Label-Free Supervision of Neural Networks with Physics and Domain Knowledge

Russell Stewart , Stefano Ermon
Department of Computer Science, Stanford University
{stewartr, ermon}@cs.stanford.edu

Abstract

In many machine learning applications, labeled data is scarce and obtaining more labels is expensive. We introduce a new approach to supervising neural networks by specifying constraints that should hold over the output space, rather than direct examples of input-output pairs. These constraints are derived from prior domain knowledge, e.g., from known laws of physics. We demonstrate the effectiveness of this approach on real world and simulated computer vision tasks. We are able to train a convolutional neural network to detect and track objects without any labeled examples.

Introduction

Applications of machine learning are often encumbered by the need for large amounts of labeled training data. Neural networks have made large amounts of labeled data even more crucial to success [8]. Nonetheless, we observe that humans are often able to learn without direct examples, opting instead for high level instructions for how a task should be performed. We ask whether a similar principle can be applied to teaching machines; can we supervise networks without individual examples by instead describing only the structure of desired outputs?

Unlike other forms of unsupervised learning, such as Autoencoders, we explicitly provide the semantics of the hidden variables we hope to discover. We train without labels by learning from constraints [13]. Intuitively, algebraic and logical constraints are used to encode structures and relationships that are known to hold because of prior domain knowledge. The priary contribution of this work is to demonstrate how encoding physics constraints may be used to supervise neural networks across practical computer vision tasks.

Problem Setup

In a traditional supervised learning setting, we are given a training set $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ of n training examples. Each example is a pair (x_i, y_i) formed by an instance $x_i \in X$ and the corresponding output (label) $y_i \in Y$. The goal is to learn a function $f : X \to Y$ mapping inputs to outputs. To quantify performance, a loss function $\ell : Y \times Y \to \mathbb{R}$ is provided, and a mapping is found via

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \ell(f(x_i), y_i) \tag{1}$$

where the optimization is over a pre-defined class of functions $\mathcal{F}$ (hypothesis class). In our case, $\mathcal{F}$ will be (convolutional) neural networks parameterized by their weights. The loss could be for example $\ell(f(x_i), y_i) = 1[f(x_i) \neq y_i]$.

While one clearly needs labels, y, to evaluate f^*, labels may not be necessary to discover f^*. If prior knowledge informs us that outputs of f^* have other unique properties among functions in $\mathcal{F}$,

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

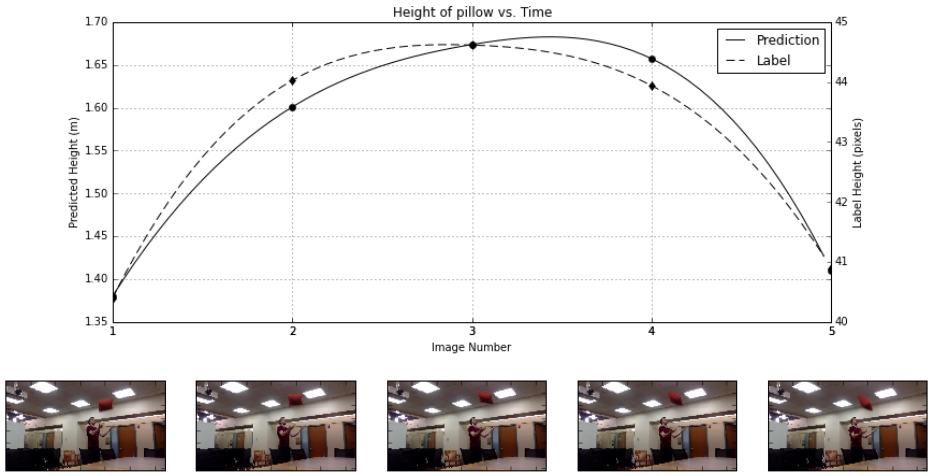

Figure 1: As the pillow is tossed, the height forms a parabola over time. We exploit this structure to independently predict the pillow's height in each frame without providing labels.

we may use these properties for training rather than direct examples y. Specifically, we consider an unsupervised approach where the labels y_i are not provided to us, and optimize for a necessary property of the output, g instead. That is, we search for

$$\hat{f}^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} g(x_i, f(x_i)) + R(f) \tag{2}$$

The process of designing the loss g and the regularization term R is a form of supervision, and can require a significant time investment. But unlike hand labeling, it does not increase proportional to the size of the data, $|D|$, and can be applied to new data sets often without modification.

Experiments

We record videos of a pillow being thrown across the field of view. Our goal is to obtain a regression network predicting the height of the pillow in an image. We will train the network as a structured prediction problem operating on a sequence of N images to produce a sequence of N heights, $\left(R^{\text{height}\times\text{width}\times 3}\right)^N \rightarrow \mathbb{R}^N$, and each piece of data x_i will be a vector of images, $\mathbf{x}$. Rather than supervising our network with direct labels, $\mathbf{y} \in \mathbb{R}^N$, we instead supervise the network to find an object obeying the elementary physics of free falling objects.

An object acting under gravity will have a fixed acceleration of $a = -9.8m/s^2$, and the plot of the object's height over time will form a parabola:

$$\mathbf{y}_i = y_0 + v_0(i\Delta t) + a(i\Delta t)^2$$

This equation provides a necessary constraint, which the correct mapping f^* must satisfy. We thus train f by making incremental improvements in the direction of better satisfying this equation.

Given any trajectory of N height predictions, $f(\mathbf{x})$, we fit a parabola with fixed curvature to those predictions, and minimize the resulting residual. Formally, we specify $\mathbf{a} = [a\Delta t^2, a(2\Delta t)^2, \ldots, a(N\Delta t)^2]$ and set

$$\hat{\mathbf{y}} = \mathbf{a} + \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(f(\mathbf{x}) - \mathbf{a}) \tag{3}$$

where

$$\mathbf{A} = \begin{bmatrix} \Delta t & 1 \\ 2\Delta t & 1 \\ 3\Delta t & 1 \\ \vdots & \vdots \\ N\Delta t & 1 \end{bmatrix}$$

The constraint loss is then defined as

$$g(\mathbf{x}, f(\mathbf{x})) = g(f(\mathbf{x})) = \sum_{i=1}^{N} |\hat{\mathbf{y}}_i - f(\mathbf{x})_i|$$

where we note that the vector $\hat{\mathbf{y}}$ from (3) is a function of the predictions $f(\mathbf{x})$, rather than ground truth labels. Because g is differentiable almost everywhere, we can optimize equation (2) with SGD. Surprisingly, we find that when combined with existing regularization methods for neural networks, this optimization is sufficient to recover f^* up to an additive constant C (specifying what object height corresponds to 0). Qualitative results from our network applied to fresh images after training are shown in Fig. (1)

Training details

Our data set is collected on a laptop webcam running at 10 frames per second ($\Delta t = 0.1s$). We fix the camera position and record 65 diverse trajectories of the object in flight, totalling 602 images. For each trajectory, we train on randomly selected intervals of $N = 5$ contiguous frames. Images are resized to 56×56 pixels before going into a small, randomly initialized 4 layer neural network with no pretraining. We train with a learning rate of 0.0001 for 4,000 iterations.

Evaluation

For evaluation, we manually labeled the height of our falling objects in pixel space. Labeling the true height in meters requires knowing the object's distance from the camera, so we instead evaluate by measuring the correlation of predicted heights with ground truth pixel measurements. Our network yields 90.1% correlation. For comparison, we also trained a supervised network on the labels to directly predict the height of the object in pixels. This network achieved a correlation of 94.5%, although this task is somewhat easier as it does not require the network to compensate for the object's distance from the camera.

This experiment demonstrates that one can teach a neural network to extract object information from real images by writing down only the equations of physics that the object obeys. Further experiments can be found in the full publication at `https://arxiv.org/abs/1609.05566`.

Related Work

In this work, we have presented a new strategy for incorporating pysics domain knowledge in computer vision tasks. The network in our experiment learns without labels by exploiting high level instructions in the form of constraints.

Constraint learning is a generalization of supervised learning that allows for more creative methods of supervision. For example, multiple-instance learning as proposed by [3] allows for more efficient labeling by providing annotations over groups of images and learning to predict properties that hold over at least one input in a group, rather than providing individual labels. In rank learning, labels may given as orderings between inputs with the objective being to find an embedding of inputs that respects the ordering relation [6]. Various types of constraints have also been used extensively to guide unsupervised learning algorithms, such as clustering and dimensionality reduction techniques [9, 1, 14, 4]. Natural language processing has seen many successful applications of constraint learning [10, 2, 5], and the recent work of [12] has provided a fresh perspective on the idea of learning with labeling functions, rather than labels, in the form of Data Programming.

The use of constraint learning for neural networks provides further advantages, as reductions in labeling effort are more impactful when feature engineering is also not required. Applications of constraint learning to neural networks have been suggested by several recent works. In [7], deep

networks were trained to predict sentiment labels of individual sentences in a review set based on constraints for the final review score. [11] and [15] trained deep convolutional neural networks to construct high level compressed embeddings of images without using labels. In [11], constraints such as invariance of embeddings to image rotations, high entropy outputs, and high standard deviation outputs were encoded to learn these embeddings. Our experiments build on these ideas in a context where we can use prior knowledge such as physical dynamics to further constrain the output's semantics.

Conclusion

We have introduced a new method for using physics and other domain constraints to supervise neural networks. By freeing the operator from collecting labels, our small scale experiments show promise for the future of training neural networks with weak supervision.

References

[1] S. Basu, I. Davidson, and K. Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.

[2] M.-W. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting-Association for Computational Linguistics*, volume 45, 2007.

[3] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1), 1997.

[4] S. Ermon, R. Le Bras, S. K. Suram, J. M. Gregoire, C. P. Gomes, B. Selman, and R. B. van Dover. Pattern decomposition with complex combinatorial constraints: Application to materials discovery. In *AAAI*, 2015.

[5] K. Ganchev, J. Gillenwater, B. Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul), 2010.

[6] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[7] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth. From group to individual labels using deep features. In *ACM SIGKDD*, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[9] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[10] P. Liang, M. I. Jordan, and D. Klein. Learning from measurements in exponential families. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009.

[11] K. Lin, J. Lu, C.-S. Chen, and J. Zhou. Learning compact binary descriptors with unsupervised deep neural networks.

[12] A. Ratner, C. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. *arXiv preprint arXiv:1605.07723*, 2016.

[13] I. Shcherbatyi and B. Andres. Convexification of learning from constraints. *arXiv preprint arXiv:1602.06746*, 2016.

[14] W. Zhi, X. Wang, B. Qian, P. Butler, N. Ramakrishnan, and I. Davidson. Clustering with complex constraints-algorithms and applications. In *AAAI*, 2013.

[15] B. Zhuang, G. Lin, C. Shen, and I. Reid. Fast training of triplet-based deep binary embedding networks. *arXiv preprint arXiv:1603.02844*, 2016.