Machine Learning + DevOps (MLOps): Road to singularity ... just kidding

... the road to intelligent services

Sasa Savic
Principal Engineer @ Telstra | Co-Founder @ Qooee
GANs
Generative Adversarial Networks

Ref: http://www.asimovinstitute.org/neural-network-zoo/
Melbourne — March 22-23
The Starry Night, Van Gogh 1889
This bird has a yellow belly and tarsus, grey back, wings, and brown throat, nape with a black face.
“I need highly available infrastructure, across multiple regions to host my web application that will serve 1 million users.”

“Fun to imagine”, Richard Feynman
A very brief introduction ...

Co-Founder

B.Sc., M.Sc., Ph.D.-Candidate Comp Sci.

Data Science

Cloud

Principal Engineer

DevOps
Machine Learning + DevOps (MLOps): Road to singularity ... just kidding

... the road to intelligent services
Machine Learning

Ability for a computer system to:

- Learn from the environment
- Improve itself from experience
- Do all of the above, without the need for any explicit programming

It focuses on:

- Enabling algorithms to learn from data
- Making predictions on previously un-analyzed data

Reference:
ML Applications: https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications
Machine Learning ( ... a branch of AI )
Singularity ...

Reference:
Singularity (Black Hole): https://www.physicsoftheuniverse.com/topics_blackholes_singularities.html
Singularity ...

Reference:
AUTO-CORRECT HAS TROUBLE WITH THE WORD "ISN'T"

TELL ME AGAIN HOW THE MACHINES WILL ONE DAY TAKE OVER

Melbourne — March 22-23
Melbourne — March 22-23

ML Ops

Learning & Development

DevOps

Value

Services & Tools

Talent
Talent

The expensive model

(Big)Data Engineer provides data ...
Data Cleaner cleans data ...
Data Scientist builds models ...
DevOps runs models ...

Melbourne — March 22-23
The less expensive model

(Big)Data Engineer provides data...
Data Scientist cleans data... builds models...
DevOps runs models...
Talent

The cheap model

(Big)Data Engineer

data ... provides data ...

Data Scientist
data ... cleans data ...
models ... builds models ...

ML Engineer
models ... builds models ...
models ... runs models ...

The cheap model provides data, the data is cleaned by the data scientist, and then models are built and run by the ML engineer.
The no-frills model

- provides data...
- cleans data...
- builds models...
- runs models...

ML Engineer
Learning & Development

- Large companies underinvest in personal development
- Employees are too busy to use personal time upskilling
- No formal recognition for personal initiative
Learning & Development

PERSONAL DEVELOPMENT

• Keen people up-to-date on relevant technologies
• Make ML fun! Gamification, Hackathons, Conferences

CUSTOMISED LEARNING

• Classroom training and online courses
• Study groups and facilitated working groups
• Conferences etc.

DEDICATED FUNDING

• It won’t happen on its own.
I know how to write software

I know how to maintain software

I can get trained on using Frameworks

ML Engineer

Melbourne — March 22-23
Value

• Business Problem Definition (Needs to be framed as a ML problem)
  • What is the problem?
    • Informal description
    • Formal description
    • Assumptions
    • Similar problems
    • Description of provided data
  • Why does the problem need to be solved?
    • Motivation
    • Benefits & Value
    • Use
  • How would I solve the problem (manually)?
Value

• How is value realized?
  • Immediately
  • Continuously
  • Over a period of time
  • Future

• How are benefits manifested?
  • Simplification
  • Reduction of labour
  • Insight
  • Savings
  • New revenue streams
  • Services, tools or platforms
DevOps

- Ingestion: Collection, Integration
- Preparation: Encoding, Normalization, Scaling
- Visualization & Analysis: Feature Engineering, Visualization, Analysis
- Train Model: Hyperparam., Data & Model Parallelization, Model
- Deploy Model: Front-end, Back-end, Predictions
- Retraining
- ETL

Data Lake

ETL

Training Infrastructure (GPU)

Serving Infrastructure

ML Pipeline
DevOps

1. train model
2. persist model
3. commit code
4. app image
5. update service
6. load model
7. serve prediction
8. feedback

CICD Pipeline

CloudWatch

Amazon ECR

Model Bucket

Applications Load Balancer

Route 53

mobile client

DevOps

Data Scientist

EC2

Auto Scaling

Spot Fleet

CloudWatch

CodePipeLine

CodeCommit

CodeBuild

CodeDeploy

CICD Pipeline

Melbourne — March 22-23
Tools & Services ... up and running quickly on AWS

AWS ML Stack

Application Services
- API Driven Services: Vision, Language & Conversational

Platform Services
- Deploy High Performance ML Algorithms, Diverse Framework Support, quick Training, Tuning and Inference

Infrastructure Services
- Develop sophisticated models, create managed auto-scaling clusters of GPUs for large scale training
Tools & Services ... up and running quickly on AWS

SageMaker

- Notebook instance: Explore AWS data in your notebooks, and use algorithms to create models via training jobs.
- Jobs: Track training jobs at your desk or remotely. Leverage high-performance AWS algorithms.
- Models: Create models for hosting from job outputs, or import externally trained models into Amazon SageMaker.
- Endpoint: Deploy endpoints for developers to use in production. A/B Test model variants via an endpoint.
Tools & Services ... getting up and running quickly

Deep Learning

Data Science

Exploratory Data Science

Visualization

Melbourne — March 22-23
Get up and running REALLY quickly on your own environment...

- virtualenv -p python3 myproject
- source myproject/bin/activate
- pip install flask gevent requests pillow scipy numpy matplotlib pandas \ sklearn tensorflow keras h5py

- vi project.py
Binary Classification Demo (Supervised)

(Customer Domain – Churn Prediction)
Let’s feed it some raw data ...

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Location</th>
<th>Gender</th>
<th>Age</th>
<th>Salary</th>
<th>Experience</th>
<th>Location</th>
<th>Gender</th>
<th>Age</th>
<th>Salary</th>
<th>Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Smith 619</td>
<td>Sydney</td>
<td>Female</td>
<td>41</td>
<td>101238.81</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Smithson 608</td>
<td>Brisbane</td>
<td>Male</td>
<td>41</td>
<td>81807.51</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Savic 502</td>
<td>Sydney</td>
<td>Male</td>
<td>42</td>
<td>159660.8</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Halgoft 699</td>
<td>Sydney</td>
<td>Female</td>
<td>39</td>
<td>93826.63</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Goulko 850</td>
<td>Brisbane</td>
<td>Male</td>
<td>43</td>
<td>125510.82</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9995</td>
<td>Xen 800</td>
<td>Sydney</td>
<td>Female</td>
<td>29</td>
<td>167773.55</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9996</td>
<td>Harroldson 771</td>
<td>Sydney</td>
<td>Male</td>
<td>39</td>
<td>96270.64</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9997</td>
<td>Laurie 516</td>
<td>Sydney</td>
<td>Female</td>
<td>25</td>
<td>90369.61</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9998</td>
<td>Georgiou 709</td>
<td>Sydney</td>
<td>Female</td>
<td>36</td>
<td>101699.77</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999</td>
<td>Wilson 772</td>
<td>Melbourne</td>
<td>Male</td>
<td>42</td>
<td>92888.52</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>Croft 792</td>
<td>Sydney</td>
<td>Female</td>
<td>28</td>
<td>130287.33</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# import libs
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import keras

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.metrics import confusion_matrix
from keras.models import Sequential
from keras.layers import Dense

# import data set
ds = pd.read_csv('data/raw_data.csv')

# create matrix of features and target
X = ds.iloc[:, 3:13].values
y = ds.iloc[:, 13].values

# encode categorical features
encoder_X_location = LabelEncoder()
encoder_X_gender = LabelEncoder()
onehotencoder = OneHotEncoder(categorical_features = [1])
X[:, 1] = encoder_X_location.fit_transform(X[:, 1])
X[:, 2] = encoder_X_gender.fit_transform(X[:, 2])
X = onehotencoder.fit_transform(X).toarray()
X = X[:, 1:]

# split the dataset into train and test (80/20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

# scale our features
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# init NN, add input and hidden layers and an output layer
model = Sequential()
model.add(Dense(units=6, kernel_initializer='uniform', activation='relu', input_dim=11))
model.add(Dense(units=6, kernel_initializer='uniform', activation='relu'))
model.add(Dense(units=1, kernel_initializer='uniform', activation='sigmoid'))

# compile
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# predicting the test results
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5)

# confusion matrix
cm = confusion_matrix(y_test, y_pred)
accuracy = (cm[0,0] + cm[1,1]) / y_pred.size
print('Accuracy:', round(accuracy*100,2))
Categorical Classification Demo (Supervised)

Assurance Domain – Image Classification
Let’s feed it some labelled image data …

1, data/modem_img_1.png, misconfigured
2, data/modem_img_2.png, misconfigured
3, data/modem_img_3.png, connected
4, data/modem_img_4.png, no_internet
5, data/modem_img_5.png, connected
6, data/modem_img_6.png, misconfigured

... 
... 

497, data/modem_img_1.png, bad_gateway
498, data/modem_img_1.png, internet_connectivity
499, data/modem_img_1.png, bad_gateway
500, data/modem_img_1.png, connected
Load

Expose as an API
```bash
~/tensorflow/images
curl -X POST -F image=@faulty_modem.jpg 'http://127.0.0.1:5000/predict'
{
  "predictions": [
    {
      "label": "misconfigured",
      "probability": 0.9901360869407654
    }
  ],
  "success": true
}"
Clustering Demo (Unsupervised)

SRE Domain – Service Degradation Cohorts
# import libraries
import numpy as np, pandas as pd, seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.cluster import KMeans

# load data
data = pd.read_csv('data/microservice_data.csv', sep=',')

# create dummy columns for categorical data
data = pd.get_dummies(data, columns=['day_part'])

# standardize
columns = ['payload_size', 'latency',
            'day_part_Afternoon', 'day_part_Evening', 'day_part_Morning']
data_std = stats.zscore(data[columns])

# cluster the data
kmeans = KMeans(n_clusters=3, random_state=0).fit(data_std)
labels = kmeans.labels_

# extend the original dataset with labels
data['clusters'] = labels
columns.extend(['clusters'])

# display
sns.scatterplot('payload_size', 'latency', data=data, fit_reg=False, hue='clusters', scatter_kws={'marker': 'D', 's': 100})
plt.title('Payload vs. Latency')
plt.xlabel('payload_size')
plt.ylabel('latency')
plt.show()

input('Press <ENTER>')
Courses

Intro to Data Science in Python (Michigan) – Coursera
https://www.coursera.org/learn/python-data-analysis

Machine Learning (Stanford) – Coursera
https://www.coursera.org/learn/machine-learning

Machine Learning Crash Course (TensorFlow API) – Google
https://developers.google.com/machine-learning/crash-course/
Thank You

GitHub: @sasasavic82
Twitter: @sasasavic82
LinkedIn: https://www.linkedin.com/in/sasasavic/
E-Mail: sasa.savic@team.telstra.com