CS-667 Advanced Machine Learning

Nazar Khan

PUCIT

Mixture Density Networks

Forward and Inverse Problems

- Goal of supervised learning: model conditional distribution $p(\mathbf{t}|\mathbf{x})$.
- For simple regression problems $p(\mathbf{t}|\mathbf{x})$ is assumed to be Gaussian.
- However, *practical machine learning problems* can have significantly non-Gaussian distributions.

Forward Problems

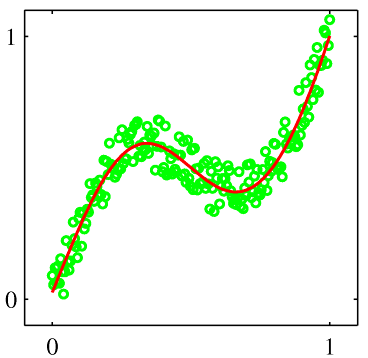

Figure: Successful neural network learning of a *uni-modal* forward problem ($t_n = x_n + 0.3\sin(2\pi x_n) + \epsilon$) using SSE function.

Inverse Problems

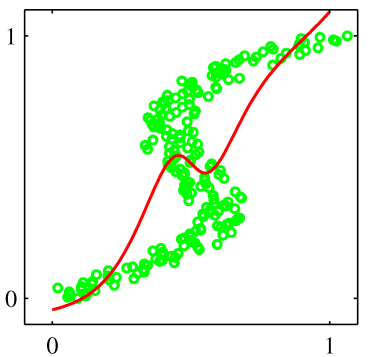

Figure: Unsuccessful neural network learning of a *multi-modal* inverse problem (roles of t_n and x_n reversed). *Reason for failure*: Training NN with SSE function implies $t \sim \mathcal{N}$. However, for multi-modal inverse problems $t \nsim \mathcal{N}$ and the learned model is a very poor fit of the underlying model.

Mixture Density Networks

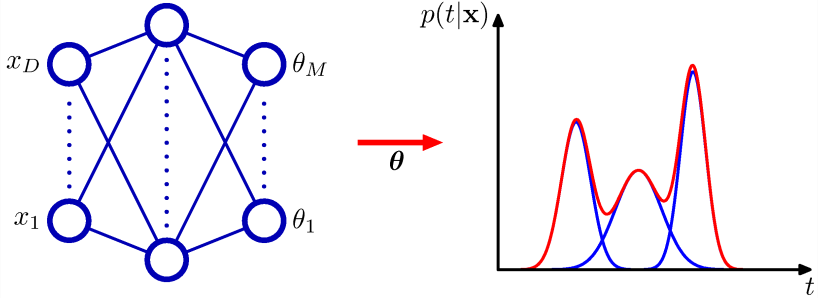

Figure: Mixture density network. Outputs are the mixture parameters $\boldsymbol{\theta}(\mathbf{x})$ corresponding to input $\mathbf{x}$. *Difference from earlier approaches*: Instead of learning parameters $\boldsymbol{\theta}$, we learn NN weights $\mathbf{w}$ that produce parameters $\boldsymbol{\theta}(\mathbf{x})$ that model the density conditioned on input $\mathbf{x}$.

The Formulation

- We will assume continuous targets and isotropic Gaussian components.
- The likelihood for one data point $(\mathbf{x}, \mathbf{t})$ can be written as

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathcal{N}(\mathbf{t}|\boldsymbol{\mu}_k(\mathbf{x}), \sigma_k^2(\mathbf{x})\mathbf{I})$$

- The component densities need not be isotropic Gaussians.
- They can be chosen according to the problem at hand (e.g Bernoulli densities if target t is a binary random variable).

The Network

- Let $\mathbf{t} \in \mathbb{R}^D$.
- Size of input layer determined by size of $\mathbf{x}$.
- Number and sizes of hidden layers are hyperparameters.
- Output layers will consist of
  1. K neurons representing the mixing coefficients $\pi_1(\mathbf{x}), \dots, \pi_K(\mathbf{x})$.
  2. KD neurons representing the mean vectors $\boldsymbol{\mu}_1(\mathbf{x}), \dots, \boldsymbol{\mu}_K(\mathbf{x})$.
  3. K neurons representing the widths of the Gaussian kernels $\sigma_1(\mathbf{x}), \dots, \sigma_K(\mathbf{x})$.

  Therefore, size of output layer will be $K + KD + K = K(D + 2)$.

The Network

- For the 3 types of output neurons, we will use the following notation
  1. a_k^{π} – activation of neuron representing k-th mixing coefficient.
  2. a_{kj}^{μ} – activation of neuron representing j-th component of k-th mean vector.
  3. a_k^{σ} – activation of neuron representing standard deviation of k-th Gaussian.

Modelling the outputs

- Mixing coefficients must satisfy $0 \leq \pi_k(\mathbf{x}) \leq 1$ and also $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$. This can be achieved via softmax outputs

$$\pi_k(\mathbf{x}) = \frac{e^{a_k^\pi}}{\sum_{i=1}^{K} e^{a_i^\pi}}$$

- Means have no constraints and can be modelled directly as

$$\mu_{kj}(\mathbf{x}) = a_{kj}^{\mu}$$

- Standard deviations must satisfy $\sigma_k(\mathbf{x}) \geq 0$ and can be modelled as

$$\sigma_k(\mathbf{x}) = e^{a_k^\sigma}$$

Training
Likelihood

- Given training data pairs $\{\mathbf{x}_n, \mathbf{t}_n\}$, the goal now will be to learn the weights $\mathbf{w}$ of the neural network *so that* it outputs $K(D+2)$ parameters
$\pi_1(\mathbf{x}_n, \mathbf{w}), \dots, \pi_K(\mathbf{x}_n, \mathbf{w})$,
$\boldsymbol{\mu}_1(\mathbf{x}_n, \mathbf{w}), \dots, \boldsymbol{\mu}_K(\mathbf{x}_n, \mathbf{w})$ and
$\sigma_1(\mathbf{x}_n, \mathbf{w}), \dots, \sigma_K(\mathbf{x}_n, \mathbf{w})$
that maximize the likelihood of targets given inputs.

$$\begin{aligned}
\mathbf{w}^* &= \arg\max_{\mathbf{w}} \prod_{n=1}^{N} p(\mathbf{t}_n | \mathbf{x}_n, \mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k(\mathbf{x}_n, \mathbf{w}) \mathcal{N}(\mathbf{t}_n | \boldsymbol{\mu}_k(\mathbf{x}_n, \mathbf{w}), \sigma_k^2(\mathbf{x}_n, \mathbf{w}) \mathbf{I})
\end{aligned}$$

Training
Negative log-likelihood

- Negative log-likelihood can be written as

$$
\begin{aligned}
E(\mathbf{w}) &= -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k(\mathbf{x}_n, \mathbf{w}) \mathcal{N}(\mathbf{t}_n | \boldsymbol{\mu}_k(\mathbf{x}_n, \mathbf{w}), \sigma_k^2(\mathbf{x}_n, \mathbf{w})\mathbf{I}) \right\} \\
&= -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_{nk} \mathcal{N}_{nk} \right\} \qquad \text{for notational clarity}
\end{aligned}
$$

- All that is required to initiate backpropagation are the partial derivatives $\frac{\partial E_n}{\partial a_k^{\pi}}, \frac{\partial E_n}{\partial a_{kj}^{\mu}}$ and $\frac{\partial E_n}{\partial a_k^{\sigma}}$.

Training

Derivatives

$$
\begin{aligned}
\frac{\partial E_n}{\partial a_k^\pi} &= \frac{-\sum_{j=1}^K \pi_{nj}(\delta_{jk}-\pi_{nk})\mathcal{N}_{nj}}{\sum_{j=1}^K \pi_{nj}\mathcal{N}_{nj}} \\
&= -\sum_{j=1}^K \frac{\pi_{nj}\mathcal{N}_{nj}(\delta_{jk}-\pi_{nk})}{\sum_{j=1}^K \pi_{nj}\mathcal{N}_{nj}} \\
&= -\sum_{j=1}^K r_{nj}(\delta_{jk}-\pi_{nk}) \\
&= -r_{nk} + \pi_{nk}\underbrace{\sum_{j=1}^K r_{nj}}_{=1} \\
&= \pi_{nk} - r_{nk}
\end{aligned}
$$

Training
Derivatives

$$
\begin{aligned}
\frac{\partial E_n}{\partial a^{\mu}_{kj}} &= \frac{-\pi_{nk}\frac{\partial \mathcal{N}_{nk}}{\partial a^{\mu}_{kj}}}{\sum_{i=1}^{K}\pi_{ni}\mathcal{N}_{ni}} = \frac{-\pi_{nk}\frac{\partial \mathcal{N}_{nk}}{\partial \mu_{nkj}}\frac{\partial \mu_{nkj}}{a^{\mu}_{kj}}}{\sum_{i=1}^{K}\pi_{ni}\mathcal{N}_{ni}} \\
&= \frac{-\pi_{nk}\mathcal{N}_{nk}\left\{\frac{-(t_{nj}-\mu_{nkj})(-1)}{\sigma^2_{nk}}\right\}}{\sum_{i=1}^{K}\pi_{ni}\mathcal{N}_{ni}} \\
&= r_{nk}\left\{\frac{\mu_{nkj}-t_{nj}}{\sigma^2_{nk}}\right\}
\end{aligned}
$$

Training
Derivatives

$$\frac{\partial E_n}{\partial a_k^\sigma} = \frac{-\pi_{nk}\frac{\partial \mathcal{N}_{nk}}{\partial a_k^\sigma}}{\sum_{i=1}^K \pi_{ni}\mathcal{N}_{ni}} = \frac{-\pi_{nk}\frac{\partial \mathcal{N}_{nk}}{\partial \sigma_{nk}}\frac{\partial \sigma_{nk}}{\partial a_k^\sigma}}{\sum_{i=1}^K \pi_{ni}\mathcal{N}_{ni}}$$
$$= r_{nk}\left\{1 - \frac{\left\|\mathbf{t}_n - \boldsymbol{\mu}_{nk}\right\|^2}{\sigma_{nk}^2}\right\}$$

Take-home Quiz 6

- Show that

$$\frac{\partial \mathcal{N}_{nk}}{\partial \sigma_{nk}} = \mathcal{N}_{nk}\left\{\frac{\left\|\mathbf{t}_n - \boldsymbol{\mu}_{nk}\right\|^2}{\sigma_{nk}^3} - \frac{1}{\sigma_{nk}}\right\}$$

to prove the formula for $\frac{\partial E_n}{\partial a_k^\sigma}$ provided above.

Please note that Equation (5.157) in Bishop's book is incorrect.

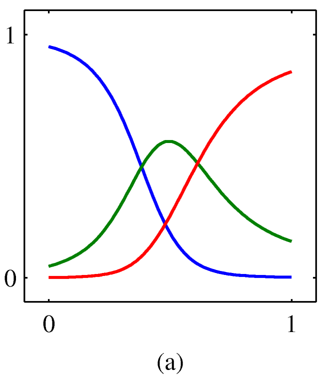

Figure: Mixing coefficients $\pi_k(x)$. At both small and large values of x where $p(t|x)$ is uni-modal, only one mixture component has a larger role. For intermediate values of x where the density is tri-modal, all 3 mixing coefficients have comparable values.

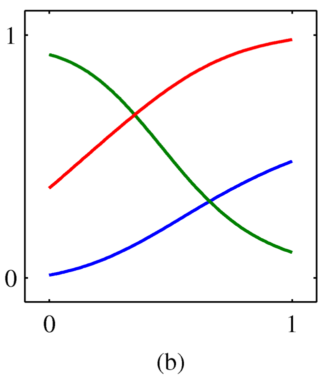

Figure: Means $\mu_k(x)$.

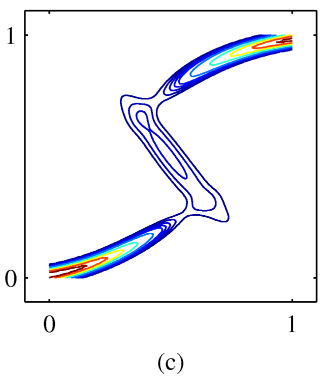

Figure: Contours of $p(t|x)$. Higher density at more certain (uni-modal) outputs

.

Obtaining a unique answer

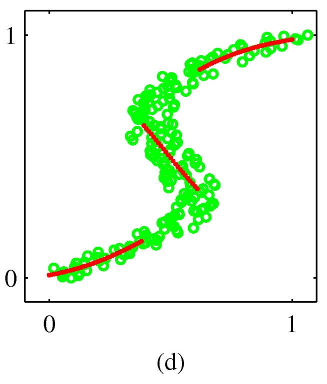

Figure: Approximate modes of conditional density $p(t|x)$ by using the mean of the component with the highest $\pi_k(\mathbf{x})$.