Introduction to Deep Learning

7. Model Selection, Weight Decay, Dropout

STAT 157, Spring 2019, UC Berkeley

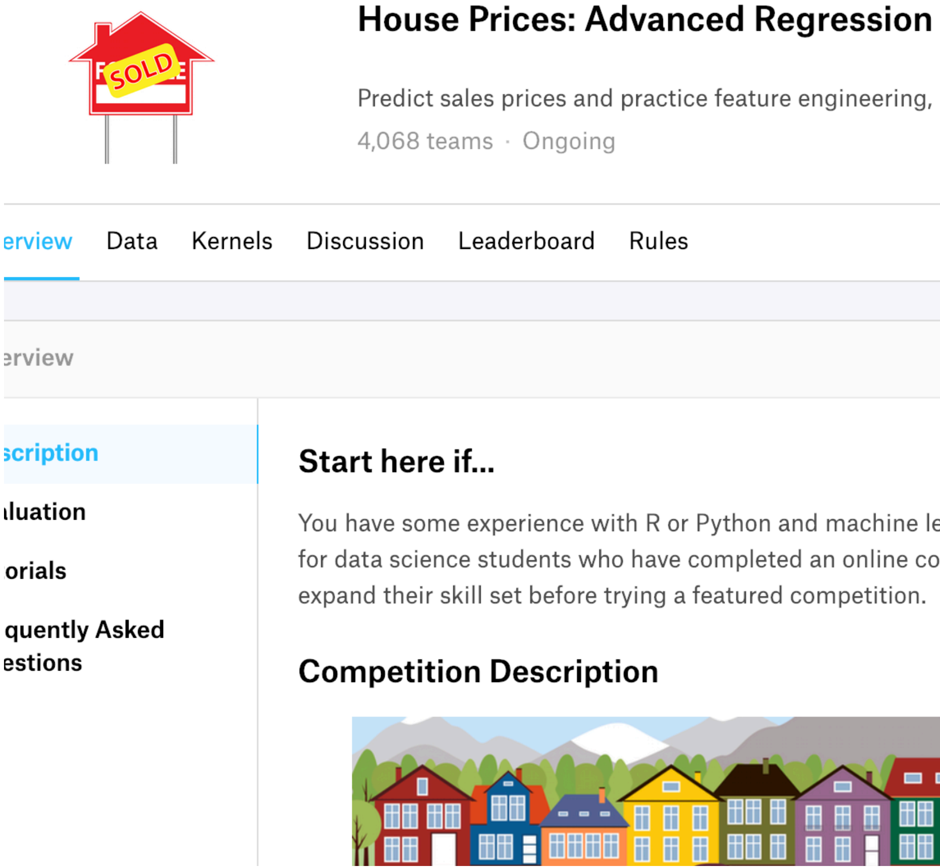
Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Homework 4

- Kaggle competition
- Works with your project teammates
- Start earlier
- Award $500 AWS credits for the top-3 team/people



Predict Who Will Repay Their Loans

- A lender hires you to investigate who will repay their loans
  - You are given complete files on 100 applicants
  - 5 defaulted within 3 years

Image credit debt.org

A Surprising Finding

- All 5 people who defaulted wore blue shirts during interviews
- Your model may find this strong signal as well

Image credit: rumble.com

Model Evaluation

Training Error and Generalization Error

- Training error: model error on the training data
- Generalization error: model error on new data
- Example: practice a future exam with past exams
  - Doing well on past exams (training error) doesn’t guarantee a good score on the future exam (generalization error)
  - Student A gets 0 error on past exams by rote learning
  - Student B understands the reasons for given answers

Validation Dataset and Test Dataset

- Validation dataset: a dataset used to evaluate the model
  - E.g. Take out 50% of the training data
  - Should not be mixed with the training data (#1 mistake)
- Test dataset: a dataset can be used once, e.g.
  - A future exam
  - The house sale price I bided
  - Dataset used in private leaderboard in Kaggle

K-fold Cross Validation

- Useful when not sufficient data
- Algorithm:
  - Partition the training data into K parts
  - For $i = 1, \ldots, K$
    - Use the i-th part as the validation set, the rest for training
  - Report the averaged the K validation errors
- Popular choices: $K = 5$ or 10

Underfitting
Overfitting

Image credit: hackernoon.com

Underfitting and Overfitting

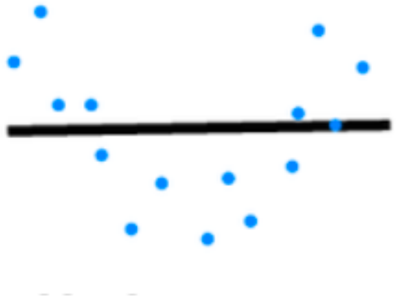
Data complexity

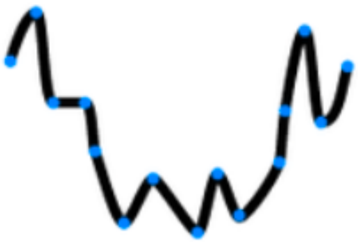
Model capacity

	Simple	Complex
Low	Normal	Underfitting
High	Overfitting	Normal

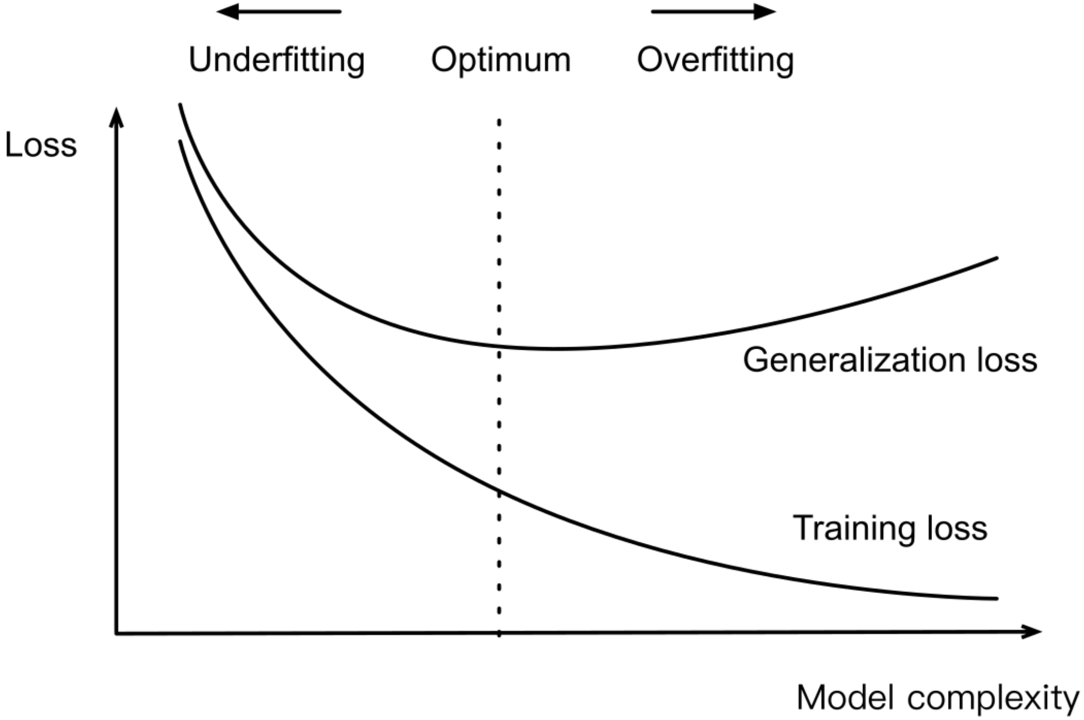
Model Capacity

- The ability to fit variety of functions
- Low capacity models struggles to fit training set
    - Underfitting
- High capacity models can memorize the training set
    - Overfitting





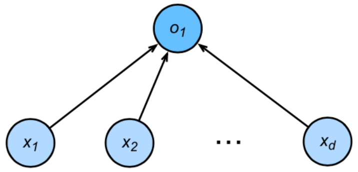
Influence of Model Complexity



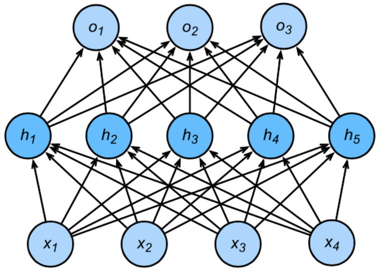
Estimate Model Capacity

- It's hard to compare complexity between different algorithms
    - e.g. tree vs neural network
- Given an algorithm family, two main factors matter:
    - The number of parameters
    - The values taken by each parameter

$$d + 1$$



$$(d + 1)m + (m + 1)k$$



VC Dimension

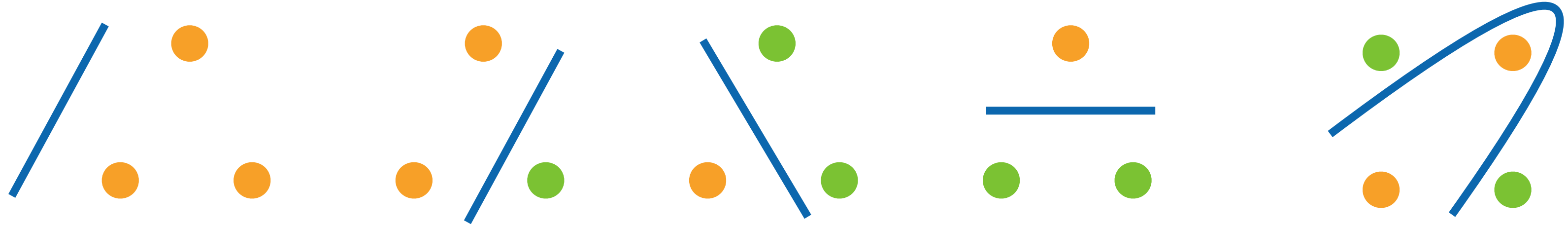
- A center topic in Statistic Learning Theory
- For a classification model, it's the size of the largest dataset, no matter how we assign labels, there exist a model to classify them perfectly

Vladimir **V**apnik

Alexey **C**hervonenkis

VC-Dimension for Linear Classifier

- 2-D perceptron: VCdim = 3
  - Can classify any 3 points, but not 4 points (xor)



- Perceptron with *N* parameters: VCdim = N
- Some Multilayer Perceptrons: VCdim = $O(N \log_2(N))$

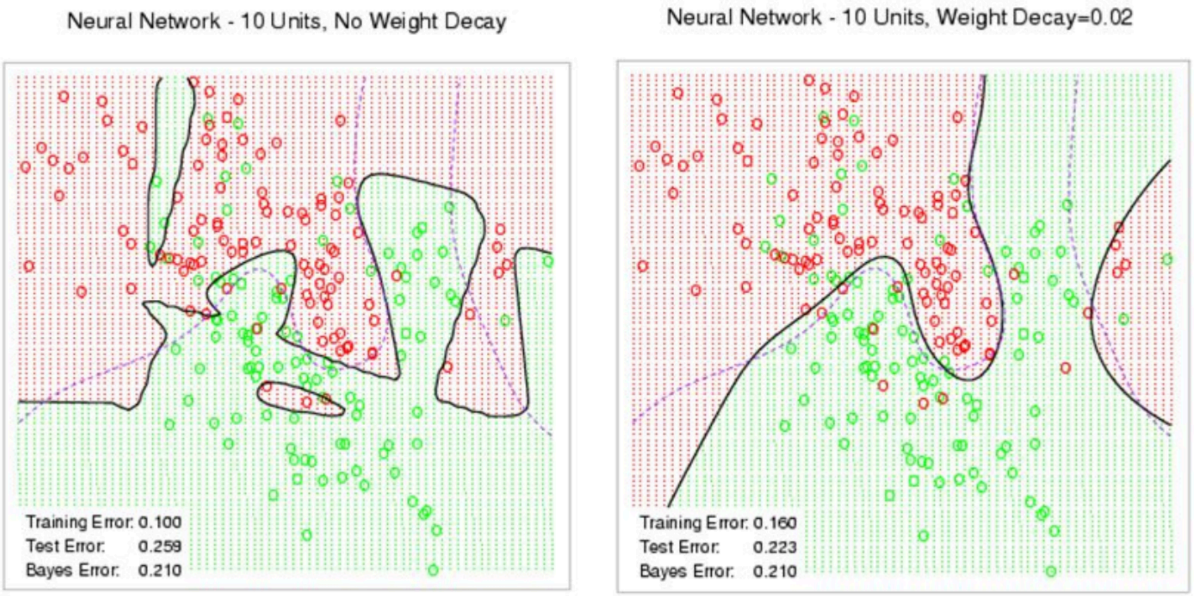
Usefulness of VC-Dimension

- Provides theory insights why a model works
    - Bound the gap between training error and generalization error
- Rarely used in practice with deep learning
    - The bounds are too loose
    - Difficulty to compute VC-dimension for deep neural networks
- Same for other statistic learning theory tools

Data Complexity

- Multiple factors matters
    - # of examples
    - # of elements in each example
    - time/space structure
    - diversity

Weight Decay

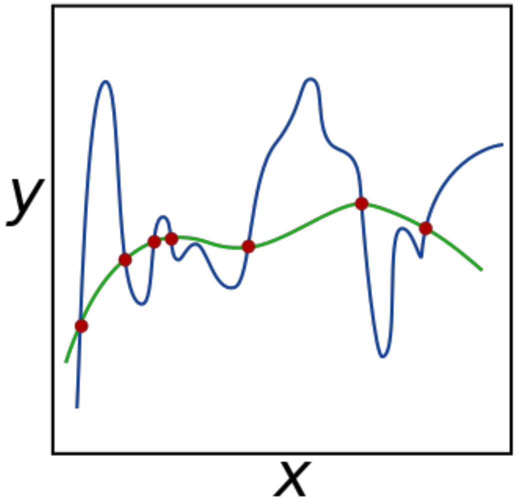


Squared Norm Regularization as Hard Constraint

- Reduce model complexity by limiting value range

$$\min \ \ell(\mathbf{w}, b) \quad \text{subject to} \quad \|\mathbf{w}\|^2 \leq \theta$$

  - Often do not regularize bias *b*
    - Doing or not doing has little difference in practice
  - A small θ means more regularization



Squared Norm Regularization as Soft Constraint

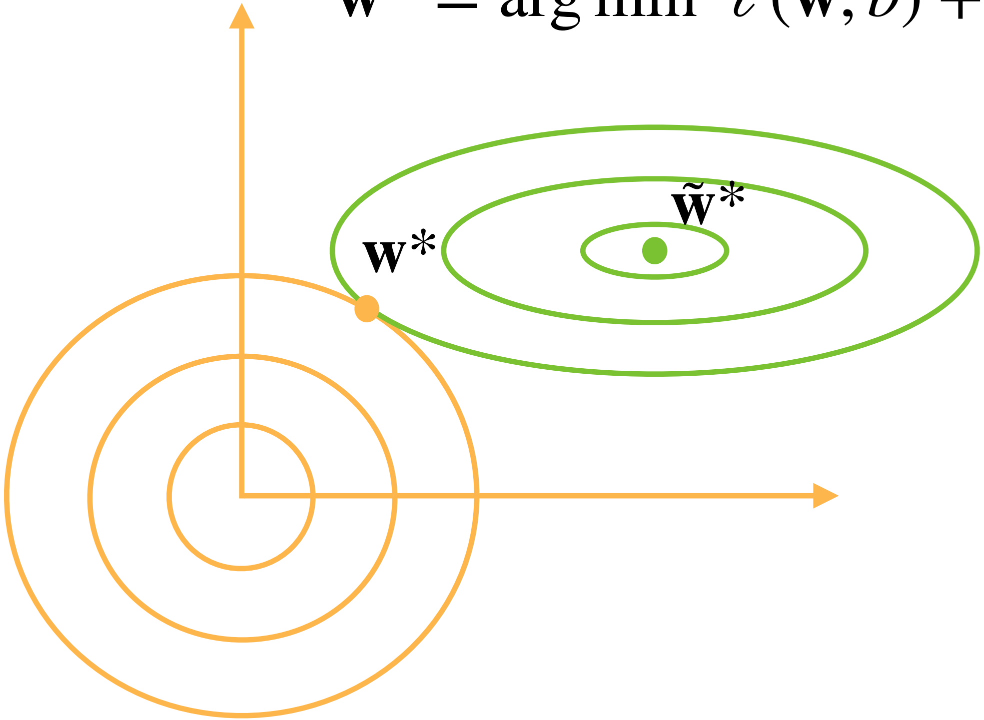
- For each θ, we can find λ to rewrite the hard constraint version as

$$\min \ \ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

  - Can prove by Lagrangian multiplier method
  - Hyper-parameter λ controls regularization importance
  - $\lambda = 0$: no effect
  - $\lambda \to \infty, \mathbf{w}^* \to \mathbf{0}$

Illustrate the Effect on Optimal Solutions

$$\mathbf{w}^* = \arg\min \ \ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$



$$\tilde{\mathbf{w}}^* = \arg\min \ \ell(\mathbf{w}, b)$$

Update Rule

- Compute the gradient

$$\frac{\partial}{\partial \mathbf{w}}\left(\ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2\right) = \frac{\partial \ell(\mathbf{w}, b)}{\partial \mathbf{w}} + \lambda \mathbf{w}$$

- Update weight at time *t*

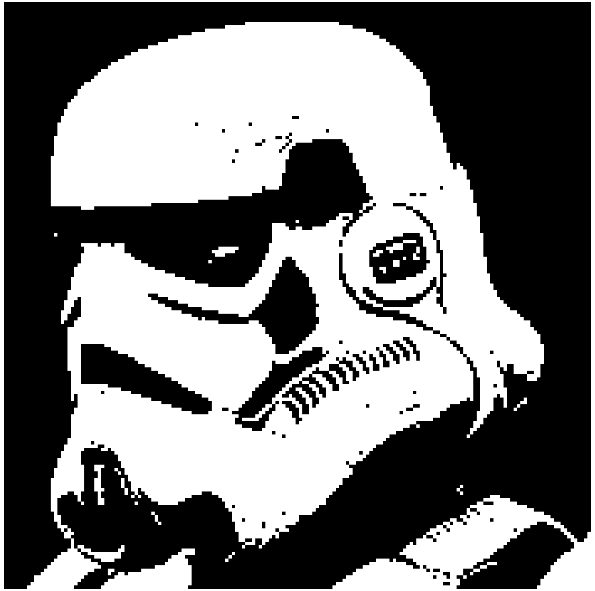
$$\mathbf{w}_{t+1} = (1 - \eta\lambda)\mathbf{w}_t - \eta \frac{\partial \ell(\mathbf{w}_t, b_t)}{\partial \mathbf{w}_t}$$

  - Often $\eta\lambda < 1$, so also called weight decay in deep learning

Dropout

Motivation

- A good model should be robust under modest changes in the input
  - Training with input noise equals to Tikhonov Regularization
  - Dropout: inject noises into internal layers



Add Noise without Bias

- Add noise into ***x*** to get ***x'***, we hope

$$\mathbf{E}[\mathbf{x}'] = \mathbf{x}$$

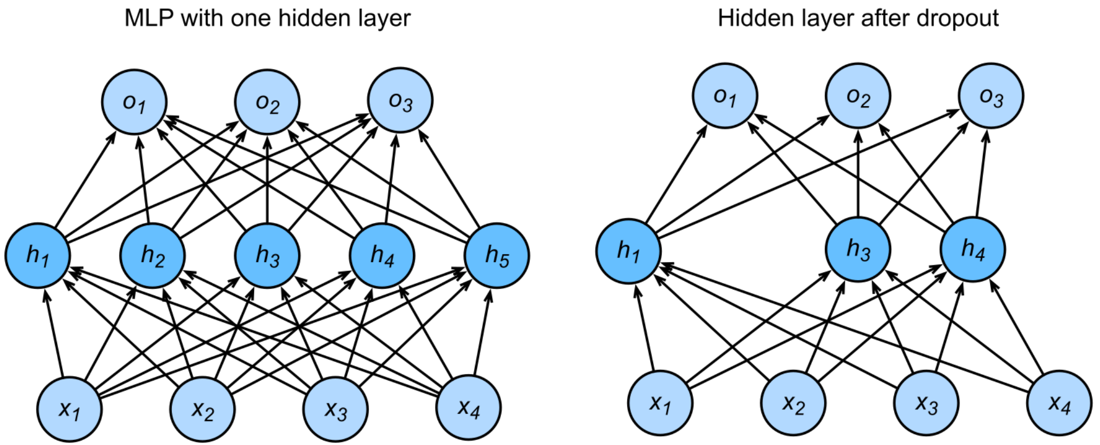
- Dropout perturbs each element by

$$x_i' = \begin{cases} 0 & \text{with probablity } p \\ \frac{x_i}{1-p} & \text{otherise} \end{cases}$$

Apply Dropout

- Often apply dropout on the output of hidden fully-connected layers

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}' = \text{dropout}(\mathbf{h})$$
$$\mathbf{o} = \mathbf{W}_2\mathbf{h}' + \mathbf{b}_2$$
$$\mathbf{y} = \text{softmax}(o)$$



Dropout in Inference

- Regularization is only used in training
- The dropout layer for inference is

$$\mathbf{h}' = \text{dropout}(\mathbf{h})$$

  - Guarantee deterministic results