An Artificial Neural Network For Spatio-Temporal Bipolar Patterns: Application to phoneme classification

SE RIM PARK
# Neural Network Generalized to Dynamic Input Patterns

## Static Network

<table>
<thead>
<tr>
<th>Input</th>
<th>$\vec{x} = [x_1 \ x_2 \ \cdots \ x_L]^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation</td>
<td>$y_i, \ i = 1,2, \ldots, N$</td>
</tr>
<tr>
<td>Output</td>
<td>$z_i, \ i = 1,2, \ldots, N$</td>
</tr>
<tr>
<td>Transmittance</td>
<td>$\vec{w} = [w_{i1} \ w_{i2} \ \cdots \ w_{iL}]^T$</td>
</tr>
<tr>
<td>Node operator</td>
<td>$\eta$ where $\eta(\cdot)$ is a nonlinear memoryless transform</td>
</tr>
<tr>
<td>Neuron operation</td>
<td>$z_i = \eta(\vec{w}_i^T \vec{x})$ (2.1)</td>
</tr>
</tbody>
</table>

## Dynamic Network

<table>
<thead>
<tr>
<th>Input</th>
<th>$\vec{x}(t) = [x_1(t) \ x_2(t) \ \cdots \ x_L(t)]^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation</td>
<td>$y_i(t), \ i = 1,2, \ldots, N$</td>
</tr>
<tr>
<td>Output</td>
<td>$z_i(t), \ i = 1,2, \ldots, N$</td>
</tr>
<tr>
<td>Transfer function</td>
<td>$\vec{w}(t) = [w_{i1}(t) \ w_{i2}(t) \ \cdots \ w_{iL}(t)]^T$</td>
</tr>
<tr>
<td>Adaptation</td>
<td>$a_i(t)$</td>
</tr>
<tr>
<td>Node operator</td>
<td>$\eta$ where $\eta(\cdot)$ is a nonlinear memoryless transform</td>
</tr>
<tr>
<td>Neuron operation</td>
<td>$z_i(t) = \eta(a_i(-t) \cdot \vec{w}_i(t)^T \cdot \vec{x}(t))$ (2.2)</td>
</tr>
</tbody>
</table>
Static network vs. dynamic network

- Static network has 1-dimensional input, ‘spatial’ dimension.
- On the other hand, dynamic network has 2-dimensional input, ‘spatial’ and ‘temporal’ dimension.
- The term dynamic is used to indicate that the network incorporates ‘temporal’ information of the data.
- It has nothing to do with network changing/evolving over time. In both cases, network’s weights are fixed once training is done.
CORRELATION: MEASURE FOR SIMILARITY

http://scidavis.sourceforge.net/manual/x4199.html
If Fourier transforms exist

\[ y_i(t) = w_i(t)^T x(t) \]

It is like filtering the signal with the filter (transfer function) to find the highest matching response

\[ y(t) = a(t) \circ w(t)^T \circ x(t) \]

\[ w(f) = F\{w(t)\} \]
\[ y(f) = F\{y(t)\} \]
\[ x(f) = F\{x(t)\} \]
\[ a(f) = F\{a(t)\} \]

\[ y(f) = a(f) \{ w(f)^* \circ x(f) \} \]

*: conjugate transpose

Transfer function
Naively, a function multiplied with the input in fourier domain to create the output is called a transfer function.
**Hebbian Learning Rule in Static Network**

Input-output pair: \( \{x^{(k)}, d^{(k)}\}, \ k=1,\ldots, M \)

\[
X = [x^{(1)}, x^{(2)}, \ldots, x^{(M)}] : \text{input matrix (LxM)}
\]

\[
W = [w_1, w_2, \ldots, w_N]' : \text{weights (NxL)}
\]

\[
Y = WX = [y^{(1)}, y^{(2)}, \ldots, y^{(M)}]' : \text{activation matrix (NxM)}
\]

\[
D = [d^{(1)}, d^{(2)}, \ldots, d^{(M)}]' : \text{desired output matrix (NxM)}
\]

e.g.) If there is only one sample per one output, output is represented with unit vector, and \( a=1 \), then \( W = X \).
HEBBIAN LEARNING RULE IN DYNAMIC NETWORK

static neuron: \[ W^{(k)} = W^{(k-1)} + \alpha d^{(k)} x^{(k)T} \]

dynamic neuron: \[ W^{(k)}(t) = W^{(k-1)}(t) + \alpha(-t) \cdot d^{(k)}(t) \cdot x^{(k)(t)T} \]

input-output pair: \{x^{(k)}(t), d^{(k)}(t)\}, k=1,..,M

\[ W = [w_1(t), w_2(t), ..., w_N(t)]' \] : weights \((N \times L)\)

e.g. Let \( a(t) = \delta(t) \), \( d^{(k)} = e_k \) a unit vector with its element \( e_{ki} = \delta(k-i) \). Then, \( W(t) = X(t)^T \)

(later used as Model 1: Spatio temporal matched filter bank)
DELTA LEARNING RULE

static neuron: \[ W^{(k)} = W^{(k-1)} - \alpha \{ W^{(k-1)} \bar{x}^{(k)} - \bar{d}^{(k)} \} \bar{x}^{(k)T} \]

dynamic neuron: \[ W^{(k)}(t) = W^{(k-1)}(t) - \alpha(-t) \cdot \{ W^{(k-1)}(t) \cdot \bar{x}^{(k)}(t) - \bar{d}^{(k)}(t) \} \cdot \bar{x}^{(k)}(t)^T \]

with respect to minimizing L2 norm between desired output and the output of NN, delta learning rule is equivalent to learning least square solution of the problem, i.e. where the goal is to minimize

\[ \min R(\sigma) = \sum_k \| \bar{y}^{(k)} - \bar{d}^{(k)} \|^2 \frac{1}{2} + \sigma^2 \sum_k \| \bar{w}_k \|^2 \frac{1}{2} \]

subject to \[ \bar{y}^{(k)} = W \bar{x}^{(k)} \] where \[ W = [\bar{w}_1 \bar{w}_2 \cdots \bar{w}_N]^T \]

yields \[ W(\sigma) = D X^T (X X^T + \sigma^2 I)^{-1} \]

\[ W(t; \sigma) = F^{-1} \{ D(f)X(f)CT(X(f)X(f)^CT + \sigma^2 I)^{-1} \} \]

Model 2: spatio temporal pseudo inverse filter
**Phoneme classification**

**Goal:** Classification of 30 phoneme patterns

**Input:**

- Length $L=100$, bipolar (either $+1$ or $-1$) patterns
- Each phoneme is represented with bipolar (either $+1$ or $-1$) patterns of length $L=100$.
- Pattern encoding scheme: Out of $L$ frequency bands, $+1$ is set for only those that are present in the frequency band of the phoneme.
Phoneme classification

Output:

- Filter response of length 30.
- Desired output encoding scheme: labels are encoded with unit vectors, [0,..., 1,..., 0], where 1 in k-th position indicates that it is the k-th phoneme.
Classification step

Learning phase: Closed form expression exists generalized from hebbian learning rule and delta rule (unless new training dataset is introduced and we have to retrain the data on top of existing data) for linear networks.

1. MODEL 1: Spatio temporal matched filter bank

\[ W(t) = X(t)^T. \]

2. MODEL 2: Spatio temporal pseudo inverse filter bank

\[ W(t) = F^{-1}\{(X(f)^{CT}X(f) + \sigma^2I)^{-1}X^{CT}\}. \]

where W(t) is the weight matrix, X(t)^T is data matrix, X(f) is corresponding fourier transform, CT is conjugate transpose of the matrix, and F^{-1} is inverse fourier transform.
**Classification step**

Testing phase:

1. Spatio-temporal filter bank: feature extraction step

   \[ \vec{y}(t) = W(t) \cdot \vec{x}(t), \text{ and } \vec{z}(t) = r(a(-t)\vec{y}(t)) \]  

   (dot missing)

2. “winner-take-all” lateral inhibition: detection step

   \[ \vec{z}(t) = \vec{z}(t), \text{ and } \vec{z}(t+\Delta) = r(A(-t) \cdot \vec{z}(t) - h), \quad A(t) = (1 + \frac{1}{SN})I\delta(t) - \frac{1}{SN} \sum_{n=0}^{4} \delta(t-n\Delta). \]

where \( W(t) \) is weight matrix, \( x(t) \) is a new testing data, \( r(.) \) is rectifier function, \( a(t) \) is adaptation (another filtering operation, in this time by convolution, e.g. if \( a(t) = \delta(t) \), then we are only considering \( y(t) \) of current \( t \)), \( h \) is a constant threshold vector.
2. “winner-take-all” lateral inhibition: detection step

\[ \mathbf{O}(t) = \mathbf{z}(t), \text{ and } \mathbf{O}(t+\Delta) = r(\mathbf{A}(-t)\cdot \mathbf{O}(t) - \mathbf{h}), \quad \mathbf{A}(t) = (1 + \frac{1}{5N})I \mathbf{\delta}(t) - \frac{1}{5N}1^T \sum_{n=0}^{4} \mathbf{\delta}(t-n\Delta). \]

\[
A(-t) \cdot o(t) = \left(1 + \frac{1}{5N}\right) I \mathbf{\delta}(t) - \frac{1}{5N}1^T \sum_{n=0}^{4} \mathbf{\delta}(t-n\Delta) \\
= \begin{bmatrix}
\delta(t) o_1(t) - \frac{1}{5N} \delta(t)(o_2(t) + \cdots + o_N(t)) \\
\delta(t) o_2(t) - \frac{1}{5N} \delta(t)(o_1(t) + \cdots + o_N(t)) \\
\vdots \\
\delta(t) o_N(t) - \frac{1}{5N} \delta(t)(o_1(t) + \cdots + o_{N-1}(t))
\end{bmatrix} - \frac{1}{5N} \begin{bmatrix}
o_1(t) \cdot \sum_{n=1}^{4} \delta(t-n\Delta) \\
o_2(t) \cdot \sum_{n=1}^{4} \delta(t-n\Delta) \\
\vdots \\
o_N(t) \cdot \sum_{n=1}^{4} \delta(t-n\Delta)
\end{bmatrix}
\]

lateral inhibition: “reduce the activity of its neighbors”

where \(W(t)\) is weight matrix, \(x(t)\) is a new testing data, \(r(.)\) is rectifier function, \(a(t)\) is adaptation (another filtering operation, in this time by convolution, e.g. if \(a(t) = \delta(t)\), then we are only considering \(y(t)\) of current \(t\)), \(h\) is a constant threshold vector.
Learning Model 1: Spatio temporal matched filter bank

(Generalized from Hebbian learning rule
\[ W^{(k)}(t) = W^{(k-1)}(t) + \alpha(-t) \cdot \vec{d}^{(k)}(t) \cdot \vec{x}^{(k)}(t)^T \]

Let \( \alpha(t) = \delta(t) \), \( d^{(k)} = e_k \), a unit vector with its element \( e_{ki} = \delta(k-i) \).

\[ W(t) = X(t)^T, h = 200, \text{ and } a(t) = \sum_{n=0}^{4} \frac{1}{5} \delta(t - n\Delta) \]

This model tries to find the testing input that is most similar with the weight matrix (= data matrix).

Matched filter? Traditionally, if given signal is \( x(t) \), the matched filter is given as \( h(t) = x(-t) \), and the output is \( y(t) = x(t) \ast h(t) = x(t) \circ h(-t) = x(t) \circ x(t) \) (\( \ast \) denotes convolution and \( \circ \) denotes correlation).
LEARNING MODEL 2: SPATIO TEMPORAL PSEUDO INVERSE FILTER BANK

Testing phase:

1. Spatio-temporal filter bank: feature extraction step
   \[ \overline{y}(t) = W(t) \circ \overline{x}(t), \text{ and } z(t) = r(a(-t)\overline{y}(t)) \]

2. "winner-take-all" lateral inhibition: detection step
   \[ \overline{\sigma}(t) = z(t), \text{ and } \overline{\sigma}(t+\Delta) = r(A(-t) \circ \overline{\sigma}(t) - h), \quad A(t) = (1 + \frac{1}{SN}) \delta(t) - \frac{1}{SN} \delta(t) \sum_{n=0}^{d} \delta(t-n\Delta). \]

(Generalized from Delta learning rule \( W^{(k)}(t) = W^{(k-1)}(t) - \alpha(-t) \cdot \{ W^{(k-1)}(t) \circ \overline{x}^{(k)}(t) - \overline{d}^{(k)}(t) \} \circ \overline{x}^{(k)}(t)^T \).)

Let \( D = I, h = 0.05, \sigma^2 = 1000.0, \) and \( a(t) = \delta(t). \) \( W(t) = F^{-1} \left\{ (X(f)^* X(f) + \sigma^2 I)^{-1} X^* \right\}. \)

This minimizes
\[ R(\sigma, f) = \sum_k ||\overline{y}^{(k)}(f) - \overline{d}^{(k)}(f)||_2^2 + \sigma^2 \sum_k ||\overline{w}_k(f)||_2^2 \quad \text{for all } f. \]

*CT: conjugate transpose
PHONEME CLASSIFICATION: GROUND TRUTH

$L = 100$

$N = 30$

$Tw = 5$

5 time units per bipolar phoneme pattern

$W$: weight tensor
Phoneme classification: Test case
PHONEME CLASSIFICATION: PATTERN DETECTION

correlation and additional filtering
PHONEME CLASSIFICATION: PATTERN DETECTION

correlation and additional filtering
DISCUSSION

- Detection is highly sensitive to different thresholds and the choice of adaptation function $a(t)$ (which is relevant to considering how much past values we are going use for detection in the example).
- Multiple iterations are necessary for lateral inhibition network to converge.
- Precaution is required when using Fourier transform because finite nature of the signal causes aliasing.
- Learning is generalized from delta and hebbian rule for linear networks only. However, it is applied to non-linear network in the example, and careful examination is required.
- Though limitations exist, notion of filtering (correlation or convolution) is well generalized from learning rules of neural networks.
SUMMARY

- The paper introduces dynamic network, which incorporates temporal information as well as spatial information of the data.

- The paper proposes to perform correlation with weights and multi-dimensional input to process information in the neurons instead of multiplication of 1-dim input and weights. (This is identical to the feature extraction layer before the activation nowadays performed in CNN)

- The paper generalizes mathematically from formal neuron how the linear dynamic network can be learned by hebbian rule and delta rule.

- The applications for phoneme classification is presented, where sequence of phonemes are detected using dynamic network presented.