Lecture 7 & 8

Expected Loss, Inductive and Transductive Learning, General approaches for Machine Learning

Aniket Nath, Diptarko Choudhury
(Group 3)

National Institute of Science Education and Research, Bhubaneswar
Homi Bhaba National Institute

January 24, 2023

Part I: Loss Function

Part II: Learning

Part III: General Approach for Machine Learning

Part I

Loss Function

Loss

Let $X \in \mathbb{R}^p$ denote a real-valued random input vector of dimension p.
$Y \in \mathbb{R}$, a real valued output variable.
We seek a map from $f(X) : \mathbb{R}^p \rightarrow \mathbb{R}$, for predicting Y for a given input X.
In general, the prediction for Y is : $f(X) = \hat{Y}$
The closer the quantity $\hat{Y}$ (prediction of output for X, to the original value of Y, the lesser the Loss.
The Loss function quantifies this closeness.
Which is basically a map from $\mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}$

$$L(Y,f(x)) \colon \mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}$$

Loss

Regret

In optimization problems like Machine Learning, we try to minimize the Loss function based on the concept of **Regret** in statistics.
If the Loss $L(Y, \hat{Y}) \to 0$ (where $f(x) = \hat{Y}$), then we claim that $f(X) \to Y$, and the optimization problem is said to be solved.

In Statistical decision-making, the goodness of any decision taken by the practitioner can be quantified using **Risk** function.

Definition 1.1

Risk Function Suppose one has to make a decision $\mathbf{d}$ *in a measurable decision space* $(\mathbf{D}, \mathbf{A})$ *with respect to a parameter* θ *based on a realization of a random variable* X *with values in a sampling space*$(\mathbf{E}, \mathcal{B}, P_\theta)$*,* $\theta \in \Theta$*. Further, let the Loss of a statistician caused by making the decision* d *when the random variable* X *follows the law* P_θ *be the Loss* $L(\theta, d)$*, where* L *is some loss function given on* $\Theta \times D$*. In this case, if the statistician uses a non-randomized decision function* $\delta : \mathbf{E} \to D$ *in the problem of decision making, then as a characteristic of this function* δ *the function.*[1]*:*

$$R(\theta, \delta) = E_\theta L(\theta, \delta(X)) = \int_{\mathbf{E}} L(\theta, \delta(X)) dP_\theta(x) \tag{1}$$

is used. It is called the risk function, or the risk of statistical procedure based on the decision function δ *with respect to the Loss* L*.*

[1] See *Risk of a statistical procedure - Encyclopedia of Mathematics* (2023).

Loss

Moving to Expected Loss

The definition **??** is defined for a generalize decision process. In case of Machine Learning, most of the above parameters can be held constant, and trivially a more simplified version of Risk can be given as.

$$
\begin{aligned}
R &= \mathbb{E}L(X, f(X)) && (2) \\
&= \int_X L(x, f(x)) dP(x); (x \in X) && (3)
\end{aligned}
$$

Where $f(X)$ is the decision function in our case.
This is the **Expected Loss** of the Loss function. The Loss is a single value, determined on a single point $x \in X$, in the entire data space. Whereas the Expected Loss is determined on the entire space. Expected Loss can be considered like a running average; given enough samples, it converges with the exact loss value for the entire space.

Expected Loss

The Expected Loss is a much more important concept than the average Loss itself, because it is directly associated with the Risk function in statistics. Hence, we can say that minimizing the Expected Loss will in turn, minimize the risk function itself, which signifies the best decision is taken and the problem is properly optimized, hence our previous claim that:

$$Loss \rightarrow 0, \; when \, f(X) \rightarrow Y$$

Part II

LEARNING

What is Learning?

Figure. Learning [2]

When we speak of **Learning**, what is it that we are actally refer to?
What defines Learning?
The answers to these questions have been pondered over by Philosophers, Psychologists and various thinkers for ages, and different people have different notions for it.

[2] Image taken from Google

What is Learning?

Machine Learning

In the context of Machine Learning, we can think of it as the capability of finding patterns in a given data, and then, predicting for a different data from similar distributions.

Types of Learning

Learning can be broadly classified into two types:

- Inductive Learning
- Transductive Learning

Types of Learning

Inductive Learning

Given a dataset, (X, Y), where $X \subset R^p$, and $Y \subset R$, inductive learning algorithm tries to learn from the, given dataset,the mapping from $X \rightarrow Y$, once the learning is complete, a new dataset X', where $X' \subset R^p$,($X' \not\subset X$) is fed to the algorithm to obtain, $\hat{Y}$ (prediction of the label value).
In this entire learning process, we have considered X' and X to belong to the same distribution.

Inductive Learning

Example I

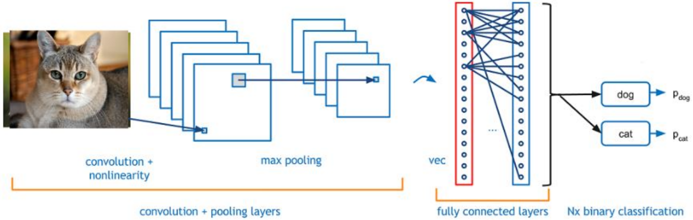

Figure. Cats and Dogs classification[3]

Let us consider, that we are building, a dogs and cat classifier. For the task, we are given grayscale images of dogs and cats, each with a specific label, either a dog or a cat. We consider, all the images to contain exactly a single dog, or a single cat.

We train a CNN[4] based model to understand the mapping between the images and the labels. We train a CNN based model, using gradient descent.

Now, we have a model M, which when given a sample $\tilde{x}$, can predict $\tilde{y}$, which is very close to the actual label value of $\tilde{x}$. A point to be noted is, $\tilde{x} \notin X$. In other words, the inference data need not to be part of the training data.

[3] Image taken from Google

[4] Convolutional Neural Network

Inductive Learning

Example II

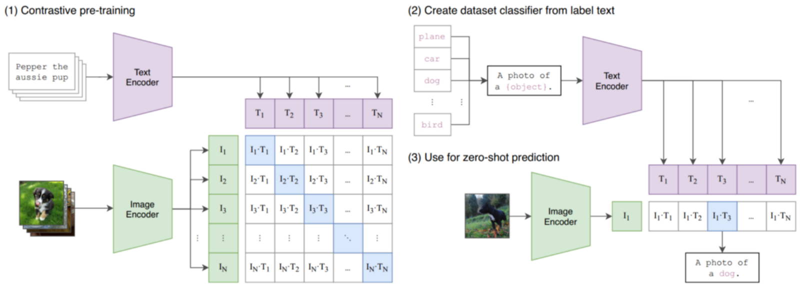
(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

Figure. OpenAI CLIP[5]

OpenAI CLIP[6] for image classification is an example from the semi-supervised domain of machine learning, which is often confused with transductive learning. The algorithm takes both labelled and unlabelled data to classify future data. Using labelled and unlabelled data simultaneously is a feature of transductive learning but CLIP will still be strongly considered as inductive, because inference data is not exposed to CLIP during training. Moreover, CLIP is a real model unlike transductive learning methods, which are Model less.

[5] Image taken from Google

[6] See Radford et al. (2021)

Types of Learning

Transductive Learning

Transductive Learning comes from the concept of transduction, which in the context of learning, refers to reasoning from specific observed instances (training) to specific observed instances (inference).
Given a dataset, (X, Y), where $X \subset R^p$, and $Y \subset R$, and another dataset, $\tilde{X} \subset R^p$, transductive learning algorithm tries to find specific behavioural similarities between elements of X and $\tilde{X}$, to predict $\tilde{Y}$, which is the label for $\tilde{X}$. The Transductive algorithm is fed , $X \cup \tilde{X}$ and Y.

Transductive Learning

Example I

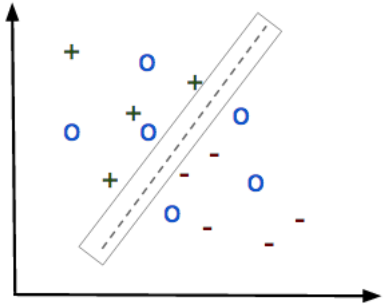

Figure. TSVM with and labelled and unlabelled data[7]

A classic example of transductive learning, which Vladimir Vapnik gives (1995)[8], is Transductive Support Vector Machines. The exact working of TSVMs is not in the context of this course, but a general idea is, having both the labelled and unlabelled datasets, the decision plane is chosen in such a way, that major clusters are formed, on it's both sides with the largest margin width. The exact nature of the clusters and the margin width depends on the amount of training(labelled) and inference(unlabelled) data, and the nature of the training data itself.

[7] Image taken from Google

[8] See Vapnik (2000)

Transductive Learning

Example II

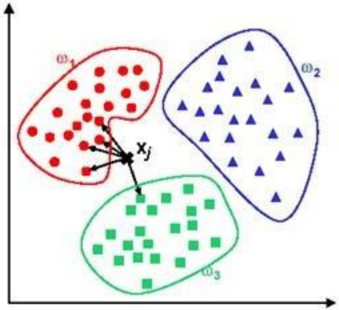

Figure. The kNN clustering[9]

Although the previous example is classic and precise, still it is bit ambiguous without the mathematical context. So for explanation, we provide a different example. The kNN[10] clustering algorithm can be thought of as a Transductive Learning algorithm, since every time a new datapoint is provided for classification, it tries to predict from the entire stored dataset. The algorithm is lazy in nature and moreover no model exists in this case. These are the two distinguishing nature of a transductive learning algorithm.

[9] Image taken from Google

[10] k-nearest neighbour

Part III

General Approach for Machine Learning

The general steps

Training a Machine Learning(ML) model, and making it converge to achieve state of the art results is, a difficult and a tedious task.
The first problem is acquiring, cleaning, pre-processing, splitting of data.
The next problem encountered while doing ML is regarding the choice of the Model. Once we have chosen a model, the next challenge is the architecture and the hyperparameters of the Model.

Definition 1.1 (Hyperparameters)

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning[11].

Finally, the Model needs to be finetuned, if necessary pruned and then deployed.

[11] See Nyuytiymbiy (2022)

Data

Acquiring Data

Data acquisition is the process of collecting data from source, which includes:

- Internet
- Simulations
- Scientific Data
- Surveys

Data

Data Cleaning

Data Cleaning refers to, the process of fixing or removing, incorrect, corrupted, incorrectly formatted, duplicate, incomplete data within a dataset.

Data

Pre-processing

Data pre-processing can refer to manipulation, dropping of data, before it is used, in order to ensure or enhance performance.

Data

Data Splitting

Data splitting refers to breaking a data into multiple parts, such that different data can be used from the same distribution, without data leaks.
In ML data is split into three parts:

- Training Dataset
- Validation Dataset
- Testing Dataset

As a rule of thumb, generally we choose the training dataset to be around 80% of the total dataset. The validation and test dataset are kept at around 10% each. The reason why we have three different datasets is, the Model is trained on the Training dataset and to prevent any form of model memorization creeping into our final result we test the Model on a different dataset. The validation dataset is used to check the accuracy or the metric of the Model. This dataset gives a rough picture of how our Model might perform. Any hyper-parameter tuning or decision regarding model training or architecture is taken on the basis of the scores obtained on this dataset. Finally, we have the test dataset which is used to check the Model's final performance after all the optimization and tuning are done. We choose to keep the test dataset even after the validation dataset because often the way we choose the hyperparameters and take other decisions might lead to bias in our Model, which might cause the Model's performance to inflate. The results of the test datasets are the ones which are published or reported.

Model Selection

The way models are selected, depends on:

- **Size of Dataset**: The larger is the training dataset, the larger models you can train on it. Large datasets generally have enough complexity in them, that they can be used to train complex models without running into the risk of overfitting.
- **Complexity of the Data**: More complex data needs larger models, data with a lot of features need wider models. Data with complex structures need deeper models.
- **problem at hand**: Different ML techniques need different types of Models, in the case of the classification of images, we can use simple CNN networks if we are working in the supervised domain, whereas if we change our domain to unsupervised, or semi-supervised, we might need to move to, variational auto-encoders, or generative adversarial nets.

Hyperparameters

Hyperparameter tuning plays a important role on model convergence and validation metrics. A professional might choose several ways to tune the hyperparameters, which might range from a few to thousands.

Some ways of tuning hyperparameters includes:

- **Random Search**
- **Grid Search**
- **Bayesian hyperparameter optimization**
- **Evolutionary hyperparameter optimization**

Active cycle of ML research

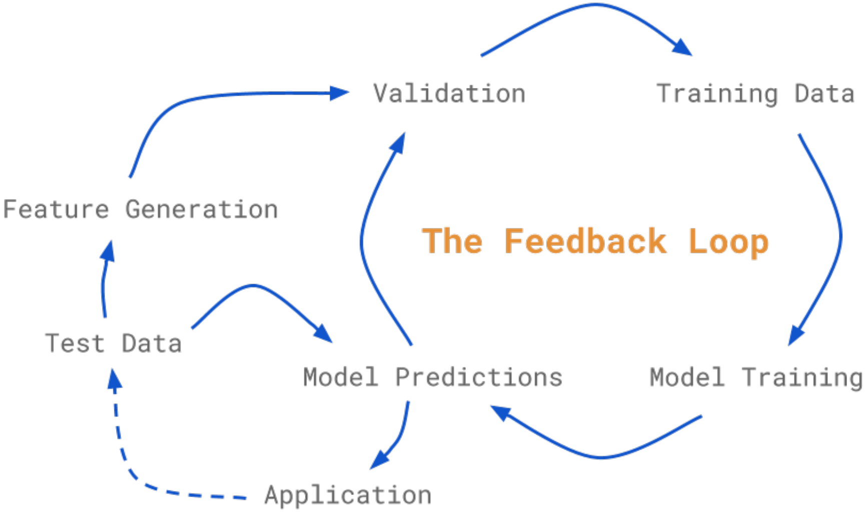

Figure. Cycle of Machine Learning[12]

[12] Image taken from Google

References I

- Nyuytiymbiy, Kizito (Mar. 2022). *Parameters and Hyperparameters in Machine Learning and Deep Learning*. en. URL: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac (visited on 01/24/2023).
- Radford, Alec et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. DOI: 10.48550/ARXIV.2103.00020. URL: https://arxiv.org/abs/2103.00020.
- *Risk of a statistical procedure - Encyclopedia of Mathematics* (2023). URL: https://encyclopediaofmath.org/wiki/Risk_of_a_statistical_procedure (visited on 01/21/2023).
- Vapnik, Vladimir N. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer. ISBN: 978-1-4419-3160-3 978-1-4757-3264-1. DOI: 10.1007/978-1-4757-3264-1. URL: http://link.springer.com/10.1007/978-1-4757-3264-1 (visited on 01/24/2023).