Deep Learning

7 Cross-Entropy Loss

Dr. Konda Reddy Mopuri
Dept. of Artificial Intelligence
IIT Hyderabad
Jan-May 2023

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. We don't generally $(C > 2)$ regress the target (Why not?)

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. We don't generally $(C > 2)$ regress the target (Why not?)
3. In other words, we don't prefer MSE loss for learning

Classification Intuition

1. Target label y is one-hot encoded

Classification Intuition

1. Target label y is one-hot encoded
2. It converts y to a pmf $(\mathbf{p})$ (e.g., $y_n = 2 \rightarrow \{0, 1, 0, 0\}$ and $y_n = 3 \rightarrow \{0, 0, 1, 0\}$ when $C = 4$)

Classification Intuition

1. Target label y is one-hot encoded
2. It converts y to a pmf ($\mathbf{p}$) (e.g., $y_n = 2 \rightarrow \{0, 1, 0, 0\}$ and $y_n = 3 \rightarrow \{0, 0, 1, 0\}$ when $C = 4$)
3. Hence, the DNN's prediction should also be a pmf ($\mathbf{q}$)

Classification Intuition

1. Target label y is one-hot encoded
2. It converts y to a pmf ($\mathbf{p}$) (e.g., $y_n = 2 \rightarrow \{0, 1, 0, 0\}$ and $y_n = 3 \rightarrow \{0, 0, 1, 0\}$ when $C = 4$)
3. Hence, the DNN's prediction should also be a pmf ($\mathbf{q}$)
4. Loss function should compare $\mathbf{p}$ and $\mathbf{q}$

Very very brief discussion on related Information Theory

1. Information contained in an event x can be computed given the probability of that event $P(x)$

Very very brief discussion on related Information Theory

1. Information contained in an event x can be computed given the probability of that event $P(x)$
2. Higher the $P(x)$, lesser is the information (less 'surprising')

1. Information contained in an event x can be computed given the probability of that event $P(x)$
2. Higher the $P(x)$, lesser is the information (less 'surprising')
3. Hence, the information can be calculated as $I(x) = -log_2(P(x))$

Very very brief discussion on related Information Theory

1. Information contained in an event x can be computed given the probability of that event $P(x)$
2. Higher the $P(x)$, lesser is the information (less 'surprising')
3. Hence, the information can be calculated as $I(x) = -log_2(P(x))$
4. This is also the number of bits required to encode x

Very very brief discussion on related Information Theory

1. Entropy is the number of bits required to encode a randomly chosen message (x) from a probability distribution $p(x)$

Very very brief discussion on related Information Theory

1. Entropy is the number of bits required to encode a randomly chosen message (x) from a probability distribution $p(x)$
2. Expected amount of information in an event drawn from that distribution $H(X) = \mathbb{E}_{x \sim p}[I(x)]$

Very very brief discussion on related Information Theory

1. One message x needs $-log(P(x))$ bits

Very very brief discussion on related Information Theory

1. One message x needs $-log(P(x))$ bits
2. There are multiple messages with associated probabilities $\rightarrow$ entropy $H(X) = -\sum P(x) \cdot log_2(P(x))$

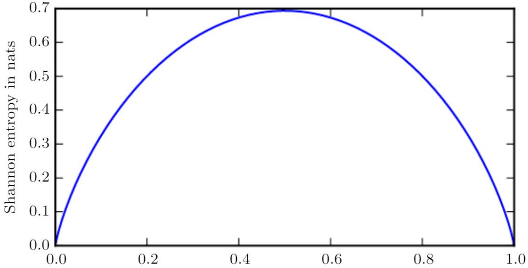
Very very brief discussion on related Information Theory

1. One message x needs $-log(P(x))$ bits
2. There are multiple messages with associated probabilities $\rightarrow$ entropy $H(X) = -\sum P(x) \cdot log_2(P(x))$
3. $H(p) = -\sum_i p_i \cdot log_2(p_i)$

Very very brief discussion on related Information Theory

1. One message x needs $-log(P(x))$ bits
2. There are multiple messages with associated probabilities $\rightarrow$ entropy $H(X) = -\sum P(x) \cdot log_2(P(x))$
3. $H(p) = -\sum_i p_i \cdot log_2(p_i)$
4. Skewed distribution has less entropy, uniform/balanced distribution has more entropy

Very very brief discussion on related Information Theory



Entropy for a binary random variable

Figure credits Goodfellow et al. 2016

Very very brief discussion on related Information Theory

1. Cross-entropy $H(p, q)$ is the average number of bits required to encode the messages from a source distribution p when encoded with a different model q

Very very brief discussion on related Information Theory

1. Cross-entropy $H(p,q)$ is the average number of bits required to encode the messages from a source distribution p when encoded with a different model q
2. $H(p,q) = -\sum_i p_i \cdot log_2(q_i)$

Very very brief discussion on related Information Theory

1. Cross-entropy $H(p, q)$ is the average number of bits required to encode the messages from a source distribution p when encoded with a different model q
2. $H(p, q) = -\sum_i p_i \cdot log_2(q_i)$
3. Note that cross-entropy is not symmetric metric, i.e, $H(p, q) \neq H(q, p)$

1. Cross-entropy $H(p, q)$ is the average number of bits required to encode the messages from a source distribution p when encoded with a different model q
2. $H(p, q) = -\sum_i p_i \cdot log_2(q_i)$
3. Note that cross-entropy is not symmetric metric, i.e, $H(p, q) \neq H(q, p)$
4. Cross-entropy between a distribution and itself $(H(p, q))$ gives the entropy of the distribution $H(p)$

1. KL-Divergence : average number of extra bits required to represent a message with distribution q instead of p

Very very brief discussion on related Information Theory

1. KL-Divergence : average number of extra bits required to represent a message with distribution q instead of p
2. $H(p, q) = H(p) + KL(p||q)$ where $KL(p||q) = \sum p_i \cdot log\left(\frac{p_i}{q_i}\right)$

Cross-entropy as a loss function

1. Widely used in classification problems (e.g. logistic regression, NNs)

Cross-entropy as a loss function

1. Widely used in classification problems (e.g. logistic regression, NNs)
2. Each label is converted into a distribution with 1 and 0s (one-hot encoding)

Cross-entropy as a loss function

1. Widely used in classification problems (e.g. logistic regression, NNs)
2. Each label is converted into a distribution with 1 and 0s (one-hot encoding)
3. Model predicts the probabilities that sample belongs to different classes

Cross-entropy as a loss function

1. Random variable is the sample

Cross-entropy as a loss function

1. Random variable is the sample
2. Events are the classes

Cross-entropy as a loss function

1. Random variable is the sample
2. Events are the classes
3. Target distribution (or, groundtruth) is one-hot encoding p, and model predicts a distribution q

Cross-entropy as a loss function

1. Random variable is the sample
2. Events are the classes
3. Target distribution (or, groundtruth) is one-hot encoding p, and model predicts a distribution q
4. `torch.nn.CrossEntropyLoss(q,p)` (PyTorch takes predicted distribution as the first argument)

Softmax

1. Typically last layer in the DNN classifier is linear (without a nonlinearity)

Softmax

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

1. Typically last layer in the DNN classifier is linear (without a nonlinearity)
2. Predicts the confidences to each class (may not lie in $[0, 1]$)

Softmax

1. Typically last layer in the DNN classifier is linear (without a nonlinearity)
2. Predicts the confidences to each class (may not lie in $[0, 1]$)
3. But, we need probabilities

Softmax

1. Typically last layer in the DNN classifier is linear (without a nonlinearity)
2. Predicts the confidences to each class (may not lie in $[0, 1]$)
3. But, we need probabilities
4. Softmax operation
   - squashes the predicted confidences to lie in $[0, 1]$

Softmax

1. Typically last layer in the DNN classifier is linear (without a nonlinearity)
2. Predicts the confidences to each class (may not lie in $[0, 1]$)
3. But, we need probabilities
4. Softmax operation
   - squashes the predicted confidences to lie in $[0, 1]$
   - make them probabilities (i.e. sum to 1)

Softmax

1. $(\alpha_1, \alpha_2, \ldots, \alpha_C) \rightarrow \left(\frac{e^{\alpha_1}}{\sum_i e^{\alpha_i}}, \frac{e^{\alpha_2}}{\sum_i e^{\alpha_i}}, \ldots, \frac{e^{\alpha_C}}{\sum_i e^{\alpha_i}} \right)$

Softmax

1. $(\alpha_1, \alpha_2, \ldots, \alpha_C) \rightarrow \left(\frac{e^{\alpha_1}}{\sum_i e^{\alpha_i}}, \frac{e^{\alpha_2}}{\sum_i e^{\alpha_i}}, \ldots, \frac{e^{\alpha_C}}{\sum_i e^{\alpha_i}} \right)$
2. $(\alpha_1, \alpha_2, \ldots, \alpha_C) \rightarrow (q_1, q_2, \ldots, q_C)$

Cross-entropy

1. Target distribution p has 1 at the position of correct label and 0 at rest of the components

Cross-entropy

1. Target distribution p has 1 at the position of correct label and 0 at rest of the components
2. $H(p,q) = -\sum p_i \cdot log(q_i) = -log(q_c)$, where c is the groundtruth class of the sample

Cross-entropy

1. Target distribution p has 1 at the position of correct label and 0 at rest of the components
2. $H(p,q) = -\sum p_i \cdot log(q_i) = -log(q_c)$, where c is the groundtruth class of the sample
3. The cross-entropy loss is
   - small when the model predicts high probability to the groundtruth class $(q_c \approx 1)$

Cross-entropy

1. Target distribution p has 1 at the position of correct label and 0 at rest of the components
2. $H(p, q) = -\sum p_i \cdot log(q_i) = -log(q_c)$, where c is the groundtruth class of the sample
3. The cross-entropy loss is
   - small when the model predicts high probability to the groundtruth class $(q_c \approx 1)$
   - large if the model assigns smaller probability for the groundtruth class $(q_c \approx 0)$

Variations/Additions

1. BCE: Binary Cross Entropy Loss

Variations/Additions

1. BCE: Binary Cross Entropy Loss
2. Softmax with temperature