The final publication is available at Springer via http://dx.doi.org/10.1007/s11263-014-0765-x

A Neural Autoregressive Approach to Attention-based Recognition

Yin Zheng
Department of Electronic Engineering,
Tsinghua University,
Beijing, China, 10084
y-zheng09@mails.tsinghua.edu.cn

R. S. Zemel
Department of Computer Science,
University of Toronto,
Toronto, Canada, M5S 3G4
zemel@cs.toronto.edu

Yu-Jin Zhang
Department of Electronic Engineering
Tsinghua University,
Beijing, China, 10084
zhang-yj@tsinghua.edu.cn

Hugo Larochelle
Départment d'Informatique
Université de Sherbrooke,
Sherbrooke (QC), Canada, J1K 2R1
hugo.larochelle@usherbrooke.ca

September 13, 2014

Abstract

Tasks that require the synchronization of perception and action are incredibly hard and pose a fundamental challenge to the fields of machine learning and computer vision. One important example of such a task is the problem of performing visual recognition through a sequence of controllable fixations; this requires *jointly* deciding *what* inference to perform from fixations and *where* to perform these fixations. While these two problems are challenging when addressed separately, they become even more formidable if solved jointly. Recently, a restricted Boltzmann machine (RBM) model was proposed that could learn meaningful fixation policies and achieve good recognition performance. In this paper, we propose an alternative approach based on a feed-forward, auto-regressive architecture, which permits exact calculation of training gradients (given the fixation sequence), unlike for the RBM model. On a problem of facial expression recognition, we demonstrate the improvement gained by this alternative approach. Additionally, we investigate several variations of the model in order to shed some light on successful strategies for fixation-based recognition.

1 Introduction

Machine learning developments have had an immense impact on how visual recognition is tackled in modern computer vision systems. For example, kernel methods facilitated the development of powerful classifiers taking advantage of the vast research on hand-designed low-level feature representations [1]. Unsupervised learning methods such as sparse coding have also provided a paradigm for automatically discovering meaningful mid-level feature representations for recognition [2].

More recently, the developments brought by research on deep learning have allowed the development of visual recognition systems trained end-to-end and relying on even less prior knowledge specific to computer vision. The best illustration of this trend is the work of Krizhevsky et al. [3], in which a deep convolutional neural network was trained to achieve state-of-the-art object recognition performance in the Large Scale Visual Recognition Challenge, outperforming other systems by a large margin.

While computer vision systems are increasingly adaptive, the way they function is still quite different from how humans solve recognition problems. One of the most obvious differences is in the role that attention plays in human perception. Human perception is intrinsically a sequential process. Indeed, we use eye saccades to move our eyes and foveate at different positions in a given visual scene. These foveated fixations allow us to assign more (brain) resources to the analysis of the most relevant details in a visual scene for the task at hand.

Such a strategy could also be beneficial in a computer system. While most systems often reduce the resolution of the input image before processing, an adaptive attentional process could learn to ignore irrelevant parts of the image and concentrate its computations to finer details that are only available in the original resolution and are crucial for recognition. Ignoring irrelevant parts could even, by itself, be a beneficial strategy that would allow the system to ignore possibly misleading information. The use of high-resolution image information is also bound to become more important as computer vision tackles recognition problems with increasingly fine-grained sets of visual categories and objects. Moreover, as the recent history of computer vision suggests, learning-based methods are likely to play an important role in the future solutions for this problem.

A step in this direction was recently made using a restricted Boltzmann machine (RBM) model [4]. The system as a whole consists of a third-order classification RBM that combines information from a sequence of fixations into a learned representation of the image, as well as an adaptive controller which makes predictions for the best fixation positions. Unlike many previous recognition systems based on fixation sequences, both the classification and controller components are trained simultaneously and jointly. This is thus a promising direction for future research on fixation-based recognition.

In this work, we propose an alternative approach which replaces the RBM recognition component with a neural, feed-forward autoregressive model. Recently, neural autoregressive networks have been shown to be competitive alternatives to RBMs as models of binary [5], real-valued [6] or text [7] data. Coined Neural Autoregressive Distribution Estimators (NADE), such models have the notable advantage of being trainable by backpropagation and not requiring approximations to their training gradients. As we will see, our proposed extension, *Fixation NADE*, is able to learn a meaningful representation of images through fixations and to outperform the RBM-based model on the task of facial emotion recognition evaluated in Larochelle and Hinton [4]. We also find this model is easier to train and investigate, thus we go beyond the experimental work in Larochelle and Hinton [4] and evaluate different variations in the learning paradigm used by the system, shedding light on successful strategies towards more competitive fixation-based recognition systems.

2 Recognition as a Sequence of Fixations

Assuming that we are given a training set of image and labels pairs $\left\{\left(\mathbf{I}^{(t)}, y^{(t)}\right)\right\}_{t=1}^{N}$, our task is to learn a model which can predict the labels $y^{(t)}$ (e.g., $y^{(t)} \in \{1, 2, ..., C\}$) given the associated image $\mathbf{I}^{(t)}$. The standard way of dealing with this problem in computer vision would be to extract some features (e.g. SIFT [8], HOG [9], etc.) densely from the whole image $\mathbf{I}^{(t)}$ and from those directly predict the label $y^{(t)}$, such as in Lazebnik et al. [1], Yang et al. [2]. These features could also be learned using a convolutional network, jointly with the classification task [3]. These methods share the requirement that perception must distribute computations uniformly over the whole image. For a given computational budget, this implies that less computations can be assigned to the more important parts of the image, containing the relevant information.

The human visual system performs visual recognition tasks differently. As shown in Mathe and Sminchisescu [10], given the same image, the fixations points that are performed across different people are highly consistent both *spatially* and *sequentially*, and are also highly influenced by the task. This suggests that humans fixate different regions to obtain information in accordance with the task at hand, with each fixation being influenced by the previous ones. Southall [11] also showed that multiple glimpses, limited by a small hole, are sufficient for people to perceive a shape. This is referred to as anorthoscopic perception. Furthermore, the work of Fazl et al. [12] suggests that shape information (the *"what"*) from each fixation cannot simply be aggregated without modulating the aggregation by the relative position of these fixations (the *where*).

Larochelle and Hinton [4] propose to combine the *what* and *where* using an RBM with third-order connections, i.e., connections that involve the multiplication of the *what* and *where* representations. This third-order RBM, termed Fixation RBM, was shown to be successful at combining the information extracted by fixations in order to recognize emotions in images of faces. Most impressively, the Fixation RBM was trained jointly with an attentional component that controlled the selection of fixations to be made in the input image. It was thus able to learn the *what* and *where* simultaneously, instead of treating these problems separately, marking an important step towards entirely adaptive, attentional computer vision systems.

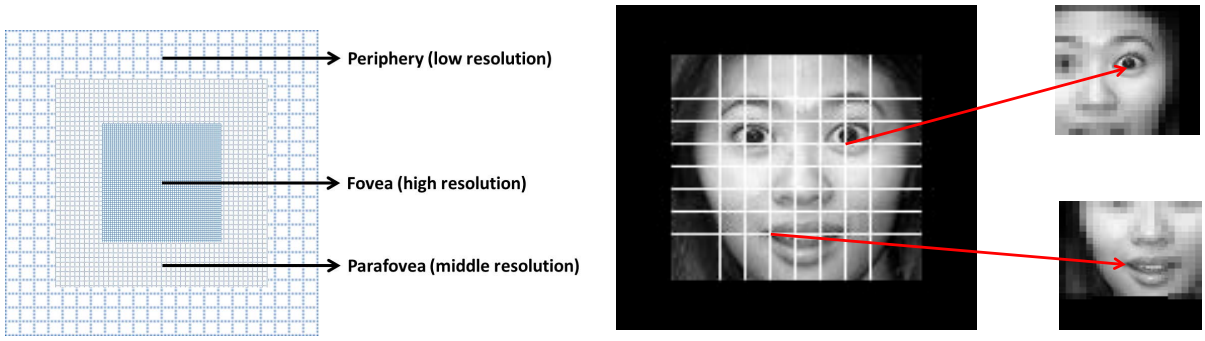

Figure 1: Illustration of the retinal transformation $\mathbf{r}(\mathbf{I}, (i, j))$. It extracts information at up to three resolutions, based on subpatches centered at the fixation position (i, j). The two lower resolution subpatches are subsampled with 2×2 and 4×4 averaging. The retinal vector output by the retinal transformation is then obtained by flattening the three subpatches and concatenating them. Possible fixation positions are constrained to a grid, laid over the image. A visualization of the information contained in the retinal transformation for two fixation positions is illustrated on the right.

However, a disadvantage of the Fixation RBM is that its training requires the estimation of gradients that are intractable. Indeed, because the RBM is an undirected graphical model, its log-likelihood gradients require a so-called negative phase expectation that can only be approximated with sampling. A biased estimator of this expectation, based on contrastive divergence [13], is often used. The difficulty of estimating a training gradient becomes even more serious as we move to deeper architectures with more than a single hidden layer. In fact, while the emotion recognition model in Larochelle and Hinton [4] used an additional hidden layer for each individual fixation, this hidden layer had to be pretrained first and was only used as a fixation preprocessing.

Feed-forward architectures are considerably easier to extend and generalize to deep architectures than undirected graphical models. Hence we propose to investigate this modeling paradigm for attention-based visual recognition. Specifically, motivated by the recent success of Neural Autoregressive Distribution Estimators (NADE) models as alternatives to the RBM, we attack the problem of modeling sequences of fixations using a neural autoregressive architecture. We propose Fixation NADE, which is a direct analogue of the Fixation RBM, but can be trained easily by backpropagation. As we will see, its performance compares favorably with that of the Fixation RBM.

2.1 Retinal Transformation

We first describe how fixation-based observations are extracted from an input image.

In this paper, a *fixation* at position (i, j) extracts a vector representation of the local image at the intersection of the ith row and jth column of a pre-defined $m \times n$ grid. This extraction is performed by what we refer to as a *retinal transformation* $\mathbf{r}(\mathbf{I}, (i, j))$ on the local image. Its output is a vector $\mathbf{x} \in \mathbb{R}^L$ where L is the length of the retinal vector representation.

In Larochelle and Hinton [4], the retinal transformation consisted of a high-resolution fovea that simply copied the pixels around the fixation position, concatenated with an array of averaging hexagons, organized into a spiral and whose size would increase with their distance with the fixation's position.

In this paper, we opted for a simpler transformation. Specifically, our transformation $\mathbf{r}(\mathbf{I}, (i, j))$ extracts subpatches from the image at up to 3 decreasing resolutions, as follows. First, we extract square image patches—the fovea P_1, the parafovea P_2, and the periphery P_3—of sizes $r_1 \times r_1$, $r_2 \times r_2$ and $r_3 \times r_3$ pixels respectively, around the grid position (i, j) from the image I. Then the square patches P_1, P_2, and P_3 are subsampled by averaging every 1×1, 2×2, and 4×4 regions positioned on a grid with step size 1, 2, and 4 respectively (i.e., there is no subsampling for P_1). We denote the resulting sampled patches $\tilde{P}_1$, $\tilde{P}_2$, and $\tilde{P}_3$. Finally, the values in the subsampled patches $\tilde{P}_1$, $\tilde{P}_2$, and $\tilde{P}_3$ are concatenated together as a vector to produce the retinal vector $\mathbf{x}$.

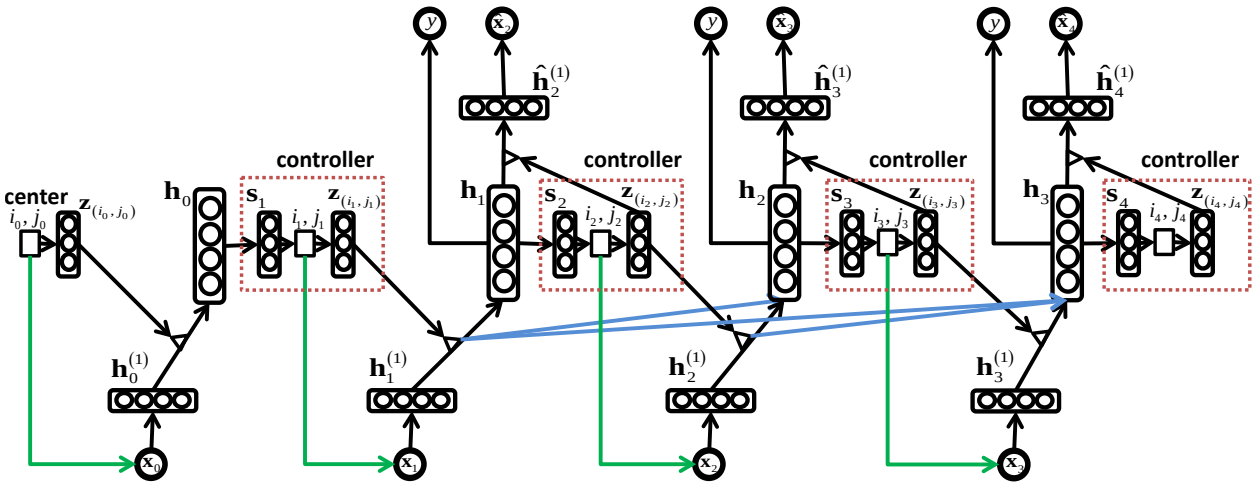

Figure 2: Flow graph of the Fixation NADE model and its controller, where $\mathbf{h}_k$ is the model's hidden representation given the previous k fixations $\mathbf{x}_{1:k}$. Third-order connections are illustrated with a small triangle. The controller is put in a dotted line square to emphasize that gradients are not backpropagated through it during training.

Thus, the retinal vectors $\mathbf{x}$ extracted by a sequence of fixations each have length L, which equals $r_1 * r_1 + r_2 * r_2/4 + r_3 * r_3/16$. Figure 1 illustrates examples of the application of the retinal transformation.

3 Fixation NADE model

We now discuss the proposed Fixation NADE model, i.e., the neural autoregressive architecture that we propose to aggregate the fixation observations into a global image representation and perform recognition. In this section, we will assume that the positions of the fixations are known and given. Thus, we try to model the joint probability of the sequence of retinal vector observations and the label for a given image.

Specifically, let $\mathbf{x}_{1:K} = [\mathbf{x}_1, \ldots, \mathbf{x}_K]$ be the retinal vectors extracted at different positions $[(i_1, j_1), \ldots, (i_K, j_K)]$, where K is the number of fixations, and y is the label to which the image belongs. The joint probability of the fixations and label can be written as follows by the probability chain rule:

$$p(\mathbf{x}_{1:K}, y) = p(y|\mathbf{x}_{1:K}) \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{<k}) \tag{1}$$

where $\mathbf{x}_{<k}$ are the retinal vectors observed before the kth fixation, and y is the target label.

Equation 1 consists of two parts: the *unsupervised* component $\prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{<k})$ which models the joint probability $p(\mathbf{x}_1, \ldots, \mathbf{x}_K)$ of all the fixations $\mathbf{x}_1, \ldots, \mathbf{x}_K$, and the *supervised* component $p(y|\mathbf{x}_{1:K})$ that represents the probability of the target given all the fixations.

Fixation NADE models each component using a feed-forward neural autoregressive network, which computes each conditional distribution in sequence while updating its hidden layer based on the gathered observations (in our case the retinal representations $\mathbf{x}_k$). Since $\mathbf{x}_k$ is real-valued, a natural choice for the unsupervised conditionals is to use Gaussian conditional distributions. For the supervised part, we utilize a logistic regression model, as is common in feed-forward neural networks. Formally, we have:

$$p(\mathbf{x}_k|\mathbf{x}_{<k}) \sim \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{\Sigma}) \tag{2}$$

$$p(y|\mathbf{x}_{1:k}) = \text{softmax}(\mathbf{d} + \mathbf{U}\mathbf{h}_k)_y \tag{3}$$

where $\hat{\mathbf{x}}_k$ is a prediction of the mean of $\mathbf{x}_k$ made by the neural network from the previous fixations $\mathbf{x}_{<k}$, $\mathbf{\Sigma}$ is a parameter covariance matrix, $\mathbf{h}_k \in \mathbb{R}^H$ is the neural network's hidden image representation given the previous k fixations $\mathbf{x}_{1:k}$, $\mathbf{U} \in \mathbb{R}^{C \times H}$ is a connection parameter matrix, $\mathbf{d} \in \mathbb{R}^C$ is a bias vector and H is the number of hidden units.

We must now specify a parametrization for the computation of the image hidden representation $\mathbf{h}_k$, as well as the prediction of $\hat{\mathbf{x}}_k$. First, our model will allow for the preprocessing of each individual fixation using a first hidden layer $\mathbf{h}_k^{(1)}$:

$$\mathbf{h}_k^{(1)} = \mathbf{g}\left(\mathbf{e} + \mathbf{V}\mathbf{x}_k\right) \tag{4}$$

where $\mathbf{g}\left(\cdot\right)$ is an element-wise nonlinear activation function, $\mathbf{V} \in \mathbb{R}^{H^{(1)} \times L}$ is a connection parameter matrix, $\mathbf{e} \in \mathbb{R}^{H^{(1)}}$ is a bias parameter vector and $H^{(1)}$ is the number of hidden units of the preprocessing layer. Then, we follow Larochelle and Hinton [4] and use a third-order order connectivity between the preprocessed fixation and the global image representation $\mathbf{h}_k$:

$$\mathbf{h}_k = \mathbf{g}\left(\mathbf{c} + \frac{1}{k}\sum_{k' \leq k} \mathbf{W}^{(i_{k'}, j_{k'})}\mathbf{h}_{k'}^{(1)}\right) \tag{5}$$

where each $\mathbf{W}^{(i,j)} \in \mathbb{R}^{H \times H^{(1)}}$ is a connection parameter matrix and $\mathbf{c} \in \mathbb{R}^H$ is a bias vector. The use of position-dependent connection matrices $\mathbf{W}^{(i,j)}$ implies that, for fixed values of the retinal representations $\mathbf{x}_k$, the hidden representation will still vary depending on where the fixations were made. Spatial information is thus incorporated into our model through these position-dependent connections.

Hence, $\mathbf{h}_k$ averages the contribution from all k fixations observed so far and can then be used to predict the next fixation $\mathbf{x}_{k+1}$. To perform this prediction, we "unroll" the computations, going from the global image's hidden representation to the next fixation's hidden representation and then to a mean prediction of the next fixation:

$$\hat{\mathbf{h}}_{k+1}^{(1)} = \mathbf{g}\left(\hat{\mathbf{e}} + \mathbf{W}^{(i_{k+1}, j_{k+1})^{\mathrm{T}}}\mathbf{h}_k\right) \tag{6}$$

$$\hat{\mathbf{x}}_{k+1} = \mathbf{b} + \mathbf{V}^{\mathrm{T}}\hat{\mathbf{h}}_{k+1}^{(1)} \tag{7}$$

where $\hat{\mathbf{e}} \in \mathbb{R}^{H^{(1)}}$ and $\mathbf{b} \in \mathbb{R}^L$ are bias parameter vectors. Notice that we use the transpose of the connection matrices $\mathbf{V}$ and $\mathbf{W}^{(i_{k+1}, j_{k+1})}$ to perform this prediction, but that predicting the next $(k+1)^{\mathrm{th}}$ fixation implies that we are using $\mathbf{W}^{(i_{k+1}, j_{k+1})}$ instead of $\mathbf{W}^{(i_k, j_k)}$. An illustration of Fixation NADE is shown in Figure 2, where $\mathbf{s}_k$ is a summary vector used by the controller to predict the next fixation position based on information from previous fixations (See Section 4 for more details).

For a given $m \times n$ grid as the possible fixation positions, all $\mathbf{W}^{(i_k, j_k)}$ matrices contain $mnHL$ parameters, which is too many to learn. As in Larochelle and Hinton [4], to reduce the number of parameters, we factorize the $\mathbf{W}^{(i_k, j_k)}$ matrices as follows:

$$\mathbf{W}^{(i_k, j_k)} = \mathbf{P}\mathrm{diag}\left(\mathbf{z}\left(i_k, j_k\right)\right)\mathbf{F} \tag{8}$$

where $\mathbf{P} \in \mathbb{R}^{H \times D}$, $\mathbf{F} \in \mathbb{R}^{D \times L}$ are two matrices that are position-independent, $\mathbf{z}\left(i_k, j_k\right) \in \mathbb{R}^D$ is a (learned) vector associated to position (i_k, j_k), $\mathrm{diag}\left(\mathbf{a}\right)$ is a diagonal matrix with vector $\mathbf{a}$ as the diagonal and D is an integer. Hence, $\mathbf{W}^{(i_k, j_k)}$ is now an outer product of D lower-dimensional bases in $\mathbf{P}$ ("pooling") and $\mathbf{F}$ ("filtering"), gated by a position specific vector $\mathbf{z}\left(i_k, j_k\right)$. Instead of learning a separate connection matrix $\mathbf{W}^{(i_k, j_k)}$ for each possible position, we now only need to learn a vector $\mathbf{z}\left(i_k, j_k\right)$ for each position plus two matrices $\mathbf{P}$ and $\mathbf{F}$ for the whole image. Hence, the total number of parameters for the position-dependent matrices $\mathbf{W}^{(i_k, j_k)}$ is reduced to $mnD + HD + DL$, which is much smaller than $mnHL$. The same factorization was used by [4]. Note that $\mathbf{z}\left(i_k, j_k\right)$ is a deterministic function of position (i_k, j_k) and is trained by gradient descent with backpropagation. In practice, we force the elements in $\mathbf{z}\left(i_k, j_k\right)$ to be in $[0, 1]$[1]

[1]This is done by setting $\mathbf{z}\left(i_k, j_k\right) = \mathrm{sigmoid}\left(\bar{\mathbf{z}}\left(i_k, j_k\right)\right)$, and learning the unconstrained $\bar{\mathbf{z}}\left(i_k, j_k\right)$ vectors instead. We also use a learning rate 100 times larger than learning the other parameters.

3.1 Learning the Fixation NADE model

Given a training set $\mathcal{D} = \{(\mathbf{x}^t_{1:K}, y^t)\}_{t=1}^N$, training Fixation NADE could be performed by minimizing the negative log-likelihood of the fixation sequence and the label y, averaged over all training images. This would imply having to predict the first fixation $\mathbf{x}_1$ from nothing, which can only result in learning the mean value of retinal fixations. Since we do not believe that information to be particularly useful to learn, we instead consider the negative log-likelihood of the fixations and label *given* the first fixation $\mathbf{x}_1$:

$$-\log p\left(\mathbf{x}_{2:K}, y|\mathbf{x}_1\right) = -\log p\left(y|\mathbf{x}_{1:K}\right) + \sum_{k=2}^{K} -\log p(\mathbf{x}_k|\mathbf{x}_{<k}) \tag{9}$$

The first term is a purely discriminative term, while the second is an unsupervised part and can be interpreted as a regularizer, that encourages a solution which also explains the unsupervised statistical relationship between the fixations.

Using Equation 2, we can replace the second term in Equation 9 and obtain

$$-\log p\left(\mathbf{x}_{2:K}, y|\mathbf{x}_1\right) = -\log p\left(y|\mathbf{x}_{1:K}\right) + \alpha \sum_{k=2}^{K} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right) \tag{10}$$

where $\alpha = \frac{1}{2\sqrt{(2\pi)^L |\mathbf{\Sigma}|}}$ is the coefficient of the multi-normal distribution.

The second term in Equation 9 is actually the squared Mahalanobis distance between $\mathbf{x}_k$ and $\hat{\mathbf{x}}_k$ weighted by α. Training the Fixation NADE model can hence be interpreted as learning an image representation that is discriminative, but that equally maintains information about the fixated regions.

In practice, we can use a hyper-parameter λ to weight the relative importance of the two terms in Equation 10, which has been observed in [14] to improve the performance of such probabilistic hidden representation models. We then obtain an hybrid cost function for Fixation NADE that can be written as

$$\mathcal{C}_{\text{hybrid}} = -\log p\left(y|\mathbf{x}_{1:K}\right) + \lambda \sum_{k=2}^{K} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right) \tag{11}$$

where λ is chosen by cross-validation.

[4] also highlights that we can take advantage of the fact that each subsequence $\mathbf{x}_{1:k}$ of the sequence of fixations $\mathbf{x}_{1:K}$ is also associated with the target y. Hence, we can incorporate such a "subsequence" signal into the cost function when training Fixation NADE. This *hybrid-sequential* cost function is written as

$$\mathcal{C}_{\text{hybrid}-\text{seq}} = -\sum_{k=1}^{K} \log p\left(y|\mathbf{x}_{1:k}\right) + \lambda \sum_{k=2}^{K} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T \mathbf{\Sigma}^{-1} \left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right) \ . \tag{12}$$

Finally, in our experiments, we limited $\mathbf{\Sigma}$ to the identity matrix, yielding an unsupervised term that simply corresponds to the squared Euclidean distance reconstruction, which is commonly used by neural network autoencoder models. The hybrid sequential cost is simplified to

$$\mathcal{C}_{\text{hybrid}-\text{seq}} = -\sum_{k=1}^{K} \log p\left(y|\mathbf{x}_{1:k}\right) + \lambda \sum_{k=2}^{K} \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|_2^2 \tag{13}$$

Optimizing the average of the cost function 13 on a training set can be performed by stochastic gradient descent, with backpropagation allowing us to compute the parameter derivatives. Optimizing the hybrid-sequential cost function is only slightly more expensive than the hybrid cost and Larochelle and Hinton [4] has shown that it tends to perform better than hybrid cost, hence we use it in our experiments.

4 The controller: learning "where to look"

In Section 3, we described a model to process the information from a sequence of fixations at different (given) positions. Now we need to define a model which will determine where these fixation should be made on an image. We refer to this model as the *controller*.

Intuitively, the controller should take the information accumulated from the previous fixations as input and produce a score indicating how much we expect to benefit from each possible next fixation position. Specifically, let $\mathbf{s}_k$ be a *summary* vector that accumulates the information from the previous $k-1$ fixations, $f\left(\mathbf{s}_k, (i_k, j_k)\right)$ will be our predicted score for the benefit from having the next fixation be at position (i_k, j_k).

As in Larochelle and Hinton [4], we will attempt to train the controller so that

$$f\left(\mathbf{s}_k, (i_k, j_k)\right) \propto Benefit\left((i_k, j_k); y, \mathbf{x}_{1:(k-1)}\right) \tag{14}$$

where we note as $Benefit\left((i_k, j_k); y, \mathbf{x}_{1:(k-1)}\right)$ the benefit of performing our next fixation at position (i_k, j_k), given that the image has label y and we have gathered the retinal observations $\mathbf{x}_{1:(k-1)}$. Ideally, the benefit should reflect the model's increased ability to predict the correct label of the image. We could use as the benefit whether the model correctly classifies the image, but such a binary signal would be weak. An alternative, suggested in Larochelle and Hinton [4], is to instead use the (log) probability of the correct class according to the model. This has the advantage of distinguishing, among fixation position choices that do allow the model to correctly classify the image, which one actually allows the model to most confidently classify it correctly.

Larochelle and Hinton [4] propose the following definition for the benefit:

$$Benefit\left((i_k, j_k); y, \mathbf{x}_{1:(k-1)}\right) = \log p\left(y|\mathbf{x}_{1:k-1}, \mathbf{x}_k = \mathbf{r}\left(\mathbf{I}, (i_k, j_k)\right)\right) \tag{15}$$

where $\mathbf{r}\left(\mathbf{I}, (i_k, j_k)\right)$ is retinal transformation as defined in Section 2. We will refer to this approach as *Bene-max*. In our experiments, we also considered this alternative definition:

$$Benefit\left((i_k, j_k); y, \mathbf{x}_{1:(k-1)}\right) = \log p\left(y|\mathbf{x}_{1:k-1}, \mathbf{x}_k = \mathbf{r}\left(\mathbf{I}, (i_k, j_k)\right)\right) - \log p\left(y|\mathbf{x}_{1:k-1}\right) \tag{16}$$

which considers instead the relative increase or boost in the log-probability of the correct class. We will thus refer to this approach as *Bene-boost*.

For the summary vector $\mathbf{s}_k$, it should provide relevant information from the previous $k-1$ fixations. Larochelle and Hinton [4] propose to use the hidden representation obtained from $\mathbf{x}_{1:k-1}$ as well as a representation of the positions that have been fixated previously. The summary $\mathbf{s}_k$ can then be written as

$$\mathbf{s}_k = [\mathbf{h}_{k-1}; \mathbf{bin}_{k-1}] \tag{17}$$

where $\mathbf{bin}_{k-1}$ is a binary vector of size mn (one component for each possible fixation position on a grid of $m \times n$) such that a component is 1 if the associated position has been fixated. We refer to this summary vector as *SHP* (Summary vector of Hidden representation and Position).

Additionally, we could try to leverage the model's uncertainty about the correct label to provide cues to the controller for promising next fixations. After all, the best fixation pattern might be dependent on the underlying class of the image. For example, for emotion recognition, the fixation points for a "smiling" face might be better placed around the corner of the mouth, while an "angry" face might be better detected based on a person's eyebrows.

Hence, we propose to incorporate the model's uncertainty about the label into the summary vector as follows:

$$\mathbf{s}_k = [\mathbf{h}_{k-1}; \mathbf{bin}_{k-1}; \mathbf{p}_{k-1}] \tag{18}$$

where $\mathbf{p}_{k-1}$ is a vector of size C whose components are the probabilities that the model assigns to each class $y \in \{1, 2, \ldots, C\}$. We refer to this summary vector as *SHPL* (Summary vector of Hidden representation, Position and Label), which we will compare with *SHP* in our experiments.

For the special case of the first fixation ($k = 1$), we follow Larochelle and Hinton [4] and compute $\mathbf{s}_1$ based on a fixation at the center of the image. The information from this "initial" fixation is then "forgotten" by the Fixation NADE model and is only used to generate the first image-dependent fixation point. It is not used by the model to

accumulate information about the image. Hence, for *SHP*, $\mathbf{s}_1 = [\mathbf{h}_0; \mathbf{bin}_0]$, where $\mathbf{h}_0$ comes from the fixation at the center of the image, and $\mathbf{bin}_0$ is a vector whose components are all 0.

We use a linear model for $f(\mathbf{s}_k, (i_k, j_k))$, with separate weights for each possible position (i_k, j_k). Specifically, $f(\mathbf{s}_k, (i_k, j_k))$ is defined as

$$f(\mathbf{s}_k, (i_k, j_k)) = \mathbf{A}_{(i_k,j_k),:}\mathbf{s}_k + q_{(i_k,j_k)} \tag{19}$$

where $\mathbf{A} \in \mathbb{R}^{mn \times S}$ is the connection matrix, $\mathbf{q} \in \mathbb{R}^S$ is a bias vector and S is the length of the summary vector $\mathbf{s}$. The training cost for the controller could then be the absolute difference between the score $f(\mathbf{s}_k, (i_k, j_k))$ that the controller outputs and the benefits for the sequence of retinal features generated during training Fixation NADE:

$$\mathcal{C}_{\text{ctrl}} = \mid f(\mathbf{s}_k, (i_k, j_k)) - Benefit\left((i_k, j_k); y, \mathbf{x}_{1:(k-1)}\right) \mid . \tag{20}$$

During training, the controller is updated after every fixation based on the stochastic gradient of this training cost.

This training strategy is akin to contextual bandit learning. The controller only predicts an immediate benefit and ignores the possible future impact of early fixation decisions. While this might technically be suboptimal, we'll see that it actually works well in practice. Moreover, unlike other reinforcement learning problems such as learning to navigate in a maze, early decisions are unlikely to have a detrimental impact on the ability of the model to correctly classify the image (whereas taking a wrong turn in a maze might put us on a path that does not lead to our goal). Indeed, fixations are rarely misleading, and only provide "real" information about the image. This is not to say that the learning problem is simple however. Indeed, unlike most reinforcement learning problems, the benefit (reward) is not static here and changes as Fixation NADE trains, generating a feedback loop between Fixation NADE and the controller that makes the learning setup especially difficult.

As in reinforcement learning problems, a balance must be made between exploration and exploitation during training. In our experiments, we generate the fixations from the Boltzmann distribution

$$p_{\text{ctrl}}((i_k, j_k) \mid \mathbf{x}_{1:k-1}) \propto \exp(f(\mathbf{s}_k, (i_k, j_k))) \tag{21}$$

which ensures that all fixation points can be sampled, but those which are currently considered more useful by the controller are more likely to be chosen. As in Larochelle and Hinton [4], in order to ensure that a fixation position is never sampled twice, we impose that $p_{\text{ctrl}} = 0$ for all positions that have been sampled previously.

We note that more principled contextual bandit algorithms, with exploration policies that have certain theoretical guaranties, could potentially be employed here as well. An interesting case would be to use contextual Gaussian process bandits [15], which would provide a more comprehensive model of our uncertainty on the benefit function.

At test time however, Larochelle and Hinton [4] suggest to greedily select the positions with the highest score from the controller for each k. This allows the model to gradually cover more of the image, until a computational budget is reached. We follow the same approach here, predicting the image's label based on the selected fixations.

When computational constraints are less of an issue, restricting the model to use very few fixations might be too detrimental. Yet, an attentional mechanism could still be useful to more appropriately weight the importance of different regions in the image. For this case, we propose an alternative approach to perform classification at test time. The idea is to leverage the fact that, at training time, Fixation NADE is in fact trained to optimize its performance on multiple fixation trajectories. Since it is well known that an ensemble of classifiers often performs much better than any individual classifier, we propose to construct such an ensemble of Fixation NADE predictors by repeatedly sampling fixation trajectories using the controller, exactly as is done at training time. If repeated M times, an ensemble prediction of the label is made by predicting the most likely label according to the following aggregated conditional distribution:

$$y^\star = \arg\max_y \frac{1}{M} \sum_{m=1}^{M} p^{(m)}\left(y | \mathbf{x}_{1:K}^{(m)}\right) \tag{22}$$

where each $p^{(m)}\left(y|\mathbf{x}_{1:K}^{(m)}\right)$ is the output of Fixation NADE for one of the M sampled fixation trajectories. Importantly, computing this distribution can be made much more efficient by caching the contribution of each fixation to the pre-activation (before the non-linear activation function) of the global image hidden layer $\mathbf{h}_K$. We will evaluate this alternative strategy in our experiments.

Algorithm 1 Basic pseudocode of Fixation NADE

Input: Image $\mathbf{I}$, Target y, Unsupervised weight λ, K fixations
Output: Hybrid sequential cost $\mathcal{C}_{\mathrm{hybrid-seq}}$ in Equation 13,

$i_0, j_0 \leftarrow$ center of image $\mathbf{I}$
$\mathbf{x}_0 \leftarrow \mathbf{r}\left(\mathbf{I}, (i_0, j_0)\right)$
$\mathbf{h}_0^{(1)} \leftarrow \mathbf{g}\left(\mathbf{e} + \mathbf{V}\mathbf{x}_0\right)$
$\mathbf{h}_0 \leftarrow \mathbf{g}\left(\mathbf{c} + \mathbf{P}\mathrm{diag}\left(\mathbf{z}\left(i_0, j_0\right)\right)\mathbf{F}\mathbf{h}_0^{(1)}\right)$
$\mathbf{bin}_0 \leftarrow$ all zeros
$\mathbf{s}_1 \leftarrow [\mathbf{h}_0; \mathbf{bin}_0]$
$i_1, j_1 \leftarrow$ Controller prediction using $\mathbf{s}_1$
$\mathbf{x}_1 \leftarrow \mathbf{r}\left(\mathbf{I}, (i_1, j_1)\right)$
for k from 1 to K **do**
  $\mathbf{h}_k^{(1)} \leftarrow \mathbf{g}\left(\mathbf{e} + \mathbf{V}\mathbf{x}_k\right)$
  $\mathbf{h}_k \leftarrow \mathbf{g}\left(\mathbf{c} + \frac{1}{k}\sum_{k' \le k}\mathbf{P}\mathrm{diag}\left(\mathbf{z}\left(i_k, j_k\right)\right)\mathbf{F}\mathbf{h}_{k'}^{(1)}\right)$
  $\mathbf{bin}_k \leftarrow$ binary vector with all zeros except $(i_{1:k}, j_{1:k})$
  $\mathbf{s}_{k+1} \leftarrow [\mathbf{h}_k; \mathbf{bin}_k]$
  $i_{k+1}, j_{k+1} \leftarrow$ Controller prediction using $\mathbf{s}_{k+1}$
  $\widehat{\mathbf{h}}_{k+1}^{(1)} \leftarrow \mathbf{g}\left(\widehat{\mathbf{e}} + \mathbf{F}^{\mathrm{T}}\mathrm{diag}\left(\mathbf{z}\left(i_{k+1}, j_{k+1}\right)\right)\mathbf{P}^{\mathrm{T}}\mathbf{h}_k\right)$
  $\widehat{\mathbf{x}}_{k+1} \leftarrow \mathbf{b} + \mathbf{V}^{\mathrm{T}}\widehat{\mathbf{h}}_{k+1}^{(1)}$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{r}\left(\mathbf{I}, (i_{k+1}, j_{k+1})\right)$
  $p\left(y|\mathbf{x}_{1:k}\right) \leftarrow \mathrm{softmax}\left(\mathbf{d} + \mathbf{U}\mathbf{h}_k\right)_y$
end for
$\mathcal{C}_{\mathrm{hybrid-seq}} = -\sum_{k=1}^{K}\log p\left(y|\mathbf{x}_{1:k}\right) + \lambda\sum_{k=2}^{K}\left\|\mathbf{x}_k - \widehat{\mathbf{x}}_k\right\|_2^2$

5 Related Work

Exploring attentional mechanisms that mimic the human visual system is an active area in computer vision, artificial intelligence, and machine learning. A common distinction made in this literature is between *saliency*-based models and *control*-based models. Most of the work belonging to saliency-based models concerns itself more with producing a saliency map for an image, the values of the map indicating the marginal probability that the corresponding pixels will be fixated by humans. For example, Itti et al. [16] present a visual attention system which combines multiscale bottom-up image features into a single topographical saliency map. Judd et al. [17] propose to train a saliency map model based on low, mid and high-level image features, which use both bottom-up and top-down information. There is also some work on obtaining saliency maps based on visual contrast, such as [18, 19]. The saliency map can then be used as the probability map to sample the fixations for recognition tasks. For example, Kanan and Cottrell [20] propose a method of obtaining a saliency map from a natural image model, from which fixation positions that are likely to be useful may be sampled. Then, a non-parametric density estimator is used for recognition. However, the fixations in [20] are sampled independently according to the saliency map, and thus cannot model how people move their eyes according to previous fixations. Also, while sampling according to the saliency map might be interpreted as a way to avoid fixating everywhere, extracting the saliency usually requires densely distributed computations across the image, which in turn are akin to fixating everywhere.

In contrast, the control-based models, as we pursue in this paper, try to model eye movements and how information obtained from previous fixations can be used to influence decisions on future fixations. For example, Schmidhuber and Huber [21] proposed to use a controller to solve the problem of locating a pre-defined 2D object in a pixel plane. More recently, Bazzani et al. [22], Denil et al. [23] adopted the Fixation RBM model to perform tracking using a fixation-based system. Najemnik and Geisler [24] also investigated what is the optimal eye movement strategy for a foveated visual system dealing with the problem of locating a pre-defined target in a cluttered environment, and whether humans

follow optimal eye movement strategies when searching a target. The model is also based on a greedy policy and a sustained posterior on the position of the target. There is also some work that models attention mechanisms with state space models. For example, Erez et al. [25] use continuous-state Partially-Observable Markov Decision Processes (POMDP) to learn to coordinate eye saccades and hand movements. Butko and Movellan [26] also used a POMDP formulation for modeling attention and proposed a controller based on an Infomax objective. Fixations are made so as to reduce the entropy of the belief state. Unlike in [25], the state of the POMDP in [26] is discrete.

While most of the control-based models focus on how to model eye movements to detect a target, less work has been proposed to model how to do a recognition task by a sequence of fixations, as we do here. Even less work has considered tackling this problem jointly, i.e., learn simultaneously a *what* and *where* component.

6 Experiments

In this section, we evaluate the success of Fixation NADE in learning *what* and *where* components jointly on the same expression recognition task used in [4] and compare it directly to the Fixation RBM model. The dataset [27] consists of 4178 aligned face images with 7 different expressions, including anger, fear, happiness, sadness, surprise and neutral. The size of the images is 100×100 pixels. We used 5 different splits of the data, such that all images belonging to the same person can only be used for either training, validation or testing. The values of the pixels are scaled to the $[-0.5, 0.5]$ interval, as in Larochelle and Hinton [4].

In our experiments, we used the rectified linear function [28] as the activation function

$$\mathbf{g}(\mathbf{a}) = \max(0, \mathbf{a}) = [\max(0, a_1), \ldots, \max(0, a_H)] \tag{23}$$

which performs better and learns faster than other (sigmoid, tanh) activation functions for our model. Nair and Hinton [28] also showed that the rectified linear function works well for image data. Given an image, the possible fixation points are arranged every 10 pixels on a 7×7 grid, with the top-left position at pixel $(20, 20)$. The number of fixations for each image is set to $K = 6$. In practice, we pretrained the model by increasing the number of fixations from 1 to 5 gradually for better initialization. The weight λ for the unsupervised part, the learning rate $lr_{\text{cost}}$for the hybrid-seq cost in 13 and the learning rate lr_{ctrl} were chosen by cross-validation. During training, we multiply the learning rate lr_{cost} and lr_{ctrl} by a decreasing factor ϵ, where $\epsilon = \frac{1}{1+n\times 10^{-6}}$ and n is the number of updates. At test time, the predicted target $y^\star$ is obtained by

$$y^\star = \arg\max_y p\left(y|\mathbf{x}_{1:K}\right) \tag{24}$$

where $p\left(y|\mathbf{x}_{1:K}\right)$ is the probability of target y given all the fixations $\mathbf{x}_{1:K}$. The performance of the model is then judged by its classification accuracy.

The experiments are organized as follows. We first compare our model with the Fixation RBM in Section 6.1. Then the impact of the hidden layer configuration is evaluated in Section 6.2. In Section 6.3, we investigate the impact of the retinal transformation configuration and, in Section 6.4, we evaluate the importance of the unsupervised part of Fixation NADE. Next, we focus on the controller module of Fixation NADE. A comparison of different variations on the controller is presented in Section 6.5. Variations on the learned fixation strategy for two different tasks based on the same images is then presented in Section 6.6, where we train Fixation NADE for gender classification instead of expression classification. Finally, we compare the greedy selection of fixation points with the use of an on-the-fly generated ensemble of Fixation NADE models in Section 6.7.

6.1 Comparison with Fixation RBM

To compare with the Fixation RBM, we used the same hyper-parameter configuration for Fixation NADE as described in Larochelle and Hinton [4]. Specifically, we set $H = 250$, $D = 250$, and the retinal transformation was set to cover about 2000 pixels[2] (with all pixels coming from the fovea, as in Larochelle and Hinton [4]). The global hidden layer size H as well as the preprocessing hidden layer $H^{(1)}$ were set to a size of 250. We used the *Bene-max* controller and the *SHP* summary vector, again as in Larochelle and Hinton [4].

[2]The retinal transformation covered a patch of 44×44 pixels, without using a lower resolution periphery. Hence, the total number of pixels is 1936.

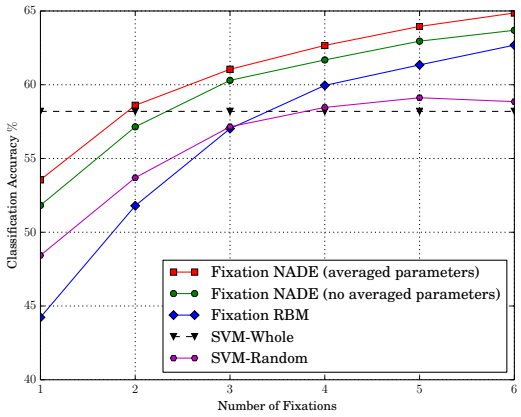

Figure 3: Comparison of Fixation NADE with Fixation RBM and a kernel SVM.

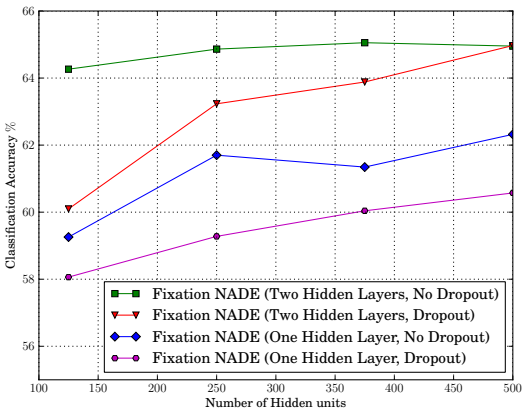

Figure 4: Comparison between different hidden layer architectures.

The results of the 5-fold experiment are shown in Figure 3, where we report the performance of different models with a varying number of fixations. The SVM-Whole corresponds to an RBF kernel SVM classifier trained on the full image, taking raw pixels as inputs, while the SVM-Random uses a random (but fixed) sequence of fixation positions and concatenates the extracted retinal representations to form the input to an RBF kernel SVM. We can see that Fixation NADE always outperforms Fixation RBM. With only 2 fixations, Fixation NADE even reaches a performance that is comparable with the SVM-Whole, that takes the whole image as input. In contrast, the number of fixations that allows Fixation RBM to perform as well as the SVM is 3. What's more, we can see that the improvement of Fixation NADE over Fixation RBM is larger when the number of fixations is small.

In [4], an exponentially decaying average of the parameters values visited during gradient descent was actually used at test time, to stabilize training and potentially prevent overfitting. This is similar to Polyak averaging, but where the uniform average is replaced by a weighting that puts more emphasis on recent parameter values. We considered using this procedure as well for the Fixation NADE and report its performance in Figure 3. We can see that Fixation NADE with averaged parameters performs even better.

Having established that Fixation NADE is competitive for learning an attention-based recognition model, the next sections will investigate many variations on this basic configuration. In particular, from now on, we will always compare with the performance of the Fixation NADE model with averaged parameters.

6.2 Deep vs. shallow Fixation NADE

In this part, we show the impact of the hidden layer architecture. There are two ways to change the hidden layer configuration. One is to vary the number of hidden units H. The other is to not use a preprocessing hidden layer, i.e., use a shallow Fixation NADE. We test the performance for $H \in \{125, 250, 375, 500\}$, with/without the preprocessing hidden layer. When using a preprocessing hidden layer, its size $H^{(1)}$ is set identical to H. We also considered the use of dropout [29] as a way to prevent overfitting. We used a dropout rate of 0.5.

The comparison is shown in Figure 4. The most noticeable observation is that using a deep Fixation NADE architecture is beneficial, with the shallow Fixation NADE model performing worse, even with dropout regularization. We also see a trend where, as the number of hidden units is increased, dropout becomes increasingly useful. While we do not observe a benefit from using dropout for the range of hidden layer sizes considered in this experiment, we suspect that, for larger hidden layers, dropout will start yielding superior performance.

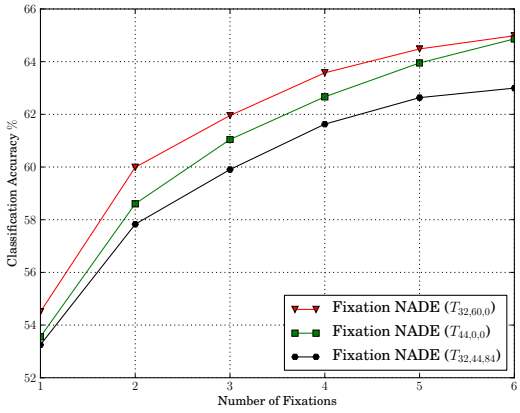

Figure 5: Comparison of the use of different retinal transformation configurations.

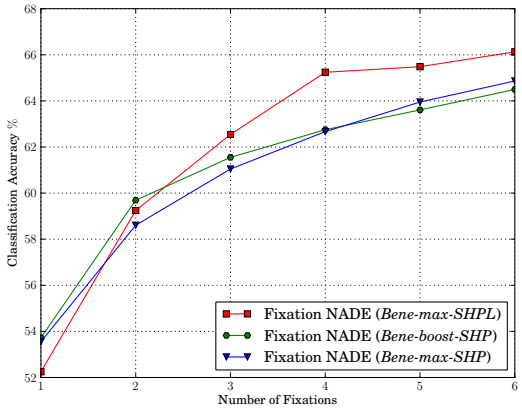

Figure 6: Comparison between different variations on the controller model.

6.3 Impact of the retinal configuration

The retinal transformation $\mathbf{r}\left(\mathbf{I}, (i_k, j_k)\right)$ plays an important role in how information is acquired from each fixation. The configuration of the retinal transformation could thus influence the performance of the model. In [4], a retinal transformation which only extracted the high resolution fovea was used for the emotion recognition data set, resulting in vectors $\mathbf{x}_k$ with a size of about 2000. In Section 6.1, we used a similar retinal transformation configuration for a fair comparison.

Here, we show the influence of the retinal configuration on the performance. Specifically, we keep the size of $\mathbf{x}_k$ to be around 2000, but try more combinations with different resolutions, as described in Section 2.1. Figure 5 shows the performance of these different choices, with a varying number of fixations. The "$T_{44,0,0}$" configuration corresponds to $\mathbf{x}_k$ that is extracted only at the fovea region centred at position (i_k, j_k), with the size of the fovea at 44×44 pixels (as in Section 6.1). The "$T_{32,60,0}$" configuration means that $\mathbf{x}_k$ is extracted at both the fovea and parafovea, with sizes 32×32 and 60×60 respectively. Hence, the length of the resulting retinal vector is $32 \times 32 + (60 \times 60)\,/4 = 1924$. Similarly, "$T_{32,44,84}$" means that $\mathbf{x}_k$ is extracted from all three regions, with sizes 32×32, 44×44 and 84×84 respectively. Thus, the length of the retinal vector with this configuration is 1949.

We can see from Figure 5 that $T_{32,60,0}$ always performs best among the 3 different retinal transformation configurations, with $T_{32,44,84}$ always performing worst. This confirms that there is a benefit in balancing between the collection of fine details in a small visual region and of coarser information on a larger region.

6.4 Impact of unsupervised learning

The unsupervised part in Equation 13 serves as a regularizer that encourages the model to learn the statistical relationship between fixations. Here, we compare its use with two alternatives. The first is to not use any regularization, i.e., set the regularization weight λ to 0. This will allow us to measure the contribution of unsupervised regularization.

The second alternative will investigate whether a more standard reconstruction term, that is not predictive and instead aims at encouraging the hidden layer to maintain information about the latest fixation, can be useful. Such a regularization term is closer in spirit to autoencoder-based regularization, e.g., when using different types of autoencoders to pretrain a neural network [30, 31].

Specifically, a reconstruction $\tilde{\mathbf{x}}_k$ for the kth fixation $\mathbf{x}_k$ is computed and trained according to the hidden representation $\mathbf{h}_k$. The reconstruction error $\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|_2^2$ is then used as an additional unsupervised regularizer, as follows:

$$\mathcal{C} = -\sum_{k=1}^{K} \log p\left(y|\mathbf{x}_{1:k}\right) + \lambda_1 \sum_{k=2}^{K} \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|_2^2 + \lambda_2 \sum_{k=1}^{K} \|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|_2^2 \tag{25}$$

Table 1: Performance comparison of different choices of unsupervised regularization. *Fixation NADE (reconstruction)* corresponds to the training objective of Equation 25.

Model	Accuracy
Fixation NADE (λ varies)†	$\mathbf{64.86}\%$
Fixation NADE ($\lambda = 0$)	63.92%
Fixation NADE (reconstruction)	63.95%

†: "λ varies" means the λ is a hyperparameter and is chosen on a validation set by cross-validation.

The performance of these different alternatives is provided in Table 1. We observe that regularization based on a predictive fixation reconstruction contributes to the performance of the model. However, regular (non-predictive) reconstruction brings no improvement to the performance.

6.5 Impact of variations on the controller

The role of the controller is to predict the next fixation point that would increase the most the probability of the true target, given all the fixations so far. In Section 4, we provide different ways of dealing with the benefit function and summary vector used by the controller. Here we evaluate the performance of these different choices, comparing them to the use of *Bene-max* for the benefit function and *SHP* for the summary vector (which we refer to as *Bene-max-SHP*).

We compare two alternatives. The first alternative is to use *Bene-boost* for the benefit function while keeping the *SHP* summary vectors (*Bene-boost-SHP*). The second is to use the *SHPL* summary vectors while sticking to the *Bene-max* benefit function (*Bene-max-SHPL*). This experiment will thus measure the potential marginal contribution of these changes to the model's performance.

From Figure 6 we can see that using the extended summary vectors significantly improves the performance of Fixation NADE, which confirms that cues on the uncertainty of the model about the correct label are really helpful in predicting good fixation positions. However, using *Bene-boost* as the benenfit function brings little gain to the performance.

Another important component of the controller learning algorithm is how it deals with the exploration-exploitation tradeoff, i.e., how frequently it performs fixations that are not considered the next best fixation by the model. We propose to investigate the importance of balancing the exploration and exploitation strategy, by explicitly defining an explorer rate γ, corresponding to the probability of selecting the fixations for a given image by sampling from the Boltzmann distribution of Equation 21 during training. Otherwise, the best fixation (as during test time) is chosen. Hence, $\gamma = 1.0$ corresponds to the strategy proposed in [4] and used so far. We tested values of $\gamma \in \{0, 0.25, 0.5, 0.75, 1.0\}$ and present the results in Figure 7.

We observe that high exploration is important, and there does not appear to be any advantage in tuning the exploration parameter further. Interestingly, exploration is particularly important when using the test-time ensemble strategy of Equation 22, with catastrophic performance when using no exploration (see Section 6.7 for more results on using the ensemble approach).

6.6 Learned fixation strategies for different tasks

Recently, Mathe and Sminchisescu [10] showed that the choice of fixation points is highly dependent on the tasks being solved, even for the same given image. Here, we try to examine whether this behavior is reflected by our model in the fixation strategy it learns for different tasks. Specifically, we compare learned fixation strategies for two classification problems: the original emotion recognition problem and a new gender classification problem. For gender classification, we manually labelled gender information for the data set of Susskind et al. [27]. Thus, the data in the gender classification experiment are exactly the same, except for labels that are changed from expressions to gender (male/female). We also adopt the configuration of Section 6.1 for Fixation NADE.

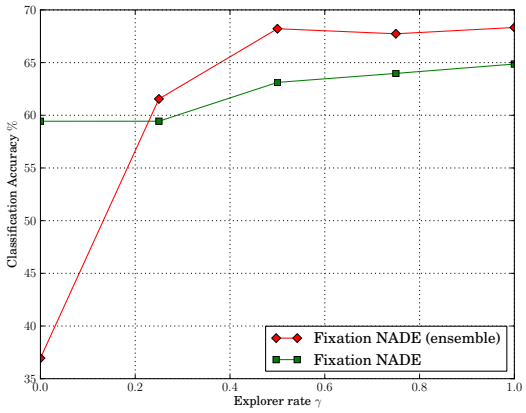

Figure 7: Comparison between using different rates of exploration by the controller.

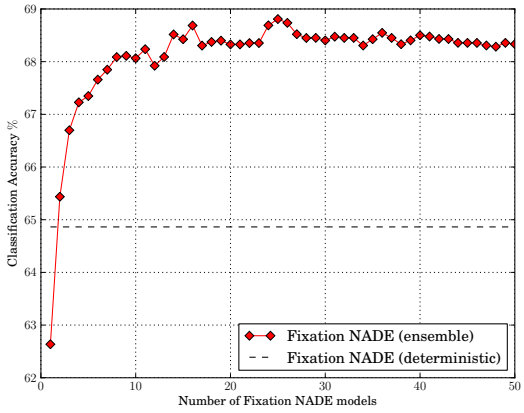

Figure 8: Performance of an ensemble of Fixation NADE models generated from sampled fixation trajectories, for a varying number of models.

The experiment was performed as follows. We trained two Fixation NADE models, one on the facial expression recognition training set and another on the gender recognition training set (which share the same images but use different labels). We then ran both models on their respective test sets, counted the number of times that each model fixated at any of the possible positions and normalized the numbers so that they summed to 1. These numbers are referred to as a *hitmap*.

Figure 9 illustrates the hitmaps for both tasks. We can see that Fixation NADE for facial expression recognition prefers fixations at the mouth or edge of the mouth, the eyes and the nose, while the model for gender recognition concentrates primarily on the eyebrows and eyes. This confirms that the model is indeed able to learn different fixation strategies depending on the task.

6.7 Fixation NADE ensembles over different fixation trajectories

We proposed in Section 4 an alternative to greedily selecting the best fixation sequence, which is appropriate when the computational budget allows for potentially fixating everywhere, while we'd still want fixations to concentrate mainly on relevant regions of the image. This can be achieved by sampling, on the fly, an ensemble of Fixation NADE models, where each performs a prediction based on a fixation sequence sampled by the controller. Classification is then performed by the ensemble, using Equation 22.

In Figure 8, we compare the performance of ensembles of varying sizes with the basic Fixation NADE model of Section 6.1 that uses a single fixation sequence. We observe that averaging just 3 models already provides a substantial boost over the single model. With about 20 models, the ensemble then reaches its asymptotic performance, corresponding to an impressive 3% improvement in accuracy.

An interesting connection with dropout can also be made here. One could view this approach as a form of dropout where whole fixations are being dropped out and an explicit ensemble is constructed at test time, instead of simply multiplying units by a dropout probability. Another distinction of course is that the dropout rates are determined by the controller and, because of the sequential nature of the selection, fixations are not dropped out independently.

7 Discussion and Conclusion

In this paper, we proposed Fixation NADE, a feed-forward, autoregressive architecture that models the recognition process as a sequence of task-specific fixations, extracted at different locations by a fixation policy, and showed how to train Fixation NADE to learn *what* and *where* components jointly. While most of the work on visual recognition

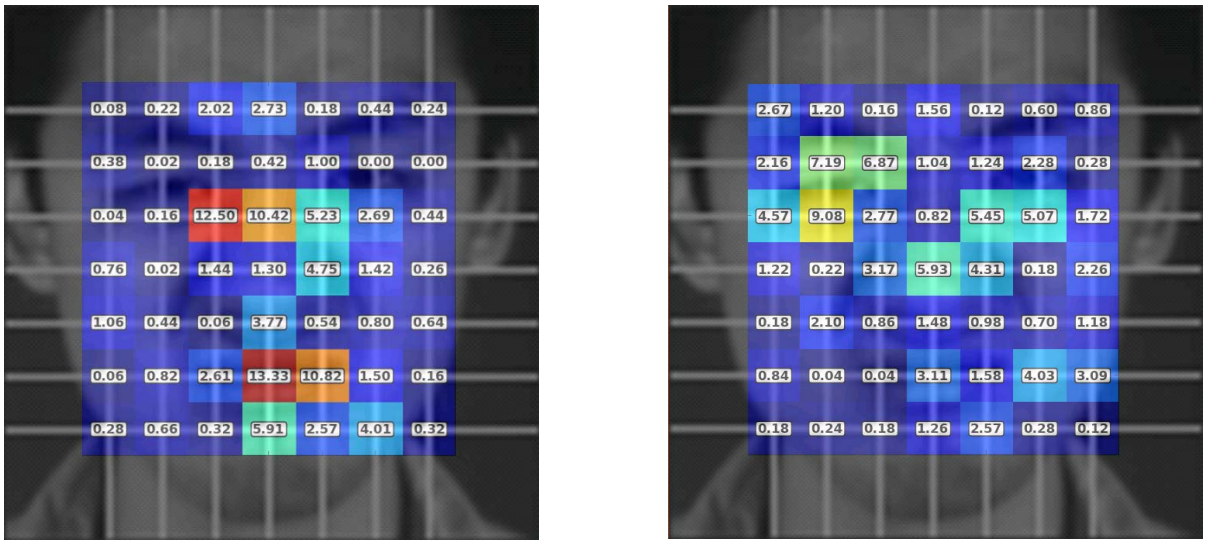

Figure 9: The fixation strategies for facial expression recognition (left) and gender recognition (right) for the same test set images. The numbers correspond to the normalized counts of fixations at the different positions (the crossing point of the white lines on the face image) across all of the test set. The selected face image on which the hitmap is displayed was selected randomly, only to better visualize what information each fixation position will typically provide.

extracts features densely over the whole image, Fixation NADE is inspired by the human visual system and learns both what local features to extract and where to extract them with a good fixation policy.

In our experiments, we showed that Fixation NADE performs better than the Fixation RBM, which inspired our model. We also showed that a deep version of Fixation NADE outperforms a shallow architecture, and that the fixation policies learned by the model varies depending on the task at hand, as for human fixation strategies. Furthermore, we investigated many variants of the model, in order to identify good modeling choices in attention-based recognition. One particularly successful strategy is to generate many fixation paths at test time and construct ensembles of Fixation NADE models from them. We thus hope that this work is one of many future successful steps in the direction of fully adaptable attention-based recognition systems.

As for future work, we wish to tackle some of the current weaknesses of our proposed model, which correspond, as far as we know, to challenges in attention-based recognition that have yet to be perfectly addressed.

First, the number of fixations is currently predefined and fixed. However, it would be interesting and computationally beneficial to design a stopping criteria for the fixation process, which would be adapted to each image. This would allow the model to stop fixating after fewer fixations if the image was considered ”easy” by the model (e.g. when the model is confident enough).

Second, moving to images with more variations is of course a very interesting and crucial direction for attention-based recognition. Being able to handle the detection of objects which are not necessarily centred in the image would be an important first step. Such a model would effectively have to perform a form of joint detection and recognition. Achieving such global invariance to the translation of objects will require more sophisticated attention and aggregation mechanisms. Moreover, our model also isn't invariant to local transformations at the level of individual receptive fields. This could possibly be dealt with by incorporating in the architecture convolutional units, a research direction we'd like to explore.

References

[1] Svetlana Lazebnik, Cordelia , and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[2] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012.

[4] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In *Advances in neural information processing systems*, pages 1243–1251, 2010.

[5] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. *Artificial Intelligence and Statistics (AISTATS)*, 15:29–37, 2011.

[6] Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems 26*, pages 2175–2183. 2013.

[7] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems 25*, pages 2717–2725. 2012.

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] Stefan Mathe and Cristian Sminchisescu. Action from still image dataset and inverse optimal control to learn task specific visual scanpaths. In *Advances in Neural Information Processing Systems*, pages 1923–1931, 2013.

[11] JPC Southall. Helmholtzs treatise on physiological optics. vol. 2: The sensation of vision, trans. jpc southall.(translated from the third german edition), 1962.

[12] Arash Fazl, Stephen Grossberg, and Ennio Mingolla. View-invariant object category learning, recognition, and search: How spatial and object attention are coordinated using surface-based attentional shrouds. *Cognitive psychology*, 58(1):1–48, 2009.

[13] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[14] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008.

[15] Andreas Krause and Cheng Soon Ong. Contextual gaussian process bandit optimization. In *NIPS*, pages 2447–2455, 2011.

[16] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998.

[17] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[18] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012.

[19] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 409–416. IEEE, 2011.

[20] Christopher Kanan and Garrison Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *CVPR*, 2010.

[21] Juergen Schmidhuber and Rudolf Huber. Learning to generate artificial fovea trajectories for target detection. *International Journal of Neural Systems*, 2(01n02):125–134, 1991.

[22] Loris Bazzani, Nando Freitas, Hugo Larochelle, Vittorio Murino, and Jo-Anne Ting. Learning attentional policies for tracking and recognition in video with deep networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, pages 937–944. ACM, 2011.

[23] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural Computation*, 24(8):2151–2184, 2012.

[24] Jiri Najemnik and Wilson S Geisler. Optimal eye movement strategies in visual search. *Nature*, 434(7031): 387–391, 2005.

[25] Tom Erez, Julian J Tramper, William D Smart, and Stan CAM Gielen. A pomdp model of eye-hand coordination. In *AAAI*, 2011.

[26] Nicholas J Butko and Javier R Movellan. Infomax control of eye movements. *Autonomous Mental Development, IEEE Transactions on*, 2(2):91–107, 2010.

[27] Josh M Susskind, Adam K Anderson, and Geoffrey E Hinton. The toronto face database. *Department of Computer Science, University of Toronto, Toronto, ON, Canada, Tech. Rep*, 2010.

[28] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[29] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 1096–1103. ACM, 2008.

[31] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML 2011)*, 2011.