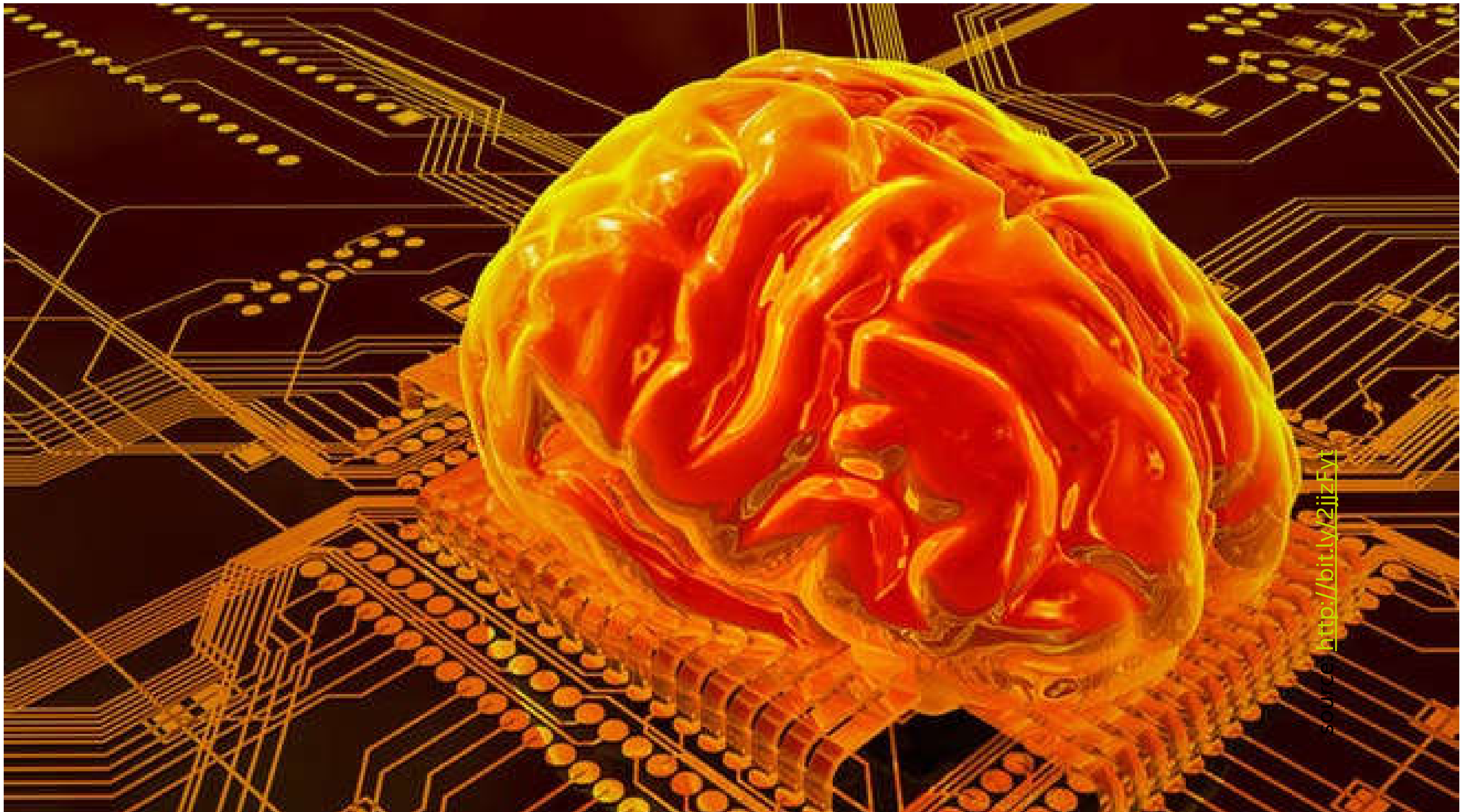
INTRODUCTION TO DEEP LEARNING

SERGEY ERMOLIN, BIG DATA TECHNOLOGIES

APRIL , 2018

sergey.v.ermolin@intel.com

github.com/sermolin



Artificial Neural Networks

Agenda

Morning: Deep Learning Intro, Part 1

- Use-cases: (Image-based property search, Recommendation engine, ~~MLS subscriber “churn”.,~~ Customized weekly real-estate newsletter generation, Anomaly detection)
- Apache Spark and BigDL

Afternoon:

- Deep Learning Frameworks (TensorFlow, PyTorch, Keras, BigDL)
- GPUs vs CPUs.
- Deep learning on Apache Spark and use-cases when it is appropriate
- Deep Learning Intro, Part 2
- Hands-on code review of Python cod

Acknowledgements

Ashish and Jeremy – for conceiving the idea of the workshop

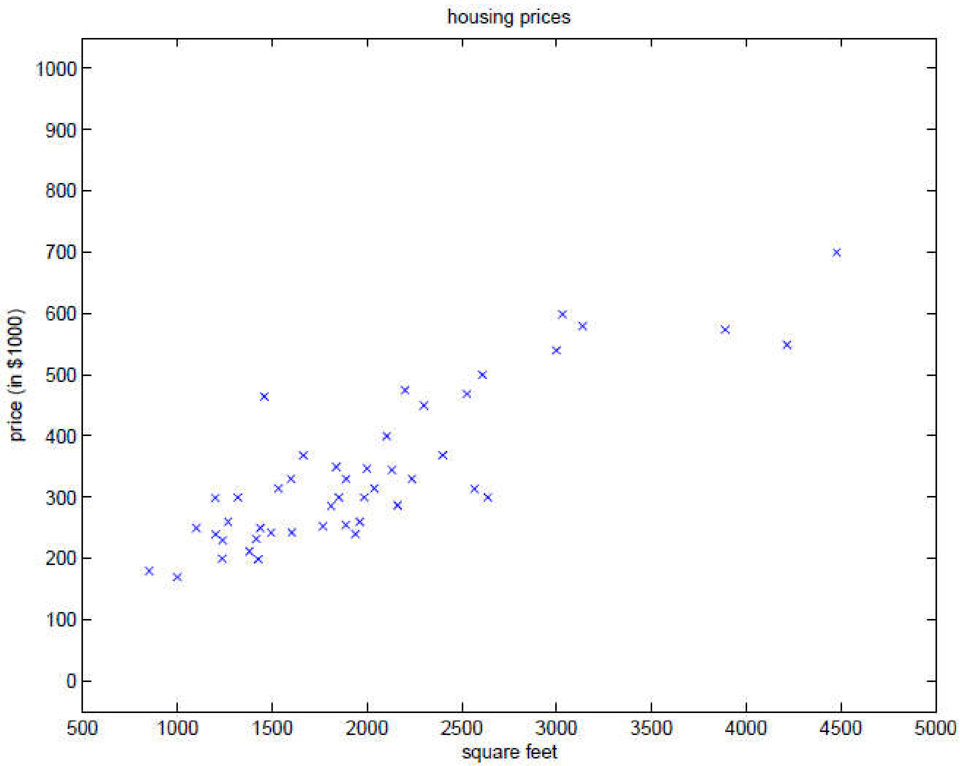
Olga Ermolin – for coming up with the house style classification use-case for it

Yuhao Yang (Intel) – for helping me prepare and debug Jupyter notebook

Alex Kalinin – for helping setting up backend infrastructure:

- Ten 96-core AWS servers.
- Twenty Docker containers running in each server.

LINEAR REGRESSION AS A NEURAL NETWORK



Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

Define cost function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

Find parameters that minimize the cost function

OBFUSCATING - NEW TERMS

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

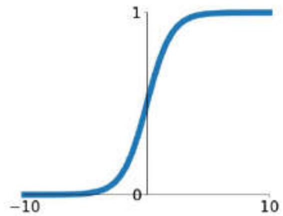
"Neuron" (plus a "non-linearity" or "activation function")

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

"Back-propagation"

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$ "Cost function"

$$h_\theta(x^{(i)}) - y^{(i)}$$ > 0 or <0?



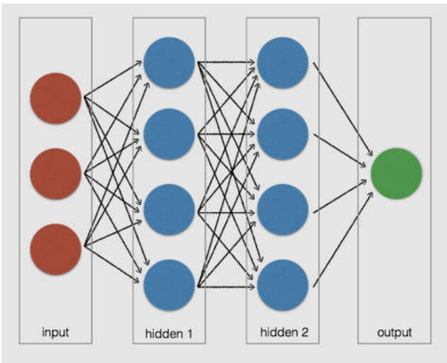
Sigmoid

"non-linearity"

Or

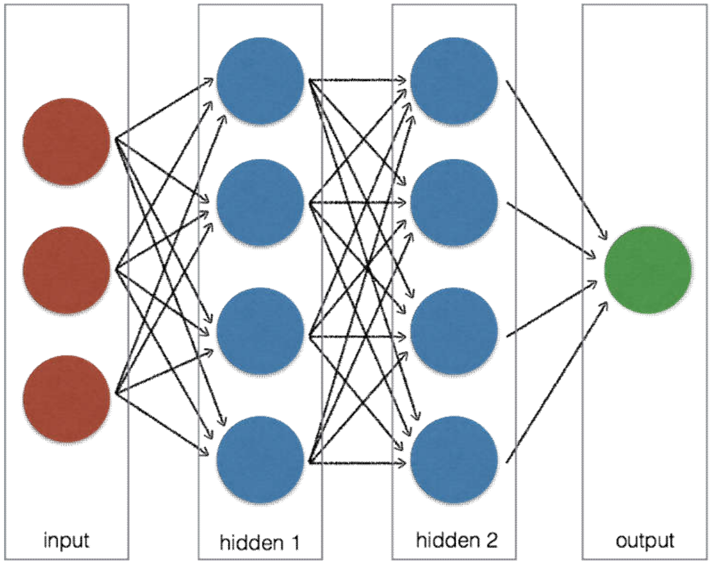
"activation function"

"Neural Net (Fully-connected)"



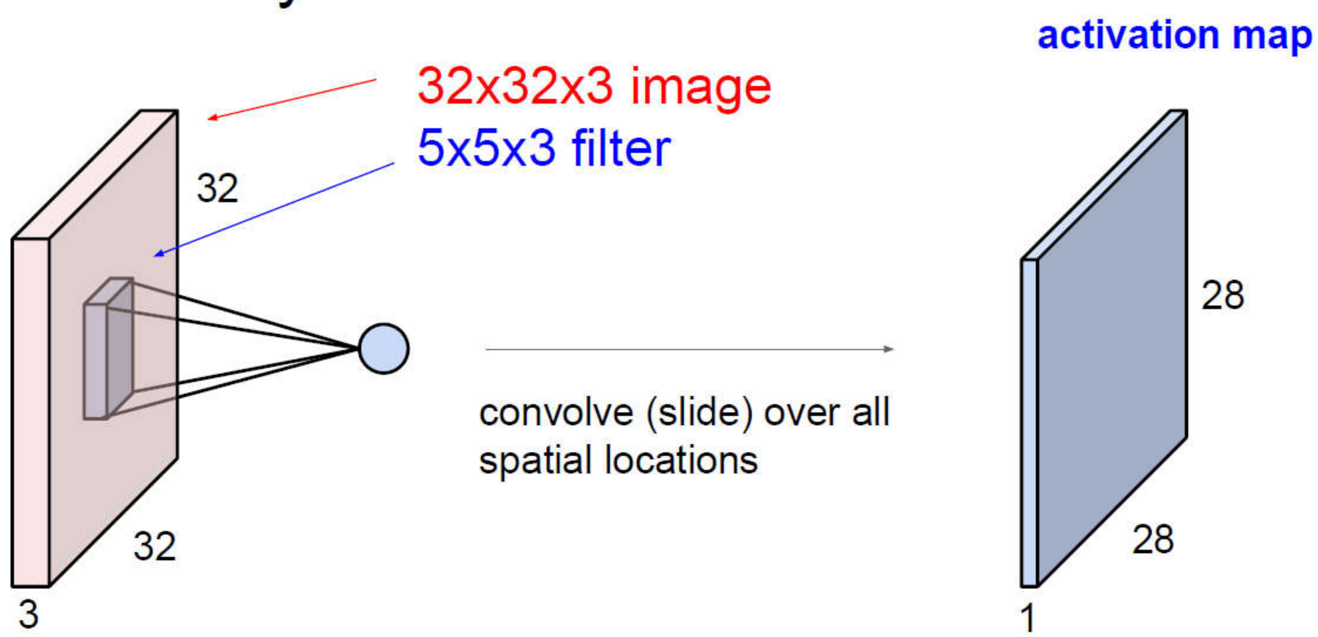
Artificial Neural Network (Fully-Connected)

- A "layer" is a set of neurons
- Typically consists of input, hidden and output layers
- Input layer acts as an interface
- The neurons in hidden layer(s) and the output modify data
- A black-box model
- Insanely computationally intensive. That's why it did not go anywhere in the 90s

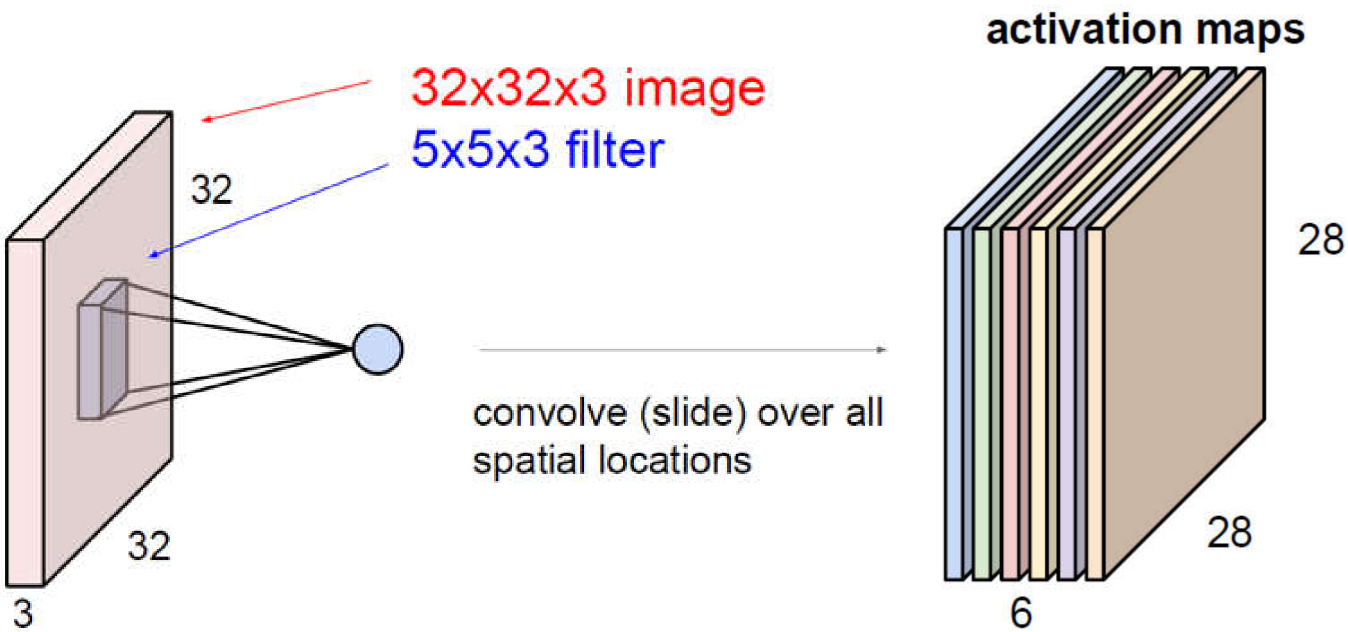


Enter Convolutional Neural Nets

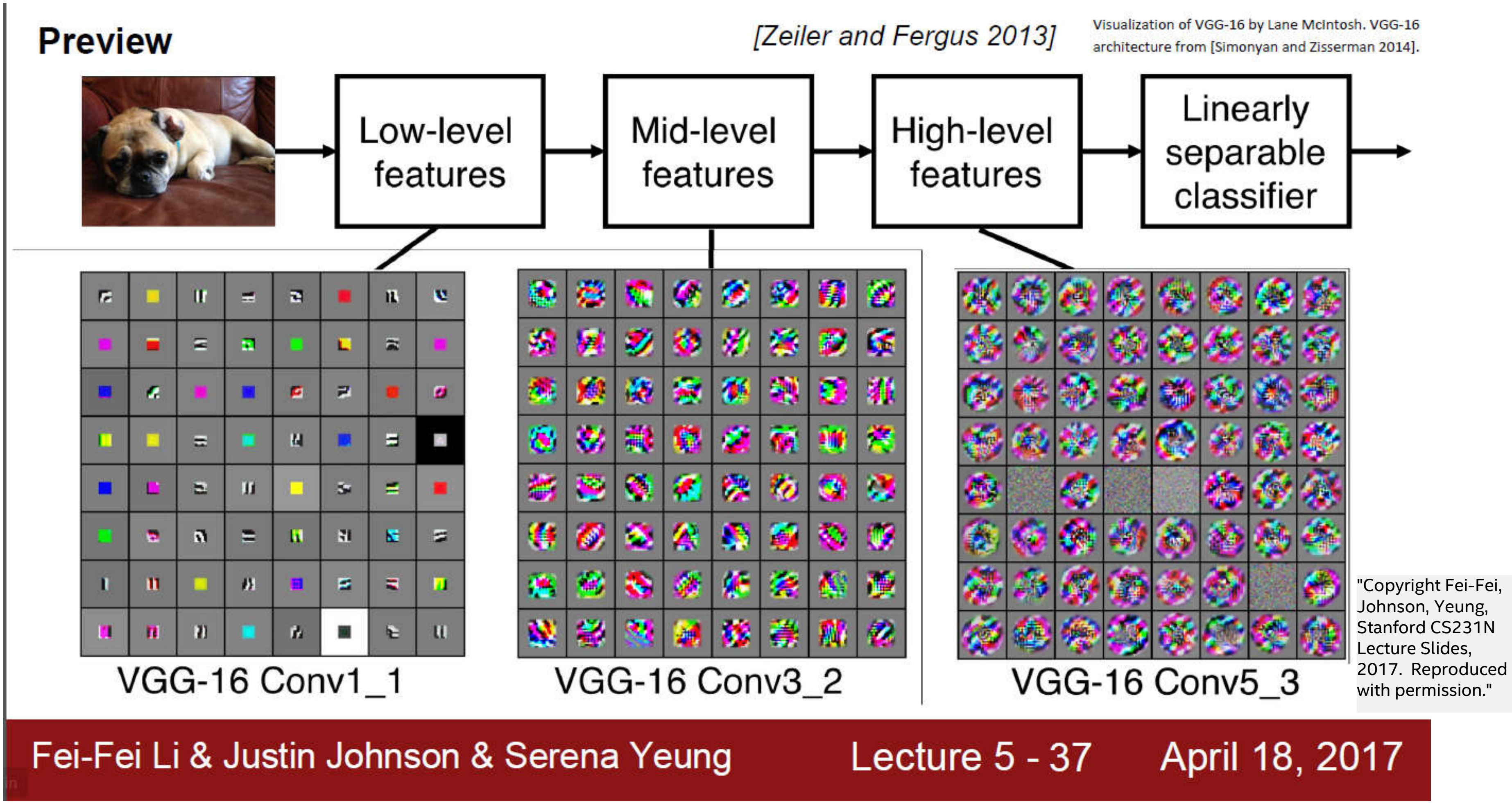
Convolution Layer

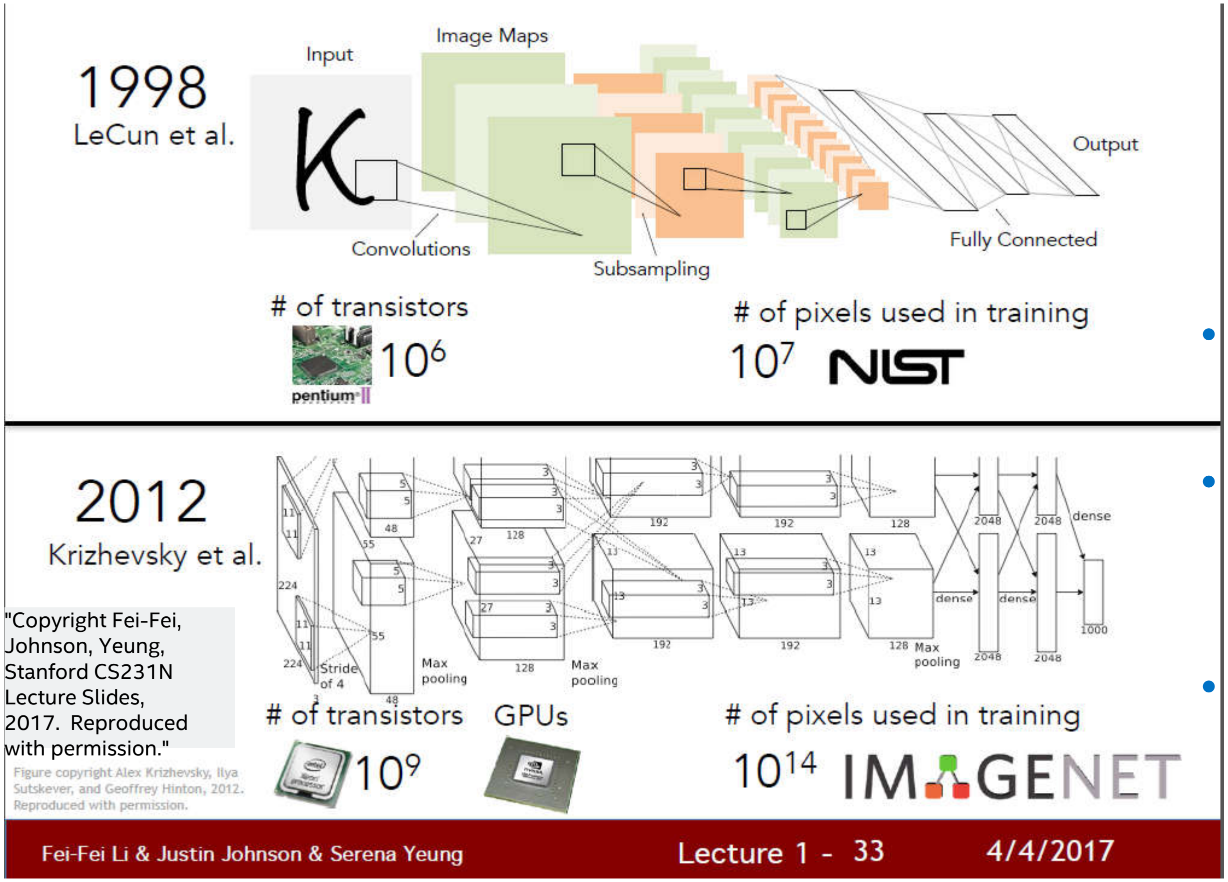


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



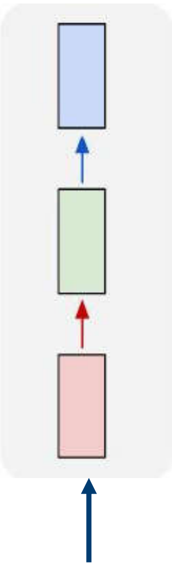
We stack these up to get a "new image" of size 28x28x6!

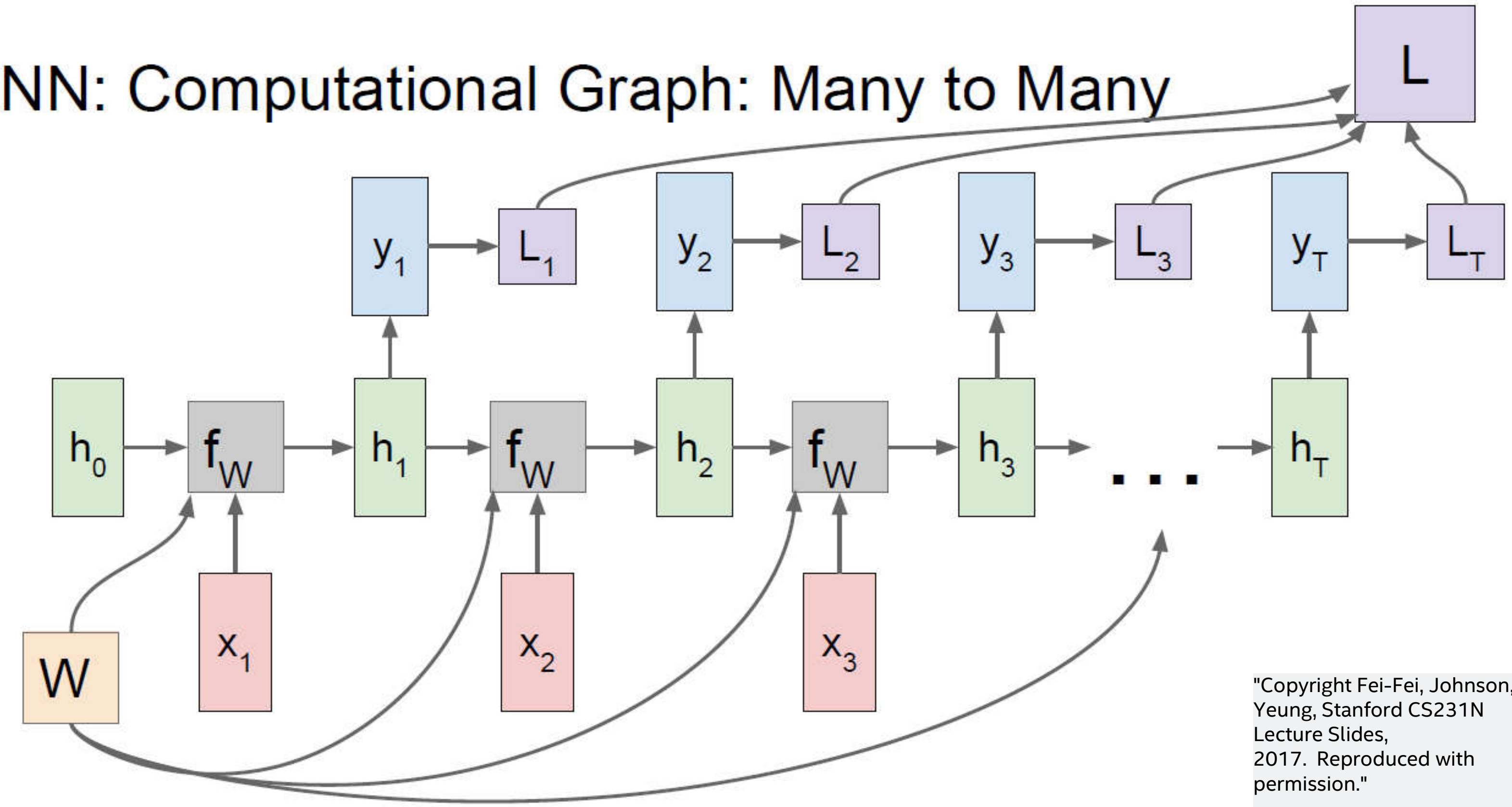

Preview
[Zeiler and Fergus 2013]
Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].
Low-level features
Mid-level features
High-level features
Linearly separable classifier
VGG-16 Conv1_1
VGG-16 Conv3_2
VGG-16 Conv5_3
"Copyright Fei-Fei, Johnson, Yeung, Stanford CS231N Lecture Slides, 2017. Reproduced with permission."
Fei-Fei Li & Justin Johnson & Serena Yeung
Lecture 5 - 37
April 18, 2017



- Still insanely computationally intensive. but better than fully-connected nets. Compute farms are a must even for small images
- Derivatives are **analytical**, not numerical
- Far more parameters than input variable. Unstable and easily fooled.
- NN research is somewhat of a dark art: "Improving performance without breaking the network"

RNN: Computational Graph: Many to Many





"vanilla" Feed-Forward NN

Deep Learning - Summary

- A way of extracting latent features from inputs and use them for predicting/affecting events
- Require to be trained on a large body of inputs to be useful
- Training is done via gradient minimization
- Insanely computationally intensive.

Future topics:

- Unsupervised learning.
- Reinforcement Learning
- Generative networks

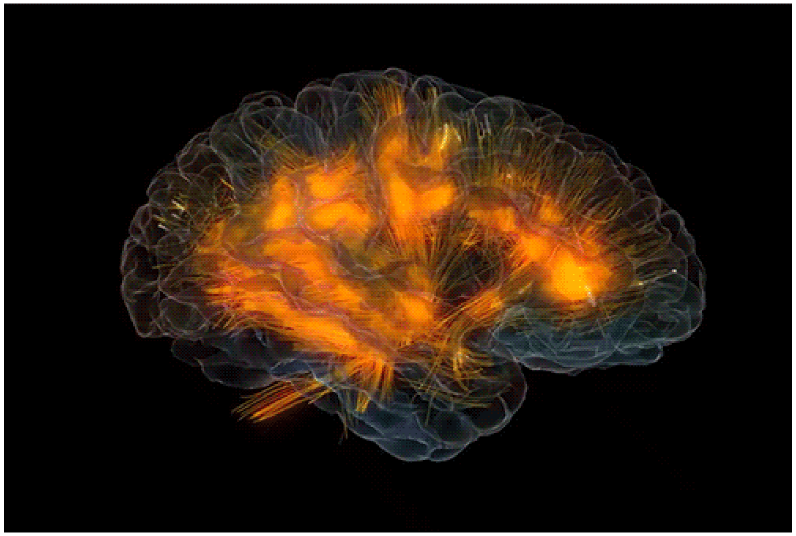
An imitation of human’s brain

Dense net of simple structures

Around 100 billion neurons

Each connected to ~10k other neurons

10^{15} synaptic connections



source: http://bit.ly/2jc9mYw

A neuron at work

Dendrites receive signals

Neuron's *cell body* acts as an accumulator

If energy level in the neuron's body exceeds certain level it fires a short pulse through the *axon* ended with *synaptic terminals*

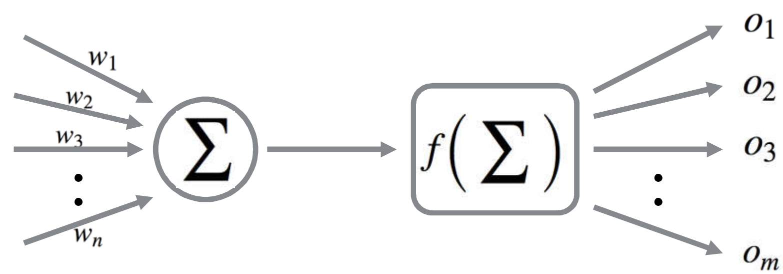
source: http://bit.ly/2jjGlNd

Conceptual mathematical model

- Receives input from *n* sources
- Computes weighted sum

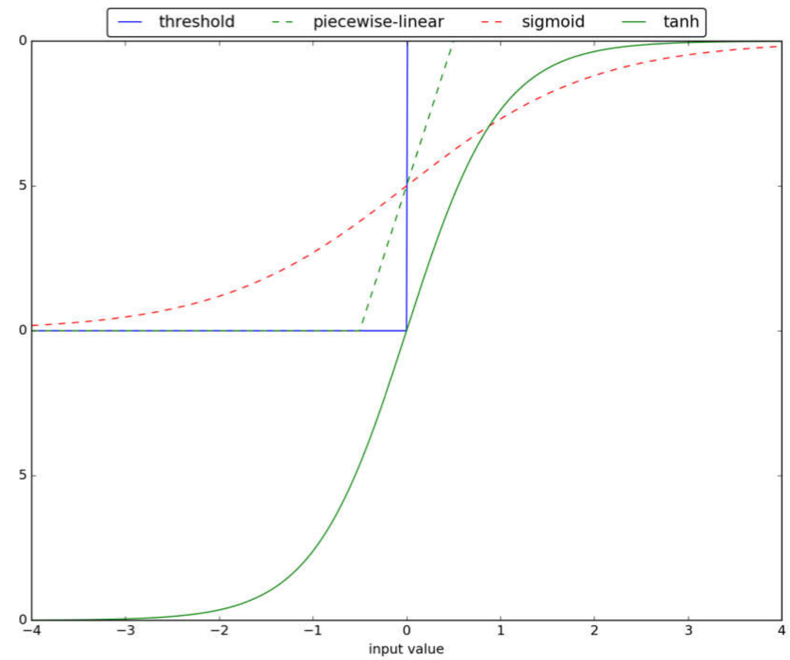
$$h_1 = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$

- Passes through an *activation function*
- Sends the signal to *m* succeeding neurons



Activation functions

- Bias (threshold) activation function was proposed first
- Sigmoid and tanh introduce non-linearity with different codomains

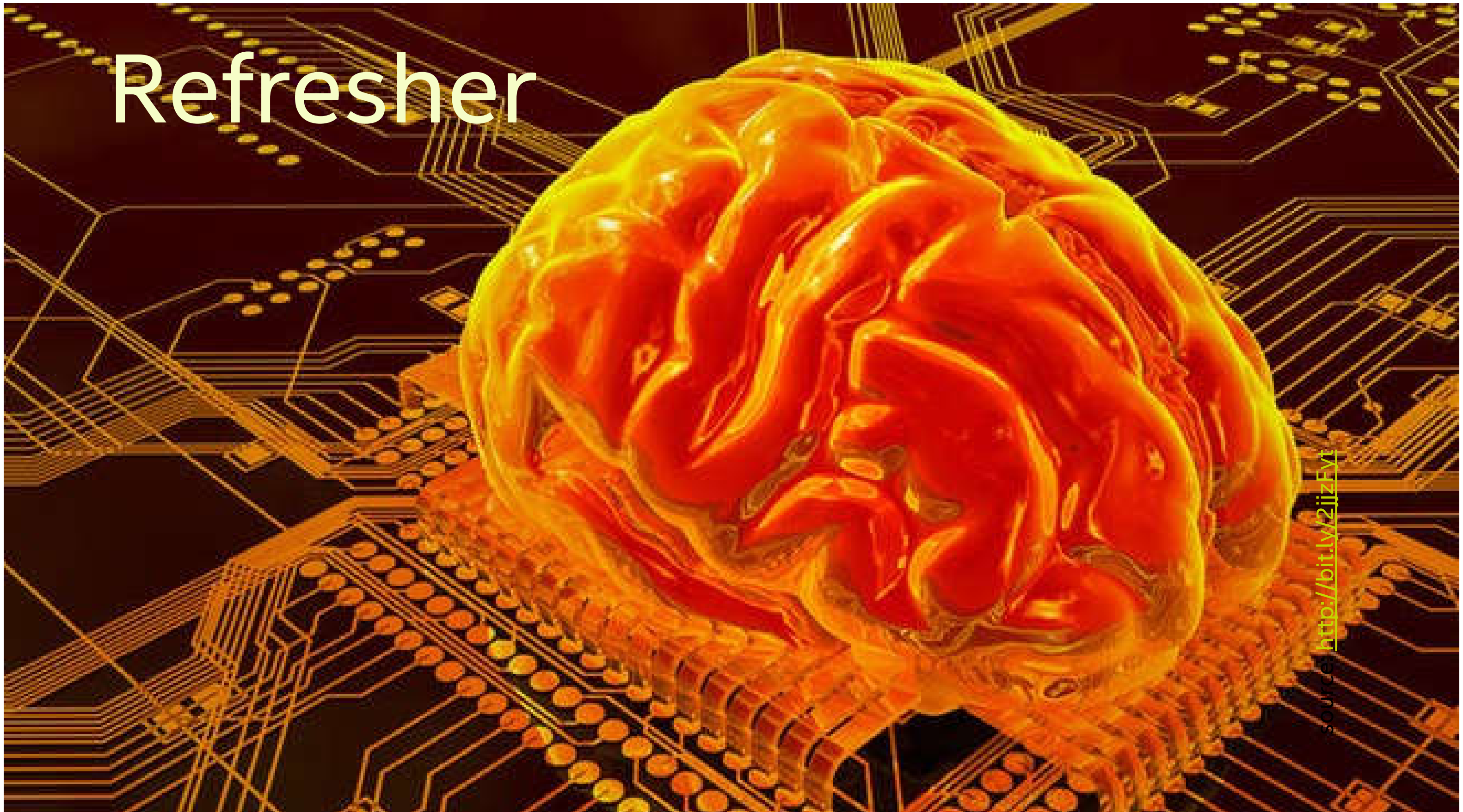


intel®
Software

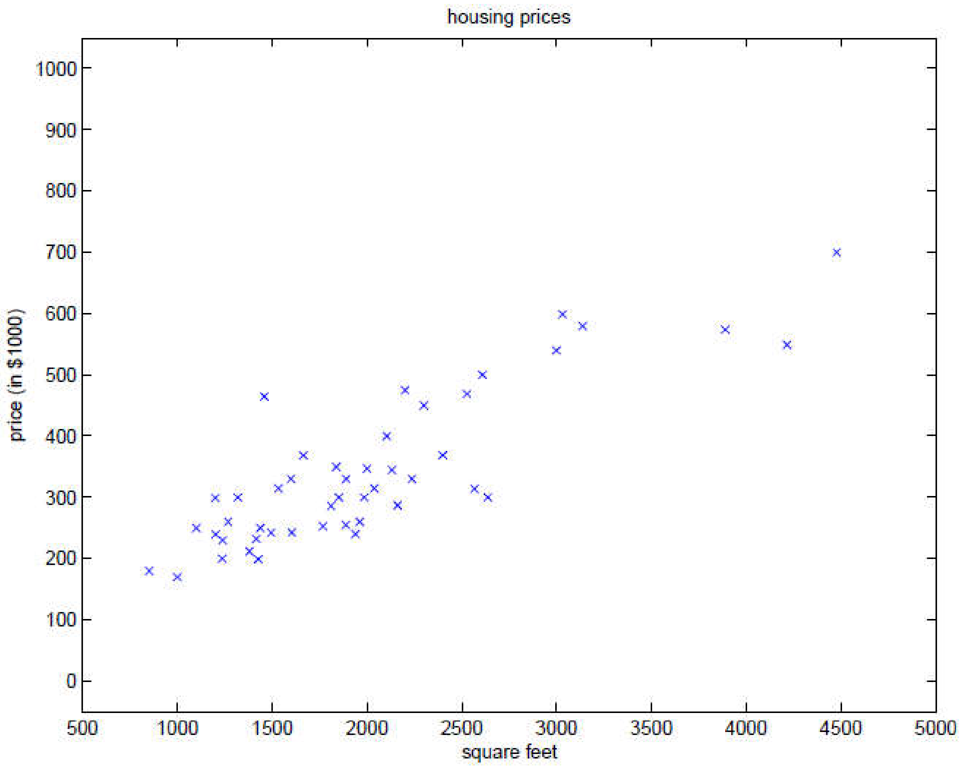
INTRODUCTION TO DEEP LEARNING – PART 2

SERGEY ERMOLIN, BIG DATA TECHNOLOGIES

APRIL , 2018


Refresher
source: http://bit.ly/2jjzFyt

LINEAR REGRESSION AS A NEURAL NETWORK



Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x,$$

Define cost function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

Model

True Values

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

Find parameters that minimize the cost function

Optimization:

1. Pick a model $h_\theta(x)$
2. Find <u>vector</u> θ that results in the smallest value of cost function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2.$$

3. Repeat step 2 until happy
4. If unhappy, repeat steps 1 & 2

Optimization:

Strategy #2: **Follow the slope**

Optimization:

Strategy #2: **Follow the slope**

In 1-dimension, the derivative of a function:

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

In multiple dimensions, the **gradient** is the vector of (partial derivatives) along each dimension

The slope in any direction is the **dot product** of the direction with the gradient
The direction of steepest descent is the **negative gradient**

Optimization:

This is silly. The loss is just a function of W:

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \sum_k W_k^2$$

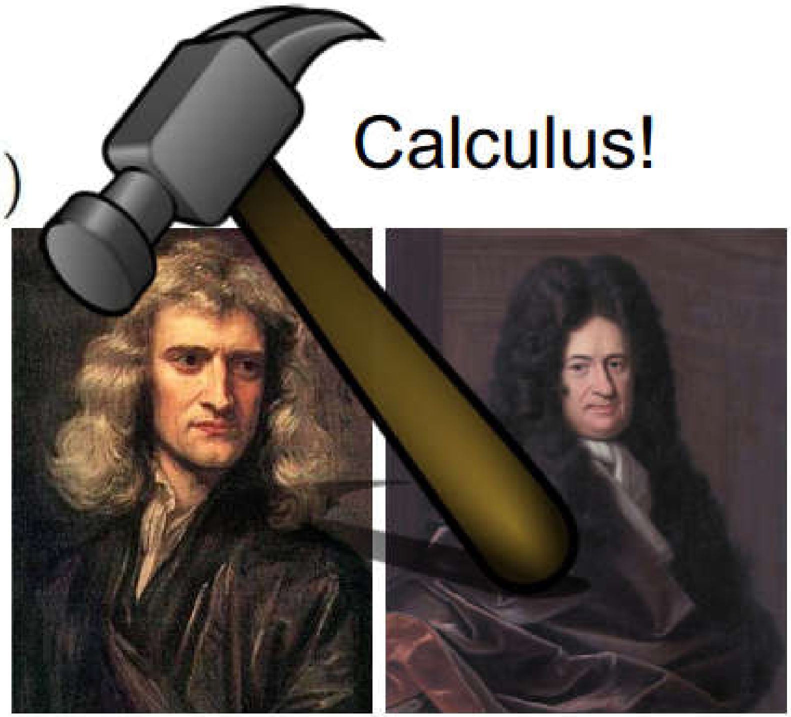
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$s = f(x; W) = Wx$$

want $\nabla_W L$

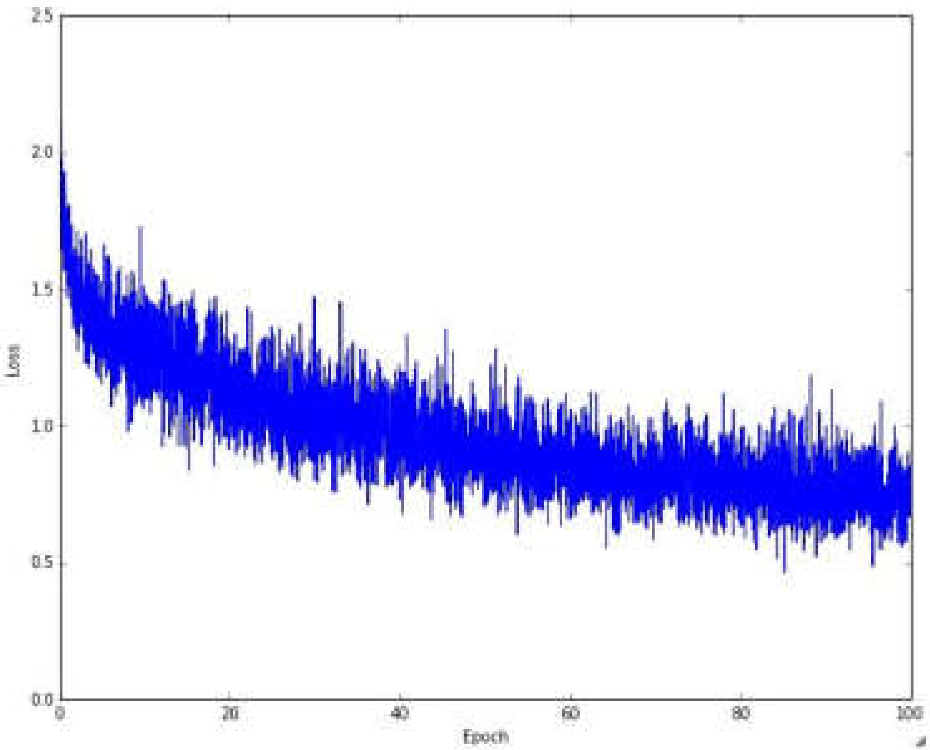
Use calculus to compute an **analytic gradient**

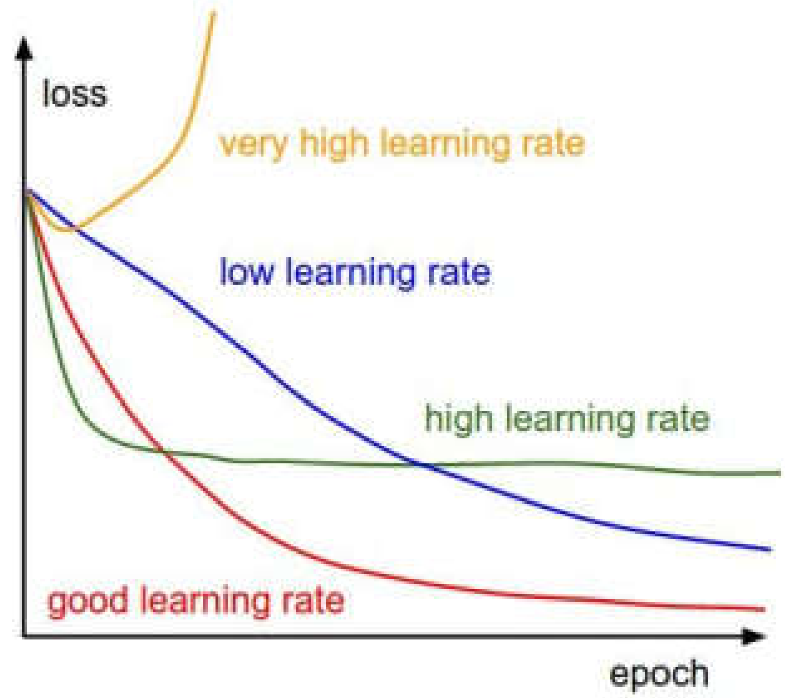
Calculus!



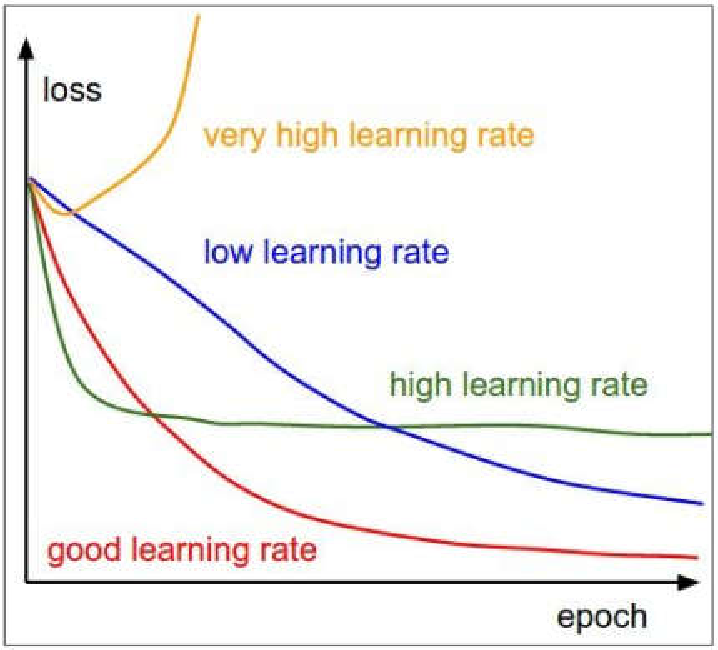
This image is in the public domain   This image is in the public domain

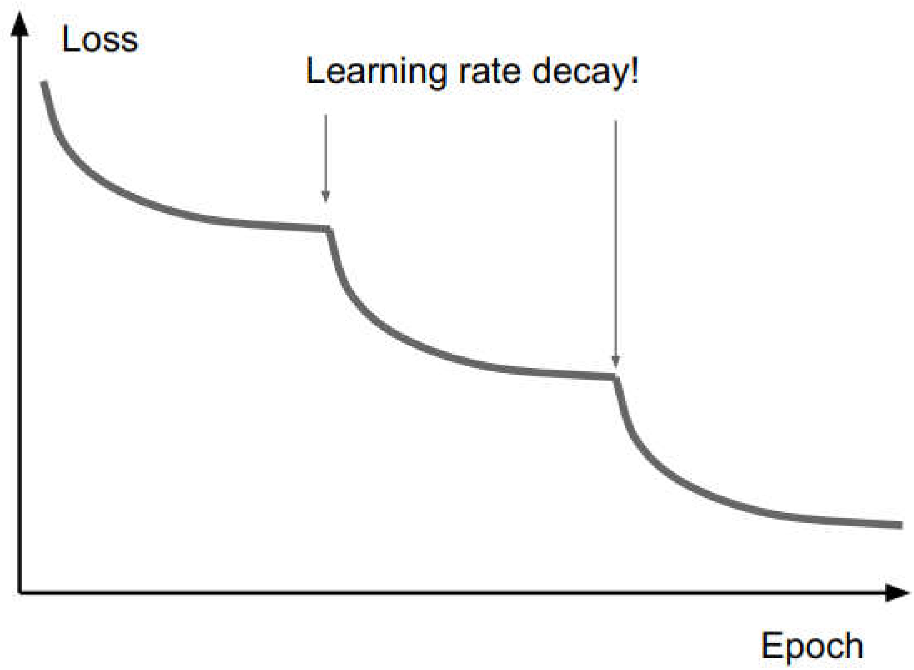
Monitor and visualize the loss curve





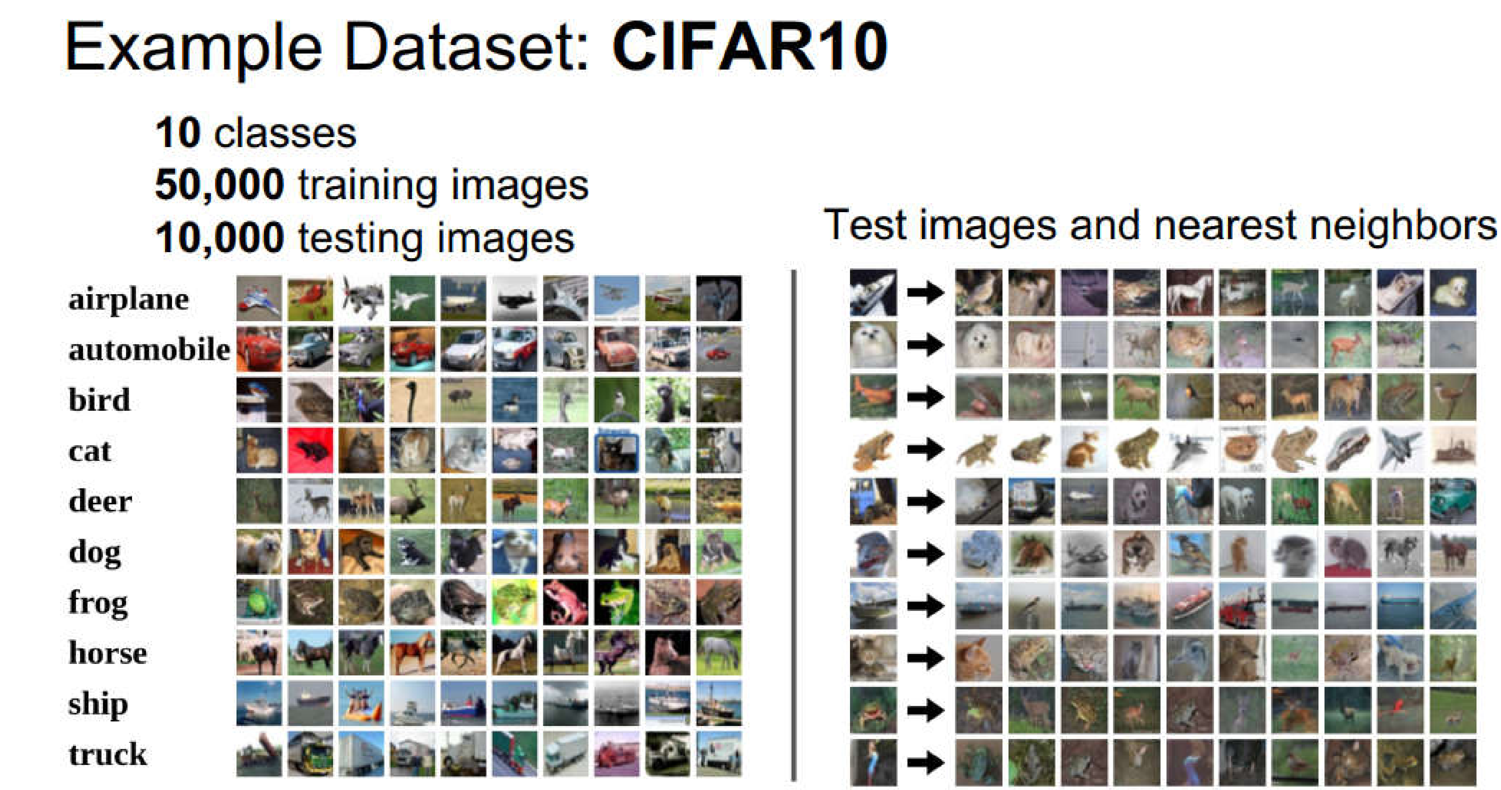
SGD, SGD+Momentum, Adagrad, RMSProp, Adam all have **learning rate** as a hyperparameter.





Yeung, Stanford CS231N Lecture Slides, 2017. Reproduced with permission."

Image Classification Problem



Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.

Task:

Given an image, place it into one of 10 class "buckets"

Image Classification Problem – K-means approach

Distance Metric to compare images

K-Nearest Neighbors

L1 distance: $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$

Instead of copying label from nearest neighbor, take **majority vote** from K closest points

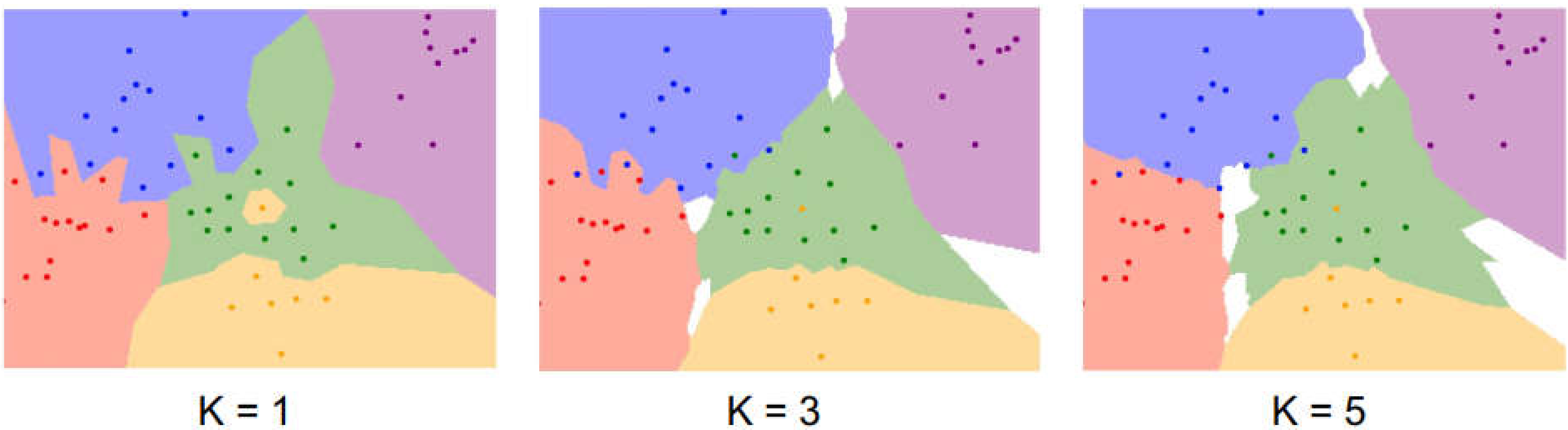


Image Classification Problem – SoftMax (better)

Softmax Classifier (Multinomial Logistic Regression)

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

unnormalized probabilities

	unnormalized log probabilities		unnormalized probabilities		probabilities	
cat	**3.2**	exp →	**24.5**	normalize →	**0.13**	→ L_i = -log(0.13) = **0.89**
car	5.1		164.0		0.87	
frog	-1.7		0.18		0.00	

unnormalized log probabilities

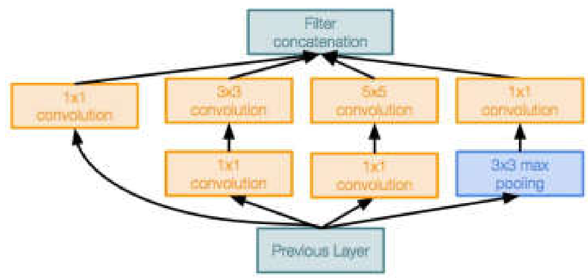
probabilities

Case Study: GoogLeNet

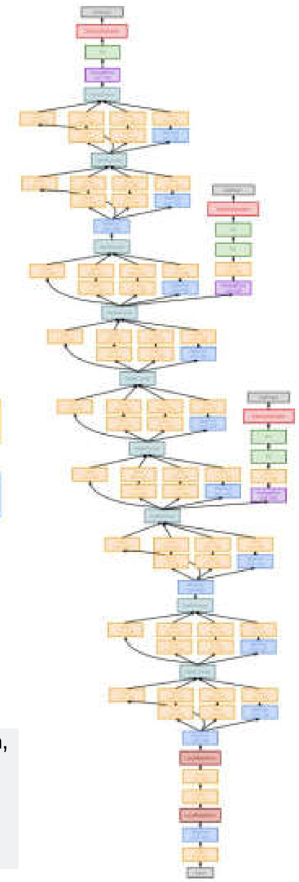
[Szegedy et al., 2014]

Deeper networks, with computational efficiency

- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters!
  12x less than AlexNet
- ILSVRC’14 classification winner
  (6.7% top 5 error)



Inception module



Transfer Learning

"You need a lot of a data if you want to train/use CNNs"

Yeung, Stanford CS231N Lecture Slides, 2017. Reproduced with permission."

Transfer Learning

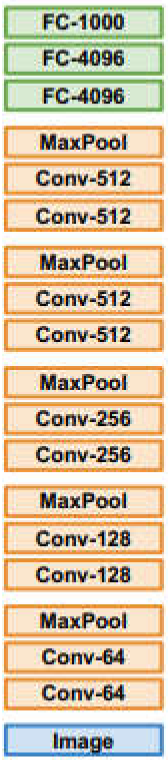
"You need a lot of a data if you want to train/use CNNs"

BUSTED

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

1. Train on Imagenet



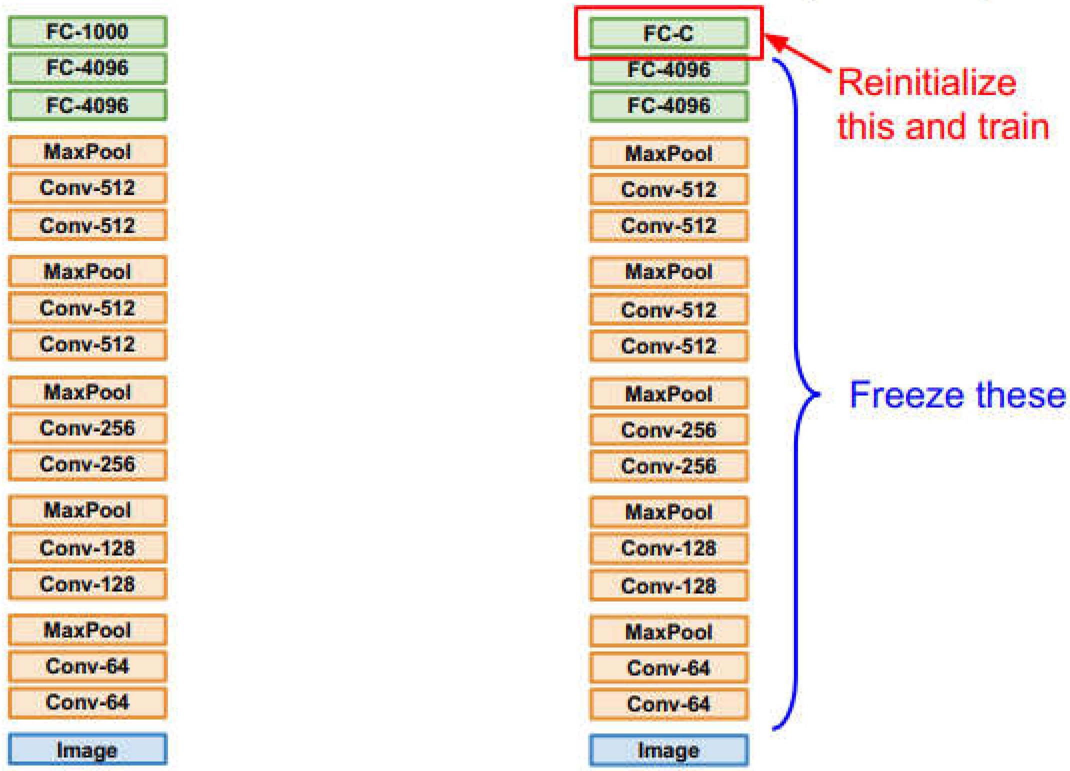
2017. Reproduced with permission."

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

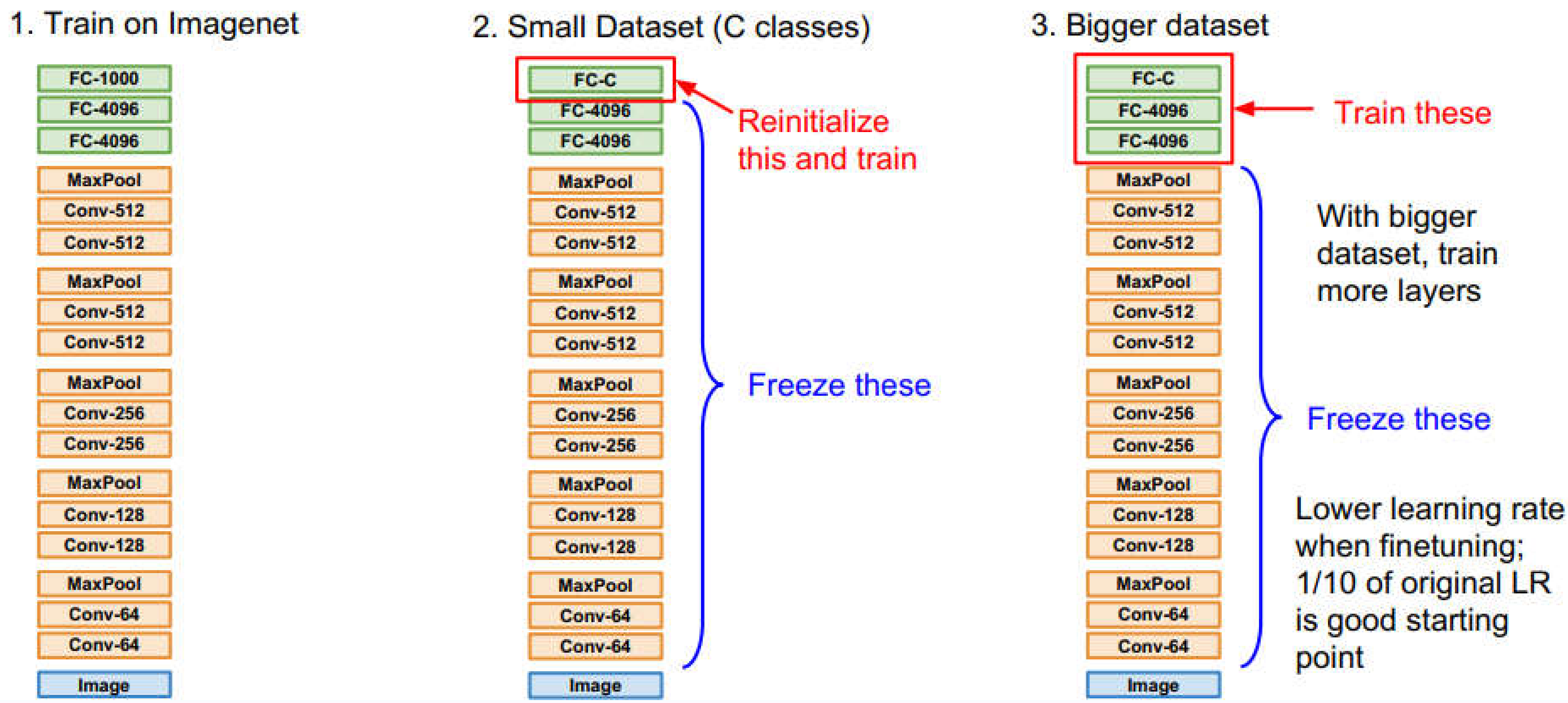
1. Train on Imagenet

2. Small Dataset (C classes)



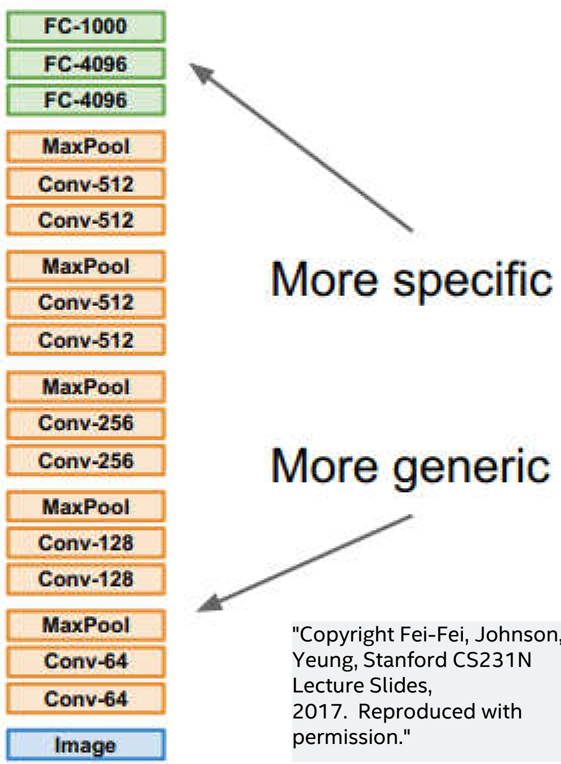
Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014



Lecture Slides, 2017. Reproduced with permission."

Transfer Learning with CNNs



	very similar dataset	**very different dataset**
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

Transfer Learning is Pervasive. It is a norm, rather than an exception

Tying it all together – House Style classification

- Training: No "deep learning" training on new images.
  - Much faster than full retraining. No parameter tuning. No retraining (or infrequent)
- Single-pass-per-image Deep Learning **prediction.**
  - Can be real-time in production (sub 1-sec). Low compute resources needed. No GPU.
- Classification is done using linear classifier
- Can be *easily* extended to 100s of house style classe.

Deep Learning - Summary

- A way of extracting latent features from inputs and use them for predicting/affecting events
- Require to be trained on a large body of inputs to be useful
- Training is done via gradient minimization
- Insanely computationally intensive.

Future topics:

- Unsupervised learning.
- Reinforcement Learning
- Generative networks

Software

MACHINE LEARNING USE-CASES

BIG DATA TECHNOLOGIES, SOFTWARE & SERVICES GROUP

WHO IS BUILDING WHAT WITH BIGDL?

CONSUMER

- ✓ Gigaspaces
- ✓ MLS Listings
- ✓ Jobs Search Engine

- ➢ Call center routing,
- ➢ Image similarity search,
- ➢ smart job search

HEALTH

- ✓ UCSF

- ➢ Analysis of 3D MRI models for knee degradation

FINANCE

- ✓ UnionPay
- ✓ ChinaLife (Insurance)
- ✓ Mastercard

- ➢ Fraud detection
- ➢ Recommendation,
- ➢ Customer/Merchant Propensity

RETAIL

- ✓ JD.Com

- ➢ Image feature extraction (Inference)

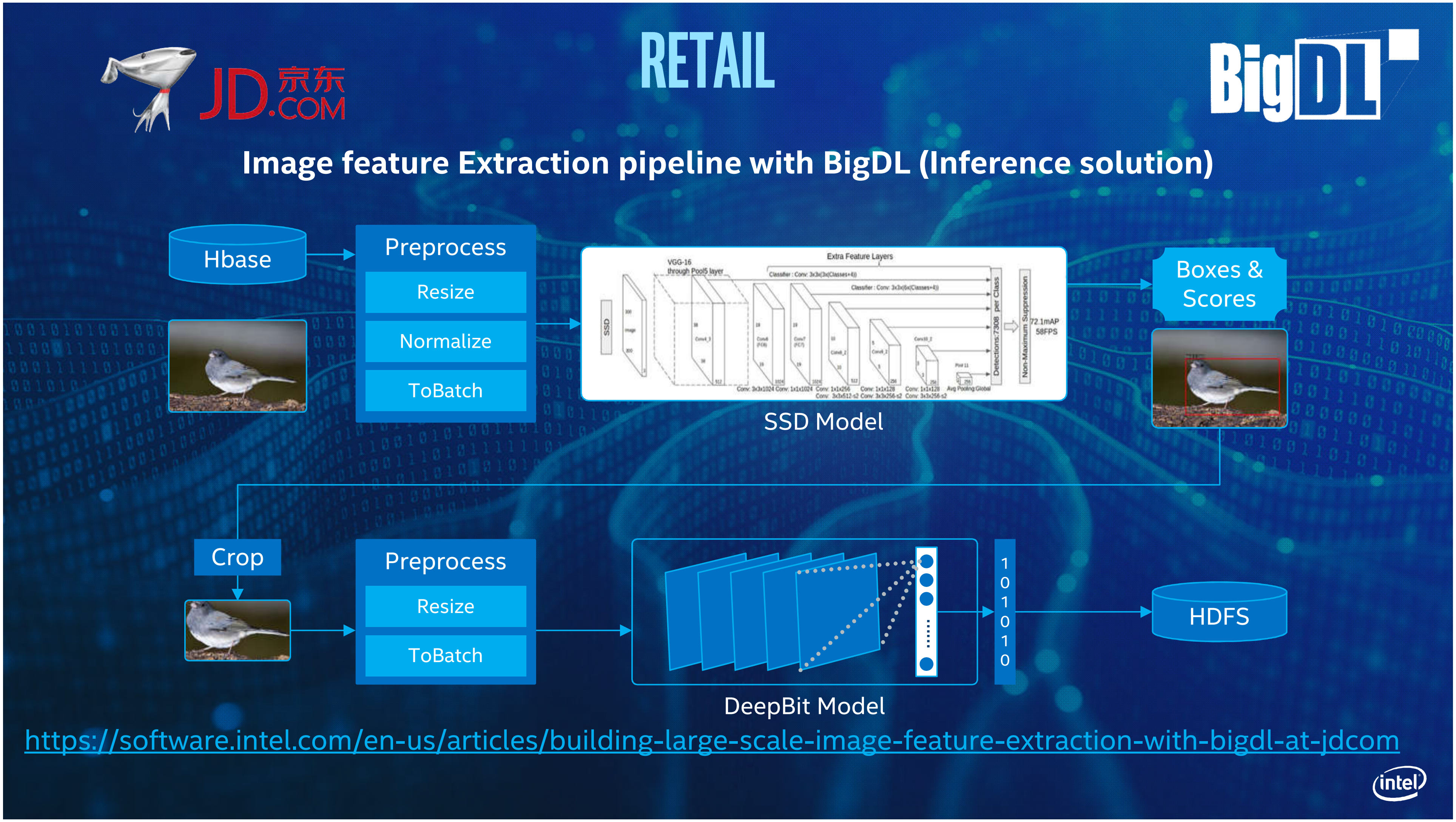
MANUFACTURING

- ✓ Steel manufacturing

- ➢ Steel Surface defect detection
- ➢ Weather forecasting

SCIENTIFIC COMPUTING

- ✓ Cray


RETAIL
Image feature Extraction pipeline with BigDL (Inference solution)
Hbase
Preprocess
Resize
Normalize
ToBatch
SSD Model
Boxes & Scores
Crop
Preprocess
Resize
ToBatch
1 0 1 0 1 0
DeepBit Model
HDFS
https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom

GIGASPACES
innovate with confidence

CUSTOMER SERVICE

BigDL

Artificial Intelligence Redefines the Call Center Customer Experience (Training)

Stop Pressing 0 or *

Automatic routing to the right agent for the perfect personalized experience

I have a windows MAC problem

NLP Processing

Training, prediction, and tuning

Route to the MAC expert

User peaks using web interface	Browser converts speech to text and sends to controller	Controller writes data to InsightEdge and to Kafka topic	Spark job listens on Kafka topic and using BIGDL mode, creates prediction	BIGDL writes Prediction to InsightEdge data grid	InsightEdge event processor listens or Prediction data and routes call session

https://blog.gigaspaces.com/gigaspaces-to-demo-with-intel-at-strata-data-conference-and-microsoft-ignite/

intel

MLS LISTINGS

REAL ESTATE

BigDL

If you looked at this house.....

You will want to look at this one, too

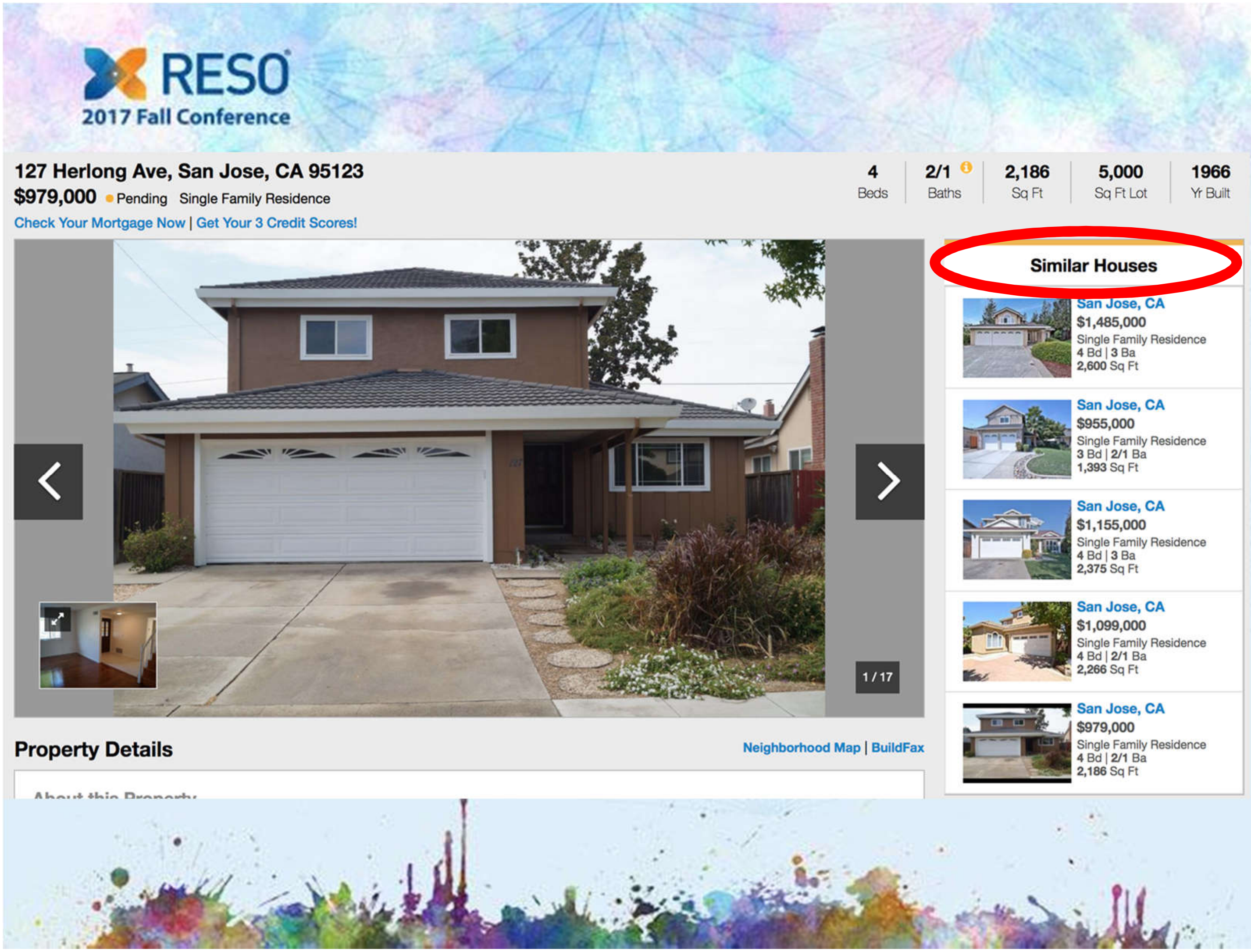
Non-real time indexing/bucketizing of similar images in the database.

Image similarity ("distance") becomes an extra parameter in addition to area, location, size, price, etc.

Runs periodically on the refreshed database.

Needs to be scalable nationwide. Distributed compute solution needed (Spark)

MLS Listings image similarity PoC demo is ready and available here: https://homes-prod-homes-poc.azurewebsites.net/



RESO
2017 Fall Conference
127 Herlong Ave, San Jose, CA 95123
$979,000 Pending Single Family Residence
Check Your Mortgage Now | Get Your 3 Credit Scores!
4 Beds
2/1 Baths
2,186 Sq Ft
5,000 Sq Ft Lot
1966 Yr Built
1 / 17
Similar Houses
San Jose, CA
$1,485,000
Single Family Residence
4 Bd | 3 Ba
2,600 Sq Ft
San Jose, CA
$955,000
Single Family Residence
3 Bd | 2/1 Ba
1,393 Sq Ft
San Jose, CA
$1,155,000
Single Family Residence
4 Bd | 3 Ba
2,375 Sq Ft
San Jose, CA
$1,099,000
Single Family Residence
4 Bd | 2/1 Ba
2,266 Sq Ft
San Jose, CA
$979,000
Single Family Residence
4 Bd | 2/1 Ba
2,186 Sq Ft
Property Details
Neighborhood Map | BuildFax

RESO
2017 Fall Conference

Deep Learning Data Flow

Train

Data Engineering | **Coding** | **Compute**

Labeled Dataset of Real Estate Images (Bing Search) + BigDL VGG Arch = BigDL Trained Model

- Long Compute. 8500 images
- 2 nodes, 28 cores/node. 3 minutes for a one single pass
- Model parameters are changing.
- Repeat until convergence
- But: only do once !

  Note: Images are *not* stored in the model

  Note: you can trade compute resources for time.

RESO 2017 Fall Conference

Deep Learning Data Flow

Score

Compute Only

+

BigDL Trained Model

=

Feature Vector

- Image Class (Front, Bdr, Bath,…)
- House Style Tag (Ranch, Victorian,…)
- House Levels (1, 2..)
- Latent Features (25k entries)

- Short Compute. 2 sec/image
- 1 node, 1 core/node
- Model parameters unchanged.
- Only run once per image
- But: need to do for every image in the search dataset !

RESO
2017 Fall Conference

Deep Learning Data Flow

Training

Labeled Dataset of Real Esate Images (Bing Search) + BigDL VGG Arch = BigDL Trained Model

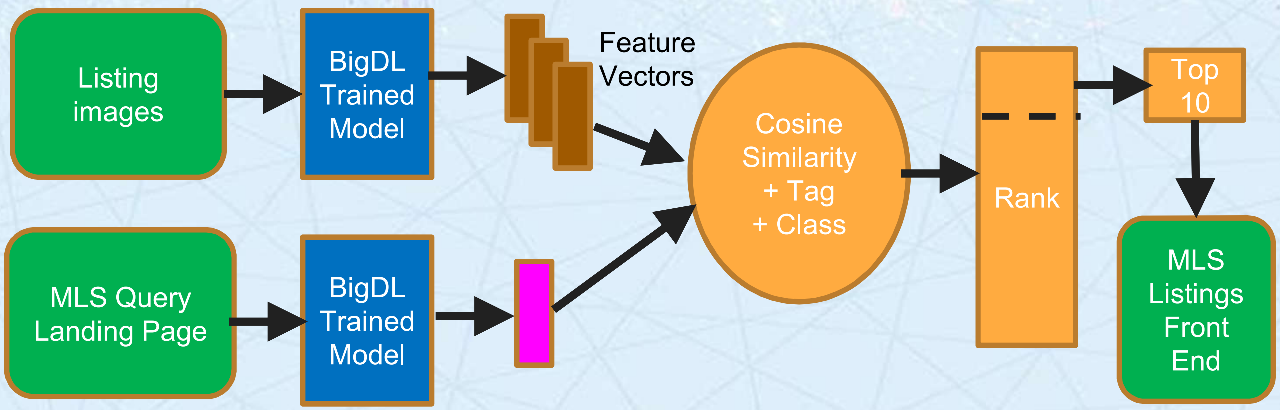
Scoring

\+ BigDL Trained Model = Feature Vector

- Image Class (Front, Bdr, Bath,…)
- House Style Tag (Ranch, Victorian,…)
- House Levels (1, 2..)
- Latent Features (25k entries)

Deep Learning Data Flow **Scoring**

RESO 2017 Fall Conference



Cosine similarity measure:

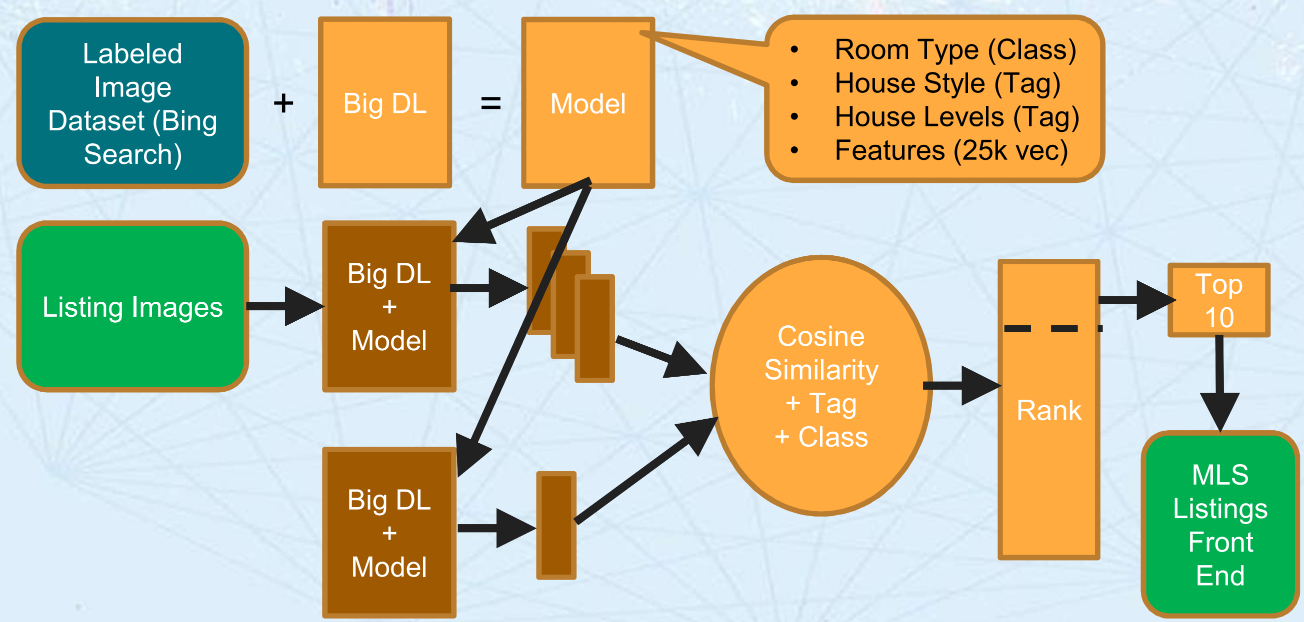
- $\mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) = \cos(\boldsymbol{r_x}, \boldsymbol{r_y}) = \dfrac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}$

r_x, r_y as points:

$r_x = \{1, 0, 0, 1, 3\}$

$r_y = \{1, 0, 2, 2, 0\}$

Deep Learning Data Flow



RESO
2017 Fall Conference

Engineering Team

- Data scientist, proficient in Machine Learning / Deep Learning
- Software Engineer, experience with Apache Spark.
- Technical project manager

Domain Expertise:

- Machine Learning / Deep Learning,
- Python, Scala

- Software Engineer, Web API
- Software Engineer, Web UI

Domain Expertise:

- OData, .net Core MSSQL
- C#, HTML, JavaScript

RETAIL: PREDICTING CUSTOMER LOYALTY

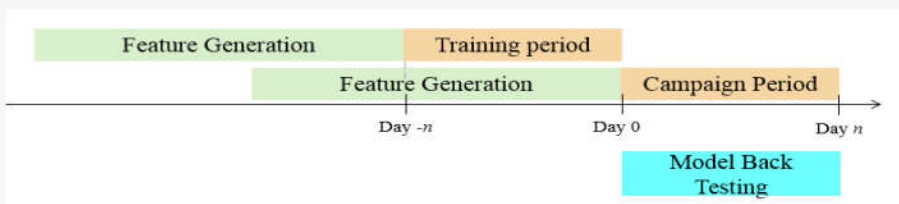
BigDL

How likely is an existing (or a new) customer to come back to a store?
What would it take to get a customer in the door?

Business Use Case

Business driven :

There are a variety of goals that advertising campaigns are targeted offering, like creating awareness or re-targeting consumers. For example, the latter could involve estimating the propensity of consumers to shop target merchants within several days. We take PCLO (Personal Card Lined Offers) use cases as our running example and focus on predicting users-merchants probability.



Goal :

implement new Users-Items Propensity Models with deep learning algorithms base on Intel BigDL framework to help our Loyalty solution to improve the quality , performance and accuracy of offer and campaigns design, targeting offer matching and linking .

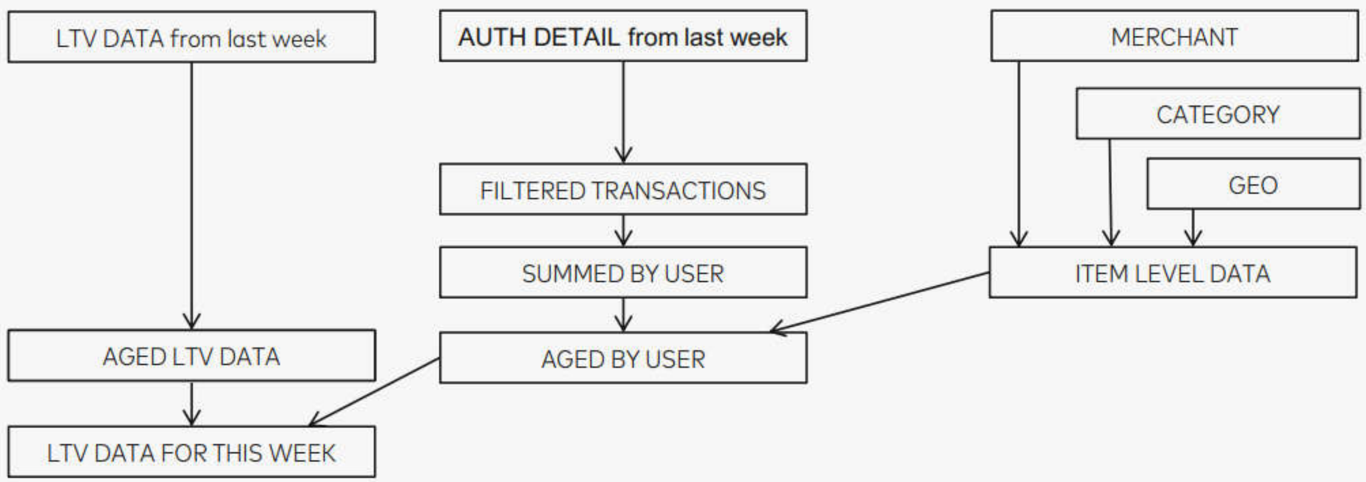
mastercard

intel

RETAIL: PREDICTING CUSTOMER LOYALTY

BigDL

Issues with Traditional ML : Feature Engineering Bottlenecks



Bottlenecks

- Need to pre-calculate 126 Long Term Variables for each user, such as total spends /visits for merchants list, category list divided by week, months and years
- The computation time for LTV features took 70% of the data processing time for the whole lifecycle and occupied lots of resources which had huge impact to other critical workloads.
- Miss the feature selection optimizations which could save the data engineering efforts a lot

mastercard

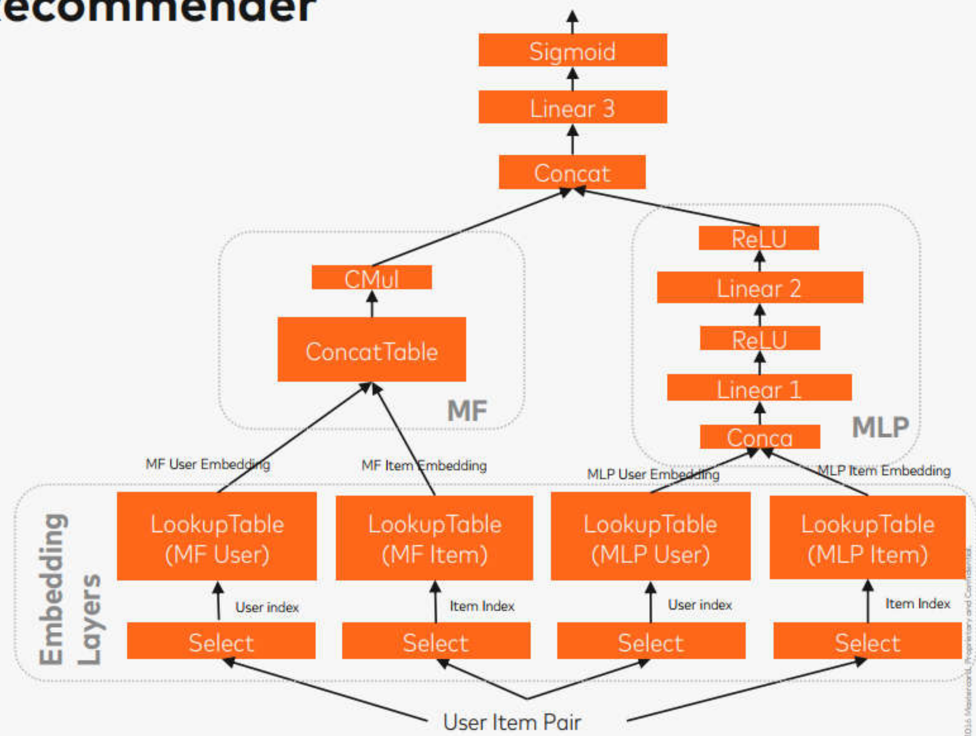
intel

RETAIL: PREDICTING CUSTOMER LOYALTY

BigDL

Approach: Neural Recommender

- **NCF**
- Scenario： Neural Collaborative Filtering ,recommend products to customers (priority is to recommend to active users) according to customers' past history activities.
- https://www.comp.nus.edu.sg/~xiangnan/papers/ncf.pdf
- **Wide & Deep learning**
- Scenario: jointly trained wide linear models and deep neural networks---to combine the benefits of memorization and generalization for recommender systems.
- https://pdfs.semanticscholar.org/aa9d/39e938c84a867ddf2a8cabc575ffba27b721.pdf



RETAIL: PREDICTING CUSTOMER LOYALTY

BigDL

Benchmark results (> 100 rounds)

Mlib AIS

AUROC: A
AUPRCs: B
recall: C
precision: D
20 precision: E

```
Parameters :
MaxIter(100)
RegParam(0.01)
Rank(200)
Alpha(0.01)
```

BigDL NCF

AUROC: A+23%
AUPRCs: B+31%
recall: C+18%
precision: D+47%
20 precision: E+51%

```
Parameters :
MaxEpoch(10)
learningRate(3e-2)
learningRateDecay(3e-7)
uOutput(100)
mOutput(200)
batchSize(1.6 M)
```

BigDL WAD

AUROC: A+20% (3 % down)
AUPRCs: B+30% (1% down)
recall: C+12% (4 % down)
precision: D+49% (2 % up)
20 precision: E+54% (3% up)

```
Parameters :
MaxEpoch(10)
learningRate(1e-2)
learningRateDecay(1e-7)
uOutput(100)
mOutput(200)
batchSize(0.6 M)
```

mastercard

intel

COUNTER-EXAMPLE: WHEN *NOT* TO USE ML

BigDL

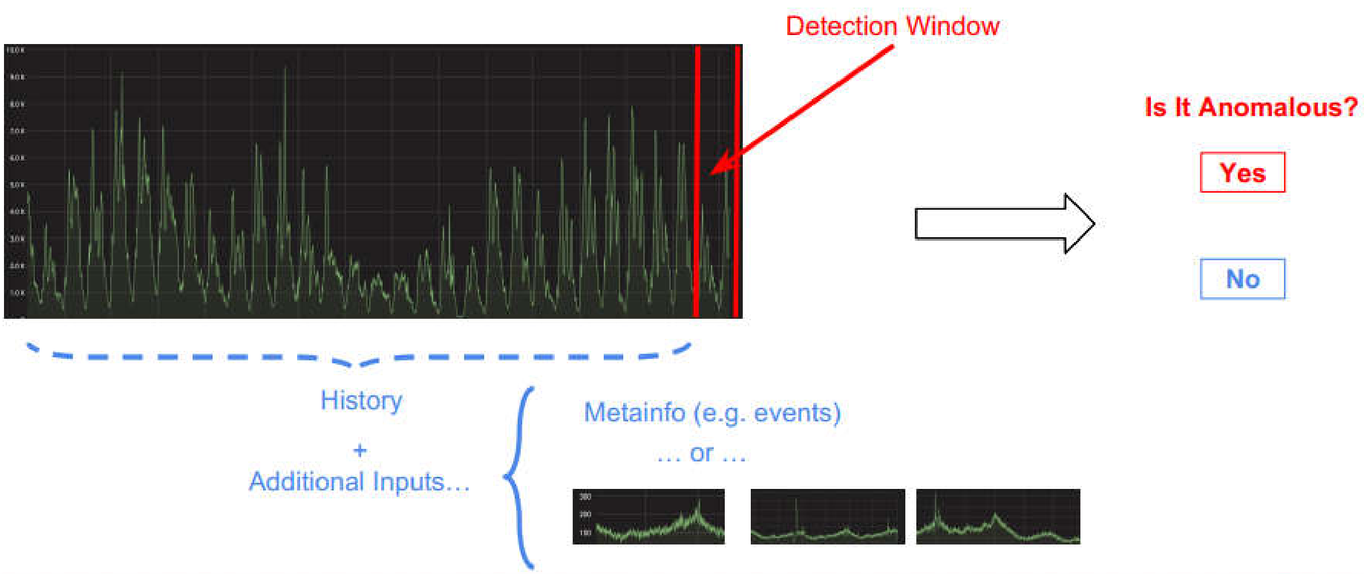
Business Need: Given monthly real-estate statistics, generate per-market area customizable Realtor Newsletters?

intel

ANOMALY DETECTION WITH RNN-LSTM

Anomaly Detection Platform

- At the core, the platform implements a stream of binary classifiers

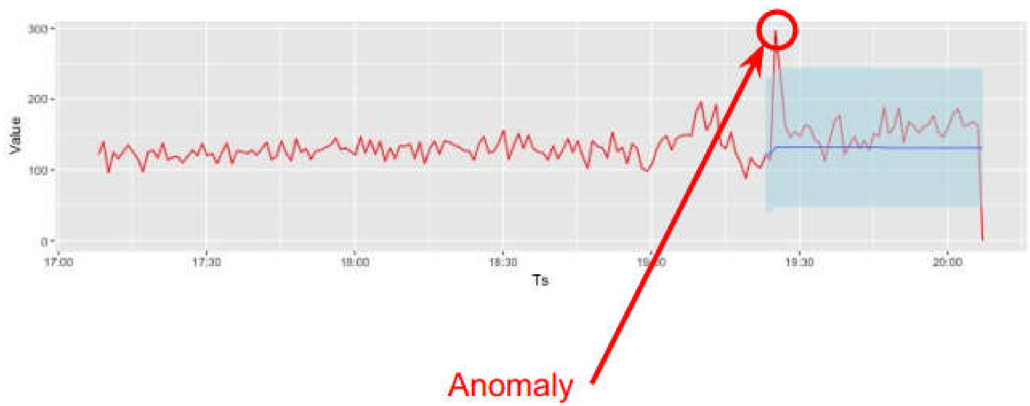


ANOMALY DETECTION WITH RNN-LSTM

BigDL

Anomaly Detection Stack

- But most carry out a density forecast behind the scenes



- Learn from the past
- Forecast our expectations…
- … and our uncertainty
- Compare with the actuals

ANOMALY DETECTION WITH RNN-LSTM

BigDL

Forecasting with Neural Networks

Use recurrent neural network forecasting

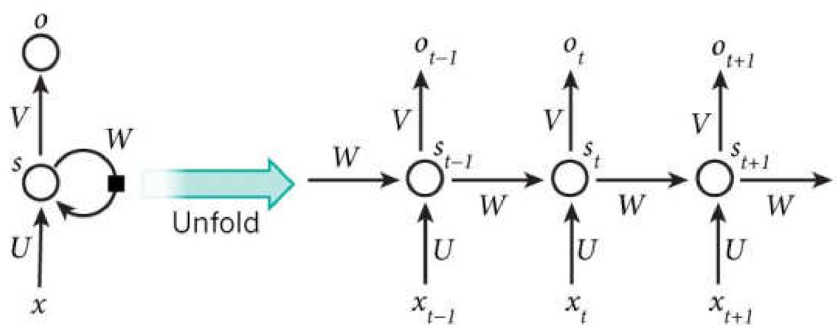
- Capable of dealing with huge amounts of data
- Has some memory of the past
- Not just univariate, could make use of other features
- Neural network could adopt many model shapes

ANOMALY DETECTION WITH RNN-LSTM

BigDL

Recurrent Neural Networks

- Inputs are sequential
  - Apply to cases like language processing, time series, etc
- Model has some memory of the past
  - Remember previous look-back steps



x_t : input

s_t : hidden state,

which is usually function of x_t and s_{t-1}

o_t : output

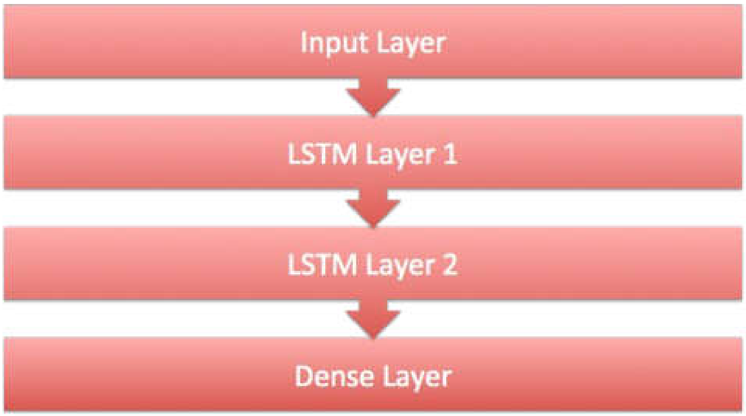
Plots from: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

ANOMALY DETECTION WITH RNN-LSTM

Forecasting with Recurrent Neural Networks

Model

- Two LSTM layers and one dense layer
- Window-wide scaling of input and output
- Adam optimization
- Minimizing absolute error instead of squared error
- Decaying learning rate

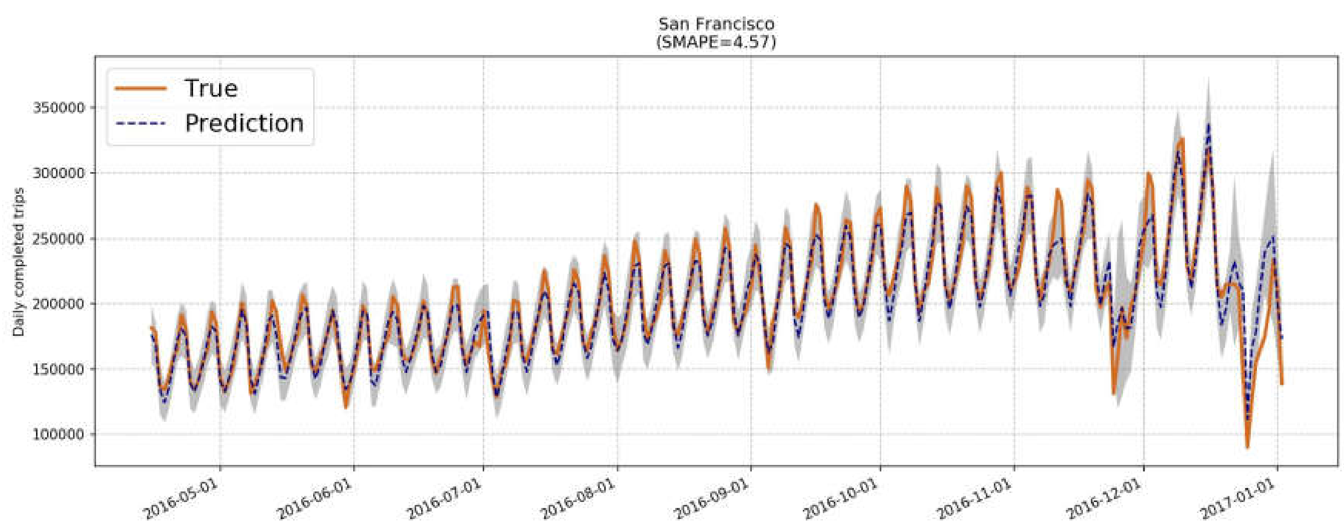


ANOMALY DETECTION WITH RNN-LSTM

BigDL

Forecasting Daily Trips with Uncertainty

Prediction with 95% prediction interval



Uber Blog: Engineering Uncertainty Estimation in Neural Networks for Time Series Prediction at Uber intel®

Software