Output Nonlinearities, Training Criteria and Bayesian Interpretation

Yang Zhang
Skeleton

- MMSE training criterion
- 2-class classification – logistic nonlinearity
- Multiclass classification – output structure
- Multiclass classification – softmax nonlinearity
- Minimum cross entropy criterion
Types of learning problem

• Feature & explained variable pairs $(x, t) \sim p_{x,t}$
  • $(x, t)$ is a pair of random variables
  • $t \in [0,1]$: 2-class classification
  • $t \in \mathbb{Z}$, each integer denotes a class: Multiclass classification
  • $t \in \mathbb{R}$: Regression
  • $t \in \mathbb{Z}$, each integer denotes level: Regression with discrete explained variable

• $\{(x^{(n)}, t^{(n)})\}$ - a set of training tokens independently drawn from $p_{x,t}$
Minimum Mean Squared Error

- Works generally well for learning tasks

\[ \mathcal{L} = \sum_{n} (y^{(n)} - t^{(n)})^2 \]

\( y^{(n)} \) - output of the classifier
\( t^{(n)} \) - labels
\[ y^{(n)} = F(x^{(n)}) \]
\[ F \in \mathcal{F} \]
\( \mathcal{F} \) - the set of all functions representable by the neural network architecture.

\[ \min_{F \in \mathcal{F}} \sum_{n} (F(x^{(n)}) - t^{(n)})^2 \]
MMSE and Bayesian framework

• Given enough hidden nodes and training tokens, neural network with MMSE criterion would give
  \[ \forall \text{test sample } x' \]
  \[ y = F(x') = E(t|x = x') \]
• MMSE leads to posterior expectation!
MMSE and Bayesian framework

- For 2-class classification problem
- Assume 0-1 loss
  \[ E(t|x = x') = 1 \cdot p_{t|x}(1|x') + 0 \cdot p_{t|x}(0|x') \]
  \[ = p_{t|x}(1|x') = \Pr(t = 1|x = x') \]
- For regression problem
MMSE and Bayesian framework

• Proof

$$\min_{F \in \mathcal{F}} \sum_n (F(x^{(n)}) - t^{(n)})^2$$
$$\approx \min_{all \ F} \sum_n (F(x^{(n)}) - t^{(n)})^2$$
$$\approx \min_{all \ F} \int \int (F(x') - t')^2 p_{x,t}(x', t') dx' dt'$$

$$= \min_{all \ F} \int \int (F(x') - E(t|x = x') + E(t|x = x') - t')^2 p_{x,t}(x', t') dx' dt'$$

$$= \min_{all \ F} \int \int (F(x') - E(t|x = x') + E(t|x = x') - t')^2 p_{t|x}(t'|x') dt' p_x(x') dx'$$

$$= \min_{all \ F} \int \int [(F(x') - E(t|x = x'))^2 + (E(t|x = x') - t')^2]$$
MMSE and Bayesian framework

• Proof (Con’d)

\[
\int \int 2(F(x') - E(t|x = x'))(E(t|x = x') - t') p_{t|x}(t'|x') dt' p_x(x') dx' \\
= \int 2(F(x'))
\]
MMSE and Bayesian framework

• Proof (Con’d)

\[ = \min_{\text{all } F} \iint (F(x') - E(t|x = x'))^2 \]
2-Class Classification

• If the training data size not that large, the output may not be so well behaved.
• May not even be within $[0, 1]$.
• Need to choose output nonlinearity such that the output is confined to $[0, 1]$. 
Skeleton

• MMSE training criterion
• 2-class classification – logistic nonlinearity
• Multiclass classification – output structure
• Multiclass classification – softmax nonlinearity
• Minimum cross entropy criterion
Logistic Function

- Logistic Function:
  \[
g(a) = \frac{1}{1 + e^{-a}}
  \]
  \[
g'(a) = \frac{e^{-a}}{(1 + e^{-a})^2}
  \]
Logistic Function – Bayesian Interpretation

• If $p_{x_{L-1}|t}$ Exponential family + additional constraints, the posterior probability naturally takes the form of

$$p_{t|x_{L-1}}(1|x_{L-1}^{(n)}) = \frac{1}{1 + e^{-(b_{L-1} + w_{L-1} x_{L-1}^{(n)})}}$$

Proof

$$p_{x_{L-1}|t}(x_{L-1}^{(n)}|0) = \alpha(x) \exp\left(a(\theta_0) + d(\theta_0)^T x_{L-1}^{(n)}\right)$$

$$p_{x_{L-1}|t}(x_{L-1}^{(n)}|1) = \alpha(x) \exp\left(a(\theta_1) + d(\theta_1)^T x_{L-1}^{(n)}\right)$$

Then

$$p_{t|x_{L-1}}(1|x_{L-1}^{(n)}) = \frac{\pi_1 p_{x_{L-1}|t}(x_{L-1}^{(n)}|1)}{\pi_0 p_{x_{L-1}|t}(x_{L-1}^{(n)}|0) + \pi_1 p_{x_{L-1}|t}(x_{L-1}^{(n)}|1)}$$
Logistic Function – Bayesian Interpretation

• Proof (cont’d)

\[ p_t | x_{L-1} \left( 1 | x_{L-1}^{(n)} \right) \]

\[
= \frac{\exp \left( \log \pi_1 + a(\theta_1) + \mathbf{d}(\theta_1)^T \mathbf{x}_{L-1}^{(n)} \right)}{\exp \left( \log \pi_0 + a(\theta_0) + \mathbf{d}(\theta_0)^T \mathbf{x}_{L-1}^{(n)} \right) + \exp \left( \log \pi_1 + a(\theta_1) + \mathbf{d}(\theta_1)^T \mathbf{x}_{L-1}^{(n)} \right)}
\]

\[
= \frac{1}{1 + \exp \left[ - \left( \log \frac{\pi_1}{\pi_0} + a(\theta_1) - a(\theta_0) \right) + (\mathbf{d}(\theta_1) - \mathbf{d}(\theta_0))^T \mathbf{x}_{L-1}^{(n)} \right]}
\]

Let

\[ b_{L-1} = \log \frac{\pi_1}{\pi_0} + a(\theta_1) - a(\theta_0) \]

\[ \mathbf{w}_{L-1} = \mathbf{d}(\theta_1) - \mathbf{d}(\theta_0) \]

Then

\[ p_t | x_{L-1} \left( 1 | x_{L-1}^{(n)} \right) = \frac{1}{1 + e^{-\left( b_{L-1} + \mathbf{w}_{L-1} x_{L-1}^{(n)} \right)}} \]
Multi-class classification

• \( t^{(n)} \in \{1,2,\ldots,M\} \)
• \( M \) – total number of classes
• Question: is

\[
\mathcal{L} = \sum_{n} \left( y^{(n)} - t^{(n)} \right)^2
\]

a good metric?

• No, because there’s no reason to assume class 3 is closer to class 2 than to class 1.
Skeleton

- MMSE training criterion
- 2-class classification – logistic nonlinearity
- Multiclass classification – output structure
- Multiclass classification – softmax nonlinearity
- Minimum cross entropy criterion
Multi indicator output

- Label as an indicator vector

\[ t^{(n)} = [0,0, \ldots, 0,1,0, \ldots, 0]^T \]

\[ c^{(n)} \in \{1,2,3, \ldots, M\} \]

\[ t^{(n)}(i) = \begin{cases} 
1 & c^{(n)} = i \\
0 & \text{otherwise}
\end{cases} \]

- Accordingly, there should be \( M \) output nodes:
Multi indicator output - MMSE

$$\mathcal{L} = \sum_n \sum_i \left( y^{(n)}(i) - t^{(n)}(i) \right)^2$$

$$= \sum_n \| y^{(n)} - t^{(n)} \|^2$$

• Similarly, with enough training data and representation power, the output gives posterior probability

$$y^{(n)}(i) = p_{t(i)|x}(1|x^{(n)}) = \Pr(t^{(n)}(i) = 1|x = x^{(n)})$$

$$= \Pr(c^{(n)} = i|x = x^{(n)})$$

• Inherent constraint

$$\sum_i y^{(n)}(i) = 1$$

• What if there is not enough training data?
Skeleton

• MMSE training criterion
• 2-class classification – logistic nonlinearity
• Multiclass classification – output structure
• Multiclass classification – softmax nonlinearity
• Minimum cross entropy criterion
Softmax function

• Definition

\[ y^{(n)}(i) = \frac{\exp(-a^{(n)}(i))}{\sum_j \exp(-a^{(n)}(j))} \]

• It can be easily verify that

\[ \sum_i y^{(n)}(i) = 1 \]
Softmax function

• Why softmax function?

If $p_{x_{L-1}|t}$ Exponential family + additional constraints, the posterior probability naturally takes the form of

$$p_{t(i)|x_{L-1}}(1|x'_{L-1}) = \frac{\exp(-\alpha'(i))}{\sum_j \exp(-\alpha'(j))}$$

where

$$\alpha'(i) = b_{L-1}(i) + W_{L-1}(i,:)x'_{L-1}$$

• You will prove it in your next homework
Softmax function

- Structure more complex -> derivative more complex
  \[
  y^{(n)}(i) = \frac{\exp(-a^{(n)}(i))}{\sum_j \exp(-a^{(n)}(j))}
  \]

- Is there a good loss function so as to make the derivative simpler?
Skeleton

- MMSE training criterion
- 2-class classification – logistic nonlinearity
- Multiclass classification – output structure
- Multiclass classification – softmax nonlinearity
- Minimum cross entropy criterion
Entropy

• Definition:

\[ p_X(i) - \text{pmf} \]

\[
H(p_X) = - \sum_i p_X(i) \log p_X(i)
\]

• Interpretation:

Expected amount of information that should be received before nailing down to one instance

• Minimum

\[ p_X(i) = \delta[i - k] \text{ for some } k \]

\[ H(p_X) = 0 \]

• Maximum

\[ p_X(i) = 1/M \text{ uniform distribution} \]

\[ H(p_X) = \log M \]
Cross Entropy

• Definition:

\[ p_X(i), q_X(i) - \text{pmf's} \]

\[ H(p_X|q_X) = - \sum_i p_X(i) \log q_X(i) \]

• Interpretation:

amount of information (encoded for another distribution \( q_X(i) \)) that should be received before nailing down to one instance

• Minimum (over \( q_X(i) \), given \( p_X(i) \))

\[ q_X(i) = p_X(i) \]

\[ H(p_X|q_X) = H(p_X) \]
Cross Entropy as a Loss Function

- $t^{(n)}(i)$ - pmf; $y^{(n)}(i)$ - pmf.

$$\ell^{(n)} = H(t^{(n)}|y^{(n)}) = \sum_i t^{(n)}(i) \log y^{(n)}(i)$$

$$\mathcal{L} = \sum_n \ell^{(n)} = \sum_n \sum_i t^{(n)}(i) \log y^{(n)}(i)$$

- Questions:
  - Does minimum cross entropy criterion given the optimum result as the posterior probability?
  - Does minimum cross entropy criterion result in simpler derivatives?
Cross Entropy and Bayesian Framework

• Given enough hidden nodes and training tokens, neural network with minimum cross entropy criterion and softmax function would give

\[ \forall \text{ test sample } \mathbf{x}' \]
\[ y'(i) = F_i(x') = P_{t(i)|x}(1|x') = \Pr(c' = i|x = x') \]

• Proof

\[
\min_{F \in \mathcal{F}} \sum_n \sum_i t^{(n)}(i) \log F_i(x^{(n)}) \\
\approx \min_{\text{all } F: \sum_i F_i = 1} \sum_n \sum_i t^{(n)}(i) \log F_i(x^{(n)}) \\
= \min_{\text{all } F: \sum_i F_i = 1} \sum_n \log F_{c^{(n)}}(x^{(n)}) \\
\approx \min_{\text{all } F: \sum_i F_i = 1} \int \sum_i p_{c,x}(i, x') \log F_i(x') \, dx' 
\]
Cross Entropy and Bayesian Framework

• Proof (Cont’d)

$$\min_{\text{all } F: \sum_i F_i = 1} \int \sum_i p_{c,x}(i, x') \log F_i(x') \, dx'$$

$$= \min_{\text{all } F: \sum_i F_i = 1} \int \left[ \sum_i p_{c|x}(i|x') \log F_i(x') \right] p_x(x') \, dx'$$

• Optimal solution

$$F_i(x') = p_{c|x}(i|x')$$
Cross Entropy – Back Prop.

\[ \mathcal{L} = \sum_n \ell^{(n)} = \sum_n \sum_i t^{(n)}(i) \log y^{(n)}(i) \]

\[ y^{(n)}(i) = \frac{\exp \left( -a^{(n)}(i) \right)}{\sum_j \exp \left( -a^{(n)}(j) \right)} \]

Chain rule

\[ \frac{\partial \ell^{(n)}}{\partial a^{(n)}(i)} = \sum_k \frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} \cdot \frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)} \]
Cross Entropy – Back Prop.

\[
\frac{\partial \ell^{(n)}}{\partial a^{(n)}(i)} = \sum_k \frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} \cdot \frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)} \\
\ell^{(n)} = - \sum_i t^{(n)}(i) \log y^{(n)}(i)
\]

First term

\[
\frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} = - \frac{t^{(n)}(k)}{y^{(n)}(k)}
\]
Cross Entropy – Back Prop.

\[
\frac{\partial \ell^{(n)}}{\partial a^{(n)}(i)} = \sum_k \frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} \cdot \frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)} \\

y^{(n)}(k) = \frac{\exp\left(-a^{(n)}(k)\right)}{\sum_j \exp\left(-a^{(n)}(j)\right)}
\]

Second term

\[
\frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)} = \begin{cases} 
-y^{(n)}(k)y^{(n)}(i), & k \neq i \\
y^{(n)}(k) - [y^{(n)}(k)]^2, & k = i 
\end{cases}
\]
Cross Entropy – Back Prop.

\[
\frac{\partial \ell^{(n)}}{\partial a^{(n)}(i)} = \sum_k \frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} \cdot \frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)}
\]

First term

\[
\frac{\partial \ell^{(n)}}{\partial y^{(n)}(k)} = -\frac{t^{(n)}(k)}{y^{(n)}(k)}
\]

Second term

\[
\frac{\partial y^{(n)}(k)}{\partial a^{(n)}(i)} = \begin{cases} 
- y^{(n)}(k) y^{(n)}(i), & k \neq i \\
y^{(n)}(k) - [y^{(n)}(k)]^2, & k = i
\end{cases}
\]

Final result

\[
\frac{\partial \ell^{(n)}}{\partial a^{(n)}(i)} = y^{(n)}(i) - t^{(n)}(i)
\]
## Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Training Criteria</th>
<th>Output Nonlinearity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>MMSE</td>
<td></td>
</tr>
<tr>
<td>2-class classification</td>
<td>MMSE</td>
<td>Logistic</td>
</tr>
<tr>
<td>Multi-class class.</td>
<td>Min cross entropy</td>
<td>Softmax</td>
</tr>
</tbody>
</table>