
Muthu Chidambaram

Department of Computer Science, University of Virginia

https://qdata.github.io/deep2Read/
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Authors: Sheng Ma, Chuanyi Ji
- Proposes internal-representation-based training of recurrent networks using EM
  - Prior work was based off of heuristics for internal targets
First establishes probabilistic models for targets of recurrent network hidden units
Then uses EM + mean-field approximation to decompose training into a set of feedforward neurons
Considers discrete-time recurrent networks with sigmoid activations
Fast Training of Recurrent Networks Based on EM Algorithm (1998)
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Overview of EM: Introduce hidden variables with missing data (Dmiss), original data (Di), complete data (Dc)
  - E step: \( Q(\Theta|\Theta^p) = E\{\ln P(D_c|\Theta)|D_I, \Theta^p\} = \int P(D_{\text{mis}}|D_I, \Theta^p) \ln P(D_c|\Theta) dD_{\text{mis}} \)
  - M step: \( \Theta^{p+1} = \arg \max_{\Theta} Q(\Theta|\Theta^p) \).

- Choose hidden random vars in EM to be the desired hidden targets
  - Markov property for modeling recurrent networks probabilistically

\[
Q(\Theta|\Theta^p) = \ln P(\{\bar{x}\}|\Theta) + \int P(\{\bar{z}\}|\{y\}, \{\bar{x}\}, \Theta^p) \cdot \ln P(\{y\}, \{\bar{z}\}|\{\bar{x}\}, \Theta) d\{\bar{z}\}.
\]

\[
P(\{\bar{z}\}|\{t\}, \{\bar{x}\}, \Theta)
= \prod_{n=1}^{N} P(\bar{x}(n+1)|\bar{z}(n), y(n+1), \bar{x}(n), \Theta)
\]

\[
P(y(n+1), \bar{z}(n)|f|\bar{z}(n), \bar{x}(n), \Theta).
\]
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Uses Gaussian to model conditional probability distributions due to correspondence with quadratic errors

\[
P(\tilde{x}(n + 1)|\tilde{x}(n), \tilde{z}(n), \Theta) = B_1 \exp \left( -\lambda_1 E_1(n + 1) \right) \\
P(y(n + 1)|\tilde{x}(n + 1), \Theta) = B_2 \exp \left( -\lambda_2 E_2(n + 1) \right)
\]

\[
E_1(n + 1) = ||\tilde{z}(n + 1) - \tilde{h}(n + 1)||^2 \\
E_2(n + 1) = (y(n + 1) - \tilde{z}(n + 1)^T \cdot \bar{w}^{(2)})^2
\]

\[
h_j(n + 1) = g(\tilde{x}(n)^T \cdot \bar{w}_j^{(1)} + \tilde{z}(n)^T \cdot \bar{w}_j^{(3)}) \\
P(y(n + 1)|\tilde{x}(n + 1), \tilde{z}(n), \Theta) = P(y(n + 1)|\tilde{z}(n + 1), \Theta)
\]
Fast Training of Recurrent Networks Based on EM Algorithm (1998)
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

\[
P(y(n+1), \tilde{z}(n+1)|\bar{x}(n), \bar{x}(n), \Theta) = P(y(n+1)|\tilde{z}(n+1), \bar{x}(n), \Theta)P(\tilde{z}(n+1)|\bar{x}(n), \Theta)
\]
\[
= P(y(n+1)|\tilde{z}(n+1), \Theta)P(\tilde{z}(n+1)|\bar{x}(n), \Theta)
\]
\[
= A_{y,z} \exp \left(-\lambda_1 E_1(n+1) - \lambda_2 E_2(n+1)\right)
\]

(16)

\[
P(y(n+1)|\tilde{z}(n), \bar{x}(n), \Theta) = A_y \exp (-\lambda_3 E_3(n+1))
\]

\[
E_3(n+1) = (y(n+1) - \tilde{h}(n+1)^T \cdot \bar{w}^{(2)})^2
\]

\[
P(\{\tilde{z}\} | \{t\}, \{\bar{x}\}, \Theta)
= \prod_{n=1}^{N} A_z \exp \left(-\frac{1}{2} (\tilde{z}(n+1) - \hat{\tilde{z}}(n+1))^T \cdot \Sigma^{-1} (\tilde{z}(n+1) - \hat{\tilde{z}}(n+1))\right)
\]

\[
P(\{t\}, \{\tilde{z}\} | \{\bar{x}\}, \Theta)
= \prod_{n=1}^{N} A_{y,z} \exp (-\lambda_1 E_1(n+1) - \lambda_2 E_2(n+1))
\]

\[
P(\tilde{z}(n+1)|y(n+1), z(n), \bar{x}(n), \Theta)
= \frac{P(y(n+1), \tilde{z}(n+1)|z(n), \bar{x}(n), \Theta)}{P(y(n+1) | z(n), \bar{x}(n), \Theta)}.
\]

\[
P(\tilde{z}(n+1)|y(n+1), \bar{x}(n), \bar{x}(n), \Theta)
= A_z \exp \left(-\frac{1}{2} (\tilde{z}(n+1) - \hat{\tilde{z}}(n+1))^T \cdot \Sigma^{-1} (\tilde{z}(n+1) - \hat{\tilde{z}}(n+1))\right)
\]

\[
\hat{\tilde{z}}(n+1) = \tilde{h}(n+1) + \frac{\lambda_2 w_j^{(2)}}{||\lambda_1 + \lambda_2||w^{(2)}||} e(n+1)
\]

\[
e(n+1) = y(n+1) - \tilde{h}(n+1)^T \cdot \bar{w}^{(2)}
\]
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Can now apply established probabilistic models to EM
- Use self-consistent mean-field approximation to approximate EM integral

\[ Q(\Theta|\Theta^p) = \ln P(\{\bar{z}\}|\Theta) + \sum_{k=1}^{N} \int \prod_{n=1}^{N} P(z(n+1)|y(n+1), \bar{z}(n), \bar{x}(n), \Theta^p) \cdot \ln P(y(k+1), \bar{z}(k+1)|\bar{z}(k), \bar{x}(k), \Theta) \cdot d\{\bar{z}\}. \]

\[ E_{u_i} = \int \int u_i P(u_i|u_j) P(u_j) du_i du_j \approx \int u_i P(u_i|E_{u_j}) du_i \]

\[ P(\bar{z}(n+1)|y(n+1), \bar{z}(n), \bar{x}(n), \Theta^p) \approx P(\bar{z}(n+1)|y(n+1), \hat{z}(n), \bar{x}(n), \Theta^p) \]

\[ \hat{h}_j(n+1) = g(\bar{x}(n)^T \cdot \bar{w}_j^{(1)} + \bar{z}(n)^T \cdot \bar{w}_j^{(3)}) \]

\[ \hat{c}(n+1) = y(n+1) - \hat{h}(n+1)^T \cdot \bar{w}^{(2)} \]

\[ \hat{z}_j(n+1) \approx \hat{h}_j(n+1) + \frac{\lambda_2 \bar{w}_j^{(2)p}}{\lambda_1 + \lambda_2 \| \bar{w}_j^{(2)} \|^p} \hat{c}(n+1) \]
Fast Training of Recurrent Networks Based on EM Algorithm (1998)
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Mean-field approximation reduces maximization step to training a single sigmoidal neuron

\[
Q(\Theta|\Theta^p) = \mathbb{E}_z \{ \ln P(y, \{z\}, \{x\}|\Theta)|\{y\}, \{x\}, \Theta^p \} \\
\approx E_c - E_p - E_h - E_o
\]

\[
(w_j^{(1)})^{(P-1)}, w_j^{(3)})^{(P-1)}
\]

\[
= \arg \min_{w_j^{(1)}, w_j^{(3)}} \sum_n (\tilde{x}(n) - g(\tilde{x}(n)^T \cdot w_j^{(1)} + \tilde{z}(n)^T \cdot w_j^{(3)}))^2
\]

\[
\bar{w}^{(2)}^{\text{opt}} = \arg \min_{\bar{w}^{(2)}} (E_o + E_p)
\]

\[
= \arg \min_{\bar{w}^{(2)}} \lambda_2 \sum_n (\tilde{w}^{(2)^T} \tilde{z}(n) - y(n))^2 + E_p
\]

\[
E_p = \frac{\lambda_2 N \|	ilde{w}^{(2)}\|^2}{2(\lambda_1 + \lambda_2 \|	ilde{w}^{(2)^P}\|^2) + \lambda_2 N (\|	ilde{w}^{(2)}\|^2 \|	ilde{w}^{(2)^P}\|^2 - (\tilde{w}^{(2)^T}(\tilde{w}^{(2)^P})(\tilde{w}^{(2)^P})^T \tilde{w}^{(2)})^2)}
\]

\[
2\lambda_1 (\lambda_1 + \lambda_2 \|	ilde{w}^{(2)^P}\|^2)
\]
Reduce nonlinear optimization of single neuron to linear using Taylor series expansion

\[ w_{\text{opt}} = \arg \min_w \sum_{n=1}^{N} (v(n) - g(w^T \cdot u(n)))^2 \]

\[ w_{\text{opt}} \approx \arg \min_w \sum_{n=1}^{N} c(n)^2 (w^T \cdot u(n) - g^{-1}(v(n)))^2 \]

\[ g(w^T \cdot u(n)) = v(n) + c(n)(w^T \cdot u(n) - g^{-1}(v(n))) + o(|w^T \cdot u(n) - g^{-1}(v(n))|) \]

\[ w_{\text{opt}} = (U^T G^T G U)^{-1} U^T G^T V \]
Randomly initialize $\bar{w}_j^{(1)}$, $\bar{w}_j^{(3)}$ and $w_j^{(2)}$ for $1 \leq j \leq L$.

**E-step:**
Compute the expectation of the desired hidden states $\tilde{z}(n)$ recursively according to (33)–(35) (illustrated by Fig. 3).

**M-step:**
- a) Compute $\bar{w}_j^{(1)}$ and $\bar{w}_j^{(3)}$ given by (41) through linear weighted regressions ((47)).
- b) Compute $\bar{w}^{(2)}$ by solving (42).

Go back to the E-step until certain convergence criteria are satisfied.
Fast Training of Recurrent Networks Based on EM Algorithm (1998)

- Conclusion: Significant speed increase due to reducing training recurrent nets to training of individual neurons
References