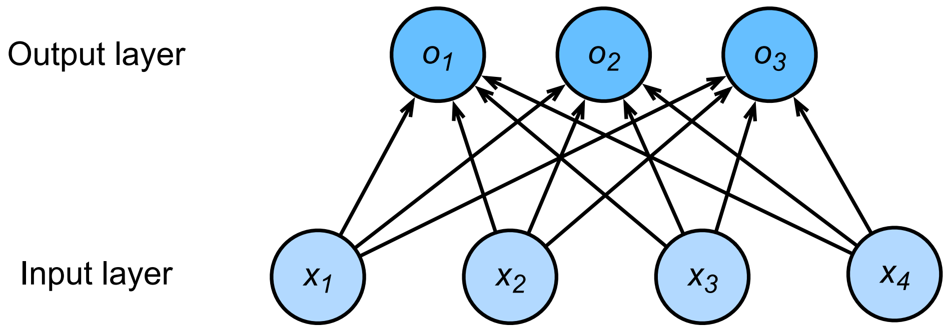
Linear Classification with Softmax

One-hot vector

- The class label is nominal (no order). So, how can be represent class 1,2,3 or a,b,c, or “cat”, “chicken”, “dog”?
- In classification, it is convenient to express a label (target) by the **one-hot vector**
  - Class 1 (or “chicken”) : 100
  - Class 2 (or “dog”) : 010
  - Class 3 (or “cat”) : 001
  - Not that the order class 1,2,3 is arbitrary
- If we build a model predicting next word given past words, we have as many classes as the size of vocabulary (say, 100k). In this case, a word is expressed a one-hot vector which contains 1 only in one place in the vector of dimension 100k, and 0 in all other places.

Linear Regression with Multiple Outputs

- Assume that the total number of classes are C
- We can simply extend the linear regression model to predict C outputs



$$\begin{aligned} o_1 &= x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1,\\ o_2 &= x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2,\\ o_3 &= x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3. \end{aligned}$$

- We want to make the output o_1 to represent the **probability** of class 1 to be the answer
  - (Cat, chicken, dog) = (0.2, 0.7, 0.1)

Softmax Operation

- This means that we need to normalize the output so that its sum becomes 1 and each output is nonnegative
- Softmax function does this

nonnegative

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \quad \text{where} \quad \hat{y}_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}.$$

Sum to 1

Loss Function for Classification

- **Cross-Entropy Loss:** Maximum-Likelihood for Classification

$$P(Y \mid X) = \prod_{i=1}^{n} P(y^{(i)} \mid x^{(i)}) \text{ and thus } -\log P(Y \mid X) = \sum_{i=1}^{n} -\log P(y^{(i)} \mid x^{(i)}).$$

- where

Predicted probability of class j

$$l = -\log P(y \mid x) = -\sum_j y_j \log \hat{y}_j.$$

Actual probability of class j

Cross-Entropy

$$l = -\log P(y \mid x) = -\sum_j y_j \log \hat{y}_j.$$

label $y_1 = 0$ $y_2 = 1$ $y_3 = 0$

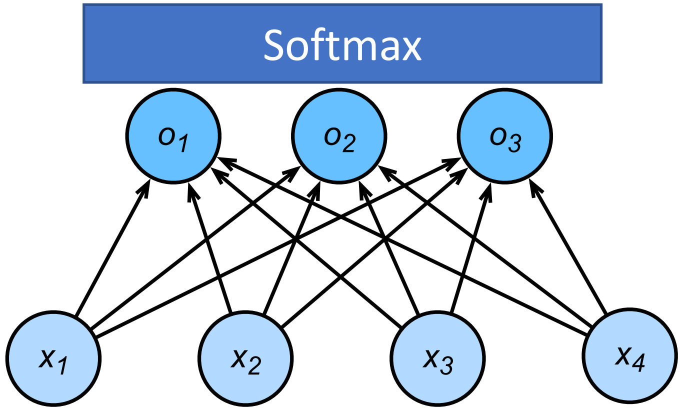
prediction $\hat{y}_1 = 0.12$ $\hat{y}_2 = 0.64$ $\hat{y}_3 = 0.24$

Cross Entropy

minimize

Output layer

Input layer



The Gradients of The Cross Entropy Loss

- The softmax function is a non-linear function (due to exp). Thus, we don’t have a close form solution. This means that we need use the gradient descent method.
- Compute the gradient of the cross-entropy loss

Kullback-Leibler Divergence

KL Divergence

- Softmax outputs a distribution over the class
- We can see one-hot encoding as a distribution where all mass is concentrated on one state
- An intuitive interpretation of cross entropy loss is to update the softmax distribution to minimize the distance between the two distributions
- Between two points in the Euclidean space, we can measure Euclidean distance. How can we measure the distance between two distributions?
- Kullback-Leibler (KL) Divergence is to do this

KL Divergence

- Definition

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

$$= \sum_d p(x_d) \log \frac{p(x_d)}{q(x_d)}$$

- KLD is
  - not symmetric
  - non-negative
  - zero if the two distribution is equivalent

KL Divergence

- Other useful form

$$KL(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

$$= \mathbb{E}_{p(x)} \left[\log \frac{p(x)}{q(x)} \right]$$

$$= -\int p(x) \log q(x)\, dx \; - H(p(x))$$

where H(p(x)) is entropy, $\mathrm{H}\big(\mathrm{p}(\mathrm{x})\big) = -\int p(x) \log p(x)\, dx$

KL Divergence and Cross-Entropy Loss

- What is KLD between softmax output and one-hot label
- Let's say one-hot label is p and softmax output q. Then,
- The KLD is

$$KL(p||q) = \sum_d p(y_d) \log \frac{p(y_d)}{q(y_d)}$$

$$= -\sum_d p(y_d) \log q(y_d) \ - H(p(y))$$

- The entropy term is independent to the model parameters, so we can ignore.
- Then, we only have the first-term which is exactly the cross-entropy loss.

MLE is equivalent to minimizing KL divergence

- Remember that the cross-entropy loss was defined simply as a negative log-likelihood of a discrete random variable
- And we saw that it is equivalent to minimizing a KL. Can this be generalized to arbitrary distributions (e.g., continuous)?
- Yes, we can use the same derivation used in the previous slide.
- That is, any maximum likelihood estimation is to minimize a KL divergence