COMP 642 — Machine Learning Mar 22, 2022

Lecture 18

Lecturer: Anshumali Shrivastava Scribe By: Haixiao Wang, Baodong Chen, Jeffery Dong, Linfeng Lou

Disclaimer: *These lecture notes are intended to develop the thought process and intuition in machine learning. The materials are not thoroughly reviewed and can contain errors.*

1 Self and Semi-Supervised Learning

1.1 The definition of supervised learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Actually, most of the machine learning like logistic regression and neural network can be allocated into the category of supervised learning.

1.2 The definition of unsupervised learning

Unsupervised learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabeled data. The simplest example is clustering. For example, a dataset for animals and there are a lot of variations among them that are not known. It can help us to know more about the data itself. For example, we can use an auto-encoder to filter the noise from the image.

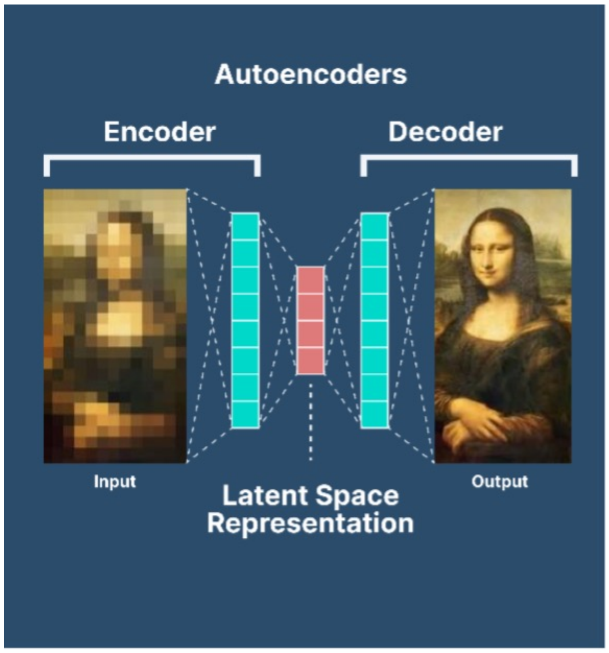

Figure 1: Example of Auto-encoders

1.3 The definition of semi-supervised learning

Semi-supervised learning is an approach to machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. It seems to be something between supervised learning and unsupervised learning. For example, you are given some labeled data and unlabeled data. The task is supervised learning, which is to predict the Yi given Xi. A key point in semi-supervised learning is about how we can better use the unlabeled dataset.

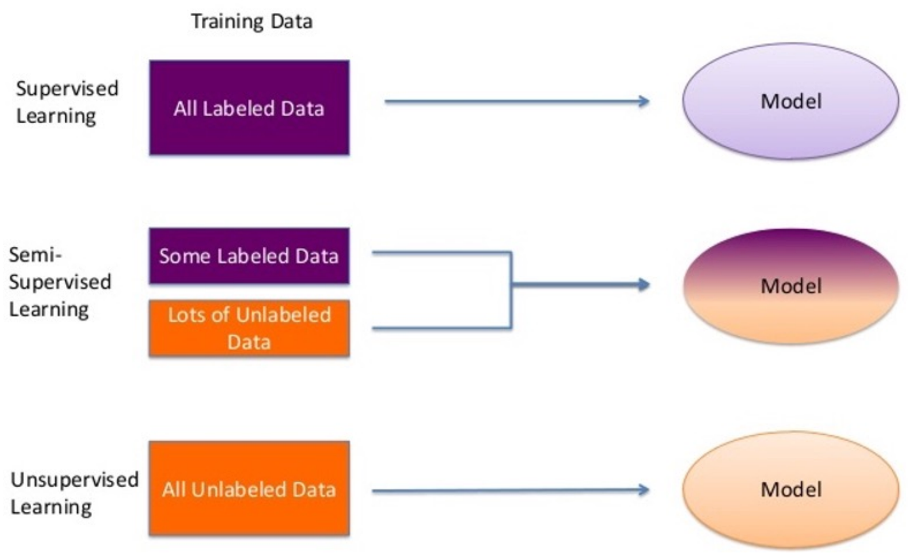

Figure 2: Different types of machine learning

1.4 Process of semi-supervised learning

The first idea is to generate the pseudo label using the trained data, which is also called self-learning or annotation. This means that we start with labeled data and implement supervised learning and get a classifier. Then, we use this classifier to get high confidence predictions on unlabeled data and put them into labeled dataset and continuously do that. Let me illustrate the idea by the following example.

Given 100 samples of labeled data and 1,000,000 samples of unlabeled data. 1. Train classifier F0 on 100 samples of labeled data. 2. Using the classifier F0 to train 1000 samples of unlabeled data and optimized model and get F1. 3. Using F1 to train on another 1000 samples of unlabeled data and continuously do that until the model does not change.

1.5 The potential problem with the semi-supervised learning

– We tell the model that the high confidence prediction on an unlabeled dataset is the actual real label so that the model can change itself. Under this circumstance, if the initial confidence prediction is wrong, the model will get worse after several iterations. – If the training data is biased, the self-training process will reinforce bias. For example. If you have a lot of dog images in labeled data, you are supposed to confidently predict dogs in the unlabeled dataset.

2 Co-training

Co-training is a machine learning algorithm used when there are only small amounts of labeled data and large amounts of unlabeled data. It is a semi-supervised learning technique that requires two views of the data. It assumes that each example is described using two different sets of features that provide complementary information about the instance. Ideally, the two views are conditionally independent (i.e., the two feature sets of each instance are conditionally independent given the class) and each view is sufficient (i.e., the class of an instance can be accurately predicted from each view alone). Co-training first learns a separate classifier for each view using any labeled examples. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data.

2.1 The loss function for co-training

The loss function for two different classifiers in co-training that need to be minimized is the cross entropy loss of F1 on the labeled dataset plus the cross entropy loss of F2 on the labeled dataset plus the disagreement on the unlabeled dataset. The knowledge that actually contribute to the model training is those that the two classifiers make different predictions, and for those which two classifiers make the same predictions, there is actually an issue that they may both make the same mistakes.

3 HybridNet

When dealing with smaller datasets,however, the need for proper regularization methods becomes more crucial to control overfitting. An appealing direction to tackle this issue is to take advantage of the huge number of unlabeled data by developing semi-supervised learning techniques.There is a model called HybridNet which is designed by this thinking. It consists in a "hybrid" auto-encoder with the feature extraction path decomposed into two branches.

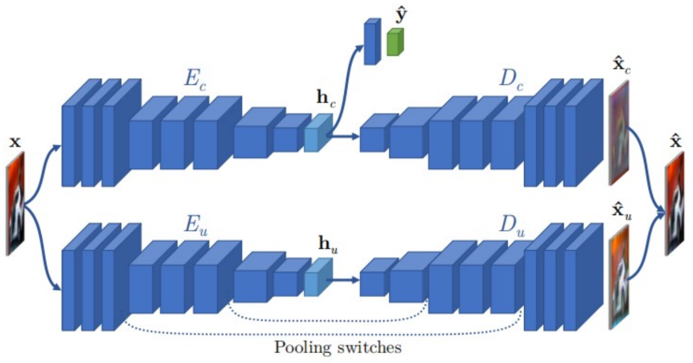

Figure 3: Example of HybridNet architecture

The top branch encoder, of parameters W_c, is connected to a classification layer that produces class predictions while the decoder from this branch is used to partly reconstruct the input image from the discriminative features, leading to $\hat{x}_c$. Since those features are expected to extract invariant class-specific patterns,information is lost and exact reconstruction is not possible. To complement it, a second encoder-decoder branch of parameters W_u is added to produce a complementary reconstruction $\hat{x}_u$ such that the sum $\hat{x} = \hat{x}_c + \hat{x}_u$ is the final complete reconstruction.

During training, the supervised classification cost impact W_c while an unsupervised reconstruction cost is applied to both W_c and W_u to properly reconstruct the input image. The main assumption behind HybridNet is that the two-path architecture helps in making classification and reconstruction cooperate. To encourage this, additional costs and training techniques is applied, namely a stability regularization in the discriminative branch and a branch complementarity training method.

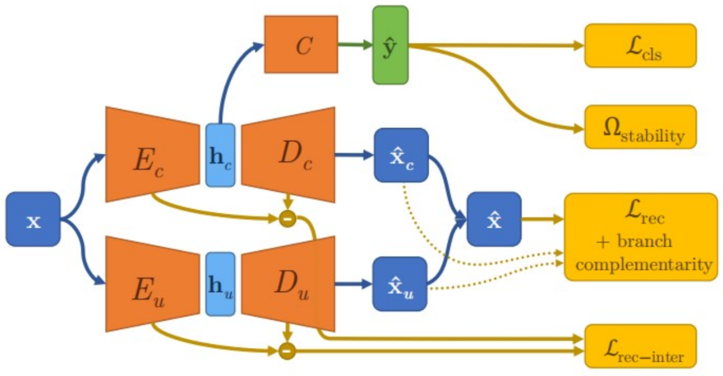

Figure 4: General description of the HybridNet framework

As we have seen, classification requires intra-class invariant features while reconstruction needs to retain all the information. To circumvent this issue, HybridNet is composed of two auto-encoding paths, the discriminative path (E_c and D_c) and the unsupervised path (E_u and D_u). Both encoders E_c and E_u take an input image x and produce representations h_c and h_u, while decoders D_c and D_u take respectively h_c and h_u as input to produce two partial reconstructions $\hat{x}_c$ and $\hat{x}_u$. Finally, a classifier C produces a class prediction using discriminative features only: $\hat{y} = C(h_c)$. Even if the two paths can have similar architectures, they should play different and complementary roles. The discriminative path must extract discriminative features h_c that should eventually be well crafted to perform a classification task effectively, and produce a purposely partial reconstruction $\hat{x}_c$ that should not be perfect since preserving all the information is not a behavior we want to encourage. Consequently, the role of the unsupervised path is to be complementary to the discriminative branch by retaining in h_u the information lost in h_c. This way, it can produce a complementary reconstruction $\hat{x}_u$ so that, when merging $\hat{x}_u$ and $\hat{x}_c$, the final reconstruction $\hat{x}$ is close to x. The HybridNet architecture, visible on Fig. 2, can be described by the following equations:

$$h_c = E_c(x) \quad \hat{x}_c = D_c(h_c) \quad \hat{y} = C(h_c)$$
$$h_u = E_u(x) \quad \hat{x}_u = D_u(h_u) \quad \hat{x} = \hat{x}_c + \hat{x}_u$$

Note that the end-role of reconstruction is just to act as a regularizer for the discriminative encoder. The main challenge and contribution is to find a way to ensure that the two paths will in fact behave in this desired way. The two main issues that we tackle are the fact that we want the discriminative branch to focus on discriminative features, and that we want both branches to cooperate and contribute to the reconstruction. Indeed, with such an architecture,we could end up with two paths that work independently: a classification path $\hat{y} = C(E_c(x))$ and a reconstruction path $\hat{x} = \hat{x}_u = D_u(E_u(x))$ and $\hat{x}_c = 0$. Both of those issues could be addressed through the design of the architecture of the encoders and decoders as well as an appropriate loss and training procedure.