Unsupervised Learning of Visual Structure using Predictive Generative Networks

William Lotter, Gabriel Kreiman & David Cox
Harvard University
Cambridge, MA 02138, USA
{lotter,davidcox}@fas.harvard.edu
gabriel.kreiman@tch.harvard.edu

Abstract

The ability to predict future states of the environment is a central pillar of intelligence. At its core, effective prediction requires an internal model of the world and an understanding of the rules by which the world changes. Here, we explore the internal models developed by deep neural networks trained using a loss based on predicting future frames in synthetic video sequences, using a CNN-LSTM-deCNN framework. We first show that this architecture can achieve excellent performance in visual sequence prediction tasks, including state-of-the-art performance in a standard "bouncing balls" dataset (Sutskever et al., 2009). Using a weighted mean-squared error and adversarial loss (Goodfellow et al., 2014), the same architecture successfully extrapolates out-of-the-plane rotations of computer-generated faces. Furthermore, despite being trained end-to-end to predict only pixel-level information, our Predictive Generative Networks learn a representation of the latent structure of the underlying three-dimensional objects themselves. Importantly, we find that this representation is naturally tolerant to object transformations, and generalizes well to new tasks, such as classification of static images. Similar models trained solely with a reconstruction loss fail to generalize as effectively. We argue that prediction can serve as a powerful unsupervised loss for learning rich internal representations of high-level object features.

1 Introduction

There is a rich literature in neuroscience concerning "predictive coding," the idea that neuronal systems predict future states of the world and primarily encode only deviations from those predictions (Rao & Ballard, 1999; Summerfield et al., 2006; Den Ouden et al., 2012; Rao & Sejnowski, 2000; Chalasani & Principe, 2013; Zhao et al., 2014). While predicting future sensory inputs can be intrinsically useful, here we explore the idea that prediction might also serve as a powerful framework for unsupervised learning. Not only do prediction errors provide a source of constant feedback, but successful prediction fundamentally requires a robust internal model of the world.

The problem of prediction is one of estimating a conditional distribution: given recent data, estimate the probability of future states. There has been much recent success in this domain within natural language processing (NLP) (Graves, 2013; Sutskever et al., 2014) and relatively low-dimensional, real-valued problems such as motion capture (Fragkiadaki et al., 2015; Gan et al., 2015). Generating realistic samples for high dimensional images, particularly predicting the next frames in videos, has proven to be much more difficult. Recently, Ranzato et al. (2014) used a close analogy to NLP models by discretizing image patches into a dictionary set, for which prediction is posed as predicting the index within this set at future time points. This approach was chosen because of the innate difficulty of using traditional losses in video prediction. In pixel space, loss functions such as mean-squared error (MSE) are unstable to slight deformations and fail to capture intuitions of image similarity. As illustrated by Srivastava et al. (2015), predictive models trained with MSE tend to react to uncertainty with blurring. Srivastava et al. (2015) propose a Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) Encoder-Decoder model for next frame prediction and, despite blurry predictions on natural image patches, their results point to the potential of prediction as unsupervised learning, as predictive pre-training improved performance on two action recognition datasets.

A promising alternative to MSE is an adversarial loss, as in the Generative Adversarial Network (GAN) (Goodfellow et al., 2014). This framework involves training a generator and discriminator in a minimax fashion. Successful extensions, including a conditional GAN (Gauthier, 2014; Mirza & Osindero, 2014) and a Laplacian pyramid of GANs (Denton et al., 2015), show its promise as a useful model for generating images.

Here we build upon recent advances in generative models, as well as classical ideas of predictive coding and unsupervised temporal learning, to investigate deep neural networks trained with a predictive loss. We use a model consisting of a convolutional neural network (CNN), an LSTM, and a deconvolutional neural network (deCNN) (Dosovitskiy & Brox, 2015; Kulkarni et al., 2015). Falling in the class of Encoder-Recurrent-Decoder (ERD) architectures (Fragkiadaki et al., 2015), our model is trained end-to-end to combine feature representation learning with the learning of temporal dynamics. In addition to MSE, we implement an adversarial loss (AL).

We demonstrate the effectiveness of our architecture, trained with MSE, on a standard "bouncing balls" experiment (Sutskever et al., 2009) before applying the same architecture to a dataset of computer-generated faces undergoing rotations. This dataset is an appropriate intermediate step between toy examples and full-scale natural images, where we can more fully study the representational learning process. We find that a weighted combination of MSE and AL leads to predictions that are simultaneously consistent with previous frames and visually convincing. Furthermore, over the course of training, the model becomes better at representing the latent variables of the underlying generative model. We test the generality of this representation in a face identification task requiring transformation tolerance. In the classification of *static* images, the model, trained with a predictive loss on dynamic stimuli, strongly outperforms comparable models trained with a reconstruction loss on static images. Thus, we illustrate the promise of prediction as unsupervised model learning from video.

2 Related Work

In addition to the work already cited, the current proposal has strong roots in the idea of learning from temporal continuity. Early efforts in this field demonstrated how invariances to particular transformations can be learned through temporal exposure (Földiák, 1991). Related algorithms, such as Slow Feature Analysis (SFA) (Wiskott & Sejnowski, 2002), take advantage of the persistence of latent causes in the world to learn representations that are robust to noisy, quickly-varying sensory input. More recent work has explicitly implemented temporal coherence in the cost function of deep learning architectures, enforcing the networks to develop a representation where feature vectors of consecutive video frames are closer together than those between non-consecutive frames (Mohabi et al., 2009; Goroshin et al., 2015a; Wang & Gupta, 2015). Related to these ideas, a recent paper proposed training models to linearize transformations observed over sequences of frames in natural video (Goroshin et al., 2015b).

Also related to our approach, especially in the context of rotating objects, is the field of relational feature learning (Memisevic & Hinton, 2007; Taylor & Hinton, 2009). This posits modeling time-series data as learning representations of the *transformations* that take one frame to the next. Recently, Michalski et al. (2014) proposed a predictive training scheme where a transformation is first inferred between two frames and is then applied again to obtain a prediction of a third frame. They reported evidence of a benefit of using predictive training versus traditional reconstruction training.

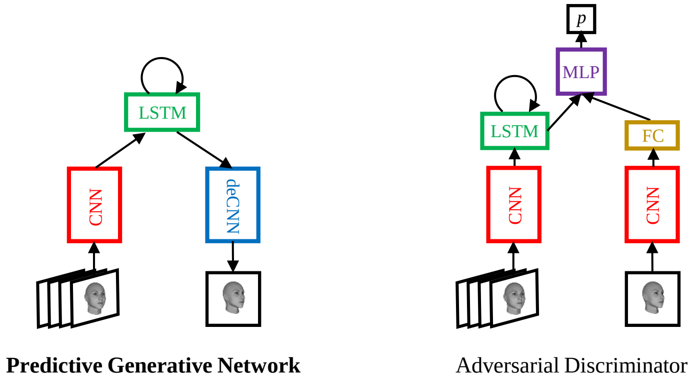
Finally, using variations of autoencoders for unsupervised learning and pre-training, is certainly not new (Erhan et al., 2010; Bengio et al., 2006). In fact, Palm (2012) coined the term "Predictive Encoder" to refer to an autoencoder that is trained to predict future input instead of reconstructing current stimuli. In preliminary experiments, it was shown that such a model could learn Gabor-like filters in training scenarios where traditional autoencoders failed to learn useful representations.

3 Predictive Generative Networks

A schematic of our framework is shown in Figure 1. The generative model first embeds a sequence of frames, successively, into a lower-dimensional feature space using a CNN. We use a CNN consisting of two layers of convolution, rectification, and max-pooling. The CNN output is passed to an LSTM network (Hochreiter & Schmidhuber, 1997). Briefly, LSTM units are a particular type of hidden unit that improve upon the vanishing gradient problem that is common when training RNNs (Bengio et al., 1994). An LSTM unit contains a cell, c_t, which can be thought of as a memory state. Access to the cell is controlled through an input gate, i_t, and a forget gate, f_t. The final output of the LSTM unit, h_t, is a function of the cell state, c_t, and an output gate, o_t. We use a version of the LSTM with the following update equations:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)\\
c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)\\
h_t &= o_t \tanh(c_t)
\end{aligned}$$

where x_t is the input to the LSTM network, $W_{\bullet\bullet}$ are the weight matrices, and σ is the elementwise logistic sigmoid function. We use 1568 LSTM units for the bouncing balls dataset and 1024 for the rotating faces. All models were implemented using the software package Keras (Chollet, 2015).



Predictive Generative Network

Adversarial Discriminator

Figure 1: Predictive Generative Network (PGN)

Upon processing the last output of the CNN, the LSTM hidden state is outputted to a deCNN, which produces a predicted image. The parameters of the network are chosen such that the predicted image is the same size of the input. For the rotating faces dataset, the deCNN consists of a fully-connected (FC) layer, followed by two layers of nearest-neighbor upsampling, convolution, and rectification. The last layer also contains a saturating non-linearity set at the maximum pixel value. Due to the lower dimensional size, the FC layer of the deCNN was omitted for the model trained on the bouncing balls dataset.

With adversarial loss, we also considered concatenating the LSTM hidden state with a random vector before it is passed to the deCNN, which would allow sampling as in previous conditional GANs (Gauthier, 2014; Mirza & Osindero, 2014). However, in our case, the sampling ended up being highly peaked, with different samples from the same input sequence being nearly indistinguishable, possibly because our sequences are deterministic, so the random vector was later discarded.

For adversarial loss, the predicted frame from the generator is passed to a discriminator network, which is also conditioned on the original input sequence. Similar to the generator, the input sequence is passed through a CNN and LSTM. We experimented with sharing the CNN and LSTM weights between the discriminator and generator, but ultimately had best results with each network having its own separate set of parameters. After processing the last input frame, the LSTM hidden state of the discriminator is concatenated with an encoding of the proposed next frame in the sequence, either the true frame of the generator's output. The encoding consists of a CNN and a FC layer, resulting in feature vector with the same size of the LSTM output. The concatenation of these two vectors is passed to a multi-layer perceptron (MLP), consisting of three FC layers with a sigmoidal read-out.

We use the original formulation of the adversarial loss function (Goodfellow et al., 2014). The discriminator outputs a probability that a proposed frame came from the ground truth data. It is trained to maximize this probability when the frame came from the true distribution and minimize it when it is produced by the generator. The generator is trained to fool the discriminator. Let $x^i_{1:t}$ be an input sequence of t frames and x^i_{t+1} be the true next frame. Let the proposed frame from the generator be $G(x^i_{1:t})$ and $D(\bullet, x^i_{1:t})$ be the discriminator's output. Given a mini-batch size of n sequences, the loss of the discriminator, $L^{(AL)}_D$, and of the generator, $L^{(AL)}_G$, have the form:

$$\begin{aligned} L^{(AL)}_D &= -\frac{1}{2n}\sum_{i=1}^{n}[\log D(x^i_{t+1}, x^i_{1:t}) + \log(1 - D(G(x^i_{1:t}), x^i_{1:t}))] \\ L^{(AL)}_G &= \frac{1}{n}\sum_{i=1}^{n}\log(1 - D(G(x^i_{1:t}), x^i_{1:t})) \end{aligned}$$

As in the original paper (Goodfellow et al., 2014), we actually train the generator to maximize $\log(D(G(x^i_{1:t}), x^i_{1:t}))$ and not minimize $\log(1 - D(G(x^i_{1:t}), x^i_{1:t}))$, as the latter tends to saturate early in training.

Due to complementary roles for MSE and AL in training the generator, we combine both in a weighted loss, $L^{(tot)}_G$, controlled by a hyperparameter λ:

$$L^{(tot)}_G = L^{(MSE)}_G + \lambda L^{(AL)}_G$$

Even for high values of λ, the MSE loss proved to be useful as it stabilizes and accelerates training.

4 Prediction Performance

We evaluated the Predictive Generative Networks (PGNs) on two datasets of synthetic video sequences. As a baseline to compare against other architectures, we first report performance on a standard bouncing balls paradigm (Sutskever et al., 2009). We then proceed to a dataset containing out-of-the-plane rotations of computer-generated faces, where we thoroughly analyze the learned representations.

4.1 Bouncing Balls

The bouncing balls dataset is a common test set for models that generate high dimensional sequences. It consists of simulations of three balls bouncing in a box. We followed standard procedure to create 4000 training videos and 200 testing videos (Sutskever et al., 2009) and used an additional 200 videos for validation. Our networks were trained to take a variable number of frames as input, selected randomly each epoch from a range of 5 to 15, and output a prediction for the next timestep. Training with MSE was very effective for this dataset, so AL was not used. Models were optimized using RMSprop (Tieleman & Hinton, 2012) with a learning rate of 0.001. In Table 1, we report the average squared one-step-ahead prediction error per frame. Our model compares favorably to the recently introduced Deep Temporal Sigmoid Belief Network (Gan et al., 2015) and restricted Boltzmann machine (RBM) variants, the recurrent temporal RBM (RTRBM) and the structured RTRBM (SRTRBM) (Mittelman et al., 2014). An example prediction sequence is shown in Figure 2, where each prediction is made one step ahead by using the ten previous frames as input.

4.2 Rotating Faces

For the rotating faces dataset, each video consists of a unique, randomly generated face rotating about the vertical axis with a random speed and initial angle. Speed is sampled uniformly from $[0, \pi/6]$ rad/frame with an initial angle sampled from $[-\pi/2, \pi/2]$, where 0 corresponds to a frontal

Table 1: Average prediction error for the bouncing balls dataset. †(Gan et al., 2015) ◇(Mittelman et al., 2014)

Model	Error
PGN (MSE)	$\mathbf{0.65 \pm 0.11}$
DTSBN †	2.79 ± 0.39
SRTRBM ◇	3.31 ± 0.33
RTRBM ◇	3.88 ± 0.33
Frame t-1	11.86 ± 0.27

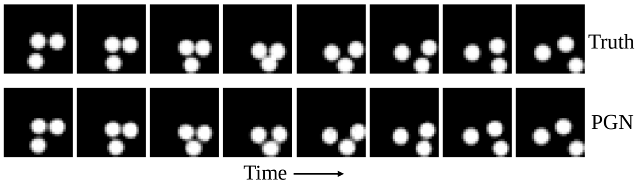


Figure 2: Example prediction sequence for bouncing balls dataset. Predictions are repeatedly generated one step ahead using the prior ten frames as input.

view. Input sequences consist of 5 frames of size 150x150 pixels. We use 4000 clips for training and 200 for validation and testing.

Generative models are often evaluated using a Parzen window estimate of the log-likelihood (Breuleux et al., 2011), but due to the deficiencies of this approach for high dimensional images, we chose values of the weighting parameter between MSE and AL, λ, based on qualitative assessment. Adversarial models are notoriously difficult to train (Chintala et al., 2015) and we empirically found benefits in giving the generator and discriminator a "warm start". For the generator, this corresponded to initializing from a solution trained solely with MSE. This is analogous to increasing the value of λ over training, thus ensuring that the models learn the low frequency components first, and then the high frequency components, akin to the LAPGAN approach (Denton et al., 2015). For the discriminator, we used a pre-training scheme where it was first trained against a generator with a high value of λ. This exposes the discriminator to a wide variety of stimuli in pixel space early in training, which helps it quickly discriminate between real and generated images when it is subsequently paired with the MSE-initialized generator. For the data shown in this paper, these initialization schemes are used and λ is set to 0.0002. The generator is optimized using RMSprop (Tieleman & Hinton, 2012) with a learning rate of 0.001. The discriminator is trained with stochastic gradient descent (SGD), with a learning rate of 0.01 and momentum of 0.5.

Example predictions are shown in Figure 3. We compare the results of training with MSE to the weighted AL/MSE model. Predictions are for faces not seen during training. Both models successfully estimate the angle and basic shape of the face very well. However, the MSE model produces blurred, low-passed versions as expected, whereas the AL/MSE model generates images with high detail. Most notably, the AL/MSE model has learned that faces contain conspicuous eyes and ears, which are largely omitted by the MSE model. When the AL/MSE model does make mistakes, it's often through generating faces that notably look realistic, but seem slightly inconsistent with the identity of the face in the preceding frames. This can be seen in the second row in the right panel of Figure 3. Weighting AL higher exaggerates this effect. One would hope that the discriminator would be able to discern if the identity changed for the proposed rotated view, but interestingly, even humans struggle with this task (Wallis & Bulthoff, 2001).

5 Exploring Latent Representation Learning

Beyond generating realistic predictions, we are interested in understanding the representations learned by the predictive models, especially in relation to the underlying generative model. The faces are created according to a principal component analysis (PCA) in "face space", which was estimated from real-world faces. In addition to the principal components (PCs), the remaining latent variables are the initial angle and rotation speed.

A decoding analysis was performed in which an L2-regularized regression was used to estimate the latent variables from the LSTM representation. We decoded from the hidden unit responses after five time steps, i.e. the last time step before the hidden representation is outputted to the deCNN to produce a predicted image. The regression was fit, validated, and tested using a different dataset than the one used to train the model.

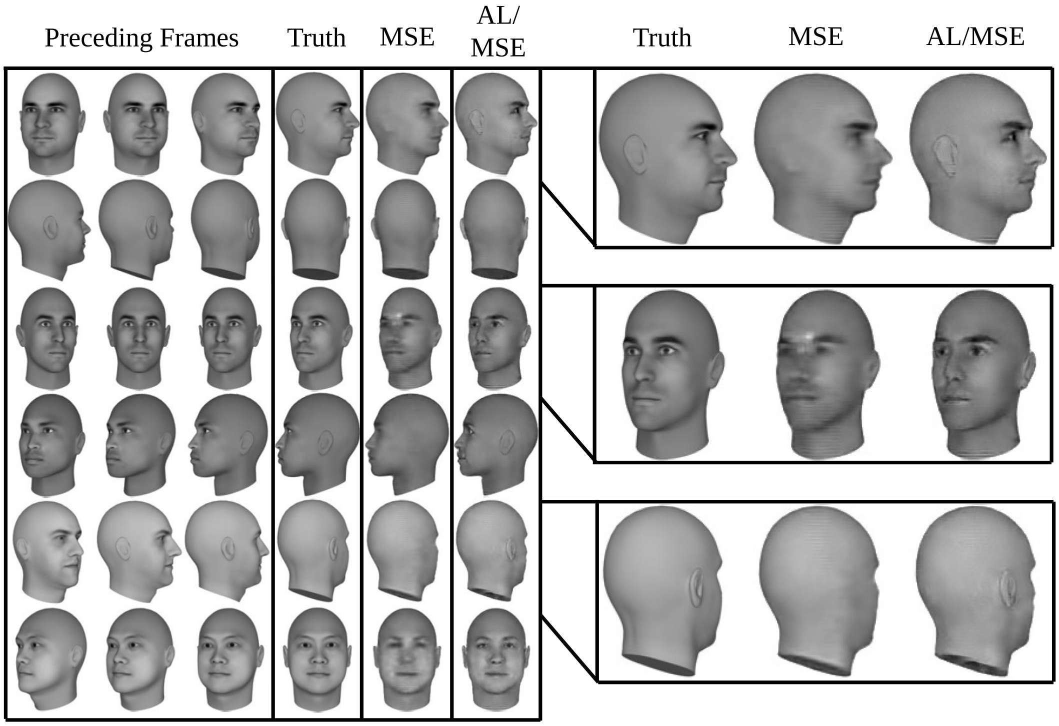


Figure 3: Example predictions for the rotating faces dataset. Predictions for models trained with MSE and a weighted MSE and adversarial loss (AL) are shown.

As a baseline, we compare decoding from the predictive models to a model with the same architecture, trained on precisely the same stimulus set, but with a reconstruction loss. Here, the input sequence is all six frames, for which the model is trained to reconstruct the last. Note, the model cannot simply copy the input, but must learn a low dimensional representation, because the LSTM has a dimension size much less than the input (1024 v.s. $150\text{x}150 = 22.5\text{K}$), e.g. the common autoencoder scenario.

The decoding results for the initial angle, rotation speed, and first four principal components are contained in Table 2. Although they produce visually distinct predictions, the MSE and AL/MSE PGNs show similar decoding performance. This is not surprising since the PCs dictate the shape of the face, which the MSE model estimates very well. Nevertheless, both predictive models strongly outperform the autoencoder. There are more sophisticated ways to train autoencoders, including denoising criteria (Vincent et al., 2008), but here we show that, for a fixed training set, a predictive loss can lead to a better representation of the underlying generative model than a reconstruction loss.

Table 2: Decoding accuracy (r^2) of latent variables from the LSTM hidden unit representation.

Model	**Angle**	**Speed**	**PC1**	**PC2**	**PC3**	**PC4**
PGN (MSE)	0.994	0.986	0.877	0.826	0.723	0.705
PGN (AL/MSE)	0.994	0.990	0.873	0.828	0.724	0.686
Autoencoder (MSE)	0.943	0.927	0.834	0.772	0.655	0.635

To gain insight into the learning dynamics, we show decoding performance for both the hidden state and cell state as a function of training epoch for the MSE model in Figure 4. Epoch 0 corresponds to the random initial weights, from which the latent variables can already be decoded fairly well, which is expected given the empirical evidence and theoretical justifications for the success of random weights in neural networks (Jarrett et al., 2009; Pinto et al., 2009; Saxe et al., 2010). Still, it is clear that the ability to estimate all latent variables increases over training. The model quickly peaks at its ability to linearly encode for speed and initial angle. The PCs are learned more slowly, with decoding accuracy for some PCs actually first decreasing while speed and angle are rapidly learned. The sequence in which the model learns is reminiscent of theoretical work supporting the notion that modes in the dataset are learned in a coarse-to-fine fashion (Saxe et al., 2013).

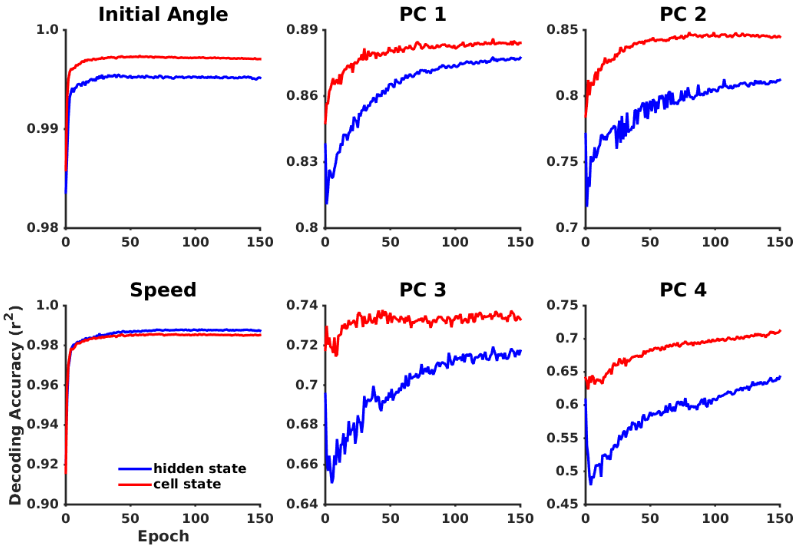


Figure 4: Dynamics of latent variable decoding from internal representation of PGN (MSE)

To understand the representational changes accompanying the increase in decoding performance, we provide visualizations of the hidden unit feature space over training in Figures 5 and 6. Figure 5 contains a multidimensional-scaling (MDS) plot for the initial random weights and the weights after Epoch 150 trained with MSE. Points are colored by PC1 value and rotation speed. Although a regression on this feature space at Epoch 0 produces an r^2 of ~0.83, it is apparent that the structure of this space changes with training. To have a more clear understanding of these changes, we linearized the feature space in two dimensions with axes pointing in the direction of the regression coefficients for decoding PC1 and rotation speed. Points from a held-out set were projected on these axes and plotted in Figure 6 and we show the evolution of the projection space, with regression coefficients calculated separately for each epoch. Over training, the points become more spread out over this manifold. This is not due to an overall increase in feature vector length, as this does not increase over training. Thus, with training, the variance in the feature vectors become more aligned with the latent variables of the generative model.

The previous analyses suggest that the PGNs learn a low dimensional, linear representation of the face space. This is further illustrated in Figure 7. Here, the feature representation of a given seed face is calculated and then the feature vector is perturbed in the direction of a principal component axis, as estimated in the decoding analysis. The new feature vector is then passed to the pre-trained deCNN to produce an image. The generated image is compared with changing the PC value directly in the face generation software. Figure 7 shows that the extrapolations produce realistic images, especially for the AL/MSE model, which correlate with the underlying model. The PC dimensions do not precisely have semantic meanings, but differences can especially be noticed in the cheeks and jaw lines. The linear extrapolations in feature space generally match changes in these features, demonstrating that the models have learned a representation where the latent variables are linear.

While the generation of frame-by-frame future predictions is useful *per se*, we were especially interested in the extent to which prediction could be used as an unsupervised loss for learning representations that are suited to other tasks. We tested this hypothesis through a task completely orthogonal to the original loss function, namely classification of static images. As a control, to specifically isolate the effect of the loss itself, we trained comparable models using a reconstruction loss and either dynamic or static stimuli. The first control was carried over from the latent variable decoding analysis and had the same architecture and training set of the PGN, but was trained with a reconstruction loss

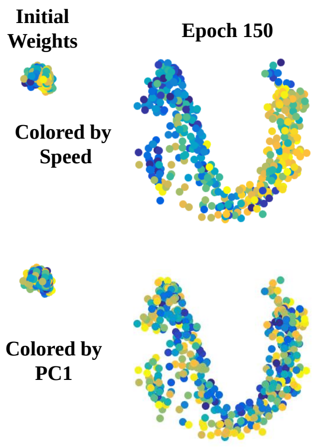


Figure 5: Multidimensional scaling plot of the LSTM representation demonstrating changes with training.

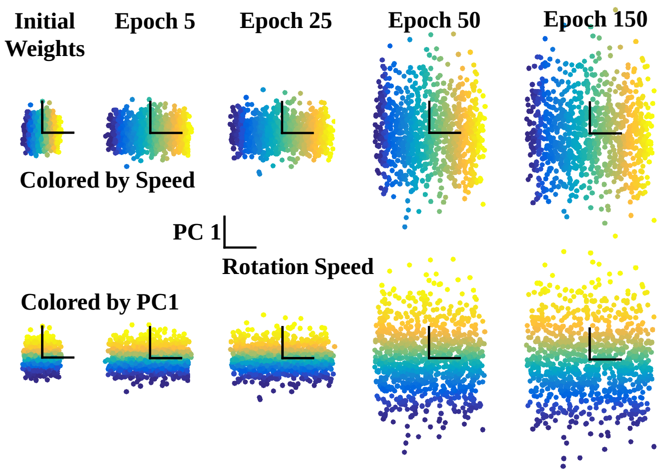


Figure 6: Projection of LSTM feature space on latent variables axes. Axes are in the direction of regression coefficients. A different regression was fit for each epoch.

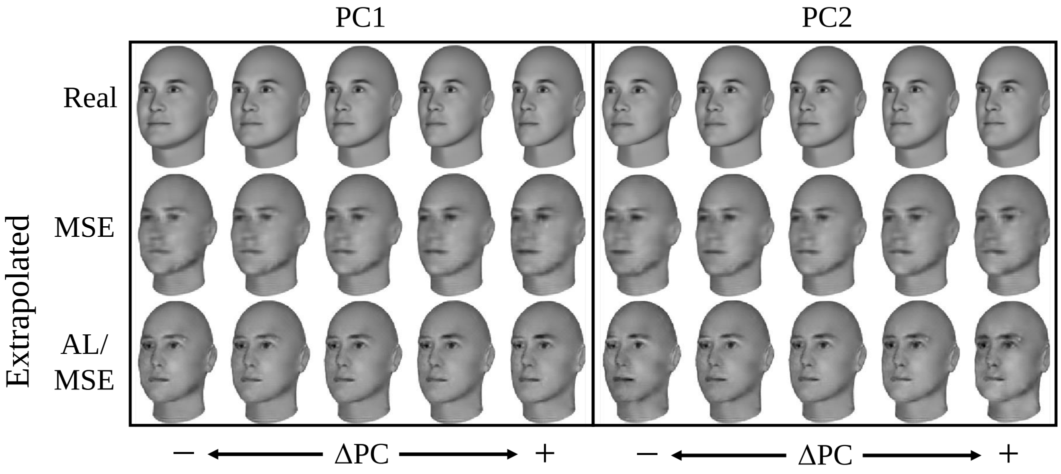


Figure 7: Linearly moving through LSTM feature space along principal component axes.

(denoted as **AE LSTM (dynamic)** in Fig. 8). The next model again had the same architecture and a reconstruction loss, but was trained on static videos [**AE LSTM (static)**]. A video was created for each unique frame in the original training set. For the last two models, the LSTM was replaced by a fully-connected (FC) layer, one with an equal number of weights [**AE FC (= # weights)**] and the other with an equal number of units [**AE FC (= # units)**] as the LSTM. These were trained in a more traditional autoencoder fashion to simply reconstruct single frames, using every frame in the original video set. All control models were trained with MSE since AL is more sensitive to hyperparameters.

The classification dataset consisted of 50 randomly generated faces at 12 equally-spaced angles between $[-\frac{\pi}{2}, \frac{\pi}{2}]$. A support vector machine (SVM) was fit on the feature representations of each model. For the models containing the LSTM layer, the feature representation at the fifth time step was chosen. To test for transformation tolerance, training and testing were done with separate sets of angles.

The classification performance curves are shown in Figure 8. While all models show improvement compared to random initial weights, the predictive models outperform the controls. The MSE PGN has the highest score for each size of the training data. The AL/MSE PGN performs moderately worse, but still better than the control models. This is likely because, as previously mentioned, the

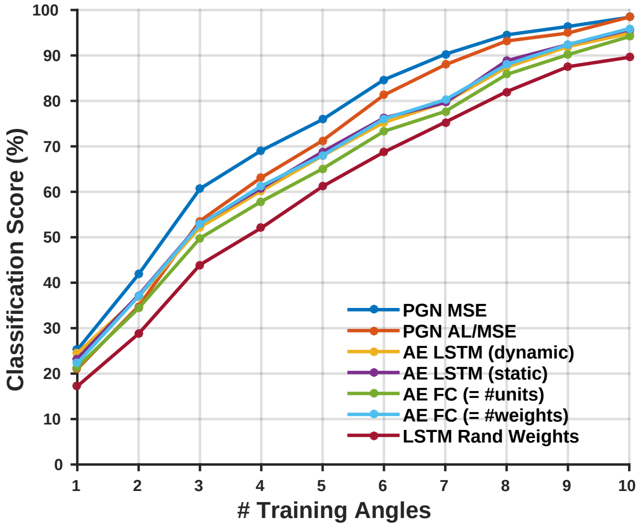


Figure 8: Classification accuracy on a 50-way face identification task. AE: Autoencoder.

AL/MSE model can tend to produce realistic faces, but are somewhat unfaithful to the underlying identity. While this is a relatively simple task compared to modern machine learning datasets, it provides a proof-of-principle that a model trained with a unsupervised, predictive loss on dynamic sequences can learn interesting structure, which is even useful for other tasks.

6 Conclusion

In extending ideas of predictive coding and learning from temporal continuity to modern, deep learning architectures, we have shown that an unsupervised, predictive loss can result in a rich internal representation of visual objects. Our CNN-LSTM-deCNN models trained with such a loss function successfully learn to predict future image frames in several contexts, ranging from the physics of simulated bouncing balls to the out-of-plane rotations of previously unseen computer-generated faces. However, importantly, models trained with a predictive unsupervised loss are also well-suited for tasks beyond the domain of video sequences. For instance, representations trained with a predictive loss outperform other models of comparable complexity in a supervised classification problem with static images. This effect is particularly pronounced in the regime where a classifier must operate from just a few example views of a new object (in this case, face). Taken together, these results support the idea that prediction can serve as a powerful framework for developing transformation-tolerant object representations of the sort needed to support one- or few-shot learning.

The experiments presented here are all done in the context of highly-simplified artificial worlds, where the underlying generative model of the stimuli is known, and where the number of degrees of freedom in the data set are few. While extending these experiments to real world imagery is an obvious future priority, we nonetheless argue that experiments with highly controlled stimuli hold the potential to yield powerful guiding insights. Understanding how to scale predictive generative models of this form to encompass all of the transformation degrees of freedom found in real-world objects is an area of great interest for future research.

Acknowledgments

We would like to thank Chuan-Yung Tsai for fruitful discussions. This work was supported by a grant from the National Science Foundation (NSF IIS 1409097) and by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

REFERENCES

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 1994.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *NIPS*. 2006.

Olivier Breuleux, Yoshua Bengio, and Pascal Vincent. Quickly generating representative samples from an rbm-derived process. *Neural Computation*, 2011.

Rakesh Chalasani and Jose C. Principe. Deep predictive coding networks. *CoRR*, abs/1301.3541, 2013.

Soumith Chintala, Emily L. Denton, Arthur Szlam, and Robert Fergus. The eyescream project, neuralnets dreaming natural images, 2015. URL `http://soumith.ch/eyescream/`.

François Chollet. Keras, 2015. URL `http://keras.io/`.

Hanneke EM Den Ouden, Peter Kok, and Floris P De Lange. How prediction errors shape perception, attention and motivation. *Frontiers in Psychology*, 2012.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Robert Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015.

Alexey Dosovitskiy and Thomas Brox. Inverting convolutional networks with convolutional networks. *CoRR*, abs/1506.02753, 2015.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, 2010.

Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 1991.

Katerina Fragkiadaki, Sergey Levine, and Jitendra Malik. Recurrent network models for kinematic tracking. *CoRR*, abs/1508.00271, 2015.

Zhe Gan, Chunyuan Li, Ricardo Henao, David Carlson, and Lawrence Carin. Deep temporal sigmoid belief networks for sequence modeling. *CoRR*, abs/1509.07087, 2015.

Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition*, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*. 2014.

Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. *CoRR*, abs/1412.6056, 2015a.

Ross Goroshin, Michaël Mathieu, and Yann LeCun. Learning to linearize under uncertainty. *CoRR*, abs/1506.03011, 2015b.

Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

Kevin Jarrett, Koray Kavukcuoglu, MarcAurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*. 2009.

Tejas D. Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. *CoRR*, abs/1503.03167, 2015.

Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations. In *CVPR*. 2007.

Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent ”grammar cells”. In *NIPS*. 2014.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

Roni Mittelman, Benjamin Kuipers, Silvio Savarese, and Honglak Lee. Structured recurrent temporal restricted boltzmann machines. In *ICML*, pp. 1647–1655. 2014.

Hossein Mohabi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, pp. 737–744. 2009.

Rasmus Berg Palm. Prediction as a candidate for learning deep hierarchical models of data. *Master's thesis, Technical University of Denmark*, 2012.

Nicolas Pinto, David Doukhan, James J. DiCarlo, and David D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput Biol*, 2009.

Marc'Aurelio Ranzato, Arthur Szlam, Joan Bruna, Michaël Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *CoRR*, abs/1412.6604, 2014.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 1999.

Rajesh P. N. Rao and Terrence J. Sejnowski. Predictive sequence learning in recurrent neocortical circuits. In *NIPS*. 2000.

Andrew Saxe, Maneesh Bhand, Zhenghao Chen, Pang Wei Koh, Bipin Suresh, and Andrew Y. Ng. On random weights and unsupervised feature learning. In *Workshop: Deep Learning and Unsupervised Feature Learning (NIPS)*. 2010.

Andrew M. Saxe, James L. Mcclelland, and Surya Ganguli. Learning hierarchical category structure in deep neural networks. *Proc. of the Cognitive Science Society*, 2013.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR*, abs/1502.04681, 2015.

Christopher Summerfield, Tobias Egner, Matthew Greene, Etienne Koechlin, Jennifer Mangels, and Joy Hirsch. Predictive codes for forthcoming perception in the frontal cortex. *Science*, 314, 2006.

Ilya Sutskever, Geoffrey E. Hinton, and Graham W. Taylor. The recurrent temporal restricted boltzmann machine. In *NIPS*. 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le. Sequence to sequence learning with neural networks. In *NIPS*. 2014.

Graham W. Taylor and Geoffrey Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of The 26th International Conference on Machine Learning*, pp. 1–8. 2009.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5 - rmsprop, coursera. *Neural networks for machine learning*, 2012.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pp. 10961103. 2008.

Guy Wallis and Heinrich H. Bulthoff. Effects of temporal association on recognition memory. *PNAS*, 2001.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. *CoRR*, abs/1505.00687, 2015.

Laurenz Wiskott and Terrence J. Sejnowski. Learning invariance from transformation sequences. *Neural Computation*, 2002.

Mingmin Zhao, Chengxu Zhuang, Yizhou Wang, and Tai Sing Lee. Predictive encoding of contextual relationships for perceptual inference, interpolation and prediction. *CoRR*, abs/1411.3815, 2014.