From Logistic Regression to Neural Networks (Part 1)

CS114B Lab 4

Kenneth Lai

February 26, 2021

Logistic Regression

- Documents are characterized by features

Logistic Regression

- Documents are characterized by features
  - No independence assumptions

Logistic Regression

- Documents are characterized by features
  - No independence assumptions
- For each feature j:
  - Value x_j
  - Weight w_j

Logistic Regression

- Documents are characterized by features
  - No independence assumptions
- For each feature j:
  - Value x_j
  - Weight w_j
- Bias term b

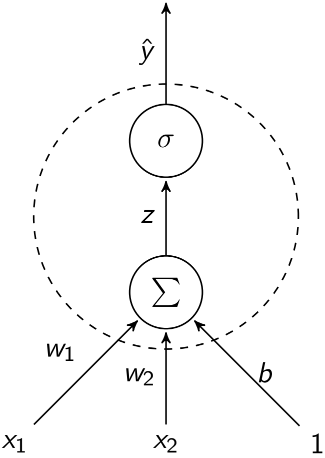
Logistic Regression

- Documents are characterized by features
  - No independence assumptions
- For each feature j:
  - Value x_j
  - Weight w_j
- Bias term b
- "Score" (log-odds) $z = \left(\sum_{j=1}^{n} w_j x_j\right) + b = \mathbf{w} \cdot \mathbf{x} + b$

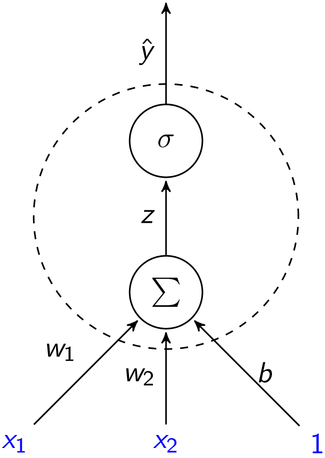
Logistic Regression

- Documents are characterized by features
  - No independence assumptions
- For each feature j:
  - Value x_j
  - Weight w_j
- Bias term b
- "Score" (log-odds) $z = \left(\sum_{j=1}^{n} w_j x_j \right) + b = \mathbf{w} \cdot \mathbf{x} + b$
- Logistic function $\sigma(z) = \frac{1}{1 + e^{-z}} = \hat{y}$

Graphical Representation of Logistic Regression

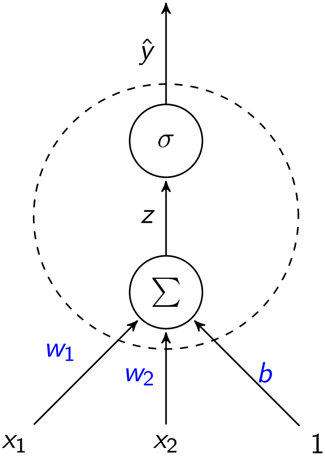


Graphical Representation of Logistic Regression



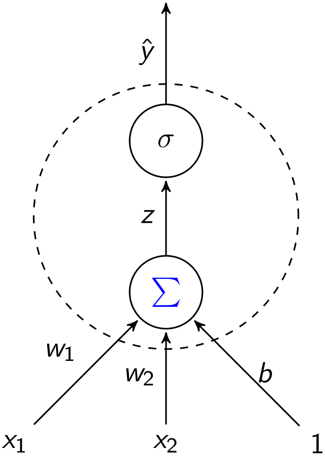
- Inputs (and dummy feature 1)

Graphical Representation of Logistic Regression



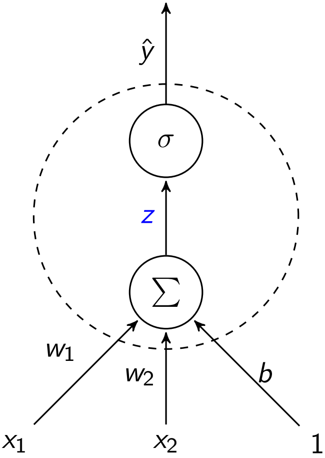
- Weights (and bias term)

Graphical Representation of Logistic Regression



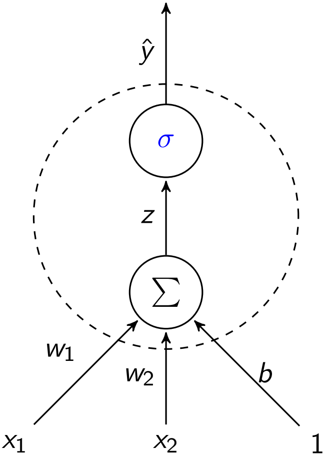
- Sum function $\sum$

Graphical Representation of Logistic Regression



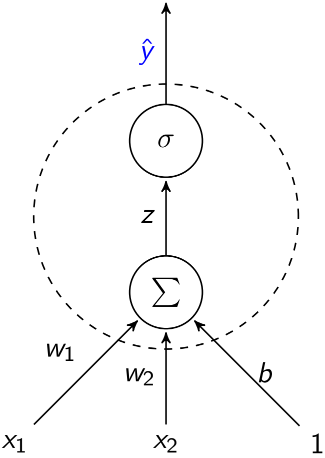
- "Score"

Graphical Representation of Logistic Regression



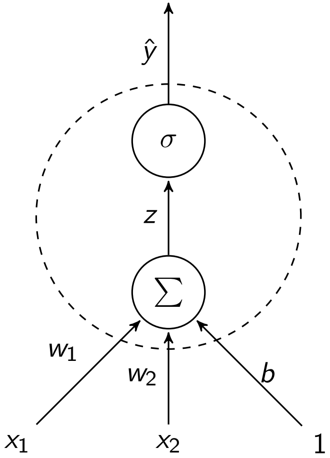
- Logistic function σ

Graphical Representation of Logistic Regression

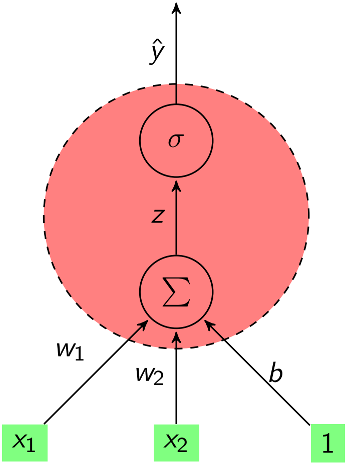


- Output

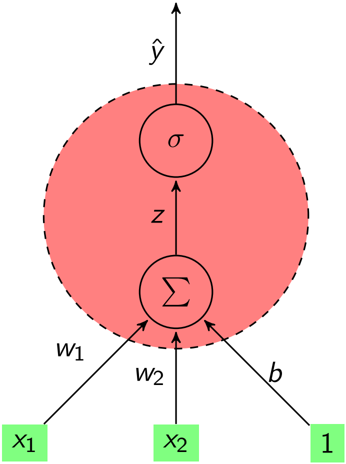
Graphical Representation of Logistic Regression



Graphical Representation of Logistic Regression

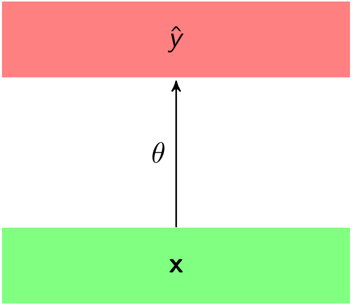


Graphical Representation of Logistic Regression



- Output
- Inputs

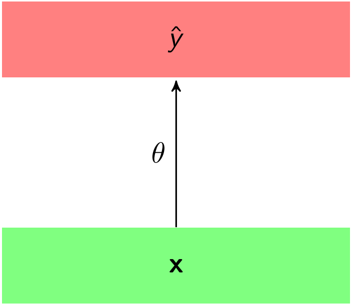
Graphical Representation of Logistic Regression



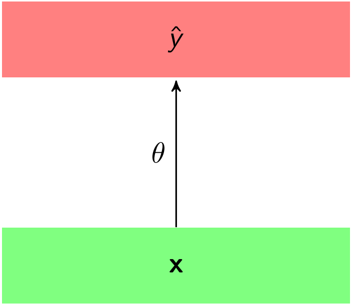
- Output
- Inputs

Graphical Representation of Logistic Regression

- Output
- Inputs
- $\hat{y} = \sigma(\mathbf{x} \cdot \theta)$

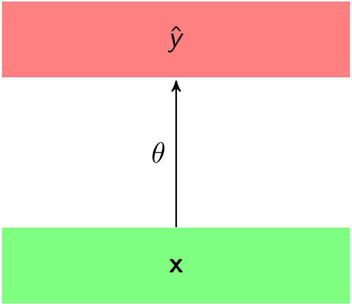


Graphical Representation of Logistic Regression



- Output
- Inputs
- $\hat{y} = \sigma(\mathbf{x} \cdot \theta)$
  - We will assume that the dummy feature 1 is part of $\mathbf{x}$

Graphical Representation of Logistic Regression



- Output
- Inputs
- $\hat{y} = \sigma(\mathbf{x} \cdot \theta)$
  - We will assume that the dummy feature 1 is part of $\mathbf{x}$
  - Why $\mathbf{x} \cdot \theta$ (rather than $\theta \cdot \mathbf{x}$)?

(Mini-)Batch Training

- Let $\mathbf{x}$ consist of the feature vectors $\mathbf{x}^{(i)}$ for each document i in the (mini-)batch of size m, stacked on top of each other

(Mini-)Batch Training

- Let $\mathbf{x}$ consist of the feature vectors $\mathbf{x}^{(i)}$ for each document i in the (mini-)batch of size m, stacked on top of each other
- $\mathbf{x} \cdot \theta = \begin{bmatrix} x_1^{(1)} & \dots & x_n^{(1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{(m)} & \dots & x_n^{(m)} & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)} \cdot \mathbf{w} + b \\ \vdots \\ \mathbf{x}^{(m)} \cdot \mathbf{w} + b \end{bmatrix}$

(Mini-)Batch Training

- Let $\mathbf{x}$ consist of the feature vectors $\mathbf{x}^{(i)}$ for each document i in the (mini-)batch of size m, stacked on top of each other
- $\mathbf{x} \cdot \theta = \begin{bmatrix} x_1^{(1)} & \dots & x_n^{(1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{(m)} & \dots & x_n^{(m)} & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{x}^{(1)} \cdot \mathbf{w} + b \\ \vdots \\ \mathbf{x}^{(m)} \cdot \mathbf{w} + b \end{bmatrix}$
- $\sigma(\mathbf{x} \cdot \theta) = \begin{bmatrix} \sigma(\mathbf{x}^{(1)} \cdot \mathbf{w} + b) \\ \vdots \\ \sigma(\mathbf{x}^{(m)} \cdot \mathbf{w} + b) \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \vdots \\ \hat{y}^{(m)} \end{bmatrix}$

Gradients in Logistic Regression

- Cross-entropy loss $L_{CE}(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$

Gradients in Logistic Regression

- Cross-entropy loss $L_{CE}(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$
- We want to compute $\dfrac{\partial L}{\partial w_j}$

Gradients in Logistic Regression

- Cross-entropy loss $L_{CE}(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$
- We want to compute $\dfrac{\partial L}{\partial w_j}$
- Chain Rule of calculus: $\dfrac{dy}{dx} = \dfrac{dy}{dz}\dfrac{dz}{dx}$

Gradients in Logistic Regression

- Cross-entropy loss $L_{CE}(\hat{y}, y) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$
- We want to compute $\dfrac{\partial L}{\partial w_j}$
- Chain Rule of calculus: $\dfrac{dy}{dx} = \dfrac{dy}{dz}\dfrac{dz}{dx}$
- Looking at the graph: $\dfrac{\partial L}{\partial w_j} = \dfrac{\partial L}{\partial \hat{y}}\dfrac{\partial \hat{y}}{\partial z}\dfrac{\partial z}{\partial w_j}$

Gradients in Logistic Regression

- $\dfrac{\partial L}{\partial w_j} = \dfrac{\partial L}{\partial \hat{y}} \dfrac{\partial \hat{y}}{\partial z} \dfrac{\partial z}{\partial w_j}$

Gradients in Logistic Regression

- $$\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} x_j$$
  - $$\frac{\partial z}{\partial w_j} = x_j$$

Gradients in Logistic Regression

- $\frac{\partial L}{\partial w_j} = \frac{\partial L}{\partial \hat{y}} \hat{y}(1 - \hat{y}) x_j$
  - $\frac{\partial z}{\partial w_j} = x_j$
  - For the logistic function: $\frac{\partial \hat{y}}{\partial z} = \hat{y}(1 - \hat{y})$

Gradients in Logistic Regression

- $\dfrac{\partial L}{\partial w_j} = \dfrac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y})x_j$
  - $\dfrac{\partial z}{\partial w_j} = x_j$
  - For the logistic function: $\dfrac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$
  - For the cross-entropy loss: $\dfrac{\partial L}{\partial \hat{y}} = \dfrac{\hat{y} - y}{\hat{y}(1-\hat{y})}$

Gradients in Logistic Regression

- $\frac{\partial L}{\partial w_j} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y})x_j = (\hat{y} - y)x_j$
  - $\frac{\partial z}{\partial w_j} = x_j$
  - For the logistic function: $\frac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$
  - For the cross-entropy loss: $\frac{\partial L}{\partial \hat{y}} = \frac{\hat{y} - y}{\hat{y}(1-\hat{y})}$

Gradients in Logistic Regression

- $\dfrac{\partial L}{\partial w_j} = \dfrac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y})x_j = (\hat{y} - y)x_j$
  - $\dfrac{\partial z}{\partial w_j} = x_j$
  - For the logistic function: $\dfrac{\partial \hat{y}}{\partial z} = \hat{y}(1-\hat{y})$
  - For the cross-entropy loss: $\dfrac{\partial L}{\partial \hat{y}} = \dfrac{\hat{y} - y}{\hat{y}(1-\hat{y})}$
- $\dfrac{\partial L}{\partial b} = \dfrac{\hat{y} - y}{\hat{y}(1-\hat{y})}\hat{y}(1-\hat{y})(1) = \hat{y} - y$
  - $\dfrac{\partial z}{\partial b} = 1$
  - ...

Gradients in Logistic Regression

- Let m be the number of documents in the (mini-)batch

Gradients in Logistic Regression

- Let m be the number of documents in the (mini-)batch
- In vector form: $\nabla L = \frac{1}{m}\left(\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})\right)$

Gradients in Logistic Regression

- Let m be the number of documents in the (mini-)batch
- In vector form: $\nabla L = \frac{1}{m}\left(\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})\right)$
  - What is $\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})$?

Gradients in Logistic Regression

$$\begin{bmatrix} x_1^{(1)} & \dots & x_1^{(m)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \dots & x_n^{(m)} \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \vdots \\ \hat{y}^{(m)} - y^{(m)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_1^{(i)} \\ \vdots \\ \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_n^{(i)} \\ \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \end{bmatrix}$$

Gradients in Logistic Regression

$$
\begin{bmatrix} x_1^{(1)} & \dots & x_1^{(m)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \dots & x_n^{(m)} \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \vdots \\ \hat{y}^{(m)} - y^{(m)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_1^{(i)} \\ \vdots \\ \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x_n^{(i)} \\ \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) \end{bmatrix}
$$

$$
= \begin{bmatrix} \displaystyle\sum_{i=1}^m \left(\frac{\partial L}{\partial w_1}\right)^{(i)} \\ \vdots \\ \displaystyle\sum_{i=1}^m \left(\frac{\partial L}{\partial w_n}\right)^{(i)} \\ \displaystyle\sum_{i=1}^m \left(\frac{\partial L}{\partial b}\right)^{(i)} \end{bmatrix}
$$

Gradients in Logistic Regression

$$\begin{bmatrix} x_1^{(1)} & \dots & x_1^{(m)} \\ \vdots & \ddots & \vdots \\ x_n^{(1)} & \dots & x_n^{(m)} \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{y}^{(1)} - y^{(1)} \\ \vdots \\ \hat{y}^{(m)} - y^{(m)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_1^{(i)} \\ \vdots \\ \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_n^{(i)} \\ \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{m} \left(\frac{\partial L}{\partial w_1} \right)^{(i)} \\ \vdots \\ \sum_{i=1}^{m} \left(\frac{\partial L}{\partial w_n} \right)^{(i)} \\ \sum_{i=1}^{m} \left(\frac{\partial L}{\partial b} \right)^{(i)} \end{bmatrix}$$

$$= \sum_{i=1}^{m} (\nabla L)^{(i)}$$

Gradients in Logistic Regression

- Let m be the number of documents in the (mini-)batch
- In vector form: $\nabla L = \dfrac{1}{m}\left(\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})\right)$
  - What is $\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})$?
    - It computes the sum of the gradients for each document i in the mini-batch!

Gradients in Logistic Regression

- Let m be the number of documents in the (mini-)batch
- In vector form: $\nabla L = \frac{1}{m}\left(\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})\right)$
  - What is $\mathbf{x}^T \cdot (\hat{\mathbf{y}} - \mathbf{y})$?
    - It computes the sum of the gradients for each document i in the mini-batch!
    - Then to get the average gradient, we just divide by m

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)
- At each time step t:

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)
- At each time step t:
  - Compute gradient ∇L

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)
- At each time step t:
  - Compute gradient ∇L
  - Move in direction of negative gradient

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)
- At each time step t:
  - Compute gradient ∇L
  - Move in direction of negative gradient
- $\theta_{t+1} = \theta_t - \eta \nabla L$

Gradient Descent

- Initialize parameters $\theta = \mathbf{w}, b$ (randomly or $\mathbf{0}$)
- At each time step t:
  - Compute gradient ∇L
  - Move in direction of negative gradient
- $\theta_{t+1} = \theta_t - \eta \nabla L$
- Because L is convex, we eventually reach a global minimum