Lecture 14: Regularization

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering
National Taiwan University
(國立台灣大學資訊工程系)
Roadmap

1. When Can Machines Learn?
2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn Better?

Lecture 13: Hazard of Overfitting

overfitting happens with excessive power, stochastic/deterministic noise, and limited data

Lecture 14: Regularization

- Regularized Hypothesis Set
- Weight Decay Regularization
- Regularization and VC Theory
- General Regularizers
Regularization: The Magic

- Regularization: $\text{Regularized Hypothesis Set}$

Data
Target
Fit

overfit
Regularization: The Magic

- 'regularized fit'
- overfit

\[ x \rightarrow y \]

\[ \mathcal{H}_0 \rightarrow \mathcal{H}_1 \rightarrow \mathcal{H}_2 \rightarrow \mathcal{H}_3 \cdots \]

- Idea: 'step back' from \( \mathcal{H}_1 \) to \( \mathcal{H}_2 \)
- Name history: function approximation for ill-posed problems

Hsuan-Tien Lin (NTU CSIE)
Regularization: The Magic

- Idea: ‘step back’ from $\mathcal{H}_1$ to $\mathcal{H}_2$

![Graph showing 'regularized fit' versus overfit](image)

- Regularization: The Magic
  - Regularized Hypothesis Set
  - $\mathcal{H}_0 \subset \mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \ldots$

- Regularized fit

- Overfit
Regularization

Regularized Hypothesis Set

Regularization: The Magic

\[
\begin{align*}
\text{Data} & \quad \text{Target} \\
\text{Fit} & \\
\end{align*}
\]

\[
\begin{align*}
\mathcal{H}_0 & \quad \mathcal{H}_1 & \quad \mathcal{H}_2 & \quad \mathcal{H}_3 & \quad \cdots
\end{align*}
\]

- idea: ‘step back’ from \( \mathcal{H}_{10} \) to \( \mathcal{H}_2 \)

- name history: function approximation for **ill-posed problems**
Regularization: The Magic

- idea: ‘step back’ from $\mathcal{H}_{10}$ to $\mathcal{H}_2$
- name history: function approximation for **ill-posed problems**

how to step back?
Stepping Back as Constraint

Q-th order polynomial transform for $x \in \mathbb{R}$:

$$\Phi_Q(x) = (1, x, x^2, \ldots, x^Q)$$

+ linear regression, denote $\tilde{w}$ by $w$
Stepping Back as Constraint

$\mathcal{H}_0 \subset \mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \ldots$

$Q$-th order polynomial transform for $x \in \mathbb{R}$:

$$\Phi_Q(x) = (1, x, x^2, \ldots, x^Q)$$

+ linear regression, denote $\tilde{w}$ by $w$

hypothesis $w$ in $\mathcal{H}_{10}$:

$$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_{10} x^{10}$$

hypothesis $w$ in $\mathcal{H}_2$:

$$w_0 + w_1 x + w_2 x^2$$
Stepping Back as Constraint

Q-th order polynomial transform for \( x \in \mathbb{R} \):

\[
\Phi_Q(x) = (1, x, x^2, \ldots, x^Q)
\]

+ linear regression, denote \( \tilde{w} \) by \( w \)

hypothesis \( w \) in \( \mathcal{H}_{10} \):
\[
w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_{10} x^{10}
\]

hypothesis \( w \) in \( \mathcal{H}_2 \):
\[
w_0 + w_1 x + w_2 x^2
\]

that is, \( \mathcal{H}_2 = \mathcal{H}_{10} \) AND ‘constraint that \( w_3 = w_4 = \ldots = w_{10} = 0 \)’
Stepping Back as Constraint

$Q$-th order polynomial transform for $x \in \mathbb{R}$:
$$\Phi_Q(x) = (1, x, x^2, \ldots, x^Q)$$

+ linear regression, denote $\tilde{w}$ by $w$

hypothesis $w$ in $\mathcal{H}_{10}$: $$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_{10} x^{10}$$

hypothesis $w$ in $\mathcal{H}_2$: $$w_0 + w_1 x + w_2 x^2$$

that is, $\mathcal{H}_2 = \mathcal{H}_{10}$ AND ‘constraint that $w_3 = w_4 = \ldots = w_{10} = 0$’

step back = constraint
Regression with Constraint

\[ \mathcal{H}_{10} \equiv \{ \mathbf{w} \in \mathbb{R}^{10+1} \} \]

regression with \( \mathcal{H}_{10} \):

\[ \min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{in}(\mathbf{w}) \]
Regularization

Regularized Hypothesis Set

Regression with Constraint

\[ \mathcal{H}_{10} \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \]

regression with \( \mathcal{H}_{10} \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

\[ \mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right. \text{ while } w_3 = w_4 = \ldots = w_{10} = 0 \]
Regression with Constraint

\[ \mathcal{H}_{10} \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \]

regression with \( \mathcal{H}_{10} \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

\[ \mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \]

while \( w_3 = w_4 = \ldots = w_{10} = 0 \)

regression with \( \mathcal{H}_2 \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

s.t. \( w_3 = w_4 = \ldots = w_{10} = 0 \)
Regression with Constraint

\[ \mathcal{H}_{10} \equiv \{ w \in \mathbb{R}^{10+1} \} \]

regression with \( \mathcal{H}_{10} \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w)
\]

\[ \mathcal{H}_2 \equiv \{ w \in \mathbb{R}^{10+1} \] \\
while \( w_3 = w_4 = \ldots = w_{10} = 0 \}

regression with \( \mathcal{H}_2 \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \]

s.t. \( w_3 = w_4 = \ldots = w_{10} = 0 \)

step back = constrained optimization of \( E_{\text{in}} \)
Regression with Constraint

\( \mathcal{H}_{10} \equiv \{ w \in \mathbb{R}^{10+1} \} \)

regression with \( \mathcal{H}_{10} \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w)
\]

\( \mathcal{H}_{2} \equiv \{ w \in \mathbb{R}^{10+1} \}
\quad \text{while } w_3 = w_4 = \ldots = w_{10} = 0 \}

regression with \( \mathcal{H}_{2} \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \quad \text{s.t. } w_3 = w_4 = \ldots = w_{10} = 0
\]

step back = constrained optimization of \( E_{\text{in}} \)

why don’t you just use \( w \in \mathbb{R}^{2+1} \) ? :-)}
\[ \mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \]

while \( w_3 = \ldots = w_{10} = 0 \}

regression with \( \mathcal{H}_2 \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

s.t. \( w_3 = \ldots = w_{10} = 0 \)
\( \mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \)
\[ \text{while } w_3 = \ldots = w_{10} = 0 \]

Regression with \( \mathcal{H}_2 \):
\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]
\[ \text{s.t. } w_3 = \ldots = w_{10} = 0 \]

\( \mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right\} \)
\[ \text{while } \geq 8 \text{ of } w_q = 0 \]
Regularization

Regularized Hypothesis Set

Regression with Looser Constraint

\( \mathcal{H}_2 \equiv \begin{cases} w \in \mathbb{R}^{10+1} \\
\text{while } w_3 = \ldots = w_{10} = 0 \end{cases} \)

regression with \( \mathcal{H}_2 \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \\
\text{s.t. } w_3 = \ldots = w_{10} = 0
\]

\( \mathcal{H}_2' \equiv \begin{cases} w \in \mathbb{R}^{10+1} \\
\text{while } \geq 8 \text{ of } w_q = 0 \end{cases} \)

regression with \( \mathcal{H}_2' \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \\
\text{s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\]
Regression with Looser Constraint

\[ \mathcal{H}_2 \equiv \begin{cases} \mathbf{w} \in \mathbb{R}^{10+1} \\ \text{while } w_3 = \ldots = w_{10} = 0 \end{cases} \]

regression with \( \mathcal{H}_2 \):

\[
\begin{align*}
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} & \quad E_{\text{in}}(\mathbf{w}) \\
\text{s.t.} & \quad w_3 = \ldots = w_{10} = 0
\end{align*}
\]

\[ \mathcal{H}'_2 \equiv \begin{cases} \mathbf{w} \in \mathbb{R}^{10+1} \\ \text{while } \geq 8 \text{ of } w_q = 0 \end{cases} \]

regression with \( \mathcal{H}'_2 \):

\[
\begin{align*}
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} & \quad E_{\text{in}}(\mathbf{w}) \\
\text{s.t.} & \quad \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\end{align*}
\]

• more flexible than \( \mathcal{H}_2 \): \( \mathcal{H}_2 \subset \mathcal{H}'_2 \)
Regularization

Regularized Hypothesis Set

Regression with Looser Constraint

\[ \mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right. \]

\[ \text{while } w_3 = \ldots = w_{10} = 0 \left\} \right. \]

regression with \( \mathcal{H}_2 \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

\[ \text{s.t. } w_3 = \ldots = w_{10} = 0 \]

\[ \mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right. \]

\[ \text{while } \geq 8 \text{ of } w_q = 0 \left\} \right. \]

regression with \( \mathcal{H}'_2 \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})
\]

\[ \text{s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3 \]

- more flexible than \( \mathcal{H}_2 \): \( \mathcal{H}_2 \subset \mathcal{H}'_2 \)
- less risky than \( \mathcal{H}_{10} \): \( \mathcal{H}'_2 \subset \mathcal{H}_{10} \)
Regularization

Regularized Hypothesis Set

Regression with Looser Constraint

\[ \mathcal{H}_2 \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

while \( w_3 = \ldots = w_{10} = 0 \}

regression with \( \mathcal{H}_2 \):

\[
\begin{align*}
\min_{w \in \mathbb{R}^{10+1}} & \quad E_{\text{in}}(w) \\
\text{s.t.} & \quad w_3 = \ldots = w_{10} = 0
\end{align*}
\]

\[ \mathcal{H}'_2 \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

while \( \geq 8 \) of \( w_q = 0 \}

regression with \( \mathcal{H}'_2 \):

\[
\begin{align*}
\min_{w \in \mathbb{R}^{10+1}} & \quad E_{\text{in}}(w) \\
\text{s.t.} & \quad \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\end{align*}
\]

- more flexible than \( \mathcal{H}_2 \): \( \mathcal{H}_2 \subset \mathcal{H}'_2 \)
- less risky than \( \mathcal{H}_{10} \): \( \mathcal{H}_2 \subset \mathcal{H}_{10} \)

bad news for sparse hypothesis set \( \mathcal{H}'_2 \):

NP-hard to solve :-(
\[
\mathcal{H}_2' \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \text{ while } \geq 8 \text{ of } w_q = 0 \right\}
\]

Regression with \( \mathcal{H}_2' \):

\[
\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \sum_{q=0}^{10} \left[ w_q \neq 0 \right] \leq 3
\]
Regression with Softer Constraint

\( \mathcal{H}'_2 \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \)

while \( \geq 8 \) of \( w_q = 0 \)

regression with \( \mathcal{H}'_2 \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \quad \text{s.t.} \quad \sum_{q=0}^{10} \left[ w_q \neq 0 \right] \leq 3
\]

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \quad \text{s.t.} \quad \sum_{q=0}^{10} w_q^2 \leq C
\]
\( \mathcal{H}'_2 \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \)
\[
\text{while } \geq 8 \text{ of } w_q = 0 \}

regression with \( \mathcal{H}'_2 \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\]

\( \mathcal{H}(C) \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \)
\[
\text{while } \|w\|^2 \leq C \}

regression with \( \mathcal{H}(C) \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} w_q^2 \leq C
\]
**Regularization**

**Regularized Hypothesis Set**

**Regression with Softer Constraint**

\[ \mathcal{H}_2' \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

while \( \geq 8 \) of \( w_q = 0 \)

regression with \( \mathcal{H}_2' \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{in}(w) \text{ s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\]

\[ \mathcal{H}(C) \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

while \( \|w\|^2 \leq C \)

regression with \( \mathcal{H}(C) \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{in}(w) \text{ s.t. } \sum_{q=0}^{10} w^2_q \leq C
\]

- \( \mathcal{H}(C) \): overlaps but not exactly the same as \( \mathcal{H}_2' \)
Regression with Softer Constraint

\[ \mathcal{H}_2' \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]
while \( \geq 8 \) of \( w_q = 0 \)

regression with \( \mathcal{H}_2' \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\]

\[ \mathcal{H}(C) \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]
while \( \|w\|^2 \leq C \)

regression with \( \mathcal{H}(C) \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} w_q^2 \leq C
\]

- \( \mathcal{H}(C) \): overlaps but not exactly the same as \( \mathcal{H}_2' \)
- soft and smooth structure over \( C \geq 0 \):
  \( \mathcal{H}(0) \subset \mathcal{H}(1.126) \subset \ldots \subset \mathcal{H}(1126) \subset \ldots \subset \mathcal{H}(\infty) = \mathcal{H}_{10} \)
Regularization

Regularized Hypothesis Set

Regression with Softer Constraint

\[ \mathcal{H}_2' \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

\[
\text{while } \geq 8 \text{ of } w_q = 0
\]

regression with \( \mathcal{H}_2' \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} [w_q \neq 0] \leq 3
\]

\[ \mathcal{H}(C) \equiv \left\{ w \in \mathbb{R}^{10+1} \right\} \]

\[
\text{while } \|w\|^2 \leq C
\]

regression with \( \mathcal{H}(C) \):

\[
\min_{w \in \mathbb{R}^{10+1}} E_{\text{in}}(w) \text{ s.t. } \sum_{q=0}^{10} w_q^2 \leq C
\]

- \( \mathcal{H}(C) \): overlaps but not exactly the same as \( \mathcal{H}_2' \)
- soft and smooth structure over \( C \geq 0 \):
  \[ \mathcal{H}(0) \subset \mathcal{H}(1.126) \subset \ldots \subset \mathcal{H}(1126) \subset \ldots \subset \mathcal{H}(\infty) = \mathcal{H}_{10} \]

**regularized hypothesis** \( w_{\text{REG}} \):

optimal solution from regularized hypothesis set \( \mathcal{H}(C) \)
For $Q \geq 1$, which of the following hypothesis (weight vector $w \in \mathbb{R}^{Q+1}$) is not in the regularized hypothesis set $\mathcal{H}(1)$?

1. $w^T = [0, 0, \ldots, 0]$
2. $w^T = [1, 0, \ldots, 0]$
3. $w^T = [1, 1, \ldots, 1]$
4. $w^T = \left[ \sqrt{\frac{1}{Q+1}}, \sqrt{\frac{1}{Q+1}}, \ldots, \sqrt{\frac{1}{Q+1}} \right]$
Regularization

Regularized Hypothesis Set

Fun Time

For $Q \geq 1$, which of the following hypotheses (weight vector $w \in \mathbb{R}^{Q+1}$) is not in the regularized hypothesis set $\mathcal{H}(1)$?

1. $w^T = [0, 0, \ldots, 0]$
2. $w^T = [1, 0, \ldots, 0]$
3. $w^T = [1, 1, \ldots, 1]$
4. $w^T = \left[ \sqrt{\frac{1}{Q+1}}, \sqrt{\frac{1}{Q+1}}, \ldots, \sqrt{\frac{1}{Q+1}} \right]$

Reference Answer: ③

The squared length of $w$ in ③ is $Q + 1$, which is not $\leq 1$. 
Matrix Form of Regularized Regression Problem

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2
\]

s.t.
\[
\sum_{q=0}^{Q} w_q^2 \leq C
\]
Matrix Form of Regularized Regression Problem

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2
\]

s.t. \[\sum_{q=0}^{Q} w_q^2 \leq C\]

\[
\sum_n \ldots = (Z\mathbf{w} - \mathbf{y})^T(Z\mathbf{w} - \mathbf{y}), \text{ remember? :-)}
\]
Matrix Form of Regularized Regression Problem

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} E_{in}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2 \\
\text{subject to} \quad \sum_{q=0}^{Q} w_q^2 \leq C
\]

- \( \sum_n \cdots = (Z\mathbf{w} - \mathbf{y})^T(Z\mathbf{w} - \mathbf{y}) \), remember? :-)

• Regularization
  • Weight Decay Regularization
Matrix Form of Regularized Regression Problem

\[
\begin{align*}
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \quad & E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2 \\
\text{s.t.} \quad & \sum_{q=0}^{Q} w_q^2 \leq C \\
& \mathbf{w}^T \mathbf{w} \\
\end{align*}
\]

- \( \sum_n \ldots = (\mathbf{Zw} - \mathbf{y})^T (\mathbf{Zw} - \mathbf{y}) \), remember? :-)
- \( \mathbf{w}^T \mathbf{w} \leq C \): feasible \( \mathbf{w} \) within a radius-\( \sqrt{C} \) hypersphere
Matrix Form of Regularized Regression Problem

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^T \mathbf{z}_n - y_n)^2
\]

\[
\text{s.t.} \quad \sum_{q=0}^{Q} w_q^2 \leq C
\]

- \( \sum_n \ldots = (\mathbf{Zw} - \mathbf{y})^T(\mathbf{Zw} - \mathbf{y}) \), remember? :-)
- \( \mathbf{w}^T \mathbf{w} \leq C \): feasible \( \mathbf{w} \) within a radius-\( \sqrt{C} \) hypersphere

how to solve constrained optimization problem?
The Lagrange Multiplier

\[
\min_{w \in \mathbb{R}^{Q+1}} E_{\text{in}}(w) = \frac{1}{N} (Zw - y)^T (Zw - y) \text{ s.t. } w^T w \leq C
\]
The Lagrange Multiplier

$$\min_{w \in \mathbb{R}^{Q+1}} E_{in}(w) = \frac{1}{N}(Zw - y)^T(Zw - y) \text{ s.t. } w^T w \leq C$$

- decreasing direction: $-\nabla E_{in}(w)$, remember? :-)

$E_{in} = \text{const.}$

$\mathbf{w}_{\text{lin}} - \nabla E_{in}$

normal

$\mathbf{w}^T \mathbf{w} = C$
The Lagrange Multiplier

\[
\min_{w \in \mathbb{R}^{Q+1}} E_{\text{in}}(w) = \frac{1}{N}(Zw - y)^T(Zw - y) \quad \text{s.t.} \quad w^T w \leq C
\]

- decreasing direction: \(-\nabla E_{\text{in}}(w)\), remember? :-)
- normal vector of \(w^T w = C\): \(w\)

\(E_{\text{in}} = \text{const.}\)

\(w_{\text{lin}} \quad -\nabla E_{\text{in}} \quad \text{normal}\)

\(w^T w = C\)
Regularization

Weight Decay Regularization

The Lagrange Multiplier

\[
\min_{w \in \mathbb{R}^{Q+1}} E_{in}(w) = \frac{1}{N} (Zw - y)^T (Zw - y) \text{ s.t. } w^T w \leq C
\]

- decreasing direction: \(-\nabla E_{in}(w)\), remember? :-)
- normal vector of \(w^T w = C\): \(w\)
- if \(-\nabla E_{in}(w)\) and \(w\) not parallel: can decrease \(E_{in}(w)\) without violating the constraint
The Lagrange Multiplier

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \quad E_{\text{in}}(\mathbf{w}) = \frac{1}{N}(\mathbf{Zw} - \mathbf{y})^T(\mathbf{Zw} - \mathbf{y}) \quad \text{s.t.} \quad \mathbf{w}^T\mathbf{w} \leq C
\]

- decreasing direction: \(-\nabla E_{\text{in}}(\mathbf{w})\), remember? :-)

- normal vector of \(\mathbf{w}^T\mathbf{w} = C\): \(\mathbf{w}\)

- if \(-\nabla E_{\text{in}}(\mathbf{w})\) and \(\mathbf{w}\) not parallel: can decrease \(E_{\text{in}}(\mathbf{w})\) without violating the constraint

- at optimal solution \(\mathbf{w}_{\text{REG}}\), \(-\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) \propto \mathbf{w}_{\text{REG}}\)
The Lagrange Multiplier

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \ E_{\text{in}}(\mathbf{w}) = \frac{1}{N} (\mathbf{Zw} - \mathbf{y})^T (\mathbf{Zw} - \mathbf{y}) \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{w} \leq C
\]

- decreasing direction: \(-\nabla E_{\text{in}}(\mathbf{w})\), remember? :-)
- normal vector of \(\mathbf{w}^T \mathbf{w} = C\): \(\mathbf{w}\)
- if \(-\nabla E_{\text{in}}(\mathbf{w})\) and \(\mathbf{w}\) not parallel: can decrease \(E_{\text{in}}(\mathbf{w})\) without violating the constraint
- at optimal solution \(\mathbf{w}_{\text{REG}}\), \(-\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) \propto \mathbf{w}_{\text{REG}}\)

want: find Lagrange multiplier \(\lambda > 0\) and \(\mathbf{w}_{\text{REG}}\) such that \(\nabla E_{\text{in}}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N} \mathbf{w}_{\text{REG}} = 0\)
Augmented Error

solving

$$\nabla E_{in}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N}\mathbf{w}_{\text{REG}} = 0$$
Augmented Error

• if oracle tells you $\lambda > 0$, then

solving $\nabla E_{in}(\mathbf{w}_{\text{REG}}) + \frac{2\lambda}{N} \mathbf{w}_{\text{REG}} = 0$
Augmented Error

- if oracle tells you $\lambda > 0$, then

\[ \nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0 \]

\[ \frac{2}{N} \left( Z^T Z w_{\text{REG}} - Z^T y \right) + \frac{2\lambda}{N} w_{\text{REG}} = 0 \]
Augmented Error

- if oracle tells you $\lambda > 0$, then

solving

$$\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

$$\frac{2}{N} (Z^T Z w_{\text{REG}} - Z^T y) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

- optimal solution:

$$w_{\text{REG}} \leftarrow (Z^T Z + \lambda I)^{-1} Z^T y$$

—called ridge regression in Statistics
Augmented Error

- if **oracle** tells you $\lambda > 0$, then

$$\nabla E_{in}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$
Augmented Error

- If oracle tells you $\lambda > 0$, then

$$\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

\[ \text{equivalent to minimizing} \]

Hsuan-Tien Lin (NTU CSIE)  
Machine Learning Foundations  
10/22
Augmented Error

- if oracle tells you $\lambda > 0$, then

$$\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

solving

equivalent to minimizing

$$E_{\text{in}}(w) + \frac{\lambda}{N} w^T w$$
Augmented Error

- if *oracle* tells you $\lambda > 0$, then

$$\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

solving

equivalent to minimizing

$$E_{\text{in}}(w) + \frac{\lambda}{N} w^T w$$

*regularizer*

*augmented error $E_{\text{aug}}(w)$*
Augmented Error

- if oracle tells you $\lambda > 0$, then solving
  $$\nabla E_{in}(w_{REG}) + \frac{2\lambda}{N} w_{REG} = 0$$
equivalent to minimizing
  $$E_{in}(w) + \frac{\lambda}{N} w^T w$$

- regularization with augmented error instead of constrained $E_{in}$
  $$w_{REG} \leftarrow \arg\min_w E_{aug}(w) \text{ for given } \lambda > 0$$
Augmented Error

- if oracle tells you $\lambda > 0$, then
  
  solving
  
  $$\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0$$

  equivalent to minimizing
  
  $$E_{\text{in}}(w) + \frac{\lambda}{N} w^T w$$

  augmented error $E_{\text{aug}}(w)$

- regularization with augmented error instead of constrained $E_{\text{in}}$

  $$w_{\text{REG}} \leftarrow \arg\min_w E_{\text{aug}}(w) \text{ for given } \lambda > 0 \text{ or } \lambda = 0$$
Augmented Error

- if oracle tells you $\lambda > 0$, then

\[
\nabla E_{\text{in}}(w_{\text{REG}}) + \frac{2\lambda}{N} w_{\text{REG}} = 0
\]

solving

\[
\text{equivalent to minimizing}
E_{\text{in}}(w) + \frac{\lambda}{N} w^T w
\]

augmented error $E_{\text{aug}}(w)$

- regularization with augmented error instead of constrained $E_{\text{in}}$

\[
w_{\text{REG}} \leftarrow \arg\min_w E_{\text{aug}}(w) \text{ for given } \lambda > 0 \text{ or } \lambda = 0
\]

minimizing unconstrained $E_{\text{aug}}$ effectively minimizes some $C$-constrained $E_{\text{in}}$
Regularization

Weight Decay Regularization

The Results

\[ \lambda = 0 \]

\[ \lambda = 0.0001 \]

\[ \lambda = 0.01 \]

\[ \lambda = 1 \]

overfitting

\[ \Rightarrow \]

\[ \Rightarrow \]

\[ \Rightarrow \]

underfitting

Data

Target

Fit

φ

哲学：一点点正则化就能走很远！

\[ -\lambda \mathbf{w} \]

\[ \mathbf{w} \text{ gets smaller} \]

- go with ‘any’ transform + linear model

Hsuan-Tien Lin (NTU CSIE)

Machine Learning Foundations

11/22
The Results

\[ \lambda = 0 \]
\[ \lambda = 0.0001 \]
\[ \lambda = 0.01 \]
\[ \lambda = 1 \]

overfitting \[\rightarrow\] \[\rightarrow\] \[\rightarrow\] underfitting

philosophy: a little goes a long way!
Regularization

Weight Decay Regularization

The Results

\( \lambda = 0 \)

\( \lambda = 0.0001 \)

\( \lambda = 0.01 \)

\( \lambda = 1 \)

overfitting

\[ \Rightarrow \]

\[ \Rightarrow \]

\[ \Rightarrow \]

underfitting

philosophy: a little **regularization** goes a long way!
The Results

\[ \lambda = 0 \]
\[ \lambda = 0.0001 \]
\[ \lambda = 0.01 \]
\[ \lambda = 1 \]

overfitting \[\Rightarrow\] \[\Rightarrow\] \[\Rightarrow\] \[\Rightarrow\] underfitting

philosophy: a little regularization goes a long way!

call ‘+\(\frac{\lambda}{N}w^Tw\)’ weight-decay regularization:

larger \(\lambda\)
\[\iff\] prefer shorter \(w\)
\[\iff\] effectively smaller \(C\)
The Results

\[ \lambda = 0 \]

\[ \lambda = 0.0001 \]

\[ \lambda = 0.01 \]

\[ \lambda = 1 \]

Data
Target
Fit

\( \lambda = 0 \) \hspace{1cm} \lambda = 0.0001 \hspace{1cm} \lambda = 0.01 \hspace{1cm} \lambda = 1

overfitting \hspace{1cm} \Rightarrow \hspace{1cm} \Rightarrow \hspace{1cm} \Rightarrow \hspace{1cm} underfitting

Philosophy: *a little regularization goes a long way!*

call ‘\( + \frac{\lambda}{N} w^T w \)’ *weight-decay* regularization:

larger \( \lambda \)

\( \iff \) prefer shorter \( w \)

\( \iff \) effectively smaller \( C \)

—go with ‘any’ transform + linear model
Some Detail: **Legendre Polynomials**

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \frac{1}{N} \sum_{n=0}^{N} (\mathbf{w}^T \Phi(x_n) - y_n)^2 + \frac{\lambda}{N} \sum_{q=0}^{Q} w_q^2
\]
Some Detail: Legendre Polynomials

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \frac{1}{N} \sum_{n=0}^{N} (\mathbf{w}^T \Phi(x_n) - y_n)^2 + \frac{\lambda}{N} \sum_{q=0}^{Q} w_q^2
\]

 naïve polynomial transform:

\[
\Phi(x) = (1, x, x^2, \ldots, x^Q)
\]
Some Detail: Legendre Polynomials

$$\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \frac{1}{N} \sum_{n=0}^{N} (\mathbf{w}^T \Phi(x_n) - y_n)^2 + \frac{\lambda}{N} \sum_{q=0}^{Q} w_q^2$$

naïve polynomial transform:

$$\Phi(x) = (1, x, x^2, \ldots, x^Q)$$

—when $x_n \in [-1, +1]$, $x_n^q$ really small, needing large $w_q$
Some Detail: **Legendre Polynomials**

Regularization

Weight Decay Regularization

\[
\min_{\mathbf{w} \in \mathbb{R}^{Q+1}} \frac{1}{N} \sum_{n=0}^{N} (\mathbf{w}^T \Phi(x_n) - y_n)^2 + \frac{\lambda}{N} \sum_{q=0}^{Q} w_q^2
\]

 naïve polynomial transform:

\[
\Phi(x) = (1, x, x^2, \ldots, x^Q)
\]

—when \( x_n \in [-1, +1] \), \( x_n^q \) really small, needing large \( w_q \)

normalized polynomial transform:

\[
(1, L_1(x), L_2(x), \ldots, L_Q(x))
\]

—‘orthonormal basis functions’ called **Legendre polynomials**

\[
L_1 = x, \quad L_2 = \frac{1}{2}(3x^2 - 1), \quad L_3 = \frac{1}{8}(5x^3 - 3x), \quad L_4 = \frac{1}{8}(35x^4 - 30x^2 + 3), \quad L_5 = \frac{1}{8}(63x^5 \ldots)
\]
When would $w_{\text{REG}}$ equal $w_{\text{LIN}}$?

1. $\lambda = 0$
2. $C = \infty$
3. $C \geq \|w_{\text{LIN}}\|^2$
4. all of the above
When would $w_{\text{REG}}$ equal $w_{\text{LIN}}$?

1. $\lambda = 0$
2. $C = \infty$
3. $C \geq \|w_{\text{LIN}}\|^2$
4. all of the above

Reference Answer: 4

1 and 2 shall be easy; 3 means that there are effectively no constraint on $w$, hence the equivalence.
Regularization and VC Theory

Regularization by Constrained-Minimizing $E_{\text{in}}$

$$\min_w E_{\text{in}}(w) \text{ s.t. } w^T w \leq C$$
Regularization by Constrained-Minimizing $E_{in}$

$$\min_w E_{in}(w) \text{ s.t. } w^T w \leq C$$

$C$ equivalent to some $\lambda$

VC Guarantee of Constrained-Minimizing $E_{in}$: indirectly getting VC guarantee without confining to $H(C)$
Regularization and VC Theory

Regularization by Constrained-Minimizing $E_{in}$

$$\min_w E_{in}(w) \text{ s.t. } w^T w \leq C$$

$\uparrow \downarrow$

$C$ equivalent to some $\lambda$

Regularization by Minimizing $E_{aug}$

$$\min_w E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} w^T w$$
Regularization and VC Theory

Regularization by Constrained-Minimizing $E_{in}$

$$\min_w E_{in}(w) \text{ s.t. } w^T w \leq C$$

$\Uparrow \Downarrow$ $C$ equivalent to some $\lambda$

Regularization by Minimizing $E_{aug}$

$$\min_w E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} w^T w$$

VC Guarantee of Constrained-Minimizing $E_{in}$

$$E_{out}(w) \leq E_{in}(w) + \Omega(\mathcal{H}(C))$$
Regularization and VC Theory

**Regularization by Constrained-Minimizing $E_{in}$**

$$\min_w E_{in}(w) \text{ s.t. } w^T w \leq C$$

$C$ equivalent to some $\lambda$

** VC Guarantee of Constrained-Minimizing $E_{in}$**

$$E_{out}(w) \leq E_{in}(w) + \Omega(H(C))$$

**Regularization by Minimizing $E_{aug}$**

$$\min_w E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} w^T w$$

minimizing $E_{aug}$: indirectly getting VC guarantee without confining to $H(C)$
Another View of Augmented Error

Augmented Error

\[ E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \mathbf{w}^T \mathbf{w} \]
Another View of Augmented Error

**Augmented Error**

\[
E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \mathbf{w}^T \mathbf{w}
\]

**VC Bound**

\[
E_{\text{out}}(\mathbf{w}) \leq E_{\text{in}}(\mathbf{w}) + \Omega(\mathcal{H})
\]
Another View of Augmented Error

Augmented Error

$$E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} w^T w$$

VC Bound

$$E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(\mathcal{H})$$

- regularizer $w^T w$ : complexity of a single hypothesis
Another View of Augmented Error

Augmented Error

\[ E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} w^T w \]

VC Bound

\[ E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(H) \]

- regularizer \( w^T w \): complexity of a single hypothesis
- generalization price \( \Omega(H) \): complexity of a hypothesis set
Another View of Augmented Error

**Augmented Error**

\[ E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \mathbf{w}^T \mathbf{w} \]

**VC Bound**

\[ E_{\text{out}}(\mathbf{w}) \leq E_{\text{in}}(\mathbf{w}) + \Omega(\mathcal{H}) \]

- regularizer \( \mathbf{w}^T \mathbf{w} = \Omega(\mathbf{w}) \): complexity of a single hypothesis
- generalization price \( \Omega(\mathcal{H}) \): complexity of a hypothesis set
Another View of Augmented Error

**Augmented Error**

\[ E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} w^T w \]

**VC Bound**

\[ E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(\mathcal{H}) \]

- regularizer \( w^T w = \Omega(w) \): complexity of a single hypothesis
- generalization price \( \Omega(\mathcal{H}) \): complexity of a hypothesis set
- if \( \frac{\lambda}{N} \Omega(w) \) ‘represents’ \( \Omega(\mathcal{H}) \) well,
  \[ E_{\text{aug}} \text{ is a better proxy of } E_{\text{out}} \text{ than } E_{\text{in}} \]
Another View of Augmented Error

Augmented Error

\[ E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} w^T w \]

VC Bound

\[ E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(H) \]

- regularizer \( w^T w = \Omega(w) \): complexity of a single hypothesis
- generalization price \( \Omega(H) \): complexity of a hypothesis set
- if \( \frac{\lambda}{N} \Omega(w) \) ‘represents’ \( \Omega(H) \) well,
  \[ E_{\text{aug}} \text{ is a better proxy of } E_{\text{out}} \text{ than } E_{\text{in}} \]

minimizing \( E_{\text{aug}} \):

(heuristically) operating with the better proxy;
Another View of Augmented Error

Augmented Error

\[ E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} w^T w \]

VC Bound

\[ E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(H) \]

- regularizer \( w^T w = \Omega(w) \): complexity of a single hypothesis
- generalization price \( \Omega(H) \): complexity of a hypothesis set
- if \( \frac{\lambda}{N} \Omega(w) \) ‘represents’ \( \Omega(H) \) well,
  \[ E_{\text{aug}} \] is a better proxy of \( E_{\text{out}} \) than \( E_{\text{in}} \)

minimizing \( E_{\text{aug}} \):

(heuristically) operating with the better proxy;
(technically) enjoying flexibility of whole \( H \)
Effective VC Dimension

\[
\min_{w \in \mathbb{R}^{d+1}} E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} \Omega(w)
\]

- model complexity?
Effective VC Dimension

\[
\min_{w \in \mathbb{R}^{\tilde{d}+1}} E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} \Omega(w)
\]

- model complexity?

\[d_{\text{VC}}(\mathcal{H}) = \tilde{d} + 1, \text{ because } \{w\} \text{ ‘all considered’ during minimization}\]
Effective VC Dimension

\[
\min_{\mathbf{w} \in \mathbb{R}^{\tilde{d}+1}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \Omega(\mathbf{w})
\]

- model complexity?
  \(d_{\text{VC}}(\mathcal{H}) = \tilde{d} + 1\), because \(\{\mathbf{w}\}\) ‘all considered’ during minimization
- \(\{\mathbf{w}\}\) ‘actually needed’:
Effective VC Dimension

\[
\min_{w \in \mathbb{R}^{\tilde{d}+1}} E_{\text{aug}}(w) = E_{\text{in}}(w) + \frac{\lambda}{N} \Omega(w)
\]

- model complexity?
  \(d_{\text{VC}}(\mathcal{H}) = \tilde{d} + 1\), because \(\{w\}\) ‘all considered’ during minimization
- \(\{w\}\) ‘actually needed’: \(\mathcal{H}(C)\), with some \(C\) equivalent to \(\lambda\)
Effective VC Dimension

\[
\min_{\mathbf{w} \in \mathbb{R}^{\tilde{d}+1}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \Omega(\mathbf{w})
\]

- model complexity?
  \(d_{\text{VC}}(\mathcal{H}) = \tilde{d} + 1\), because \(\{\mathbf{w}\}\) ‘all considered’ during minimization

- \(\{\mathbf{w}\}\) ‘actually needed’: \(\mathcal{H}(C)\), with some \(C\) equivalent to \(\lambda\)

- \(d_{\text{VC}}(\mathcal{H}(C))\):
  effective VC dimension \(d_{\text{EFF}}(\mathcal{H}, \mathcal{A})\)

\[
\min_{\mathbf{w}} E_{\text{aug}}(\mathbf{w})
\]
Effective VC Dimension

\[
\min_{w \in \mathbb{R}^{\tilde{d}+1}} E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N} \Omega(w)
\]

- model complexity?
  \(d_{VC}(\mathcal{H}) = \tilde{d} + 1\), because \(\{w\}\) ‘all considered’ during minimization
- \(\{w\}\) ‘actually needed’:
  \(\mathcal{H}(C)\), with some \(C\) equivalent to \(\lambda\)
- \(d_{VC}(\mathcal{H}(C))\):
  effective VC dimension \(d_{EFF}(\mathcal{H}, A)\)

explanation of regularization:
  \(d_{VC}(\mathcal{H})\) large,
  while \(d_{EFF}(\mathcal{H}, A)\) small if \(A\) regularized
Consider the weight-decay regularization with regression. When increasing $\lambda$ in $\mathcal{A}$, what would happen with $d_{\text{EFF}}(\mathcal{H},\mathcal{A})$?

1. $d_{\text{EFF}} \uparrow$
2. $d_{\text{EFF}} \downarrow$
3. $d_{\text{EFF}} = d_{\text{VC}}(\mathcal{H})$ and does not depend on $\lambda$
4. $d_{\text{EFF}} = 1126$ and does not depend on $\lambda$
Fun Time

Consider the weight-decay regularization with regression. When increasing $\lambda$ in $A$, what would happen with $d_{\text{EFF}}(H, A)$?

1. $d_{\text{EFF}} \uparrow$
2. $d_{\text{EFF}} \downarrow$
3. $d_{\text{EFF}} = d_{\text{VC}}(H)$ and does not depend on $\lambda$
4. $d_{\text{EFF}} = 1126$ and does not depend on $\lambda$

Reference Answer: 2

- larger $\lambda$
- $\iff$ smaller $C$
- $\iff$ smaller $\mathcal{H}(C)$
- $\iff$ smaller $d_{\text{EFF}}$
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

**Symmetry Regularizer:**
$$\sum_{J} q K w^2_{q}$$

**Sparsity (L1) Regularizer:**
$$\sum_{q} |w_{q}|$$

**Weight-decay (L2) Regularizer:**
$$\sum_{q} w_{q}^2$$

**Augmented Error:**
$$\hat{\text{err}} + \text{regularizer}$$

regularizer: target-dependent, plausible, or friendly

error measure: user-dependent, plausible, or friendly

no worries, guard by $\lambda$
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$

- plausible: direction towards smoother or simpler

- stochastic/deterministic noise both non-smooth

- sparsity (L1) regularizer: $\sum |w_q|$

- friendly: easy to optimize

- weight-decay (L2) regularizer: $\sum w_q^2$

- bad? :-)
  - no worries, guard by $\lambda$

augmented error = error $\hat{\text{err}}$ + regularizer

$\Omega$ regularizer: target-dependent, plausible, or friendly

error measure: user-dependent, plausible, or friendly
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler

$augmented\ error = error + \text{regularizer}$

regularizer: target-dependent, plausible, or friendly

error measure: user-dependent, plausible, or friendly
General Regularizers $\Omega(w)$

**want:** constraint in the *direction* of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w^2_q$

- plausible: direction towards **smoother or simpler**

  stochastic/deterministic noise both **non-smooth**
Regularization

General Regularizers

General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler
  stochastic/deterministic noise both non-smooth
- sparsity (L1) regularizer: $\sum |w_q|$ (next slide)

Hsuan-Tien Lin (NTU CSIE)  Machine Learning Foundations

18/22
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$

- plausible: direction towards smoother or simpler

  stochastic/deterministic noise both non-smooth

- sparsity (L1) regularizer: $\sum |w_q|$ (next slide)

- friendly: easy to optimize
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] \, w_q^2$
- plausible: direction towards smoother or simpler
  stochastic/deterministic noise both non-smooth
  - sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
General Regularizers $Ω(\mathbf{w})$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards simpler or smoother
  - stochastic/deterministic noise both non-smooth
  - sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? :-):
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler
  stochastic/deterministic noise both non-smooth
  
  - sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? :-(: no worries, guard by $\lambda$
General Regularizers $\Omega(w)$

want: constraint in the ‘**direction**’ of target function

- target-dependent: some **properties** of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] \ w_q^2$
- plausible: direction towards **smoother or simpler**
  - stochastic/deterministic noise both **non-smooth**
  - **sparsity** (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - **weight-decay** (L2) regularizer: $\sum w_q^2$
- bad? :-): no worries, guard by $\lambda$

regularizer: target-dependent, plausible, or friendly
Regularization

General Regularizers

General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler
  - stochastic/deterministic noise both non-smooth
- sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? :-) : no worries, guard by $\lambda$

regularizer: target-dependent, plausible, or friendly

ringing a bell? :-)
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler
  stochastic/deterministic noise both non-smooth
  - sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? :-): no worries, guard by $\lambda$

regularizer: target-dependent, plausible, or friendly
ringing a bell? :-) 
error measure: user-dependent, plausible, or friendly
General Regularizers $\Omega(w)$

want: constraint in the ‘direction’ of target function

- target-dependent: some properties of target, if known
  - symmetry regularizer: $\sum [q \text{ is odd}] w_q^2$
- plausible: direction towards smoother or simpler
  stochastic/deterministic noise both non-smooth
- sparsity (L1) regularizer: $\sum |w_q|$ (next slide)
- friendly: easy to optimize
  - weight-decay (L2) regularizer: $\sum w_q^2$
- bad? :-): no worries, guard by $\lambda$

augmented error = error $\hat{\text{err}}$ + regularizer $\Omega$

regularizer: target-dependent, plausible, or friendly

ringing a bell? :-)

error measure: user-dependent, plausible, or friendly
L2 and L1 Regularizer

\[ E_{\text{in}} = \text{const.} \]

\[ w_{\text{lin}} - \nabla E_{\text{in}} \]

\[ w^T w = C \]

L2 Regularizer

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|_2^2 \]
L2 and L1 Regularizer

\( E_{in} = \text{const.} \)

\[ w_{\text{lin}} \]

\[ -\nabla E_{in} \]

\[ w_{\text{lin}} \]

\[ w^T w = C \]

**L2 Regularizer**

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \| w \|_2^2 \]

- convex, differentiable everywhere

\[ \nabla E_{in} \]

\[ \text{normal} \]
**L2 Regularizer**

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|_2^2 \]

- convex, differentiable everywhere
- easy to optimize
L2 and L1 Regularizer

**L2 Regularizer**

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|_2^2 \]

- convex, differentiable everywhere
- easy to optimize

**L1 Regularizer**

\[ \Omega(w) = \sum_{q=0}^{Q} |w_q| = \|w\|_1 \]
**Regularization**

**General Regularizers**

### L2 and L1 Regularizer

**L2 Regularizer**

\[
\Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|^2_2
\]

- convex, differentiable everywhere
- easy to optimize

**L1 Regularizer**

\[
\Omega(w) = \sum_{q=0}^{Q} |w_q| = \|w\|_1
\]

- convex, **not** differentiable everywhere

---

Hsuan-Tien Lin  (NTU CSIE)
L2 and L1 Regularizer

L2 Regularizer

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|_2^2 \]

- convex, differentiable everywhere
- easy to optimize

L1 Regularizer

\[ \Omega(w) = \sum_{q=0}^{Q} |w_q| = \|w\|_1 \]

- convex, not differentiable everywhere
- sparsity in solution
L2 Regularizer

\[ \Omega(w) = \sum_{q=0}^{Q} w_q^2 = \|w\|_2^2 \]

- convex, differentiable everywhere
- easy to optimize

L1 Regularizer

\[ \Omega(w) = \sum_{q=0}^{Q} |w_q| = \|w\|_1 \]

- convex, not differentiable everywhere
- sparsity in solution

L1 useful if needing **sparse solution**
The Optimal $\lambda$

**stochastic noise**

- $\sigma^2 = 0.5$
- $\sigma^2 = 0.25$
- $\sigma^2 = 0$

### Regularization Parameters
- $\lambda$

### Expected $E_{out}$
- $Q_f = 15$
- $Q_f = 30$
- $Q_f = 100$

- More noise $\Rightarrow$ more regularization needed
- More bumpy road $\Rightarrow$ putting brakes more

- Noise unknown — important to make proper choices

To choose: stay tuned for the next lecture! :-)

Hsuan-Tien Lin (NTU CSIE)
The Optimal $\lambda$

**stochastic noise**

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>$E_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.25</td>
<td>0.75</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**deterministic noise**

<table>
<thead>
<tr>
<th>$Q_f$</th>
<th>$E_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.6</td>
</tr>
<tr>
<td>30</td>
<td>0.4</td>
</tr>
<tr>
<td>100</td>
<td>0.2</td>
</tr>
</tbody>
</table>

- More noise $\Rightarrow$ More regularization needed
- More bumpy road $\Rightarrow$ Putting brakes more
- Noise unknown $\Rightarrow$ Important to make proper choices

How to choose? Stay tuned for the next lecture! :-)

Hsuan-Tien Lin (NTU CSIE)
The Optimal $\lambda$

**Stochastic noise**
- $\sigma^2 = 0.5$
- $\sigma^2 = 0.25$
- $\sigma^2 = 0$

**Deterministic noise**
- $Q_f = 15$
- $Q_f = 30$
- $Q_f = 100$

- More noise $\iff$ more regularization needed
  --- More bumpy road $\iff$ putting brakes more

- Expected $E_{out}$ vs. Regularization Parameter $\lambda$
The Optimal $\lambda$

- more noise $\iff$ more regularization needed
  —more bumpy road $\iff$ putting brakes more
- noise **unknown**$\iff$ important to **make proper choices**
Regularization

General Regularizers

The Optimal $\lambda$

stochastic noise

- more noise $\iff$ more regularization needed
  —more bumpy road $\iff$ putting brakes more

- noise unknown—important to make proper choices

how to choose?

stay tuned for the next lecture! :-)
Consider using a regularizer $\Omega(w) = \sum_{q=0}^{Q} 2^q w_q^2$ to work with Legendre polynomial regression. Which kind of hypothesis does the regularizer prefer?

1. symmetric polynomials satisfying $h(x) = h(-x)$
2. low-dimensional polynomials
3. high-dimensional polynomials
4. no specific preference

Reference Answer: 2

There is a higher 'penalty' for higher-order terms, and hence the regularizer prefers low-dimensional polynomials.
Consider using a regularizer \( \Omega(w) = \sum_{q=0}^{Q} 2^q w_q^2 \) to work with Legendre polynomial regression. Which kind of hypothesis does the regularizer prefer?

1. symmetric polynomials satisfying \( h(x) = h(-x) \)
2. low-dimensional polynomials
3. high-dimensional polynomials
4. no specific preference

Reference Answer: 2

There is a higher ‘penalty’ for higher-order terms, and hence the regularizer prefers low-dimensional polynomials.
Summary

1. When Can Machines Learn?
2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn Better?

Lecture 13: Hazard of Overfitting

Lecture 14: Regularization

- Regularized Hypothesis Set
  \[ \mathcal{H} + \text{constraint} \]
- Weight Decay Regularization
  \[ \text{add } \frac{\lambda}{N} w^T w \text{ in } E_{\text{aug}} \]
- Regularization and VC Theory
  \[ \text{regularization decreases } d_{\text{EFF}} \]
- General Regularizers
  \[ \text{target-dependent, [plausible], or [friendly]} \]

next: choosing from the so-many models/parameters