Gradient-based Algorithms for Machine Teaching

Pei Wang
UC, San Diego
pew062@ucsd.edu

Kabir Nagrecha
UC, San Diego
kabir.nagrecha@gmail.com

Nuno Vasconcelos
UC, San Diego
nuno@ucsd.edu

Abstract

The problem of machine teaching is considered. A new formulation is proposed under the assumption of an optimal student, where optimality is defined in the usual machine learning sense of empirical risk minimization. This is a sensible assumption for machine learning students and for human students in crowdsourcing platforms, who tend to perform at least as well as machine learning systems. It is shown that, if allowed unbounded effort, the optimal student always learns the optimal predictor for a classification task. Hence, the role of the optimal teacher is to select the teaching set that minimizes student effort. This is formulated as a problem of functional optimization where, at each teaching iteration, the teacher seeks to align the steepest descent directions of the risk of (1) the teaching set and (2) entire example population. The optimal teacher, denoted MaxGrad, is then shown to maximize the gradient of the risk on the set of new examples selected per iteration. MaxGrad teaching algorithms are finally provided for both binary and multiclass tasks, and shown to have some similarities with boosting algorithms. Experimental evaluations demonstrate the effectiveness of MaxGrad, which outperforms previous algorithms on the classification task, for both machine learning and human students from MTurk, by a substantial margin.

1. Introduction

The success of deep learning has been driven, in large part, by the availability of large and carefully curated datasets for tasks such as image recognition [8, 22], action recognition [19, 11], object detection [24], etc. These datasets usually contain everyday objects, actions, or scenes and can be scalably annotated on crowdsourcing platforms such as Amazon Mechanical Turk (MTurk). This is, however, usually not true for expert domains, such as biology or medical imaging. While data collection can still be easy in these domains, annotations require highly specialized and domain specific knowledge. This is beyond the reach of crowdsourcing annotators. On the other hand, annotation by specialists is usually too expensive and rarely feasible at a large scale.

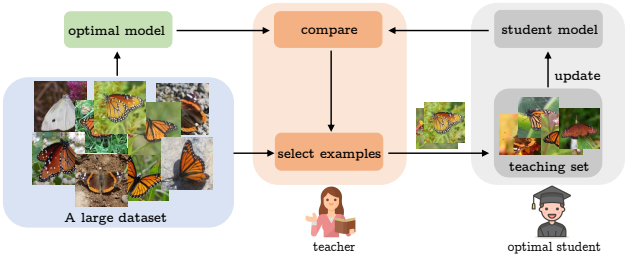


Figure 1: MaxGrad is an iterative machine teaching algorithm. At each iteration, the teacher selects new examples from a large dataset, to complement a small teaching set used by a student to learn a task. Examples are selected by comparing the current student model to the optimal model for the large dataset. The optimal student uses the teaching set to update his/her model and feeds back its predictions to the teacher.

This has motivated extensive research in alternative and less label-intensive forms of learning, including few-shot learning [42, 1], transfer learning [21, 46], semi-supervised learning [47, 23], and self-supervised learning [35, 20]. However, these approaches usually underperform supervised learning from large and fully labeled datasets. In result, there has recently been interest in machine teaching algorithms capable of training crowdsource annotators to label data from specialized domains.

The goal of machine teaching is to design systems that can teach students efficiently and automatically. Machine teaching is a broad research problem [52], where humans can utilize domain knowledge to teach machines or vice-versa. In this work, we restrict the discussion to the narrow task of image classification, where a machine teaches human learners to discriminate between different image classes. Although the proposed ideas are general, we target the application of teaching image annotators in crowd-sourcing platforms. This exploits the fact that a relatively small annotated dataset can be leveraged to train crowd workers, which can then annotate large numbers of images, enabling scalable supervised learning of image classifiers. While classification has been the task of choice for much machine teaching work, it should be noted that several other tasks and applications have also been investigated [6, 5, 44].

The machine teaching set-up considered in this work is

the iterative interaction set-up of Figure 1. At each iteration, the teacher selects new examples from a large dataset, to complement a small set of examples, known as the *teaching set,* which is used by the student to learn the target task. By comparing the current student model and the optimal model for the large dataset, the teacher seeks to select the examples that most help the student learn. The central question in this set-up is how to select the teaching set. Ideally, this set should pack as much information for class discrimination as possible into the smallest number of examples.

In the literature, there have been many attempts to design optimal teaching algorithms [39, 26, 27, 30]. This usually requires the assumption of certain student properties. Although past works have proposed different student models, these frequently rely on assumptions that are questionable for the crowdsource annotation context. For example, a popular assumption [39, 29, 7] is that the student only has access to a countable set of hyperplane hypotheses. While justified by the fact that human students have limited ability and memory, this assumption overly underestimates their learning ability. In fact, several machine teaching works explicitly assume that students have limited capacity or are otherwise sub-optimal learners [30, 18, 39, 50]. This is not supported by studies with real students, which found that humans have strong learning ability [13, 9, 15, 38].

In this work, we assume that the student is an optimal learner. Optimality is defined in the standard machine learning sense, i.e. that the student learns a predictor of minimum empirical risk in the teaching set. This always holds for machine learning students, which are defined in this way, and is sensible for human students, who usually do not underperform machine learning students, especially on few-shot learning scenery in practice. It does assume that students are engaged in the learning task, i.e. giving their best effort. This is sensible in the crowdsourcing scenario, where students are free-willing participants rated by their task performance. We show that, if allowed unbounded effort, the optimal student will always learn the optimal predictor for the task. This implies that the only role of the teacher is to optimize learning speed, i.e. select the teaching examples that enable the student to learn with least effort.

We then formulate the search for the optimal teacher as a problem of functional optimization where, at each teaching iteration, the teacher aims to align the steepest descent direction of the teaching set risk with that of the empirical risk over the entire example population. This is shown to have as optimal solution the *MaxGrad* teacher, which maximizes the gradient of the risk on the set of new examples selected per iteration. MaxGrad teaching algorithms are finally provided for both binary and multiclass tasks, and shown to have some similarities with boosting algorithms [33, 12, 34]. Experimental evaluations demonstrate the effectiveness of MaxGrad, which outperforms previous algorithms on the classification task, for both machine learning and human students from MTurk.

2. Related work

Simulated studies: In the past two decades, a variety of algorithms have been proposed to model the teacher-student interaction and seek the optimal teaching sequence. [3] explored several heuristics for the selection of the teaching set, based on insights derived from active learning, including a preference for points closest to the boundary, a handcrafted indicator of classification difficulty, curriculum learning, and a coverage model. [31] explored the use of recurrent neural networks as models of student learning. [51] modeled student learning as a Bayesian update process. [2, 32] used reinforcement learning based models to develop teaching policies for computer-based tutoring systems. All these methods have been developed and evaluated with synthetic data or handcrafted features, and did not explore the teaching of human learners with natural images. Note that there are some related algorithm families to machine teaching, including active learning [37, 45], few-shot learning [42, 41], curriculum learning [4, 14] and knowledge distillation [17]. For example, the main difference from active learning is that in the latter the learner selects examples without knowing the ground truth. In machine teaching, examples are selected by the teacher, who knows all labels. We recommend [26, 52] for extensive comparisons.

Human studies: Most of existing literature on human evaluations only work on simple binary classification problem [50, 39, 40]. A representative is STRICT [39]. It simulates the student as a hyperplane in a finite hypothesis space. The learning process is modeled as a Markov chain, assuming that learners perform a random walk in hypothesis space, according to the teacher's feedback. Expected error rate is the criterion for teaching set selection. Since its minimization is NP-hard, a surrogate objective is optimized in a greedy manner. Following STRICT, many extensions or generalizations have been proposed [40, 29, 7]. For example, beyond pure label feedback, methods have been proposed to account for feature-based feedback, both for synthetic data [40] and real images [29], using an attribution map [49]. [29] also extended STRICT to multiclass problems.

Alternatively to STRICT, [26, 27] modeled teaching as an iterative process and the learner as a linear classifier, which is updated at each iteration based uniquely on the example seen at that iteration. Beyond [26], [27] treats the student network as a black-box, which more closely resembles real student learning. [18] approximates the student's class conditional distribution given the teaching set with a Gaussian random field but it is designed for online learning, a different setting from that studied in this paper. All these methods assume that the learner is sub-optimal or has limited capacity. However, there is little evidence to support this. On the

contrary, many studies have found that humans have strong learning ability [13, 9, 15, 38], which is also intuitive. We argue that assuming an optimal learner is more sensible in very specialised domains, at least for image classification in the crowdsourcing context.

Feature space: The practical implementation of machine teaching requires a feature extractor to implement the simulated student. Since several prior works were introduced before the popularization of deep learning, they rely on handcrafted features [39, 7]. These are unlikely to be close to human perception and tend to produce low-accuracy classifiers. More recently, it has become standard practice to use features extracted by a deep convolutional network, which is a better model of human perception [36, 10, 48] and produces better classifiers. This is a practice that we also adopt. However, previous works have used networks fine-tuned on a dataset from the target domain [29]. This vastly simplifies the teaching problem, as it is equivalent to assuming that the student already is an expert in the target domain before the teaching starts. We instead rely on a model pretrained on ImageNet. This reflects the assumption that the student is competent in generic image classification tasks, but has no experience in the target domain. This assumption usually holds for the crowdsource setting, whenever the target domain requires specific expertise.

Other approaches: Recently, several works have investigated the use of explanations during the teaching phase, to improve teaching performance. The results are so far inconclusive, as these works show limited improvements [29, 7, 40], particularly in light of the noise inherent to human evaluations, or even a negative impact [29, 7]. While MaxGrad could in principle be combined with visual explanations, we leave this for future work. There have also been proposals for interactive online machine teaching [18], where the selection of teaching examples is not based on a simulated student, but derived from the responses of human users in real-time. However, online updates are costly and difficult to scale to large numbers of simultaneous users. The extension of the ideas used to derive MaxGrad to this setting is a topic that we intend to investigate in the future.

3. Gradient-Based Machine Teaching

In this section, we introduce the MaxGrad algorithm.

3.1. Machine Teaching

In machine teaching for classification, the goal of the teacher is to assemble a *teaching set* $\mathcal{L} = \{(x_i^l, y_i^l)\}_{i=1}^K$ of examples x_i^l and class labels y_i^l, which a student uses to learn a classifier. In this paper, we adopt the pool-based teaching setting [52]. This assumes that the teacher has access to a much larger example dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ from which it selects a subset to assemble $\mathcal{L}$. This is different from synthesis teaching where the teaching examples are synthetically generated. Pool-based teaching is more realistic for image labeling applications, because artificial images may appear nonsensical to a (human) student. The goal of machine teaching is to enable the student to learn the optimal predictor $f^*(x)$ for the entire example population $\mathcal{D}$, from the smallest teaching set $\mathcal{L}$, i.e. the smallest possible number of teaching examples K.

As usual in machine learning, the optimal predictor f^* is defined as the predictor that minimizes the risk $\mathcal{R}_{\mathcal{D}}[f]$ associated with a loss function on $\mathcal{D}$. The details of the loss function depend on the task. For simplicity, we discuss binary classification firstly and extend all ideas to the multi-class setting in section 3.5. For binary classification, $y \in \mathcal{Y} = \{-1, +1\}$, $f(x)$ maps $x \in \mathcal{X}$ to $\mathbb{R}$ and the optimal predictor is

$$f^* = \arg\min_f \mathcal{R}_{\mathcal{D}}[f] = \arg\min_f \sum_{(x_i,y_i)\in\mathcal{D}} \phi(y_i f(x_i)), \tag{1}$$

where $\phi(.)$ is a margin loss function. This predictor is assumed known to the teacher.

The end-goal of the teacher is to assemble the teaching set $\mathcal{L} \subset \mathcal{D}$ that achieves the best trade-off between two conflicting requirements: the student learns the optimal predictor f^* while spending the least effort. This reflects the fact that longer teaching sequences lead to better student performance, but the student has a limited set of learning resources, e.g. a limited attention span. For example, image annotators on crowd-sourcing platforms are well known to drop tasks that are too tedious to master. In this work, we assume that student effort is proportional to the cardinality of the teaching set $|\mathcal{L}|$. This leads to the formulation of the optimal teacher as the one which minimizes some distance $d(f^*, f^s)$ between the predictor f^s learned by the student from $\mathcal{L}$ and the optimal predictor f^*, under a constraint on student effort $|\mathcal{L}| \leq \zeta$.

3.2. The optimal student assumption

In this work, we rely on the assumption that the student is an optimal learner.

Definition 1 *The student is an optimal learner with respect to loss ϕ if and only if, given a teaching set $\mathcal{L}$, it learns the predictor that minimizes the risk defined by ϕ and $\mathcal{L}$,*

$$\mathcal{R}_{\mathcal{L}}[f] = \sum_{(x_i,y_i)\in\mathcal{L}} \phi(y_i f(x_i)). \tag{2}$$

Note that the risk of (2) is defined over $\mathcal{L}$, the teaching set that the student has access to, not the entire population $\mathcal{D}$. The optimal student assumption holds trivially when the student is a machine learning algorithm, because learning algorithms are designed to minimize (2). Since human learners tend to perform at least as well as machine learning algorithms for most tasks, it is a sensible assumption for

human students as well. Under this definition of student, the machine teaching problem can then be formalized as a bilevel optimization problem.

Definition 2 *Under the assumption of an optimal learner with respect to loss ϕ, a teacher is optimal if and only if it produces the teaching set*

$$\mathcal{L}^* = \arg\min_{\mathcal{L}} d(f^*, f^s(\mathcal{L})) \quad (3)$$

$$f^s(\mathcal{L}) = \arg\min_{f} \sum_{(x_i,y_i)\in\mathcal{L}} \phi(y_i f(x_i)). \quad (4)$$

$$|\mathcal{L}| \leq \zeta \quad (5)$$

where f^* is given by (1), $d(.,.)$ is a distance function, and ζ a bound on student effort to process the examples in $\mathcal{L}$.

In what follows, the teaching process is assumed to be iterative.

Definition 3 *An iterative machine teaching procedure iterates between a step of example selection, by the teacher, and a learning step by the student. At iteration t, the teacher produces a teaching set $\mathcal{L}^t$, which the student uses to learn a predictor $f^t(x)$. The teacher then selects from $\mathcal{D}^t = \mathcal{D} - \mathcal{L}^t$ the examples to add to $\mathcal{L}^t$ in order to produce $\mathcal{L}^{t+1}$. The student starts the process with an initial predictor $f^0(x)$. This can be derived from prior experience or $f^0(x) = 0$.*

The following result is an immediate consequence of these definitions.

Corollary 1 *Consider the iterative machine teaching procedure of Definition 3 and assume that the teacher selects at least one new example per learning iteration. If ζ is large enough, the optimal student of Definition 1 is guaranteed to learn the optimal predictor f^* of (1) after a finite number of iterations.*

Proof See Appendix A.1.

In summary, for an optimal student and a sufficient level of effort, the distance $d(f^*, f^s)$ of (3) always converges to zero. It follows that the only role of the teacher is to optimize learning speed, i.e. select the set of examples that enable the student to learn with the least effort. We next define an optimal teacher from this point of view. This, however, requires a brief review of basic concepts in functional optimization.

3.3. Functional optimization

Given two vector spaces $\mathcal{X}$, $\mathcal{Y}$ and a differentiable function $R : \mathcal{X} \to \mathcal{Y}$, the differential $dR(u, \psi)$ of R at $u \in \mathcal{X}$ in the direction $\psi \in \mathcal{X}$ is given by

$$dR(u, \psi) = \frac{d}{d\tau} R(u + \tau\psi)\Big|_{\tau=0}. \quad (6)$$

For example, the margin loss function $\mathcal{M}(f) = \phi(y(x)f(x))$ has differential $d\mathcal{M}(f, \psi) = y\,\phi'(yf)\psi$. Given a set of directions $\Psi = \{\psi_1, \ldots, \psi_n\}$ such that $\psi_i \in \mathcal{X}, \forall i$, the gradient of R with respect to Ψ at u is the vector

$$\nabla_\Psi R(u) = \left(\langle dR(u, \psi_1), \psi_1\rangle, \ldots, \langle dR(u, \psi_n), \psi_n\rangle\right)^T. \quad (7)$$

Let $Sp(\Psi)$ be the span of Ψ and γ a direction in $Sp(\Psi)$, i.e. $\gamma = \sum_i \alpha_i \psi_i$ for some vector α. The derivative of R at u along direction $\gamma \in Sp(\Psi)$ is

$$\partial_\gamma R(u) = \langle \nabla_\Psi R, \alpha \rangle, \quad (8)$$

where $< \alpha, \beta >= \int \alpha(x)\beta(x)dx$ when α and β are functions and $< \alpha, \beta >= \sum_i \alpha_i\beta_i$ when they are finite dimensional vectors.

A dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, defines a set of canonical directions $\Psi(\mathcal{D}) = \{\delta(x - x_i)\}_{i=1}^n$, where $\delta(x)$ is the Dirac delta function. The differentials of the margin loss along these directions are $d\mathcal{M}(f, \psi_k) = y\,\phi'(yf)\,\delta(x - x_k)$ and the empirical risk

$$R_\mathcal{D}(f) = \sum_{(x_i,y_i)\in\mathcal{D}} \phi(y_i f(x_i)) = \sum_{(x_i,y_i)\in\mathcal{D}} \mathcal{M}(f(x_i)) \quad (9)$$

has gradient

$$\nabla_{\Psi(\mathcal{D})} R_\mathcal{D}(f) = (w_1, \ldots, w_n)^T, \quad w_i = y_i\phi'(y_i f(x_i)) \quad (10)$$

where ϕ' is the derivative of ϕ. For any function g in the span of $\Psi(\mathcal{D})$, i.e.

$$g(x) = \sum_i g(x_i)\delta(x - x_i), \quad (11)$$

the derivative of the risk at f along the direction of g is

$$\partial_g R_\mathcal{D}(f) = \sum_{(x_i,y_i)\in\mathcal{D}} w_i g(x_i). \quad (12)$$

The risk $R_\mathcal{D}(f)$ is minimized at f^* if $\partial_g R_\mathcal{D}(f^*) = 0, \forall g \in Sp(\Psi(\mathcal{D}))$, which holds if

$$\nabla_{\Psi(\mathcal{D})} R_\mathcal{D}(f^*) = 0. \quad (13)$$

3.4. The optimal teacher

With these results we are ready to introduce a criterion for teacher optimality, under the iterative teaching procedure of Definition 3. We start by introducing the set of permissible choices for the teaching set, i.e the set of teaching sets that the teacher is allowed to choose from at iteration t. Under the iterative teaching procedure, $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$, i.e. the teacher augments $\mathcal{L}^{t-1}$ with a set of examples $\mathcal{N}^t$ not contained in it, which we denote as the *novel examples* of

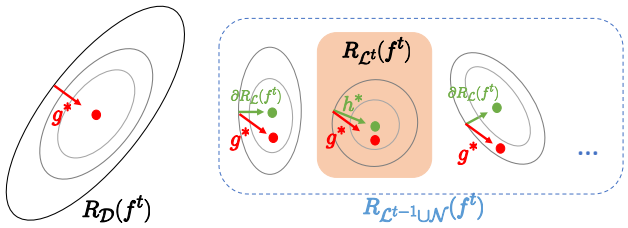


Figure 2: Left: at iteration t, the teacher has access to the population risk $R_{\mathcal{D}}(f^t)$ and corresponding steepest descent direction. Right: the student can only learn from the teaching set $\mathcal{L}^{t-1}$ of iteration $t-1$ and the newly selected examples $\mathcal{N}$. MaxGrad selects $\mathcal{N}$ so that the steepest descent direction on $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}$, i.e. $\partial R_{\mathcal{L}^{t-1}\cup\mathcal{N}}(f^t)$ is closest to g^*.

iteration t. The set of permissible choices includes all such novel sets

$$\mathcal{P}^t(\tau) = \{\mathcal{N} \subset \mathcal{D}^{t-1} | \ |\mathcal{N}| \leq \tau\} \tag{14}$$

The parameter τ upper-bounds the student effort per teaching iteration, enabling the teacher to control the trade-off between number of teaching iterations and student effort. Since, the total effort spent up to iteration t is upper-bounded by $t\tau$, it follows from (5) that the student can learn for up to $T = \zeta/\tau$ iterations. In the iterative setting, it is easier to control the level of effort per iteration than the overall level of effort ζ. In fact, the standard practice in the literature [39, 29, 26] is to allow a single novel example per iteration, i.e. set $\tau = 1$, and then limit the number of iterations T. The definition of set of permissible choices above loosens this constraint.

The question for the teacher is how to select the set of novel examples $\mathcal{N}^t$ is some optimal way. We next introduce the definition of optimality used in this work.

Definition 4 *Consider the iterative machine teaching procedure of Definition 3, with optimal student of Definition 1. Let g^* be the direction of steepest descent of the population risk*

$$g^* = \arg \min_{g \in Sp(\mathcal{D}), ||g||=1} \partial_g R_{\mathcal{D}}(f^t) \tag{15}$$

and $\mathcal{P}^t$ be the set of permissible choices for iteration t. The optimal teacher selects the set of novel examples

$$\mathcal{N}^t = \arg \max_{\mathcal{N} \in \mathcal{P}^t} \langle g^*, h^*(\mathcal{L}^{t-1} \cup \mathcal{N}) \rangle \tag{16}$$

where

$$h^*(\mathcal{L}) = \arg \min_{h \in Sp(\mathcal{L}), ||h||=1} \partial_h R_{\mathcal{L}}(f^t) \tag{17}$$

is the direction of steepest descent on the teaching set risk. The teaching set is then updated into $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$.

This definition encodes the fact that the ideal teaching set $\mathcal{L}^t$ would allow the student to give steepest descent steps on the population risk $R_{\mathcal{D}}$, to enable the fastest progress towards f^*. However, the student does not have access to $\mathcal{D}$, only to $\mathcal{L}^{t-1}$ and a set of novel examples from $\mathcal{P}^t$. The optimal teacher of (16) selects the novel set $\mathcal{N}^t \in \mathcal{P}^t$ that leads to the teaching set $\mathcal{L}^t$ whose steepest descent direction $h^*(\mathcal{L}^t)$ is *closest* to the steepest descent direction g^* of $\mathcal{D}$. This is illustrated in Figure 2.

To derive the solution of (16), we leverage the following property of functional derivatives.

Lemma 1 *For any decomposition of $\mathcal{D} = \mathcal{A} \cup \mathcal{B}$ into two disjoint subsets $\mathcal{A}$ and $\mathcal{B}$ (such that $\mathcal{A} \cap \mathcal{B} = \emptyset$) and any direction $g \in Sp(\Psi(\mathcal{D}))$*

$$\partial_g R_{\mathcal{D}}(f) = \partial_g R_{\mathcal{A}}(f) + \partial_g R_{\mathcal{B}}(f). \tag{18}$$

Proof See Appendix A.2.

The following result uses this property to show that, given what the optimal student has learned until iteration t, the derivative of the population risk is independent of the teaching set $\mathcal{L}^{t-1}$ already studied.

Lemma 2 *Consider the iterative machine teaching procedure of Definition 3. Then, the predictor f^t learned by the optimal student of Definition 1 at iteration t is such that, for any direction g in $Sp(\Psi(\mathcal{D}))$*

$$\partial_g R_{\mathcal{D}}(f^t) = \partial_g R_{\mathcal{D}^{t-1}}(f^t). \tag{19}$$

Proof See Appendix A.3.

The following theorem uses these results to derive the example selection strategy of the optimal teacher.

Theorem 1 *Consider the iterative machine teaching procedure of Definition 3, with optimal student as in Definition 1, and set of permissible choices of (14). The optimal teacher of Definition 4 selects the teaching set $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$ with novel examples*

$$\mathcal{N}^t = \arg \max_{\mathcal{N} \in \mathcal{P}^t} ||\nabla^T_{\Psi(\mathcal{N})} R_{\mathcal{N}}(f^t)||^2 \tag{20}$$

$$= \arg \max_{\mathcal{N} \in \mathcal{P}^t} \sum_{(x_i, y_i) \in \mathcal{N}} w_i^2 \tag{21}$$

where $w_i = \phi'(y_i f^t(x_i))$.

Proof See Appendix A.4.

The theorem shows that the optimal teacher strategy is to select the set of novel examples $\mathcal{N}$ available in $\mathcal{P}^t$ of *largest* risk gradient. For this reason, we denote the teacher as the *MaxGrad* teacher. Since, for margin losses, ϕ' has largest magnitude for negative arguments, w_i is largest for examples

Algorithm 1 MaxGrad

Input Data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, codewords $\mathcal{Y}$, max iter. T, effort τ.

Initialization: $\mathcal{L}^0 \leftarrow \emptyset$, f^1, $\mathcal{D}^0 \leftarrow \mathcal{D}$.
for $t = \{1, \dots, T\}$ **do**
  compute ξ_i for all examples in $\mathcal{D}^{t-1}$.
  order examples by decreasing ξ_i and select top τ to create $\mathcal{N}^t$.
  teaching set update: $\mathcal{L}^t \leftarrow \mathcal{L}^{t-1} \cup \mathcal{N}^t$
  student update: $f^{t+1} = f^*(\mathcal{L}^t)$.
  $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \setminus \mathcal{N}^t$
end for

Output $\mathcal{L}^t$

	binary
$\mathcal{Y}$	$\{-1, +1\}$
ξ_i	$(\phi'(y_i f^t(x_i)))^2$
$f^*(\mathcal{L}^t)$	$\arg\min_f \sum_{(x_i,y_i)\in\mathcal{L}^t} \phi(y_i f(x_i))$

	multi-class
$\mathcal{Y}$	$\{y^1, ..., y^C\}$, $y^i \in \mathcal{R}^d$
ξ_i	$w_i^2 \, \|\|y^{c_i} - \sum_{k\neq c_i} y^k \epsilon_k(x_i, c_i)\|\|^2$
w_i	$\sum_{k\neq c_i} \phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i} - y^k\right\rangle\right]$
$\epsilon_k(x, c)$	$\frac{\phi'\left[\frac{1}{2}\left\langle f^t(x), y^c - y^k\right\rangle\right]}{\sum_{k\neq c}\phi'\left[\frac{1}{2}\left\langle f^t(x), y^c - y^k\right\rangle\right]}$
$\phi(v)$	e^{-v}
$f^*(\mathcal{L}^t)$	$\arg\min_f \sum_{(x_i,y_i)\in\mathcal{L}^t} \sum_{l=1,l\neq y_i}^C \phi\left(\frac{1}{2}\left\langle y^{y_i} - y^l, f(x_i)\right\rangle\right)$

of *negative* margin, i.e. which are incorrectly classified by the current student predictor f^t. Hence, w_i is a measure of how difficult each example is, under the current state of student knowledge. Similarly $\mathcal{H}(\mathcal{N}) = \sum_{(x_i,y_i)\in\mathcal{N}} w_i^2$ measures the difficulty, for the student, of the novel examples in $\mathcal{N}$. It follows from (21) that the MaxGrad teacher always selects the *hardest set of novel examples* in $\mathcal{P}^t$. Furthermore, since $\mathcal{H}(\mathcal{N})$ is a sum of non-negative terms, it is an increasing function of $|\mathcal{N}|$. This implies that the teacher has a preference for larger sets of novel examples. As long as there are examples that the student has not mastered ($w_i > 0$), it will choose a set of τ examples per iteration. Hence, $|\mathcal{N}^t| = \tau$ for all $t < T$ and the overall learning complexity is $T\tau$. This implies that the number of iterations is upper bounded by ζ/τ, which makes it equivalent to specifying a maximum level of effort ζ or a maximum number of iterations T for the teaching process. Finally, because the set of permissible choices includes all novel sets of cardinality τ, the solution of (21) is trivial: it suffices to compute w_i for all examples in $\mathcal{D}^{t-1}$ and select the τ examples of largest w_i^2. The resulting machine teaching procedure is summarized by Algorithm 1.

3.5. Multi-class extension

We have discussed binary classification tasks, were $f(x) \in \mathbb{R}$, class labels $y \in \{-1, 1\}$, the margin of example (x, y) is defined as $yf(x)$ and a margin loss is a function $\phi(yf(x))$ for some decreasing $\phi \in \mathbb{R}^+$. All ideas can be generalized for the C-class case, by extending these definitions. A common generalization is to use a d-dimensional predictor, $f(x) \in \mathbb{R}^d$, a set of C class label codewords $y^c \in \mathcal{Y} = \{y^1, ..., y^C\}$, where $y^c \in \mathbb{R}^d$, and define the margin of example x with respect to class y^k as

$$\mathcal{M}(y^k, f(x)) = \min_{l\neq k} \frac{1}{2}\left\langle y^k - y^l, f(x)\right\rangle. \quad (22)$$

A family of margin losses is then defined as [33]

$$L[y^k, f(x)] = \sum_{l=1,l\neq k}^{C} \phi\left(\frac{1}{2}\left\langle y^k - y^l, f(x)\right\rangle\right), \quad (23)$$

where $\phi : \mathbb{R} \to \mathcal{R}^+$ are strictly positive. A theoretical discussion of the properties of these losses can be found in [33]. The empirical risk then becomes

$$R_{\mathcal{D}}(f) = \sum_{(x_i,y_i)\in\mathcal{D}} L[y^{y_i}, f(x_i)]. \quad (24)$$

and, given a dataset $\mathcal{D} = \{(x_i, c_i)\}$ and a corresponding set of directions $\Psi(\mathcal{D}) = \{\psi_1, \dots, \psi_n\}$ such that $\psi_i = \delta(x - x_i)$ the gradient of $R_{\mathcal{D}}(f)$ evaluated at f^t with respect to $\Psi(\mathcal{D})$ has entries

$$\left[\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)\right]_i = w_i \left(y^{c_i} - \sum_{k\neq c_i} y^k \epsilon_k(x_i, c_i)\right), \quad (25)$$

with

$$w_i = \sum_{k\neq c_i} \phi'\left[\frac{1}{2}\left\langle f^t(x_i), y^{c_i} - y^k\right\rangle\right] \quad (26)$$

$$\epsilon_k(x, c) = \frac{\phi'\left[\frac{1}{2}\left\langle f^t(x), y^c - y^k\right\rangle\right]}{\sum_{k\neq c}\phi'\left[\frac{1}{2}\left\langle f^t(x), y^c - y^k\right\rangle\right]}. \quad (27)$$

Note that $\left[\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)\right]_i$ is a d-dimensional vector. The gradient norm of (21) is then

$$||\nabla^T_{\Psi(\mathcal{N})} R_{\mathcal{N}}(f^t)||^2 = \sum_{(x_i,c_i)\in\mathcal{N}} ||\left[\nabla_{\Psi(\mathcal{D})} R_{\mathcal{D}}(f^t)\right]_i||^2 \quad (28)$$

$$= \sum_{(x_i,c_i)\in\mathcal{N}} \xi_i \quad (29)$$

where $\xi_i = w_i^2 \, ||y^{c_i} - \sum_{k\neq c_i} y^k \epsilon_k(x_i, c_i)||^2$. In this work, we adopt the exponential loss by setting $\phi(v) = e^{-v}$, leading to the multi-class version of Algorithm 1 for the implementation of the optimal multi-class teacher.

4. Connections to boosting

The algorithm above has certain similarities with boosting. Note that the weights of (10) are the weights of boosting

(a) butterflies (b) Chinese Characters

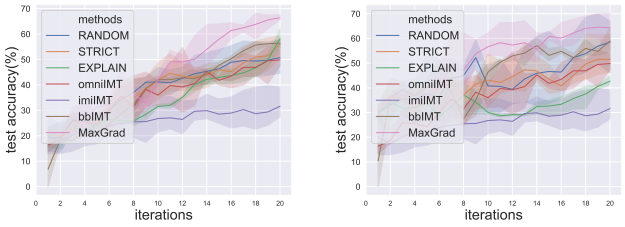
Figure 3: Example images from our two datasets

at the end of the iteration that produces f as strong classifier. Boosting then selects the weak learner g^* that maximizes (12), adds this to f to produce the new strong classifier and iterates. Since examples of large weight are those worse classified by f, the algorithm focuses on the hardest examples (for the currently learned classifier) to pick the next weak learner. The MaxGrad teacher does essentially the same. In this case, f^t is the predictor currently learned by the optimal student, and the teacher selects the hardest examples for the student. However, in boosting, this is used to perform one learning iteration and select one weak learner. In machine teaching, the student is assumed to be able to fully learn $\mathcal{L}^t$, i.e. does not simply perform a gradient iteration on $R_{\mathcal{D}}(f^t)$ but actually solves (4). If, for example, the student is a machine learning algorithm, this can be done by implementing the complete boosting algorithm on $R_{\mathcal{L}^t}(f)$. While boosting uses the entire example population $\mathcal{D}$ to perform a boosting iteration and select a single weak learner, the machine teaching algorithm selects the best set $\mathcal{N}^t$ of τ novel examples to add to $\mathcal{L}^t$ and performs any number of boosting iterations needed to solve the new teaching set $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \mathcal{N}^t$. In summary, while boosting assumes a weak learner with access to the entire dataset $\mathcal{D}$, the machine teaching algorithm assumes a strong learner with access to the limited information available in $\mathcal{L}^t$.

5. Experiments

Dataset: MaxGrad was evaluated on two datasets, Butterflies and Chinese Characters illustrated in Figure 3. Butterflies [29] is a fine-grained multi-class dataset of images of five butterfly species, captured in a large variety of settings, from the iNaturalist dataset [43]. It is a challenging dataset due to the large intra-class image diversity, low image resolution, and high similarity of some of the species. Chinese Characters [25, 18] consists of three similar Chinese characters: Grass, Mound, and Stem. The images vary in difficulty, due to a large variety of handwriting styles and image qualities. We use the training-testing split of [29] on both cases. The data is accessible in [28]. The teaching set is selected from the training set.

Implementation details: Both datasets were subject to standard normalizations. The pre-trained ResNet-18 [16] on ImageNet is used to simulate the student. This is equivalent to assuming a student that starts from a good generic understanding of image classification. The student learners are trained 10 epochs by gradient descent with batch size equal to $|\mathcal{L}^t|$ and weight decay of $1e-4$. The learning rate is set to $1e-4$ with 0.9 momentum. For fair comparison with other methods [26, 39, 29, 27], novel sets of size $\tau = 1$ were used in all experiments, i.e. a single example is selected per iteration.



(a) butterflies (b) Chinese Characters

Figure 4: Test set accuracy of simulated students as a function of teaching iterations.

	Butterflies	Chinese Char.
RANDOM	65.20	47.05
STRICT [39]	65.00	51.51
EXPLAIN [29]	68.33	65.44
omniIMT* [26]	70.07(18.30)	64.36(19.58)
imiIMT* [26]	72.70(17.63)	64.46(23.72)
bbIMT* [27]	76.09(18.05)	64.37(19.57)
RANDOM*	63.15(18.17)	51.53(24.47)
MaxGrad	**80.33**(19.76)	**81.89**(12.93)

Table 1: Test set accuracies for MTurk learners. Methods with superscript "$*$" represent our implementations. For top RANDOM, value is from [29]. Values are presented by mean(std).

5.1. Evaluation with simulated learners

We start with evaluations on simulated learners, i.e. a classifier. This enables a simple evaluation setting and fully reproducible experiments. Figure 4 shows the accuracies of student networks taught with examples selected randomly (RANDOM), by STRICT [39], EXPLAIN [29], omniscient teacher (omniIMT) [26], imitation teacher (imiIMT) [26], black-box IMT (bbIMT) [27], and MaxGrad. While student performance improves with teaching set size for all methods, MaxGrad has the fastest growth and the best performance for all iterations. The gains are significant: in butterflies and characters it achieves an accuracy at 15 iterations that others do not reach before 20 iterations. Of all algorithms, it is also the only to stably outperform RANDOM.

5.2. Evaluation with real learners

We next tested the algorithm on MTurk users. Note that a student network was still used to assemble the teaching set, which was then used to train MTurkers. In this case, the student network was trained without any stochasticity. We used gradient descent and gave up data augmentation techniques (e.g. random crops or flips) that are not accessible to the human students. The codewords y^c of Algorithm 1 were initialized with the canonical basis and refined during the student optimization.

The MTurk experiments followed the setting of [29, 39], using 40 workers per dataset. The teaching process consists

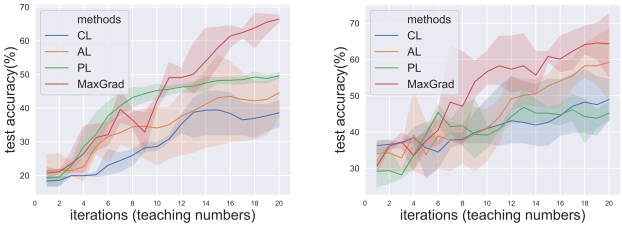
of two phases, teaching and testing. Before teaching, workers were shown a brief introduction to the teaching task. In the teaching stage, they were shown a sequence of 20 images. At each iteration, they were asked to select a category from a list of candidate options, and received feedback declaring their choice 'Correct' or 'Incorrect,' as well as the true class. Upon this, learners had to wait for a minimum of 2 seconds before proceeding to the next iteration. After teaching, 20 randomly selected test images were assigned to each learner, who was asked to classify them. These random images were different per learner and no feedback was provided as they were classified.

Table 1 reports the accuracy of image classification by the students on the test set. The results shown in the top third of the table (RANDOM, STRICT and EXPLAIN) are taken from [29]. For completeness, we repeated the experiments with random image selection, which produced similar results, as shown in the bottom third. The remaining three results of previous methods (center third of the table) are obtained with our own implementation. MaxGrad significantly outperforms the previous approaches, achieving gains of almost 5 (17) points on Butterflies (Chinese characters). Finally, we observe that human test accuracy is higher than that of the simulated student used to collect the training set, shown in Figure 4. This confirms that the optimal student assumption is realistic for human learners.

5.3. Comparison with related methods

We use simulated learners to compare MaxGrad to related algorithms beyond the machine teaching literature: curriculum learning [4, 14], active learning [37, 45] and passive learning. Curriculum learning (CL) orders training examples by complexity. The student is introduced to easier examples first and then harder ones. CL has been empirically shown to accelerate and improve learning. In active learning (AL), the student actively selects examples to be annotated. The annotator can be seen as an oracle teacher. Passive learning (PL) denotes standard supervised learning on the whole training set. Two representative methods, [14] for CL and [45] for AL, are adopted for comparison, with the results of Figure 5.

To compare to CL and AL, we plot the curves of test set accuracy v.s. teaching example number. MaxGrad outperforms the two algorithms substantially. This is not surprising, because CL only sees a few (20) easy examples and cannot generalize to hard ones during testing. It requires long-term training on the whole dataset to reach good accuracy. The weaker performance of AL is explained by the fact that the examples are selected by the student not the teacher. The teacher only labels them. Hence, the teaching set is not globally optimal. To compare with PL, we plot the curves of test accuracy vs teaching iteration. MaxGrad ascends faster than PL after 10 (7) iterations on butterflies (characters). This suggests that, when only a limited number of iterations



(a) butterflies (b) Chinese Characters

Figure 5: Test set accuracy of simulated students as a function of teaching iterations (teaching example number).

Figure 6: Randomly selected teaching sequence from the Butterflies teaching set

is allowed, teaching with a MaxGrad teaching set is more efficient than with the whole training set.

5.4. Sample Images

To gain some intuition about the teaching sets selected by MaxGrad, we show a (randomly selected) sequence of 5 teaching examples from Butterflies in Figure 6. It can be seen that, beyond a few "normal" images that teach students the appearance of the concepts, the algorithm selects images with unusual poses, low-resolution, and even camouflage. This is consistent with MaxGrad's example selection based on large negative margins. These images force the students to focus attention on features that are essential for class discrimination, speeding up the learning process. The improved student performance suggests some degree of overlap between the criteria used by MaxGrad and humans to assess example difficulty.

6. Conclusion

In this paper, we have proposed MaxGrad, a new gradient-based machine teaching algorithm derived from the optimal student assumption. We have demonstrated its effectiveness on both synthetic and human student teaching experiments. While we have not considered the integration of teaching and explanations yet, MaxGrad can be generalized to accommodate the latter. For example, explanations can be merged with classifier training using attention mechanisms. This is left for future research.

Acknowledgement This work was partially funded by NSF awards IIS-1924937, IIS-2041009, a gift from Amazon, a gift from Qualcomm, and NVIDIA GPU donations. We also acknowledge and thank the use of the Nautilus platform for some of the experiments discussed above.

References

[1] Antreas Antoniou and Amos J Storkey. Learning to learn by self-critique. In *Advances in Neural Information Processing Systems*, pages 9936–9946, 2019.

[2] Ji Hyun Bak, Jung Yoon Choi, Athena Akrami, Ilana Witten, and Jonathan W Pillow. Adaptive optimal training of animal behavior. In *Advances in neural information processing systems*, pages 1947–1955, 2016.

[3] Sumit Basu and Janara Christensen. Teaching classification boundaries to humans. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[5] Cory J Butz, Shan Hua, and R Brien Maguire. A web-based intelligent tutoring system for computer programming. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pages 159–165. IEEE, 2004.

[6] Maya Cakmak and Manuel Lopes. Algorithmic and human teaching of sequential decision tasks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[7] Yuxin Chen, Oisin Mac Aodha, Shihan Su, Pietro Perona, and Yisong Yue. Near-optimal machine teaching via explanatory teaching sets. In *International Conference on Artificial Intelligence and Statistics*, pages 1970–1978, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Sharon J Derry and Debra A Murphy. Designing systems that train learning ability: From theory to practice. *Review of educational research*, 56(1):1–39, 1986.

[10] Michael Eickenberg, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, 152:184–194, 2017.

[11] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[12] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[13] Elena L Grigornko, Robert J Sternberg, and Madeline E Ehrman. A theory-based approach to the measurement of foreign language learning ability: The canal-f theory and test. *The Modern Language Journal*, 84(3):390–405, 2000.

[14] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. *ICML*, 2019.

[15] Nile W Hatch and Jeffrey H Dyer. Human capital and learning as a source of sustainable competitive advantage. *Strategic management journal*, 25(12):1155–1178, 2004.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Edward Johns, Oisin Mac Aodha, and Gabriel J Brostow. Becoming the expert-interactive multi-class machine teaching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2624, 2015.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[21] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[22] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[23] Dong Li, Wei-Chih Hung, Jia-Bin Huang, Shengjin Wang, Narendra Ahuja, and Ming-Hsuan Yang. Unsupervised visual representation learning by graph-based consistent constraints. In *European Conference on Computer Vision*, pages 678–694. Springer, 2016.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition*, pages 37–41. IEEE, 2011.

[26] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2149–2158. JMLR. org, 2017.

[27] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James M Rehg, and Le Song. Towards black-box iterative machine teaching. *International Conference on Machine Learning*, 2018.

[28] Oisin Mac Aodha, Shihan Su, Yuxin Chen, Pietro Perona, and Yisong Yue. https://github.com/macaodha/explain_teach/tree/master/data.

[29] Oisin Mac Aodha, Shihan Su, Yuxin Chen, Pietro Perona, and Yisong Yue. Teaching categories to human learners with visual explanations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3820–3828, 2018.

[30] Kaustubh R Patil, Jerry Zhu, Łukasz Kopeć, and Bradley C Love. Optimal teaching for limited-capacity human learners. In *Advances in neural information processing systems*, pages 2465–2473, 2014.

[31] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[32] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive science*, 40(6):1290–1332, 2016.

[33] Mohammad Saberian and Nuno Vasconcelos. Multiclass boosting: Margins, codewords, losses, and algorithms. *Journal of Machine Learning Research*, 20(137):1–68, 2019.

[34] Mohammad J Saberian and Nuno Vasconcelos. Multiclass boosting: Theory and algorithms. In *Advances in Neural Information Processing Systems*, pages 2124–2132, 2011.

[35] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems*, pages 12942–12952, 2019.

[36] Katja Seeliger, Matthias Fritsche, Umut Güçlü, Sanne Schoenmakers, J-M Schoffelen, SE Bosch, and MAJ Van Gerven. Convolutional neural network-based encoding and decoding of visual object recognition in space and time. *NeuroImage*, 180:253–266, 2018.

[37] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[38] Herbert A Simon. Bounded rationality and organizational learning. *Organization science*, 2(1):125–134, 1991.

[39] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, volume 1, page 3, 2014.

[40] Shihan Su, Yuxin Chen, Oisin Mac Aodha, Pietro Perona, and Yisong Yue. Interpretable machine teaching via feature feedback. 2017.

[41] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[42] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[43] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[44] Luis Von Ahn. Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 1–2, 2013.

[45] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

[46] Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *International Conference on Machine Learning*, pages 5085–5094, 2018.

[47] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2907–2916, 2019.

[48] Astrid A Zeman, J Brendan Ritchie, Stefania Bracci, and Hans Op de Beeck. orthogonal representations of object shape and category in deep convolutional neural networks and human visual cortex. *Scientific Reports*, 10(1):1–12, 2020.

[49] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[50] Yao Zhou, Arun Reddy Nelakurthi, and Jingrui He. Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2817–2826, 2018.

[51] Jerry Zhu. Machine teaching for bayesian learners in the exponential family. In *Advances in Neural Information Processing Systems*, pages 1905–1913, 2013.

[52] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N Rafferty. An overview of machine teaching. *arXiv preprint arXiv:1801.05927*, 2018.