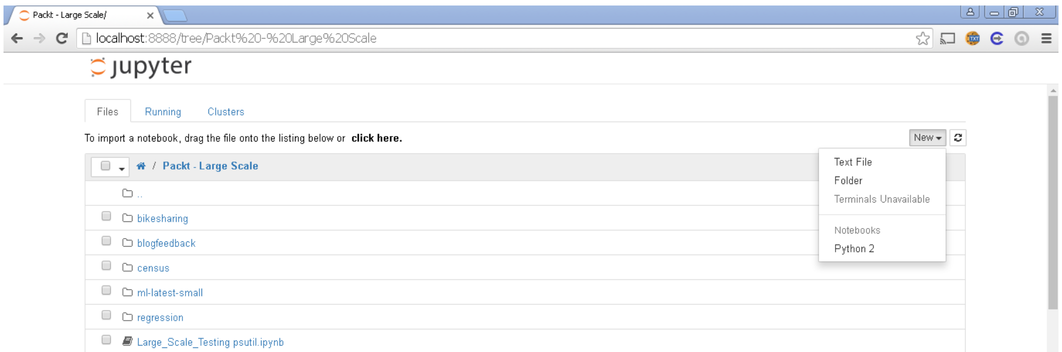
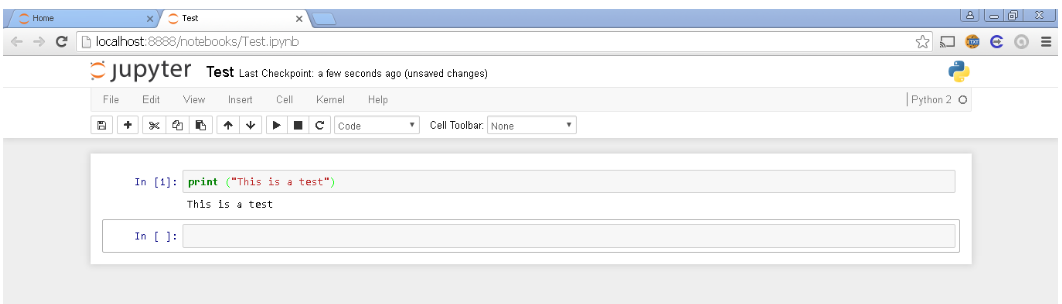
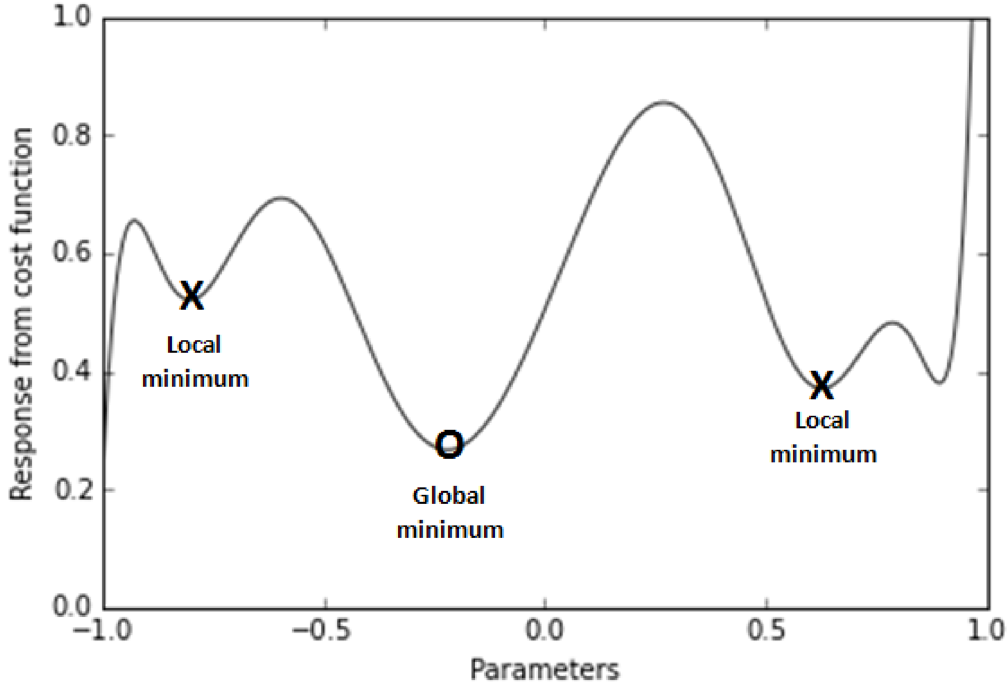
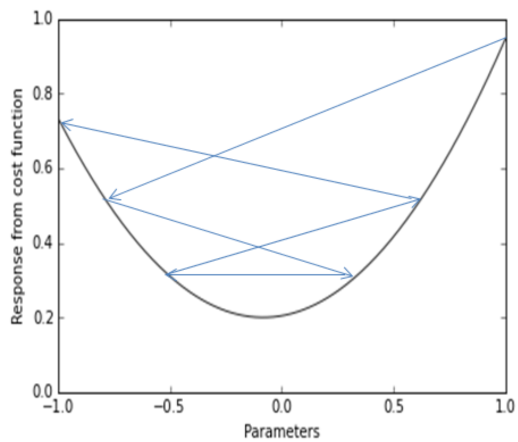
Chapter 1, First Steps to Scalability

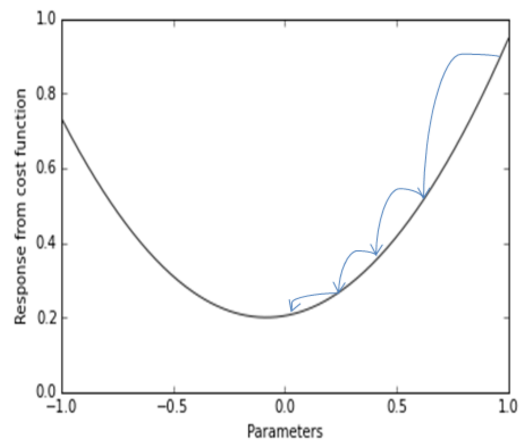




Chapter 2, Scalable Learning in Scikit-learn







$$J(w) = \frac{1}{2n}\sum(Xw - y)^2$$

$$w_j = w_j - \alpha\frac{\partial}{\partial w}J(w)$$

$$w_j = w_j - \alpha\frac{1}{n}\sum(Xw - y)x_j$$

$$w_j = w_j - \alpha(x_j w - y)x_j$$

$$y \approx h(X) = \beta X + \beta_0$$

$$\frac{1}{2n} * \sum(h(X) - y)^2$$

$$y \approx h(X) = \frac{1}{1 + e^{\beta X + \beta_0}}$$

$$-\frac{1}{n} * \sum[y * \ln(h(X)) + (1 - y) * \ln(1 - h(X))]$$

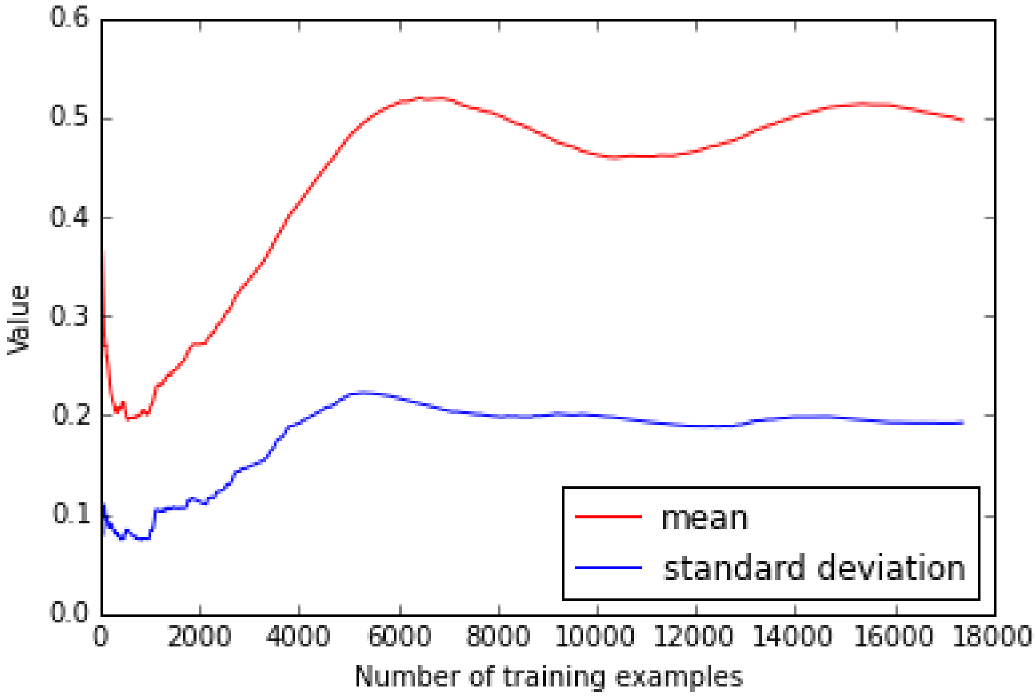
$$\eta$$

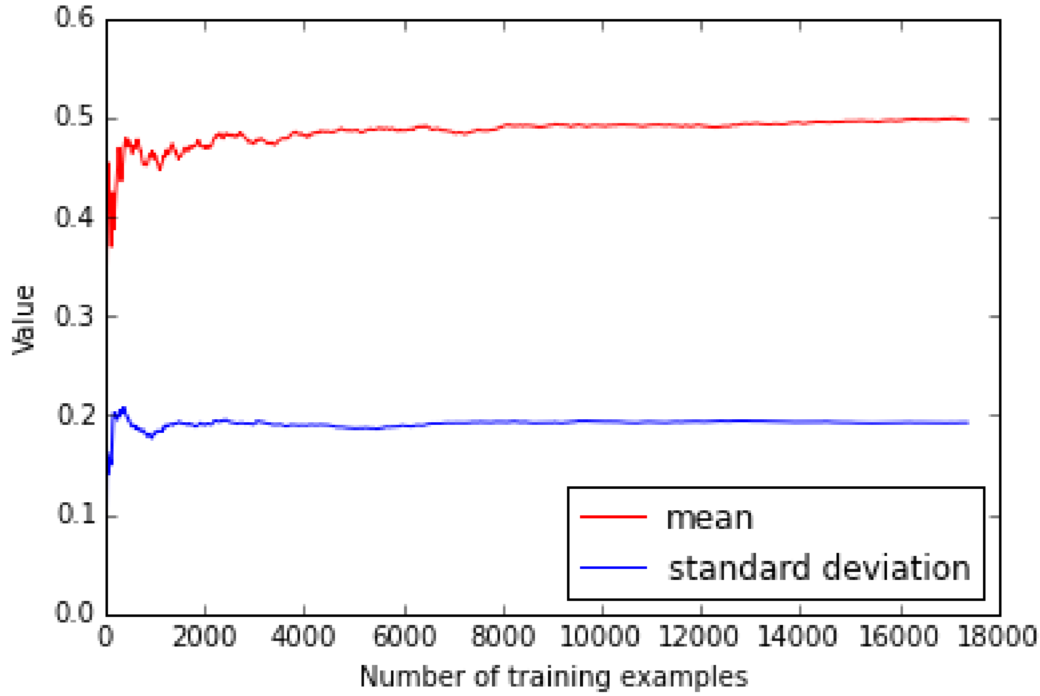
$$\eta^t = \frac{1}{\alpha_{t0} + \alpha_t}$$

$$\eta^t = \frac{eta_0}{t^{power\_t}}$$

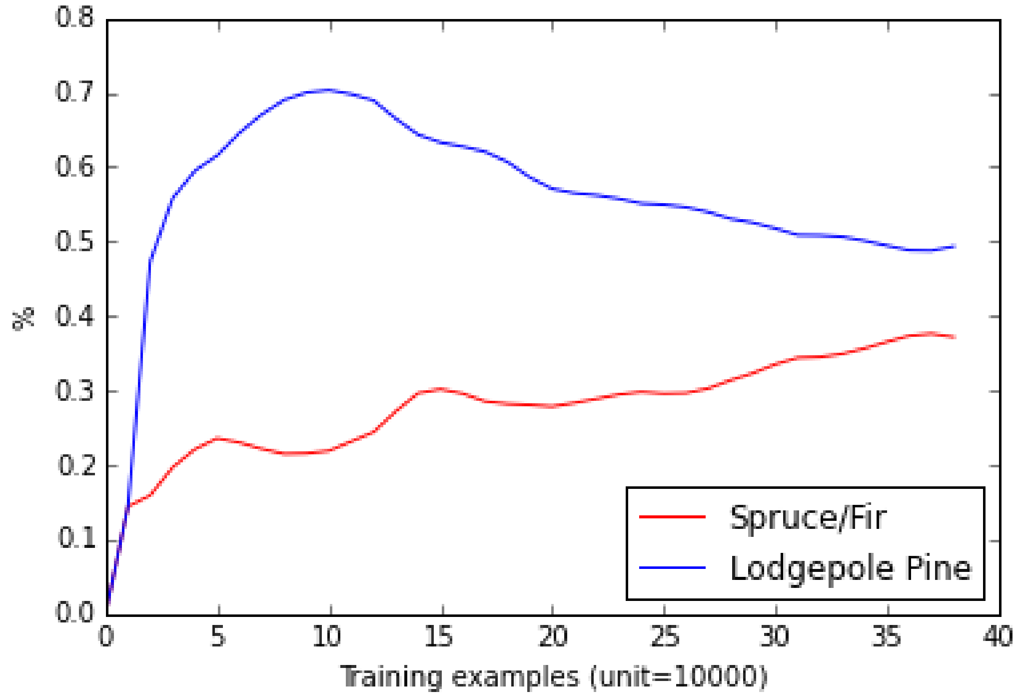
$$\sigma^2 = \frac{1}{n}\sum(x-\mu)^2$$

$$\sigma^2 = \frac{1}{n}\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)$$


0.6
0.5
0.4
0.3
0.2
0.1
0.0
Value
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
Number of training examples
mean
standard deviation



0.6
0.5
0.4
0.3
0.2
0.1
0.0
Value
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
Number of training examples
mean
standard deviation



0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
%
0
5
10
15
20
25
30
35
40
Training examples (unit=10000)
Spruce/Fir
Lodgepole Pine

Chapter 3, Fast-Learning SVMs

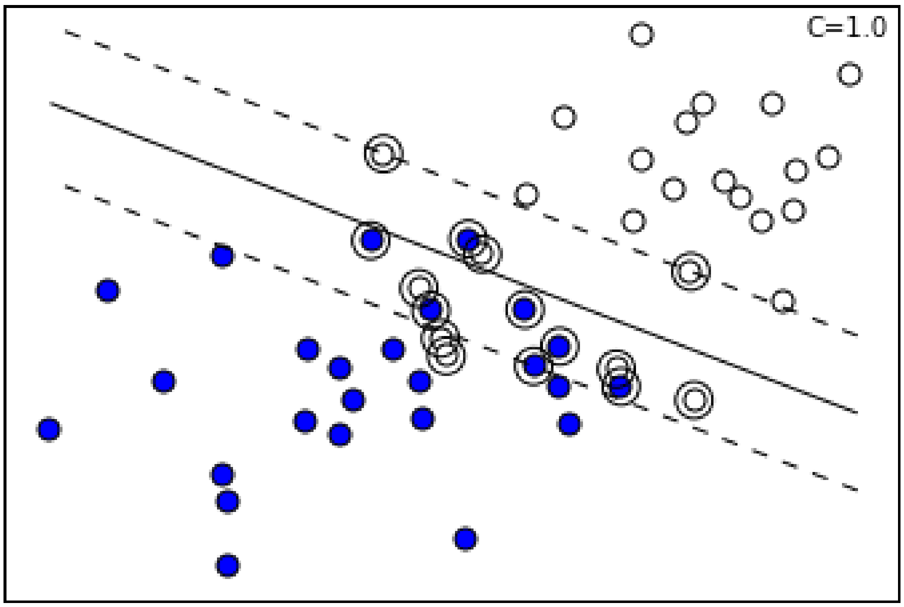
$$\frac{\lambda}{2}||w^2|| + \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y(wX + b)\right)\right]$$

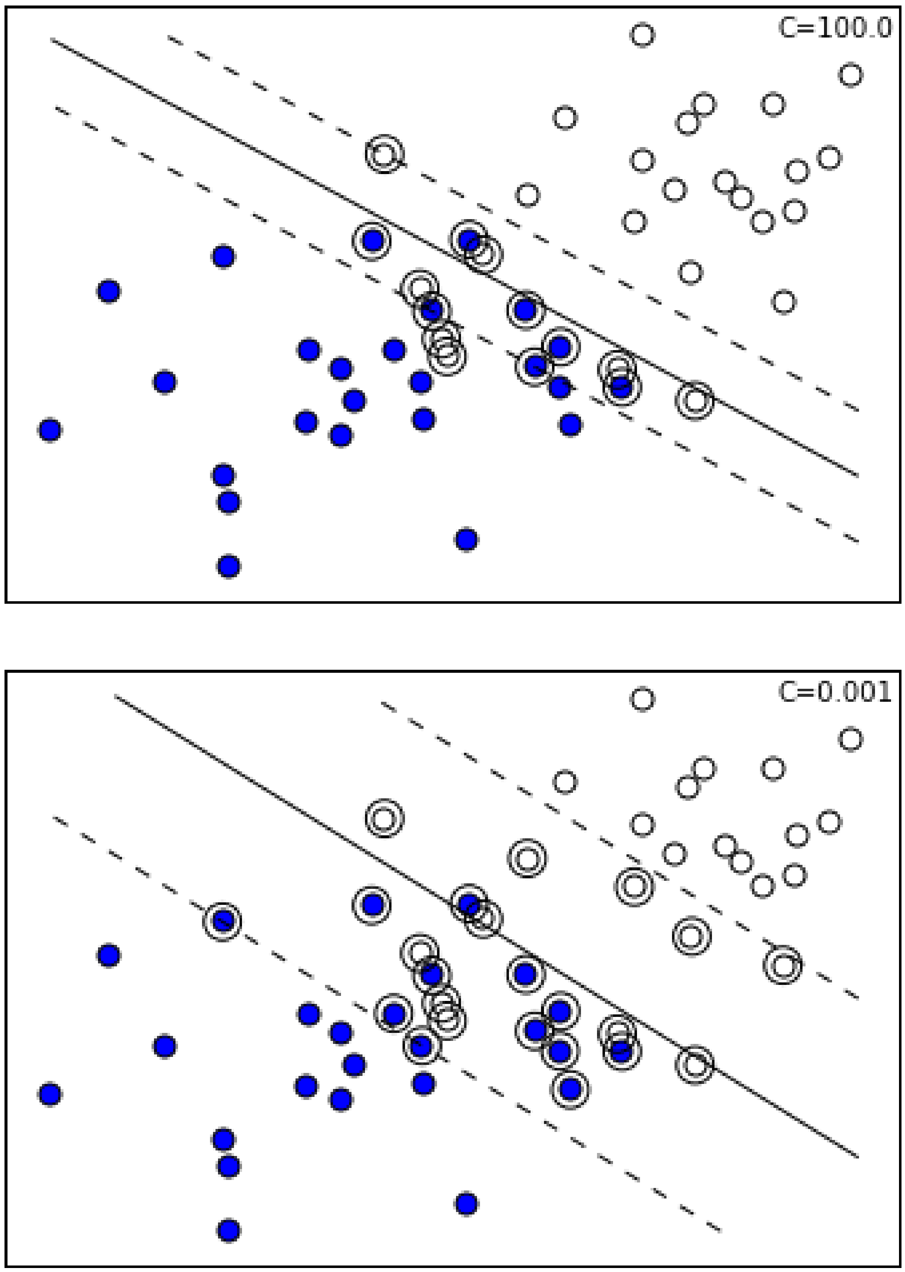
$$\frac{\lambda}{2}||w^2||$$

$$\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y(wX + b)\right)$$

$$\frac{1}{2}||w^2|| + C\left[\sum_{i=1}^{n} \max\left(0, 1 - y(wX + b)\right)\right]$$

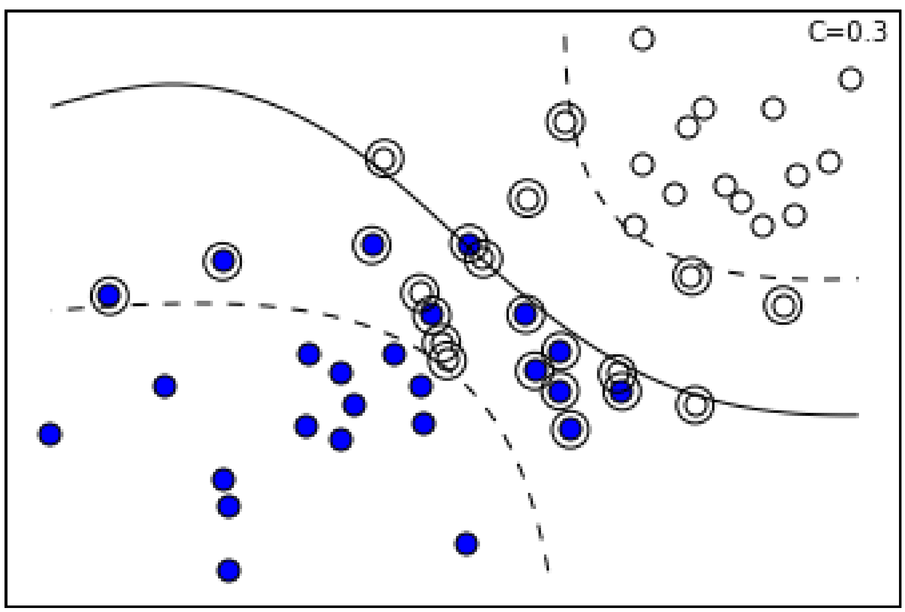
$$\lambda = \frac{1}{nC}$$

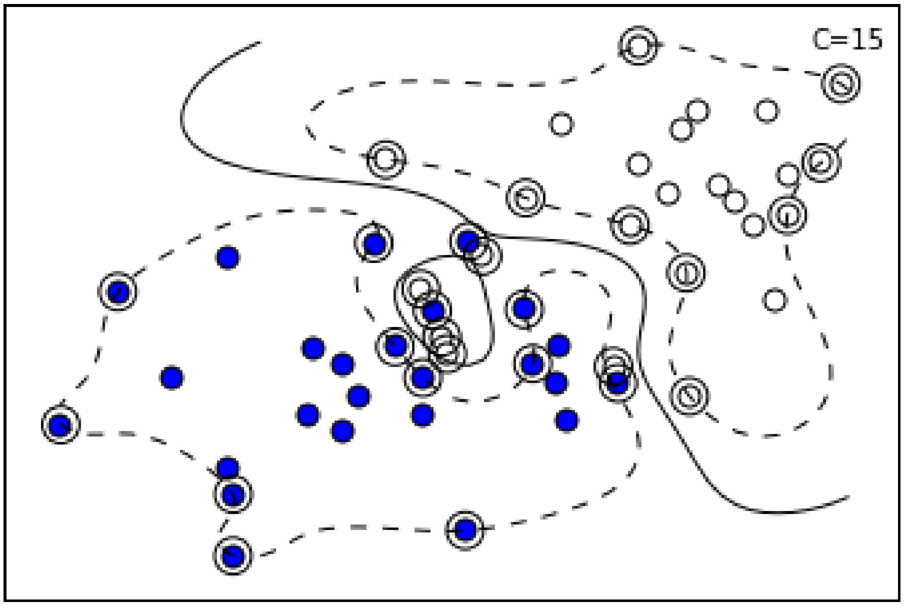



C=100.0
C=0.001

$$K(x_i, x) = \exp\left(-\|x_i - x\|^2 / 2\sigma\right)$$

$$f(x) = \sum_{i=1}^{n} \alpha_i \, y_i K(x_i, x) + b$$





$$k(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$$

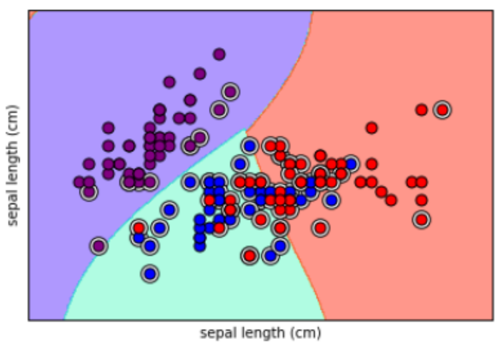
$$k(x_i, x_j) = \tanh\left(\gamma \langle x_i x_j \rangle^2 + r\right)$$

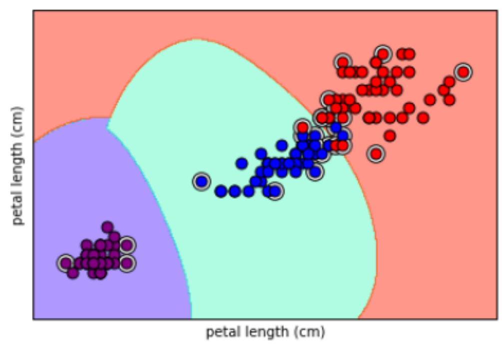
$$\sum_{j=1}^{m} \beta_j^2 + C \sum_{i=1}^{n} \mathrm{L}_\epsilon\left(y_i - \hat{y}_i\right)$$

$$loss(y, \hat{y}) = \max(0, 1 - y\hat{y})$$

$$\hat{y} = wX + b$$

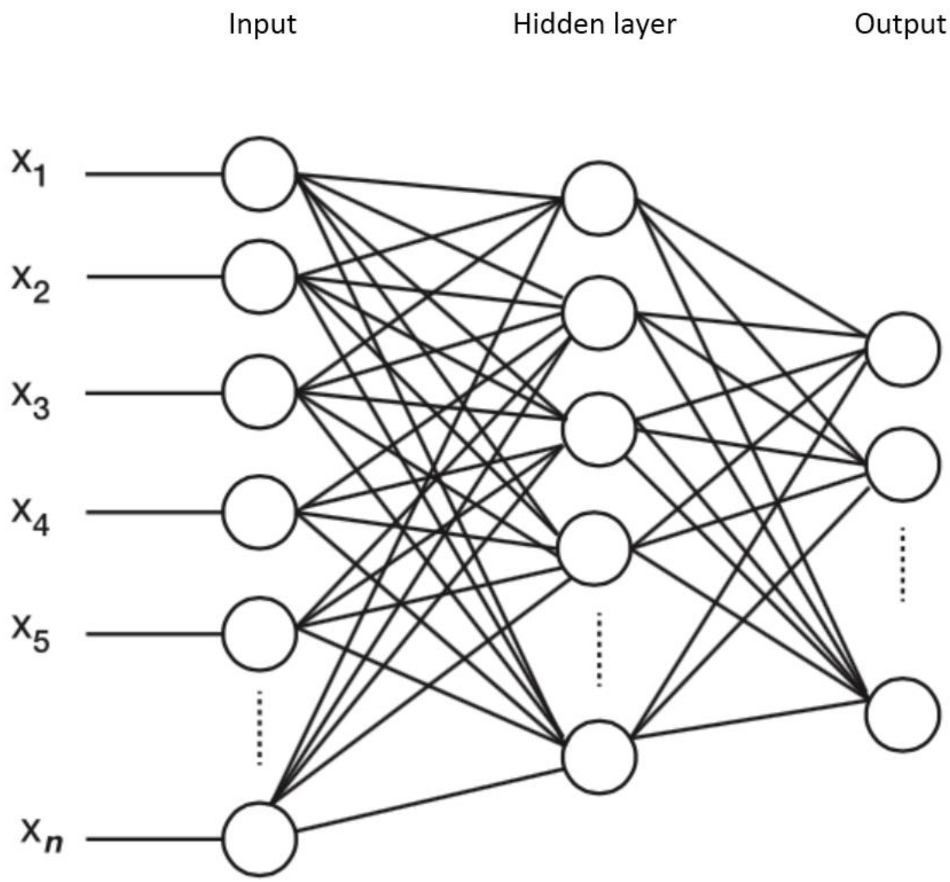
$$L2\_loss(y, \hat{y}) = \max(0, 1 - y\hat{y})^2$$

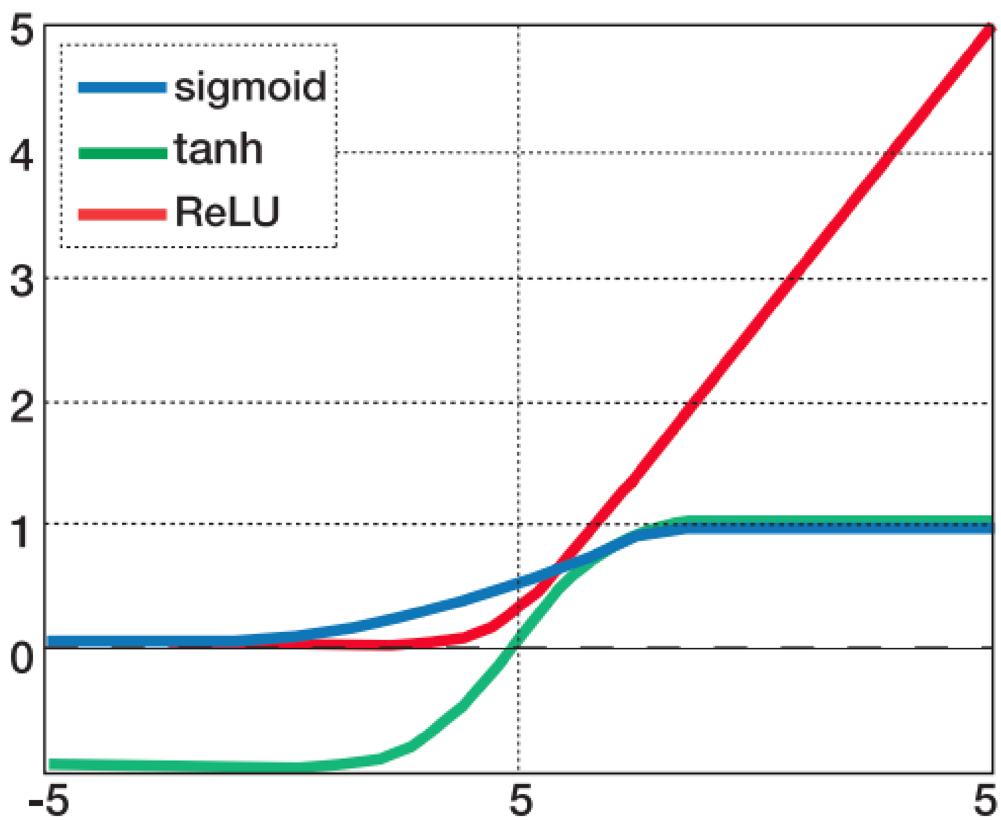




$$\lambda_1 \|w\|_1 + \frac{\lambda_2}{2} \|w\|_2^2 + \sum_{i=1}^{n} \text{loss}(x_i, y_i, w)$$

Chapter 4, Neural Networks and Deep Learning



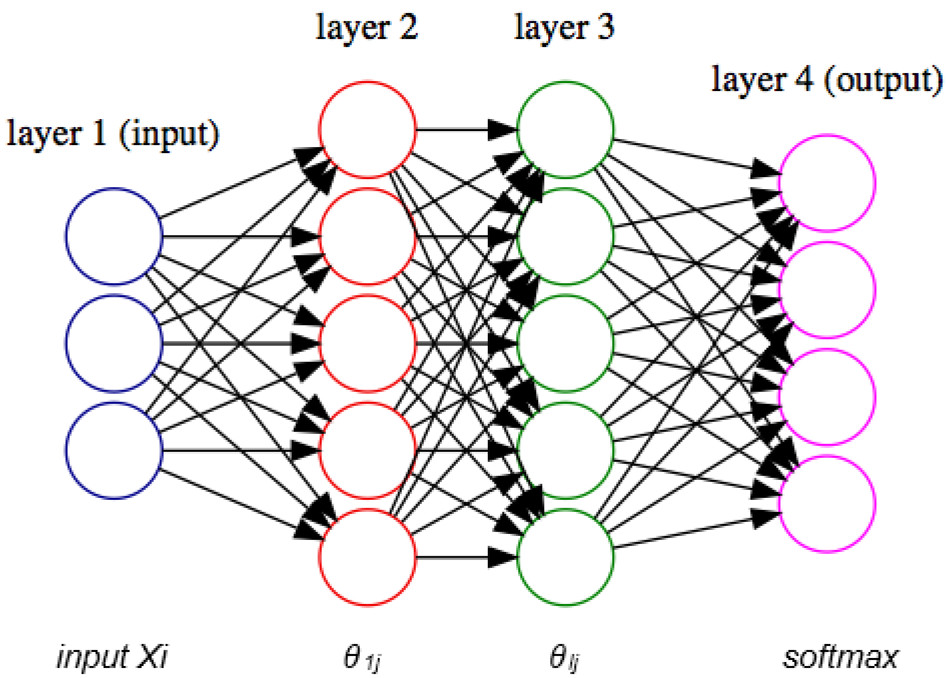


sigmoid	$\frac{1}{(1 + e^{-x})}$	Active range: [sqrt(3), sqrt(3)] Output range: (0, 1)

tanh function	$\frac{e^{t} - e^{-t}}{e^{t} + e^{-t}}$	Active range: [-2,2] Output range: (-1,+1)

rectified linear unit (ReLU)	$f(x) = \max(0, x)$	Active range: [0, inf]

$$softmax(k, x_1, \ldots, x_n) = \frac{e^{xk}}{\sum_{i=1}^{n} e^{xi}}$$



$$\theta$$

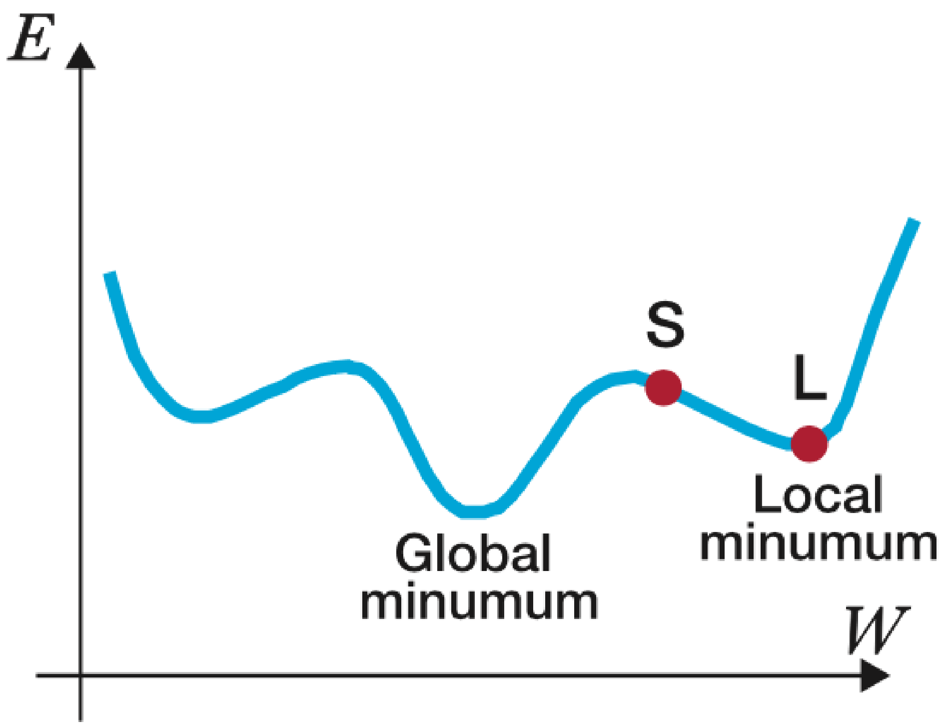
$$z_{(2)} = \theta_{(1)} x + b_{(1)}$$

$a_{(2)} = f(z_{(2)})$ *transformation after layer 1*

$$z_{(3)} = \theta_{(2)} a_{(2)} + b_{(2)}$$

$h_{W,b}(x) = a_{(3)} = f_{(softmax)}(z_{(3)})$ *final output with softmax transformation*

$$\theta ij := \theta ij - \boldsymbol{\eta} * \Delta_{\theta} J(\theta ij)$$



$$v_{t+1} = \mu v_t - \eta \nabla \mathcal{L}(\theta_t)$$
$$\theta_{t+1} = \theta_t + v_{t+1}$$

$$\eta$$

$$g_{t+1} = g_t + \nabla\mathcal{L}(\theta_t)^2$$

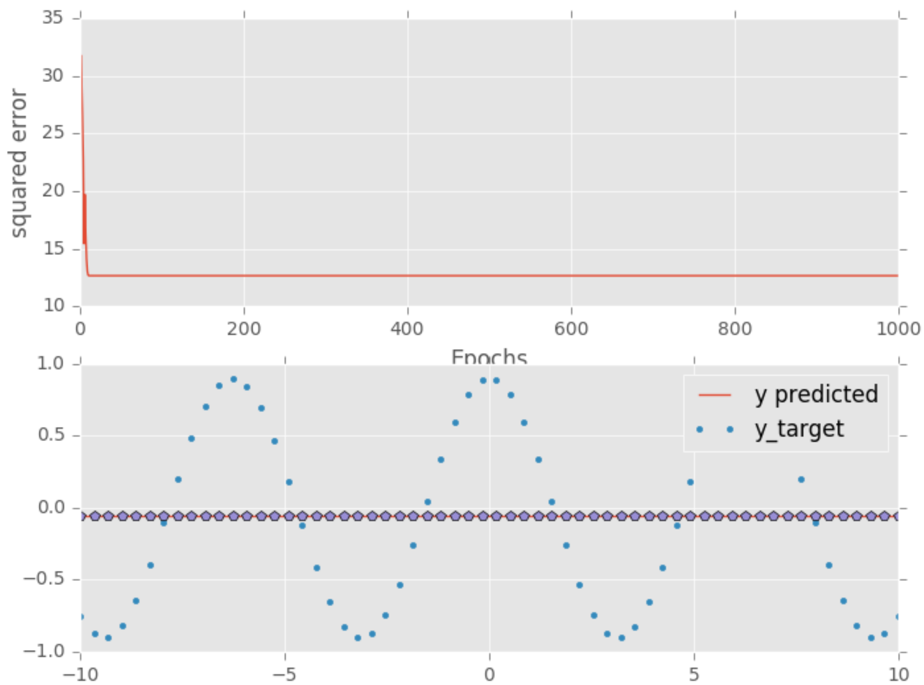
$$\theta_{t+1} = \theta_t - \frac{\eta\nabla\mathcal{L}(\theta_t)}{\sqrt{g_{t+1} + \in}}$$

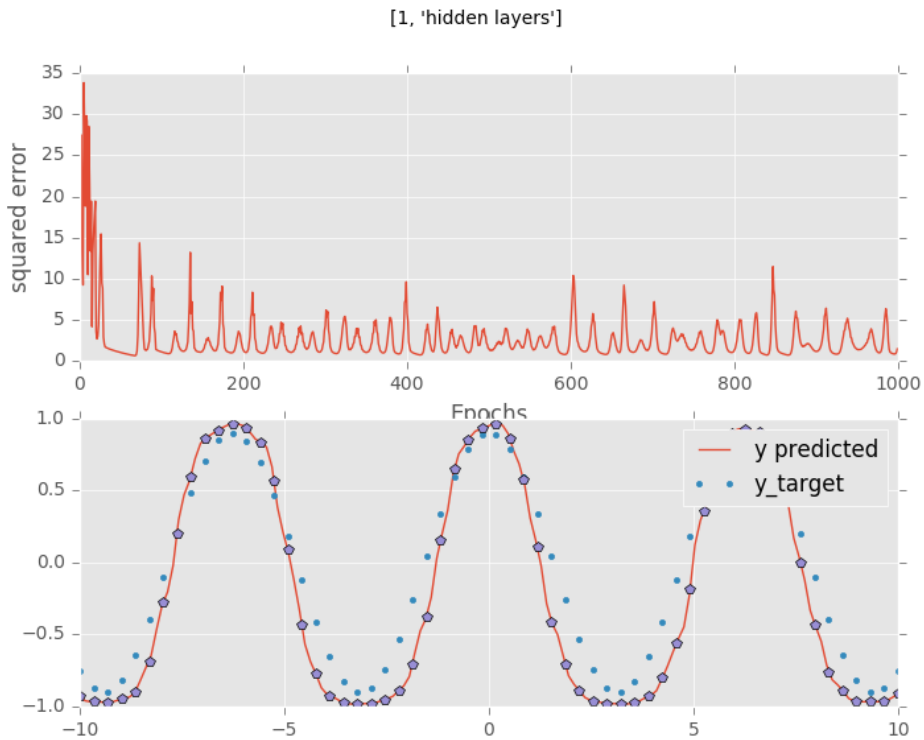
$$\Delta_{ij}^{(t)} = \begin{cases} n^+ * \Delta_{ij}^{(t-1)}, if \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ n^- * \Delta_{ij}^{(t-1)}, if \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, else \end{cases}$$

$$where\, 0 < \eta^- < 1 < \eta^+$$

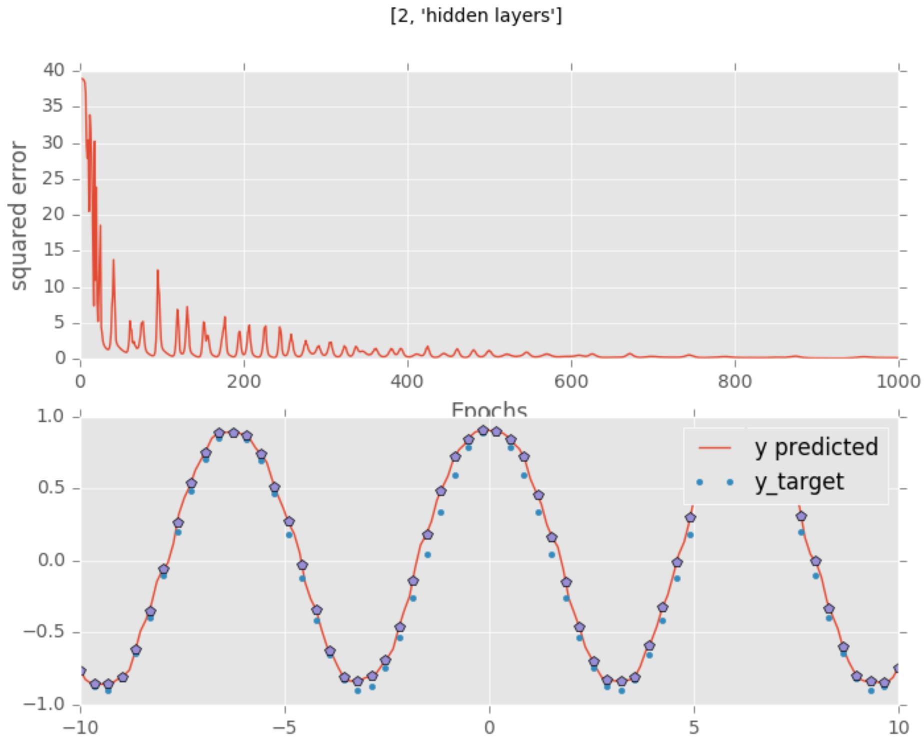
$$\theta_{t+1} = \theta t - \frac{\eta}{\sqrt{E[g2]t + e}} g_t$$

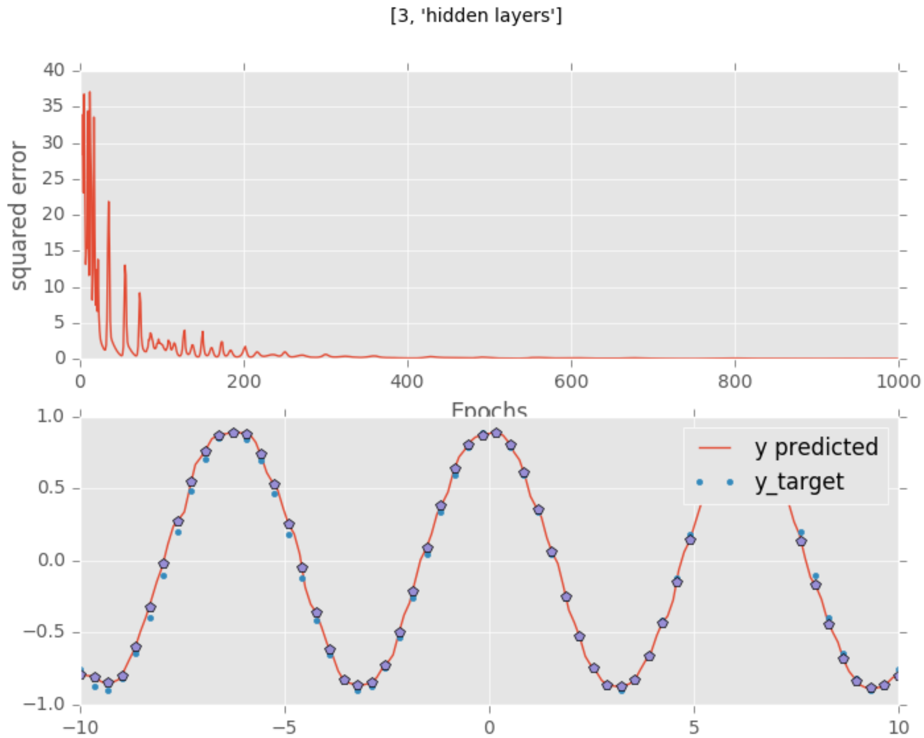
[0, 'hidden layers']

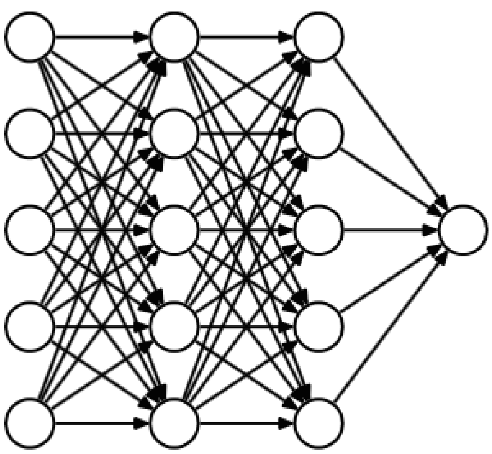


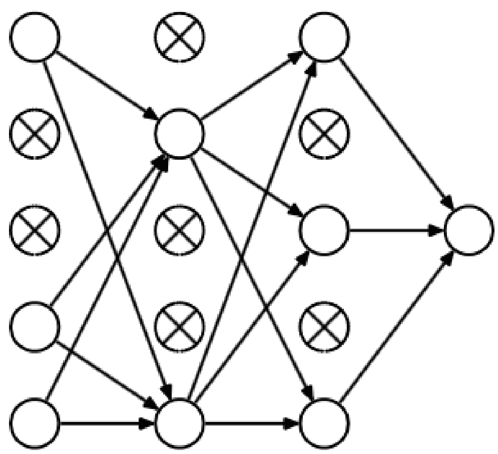


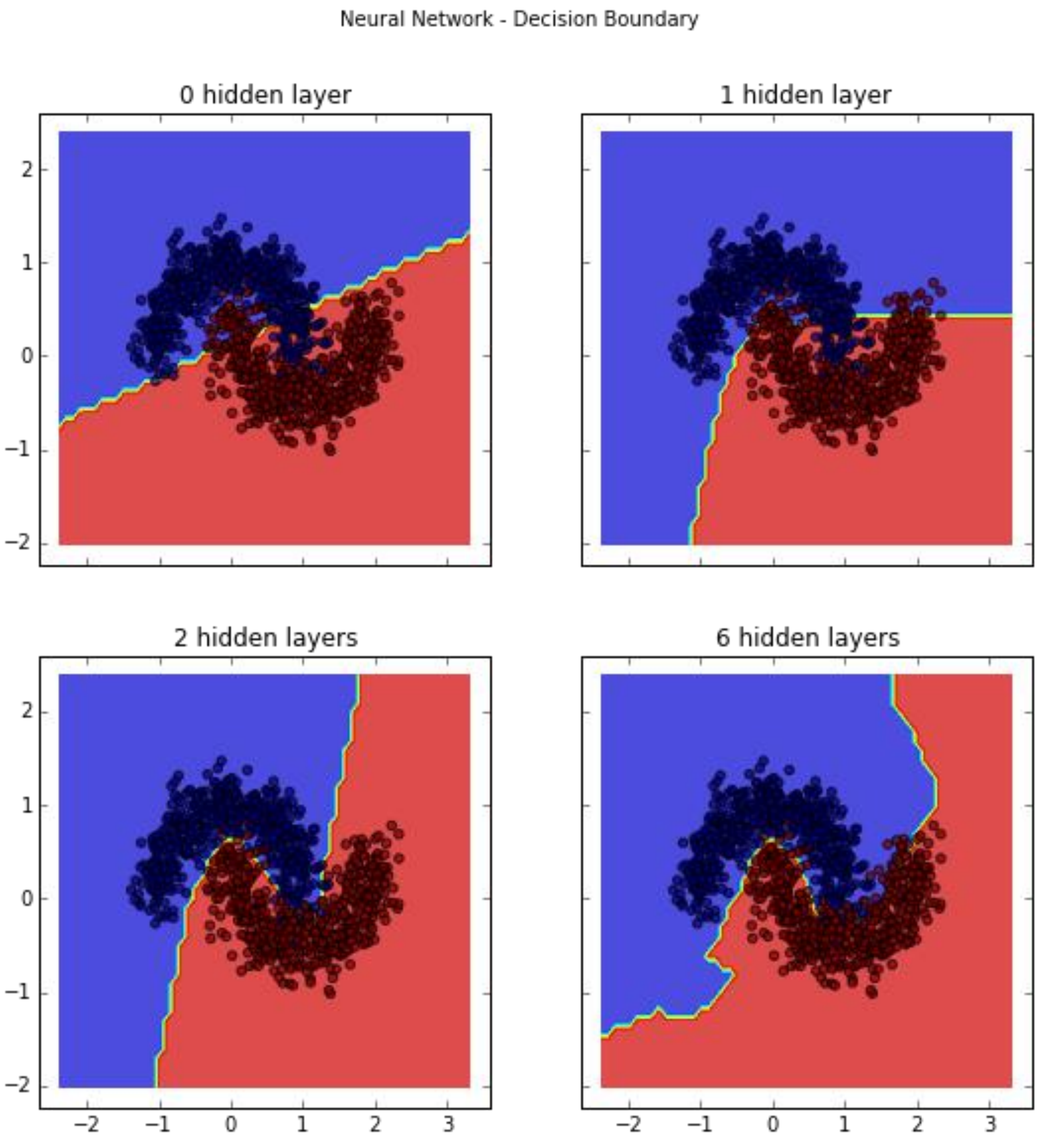
[1, 'hidden layers']
squared error
Epochs
y predicted
y_target


[2, 'hidden layers']
squared error
Epochs
y predicted
y_target


[3, 'hidden layers']
squared error
Epochs
y predicted
y_target






Neural Network - Decision Boundary
0 hidden layer
1 hidden layer
2 hidden layers
6 hidden layers
2
1
0
−1
−2
−2
−1
0
1
2
3

```
Model Details
=============
H2ODeepLearningEstimator :  Deep Learning
Model Key:  DeepLearning_model_python_1463889677812_3

Status of Neuron Layers: predicting C785, 10-class classification, multinomial distribution, CrossEntropy loss, 25,418
weights/biases, 371.3 KB, 300,525 training samples, mini-batch size 1
```

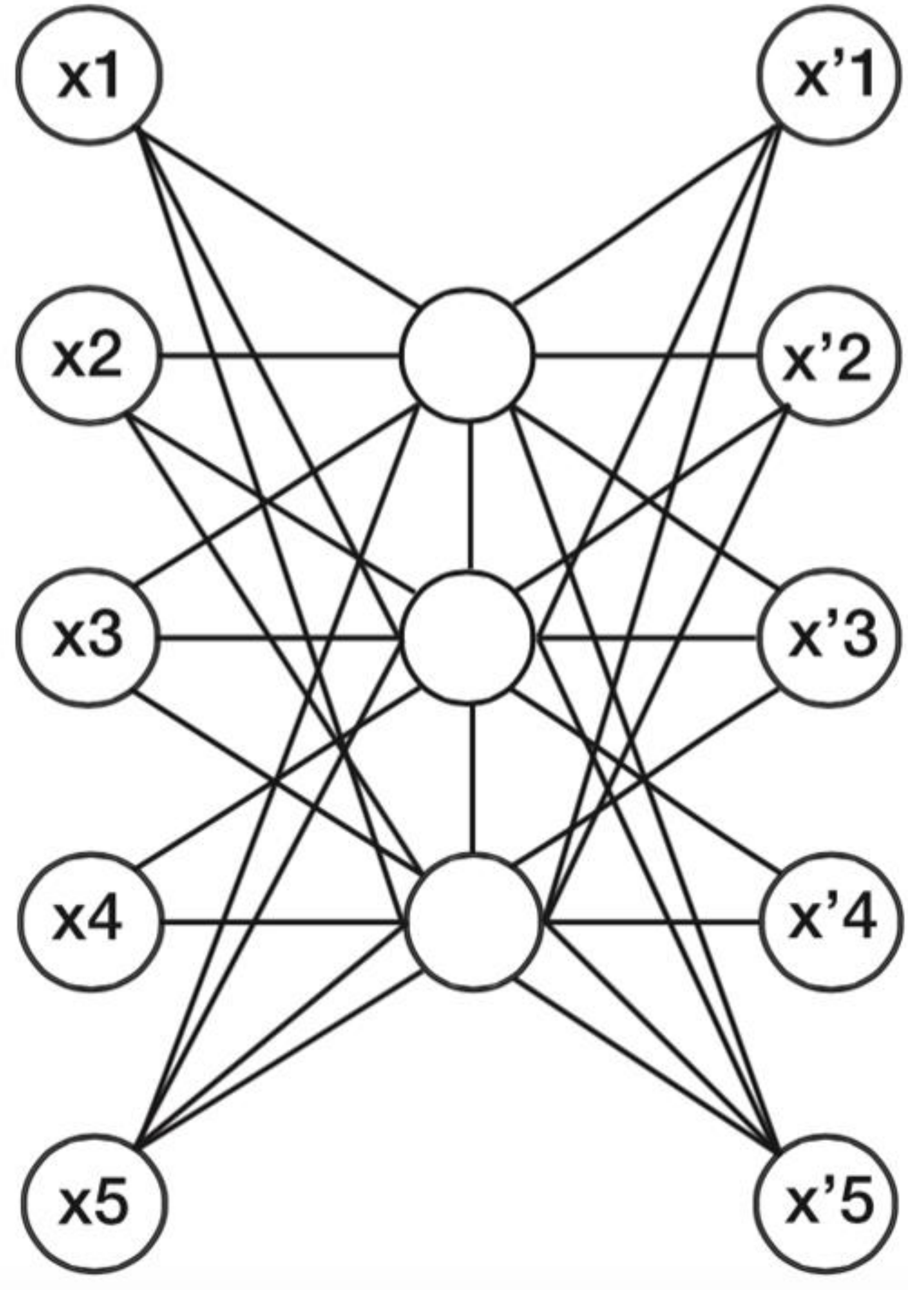
	layer	units	type	dropout	l1	l2	mean_rate	rate_RMS	momentum	mean_weight	weight_RMS	mean_bias	bias_RMS
	1	717	Input	20.0									
	2	32	RectifierDropout	50.0	0.0005	0.0	0.0370441	0.1916480	0.0	-0.0061157	0.0612413	0.4243763	0.0918573
	3	32	RectifierDropout	50.0	0.0005	0.0	0.0004112	0.0002142	0.0	-0.0279839	0.1946866	0.7527754	0.2369041
	4	32	RectifierDropout	50.0	0.0005	0.0	0.0006548	0.0002914	0.0	-0.0397208	0.2000279	0.6407341	0.3597416
	5	10	Softmax		0.0005	0.0	0.0025825	0.0024549	0.0	-0.2988227	0.8903637	-1.0314634	0.8309324

```
ModelMetricsMultinomial: deeplearning
** Reported on train data. **

MSE: 0.142497867237
R^2: 0.982924289006
LogLoss: 0.455262748035
```

```
             timestamp    duration  training_speed    epochs  iterations  \
0  2016-05-22 06:09:35   0.000 sec            None  0.000000           0
1  2016-05-22 06:09:36   3.161 sec  30039 rows/sec  0.500650           1
2  2016-05-22 06:09:41   8.279 sec  40768 rows/sec  4.008217           8
3  2016-05-22 06:09:43  10.002 sec  40360 rows/sec  5.008750          10

   samples  training_MSE  training_r2  training_logloss  \
0        0           NaN          NaN               NaN
1    30039      0.434316     0.947955          1.154869
2   240493      0.163368     0.980423          0.507394
3   300525      0.142498     0.982924          0.455263

   training_classification_error
0                            NaN
1                       0.327284
2                       0.114081
3                       0.096430
```

Scoring History:

ing_MSE	training_r2	training_logloss	training_classification_error	validation_MSE	validation_r2	validation_logloss	validation_classification_error
	nan	nan	nan	nan	nan	nan	nan
23902	0.9412354	1.2943441	0.3213827	0.4909360	0.9414521	1.2906389	0.3221
91473	0.9857803	0.4101554	0.0889877	0.1234898	0.9852729	0.4234574	0.0954



Input vector X weight vector *W* Output vector X'

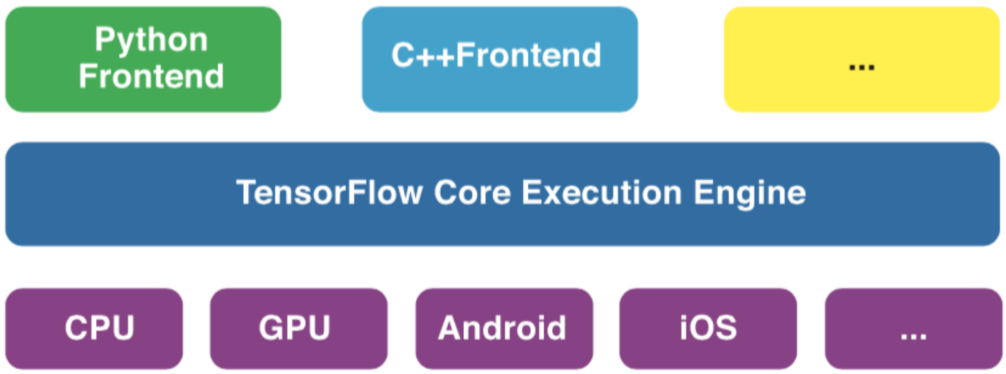
$$h_i=\text{sigmoid}((W_1.x*x)+b_1(i,1))$$

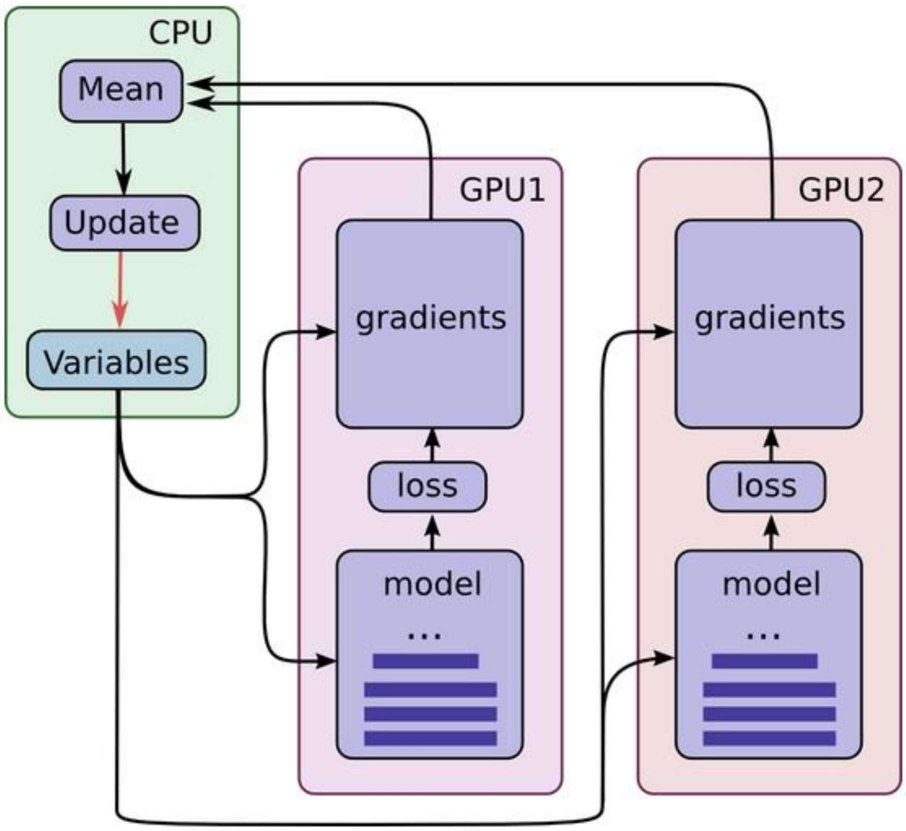
$$x'=\text{sigmoid}(W_2.x*h_i)+b_2(i,1)$$

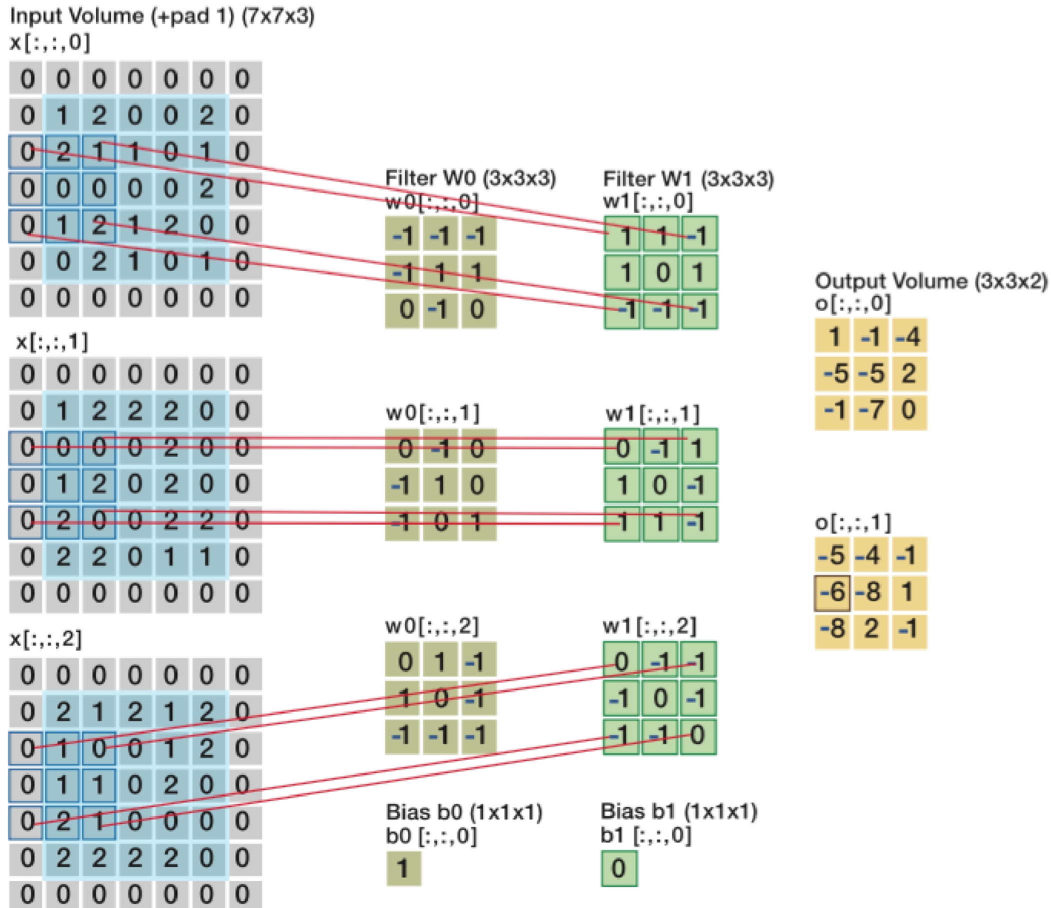
Cross entropy $L(x,y)=-\frac{1}{m}\sum_{i=l}^{m}[y_n\log\hat{y}_n+(1-y_n)\log(1-\hat{y}_n)]$

$$\hat{\rho}_j=\frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right]$$

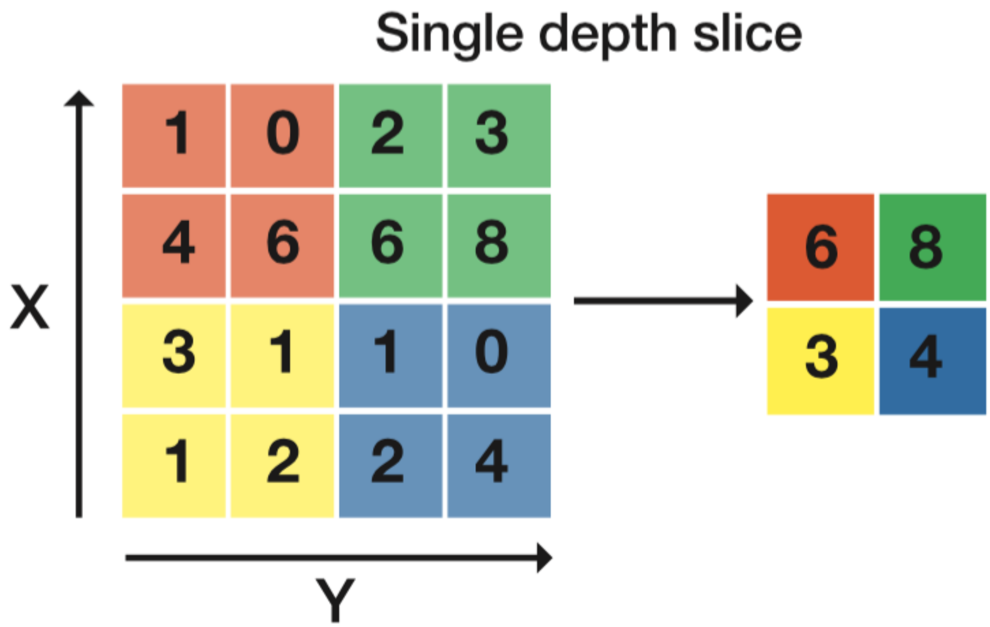
Chapter 5, Deep Learning with TensorFlow

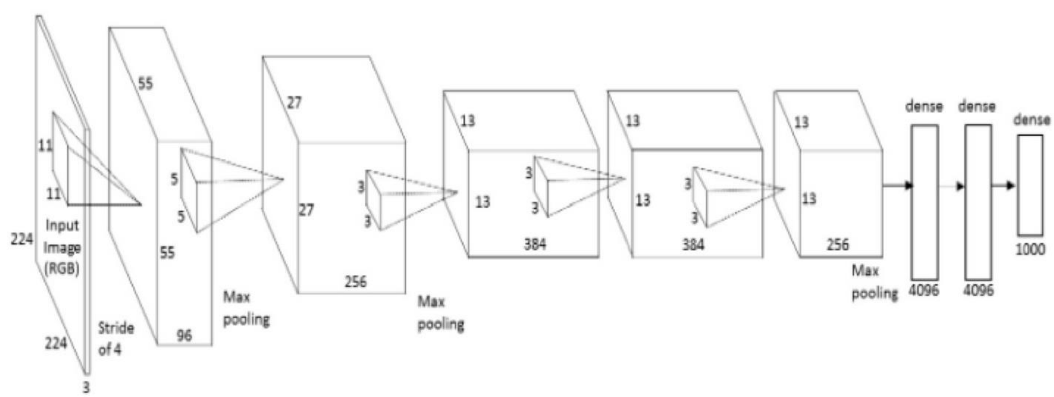


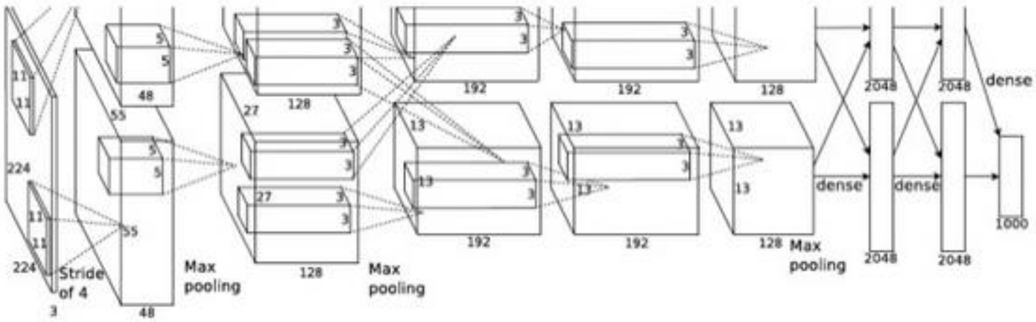

CPU
Mean
Update
Variables
GPU1
gradients
loss
model
...
GPU2
gradients
loss
model
...



Input Volume (+pad 1) (7x7x3)
x[:,:,0]
x[:,:,1]
x[:,:,2]
Filter W0 (3x3x3)
w0[:,:,0]
w0[:,:,1]
w0[:,:,2]
Filter W1 (3x3x3)
w1[:,:,0]
w1[:,:,1]
w1[:,:,2]
Output Volume (3x3x2)
o[:,:,0]
o[:,:,1]
Bias b0 (1x1x1)
b0 [:,:,0]
Bias b1 (1x1x1)
b1 [:,:,0]


Single depth slice
X
Y
1 0 2 3
4 6 6 8
3 1 1 0
1 2 2 4
6 8
3 4


224
224
3
11
11
Input Image (RGB)
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
13
13
384
13
13
256
Max pooling
dense
dense
dense
4096
4096
1000



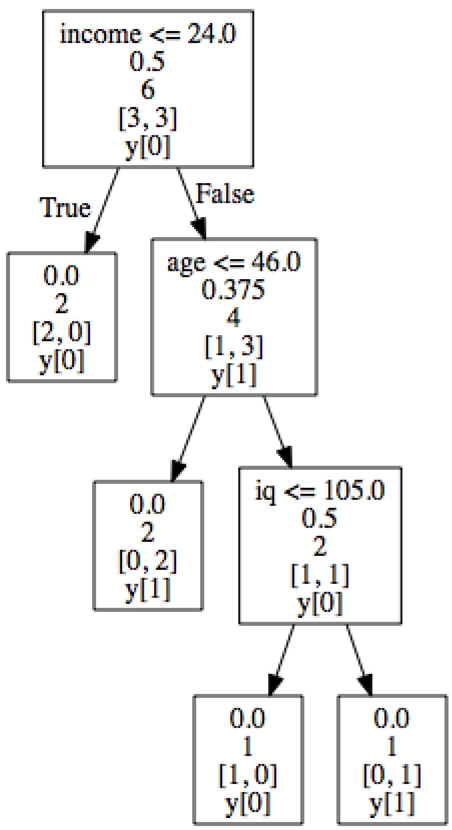
A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

```
Train on 50000 samples, validate on 10000 samples
Epoch 1/200
11552/50000 [=====>........................] - ETA: 1267s - loss: 1.8730 - acc: 0.3183
```

Chapter 6, Classification and Regression Trees at Scale

$$Gini(S) = 1 - \sum_{i=1}^{k} p_i^2$$

$$D = -\sum_{k=1}^{k} \hat{p}_{mk} \log \hat{p}_{mk}.$$



$$\phi$$

$$\hat{y} = \sum_{m=1}^{M} f_k\left(x_i\right), f_k \in \phi$$

$\hat{y}_i^{(0)}=0$

$\hat{y}_i^{(1)}= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$ *(this is our first tree)*

$\hat{y}_i^{(2)}=f_1(x_i)+f_2(x_i)=\hat{y}_i^{(1)} + f_2(x_1)$ *(our second tree added to the previous)*

And so on… till stopping criteria is reached

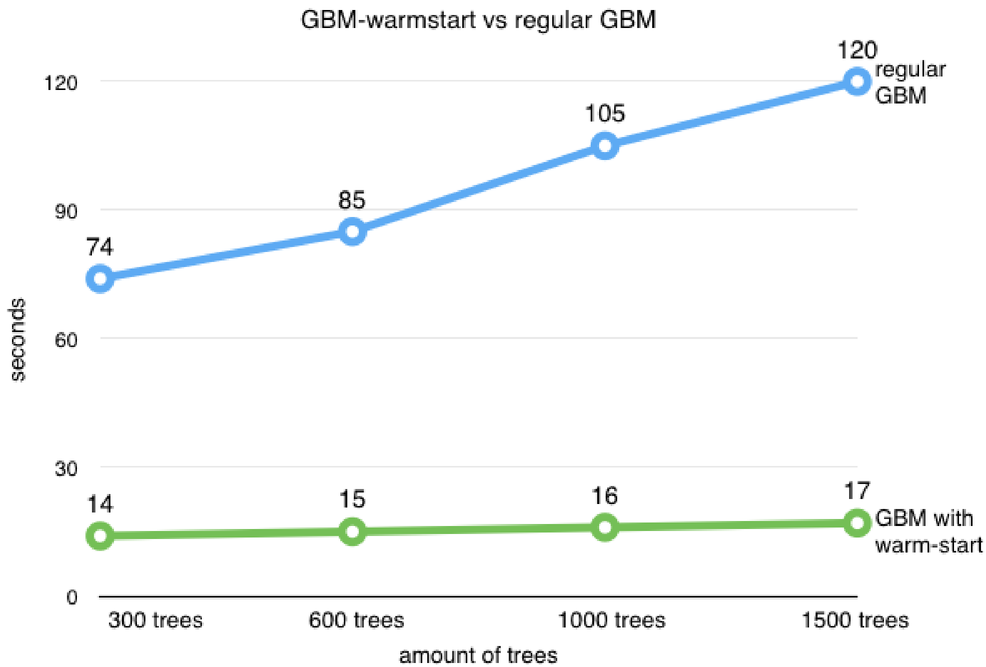
$$(\hat{y}_i^{(t-1)}) + f_1(x_i)$$

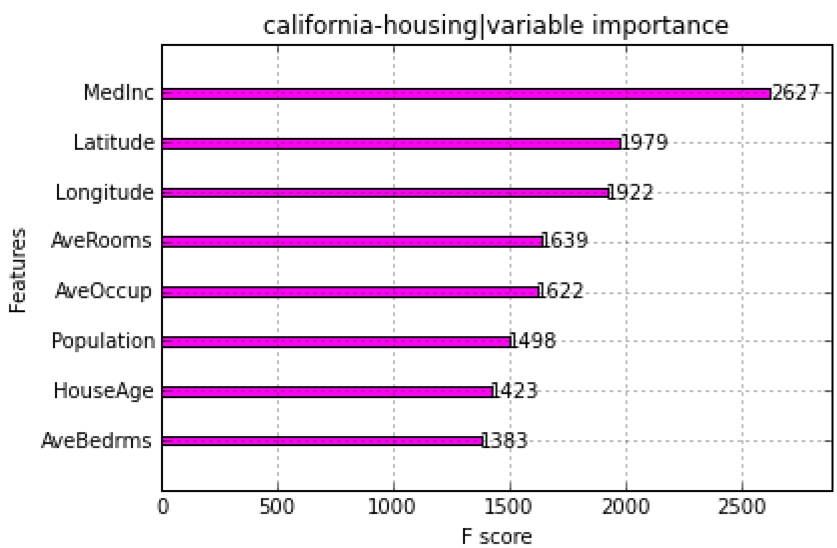
$$\hat{y}_i^{(t)} = \sum_{m=1}^{M} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + f_i\left(x_i\right)$$

$$\lambda$$

$$\gamma_{mi}$$

$$\hat{y}_i^{(t)} = \sum_{m=1}^{M} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + \lambda \gamma_{mi} f_i\left(x_i\right)$$


GBM-warmstart vs regular GBM
seconds
120
90
60
30
0
74
85
105
120
regular GBM
14
15
16
17
GBM with warm-start
300 trees
600 trees
1000 trees
1500 trees
amount of trees


california-housing|variable importance
Features
MedInc
Latitude
Longitude
AveRooms
AveOccup
Population
HouseAge
AveBedrms
2627
1979
1922
1639
1622
1498
1423
1383
0
500
1000
1500
2000
2500
F score

H2O cluster uptime:	1 days 1 hours 33 minutes 47 seconds 112 milliseconds
H2O cluster version:	3.6.0.8
H2O cluster name:	H2O_started_from_python
H2O cluster total nodes:	1
H2O cluster total memory:	1.78 GB
H2O cluster total cores:	4
H2O cluster allowed cores:	4
H2O cluster healthy:	True
H2O Connection ip:	127.0.0.1
H2O Connection port:	54321

```
drf Grid Build Progress: [##################################################] 100%
```

```
Grid Search Results for H2ORandomForestEstimator:
```

Model Id	**Hyperparameters: [ntrees, sample_rate, max_depth, balance_classes]**	**mse**
Grid_DRF_py_87_model_python_1466382079157_49_model_19	[300, 0.9, 50, True]	0.0249340
Grid_DRF_py_87_model_python_1466382079157_49_model_18	[300, 0.8, 50, True]	0.0258412
Grid_DRF_py_87_model_python_1466382079157_49_model_17	[300, 0.6, 50, True]	0.0289790
Grid_DRF_py_87_model_python_1466382079157_49_model_16	[300, 0.5, 50, True]	0.0314358
Grid_DRF_py_87_model_python_1466382079157_49_model_38	[300, 0.8, 50, False]	0.0417964
---	---	---
Grid_DRF_py_87_model_python_1466382079157_49_model_23	[300, 0.9, 3, False]	0.0914980
Grid_DRF_py_87_model_python_1466382079157_49_model_1	[300, 0.6, 3, True]	0.1040888
Grid_DRF_py_87_model_python_1466382079157_49_model_0	[300, 0.5, 3, True]	0.1042337
Grid_DRF_py_87_model_python_1466382079157_49_model_2	[300, 0.8, 3, True]	0.1042843
Grid_DRF_py_87_model_python_1466382079157_49_model_3	[300, 0.9, 3, True]	0.1060737

predict	p0	p1
1	0.531042	0.468958
1	0.510856	0.489144
1	0.51637	0.48363
1	0.542997	0.457003
1	0.544576	0.455424
1	0.560277	0.439723
1	0.544576	0.455424
1	0.5408	0.4592
1	0.535741	0.464259
1	0.498822	0.501178

```
gbm Grid Build Progress: [##################################################] 100%

Grid Search Results for H2OGradientBoostingEstimator:
```

Model Id	Hyperparameters: [learn_rate, col_sample_rate, ntrees, sample_rate, max_depth]	mse
Grid_GBM_py_87_model_python_1466382079157_52_model_23	[0.3, 0.9, 300, 1.0, 30]	0.0001859
Grid_GBM_py_87_model_python_1466382079157_52_model_20	[0.3, 0.9, 300, 1.0, 12]	0.0001859
Grid_GBM_py_87_model_python_1466382079157_52_model_47	[0.3, 1.0, 300, 1.0, 12]	0.0001859
Grid_GBM_py_87_model_python_1466382079157_52_model_26	[0.3, 0.9, 300, 1.0, 50]	0.0001859
Grid_GBM_py_87_model_python_1466382079157_52_model_53	[0.3, 1.0, 300, 1.0, 50]	0.0001859
---	---	---
Grid_GBM_py_87_model_python_1466382079157_52_model_33	[0.01, 1.0, 300, 0.5, 50]	0.0196867
Grid_GBM_py_87_model_python_1466382079157_52_model_6	[0.01, 0.9, 300, 0.5, 50]	0.0197013

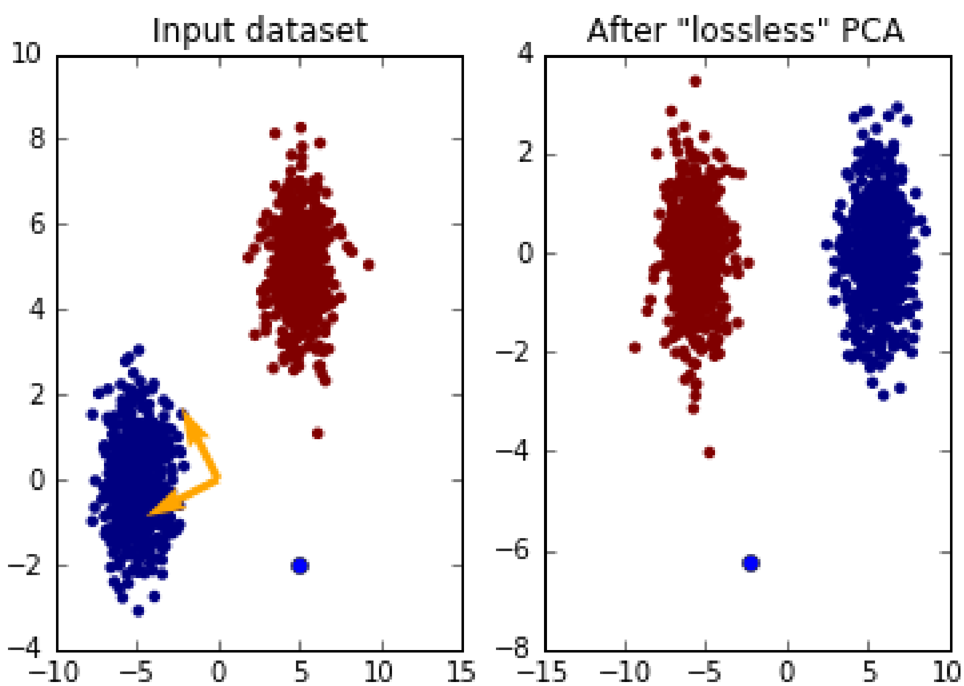
```
gbm Model Build Progress: [##################################################] 100%

Confusion Matrix (Act/Pred) for max accuracy @ threshold = 0.99983413575:
```

	0	1	Error	Rate
0	1639.0	0.0	0.0	(0.0/1639.0)
1	1.0	1050.0	0.001	(1.0/1051.0)
Total	1640.0	1050.0	0.0004	(1.0/2690.0)

Chapter 7, Unsupervised Learning at Scale

$$\hat{X} = X \cdot T$$

$$\widehat{X}$$



$$(U, \Sigma, W)$$

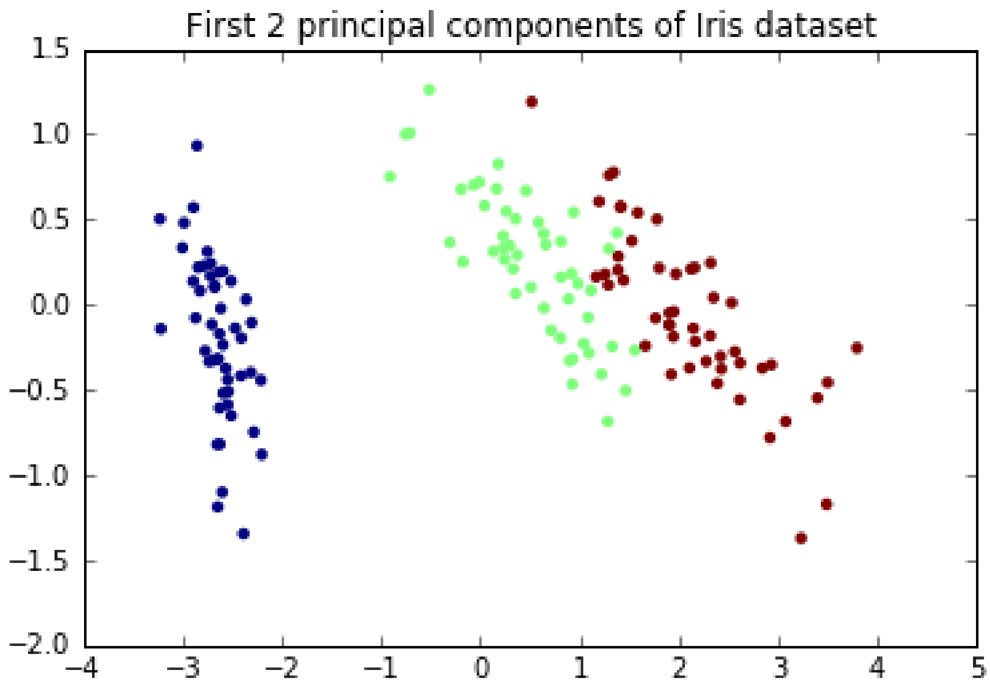
$$M = U \cdot \Sigma \cdot W^T$$

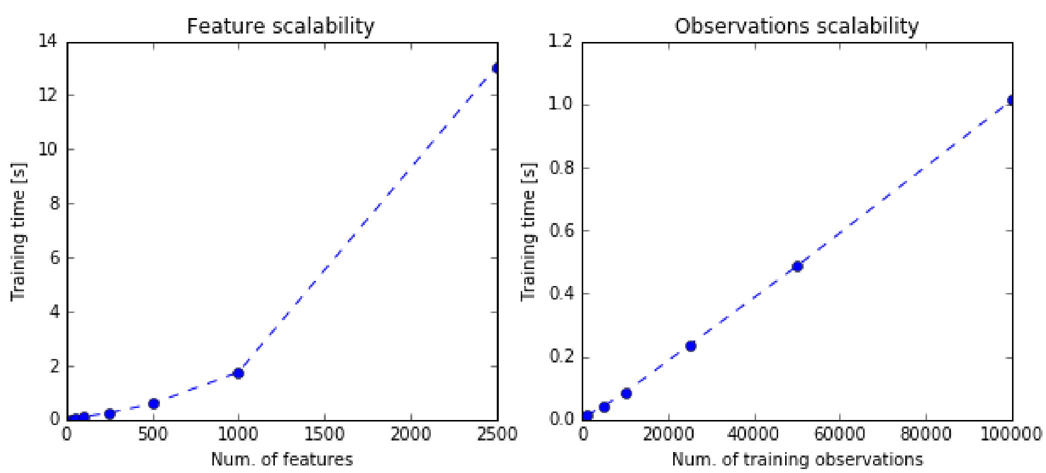
$$M \cdot M^T$$

$$\Sigma$$

$$M^T \cdot M$$

$$X^T \cdot X$$


First 2 principal components of Iris dataset
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4
−3
−2
−1
0
1
2
3
4
5


Feature scalability
Training time [s]
Num. of features
0
2
4
6
8
10
12
14
0
500
1000
1500
2000
2500
Observations scalability
Training time [s]
Num. of training observations
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
20000
40000
60000
80000
100000

$$X \approx Q \cdot Q^T \cdot X$$

$$Q^T \cdot X = U \cdot \Sigma \cdot W^T$$

$$X \approx Q \cdot Q^T \cdot X = Q \cdot U \cdot \Sigma \cdot W^T$$

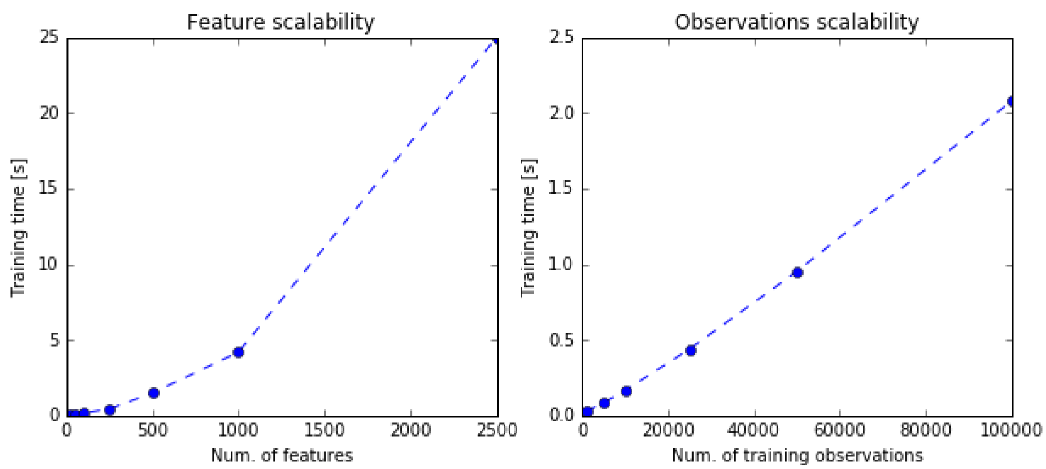
$$Q \cdot U = S$$

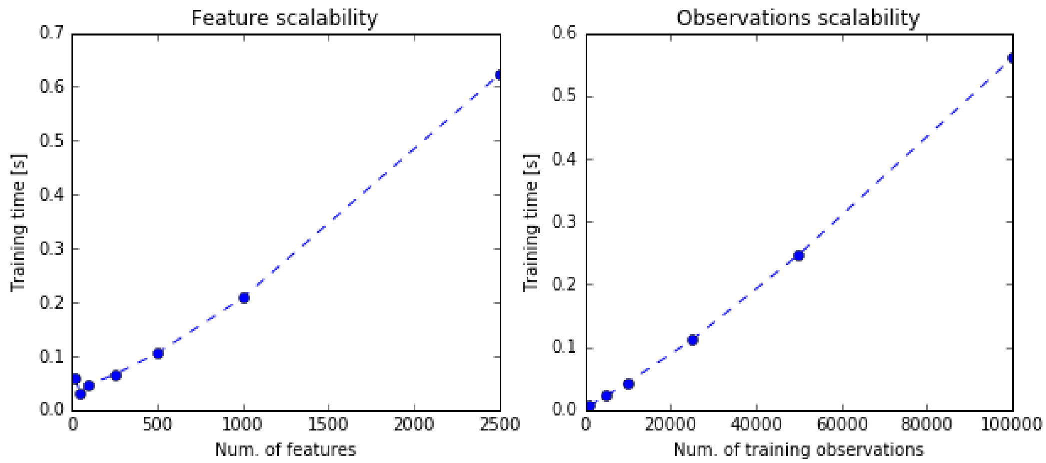
$$X \approx S \cdot \Sigma \cdot W^T$$

$$\Omega$$

$$Y = X \cdot \Omega$$

$$Y = Q \cdot R$$

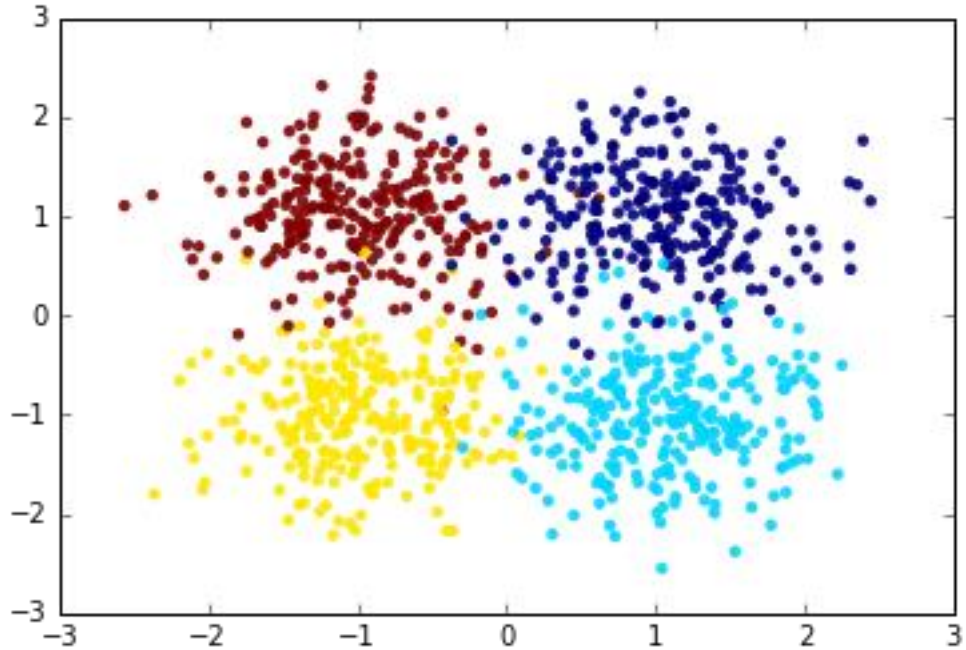




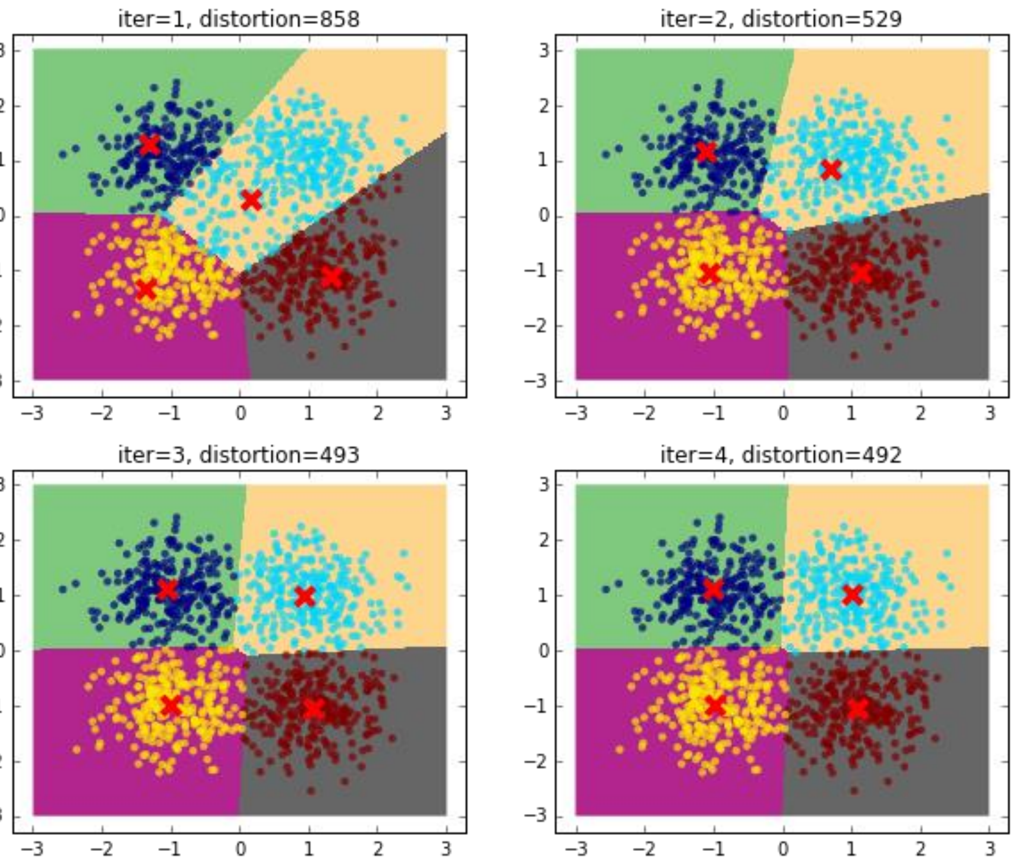
$$D = \sum_{i=i}^{K} \sum_{x \in S_i} ||x - Ci||^2$$

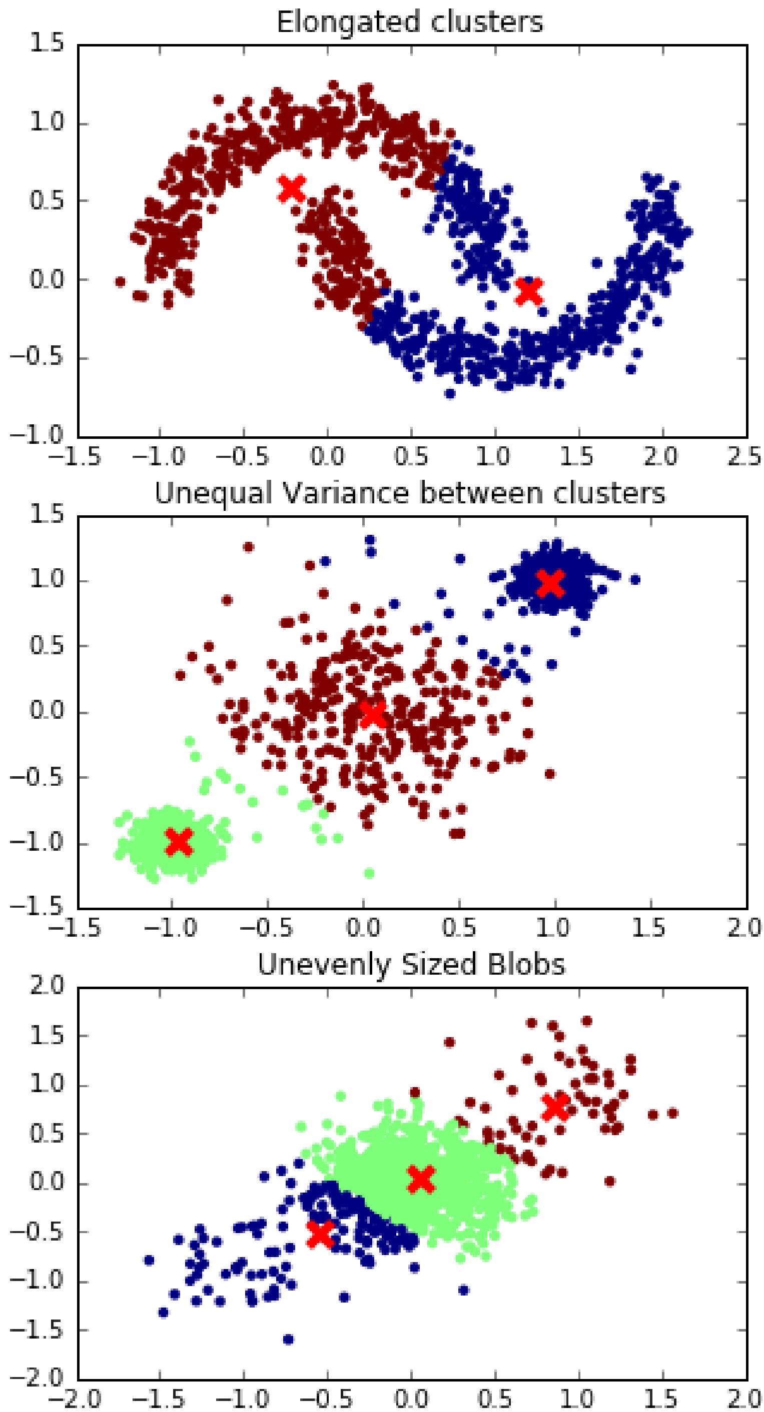
$$S_i^{(t)} = \{ \ x: \|x - Ci\|^2 = \min_j \|x - Cj\|^2 \ \}$$

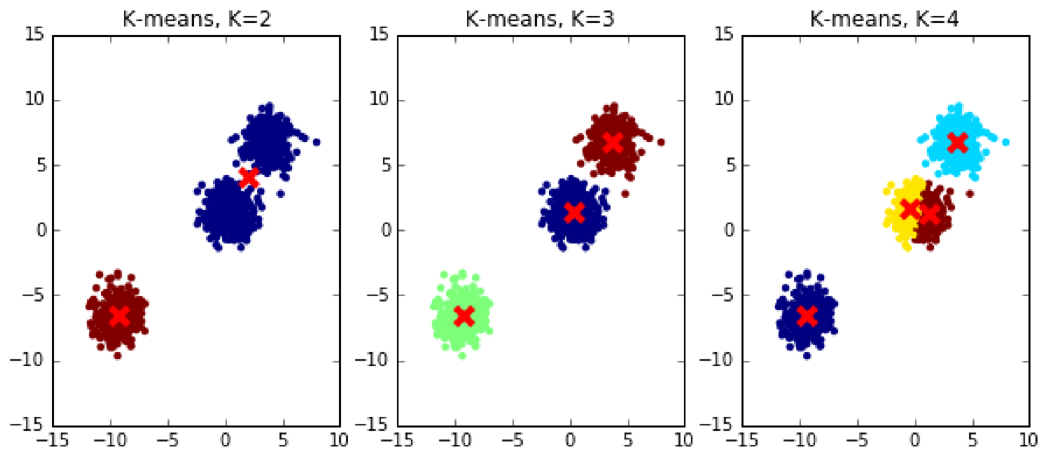
$$C_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \cdot \sum_{x \in S_i^{(t)}} x \ for \ i = 1, \ldots, K$$



3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


iter=1, distortion=858
iter=2, distortion=529
iter=3, distortion=493
iter=4, distortion=492

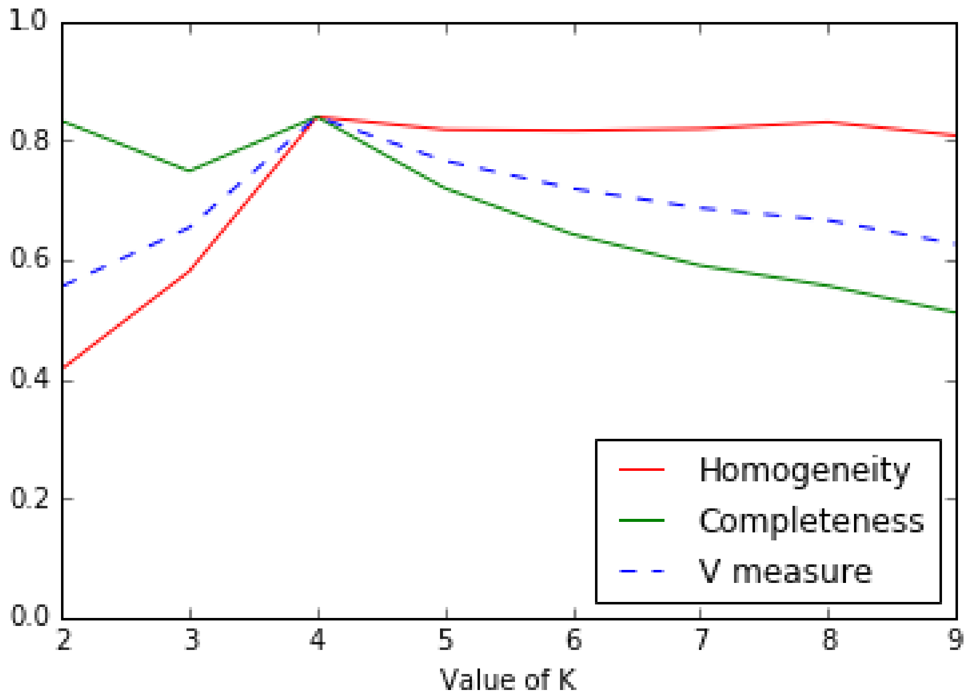

Elongated clusters
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Unequal Variance between clusters
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Unevenly Sized Blobs
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

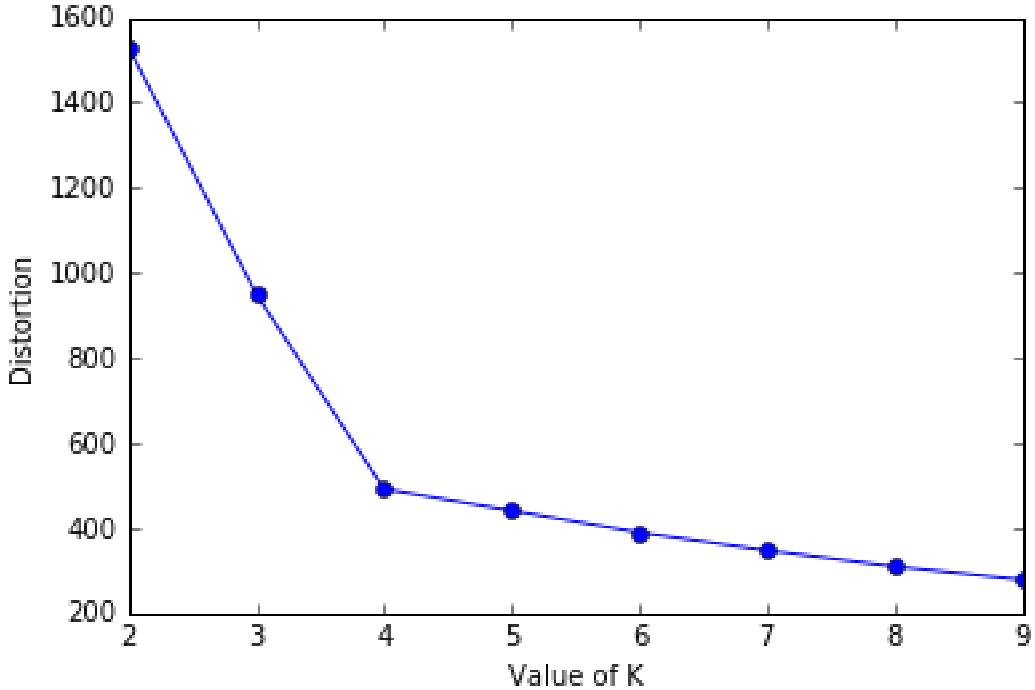


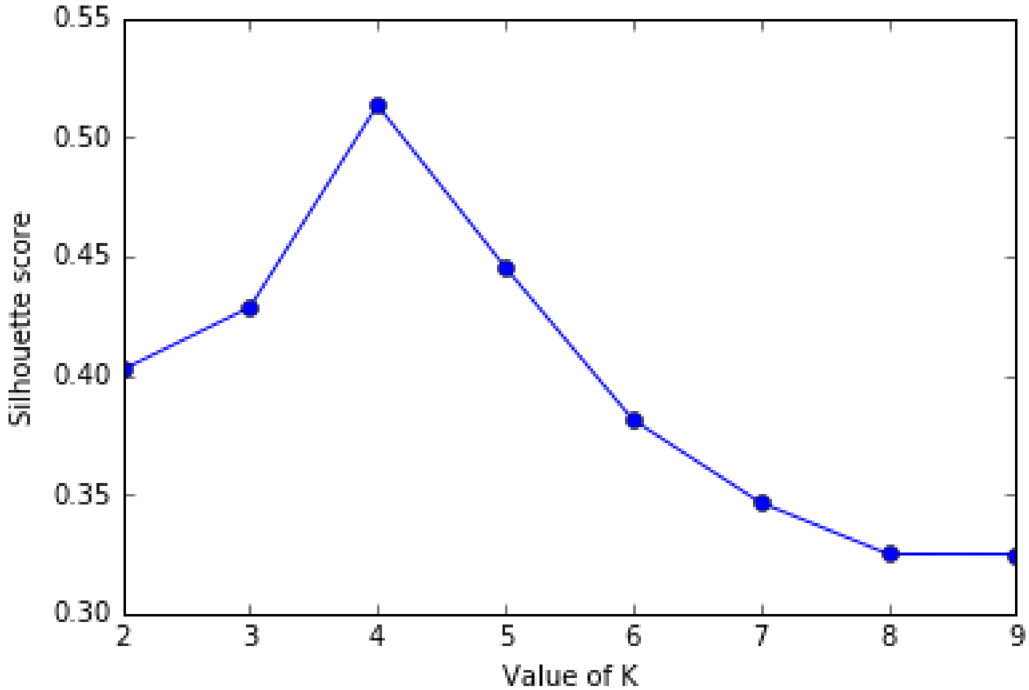
$$h = 1 - \frac{H(C|K)}{H(C)}$$

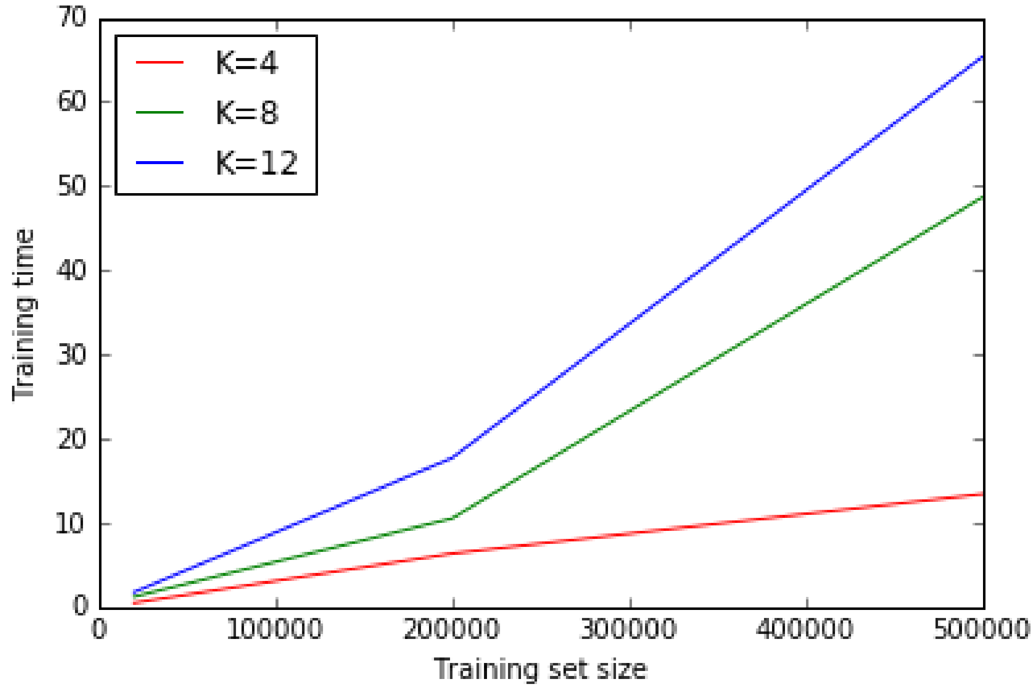
$$c = 1 - \frac{H(K|C)}{H(K)}$$

$$v = 2 \cdot \frac{h \cdot c}{h + c}$$

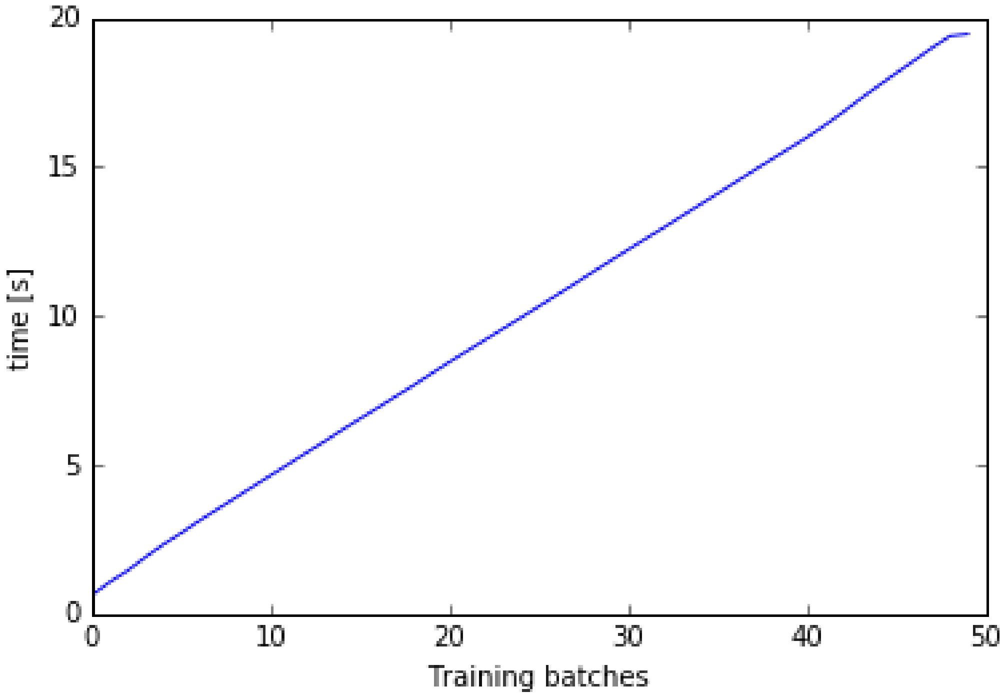


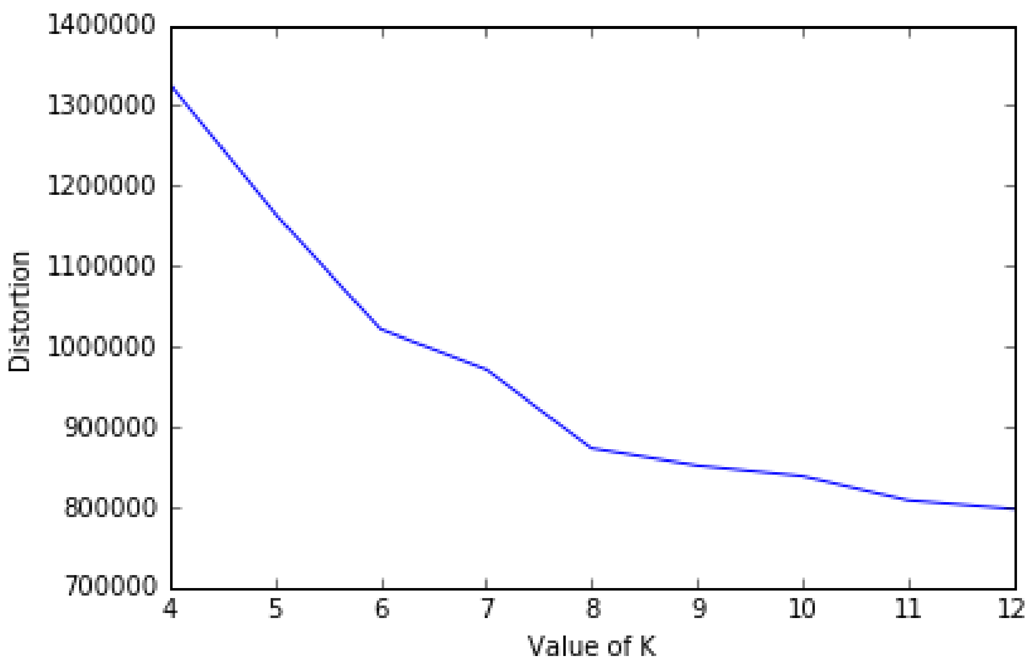

1600
1400
1200
1000
800
600
400
200
Distortion
2
3
4
5
6
7
8
9
Value of K

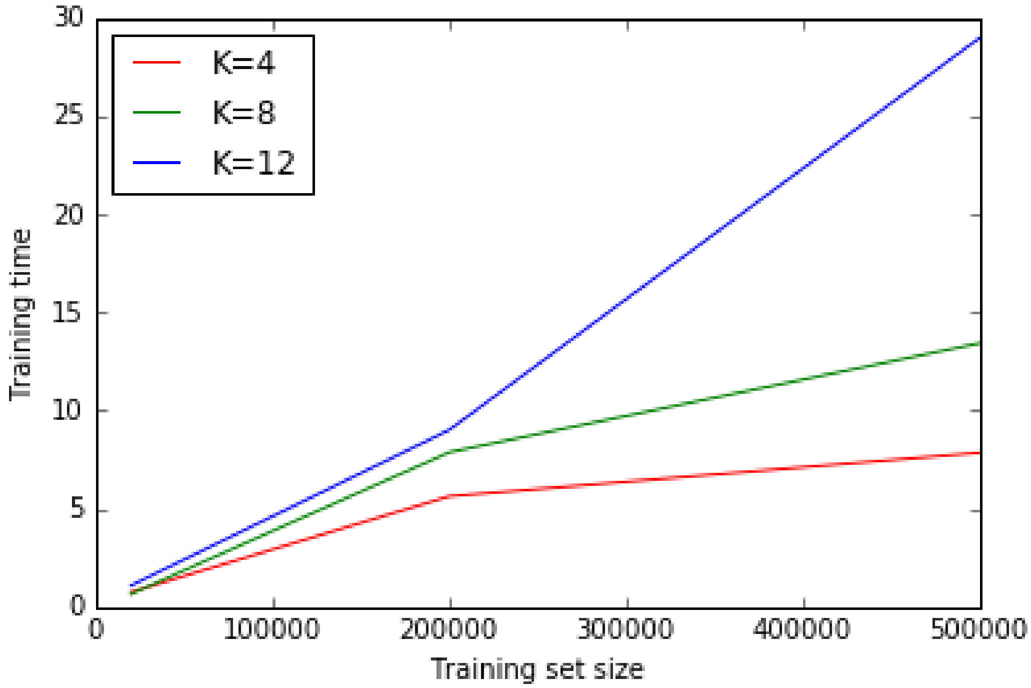

0.55
0.50
0.45
0.40
0.35
0.30
Silhouette score
2
3
4
5
6
7
8
9
Value of K



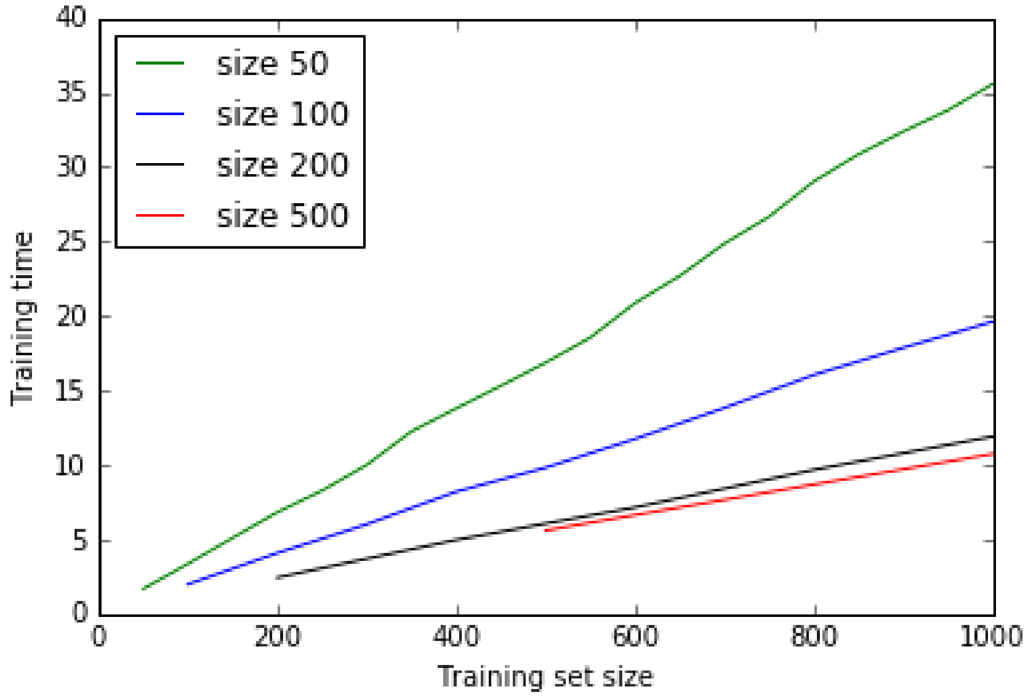
K=4
K=8
K=12
Training time
Training set size
0
10
20
30
40
50
60
70
100000
200000
300000
400000
500000


time [s]
Training batches
0
5
10
15
20
0
10
20
30
40
50


Distortion
Value of K
700000
800000
900000
1000000
1100000
1200000
1300000
1400000
4
5
6
7
8
9
10
11
12



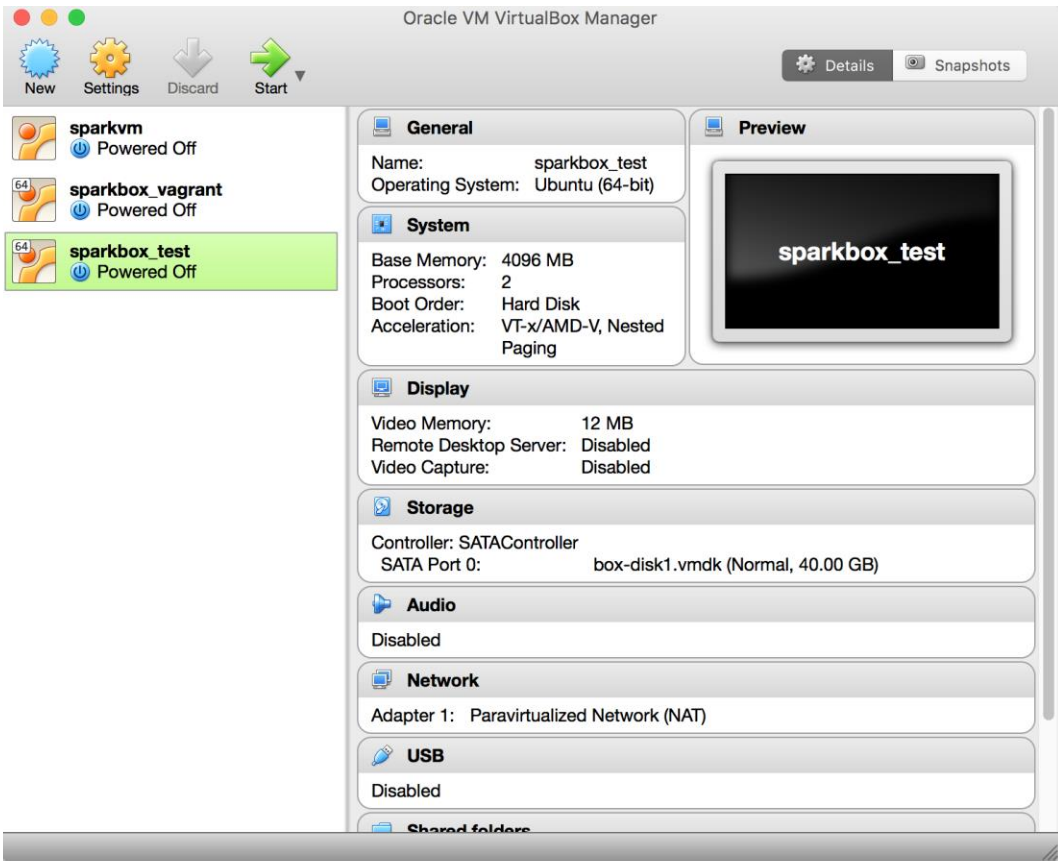
30
25
20
15
10
5
0
0
100000
200000
300000
400000
500000
Training time
Training set size
K=4
K=8
K=12

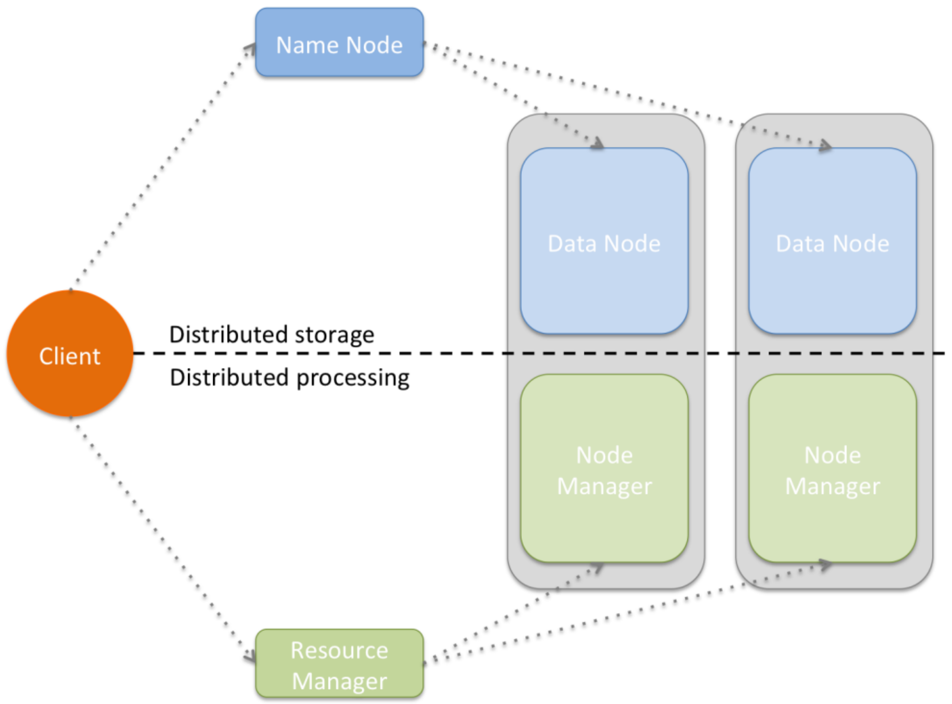


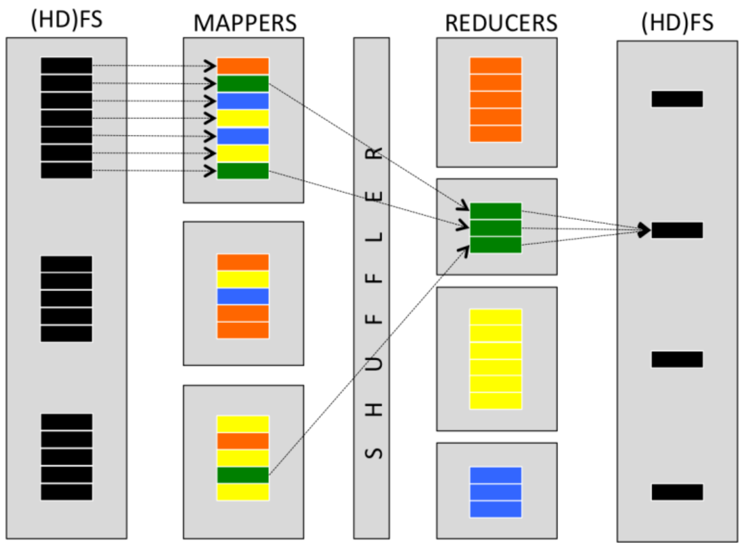
size 50
size 100
size 200
size 500
Training time
Training set size
0
5
10
15
20
25
30
35
40
0
200
400
600
800
1000

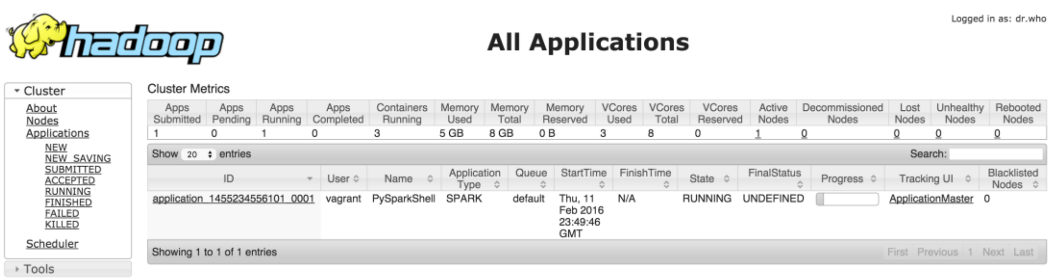
Chapter 8, Distributed Environments – Hadoop and Spark

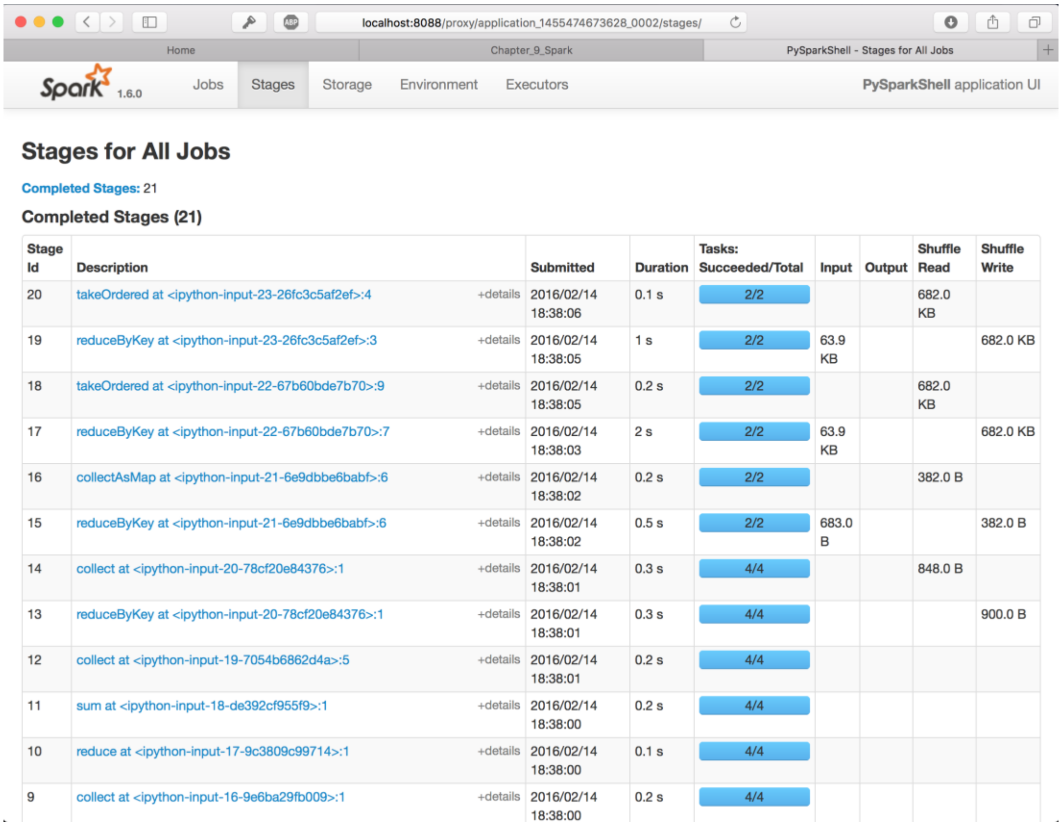
$$
\begin{aligned}
P(cluster = ok) &= P(note_1 = ok, node_2 = ok, ..., node_{100} = ok) \\
&= \left(1 - P(fail)\right)^{100} \\
&= 37\%
\end{aligned}
$$




Name Node
Data Node
Data Node
Client
Distributed storage
Distributed processing
Node Manager
Node Manager
Resource Manager


(HD)FS
MAPPERS
SHUFFLER
REDUCERS
(HD)FS


hadoop
Logged in as: dr.who
All Applications
Cluster
About
Nodes
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved | Active Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted Nodes
1 | 0 | 1 | 0 | 3 | 5 GB | 8 GB | 0 B | 3 | 8 | 0 | 1 | 0 | 0 | 0 | 0
Show 20 entries
Search:
ID | User | Name | Application Type | Queue | StartTime | FinishTime | State | FinalStatus | Progress | Tracking UI | Blacklisted Nodes
application_1455234556101_0001 | vagrant | PySparkShell | SPARK | default | Thu, 11 Feb 2016 23:49:46 GMT | N/A | RUNNING | UNDEFINED | | ApplicationMaster | 0
Showing 1 to 1 of 1 entries
First Previous 1 Next Last



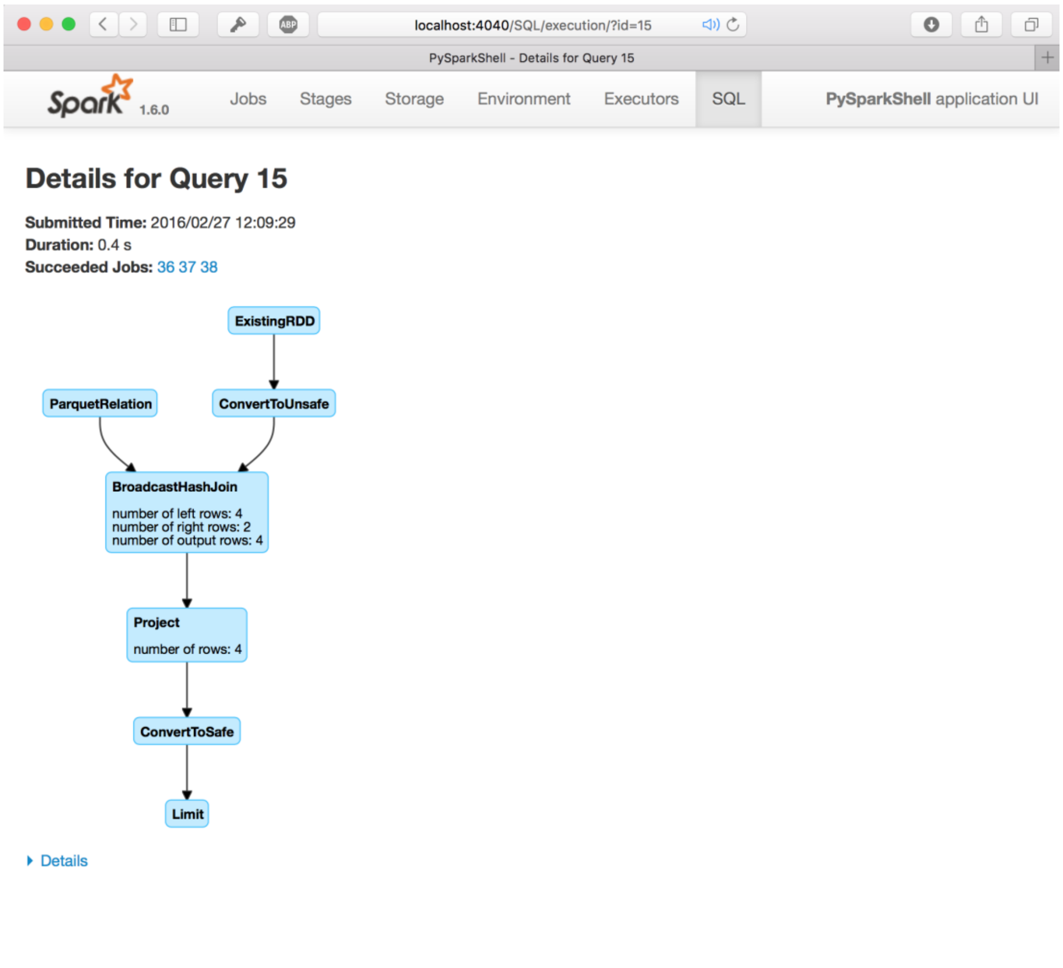
Stages for All Jobs

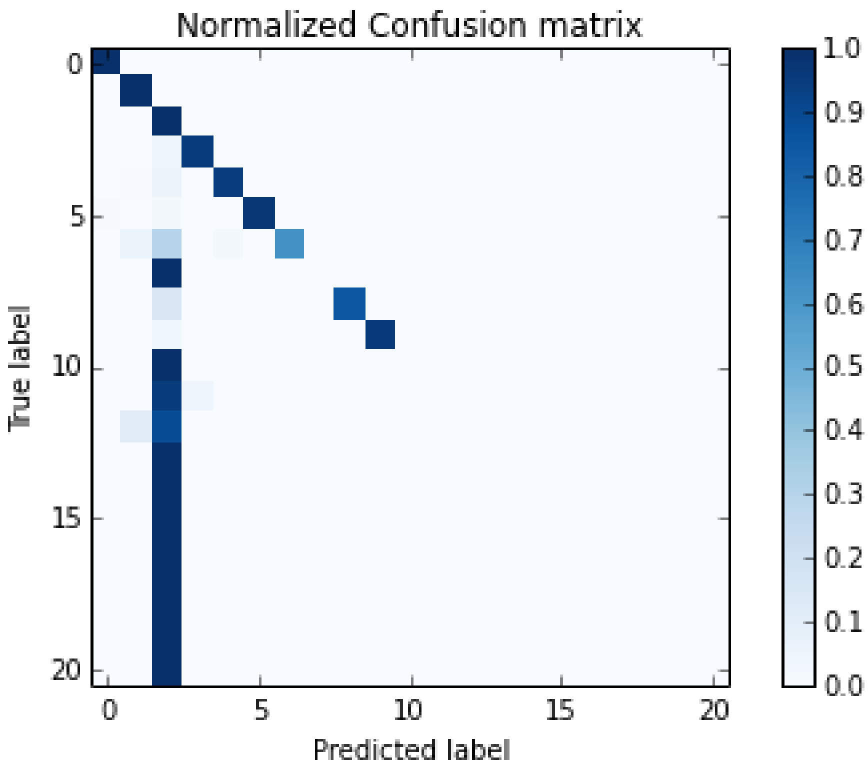
Completed Stages: 21

Completed Stages (21)

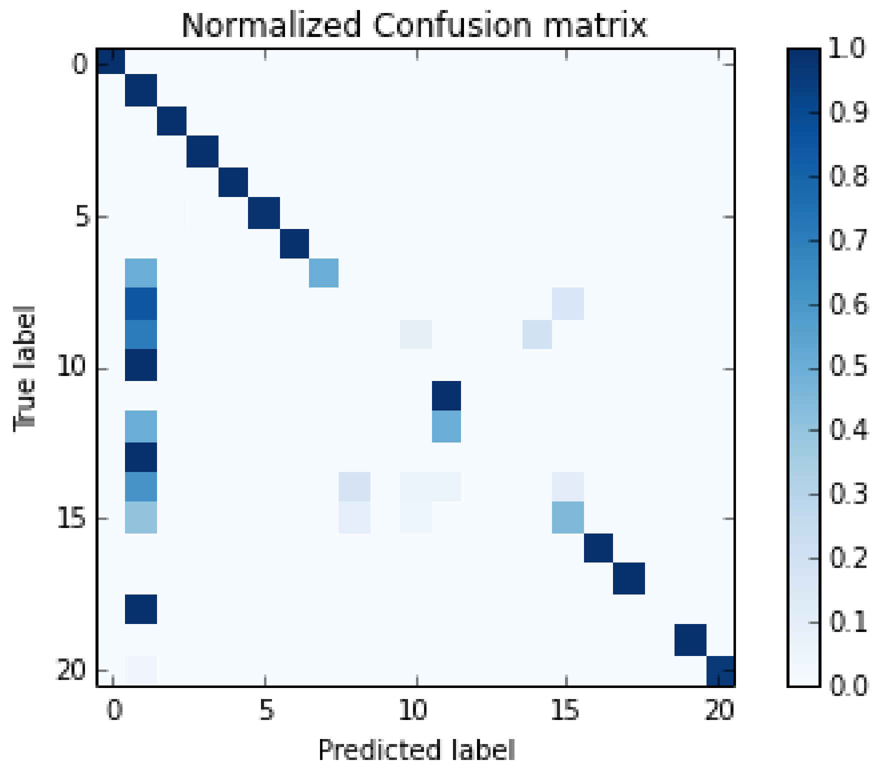
Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
20	takeOrdered at <ipython-input-23-26fc3c5af2ef>:4 +details	2016/02/14 18:38:06	0.1 s	2/2			682.0 KB	
19	reduceByKey at <ipython-input-23-26fc3c5af2ef>:3 +details	2016/02/14 18:38:05	1 s	2/2	63.9 KB			682.0 KB
18	takeOrdered at <ipython-input-22-67b60bde7b70>:9 +details	2016/02/14 18:38:05	0.2 s	2/2			682.0 KB	
17	reduceByKey at <ipython-input-22-67b60bde7b70>:7 +details	2016/02/14 18:38:03	2 s	2/2	63.9 KB			682.0 KB
16	collectAsMap at <ipython-input-21-6e9dbbe6babf>:6 +details	2016/02/14 18:38:02	0.2 s	2/2			382.0 B	
15	reduceByKey at <ipython-input-21-6e9dbbe6babf>:6 +details	2016/02/14 18:38:02	0.5 s	2/2	683.0 B			382.0 B
14	collect at <ipython-input-20-78cf20e84376>:1 +details	2016/02/14 18:38:01	0.3 s	4/4			848.0 B	
13	reduceByKey at <ipython-input-20-78cf20e84376>:1 +details	2016/02/14 18:38:01	0.3 s	4/4				900.0 B
12	collect at <ipython-input-19-7054b6862d4a>:5 +details	2016/02/14 18:38:01	0.2 s	4/4				
11	sum at <ipython-input-18-de392cf955f9>:1 +details	2016/02/14 18:38:00	0.2 s	4/4				
10	reduce at <ipython-input-17-9c3809c99714>:1 +details	2016/02/14 18:38:00	0.1 s	4/4				
9	collect at <ipython-input-16-9e6ba29fb009>:1 +details	2016/02/14 18:38:00	0.2 s	4/4				

Chapter 9, Practical Machine Learning with Spark





Normalized Confusion matrix
True label
Predicted label
0
5
10
15
20
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0



Normalized Confusion matrix
True label
Predicted label
0
5
10
15
20
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Appendix, Introduction to GPUs and Theano

GPU	CPU
Large number of cores (but slower than CPU cores)	Small number of cores, but much faster than GPU-cores
High memory bandwidth to control the cores	Lower memory bandwidth
Special purpose	General purpose
highly parallel processing	sequential processing