Example Generation Under Constraints Using Cascade Correlation Neural Nets

Ardavan S. Nobandegani[1,3] **Thomas R. Shultz**[2,3]
{ardavan.salehinobandegani@mail.mcgill.ca, thomas.shultz@mcgill.ca}
[1]Department of Electrical & Computer Engineering, McGill University
[2]School of Computer Science, McGill University
[3]Department of Psychology, McGill University

Abstract

Humans not only can effortlessly imagine a wide range of novel instances and scenarios when prompted (e.g., a new shirt), but more remarkably, they can adequately generate examples which satisfy a given set of constraints (e.g., a new, dotted, pink shirt). Recently, Nobandegani and Shultz (2017) proposed a framework which permits converting deterministic, discriminative neural nets into probabilistic generative models. In this work, we formally show that an extension of this framework allows for generating examples under a wide range of constraints. Furthermore, we show that this framework is consistent with developmental findings on children's generative abilities, and can account for a developmental shift in infants' probabilistic learning and reasoning. We discuss the importance of integrating Bayesian and connectionist approaches to computational developmental psychology, and how our work contributes to that research.

Keywords: Cascade correlation neural networks; Deterministic discriminative models; Probabilistic generative models; Bayesian vs. connectionist modeling of development

1 Introduction

Can you imagine a pink shirt? How about a dotted pink shirt? Or a slim, dotted, pink shirt? Humans often can not only effortlessly imagine a wide range of novel instances and scenarios when prompted (e.g., a shirt), but more remarkably, they can adequately imagine exemplars which satisfy a given set of constraints (e.g., a dotted, pink shirt), suggesting flexible generative abilities. For example, when given incomplete sentences or fragments of a picture, people can generate possible completions (Sanborn & Chater, 2016). However, example generation under some constraints appears to be easier than others (e.g., generating a word ending in `ing` vs. generating a nine-letter word whose third, fifth, and seventh letters are `x,a,o,` respectively), leading us sometimes to settle for partially satisfied constraints (e.g., a word whose third letter is `x`, with the other constraints dropped).

But how could human abilities of example generation under constraints be formalized in computational terms? Given that example generation can be computationally characterized in terms of *sampling* from some underlying probability distribution (e.g., Nobandegani & Shultz, 2017; Jern & Kemp, 2013), constraints can be viewed as an inductive bias or a prior distribution over the domain of interest.

Bridging the computational, algorithmic, and implementational levels of analysis (Marr, 1982), Nobandegani and Shultz (2017) presented a framework which allows for converting cascade correlation neural networks (CCNNs) (Fahlman & Lebiere, 1989) into probabilistic generative models. CCNNs are a well-known class of self-organized, discriminative (as opposed to generative) models that have been successful in simulating a variety of phenomena in the developmental literature, e.g., infant learning of word-stress patterns in artificial languages (Shultz & Bale, 2006), syllable boundaries (Shultz & Bale, 2006), visual concepts (Shultz, 2006), and have also been successful in capturing important developmental regularities in a variety of tasks, e.g., the balance scale task (Shultz, Mareschal, & Schmidt, 1994; Shultz & Takane, 2007), transitivity (Shultz & Vogel, 2004), conservation (Shultz, 1998), and seriation (Mareschal & Shultz, 1999).

In this work, we formally show that an extension of the Nobandegani and Shultz (2017) framework allows for probabilistically generating examples under a wide range of constraints. We show how both equality and inequality constraints can be effectively incorporated into that framework in the form of prior distributions, thereby tailoring the generated samples into regions of interest. Also, we formally demonstrate how hard-to-satisfy constraints can be encoded in a relaxed fashion (i.e., by *softening* those constraints) such that they can be partially satisfied. As suggested by Nobandegani and Shultz (2017), converting CCNNs into generative models also gives rise to the notion of *self-organized, probabilistic generative models*: probabilistic generative models possessing the self-constructive property of CCNNs. Such self-organized generative models could provide a wealth of quantitative developmental hypotheses as to how the generative and probabilistic abilities of children change over development. We show that the Nobandegani and Shultz (2017) framework is consistent with developmental findings on children's generative abilities, and can account for a developmental shift in infants' probabilistic learning and reasoning.

After a brief overview of CCNNs and the main ingredients of the Nobandegani and Shultz (2017) framework, we formally present an extension of our framework enabling it to generate examples under a wide range of constraints, followed by extensive simulations confirming the efficacy of the proposed extension. We then turn our attention to features of Nobandegani and Shultz's (2017) framework which are of significance for the developmental literature, and conclude by discussing the importance of integrating Bayesian and connectionist approaches to computational developmental psychology, and how the research presented here may contribute to that line of work.

2 Background

Cascade-Correlation Neural Nets (CCNNs): CCNNs are a special class of deterministic, discriminative neural net-

works, which construct their topology in an autonomous fashion—an appealing property for simulating developmental phenomena (Westermann et al., 2006). CCNN training starts with a two-layer network (i.e., the input and the output layer) with no hidden units, and proceeds by recruiting hidden units one at a time, as needed. Each new hidden unit is trained to correlate with residual error in the network built so far, and is recruited into a hidden layer of its own, giving rise to a deep network with as many hidden layers as the number of recruited hidden units. CCNNs use sum-of-squared error as an objective function, and typically use symmetric (with range -0.5 to +0.5) or asymmetric (with range 0 to +1) sigmoidal activation functions for hidden and output units. Some variants have been proposed: Sibling-Descendant Cascade-Correlation (SDCC) (Baluja & Fahlman, 1994) and Knowledge-Based Cascade-Correlation (KBCC) (Shultz & Rivest, 2001). Although this work focuses on standard CCNN and SDCC, the proposed extension is applicable to KBCC as well.

Nobandegani and Shultz (2017) Framework: Nobandegani and Shultz (2017) presented a framework that converts CCNNs into probabilistic generative models. Importantly, the framework bridges computational, algorithmic, and implementational levels of analysis (Marr, 1982). Concretely, this framework induces a probability distribution $p(\mathbf{X}|\mathbf{Y})$ (in the form a Gibbs distribution for non-probabilistic energy-based models) on the deterministic input-output mapping $f(\mathbf{X};W^*)$ learned by a CCNN, and uses a Markov chain Monte Carlo (MCMC) method to sample from the induced distribution:[1]

$$p(\mathbf{X}|\mathbf{Y}=Y) = \frac{1}{Z_1}\exp(-\beta||Y - f(\mathbf{X};W^*)||_2^2), \tag{1}$$

where $||\cdot||_2$ denotes the l_2-norm, $\beta \in \mathbb{R}_+$ is a damping factor, W^* the set of weights for a CCNN after training, and Z_1 is a normalizing constant (aka partition factor).

It is worth noting that this framework allows for converting *any* deterministic, discriminative neural network into a probabilistic generative model.

3 Example Generation Under Constraints

In this section, we present a formal extension of our framework, allowing probabilistic generation of examples under a wide range of equality and inequality constraints. As noted earlier, the notion of example generation under constraints computationally amounts to sampling from a distribution, with constraints serving as an inductive bias or a prior distribution over the domain of interest. To further flesh out this understanding, we formally show that a wide range of equality and inequality constraints (or soft variants thereof) can be encoded, as an inductive bias, into the expression given in (1) in the following generic format:

$$p(\mathbf{X}|\mathbf{Y}=L_j) = \frac{1}{Z_2}\exp\big(-\beta\big\{||L_j - f(\mathbf{X};W^*)||_2^2 + \gamma\,\phi(\mathbf{X})\big\}\big), \tag{2}$$

with $\phi(\mathbf{X})$ compactly encoding the set of constraints of interest, $\gamma \in \mathbb{R}_+$ denoting a *trade-off factor*, and L_j a vector whose element corresponding to the desired class is +0.5 (i.e., its j^{th} element) and the rest of its elements are -0.5s. (Likewise, in case the activation function of the output units are asymmetric sigmoidals, L_j denotes a vector whose element corresponding to the desired class is +1 and the rest of its elements are 0s.) Concretely, the parameter γ moderates how deviations from the desired class (encoded in the term $||L_j - f(\mathbf{X};W^*)||_2^2$) and deviations from the constraints (compactly encoded in $\phi(\mathbf{X})$) should trade off.

Next, we formally articulate how (2) lets us handle a wide range of equality (Sec. 3.1) and an arbitrary set of inequality (Sec. 3.2) constraints, in a unified fashion.

3.1 Handling Hard and Soft Equality Constraints

Consider the general case of having a CCNN with $n \in \mathbb{N}$ inputs and $m \in \mathbb{N}$ outputs, i.e., $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$. A broad class of equality constraints corresponds to having a subset of $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{x}_j\}_j$ (i.e., $\{\mathbf{x}_j\}_j \subseteq \{\mathbf{x}_i\}_{i=1}^n$), expressible as functions of the remaining variables $\mathbf{X} \setminus \{\mathbf{x}_j\}_j \triangleq: \mathbf{X}_{\text{rest}}$. That is, formally, $\forall j:\ \mathbf{x}_j = c_j(\mathbf{X}_{\text{rest}})$, with $c_j(\cdot)$ denoting the corresponding function. Note that, in the simplest case, $c_j(\cdot)$ can be a constant function, corresponding to setting $\mathbf{x}_j$ to a fixed value. Then, all these equality constraints can be compactly encoded into (2) by having:

$$p(\mathbf{X}|\mathbf{Y}=L_j) \propto \exp\big(-\beta\big||L_j - f(\mathbf{X}_{\text{rest}}, \{c_k(\mathbf{X}_{\text{rest}})\}_k; W^*)||_2^2\big), \tag{3}$$

Although the above formalism can capture a wide range of equality constraints, not all equality constraints lend themselves to this formulation. For example, consider the equality constraint $\mathbf{x}_1 \sin(\mathbf{x}_2) = 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2})$. A glance at this expression reveals that $\mathbf{x}_1$ cannot be cast as a function of $\mathbf{x}_2$ or vice versa (i.e., separability is not attainable). In such cases, the idea of only *approximately* satisfying a constraint (i.e., softening a constraint) comes into play. For example, instead of aiming at perfectly satisfying the said constraint (i.e., to find pairs of $(\mathbf{x}_1, \mathbf{x}_2)$ which exactly satisfy the constraint), we can strive for approximate satisfaction of the constraint, by finding pairs of $(\mathbf{x}_1, \mathbf{x}_2)$ for which $\big(\mathbf{x}_1 \sin(\mathbf{x}_2) - 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2})\big)^2$ is sufficiently small, with deviations from zero (i.e., *loss*) being penalized quadratically. Assuming that $L(\cdot)$ denotes a (non-negative) loss function, the idea of approximately satisfying a set of equality constraints of arbitrary forms $\{(c_k(\mathbf{X}) = 0)\}_k$ (e.g., $c_1(\mathbf{X}) : \mathbf{x}_1 \sin(\mathbf{x}_2) - 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2})$) can be compactly encoded into (2) by having:

$$\phi(\mathbf{X}) := \sum_k L(c_k(\mathbf{X})). \tag{4}$$

A wide range of loss functions have been entertained in the mathematical statistics, decision theory, optimization, and

[1] Although our framework advocates a particular neurally-plausible and computationally-efficient gradient-based MCMC, called the Metropolis-Adjusted Langevin (MAL) (Savin & Deneve, 2014; Moreno-Bote et al., 2011), it can accommodate *any* MCMC method.

statistical machine leaning literature, e.g., the quadratic loss function, the $0-1$ loss function, the hing loss function, the l_p-norm of the error for various p, etc. The choice of loss function is made based on the the requirements of the task. In Sec. 4 we present several simulations to demonstrate the efficacy of the formalism developed in this section.

3.2 Handling Hard and Soft Inequality Constraints

Drawing further on the formalism developed in Sec. 3.1, in this section we formally articulate how (2) lets us generate examples under an arbitrary set of *in*equality constraints. Let us first consider the case of satisfying arbitrary inequality constraints only approximately (i.e., softened inequality constraints). We then formally discuss the case of satisfying an arbitrary set of inequality constraints exactly (i.e., without any approximation, hence hard inequality constraints).

Before presenting our general result, and to convey the intuition behind it, let us consider our old example, but this time an inequality variant of it: Generating examples under the inequality constraint $\mathbf{x}_1 \sin(\mathbf{x}_2) - 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2}) \leq 0$. Satisfying exactly this inequality constraint would formally amount to having $\phi(\mathbf{X}) := L_\infty(\mathbf{x}_1 \sin(\mathbf{x}_2) - 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2}))$ with the loss function $L_\infty(\cdot)$ defined as:

$$L_\infty(e) = \begin{cases} +\infty & \text{if } e > 0, \\ 0 & \text{if } e \leq 0, \end{cases} \tag{5}$$

wherein any pair $(\mathbf{x}_1, \mathbf{x}_2)$ violating that inequality constraint would be assigned zero likelihood (see expression (2), and note that $\exp(-\infty) = 0$), ensuring that such a pair would never be generated.

A relaxed version the above loss function, $L_\alpha(\cdot)$ ($\forall \alpha \geq 1$), as defined below, allows us to satisfy the said inequality constraint only approximately when generating examples:

$$L_\alpha(e) = \begin{cases} e^\alpha & \text{if } e > 0, \\ 0 & \text{if } e \leq 0, \end{cases} \tag{6}$$

Simply put, $L_\alpha(\cdot)$ penalizes any violations of an inequality constraint polynomially with respect to the error, e.

We are now well-positioned to formally present how (2) allows us to generate examples under an arbitrary set of inequality constraints, with those constraints being only approximately satisfied. Consider again the general case of having a CCNN with $n \in \mathbb{N}$ inputs and $m \in \mathbb{N}$ outputs, i.e., $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$. Let $\{(c_j(\mathbf{X}) \leq 0)\}_j$ denote an arbitrary set of constraints (e.g., in above $c_1(\mathbf{X})$: $\mathbf{x}_1 \sin(\mathbf{x}_2) - 2\exp(\mathbf{x}_1 \sqrt[3]{\mathbf{x}_2})$). Assuming that $L(\cdot)$ denotes a (non-negative) loss function, the problem of approximately satisfying an arbitrary set of inequality constraints $\{(c_j \leq 0)\}_j$ can be compactly encoded by having $\phi(\mathbf{X}) := \sum_j L(c_j)$. Again, the loss function $L(\cdot)$ can be selected from the many available loss function already entertained in the mathematical statistics, decision theory, optimization, and statistical machine leaning literature, or it can be designed according to the specifications and requirements of the example generation task of interest. Also note that, since $c(\mathbf{X}) \geq 0 \iff \bar{c}(\mathbf{X}) \leq 0$ with $\bar{c}(\mathbf{X}) :\triangleq -c(\mathbf{X})$, all forms of inequality can be cast into the formalism developed above.

Finally, note that the problem of exactly satisfying an arbitrary set of inequality constraints $\{(c_j \leq 0)\}_j$ can be compactly encoded by having $\phi(\mathbf{X}) := \sum_j L_\infty(c_j(\mathbf{X}))$, where L_∞ is given in (5). In Sec. 4, we present several simulations to demonstrate the efficacy of the formalism developed in this section.

4 Simulations

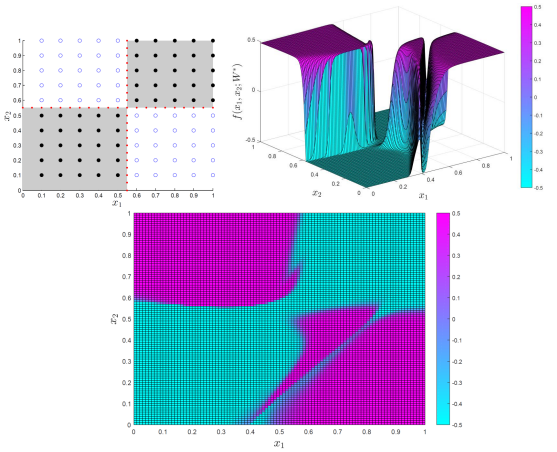
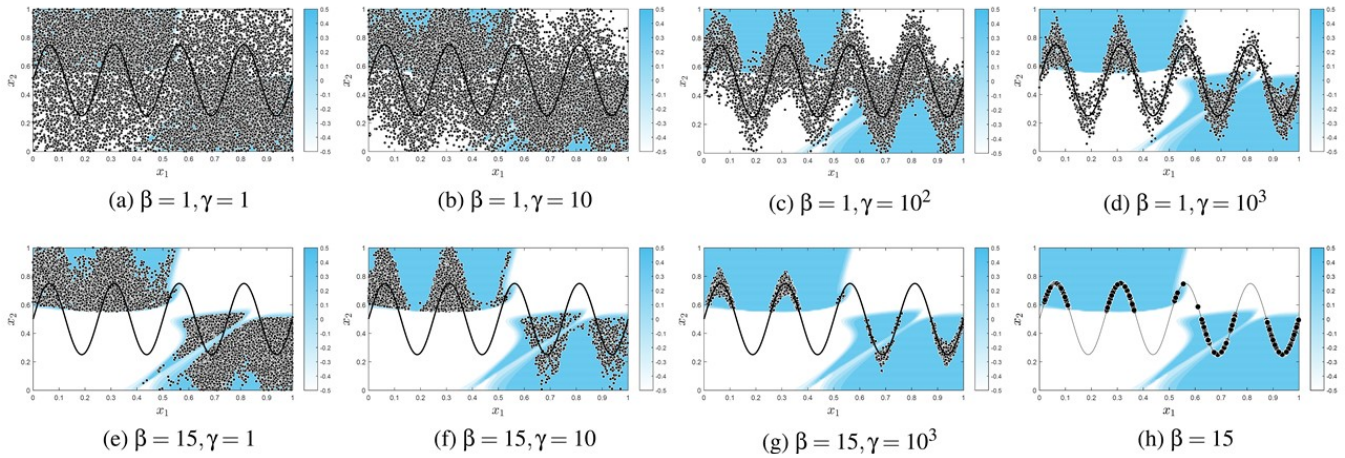


Figure 1: A CCNN trained on the continuous-XOR classification task. Top-left: Training patterns. All the patterns in the gray quadrants are negative examples with label -0.5, and all the patterns in the white quadrants are positive examples with label +0.5. Red dotted lines depict the boundaries. Top-right: The input-output mapping, $f(x_1, x_2; W^*)$, learned by a CCNN, along with a colorbar. Bottom: The top-down view of the curve depicted in top-right, along with a colorbar.

In this section we demonstrate the efficacy of our proposed formalism (Sec. 3) for generating examples under equality and inequality constraints through simulations. We particularly focus on learning which can be accomplished by two input and one output units. This permits visualization of the input-output space, which lies in $\mathbb{R}^3$. Note that our formalism can handle an arbitrary number of input and output units; this restriction is solely for ease of visualization. As a running example, we show how we can get a CCNN trained on the continuous-XOR classification task (Fig. 1) to generate examples under a wide range of equality (Sec. 3.1) and an arbitrary set of inequality (Sec. 3.2) constraints. The output unit has a symmetric sigmoidal activation function with range -0.5 and +0.5. After training, a CCNN with 6 hidden layers is obtained whose input-output mapping, $f(x_1, x_2; W^*)$, is shown in Fig. 1(top-right).

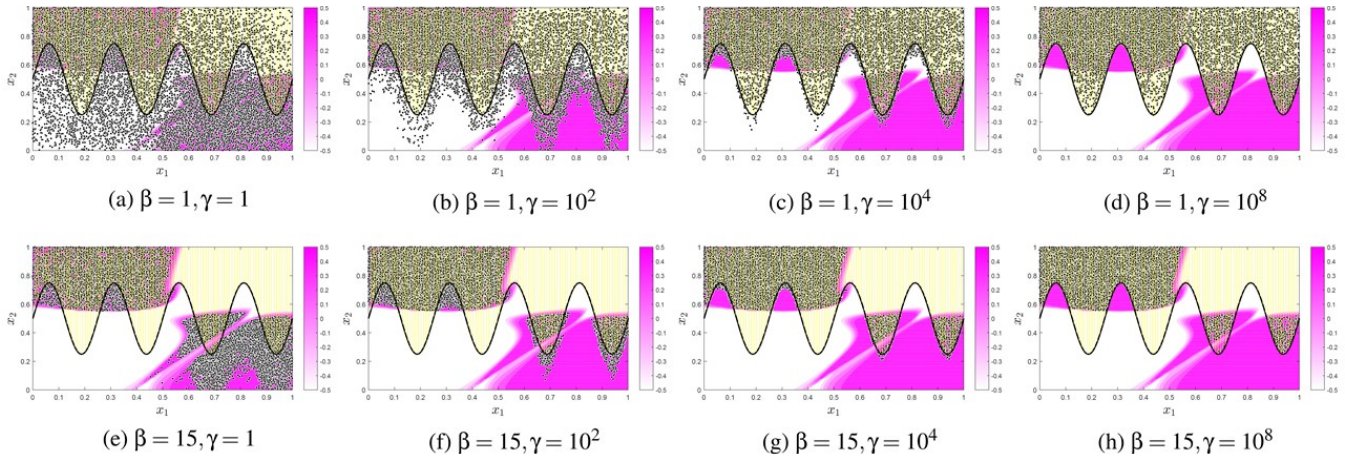
Following Nobandegani and Shultz (2017), for all the following simulations (Figs. 2-4), we used the Metropolis-adjusted Langevin (MAL) algorithm, with the time-step parameter $\tau = 5 \times 10^{-3}$; number of generated samples is set to $N = 2 \times 10^4$. Parameter τ featured in MAL (see Noban-



(a) $\beta = 1, \gamma = 1$ (b) $\beta = 1, \gamma = 10$ (c) $\beta = 1, \gamma = 10^2$ (d) $\beta = 1, \gamma = 10^3$

(e) $\beta = 15, \gamma = 1$ (f) $\beta = 15, \gamma = 10$ (g) $\beta = 15, \gamma = 10^3$ (h) $\beta = 15$

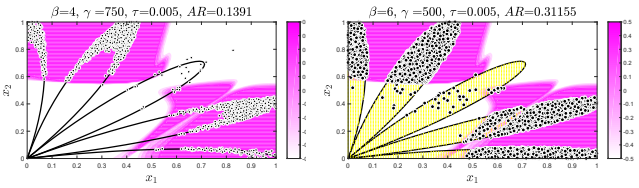
Figure 2: Generating examples (black dots) from the positive category of continuous-XOR (blue regions) under the equality constraint $x_2 = 0.25\sin(8\pi x_1) + 0.5$ (solid black curve). The quartic loss function $L(e) = e^4$ was adopted. Top-row (left to right): Increasing γ ensures that deviations from the sin curve are increasingly more penalized, all while lying outside the blue regions is only negligibly penalized ($\beta = 1$). Bottom-row (left to right): Lying outside the blue regions is more heavily penalized ($\beta = 15$). **(h)** The equality constraint is treated as a hard constraint, hence satisfied exactly, using the formalism introduced in (3).



(a) $\beta = 1, \gamma = 1$ (b) $\beta = 1, \gamma = 10^2$ (c) $\beta = 1, \gamma = 10^4$ (d) $\beta = 1, \gamma = 10^8$

(e) $\beta = 15, \gamma = 1$ (f) $\beta = 15, \gamma = 10^2$ (g) $\beta = 15, \gamma = 10^4$ (h) $\beta = 15, \gamma = 10^8$

Figure 3: Generating examples (black dots) from the positive category of continuous-XOR (magenta regions) under the inequality constraint $x_2 > 0.25\sin(8\pi x_1) + 0.5$ (yellow region). $L_\alpha(x)$ with $\alpha = 4$ was adopted as a loss function; see Eq. (6). Top-row (left to right): Increasing γ ensures that lying outside the yellow region is increasingly more penalized, all while lying outside the magenta regions is only negligibly penalized ($\beta = 1$). Bottom-row (left to right): Lying outside the magenta regions is more heavily penalized ($\beta = 15$).



(a) $\beta = 4, \gamma = 750$ (b) $\beta = 6, \gamma = 500$

Figure 4: Generating examples (black dots) from the positive category of continuous-XOR (magenta regions) under non-separable, complicated constraints. **(a)** Generating examples under the equality constraint: $(x_1^2 + x_2^2)^{0.5} = |\cos(8\tan^{-1}(\frac{x_2}{x_1}))|$ (solid black curve, on petals' boundaries); $L(e) = e^4$ was adopted as the loss functions. **(b)** Example generation under the inequality constraint: $(x_1^2 + x_2^2)^{0.5} < |\cos(8\tan^{-1}(\frac{x_2}{x_1}))|$ (yellow region, inside the petals); $L_\alpha(x)$ with $\alpha = 4$ was adopted as the loss function.

degani and Shultz, 2017, Algorithm 1) controls how distant proposed examples should be from each other, with large values of τ inducing larger distances; the effect of parameter τ is qualitatively investigated in Nobandegani and Shultz (2017).

5 Simulation of Denison et al. (2013)

As a demonstration, we simulated the experiment of Denison et al. (2013) with a neural network learning algorithm known as Sibling-Descendant Cascade-Correlation (Baluja & Fahlman, 1994). SDCC has been used to simulate many phenomena in cognitive development and learning (Shultz, 2012; Shultz & Fahlman, 2010). It automatically constructs the network in between the input and output layers by recruiting as many asigmoidal hidden units as needed to solve the problem being learned, thus capturing both development (unit recruitment) and learning (weight adjustment).

In the experiment, 4.5- and 6-month-olds were shown two boxes, one containing a ratio of 1 pink to 4 yellow balls, the other containing the opposite ratio (Denison et al., 2013). The experimenter drew from, say, the mostly yellow box, removing a sample of either 1 pink and 4 yellow balls (expected) or 4 pink and 1 yellow balls (unexpected) on alternating trials. Only the older infants looked longer at an unexpected, improbable sample than at an expected, probable sample.

Because depth of learning, manipulated by the score-threshold (ST) parameter in SDCC, has been shown to capture many developmental phenomena (Shultz, 2011, 2012), we set ST to either the default 0.4 to represent the apparent deeper learning of the older infants or the higher value of 0.6 to represent the apparent shallower learning of the younger infants. Technically, ST is the maximum distance from target training values (in this case 0 or 1) considered to be correct. In both conditions, SDCC was run in the learning cessation mode, which insured quitting when no further progress is being made in reducing network error (Shultz & Doty, 2014). Previous work has established that SDCC with automatic learning cessation accurately learns either discrete or (un-normalized) continuous probability distributions from patterns of positive and negative outcomes (Kharratzadeh & Shultz, 2016). Because SDCC is a deterministic learner, learning cessation ensures that learning stops when error reduction does. This affords a realistic learning, and compact representation, of probability distributions.

We trained 20 SDCC networks in each condition on 10 samples illustrating the $1/5$ or $4/5$ color ratios of the boxes from Denison et al. (2013). After training, the networks were tested on 5 samples representing either expected or unexpected outcomes. Fig. 5 shows that these probability distributions were accurately learned only with deeper learning characteristic of the older infants. SDCC recruited 3 hidden units with a ST of 0.4 and no hidden units with an ST of 0.6. As in previous work, error on test patterns represents surprise at seeing an unexpected event, in this case an improbable sample of 5 balls. Fig. 6 shows that this surprise was noted only by networks which had successfully learned the probability distribution, with ST of 0.4.

Fully consistent with Nobandegani and Shultz (2017), the joint probability distribution assigned to an input-output pair $(\mathbf{X},\mathbf{Y})$ can be modeled by (see the expression given in (1)):

$$p(\mathbf{X},\mathbf{Y}) = \frac{1}{Z_3}\exp(-\beta||\mathbf{Y}-f(\mathbf{X};W^*)||_2^2), \tag{7}$$

with Z_3 denoting a normalizing constant. Note that the term $||\mathbf{Y}-f(\mathbf{X};W^*)||_2^2$, when unpacked, evaluates to sum-of-squared error corresponding to the input-output pair $(\mathbf{X},\mathbf{Y})$. Hence, the probability of observing a sample is proportional to $\exp(-\beta\Sigma)$, with Σ denoting the sum-of-squared error corresponding to that sample.

Given the pattern of error observed for 4.5- and 6-month-olds reported in Fig. 6, it is clear that the differential looking-times of 4.5- and 6-month-olds for the expected, probable

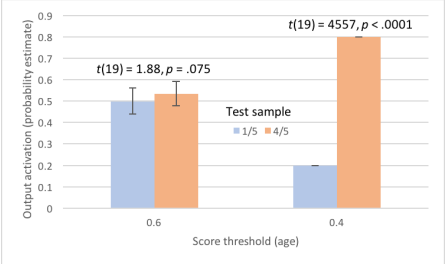


Figure 5: Mean output activations (with SDs) as a function of score-threshold and test sample. Probabilities are estimated accurately only with deeper learning. The output activation of shallower networks (ST = 0.6, for 4.5-month-olds) estimates the probability of both probable and improbable balls as near 0.5. However, the output activation of deeper networks (ST = 0.4, for 6-month-olds) accurately estimates the probability of a probable (= 0.8) and an improbable (= 0.2) sample of balls.

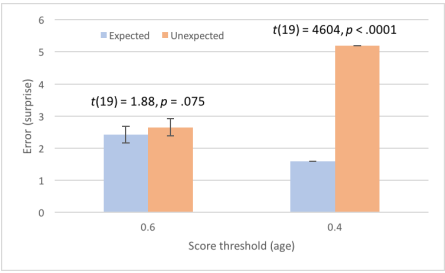


Figure 6: Mean error (with SDs) as a function of score-threshold and test-pattern expectedness. Substantial surprise to unexpected events only with deeper learning.

vs. unexpected, improbable sample can be accounted for by the joint distribution given in (7).

Interestingly, the Nobandegani and Shultz (2017) framework enables sampling from the SDCC-learned probability distribution, thus quantitatively simulating the samples children would generate; see Fig. 7. In that light, the Nobandegani and Shultz (2017) framework is consistent with a substantial body of work in developmental psychology providing evidence for the sampling abilities of children, including infants (e.g., Denison et al., 2010; Bonawitz et al., 2014a; Denison et al., 2013; Bonawitz et al., 2014b).

6 General Discussion

Recently, Nobandegani and Shultz (2017) presented an integrative framework which allows transforming any deterministic, discriminative neural network into a probabilistic, generative model. Most notably, the Nobandegani and Shultz (2017) framework: (i) bridges computational, algorithmic, and implementational levels of analysis, (ii) gives rise to self-organized, probabilistic generative models, and finally (iii) connects two dominant schools of thought in cognitive science: connectionism and Bayesian cognition. Importantly, by virtue of being able to handle any MCMC method, Nobandegani and Shultz's (2017) framework is consistent with substantial developmental findings suggesting that children mentally engage in sampling processes closely resem-

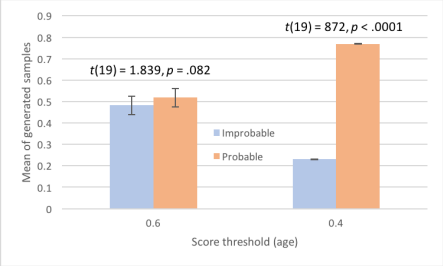


Figure 7: Means (and SDs) of MCMC-generated probable and improbable samples for Denison et al. (2013). The average proportions are plotted for the probable and improbable outcomes within a sequence of 2×10^5 MCMC-generated samples using 20 SDCC networks trained in each condition of Denison et al. (2013). Eq. (1) with $\beta = 2$ is used, together with the Metropolis-Hastings MCMC using a uniform proposal distribution.

bling MCMC (e.g., Denison et al., 2010; Bonawitz et al., 2014a; Denison et al., 2013; Bonawitz et al., 2014b).

It is worth noting that the Nobandegani and Shultz (2017) framework promotes the Metropolis-adjusted Langevin (MAL), a gradient-based MCMC, which can be implemented in a neurally-plausible manner (Savin & Deneve, 2014; Moreno-Bote et al., 2011), and which exploits the gradient of the target distribution to guide its explorations towards regions of high probability, thereby significantly reducing the undesirable random walk often observed at the beginning of an MCMC run (aka the burn-in period). Importantly, in the Nobandegani and Shultz (2017) framework, by exploiting the gradient signal (which can be efficiently computed by Backprop) learned by a deterministic neural net, MAL allows for computationally-efficient example generation, showcasing a computational complementarity of MAL and neural nets.

In this paper, we show that a novel extension of this framework nicely allows for generating exemplars under a wide range of equality (Sec. 3.1) and an arbitrary set of inequality (Sec. 3.2) constraints, either exactly (i.e., hard constraints) or approximately (i.e., soft constraints). Extensive simulations demonstrated the efficacy of this extension. We also showed how a joint probability distribution fully consistent with the Nobandegani and Shultz (2017) framework can account for a developmental shift in infants' probabilistic learning and reasoning abilities. This framework allows for sampling from the distribution (implicitly) learned by the model (Eqs. 2, 7), and hence is capable of quantitatively modeling infants' behavior which can be seen as sampling, e.g., infants' search for desired objects (Denison & Xu, 2010).

In recent years, there has been a surge of interest in Bayesian approaches to computational developmental psychology (e.g., Denison et al., 2010; Buchsbaum et al., 2011; Denison et al, 2013; Bonawitz et al., 2014a; Hu et al., 2015; Buchsbaum et al., 2012; Bonawitz et al., 2014b; Otsubo et al., 2017), primarily focusing on the computational (e.g., Buchsbaum et al., 2011; Hu et al. 2015; Buchsbaum et al., 2012; Otsubo et al., 2017) and, to a lesser extent, on the algorithmic level of analysis (e.g., Bonawitz et al., 2014), with relatively little attention to the implementational-level. Not long ago, however, the primary focus of computational developmental psychology was on connectionist approaches to development (Shultz, 2003). By now, evidence for both Bayesian and connectionist approaches to development abounds, calling for a unified take on phenomena linking connectionism and Bayesian cognition. Moving forward, integrated approaches—which bridge computational, algorithmic, and implementational levels of analysis—become increasingly more imperative to achieve. We believe the Nobandegani and Shultz (2017) framework, together with the extension formally outlined and experimentally tested in this paper would be a satisfying first step in this direction.

Acknowledgments: This works is supported by an operating grant to TRS from NSERC of Canada.

References

Baluja, S., & Fahlman, S. E. (1994). Reducing network depth in the cascade-correlation learning architecture. Technical Report # CMU-CS-94-209, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA..

Bonawitz, E., Denison, S., Gopnik, A., & Griffiths, T. L. (2014a). Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cognitive Psychology*, *74*, 35–65.

Bonawitz, E., Denison, S., Griffiths, T. L., & Gopnik, A. (2014b). Probabilistic models, learning algorithms, and response variability: sampling in cognitive development. *Trends in Cognitive Sciences*, *18*(10), 497–500.

Buchsbaum, D., Bridgers, S., Whalen, A., Seiver, E., Griffiths, T., & Gopnik, A. (2012). Do i know that you know what you know? modeling testimony in causal inference. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 34).

Buchsbaum, D., Gopnik, A., Griffiths, T. L., & Shafto, P. (2011). Childrens imitation of causal action sequences is influenced by statistical and pedagogical evidence. *Cognition*, *120*(3), 331–340.

Denison, S., Bonawitz, E., Gopnik, A., & Griffiths, T. L. (2013). Rational variability in childrens causal inferences: The sampling hypothesis. *Cognition*, *126*(2), 285–300.

Denison, S., Bonawitz, E. B., Gopnik, A., & Griffiths, T. L. (2010). Preschoolers sample from probability distributions. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 32).

Denison, S., Reed, C., & Xu, F. (2013). The emergence of probabilistic reasoning in very young infants: Evidence from 4.5-and 6-month-olds. *Developmental Psychology*, *49*(2), 243.

Denison, S., & Xu, F. (2010). Twelve-to 14-month-old infants can predict single-event probability with large set sizes. *Developmental Science*, *13*(5), 798–803.

Fahlman, S. E., & Lebiere, C. (1989). The cascade-correlation learning architecture. In *Advances in Neural Information Processing Systems*, pp. 524-532.

Hu, J. C., Whalen, A., Buchsbaum, D., Griffiths, T. L., & Xu, F. (2015). Can children balance the size of a majority with the quality of their information? In *CogSci Conference.*

Jern, A., & Kemp, C. (2013). A probabilistic account of exemplar and category generation. *Cognitive Psychology*, *66*(1), 85–125.

Kharratzadeh, M., & Shultz, T. (2016). Neural implementation of probabilistic models of cognition. *Cognitive Systems Research*, *40*, 99–113.

Marr, D. (1982). *Vision: A Computational Approach.*

Moreno-Bote, R., Knill, D. C., & Pouget, A. (2011). Bayesian sampling in visual perception. *Proceedings of the National Academy of Sciences*, *108*(30), 12491–12496.

Nobandegani, A. S., & Shultz, T. R. (2017). Converting cascade-correlation neural nets into probabilistic generative models. In *Proceedings of the* 39^{th} *Annual Meeting of the Cognitive Science Society. Austin, TX: Cognitive Science Society.*

Otsubo, K., Whalen, A., & Buchsbaum, D. (2017). Investigating sensitivity to shared information and personal experience in childrens use of majority information. In *Proceedings of the* 39^{th} *Annual Meeting of the Cognitive Science Society. Austin, TX: Cognitive Science Society.*

Sanborn, A. N., & Chater, N. (2016). Bayesian brains without probabilities. *Trends in Cognitive Sciences*, *20*(12), 883–893.

Savin, C., & Deneve, S. (2014). Spatio-temporal representations of uncertainty in spiking neural networks. In *Advances in Neural Information Processing Systems.*

Shultz, T. R. (1998). A computational analysis of conservation. *Developmental Science*, *1*(1), 103–126.

Shultz, T. R. (2003). Computational Developmental Psychology. MIT Press.

Shultz, T. R. (2006). Constructive learning in the modeling of psychological development. *Processes of Change in Brain and Cognitive Development: Attention and Performance*, *21*, 61–86.

Shultz, T. R. (2011). Computational modeling of infant concept learning: The developmental shift from features to correlations. *Infant perception and cognition: Recent advances, emerging theories, and future directions*, 125–152.

Shultz, T. R. (2012). A constructive neural-network approach to modeling psychological development. *Cognitive Development*, *27*(4), 383–400.

Shultz, T. R., & Bale, A. C. (2006). Neural networks discover a near-identity relation to distinguish simple syntactic forms. *Minds and Machines*, *16*(2), 107–139.

Shultz, T. R., & Doty, E. (2014). Knowing when to quit on unlearnable problems: another step towards autonomous learning. In *Computational Models of Cognitive Processes: Proceedings of the* 13^{th} *Neural Computation and Psychology Workshop* (pp. 211–221).

Shultz, T. R., & Fahlman, S. E. (2010). Cascade-correlation. In *Encyclopedia of Machine Learning* (pp. 139–147). Springer.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning*, *16*(1-2), 57–86.

Shultz, T. R., & Rivest, F. (2001). Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science*, *13*(1), 43–72.

Shultz, T. R., & Takane, Y. (2007). Rule following and rule use in the balance-scale task. *Cognition*, *103*(3), 460–472.

Shultz, T. R., & Vogel, A. (2004). A connectionist model of the development of transitivity. In *Proceedings of the* 26^{th} *Annual Conference of the Cognitive Science Society* (pp. 1243–1248).

Westermann, G., Sirois, S., Shultz, T. R., & Mareschal, D. (2006). Modeling developmental cognitive neuroscience. *Trends in Cognitive Sciences*, *10*(5), 227–232.