One shot learning of simple visual concepts

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology

Abstract

People can learn visual concepts from just one example, but it remains a mystery how this is accomplished. Many authors have proposed that transferred knowledge from more familiar concepts is a route to one shot learning, but what is the form of this abstract knowledge? One hypothesis is that the sharing of parts is core to one shot learning, and we evaluate this idea in the domain of handwritten characters, using a massive new dataset. These simple visual concepts have a rich internal part structure, yet they are particularly tractable for computational models. We introduce a generative model of how characters are composed from strokes, where knowledge from previous characters helps to infer the latent strokes in novel characters. The stroke model outperforms a competing state-of-the-art character model on a challenging one shot learning task, and it provides a good fit to human perceptual data.

Keywords: category learning; transfer learning; Bayesian modeling; neural networks

A hallmark of human cognition is learning from just a few examples. For instance, a person only needs to see one Segway to acquire the concept and be able to discriminate future Segways from other vehicles like scooters and unicycles (Fig. 1 left). Similarly, children can acquire a new word from one encounter (Carey & Bartlett, 1978). How is one shot learning possible?

New concepts are almost never learned in a vacuum. Past experience with other concepts in a domain can support the rapid learning of novel concepts, by showing the learner what matters for generalization. Many authors have suggested this as a route to one shot learning: transfer of abstract knowledge from old to new concepts, often called *transfer learning*, *representation learning*, or *learning to learn*. But what is the nature of the learned abstract knowledge that lets humans acquire new object concepts so quickly?

The most straightforward proposals invoke attentional learning (Smith, Jones, Landau, Gershkoff-Stowe, & Samuelson, 2002) or overhypotheses (Kemp, Perfors, & Tenenbaum, 2007; Dewar & Xu, in press), like the shape bias in word learning. Prior experience with concepts that are clearly organized along one dimension (e.g., shape, as opposed to color or material) draws a learner's attention to that same dimension (Smith et al., 2002) – or increases the prior probability of new concepts concentrating on that same dimension (Kemp et al., 2007). But this approach is limited since it requires that the relevant dimensions of similarity be defined in advance.

For many real-world concepts, the relevant dimensions of similarity may be constructed in the course of learning to learn. For instance, when we first see a Segway, we may parse it into a structure of familiar parts arranged in a novel configuration: it has two *wheels*, connected by a *platform*, supporting a *motor* and a central *post* at the top of which are two *handlebars*. These parts and their relations comprise a

Figure 1: Test yourself on one shot learning. From the example boxed in red, can you find the others in the array? On the left is a Segway and on the right is the first character of the Bengali alphabet.

Answer for the Bengali character: Row 2, Column 3; Row 4, Column 2.

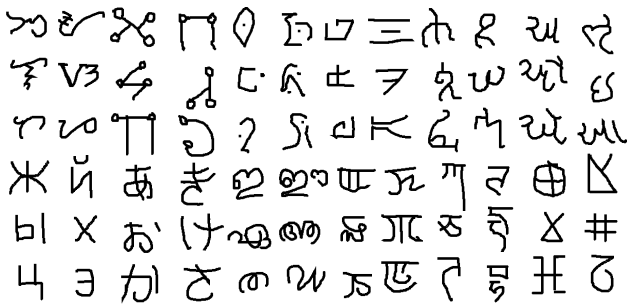

Figure 2: Examples from a new 1600 character database.

useful representational basis for many different vehicle and artifact concepts – a representation that is likely learned in the course of learning the concepts that they support. Several papers from the recent machine learning and computer vision literature argue for such an approach: joint learning of many concepts and a high-level part vocabulary that underlies those concepts (e.g., Torralba, Murphy, & Freeman, 2007; Fei-Fei, Fergus, & Perona, 2006). Another recently popular machine learning approach is based on *deep learning* (Salakhutdinov & Hinton, 2009): unsupervised learning of hierarchies of distributed feature representations in neural-network-style probabilistic generative models. These models do not specify explicit parts and structural relations, but they can still construct meaningful representations of what makes two objects deeply similar that go substantially beyond low-level image features.

These approaches from machine learning may be compelling ways to understand how humans learn so quickly, but there is little experimental evidence that directly supports them. Models that construct parts or features from sensory data (pixels) while learning object concepts have been tested in elegant behavioral experiments with very simple stimuli and a very small number of concepts (Austerweil & Griffiths, 2009; Schyns, Goldstone, & Thibaut, 1998). But there have been few systematic comparisons of multiple state-of-the-art computational approaches to representation learning with hu-

man learners on a large scale, using a large number of interesting natural concepts. This is our goal here.

We work in the domain of handwritten characters, an ideal setting for studying one shot learning at the interface of human and machine learning. Handwritten characters contain a rich internal part structure of pen strokes, providing good a priori reason to explore a parts-based approach to representation learning. Supporting this notion, psychological studies have shown that knowledge about how characters are produced from strokes influences basic perception, including classification (Freyd, 1983) and apparent motion (Tse & Cavanagh, 2000). While characters contain complex internal structure (Fig. 2), they are simple enough for us to hope that tractable computational models can represent all the structure people see in them – unlike natural images. Handwritten digit recognition (0 to 9) has received major attention in machine learning, with genuinely successful algorithms. Classifiers based on deep learning can obtain over 99 percent accuracy on the standard MNIST dataset (e.g., LeCun, Bottou, Bengio, & Haffner, 1998; Salakhutdinov & Hinton, 2009). Yet these state-of-the-art models are still probably far from human-level competence; there is much room to improve on them. The MNIST dataset provides thousands of training examples for each class. In stark contrast, humans only need one example to learn a new character (Fig. 1 right).

Can this gap be closed by exploring different forms of prior knowledge? Earlier work on one shot digit learning investigated transferable knowledge of image deformations, such as scale and rotation (Miller, Matsakis, & Viola, 2000). These factors are important, but we suggest there is much more to the knowledge that supports one shot learning. People have a rich understanding of how characters are formed from the strokes of a pen, guided by the human motor system.

There are challenges with conducting a large scale study of character learning. People already know the digits and the Latin alphabet, so experiments must be conducted on new characters. Also, people receive massive exposure to domestic and foreign characters over a lifetime, including extensive first hand drawing experience. To simulate some of this experience for machines, we collected a massive new dataset of over 1600 characters from around the world. By having participants draw characters online, it was possible to record both the images, the strokes, and the time course of drawing (Fig. 3). Using the dataset, we can investigate the dual problems of understanding human concept learning and building machines that learn as rapidly as people can. We propose a new model of character learning based on inducing probabilistic part-based representations, similar to the computer vision approaches of Torralba, Fei-Fei, Perona and colleagues. Given an example image of a new character type, the model infers a sequence of latent strokes that best explains the pixels in the image, drawing on a large stroke vocabulary abstracted from many previous characters. This stroke-based representation guides generalization to new examples of the concept. We test the model against both human perceptual discrimination data and human accuracy in a challenging one shot classification task, while comparing it with a leading alternative approach from machine learning, the Deep Boltzmann Machine (DBM; Salakhutdinov & Hinton, 2009). The DBM is an interesting comparison because it is also a generative probabilistic model, it achieves state-of-the-art performance on the permutation invariant version of the MNIST task, and it has no special knowledge of strokes or even image geometry. We find that the stroke model outperforms the DBM by a large margin on one shot learning accuracy, and both models provide a good fit to human perceptual discrimination.

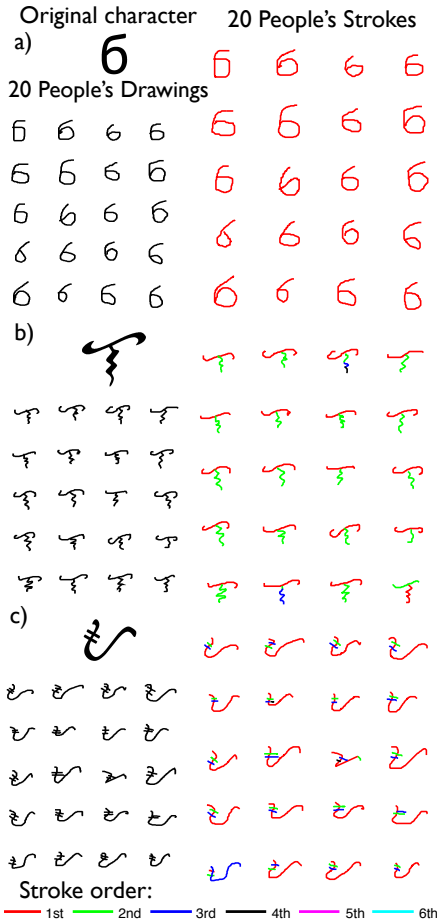

Figure 3: Illustration of the drawing data. Each panel shows the original character, 20 people's image drawings, and 20 people's strokes color coded for order.

New dataset of 1600 characters

We collected a new dataset suitable for large scale concept learning from few examples. The dataset can be viewed as the "transpose" of MNIST; rather than having 10 character (digit) classes with thousands of examples each like MNIST, the new dataset has over 1600 characters with only 20 examples each. These characters are from 50 alphabets from around the world, including Bengali, Cyrillic, Avorentas, Sanskrit, Tagalog, and even synthetic alphabets used for sci-fi novels. Prints of the original characters were downloaded from www.omniglot.com and several original images are shown in Fig. 3 (top left in each panel). Perception and modeling

should not be tested on these original typed versions, since they contain differences in style and line width across alphabets. Instead each alphabet was posted on Amazon Mechanical Turk using the printed forms as reference, and all characters were drawn by 20 different non-experts with computer mice (Fig. 3, bottom left). In addition to capturing the image, the interface captures the drawer's parse into strokes, shown in Fig. 3 (right) where color denotes stroke order.

Drawing methods are remarkably consistent across participants. For instance, Fig. 3a shows a Cyrillic character where all 20 people used one stroke. While not visible from the static trace, each drawer started the trajectory from the top right. Fig. 3b shows a Tagalog character where 19 drawers started with top stroke (red), followed by a second dangling stroke (green). But there are also slight variations in stroke order and number. Videos of the drawing process for these characters and others can be downloaded at http://web.mit.edu/brenden/www/charactervideos.html.

Generative stroke model of characters

The consistent drawing pattern suggests a principled inference from static character to stroke representation (see Babcock & Freyd, 1988). Here we introduce a stroke model that captures this basic principle. When shown just one new example of a character, the model tries to infer a set of latent strokes and their configuration that explains the pixels in an image. This high-level representation is then used to classify new images with unknown identity. Fig. 4 describes the generative process. Character types (A, B, etc.) are generated from general knowledge which includes knowledge of strokes. These types are abstract descriptions defined by strokes: their number, identity, and general configuration. Character tokens (images) are generated from the types by perturbing stroke positions and inking the pixels.

Generating a character type A character type is defined by a set of strokes S, their positions W, and their mixing strengths π. The number of strokes m is picked from a uniform distribution (1 to 10 for simplicity). The first stroke identity is drawn from the uniform distribution $P(S_1) = 1/K$ where $K = 1000$ is the size of the stroke set. Each stroke also has a starting position for its trajectory, denoted W_i where $W_i = [w_{x_i}, w_{y_i}]$ which has discrete x and y coordinates. The first stroke's position is uniform across the R^2 discrete pixel locations in the image (the image size is $R \times R$). Subsequent strokes $P(S_{i+1}|S_i)$ and positions $P(W_{i+1}|W_i)$ are drawn from a transition model, which is uniform and independent of the past. The transition models could be extended, both for the strokes and positions, to include a more accurate sequential process. Finally, we draw the mixing weights $\pi \sim \text{Dirichlet}(1, 1, ..., 1)$ which is a vector of length m.

Generating a character token A character type then generates a character token $I^{(j)}$, which is a pixel image. While W specifies a character type position template, the token specific positions $Z^{(j)}$ can vary in both relative positions

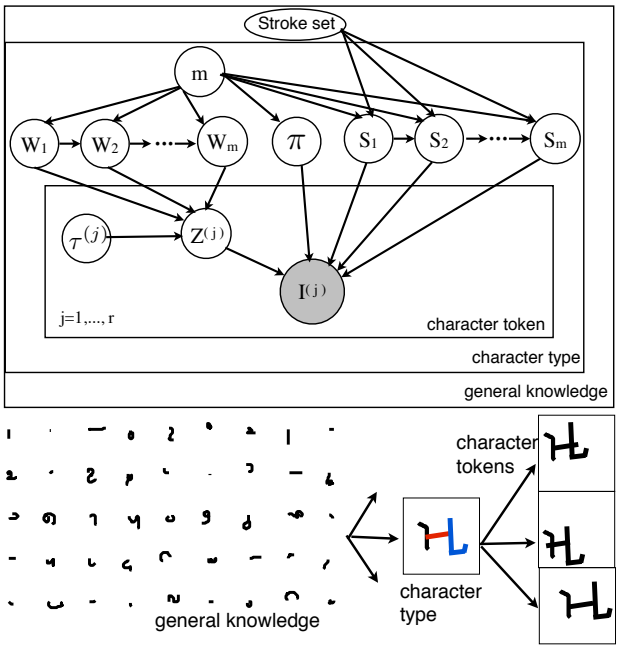

Figure 4: Illustration of the generative process as described in text. All variables inside the character type plate are implicitly indexed by character type.

of the strokes and through a global translation controlled by $\tau^{(j)}$. As with W, the image specific positions $Z^{(j)} = \{Z_1^{(j)}, ..., Z_m^{(j)}\} = \{z_{x_1}^{(j)}, z_{y_1}^{(j)}, ..., z_{x_m}^{(j)}, z_{y_m}^{(j)}\}$ specify discrete x and y coordinates in the image. The distribution is

$$P(Z^{(j)}|W, \tau^{(j)}) \propto \prod_{i=1}^{m} \exp(-\frac{1}{2\sigma_z^2}||(Z_i^{(j)} - W_i - \tau^{(j)}||_2^2),$$

which is like a spherical Gaussian but with support on a discrete set. The translation $\tau^{(j)}$ is distributed as $P(\tau^{(j)}) \propto \exp(-\frac{1}{2\sigma_t^2}||\tau^{(j)}||_2^2)$ with support on $\{-R, ..., R\} \times \{-R, ..., R\}$. Given the positions, the image can be generated by G[1] hypothetical draws of pixels to "ink" from a distribution over pixels. The ink model is based on Revow, Williams, and Hinton (1996) although we extend it to the multi-stroke case. The probability that *none* of the draws landed in a pixel slot g (g is white) is

$$P(I_g^{(j)} = 0|S, Z^{(j)}, \pi) = (1 - Q(I_g^{(j)}|S, Z^{(j)}, \pi))^G,$$

and the probability of a pixel being inked is the complement $P(I_g^{(j)} = 1|S, Z^{(j)}, \pi) = 1 - P(I_g^{(j)} = 0|S, Z^{(j)}, \pi)$. Intuitively, the function Q distributes ink across the strokes with Gaussian spray paint. This is captured by lining each stroke with little Gaussian beads that generate ink. Q is defined by a nested mixture:[2] an inked pixel is a mixture of noise (parameter β) and another mixture over the m strokes, and each stroke is yet another mixture (V) of the Gaussian beads

[1] We use the actual number of inked pixels for G. But as Revow et al. point out, other values would increase the probability of the data since hypothetical draws will overlap. But this inaccuracy will hurt both correct and incorrect candidate models during classification.

[2] Note that if Q can have values greater than 1, this is no longer a valid distribution. But it can be shown that if $\sigma_b > 0.4$, then $Q < 1$.

$$Q(I_g^{(j)}|S,Z^{(j)},\pi) = \frac{\beta}{R^2} + (1-\beta)\sum_{i=1}^{m}\pi_i V(I_g^{(j)}|S_i,Z_i^{(j)})$$

$$V(I_g^{(j)}|S_i,Z_i^{(j)}) = \frac{1}{B}\sum_{b=1}^{B} N(I_g^{(j)}|X_b + Z_i^{(j)}, \sigma_b^2 I),$$

where I is the identity matrix, N is a Gaussian, and $X_b \in \mathbb{R}^2$ are the bead coordinates for the stroke S_i. Evaluating the ink model is expensive, but each stroke's V can be computed offline, cached, and then translated by $Z_i^{(j)}$ as needed. We used $B = 28$, $\sigma_b = 1.5$, $\beta = 0.01$, $\sigma_z = 2$, and $\sigma_t = 10$.

Learning a library of strokes General knowledge of strokes was learned from the drawing data. The entire dataset was split randomly into a 25 alphabet "background set" and a 25 alphabet "experiment set." The stroke library was learned from the background set, and the models and people were tested on the experiment set. About 40,000 strokes were aligned and clustered (using k-means) to form $K = 1000$ centroids that comprise the model's library (Fig. 4). Stroke trajectories vary widely in length so they were reduced to a common dimensionality by fitting a cubic B-spline with 10 control points and clustering was done in this new space (Revow et al., 1996; Branson, 2004).[3] Strokes are direction specific, meaning a left to right line and a right to left line are different.

Inference for one shot learning For one shot learning, the model is given a single example image $I^{(e)}$ and a candidate image $I^{(t)}$. Exact computation of $P(I^{(t)}|I^{(e)})$ is intractable and even a maximum a posteriori (MAP) estimate involves fitting every pair of images $I^{(e)}$ and $I^{(t)}$, which is very expensive. Instead, the computation is approximated as follows:

$$\begin{aligned} & P(I^{(t)}|I^{(e)}) \\ = & \sum_{S,W,\pi} P(I^{(t)}|S,W,\pi) \sum_{\tau^{(e)},Z^{(e)}} P(S,W,\pi,Z^{(e)},\tau^{(e)}|I^{(e)}) \\ \approx & \; P(I^{(t)}|S^*,W^*,\pi^*), \text{where} \end{aligned}$$

$$\{S^*,W^*,\pi^*\} = \underset{S,W,\pi,Z^{(e)},\tau^{(e)}}{\text{argmax}} \; P(S,W,\pi,Z^{(e)},\tau^{(e)}|I^{(e)}).$$

To compute the maximization, we run Markov Chain Monte Carlo (MCMC) and the Metropolis-Hastings algorithm, taking the most probable sample after 50,000 proposals. Intuitively, this picks the best strokes it can find to explain just the training image. Proposals include a replacement of a stroke S_i and position W_i with a similar stroke and position, moves to change π, moves to permute stroke indices, and reversible jump moves to add or remove the last stroke while also perturbing all the other variables. There is just one image so far, so we fix $Z^{(e)} = W$ and $\tau^{(e)} = 0$. The sampler is initialized after exploring a set of bottom-up parses, using a stochastic tracing algorithm inspired by Edelman, Flash, and Ullman (1990). Each bottom-up parse is mapped to the closest library strokes, scored, and the best is picked for initialization. We then approximate

$$\begin{aligned} & P(I^{(t)}|S^*,W^*,\pi^*) \\ = & \sum_{Z^{(t)},\tau^{(t)}} P(I^{(t)},Z^{(t)},\tau^{(t)}|S^*,W^*,\pi^*) \\ \approx & \; P(I^{(t)},Z^{(t)*},\tau^{(t)*}|S^*,W^*,\pi^*), \text{where} \end{aligned}$$

$$\{Z^{(t)*},\tau^{(t)*}\} = \underset{Z^{(t)},\tau^{(t)}}{\text{argmax}} \; P(Z^{(t)},\tau^{(t)}|I^{(t)},S^*,W^*,\pi^*).$$

Again, we use Metropolis-Hastings and take the most probable sample after 2000 proposals. Moves include proposing a new $Z_i^{(t)}$ or jointly proposing changes in $Z^{(t)}$ and $\tau^{(t)}$.

20-way classification from one example

We tested three models on one shot learning: the stroke model, the Deep Boltzmann Machine (DBM, Salakhutdinov & Hinton, 2009), and Nearest Neighbor (NN) in pixel space. Performance was evaluated on 20-way classification, where each training class gets only one example. For a given run, 20 characters were picked at random from different alphabets in the experiment set. The models have never seen any of these alphabets or characters before. Accuracy was then tested on novel images drawn from this set of 20 characters.

All models received 28×28 images (binary for the stroke model and NN, grayscale for the DBM). The stroke model fits a latent stroke representation to each training image $I^{(e)}$, and a test image $I^{(t)}$ is classified by picking the largest $P(I^{(t)}|I^{(e)})$ across the 20 possible training characters. The DBM was pretrained on the 25 background alphabets using a combination of MCMC and variational approximation (see Salakhutdinov & Hinton, 2009). The architecture was two hidden layers with 1000 units each. DBM classification is performed by nearest neighbor in the hidden representation space, combining vectors from both hidden layers and using cosine similarity. NN classification uses Euclidean distance but cosine performs similarly.

The stroke model achieves 54.9% correct, compared to 39.6% for the DBM and 15.7% for nearest neighbor in pixels (Fig. 5 left). This was averaged over many random runs (27) of the training characters and four test examples per class. Figure 6 illustrates model fits to the training images in one run. The stroke model is reasonable but imperfect parser. To disentangle the imperfect parsing from the general approach, we replaced the inferred strokes at training time (like Fig. 6) with the real strokes produced by the drawer of the image. Classification was then conducted after inferring the test image parameters ($Z^{(t)}$ and $\tau^{(t)}$) like in the standard stroke model. The real strokes achieve 63.7% correct, which is likely an upper bound for the current stroke model implementation. But there are many promising avenues for overcoming this bound, which are outlined in the discussion. How would people perform? We found that people were 97.6% correct on a Same/Different task (baseline is 75%); see footnote for details.[4] Although this is a different task, it confirms there is

[3] B-splines are a compact representation of a smooth curve, providing a function $B(s)$ that maps a dimension s (similar to time for strokes) to an x and y position. It smoothly interpolates between the 10 control points (which are x and y coordinates) such that the curve starts near the first control point and ends near the last. The least-squares fit can be computed in closed form (Branson, 2004).

[4] If people were run directly on 20-way classification, they could

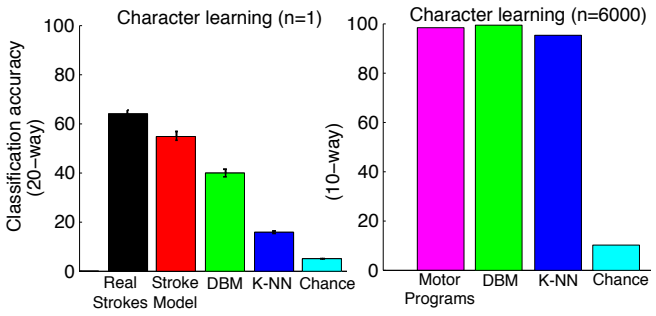

Figure 5: Classification accuracy from one example (left, our results) and on the MNIST digits (right, published results not from this work). DBM = Deep Boltzmann Machine; K-NN = K-Nearest Neighbors; Motor programs model is from Hinton and Nair (2006). Error bars are standard error.

a substantial gap between human and machine competence.

To create an interesting juxtaposition with one shot learning, some previously published results on MNIST, not from this work, are displayed (Fig. 5 right). Even simple methods like K-nearest neighbors perform extremely well (95% correct LeCun et al., 1998) due to the huge training set ($n \approx 6,000$ per character). As a possible analog to the stroke model, Hinton and Nair (2006) learned motor programs for the MNIST digits where characters were represented by just one, more flexible stroke (unless a second stroke was added by hand). As evident from the figure, the one example setting provides more room for both model comparison and progress.

Fit to human perceptual discrimination

The models were also compared to human perceptual judgments. A set of six alphabets and four characters each was selected for high confusability within alphabets. Fig. 7 shows the original images, but participants saw the handwritten copies. Participants were asked to make 200 same vs. different judgments, where the proportion of same trials was $1/4$. The task was speeded and the first of two images was flashed on the screen for just 50 milliseconds before it was covered by a mask. The second image then appeared and remained visible until a response was made. There was an option for "I wasn't looking at the computer screen" and these responses were discarded. Sixty people were run on Mechanical Turk, and 13 subjects were removed for having a d-prime less than 0.5.[5] Of the remaining, accuracy was 80 percent.

Trials were pooled across participants to create a character by character similarity matrix. Cells show the percentage of responses that were "same," and the matrix was made symmetrical by averaging with its transpose (Fig. 8). There is a clear block structure showing confusion within alphabets, except Inuktitut that contains shapes already familiar to people (e.g. triangle). The stroke model, DBM, and image distance

learn from the test examples. Instead, people made same vs. different judgements using the whole experiment set of characters. "Same" trials were two images, side by side, of the same character from two drawers, and "Different" trials were two different characters. Each of 20 participants saw 200 trials using a web interface on Mechanical Turk, and the ratio of same to different trials was $1/4$.

[5]The number of false alarms was divided by 3 to correct for having 3 times as many different trials.

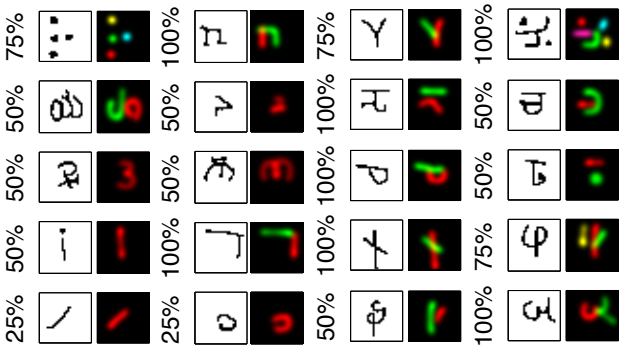

Figure 6: Example run of 20-way classification, showing the training images/classes (white background) and the stroke model's fits (black background). Accuracy rates are indicated on the 4 test examples (not shown) per class.

were compared to the perceptual data. Perceptual discrimination was modeled using the same procedure for all models, where each model saw many replications (76) of mock 24-way classification, as in the previous section. For each test image, the goodness of fit to each of the 24 training classes was assigned a rank r from best (1) to worst (24). For each pairing of stimuli, the similarity $s = 1/r$ was added to the corresponding cells, averaging across replications. Both the stroke model (r=0.80) and the DBM (r=.77) show clear alphabet block structure and correlate well with the human judgments, while image distance does not fit well (r=0.30).

Discussion

This paper introduced a generative model of how characters are composed from parts. Given a new character type, the model attempts to infer latent strokes that explain the pixels in the image. This approach performs well on one shot classification, beating Deep Boltzmann Machines (DBM) by a wide margin. Both of these models provide good fits to human perceptual judgements on a small set of characters.

The stroke model is still far from human competence, although there are many avenues for extension and improvement. There is a clear need for both a richer basis of compositional elements and the ability to expand this basis to include new strokes when needed. The strokes in the current model are rigid, allowing for translations but no scaling, rotations, or deformation within individual strokes (see Revow et al., 1996). As suggested in the classification results, there is not much room for improvement within the rigid stroke regime, given the upper bound obtained by using the real but still rigid strokes. Additionally, novel characters often contain novel strokes, and a model could benefit from moving beyond a finite library with a non-parametric Bayesian approach. While our general framework allows for these improvements, the present choices were made for computational efficiency, and it will be critical to overcome these limitations in future work.

While more flexible models introduce new problems for inference, bottom-up parsers are a promising means for tackling these challenges. In preliminary simulations using an image tracer modified from Edelman et al. (1990), we found that these methods can work well for classification, even without a

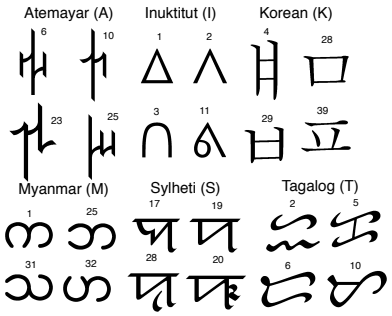

Figure 7: Human perceptual discrimination was measured on pairs of these characters, which are from 6 alphabets. The original printed images are shown, but participants saw handwritten drawings. Character index within an alphabet is denoted.

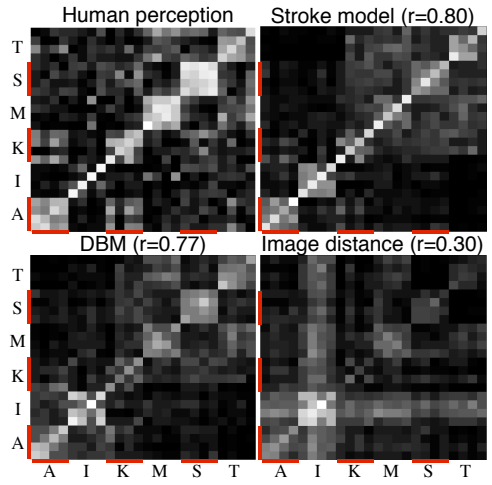

Figure 8: Similarity matrices (lighter is more similar) for the 24 characters in Fig. 7. Alphabets are blocked and denoted by their starting letter (A, I, etc.). Character index (Fig. 7) within alphabet blocks is in increasing order, left to right.

notion of shared parts. If the stroke model's parse is replaced with a bottom-up parse and inference is then performed on the remaining parameters in the generative model, classification accuracy is often higher, likely because complex strokes are fit more precisely. But performance is still limited by the upper bound for rigid strokes, and without a notion of shared parts, it is unclear how to incorporate multiple training examples. Despite the limitations of a pure bottom-up approach, these methods could play an important role by making data driven proposals, either by proposing new strokes or by fine-tuning existing strokes within a richer generative framework.

What other domains are like our simple visual concepts? Like characters, artifacts are complex concepts composed of parts. Bicycles, cars, and scooters share parts like wheels, handlebars, and motors, and new artifacts like the Segway can be generated by combining these parts in novel ways. Our simple visual concepts also share deep similarities with other symbols used for communication, such as spoken words, gestures, and sign language. When defining the concept as the raw symbol rather than its meaning, people readily both generate and perceive these concepts, and there is a similar emphasis on building objects from primitives. For instance, a spoken word is a sequence of phonemes just as a character is a sequence of strokes. Learning in these domains could involve similar computational mechanisms. For spoken words, Feldman, Griffiths, and Morgan (2009) proposed that concepts are learned in conjunction with their components parts, and this is also a guiding principle behind the stroke model. Most speculatively, people learn rich visual concepts like animals and faces from few examples, although their forms are governed by very complicated generative processes. Could similar computational principles explain rapid learning in even these domains?

References

Austerweil, J., & Griffiths, T. L. (2009). Analyzing human feature learning as nonparametric bayesian inference. In *Advances in Neural Information Processing Systems.*

Babcock, M. K., & Freyd, J. (1988). Perception of dynamic information in static handwritten forms. *American Journal of Psychology*, *101*(1), 111-130.

Branson, K. (2004). *A practical review of uniform b-splines.*

Carey, S., & Bartlett, E. (1978). Acquiring a single new word. *Papers and Reports on Child Language Development*, *15*, 17-29.

Dewar, K., & Xu, F. (in press). Induction, overhypothesis, and the origin of abstract knowledge: evidence from 9-month-old infants. *Psychological Science*.

Edelman, S., Flash, T., & Ullman, S. (1990). Reading cursive handwriting by alignment of letter prototypes. *International Journal of Computer Vision*, *5*(3), 303-331.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 594-611.

Feldman, N. H., Griffiths, T. L., & Morgan, J. L. (2009). Learning phonetic categories by learning a lexicon. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society.*

Freyd, J. (1983). Representing the dynamics of a static form. *Memory & Cognition*, *11*(4), 342-346.

Hinton, G. E., & Nair, V. (2006). Inferring motor programs from images of handwritten digits. In *Advances in Neural Information Processing Systems* (Vol. 18). Cambridge, MA: MIT Press.

Kemp, C., Perfors, A., & Tenenbaum, J. B. (2007). Learning overhypotheses with hierarchical Bayesian models. *Developmental Science*, *10*(3), 307-321.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2323.

Miller, E. G., Matsakis, N. E., & Viola, P. A. (2000). Learning from one example through shared densities on transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Revow, M., Williams, C. K. I., & Hinton, G. E. (1996). Using generative models for handwritten digit recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(6), 592-606.

Salakhutdinov, R., & Hinton, G. E. (2009). Deep boltzmann machines. In *12th Internationcal Conference on Artificial Intelligence and Statistics.*

Schyns, P. G., Goldstone, R. L., & Thibaut, J.-P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences*, *21*, 1-54.

Smith, L., Jones, S. S., Landau, B., Gershkoff-Stowe, L., & Samuelson, L. (2002). Object name learning provides on-the-job training for attention. *Psychological Science*, *13*, 13-19.

Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(5), 854-869.

Tse, P. U., & Cavanagh, P. (2000). Chinese and americans see opposite apparent motions in a chinese character. *Cognition*, *74*, B27-B32.