Deep Learning

5.1 Crossentropy loss

Dr. Konda Reddy Mopuri
kmopuri@iittp.ac.in
Dept. of CSE, IIT Tirupati

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. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$
3. One-hot encoding $\rightarrow \{0, 1, 0\}$

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$
3. One-hot encoding $\rightarrow \{0, 1, 0\}$
4. Consider two predictions $y_1 = \{-2.0, 1.5 - 3.0\}$ and $y_2 = \{0.5, -3.0, -3.0\}$

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$
3. One-hot encoding $\rightarrow \{0, 1, 0\}$
4. Consider two predictions $y_1 = \{-2.0, 1.5 - 3.0\}$ and $y_2 = \{0.5, -3.0, -3.0\}$
5. Both result in the same (13.25) loss value

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$
3. One-hot encoding $\rightarrow \{0, 1, 0\}$
4. Consider two predictions $y_1 = \{-2.0, 1.5 - 3.0\}$ and $y_2 = \{0.5, -3.0, -3.0\}$
5. Both result in the same (13.25) loss value
6. However, y_1 is clearly predicting more score for the groundtruth class and y_2 predicts strongest to a wrong class

Classification

1. Dataset looks like $(x_n, y_n) \in \mathcal{R}^D \times \{1, 2, \ldots, C\}, n = 1, 2, \ldots, N$
2. Say $C = 3$, and we employ MSE loss on a sample with $y_n = 1$
3. One-hot encoding $\rightarrow \{0, 1, 0\}$
4. Consider two predictions $y_1 = \{-2.0, 1.5 - 3.0\}$ and $y_2 = \{0.5, -3.0, -3.0\}$
5. Both result in the same (13.25) loss value
6. However, y_1 is clearly predicting more score for the groundtruth class and y_2 predicts strongest to a wrong class
7. MSE in not suitable for classification

Classification Intuition

1. For each input $\mathbf{x}$ we have a target label y

Classification Intuition

1. For each input **x** we have a target label y
2. One-hot encoding converts y to a pmf ($\mathbf{p}$) (e.g., $y_n = 2 \rightarrow \{0, 0, 1, 0\}$)

Classification Intuition

1. For each input **x** we have a target label y
2. One-hot encoding converts y to a pmf ($\mathbf{p}$) (e.g., $y_n = 2 \rightarrow \{0, 0, 1, 0\}$)
3. Hence, the DNN's prediction should also be a pmf ($\mathbf{q}$)

Classification Intuition

1. For each input $\mathbf{x}$ we have a target label y
2. One-hot encoding converts y to a pmf ($\mathbf{p}$) (e.g., $y_n = 2 \rightarrow \{0, 0, 1, 0\}$)
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')

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 $h(x) = -log_2(P(x))$

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 $h(x) = -log_2(P(x))$
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. Skewed distribution has less entropy, uniform/balanced distribution has more entropy

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. 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. $H(p, q) = -\sum_i p_i \cdot log_2(q_i)$
3. Note that cross-entropy is not symmetric metric, i.e, $H(p, q) \neq H(q, p)$

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, $H(p, q) \neq H(q, p)$

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, $H(p, q) \neq H(q, p)$
4. Cross-entropy between a distribution and itself $(H(p, q))$ gives the entropy of the distribution $H(p)$

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

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 has a known label that 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 has a known label that is converted into a distribution with 1 and 0s (one-hot encoding)
3. A model predicts probability that sample belongs to each of the classes

Cross-entropy as a loss function

1. Widely used in classification problems (e.g. logistic regression, NNs)
2. Each label has a known label that is converted into a distribution with 1 and 0s (one-hot encoding)
3. A model predicts probability that sample belongs to each of the classes
4. Cross-entropy can be used to calculate the difference between the distributions

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

Softmax

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

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 $[0, 1]$)

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

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 $(q_c \approx 0)$

Colab Notebook: Backword()