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 encounter, 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, but there have been few attempts to test this hypothesis on a large scale. This paper works in the domain of handwritten characters, which contain a rich component structure of strokes. We introduce a generative model of how characters are composed from strokes, and how knowledge from previous characters helps to infer the latent strokes in novel characters. After comparing several models and humans on one shot character learning, we find that our stroke model outperforms a state-of-the-art character model by a large margin, and it provides a closer 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. Experience with other, more familiar concepts in a domain can support more rapid learning of novel concepts by showing the learner what aspects of objects matter 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, the learned representational capacities, that lets humans learn 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. Given several dimensions along which objects may be similar, 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, in a hierarchical Bayesian model of overhypothesis learning (Kemp et al., 2007). But this approach is limited since it requires that relevant dimensions of similarity be defined in advance.

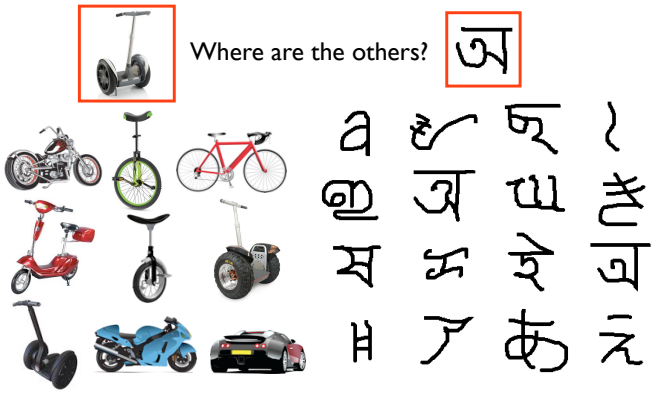

Figure 1: Test yourself on one shot learning. From the example boxed in red, can you find the others in the grid below? On the left is a Segway and on the right is the first character of the Bengali alphabet.

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

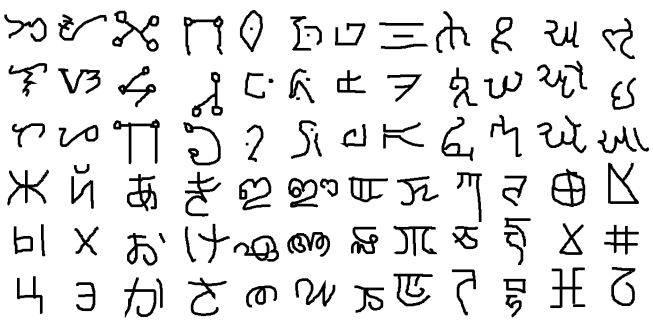

Figure 2: Examples from a new 1600 character database.

In contrast, for many interesting, 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*. Such parts and their relations comprise a useful representational basis for many different vehicle and artifact concepts more generally – a representation that is likely learned in the course of learning the many different object 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* (LeCun, Bottou, Bengio, & Haffner, 1998; Salakhutdinov & Hinton, 2009): unsupervised learning

of hierarchies of distributed feature representations in multilayered 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 in a domain deeply similar that go substantially beyond low-level image features or pixel-wise similarity.

These approaches from the recent machine learning literature may be compelling ways to understand how humans learn to learn new concepts from few examples, 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 human 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. This is an ideal setting for studying one shot learning and knowledge transfer 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. Furthermore, studies have shown that knowledge about how characters decompose into strokes influences basic perception, including classification (Freyd, 1983) and apparent motion (Tse & Cavanagh, 2000). While characters contain complex internal structure (Fig. 2), they are still simple enough for us to hope that tractable computational models can fully represent all the structure people see in them – unlike for natural visual 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 et al., 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 a different form of prior knowledge? Earlier work on one shot digit learning investigated how to transfer knowledge of common image deformations, such as scale, position, and rotation (Miller, Matsakis, & Viola, 2000). While these factors are important, 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.

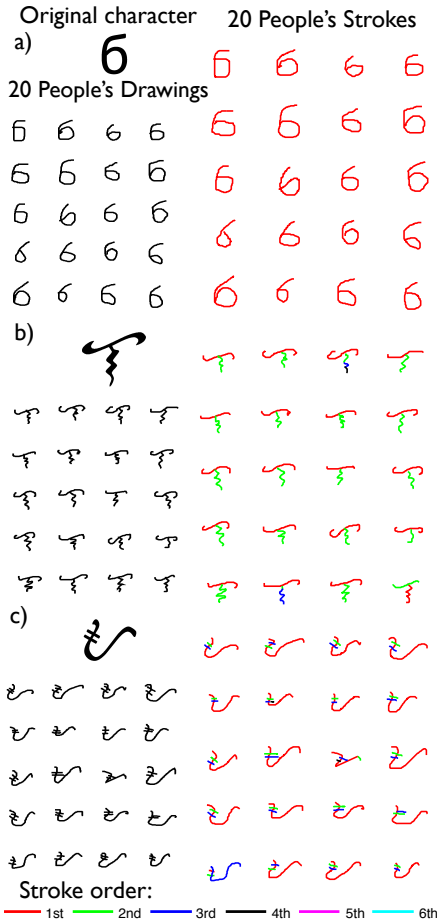

Figure 3: Illustration of the character dataset. Each panel shows the original character, 20 people's image drawings, and 20 people's strokes color coded for order.

There are challenges with conducting a large scale study of character learning. First, people already know the digits and the Latin alphabet, so experiments must be conducted on new characters. Second, 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 each drawing (Fig. 3). Using the dataset, we can investigate the dual problems of understanding human concept learning and building machines that learn from one example. We propose a new model of character learning based on inducing probabilistic part-based templates, 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 vocabulary of candidate strokes 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

similarity data and human accuracy in a challenging one-shot classification task, and compare it with a leading alternative approach from the machine learning literature, 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. We find that the stroke model outperforms the DBM by a large margin on one-shot learning accuracy, and it provides a closer fit to human perceptual similarity.

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 illustrated in Figure 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 (Figure 3, bottom left). In addition to capturing the image, the interface captures the drawer's parse into strokes, shown in Figure 3 (right) where color denotes stroke order.

Drawing methods are remarkably consistent across participants. For instance, Figure 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. Figure 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

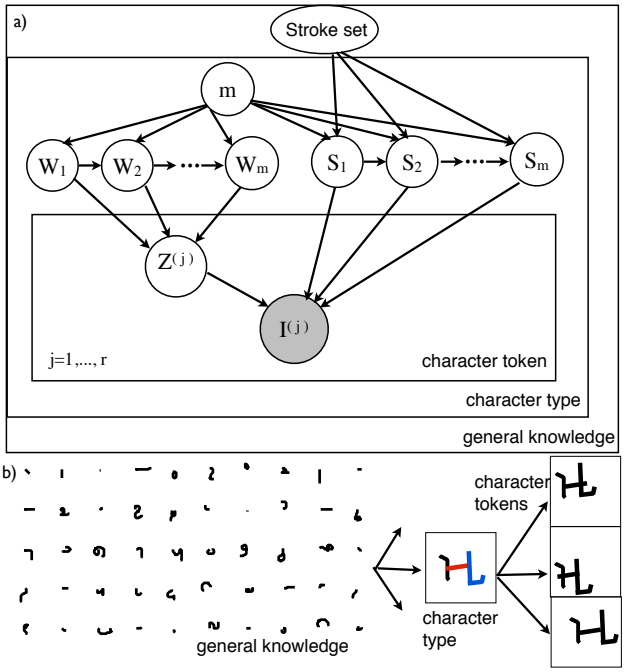

Figure 4: Illustration of the generative process. Character types are defined by strokes S_i and their template locations W_i. Tokens are generated by choosing image specific positions $Z^{(j)}$ from W and producing a pixel image $I^{(j)}$. All variables inside the character type plate are implicitly indexed by character type.

unknown identity. Figure 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 The number of strokes m is picked from a uniform distribution (1 to 10 for simplicity). The first stroke is drawn uniformly from $P(S_1) = 1/K$, where $K = 1000$ is the size of the stroke set, and its general position is drawn uniformly from $P(W_1) = 1/N$ where there are N pixels in the image. Position $W_1 = [w_{x_1}, w_{y_1}]$ has a discrete x and y coordinate. Subsequent strokes $P(S_{i+1}|S_i)$ and positions $P(W_{i+1}|W_i)$ are drawn from the transition model. The transitions are also uniform, but the model could be extended to include a more accurate sequential process.

Generating a character token A character type (S and W) generates a character token $I^{(j)}$, which is a pixel image. While W specifies a rough template for the relative positions of strokes, the image specific positions $Z^{(j)}$ vary slightly from token to token. The conditional distribution $P(Z^{(j)}|W)$ samples a configuration $Z^{(j)}$ similar to W in relative location (if S_i is to the left of S_j in W, it will likely be the case in $Z^{(j)}$), the distribution is translation invariant, meaning shifting the entire image to the left by 4 pixels has equal probability un-

der the probability mass function (Figure 4b). The image specific positions $Z^{(j)} = \{z_{x_1}^{(j)}, z_{y_1}^{(j)}, ..., z_{x_m}^{(j)}, z_{y_m}^{(j)}\}$, as with W, specify discrete x and y coordinates. Denoting $d[x_i, x_j] = (w_{x_i} - w_{x_j}) - (z_{x_i}^{(j)} - z_{x_j}^{(j)})$ as the difference between pairwise offsets for coordinates x_i and x_j,

$$P(Z^{(j)}|W) \propto \exp(-\sum_{i<j}^{m}(\frac{1}{\sigma_w^2}d[x_i, x_j]^2 + \frac{1}{\sigma_w^2}d[y_i, y_j]^2)).$$

Given the strokes $S_1, ..., S_m$ and now their image specific positions $Z^{(j)}$, an image is generated by choosing G pixels to "ink." Intuitively, ink is sprayed across the strokes with Gaussian spray paint. This is captured by lining each stroke with little Gaussian beads that generate ink, borrowed from Revow, Williams, and Hinton (1996). The ink model has a recursive mixture structure; an inked pixel is drawn from a mixture of uniform noise (parameter β) and another mixture over the m strokes, and each stroke is yet another mixture of the Gaussian beads. The mixture over strokes is

$$P(I^{(j)}|S, Z^{(j)}) = \prod_{g=1}^{G}\frac{\beta}{N} + \frac{1-\beta}{m}\sum_{i=1}^{m}P(I_g^{(j)}|S_i, z_{x_i}^{(j)}, z_{y_i}^{(j)}),$$

where index g includes just the inked (black) pixels. The within stroke mixture model is

$$P(I_g^{(j)}|S_i, z_{x_i}^{(j)}, z_{y_i}^{(j)}) \propto \frac{1}{B}\sum_{b=1}^{B}\exp(-\frac{1}{\sigma_b^2}(x_b + z_{x_i}^{(j)} - x_g)^2 - \frac{1}{\sigma_b^2}(y_b + z_{y_i}^{(j)} - y_g)^2),$$

where $x_b + z_{x_i}^{(j)}$ and $y_b + z_{y_i}^{(j)}$ are the bead coordinates translated into position and x_g and y_g are the inked pixel $I_g^{(j)}$ coordinates. The parameters settings $B = 28$, $\sigma_b = 1.5$, $\beta = 0.1$, and $\sigma_w = 2$ provide a good ink/position model but many others are reasonable.

Learning a library of strokes General knowledge of strokes was learned from the drawing data. The character dataset was split randomly into a 25 alphabet "background set" and a 25 alphabet "experiment set." The background set was used to learn the strokes, and the models and people were tested on the experiment set. About 40,000 strokes were aligned and clustered to form $K = 1000$ centroids that comprise the model's library (Figure 4). Stroke trajectories vary widely in length, and each stroke is reduced to a common dimensionality by fitting a cubic B-spline with 10 control points (Revow et al., 1996; Branson, 2004).[1] The cluster centroids are learned in control point space using k-means. Strokes are direction specific meaning left to right and right to left are different centroids.

Inference for one shot learning For one shot learning, the model is given a single example image I and a candidate image $I^{(t)}$. Exact computation of $P(I^{(t)}|I)$ is intractable so it is approximated with maximum a posteriori (MAP) estimate, where

$$\begin{aligned} P(I^{(t)}|I) &= \int_{S,W} P(I^{(t)}|S, W)\int_Z P(S, W, Z|I)\, dZdSdW \\ &\approx P(I^{(t)}|S^*, W^*), \end{aligned}$$

where $S^*, W^* = \underset{S,W,Z}{\text{argmax}}\ P(S, W, Z|I)$. This involves finding strokes S^* and configuration W^* that best explain I, and the space is searched with Markov Chain Monte Carlo (MCMC) and the Metropolis-Hastings algorithm for 50,000 proposals. Given a current state S and W, the algorithm jointly proposes swaps of a stroke S_i and its positions W_i with similar strokes and positions. There are also reversible jump proposals that jointly perturb S_1 to S_m and W while adding a new stroke S_{m+1} and W_{m+1}. There is a corresponding removal proposal and a proposal that permutes indices. We also approximate

$$\begin{aligned} P(I^{(t)}|S^*, W^*) &= \int_{Z^{(t)}} P(I^{(t)}, Z^{(t)}|S^*, W^*)dZ^{(t)} \\ &\approx P(I^{(t)}, Z^{(t)*}|S^*, W^*), \end{aligned}$$

where $Z^{(t)*} = \underset{Z^{(t)}}{\text{argmax}}\ P(I^{(t)}, Z^{(t)}|S^*, W^*)$ using MCMC with 2000 position-specific proposals.

Results

The stroke model, the Deep Boltzmann Machine (DBM, Salakhutdinov & Hinton, 2009), and Nearest Neighbor (NN) in pixel space were evaluated on character learning. These three models were compared on accuracy for one shot learning and also on their fit to human perceptual similarity.

20-way classification from one example

We tested accuracy on 20-way classification, where each training class gets only one example. The 20 training characters are picked at random from different alphabets in the 25 alphabet experiment set, and the models see only one image of each. The models have never seen any of these alphabets or characters before. Accuracy is then tested on novel images drawn from this set of 20 characters. All models receive 28x28 images (binary for the stroke model and NN, grayscale for the DBM). The DBM was pretrained on the 25 background alphabets using a combination of MCMC and variational approximation (see Salakhutdinov & Hinton, 2009). The architecture

[1] 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).

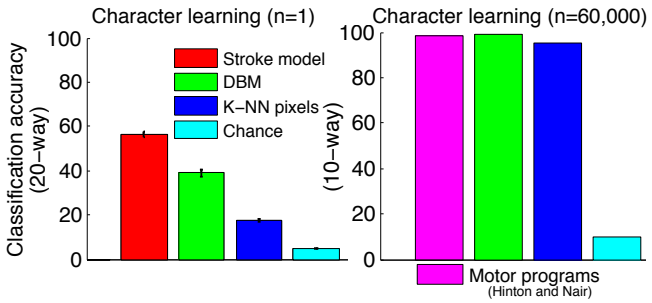

Figure 5: Classification accuracy from one example (left, our results) and on the MNIST digits (right, previously published results not from this work). Error bars are standard error.

was two hidden layers with 1000 units each. After receiving the 20 new characters for one shot learning, the stroke model fits a latent stroke representation to each, and a test image $I^{(t)}$ is classified by picking the largest $P(I^{(t)}|I)$ across the 20 possible training characters I using the MAP approximation. 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 56.3% correct, compared to 39.6% for the DBM and 17.5% for nearest neighbor in pixels (Figure 5 left). This was averaged over many random draws (26) of the training characters and four test examples per class. Figure 6 illustrates one of these draws of 20 characters and the model fits to each image. How would people perform on this task? If people were run directly on 20-way classification, they would continue learning from the test examples. Instead participants were asked to make same vs. different judgements using the experiment set of characters. "Same" trials were two images, side by side, of the same character from two unique drawers, and "different" trials were two different characters. Each participant saw 200 trials using a web interface on Mechanical Turk, and the ratio of same to different trials was 1/4. Across 20 participants who saw different pairings randomly selected from the experiment set of characters, performance was 97.6% correct. Although this is a different task than 20-way classification, it confirms there is a substantial gap between human and machine competence.

To create an interesting juxtaposition with the 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 ($n = 60,000$). 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.

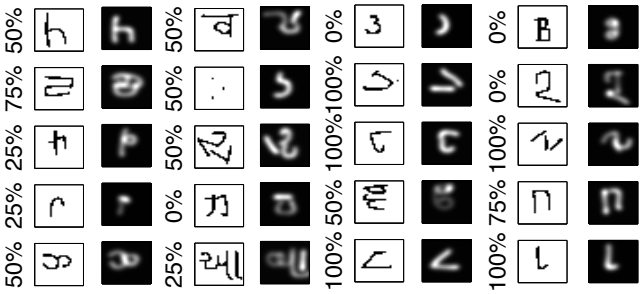

Figure 6: Example run of 20-way classification, showing the training images/classes (white background) and the stroke model's fits (black background) where lighter pixels are higher probability. Accuracy rates are indicated on the 4 test examples (not shown) per class.

Fit to human perceptual similarity

The models were also compared with human perceptual judgments. A collection of six alphabets and four characters each was selected for high confusability within alphabets. Fig. 7 shows the original print images, but participants were shown the handwritten copies. Like the previous experiment, participants were asked to make 200 same vs. different judgments, where the proportion of same trials was 1/4. The task was speeded to avoid near perfect performance, 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.[2] 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" when either character in a pair appeared on the screen (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 were compared to the perceptual data. Each model saw many replications (20) of mock 24-way classification, exactly like the 20-way classification results. For each test image, the goodness of fit to each of the 24 training classes was calculated and ranked from best (24) to worst (1). For each pairing of stimuli, these numbers were added to the similarity matrix while averaging across replications in each cell. Of the models, the stroke model had the strongest block structure. Along with the DBM, it shows additional structure between alphabets that is not evident in the subject responses. After computing a correlation with the human matrix over the unique cells, the stroke model (r=0.72) fits better than the DBM (r=0.63) and image distance (r=0.14).

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

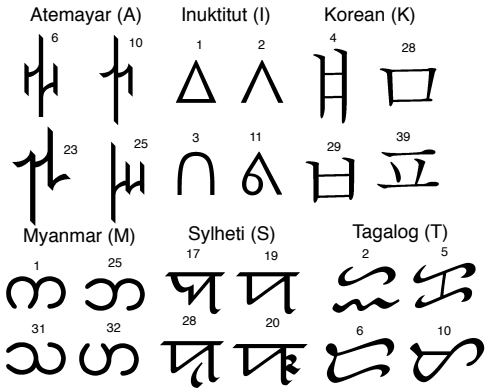

Figure 7: Human perceptual similarity was measured on pairs of these characters, which are from 6 alphabets. The original printed images are shown, but participants saw handwritten drawings. Character index within an alphabet is denoted.

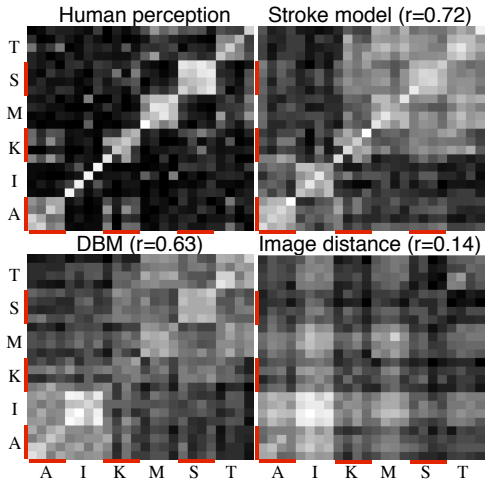

Figure 8: Similarity matrices (lighter is more similar) for the 24 characters in Figure 7. Alphabets are blocked and denoted by their starting letter (A, I, etc.). Character index (Fig. 7) within alphabet blocks is in increasing order, left to right.

Discussion

This paper introduced a generative model where characters are composed of strokes. When shown an image of a character, it attempts to infer a latent set of strokes that explain the pixels in the image. This approach performs well on one shot learning and classification, beating Deep Boltzmann Machines by a wide margin. The stroke model also provided the closest fit to human perceptual judgements across a set of confusable characters.

The stroke model is still far from human competence, although there are many avenues for extension and improvement. For instance, the basic elements in the stroke set are rigid, allowing for translations but no scaling, rotation, or deformation within individual strokes (see Revow et al., 1996). Our large stroke basis provides only discrete approximation to a much richer continuous space. Stroke deformations are completely consistent with our framework, but they make an already challenging inference problem worse. Despite this challenge, data driven proposals for inference could provide such an improvement, where proposals are influenced by bottom-up stroke detectors that look at the image when deciding what to propose.

The new 1600 character dataset supports other interesting problems like alphabet recognition. A unique stroke found in a few examples of an alphabet could be acquired on the fly, allowing generalization to new alphabet examples. By studying more problems like these at the interface of human and machine learning, we can gain new insight into both 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.*

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.

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.