Using **Big Data, Cloud and AI** to Enable **Intelligence @ Scale**

Danielle Dean, Wee Hyong Tok  
Data Scientist Leads  
Algorithms and Data Science  
Microsoft

@weehyong  
@danielleodean
Why does **AI** matter to the **Big Data** Community?
Major Generators of Big Data

1.4 PB/year
Twitter

1 EB/year
Astronomy

1-2 EB/year
YouTube

2 – 40 EB/year
Genomics

Projected Annual Storage (2025)


* Not to scale
My Data

Organization’s Data

Database / Data Warehouse
Web logs
Call Logs Data

Images
Video

My Company
What has changed since mid-1980s?
Data
Scale
Overview

Train/Create New Models → Ready-to-use AI → Leverage what’s already built → Ready-to-use AI Apps → Solutions
Intelligence in Every Software
What can AI do?
Using AI to help measure tumors in ways that doctors cannot
Using AI for Realtime ID Check
Deep Learning for automated inspections
Show me AI ...
Demo
How can you build AI applications?
Give your apps a human side

Vision
From faces to feelings, allow your apps to understand images and video

Speech
Hear and speak to your users by filtering noise, identifying speakers, and understanding intent

Language
Process text and learn how to recognize what users want

Knowledge
Tap into rich knowledge amassed from the web, academia, or your own data

Search
Access billions of web pages, images, videos, and news with the power of Bing APIs
Demo

Cognitive Services
Demo

Real-time Video Intelligence
Intelligent solutions will enable differentiation

Cognitive Understanding

- Identify objects, people and actions
- Hear and recognize language
- Infer emotions and reactions
- Develop deeper context & understanding over time

Conversation as a platform

- Natural language conversational UI
- On any canvas e.g. Skype, Slack, Facebook, etc.
- Powered by Intelligent Bots
- Accessible through personal digital assistants
How can you build your own AI?
Cortana Intelligence Suite
Transform data into intelligent action

Data Sources
- Data Factory
- Data Catalog
- Event Hubs

Apps
- Data Lake Store
- SQL Data Warehouse

Big Data Stores
- Machine Learning
- Data Lake Analytics
- HDInsight
- Stream Analytics

Machine Learning and Analytics
- Cognitive Services
- Bot Framework
- Cortana
- Power BI

Intelligence
- Dashboards & Visualizations

People
- Web
- Mobile
- Bots

Automated Systems

Data
- Sensors and devices

Data
- Information Management

Intelligence
- Apps

Action
Data Science Virtual Machine

- Custom VM image on Azure Marketplace
- Contains a set of data science, Azure tools/SDKs
- All pre-configured and ready to use
- Pay for cloud hardware usage only. No separate software charges!
- Pointers to gallery, samples, documentation
- Windows and Linux Versions

Build your first model in 30 minutes or less!
Try the DSVM for Free today...
http://aka.ms/dsvm
http://aka.ms/dsvmhandout
http://aka.ms/dsvmtenthings
Deep Learning Toolkit on Data Science VM

- Comes with a number of Deep Learning framework already installed
- Bootstrap your Deep Learning exploration
Lung Cancer Detection

Based on work by Miguel Fierro, Ye Xing, Tao Wu
Kaggle Data Science Bowl 2017

Determine whether a patient has cancer

Jumpstart in less than 1h
Top 10% (Jan 19\textsuperscript{th})
Solution Pieces

Data Science VM (GPU)
Microsoft Cognitive Toolkit (CNTK)
LightGBM for classification
Deep Neural Nets
ResNet, 152 layers

Microsoft
Solution: Transfer Learning

ImageNet ResNet N layers

224
224
3

penultimate layer
last layer

tabby cat
Solution: Transfer Learning

- ResNet N-1 layers
- CNTK (53min)
- LightGBM (2min)
- no cancer

k batch of images = 1 patient

Boosted tree

features

penultimate layer
Challenges and Improvements

Nodule detection + classification

• Use other pretrained CNNs such as AlexNet, AlexNet with Batch Normalization and ResNet with 18 layers.

• Perform image augmentation: You can try to increment the training set by performing transformations in the images. For that you can apply different filters or rotate them.

• Use a customize network with 3D images: CNTK allows 3D convolutions. On GPU, 1D, 2D and 3D convolutions will use cuDNN (fast), all other convolutions will use reference engine (slow). You can create a CNN that accepts 3D images.

• Try some feature engineering: Before training the tree, the features are averaged. You can try to feed the tree without this operation or try a different one.

How do I get started?

Kaggle script
https://aka.ms/dsb2017-cntk-script

Cortana Gallery notebook
https://aka.ms/dsb2017-cntk-notebook

Blog in TechNet
https://aka.ms/dsb2017-cntk-blog
Cloud Scale Text Classification

Deep Learning and Sentiment Analysis

Cloud Atlas is a very interesting piece. I really like the adaptation. It respects the format and the atmosphere of the book.

Categories

- Books
- Electronics
- Home & Kitchen
- Clothing, Shoes & Jewelry
- Movies & TV
- Health & Personal Care
- Sports & Outdoors

Sentiment

- crepe: 14% (99%)
- Cortana: 14% (87%)

Cloud Scale Text Classification

Text Classification with Convolutional Neural Networks at the Character Level

**Input text:**
a cat is playing with a toy

**Matrix representation of input text**

**Scheme of Crepe CNN.**
CAP stands for Convolution, Activation and Pooling. CA stands for Convolution and Activation. FC stands for Fully Connected. This scheme assumes a batch size of 1.

Trained with R API of MXNet on high-performance GPU DSVMs by distributed computation with 4 GPUs
Aka.ms/hostDNN
Architecture

Clients → POST → API → Response

- Image Classification, CNTK
- Virtual machine scale set
- Cognitive Services (API)
Example Production Workflow

Big Cognition

Language
- Speech
- Customer feedback analysis
- Recommendation API
- Custom recognition (CRIS)

Knowledge
- Bing news search
- Text analytics
- Thumbnail generation
- Web language model
- Anomaly detection
- Sentiment scoring
- Bing autosuggest

Search
- Computer vision
- ORC, tagging, captioning
- Bing web search

Vision
- Bing image search
- Speech API
- Text to speech
- Spell check
- Academic knowledge
- Emotion

Machine Learning
- Entity linking
- Forecasting
### Job Summary

- **Preparation**: 23 seconds
- **Queue**: 0 seconds
- **Running**: 24 seconds

**Job Result**: Succeeded

**Total Duration**: 1.1 minutes
**Total Compute Time**: 19 seconds

- **Submit Time**: 2/7/2017 11:15:29 PM
- **Start Time**: 2/7/2017 11:16:20 PM
- **End Time**: 2/7/2017 11:16:44 PM

**Compilation**: 23 seconds
**Queued**: 0 seconds
**Running**: 24 seconds

**Account**: ussst Team Demos
**Author**: saelipi@microsoft.com
**Priority**: 1000
**Parallelism**: 5

**Bytes Left**: 0
**Bytes Read**: 514,254,293
**Bytes Written**: 79,629,414

**Total Vertices**: 36
**Completed**: 36
**Failed**: 0

---

### Job Graph

- **SV3 Extract**
  - 25 vertices
  - 538 edges
  - 791,829 rows written

- **SV4 PodAggregate**
  - 3 vertices
  - 112,760 rows written

- **SV5 Process**
  - 5 vertices
  - 9,082 edges
  - 9,831 rows written

- **SV6 Aggregate**
  - 1 vertex
  - 802 vertices
  - 9 bytes

- **SV7 Process**
  - 1 vertex
  - 678 edges
  - 19 bytes

- **SV8 Aggregate**
  - 1 vertex
  - 100 vertices
  - 9 bytes

- **SV9 Process**
  - 1 vertex
  - 335 edges
  - 11 bytes

- **SV10 Aggregate**
  - 1 vertex
  - 100 vertices
  - 9 bytes

- **SV11 Combine**
  - 1 vertex
  - 100 vertices
  - 9 bytes

**Job Playback**

- **Step 1**: 0.00:10
// Extract key phrases for each paragraph.

@keyphrase =

PROCESS CallEngineersLog |
PRODUCE Product,
NoteCreatedOn,
ServiceRequestNumber,
Text,
KeyPhrase string 
READONLY Product,
NoteCreatedOn,
ServiceRequestNumber,
Text 

OUTPUT @keyphrase TO @"/users/SupportCalls/Results/supportkeyphrase.csv"
ORDER BY ServiceRequestNumber 
USING Outputters.Csv();
for(product in products){
    # filter for SQL Server 2012
    callsASQL <- callVolumeDaily2[callVolumeDaily2[,callVolumeDaily2$V3 == product,]
    callsASQL <- callVolumeDaily2[callVolumeDaily2$V3 == product,]

    # order to get the time series
    yraw <- ts(data = callsASQL[['V4']], start = 2015, frequency = 12)
    y <- log(yraw)
    # model, predict and store
    m1 <- arima(y, order = c(1, 0, 1), include.mean = TRUE)
    f1 <- predict(m1, n.ahead = 3)
    flevel <- ceiling(exp(f1$pred))
    actualsAndPredicted[[product]] <- c(yraw, flevel)
}

output2USQL = actualsAndPredicted
output2USQL
#write.csv(output2USQL, "out.csv")
Your bots – wherever your users converse

Microsoft Bot Framework

Build and connect intelligent bots to interact with your users naturally wherever they are, from text/sms to Skype, Slack, Office 365 mail and other popular services.

Get started

- Bot Connector Service: A service to register your bot, configure channels and publish to the Bot Directory. Connect your bot(s) seamlessly to text/sms, Office 365 mail, Skype, Slack, Twitter, and more.
- Bot Builder SDK: An open source SDK hosted on GitHub. Everything you need to build great dialogs within your Node.js or C# bot.
- Bot Directory: A public directory of bots registered through the Bot Connector Service. Discover, try, and add bots to conversation experiences.
Summary

AI -> Ready-to-use AI

Train/Create New Models
Leverage what's already built

Ready-to-use AI Apps

Solutions
Thank you!