Derivation of Semi-Supervised Loss Terms

Matthew Scott

October 27, 2020

1 Introduction

This writing aims to unify loss terms under some consistent logic. It takes a simplified mathematical model of neural nets with their input and feature spaces, and uses the vague notion of similarity to justify losses. In the most general sense, if two items are similar $a \sim b$ then given a metric in the feature space we immediately have a loss term $d(\text{features}(a), \text{features}(b))$ which we aim to minimize. What follows is therefore a categorisation of similarity relations. The losses that are included here come from The Energy Clustering Notes by Prof. Oberman, the MixMatch paper[2] as well as Mido's paper[1].

The main takeway of this writing is to separate loss terms into two categories: Contrastive and Wrapped.

1. "Contrastive": Loss terms that aims to learn a metric in the feature space by training similar elements to be close to each other, and dissimilar elements far from one another.
2. "Wrapped": Loss terms that identify a target in the feature space and attempts to map elements as closely as possible to the target.

2 Notation

Spaces:

X: The input space. It is a vector space. This is the set of all possible data that could be fed to the network, not only the data collected for training.
Y: The space of labels. It has no special structure. This space if finite, of dimension d.
F: The feature space. It is a vector space. This is an intermediate space which attempts to capture the essential features of X in a lower dimensional space.
Θ: The parameter space. It is a vector space. It contains all possible parameterisation of the neural net, and so fully determines the workings of our net. It is over this space that the learning is done.

Functions:
$d : F \times F \to \mathbb{R}$ A metric on the feature space, which is usually the euclidean norm.
$I : Y \to F$ embedding function for the labels into the feature space.
$h : X \times \Theta \to F$. The encoding function. This is the function that finds the feature vectors for a given sample x in the input space. It is for the purposes of this writing the full neural net.
Data:
X_l the set of labeled samples that we have to learn.
$f : X_l \to Y$ The labelling function.

$U \in X$ the set of unlabeled samples that we have at our disposition to learn.

Additionally, we need functions that will enable us to do Data Augmentation. Define a class of functions which are label-preserving $\Psi = \{\psi : X \to X | \forall x \in X, f(x_1) = f(\psi(x_1))\}$. These functions can be found by using some symmetries known in X to perform transformations on X which are class-preserving. Example: If we try to learn a function to differentiate pictures of cats and dogs, then flipping the images is label-preserving. We find this because of a symmetry in the data set: flipping images in that way does not change which animal is in the picture.
These symetries are dataset-dependent. For example, the MNIST dataset has no flipping symetry, since flipping a number does not preserve the label of a number. A flipped 3 is no longer a 3.

3 Contrastive Data

A contrastive loss term takes the form: $d(h(x_1, \theta), h(x_2, \theta))$ for $x_1, x_2 \in X, \theta \in \Theta$. To construct such a term, we need some notion of similarity between x_1 and x_2: $x_1 \sim x_2$ in order to know that their feature vectors should be close (minimize $d(h(x_1, \theta), h(x_2, \theta))$). Therefore we will define similarity relations $\sim$ where this relation is a subset of $X \times X$. We will re-use the $\sim$ symbol for every definition. To find loss terms of this sort, we therefore find notions of similarity.

1. Similarity through Data Augmentation:

$$x_1 \sim x_2 \iff \exists \psi \in \Psi : x_2 = \psi(x_1)$$

2. Generalisation of similarity through Data Augmentation:

$$x_1 \sim x_2 \iff \exists x_3 \in X \text{ and } \psi_1, \psi_2 \in \Psi : x_1 = \psi_1(x_3) \wedge x_3 = \psi_2(x_3)$$

   Notice that this is a strict generalization of the first notion of similarity by considering that the identity function is in Ψ.

3. Similarity by Proximity in the Input Space:
   For some $\delta > 0$ and a metric on X $d' : X \times X \to \mathbb{R}$,

$$x_1 \sim x_2 \iff d'(x_1, x_2) < \delta$$

   This type of notion of similarity is used when randomly perturbing the input data.

4. Similarity through Identical Labelling:
   For $x_1, x_2 \in X_l$(i.e. x_1 and x_2 are labeled samples), then

$$x_1 \sim x_2 \iff f(x_1) = f(x_2)$$

   This notion of similarity results in the SUNCET loss proposed by Mido[1].

4 Wrapped Data

To find a Wrapped loss term, we need a sample $x \in X$, a target $c \in F$ which have some similarity relation $x \approx c$. Given a metric d in F, we which then have a loss term $d(h(x,\theta),c)$. In this section we discuss the ways that $x \in X$ can be obtained, and which targets c can be related to the input. We discuss similarity relations from input data to targets with the symbol $\approx$ which denotes a relation in $X \times F$. We will re-use the $\approx$ symbol for every definition.

1. Supervised Learning Target:
   Let $x \in X_l$.
   $$x \approx c \iff c = E(f(x))$$

2. Supervised Learning Through Input Similarity:
   For $x_1 \in X$. Take $\sim$ as any of the defined similarity relations in the Contrastive section. We define
   $$x_1 \approx c \iff \exists x_2 \in X_l : x_1 \sim x_2 \wedge I(f(x_2)) = c$$
   Supervised Learning with Data augmentation is obtained by taking $\sim$ to be defined in # 1 of the Contrastive section.

3. Initialisation Wrapping (Center Targets in Clustering Energy Notes):
   $\forall x \in U$ we fix the target to be the nearest target at initialisation. Let (θ_t) be the sequence of parameter vectors during the learning process.
   $$x \approx \text{ at time t } \iff c = I(\arg\min_y \{d(h(x,\theta_t), I(y))\})$$
   It is worth noting that this method of choosing a target is essentially random modulo some smoothness in the assignmnent of targets at initialisation. Indeed, once a target is chosen at initialisation, it is unlikely that it will change since training attempts to reinforce the initial target. The main motivation to use this target is that targets will be common to many data points, and so it will lead to clustering of the data in feature space.
   This type of relation corresponds to the the centers specified in the Cluster Energy notes provided by Prof. Oberman.

4. Combination of Similarity Relations:
   Given two similarity relations $\approx_1, \approx_2$ we define their combination $\approx$ with the logical "or":
   $$\forall x \in X, c \in F \quad x \approx c \iff [x \approx_1 c] \vee [x \approx_2 c]$$
   This can trivially be generalised to combine any number of similarity relations together.

5. Convex Combination(MixUP algorithm):
   For any similarity relation $\approx^*$ let $T_{\approx^*} = \{(x,c) \in X \times F | x \approx^* c\}$, that is the set of pairs which obey the similarity relation $\approx^*$. Note that $\approx^*$ may be any one of the similarity relations defined above(including combinations of them as defined in 4.). Then we define the relation:
   $$x \approx c \iff \exists S \in T_{\approx^*}, |S| < \infty, \{\lambda_i\}_{i=1}^{|S|} \in \mathbb{R}^{|S|} : \left[\sum_{i=1}^{n} \lambda_i = 1\right] \wedge \left[x = \sum_{i=1}^{|S|} \lambda_i x_i\right] \wedge \left[c = \sum_{i=1}^{|S|} \lambda_i c_i\right]$$

The idea of this relation is that relations between inputs and targets can be extended by convex combinations. This is the relation corresponding to the loss defined for MixUp loss used in MixMatch[2].

5 Motivation For the Loss Terms

The motivation for the terms introduced above are (as far as I can tell) to:

1. Cluster Data (essentially no matter what)
   This is most apparent in the less intuitive wrapping mechanisms. It seems like for a neural net wrongly clustered data allows for easier correction by fine-tuning than starting from unclustered data. That is, even while learning the wrong clustering, some information is picked up somehow. The abstract of [3] indicates that this seems to be true to some extent.
   Remark: the extent to which this motivation applies relative to "Cluster Similar Data"(next item on the list) is often not very clear.
   The terms which do this are: Wrapped #3.

2. Cluster Similar Data
   This appears first and foremost in the Contrastive loss terms. This is also done indirerctly through wrapping methods by training similar data to a common target, which in effect should cluster the data
   The terms which do this are: Contrastive #1,2,3,4 and Wrapped #3,5

3. Map Data to Correct Labels
   This happens with the supervised learning losses or the ones closely related to supervised learning losses (Ex: data-augmented supervised learning). These loss terms are the ones used in the fine-tuning process for in the pre-processing/fine-tuning paradigm.
   The terms which do this are: Wrapped #1,2

6 Possible Improvements

1. The notion of "similarity" in this writing investigates if there is any reason to associate two pieces of data together. It is a binary choice (are they related or not). A great improvement would be to add the notion of strength to these similarities by finding some kind of ordering or bounds on results. Needless to say this looks much more challenging, and can probably only be done by focussing on each loss term individually.

2. Similarities in the feature space could and should be added, such as when a target at initialisation is projected to a label center.

3. In the sections above we have assumed for simplicity the existence of a single feature space F. This is very often inaccurate in practice, where often the Contrastive loss terms will operate in a different space from the Wrapper methods. The Feature space will be located in a earlier layer of the neural net, whereas the target elements will be in the logit space, that is the space of the output of the last layer of the net before softmax is applied. An improvement (which also makes this writing more complex) would be to generalise by treating two different spaces for the features and targets.

References

[1] Mahmoud Assran et al. *Recovering Petaflops in Contrastive Semi-Supervised Learning of Visual Representations*. 2020. arXiv: 2006.10803 [cs.LG].

[2] David Berthelot et al. "MixMatch: A Holistic Approach to Semi-Supervised Learning". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 5049–5059. URL: http://papers.nips.cc/paper/8749-mixmatch-a-holistic-approach-to-semi-supervised-learning.pdf.

[3] Vinaychandran Pondenkandath et al. *Leveraging Random Label Memorization for Unsupervised Pre-Training*. 2018. arXiv: 1811.01640 [cs.LG].