SAPIENT
RAZORFISH

Deep Learning-based Search and Recommendation systems using TensorFlow

Dr. Vijay Agneeswaran
Abhishek Kumar

MARCH 06H, 2018

Session Logistics

1. Access the work environment using the following link
2. Code and presentation available at link : [http://bit.ly/strata-dl-ca-2018]
3. Connecting to the speakers [*Please send introductory note*]
    1. Dr. Vijay Agneeswaran ( http://bit.ly/vijaysa  **@a_vijaysrinivas**)
    2. Abhishek Kumar ( http://bit.ly/kumarabhishek  **@meabhishekkumar**)
4. Don’t forget to tweet **#stratadata**

About the Speaker

DR. VIJAY AGNEESWARAN

Senior Director and Head of Data Science,

SapientRazorfish

MS (Research) & PhD , IIT Madras

Post doctoral research fellowship, LSIR Labs

Professional member : ACM, IEEE (Senior)

4 Full US Patents and multiple publications (including IEEE journals)

Regular Speaker @ O'Reilly Strata conference

About the Speaker

ABHISHEK KUMAR

Senior Data Scientist , **SapientRazorfish**

Masters from **University of California, Berkeley**

Pluralsight Author

- Doing Data Science with Python
- R Programming Fundamentals
- Machine Learning with ENCOG
- Currently authoring : "*Deploying Machine Learning Models with Tensorflow Serving*"

Audience Profiling

1. Machine Learning ?
2. Deep Learning ?
3. Search and Recommendation Systems ?
4. Tensorflow ?

Session Agenda

4 Levels of Learning

FOUNDATION [30 MINS]

1. High Level Overview For Problem Space [Search, Recommendations, Learning to Rank]
2. Deep Learning Primer
3. Why Tensorflow for Deep Learning ?

Search [1 HR]

1. Embedding
2. Demo : Embedding in TF
3. Image Search using CovNet
4. Demo : Image Search in TF

BREAK

Recommendation [1 HR]

1. Embedding in RecSys
2. Demo : Build DL based RecSys with Explicit Feedback using TF
3. Demo : Build hybrid RecSys using TF
4. Learning on Rank
5. Demo : Build DL based RecSys with Implicit Feedback using TF

PRODUCTION [30 Mins]

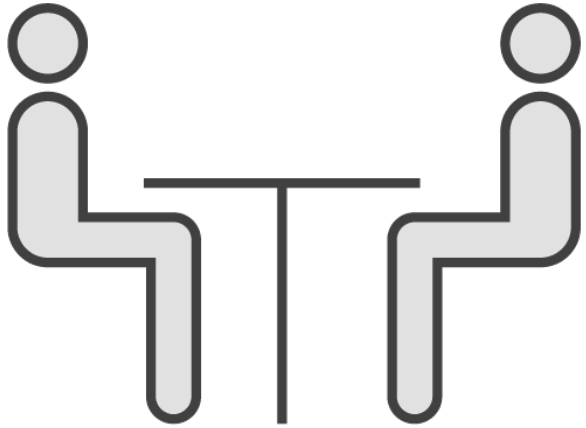
1. TF in Production : Training & Inference
2. Tensorflow Serving
3. RecSys Architecture

By the end of this session...

1. You will have solid foundation of Deep Learning.
2. You will have good understanding of Recommendation System , Search and Ranking System
3. You will be able to transform the concepts and build DL models using Tensorflow
4. You will have high level idea to take the lab scale solution to a production ready system

Quick Chat with Your Neighbor

1. Introduce yourself to your neighbor
2. What they are looking to learn from the tutorial



Problem Space : Search and Recommendation

We now live in the connected age

1900

Manufacturing Economy

Mass manufacturing makes industrial powerhouses successful

"A customer can have a car painted any color he wants as long as it's black"

1960

Distribution Economy

Global connections and transportation systems make distribution key

"Strategy is…globalization, taking your products around the world; be the low-cost producer"

1990

Information Economy

Connected PCs and supply chains mean those that control information flow dominate

"The great challenge… is to make productive the tremendous new resource, the knowledge worker. "

2008+

Connected Economy

iPhone and Facebook launch in 2007/8 heralding a new era in transparency, empowerment, and experimentation.

"The customer is the center of your universe."

**Adapted from Forrester Research "Age of the Customer" graphic*

The Connected Consumer is in charge

Empowered and demand transparency

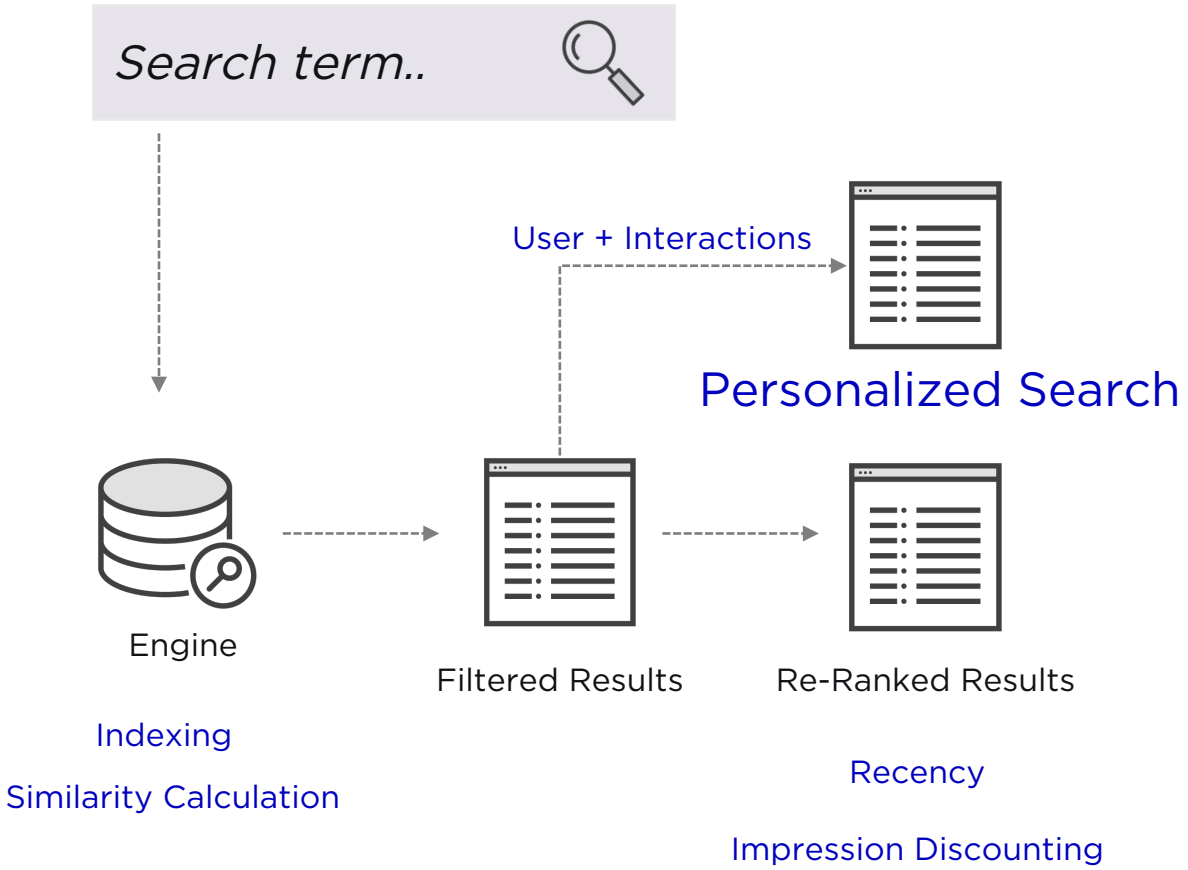
Value experiences over most things

Embrace and seek new companies to engage with

Demand personalization and real-time relevancy

Problem Space : Search

Search Engines



Challenges

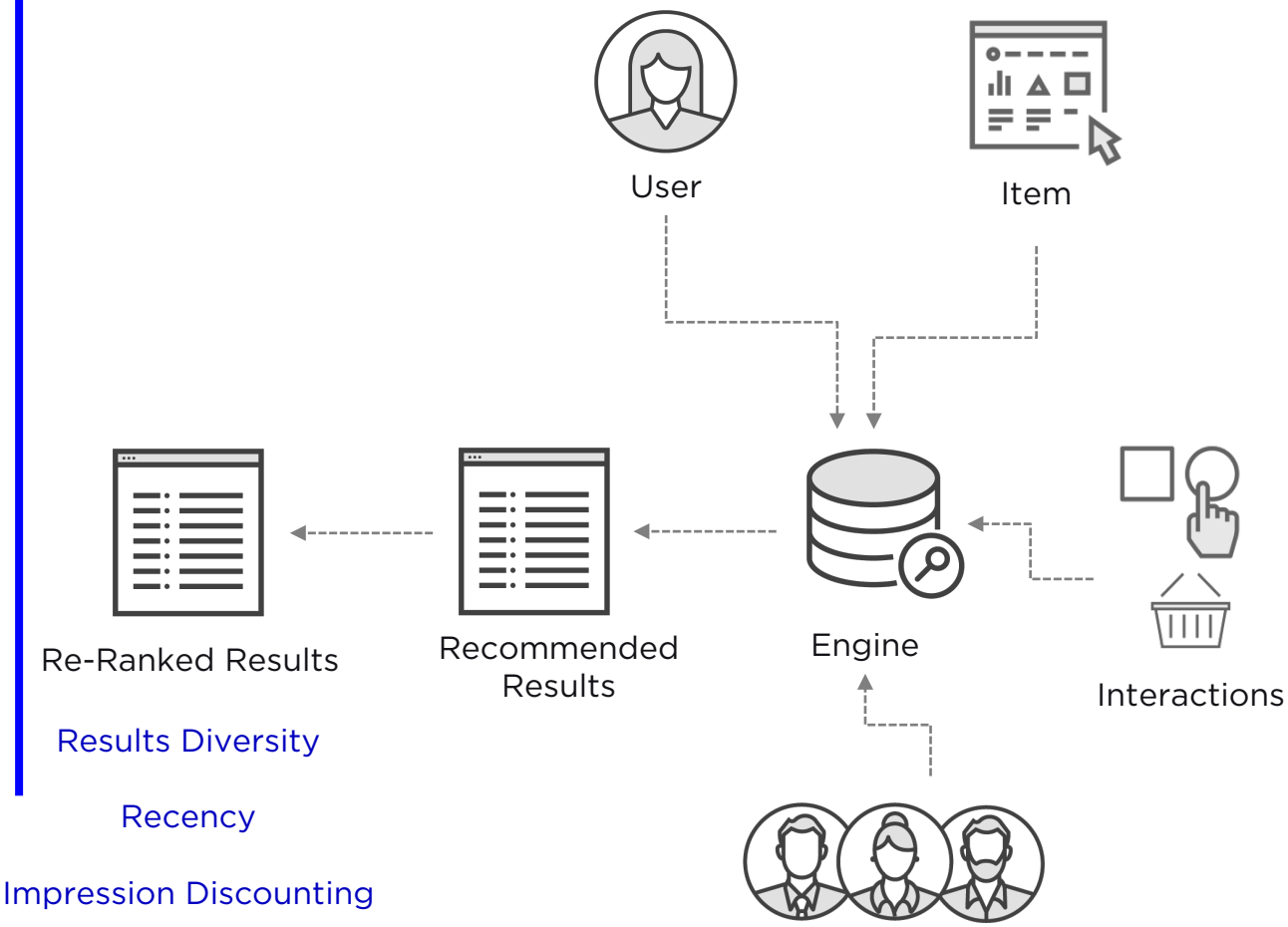
- How to represent text, images, audios
  - TF-IDFs ?
  - Metadata for non-text ?
- Search in other languages ?
- Search Quality
  - Well-ranked results
  - *By providing better search results, Netflix estimates that it is avoiding canceled subscriptions that would reduce its revenue by $1B annually. [Link]*

Problem Space : Recommendation

Challenges

- How to represent users and items ?
- How to build hybrid systems with both interactions (collaborative) and user/item metadata ?
- How to use dynamic user behaviors ?
- How to use implicit (view, share) feedback ?

Recommendation Engines



Results Diversity

Recency

Impression Discounting

Why Deep Learning for Search and Recommender System?

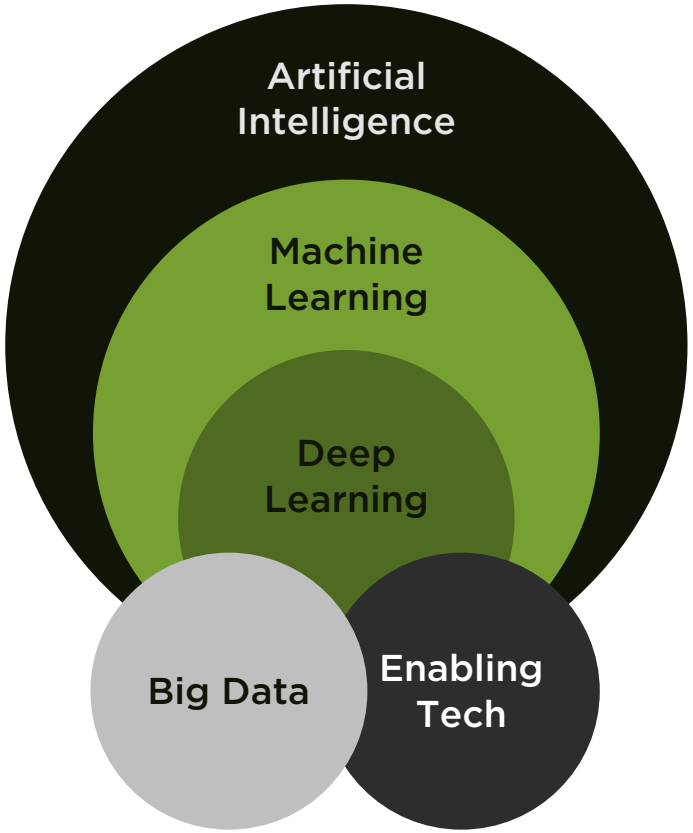
- Direct content Feature extraction instead of metadata
  - Text, Image, Audio
- Better representation of users and items for Recsys
- Hybrid algorithms and heterogeneous data can be used
- Better suited to model dynamic behavioral patterns

Deep Learning Primer

What is Deep Learning ?

Class of machine learning algorithms

- That uses hierarchy of non-linear processing layers and complex model structures
- Layers learn to represent different representation of data
- Higher level features are constructed from lower level abstract features
- Trendy name for “Neural Networks with deep layers”



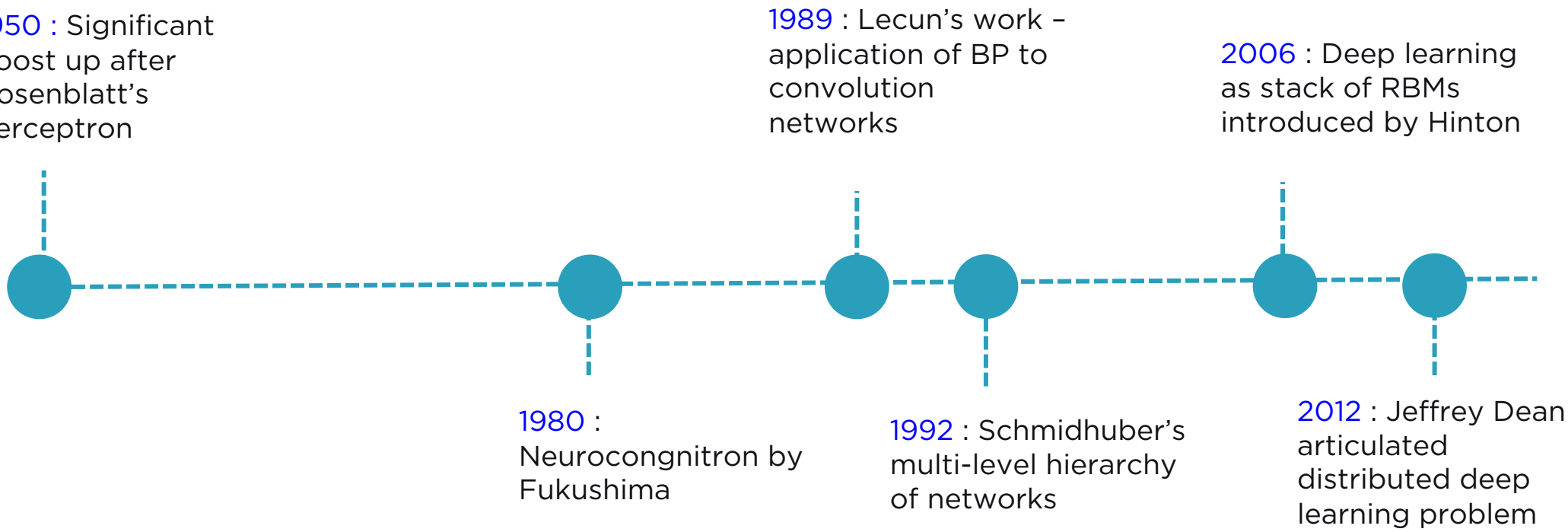
What Deep Learning is Not ?

- Deep learning **is not AI**
  - AI has other facets than just machine learning and deep learning

- Deep learning **is not artificial brain**
  - Though neural network techniques are inspired by neuroscience

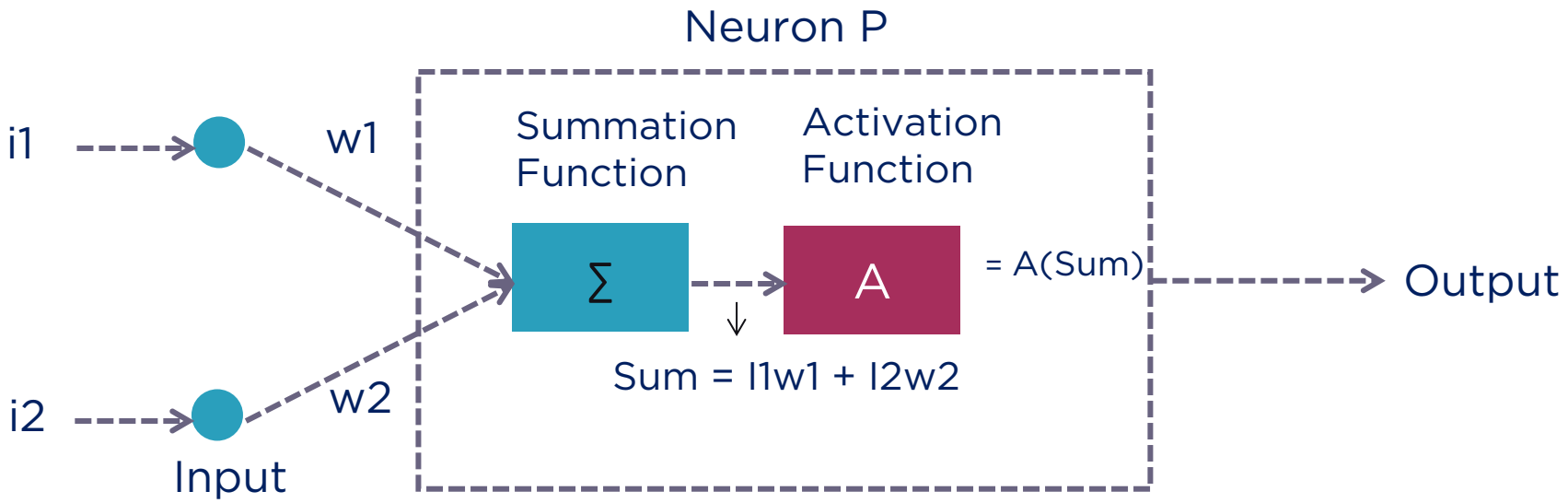
- Deep learning is **not the best tool for all ML task**
  - Requires lots of data
  - Computationally expensive
  - No theoretical guarantees
  - Black-box model and fall short in terms of **explainability**

Deep Learning : Timeline

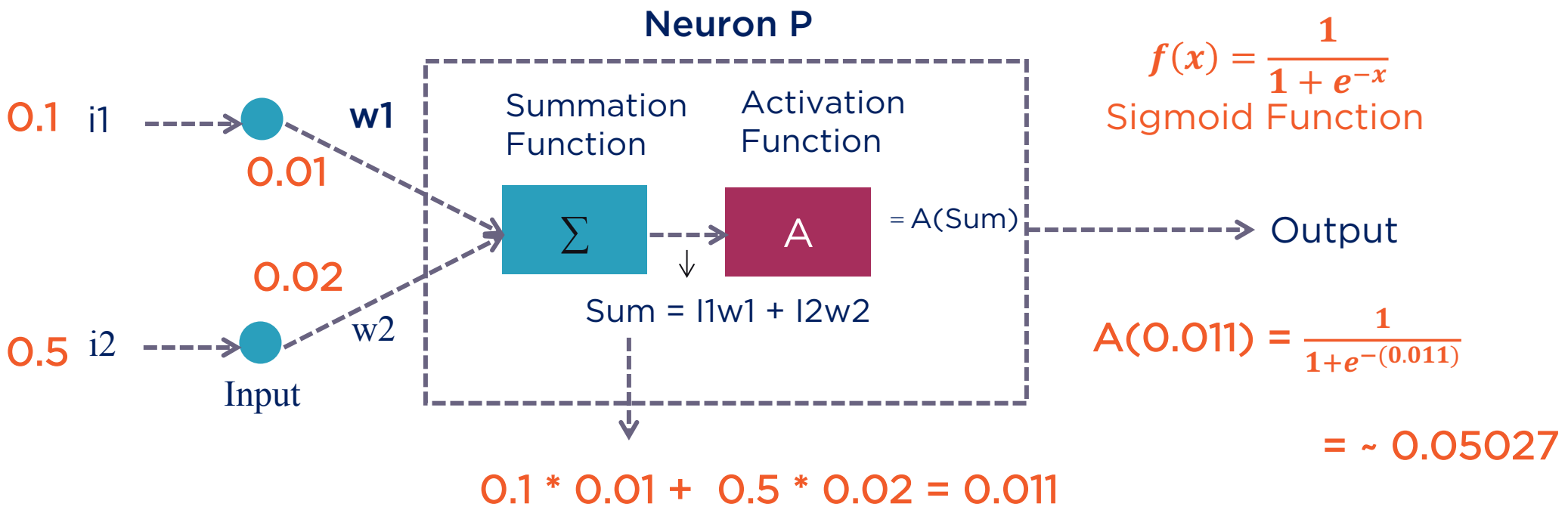


Artificial Neuron

- Combination of a linear model and an activation function

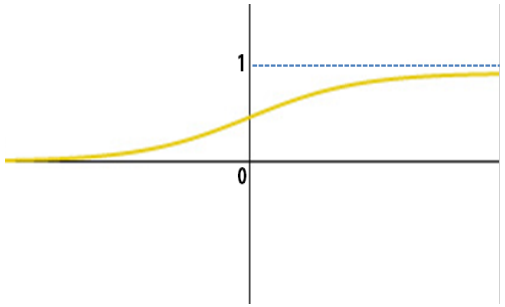


Neuron Computation



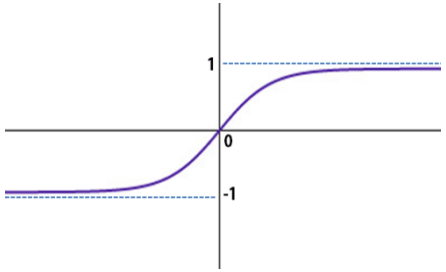
Activation Function

Sigmoid



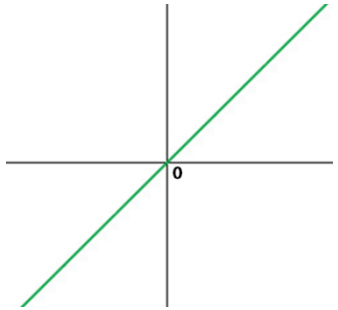
$$f(x) = \frac{1}{1 + e^{-x}}$$

Hyperbolic Tangent



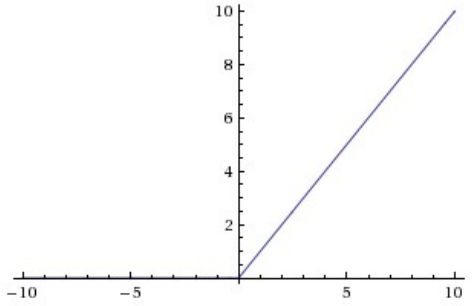
$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Linear



$$f(x) = x$$

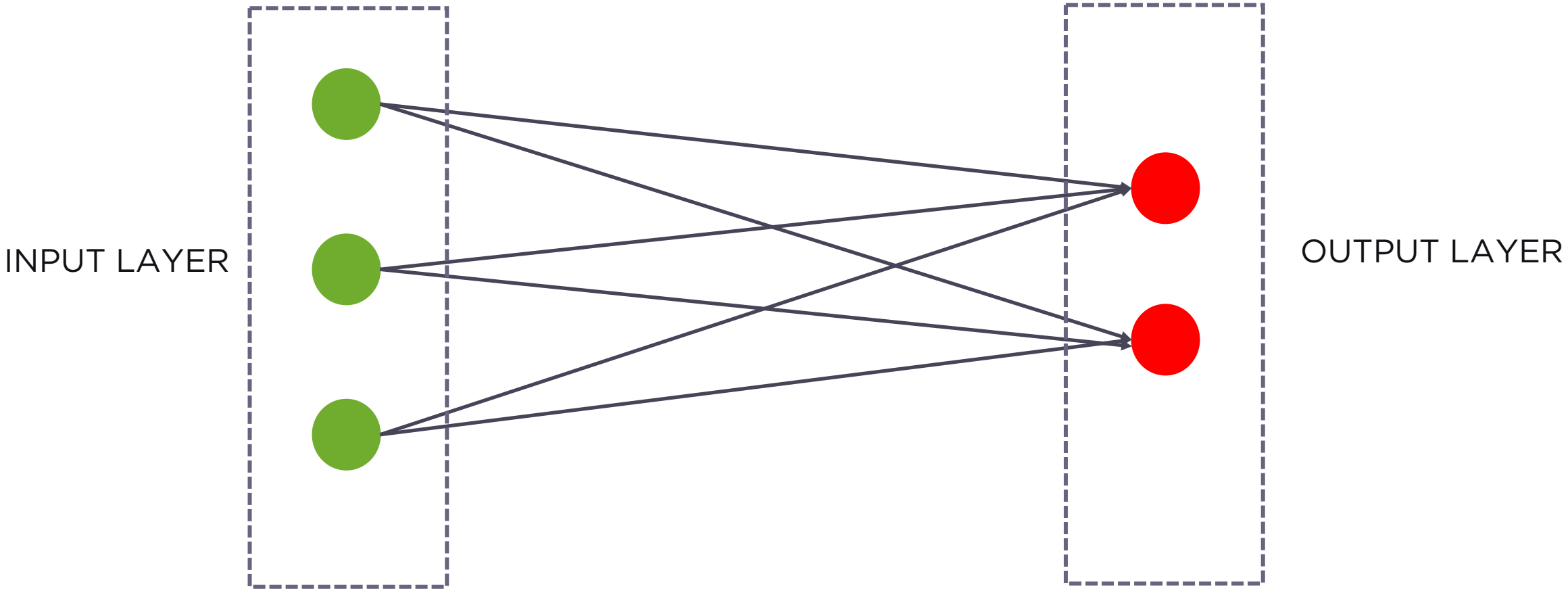
ReLU



f(x)=max(0,x)

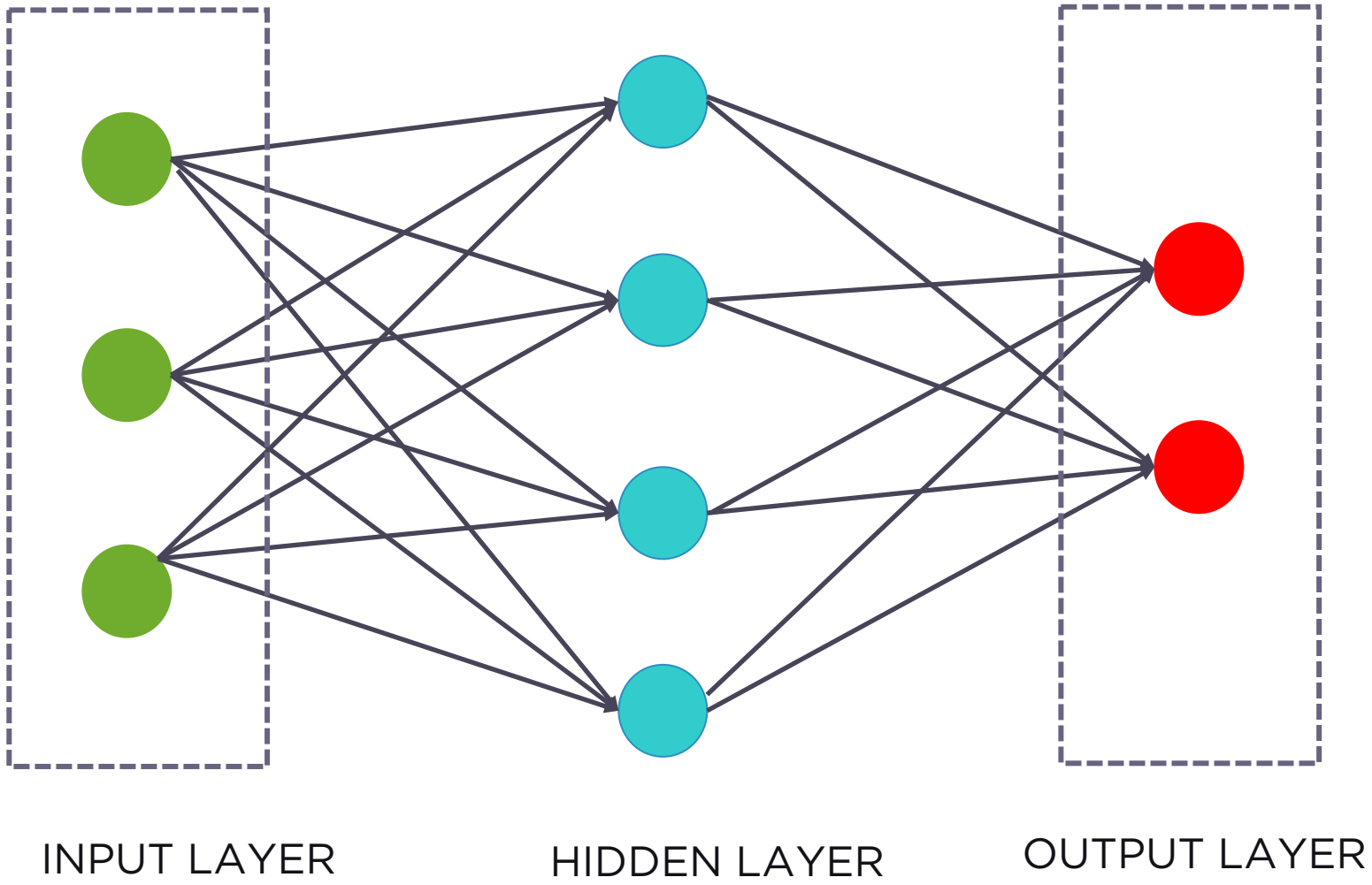
x = Sum or Net

Simple Neural Network



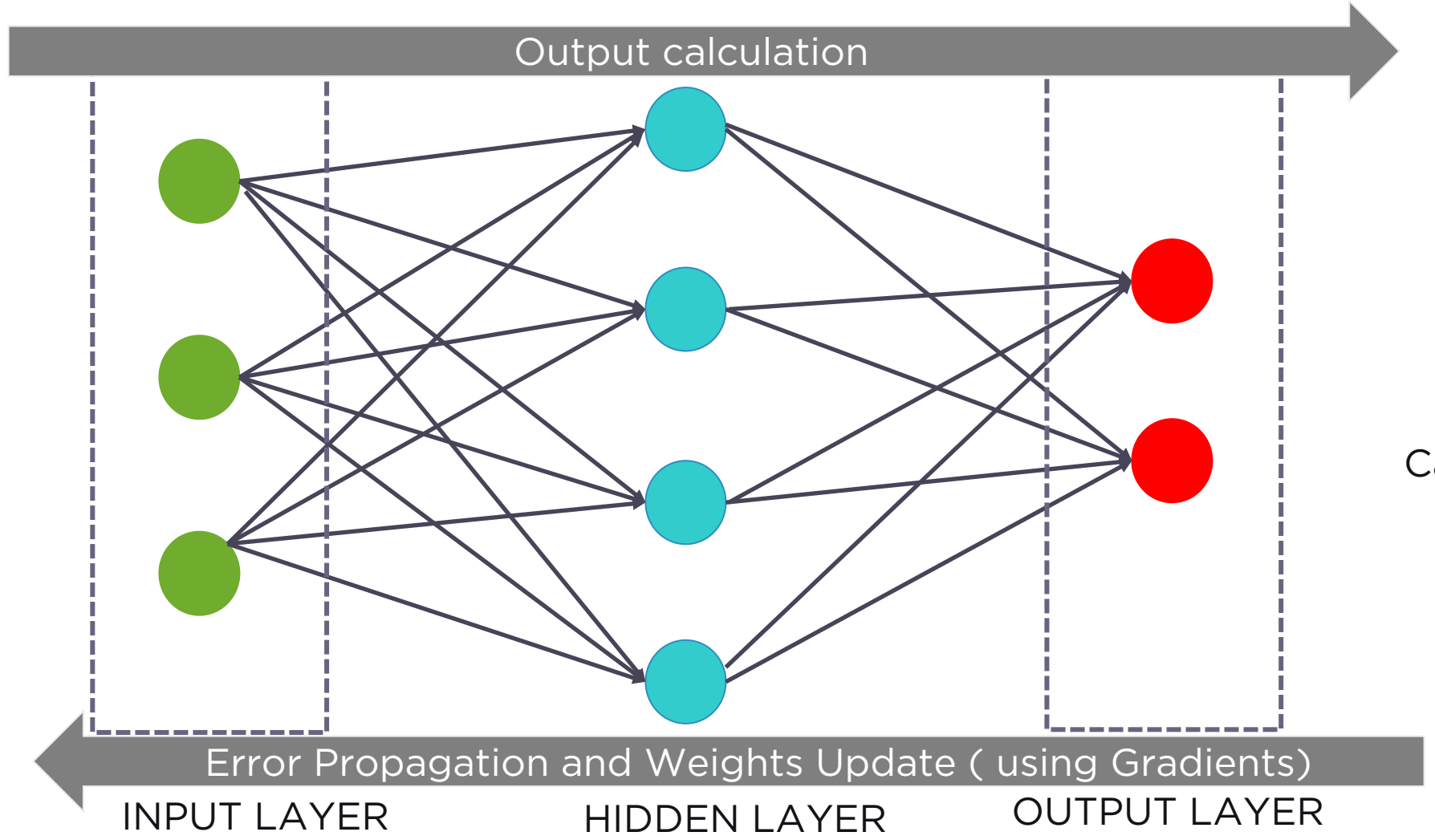
Limitation : Can learn only linear relationship

Simple Neural Network



Universal Approximator

Neural Network Training : Backpropagation



Why Deep Layers Are Required ?

- Feed forward network are universal approximators and theoretically can learn any function mapping from input to output if network is “Big Enough” in terms of layers and neurons

- For certain functions it has been shown that
  - if network is not deep, number of neurons required grow exponential to input

  - With deep network (at least k hidden layer) number of neurons required grow polynomial to input

But Even Deep Layers Had Challenges ?

- Vanishing Gradient and Saturation :
  - If input to certain layers are too small or too big, gradients become zero
  - Negligible update in some layers
  - Different initialization and activation attempted

- Overfitting
  - Highly complex model
  - Regularization, Early Stopping helps slightly

- Convergence
  - SGD often gets stuck to local minima
  - Highly sensitive to learning rate

All these issues led to “Neural Winter”

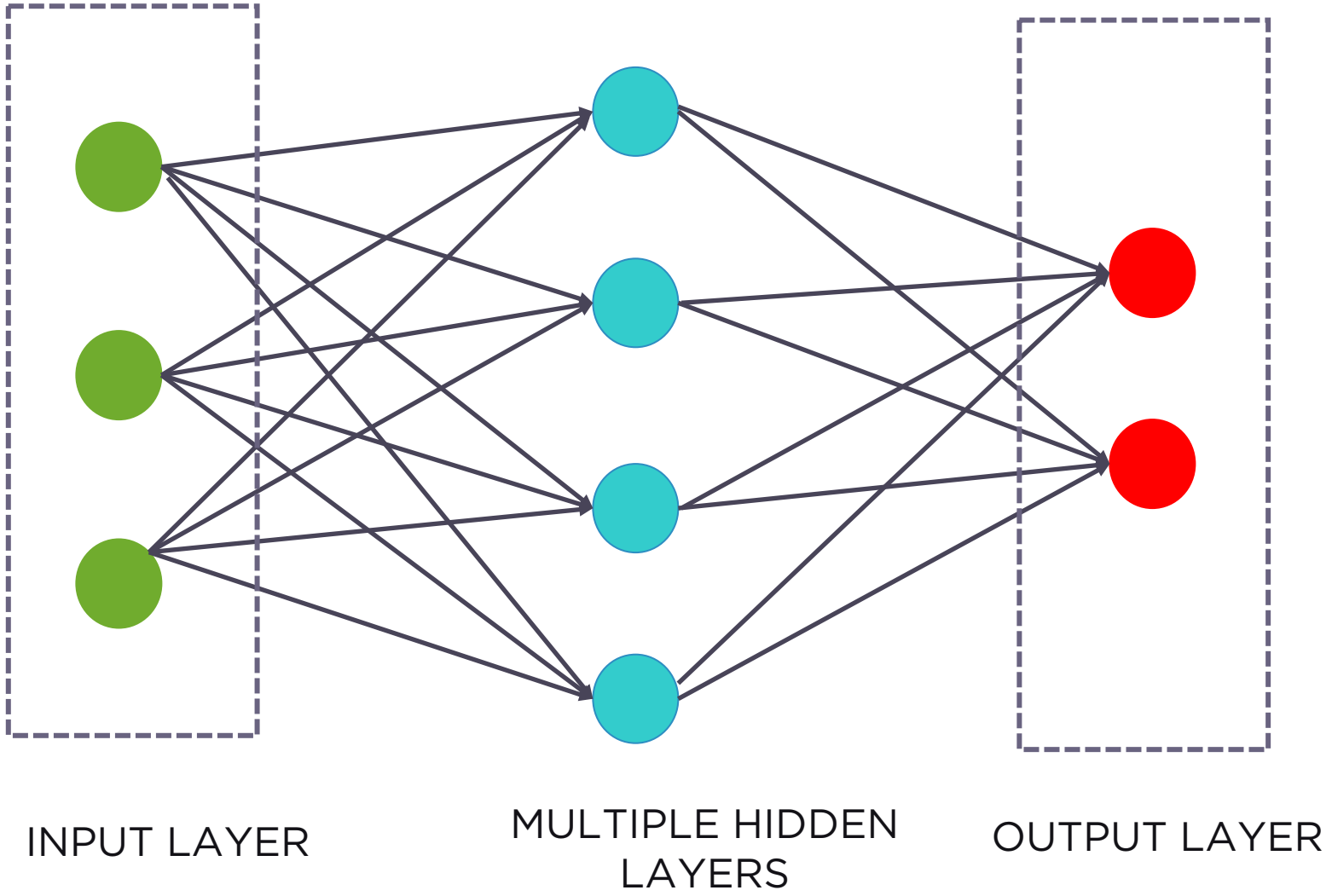
What Changed Now ?

- More data
  - More complex model like Deep Learning required more and more data to avoid problems of overfitting
  - With more labelled examples available these days, Deep learning models have shown promising results

- Computing Power
  - Computing power have increased significantly
  - Specialized hardware such as GPUs and TPUs helped to parallelize operations and process large datasets and complex models

- Research Breakthrough
  - Hinton's work on layerwise training led a new paradigm to train deep networks
  - Non-saturating activation functions (variation of ReLUs) reduced the problem of saturation
  - Dropouts helped to achieve regularization easily
  - Adaptive learning rate helped to avoid problems of local minima and led to better convergence

Popular Neural Network Architectures : Deep Feed forward



Popular Neural Network Architectures : Convolution Neural Network (CovNet)

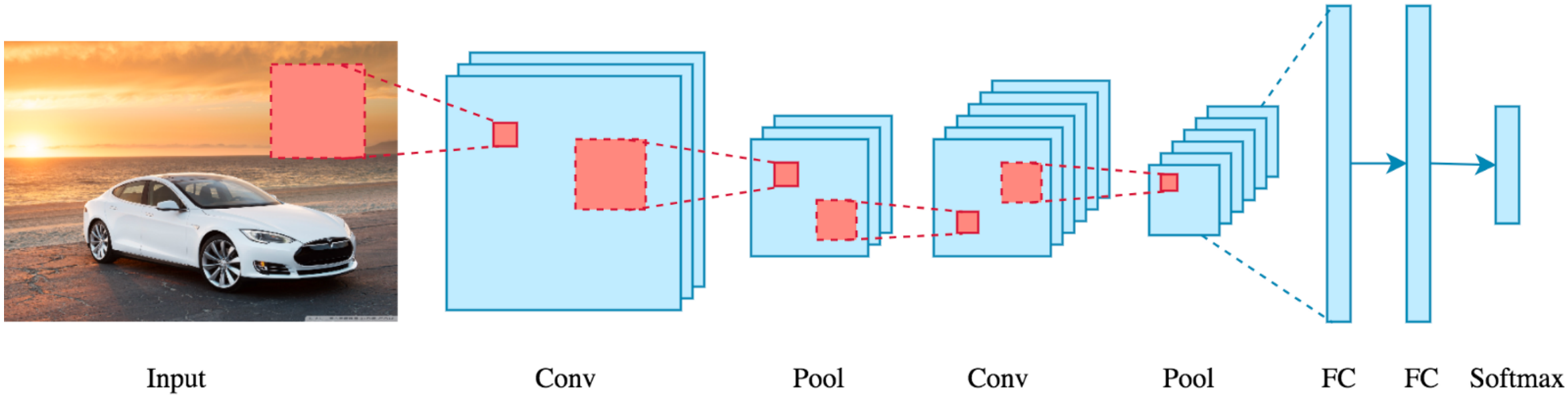


Image Credit : https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Convolution Neural Network (CovNet) : Components

CONVOLUTION

Mathematical Operation on two sets of information

NON-LINEARITY

POOLING

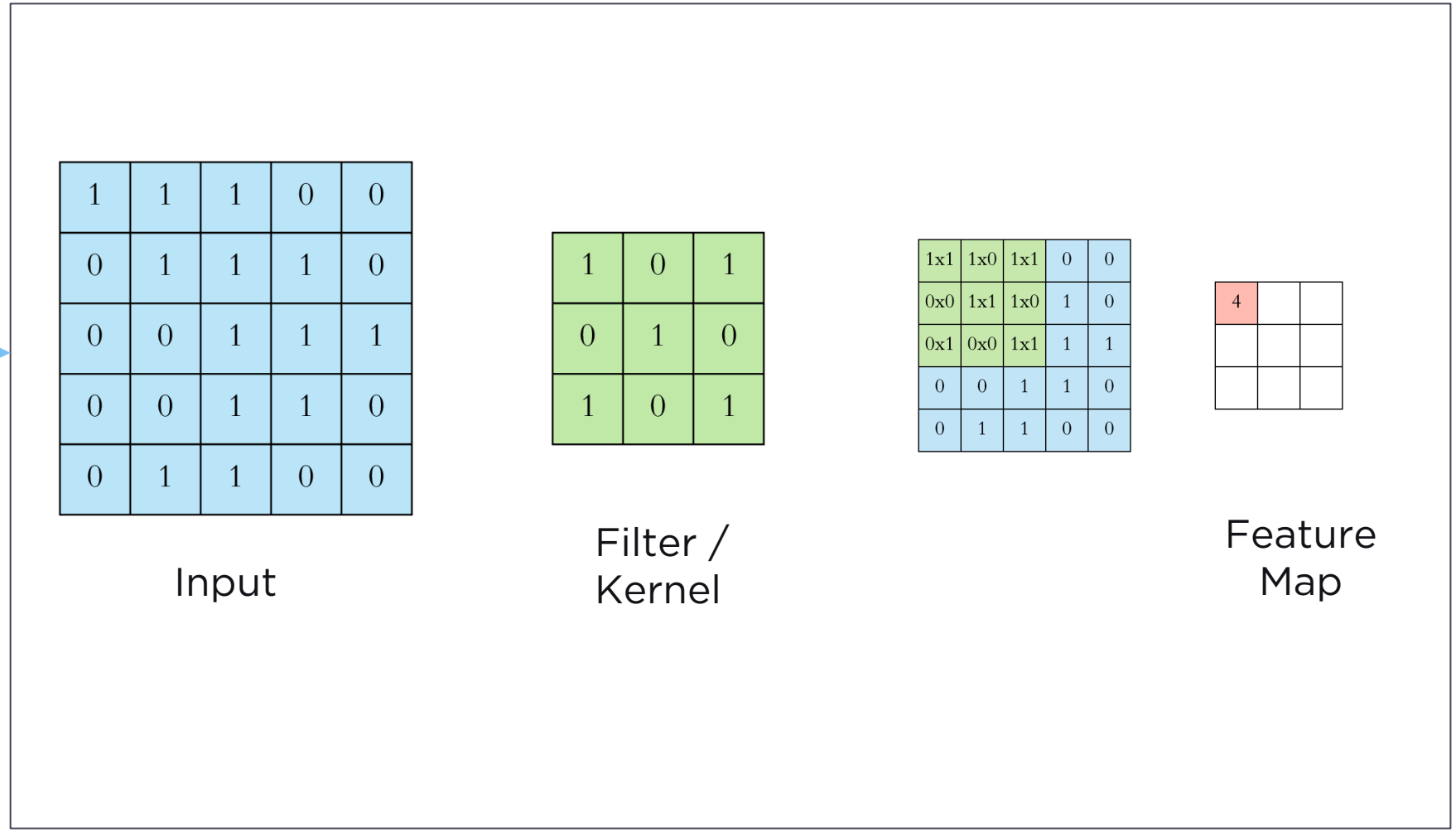


Image Credit : https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Convolution Neural Network (CovNet) : Components

CONVOLUTION

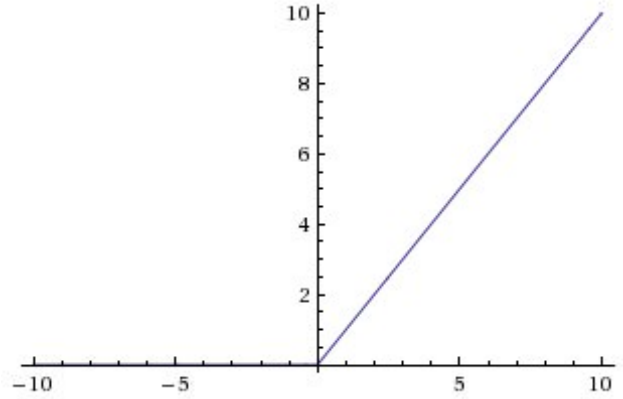
NON-LINEARITY

Non-linear operation on input

POOLING

ReLU (Rectified Linear Unit)

- Capture Interaction
  - E.g Input : 3 * x1 + 4 * x2, Output : f(Input)
- Introduce Non-Linearity
  - Slope is not constant (zero for negative value, 1 for positive)
- Reduce the chances of vanishing gradient
  - Average derivative rarely become 0 (some data points have positive derivative)



$f(x)=\max(0,x)$.

Image Credit : https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Convolution Neural Network (CovNet) : Components

CONVOLUTION

NON-LINEARITY

POOLING

Downsample feature map to reduce dimensionality

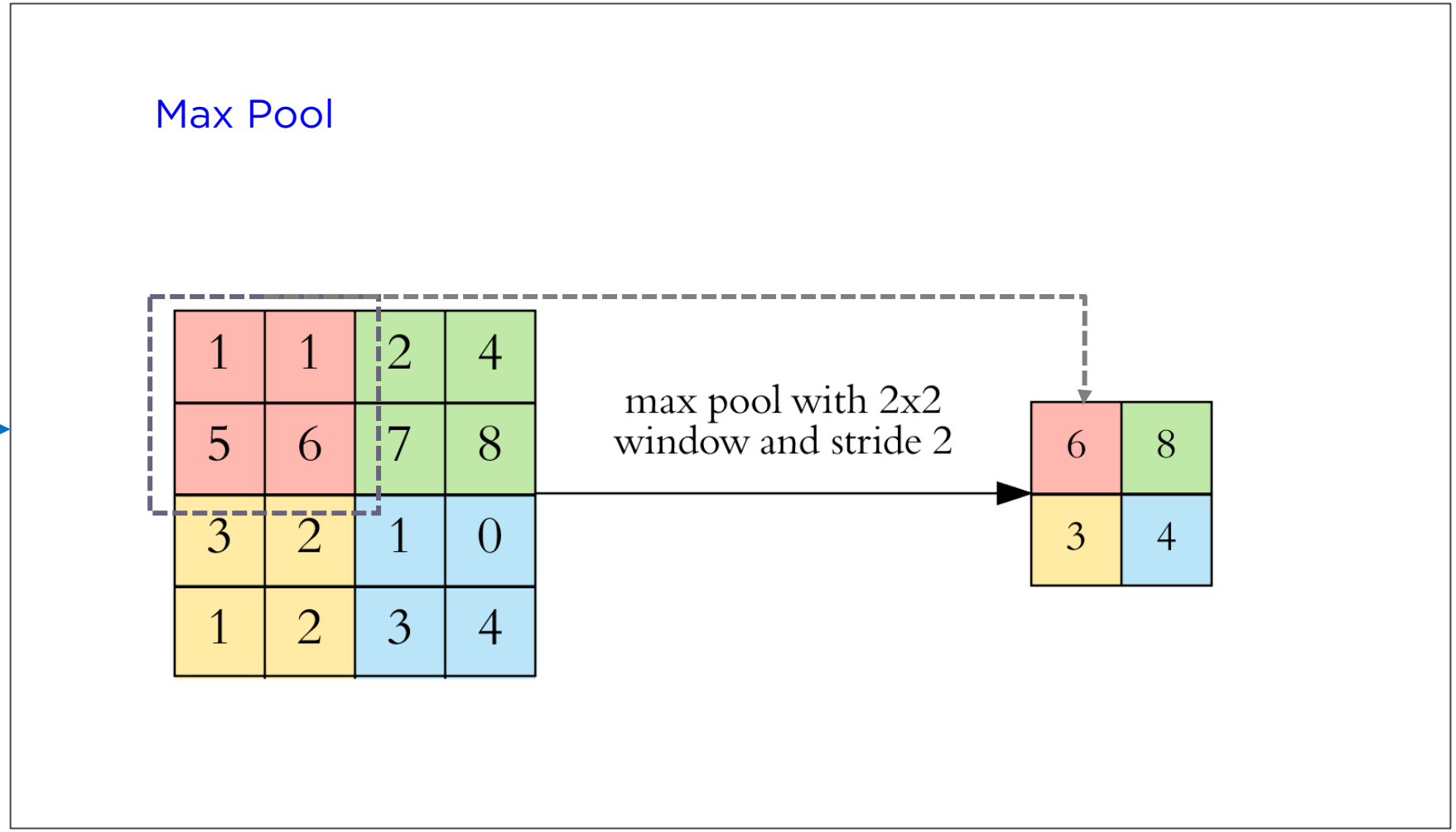
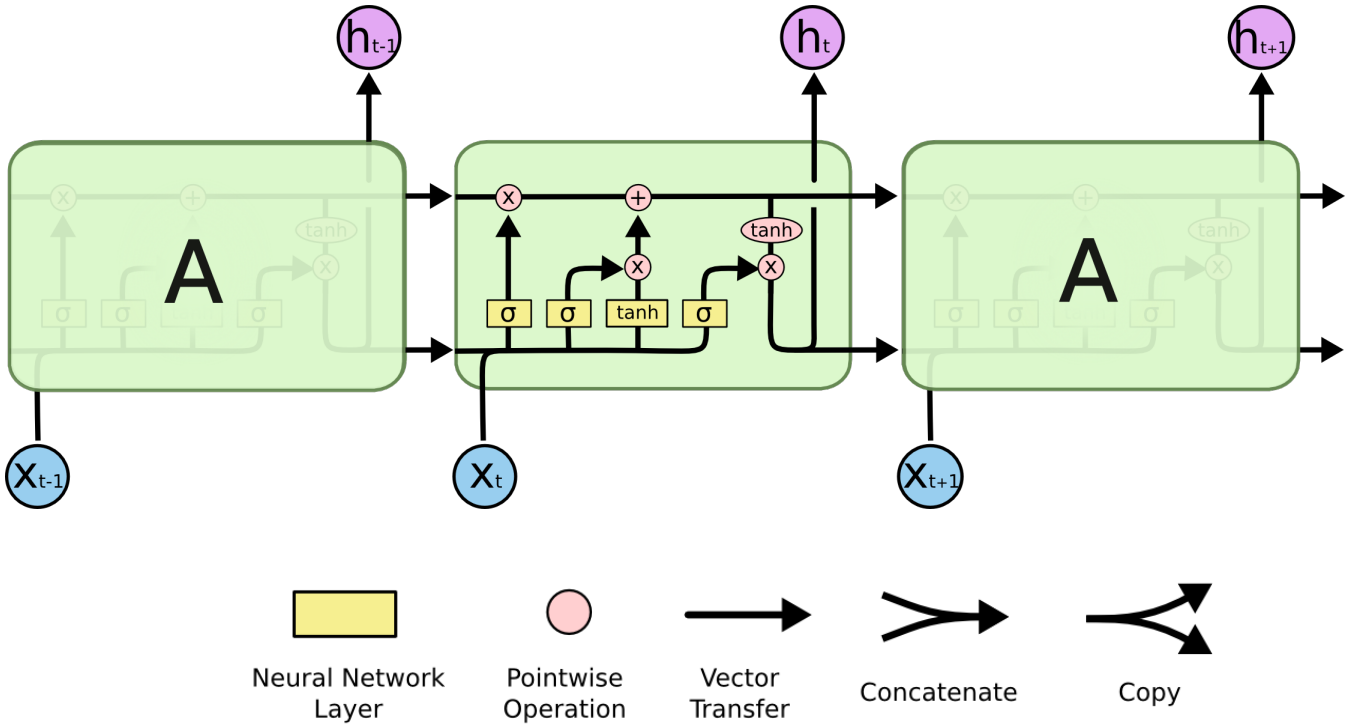


Image Credit : https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Popular Neural Network Architectures : LSTM (Long Short Term Memory)

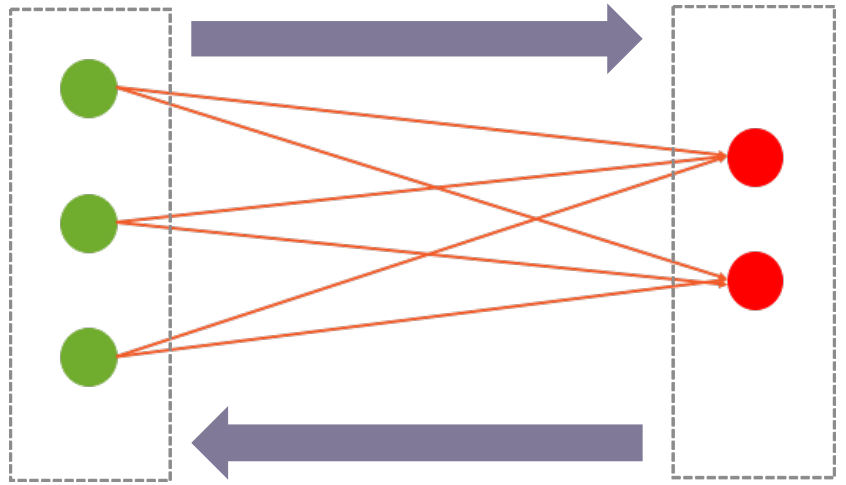
- Special kind of Recurrent Neural Network (RNN)
- Can learn long-term dependencies (as default behavior)
- Use gates
  - Forget gate
  - Input gate
  - Output gate



Source : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Popular Neural Network Architectures : RBM (Restricted Boltzmann Machine)

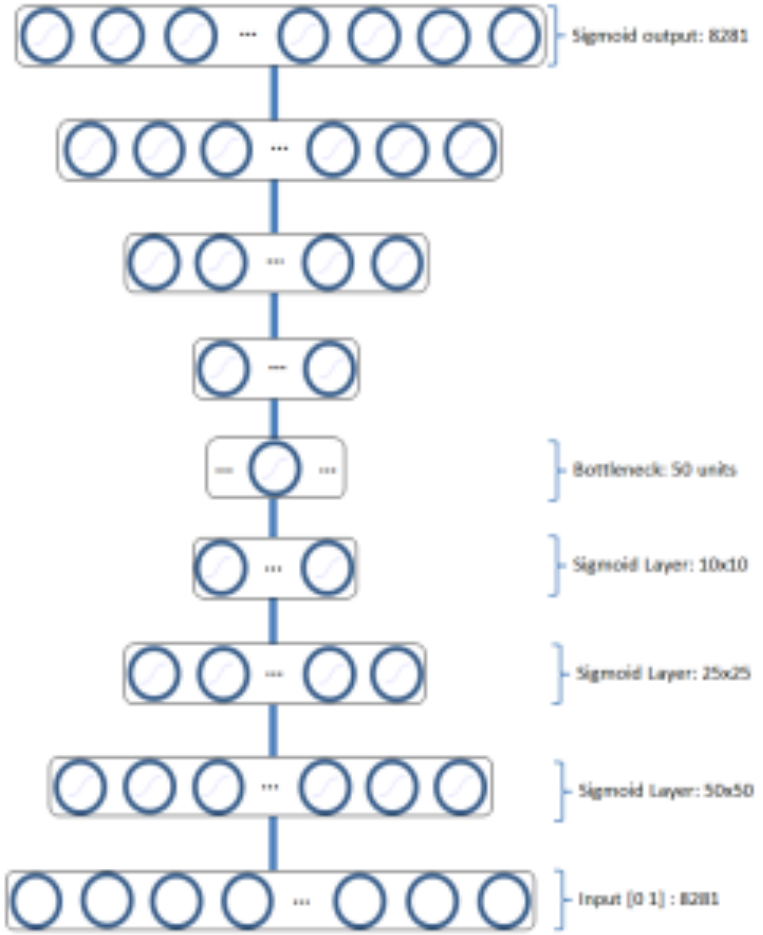
- Shallow network
- Can be used for unsupervised learning
- Reconstruct input
- Belongs to auto-encoder family
- Useful in Collaborative filtering, dimensionality reduction



Source : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Popular Neural Network Architectures : Auto-Encoders

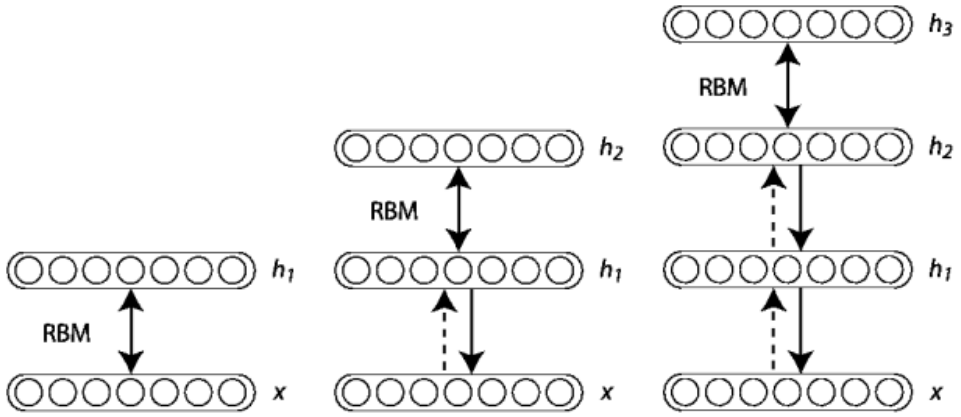
- Aim of auto encoders network is to learn a compressed representation for set of data
- Unsupervised learning algorithm that applies back propagation, setting the target values equal to inputs (identity function)
- Denoising auto encoder addresses identity function by randomly corrupting input that the auto encoder must then reconstruct or denoise
- Best applied when there is structure in the data
- Applications : Dimensionality reduction, feature selection



Source : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Popular Neural Network Architectures : Deep Belief Networks

- Boltzmann Machine is a specific energy model with linear energy function.
- This is a deep neural network composed of multiple layers of latent variables (hidden units or feature detectors)
- Can be viewed as a stack of RBMs
- Hinton along with his student proposed that these networks can be trained greedily one layer at a time



Source : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Why Tensorflow for Deep Learning ?

Why Tensorflow for Deep Learning ?

Extensive built-in support

Assemble neural networks

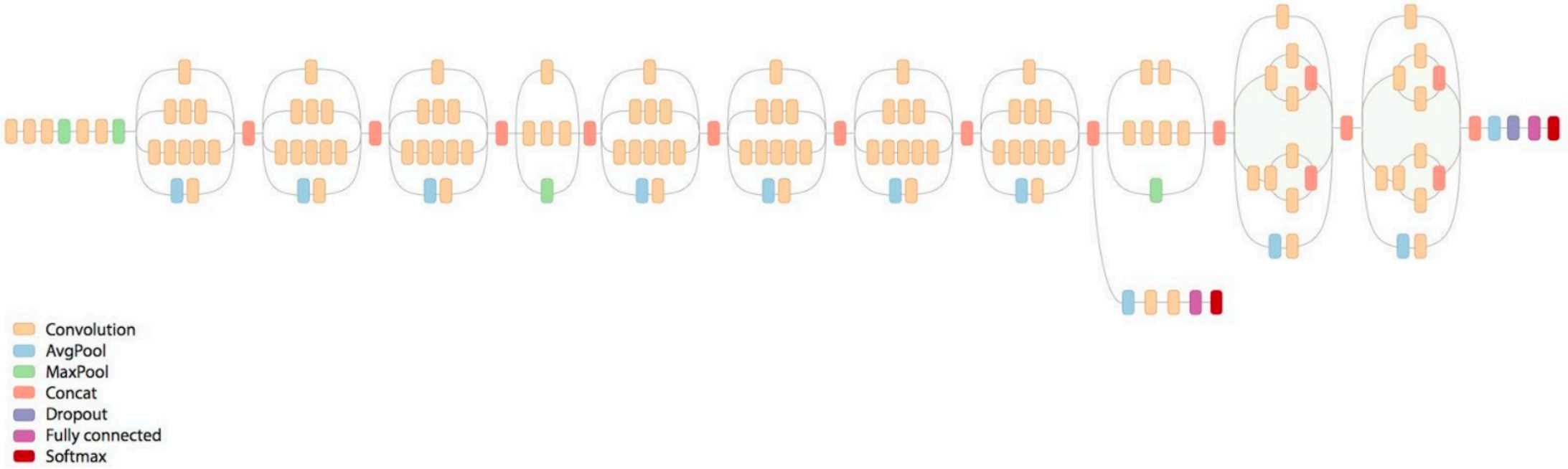
Mathematical functions

Auto-differentiation & first-rate optimizers

Versatility

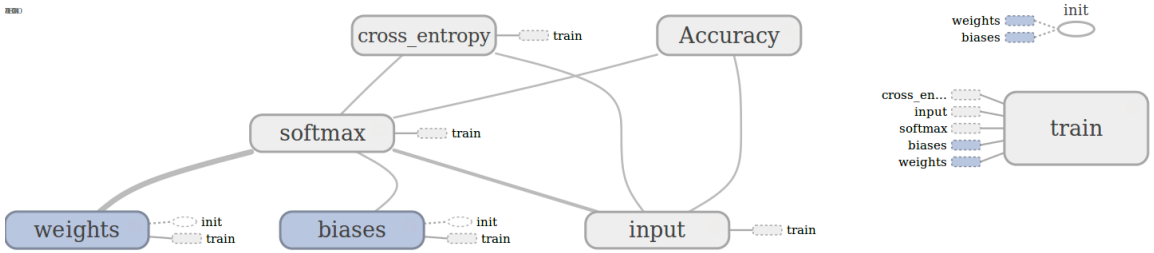
Why Tensorflow for Deep Learning ?

Align cognitive model to programming model

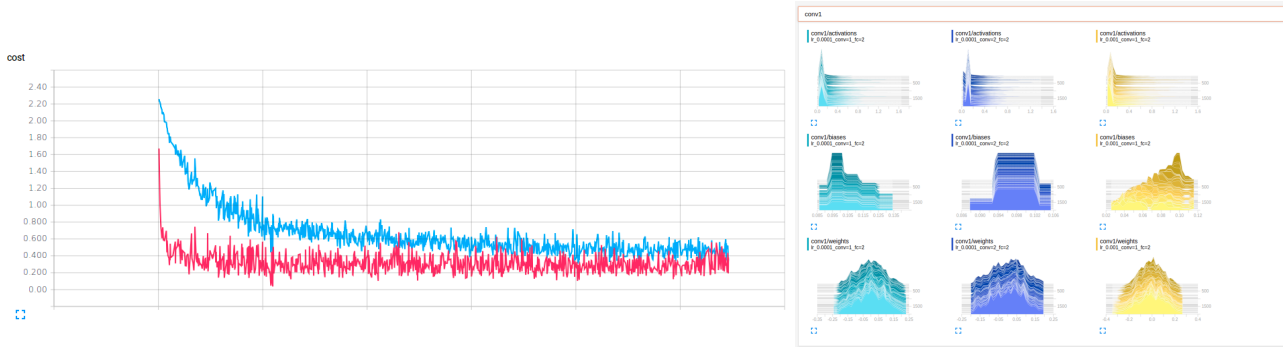


Why Tensorflow for Deep Learning ?

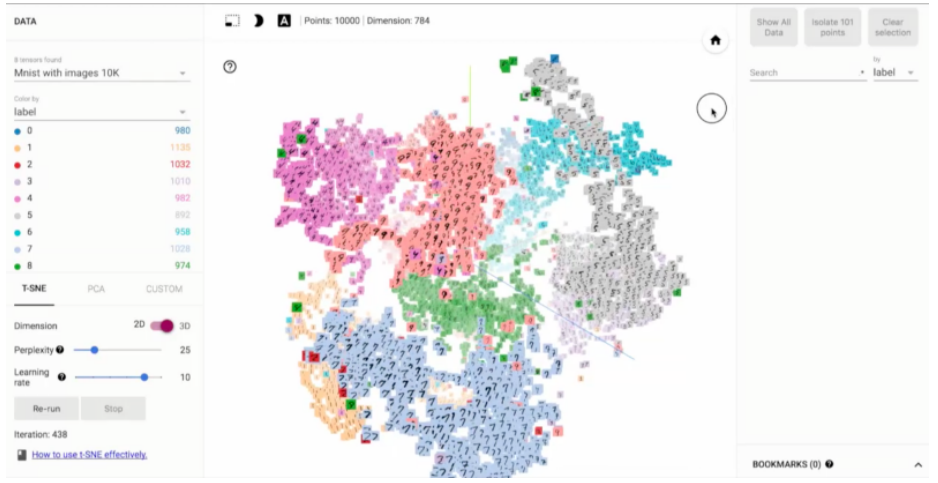
Use Tensorboard to Visualize and Debug Deep Learning Network



Network Graph



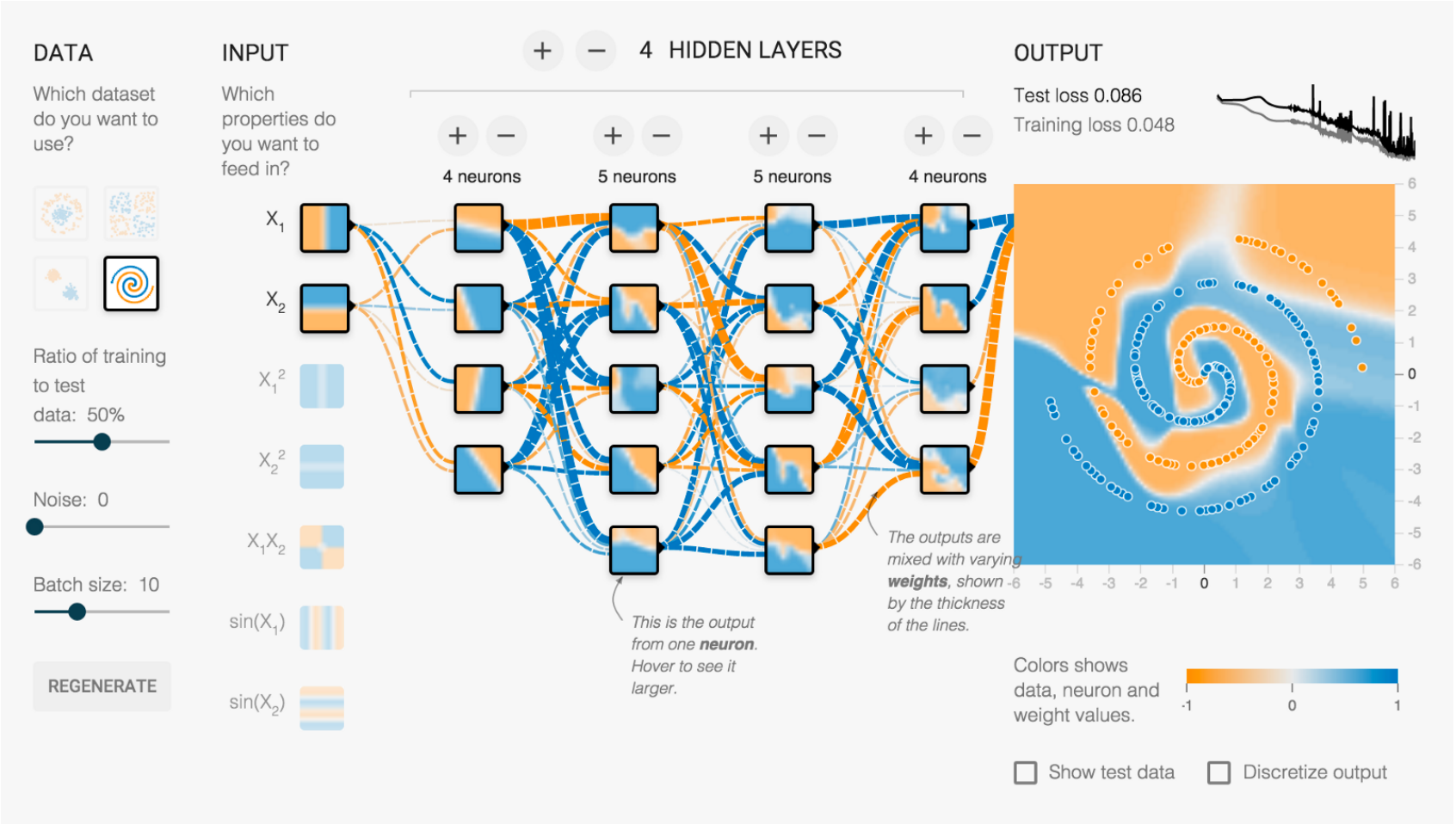
Cost Trends



Visualize Embedding

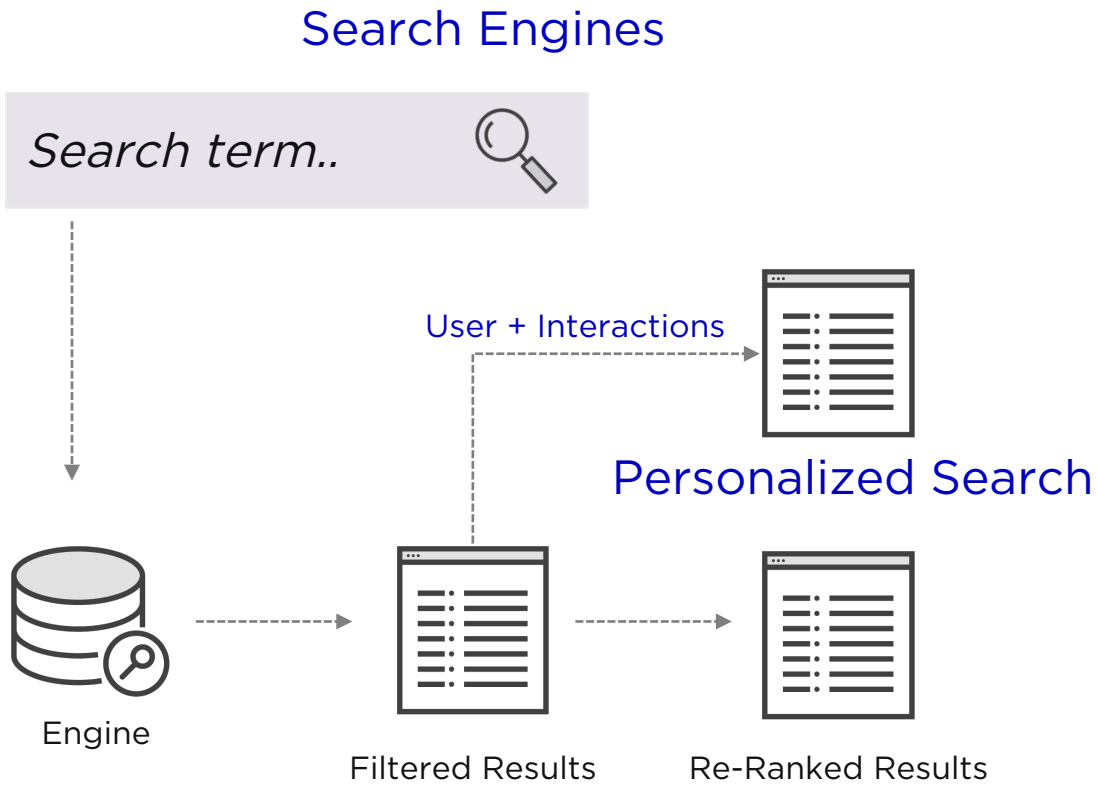
Why Tensorflow for Deep Learning ?

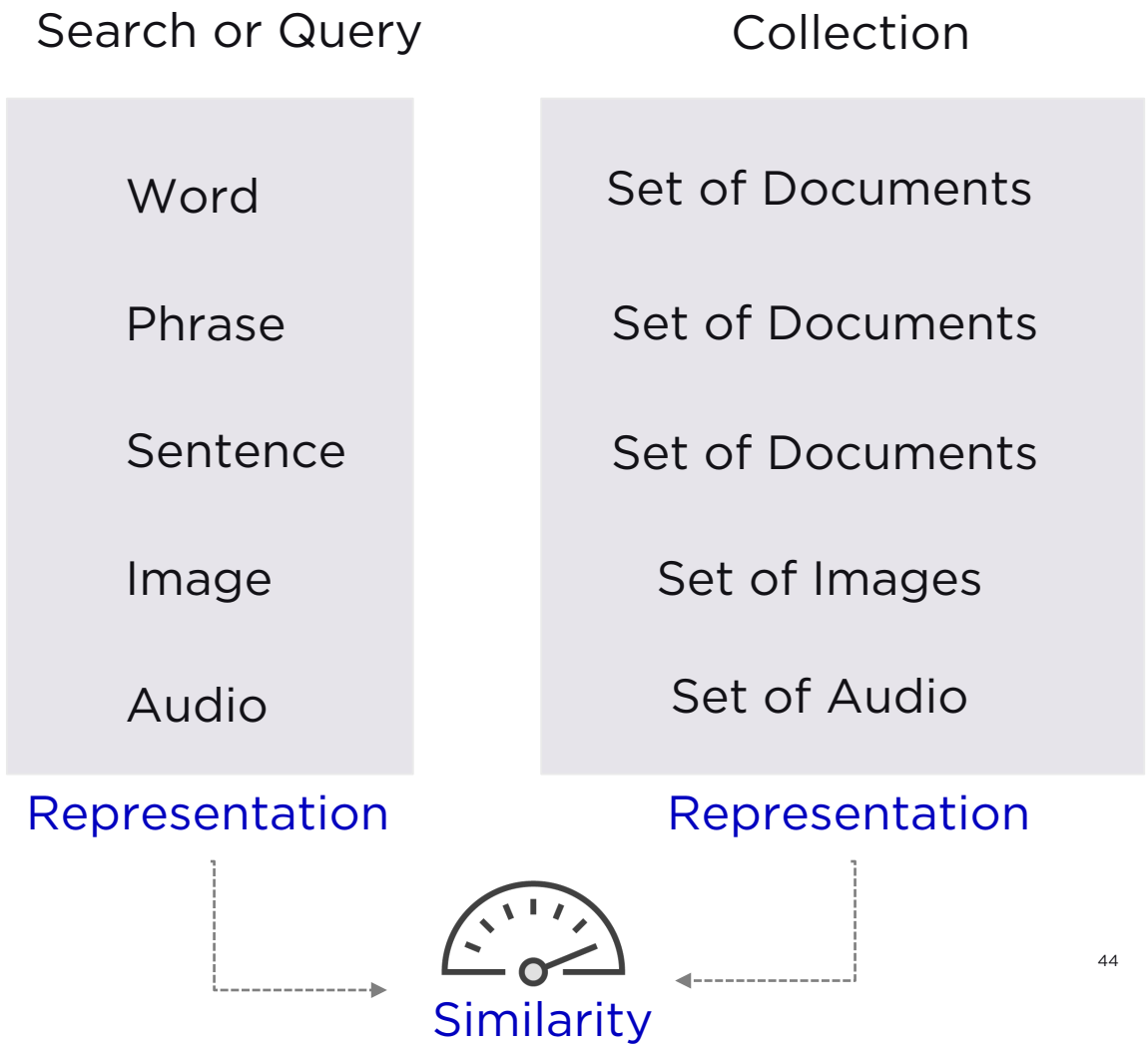
Tensorflow Playbook : http://playground.tensorflow.org/



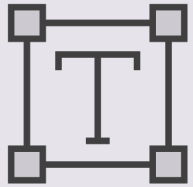
Deep Learning in Search

Representation : A Key Aspect





Representation : A Key Challenge



Text

Query	Collection
Word Sentence	Documents

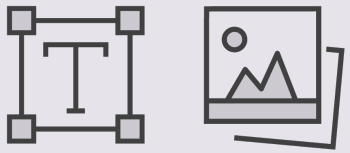
Image

Query	Collection
Image	Set of Images

Treat “Representation” as the problem of “**Embedding**” to encode objects (text, images)
into continuous space (set of numeric values)

Search Problem In Context of Embedding

1



Objects

Create embedding for objects into continuous space

2

Sneakers

Shoes

How to tie laces ?

Black holes

How time changes in black hole ?

Put similar objects together Based on embedding

3

Sneakers

Shoes

How to tie laces ?

Black holes

How time changes in black hole ?

Given a query embedding, find neighbors quickly

Word Embedding : One-Hot Encoding

	shoe	sneakers	tree	book	black	...	..	..
Sneakers	0	1	0	0	0	0	0	0

Number of columns = Number of unique words in the vocabulary

Issues :

- Sparse (all values are zero except one)
- Large embedding dimension (equal to vocab size)
- Semantic meaning not captured
  - “shoe” is at same distance as “tree”

Word Embedding : Prediction Based Encoding

Sneakers
+
Near by words

[0.322 0.122 0.231 0.111 0.222 ..... 0.445]

Embedding Size : d

Word2Vec Model
(Google, 2013)

CBOW

Use surrounding words to predict target word

Skip-Gram

Use target word to predict surrounding words

Benefits:

- Dense representation
- Smaller embedding dimension (equal to embedding size :d)
- Semantic meaning captured
  - “shoe” is at smaller distance than “tree”

GloVe Model
(Stanford, 2014)

Use word-word co-occurance matrix and nearest neighbor to create embedding

Demo : Short Introduction to Embedding

Goal :

- Embedding in Tensorflow
- Word Embedding using GloVe pre-trained model

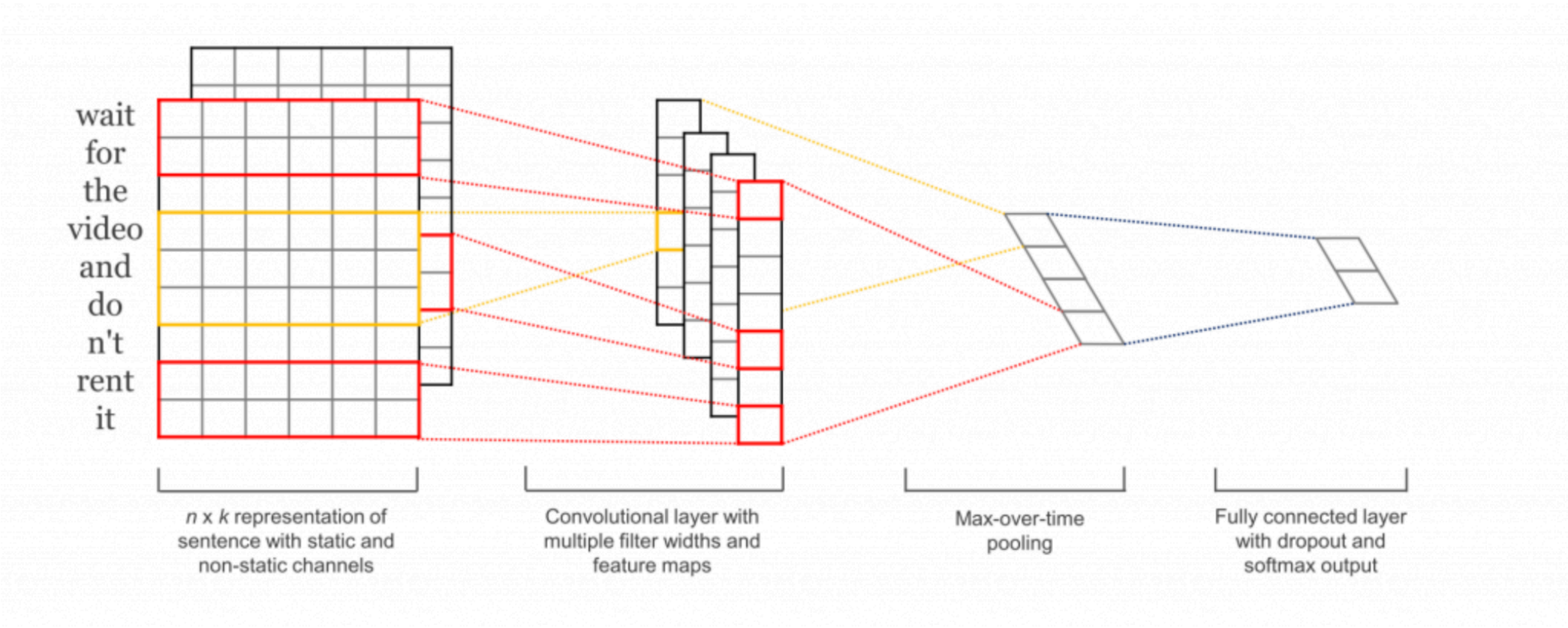
Sentence Embedding

I like white sneakers.

	I	Like	White	Sneakers
Embedding size (d)	..	...	..	..
	x1	x2	x3	x4
	..	..	..	..

- Sentence embedding can be considered as word embedding matrix where each word is represented as a column of size d
- Leverage pre-trained word embedding to learn sentence embedding
- Weights are further tuned during training process

Sentence Embedding Using Convolution Neural Network



Paper : Convolutional Neural Networks for Sentence Classification" by Yoon Kim

Image Embedding : Option 1 : Flattened Arrays

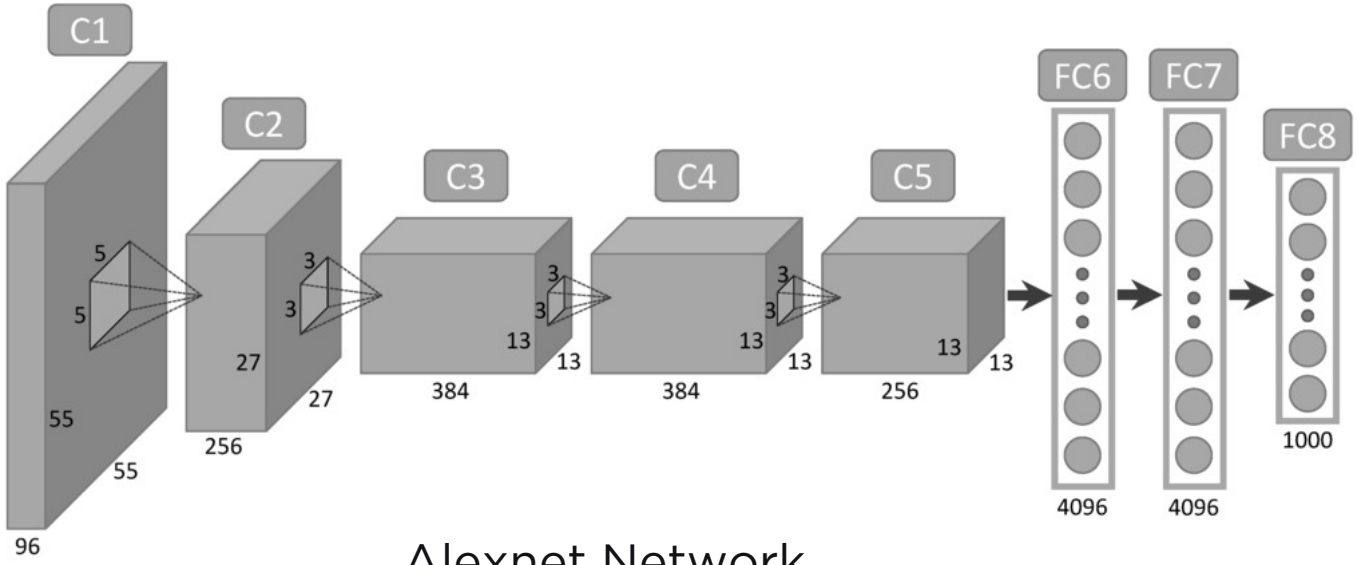
Each image is set of pixels
Flatten the pixels matrix into arrays

100 px by 100 px image : 10000 dimensional array

Issues :

- Very large embedding dimension
- Search in a very large embedding space will be very expensive
- Spatial features (edges, contours, textures) are not captured : Poor search results

Image Embedding : Option 2 : Pre-Trained Deep Learning Models



Alexnet Network

Inception
VGG
Lenet
GoogLenet
ResNet

Benefits:

- Smaller embedding dimension
- Spatial features (edges, contours, textures) are captured in intermediate layers
- Enhanced search experience

Image Source : https://www.saagie.com/blog/object-detection-part1

Demo : Image search using Alexnet Pre-Trained Model

Goal :

- Use Alexnet Pre-trained model to create image embedding
- Image Search

Deep Learning in Recommendation System

RecSys 101 : What is RecSys?

"Serve the relevant items to users in an automated fashion to optimize short and long term business objectives"

RELEVANT (WHAT)

1. Novelty
2. Serendipity
3. Diversity

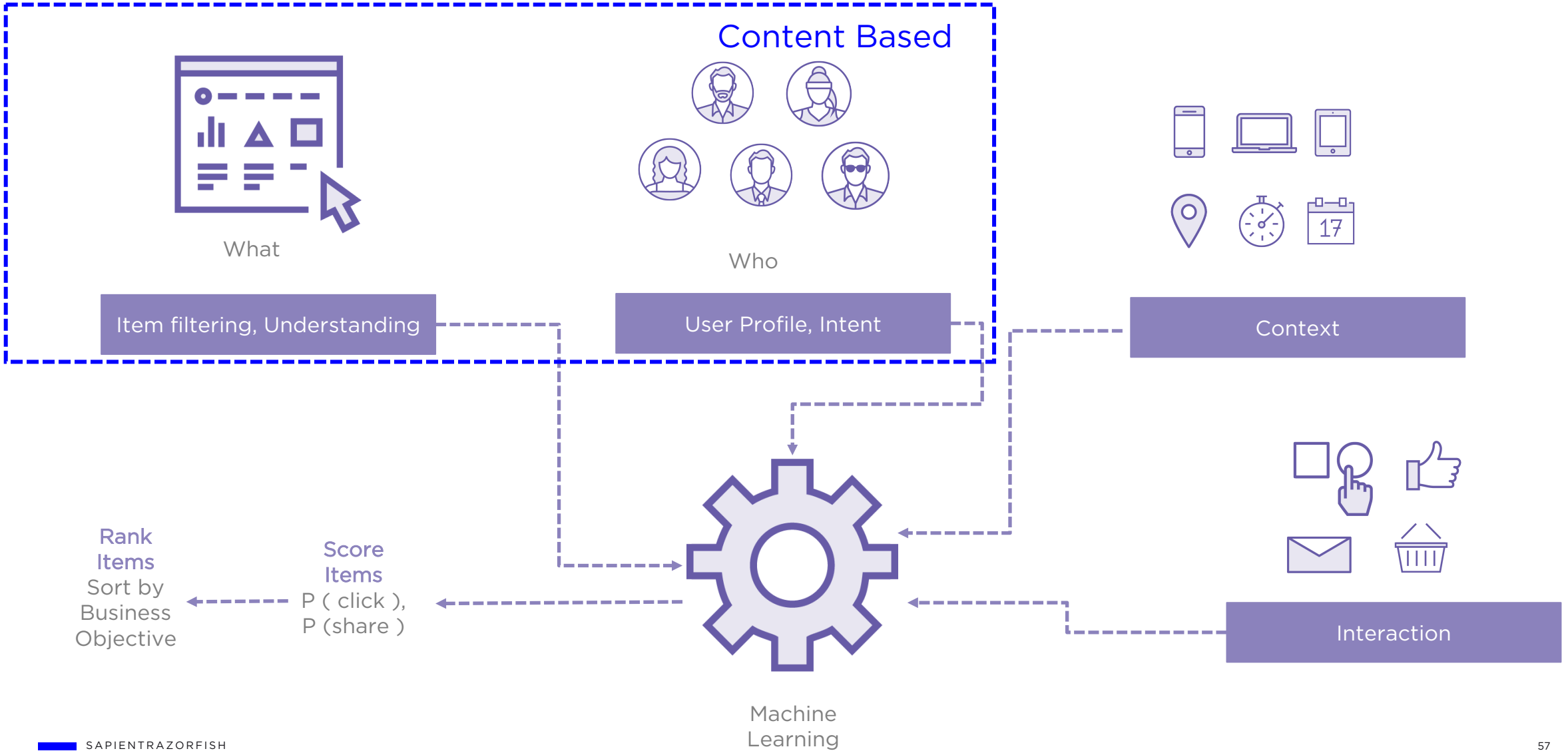
AUTOMATED (HOW)

1. No manual intervention
2. Scale Up

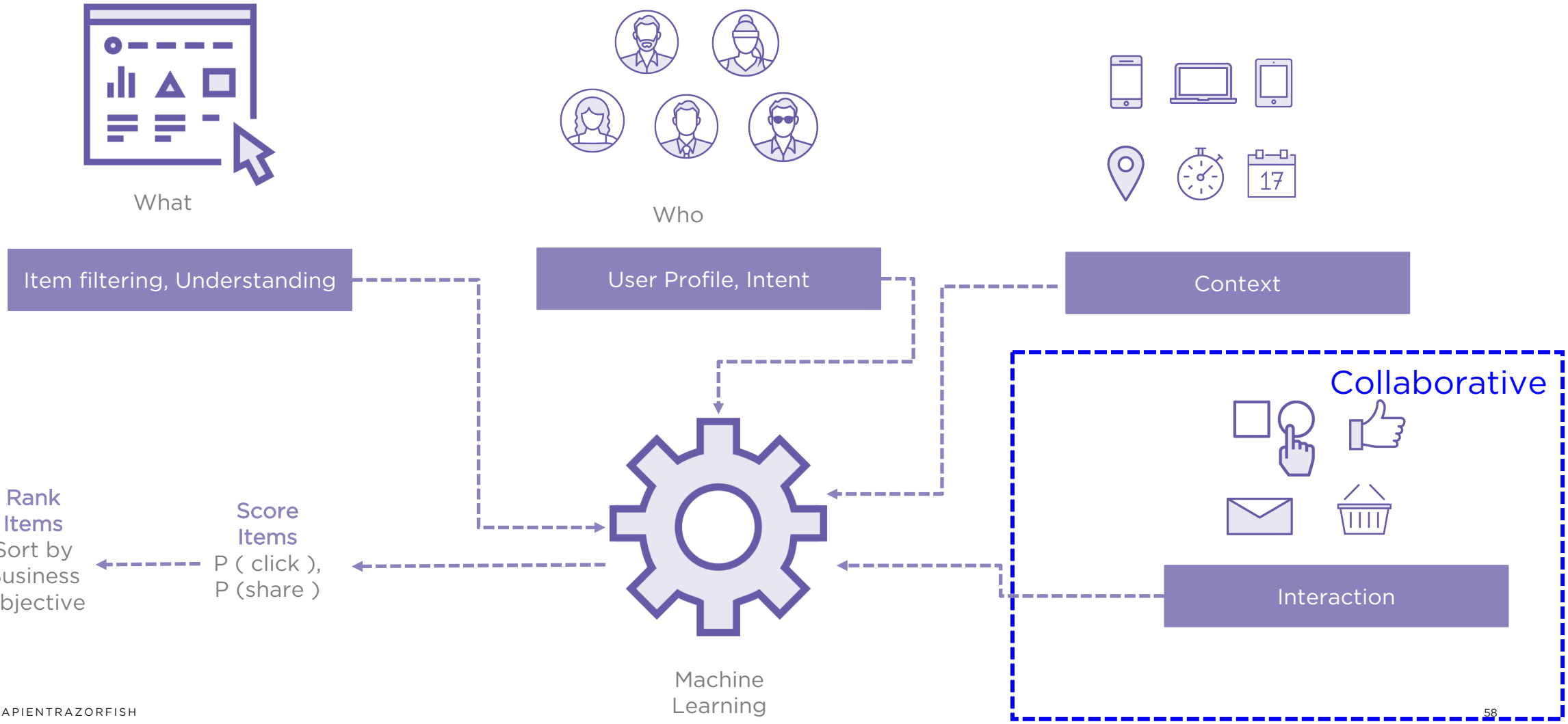
BUSINESS OBJECTIVES (WHY)

1. Short Term Business Objectives
   a. High clicks
   b. Revenue
   c. Positive explicit ratings
2. Long Term Business Objectives
   a. Increased engagement
   b. Increase in social action
   c. Increase in Subscriptions

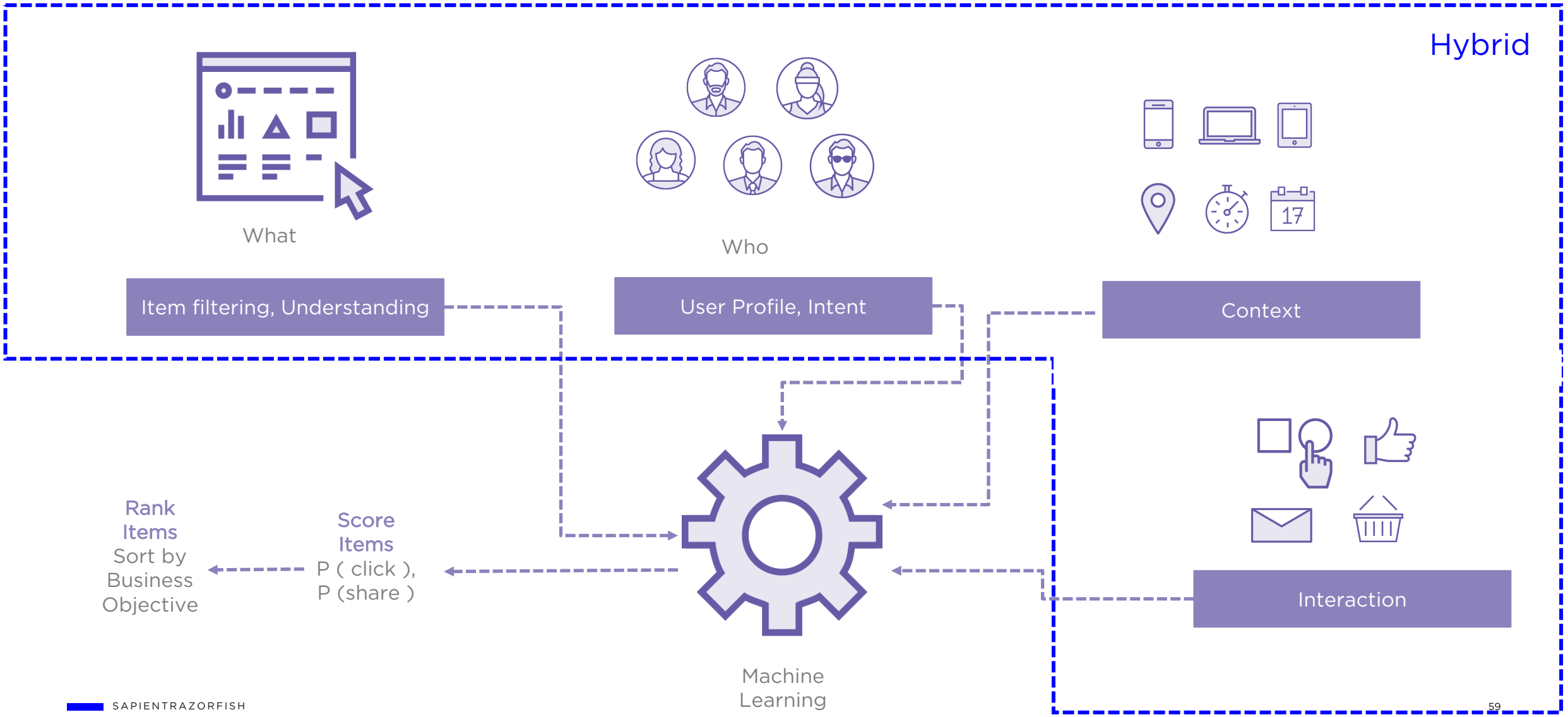
RecSys 101 : Internals



RecSys 101 : Internals



RecSys 101 : Internals



RecSys Traditional Approaches

Challenges with Traditional Approach

ADD SLIDE FOR LIMITATIONS OF TRADITIONAL APPROACH: SUMMARY

- Sparse Features
- Cold Start Problem

Representation : A Key Aspect

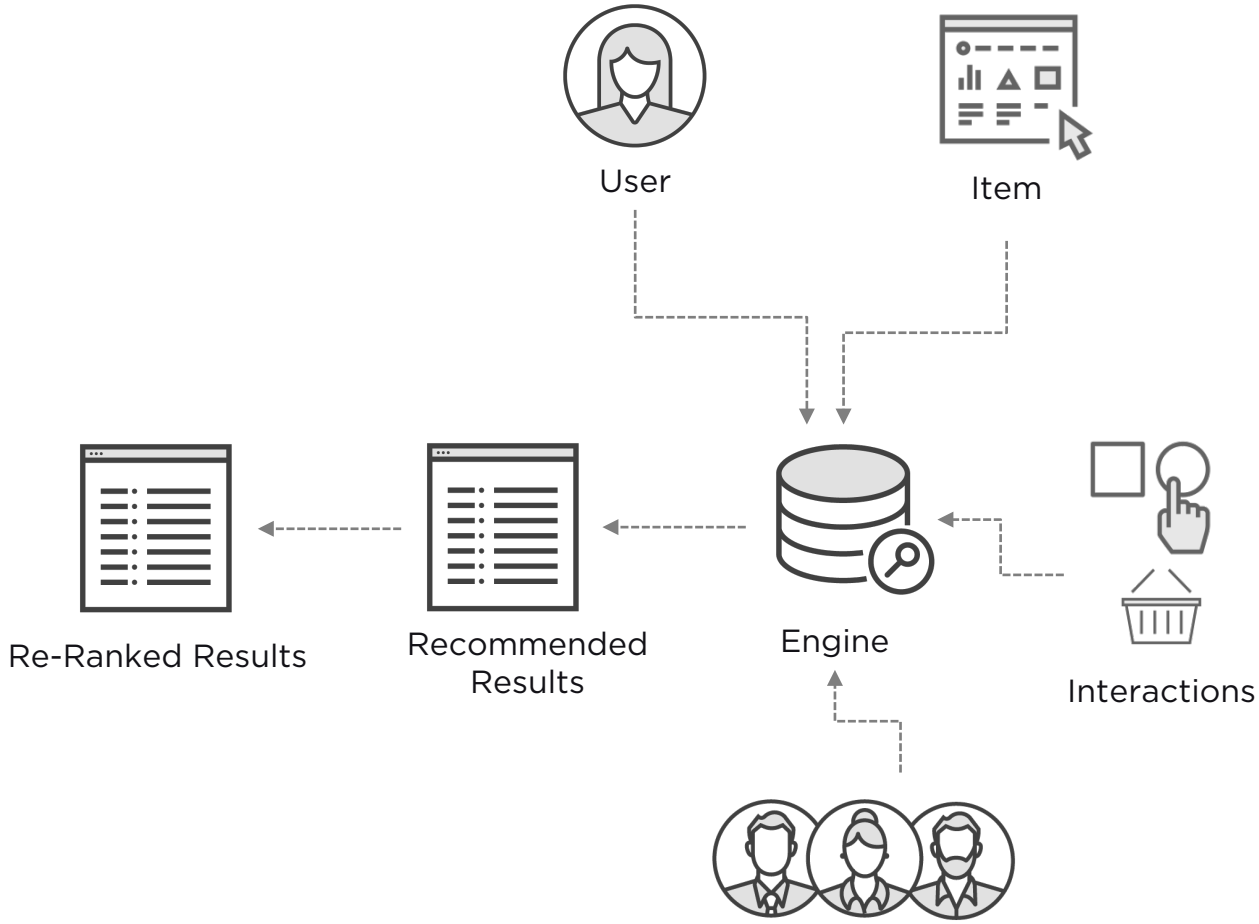
User

Item

Interactions

Representation

Recommendation Engines



Matrix Factorization

Items

Users

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	X		X		X	
User 2		X	X			
User 3				X		X
User 4					X	
User 5	X	X		X		X
User 6			X	X		
User 7	X	X	X		X	X
User 8		X		X		
User 9			X			

R

$\cong$

	UF1	UF2
User 1		
User 2		
User 3		
User 4		
User 5		
User 6		
User 7		
User 8		
User 9		

U

X

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
IF1						
IF2						

V

How to better represent users and items ?

What about item and user metadata ?

Demo : User and Item Embedding in Matrix Factorization using Tensorflow

Goal :

- Use embedding in RecSys
- How to add metadata for building hybrid RecSys

Learning to Rank

Problem Space : Learning to Rank

Classical Information Retrieval

- Depends on few features : TF , IDF, Length of Document, Probability of term given corpus
- Not many parameters to tune with limited features
- But Don’t scale for more features

Learning to Rank

- Can work on hundreds of features
- Not many parameters to tune with limited features
- But Don’t scale for more features
- How to use both explicit (rating) and implicit (view, share) feedback

Demo : Using Triplet Loss for Implicit Feedback using Tensorflow

Goal :

- Use implicit feedback
- Use triplet network

Phase 4 : Production

Tensorflow in Production

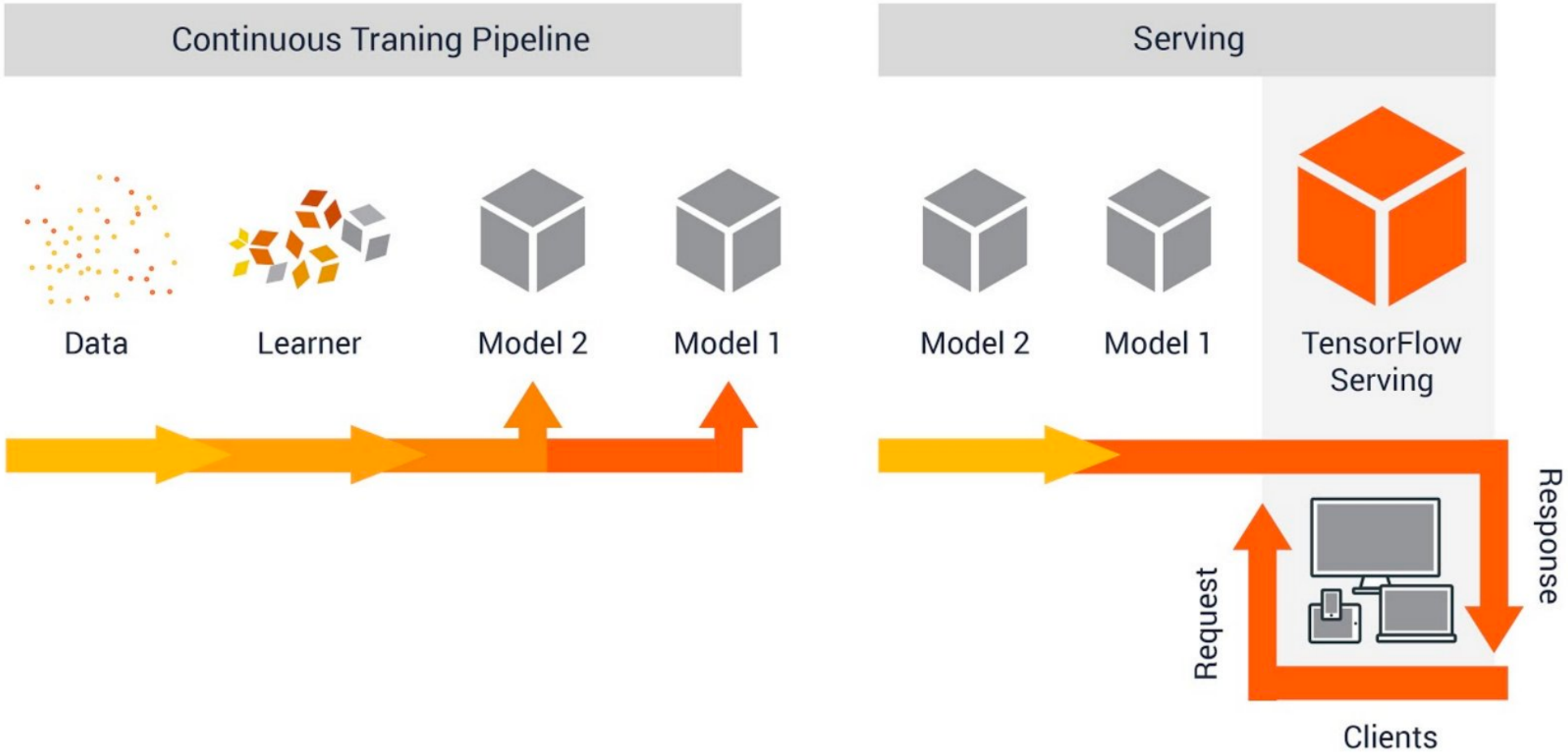
Training

- Distributed training
  - Distributed Tensorflow
  - Leverage Kubernetes to auto-scale training process
  - Tensorflow + Spark to get the best of both world
  - GCP Cloud ML for serverless processing
- HyperParameter Tuning
  - Kubernetes for parallel execution of hyperparameter tuning
  - Leverage Bayesian Optimization (Scikit-Optimize, SigOpt)
- Specialized Hardware
  - Leverage GPUs, TPUs for faster training

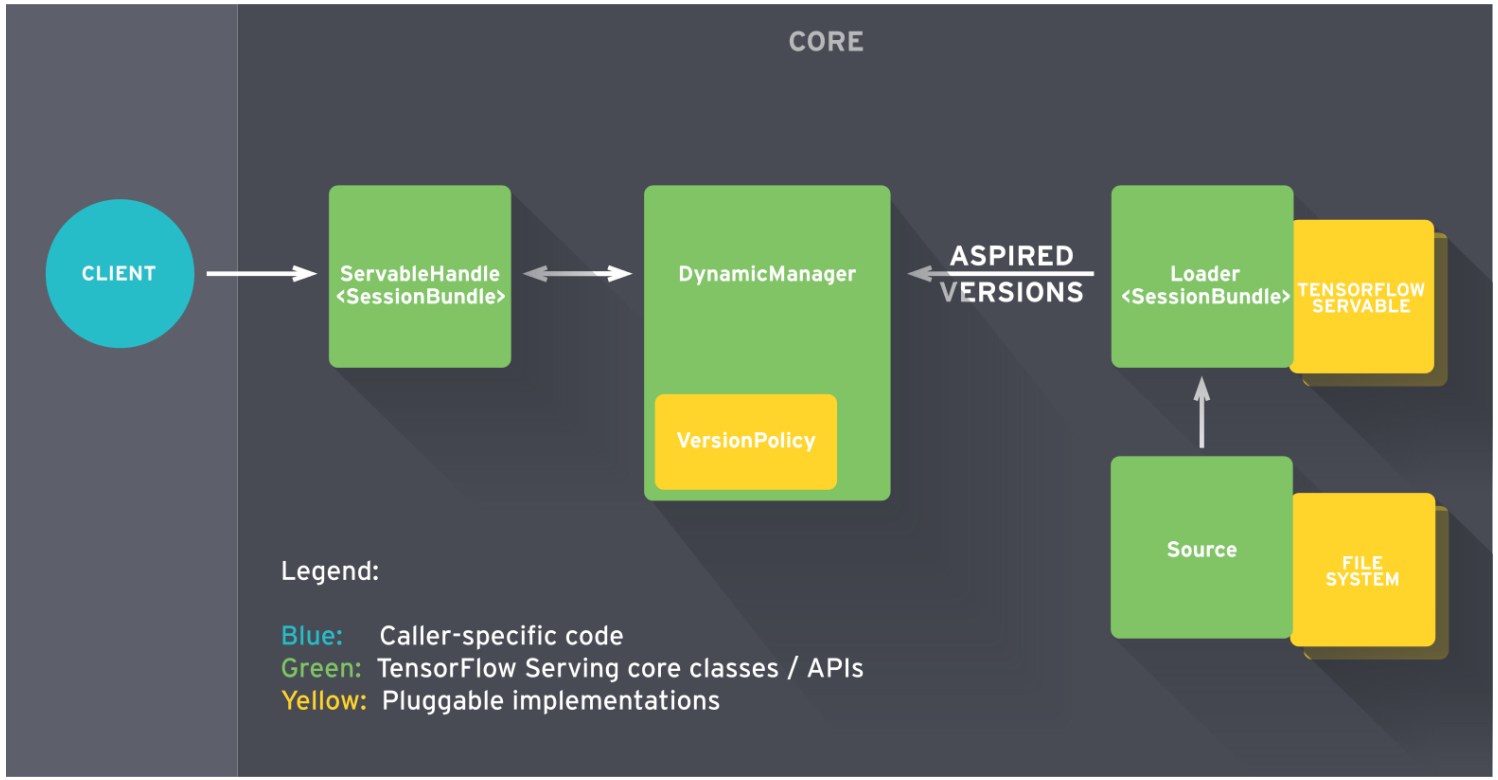
Inference

- Model Optimization
  - XLA
- Low latency inference
  - Tensorflow Serving to serve TF models

Tensorflow Serving

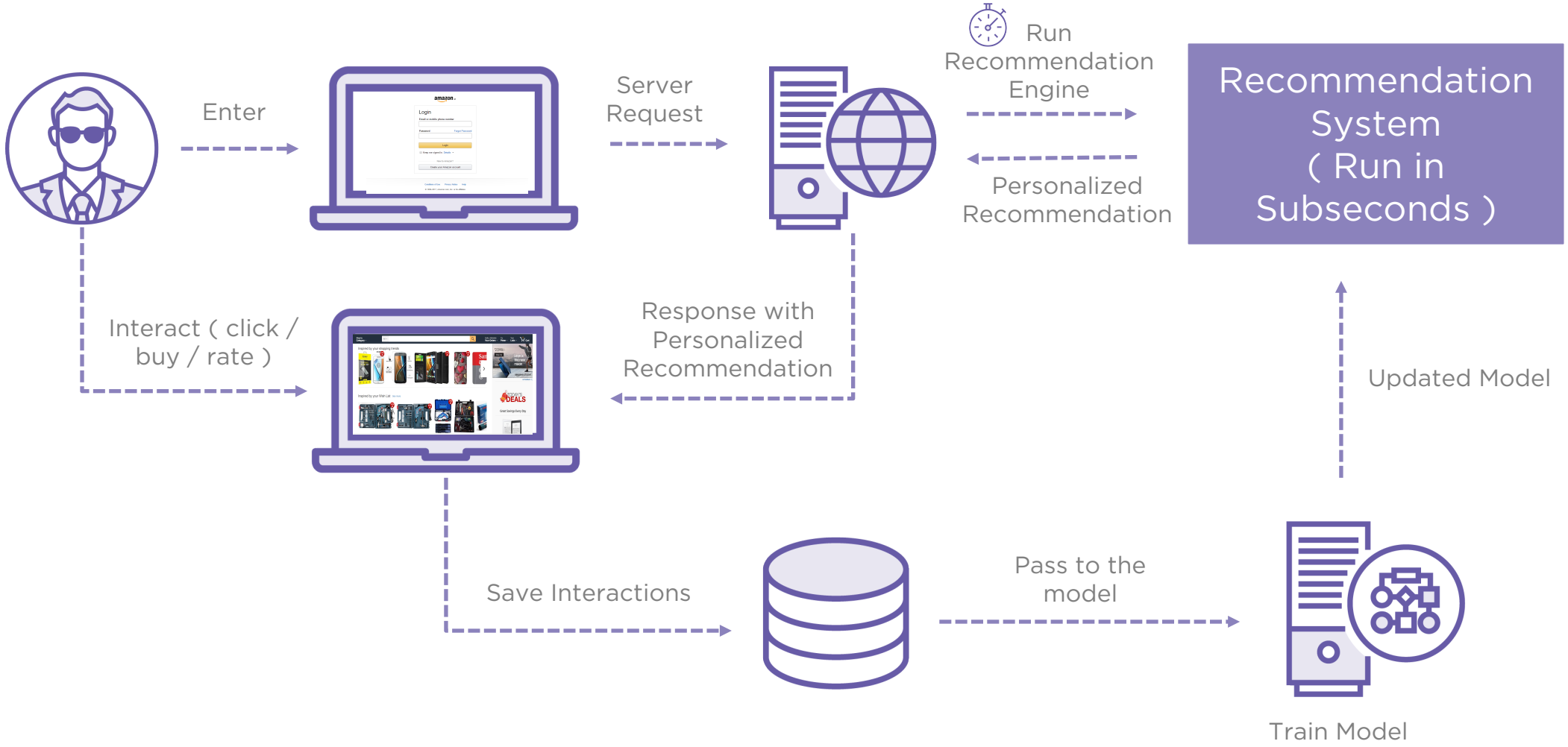


Tensorflow Serving

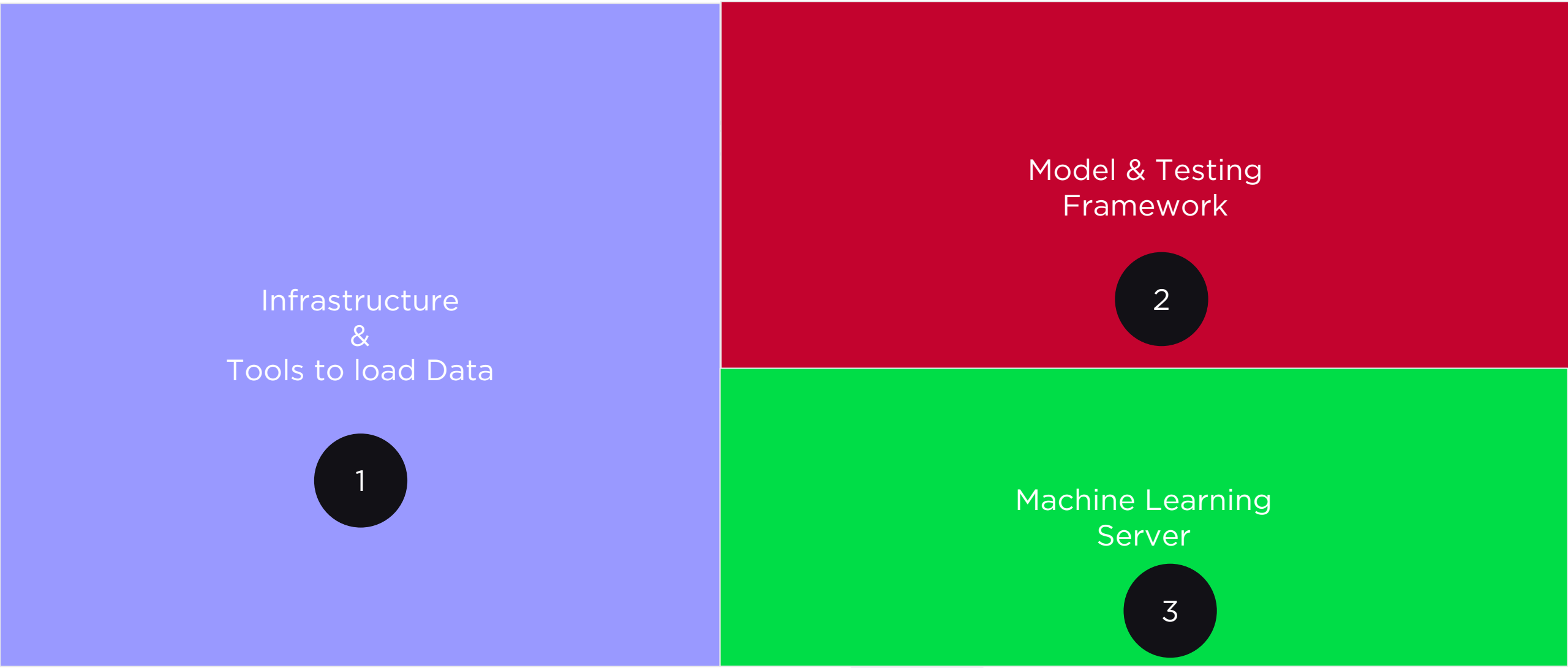


- Dynamic model loader
- Model versioning & rollback
- Written in C/C++ for low latency inference

How Recommendation System Works



Connecting the Dots



Credit : Lazada

Thanks

06-MARCH-2018

SAPIENT
RAZORFISH_

Next Steps

- Provide feedback on the tutorial
- Download & review tutorial material
  - Concepts
  - Demos
- Share
  - Progress, Issues, Use-cases
  - Twitter
    - @a_vijaysrinivas
    - @meabhishekkumar