Machine Learning Basics

Lecture slides for Chapter 5 of Deep Learning

www.deeplearningbook.org

Maximum Likelihood Estimation

- derive specific functions that are good estimators for different models
- Consider m examples $\mathbb{X} = \{x^{(1)}, \ldots, x^{(m)}\}$ drawn independently from the true but unknown data generating distribution $p_{data}(x)$
- Let $p_{model}(x; \theta)$ be a parametric family of probability distributions over the same space indexed by θ
- MLE for θ is defined as

$$\theta_{ML} = \arg\max_{\theta} p_{model}(\mathbb{X}; \theta)$$

$$= \arg\max_{\theta} \prod_{i=1}^{m} p_{model}(x^{(i)}; \theta)$$

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{m} \log p_{model}(x^{(i)}; \theta)$$

Conditional Log-Likelihood and Mean Squared Error

- The maximum likelihood estimator can be generalized to estimate a conditional probability $P(y|x;\theta)$
- If X represents all inputs and Y observed targets, then the conditional maximum likelihood estimator is

$$\theta_{ML} = \underset{\theta}{argmax} P(Y|X;\theta)$$

- If the examples are assumed to be i.i.d., then this can be decomposed into

$$\theta_{ML} = \underset{\theta}{argmax} \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)};\theta)$$

Example: Linear Regression as Maximum Likelihood

- Previously, we motivated linear regression as an algorithm that learns to take input x and produce an output $\hat{y}$. The mapping is chosen to minimize MSE.
- Instead of producing a single prediction $\hat{y}$, we now think of the model as producing a conditional distribution $p(y|x)$
- We define $p(y|x) = \mathcal{N}(y;\hat{y}(x,w),\sigma^2)$
- $\hat{y}(x,w)$ gives the prediction of the mean of the Gaussian. In this example, we assume that the variance is fixed to some constant σ^2

Since the examples are assumed to be i.i.d., the conditional log-likelihood is given by

$$\sum_{i=1}^{m} \log p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \tag{5.64}$$

$$= -m \log \sigma - \frac{m}{2} \log(2\pi) - \sum_{i=1}^{m} \frac{\|\hat{y}^{(i)} - y^{(i)}\|^2}{2\sigma^2}, \tag{5.65}$$

where $\hat{y}^{(i)}$ is the output of the linear regression on the i-th input $\boldsymbol{x}^{(i)}$ and m is the number of the training examples. Comparing the log-likelihood with the mean squared error,

$$\text{MSE}_{\text{train}} = \frac{1}{m} \sum_{i=1}^{m} ||\hat{y}^{(i)} - y^{(i)}||^2, \tag{5.66}$$

we immediately see that maximizing the log-likelihood with respect to $\boldsymbol{w}$ yields the same estimate of the parameters $\boldsymbol{w}$ as does minimizing the mean squared error. The two criteria have different values but the same location of the optimum.

Supervised Learning Algorithms

- Linear regression
- Logistic Regression
- Support Vector Machines
- kNN
- Decision Tree

Unsupervised Learning Algorithms

- Informally, unsupervised learning refers to most attempts to extract information from a distribution that do not require human labour to annotate examples.
  - density estimation,
  - learning to draw samples from a distribution,
  - learning to denoise data from some distribution,
  - finding a manifold that the data lies near,
  - clustering the data into groups of related examples
- A classic unsupervised learning task is to find the "best" representation of the data
  - preserves as much information about x as possible while obeying some penalty or constraint aimed at keeping the representation simpler

Simpler representation

1. lower dimensional representations
   - compress as much information about x as possible in a smaller representation
2. sparse representations
   - embed the dataset into a representation whose entries are mostly zeroes for most inputs
3. independent representations
   - Disentangle the sources of variation underlying the data distribution such that the dimensions of the representation are statistically independent

- The notion of representation is one of the central themes of deep learning

Principal Components Analysis

- PCA provides a means of compressing data
  - We can view PCA as an unsupervised learning algorithm that learns a representation of data.
  - Learn lower dimension representation
  - elements have no linear correlation with each other
- a first step toward the criterion of learning representations whose elements are statistically independent.
- To achieve full independence, a representation learning algorithm must also remove the nonlinear relationships between variables
- PCA learns an orthogonal, linear transformation of the data that projects an input x to a representation z

k-means Clustering

- An example of a simple representation learning algorithm
- We can thus think of the algorithm as providing a k-dimensional one-hot code vector h representing an input x
- The one-hot code provided by k-means clustering is an example of a sparse representation
- we may prefer a distributed representation to a one-hot representation.
- A distributed representation could have two attributes for each vehicle—one representing its color and one representing whether it is a car or a truck.
- having many attributes reduces the burden on the algorithm to guess which single attribute we care about, and allows us to measure similarity between objects in a fine-grained way by comparing many attributes

Stochastic Gradient Descent

- Nearly all of deep learning is powered by one very important algorithm: stochastic gradient descent or SGD
- large training sets computationally expensive but necessary for generalization
- The cost function used by a machine learning algorithm often decomposes as a sum over training examples of some per-example loss function.
- The insight of stochastic gradient descent is that the gradient is an expectation. The expectation may be approximately estimated using a small set of samples

Building a ML Algorithm

- Nearly all deep learning algorithms can be described as particular instances of a fairly simple recipe:
- combine a specification of
  - a dataset,
  - a cost function,
  - An optimization procedure and
  - a model.

Challenges Motivating Deep Learning

- motivated in part by the failure oftraditional algorithms to generalize well on such AI tasks as speech recognition, object recognition NLP
- the challenge of generalizing to new examples becomes exponentially more difficult when working with high-dimensional data
- the mechanisms used to achieve generalization in traditional ML are insufficient to learn complicated functions in high-dimensional spaces.

The Curse of Dimensionality

- The number of possible distinct configurations of a set of variables increases exponentially as the number of variables increases
- the number of possible configurations of x is much larger than the number of training examples.
- The core idea in deep learning is that we assume that the data was generated by the composition of factors or features, potentially at multiple levels in a hierarchy
- Many other similarly generic assumptions can further improve deep learning algorithms.
- The exponential advantages conferred by the use of deep, distributed representations counter the exponential challenges posed by the curse of dimensionality

Manifold Learning

- A manifold is a connected region.
- Mathematically, it is a set of points, associated with a neighborhood around each point.
- The definition of a neighborhood surrounding each point implies the existence of transformations that can be applied to move on the manifold from one position to a neighboring one.
- In ML it tends to be used more loosely to designate a connected set of points that can be approximated well by considering only a small number of degrees of freedom, or dimensions, embedded in a higher-dimensional space.
- Each dimension corresponds to a local direction of variation

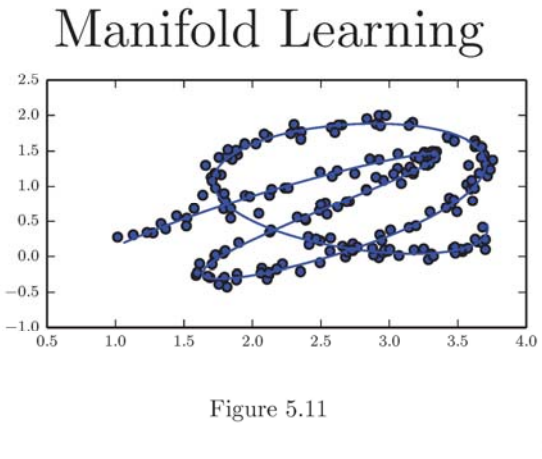

Figure 5.11

(Goodfellow 2016)

- Manifold learning algorithms assume that
  - most of R^n consists of invalid inputs
  - interesting inputs occur only along a collection of manifolds containing a small subset of points,
  - with interesting variations in the output of the learned function occurring only along directions that lie on the manifold,
  - or with interesting variations happening only when we move from one manifold to another.

Manifold hypothesis

- We argue that in the context of AI tasks, such as those that involve processing images, sounds, or text, the manifold assumption is at least approximately correct.
- Obs1: the probability distribution over images, text strings, and sounds that occur in real life is highly concentrated
- 2. we can also imagine such neighbourhoods and transformations informally.
  - images
  - Images: we can think of transformations that allow us to trace out a manifold in image space: we can gradually dim or brighten the lights, gradually move or rotate objects in the image, gradually alter the colors on the surfaces of objects, etc.
  - It remains likely that there are multiple manifolds involved in most applications. For example, the manifold of images of human faces may not be connected to the manifold of images of cat faces.

- When the data lies on a low-dimensional manifold, it can be most natural for machine learning algorithms to represent the data in terms of coordinates on the manifold, rather than in terms of coordinates in $\mathcal{R}^n$.

Optimization (Chapter 4) Sec 4.3

Gradient-Based Optimization

- Most deep learning algorithms involve optimization of some sort.
- Optimization refers to the task of minimizing or maximizing some function $f(x)$ by altering x
  - Called the objective function (cost function, loss function)
- Derivative of $f(x)$ denoted as $f'(x)$or $\frac{dy}{dx}$
  - Gives slope of $f(x)$ at x
- We canthus reduce $f(x)$ by moving x in small steps with opposite sign of the derivative. This technique is called gradient descent

- For functions with multiple inputs, we must make use of the concept of partial derivatives. The partial derivative $\frac{\partial}{\partial x_i} f(x)$ measures how f changes as only the variable x_i increases at point x. The gradient generalizes the notion of derivative to the case where the derivative is with respect to a vector: the gradient of f is thevector containing all of the partial derivatives, denoted $\nabla_x f(x)$
- In multiple dimensions, critical points are points where every element of the gradient is equal to zero