Notations

Training data:

$$\{\ (x^{\{i\}}, y^{\{i\}}) \mid i = 1, \cdots, m\ \}$$

where$x^{\{i\}}$ is the input and $y^{\{i\}}$ is the corresponding target.

For example, for image classification problems, the input is an image

$$x^{\{i\}} \in \mathbb{R}^{h \times w \times 3}$$

the output (target) is a class label.

$$y^{\{i\}} \in \{1, 2, \cdots, C\}$$

Loss functions

- Loss function measures the inconsistency between the prediction and the ground truth. For one pair of sample (x, y):

$$\mathcal{L}(h(x),\ y)$$

- The overall loss calculated over the entire training set $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$

$$\mathcal{L}(W) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(h(x^{\{i\}}),\ y^{\{i\}})$$

- Learning is treated as optimizing parameters over the loss function

$$W^* = \arg\max_{W}\ \mathcal{L}(W)$$

Linear Regression

- Loss function for one pair of sample (x, y):

$$\mathcal{L}(h_\theta(x),\ y) = \frac{1}{2}\|\theta^T x - y\|^2$$

- The overall loss calculated over the entire training set $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^m$

$$\mathcal{L}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\|\theta^T x^{\{i\}} - y^{\{i\}}\|^2$$

- Learning is treated as optimizing parameters over the loss function

$$\theta^* = \arg\max_\theta\ \mathcal{L}(\theta)$$

Linear Regression - Gradient

- Loss function for one pair of sample (x, y):

$$\mathcal{L}(h_\theta(x),\ y) = \frac{1}{2}\|\theta^T x - y\|^2$$

- Gradient of the loss over one sample:

$$\frac{\partial \mathcal{L}}{\partial \theta} = (\theta^T x - y)x$$

- The overall loss calculated over the entire training set $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^m$

$$\mathcal{L}(\theta) = \frac{1}{2m}\sum_{i=1}^{m} \|\theta^T x^{\{i\}} - y^{\{i\}}\|^2$$

- Gradient of the loss over all samples:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{m}\sum_{i=1}^{m}(\theta^T x^{\{i\}} - y^{\{i\}})x^{\{i\}}$$

Logistic Regression

- Training set: $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$ where the label $y \in \{0, 1\}$:
- Model output:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

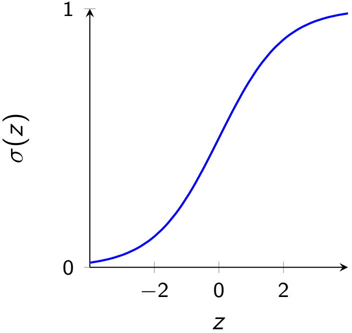
- Loss function for one pair of sample (x, y)

$$\begin{aligned}\mathcal{L}(h_\theta(x), y) &= \begin{cases} -\log h_\theta(x), & \text{if } y = 1 \\ -\log(1 - h_\theta(x)), & \text{if } y = 0 \end{cases} \\ &= -y \log h_\theta(x) - (1 - y) \log(1 - h_\theta(x))\end{aligned}$$

Activation Functions

Usually the activation function is chosen to be the logistic sigmoid:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



which is non-linear, monotonic and differentiable.

Logistic Regression - Gradient

- Given $x \in R^n$, the model outputs

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Loss function for one pair of sample (x, y)

$$\mathcal{L}(h_\theta(x), y) = -y \log h_\theta(x) - (1 - y) \log(1 - h_\theta(x))$$

- Gradient of the loss over one sample:

$$\frac{\partial \mathcal{L}}{\partial \theta} = (h_\theta(x) - y)x$$

Binary SVM

- Training set: $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$ where the label $y \in \{-1, 1\}$
- Model outputs: $h_\theta(x) = \theta^T x$
- Loss function for one pair of sample (x, y)

$$\begin{aligned}\mathcal{L}(h_\theta(x), y) &= \begin{cases} 1 - y\theta^T x, & \text{if } y\theta^T x < 1 \\ 0, & \text{if } y\theta^T x \geq 1 \end{cases} \\ &= \max\{0, 1 - y\theta^T x\}\end{aligned}$$

- The overall loss calculated over the entire training set

$$\mathcal{L}(\theta) = \frac{C}{m} \sum_{i=1}^{m} \max\{0, 1 - y^{\{i\}}\theta^T x^{\{i\}}\} + \frac{1}{2}\|\theta\|^2$$

Binary SVM - Gradient

- Given one input x, the model outputs

$$h_\theta(x) = \theta^T x$$

- Loss over one pair of sample (x, y) where $y \in \{-1, 1\}$,

$$\mathcal{L}(h_\theta(x), y) = \max\{0, 1 - y\theta^T x\}$$

- Gradient (more rigorously, subgradient) of the loss over one sample:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \begin{cases} -yx, & \text{if } y\theta^T x < 1 \\ 0, & \text{if } y\theta^T x \geq 1 \end{cases}$$

Multiclass SVM

- Training set: $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$ where the label $y \in \{1, 2, \cdots, C\}$:
- Model outputs a score given an input: $s = h_W(x) = Wx$
- Loss function for one pair of sample (x, y). First, find the predicted best class other than the ground truth:

$$\hat{y} = \arg\max_{j \neq y} s_j$$

Then calculate the hinge loss between scores of the ground truth class and the predicted best non-ground-truth class

$$\begin{aligned} \mathcal{L}(s, y) &= \begin{cases} 1 - s_y + s_{\hat{y}}, & \text{if } s_y - s_{\hat{y}} < 1 \\ 0, & \text{if } s_y - s_{\hat{y}} \geq 1 \end{cases} \\ &= \max\{0, 1 - s_y + s_{\hat{y}}\} \end{aligned}$$

- Or a relaxed form:

$$\mathcal{L}(s, y) = \sum_{j \neq y} \max\{0, 1 - s_y + s_j\}$$

Multiclass SVM

- Training set: $\{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$ where the label y is in $\{1, 2, \cdots, C\}$
- Loss function over one training sample (x, y)

$$\mathcal{L}(Wx, y) = \sum_{j \neq y} \max\{0, 1 - (Wx)_y + (Wx)_j\}$$

- Loss over all training samples

$$\mathcal{L}(W) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j \neq y^{\{i\}}} \max\{0, 1 - (Wx^{\{i\}})_{y^{\{i\}}} + (Wx^{\{i\}})_j\} + \lambda R(W)$$

where $R(W)$ is a regularization term, e.g., L1 or L2.

Multiclass SVM - Gradient

- For a given input x, the model outputs a vector of scores

$$s = h_W(x) = Wx$$

- The loss for one pair of sample (x, y) in the relaxed form:

$$\mathcal{L}(s, y) = \sum_{j \neq y} \max\{0, 1 - s_y + s_j\}$$

- Gradient (more rigorously, subgradient) of the loss over one sample:

$$\frac{\partial \mathcal{L}}{\partial s_j} = \begin{cases} -\sum_{k \neq y} I(s_y - s_k < 1), & \text{if } j = y \\ I(s_y - s_j < 1), & \text{if } j \neq y \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial s} \, x^T$$

Softmax

- A generalization of logistic regression to classification on more than 2 classes.
- Given input x, the model outputs the probability of assigning the label to each class

$$\mathbb{P}(Y = k|x) = \frac{e^{s_k}}{\sum_{j=1}^{C} e^{s_j}}$$

  where $s = Wx \in R^C$.
- Loss function for one pair of sample (x, y) is the negative log likelihood of the correct class

$$\begin{aligned}\mathcal{L}(s, x) &= -\log \mathbb{P}(Y = y|x) \\ &= -\log \frac{e^{s_y}}{\sum_{j=1}^{C} e^{s_j}}\end{aligned}$$

  also called cross-entropy loss.

Softmax - in vector form

- Given input x, the model outputs the probability of assigning the label to each class

$$\hat{y} = \frac{1}{\sum_{j=1}^{C} e^{s_j}} \begin{bmatrix} e^{s_1} \\ e^{s_2} \\ \vdots \\ e^{s_C} \end{bmatrix}$$

  where $s = Wx \in R^C$.
- Use one-hot encoding of the class label $y \in \mathbb{R}^C$
- Then the loss for one pair of sample (x, y) is

$$\mathcal{L}(\hat{y}, y) = -y^T \log \hat{y}$$

One-Hot Encoding

- One discrete feature with n values $\rightarrow$ n real values
- The i^{th} feature value sets the i^{th} input to 1 and others to 0
- Suppose the total number of discrete categories is 5. Then using one-hot encoding,

$$y = 2 \rightarrow y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad y = 5 \rightarrow y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Softmax - Gradient

- Given input x, the model outputs the probability of assigning the label to each class

$$\hat{y} = \frac{1}{\sum_{j=1}^{C} e^{s_j}} \begin{bmatrix} e^{s_1} \\ e^{s_2} \\ \vdots \\ e^{s_C} \end{bmatrix}$$

where $s = Wx \in R^C$.

- Use one-hot encoding of the class label $y \in \mathbb{R}^C$, the loss for one pair of sample (x, y) is

$$\mathcal{L}(\hat{y}, y) = -y^T \log \hat{y}$$

- Gradient of the loss over one sample

$$\frac{\partial \mathcal{L}}{\partial W} = (\hat{y} - y)\, x^T$$