Lecture 5: Generalized Linear Models, MLP, and Back-Propagation

Tao LIN

March 16, 2023

WESTLAKE UNIVERSITY | SCHOOL OF ENGINEERING

Reading materials

- Chapter 3, Stanford CS 229 Lecture Notes, `https://cs229.stanford.edu/notes2022fall/main_notes.pdf`
- Lecture 4, Stanford CS 231n, `http://cs231n.stanford.edu/schedule.html`

Reference

- EPFL, CS-433 Machine Learning, `https://github.com/epfml/ML_course`

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression
2. Exponential Families and Generalized Linear Models
3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression
2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)
3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

Generalization gap: How far is the test from the true error?

- **True Error:**

$$L_{\mathcal{D}}(f) = \mathbb{E}_{(y,\mathbf{x})\sim\mathcal{D}}\left[\ell\left(y, f(\mathbf{x})\right)\right]. \tag{1}$$

- **Test/Empirical Error:**

$$L_{S_{\text{test}}}(f) = \frac{1}{|S_{\text{test}}|}\sum_{(\mathbf{x}_n,y_n)\in S_{\text{test}}} \ell\left(y_n, f(\mathbf{x}_n)\right). \tag{2}$$

- **Generalization Error:**

$$\left|L_{\mathcal{D}}(f) - L_{S_{\text{test}}}(f)\right| . \tag{3}$$

- **In Expectation:**

$$L_{\mathcal{D}}(f) = \mathbb{E}_{S_{\text{test}}\sim\mathcal{D}}\left[L_{S_{\text{test}}}(f)\right] , \tag{4}$$

Generalization gap: How far is the test from the true error?

- **True Error:**

$$L_{\mathcal{D}}(f) = \mathbb{E}_{(y,\mathbf{x})\sim\mathcal{D}}\left[\ell\left(y, f(\mathbf{x})\right)\right]. \tag{1}$$

- **Test/Empirical Error:**

$$L_{S_{\text{test}}}(f) = \frac{1}{|S_{\text{test}}|}\sum_{(\mathbf{x}_n, y_n)\in S_{\text{test}}} \ell\left(y_n, f(\mathbf{x}_n)\right). \tag{2}$$

- **Generalization Error:**

$$\left|L_{\mathcal{D}}(f) - L_{S_{\text{test}}}(f)\right| . \tag{3}$$

- **In Expectation:**

$$L_{\mathcal{D}}(f) = \mathbb{E}_{S_{\text{test}}\sim\mathcal{D}}\left[L_{S_{\text{test}}}(f)\right] , \tag{4}$$

The variation of the $|L_{\mathcal{D}}(f) - L_{S_{\text{test}}}(f)|$ matters.

Theorem 1

Given a model f and a test set $S_{test} \sim \mathcal{D}$ i.i.d. (not used to learn f) and a loss $\ell(\cdot,\cdot) \in [a,b]$:

$$\Pr\left[|L_{\mathcal{D}}(f) - L_{S_{test}}(f)| \geq \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2\,|S_{test}|}} \right] \leq \delta \tag{5}$$

Theorem 1

Given a model f and a test set $S_{test} \sim \mathcal{D}$ i.i.d. (not used to learn f) and a loss $\ell(\cdot,\cdot) \in [a,b]$:

$$\Pr\left[\left|L_{\mathcal{D}}(f) - L_{S_{test}}(f)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2\left|S_{test}\right|}}\right] \leq \delta \tag{5}$$

- The error decreases as $\mathcal{O}\left(1/\sqrt{S_{\text{test}}}\right)$ with the number test points.

Theorem 1

Given a model f and a test set $S_{\mathit{test}} \sim \mathcal{D}$ i.i.d. (not used to learn f) and a loss $\ell(\cdot,\cdot) \in [a,b]$:

$$\Pr\left[\left|L_{\mathcal{D}}(f) - L_{S_{\mathit{test}}}(f)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2\left|S_{\mathit{test}}\right|}}\right] \leq \delta \tag{5}$$

- The error decreases as $\mathcal{O}\left(1/\sqrt{S_{\text{test}}}\right)$ with the number test points.

⇒ The more data points we have, the more confident we are that the empirical loss we measure is close to the true loss.

Theorem 1

Given a model f and a test set $S_{test} \sim \mathcal{D}$ i.i.d. (not used to learn f) and a loss $\ell(\cdot,\cdot) \in [a,b]$:

$$\Pr\left[\left|L_{\mathcal{D}}(f) - L_{S_{test}}(f)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2\left|S_{test}\right|}}\right] \leq \delta \tag{5}$$

- The error decreases as $\mathcal{O}\left(1/\sqrt{S_{\text{test}}}\right)$ with the number test points.

⇒ The more data points we have, the more confident we are that the empirical loss we measure is close to the true loss.

Given a predictor f and a dataset S, we can control the expected risk:

$$\Pr\left[\underbrace{L_{\mathcal{D}}(f)}_{\text{not computable}} \geq \underbrace{L_{S_{\text{test}}}(f)}_{\text{computable}} + \underbrace{\sqrt{\frac{(b-a)^2 \ln(2/\delta)}{2\left|S_{\text{test}}\right|}}}_{\text{deviation}}\right] \leq \delta\,. \tag{6}$$

How far is each of the K test errors $L_{S_{\text{test}}}(f_k)$ from the true $L_{\mathcal{D}}(f_k)$?

Theorem 2

We can bound the maximum deviation for all K candidates, by

$$\Pr\left[\max_k \left|L_{\mathcal{D}}(f_k) - L_{S_{\mathit{test}}}(f_k)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2\left|S_{\mathit{test}}\right|}}\right] \leq \delta \tag{7}$$

How far is each of the K test errors $L_{S_{\text{test}}}(f_k)$ from the true $L_{\mathcal{D}}(f_k)$?

Theorem 2

We can bound the maximum deviation for all K candidates, by

$$\Pr\left[\max_k |L_{\mathcal{D}}(f_k) - L_{S_{test}}(f_k)| \geq \sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2\,|S_{test}|}}\right] \leq \delta \tag{7}$$

- The error decreases as $\mathcal{O}(1/\sqrt{|S_{\text{test}}|})$ with the number test points.

How far is each of the K test errors $L_{S_{\text{test}}}(f_k)$ from the true $L_{\mathcal{D}}(f_k)$?

Theorem 2

We can bound the maximum deviation for all K candidates, by

$$\Pr\left[\max_k \left|L_{\mathcal{D}}(f_k) - L_{S_{test}}(f_k)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2\left|S_{test}\right|}}\right] \leq \delta \tag{7}$$

- The error decreases as $\mathcal{O}(1/\sqrt{|S_{\text{test}}|})$ with the number test points.
- When testing K hyper-parameters, the error only goes up by $\sqrt{\ln(K)}$.

How far is each of the K test errors $L_{S_{\text{test}}}(f_k)$ from the true $L_{\mathcal{D}}(f_k)$?

Theorem 2

We can bound the maximum deviation for all K candidates, by

$$\Pr\left[\max_k \left|L_{\mathcal{D}}(f_k) - L_{S_{test}}(f_k)\right| \geq \sqrt{\frac{(b-a)^2 \ln(2K/\delta)}{2\left|S_{test}\right|}}\right] \leq \delta \tag{7}$$

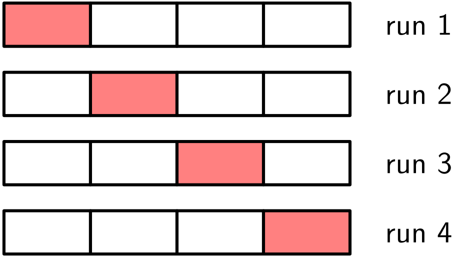
- The error decreases as $\mathcal{O}(1/\sqrt{|S_{\text{test}}|})$ with the number test points.
- When testing K hyper-parameters, the error only goes up by $\sqrt{\ln(K)}$.

⇒ So we can test many different models without incurring a large penalty.

Issues: Splitting the data once into two parts (one for training and one for testing) is not the most efficient way to use the data!

K-fold cross-validation:

1. Randomly partition the data into K groups
2. Train K times. Each time leave out exactly one of the K groups for testing and use the remaining $K-1$ groups for training.
3. Average the K results

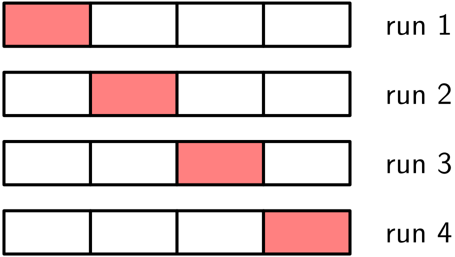


Benefits:

Issues: Splitting the data once into two parts (one for training and one for testing) is not the most efficient way to use the data!

K-fold cross-validation:

1. Randomly partition the data into K groups
2. Train K times. Each time leave out exactly one of the K groups for testing and use the remaining $K-1$ groups for training.
3. Average the K results



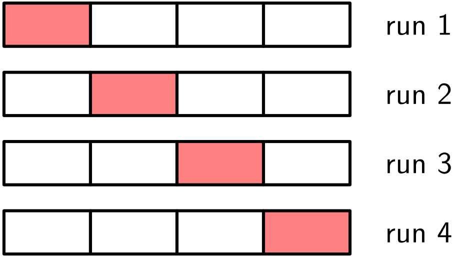
Benefits:

- We have used all data for training, and all data for testing, and used each data point the same number of times.

Issues: Splitting the data once into two parts (one for training and one for testing) is not the most efficient way to use the data!

K-fold cross-validation:

1. Randomly partition the data into K groups
2. Train K times. Each time leave out exactly one of the K groups for testing and use the remaining $K-1$ groups for training.
3. Average the K results



Benefits:

- We have used all data for training, and all data for testing, and used each data point the same number of times.
- Cross-validation returns an unbiased estimate of the generalization error and its variance.

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

Bias-Variance Decomposition

$$\begin{aligned}
\mathbb{E}_{S_{\text{train}} \in \mathcal{D}}\left[L(f_{S_{\text{train}}})\right] &= \mathsf{Var}_{\boldsymbol{\epsilon} \sim \mathcal{D}_{\boldsymbol{\epsilon}}}\left[\boldsymbol{\epsilon}\right] \tag*{(noise variance)} \\
&+ \left(f(\mathbf{x}_0) - \mathbb{E}_{S_{\text{train'}}}\left[f_{S_{\text{train'}}}(\mathbf{x}_0)\right]\right)^2 \tag*{(Bias)} \\
&+ \mathbb{E}_{S_{\text{train}} \sim \mathcal{D}}\left[\left(\mathbb{E}_{S_{\text{train'}}}\left[f_{S_{\text{train'}}}(\mathbf{x}_0)\right] - f_{S_{\text{train}}}(\mathbf{x}_0)\right)^2\right], \tag*{(Variance)}
\end{aligned}$$

which always lower-bounds the true error.

Bias-Variance Decomposition

$$\begin{aligned}
\mathbb{E}_{S_{\text{train}} \in \mathcal{D}} \left[L(f_{S_{\text{train}}}) \right] &= \text{Var}_{\boldsymbol{\epsilon} \sim \mathcal{D}_{\boldsymbol{\epsilon}}} [\boldsymbol{\epsilon}] && \text{(noise variance)} \\
&+ \left(f(\mathbf{x}_0) - \mathbb{E}_{S_{\text{train'}}} \left[f_{S_{\text{train'}}}(\mathbf{x}_0) \right] \right)^2 && \text{(Bias)} \\
&+ \mathbb{E}_{S_{\text{train}} \sim \mathcal{D}} \left[\left(\mathbb{E}_{S_{\text{train'}}} \left[f_{S_{\text{train'}}}(\mathbf{x}_0) \right] - f_{S_{\text{train}}}(\mathbf{x}_0) \right)^2 \right], && \text{(Variance)}
\end{aligned}$$

which always lower-bounds the true error.

$\Rightarrow$ To minimize the true error, we need to select a method that **simultaneously achieves low bias and low variance**.

Double descent curve in Deep Learning

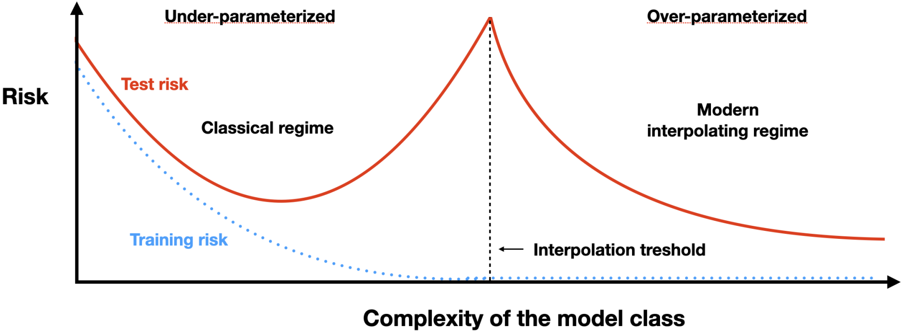


Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

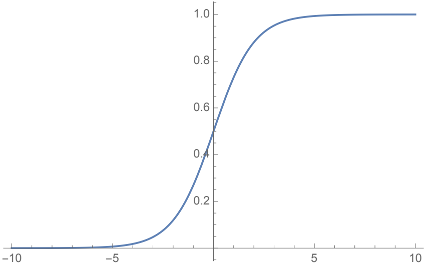
3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

The logistic function

Consider first of all the case of two classes.
The posterior probability for class $\mathcal{C}_1$:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{8}$$

$$= \frac{1}{1+\exp(-\eta)} = \sigma(\eta) \tag{9}$$



Properties of the logistic function:

- $1 - \sigma(\eta) = \sigma(-\eta)$
- $\sigma'(\eta) = \sigma(\eta)\,(1 - \sigma(\eta))$

The logistic function

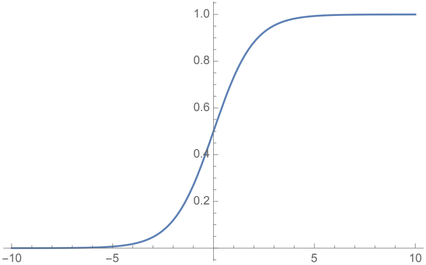
Consider first of all the case of two classes.
The posterior probability for class $\mathcal{C}_1$:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{8}$$

$$= \frac{1}{1 + \exp(-\eta)} = \sigma(\eta) \tag{9}$$

where we have defined

$$\eta = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \text{ and } \sigma(\eta) := \frac{e^\eta}{1 + e^\eta} \tag{10}$$



Properties of the logistic function:

- $1 - \sigma(\eta) = \sigma(-\eta)$
- $\sigma'(\eta) = \sigma(\eta)\,(1 - \sigma(\eta))$

The logistic function

Consider first of all the case of two classes.
The posterior probability for class $\mathcal{C}_1$:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{8}$$

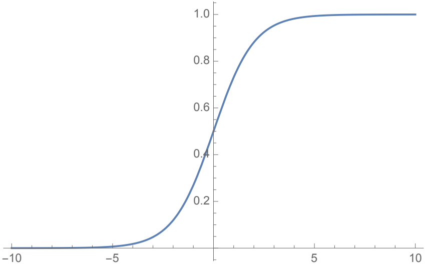
$$= \frac{1}{1 + \exp(-\eta)} = \sigma(\eta) \tag{9}$$

where we have defined

$$\eta = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \text{ and } \sigma(\eta) := \frac{e^\eta}{1 + e^\eta} \tag{10}$$

For the case of $K > 2$ classes, we have

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(\eta_k)}{\sum_j \exp(\eta_j)} \tag{11}$$



Properties of the logistic function:

- $1 - \sigma(\eta) = \sigma(-\eta)$
- $\sigma'(\eta) = \sigma(\eta)\,(1 - \sigma(\eta))$

Logistic Regression

Given a “new” feature vector $\mathbf{x}$, we predict the (posterior) probability of the two class labels given $\mathbf{x}$ by means of

$$p(1|\mathbf{x}) := \Pr\left[Y = 1|\mathbf{X} = \mathbf{x}\right] = \sigma\left(\mathbf{x}^\top \mathbf{w} + w_0\right) \tag{12}$$

$$p(0|\mathbf{x}) := \Pr\left[Y = 0|\mathbf{X} = \mathbf{x}\right] = 1 - \sigma\left(\mathbf{x}^\top \mathbf{w} + w_0\right) \ , \tag{13}$$

where we predict a real value (a probability) and not a label.

MLE is a method of estimating the parameters of a statistical model

The MLE finds the parameters $\mathbf{w}^\star$ under which $\{\mathbf{y}, \mathbf{X}\}$ are the most likely:

$$\mathbf{w}^\star = \arg\max_{\mathbf{w}} \left(\mathcal{L}(\mathbf{w}) := \prod_{n=1}^{N} p(\{\mathbf{x}_n, y_n\} | \mathbf{w}) \right) = \arg\min_{\mathbf{w}} \left[-\log \mathcal{L}(\mathbf{w}) \right] \; . \tag{14}$$

MLE is a method of estimating the parameters of a statistical model

The MLE finds the parameters $\mathbf{w}^\star$ under which $\{\mathbf{y}, \mathbf{X}\}$ are the most likely:

$$\mathbf{w}^\star = \arg\max_{\mathbf{w}} \left(\mathcal{L}(\mathbf{w}) := \prod_{n=1}^{N} p(\{\mathbf{x}_n, y_n\}|\mathbf{w}) \right) = \arg\min_{\mathbf{w}} \left[-\log \mathcal{L}(\mathbf{w}) \right] . \tag{14}$$

The likelihood of the data $\{\mathbf{y}, \mathbf{X}\}$ given the parameter $\mathbf{w}$, i.e., $p(\mathbf{y}, \mathbf{X}|\mathbf{w})$.

$$p(\mathbf{y}, \mathbf{X}|\mathbf{w}) = p(\mathbf{X}|\mathbf{w}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = p(\mathbf{X}) p(\mathbf{y}|\mathbf{X}, \mathbf{w}) \,, \tag{15}$$

where $\mathbf{X}$ does not depend on $\mathbf{w}$.

MLE for Logistic Regression

For Logistic Regression, we have:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n) = \prod_{n:y_n=1} p(y_n = 1|\mathbf{x}_n) \prod_{n:y_n=0} p(y_n = 0|\mathbf{x}_n) \tag{16}$$

$$= \prod_{n=1}^{N} \sigma(\mathbf{x}_n^\top \mathbf{w})^{y_n} \left[1 - \sigma(\mathbf{x}_n^\top \mathbf{w})\right]^{1-y_n} \tag{17}$$

MLE for Logistic Regression

For Logistic Regression, we have:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n) = \prod_{n:y_n=1} p(y_n = 1|\mathbf{x}_n) \prod_{n:y_n=0} p(y_n = 0|\mathbf{x}_n) \tag{16}$$

$$= \prod_{n=1}^{N} \sigma(\mathbf{x}_n^\top \mathbf{w})^{y_n} \left[1 - \sigma(\mathbf{x}_n^\top \mathbf{w})\right]^{1-y_n} \tag{17}$$

Minimizing $\mathcal{L}(\mathbf{w})$ through the property of stationary points.

$$\nabla\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \left(\sigma(\mathbf{x}_n^\top \mathbf{w}) - y_n\right) = \frac{1}{N}\mathbf{X}^\top \left[\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}\right], \tag{18}$$

where $\mathbf{X} \in \mathbb{R}^{N\times d}$. It has no closed-form solution to $\nabla\mathcal{L}(\mathbf{w}) = 0$.

Last lecture:

- Generalization Gap and Model Selection
- Bias-Variance Decomposition
- Before Introducing Multilayer Perceptron: Logistic Regression

Last lecture:

- Generalization Gap and Model Selection
- Bias-Variance Decomposition
- Before Introducing Multilayer Perceptron: Logistic Regression

This lecture:

- Exponential Families and Generalized Linear Models
- Multi-Layer Perceptron
- Back-Propagation

Table of Contents

1 Review of Last Week

2 Exponential Families and Generalized Linear Models

- Motivation
- Exponential family
- Application in ML (Generalized Linear Models)

3 Multi-Layer Perceptron (MLP) and Back-Propagation (BP)

Table of Contents

1 Review of Last Week
- Generalization Gap and Model Selection
- Bias-Variance Decomposition
- Logistic Regression

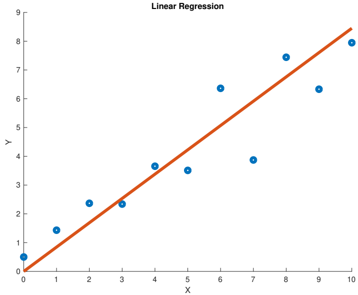
2 Exponential Families and Generalized Linear Models
- Motivation
- Exponential family
- Application in ML (Generalized Linear Models)

3 Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
- The Basic Structure of MLP
- Training of NNs and BP

The Least-Squares can be defined in two different ways

- **Geometric way:**
  Minimizing the sum of the squares of the residuals:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{19}$$



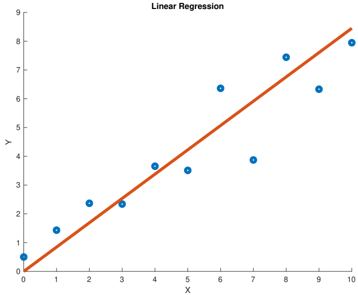
The Least-Squares can be defined in two different ways

- **Geometric way:**
  Minimizing the sum of the squares of the residuals:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{19}$$

- **Probabilistic way:**
  Assume the data follow a linear Gaussian model:

$$\mathbf{y} = \mathbf{x}^\top \mathbf{w} + \boldsymbol{\epsilon} \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2) \tag{20}$$



The Least-Squares can be defined in two different ways

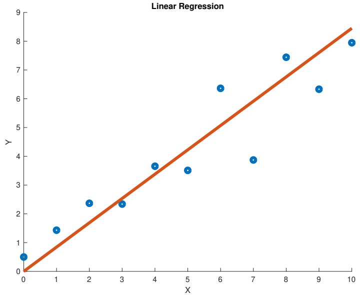
- **Geometric way:**
  Minimizing the sum of the squares of the residuals:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{2N} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\top \mathbf{w})^2 \tag{19}$$

- **Probabilistic way:**
  Assume the data follow a linear Gaussian model:

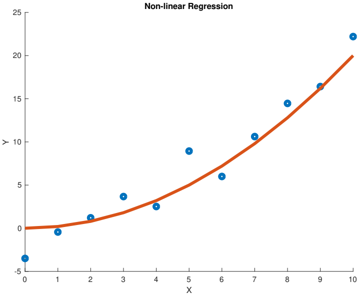
$$\mathbf{y} = \mathbf{x}^\top \mathbf{w} + \boldsymbol{\epsilon} \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\sigma}^2) \tag{20}$$

Doing MLE recovers the LS estimator $\hat{\mathbf{w}}$.



How to get non-linear models?

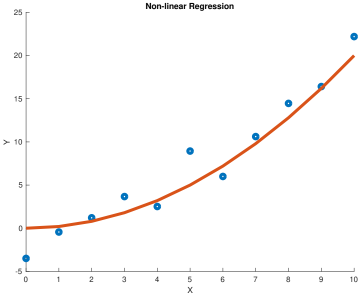
- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \ldots)$



How to get non-linear models?

- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \ldots)$

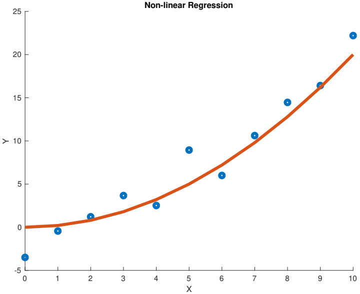
- **Different probabilistic models:**



How to get non-linear models?

- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \ldots)$

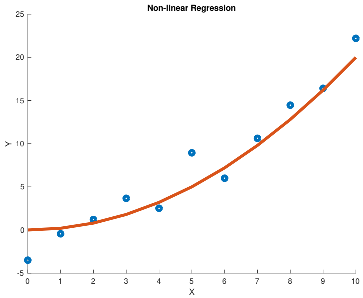
- **Different probabilistic models:**
  - Least Squares: $y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \boldsymbol{\sigma}^2\right)$



How to get non-linear models?

- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \ldots)$
- **Different probabilistic models:**
  - Least Squares: $y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \boldsymbol{\sigma}^2\right)$

    $\Rightarrow$ The linear model predicts the mean of a distribution μ (from which the data are sampled).



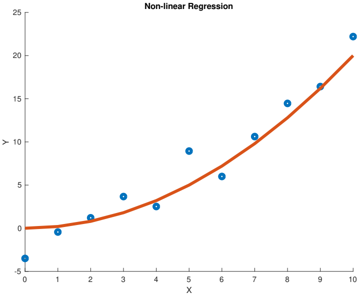
How to get non-linear models?

- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \ldots)$

- **Different probabilistic models:**
  - Least Squares: $y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \boldsymbol{\sigma}^2\right)$

    $\Rightarrow$ The linear model predicts the mean of a distribution μ (from which the data are sampled).

  - Logistic Regression: $y \sim \mathcal{B}\left(\sigma(\mathbf{x}^\top \mathbf{w})\right)$



How to get non-linear models?

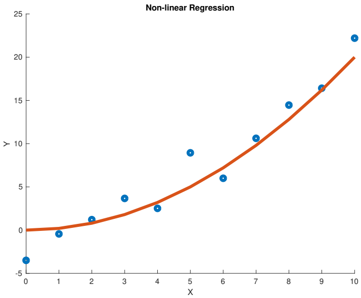
- **Features augmentations:** add non-linear features $(\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \dots)$

- **Different probabilistic models:**
  - Least Squares: $y \sim \mathcal{N}\left(\mathbf{x}^\top \mathbf{w}, \boldsymbol{\sigma}^2\right)$

    $\Rightarrow$ The linear model predicts the mean of a distribution μ (from which the data are sampled).

  - Logistic Regression: $y \sim \mathcal{B}\left(\sigma(\mathbf{x}^\top \mathbf{w})\right)$

    $\Rightarrow$ The linear model predicts another quantity $\eta := \mathbf{x}^\top \mathbf{w}$.



Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$.

Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$. This can be compactly written as

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{22}$$

Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$. This can be compactly written as

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{22}$$

- The linear model predicts $\sigma(\eta)$ which is not the mean of the distribution.

Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$. This can be compactly written as

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^\eta} = \exp\left(\eta y - \ln(1 + e^\eta)\right) \tag{22}$$

- The linear model predicts $\sigma(\eta)$ which is not the mean of the distribution.
- η is related to the mean μ by the non-linear relation

Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$. This can be compactly written as

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{22}$$

- The linear model predicts $\sigma(\eta)$ which is not the mean of the distribution.
- η is related to the mean μ by the non-linear relation

⇒ The *link function*:

Recall the definition of Logistic Regression

Logistic Regression models the probability of the two classes $\{0, 1\}$ by

$$p(1|\eta) = \sigma(\eta) \text{ and } p(0|\eta) = 1 - \sigma(\eta)\,, \tag{21}$$

where $\eta = \mathbf{x}^\top \mathbf{w}$. This can be compactly written as

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{22}$$

- The linear model predicts $\sigma(\eta)$ which is not the mean of the distribution.
- η is related to the mean μ by the non-linear relation

$\Rightarrow$ The *link function*:
the relation between (1) the value η we predict by the linear model and (2) the mean μ.

A unified framework

The distribution used in Logistic Regression can be written in a very specific form:

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{23}$$

A unified framework

The distribution used in Logistic Regression can be written in a very specific form:

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{23}$$

Goals: a unified framework to generalize other forms of distributions.

A unified framework

The distribution used in Logistic Regression can be written in a very specific form:

$$p(y|\eta) = \frac{e^{\eta y}}{1+e^{\eta}} = \exp\left(\eta y - \ln(1+e^{\eta})\right) \tag{23}$$

Goals: a unified framework to generalize other forms of distributions.

- The discussion on a class of distributions, known as *exponential families*.

A unified framework

The distribution used in Logistic Regression can be written in a very specific form:

$$p(y|\eta) = \frac{e^{\eta y}}{1 + e^{\eta}} = \exp\left(\eta y - \ln(1 + e^{\eta})\right) \tag{23}$$

Goals: a unified framework to generalize other forms of distributions.

- The discussion on a class of distributions, known as *exponential families*.
- Many distributions (but not all) fit into this framework and that distributions in this family have many nice properties.

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

[1] Assume that we are given independent samples from this distribution. We do know $\boldsymbol{\phi}(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\boldsymbol{\phi}(y)$.

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

- natural or canonical parameter of the distribution $\boldsymbol{\eta}$

[1] Assume that we are given independent samples from this distribution. We do know $\boldsymbol{\phi}(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\boldsymbol{\phi}(y)$.

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

- natural or canonical parameter of the distribution $\boldsymbol{\eta}$
- sufficient statistics[1] $\phi(y)$ contains all the relevant information

[1] Assume that we are given independent samples from this distribution. We do know $\boldsymbol{\phi}(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\boldsymbol{\phi}(y)$.

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

- natural or canonical parameter of the distribution $\boldsymbol{\eta}$
- sufficient statistics[1] $\boldsymbol{\phi}(y)$ contains all the relevant information
- $A(\boldsymbol{\eta})$: log partition, the quantity $e^{-A(\boldsymbol{\eta})}$ is used as a normalization constant:

[1] Assume that we are given independent samples from this distribution. We do know $\boldsymbol{\phi}(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\boldsymbol{\phi}(y)$.

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

- natural or canonical parameter of the distribution $\boldsymbol{\eta}$
- sufficient statistics[1] $\phi(y)$ contains all the relevant information
- $A(\boldsymbol{\eta})$: log partition, the quantity $e^{-A(\boldsymbol{\eta})}$ is used as a normalization constant:

$$\int p(y|\boldsymbol{\eta})dy = \int h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] dy = 1 \tag{25}$$

[1] Assume that we are given independent samples from this distribution. We do know $\phi(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\phi(y)$.

Exponential family — Definition

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{24}$$

- natural or canonical parameter of the distribution $\boldsymbol{\eta}$
- sufficient statistics[1] $\phi(y)$ contains all the relevant information
- $A(\boldsymbol{\eta})$: log partition, the quantity $e^{-A(\boldsymbol{\eta})}$ is used as a normalization constant:

$$\int p(y|\boldsymbol{\eta}) dy = \int h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] dy = 1 \tag{25}$$

$$\implies \int h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y)\right] dy = \int h(y) \exp\left[A(\boldsymbol{\eta})\right] dy = \exp\left[A(\boldsymbol{\eta})\right].$$

[1] Assume that we are given independent samples from this distribution. We do know $\phi(y)$ and $h(y)$ but not $\boldsymbol{\eta}$. In order to optimally estimate $\boldsymbol{\eta}$ given these samples, all we need is the empirical average of the $\phi(y)$.

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

- A fixed choice of $\boldsymbol{\phi}(y)$, $A(\boldsymbol{\eta})$ and $h(y)$ defines a family of distributions (parameterized by $\boldsymbol{\eta}$).

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

- A fixed choice of $\boldsymbol{\phi}(y)$, $A(\boldsymbol{\eta})$ and $h(y)$ defines a family of distributions (parameterized by $\boldsymbol{\eta}$).
- As we very $\boldsymbol{\eta}$, we then get different distribution within this family.

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

- A fixed choice of $\boldsymbol{\phi}(y)$, $A(\boldsymbol{\eta})$ and $h(y)$ defines a family of distributions (parameterized by $\boldsymbol{\eta}$).
- As we very $\boldsymbol{\eta}$, we then get different distribution within this family.
- For some parameters $\boldsymbol{\eta}$, there exists some $h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y)\right]$ that cannot be normalized.

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

- A fixed choice of $\boldsymbol{\phi}(y)$, $A(\boldsymbol{\eta})$ and $h(y)$ defines a family of distributions (parameterized by $\boldsymbol{\eta}$).

- As we very $\boldsymbol{\eta}$, we then get different distribution within this family.

- For some parameters $\boldsymbol{\eta}$, there exists some $h(y)\exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y)\right]$ that cannot be normalized.

  For example, $h(y) = 1, \boldsymbol{\phi}(y) = y^2$ and $\boldsymbol{\eta} = 1$.

A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = \underbrace{h(y)}_{\geq 0} \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{26}$$

- A fixed choice of $\boldsymbol{\phi}(y)$, $A(\boldsymbol{\eta})$ and $h(y)$ defines a family of distributions (parameterized by $\boldsymbol{\eta}$).
- As we very $\boldsymbol{\eta}$, we then get different distribution within this family.
- For some parameters $\boldsymbol{\eta}$, there exists some $h(y)\exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y)\right]$ that cannot be normalized.

  For example, $h(y) = 1, \boldsymbol{\phi}(y) = y^2$ and $\boldsymbol{\eta} = 1$.

We will exclude such parameters by only looking at the set of parameters

$$M := \left\{ \boldsymbol{\eta} : \int_y h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y)\right] dy < \infty \right\} \tag{27}$$

Why?

Bernoulli distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{28}$$

Bernoulli distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y)\exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{28}$$

The Bernoulli distribution is the binary random variable such that for $\mu \in [0, 1]$:

$$\Pr(Y = 1) = \mu \qquad \text{and} \qquad \Pr(Y = 0) = 1 - \mu \tag{29}$$

Bernoulli distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{28}$$

The Bernoulli distribution is the binary random variable such that for $\mu \in [0, 1]$:

$$\Pr(Y = 1) = \mu \qquad \text{and} \qquad \Pr(Y = 0) = 1 - \mu \tag{29}$$

Claim: the Bernoulli distribution is a member of the exponential family.

$$p(y|\mu) = \mu^y (1-\mu)^{1-y}, \text{ where } \mu \in (0, 1) \tag{30}$$

$$= \exp\left\{(\ln \frac{\mu}{1-\mu})y + \ln(1-\mu)\right\} = \exp\left\{\eta \boldsymbol{\phi}(y) - A(\eta)\right\}. \tag{31}$$

Bernoulli distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{28}$$

The Bernoulli distribution is the binary random variable such that for $\mu \in [0, 1]$:

$$\Pr(Y = 1) = \mu \qquad \text{and} \qquad \Pr(Y = 0) = 1 - \mu \tag{29}$$

Claim: the Bernoulli distribution is a member of the exponential family.

$$p(y|\mu) = \mu^y (1-\mu)^{1-y}, \text{ where } \mu \in (0, 1) \tag{30}$$

$$= \exp\left\{(\ln \frac{\mu}{1-\mu})y + \ln(1-\mu)\right\} = \exp\left\{\eta \boldsymbol{\phi}(y) - A(\eta)\right\}. \tag{31}$$

where we can identify:

$$\boldsymbol{\phi}(y) = y\,, \quad \eta = \ln \tfrac{\mu}{1-\mu}\,, \quad A(\eta) = -\ln(1-\mu) = \ln(1+e^\eta)\,, \quad h(y) = 1. \tag{32}$$

Bernoulli distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y)\exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{28}$$

The Bernoulli distribution is the binary random variable such that for $\mu \in [0,1]$:

$$\Pr(Y=1) = \mu \qquad \text{and} \qquad \Pr(Y=0) = 1-\mu \tag{29}$$

Claim: the Bernoulli distribution is a member of the exponential family.

$$p(y|\mu) = \mu^y (1-\mu)^{1-y}, \text{ where } \mu \in (0,1) \tag{30}$$

$$= \exp\left\{(\ln \frac{\mu}{1-\mu})y + \ln(1-\mu)\right\} = \exp\left\{\eta\boldsymbol{\phi}(y) - A(\eta)\right\}. \tag{31}$$

where we can identify:

$$\boldsymbol{\phi}(y) = y\,, \quad \eta = \ln\tfrac{\mu}{1-\mu}\,, \quad A(\eta) = -\ln(1-\mu) = \ln(1+e^\eta)\,, \quad h(y) = 1. \tag{32}$$

$\Rightarrow \eta = g(\mu) = \ln\frac{\mu}{1-\mu} \iff \mu = g^{-1}(\eta) = \frac{e^\eta}{1+e^\eta}$, and $g(\mu)$ links the mean of $\boldsymbol{\phi}(y)$ to $\boldsymbol{\eta}$.

Gaussian distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{33}$$

Gaussian distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{33}$$

Claim: the Gaussian distribution with mean μ and variance σ^2 is also a member of the exponential family.

Gaussian distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{33}$$

Claim: the Gaussian distribution with mean μ and variance σ^2 is also a member of the exponential family.

$$\begin{aligned} p(y|\mu) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \qquad \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}^+ && (34) \\ &= \exp\left[(\mu/\sigma^2, -1/(2\sigma^2))(y, y^2)^\top - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right]. && (35) \end{aligned}$$

Gaussian distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{33}$$

Claim: the Gaussian distribution with mean μ and variance σ^2 is also a member of the exponential family.

$$p(y|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \qquad \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}^+ \tag{34}$$

$$= \exp\left[(\mu/\sigma^2, -1/(2\sigma^2))(y, y^2)^\top - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right]. \tag{35}$$

$$\boldsymbol{\phi}(y) = (y, y^2)^\top, \qquad \boldsymbol{\eta} = (\eta_1 = \mu/\sigma^2, \eta_2 = -1/(2\sigma^2))^\top, \tag{36}$$

$$A(\boldsymbol{\eta}) = -\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\ln(-\eta_2/\pi), \qquad h(y) = 1. \tag{37}$$

Gaussian distributions belong to the exponential family

Recall: A distribution belongs to the exponential family if it can be written in the form

$$p(y|\boldsymbol{\eta}) = h(y)\exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right] \tag{33}$$

Claim: the Gaussian distribution with mean μ and variance σ^2 is also a member of the exponential family.

$$\begin{aligned} p(y|\mu) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}, \qquad \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}^+ && (34) \\ &= \exp\left[(\mu/\sigma^2, -1/(2\sigma^2))(y, y^2)^\top - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\ln(2\pi\sigma^2)\right]. && (35) \end{aligned}$$

$$\begin{aligned} \boldsymbol{\phi}(y) &= (y, y^2)^\top, & \boldsymbol{\eta} &= (\eta_1 = \mu/\sigma^2, \eta_2 = -1/(2\sigma^2))^\top, && (36) \\ A(\boldsymbol{\eta}) &= -\frac{\eta_1^2}{4\eta_2} - \frac{1}{2}\ln(-\eta_2/\pi), \quad h(y) = 1. && && (37) \end{aligned}$$

Link function ($\eta := g(\mu)$): $\eta_1 = \frac{\mu}{\sigma^2}, \eta_2 = -\frac{1}{2\sigma^2} \iff \mu = -\frac{\eta_1}{2\eta_2}, \sigma^2 = -\frac{1}{2\eta_2}$.

Basic properties

- Cumulant $A(\boldsymbol{\eta})$ is convex.

Basic properties

- Cumulant $A(\boldsymbol{\eta})$ is convex.
- $\nabla A(\boldsymbol{\eta}) = \mathbb{E}\left[\boldsymbol{\phi}(y)\right]$

Basic properties

- Cumulant $A(\boldsymbol{\eta})$ is convex.
- $\nabla A(\boldsymbol{\eta}) = \mathbb{E}\left[\boldsymbol{\phi}(y)\right]$
- $\nabla^2 A(\boldsymbol{\eta}) = \mathbb{E}\left[\boldsymbol{\phi}(y)\boldsymbol{\phi}(y)^\top\right] - \mathbb{E}\left[\boldsymbol{\phi}(y)\right]\mathbb{E}\left[\boldsymbol{\phi}(y)\right]^\top$

Basic properties

- Cumulant $A(\boldsymbol{\eta})$ is convex.
- $\nabla A(\boldsymbol{\eta}) = \mathbb{E}\left[\phi(y)\right]$
- $\nabla^2 A(\boldsymbol{\eta}) = \mathbb{E}\left[\phi(y)\phi(y)^\top\right] - \mathbb{E}\left[\phi(y)\right]\mathbb{E}\left[\phi(y)\right]^\top$
- There is a $1-1$ relationship between the “mean” $\boldsymbol{\mu} := \mathbb{E}\left[\phi(y)\right]$ and natural parameter $\boldsymbol{\eta}$, defined using a so-called *link function* $\mathbf{g}$:

Basic properties

- Cumulant $A(\boldsymbol{\eta})$ is convex.

- $\nabla A(\boldsymbol{\eta}) = \mathbb{E}\left[\phi(y)\right]$

- $\nabla^2 A(\boldsymbol{\eta}) = \mathbb{E}\left[\phi(y)\phi(y)^\top\right] - \mathbb{E}\left[\phi(y)\right]\mathbb{E}\left[\phi(y)\right]^\top$

- There is a $1-1$ relationship between the “mean” $\boldsymbol{\mu} := \mathbb{E}\left[\phi(y)\right]$ and natural parameter $\boldsymbol{\eta}$, defined using a so-called *link function* $\mathbf{g}$:

$$\boldsymbol{\eta} = \mathbf{g}\left(\boldsymbol{\mu} := \mathbb{E}\left[\phi(y)\right]\right) \Longleftrightarrow \boldsymbol{\mu} = \mathbf{g}^{-1}(\boldsymbol{\eta}) = \nabla A(\boldsymbol{\eta}) \tag{38}$$

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

Maximum Likelihood Estimation (MLE)

Assume a set of iid samples $\{y_n\}_{n=1}^N$, sampled from a member of the exponential family with given $h(y)$, sufficient statistics $\phi(y)$, but unknown parameter $\boldsymbol{\eta}$.

Maximum Likelihood Estimation (MLE)

Assume a set of iid samples $\{y_n\}_{n=1}^{N}$, sampled from a member of the exponential family with given $h(y)$, sufficient statistics $\phi(y)$, but unknown parameter $\boldsymbol{\eta}$.

Goal: Estimate the natural parameter $\boldsymbol{\eta}$.

Maximum Likelihood Estimation (MLE)

Assume a set of iid samples $\{y_n\}_{n=1}^N$, sampled from a member of the exponential family with given $h(y)$, sufficient statistics $\boldsymbol{\phi}(y)$, but unknown parameter $\boldsymbol{\eta}$.

Goal: Estimate the natural parameter $\boldsymbol{\eta}$.

How: MLE for $p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right]$.

Maximum Likelihood Estimation (MLE)

Assume a set of iid samples $\{y_n\}_{n=1}^{N}$, sampled from a member of the exponential family with given $h(y)$, sufficient statistics $\phi(y)$, but unknown parameter $\boldsymbol{\eta}$.

Goal: Estimate the natural parameter $\boldsymbol{\eta}$.

How: MLE for $p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right]$.

$$\mathcal{L}(\boldsymbol{\eta}) = -\frac{1}{N} \ln\left(p(y|\boldsymbol{\eta})\right) \tag{39}$$

$$= \frac{1}{N} \sum_{n=1}^{N} \left[-\ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \ . \tag{40}$$

Maximum Likelihood Estimation (MLE)

Assume a set of iid samples $\{y_n\}_{n=1}^{N}$, sampled from a member of the exponential family with given $h(y)$, sufficient statistics $\phi(y)$, but unknown parameter $\boldsymbol{\eta}$.

Goal: Estimate the natural parameter $\boldsymbol{\eta}$.

How: MLE for $p(y|\boldsymbol{\eta}) = h(y) \exp\left[\boldsymbol{\eta}^\top \boldsymbol{\phi}(y) - A(\boldsymbol{\eta})\right]$.

$$\mathcal{L}(\boldsymbol{\eta}) = -\frac{1}{N} \ln\left(p(y|\boldsymbol{\eta})\right) \tag{39}$$

$$= \frac{1}{N} \sum_{n=1}^{N} \left[- \ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \,. \tag{40}$$

⇒ The cost function $\mathcal{L}$ is a convex function in $\boldsymbol{\eta}$ since $A(\boldsymbol{\eta})$ is convex.

Given the definition

$$\mathcal{L}(\boldsymbol{\eta}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \tag{41}$$

[2] It says that we should pick $\boldsymbol{\eta}$ s.t. the expected value of the sufficient statistics is equal to its empirical value!

Given the definition

$$\mathcal{L}(\boldsymbol{\eta}) = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \left[- \ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \tag{41}$$

Gradient:

$$\nabla_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\eta}) = -\tfrac{1}{N} \textstyle\sum_{n=1}^{N} \boldsymbol{\phi}(y_n) + \mathbb{E}\left[\boldsymbol{\phi}(y)\right] , \tag{42}$$

[2] It says that we should pick $\boldsymbol{\eta}$ s.t. the expected value of the sufficient statistics is equal to its empirical value!

Given the definition

$$\mathcal{L}(\boldsymbol{\eta}) = \frac{1}{N} \sum_{n=1}^{N} \left[-\ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta}) \right] \tag{41}$$

Gradient:

$$\nabla_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\eta}) = -\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\phi}(y_n) + \mathbb{E}\left[\boldsymbol{\phi}(y)\right] , \tag{42}$$

Stationary point:[2]

[2] It says that we should pick $\boldsymbol{\eta}$ s.t. the expected value of the sufficient statistics is equal to its empirical value!

Given the definition

$$\mathcal{L}(\boldsymbol{\eta}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \tag{41}$$

Gradient:

$$\nabla_{\boldsymbol{\eta}}\mathcal{L}(\boldsymbol{\eta}) = -\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(y_n) + \mathbb{E}\left[\boldsymbol{\phi}(y)\right], \tag{42}$$

Stationary point:[2]

$$\boldsymbol{\mu} := \mathbb{E}\left[\boldsymbol{\phi}(y)\right] = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(y_n), \tag{43}$$

[2] It says that we should pick $\boldsymbol{\eta}$ s.t. the expected value of the sufficient statistics is equal to its empirical value!

Given the definition

$$\mathcal{L}(\boldsymbol{\eta}) = \tfrac{1}{N}\sum_{n=1}^{N}\left[-\ln\left(h(y_n)\right) - \boldsymbol{\eta}^\top \boldsymbol{\phi}(y_n) + A(\boldsymbol{\eta})\right] \tag{41}$$

Gradient:

$$\nabla_{\boldsymbol{\eta}}\mathcal{L}(\boldsymbol{\eta}) = -\tfrac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(y_n) + \mathbb{E}\left[\boldsymbol{\phi}(y)\right], \tag{42}$$

Stationary point:[2]

$$\boldsymbol{\mu} := \mathbb{E}\left[\boldsymbol{\phi}(y)\right] = \tfrac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(y_n), \tag{43}$$

Closed-form: assume we have determined the link function $\mathbf{g}(\boldsymbol{\mu}) = \boldsymbol{\eta}$

$$\boldsymbol{\eta} = \mathbf{g}\left(\tfrac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(y_n)\right), \tag{44}$$

and justify why we called $\phi(y)$ a sufficient statistics.

[2] It says that we should pick $\boldsymbol{\eta}$ s.t. the expected value of the sufficient statistics is equal to its empirical value!

Generalized Linear Models (GLM)

Scenario:

Generalized Linear Models (GLM)

Scenario:

- We would like to build a model to estimate the number y of customers arriving in a store.

Generalized Linear Models (GLM)

Scenario:

- We would like to build a model to estimate the number y of customers arriving in a store.
- The Poisson distribution usually gives a good model for the number of visitors.

Generalized Linear Models (GLM)

Scenario:

- We would like to build a model to estimate the number y of customers arriving in a store.
- The Poisson distribution usually gives a good model for the number of visitors.
- How can we come up with a model for our problem?

Generalized Linear Models (GLM)

Scenario:

- We would like to build a model to estimate the number y of customers arriving in a store.
- The Poisson distribution usually gives a good model for the number of visitors.
- How can we come up with a model for our problem?
- Fortunately the Poisson is an exponential family distribution.

Generalized Linear Models (GLM)

Scenario:

- We would like to build a model to estimate the number y of customers arriving in a store.
- The Poisson distribution usually gives a good model for the number of visitors.
- How can we come up with a model for our problem?
- Fortunately the Poisson is an exponential family distribution.

We can apply a Generalized Linear Model (GLM)!

Constructing GLMs

To derive a GLM for a classification/regression problem (the conditional dist. of y given $\mathbf{x}$):

Constructing GLMs

To derive a GLM for a classification/regression problem (the conditional dist. of y given $\mathbf{x}$):

1. The natural parameter η and the observed inputs $\mathbf{x}$ are related linearly: $\eta = \mathbf{x}^\top \mathbf{w}$

Constructing GLMs

To derive a GLM for a classification/regression problem (the conditional dist. of y given $\mathbf{x}$):

1. The natural parameter η and the observed inputs $\mathbf{x}$ are related linearly: $\eta = \mathbf{x}^\top \mathbf{w}$
2. The conditional mean μ is represented as a function $f(\eta)$ of the linear combination $\boldsymbol{\eta}$

Constructing GLMs

To derive a GLM for a classification/regression problem (the conditional dist. of y given $\mathbf{x}$):

1. The natural parameter η and the observed inputs $\mathbf{x}$ are related linearly: $\eta = \mathbf{x}^\top \mathbf{w}$
2. The conditional mean μ is represented as a function $f(\eta)$ of the linear combination $\boldsymbol{\eta}$
3. The observed output y is assumed to be characterized by an exponential family distribution with conditional mean μ, i.e.,

Constructing GLMs

To derive a GLM for a classification/regression problem (the conditional dist. of y given $\mathbf{x}$):

1. The natural parameter η and the observed inputs $\mathbf{x}$ are related linearly: $\eta = \mathbf{x}^\top \mathbf{w}$
2. The conditional mean μ is represented as a function $f(\eta)$ of the linear combination $\boldsymbol{\eta}$
3. The observed output y is assumed to be characterized by an exponential family distribution with conditional mean μ, i.e.,

The condition probability is thus modeled as:

$$p(y|\mathbf{x};\mathbf{w}) = h(y_n)\exp\left(\eta\phi(y) - A(\eta)\right) \qquad \text{for} \quad \eta = g \circ f(\mathbf{x}^\top \mathbf{w}) \tag{45}$$

Negative log-likelihood estimation:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \ln p(y_n|\mathbf{x}_n^\top \mathbf{w}) \tag{46}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} (\ln(h(y_n)) + \eta_n \phi(y_n) - A(\eta_n)) \tag{47}$$

Negative log-likelihood estimation:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \ln p(y_n|\mathbf{x}_n^\top \mathbf{w}) \tag{46}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} (\ln(h(y_n)) + \eta_n \phi(y_n) - A(\eta_n)) \tag{47}$$

If we rewrite this sum by using the matrix notation, we get

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \phi(y_n) - \nabla_{\mathbf{w}} A(\eta_n) \tag{48}$$

Negative log-likelihood estimation:

$$\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \ln p(y_n|\mathbf{x}_n^\top \mathbf{w}) \tag{46}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} \left(\ln(h(y_n)) + \eta_n \phi(y_n) - A(\eta_n)\right) \tag{47}$$

If we rewrite this sum by using the matrix notation, we get

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \phi(y_n) - \nabla_{\mathbf{w}} A(\eta_n) \tag{48}$$

In the case of Logistic Regression:

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}) = \frac{1}{N}\mathbf{X}^\top \left[\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}\right] \tag{49}$$

Some examples

- Gaussian distribution $\Longrightarrow$ Least Squares

Some examples

- Gaussian distribution $\Longrightarrow$ Least Squares
- Bernoulli distribution $\Longrightarrow$ Logistic Regression

Some examples

- Gaussian distribution ⟹ Least Squares
- Bernoulli distribution ⟹ Logistic Regression
- Multi-nomial distribution ⟹ Softmax Regression

Table of Contents

1 Review of Last Week

2 Exponential Families and Generalized Linear Models

3 Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
- The Basic Structure of MLP
- Training of NNs and BP

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

MLP from a single neuron view

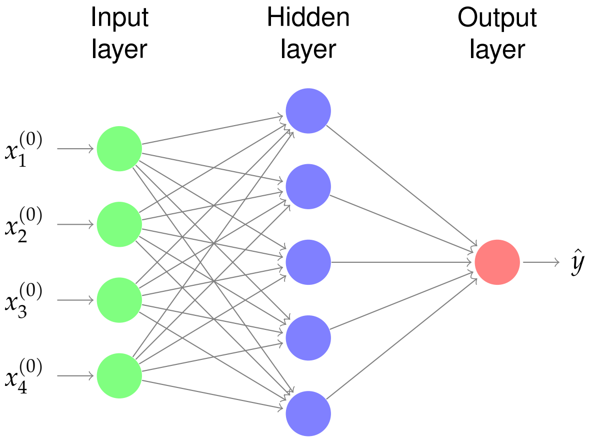


Figure: A simple MLP.

The output at the node j in layer l is denoted by $x_j^{(l)}$ and it is given by

$$x_j^{(l)} = \phi\Big(\sum w_{i,j}^{(l)} x_i^{(l-1)} + b_j^{(l)}\Big). \tag{50}$$

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

  This function is defined by

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

  This function is defined by

  - The biases $\{\mathbf{b}^{(l)}\}_{l \in [L]}$ and weights $\{\mathbf{W}^{(l)}\}_{l \in [L]}$

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

  This function is defined by

  - The biases $\{\mathbf{b}^{(l)}\}_{l\in[L]}$ and weights $\{\mathbf{W}^{(l)}\}_{l\in[L]}$
  - The activation function σ we pick

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

  This function is defined by
  - The biases $\{\mathbf{b}^{(l)}\}_{l\in[L]}$ and weights $\{\mathbf{W}^{(l)}\}_{l\in[L]}$
  - The activation function σ we pick

  In practice: both L and K are large — over-parameterized NNs.

NNs extract suitable features from the input

A NN can be decomposed into a feature extractor and the output layer.

- Feature extractor $\mathbb{R}^d \to \mathbb{R}^K$: It transforms data into a suitable representation.

  This function is defined by

  - The biases $\{\mathbf{b}^{(l)}\}_{l\in[L]}$ and weights $\{\mathbf{W}^{(l)}\}_{l\in[L]}$
  - The activation function σ we pick

  In practice: both L and K are large — over-parameterized NNs.

- The last layer $\mathbb{R}^K \to \mathbb{R}$: It performs the desired ML task, either linear regression or classification.

Table of Contents

1. Review of Last Week
   - Generalization Gap and Model Selection
   - Bias-Variance Decomposition
   - Logistic Regression

2. Exponential Families and Generalized Linear Models
   - Motivation
   - Exponential family
   - Application in ML (Generalized Linear Models)

3. Multi-Layer Perceptron (MLP) and Back-Propagation (BP)
   - The Basic Structure of MLP
   - Training of NNs and BP

Training loss for a regression problem with $S_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$:

$$\mathcal{L}(f) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \,, \tag{51}$$

where

Training loss for a regression problem with $S_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$:

$$\mathcal{L}(f) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \, , \tag{51}$$

where

- f is the function represented by a NN.

Training loss for a regression problem with $S_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$:

$$\mathcal{L}(f) = \frac{1}{2N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \,, \tag{51}$$

where

- f is the function represented by a NN.
- The overall function $y = f(\mathbf{x}^{(0)})$ can then be written as the composition:

$$f(\mathbf{x}^{(0)}) = f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}^{(0)}).$$

Compact description of output

Compact description of output

- The function that is implemented by each layer in the form

$$\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)})\,. \tag{52}$$

Compact description of output

- The function that is implemented by each layer in the form

$$\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)})\,. \tag{52}$$

- Let $\mathbf{W}^{(l)}$ denote the *weight* matrix that connects layer $l - 1$ to layer l.

Compact description of output

- The function that is implemented by each layer in the form

$$\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)})\,. \tag{52}$$

- Let $\mathbf{W}^{(l)}$ denote the *weight* matrix that connects layer $l-1$ to layer l.
- The matrix $\mathbf{W}^{(1)}$ is of dimension $D \times K$, the matrices $\mathbf{W}^{(l)}$, $2 \leq l \leq L$, are of dimension $K \times K$, and the matrix $\mathbf{W}^{(L+1)}$ is of dimension $K \times 1$.

Compact description of output

- The function that is implemented by each layer in the form

$$\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)})\,. \tag{52}$$

- Let $\mathbf{W}^{(l)}$ denote the *weight* matrix that connects layer $l-1$ to layer l.
- The matrix $\mathbf{W}^{(1)}$ is of dimension $D \times K$, the matrices $\mathbf{W}^{(l)}$, $2 \leq l \leq L$, are of dimension $K \times K$, and the matrix $\mathbf{W}^{(L+1)}$ is of dimension $K \times 1$.
- The entries of each matrix $\mathbf{W}$ are given by

$$\mathbf{W}_{i,j}^{(l)} = w_{i,j}^{(l)}\,, \tag{53}$$

Compact description of output

- The function that is implemented by each layer in the form

$$\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)})\,. \tag{52}$$

- Let $\mathbf{W}^{(l)}$ denote the *weight* matrix that connects layer $l-1$ to layer l.
- The matrix $\mathbf{W}^{(1)}$ is of dimension $D \times K$, the matrices $\mathbf{W}^{(l)}$, $2 \leq l \leq L$, are of dimension $K \times K$, and the matrix $\mathbf{W}^{(L+1)}$ is of dimension $K \times 1$.
- The entries of each matrix $\mathbf{W}$ are given by

$$\mathbf{W}_{i,j}^{(l)} = w_{i,j}^{(l)}\,, \tag{53}$$

where $w_{i,j}^{(l)}$ is the edge weight that connects node i on layer $l-1$ to node j on layer l.

The back-propagation algorithm

Cost function:

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n^{(0)})\right)^2 ,$$

where $\mathbf{x}_n^{(l)} = f^{(l)}(\mathbf{x}_n^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}_n^{(l-1)} + \mathbf{b}^{(l)})$.

The back-propagation algorithm

Cost function:

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n^{(0)})\right)^2 ,$$

where $\mathbf{x}_n^{(l)} = f^{(l)}(\mathbf{x}_n^{(l-1)}) = \phi((\mathbf{W}^{(l)})^\top \mathbf{x}_n^{(l-1)} + \mathbf{b}^{(l)})$.

Recall that we aim to compute:

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}}, \qquad l = 1, \cdots, L+1,$$

$$\frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}}, \qquad l = 1, \cdots, L+1.$$

Let's use two quantities (i.e., $\mathbf{z}^{(l)}$ and $\delta^{(l)}$) to aid the computation:

Let's use two quantities (i.e., $\mathbf{z}^{(l)}$ and $\delta^{(l)}$) to aid the computation:

- Quantity computed in the **forward pass**:

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \tag{54}$$

  be the input at the l-th layer before applying the activation function, where $\mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)})$.

Let’s use two quantities (i.e., $\mathbf{z}^{(l)}$ and $\delta^{(l)}$) to aid the computation:

- Quantity computed in the **forward pass**:

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \tag{54}$$

  be the input at the l-th layer before applying the activation function, where $\mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)})$.

- Quantity computed in the **backward pass**:

$$\begin{aligned}
\delta_j^{(l)} &= \frac{\partial \mathcal{L}_n}{\partial z_j^{(l)}} && (55)\\
&= \sum_k \frac{\partial \mathcal{L}_n}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} && (56)\\
&= \sum_k \delta_k^{(l+1)} \mathbf{W}_{j,k}^{(l+1)} \phi'(z_j^{(l)}) \,, && (57)
\end{aligned}$$

Let's use two quantities (i.e., $\mathbf{z}^{(l)}$ and $\delta^{(l)}$) to aid the computation:

- Quantity computed in the **forward pass**:

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \tag{54}$$

  be the input at the l-th layer before applying the activation function, where $\mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)})$.

- Quantity computed in the **backward pass**:

$$\delta_j^{(l)} = \frac{\partial \mathcal{L}_n}{\partial z_j^{(l)}} \tag{55}$$

$$= \sum_k \frac{\partial \mathcal{L}_n}{\partial z_k^{(l+1)}} \frac{\partial z_k^{(l+1)}}{\partial z_j^{(l)}} \tag{56}$$

$$= \sum_k \delta_k^{(l+1)} \mathbf{W}_{j,k}^{(l+1)} \phi'(z_j^{(l)}) \,, \tag{57}$$

  In vector form, we can write this as

$$\delta^{(l)} = (\mathbf{W}^{(l+1)} \delta^{(l+1)}) \odot \phi'(\mathbf{z}^{(l)}) \,, \tag{58}$$

  where $\odot$ denotes the Hadamard product (the point-wise multiplication of vectors).

Now that we have both $\mathbf{z}^{(l)}$ and $\delta^{(l)}$ let us get back to our initial goal.

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}} = \sum_k \frac{\partial \mathcal{L}_n}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial w_{i,j}^{(l)}} = \underbrace{\frac{\partial \mathcal{L}_n}{\partial z_j^{(l)}}}_{\delta_j^{(l)}} \underbrace{\frac{\partial z_j^{(l)}}{\partial w_{i,j}^{(l)}}}_{\mathbf{x}_i^{(l-1)}} = \delta_j^{(l)} \mathbf{x}_i^{(l-1)}$$

$$\frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}} = \sum_k \frac{\partial \mathcal{L}_n}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial b_j^{(l)}} = \underbrace{\frac{\partial \mathcal{L}_n}{\partial z_j^{(l)}}}_{\delta_j^{(l)}} \underbrace{\frac{\partial z_j^{(l)}}{\partial b_j^{(l)}}}_{1} = \delta_j^{(l)} \cdot 1 = \delta_j^{(l)} \,.$$

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.
- We are given in addition a sample $(\mathbf{x}_n, y_n)$.

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.
- We are given in addition a sample $(\mathbf{x}_n, y_n)$.
- We want to compute the derivatives

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}}\,, \qquad \frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}}\,, \qquad l = 1, \cdots, L+1\,,$$

where

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n)\right)^2.$$

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.
- We are given in addition a sample $(\mathbf{x}_n, y_n)$.
- We want to compute the derivatives

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}}, \qquad \frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}}, \qquad l = 1, \cdots, L+1,$$

where

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n)\right)^2.$$

Forward pass: Set $\mathbf{x}^{(0)} = \mathbf{x}_n$. Compute for $l = 1, \cdots, L+1$,

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}, \qquad \mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)}).$$

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.
- We are given in addition a sample $(\mathbf{x}_n, y_n)$.
- We want to compute the derivatives

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}}, \qquad \frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}}, \qquad l = 1, \cdots, L+1,$$

where

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n)\right)^2.$$

Forward pass: Set $\mathbf{x}^{(0)} = \mathbf{x}_n$. Compute for $l = 1, \cdots, L+1$,

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}, \qquad \mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)}).$$

Backward pass: Set $\delta^{(L+1)} = -2(y_n - \mathbf{x}^{(L+1)})\phi'(z^{(L+1)})$. Compute for $l = L, \cdots 1$,

$$\delta^{(l)} = (\mathbf{W}^{(l+1)}\delta^{(l+1)}) \odot \phi'(\mathbf{z}^{(l)}).$$

Summary: Backpropagation Algorithm for Computing the Derivatives

Settings: We are given a NN with L hidden layers

- All weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$, $l = 1, \cdots, L+1$, are fixed.
- We are given in addition a sample $(\mathbf{x}_n, y_n)$.
- We want to compute the derivatives

$$\frac{\partial \mathcal{L}_n}{\partial w_{i,j}^{(l)}}, \qquad \frac{\partial \mathcal{L}_n}{\partial b_j^{(l)}}, \qquad l = 1, \cdots, L+1,$$

where

$$\mathcal{L}_n = \left(y_n - f^{(L+1)} \circ \cdots \circ f^{(2)} \circ f^{(1)}(\mathbf{x}_n)\right)^2.$$

Forward pass: Set $\mathbf{x}^{(0)} = \mathbf{x}_n$. Compute for $l = 1, \cdots, L+1$,

$$\mathbf{z}^{(l)} = (\mathbf{W}^{(l)})^\top \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}, \qquad \mathbf{x}^{(l)} = \phi(\mathbf{z}^{(l)}).$$

Backward pass: Set $\delta^{(L+1)} = -2(y_n - \mathbf{x}^{(L+1)})\phi'(z^{(L+1)})$. Compute for $l = L, \cdots 1$,

$$\delta^{(l)} = (\mathbf{W}^{(l+1)}\delta^{(l+1)}) \odot \phi'(\mathbf{z}^{(l)}).$$

Final computation: For all parameters compute

This lecture:

- Exponential Families and Generalized Linear Models
- Multi-Layer Perceptron
- Back-Propagation

This lecture:

- Exponential Families and Generalized Linear Models
- Multi-Layer Perceptron
- Back-Propagation

Next lecture ?