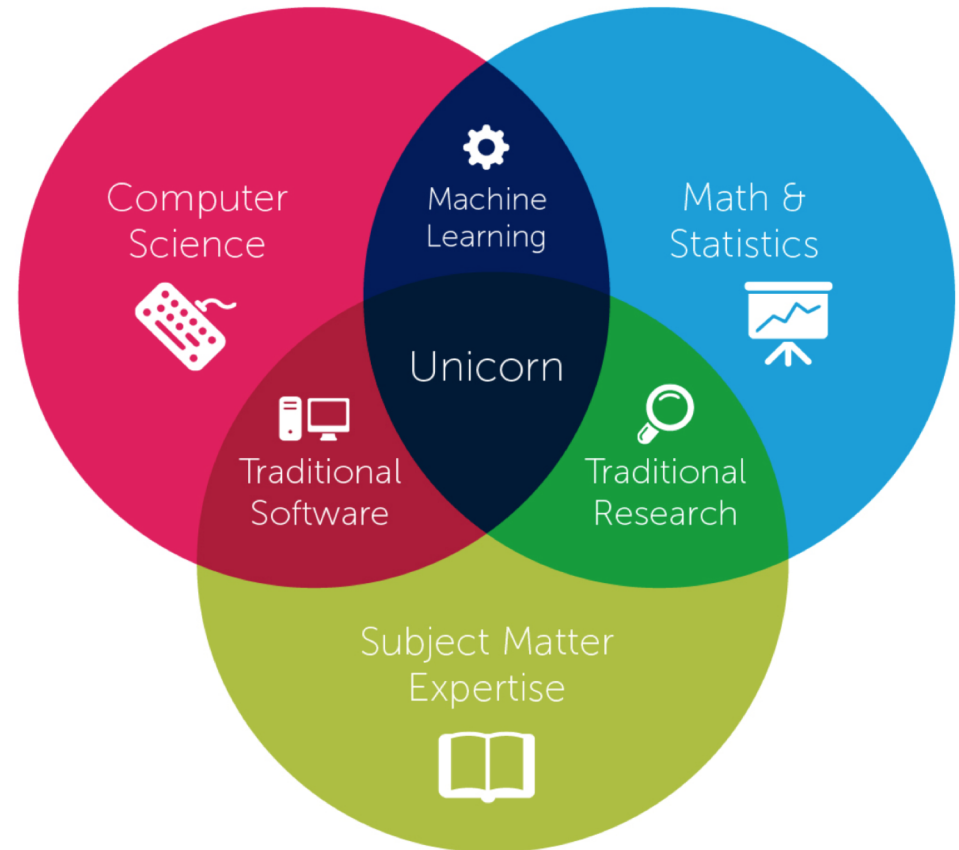
ML is not a replacement for expert analysts, or engineers.

ML requires Subject Matter Experts to enhance security & IT operations.

Analysts are required to provide feedback to the models to adjust thresholding rules and reduce false positives.

If you need more examples check out some of the past conf talks:

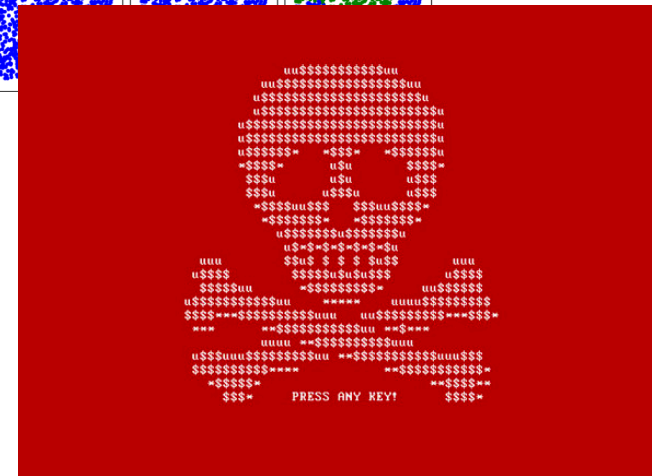
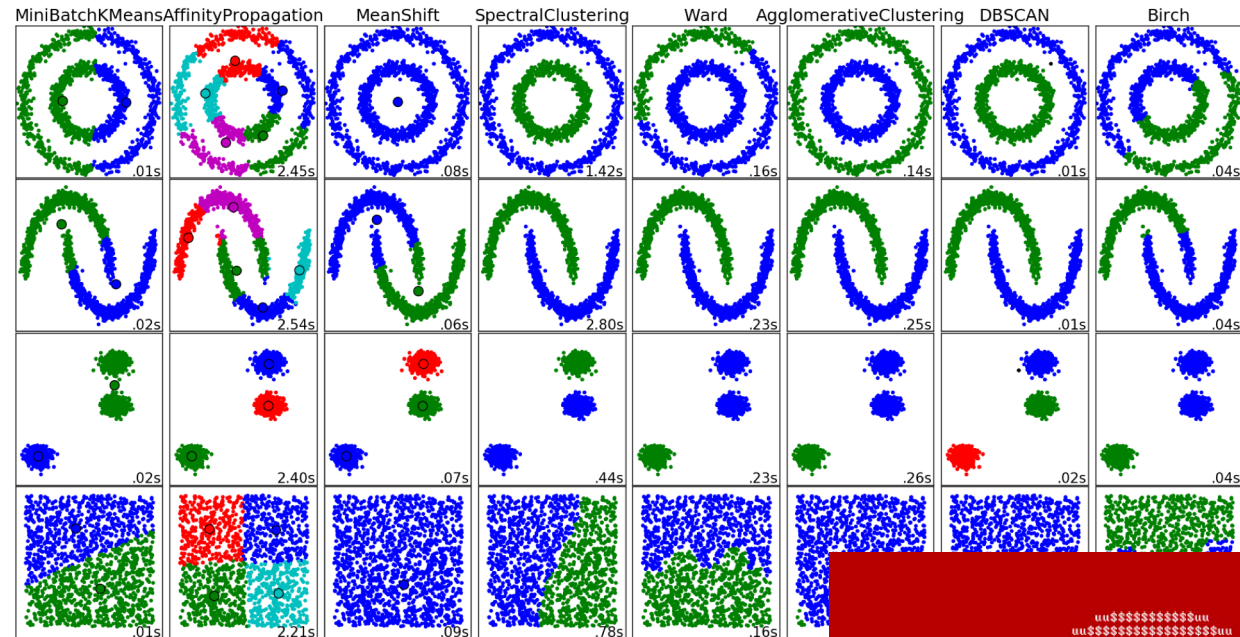
- .conf18 Getting Your Data Ready for Machine Learning - Kristal Curtis
- .conf18 Using the Latest Features from the Splunk Machine Learning Toolkit to Create Your Own Custom Models – Harsh Keswani
- .conf18 Turning Security Use Cases into SPL – Marquis Montgomery



Machine Learning & AI

What does the scientific method look like in the IT & Security Space?

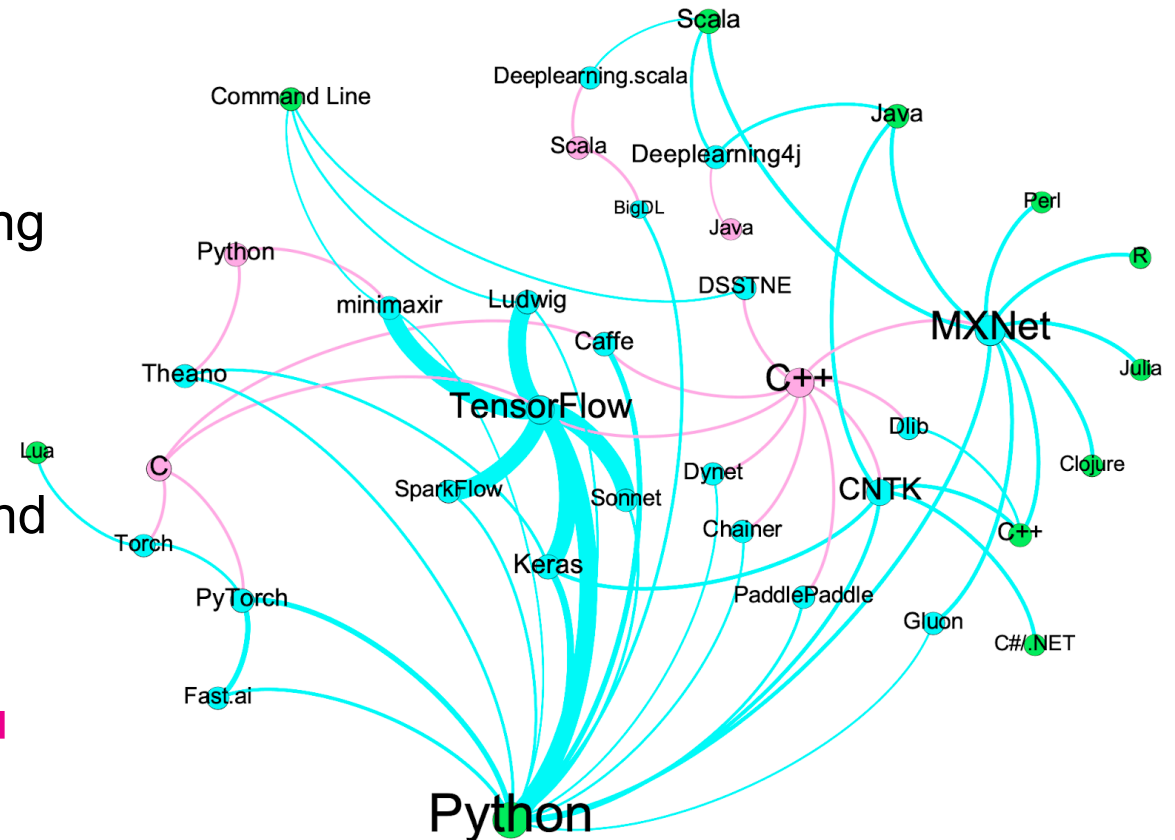
- ▶ **Problem:** DGA domains are computer generated pseudo-random character strings used by attackers, blacklisting an infinite number of domains is not feasible.
- ▶ **Hypothesis:** “Are there *patterns in domain generation algorithms* that can be exploited to identify newly generated domains as threats in real-time?”
- ▶ **Example Domains:**
<http://87hfdredwertyfdvvlkgdrsadm.net/af/GHFbfsalku65>
<http://87hfdredwertyfdvvlkgdrsadm.net/af/sdgLKJvgh>
<http://wszystkodokuchni.pl/34f43>



Machine Learning & AI

What does the Deep Learning Marketplace look like?

- ▶ **Not every problem** can be solved with ML
- ▶ If you understand your underlying business problem and can clearly state your hypothesis... ML provides you a statistical framework for testing
- ▶ Deep Learning is designed to help customers leverage the power of more advanced math & parallel processing power.
- ▶ Deep Learning frameworks such as **PyTorch**, and **Tensorflow** can leverage both **CPU & GPU** resources to reduce training time.
- ▶ You still need to **understand the problem you are solving** to optimize the neural network's layers & hyperparameter tuning.



<https://www.rtinsights.com/top-deep-learning-tools/>

MLTK Container

.conf19

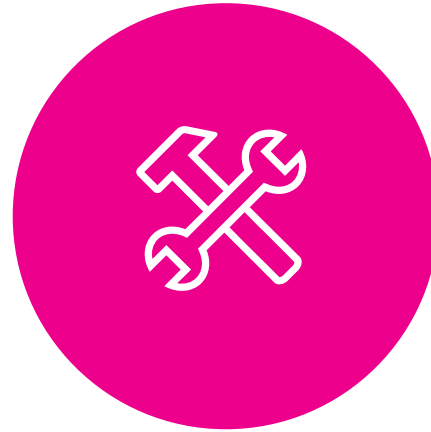
splunk>



Key Benefits of the MLTK Container



Seamlessly Integrate with
Splunk Enterprise and
Machine Learning Toolkit
Workflows

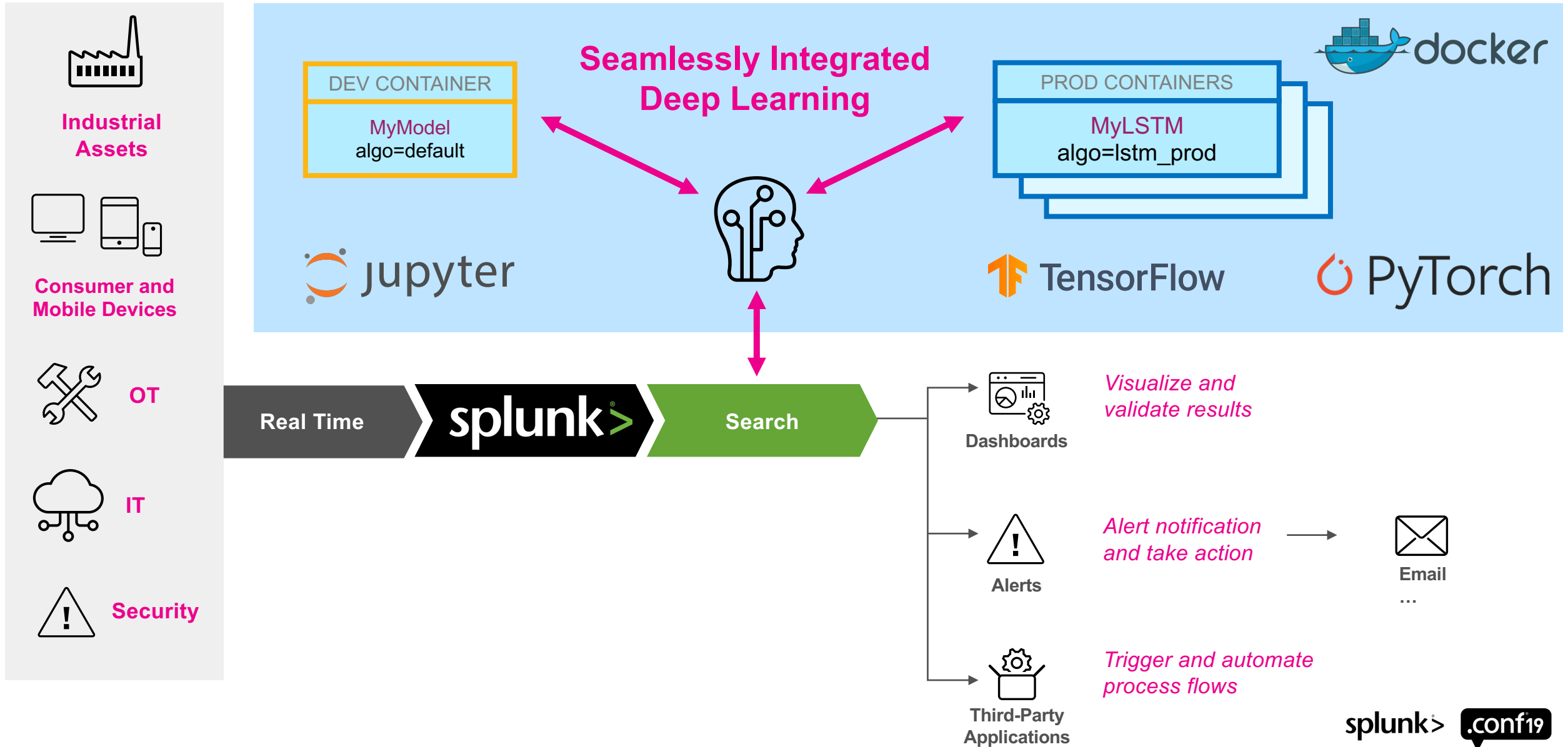


Freedom of Code within
Jupyter Lab Notebooks for
Advanced Modelling with
TensorFlow and PyTorch



GPU accelerated Deep
Learning for Compute
Intensive Training
Workloads

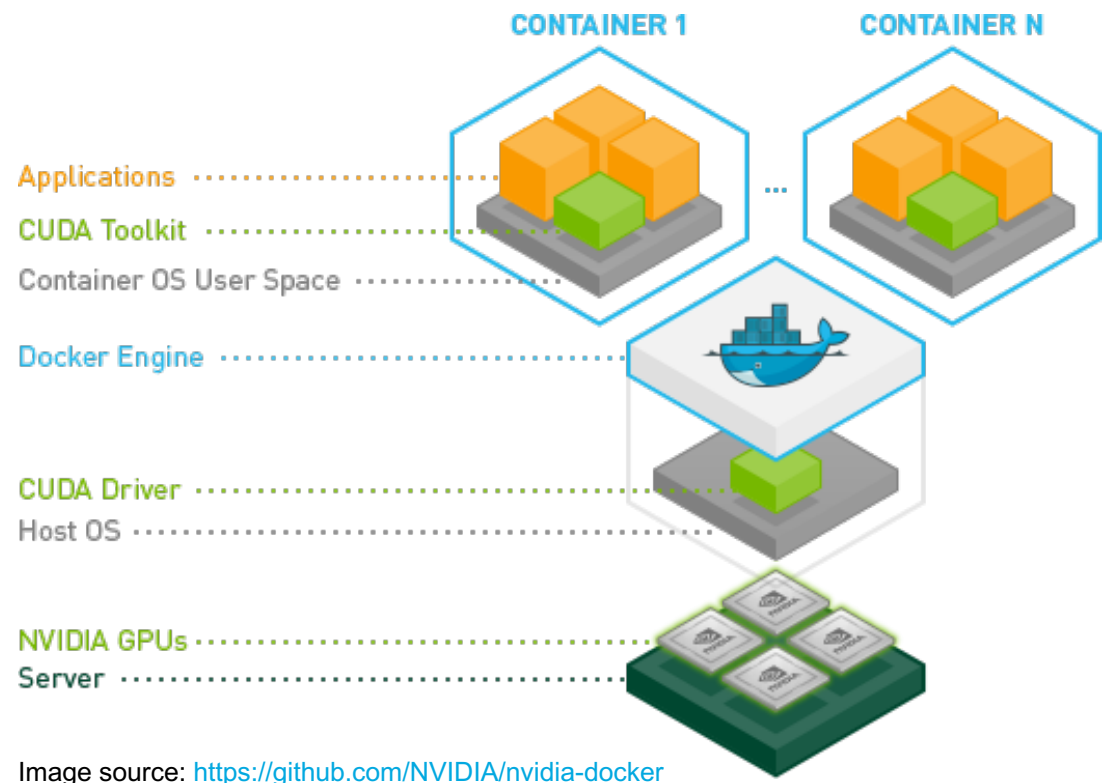
Integrated Architecture with Splunk's MLTK



GPU Accelerated Training of Deep Learning Models

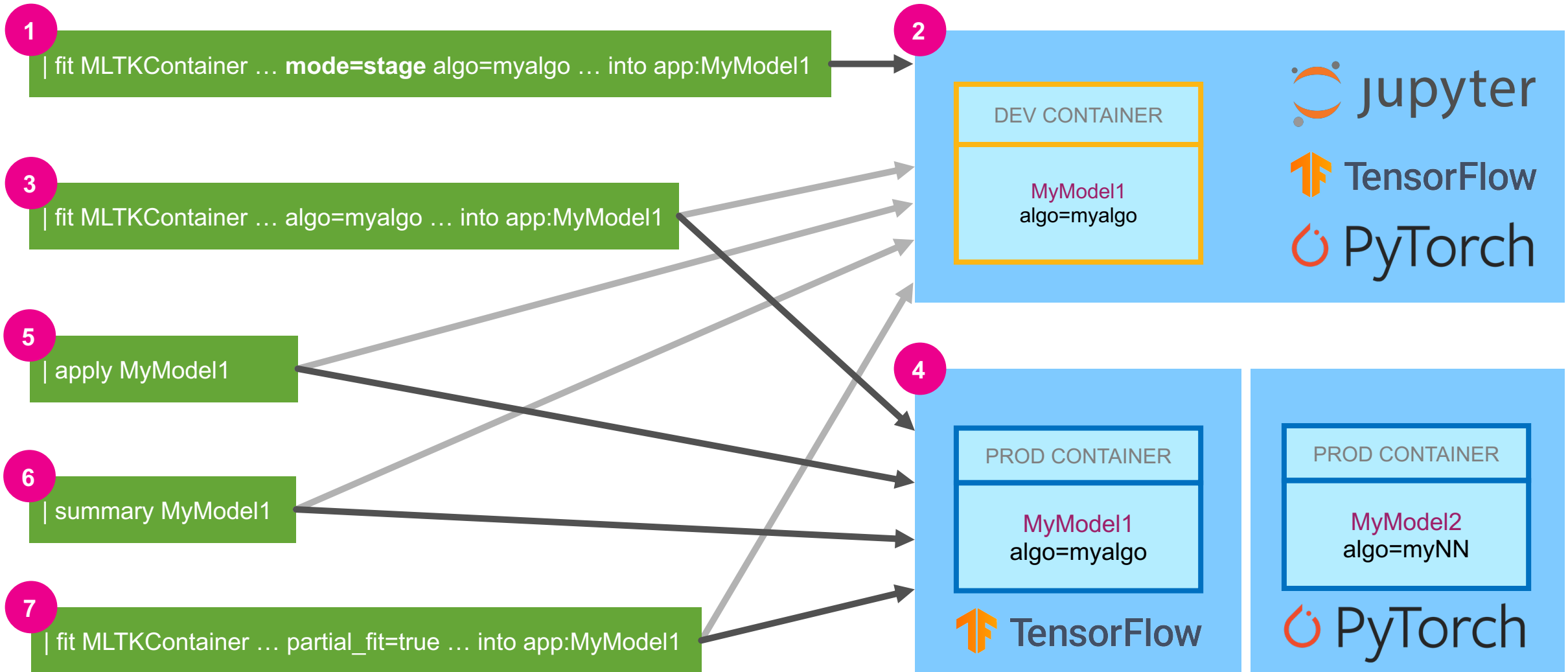
- ▶ Many Deep Learning Algorithms like Neural Networks require intense numerical computations
- ▶ GPUs can speed up such workloads by parallelizing over many cores
- ▶ GPU Computing is a complex topic, so please prepare yourself and set the right expectations. Study the best practise and available mechanisms in your framework of choice to achieve desired speedups and increased computational throughput by leveraging GPUs.

- ▶ Containerized Multi GPU Computing



Jupyter Lab Notebooks Workflow

Develop and Operationalize your Model with a few simple steps



MLTK Syntax Overview

- **Fit (i.e. train) a model** from search results

```
... | fit <ALGORITHM> <TARGET> from <VARIABLES ...>  
      <PARAMETERS> into <MODEL>
```

- **Apply a model** to obtain predictions from (new) search results

```
... | apply <MODEL>
```

- **Inspect the model** inferred by <ALGORITHM> (e.g. display coefficients)

```
| summary <MODEL>
```

- **DLTK Example Syntax**

```
... | fit MLTKContainer response from age blood_pressure  
      diabetes_pedigree glucose_concentration mode=stage algo=myalgo  
      epochs=10 batch_size=32 partial_fit=true into  
      app:diabetes_classifier_model
```

DLTK Syntax Overview

MLTKContainer

- Used as a bridge algorithm, converts data using MLTK's API

Algo

- Defines the Python notebook algorithm & container used as part of the search job

Mode

- Used to add data into a notebook as a csv, example: mode=**stage** *Optional*

Epochs

- “One epoch is when an entire dataset is passed both forward and backward through the **neural network** only once”

Batch_size

- Refers to the number of training examples utilized in one iteration

Partial_fit

- **Online learning** or “Incremental fit on a batch of samples” used to add incremental updates to a model file as new data is made available

App:

- Describes the model in a shared app context specifies container endpoint to be utilized for a production, more info:



Demo

Content

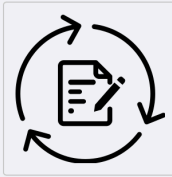
Edit Export ...

- Overview
- Configuration
- Classifier
- Regressor
- Forecasting
- Clustering
- NLP

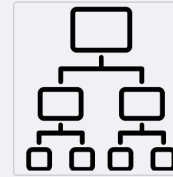
Overview



Architecture
Learn about the architecture how the Deep Learning Toolkit works



Model Development Workflow
Learn about the predefined rapid model development workflows using Jupyter Lab Notebooks



Multi GPU Computing
Learn how you can leverage multi GPU computing for high computational workloads

Configuration



Setup Information
Learn about how to setup the MLTK Container step by step, further details, limitations and security considerations



Docker Setup
Configure how the MLTK Container connects to your Docker environment



Container Management
Controls to start and stop the MLTK Container and check its status

Classifier

serum_insulin	skin_thickness
0	0
0	0
0	0
0	0
110	32
3	4
5	6
7	8
9	10
next	

Predicted 0	Predicted 1
79 (77.5%)	23 (22.5%)
11 (20.8%)	42 (79.2%)

Neural Network Classifier
This example shows how to use a binary neural network classifier build on keras and TensorFlow™

TensorFlow™

serum_insulin	skin_thickness
0	0
0	0
0	0
0	0
110	32
3	4
5	6
7	8
9	10
next	

Predicted 0	Predicted 1
79 (77.5%)	23 (22.5%)
11 (20.8%)	42 (79.2%)

Logistic Regression
This example shows a simple logistic regression using PyTorch

PyTorch

serum_insulin	skin_thickness
0	0
0	0
0	0
0	0
110	32
3	4
5	6
7	8
9	10
next	

Predicted 0	Predicted 1
79 (77.5%)	23 (22.5%)
11 (20.8%)	42 (79.2%)

LSTM
This example shows the results of a LSTM to classify DGA domains

TensorFlow™

Setup

Edit Export ...

1. Machine Learning Toolkit Installation

Please make sure that [Splunk Machine Learning Toolkit](#) and [Python for Scientific Computing](#) package (choose for your OS platform!) is properly installed to work with this app.

You need to set the Machine Learning Toolkit app to **global permissions** so that knowledge objects are shared.

label	version
Splunk Machine Learning Toolkit	4.3.0
Python for Scientific Computing	1.4

2x dependencies found

2. Configuration of your connection to Docker

Description

On this setup page the connection to your docker environment is configured. Typical scenarios are:

- **single-instance** deployment with docker and splunk running on the same instance
- **side-by-side** deployment where the splunk instance communicates with another instance that is the docker host

Settings

Docker Host

Endpoint URL

Save

Example configurations

deployment	docker host	endpoint url	host os
single-instance	unix://var/run/docker.sock	localhost	mac
single-instance	unix://var/run/docker.sock	yourhostname	linux
side-by-side	tcp://remote.host.com:2375	remote.host.com	any

Containers

Edit Export ...

Overview of all Container Models

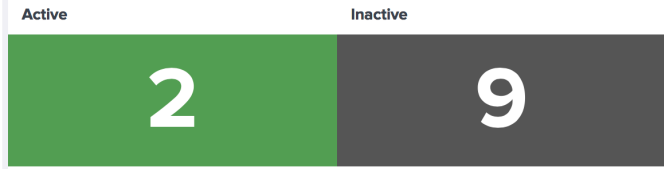
Development Container

Container Image: PyTorch Docker Runtime: nvidia Use fixed port: Jupyter Lab (8888) Tensorboard (6006)

✓ RUNNING

START STOP JUPYTER LAB TENSORBOARD

Status of all Container Models



Container Model Status

	model	sharing	image	runtime	api_url	jupyter_url	tensorboard_url
1	__dev__	global	mltk-container-pytorch	None	http://localhost:32774	http://localhost:8888	http://localhost:6006
2	call_center_rnn_forecast_model	global					
3	diabetes_classifier_model	global	mltk-container-tf-cpu	None	http://localhost:32777	http://localhost:32775	http://localhost:32776
4	diabetes_classifier_model2	global					
5	internet_traffic_forecast_model	global					
6	pytorch_rnn_test	global					
7	spacy_entity_extraction_model	global					
8	DNN_iris_model	user					
9	LR_Model	user					
10	PyTorch_iris_model	user					
11	RNN_Model	user					

Neural Network Classifier Example

Edit Export ...

Epochs

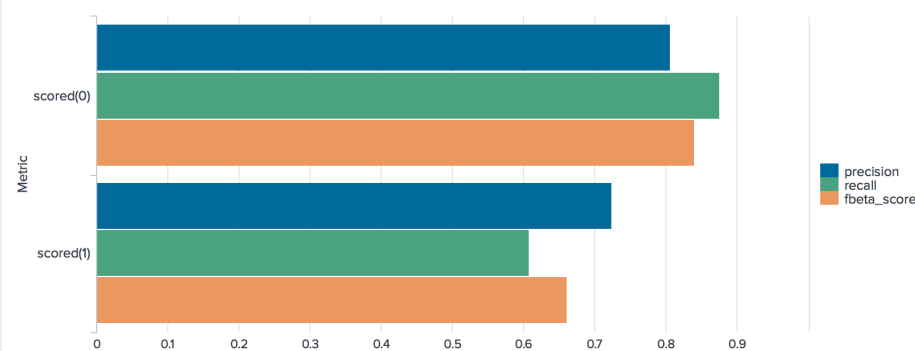
1000 Submit Hide Filters

Example for TensorFlow

This example shows a multi layer fully connected neural network for binary classification using TensorFlow.

response	response_prediction	response_prediction_raw	BMI	age	blood_pressure	diabetes_pedigree	glucose_concentration	number_pregnant	serum_insulin	skin_thickness
1	1	0.61574805	33.6	50	72	0.627	148	6	0	35
0	0	0.13517466	26.6	31	66	0.35100000000000003	85	1	0	29
1	1	0.74884856	23.3	32	64	0.672	183	8	0	0
0	0	0.11605233	28.1	21	66	0.16699999999999998	89	1	94	23
1	1	0.85622925	43.1	33	40	2.2880000000000003	137	0	168	35

< Prev 1 2 3 4 5 6 7 8 9 10 Next >



Confusion Matrix

accuracy	precision	recall	f1
0.7826	0.7780	0.7826	0.7776

Label	predicted(0)	predicted(1)
actual(0)	438	62
actual(1)	105	163

- Show data download links
- Ignore outliers in chart scaling
- Tooltip sorting method: default ▾

Smoothing



Horizontal Axis

- STEP RELATIVE WALL

Runs

train

- fit/20190829-172737/train
- fit/20190829-142829/train
- fit/20190830-155546/train
- fit/20190830-180436/train
- fit/20190902-064625/train
- fit/20190902-105000/train
- fit/20190905-105939/train
- fit/20190905-110403/train
- fit/20190905-110415/train
- fit/20190905-110635/train
- fit/20190905-110641/train
- fit/20190905-110825/train
- fit/20190905-110832/train
- fit/20190905-110836/train
- fit/20190905-165518/train

TOGGLE ALL RUNS

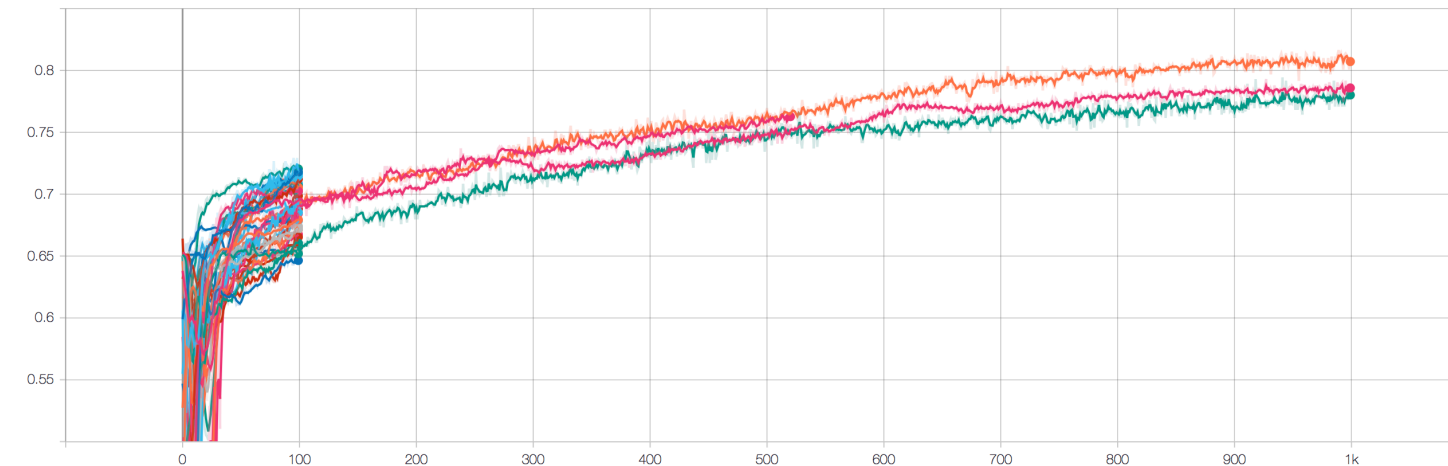
/srv/notebooks/logs/

Filter tags (regular expressions supported)

epoch_accuracy

1

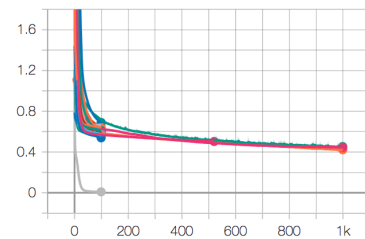
epoch_accuracy



epoch_loss

1

epoch_loss



Logistic Regression Classifier Example

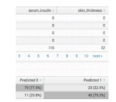
Edit Export ...

Epochs

10000 Submit Hide Filters

Example for PyTorch

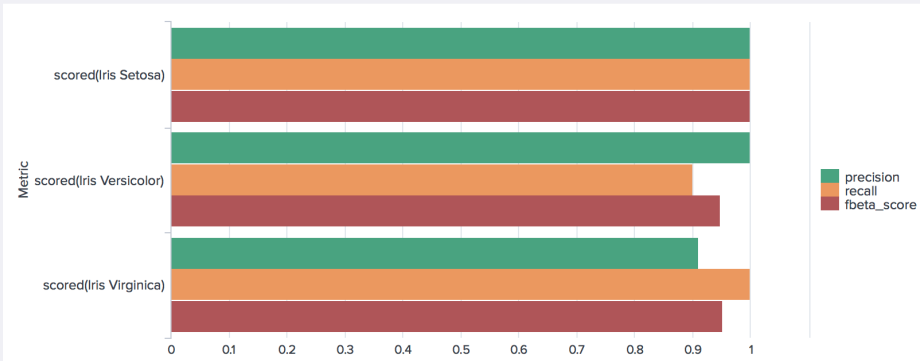
This example shows a simple logistic regression using PyTorch for building a multiclass classifier applied to the favourite iris dataset. The more training epochs you run the model the better classification results get.



Dataset Results and Predictions

species	species_predicted	petal_length	petal_width	sepal_length	sepal_width
Iris Setosa	Iris Setosa	1.4	0.2	5.1	3.5
Iris Setosa	Iris Setosa	1.4	0.2	4.9	3.0
Iris Setosa	Iris Setosa	1.3	0.2	4.7	3.2
Iris Setosa	Iris Setosa	1.5	0.2	4.6	3.1
Iris Setosa	Iris Setosa	1.4	0.2	5.0	3.6

< Prev 1 2 3 4 5 6 7 8 9 10 Next >



Confusion Matrix

	accuracy	precision	recall	f1
	0.9667	0.9697	0.9667	0.9666
Label	predicted(Iris Setosa)	predicted(Iris Versicolor)	predicted(Iris Virginica)	
actual(Iris Setosa)	50	0	0	
actual(Iris Versicolor)	0	45	5	
actual(Iris Virginica)	0	0	50	

File Edit View Run Kernel Tabs Settings Help

/ notebooks /

Name	Last Modified
data	6 days ago
logs	a month ago
barebone.ipynb	6 days ago
default_pytorch.ipynb	12 days ago
default.ipynb	12 days ago
dnn_classifier.ipynb	13 days ago
flair_test.ipynb	8 hours ago
linear_regression.ipynb	13 days ago
linear_regression2.ipynb	13 days ago
pytorch_logistic_regression.ipy...	10 minutes ago
rnn.ipynb	13 days ago
Untitled.ipynb	a month ago

pytorch_logistic_regressor Python 3

Notebook for Deep Learning App for Splunk

Logistic Regression in PyTorch

This notebook contains an example for a simple logistic regression in PyTorch.
By default every time you save this notebook the cells are exported into a python module which is then used for executing your custom model invoked by Splunk MLTK Container App.

Stage 0 - import libraries

At stage 0 we define all imports necessary to run our subsequent code depending on various libraries.

```
[56]: # mltkc_import
# this definition exposes all python module imports that should be available in all subsequent commands
import json
import datetime
import numpy as np
import scipy as sp
import pandas as pd
import torch
# global constants
MODEL_DIRECTORY = "/srv/app/model/data/"
```

```
[57]: # THIS CELL IS NOT EXPORTED - free notebook cell for testing purposes
print("numpy version: " + np.__version__)
print("scipy version: " + sp.__version__)
print("pandas version: " + pd.__version__)
print("PyTorch: " + torch.__version__)
if torch.cuda.is_available():
    print(f"There are {torch.cuda.device_count()} CUDA devices available")
    for i in range(0,torch.cuda.device_count()):
        print(f"Device {i:0}: {torch.cuda.get_device_name(i)} ")
else:
    print("No GPU found")

numpy version: 1.15.4
scipy version: 1.3.1
pandas version: 0.25.0
PyTorch: 1.0.1.post2
No GPU found
```

```
[60]: |
```

0 3 Python 3 | Idle Mode: Edit Ln 1, Col 1 pytorch_logistic_regression.ipynb

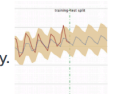
App Expense Forecast using LSTM

Edit Export ...

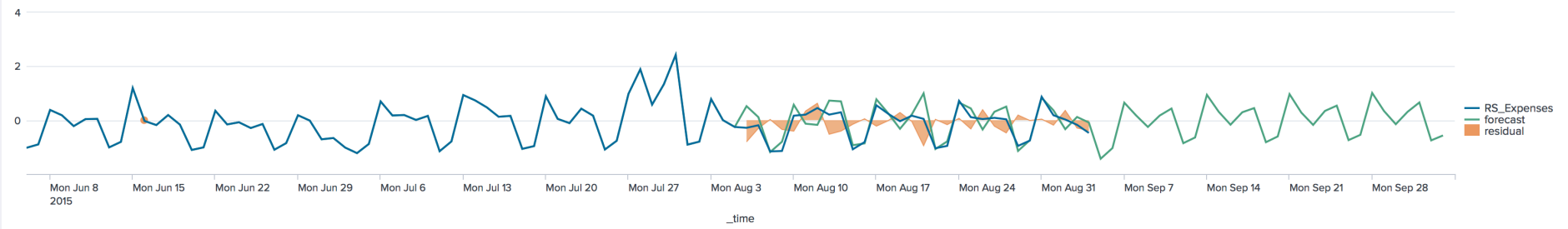
Select Epochs: X Select Batch Size: [Hide Filters](#)

Example for TensorFlow

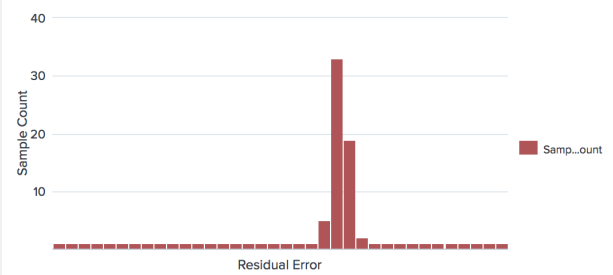
This example shows a recursive neural network (RNN) forecast on univariate data using a long-short term memory (LSTM) approach. Note that the data has also been scaled using a RobustScaler to ensure that the LSTM forecast converges successfully. Make sure to check the [information](#) and [setup](#) page and perform all steps needed to run this dashboard successfully.



LSTM Forecast



Residuals Histogram



R Squared

0.915

Root Mean Squared Error

0.210

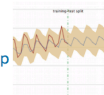
Internet Traffic Forecast using a Convolutional Neural Network

Edit Export ...

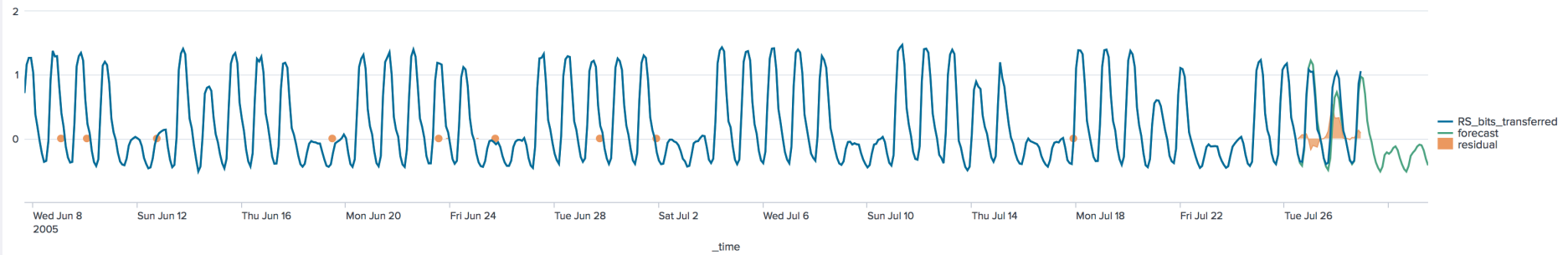
Select Epochs: X Select Batch Size: [Hide Filters](#)

Example for TensorFlow

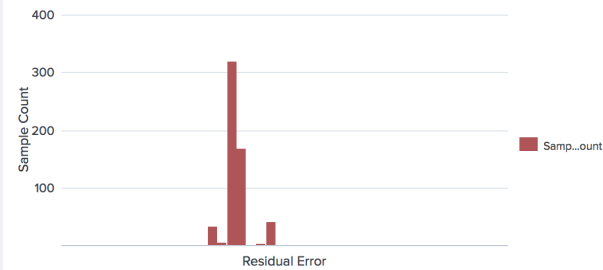
This example shows a convolutional neural network (CNN) forecast on univariate data. Note that the data has also been scaled using a RobustScaler to ensure that the CNN forecast converges successfully. Make sure to check the [information](#) and [setup](#) page and perform all steps needed to run this dashboard successfully.



CNN Model Prediction



Residuals Histogram



R Squared

0.996

Root Mean Squared Error

0.040

Entity Recognition and Extraction Example using the spaCy Library

Edit Export ...

Enter Text:

Baroness is an American heavy metal band from Savannah, Georgia whose original members grew up together in Lexington, Virginia. Baroness formed in mid-2003, founded by former members of the punk/met

Raw Data and Extracted Entities

_time	text	extracted_0
2019-09-13 18:34:46	Baroness is an American heavy metal band from Savannah, Georgia whose original members grew up together in Lexington, Virginia. Baroness formed in mid-2003, founded by former members of the punk/metal band Johnny Welfare and the Paychecks. Singer John Dyer Baizley creates the artwork for all Baroness albums, and has done artwork for other bands.	Baroness:PRODUCT American:NORP Savannah:GPE Georgia:GPE Lexington:GPE Virginia:GPE Baroness:ORG mid-2003:DATE Johnny Welfare:PERSON Paychecks:ORG John Dyer Baizley:PERSON Baroness:NORP
2019-09-13 18:34:46	From 2004 to 2007, Baroness recorded and released three EPs, named First, Second and A Grey Sigh in a Flower Husk (aka Third), with the third one being a split album with Unpersons.	From 2004 to 2007:DATE Baroness:ORG three:CARDINAL First:ORDINAL Second:ORDINAL A Grey Sigh:ORG a Flower Husk:ORG Third:ORDINAL third:ORDINAL Unpersons:ORG
2019-09-13 18:34:46	Baroness started recording their first full-length album in March 2007. Phillip Cope from Kylesa continued to produce Baroness on this album. The Red Album was released on September 4, 2007, and met positive reception. Heavy metal magazine Revolver named it Album of the Year. On December 1, 2007, Baroness performed at New York City's Bowery Ballroom. On September 20, 2008, the band announced via MySpace Brian Blickle would be parting ways with the band, while also introducing a new guitarist named Peter Adams, also of Virginia-based band Valkyrie.	Baroness:ORG first:ORDINAL March 2007:DATE Phillip Cope:PERSON Kylesa:PERSON Baroness:PRODUCT The Red Album:ORG September 4, 2007:DATE December 1, 2007:DATE Baroness:ORG New York City's:GPE September 20, 2008:DATE MySpace Brian Blickle:PERSON Peter Adams:PERSON Virginia:GPE Valkyrie:ORG

< Prev 1 2 3 4 5 6 7 8 9 Next >

Top Entities

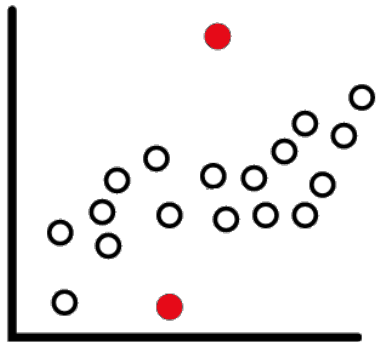
extracted_0	Entity	Entity_Count	Entity_Type	Entity_Type_Count
Baroness:ORG	Baroness	12	ORG	32
Baroness:PRODUCT	Baroness	4	PRODUCT	1
first:ORDINAL	first	4	ORDINAL	9
John Baizley:PERSON	John Baizley	3	PERSON	33
Peter Adams:PERSON	Peter Adams	3	PERSON	33
two:CARDINAL	two	3	CARDINAL	7
9:CARDINAL	9	2	CARDINAL	7
Allen Blickle:PERSON	Allen Blickle	2	PERSON	33
Bath:GPE	Bath	2	GPE	26
Chlorine & Wine':ORG	Chlorine & Wine'	2	ORG	32

Use Cases



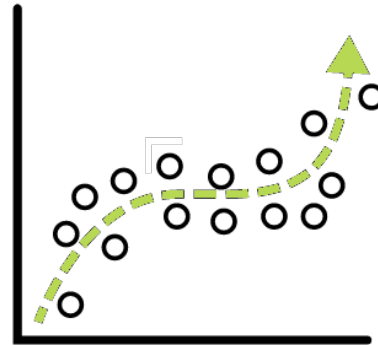
Splunk customers want answers from their data

Anomaly detection



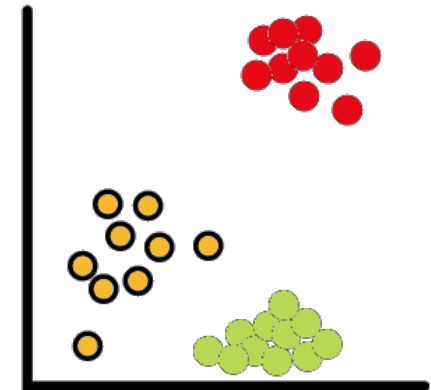
- Advanced Anomaly Detection with Deep Learning Approaches
 - RNN
 - LSTM

Predictive Analytics



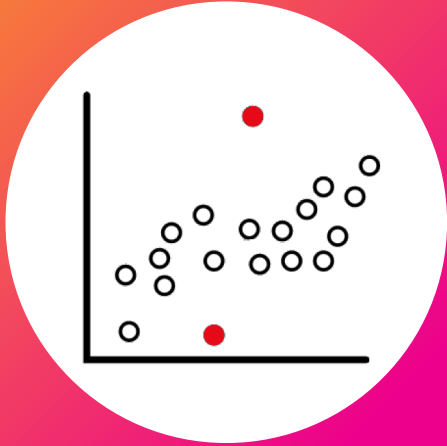
- Deep Learning based Regression and Classification
 - Deep Neural Networks
- Sophisticated Predictive Analytics and Time Series Forecasting
 - RNN
 - LSTM

Clustering



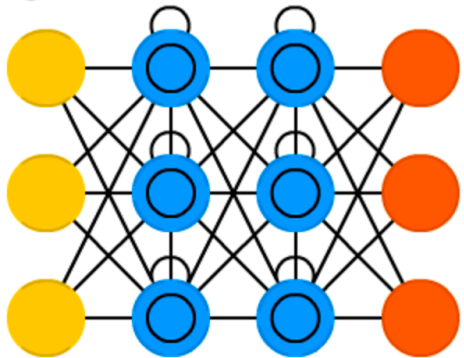
- Deep Learning based approaches
 - Autoencoder
 - Variational Autoencoder

Anomalous Access Patterns



- ▶ **Goal:** Identify entities with deviations from past observations
- ▶ **Plain Language:** Finding Users, KPIs or Devices that are acting differently than they have in the past.
- ▶ **Example:** A user authenticates to a higher than normal number of servers using administrative credentials.
- ▶ LSTM / RNN neural networks are good for learning the historical and contextual patterns in your data.
“decisions from past iterations or samples can influence current ones”

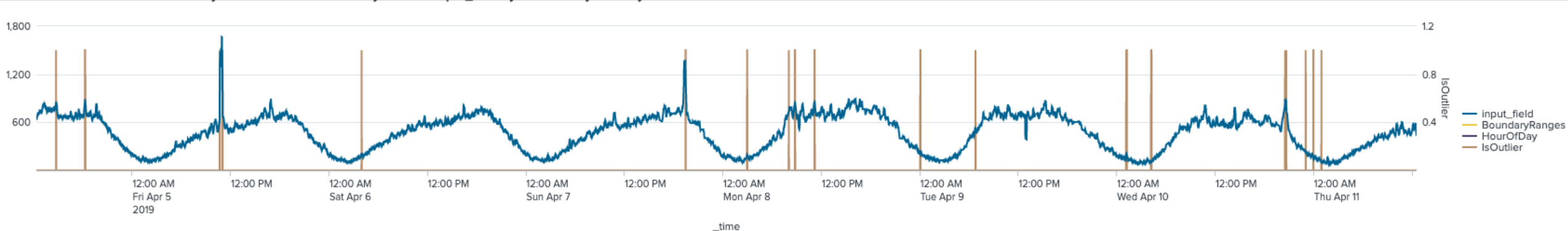
Long / Short Term Memory (LSTM)



<https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

Why RNN/LSTM?

- ▶ **Basic Idea:** LSTM Learns Patterns in the data.
 - Business Rules: Day of Week, Time of Day, & Seasonality
- ▶ Model is used to forecast KPI's value the next minute
 - If the actual value is within 1 standard deviation = **√OK**
 - If the actual value is outside 1 standard deviation = **ANOMALY!**

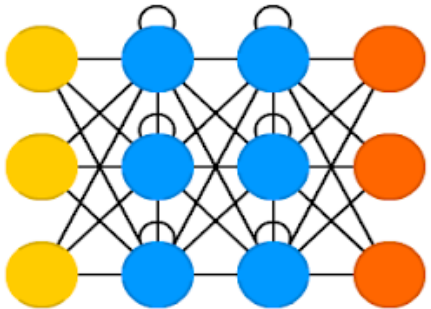


<https://medium.com/datadriveninvestor/lstm-neural-networks-for-anomaly-detection-4328cb9b6e27>

Predictive Maintenance at Volkswagen



Recurrent Neural Network (RNN)



- ▶ Detect wear of industrial equipment that is used in car assembly lines
- ▶ Integration of custom recurrent neural network (RNN) models to detect deviations in operational behavior of industrial equipment
- ▶ Visualize results to operators on real time dashboards and generate alerts based on anomaly scores of the RNN models being continuously applied to live data

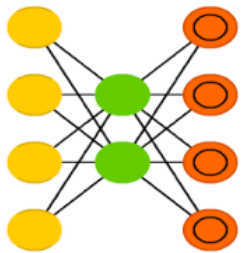
FRAUD

Autoencoder Example

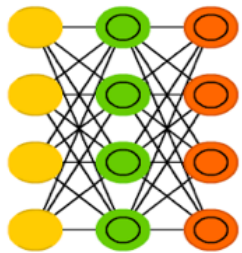


- ▶ **Goal:** Identify transactions with unexpected values or behaviors in the data using reconstruction error
- ▶ **Plain Language:** Finding transactions that are abnormal compared to some set of normal transactions.
- ▶ **Example:** a user transaction request to an unexpected merchant.
- ▶ Train an autoencoder to understand how the different features in our data are related to each other.
- ▶ **“Predict”** transaction_amount, merchant_id, account_id, channel, zipcode, etc.

Auto Encoder (AE)



Variational AE (VAE)

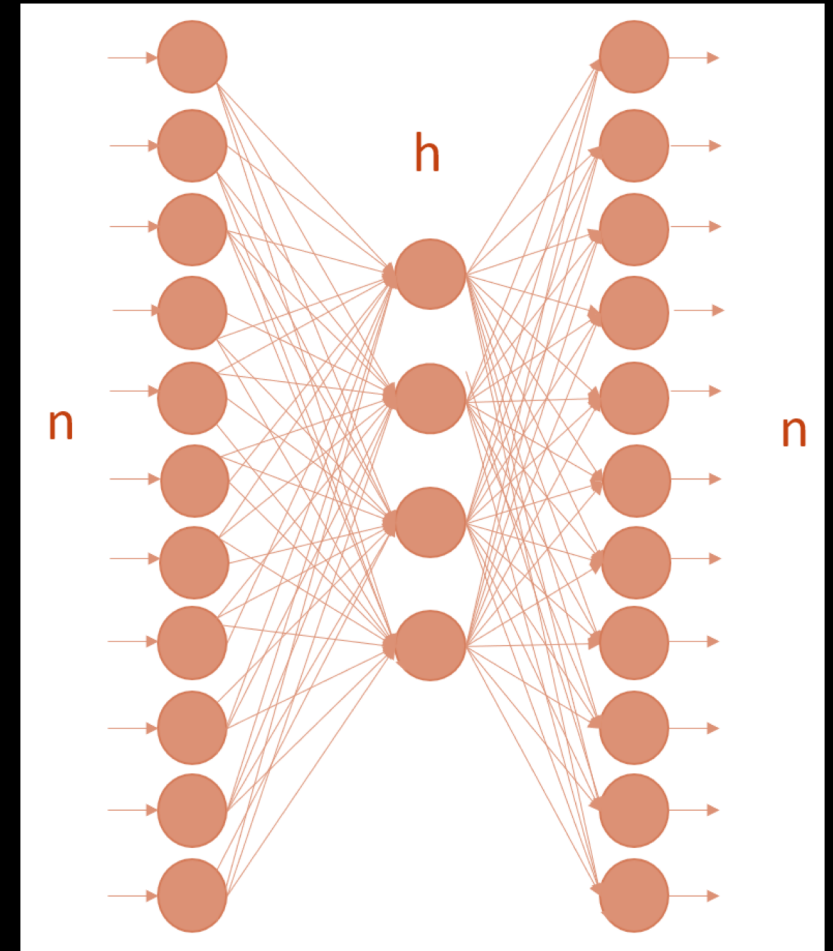


<https://www.dataversity.net/fraud-detection-using-a-neural-autoencoder/>

<https://www.kaggle.com/mlg-ulb/creditcardfraud>

Why AutoEncoder?

- ▶ **Basic Idea:** Train model on legitimate transactions
 - Train algorithm to reproduce the feature vectors of each transaction
 - Each feature input, will map to an output value.
 - n input variables = n output variables
- ▶ Model is used to reconstruct **transaction** x_k
 - **Reconstruction Error** ε_k is calculated as distance between
 - **original** x and **reproduced** \hat{x} transaction
 - $x_k \rightarrow$ "normal" if $\varepsilon_k \leq K$
 - If the distance between x and \hat{x} is small = **OK**
 - $x_k \rightarrow$ "anomaly" if $\varepsilon_k > K$
 - If the distance between x and \hat{x} is above threshold K = **FRAUD!**
 - Optimize for **threshold value** K against **reconstruction error** ε using the validation set



$$\varepsilon = \frac{1}{N} (x - \hat{x})^2$$

Wrap up



Key Takeaways

Deep Learning Toolkit
for Splunk

1. Extend your Splunk platform with the Deep Learning Toolkit for Splunk
2. Integrate custom advanced deep learning and NLP models into Splunk using a predefined Jupyter Notebook workflow for rapid model development.
3. Leverage GPUs for compute intense training tasks

Deep Learning Toolkit for Splunk

Download from SplunkBase

- ▶ If you're a Splunk admin you can download the DLTK:
- ▶ splunkbase.splunk.com/app/4607
- ▶ Utilize the install guide
 - Github docs placeholder
- ▶ Reach out to Sales & PS for additional support needs
- ▶ Splunk Answers

Consult Professional Services

- ▶ Machine Learning & Analytics Workshop is a paid offering where Splunk's expert consultants help you:
 - Configuration of Splunk Enterprise, MLTK & DLTK
 - Define use case criteria & key business objectives of the ML use case
 - Prepare your data for machine learning
 - Implement use cases
 - Provide guidance on additional value and use cases based on the customer's data sources

Q&A

.conf19
splunk>





**Thank
You!**