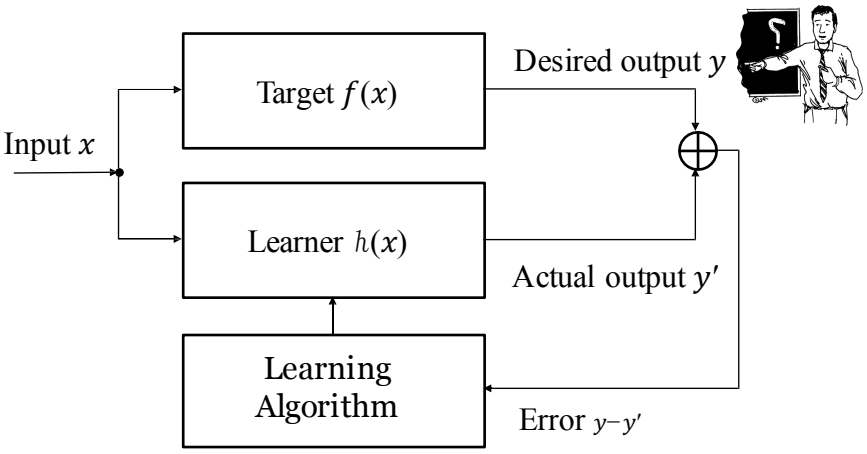
Artificial Intelligence
Fundamentals of machine learning

Topics of this lecture

- Formulation of machine learning
- Taxonomy of learning algorithms
  - Supervised, semi-supervised, and unsupervised learning
  - Parametric and non-parametric learning
  - Online and offline learning
  - Evolutionary learning
  - Reinforcement learning
  - Deterministic and statistical learning

Formulation of machine learning (1)



Formulation of machine learning (2)

- Concepts to learn: $X_1, X_2, .., X_{N_c}$

$$X_i = \{x \in X | f(x) = y_i, \quad y_i \in Y\}$$

where $Y = \{y_1, y_2, .., y_{N_c}\}$ is the label set.

- A training datum is usually given as a pair (x, y), where x is the observation and y is the label given by a "teacher".
- Learning is the process to find a good "learner" or learning model $h(x)$ to approximate the target function $f(x)$.

Formulation of machine learning (3)

- In machine learning, we call $h(x)$ a **hypothesis**. The set of all hypotheses *H* is called the **hypothesis space**.
- *H* is a set of functions (e.g. all linear functions defined in R^n).
- Machine learning is an **optimization problem** for finding the best hypothesis $h(x)$ from *H*.
- The goodness of a hypothesis can be evaluated by using the followin error function:

$$E = \frac{1}{|m|} \sum_{x \epsilon m} |f(x) - h(x)|^2$$

- This is known as the "mean squared error" (MSE).
- More theoretically, *H* can be considered a Hilbert space, and the error can be defined using the norm $|f(x) - h(x)|$

Formulation of machine learning (4)

- We may use a **loss function** instead of using the error function directly. The simplest loss function is 0-1 loss defined by:

$$L = \sum_{x \epsilon m} 1(f(x) \neq h(x))$$

  Where $1P(x)$ is 1 if P is true, and 0 otherwise.
- The error or loss defined above is **empirical** in the sense that they are defined based on the observed data only. The empirical cost or loss may not be the same as the **predictive value** when we observe more data.
- The best predictive error E^* or loss L^* is called the Bayes error or Bayes loss, and the hypothesis $h^*(x)$ that achieves the best error/loss is called the **Bayes Rule**. The goal of machine learning is to find $h^*(x)$ from *H*.
- To find the best hypothesis, however, we cannot use the MSE directly because the problem is **ill-posed**. That is, even if the hypothesis so obtained is good for the given training data set, it may not **generalize well** for unknown data.

Formulation of machine learning (5)

- To avoid the problem, we usually introduce a **regularization factor** In the objective function.
- For example, if the hypothesis depends on a set of parameters $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$, we may consider θ as m-dimension vector, and define the objective function as follows:

$$\min_{\theta} \sum_{x \epsilon m} |f(x) - h_\theta(x)|^2 + \lambda(x)$$

- where λ is a parameter for judging the balance between the error and regularization factor.
- The often used norm for the regularization factor is Euclidean norm. We may also use the norm of $h(x)$ defined in the Hilbert space H.
- The physical meaning of regularization is to find the **most smooth solution** amount others, to improve the generalization ability.
- For *sparse learning*, we can introduce a factor to encourage learner *parsimony*.

Formulation of machine learning (6)

- If $f(x)$ takes values from R^{N_o}, the problem is called **regression**, where N_o is the number of output variables.
- That is, regression is also a function approximation problem.
- For regression problem, 0-1 loss is not suitable because a good hypothesis $h(x)$ may not exactly equal to $f(x)$ for $x \in \mathrm{m}$.
- Instead, we can use other loss functions, such as

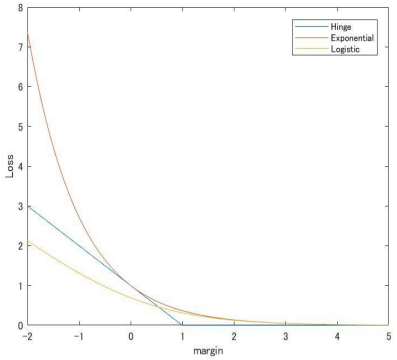
$$Hinge\ loss: L(u) = \max\{1 - u, 0\}$$

$$Exponential\ loss: L(u) = e^{-u}$$

$$Logistic\ loss: L(u) = \log(1 + e^{-u})$$

Formulation of machine learning (7)

- Note u in the loss function can be defined as $f(x)\times h(x)$, which is called the ***margin***.
- For example, if the desired value is $f(x)$=1, and the actual output is $h(x)$=0.9, the Hinge loss is 0.1, the exponential loss is 0.41, and the logistic loss is 0.34; but if the desired value is $f(x)$=1, and the actual output is $h(x)$=-0.2, the Hinge loss is 1.2, the exponential loss is 1.22, and the logistic loss is 0.798.
- We may also define u using the difference between $f(x)$ and $h(x)$.



Supervised, semi-supervised, and unsupervised learning (1)

- Supervised learning: If teacher signals or labels are available for training data.
- Un-supervised learning: If teacher signals are not available.
- Semi-supervised learning: If part of the signals are available.

Supervised, semi-supervised, and unsupervised learning (2)

- Teacher signals can be provided in different forms.
  - Correct answers for all input patterns.
    - Most informative, often used for pattern recognition.
  - Reward or penalty
    - The learner must learn what is the correct answer for each input pattern, to achieve a high score.
    - This is commonly known as **reinforcement learning**.
  - Goodness (fitness) of the current hypothesis
    - Each learner knows how good it is, and
    - Many learners can work together to find a good learner, through information exchange, or through self-improvement.
    - This is commonly known as **evolutionary learning**, **or meta-heuristic-based learning** in general.

Supervised, semi-supervised, and unsupervised learning (3)

- When there is no teacher signal at all, we need to partition the feature space into several disjoint clusters, and patterns in each cluster should share some common properties.
- This is in general a chicken-and-egg problem:
  - Define the clusters first, and then divide the space.
  - Divide the space first, and then define the clusters.
- The k-means algorithms is a heuristic algorithm for resolving the dilemma.
- Using different "similarity" measures, we can obtain different results → Some results may not be consistent with our expectation.

Supervised, semi-supervised, and unsupervised learning (4)

- When we have many un-labeled data, we can first define the "structure" of the feature space roughly based on un-supervised learning, and then use the labeled data to define (calibrate) the label of each cluster.
- This is also a heuristic based on the observation that "probably similar patterns have the same label".
- In big data analytics, each datum may have many labels. Algorithms proposed for single label data are certainly not enough. → further study needed!

Parametric and non-parametric learning (1)

- Parametric learning: If each hypothesis in the hypothesis space can be defined by a set of parameters.
  - Example 1: Similar data can be generated following a Gaussian distribution in the feature space. The mean and standard deviation can be used as parameters to determine this group of data.
  - Example 2: A neural network with a given structure is defined by its weights, and the weights are the parameters.
- The point is to find the best set of parameters to fit given training data.

Parametric and non-parametric learning (2)

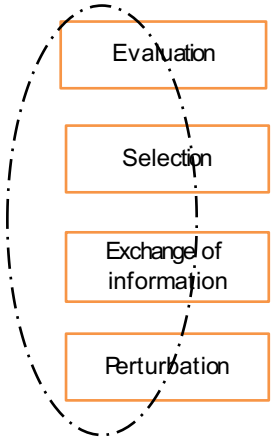
- Non-parametric learning: If the hypotheses do not depends on a certain number of parameters.
  - Example 1: A nearest neighbor classifier using all training data cannot be defined by a "small" set of parameters, especially when the number of data is large, and changing.
  - Example 2: Support vector machine (SVM) is similar to a neural network in structure, but the number of support vectors depends on the training set size. So an SVM is non-parametric.

Online and off line learning

- Online learning:
  - Update the learner using newly observed data.
  - Do not use the data all at once.
  - → May obtain a good learner efficient by starting from a small training set.
  - → Suitable for learning with mobile devices.
- Offline learning:
  - Train the learner using all data.
  - Can obtain a better learner.
  - → Need more computing power for learning.
  - → Suitable for learning with strong platforms.

Evolutionary learning or population-based learning (1)

- Typical evolutionary algorithms include genetic algorithm (GA), evolutionary programing (EP), genetic programming (GP), evolution strategy (ES), etc.
- One important advantage of these algorithms is that they can find both structure and parameters together.



Evolutionary learning or population-based learning (2)

- In recent years, many other meta-heuristic algorithms have been proposed.
- Examples include particle swarm optimization (PSO), differential evolution (DE), etc.
- These algorithms can be adopted to machine learning because machine learning is nothing but an optimization problem.
- After finding a good solution, we may improve the search path (or the learning process) using some other meta-heuristic algorithm (e.g. ant colony optimization) → learning of learning.

Reinforcement learning (1)

- Reinforcement learning (RL) is important for "strategy learning". It is useful for robotics, for playing games, etc.
- The well-known alpha-GO actually combined RL with deep learning, and was the first program that defeated human expert Go-players.

Reinforcement learning (2)

- In RL, a learner is called an agent. The point is to take a correct "action" for each environment "situation".
- If there is a teacher who can tell the correct actions for all situations, we can use supervised learning. In RL, we suppose that the teacher only "rewards" or "punishes" the agent under some (not all) situations.
- RL can find a "map" (a Q-table) that defines the relation between the situation set and the action set, so that the agent can get the largest reward by following this map.

Reinforcement learning (3)

- To play a game successfully, the computer can generate many different situations, and find a map between situation set and action set in such a way to win the game (with a high probability).
- Thus, even if there is no human opponent, a machine can improve its skill by playing with itself, using RL.
- Of course, if the machine has the honor to play many games with human experts, it can find the best strategy more "efficiently" without generating many "impossible" situations.

Deterministic learning and statistic learning (1)

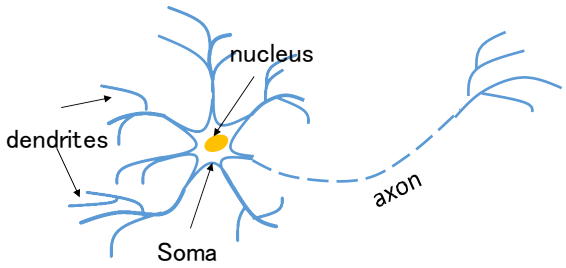
- Given a hypothesis space H, we can find the best (in some given criterion) hypothesis deterministically or statistically.
- In deterministic learning, we usually assume that all functions are defined in a high dimensional Euclidean space, and do not use "probability" explicitly.
- For example, in the case we want to find a neural network, we can use some method proposed in the context of ***mathematical programming*** (e.g. the well known BP algorithm).
- Generally speaking, basis function-based methods are also deterministic.

Deterministic learning and statistic learning (2)

- In most cases, however, it is natural to assume that the data are generated by following some probability distribution (e.g. Gaussian, or combination of several Gaussians).
- Instead of finding a deterministic function, it is natural to find the probabilities such as
  - Given a pattern x, the probability that x belongs to a certain class,
  - given a class, the probability that x is observed,
  - and so on.
- Based on these probabilities, we may make some "recommended" decisions, instead of telling yes or no.

Mechanism of a bio-neuron

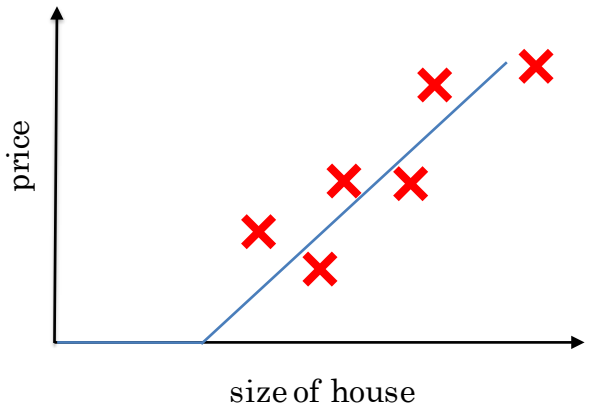
- A neuron (mainly) consists of a soma, many dendrites and an axon. Pulses from other neurons are inputted to the soma from the dendrites via synapses, and are integrated there. The neuron sends a pulse to other neurons through the axon when the potential of the soma is higher than a threshold.



Artificial Neural Network

What is a Neural Network?

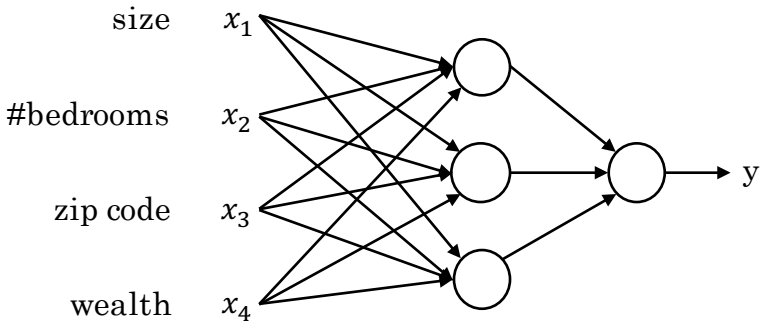
Housing Price Prediction



Artificial Neural Network

What is a Neural Network?

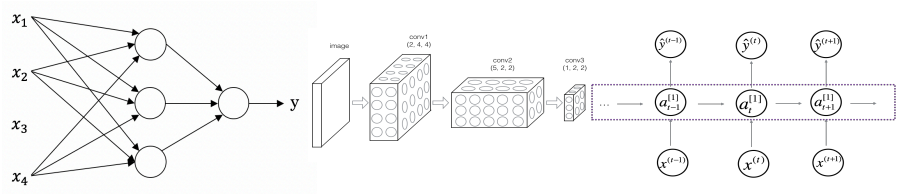
Housing Price Prediction



Supervised Learning

Input(x)	Output (y)	Application
Home features	Price	Real Estate
Ad, user info	Click on ad? (0/1)	Online Advertising
Image	Object (1,…,1000)	Photo tagging
Audio	Text transcript	Speech recognition
English	Chinese	Machine translation
Image, Radar info	Position of other cars	Autonomous driving

Neural Network examples



Standard NN

Convolutional NN

Recurrent NN

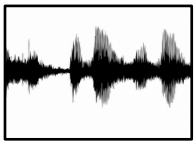
Supervised Learning

Structured Data

Size	#bedrooms	...	Price (1000$s)
2104	3		400
1600	3		330
2400	3		369
⋮	⋮		⋮
3000	4		540

User Age	Ad Id	...	Click
41	93242		1
80	93287		0
18	87312		1
⋮	⋮		⋮
27	71244		1

Unstructured Data



Audio

Image

I pledge to Nigeria my country to be faithful ...

Text

Basics of Neural Network Programming

- All about Implementing NN for Learning
- Iterating through the entire training set of n training examples.
- How to avoid explicit looping
- Forward and Backward Propagation

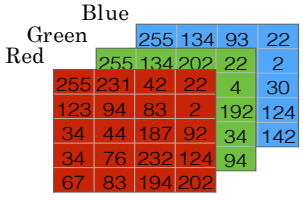
Contents

1. Binary Classification
2. Logistic Regression
3. Logistic cost Function
4. Gradient Descent
5. Derivatives
6. Logistic Regression Gradient Descent

Binary Classification

Logistic regression is an algorithm for Binary classification.

→ 1 (cat) vs 0 (non cat)



Notations

Single training example is of the form:

(x, y) x ∈ $\mathbb{R}^{n_x}$ and y∈ $\{0,1\}$

For m training examples, we've:

$\{(x^1, y^1), (x^2, y^2), (x^3, y^3), \ldots (x^m, y^m)\}$

The m training examples can be put in a more compact way using $n_x \times m$ matrix.

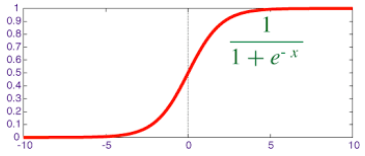
Logistic Regression

An algorithm for Binary classification
E.g given an $image \rightarrow \odot \rightarrow \hat{y} = p(y = 1|x)$

That is, $Given\ X, we\ want\ to\ predict\ \hat{y} = p(y = 1|x)$

Parameters: $w \in \mathbb{R}^{n_x}$ and $b \in \mathbb{R}$

Output: $\hat{y} = \delta(w^T X + b)$



Logistic Regression Cost function

$\hat{y} = \sigma(w^T x + b)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$

Given $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, want $\hat{y}^{(i)} \approx y^{(i)}$.

Loss (error) function: $\mathcal{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$

$\mathcal{L}(\hat{y}, y) = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$

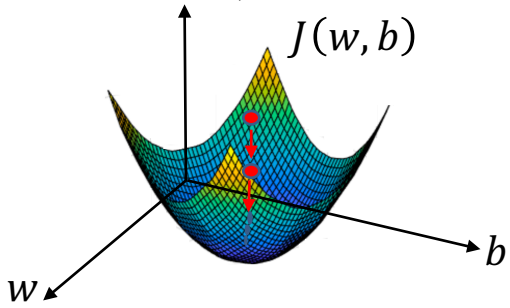
$$Cost\ Function\ \mathcal{J}(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)}) \right)$$

Gradient Descent

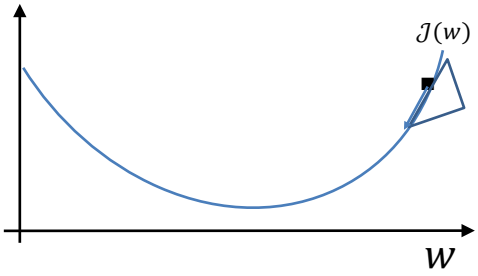
Recap: $\hat{y} = \sigma(w^T x + b)$, $\sigma(z) = \frac{1}{1+e^{-z}}$

$$\mathcal{J}(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)}) \right)$$

Want to find w, b that minimize $\mathcal{J}(w,b)$



Gradient Descent



$Repeat\{$

$$w = w - \alpha \frac{d\mathcal{J}(w)}{dw}$$

$\}$

$For\ \mathcal{J}(w,b)$

$Repeat\{$

$$w = w - \alpha \frac{d\mathcal{J}(w,b)}{dw}$$

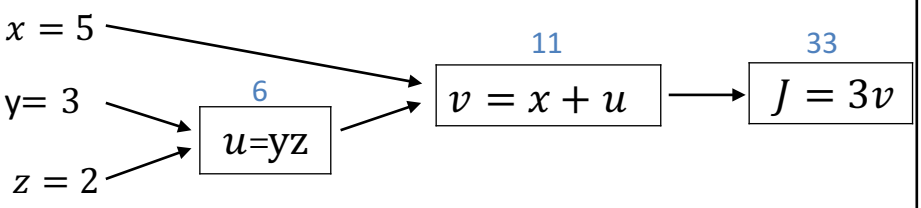
$$b = b - \alpha \frac{d\mathcal{J}(w,b)}{db}$$

$\}$

Computation Graph

$Assume\ we\ want\ compute\ \mathcal{J}(x,y,z) = 3(x + yz)$

Computing derivatives

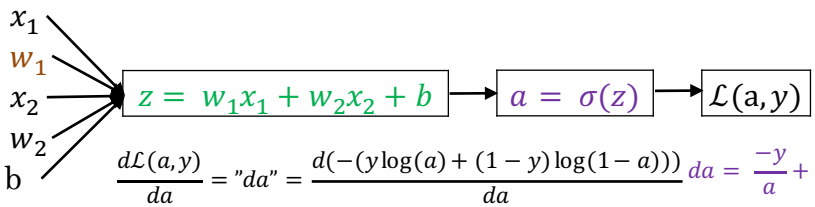


Logistic Regression Gradient descent

Logistic regression recap

$$z = w^T x + b$$
$$\hat{y} = a = \sigma(z)$$
$$\mathcal{L}(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$$

Logistic regression derivatives



$$\frac{d\mathcal{L}(a,y)}{da} = "da" = \frac{d(-(y\log(a) + (1-y)\log(1-a)))}{da} \qquad da = \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dz = \frac{d\mathcal{L}(a,y)}{dz} = \frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{da}\frac{da}{dz} = (a-y) \qquad dz = (a-y)$$

$$dw_1 = \frac{d\mathcal{L}}{dw_1} = \frac{dz}{dw_1}\frac{d\mathcal{L}}{dz} = x_1 dz \qquad dw_1 = x_1 dz$$

$$dw_2 = x_2 dz$$

$$db = dz$$

Logistic regression on *m* examples

To train an algorithm, we deal with the entire training examples to obtain

$$\mathcal{J}(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(a^{(i)}, y^{(i)})$$

$$where\ a^{(i)} = \hat{y}^{(i)} = \sigma(z^{(i)}) = \sigma(w^T x^{(i)} + b)$$

$$\frac{d}{dw_1}\mathcal{J}(w,b) = \frac{1}{m}\sum_{i=1}^{m}\frac{d}{dw_1}\mathcal{L}(a^{(i)}, y^{(i)})$$

Update:

$$w_1 = w_1 - \alpha dw_1$$
$$w_2 = w_2 - \alpha dw_2$$
$$b = w_1 - \alpha db$$

Implementing logistic regression:

$Initialize\ \mathcal{J} = 0, dw_1 = 0, dw_1 = 0, db = 0$

$for\ i = 1\ to\ m$:

$z^{(i)} = w^T x^{(i)} + b$

$a^{(i)} = \sigma(z^{(i)})$

$\mathcal{J} = -(y^{(i)}\log(a^{(i)}) + (1-y^{(i)})\log(1-a^{(i)}))$

$dz^{(i)} = a^{(i)} - y^{(i)}$

$dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)}$

$dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)}$

$db \mathrel{+}= dz^{(i)}$

$\mathcal{J} /= m$

$dw_1 /= m \quad dw_2 /= m \quad db /= m$

End