Logistic Regression and Softmax Regression

Prof.Mingkui Tan

South China University of Technology
Southern Artificial Intelligence Laboratory(SAIL)

December 1, 2018

Content

Contents

Linear Classification and Regression

The linear signal:

$$s = \mathbf{w}^\top \mathbf{x}$$

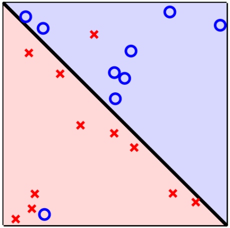


Figure: Linear Classification

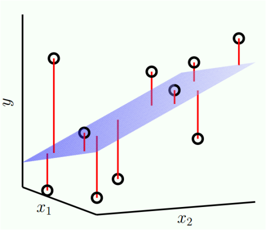


Figure: Linear Regression

Predicting a Probability

Will someone have a heart attack over the next year?

age	62 years
gender	male
blood sugar	120 mg/dL40,000
HDL	50
LDL	120
Mass	190 lbs
Height	5′ 10″
…	…

Classification: Yes/No
Logistic Regression: Likelihood of heart attack

$$h_{\mathbf{w}}(\mathbf{x}) = g\left(\sum_{i=1}^{m} w_i x_i\right) = g(\mathbf{w}^\top \mathbf{x})$$

Logistic function

Definition

$$g(z) = \frac{1}{1 + e^{-z}}$$

- The function is a continuous function.
- If $z \to +\infty$, then $g(z) \to 1$; If $z \to -\infty$, then $g(z) \to 0$.

Logistic function

Definition

$$g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

$$g(-z) = \frac{e^{-z}}{1+e^{-z}} = \frac{1}{1+e^{z}} = 1 - g(z)$$

The Data is Still Binary

$$\mathcal{D} = \{(\mathbf{x}_1, y_1 = \pm 1), ..., (\mathbf{x}_n, y_n = \pm 1)\}$$

- $\mathbf{x}_n \leftarrow$ a persons health information.
- $y_n = \pm 1 \leftarrow$ did they have a heart attack or not.
- We cannot measure a probability.
- We can only see the occurence of an event and try to infer a probability.

The Target Function is Inherently Noisy

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbb{P}[y = +1|\mathbf{x}]$$

The data is generated from a noisy target function:

$$P(y|\mathbf{x}) = \begin{cases} h_{\mathbf{w}}(\mathbf{x}) & y = 1 \\ 1 - h_{\mathbf{w}}(\mathbf{x}) & y = -1 \end{cases}$$

What Makes an h Good?

Fitting the data means finding a good h

$$\text{h is good if: } \begin{cases} h_{\mathbf{w}}(\mathbf{x}) \approx 1 & y = 1 \\ h_{\mathbf{w}}(\mathbf{x}) \approx 0 & y = -1 \end{cases}$$

A simple error measure that captures this:

$$\mathbf{E}_{in}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(h_{\mathbf{w}}(\mathbf{x}_i) - \frac{1}{2}(1 + y_i)\right)^2$$

Not very convenient (hard to minimize).

The Cross Entropy Error Measure

$$\mathbf{E}_{in}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i \cdot \mathbf{w}^\top \mathbf{x}})$$

- It is based on an intuitive probabilistic interpretation of h.
- It is very convenient and mathematically friendly (easy to minimize).

The Probabilistic Interpretation

Suppose that $h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x})$ closely captures $\mathbb{P}[+1|\mathbf{x}]$:

$$P(y|\mathbf{x}) = \begin{cases} g(\mathbf{w}^\top \mathbf{x}) & y = 1 \\ 1 - g(\mathbf{w}^\top \mathbf{x}) & y = -1 \end{cases}$$

The Probabilistic Interpretation

So, if $h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x})$ closely captures $\mathbb{P}[+1|\mathbf{x}]$:

$$P(y|\mathbf{x}) = \begin{cases} g(+\mathbf{w}^\top \mathbf{x}) & y = 1 \\ 1 - g(+\mathbf{w}^\top \mathbf{x}) = g(-\mathbf{w}^\top \mathbf{x}) & y = -1 \end{cases}$$

...or, more compactly,

$$P(y|\mathbf{x}) = g(y \cdot \mathbf{w}^\top \mathbf{x})$$

The Likelihood

$$P(y|\mathbf{x}) = g(y \cdot \mathbf{w}^\top \mathbf{x})$$

Recall: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are independently generated.

Likelihood:

The probability of getting the $y_1, \ldots, y_n$ in $\mathcal{D}$ from the corresponding $\mathbf{x}_1, \ldots, \mathbf{x}_n$:

$$P(y_1, \ldots, y_n | \mathbf{x}_1, \ldots, \mathbf{x}_n) = \prod_{i=1}^{n} P(y_i | \mathbf{x}_i)$$

Maximizing The Likelihood

$$
\begin{aligned}
\max \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) \Leftrightarrow \max \log \left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i) \right)
&\equiv \max \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \\
&\Leftrightarrow \min -\frac{1}{n} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \\
&\equiv \min \frac{1}{n} \sum_{i=1}^{n} \log \frac{1}{P(y_i|\mathbf{x}_i)} \\
&\equiv \min \frac{1}{n} \sum_{i=1}^{n} \log \frac{1}{g(y_i \cdot \mathbf{w}^\top \mathbf{x}_i)} \\
&\equiv \min \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \cdot \mathbf{w}^\top \mathbf{x}_i}) = \min E_{in}(\mathbf{w})
\end{aligned}
$$

Regularization

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i\cdot\mathbf{w}^\top\mathbf{x}_i}) + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

Small values for parameters $w_0, w_1, ..., w_{m-1}$

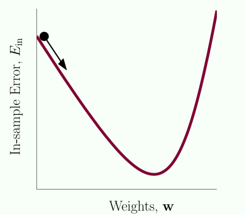
- "Simpler" model
- Less prone to overfitting

Regularization parameter λ

- Trade off between fitting the training set well and keeping the model relatively simple

Finding The Best Weights

Use the Gradient Descent



Minimize $E_{in}(\mathbf{w})$ by repeated gradient steps:

- Compute gradient of loss with respect to parameters $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$
- Update parameters with rate η

$$\mathbf{w}' \rightarrow \mathbf{w} - \eta \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = (1 - \eta\lambda)\mathbf{w} + \eta \frac{1}{n} \sum_{i=1}^{n} \frac{y_i \mathbf{x}_i}{1 + e^{y_i \cdot \mathbf{w}^\top \mathbf{x}_i}}$$

Logistic Regression: $y_i \in \{0, 1\}$

Assume that the labels are binary: $y_i \in \{0, 1\}$

$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{w}^\top \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}}$$

Probability:

$$p = \begin{cases} h_{\mathbf{w}}(\mathbf{x}_i) & y_i = 1 \\ 1 - h_{\mathbf{w}}(\mathbf{x}_i) & y_i = 0 \end{cases}$$

Log-likehood loss function:

$$
\begin{aligned}
\max \prod_{i=1}^{n} P(y_i|\mathbf{x}_i) &\Leftrightarrow \max \log \left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i) \right) \\
&\equiv \max \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \\
&\Leftrightarrow \min -\frac{1}{n} \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) \\
&\equiv \min -\frac{1}{n} \sum_{i=1}^{n} \log h_{\mathbf{w}}(\mathbf{x}_i)^{y_i} \cdot (1 - h_{\mathbf{w}}(\mathbf{x}_i))^{(1-y_i)}
\end{aligned}
$$

$$
J(\mathbf{w}) = -\frac{1}{n} \left[\sum_{i=1}^{n} y_i \log\, h_{\mathbf{w}}(\mathbf{x}_i) + (1 - y_i) \log\, (1 - h_{\mathbf{w}}(\mathbf{x}_i)) \right]
$$

The Gradient of The Loss Function

For a sample:

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -\frac{1}{\partial \mathbf{w}} \cdot \partial \left[y \cdot \log\, h_{\mathbf{w}}(\mathbf{x}) + (1-y) \log\, (1 - h_{\mathbf{w}}(\mathbf{x})) \right]$$
$$= -y \cdot \frac{1}{h_{\mathbf{w}}(\mathbf{x})} \cdot \frac{\partial h_{\mathbf{w}}(\mathbf{x})}{\partial \mathbf{w}} + (1-y) \cdot \frac{1}{1 - h_{\mathbf{w}}(\mathbf{x})} \frac{\partial h_{\mathbf{w}}(\mathbf{x})}{\partial \mathbf{w}}$$

Note:

$$g(z) = \frac{1}{1+e^{-z}}, \;\; g'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = g(z)\left[1 - g(z)\right]$$

The Gradient of The Loss Function

For a sample:

$$
\begin{aligned}
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= -y \cdot \frac{1}{h_{\mathbf{w}}(\mathbf{x})} \cdot \frac{\partial h_{\mathbf{w}}(\mathbf{x})}{\partial \mathbf{w}} + (1-y) \cdot \frac{1}{1-h_{\mathbf{w}}(\mathbf{x})} \frac{\partial h_{\mathbf{w}}(\mathbf{x})}{\partial \mathbf{w}} \\
&= -y \cdot \frac{1}{h_{\mathbf{w}}(\mathbf{x})} \cdot \frac{\partial g\left(\mathbf{w}^{\top}\mathbf{x}\right)}{\partial \mathbf{w}} + (1-y) \cdot \frac{1}{1-h_{\mathbf{w}}(\mathbf{x})} \frac{\partial g\left(\mathbf{w}^{\top}\mathbf{x}\right)}{\partial \mathbf{w}} \\
&= \left(-\frac{\mathbf{x}y}{h_{\mathbf{w}}(\mathbf{x})} + \frac{\mathbf{x}(1-y)}{1-h_{\mathbf{w}}(\mathbf{x})}\right) \cdot g\left(\mathbf{w}^{\top}\mathbf{x}\right) \cdot \left[1-g\left(\mathbf{w}^{\top}\mathbf{x}\right)\right] \\
&= \left(h_{\mathbf{w}}(\mathbf{x})-y\right)\mathbf{x}
\end{aligned}
$$

Use The Gradient Descent to Get $\mathbf{w}$

For a sample:

$$\frac{\partial J\left(\mathbf{w}\right)}{\partial \mathbf{w}} = \left(h_{\mathbf{w}}\left(\mathbf{x}\right) - y\right)\mathbf{x}$$

$$\mathbf{w} := \mathbf{w} - \alpha\left(h_{\mathbf{w}}\left(\mathbf{x}\right) - y\right)\mathbf{x}$$

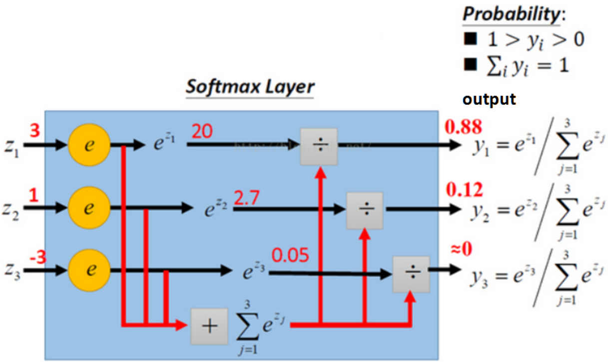
For all samples:

$$\frac{\partial J\left(\mathbf{w}\right)}{\partial \mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n}\left(h_{\mathbf{w}}\left(\mathbf{x}_i\right) - y\right)\mathbf{x}_i$$

$$\mathbf{w} := \mathbf{w} - \frac{1}{n}\sum_{i=1}^{n}\alpha\left(h_{\mathbf{w}}\left(\mathbf{x}_i\right) - y_i\right)\mathbf{x}_i$$

Contents

Softmax Regression
Multi-class classification



$$p\left(y_i = j \mid \mathbf{x}_i; \mathbf{w}\right) = \frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}$$

Softmax Regression

Multi-class classification

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{bmatrix} p(y_i = 1|\mathbf{x}_i; \mathbf{w}) \\ p(y_i = 2|\mathbf{x}_i; \mathbf{w}) \\ \vdots \\ p(y_i = k|\mathbf{x}_i; \mathbf{w}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\mathbf{w}_j^\top \mathbf{x}_i}} \begin{bmatrix} e^{\mathbf{w}_1^\top \mathbf{x}_i} \\ e^{\mathbf{w}_2^\top \mathbf{x}_i} \\ \vdots \\ e^{\mathbf{w}_k^\top \mathbf{x}_i} \end{bmatrix}$$

- Multi-class classification: $y \in \{1, 2, ..., k\}$.
- $p\left(y = j \mid \mathbf{x}\right)$ represents the probability of the class label.
- The term $\frac{1}{\sum_{j=1}^{k} e^{\mathbf{w}_j^\top \mathbf{x}^{(i)}}}$ normalizes the distribution, so the elements sum to 1.

Softmax function

Logistic function vs Softmax function

When the number of the classes is two:

$$\begin{aligned}
h_{\mathbf{w}}(\mathbf{x}) &= \begin{bmatrix} p\left(y=0 \mid \mathbf{x}; \mathbf{w}\right) \\ p\left(y=1 \mid \mathbf{x}; \mathbf{w}\right) \end{bmatrix} \\
&= \frac{1}{e^{\mathbf{w}_0^\top \mathbf{x}} + e^{\mathbf{w}_1^\top \mathbf{x}}} \begin{bmatrix} e^{\mathbf{w}_0^\top \mathbf{x}} \\ e^{\mathbf{w}_1^\top \mathbf{x}} \end{bmatrix} \\
&= \frac{1}{e^{(\mathbf{w}_0-\mathbf{w}_1)^\top \mathbf{x}} + e^{(\mathbf{w}_1-\mathbf{w}_1)^\top \mathbf{x}}} \begin{bmatrix} e^{(\mathbf{w}_0-\mathbf{w}_1)^\top \mathbf{x}} \\ e^{(\mathbf{w}_1-\mathbf{w}_1)^\top \mathbf{x}} \end{bmatrix} \\
&= \frac{1}{e^{(\mathbf{w}_0-\mathbf{w}_1)^\top \mathbf{x}} + e^{(\mathbf{0})^\top \mathbf{x}}} \begin{bmatrix} e^{(\mathbf{w}_0-\mathbf{w}_1)^\top \mathbf{x}} \\ e^{(\mathbf{0})^\top \mathbf{x}} \end{bmatrix}
\end{aligned}$$

Let $-\mathbf{w} = \mathbf{w}_0 - \mathbf{w}_1$

Softmax function

Logistic function vs Softmax function

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} \begin{bmatrix} e^{-\mathbf{w}^\top \mathbf{x}} \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 - \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} \\ \\ \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}}} \end{bmatrix}$$

- Softmax regression is a generalization of logistic regression.

Softmax function

Loss function

Represent $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_k \end{bmatrix}$, the softmax cost function

$$J(\mathbf{w}) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{I}\{y_i = j\}\log\frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}\right]$$

- $\mathbb{I}\{\cdot\}$ is the indicator function.
- $\mathbb{I}$\{a true statement\}=1.
- $\mathbb{I}$\{a false statement\}=0.

The logistic regression cost function could also have been written:

$$J\left(\mathbf{w}\right) = -\frac{1}{n}\left[\sum_{i=1}^{n} y_i \log\, h_{\mathbf{w}}(\mathbf{x}_i) + (1-y_i)\log\,(1-h_{\mathbf{w}}(\mathbf{x}_i))\right]$$

$$= -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=0}^{1}\mathbb{I}\left\{y_i = j\right\}\log P(y_i = j|\mathbf{x}_i;\mathbf{w})\right]$$

Softmax function

Derivation

For $\mathbf{w}_j \ (j = 1, ..., k)$

$$
\begin{aligned}
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}_j} &= \frac{\partial \left\{-\frac{1}{n} \cdot \left[\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}\{y_i = j\} \, log \frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}\right]\right\}}{\partial \mathbf{w}_j} \\
&= -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial \sum_{j=1}^{k} \mathbb{I}\{y_i = j\} \left(\log e^{\mathbf{w}_j^\top \mathbf{x}_i} - \log \sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}\right)}{\partial \mathbf{w}_j} \\
&= -\frac{1}{n} \sum_{i=1}^{n} \left[\mathbb{I}\{y_i = j\} \mathbf{x}_i - \frac{1}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}} \cdot \frac{\partial \sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}{\partial \mathbf{w}_j}\right] \\
&= -\frac{1}{n} \sum_{i=1}^{n} \left[\mathbb{I}\{y_i = j\} \mathbf{x}_i - \frac{\mathbf{x}_i \cdot e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}\right] \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(p(y_i = j \mid \mathbf{x}_i; \mathbf{w}) - \mathbb{I}\{y_i = j\}\right) \mathbf{x}_i
\end{aligned}
$$

Softmax function

Properties

Softmax function has a redundant set of parameters.

$$p\left(y_i = j \mid \mathbf{x}_i; \mathbf{w}\right) = \frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}$$

$$= \frac{e^{\mathbf{w}_j^\top \mathbf{x}_i} \div e^{\varphi^\top \mathbf{x}_i}}{\sum_{l=1}^{k}\left(e^{\mathbf{w}_l^\top \mathbf{x}_i} \div e^{\varphi^\top \mathbf{x}_i}\right)}$$

$$= \frac{e^{\left(\mathbf{w}_j - \varphi\right)^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\left(\mathbf{w}_l - \varphi\right)^\top \mathbf{x}_i}}$$

- Subtract φ from every $\mathbf{w}_j$ does not affect the hypothesis predictions

Softmax function
Loss function

The cost function $J(\mathbf{w})$ is minimized by some setting of the parameters $(\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_k)$, then it is also minimized by $(\mathbf{w}_1 - \varphi, \mathbf{w}_2 - \varphi, \ldots \mathbf{w}_k - \varphi)$ for any value of φ.

- However using the weight decay method, the minimizer of $J(\mathbf{w})$ is unique.

$$J(\mathbf{w}) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{I}\{y_i = j\}\log\frac{e^{\mathbf{w}_j^\top \mathbf{x}_i}}{\sum_{l=1}^{k} e^{\mathbf{w}_l^\top \mathbf{x}_i}}\right] + \frac{\lambda}{2}||\mathbf{w}||_2^2$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}_j} = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{x}_i\left(p\left(y_i = j \mid \mathbf{x}_i; \mathbf{w}\right) - \mathbb{I}\{y_i = j\}\right)\right] + \lambda \mathbf{w}_j$$

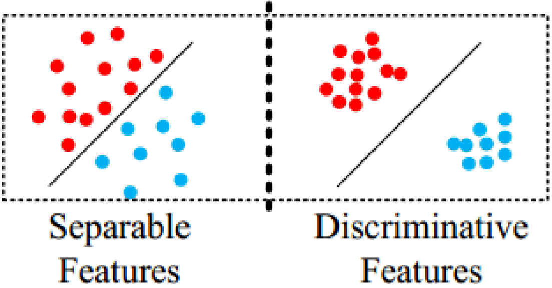
Contents

Two variants of the softmax loss

- Large-Margin Softmax Loss
- Angular Softmax Loss

Motivation

- Learn a discriminative features

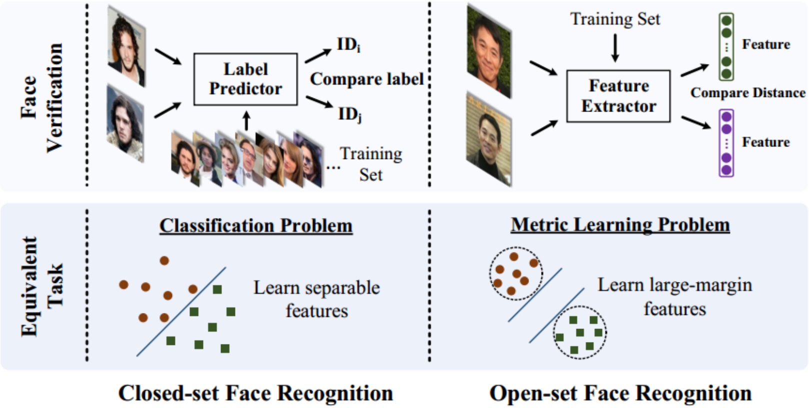


Motivation

Closed-set and open-set face recognition

Motivation

Closed-set and open-set face recognition



Softmax loss

Given input feature x_i with the label y_i, the softmax loss function is:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\frac{e^{f_{y_i}}}{\sum_j e^{f_j}},$$

- f_j denotes the j-th element of the vector of class scores f
- N is the number of training data

Softmax Loss

$$f_{y_i} = W_{y_i}^T x_i = \|W_{y_i}\| \|x_i\| \cos(\theta_j)$$

$$L_i = -\log\left(\frac{e^{\|W_{y_i}\| \|x_i\| \cos(\theta_{y_i})}}{\sum_j e^{\|W_j\| \|x_i\| \cos(\theta_j)}}\right)$$

- θ_j $(0 \leq \theta_j \leq \pi)$ is the angle between the vector W_j and x_i

Large-Margin Softmax Loss [1]

Consider the binary classification and we have a sample x from class 1.

- Original softmax

$$\|W_1\|\|x\|\cos(\theta_1) > \|W_2\|\|x\|cos(\theta_2)$$

- Large-Margin softmax

$$\|W_1\|\|x\|\cos(m\theta_1) > \|W_2\|\|x\|cos(\theta_2)\ (0 \le \theta_1 \le \frac{\pi}{m})$$

Large-Margin Softmax Loss

Large-Margin Softmax Loss:

$$L_i = -\log\left(\frac{e^{\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i})}}{e^{\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i})} + \sum_{j\neq y_i} e^{\|W_j\|\|x_i\|\cos(\theta_j)}}\right)$$

$$\psi(\theta) = \begin{cases} cos(m\theta), & 0 \leq \theta \leq \frac{\pi}{m} \\ \mathcal{D}(\theta), & \frac{\pi}{m} < \theta \leq \pi \end{cases}$$

Large-Margin Softmax Loss

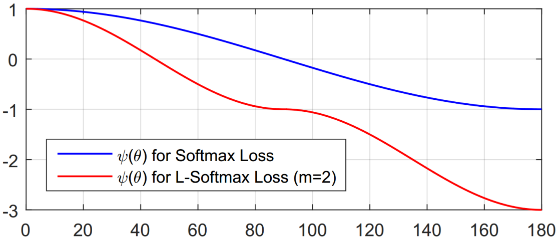


Figure: $\psi(\theta)$ for softmax loss and L-Softmax loss

- Construct a specific $\psi(\theta)$:

$$\psi(\theta) = (-1)^k \cos(m\theta) - 2k, \;\; \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}],$$

where $k \in [0, m-1]$ and k is an integer

Large-Margin Softmax Loss

Replace $\cos(\theta_j)$ with

$$\frac{W_j^T x_i}{\|W_j\|\|x_i\|}$$

Replace $\cos(m\theta_{y_i})$ with

$$\begin{aligned}\cos(m\theta_{y_i}) = & C_m^0 \cos^m(\theta_{y_i}) - C_m^2 \cos^{m-2}(\theta_{y_i})(1-\cos^2(\theta_{y_i})) \\ & + C_m^4 \cos^{m-4}(\theta_{y_i})(1-\cos^2(\theta_{y_i}))^2 + \cdots \\ & (-1)^n C_m^{2n} \cos^{m-2n}(\theta_{y_i})(1-\cos^2(\theta_{y_i}))^n + \cdots\end{aligned}$$

Large-Margin Softmax Loss

$$
\begin{aligned}
f_{y_i} =& (-1)^k \cdot \|W_{y_i}\|\|x_i\| \cos(m\theta_i) - 2k \cdot \|W_{y_i}\|\|x_i\| \\
=& (-1)^k \cdot \|W_{y_i}\|\|x_i\| \left(C_m^0 \left(\frac{W_{y_i}^T x_i}{\|W_{y_i}\|\|x_i\|}\right)^m - \right. \\
& \left. C_m^2 \left(\frac{W_{y_i}^T x_i}{\|W_{y_i}\|\|x_i\|}\right)^{m-2} \left(1 - \left(\frac{W_{y_i}^T x_i}{\|W_{y_i}\|\|x_i\|}\right)^2\right) + \cdots \right) \\
& - 2k \cdot \|W_{y_i}\|\|x_i\|,
\end{aligned}
$$

where $\frac{W_{y_i}^T x_i}{\|W_{y_i}\|\|x_i\|} \in [\cos(\frac{k\pi}{m}), \cos(\frac{(k+1)\pi}{m})]$ and $k \in [0, m-1]$

Large-Margin Softmax Loss
Optimization

$$
\begin{aligned}
\frac{\partial f_{y_i}}{\partial x_i} = (-1)^k \cdot \Bigg(& C_m^0 \Big(\frac{m(W_{y_i}^T x_i)^{m-1} W_{y_i}}{(\|W_{y_i}\| \|x_i\|)^{m-1}} \Big) - \\
& C_m^0 \Big(\frac{(m-1)(W_{y_i}^T x_i)^m x_i}{\|W_{y_i}\|^{m-1} \|x_i\|^{m+1}} \Big) - C_m^2 \Big(\frac{(m-2)(W_{y_i}^T x_i)^{m-3} W_{y_i}}{(\|W_{y_i}\| \|x_i\|)^{m-3}} \Big) \\
& + C_m^2 \Big(\frac{(m-3)(W_{y_i}^T x_i)^{m-2} x_i}{\|W_{y_i}\|^{m-3} \|x_i\|^{m-1}} + C_m^2 \Big(\frac{m(W_{y_i}^T x_i)^{m-1} W_{y_i}}{(\|W_{y_i}\| \|x_i\|)^{m-1}} \Big) \\
& - C_m^2 \Big(\frac{(m-1)(W_{y_i}^T x_i)^m x_i}{\|W_{y_i}\|^{m-1} \|x_i\|^{m+1}} + \cdots \Bigg) - 2k \cdot \frac{\|W_{y_i}\| x_i}{\|x_i\|}
\end{aligned}
$$

Large-Margin Softmax Loss

Optimization

$$
\begin{aligned}
\frac{\partial f_{y_i}}{\partial W_{y_i}} = (-1)^k \cdot \Bigg(C_m^0 \frac{m(W_{y_i}^T x_i)^{m-1} x_i}{(\|W_{y_i}\| \|x_i\|)^{m-1}} - \\
C_m^0 \frac{(m-1)(W_{y_i}^T x_i)^m W_{y_i}}{\|W_{y_i}\|^{m+1} \|x_i\|^{m-1}} - C_m^2 \frac{(m-2)(W_{y_i}^T x_i)^{m-3} x_i}{(\|W_{y_i}\| \|x_i\|)^{m-3}} \\
+ C_m^2 \frac{(m-3)(W_{y_i}^T x_i)^{m-2} W_{y_i}}{\|W_{y_i}\|^{m-1} \|x_i\|^{m-3}} + C_m^2 \frac{m(W_{y_i}^T x_i)^{m-1} x_i}{(\|W_{y_i}\| \|x_i\|)^{m-1}} \\
- C_m^2 \frac{(m-1)(W_{y_i}^T x_i)^m W_{y_i}}{\|W_{y_i}\|^{m+1} \|x_i\|^{m-1}} + \cdots \Bigg) - 2k \cdot \frac{\|x_i\| W_{y_i}}{\|W_{y_i}\|}
\end{aligned}
$$

Geometric Interpretation

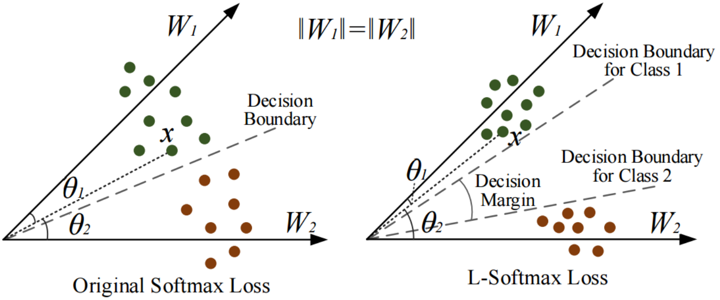


Figure: Example of Geometric Interpretation when $\|W_1\| = \|W_2\|$

Geometric Interpretation

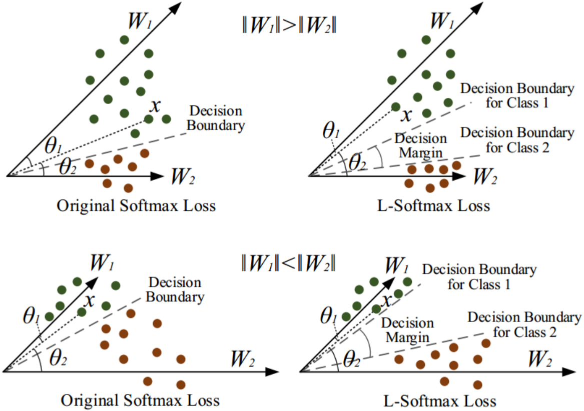


Figure: Examples of Geometric Interpretation when $\|W_1\| > \|W_2\|$ and $\|W_1\| < \|W_2\|$

Two variants of the softmax loss

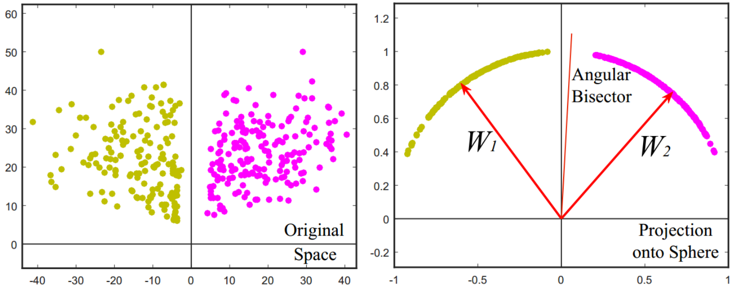
- Large-Margin Softmax Loss
- Angular Softmax Loss (A-Softmax Loss)

Modified Softmax Loss Function

- Normalize $\|W_j\| = 1, \forall j$ in each iteration

$$L_{\text{modified}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|x_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|x_i\|\cos(\theta_j,i)}}\right)$$

Modified Softmax Loss Function



- Learn a 2-D features on a subset of CASIA face dataset

A-Softmax Loss [2]

Consider the binary classification and we have a sample x from class 1

- Modified softmax loss need

$$\|x\| \cos(\theta_1) > \|x\| cos(\theta_2)$$

- A-Softmax loss need

$$\|x\| \cos(m\theta_1) > \|x\| cos(\theta_2) \ (0 \le \theta_1 \le \frac{\pi}{m})$$

A-Softmax Loss

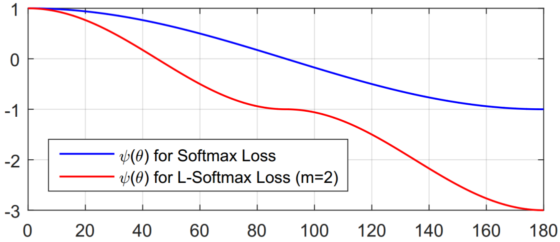
$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|x_i\|\cos(m\theta_{y_i,i})}}{e^{\|x_i\|\cos(m\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|x_i\|\cos(\theta_j,i)}}\right),$$

where $\theta_{y_i,i}$ has to be in the range of $[0, \frac{\pi}{m}]$

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|x_i\|\psi(\theta_{y_i,i})}}{e^{\|x_i\|\psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\|x_i\|\cos(\theta_{j,i})}}\right)$$

$$\psi(\theta) = \begin{cases} cos(m\theta), & 0 \le \theta \le \frac{\pi}{m} \\ \mathcal{D}(\theta), & \frac{\pi}{m} < \theta \le \pi \end{cases}$$

A-Softmax Loss

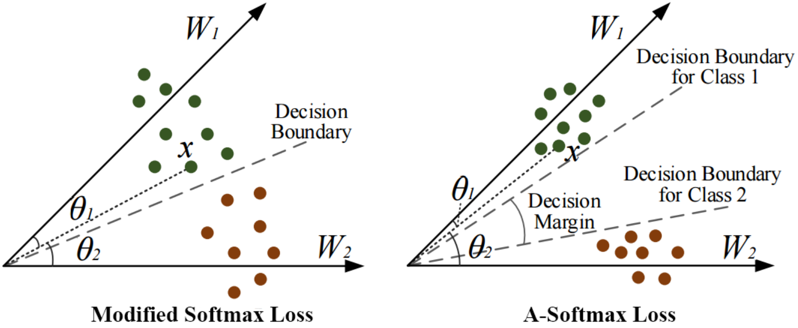


- Construct a specific $\psi(\theta)$:

$$\psi(\theta) = (-1)^k \cos(m\theta) - 2k, \ \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}],$$

where $k \in [0, m-1]$ and k is an integer

A-Softmax Loss
Geometric Interpretation

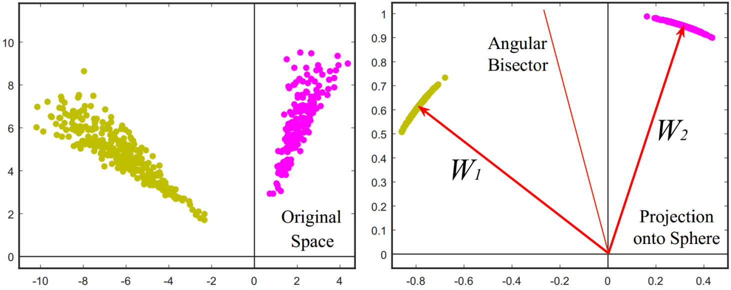


A-Softmax Loss
Decision Boundary

Loss Function	Decision Boundary
Softmax Loss	$(\boldsymbol{W}_1 - \boldsymbol{W}_2)\boldsymbol{x} + b_1 - b_2 = 0$
Modified Softmax Loss	$\|\boldsymbol{x}\|(\cos\theta_1 - \cos\theta_2) = 0$
A-Softmax Loss	$\|\boldsymbol{x}\|(\cos m\theta_1 - \cos\theta_2) = 0$ for class 1 $\|\boldsymbol{x}\|(\cos\theta_1 - \cos m\theta_2) = 0$ for class 2

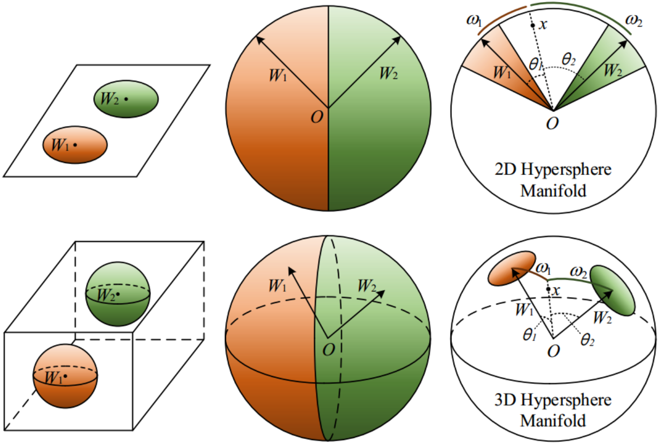
- θ_i is the angle between W_i and x

A-Softmax Loss



- Learn a 2-D features on a subset of CASIA face dataset

Hypersphere Interpretation



Euclidean Margin Loss Modified Softmax Loss A-Softmax Loss (m⩾2)

References

[1] Liu W, Wen Y, Yu Z, et al. Large-Margin Softmax Loss for Convolutional Neural Networks[C] ICML. 2016: 507-516.
[2] Liu W, Wen Y, Yu Z, et al. Sphereface: Deep hypersphere embedding for face recognition[C] The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, 1: 1.

THANK YOU!