MACHINE LEARNING FROM OPTIMIZATION PERSPECTIVE

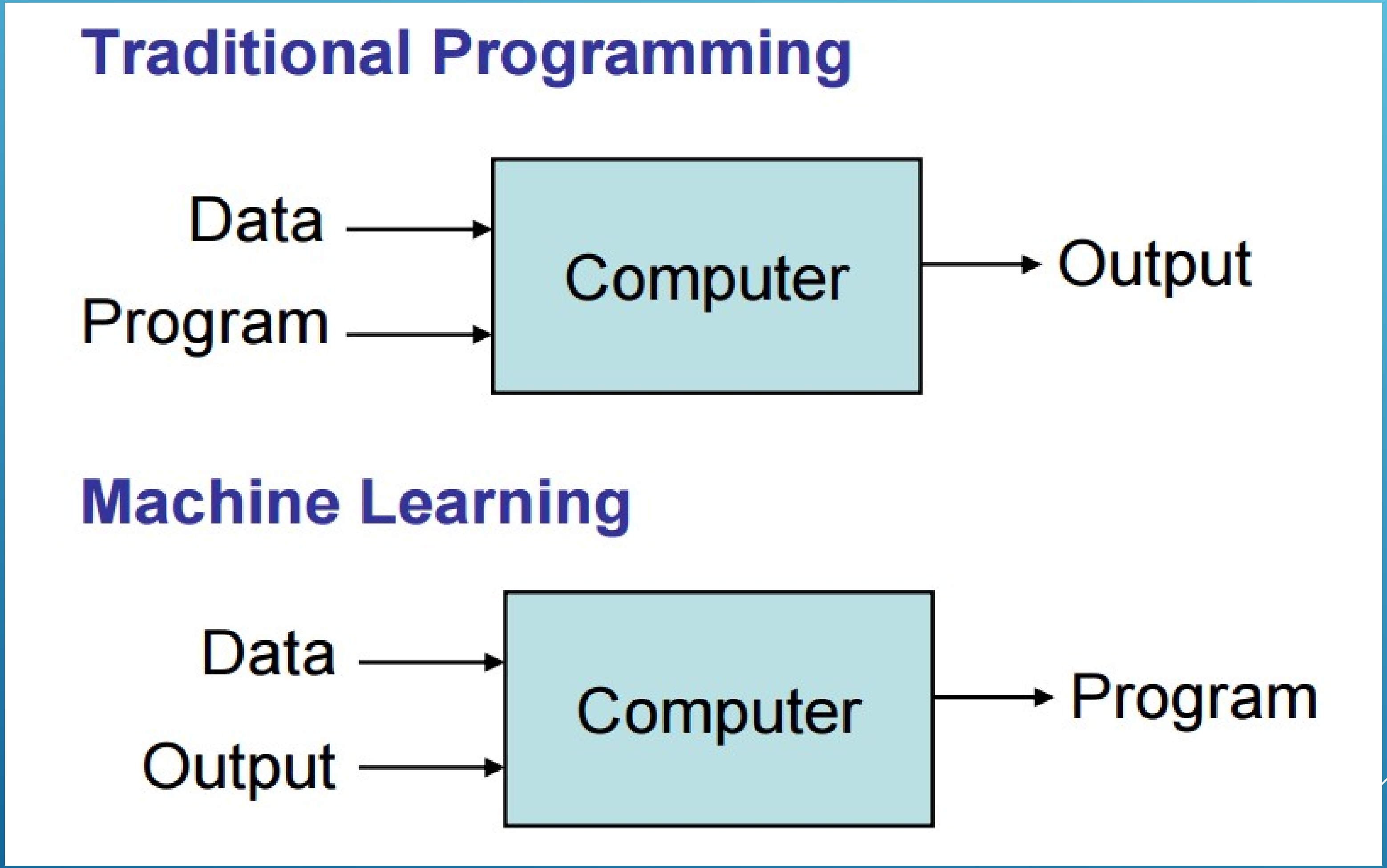
Zheng Han

June 15th, 2017

CONTENT

- Machine Learning (ML)
- Case Study
- Optimization in ML
- - Learning Theory
- - Regularization
- - Learning Algorithm

ML – A FAVORABLE PERSPECTIVE



ML – ANALOGY

Magic?

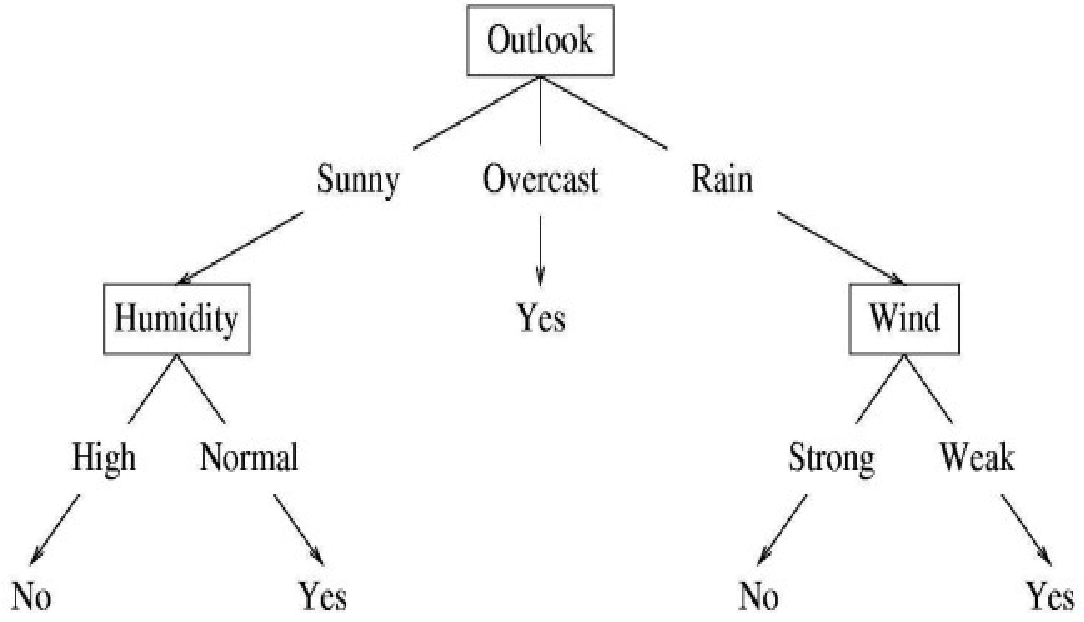
No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

ML – THREE COMPONENTS

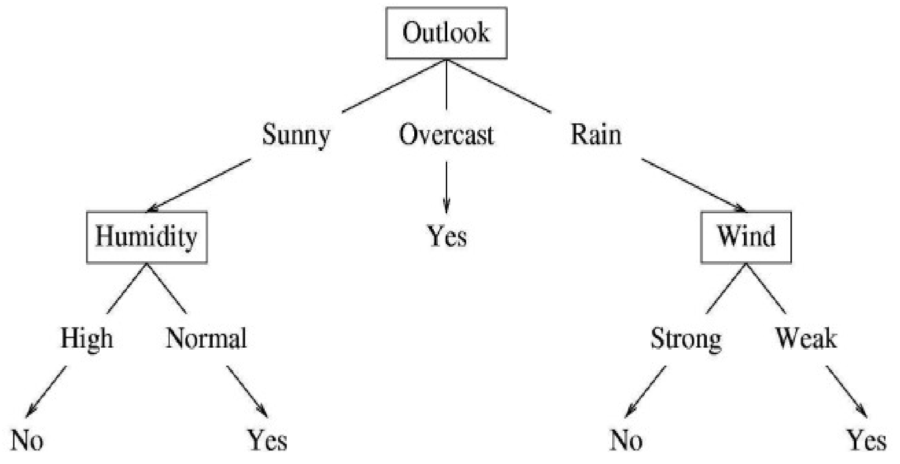
Machine Learning = Representation + Evaluation + Optimization

CASE STUDY – DECISION TREE



Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

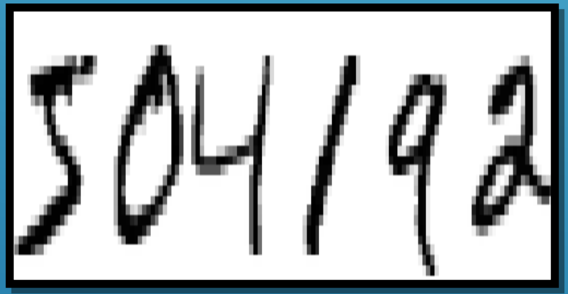
CASE STUDY – DECISION TREE



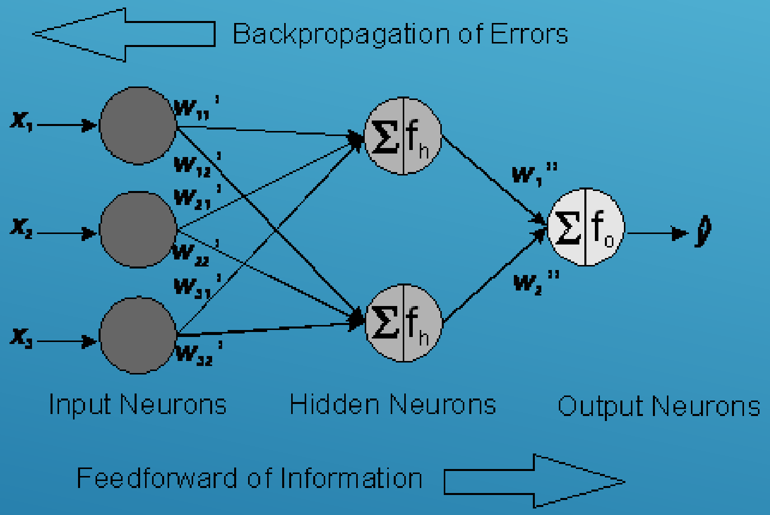
Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

- Representation: x = (Sunny Hot, High, Strong), tree structure to represent boolean function
- Evaluation: false positive rate, false negative rate, etc..
- Optimization: efficiently construct a tree that gives relatively low predictive error

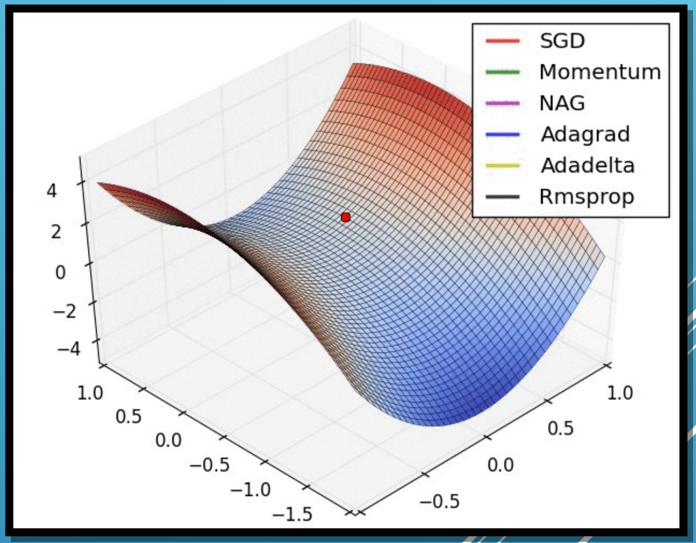
CASE STUDY – NEURAL NETWORK



Representation: images -> pixels -> matrices



Backpropagation



Optimization:
Stochastic Gradient Descent(SGD)s:
Momentum / Nesterov accelerated gradient
Adagrad / Adadelta / RMSprop / Adam

ML – AUTOMATE AUTOMATION

Table 1: The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

OPTIMIZATION

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & e(x) = 0 \\ & c(x) \leq 0 \end{aligned}$$

OPTIMIZATION IN ML

Representation

$\mathrm{x} \in \mathcal{X}, \; y \in \mathcal{Y}$

$(\mathrm{x}_1, \mathrm{y}_1), (\mathrm{x}_2, \mathrm{y}_2), \ldots, (\mathrm{x}_n, \mathrm{y}_n)$

$h : \mathcal{X} \mapsto \mathcal{Y}$

Evaluation

$\ell(h(x), y)$

Optimization

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$$

Parameterized Optimization

$$\min_{\omega \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; \omega), y_i)$$

OPTIMIZATION IN ML

Expected Risk $$R(h) := \int_{(\mathcal{X},\mathcal{Y})} \ell(h(x),y)dP(x,y) = E[\ell(h(x),y)]$$

Empirical Risk $$R_n(h) := \frac{1}{n}\sum_{i=1}^{n}\ell(h(x_i),y_i)$$

Learning Theory $$\sup_{h\in\mathcal{H}}|R(h) - R_n(h)| \leq \mathcal{O}\left(\sqrt{\frac{1}{2n}\log\left(\frac{2}{n}\right) + \frac{d_{\mathcal{H}}}{n}\log\left(\frac{n}{d_{\mathcal{H}}}\right)}\right)$$

$d_{\mathcal{H}}$: VC dimension, measures the capacity of $\mathcal{H}$

OPTIMIZATION IN ML

Expected Risk

$$R(h) := \int_{(\mathcal{X},\mathcal{Y})} \ell(h(x), y) dP(x, y) = E[\ell(h(x), y)]$$

Empirical Risk

$$R_n(h) := \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i), y_i)$$

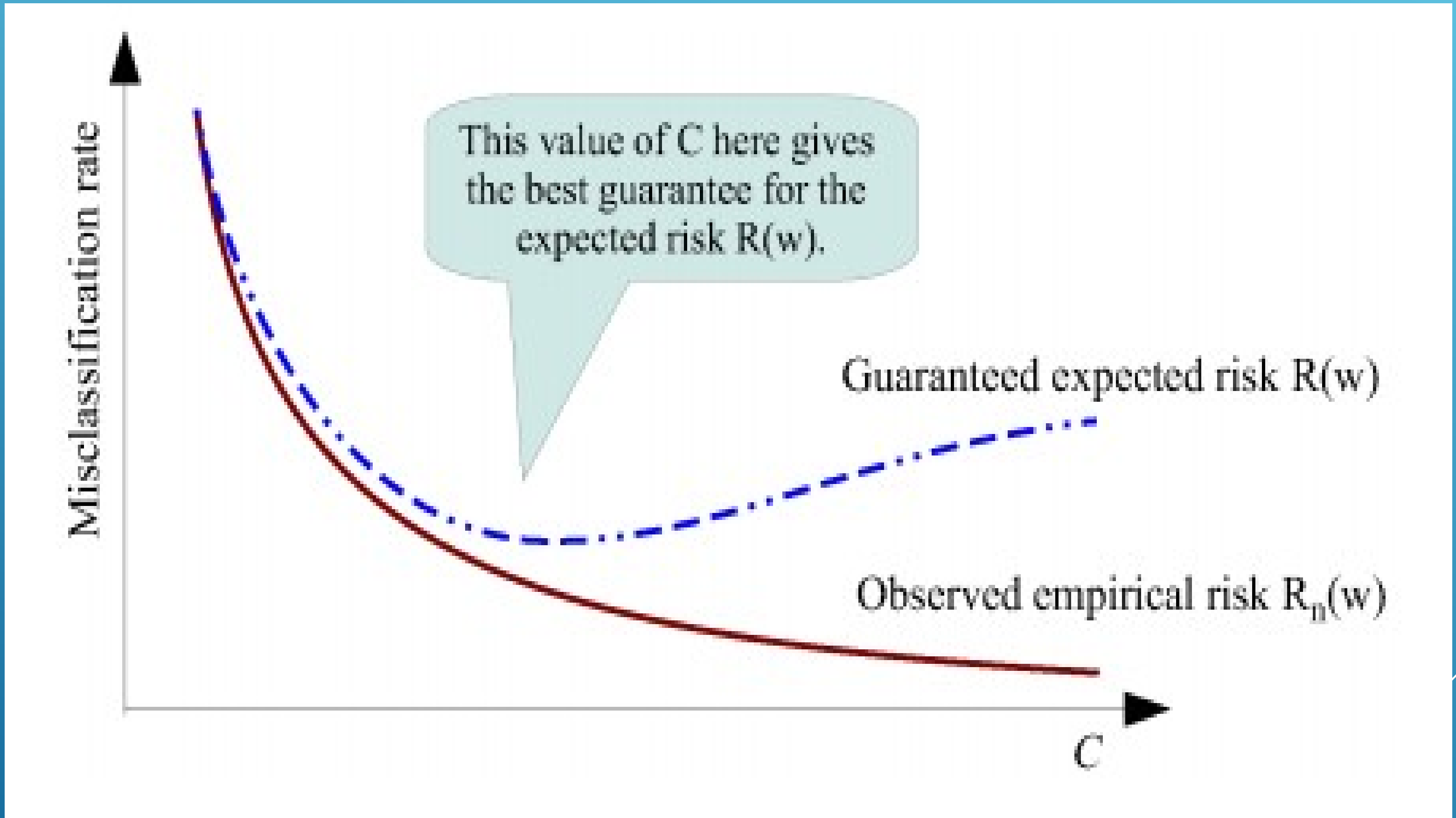
Model Complexity

$$\mathcal{H}_C := \{h \in \mathcal{H} : \ \Omega(h) \leq C\}$$

Structural Risk Minimization

$$\min_{h \in \mathcal{H}_C} \ R_n(h)$$

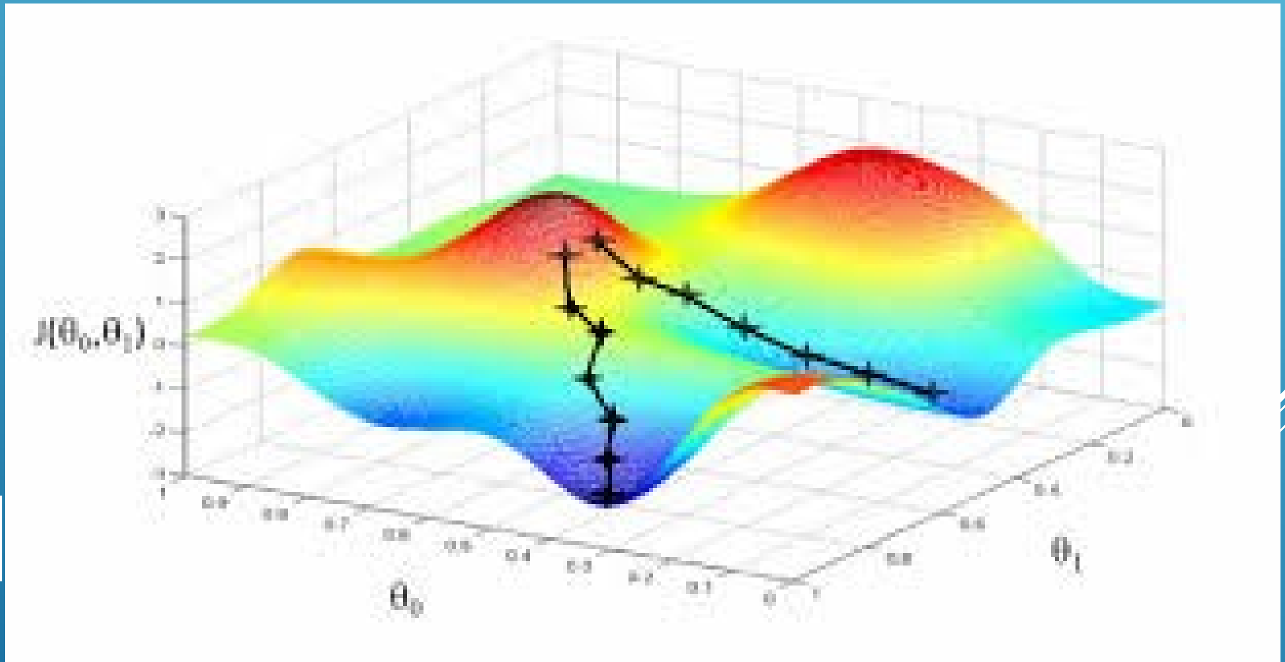
STRUCTURAL RISK MINIMIZATION BY REGULARIZATION



OPTIMIZATION ALGORITHM IN ML

Gradient Descent Method:

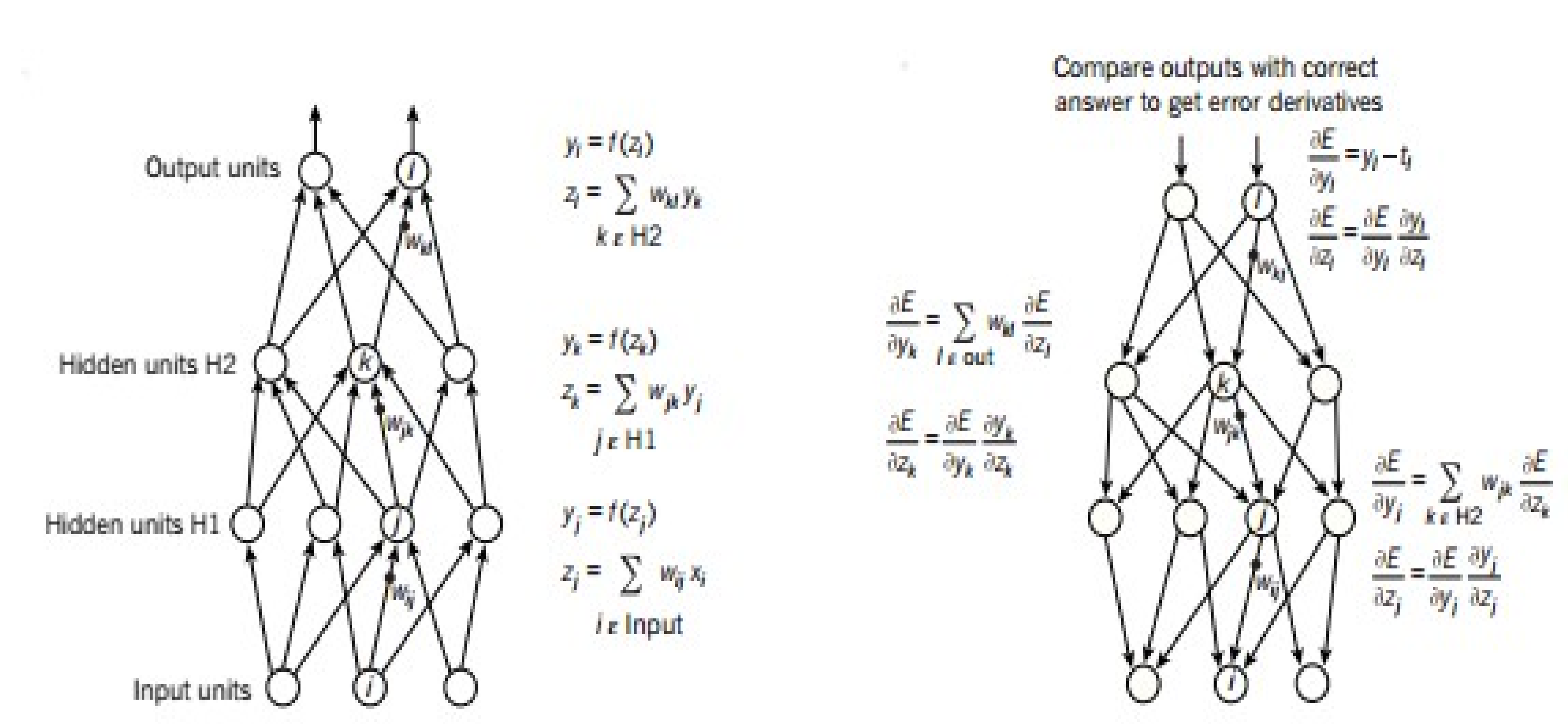
$$\theta^{k+1} = \theta^k - \alpha_k \nabla J(\theta^k)$$



Second-order Method:

$$\theta^{k+1} = \theta^k - \alpha_k G_k \nabla J(\theta^k)$$

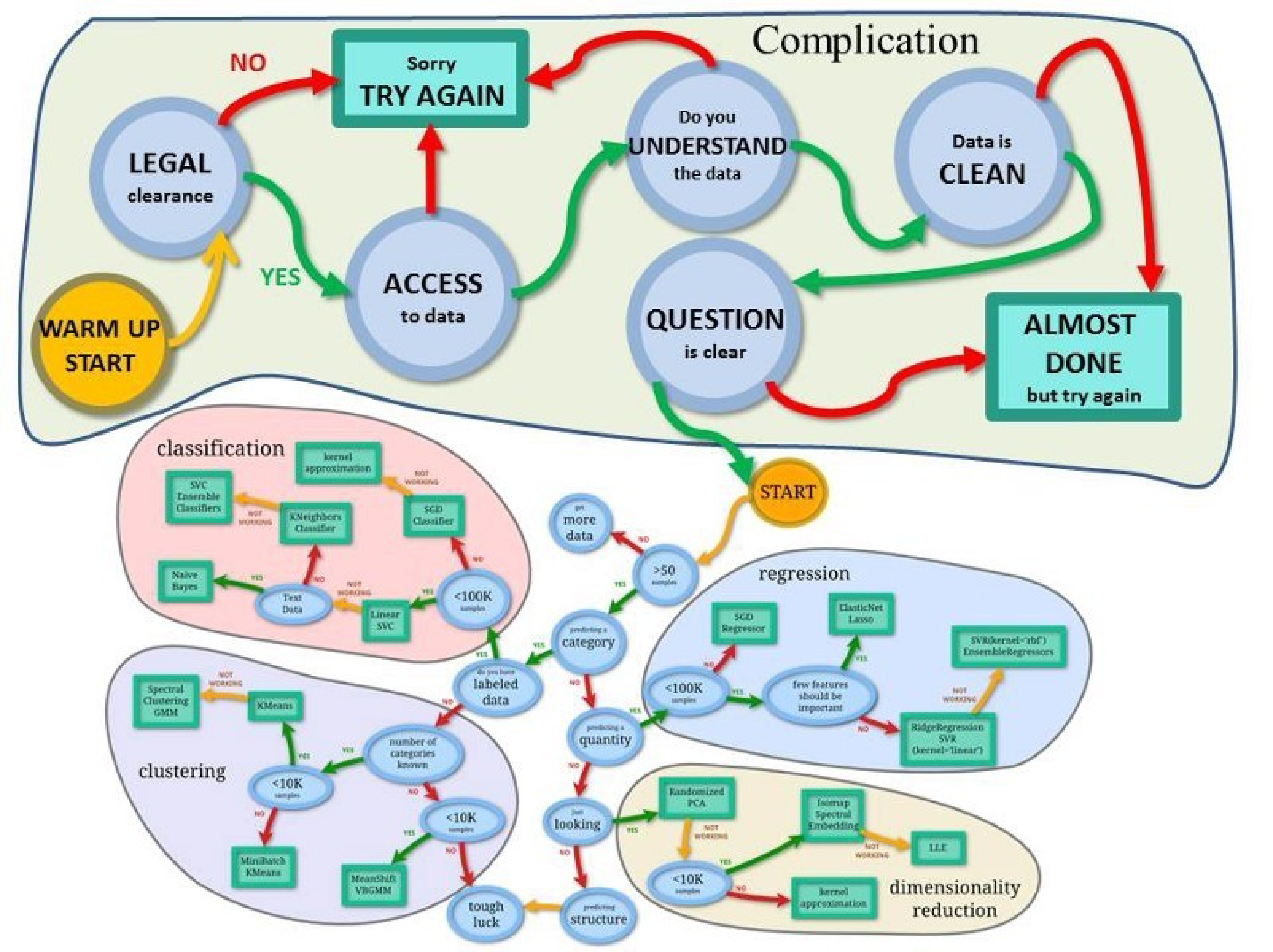
OPTIMIZATION ALG FOR DEEP LEARNING



OTHER RELEVANT TOPICS

- Reinforcement Learning: dynamic programming
- Online Learning: online convex optimization
- Evolutionary Algorithms: generic algorithms
- Big Data: distributed optimization, sparse optimization

ML IN REAL-WORLD



REFERENCES

- A Few Useful Things to Know about Machine Learning. Pedro Domingos
- The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. Pedro Domingos
- Scikit-Learn: http://scikit-learn.org
- An Extended Version Of The Scikit-Learn Cheat Sheet. Christophe Bourguignat
- Deep Learning. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton
- Optimization Methods for Large-Scale Machine Learning. Leon Bottou, Frank E. Curtis, and Jorge Nocedal
- The LION Way: Machine Learning plus Intelligent Optimization. Roberto Battiti, Mauro Brunato