# Deep Learning Book

## Chapter 7
## Regularization for Deep Learning

Botian Shi
botianshi@bit.edu.cn

March 7, 2017

You can download the LaTeX source code of this file from Here.

# Generalization and Strategy

- How to make an algorithm that will perform well not just on the training data, but also on new inputs?
- Many strategies designed to reduce the test error, possibly at the expense of increased training error.
- These strategies are known collectively as **regularization**.
- Many regularization algorithm have been developed.
- Developing more effective regularization strategies is one of the major research efforts in the field.
- In this chapter, we describe regularization in more detail, focusing on regularization strategies for deep models or models that may be used as building blocks to form deep models.

## Generalization and Strategy

- How to make an algorithm that will perform well not just on the training data, but also on new inputs?
- Many strategies designed to reduce the test error, possibly at the expense of increased training error.
- These strategies are known collectively as **regularization**.
- Many regularization algorithm have been developed.
- Developing more effective regularization strategies is one of the major research efforts in the field.
- In this chapter, we describe regularization in more detail, focusing on regularization strategies for deep models or models that may be used as building blocks to form deep models.

# Generalization and Strategy

- How to make an algorithm that will perform well not just on the training data, but also on new inputs?
- Many strategies designed to reduce the test error, possibly at the expense of increased training error.
- These strategies are known collectively as **regularization**.
- Many regularization algorithm have been developed.
- Developing more effective regularization strategies is one of the major research efforts in the field.
- In this chapter, we describe regularization in more detail, focusing on regularization strategies for deep models or models that may be used as building blocks to form deep models.

# Generalization and Strategy

- How to make an algorithm that will perform well not just on the training data, but also on new inputs?
- Many strategies designed to reduce the test error, possibly at the expense of increased training error.
- These strategies are known collectively as **regularization**.
- Many regularization algorithm have been developed.
- Developing more effective regularization strategies is one of the major research efforts in the field.
- In this chapter, we describe regularization in more detail, focusing on regularization strategies for deep models or models that may be used as building blocks to form deep models.

# Generalization and Strategy

- How to make an algorithm that will perform well not just on the training data, but also on new inputs?
- Many strategies designed to reduce the test error, possibly at the expense of increased training error.
- These strategies are known collectively as **regularization**.
- Many regularization algorithm have been developed.
- Developing more effective regularization strategies is one of the major research efforts in the field.
- In this chapter, we describe regularization in more detail, focusing on regularization strategies for deep models or models that may be used as building blocks to form deep models.

# Generalization and Strategy

- There are many regularization strategies.
  1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
  2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
  1. **encode** specific kinds of **prior knowledge**.
  2. Express a generic preference for a simpler model class in order to promote generalization.
  3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- There are many regularization strategies.
  1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
  2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
  1. **encode** specific kinds of **prior knowledge**.
  2. Express a generic preference for a simpler model class in order to promote generalization.
  3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- There are many regularization strategies.
  1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
  2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
  1. **encode** specific kinds of **prior knowledge**.
  2. Express a generic preference for a simpler model class in order to promote generalization.
  3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- There are many regularization strategies.
  1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
  2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
  1. **encode** specific kinds of **prior knowledge**.
  2. Express a generic preference for a simpler model class in order to promote generalization.
  3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- There are many regularization strategies.
  1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
  2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
  1. **encode** specific kinds of **prior knowledge**.
  2. Express a generic preference for a simpler model class in order to promote generalization.
  3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- There are many regularization strategies.
    1. Put extra constrains on a machine learning model. (Adding restrictions on the parameter values.)
    2. Add extra terms in the objective function that can be thought of as corresponding to a soft constraint on the parameter values.
- If chosen carefully, these extra constraints and penalties can lead to improved performance on the test set.
- Sometimes these constraints and penalties are designed to
    1. **encode** specific kinds of **prior knowledge**.
    2. Express a generic preference for a simpler model class in order to promote generalization.
    3. make an under-determined problem determined. (Provide more information)
- Other forms of regularization, known as ensemble methods, combine multiple hypotheses that explain the training data.

# Generalization and Strategy

- Treading increased bias for reduced variance.
- An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.
- We now review several strategies for how to create such a large, deep, regularized model.

# Generalization and Strategy

- Treading increased bias for reduced variance.
- An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.
- We now review several strategies for how to create such a large, deep, regularized model.

# Generalization and Strategy

- Treading increased bias for reduced variance.
- An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.
- We now review several strategies for how to create such a large, deep, regularized model.

# Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:

$$\tilde{J}(\boldsymbol{\theta}; X, \boldsymbol{y}) = J(\boldsymbol{\theta}; X, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

# Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:

$$\tilde{J}(\boldsymbol{\theta}; X, \boldsymbol{y}) = J(\boldsymbol{\theta}; X, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

# Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:

$$\tilde{J}(\boldsymbol{\theta}; X, y) = J(\boldsymbol{\theta}; X, y) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

# Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:

  $$\tilde{J}(\boldsymbol{\theta}; X, \boldsymbol{y}) = J(\boldsymbol{\theta}; X, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

## Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:

$$\tilde{J}(\boldsymbol{\theta}; X, y) = J(\boldsymbol{\theta}; X, y) + \alpha\Omega(\boldsymbol{\theta})$$

  where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

# Parameter Norm Penalties

- Regularization has been used for decades prior to the advent of deep learning.
- Linear models allow simple straightforward and effective regularization strategies.
- Most approaches are based on limiting the capacity of models by adding a **parameter norm penalty** $\Omega(\theta)$ to the objective function $J$:
$$\tilde{J}(\boldsymbol{\theta}; X, y) = J(\boldsymbol{\theta}; X, y) + \alpha\Omega(\boldsymbol{\theta})$$
where $\alpha \in [0, +\infty)$ weights the relative contribution of the norm penalty term.
- Setting $\alpha$ to 0 results in no regularization. Larger values of $\alpha$ correspond to more regularization.
- Optimize both $J$ and norm
- Different $\Omega$ has different result.

# Parameter Norm Penalties

- We penalize **only the weights** of the affine transformation at each layer and leaves the biases unregularized.
- We do not induce too much variance by leaving the biases unregularized.
- Regularizing the bias parameters can introduce a significant amount of under-fitting.
- We therefore use the vector $\boldsymbol{w}$ to indicate all of the weights that should be affected by a norm penalty, while the vector $\boldsymbol{\theta}$ denotes all of the parameters, including both $\boldsymbol{w}$ and the unregularized parameters.

# Parameter Norm Penalties

- We penalize **only the weights** of the affine transformation at each layer and leaves the biases unregularized.
- We do not induce too much variance by leaving the biases unregularized.
- Regularizing the bias parameters can introduce a significant amount of under-fitting.
- We therefore use the vector $\boldsymbol{w}$ to indicate all of the weights that should be affected by a norm penalty, while the vector $\boldsymbol{\theta}$ denotes all of the parameters, including both $\boldsymbol{w}$ and the unregularized parameters.

# Parameter Norm Penalties

- We penalize **only the weights** of the affine transformation at each layer and leaves the biases unregularized.
- We do not induce too much variance by leaving the biases unregularized.
- Regularizing the bias parameters can introduce a significant amount of under-fitting.
- We therefore use the vector $\boldsymbol{w}$ to indicate all of the weights that should be affected by a norm penalty, while the vector $\boldsymbol{\theta}$ denotes all of the parameters, including both $\boldsymbol{w}$ and the unregularized parameters.

# Parameter Norm Penalties

- We penalize **only the weights** of the affine transformation at each layer and leaves the biases unregularized.
- We do not induce too much variance by leaving the biases unregularized.
- Regularizing the bias parameters can introduce a significant amount of under-fitting.
- We therefore use the vector $\boldsymbol{w}$ to indicate all of the weights that should be affected by a norm penalty, while the vector $\boldsymbol{\theta}$ denotes all of the parameters, including both $\boldsymbol{w}$ and the unregularized parameters.

# $L^2$ Parameter Regularization

- The $L^2$ norm penalty commonly known as *weight decay*.

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|w\|_2^2$$

- We can gain some insight into the behavior of weight decay regularization.

$$\tilde{J}(w; X, y) = \frac{\alpha}{2} w^T w + J(w; X, y)$$

$$\nabla_w \tilde{J}(w; X, y) = \alpha w + \nabla_w J(w; X, y)$$

- The update

$$w \leftarrow w - \epsilon(\alpha w + \nabla_w J(w; X, y))$$

$$w \leftarrow (1 - \epsilon\alpha) w - \epsilon \nabla_w J(w; X, y)$$

- Shrink the weight vector by a constant factor on each step.
- What happens over the entire course of training?

# $L^2$ Parameter Regularization

- The $L^2$ norm penalty commonly known as *weight decay*.

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|w\|_2^2$$

- We can gain some insight into the behavior of weight decay regularization.

$$\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

$$\nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- The update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon(\alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}))$$

$$\boldsymbol{w} \leftarrow (1 - \epsilon\alpha)\boldsymbol{w} - \epsilon\nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- Shrink the weight vector by a constant factor on each step.
- What happens over the entire course of training?

# $L^2$ Parameter Regularization

- The $L^2$ norm penalty commonly known as *weight decay*.

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|w\|_2^2$$

- We can gain some insight into the behavior of weight decay regularization.

$$\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

$$\nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- The update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon(\alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}))$$

$$\boldsymbol{w} \leftarrow (1 - \epsilon\alpha)\boldsymbol{w} - \epsilon\nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- Shrink the weight vector by a constant factor on each step.
- What happens over the entire course of training?

# $L^2$ Parameter Regularization

- The $L^2$ norm penalty commonly known as *weight decay*.

$$\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|w\|_2^2$$

- We can gain some insight into the behavior of weight decay regularization.

$$\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

$$\nabla_{\boldsymbol{w}}\tilde{J}(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}) = \alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- The update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon(\alpha\boldsymbol{w} + \nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}))$$

$$\boldsymbol{w} \leftarrow (1 - \epsilon\alpha)\boldsymbol{w} - \epsilon\nabla_{\boldsymbol{w}}J(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y})$$

- Shrink the weight vector by a constant factor on each step.
- What happens over the entire course of training?

# References