Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks

Daphna Weinshall [1] **Gad Cohen** [1] **Dan Amir** [1]

Abstract

We provide theoretical investigation of curriculum learning in the context of stochastic gradient descent when optimizing the convex linear regression loss. We prove that the rate of convergence of an ideal curriculum learning method is monotonically increasing with the difficulty of the examples. Moreover, among all equally difficult points, convergence is faster when using points which incur higher loss with respect to the current hypothesis. We then analyze curriculum learning in the context of training a CNN. We describe a method which infers the curriculum by way of transfer learning from another network, pre-trained on a different task. While this approach can only approximate the ideal curriculum, we observe empirically similar behavior to the one predicted by the theory, namely, a significant boost in convergence speed at the beginning of training. When the task is made more difficult, improvement in generalization performance is also observed. Finally, curriculum learning exhibits robustness against unfavorable conditions such as excessive regularization.

1. Introduction

Biological organisms can learn to perform tasks (and often do) by observing a sequence of labeled events, just like supervised machine learning. But unlike machine learning, in human learning supervision is often accompanied by a curriculum. Thus the order of presented examples is rarely random when a human teacher teaches another human. Likewise, the task may be divided by the teacher into smaller sub-tasks, a process sometimes called shaping (Krueger & Dayan, 2009) and typically studied in the context of reinforcement learning (e.g. Graves et al., 2017). Although it remained for the most part in the fringes of machine learning research, curriculum learning has been identified as a key challenge for machine learning throughout (e.g., Mitchell, 1980; 2006; Wang & Cottrell, 2015).

We focus here on curriculum learning based on ranking (or weighting as in (Bengio et al., 2009)) of the training examples, which is used to guide the order of presentation of examples to the learner. Risking over simplification, the idea is to first present the learner primarily with examples of higher weight or rank, later to be followed by examples with lower weight or rank. Ranking may be based on the difficulty of each training example as evaluated by the teacher, from easiest to the most difficult.

In Section 2 we investigate this strict definition of curriculum learning theoretically, in the context of stochastic gradient descent used to optimize the convex linear regression loss function. We first define the (ideal) difficulty of a training point as its loss with respect to the optimal classifier. We then prove that curriculum learning, when given the ranking of training points by their difficulty thus defined, is expected (probabilistically) to significantly speed up learning especially at the beginning of training. This theoretical result is supported by empirical evidence obtained in the deep learning scenario of curriculum learning described in Section 3, where similar behavior is observed. We also show that when the difficulty of the sampled training points is fixed, convergence is faster when sampling points that incur higher loss with respect to the current hypothesis as suggested in (Shrivastava et al., 2016). This result is *not* always true when the difficulty of the sampled training points is not fixed.

But such ideal ranking is rarely available. In fact, the need for such supervision has rendered curriculum learning less useful in machine learning, since ranking by difficulty is hard to obtain. Moreover, even when it is provided by a human teacher, it may not reflect the true difficulty as it affects the machine learner. For example, in visual object recognition it has been demonstrated that what makes an image difficult to a neural network classifier may not always match whatever makes it difficult to a human observer, an observation that has been taken advantage of in the recent

[1]School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel. Correspondence to: Daphna Weinshall <daphna@mail.huji.ac.il>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

work on adversarial examples (Szegedy et al., 2013). Possibly, this is one of the reasons why curriculum learning is rarely used in practice (but see, e.g., Zaremba & Sutskever, 2014; Amodei et al., 2016; Jesson et al., 2017).

In the second part of this paper we focus on this question - how to rank (or weight) the training examples without the aid of a human teacher. This is paramount when a human teacher cannot provide a reliable difficulty score for the task at hand, or when obtaining such a score by human teachers is too costly. This question is also closely related to transfer learning: here we investigate the use of another classifier to provide the ranking of the training examples by their presumed difficulty. This form of transfer should not be confused with the notion of transfer discussed in (Bengio et al., 2009) in the context of multi-task and life-long learning (Thrun & Pratt, 2012), where knowledge is transferred from earlier tasks (e.g. the discrimination of easy examples) to later tasks (e.g. the discrimination of difficult examples). Rather, we investigate the transfer of knowledge from one classifier to another, as in *teacher classifier* to *student classifier*. In this form curriculum learning has not been studied in the context of deep learning, and hardly ever in the context of other classification paradigms.

Differently from previous work, it is not the instance representation which is being transferred but rather the ranking of training examples. Why is this a good idea? This kind of transfer assumes that a powerful pre-trained network is only available at train time, and cannot be used at test time even for the computation of a test point's representation. This may be the case, for example, when the powerful network is too big to run on the target device. One can no longer expect to have access to the transferred representation at test time, while ranking can be used at train time in order to improve the learning of the target smaller network (see related discussion of network compression in (Chen et al., 2015; Kim et al., 2015), for example).

In Section 3 we describe our method, an algorithm which uses the ranking to construct a schedule for the order of presentation of training examples. In subsequent empirical evaluations we compare the performance of the method when using a curriculum which is based on different scheduling options, including 2 control conditions where difficult examples are presented first or when using arbitrary scheduling. The main results of this empirical study can be summarized as follows: (i) Learning rate is always faster with curriculum learning, especially at the beginning of training. (ii) Final generalization is sometimes improved with curriculum learning, especially when the conditions for learning are hard: the task is difficult, the network is small, or when strong regularization is enforced. These results are consistent with prior art (see e.g. Bengio et al., 2009).

2. Theoretical analysis

We start with some notations in Section 2.1, followed in Sections 2.2 by the rigorous analysis of curriculum learning when used to optimize the linear regression loss. In Section 2.3 we report supporting empirical evidence for the main theoretical results, obtained using the deep learning setup described later in Section 3.

2.1. Notations and definitions

Let $\mathbb{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the training data, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the i-th data point and y_i its corresponding label. In general, the goal is to find a hypothesis $\bar{h}(\mathbf{x}) \in \mathcal{H}$ that minimizes the risk function (the expected loss). In order to minimize this objective, Stochastic Gradient Descent (SGD) is often used with various extensions and regularization.

We start with two definitions:

Definition 1 (Ideal Difficulty Score)**.** *The difficulty of point* $\mathbf{x}$ *is measured by its minimal loss with respect to the set of optimal hypotheses* $\{L(\bar{h}(\mathbf{x}_i), y_i)\}$.

Definition 2 (Stochastic Curriculum Learning)**.** *SCL is a variation on Stochastic Gradient Descent (SGD), where the learner is exposed to the data gradually based on the difficulty score of the training points.*

In vanilla SGD training, at each iteration the learner is presented with a new datapoint (or mini-batch) sampled from the training data based on some probability function $\mathcal{D}(\mathbb{X})$. In SCL, the sampling is biased to favor easier examples at the beginning of the training. This bias is decreased following some scheduling procedure. At the end of training, points are sampled according to $\mathcal{D}(\mathbb{X})$ as in vanilla SGD.

In practice, an SCL algorithm should solve two problems: (i) Score the training points by difficulty; in prior art this score was typically provided by the teacher in a supervised manner. (ii) Define the scheduling procedure.

2.2. The linear regression loss

Here we analyze SCL when used to minimize the linear regression model. Specifically, we investigate the differential effect of a point's *Difficulty Score* on convergence towards the global minimum of the expected least squares loss, when the family of hypotheses $\mathcal{H}$ includes the linear functions $h(\mathbf{x}) = \mathbf{a}^t\mathbf{x} + b$ and $y \in \mathbb{R}$.

The risk function of the regression model is the following

$$\begin{aligned} \mathcal{R}(\mathbb{X}, \mathbf{w}) &= \mathbb{E}_{\mathcal{D}(\mathbb{X})} L(h_{\mathbf{w}}(\mathbf{x}), y) \\ L(h_{\mathbf{w}}(\mathbf{x}_i), y_i) &= (h(\mathbf{x}_i) - y_i)^2 = (\mathbf{a}^t\mathbf{x}_i + b - y_i)^2 \\ &\triangleq (\mathbf{x}_i^t\mathbf{w} - y_i)^2 \triangleq L(\mathbf{X}_i, \mathbf{w}) \end{aligned} \quad (1)$$

In the last transition above, $\mathbf{w} = \begin{bmatrix} \mathbf{a} \\ b \end{bmatrix} \in \mathbb{R}^{d+1}$. With some

abuse of notation, $\mathbf{x}_i$ denotes the vector $\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$. $\mathbf{X}_i$ denotes the vector $[\mathbf{x}_i, y_i]$, with *Difficulty Score* $L(\mathbf{X}_i, \bar{\mathbf{w}})$.

In general the output hypothesis $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}_i^t\mathbf{w}$ is determined by minimizing $\mathcal{R}(\mathbb{X}, \mathbf{w})$ with respect to $\mathbf{w}$. The global minimum $\bar{\mathbf{w}}$ of the empirical loss can be computed in closed form from the training data. However, gradient descent can be used to find $\bar{\mathbf{w}}$ with guaranteed convergence, which is efficient when n is very large.

Recall that SCL computes a sequence of estimators $\{\mathbf{w}_t\}_{t=1}^T$ for the parameters of the optimal hypothesis $\bar{\mathbf{w}}$. This is based on a sequence of training points $\{\mathbf{X}_t = [\mathbf{x}_t, y_t]\}_{t=1}^T$, sampled from the training data while favoring easy points at the beginning of training. Other than sampling probability, the update step at time t follows SGD:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial L(\mathbf{X}_t, \mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}_t} \tag{2}$$

CONVERGENCE RATE DECREASES WITH DIFFICULTY

The main theorem in this sub-section states that the expected rate of convergence of gradient descent is monotonically *decreasing* with the *Difficulty Score* of the sample $\mathbf{X}_t$. We prove it below for the gradient step as defined in (2). If the size of the gradient step is fixed at η, a somewhat stronger theorem can be obtained where the constraint on the step size being small is not required.

We first derive the gradient step at time t:

$$\mathbf{s} = -\eta \frac{\partial L(\mathbf{X}_i, \mathbf{w})}{\partial \mathbf{w}} = -2\eta(\mathbf{x}_i^t\mathbf{w} - y_i)\mathbf{x}_i \tag{3}$$

Let Ω_i denote the hyperplane on which this gradient vanishes $\frac{\partial L(\mathbf{X}_i, \mathbf{w})}{\partial \mathbf{w}} = 0$. This hyperplane is defined by $\mathbf{x}_i^t\mathbf{w} = y_i$, namely, $\mathbf{x}_i$ defines its normal direction. Thus (3) implies that the gradient step at time t is perpendicular to Ω_i. Let $\bar{\mathbf{z}}$ denote the projection of $\bar{\mathbf{w}}$ on Ω_i. Let $\Psi^2 = L(\mathbf{X}_i, \bar{\mathbf{w}})$ denote the *Difficulty Score* of $\mathbf{X}_i$.

Lemma 1. *Fix the training point* $\mathbf{X}_i$. *The* Difficulty Score *of* $\mathbf{X}_i$ *is* $\Psi^2 = r^2\|\bar{\mathbf{w}} - \bar{\mathbf{z}}\|^2$.

Proof.

$$\begin{aligned} \Psi^2 &= L(\mathbf{X}_i, \bar{\mathbf{w}}) = L(\mathbf{X}_i, \bar{\mathbf{z}} + (\bar{\mathbf{w}} - \bar{\mathbf{z}})) \\ &= [\mathbf{x}_i^t\bar{\mathbf{z}} + \mathbf{x}_i^t(\bar{\mathbf{w}} - \bar{\mathbf{z}}) - y_i]^2 \\ &= [\mathbf{x}_i^t(\bar{\mathbf{w}} - \bar{\mathbf{z}})]^2 = \|\mathbf{x}_i\|^2\|\bar{\mathbf{w}} - \bar{\mathbf{z}}\|^2 \end{aligned} \tag{4}$$

□

Recall that $\mathbf{x}_i, \mathbf{w} \in \mathbb{R}^{d+1}$. We continue the analysis in the parameter space $\mathbf{w} \in \mathbb{R}^{d+1}$, where parameter vector $\mathbf{w}$ corresponds to a point, and data vector $\mathbf{x}_i$ describes a hyperplane. In this space we represent each vector $\mathbf{x}_i$ in a hyperspherical coordinate system $[r, \vartheta, \Phi]$, with pole (origin) fixed at $\bar{\mathbf{w}}$ and polar axis (zenith direction) $\vec{\mathcal{O}} = \bar{\mathbf{w}} - \mathbf{w}_t$ (see Fig. 1). r denotes the vector's length, while $0 \le \vartheta \le \pi$ denotes the polar angle with respect to $\vec{\mathcal{O}}$. Let $\Phi = [\varphi_1 \ldots, \varphi_{d-1}]$ denote the remaining polar angles.

To illustrate, Fig. 1 shows a planar section of the parameter space, the $2D$ plane formed by the two intersecting lines $\vec{\mathcal{O}}$ and $\bar{\mathbf{z}} - \bar{\mathbf{w}}$. The gradient step $\mathbf{s}$ points from $\mathbf{w}_t$ towards Ω_i. Ω_i is perpendicular to $\mathbf{x}_i$, which is parallel to $\bar{\mathbf{z}} - \bar{\mathbf{w}}$ and to $\mathbf{s}$, and therefore Ω_i is projected onto a line in this plane. We introduce the notation $\lambda = \|\bar{\mathbf{w}} - \mathbf{w}_t\|$.

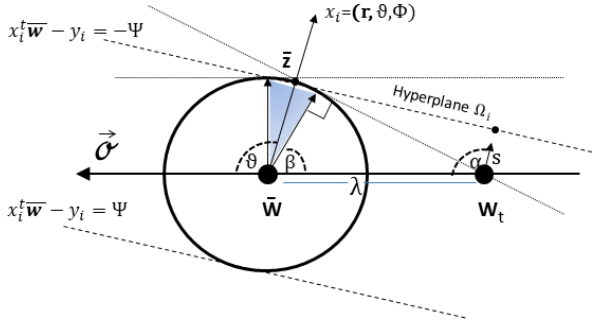

Figure 1. The $2D$ planar section defined by the vectors $\vec{\mathcal{O}} = \bar{\mathbf{w}} - \mathbf{w}_t$ and $\bar{\mathbf{z}} - \bar{\mathbf{w}}$. The circle centered on $\bar{\mathbf{w}}$ has radius $\|\bar{\mathbf{w}} - \bar{\mathbf{z}}\| = \frac{\Psi}{\|\mathbf{x}_i\|}$ from Lemma 1. It traces the location of $\bar{\mathbf{z}}$ for all the points $\mathbf{x}_i$ with the same length r and the same difficulty score Ψ.

Let $\mathbf{s}_{\mathcal{O}}$ denote the projection of the gradient vector $\mathbf{s}$ on the polar axis $\vec{\mathcal{O}}$, and let $\mathbf{s}_{\perp}$ denote the perpendicular component. From (3) and the definition of Ψ

$$\begin{aligned} \mathbf{s} &= -2\eta\mathbf{x}_i(\mathbf{x}_i^t\mathbf{w}_t - y_i) = -2\eta\mathbf{x}_i[\mathbf{x}_i^t(\mathbf{w}_t - \bar{\mathbf{w}}) \pm \Psi] \\ \mathbf{s}_{\mathcal{O}} &= \mathbf{s} \cdot \frac{\bar{\mathbf{w}} - \mathbf{w}_t}{\lambda} = 2\frac{\eta}{\lambda}[r^2\lambda^2\cos^2\vartheta \mp \Psi r\lambda\cos\vartheta] \end{aligned} \tag{5}$$

Let $\mathbf{x} = (r, \vartheta, \Phi)$. Let $f_{\mathcal{D}(\mathbb{X})} = f(r, \vartheta, \Phi)f_Y(|y - \mathbf{x}^t\bar{\mathbf{w}}|)$ denote the density function of the data $\mathbb{X}$. This choice assumes that the density of the label y only depends on the absolute error $|y - \mathbf{x}^t\bar{\mathbf{w}}|$.

For the subsequent derivations we need the conditional distribution of the data $\mathbb{X}$ given difficulty score Ψ. Fixing the difficulty score determines one of two labels $y(\mathbf{x}, \Psi) = \mathbf{x}^t\bar{\mathbf{w}} \pm \Psi$. We further assume that both labels are equally likely[1], and therefore $f_{\mathcal{D}(\mathbb{X})/\Psi}([\mathbf{x}, y]) = \frac{1}{2}f(r, \vartheta, \Phi)$.

Let $\Delta(\Psi)$ denote the expected convergence rate at time t, given fixed difficulty score Ψ.

$$\Delta(\Psi) = \mathbb{E}[\|\mathbf{w}_t - \bar{\mathbf{w}}\|^2 - \|\mathbf{w}_{t+1} - \bar{\mathbf{w}}\|^2/_{\Psi}] \tag{6}$$

Lemma 2.

$$\Delta(\Psi) = 2\lambda\mathbb{E}[\mathbf{s}_{\mathcal{O}}/_{\Psi}] - \mathbb{E}[\mathbf{s}^2/_{\Psi}] \tag{7}$$

[1]This assumption can be somewhat relaxed, but the strict form is used to simplify the exposition.

Proof. From (6)

$$\begin{aligned}\mathbb{E}[\Delta] &= \lambda^2 - \mathbb{E}[(\lambda - \mathbf{s}_{\mathcal{O}})^2 + \mathbf{s}_{\perp}^2] \\ &= \lambda^2 - (\lambda^2 - 2\lambda\mathbb{E}[\mathbf{s}_{\mathcal{O}}] + \mathbb{E}[\mathbf{s}_{\mathcal{O}}^2]) - \mathbb{E}[\mathbf{s}_{\perp}^2] \\ &= 2\lambda\mathbb{E}[\mathbf{s}_{\mathcal{O}}] - \mathbb{E}[\mathbf{s}^2]\end{aligned}$$

□

From Lemma 2 and (5)[2]

$$\begin{aligned}\frac{1}{4}\Delta(\Psi) = \eta\mathbb{E}[r^2\lambda^2\cos^2\vartheta] - \eta^2\mathbb{E}[r^4\lambda^2\cos^2\vartheta] \\ -\eta^2\Psi^2\mathbb{E}[r^2] \\ -\eta\mathbb{E}[(\pm\Psi)r\lambda\cos\vartheta] - 2\eta^2\mathbb{E}[(\pm\Psi)r^3\lambda\cos\vartheta]\end{aligned} \tag{8}$$

Lemma 3.

$$\mathbb{E}[(\pm\Psi)r\lambda\cos\vartheta] = \mathbb{E}[(\pm\Psi)r^3\lambda\cos\vartheta] = 0$$

Proof. The lemma follows from the assumed symmetry of $\mathcal{D}(\mathbb{X})$ with respect to the sign of $y_i - \mathbf{x}_i^t\bar{\mathbf{w}}$. □

It follows from Lemma 3 that

$$\begin{aligned}\frac{1}{4}\Delta(\Psi) =& \eta\mathbb{E}[r^2\lambda^2\cos^2\vartheta] - \eta^2\mathbb{E}[r^4\lambda^2\cos^2\vartheta] \\ &- \eta^2\Psi^2\mathbb{E}[r^2]\end{aligned} \tag{9}$$

We can now state the main theorem of this section.

Theorem 1. *At time t the expected convergence rate for training point $\mathbf{x}$ is monotonically decreasing with the Difficulty Score Ψ of $\mathbf{x}$. If the step size coefficient is sufficiently small so that $\eta \leq \frac{\mathbb{E}[r^2\cos^2\vartheta]}{\mathbb{E}[r^4\cos^2\vartheta]}$, it is likewise monotonically increasing with the distance λ between the current estimate of the hypothesis $\mathbf{w}_t$ and the optimal hypothesis $\bar{\mathbf{w}}$.*

Proof. From (9)

$$\frac{\partial\Delta(\Psi)}{\partial\Psi} = -8\eta^2\mathbb{E}[r^2]\Psi \leq 0$$

which proves the first statement. In addition,

$$\frac{\partial\Delta(\Psi)}{\partial\lambda} = 8\eta\lambda\left(\mathbb{E}[r^2\cos^2\vartheta] - \eta\mathbb{E}[r^4\cos^2\vartheta]\right)$$

If $\eta \leq \frac{\mathbb{E}[r^2\cos^2\vartheta]}{\mathbb{E}[r^4\cos^2\vartheta]}$ then $\frac{\partial\Delta(\Psi)}{\partial\lambda} \geq 0$, and the second statement follows. □

Corollary 1. *Although $\mathbb{E}[\Delta(\Psi)]$ may be negative, $\mathbf{w}_t$ always converges faster to $\bar{\mathbf{w}}$ when the training points are sampled from easier examples with smaller Ψ.*

Corollary 2. *If the step size coefficient η is small enough so that $\eta \leq \frac{\mathbb{E}[r^2\cos^2\vartheta]}{\mathbb{E}[r^4\cos^2\vartheta]}$, we should expect faster convergence at the beginning of SCL.*

[2]Below the short-hand notation $\mathbb{E}[(\pm\Psi)]$ implies that the 2 cases of $y(\mathbf{x}, \Psi) = \mathbf{x}^t\bar{\mathbf{w}} \pm \Psi$ should be considered, with equal probability $\frac{1}{2}$ by assumption.

CONVERGENCE RATE INCREASES WITH CURRENT LOSS

The main theorem in this sub-section states that for a fixed difficulty score Ψ, when the gradient step is small enough, convergence is monotonically *increasing* with the loss of the point with respect to the current hypothesis. *This is not true in general.* The second theorem in this section shows that when the difficulty score is not fixed, there exist hypotheses $\mathbf{w} \in \mathcal{H}$ for which the convergence rate is decreasing with the current loss.

Let $\Upsilon^2 = L(\mathbf{X}_i, \mathbf{w}_t)$ denote the loss of $\mathbf{X}_i$ with respect to the current hypothesis $\mathbf{w}_t$. Define the angle $\beta \in [0, \frac{\pi}{2})$ as follows (see Fig. 1)

$$\beta = \beta(r, \Psi, \lambda) = \arccos(\min(\frac{\Psi}{\lambda r}, 1)) \tag{10}$$

Lemma 4. *The relation between $\Upsilon, \Psi, r, \vartheta$ can be written separately in 4 regions as follows (see Fig. 1):*

A1 $0 \leq \vartheta \leq \pi - \beta,\ y_i = \mathbf{x}_i^t\bar{\mathbf{w}} + \Psi \implies y_i = \mathbf{x}_i^t\mathbf{w}_t + \Upsilon,$
$\lambda r\cos\vartheta = \mathbf{x}_i^t(\bar{\mathbf{w}} - \mathbf{w}_t) = -\Psi + \Upsilon$

A2 $\pi - \beta \leq \vartheta \leq \pi,\ y_i = \mathbf{x}_i^t\bar{\mathbf{w}} + \Psi \implies y_i = \mathbf{x}_i^t\mathbf{w}_t - \Upsilon,$
$\lambda r\cos\vartheta = -\Psi - \Upsilon$

A3 $0 \leq \vartheta \leq \beta,\ y_i = \mathbf{x}_i^t\bar{\mathbf{w}} - \Psi \implies y_i = \mathbf{x}_i^t\mathbf{w}_t + \Upsilon,$
$\lambda r\cos\vartheta = \Psi + \Upsilon$

A4 $\beta \leq \vartheta \leq \pi,\ y_i = \mathbf{x}_i^t\bar{\mathbf{w}} - \Psi \implies y_i = \mathbf{x}_i^t\mathbf{w}_t - \Upsilon,$
$\lambda r\cos\vartheta = \Psi - \Upsilon$

Proof. We keep in mind that $\forall\mathbf{x}_i$ and Ψ, there are 2 possible y_i with equal probability. Recall that $\bar{\mathbf{z}}$ denotes the projection of $\bar{\mathbf{w}}$ on Ω_i. In the planar section shown in Fig. 1,

$$\bar{\mathbf{z}} \text{ lies in the upper half space} \iff y_i = \mathbf{x}_i^t\bar{\mathbf{w}} + \Psi$$

$$\bar{\mathbf{z}} \text{ lies in the lower half space} \iff y_i = \mathbf{x}_i^t\bar{\mathbf{w}} - \Psi$$

This follows from 3 observations: $\bar{\mathbf{x}}_i$ lies in the upper half space by the definition of the polar coordinate system, $\mathbf{x}_i^t\bar{\mathbf{w}} - y_i = \pm\Psi$, and

$$0 = \mathbf{x}_i^t\bar{\mathbf{z}} - y_i = \mathbf{x}_i^t(\bar{\mathbf{z}} - \bar{\mathbf{w}}) + \mathbf{x}_i^t\bar{\mathbf{w}} - y_i$$

Next, let $\mathbf{z}_t$ denote the projection of $\mathbf{w}_t$ on Ω_i. Then

$$0 = \mathbf{x}_i^t\mathbf{z}_t - y_i = \mathbf{x}_i^t(\mathbf{z}_t - \mathbf{w}_t) + \mathbf{x}_i^t\mathbf{w}_t - y_i$$

When $\bar{\mathbf{z}}$ lies in the upper half space, the following can be verified geometrically from Fig. 1:

$$0 \leq \vartheta \leq \pi - \beta \ \Rightarrow\ \mathbf{x}_i^t(\mathbf{z}_t - \mathbf{w}_t) \geq 0 \ \Rightarrow\ y_i = \mathbf{x}_i^t\mathbf{w}_t + \Upsilon$$

$$\pi - \beta \leq \vartheta \leq \pi \ \Rightarrow\ \mathbf{x}_i^t(\mathbf{z}_t - \mathbf{w}_t) \leq 0 \ \Rightarrow\ y_i = \mathbf{x}_i^t\mathbf{w}_t - \Upsilon$$

When $\bar{\mathbf{z}}$ lies in the lower half space

$$0 \le \vartheta \le \beta \implies \mathbf{x}_i^t(\mathbf{z}_t - \mathbf{w}_t) \ge 0 \implies y_i = \mathbf{x}_i^t\mathbf{w}_t + \Upsilon$$
$$\beta \le \vartheta \le \pi \implies \mathbf{x}_i^t(\mathbf{z}_t - \mathbf{w}_t) \le 0 \implies y_i = \mathbf{x}_i^t\mathbf{w}_t - \Upsilon$$

□

Next we analyze how the convergence rate at $\mathbf{x}_i$ changes with Υ. Let $\Delta(\Psi, \Upsilon)$ denote the expected convergence rate at time t, given fixed difficulty score Ψ and fixed loss Υ. From (9) $\Delta(\Psi, \Upsilon) = 4\eta\mathbb{E}[r^2\lambda^2\cos^2\vartheta/\Upsilon] + O(\eta^2)$.

It is easier to analyze $\Delta(\Psi, \Upsilon)$ when using the Cartesian coordinates, rather than polar, in the $2D$ plane defined by the vectors $\vec{\mathcal{O}} = \bar{\mathbf{w}} - \mathbf{w}_t$ and $\bar{\mathbf{z}} - \bar{\mathbf{w}}$ (see Fig. 1); thus we define $u = r\cos\vartheta,\ v = r\sin\vartheta$. The 4 cases listed in Lemma 4 can be readily transformed to this coordinate system as follows $\{0 \le \vartheta \le \beta\} \Leftrightarrow \{\lambda u \ge \Psi\}$, $\{\beta \le \vartheta \le \pi - \beta\} \Leftrightarrow \{-\Psi \le \lambda u \le \Psi\}$, and $\{\pi - \beta \le \vartheta \le \pi\} \Leftrightarrow \{\lambda u \le -\Psi\}$:

A1 $\lambda u \ge -\Psi:\ \ \lambda u = -\Psi + \Upsilon$

A2 $\lambda u \le -\Psi:\ \ \lambda u = -\Psi - \Upsilon$

A3 $\lambda u \ge \Psi:\ \ \lambda u = \Psi + \Upsilon$

A4 $\lambda u \le \Psi:\ \ \lambda u = \Psi - \Upsilon$

Define

$$\nabla = \frac{f(\frac{\psi+\Upsilon}{\lambda}) - f(\frac{\psi-\Upsilon}{\lambda}) - f(\frac{-\psi+\Upsilon}{\lambda}) + f(\frac{-\psi-\Upsilon}{\lambda})}{f(\frac{\psi+\Upsilon}{\lambda}) + f(\frac{\psi-\Upsilon}{\lambda}) + f(\frac{-\psi+\Upsilon}{\lambda}) + f(\frac{-\psi-\Upsilon}{\lambda})}$$

Clearly $-1 \le \nabla \le 1$.

Theorem 2. *Assume that the gradient step size is small enough so that we can neglect second order terms $O(\eta^2)$, and that $\frac{\partial\nabla}{\partial\Upsilon} \ge \frac{\psi}{\Upsilon} - \frac{\Upsilon}{\psi}\ \forall\Upsilon$. Fix the difficulty score at Ψ. At time t the expected convergence rate is monotonically increasing with the loss Υ of the training point $\mathbf{x}$.*

Proof. In the coordinate system defined above $\Delta(\Psi, \Upsilon) = 4\eta\mathbb{E}[\lambda^2u^2/\Upsilon] + O(\eta^2)$. We compute $\Delta(\Psi, \Upsilon)$ separately in each region, marginalizing out v based on the following

$$\int\int_0^\infty \lambda^2u^2v^{d-1}f(u,v)dvdu = \int \lambda^2u^2f(u)du$$

where $f(u)$ denotes the marginal distribution of u.

Let u_i denote the value of u corresponding to loss Υ in each region A1-A4, and $\frac{1}{2}f(u_i)$ its density. $\Delta(\Psi, \Upsilon)$ takes 4 discrete values, one in each region, and its expected value is therefore $\Delta(\Psi, \Upsilon) = 4\eta\sum_{i=1}^4 \lambda^2u_i^2\frac{f(u_i)}{\sum_{i=1}^4 f(u_i)}$. It can readily be shown that

$$\frac{1}{4\eta}\Delta(\psi, \Upsilon) = \psi^2 + \Upsilon^2 + 2\psi\Upsilon\nabla \tag{11}$$

and subsequently

$$\begin{aligned}\frac{1}{4\eta}\frac{\partial\Delta(\psi,\Upsilon)}{\partial\Upsilon} &= 2\Upsilon + 2\psi\Upsilon\frac{\partial\nabla}{\partial\Upsilon} + 2\psi\nabla \\ &\ge 2\Upsilon + 2\psi\Upsilon\frac{\partial\nabla}{\partial\Upsilon} - 2\psi\end{aligned} \tag{12}$$

Using the assumption that $\frac{\partial\nabla}{\partial\Upsilon} \ge \frac{\psi}{\Upsilon} - \frac{\Upsilon}{\psi}\ \forall\Upsilon$, we have that

$$\frac{1}{8\eta}\frac{\partial\Delta(\psi,\Upsilon)}{\partial\Upsilon} \ge \Upsilon + \psi\Upsilon\frac{\psi - \Upsilon}{\psi\Upsilon} - \psi = 0$$

□

Corollary 3. *For any $c \in \mathbb{R}^+$, if ∇ is $(c - \frac{1}{c})$-lipschitz then $\frac{\partial\Delta(\psi,\Upsilon)}{\partial\Upsilon} \ge 0$ for any $\Upsilon \ge c\,\psi$.*

Corollary 4. *If $\mathcal{D}(\mathbb{X}/_\Psi) = k(\Psi)$ over a compact region and η small enough, then $\frac{\partial\Delta(\psi,\Upsilon)}{\partial\Upsilon} \ge 0$ for all Υ excluding the boundaries of the compact region. If in addition $\Upsilon > \Psi$, then $\frac{\partial\Delta(\psi,\Upsilon)}{\partial\Upsilon} \ge 0$ almost surely.*

Theorem 3. *Assume that $\mathcal{D}(\mathbb{X})$ is continuous and that $\bar{\mathbf{w}}$ is realizable. Then there are always hypotheses $\mathbf{w} \in \mathcal{H}$ for which the expected convergence rate under $\mathcal{D}(\mathbb{X})$ is monotonically decreasing with the loss Υ of the sampled points.*

Proof. We shift to a hyperspherical coordinate system in $\mathbb{R}^{d+1}$ similar as before, but now the pole (origin) is fixed at $\mathbf{w}_t$. For the gradient step $\mathbf{s}$, it can be shown that:

$$\begin{aligned}\mathbf{s} &= -\operatorname{sgn}(\mathbf{x}_i^t\mathbf{w}_t - y_i)2\eta\mathbf{x}_i\Upsilon \\ \mathbf{s}_\mathcal{O} &= \mathbf{s}\cdot\frac{\bar{\mathbf{w}} - \mathbf{w}_t}{\lambda} = \pm\frac{2\eta}{\lambda}r\lambda\cos\vartheta\,\Upsilon\end{aligned} \tag{13}$$

Let $\Delta(\Upsilon)$ denote the expected convergence rate at time t, given a fixed loss Υ. From Lemma 2

$$\begin{aligned}\Delta(\Upsilon) &= 2\eta\Upsilon\Big(\mathbb{E}[r\cos\vartheta/_{\mathbf{x}_i^t\mathbf{w}_t - y_i = -\Upsilon}] - \\ &\qquad \mathbb{E}[r\cos\vartheta/_{\mathbf{x}_i^t\mathbf{w}_t - y_i = \Upsilon}]\Big) - \mathbb{E}[(2\eta r\Upsilon)^2] \\ &\triangleq 2\eta\Upsilon Q(r, \vartheta, \mathbf{w}_t) - 4\eta^2\Upsilon^2\mathbb{E}[r^2]\end{aligned}$$

If $\mathbf{w} = \bar{\mathbf{w}}$, then $Q(r, \vartheta, \mathbf{w}) = 0$ from the symmetry of $\mathcal{D}(\mathbb{X})$ with respect to Ψ. From the continuity of $\mathcal{D}(\mathbb{X})$, there exists $\delta > 0$ such that if $\|\mathbf{w} - \bar{\mathbf{w}}\|_2 < \delta$, then $\|Q(r, \vartheta, \mathbf{w}) - Q(r, \vartheta, \bar{\mathbf{w}})\|_2 < \eta\Upsilon\mathbb{E}[r^2]$, which implies that $\Delta(\Upsilon) < -2\eta^2\Upsilon^2\mathbb{E}[r^2] < 0$. □

2.3. Deep learning: simulation results

While the corollaries above apply to a rather simple situation, when using the *Difficulty Score* to guide SGD while

minimizing the convex regression loss, their predictions can be empirically tested with the deep learning architecture and loss which are described in Section 3. There an additional challenge is posed by the fact that the empirical ranking is not based on the ideal definition given in Def. 1, but rather on an estimate derived from another classifier.

Still, the empirical results as shown in Fig. 2 demonstrate agreement with the theoretical analysis of the linear regression loss. Specifically, in epoch 0 there is a big difference between the average errors in estimating the gradient direction, which is smallest for the easiest examples and highest for the most difficult examples as predicted by Corollary 1. This difference in significantly reduced after 10 epochs, and becomes insignificant after 20 epochs, in agreement with Corollary 2.

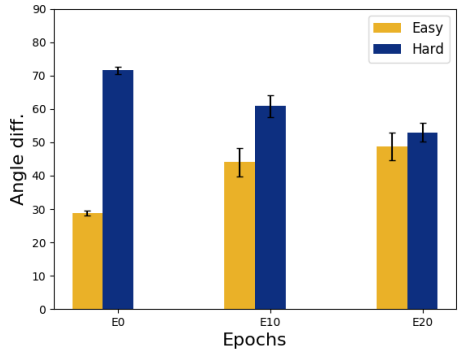

Figure 2. Results using the empirical setup described in Section 3. The average angular difference (in degrees) between the gradient computed based on a batch of 100 examples, and the true gradient based on all the training examples, is shown for 2 cases: the easiest examples (yellow) and the most difficult training examples (blue). Standard error bars are plotted based on 100 repetitions. Three Condition are shown: beginning of training (E0), 10 epochs into training (E10), and 20 epochs into training (E20).

Discussion. Fig. 2 shows that the variance in the direction of the gradient step defined by easier points is significantly smaller than that defined by difficult points, especially at the beginning of training. This is advantageous when the initial point $\mathbf{w}_0$ does not lie in the basin of attraction of the desired global minimum $\bar{\mathbf{w}}$, and if, in agreement with Lemma 1, the pronounced shared component of the easy gradient steps points in the direction of the global minimum, or a more favorable local minimum; then the likelihood of escaping the local minimum decreases with a point's *Difficulty Score*. This scenario suggests another possible advantage for curriculum learning at the initial stages of training.

3. Curriculum learning in deep networks

As discussed in the introduction, a practical curriculum learning method should address two main questions: how to rank the training examples, and how to modify the sampling procedure based on this ranking. Solutions to these issues are discussed in Section 3.1. In Section 3.2 we discuss the empirical evaluation of our method.

3.1. Method

Ranking examples by knowledge transfer

The main novelty of our proposed method lies in this step, where we rank the training examples by estimated difficulty in the absence of human supervision. Difficulty is estimated based on knowledge transfer from another classifier. Here we investigate transfer from a more powerful learner.

It is a common practice now to treat one of the upstream layers of a pre-trained network as a representation (or embedding) layer. This layer activation is then used for representing similar objects and train a simpler classifier (such as SVM, or shallower NNs) to perform a different task, related but not identical to the original task the network had been trained on. In computer vision such embeddings are commonly obtained by training a deep network on the recognition of a very large database such as ImageNet (Deng et al., 2009). These embeddings have been shown to provide better semantic representations of images (as compared to more traditional image features) in a number of related tasks, including the classification of small datasets (Sharif Razavian et al., 2014), image annotation (Donahue et al., 2015) and structured predictions (Hu et al., 2016).

Following this practice, the activation in the penultimate layer of a large and powerful pre-trained network is the loci of knowledge transfer from one network to another. Repeatedly, as in (Sharif Razavian et al., 2014), it has been shown that competitive performance can be obtained by training a shallow classifier on this representation in a new related task. Here we propose to use the confidence of such a classifier, e.g. the margin of an SVM classifier, as the estimator for the difficulty of each training example. This measure is then used to sort the training data. We note that unlike the traditional practice of reusing a pre-trained network, here we only transfer information from one learner to another. The goal is to achieve a smaller classifier that can conceivably be used with simpler hardware, without depending on access to the powerful learner at test time.

Scheduling the appearance of training examples

In agreement with prior art, e.g. the definition of curriculum in (Bengio et al., 2009), we investigate curriculum learning

where the scheduling of examples changes with time, giving priority to easier examples at the beginning of training. We explored two variants of the basic scheduling idea:

Fixed. The distribution used to sample examples from the training data is gradually changed in fixed steps. Initially all the weight is put on the easiest examples. In subsequent steps the weight of more difficult examples is gradually increased, until the final step in which the training data is sampled uniformly (or based on some prior distribution on the training set).

Adaptive. Similar to the previous mode, but where the length of each step is not fixed, but is being determined adaptively based on the current loss of the training data.

3.2. Empirical evaluation

EXPERIMENTAL SETUP

Datasets. For evaluation we used 2 data sets: CIFAR-100 (Krizhevsky & Hinton, 2009) and STL-10 (Coates et al., 2010). In all cases, as is commonly done, the data was pre-processed using global contrast normalization; cropping and flipping were used for STL-10.

Network architecture. We used convolutional Neural Networks (CNN) which excel at image classification tasks. Specifically, we used two architectures which are henceforth denoted *Large* and *Small*, in accordance with the number of parameters. The *Large* network is comprised of four blocks, each with two convolutional layers, ELU activation, and max-pooling. This is followed by a fully connected layer, for a total of 1,208,101 parameters. The *Small* network consists of only three hidden layers, for a total of 4,557 parameters. During training, we applied dropout and $l2$ regularization on the weights, and used either SGD or ADAM to optimize the cross-entropy loss.

Scheduling mechanisms: control. As described above, our method is based on a scheduling design which favors the presentation of easier examples at the beginning of training. In order to isolate the contribution of scheduling by increasing level of difficulty as against other spurious consequences of data scheduling, we compared performance with the following control conditions: `control-curriculum`, identical scheduling mechanism but where the underlying ranking of the training examples is random and unrelated to estimated difficulty; and `anti-curriculum`, identical scheduling mechanism but favoring the more difficult examples at the beginning of training.

CONTROLLING FOR TASK DIFFICULTY

Evidence from prior art is conflicting regarding where the benefits of curriculum learning lie, which is to be expected given the variability in the unknown sources of the curriculum supervision information and its quality. We observed in our empirical study that the benefits depended to a large extent on the difficulty of the task. We always saw faster learning at the beginning of the training process, while lower generalization error was seen only when the task was relatively difficult. We therefore employed controls for the following 3 sources of task difficulty:

Inherent task difficulty. To investigate this factor, we take advantage of the fact that CIFAR-100 is a hierarchical dataset with 100 classes and 20 super-classes, each including 5 member classes. We therefore trained a network to discriminate the 5 member classes of 2 super-classes as 2 separate tasks: 'small mammals' (task 1) and 'aquatic mammals' (task 2). These are expected to be relatively hard learning tasks. We also trained a network to discriminate 5 random well separated classes: 'camel', 'clock', 'bus', 'dolphin' and 'orchid' (task 3). This task is expected to be relatively easy.

Size of classification network. For a given task, classification performance is significantly affected by the size of the network and its architecture. We assume, of course, that we operate in the domain where the number of model parameters is smaller than can be justified by the training data (i.e., there is no overfit). We therefore used networks of different sizes in order to evaluate how curriculum learning is affected by *task difficulty* as determined by the network's strength (see Fig. 3a-b). In this comparative evaluation, the smaller the network is, the more difficult the task is likely to be (clearly, many other factors participate in the determination of task difficulty).

Regularization and optimization. Regularization is used to constrain the family of hypotheses, or models, so that they possess such desirable properties as smoothness. Regularization effectively decreases the number of degrees of freedom in the model. In fact, most optimization methods, other then vanilla stochastic gradient descent, incorporate some form of regularization and smoothing, among other inherent properties. Therefore the selection of optimization method also plays a role in determining the effective size of the final network.

RESULTS

Fig. 3a shows typical results when training the *Large* CNN (see network's details above) to classify a subset of 5 CIFAR100 images (task 1 as defined above), using slow learning rate and Adam optimization. In this setup we see that curriculum learning speeds up the learning rate at the beginning of the training, but converges to the same performance as regular training. When we make learning more difficulty by using the *Small* network, performance naturally decreases, but now we see that curriculum learning also improves the final generalization performance (Fig. 3b).

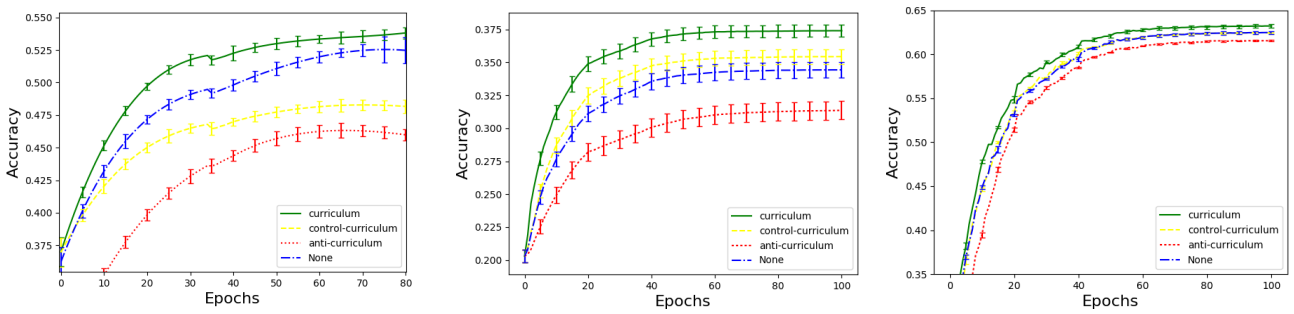

Figure 3. Accuracy in test classification as a function of training time. We show results with 4 scheduling methods: our method denoted `curriculum` (solid green), `control-curriculum` with random ordering (dashed yellow), `anti-curriculum` where more difficult examples are preferred at the beginning of training (dotted red), and `none` with no curriculum learning (dashdotted blue). a-b) Learning CIFAR100 task 1, where the *Large* network is used in a) and the *Small* network in b). c) Learning to classify STL-10 images.

Similar results are shown for the STL-10 dataset (Fig. 3c).

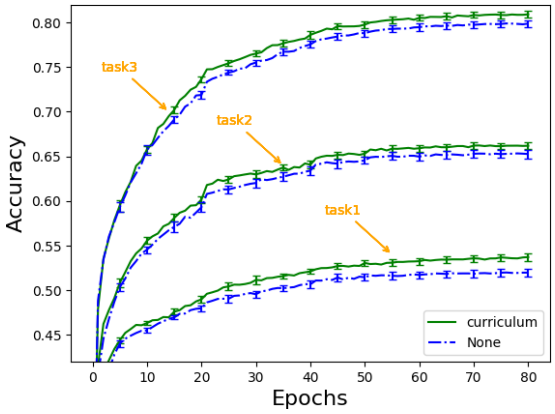

Figure 4. Accuracy in test classification in 3 tasks each involving 5 classes from the CIFAR100 datasets (see Section 3.2). In each task, the performance of learning with a curriculum (solid green) and without (dashdotted blue) is shown.

Fig. 4 shows comparative results when controlling for inherent task difficulty in the 3 tasks described above, using faster learning rate and SGD optimization. Task difficulty can be evaluated in retrospect from the final performance seen in each plot. As can be clearly seen in the figure, the improvement in final accuracy with curriculum learning is larger when the task is more difficult. When manipulating the level of regularization, we see that while too much regularization always harms performance, curriculum learning is least affected by this degradation (results are omitted).

4. Summary and Discussion

We investigated curriculum learning, an extension of stochastic gradient descent in which easy examples are more frequently sampled at the beginning of training. We started with the theoretical investigation of this strict definition in the context of linear regression, showing that curriculum learning accelerates the learning rate in agreement with prior empirical evidence. While not shedding light on its affect on the classifier's final performance, our analysis suggests that the direction of a gradient step based on "easy" examples may be more effective in traversing the input space towards the ideal minimum of the loss function. Specifically, we have empirically shown that the variance in the gradient direction of points increases with their difficulty when optimizing a non-convex loss function. Over-sampling the more coherent easier examples may therefore increase the likelihood to escape the basin of attraction of a low quality local minimum in favor of higher quality local minima even in the general non-convex case.

We also showed theoretically that when the difficulty score of the training points is fixed, convergence is faster if the loss with respect to the current hypothesis is higher. This seems to be a very intuitive result, an intuition that underlies the boosting method for example. However, as intuitive as it might be, this is not always true when the prior data density is assumed to be continuous and when the optimal hypothesis is realizable. Thus the requirement that the difficulty score is fixed is necessary.

In the second part of this paper we described a curriculum learning method for deep networks. The method relies on knowledge transfer from other (pre-trained) networks in order to rank the training examples by difficulty. We described extensive experiments where we evaluated our proposed method under different task difficulty conditions and against a variety of control conditions. In all cases curriculum learning has been shown to increase the rate of convergence at the beginning of training, in agreement with the theoretical results. With more difficult tasks, curriculum learning improved generalization performance.

Acknowledgements

This work was supported in part by a grant from the Israel Science Foundation (ISF) and by the Gatsby Charitable Foundations.

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pp. 173–182, 2016.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, 2009.

Chen, W., Wilson, J., Tyree, S., Weinberger, K., and Chen, Y. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*, pp. 2285–2294, 2015.

Coates, A., Lee, H., and Ng, A. Y. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001(48109):2, 2010.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

Hu, H., Zhou, G.-T., Deng, Z., Liao, Z., and Mori, G. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2960–2968, 2016.

Jesson, A., Guizard, N., Ghalehjegh, S. H., Goblot, D., Soudan, F., and Chapados, N. Cased: Curriculum adaptive sampling for extreme data imbalance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 639–646. Springer, 2017.

Kim, Y.-D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Krueger, K. A. and Dayan, P. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009.

Mitchell, T. M. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ. New Jersey, 1980.

Mitchell, T. M. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

Shrivastava, A., Gupta, A., and Girshick, R. B. Training region-based object detectors with online hard example mining. *CoRR*, abs/1604.03540, 2016. URL http://arxiv.org/abs/1604.03540.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Thrun, S. and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.

Wang, P. and Cottrell, G. W. Basic level categorization facilitates visual object recognition. *arXiv preprint arXiv:1511.04103*, 2015.

Zaremba, W. and Sutskever, I. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.