Recognizing Faces by Dynamic Link Matching

Laurenz Wiskott and Christoph von der Malsburg†

Ruhr-Universität Bochum
Institut für Neuroinformatik
D-44780 Bochum, Germany
WWW: http://www.neuroinformatik.ruhr-uni-bochum.de
E-mail: laurenz@neuroinformatik.ruhr-uni-bochum.de

Abstract

Dynamic Link Matching is a neural dynamics for translation invariant object recognition that is robust against distortion. We here demonstrate human face recognition against a gallery of 112 neutral frontal view faces. Probe images are distorted due to rotation in depth and changing facial expression. Probe images and gallery models are represented by layers of neurons interpreted as labeled graphs. Nodes are labeled by local features based on the Gabor wavelet transform. Probe images are matched to the gallery of face models by Dynamic Link Matching. Concurrently with the matching process a simple winner-take-all mechanism identifies the correct model. A dynamic window of attention restricts the match to the part of the image occupied by the face.

1 Introduction

In a large class of neural networks, position invariance is achieved by extracting features and discarding positional information for each feature individually. This has a number of serious drawbacks. The necessary loss of information also about relative position of features introduces ambiguity, unless a system of high-level interlocking features is found (a goal which can in general only be achieved with extensive training and relative to a specific class of objects). The loss of information on feature position makes it also impossible to refer back to their positions in the image, which is required, for instance, for motion planning.

Dynamic Link Matching [1, 4, 5] overcomes these draw-backs by creating a set of one-to-one correspondences between image and model during the matching process. This is achieved with the help of temporal binding and fast synaptic plasticity. We show here that a mapping can be set up by rapid self-organization during single fixations. Without requiring a training period, DLM provides invariance against translation, mirror-reflection and rotation as well as robustness against image distortion. The performance of DLM was previously demonstrated for symmetry detection [1] and simple recognition tasks [2]. We here extend the model to an autonomous object recognition system, able to deal with galleries of more than one hundred models.

*This work has been funded by grants from the German Federal Ministry of Science and Technology (413-5839-01 IN 301 B/9) and from AFOSR (F49620-93-1-0109).
†also Dept. of Computer Science and Section for Neurobiology, University of Southern California, Los Angeles, CA 90089
2 The Principle of Dynamic Link Matching

In Dynamic Link Matching, see Figure 1, image and all models are represented by rectangular layers of neurons. These are labeled by jets as local features (jets are vectors of Gabor wavelet components [3]), symbolized here by different textures. The initial connectivity is all-to-all with synaptic weights depending on the similarities between local features, indicated by arrows of different width. In each layer, neural activity dynamics generates one small moving blob of activity (the blob can be interpreted as covert attention scanning the image or model). If a model is similar in feature distribution to the image, its initial connectivity matrix contains a strong regular component, connecting corresponding points (which by definition have high feature similarity), plus noise in the form of accidental similarities. Hence the blobs in the image and that model tend to align and synchronize in the sense of simultaneously activating, and thus generating correlations, between corresponding regions. These correlations are used, in a process of rapid reversible synaptic plasticity, to restructure the connectivity matrix. The mapping implicit in the signal correlations is more regularly structured than the connectivity itself, and correlation-controlled plasticity thus improves the connectivity matrix. Iteration of this game rapidly leads to a neighborhood preserving one-to-one mapping connecting neurons with similar features.

For recognition purposes, DLM has to be applied in parallel to many models. The best fitting model, i.e., the model most similar to the image, will finally have the strongest connections to the image and will have attracted the greatest share of blob activity. A simple integrating winner-take-all mechanism detects the correct model.

3 System Dynamics

System dynamics is formulated as a set of coupled differential equations, see Table 1. The different functional parts are now going to be explained step by step.

**Blob formation**: The formation of a stationary blob on a layer of neurons is achieved by local excitation and global inhibition. Local excitation (in the form of a Gaussian) generates clusters of activity and global inhibition lets the clusters compete against each other. The strongest one will finally suppress all others and grow to an equilibrium size, determined by the strengths of excitation and inhibition.
Running blob: Making the blob running is achieved by delayed self-inhibition $s$ to repel the blob from regions where it has recently been. It is realized by a leaky integrator with time constant $\mu_\pm$. The blob actually generates a tail of self-inhibition as a memory where not to go. The speed with which self-inhibition is built up and the memory span keeping it away from where it was are controlled by parameters $\mu_+$ and $\mu_-$, respectively.

Layer interaction: The trajectory of the running blob is guided by excitatory input from another layer, coupled through connection matrix $W$. If two layers of the same size are connected by the identity matrix the input to one is a copy of the blob in the other layer. This coupling would stabilize in-step motion of the two blobs. This “synchronization principle” prevails also when the connection matrix is not perfect as when generated by real image data.

Link dynamics: Once the dynamics of the layers are coordinated and in line with the connection matrix, the induced correlations $\sigma(h^p_i)\sigma(h^q_j)$ between the neurons are a good cue to further clean up and structure the connectivity. The link dynamics typically consists of a growth rule and a normalization rule. The former lets the weights grow according to the correlation between the connected neurons in a quasi-Hebbian manner. The latter prevents the links from infinite growth and induces competition such that eventually one link per neuron survives and suppresses all others.

Attention dynamics: The alignment between the running blobs depends very much on the constraints, in particular on the size and form of the layer on which they are running. This causes a problem since image and model have different size. We have therefore introduced an attention blob $a$ which restricts the movement of the running blob on the image layer to a region of about the same size as the model layers. The running blob on the other side can shift the attention blob into a region where input is especially large and favors activity. Therefore the attention blob automatically aligns with the actual face position. The attention blob layer is initialized with a primitive segmentation cue, in this case the norm $N(J)$ of the respective jets. The model layers have the same attention blob to keep the system symmetric.

Recognition dynamics: Each model cooperates with the image depending on its similarity. The most similar model will cooperate most successfully and will be most active. Hence the total activity of the model layers indicates which is the correct one. A simple winner-take-all mechanism is applied to detect the best model and to suppress all other models.

Blob alignment in the model domain: Since faces have a common general structure it is advantageous to align the blobs in the model domain to insure that the blobs are always at the same position in the faces, all at the left eye or all at the chin, for instance. This is achieved by connections between the layers and is implemented by the term $+\sum_{\ell'} \max_{\sigma'} \left( g_{i\ell'} \sigma(h^p_{i\ell'}) \right)$, instead of merely $+\sum_{\ell'} \left( g_{i\ell'} \sigma(h^p_{i\ell'}) \right)$, in the first equation. Otherwise the image would get input from all different parts of the model faces at the same time and could not align with the correct model reliably. An alternative would be to let the models inhibit each other such that only one model can have a blob at a time. The models then would share time to match onto the image and the best fitting one would get most of the time. This would probably be the appropriate setup if the models were very different and without a common structure, as is the case for general objects. The disadvantage would be that the system needed much more time to decide which model to take, because the relative layer activities in the beginning would depend much more on accident than in the setup we have chosen here.

Maximum versus sum neurons: The model neurons used here use the maximum over all input signals instead of the sum. The reason is that the sum would mix up many different signals while only one is actually the correct one. Hence the signal to noise ratio would be very low. The maximum rule has the additional advantage that the dynamic range of the input into a single cell does not vary very much when the connectivity develops, whereas the signal sum would grow very much during synaptic re-organization and would drive cells into saturation.
Layer dynamics:

\[ \dot{h}_i^p(t) = -h_i^p + \sum_{\delta} \max(\sigma(h_i^\delta), \delta) - \beta_h \sum \sigma(h_i^p) - \alpha s_i^p \\
+ \kappa \max_{\delta j} (W_{ij}^p \sigma(h_j^p)) + \kappa a \sigma(a_i^p) - \beta_r \Theta(r - r^p) \]

Attention dynamics:

\[ \dot{a}_i^p(t) = -a_i^p + \sum_{\delta} \sigma(a_i^\delta) - \beta_a \sum \sigma(a_i^p) + \kappa a \sigma(h_i^p) \]

Link dynamics:

\[
\begin{align*}
W_{ij}^p(t_0) &= S_{ij}^p = S(J_i^p, J_j^p) \\
W_{ij}^p(t) &= \lambda W_{ij}^p \sigma(h_i^p) \sigma(h_j^p) \\
W_{ij}^p &\to W_{ij}^p \min(S_{ij}^p/W_{ij}^p, 1)
\end{align*}
\]

Recognition dynamics:

\[
\begin{align*}
\dot{r}^p(t) &= \lambda_r r^p \left( F^p - \max_{\delta' r' \delta' r'} \right) \\
F^p(t) &= \sum_i \sigma(h_i^p)
\end{align*}
\]

Table 1: DLM dynamics for face recognition system. **Variables:** \( h \): internal state (membrane potential) of neurons; \( s \): delayed self-inhibition; \( a \): attention variable; \( W \): synaptic weights between the image and model layers; \( J \): feature jet; \( r \): recognition variable; \( F \): summed activity of each layer. **Indices:** \( (p, i, j; q, q') \): layer indices, 0 indicates image layer, 1, ..., \( M \) indicate model layers; \( (p, i, j; q, q') = (0, 0, 1, ..., M; 1, ..., M) \) for image layer dynamics; \( (q, i, j; q') = (1, ..., M; 1, ..., M; 0, 0) \) for model layer dynamics; \( (i; i', j; j') \): two-dimensional indices for the individual neurons in layers \( (p; i', j; q') \), respectively. **Functions:** \( g_{\delta \delta'} = \exp(-(i - i')^2/(2\sigma_i^2)) \): Gaussian interaction kernel; \( \sigma(h) = 0 \) for \( h \leq 0 \), \( \sqrt{h/\rho} \) for \( 0 < h < \rho \); \( 1 \) for \( h \geq \rho \); nonlinear squashing function; \( \Theta(r) \): Heavyside function; \( S(J, J') \): similarity between jets \( J \) and \( J' \). **Parameters:** \( \beta_h \): strength of global inhibition; \( \beta_a \): strength of global inhibition for attention blob; \( \alpha \): strength of global inhibition compensating the attention blob; \( \beta_r \): global inhibition for model suppression; \( r_s \): threshold for model suppression; \( \alpha \): strength of self-inhibition; \( \kappa \): strength of interaction between image and model layers; \( \kappa_a \): effect of the attention blob on the running blob; \( \kappa_a \): effect of the running blob on the attention blob; \( \lambda_W \): time constant for the link dynamics; \( \lambda_r \): time constant for the recognition dynamics; \( \mu_{\pm} \): decay constant for delayed self-inhibition; \( \mu_{\perp} = \mu_{\perp} \): if \( h - s > 0 \); \( \mu_{\perp} = \mu_{\perp} \): if \( h - s < 0 \); \( \rho \): slope radius of squashing function.
4 Experiments

Data base: As a face data base we have used a gallery of 112 different persons, taken with neutral frontal views. As probes we used sets of frontal views with altered facial expression and neutral views rotated in depth. The gallery models, i.e. the neutral frontal views, are represented by layers of size 10 × 10. The x- and y-position and the grid spacings in the images were controlled manually such as to align them with each other. Grid spacing varies for different models by about a factor of 1.5. Input images of the test faces are represented by layers of size 16 × 17, with a constant spacing similar to that of the models.

Recognition example: Figure 2 shows a recognition example of a test face with a different expression. The gallery contains five models. Due to the tight connections between the models, the layer activities show the same variations and differ only little in intensity. This small difference is averaged over time and amplified by the recognition dynamics, which rules out one model after the other until only the correct one survives. The example was monitored for 2000 time units of recognition time. A phase of 1000 time units where the attention blob aligned itself with the face preceded this, but is not shown here.

Figure 2: Simulation example of DLM recognition. The test image is shown on the left with 16 × 17 neurons indicated by black dots. The models have 10 × 10 neurons and are aligned with each other. The respective total layer activities, i.e. the sum over all neurons of one model, are shown in the upper graph. The most similar model is usually slightly more active than the others. On that basis the models compete against each other, and eventually the correct one survives, as indicated by the recognition variable. The sum over all links of each connection matrix is shown in the lower graph. It gives an impression of the extent to which the matrices self-organize before the recognition decision is made.
**Results:** Recognition rates for a gallery of 112 models are given in Table 2. As is already known from previous work, recognition of depth-rotated faces is less reliable than for instance that of faces with altered expression. It is interesting to observe recognition times. Although they vary significantly, a general tendency is noticeable. Firstly the more difficult task of recognizing depth-rotated faces usually takes more time than for frontal views. Secondly incorrect recognition takes much more time than correct recognition.

<table>
<thead>
<tr>
<th>test images</th>
<th>correct recognition #</th>
<th>rate</th>
<th>recognition time for correct recognition</th>
<th>incorrect recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>109 frontal views (grimace)</td>
<td>89</td>
<td>81.2</td>
<td>700 ± 980</td>
<td>3600 ± 1700</td>
</tr>
<tr>
<td>111 rotated faces (15 degrees)</td>
<td>92</td>
<td>86.5</td>
<td>630 ± 600</td>
<td>3900 ± 3000</td>
</tr>
<tr>
<td>110 rotated faces (30 degrees)</td>
<td>73</td>
<td>66.4</td>
<td>1400 ± 1600</td>
<td>4200 ± 3600</td>
</tr>
</tbody>
</table>

Table 2: Recognition results against a gallery of 112 neutral frontal views. Differential equations were iterated twice per time unit.

5 Discussion

For the first time DLM was successfully applied to a larger recognition task, here face recognition against a gallery of 112 faces. The system shows translational invariance and robustness against distortion. Whereas it had been previously shown that DLM has the potential for invariance against rotation and mirror reflection [2], we didn’t attempt it here, our features not being invariant under these transformations. The system requires only one model per person and no training time, models just being stored and added to the gallery. On the other hand, DLM is relatively expensive in terms of (initial) connectivity and in terms of the processing time necessary for map self-organization. Both problems can be alleviated by the introduction of a coarse-to-fine hierarchy and the use of more sophisticated dynamics than simple running blobs.

References


