شبکه‌های عصبی مصنوعی

درس ۲۲

موضوعات مطرح در آموزش عملی

Practical Training Issues

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/nn

Practical Training Issues

موضوعات مطرح در آموزش عملی

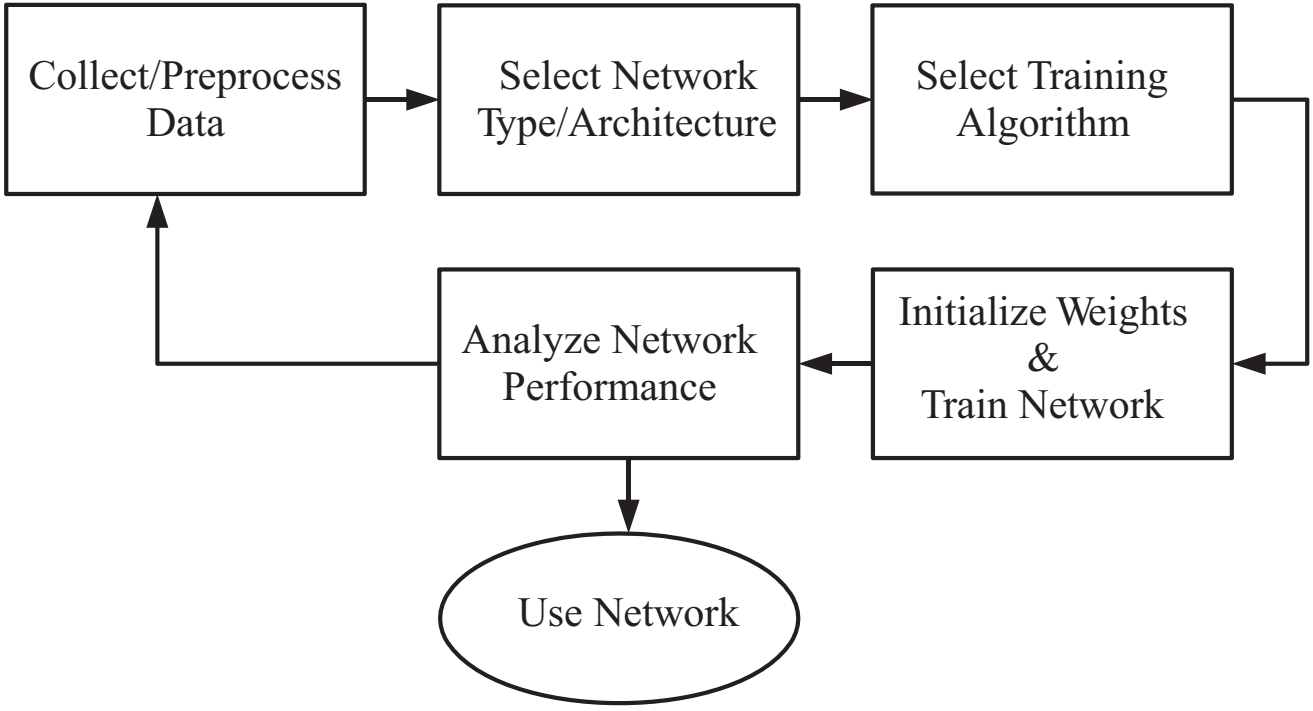
PRACTICAL TRAINING ISSUES

در این فصل، به برخی از نکات برای آموزش عملی
که به انواع مختلفی از شبکه‌های عصبی قابل اعمال است، پرداخته می‌شود.

موضوعات مطرح در آموزش عملی شبکه‌های عصبی

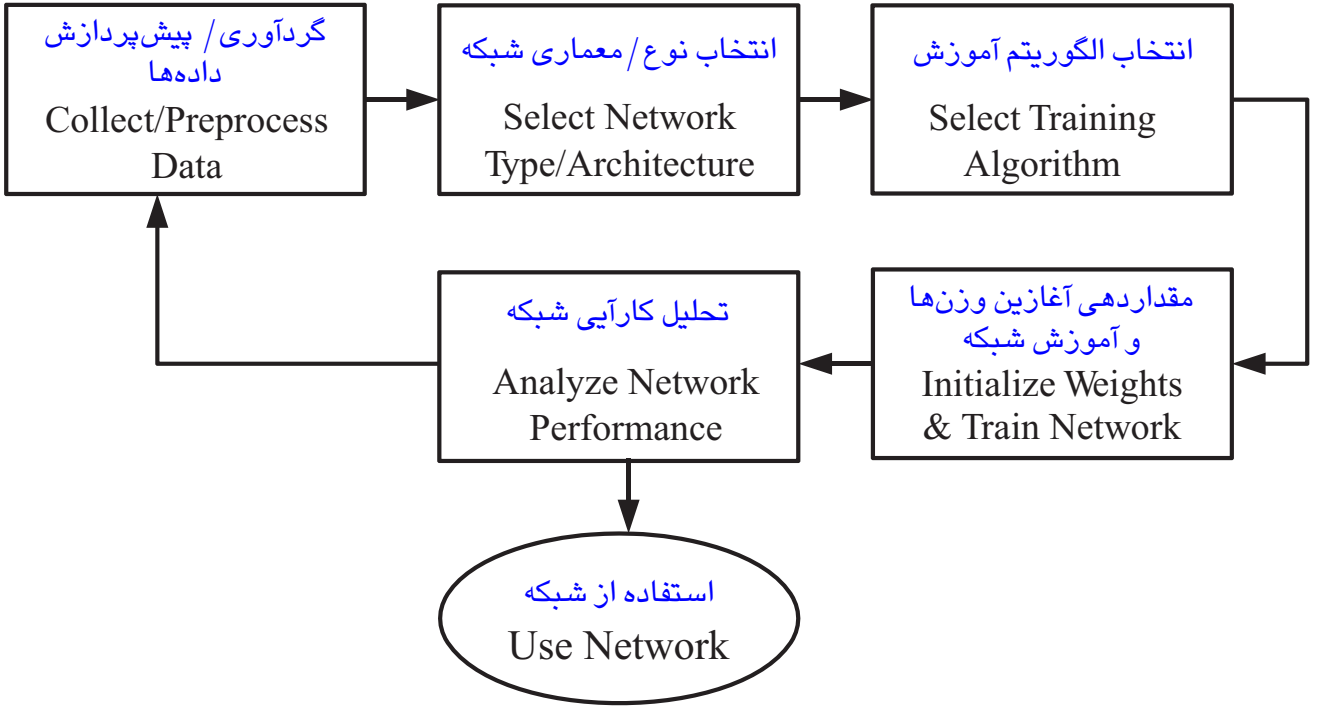
- کارهای لازم پیش از آموزش یک شبکه (پیشا –آموزش):
  (گردآوری داده‌ها، پیش‌پردازش آنها، انتخاب معماری شبکه)
- کارهای لازم هنگام آموزش شبکه (آموزش)
- کارهای لازم برای تحلیل‌های پس از آموزش (پسا –آموزش)

Network Training Steps



گام‌های آموزش شبکه

NETWORK TRAINING STEPS



گام‌های آموزش شبکه

NETWORK TRAINING STEPS

گام‌های آموزش شبکه

۱
گام‌های پیشا–آموزش
Pre-Training Steps

- انتخاب داده‌ها
- پیش‌پردازش داده‌ها
- انتخاب معماری شبکه

۲
گام‌های هنگام آموزش
Training Steps

- مقداردهی آغازین وزن‌ها
- انتخاب الگوریتم آموزش
- تعیین ضابطه‌ی توقف
- انتخاب تابع کارآیی
- اجرای چند سلسله‌ی یادگیری

۳
گام‌های پسا–آموزش
Post-Training Steps

- تحلیل کارآیی
- تحلیل بیش‌برازش / برون‌یابی
- تحلیل حساسیت

آیا شبکه‌ی عصبی برای مسئله‌ی شما راه‌حل مناسبی است؟

Before beginning the neural network training process, you should first determine if a neural network is needed to solve your problem, or if some simpler linear technique might be adequate.

For example, there is no need to use a neural network for a **fitting problem**, if *standard linear regression* will produce a satisfactory result.
The neural network techniques provide additional power, but at the expense of more challenging training requirements.

When linear methods will work, they are the first choice.

شبکه‌های عصبی مصنوعی

موضوعات مطرح در آموزش عملی

۱

گام‌های پیش از آموزش

گام‌های آموزش شبکه

گام‌های پیشا–آموزش

PRE-TRAINING STEPS

گام‌های آموزش شبکه

۱

گام‌های پیشا–آموزش
Pre-Training Steps

- انتخاب داده‌ها
- پیش‌پردازش داده‌ها
- انتخاب معماری شبکه

۲

گام‌های هنگام آموزش
Training Steps

- مقداردهی آغازین وزن‌ها
- انتخاب الگوریتم آموزش
- تعیین ضابطه‌ی توقف
- انتخاب تابع کارآیی
- اجرای چند سلسله‌ی یادگیری

۳

گام‌های پسا–آموزش
Post-Training Steps

- تحلیل کارآیی
- تحلیل بیش‌برازش / برون‌یابی
- تحلیل حساسیت

Selection of Data

- Data must adequately cover the relevant regions of the input space (to avoid extrapolation).
- Divide the data into training, validation and testing subsets (70%, 15%, 15%).
- Each of the subsets must cover the same parts of the input space.
- The amount of data required depends on the complexity of the function being approximated (or the complexity of the decision boundary).
- Post-training analysis may be needed to determine the adequacy of the data.

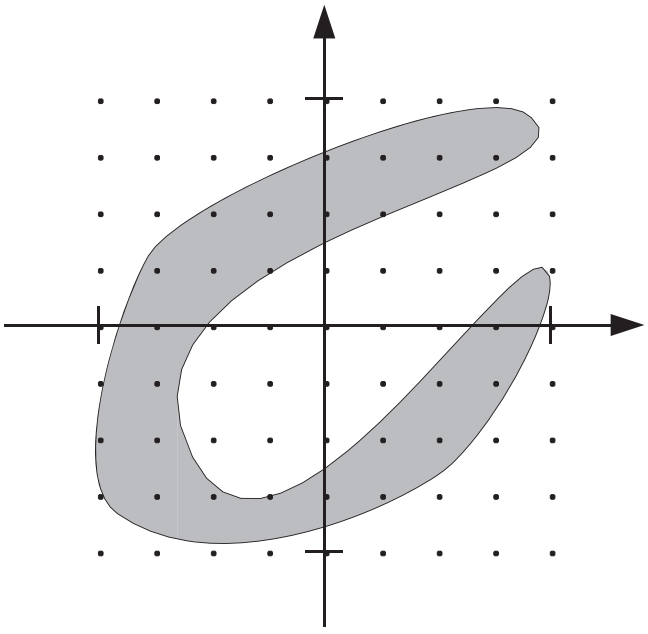
انتخاب داده‌ها

SELECTION OF DATA

- Data must adequately cover the relevant regions of the input space (to avoid extrapolation).
- Divide the data into training, validation and testing subsets (70%, 15%, 15%).
- Each of the subsets must cover the same parts of the input space.
- The amount of data required depends on the complexity of the function being approximated (or the complexity of the decision boundary).
- Post-training analysis may be needed to determine the adequacy of the data.

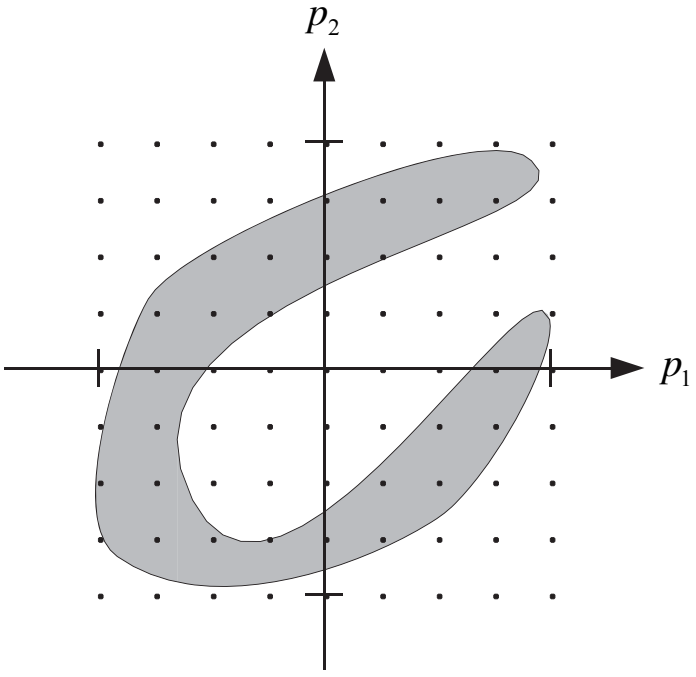
- داده‌ها باید به‌طور کافی نواحی مربوط در فضای ورودی را پوشش دهند (برای اجتناب از برون‌یابی).
- داده‌ها را به سه زیرمجموعه‌ی آموزشی، اعتبارسنجی و آزمایشی تقسیم کنید (۷۰%، ۱۵%، ۱۵%).
- هر یک از این زیرمجموعه‌ها باید بخش‌های مشابهی از فضای ورودی را پوشش دهند.
- میزان داده‌های لازم، وابسته به پیچیدگی تابع مورد تقریب (یا پیچیدگی مرز تصمیم) است.
- ممکن است تحلیل‌های پسا–آموزش برای تعیین کافی بودن داده‌ها لازم باشد.

Determination of Input Range



تعیین بازه‌ی ورودی

DETERMINATION OF INPUT RANGE



محدوده‌ی ورودی با در نظر گرفتن متغیرهای ورودی مستقل

Data Preprocessing

•Normalize inputs/targets to the range [-1,1].

$$\mathbf{p}^n = 2(\mathbf{p} - \mathbf{p}^{min})./(\mathbf{p}^{max} - \mathbf{p}^{min}) - 1$$

•Normalize inputs/targets to zero mean and unity variance.

$$\mathbf{p}^n = (\mathbf{p} - \mathbf{p}^{mean})./\mathbf{p}^{std}$$

•Nonlinear transformations.

$$\mathbf{p}^t = 1/\mathbf{p} \qquad \mathbf{p}^t = \exp(\mathbf{p})$$

•Feature extraction (dimensionality reduction).

  •Principal components

پیش‌پردازش داده‌ها

نرمال‌سازی داده‌ها

DATA PREPROCESSING

روش استاندارد اول: نرمال‌سازی به مقادیر بازه‌ی [-1, 1]

Normalize inputs/targets to the range [-1,1].

$$\mathbf{p}^n = 2(\mathbf{p} - \mathbf{p}^{min})./(\mathbf{p}^{max} - \mathbf{p}^{min}) - 1$$

روش استاندارد دوم: نرمال‌سازی به میانگین صفر و واریانس یک

Normalize inputs/targets to zero mean and unity variance.

$$\mathbf{p}^n = (\mathbf{p} - \mathbf{p}^{mean}).\cdot/\mathbf{p}^{std}$$

گام نرمال‌سازی به هر دوی بردارهای ورودی و تارگت در مجموعه‌ی داده‌ها اعمال می‌شود.

Nonlinear transformations.

تبدیل‌های غیرخطی

$$\mathbf{p}^t = 1/\mathbf{p} \qquad \mathbf{p}^t = \exp(\mathbf{p})$$

تبدیل‌های غیرخطی، برای هر مجموعه داده‌ای قابل اعمال نیست.

پیش‌پردازش داده‌ها

استخراج ویژگی

DATA PREPROCESSING

استخراج ویژگی (کاهش بعد)

Feature extraction (dimensionality reduction).

- Principal components

مانند تکنیک تحلیل مؤلفه‌های اصلی (به‌عنوان یک روش عمومی برای کاهش بعد و استخراج ویژگی)

هدف استخراج ویژگی:

کاهش بعد فضای ورودی با محاسبه‌ی مجموعه‌ی کوچکی از ویژگی‌ها از روی هر بردار ورودی و استفاده از ویژگی‌ها به‌عنوان ورودی شبکه‌ی عصبی

22 Coding Targets (Pattern Classification)

- There are three common ways to code targets. Assume that we have N classes.
- 1) You can have a scalar target that takes on N possible values (e.g., 1, 2, …, N)
- 2) You can code the target in binary code. This requires P output neurons, where 2^P is greater than or equal to N.
- 3) You can have N neurons in the output layer. The targets will be vectors whose elements are all equal to zero, except for the neuron that corresponds to the correct class.
- Method 3) generally produces the best results.

کدگذاری تارگت‌ها

(در کاربرد طبقه‌بندی الگو)

CODING TARGETS (PATTERN CLASSIFICATION)

- There are three common ways to code targets. Assume that we have N classes.
  1) You can have a scalar target that takes on N possible values (e.g., 1, 2, …, N)
  2) You can code the target in binary code. This requires P output neurons, where 2^P is greater than or equal to N.
  3) You can have N neurons in the output layer. The targets will be vectors whose elements are all equal to zero, except for the neuron that corresponds to the correct class.
- Method 3) generally produces the best results.

- سه راه متداول برای کدگذاری تارگت‌ها وجود دارد. با فرض اینکه N طبقه داشته باشیم:
  ۱. می‌توانیم یک تارگت اسکالر داشته باشیم که مقادیرش را از N مقدار ممکن می‌گیرد (مثلاً، $1, 2, \ldots, N$).
  ۲. می‌توانیم تارگت را به‌صورت دودویی کدگذاری کنیم. این نیازمند P نرون خروجی است، که در آن $2^P \geq N$.
  ۳. می‌توانیم N نرون در لایه‌ی خروجی داشته باشیم. این تارگت‌ها بردارهایی خواهند بود که عناصر آنها همگی صفر است، بجز برای نرون متناظر با طبقه‌ی درست.
- روش ۳، عموماً بهترین نتایج را ایجاد می‌کند.

کدگذاری تارگت‌ها زمانی انجام می‌شود که ورودی/تارگت‌های شبکه مقدار گسسته دارد.
* کدگذاری ورودی‌ها هم مشابه کدگذاری تارگت‌ها قابل انجام است.

Importance of Transfer Function

- When coding the targets, we need to consider the output layer transfer function.
- For pattern recognition problems, we would typically use log-sigmoid or tangent-sigmoid.
- If we use the tangent-sigmoid in the last layer, which is more common, then we might consider assigning target values to -1 or 1.
- This tends to cause training difficulties (saturation of the sigmoid function).
- It is better to assign target values at the point where the second derivative of the sigmoid function is maximum. This occurs when the net input is -1 and 1, which corresponds to output values of -0.76 and +0.76.

اهمیت تابع انتقال

IMPORTANCE OF TRANSFER FUNCTION

- When coding the targets, we need to consider the output layer transfer function.
- وقتی تارگت‌ها را کدگذاری می‌کنیم، لازم است تابع انتقال لایه‌ی خروجی را در نظر داشته باشیم.
- For pattern recognition problems, we would typically use log-sigmoid or tangent-sigmoid.
- برای مسئله‌های بازشناسی الگو، معمولاً از سیگموئید–لگاریتمی یا سیگموئید–تانژانتی هایپربولیک استفاده می‌کنیم.
- If we use the tangent-sigmoid in the last layer, which is more common, then we might consider assigning target values to −1 or +1.
- اگر از سیگموئید تانژانتی در لایه‌ی آخر استفاده کنیم، که معمولاً متداول‌تر است، باید انتساب مقادیر تارگت به 1− یا 1+ را در نظر داشته باشیم.
- This tends to cause training difficulties (saturation of the sigmoid function).
- این موجب بروز دشواری‌هایی در حین آموزش می‌شود (اشباع تابع سیگموئید)
- It is better to assign target values at the point where the second derivative of the sigmoid function is maximum. This occurs when the net input is -1 and 1, which corresponds to output values of −0.76 and +0.76.
- بهتر است مقادیر تارگت‌ها را به نقطه‌ای منتسب کنیم که مقدار مشتق دوم تابع سیگموئید در آن ماکزیمم می‌شود. این وقتی اتفاق می‌افتد که ورودی خالص 1− یا 1+ باشد که به ترتیب متناظر با مقادیر 0.76− و 0.76+ است.

Softmax Transfer Function

- If the network outputs should correspond to probabilities of belonging to a certain class, the softmax transfer function can be used.

$$a_i = f(n_i) = \frac{\exp(n_i)}{\sum_{j=1}^{S} \exp(n_j)}$$

اهمیت تابع انتقال

تابع انتقال «سافت‌مکس»

SOFTMAX TRANSFER FUNCTION

If the network outputs should correspond to probabilities of belonging to a certain class, the softmax transfer function can be used.

اگر خروجی شبکه باید متناظر با احتمالات تعلق داشتن به یک طبقه باشد،
تابع انتقال *softmax* می‌تواند مورد استفاده قرار گیرد.

$$a_i = f(n_i) = \frac{\exp(n_i)}{\sum_{j=1}^{S} \exp(n_j)}$$

Missing Data

- Replace the missing values in the input vector with the average value for that element of the input. Add an additional variable to the input vector as a flag to indicate missing data.
- For missing elements of the target vectors, do not include them in the calculation of squared error.

داده‌های گم‌شده

MISSING DATA

- Replace the missing values in the input vector with the average value for that element of the input.
  Add an additional variable to the input vector as a flag to indicate missing data.

  مقادیر گم‌شده در بردار ورودی را با مقدار متوسط برای آن عنصر ورودی جایگزین می‌کنیم.
  یک متغیر اضافی به بردار ورودی اضافه می‌کنیم تا به‌عنوان پرچمی که نشان‌دهنده‌ی داده‌ی گم‌شده است استفاده شود.

- For missing elements of the target vectors, do not include them in the calculation of squared error.

  برای عناصر گم‌شده‌ی بردارهای تارگت، آنها را در محاسبه‌ی مربع خطا وارد نمی‌کنیم.

Problem Types

- Fitting (nonlinear regression). Map between a set of inputs and a corresponding set of targets. (e.g., estimate home prices from tax rate, pupil/teacher ratio, etc.; estimate emission levels from fuel consumption and speed; predict body fat level from body measurements.)
- Pattern recognition (classification). Classify inputs into a set of target categories. (e.g., recognize the vineyard from a chemical analysis of the wine; classify a tumor as benign or malignant, from uniformity of cell size, clump thickness and mitosis.)
- Clustering (segmentation) Group data by similarity. (e.g., group customers according to buying patterns, group genes with related expression patterns.)
- Prediction (time series analysis, system identification, filtering or dynamic modeling). Predict the future value of some time series. (e.g., predict the future value of some stock; predict the future value of the concentration of some chemical; predict outages on the electric grid.)

انتخاب معماری شبکه

انواع مسئله

PROBLEM TYPES

برازش (رگرسیون غیرخطی)
Fitting (Nonlinear Regression)

نگاشت میان یک مجموعه از ورودی‌ها و مجموعه‌ی متناظر از تارگت‌ها

(مثال: تخمین قیمت خانه از روی نرخ مالیات، نسبت شاگرد/معلم در مدارس منطقه و ...، تخمین سطح انتشار آلاینده از روی مصرف سوخت و سرعت، پیش‌بینی سطح چربی بدن از روی اندازه‌گیری‌های بدنی)

Fitting (nonlinear regression, function approximation).
Map between a set of inputs
and a corresponding set of targets.
(e.g., estimate home prices from tax rate, pupil/teacher ratio, etc.; estimate emission levels from fuel consumption and speed; predict body fat level from body measurements.)

بازشناسی الگو (طبقه‌بندی)
Pattern recognition (Classification)

طبقه‌بندی ورودی‌ها در مجموعه‌ای از دسته‌ها/طبقه‌های تارگت

(مثال: طبقه بندی یک تومور به خوش‌خیم و بدخیم از روی نایکنواختی اندازه‌ی سلول، ضخامت توده و تقسیم میتوز)

Pattern recognition (classification).
Classify inputs into a set of target categories.
(e.g., classify a tumor as benign or malignant, from uniformity of cell size, clump thickness and mitosis.)

خوشه‌بندی (بخش‌بندی)
Clustering (Segmentation)

گروه‌بندی داده‌ها بر اساس شباهت

(مثال: گروه‌بندی مشتریان براساس الگوهای خرید، داده‌کاوی، گروه‌بندی ژن‌ها بر اساس الگوهای عبارت ژنی مرتبط)

Clustering (segmentation)
Group data by similarity.
(e.g., group customers according to buying patterns, data mining, group genes with related expression patterns.)

پیش‌بینی
Prediction

پیش‌بینی مقدار آینده‌ی یک سری زمانی

(مثال: پیش‌بینی مقدار آینده‌ی یک سهام، پیش‌بینی مقدار آینده‌ی غلظت یک ماده‌ی شیمیایی، پیش‌بینی قطع‌شدگی‌ها از شبکه‌ی برق)

Prediction (time series analysis, system identification, filtering or dynamic modeling).
Predict the future value of some time series.
(e.g., predict the future value of some stock; predict the future value of the concentration of some chemical; predict outages on the electric grid.)

Choice of Network Architecture

- Fitting
  - Multilayer networks with sigmoid hidden layers and linear output layers.
  - Radial basis networks
- Pattern Recognition
  - Multilayer networks with sigmoid hidden layers and sigmoid output layers.
  - Radial basis networks.
- Clustering
  - Self-organizing feature map
- Prediction
  - Focused time-delay neural network
  - NARX network

انتخاب معماری شبکه

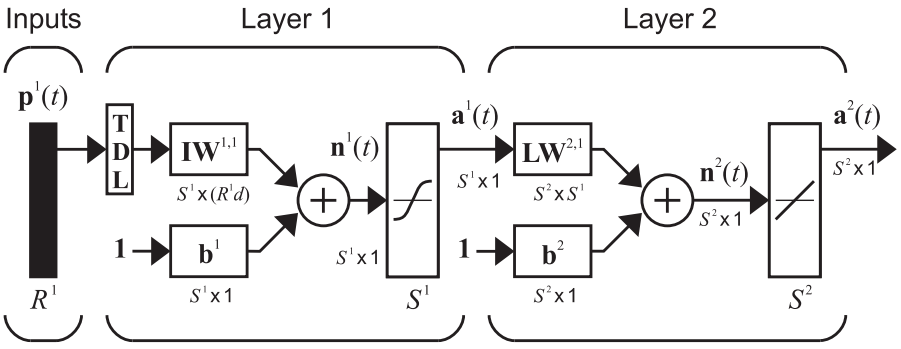
معماری پایه‌ی شبکه

CHOICE OF NETWORK ARCHITECTURE

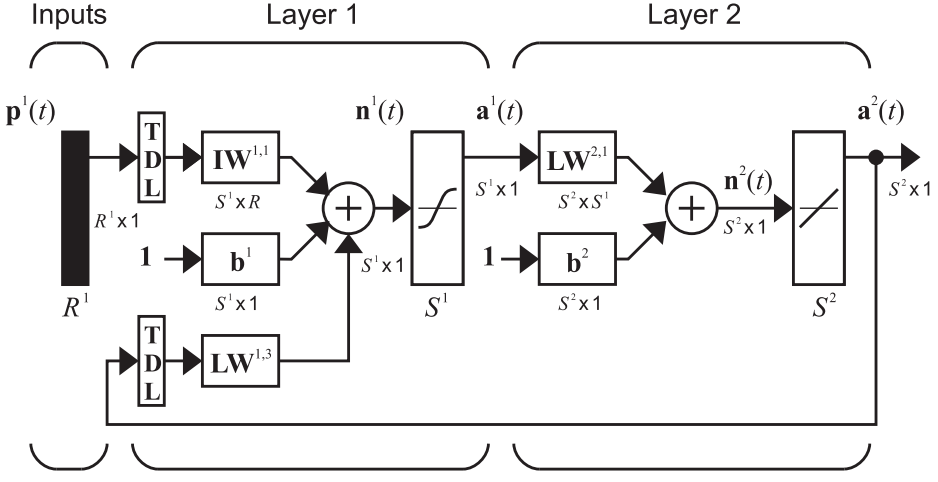
برازش (رگرسیون غیرخطی) *Fitting (Nonlinear Regression)*	○ Multilayer networks with sigmoid hidden layers and linear output layers. ○ Radial basis networks
بازشناسی الگو (طبقه‌بندی) *Pattern recognition (Classification)*	○ Multilayer networks with sigmoid hidden layers and sigmoid output layers. ○ Radial basis networks
خوشه‌بندی (بخش‌بندی) *Clustering (Segmentation)*	○ Self-organizing feature map
پیش‌بینی *Prediction*	○ Focused time-delay neural network ○ NARX network (Nonlinear AutoRegressive model with eXogenous input)

Prediction Networks

Focused Time Delay



NARX

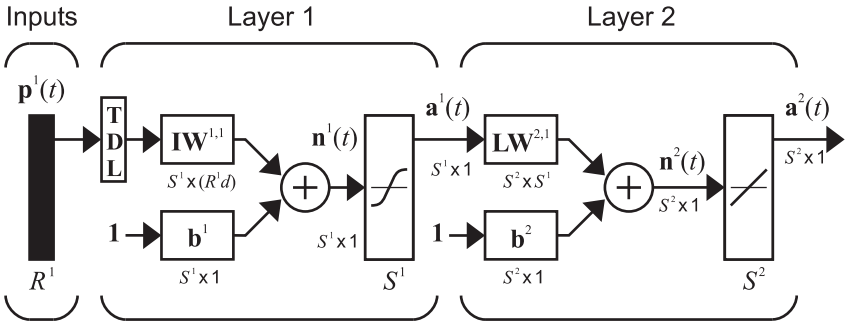


انتخاب معماری شبکه

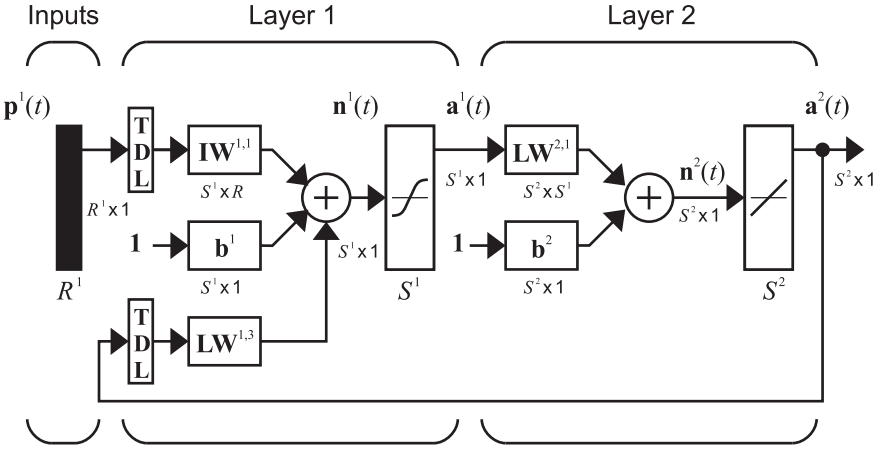
معماری پایه‌ی شبکه : شبکه‌های پیش‌بینی

PREDICTION NETWORKS

Focused Time Delay



NARX



Architecture Specifics

- Number of layers/neurons
  - For multilayer network, start with two layers. Increase number of layers if result is not satisfactory.
  - Use a reasonably large number of neurons in the hidden layer (20). Use early stopping or Bayesian regularization to prevent overfitting.
  - Number of neurons in output layer = number of targets. You can use multiple networks instead of multiple outputs.
- Input selection
  - Sensitivity analysis (see later slide)
  - Bayesian regularization with separate α for each column of the input weight matrix.

انتخاب معماری شبکه

انتخاب مشخصه‌های معماری

SELECTION OF ARCHITECTURE SPECIFICS

Number of layers/neurons

- For multilayer network, start with two layers. Increase number of layers if result is not satisfactory.
- Use a reasonably large number of neurons in the hidden layer (20). Use early stopping or Bayesian regularization to prevent overfitting.
- Number of neurons in output layer = number of targets. You can use multiple networks instead of multiple outputs.

Input selection

- Sensitivity analysis (see later slide)
- Bayesian regularization with separate α for each column of the input weight matrix.

تعداد لایه‌ها/نرون‌ها

- برای شبکه‌های چندلایه، با دو لایه شروع می‌کنیم. اگر نتایج راضی‌کننده نبودند، تعداد لایه‌ها را افزایش می‌دهیم.
- از تعداد منطقاً بزرگ نرون‌ها در لایه‌ی پنهان استفاده می‌کنیم (۲۰). از توقف زودهنگام یا رگولاریزاسیون بیزی برای اجتناب از بیش‌برازش استفاده می‌کنیم.
- تعداد نرون‌ها در لایه‌ی خروجی = تعداد تارگت‌ها. می‌توانیم از چند شبکه به‌جای خروجی‌های چندگانه استفاده کنیم.

انتخاب ورودی

- تحلیل حساسیت (در ادامه)
- رگولاریزاسیون بیزی با مقادیر α جداگانه برای هر ستون از ماتریس وزن ورودی

شبکه‌های عصبی مصنوعی

موضوعات مطرح در آموزش عملی

۲

آموزش شبکه

گام‌های آموزش شبکه

گام‌های هنگام آموزش

TRAINING STEPS

گام‌های آموزش شبکه

۱

گام‌های پیشا–آموزش
Pre-Training Steps

- انتخاب داده‌ها
- پیش‌پردازش داده‌ها
- انتخاب معماری شبکه

۲

گام‌های هنگام آموزش
Training Steps

- مقداردهی آغازین وزن‌ها
- انتخاب الگوریتم آموزش
- تعیین ضابطه‌ی توقف
- انتخاب تابع کارآیی
- اجرای چند سلسله‌ی یادگیری

۳

گام‌های پسا–آموزش
Post-Training Steps

- تحلیل کارآیی
- تحلیل بیش‌برازش/ برون‌یابی
- تحلیل حساسیت

Weight Initialization

For Multilayer Networks

- Random weights. Uniformly distributed between -0.5 and 0.5, if the inputs are normalized to fall between -1 and 1.
- Random direction for weights, with magnitude set to

$$\left\| {}_i\mathbf{w} \right\| = 0.7\left(S^1\right)^{1/R}$$

and biases randomly distributed between

$$-\left\| {}_i\mathbf{w} \right\| \quad \text{and} \quad \left\| {}_i\mathbf{w} \right\|.$$

مقداردهی آغازین وزن‌ها

برای شبکه‌های چندلایه

WEIGHT INITIALIZATION

For Multilayer Networks

برای شبکه‌های چندلایه

- Random weights. Uniformly distributed between -0.5 and 0.5, if the inputs are normalized to fall between -1 and 1.

وزن‌های تصادفی: توزیع یکنواخت بین 0.5- و 0.5+، اگر ورودی‌ها پس از نرمال‌سازی بین 1- و 1+ باشند.

- Random direction for weights, with magnitude set to

جهت تصادفی برای وزن‌ها، با تنظیم اندازه‌ی آنها به:

$$\|{}_i\mathbf{w}\| = 0.7\left(S^1\right)^{1/R}$$

S^1 = تعدادنرون لایه‌ی اول
R = بعد ورودی

هر سیگموئید $1/S^1$ بازه‌ی ورودی را پوشش می‌دهد

and biases randomly distributed between

و تنظیم بایاس‌ها به‌صورت تصادفی بین:

$$-\|{}_i\mathbf{w}\| \quad \text{and} \quad \|{}_i\mathbf{w}\|.$$

Weight Initialization

- For Competitive Networks
  - Small random numbers
  - Randomly selected input vectors
  - Principal components of the input vectors

مقداردهی آغازین وزن‌ها

برای شبکه‌های چندلایه

WEIGHT INITIALIZATION

For Competitive Networks

برای شبکه‌های رقابتی

- Small random numbers

  اعداد تصادفی کوچک

- Randomly selected input vectors

  بردارهای ورودی انتخاب‌شده به‌صورت تصادفی

- Principal components of the input vectors

  مؤلفه‌های اصلی بردارهای ورودی

Choice of Training Algorithm

- For medium sized networks (several hundred weights) used for fitting or prediction problems, use the Levenberg-Marquardt algorithm (trainlm).
- For large networks (thousands of weights) used for fitting or prediction problems, or networks used for pattern recognition problems, conjugate gradient algorithms, such as the scaled conjugate gradient algorithm (trainscg) are generally faster.
- Of the sequential algorithms, the extended Kalman filter algorithm are generally fastest.

انتخاب الگوریتم آموزش

CHOICE OF TRAINING ALGORITHM

- For medium sized networks (several hundred weights) used for fitting or prediction problems, use the Levenberg-Marquardt algorithm (trainlm).
  - برای شبکه‌های دارای اندازه‌ی متوسط (دارای چند صد وزن) که برای مسائل برازش یا پیش‌بینی استفاده می‌شوند، از الگوریتم لونبرگ–مارکوآرد (trainlm) استفاده کنید.
- For large networks (thousands of weights) used for fitting or prediction problems, or networks used for pattern recognition problems, conjugate gradient algorithms, such as the scaled conjugate gradient algorithm (trainscg) are generally faster.
  - برای شبکه‌های بزرگ (دارای هزاران وزن) که برای مسائل برازش یا پیش‌بینی استفاده می‌شوند، یا شبکه‌هایی که برای مسائل بازشناسی الگو استفاده می‌شوند، الگوریتم‌های گرادیان مزدوج، مانند الگوریتم گرادیان مزدوج مقیاس‌یافته (trainscg) عموماً سریع‌تر هستند.
- Of the sequential algorithms, the extended Kalman filter algorithm are generally fastest.
  - از میان الگوریتم‌های ترتیبی، الگوریتم فیلتر کالمن گسترش‌یافته، عموماً سریع‌ترین است.

Stopping Criteria

- Norm of the gradient (of the mean squared error) less than a pre-specified amount (for example, 10^{-6}).
- Early stopping because the validation error increases.
- Maximum number of iterations reached.
- Mean square error drops below a specified threshold (not generally a useful method).
- Mean square error curve (on a log-log scale) becomes flat for some time (user stop).

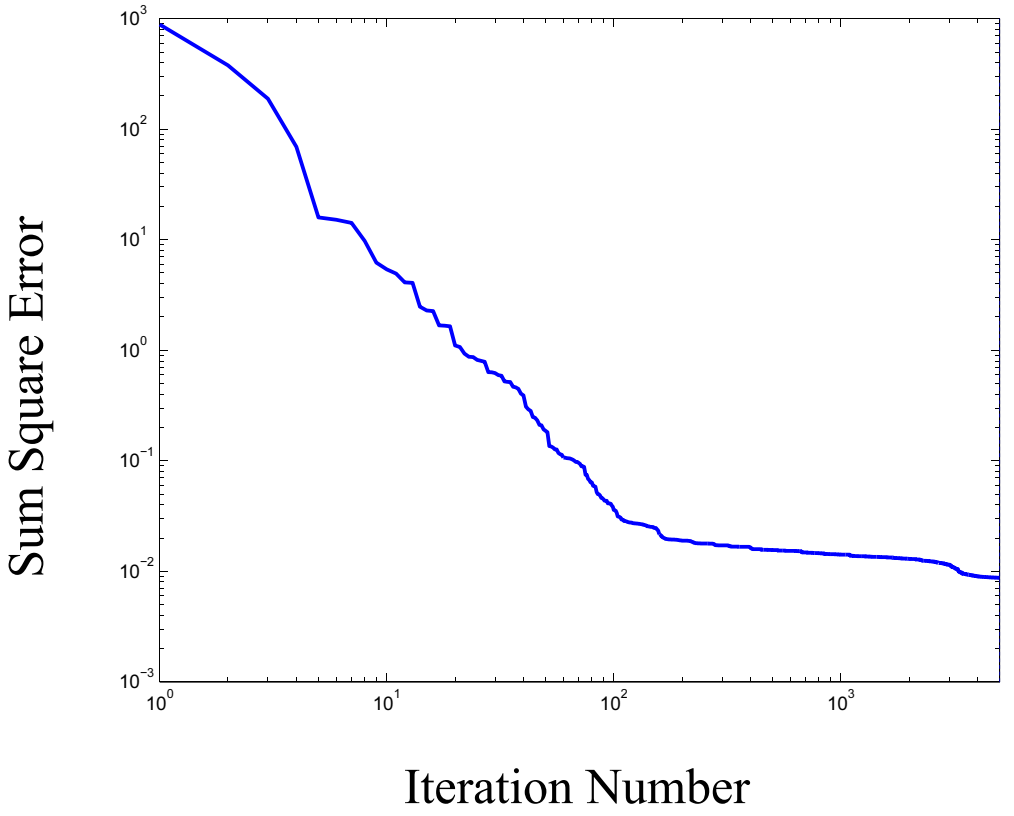
ضابطه‌ی توقف

برای شبکه‌های چندلایه

STOPPING CRITERIA

- Norm of the gradient (of the mean squared error) less than a pre-specified amount (for example, 10^{-6}).
  - نُرم گرادیان (برای خطای میانگین مربعات) کوچک‌تر از یک مقدار از پیش‌تعیین شده (برای مثال، 10^{-6})
- Early stopping because the validation error increases.
  - توقف زودهنگام به‌دلیل افزایش خطا در اعتبارسنجی
- Maximum number of iterations reached.
  - رسیدن به حداکثر تعداد تکرارها
- Mean square error drops below a specified threshold (not generally a useful method).
  - قرار گرفتن خطای میانگین مربعات زیر یک مقدار آستانه‌ی مشخص (عموماً روش مفیدی نیست)
- Mean square error curve (on a log-log scale) becomes flat for some time (user stop).
  - منحنی خطای میانگین مربعات (بر روی مقیاس لگاریتمی) برای یک مدت زمانی تخت شود.

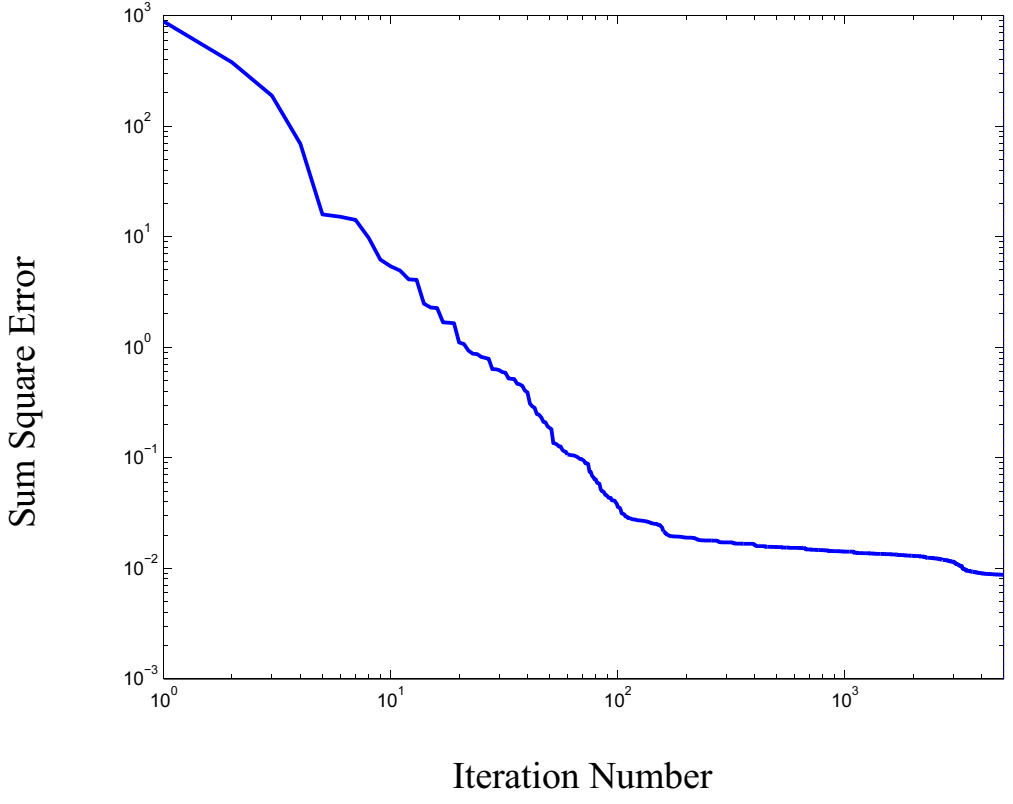
Typical Training Curve



ضابطه‌ی توقف

منحنی آموزش نوعی

TYPICAL TRAINING CURVE



22 Competitive Network Stopping Criteria

- Stop when a specified number of iterations has been reached.
- Learning rate and neighborhood size (SOM) are decreased during training, so that they reach their smallest values when the maximum number of iterations have been reached.
- Post-training analysis is used to determine if retraining is required.

ضابطه‌ی توقف

برای شبکه‌های رقابتی

COMPETITIVE NETWORK STOPPING CRITERIA

- Stop when a specified number of iterations has been reached.
  - هنگامی که به تعداد تکرار مشخص‌شده رسیدید، توقف کنید.
- Learning rate and neighborhood size (SOM) are decreased during training, so that they reach their smallest values when the maximum number of iterations have been reached.
  - نرخ یادگیری و اندازه‌ی همسایگی (SOM) در طول آموزش کاهش می‌یابد، به‌طوری که به کوچک‌ترین مقادیر خود می‌رسند، وقتی که به حداکثر تعداد تکرارها می‌رسیم.
- Post-training analysis is used to determine if retraining is required.
  - تحلیل‌های پسا–آموزش برای تعیین اینکه آیا آموزش مجدد لازم است یا خیر، استفاده می‌شوند.

Choice of Performance Function

Mean Square Error

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}(\mathbf{t}_q - \mathbf{a}_q)^T(\mathbf{t}_q - \mathbf{a}_q)$$

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}\sum_{i=1}^{S^M}(t_{i,q} - a_{i,q})^2$$

Minkowski error

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}\sum_{i=1}^{S^M}|t_{i,q} - a_{i,q}|^K$$

Cross-Entropy

$$F(\mathbf{x}) = -\sum_{q=1}^{Q}\sum_{i=1}^{S^M} t_{i,q}\ln\frac{a_{i,q}}{t_{i,q}}$$

انتخاب تابع کارآیی

CHOICE OF PERFORMANCE FUNCTION

Mean Square Error

ضریب نرمال‌سازی برای مقایسه‌ی خطا بر روی دو مجموعه داده با اندازه‌ی متفاوت، مفید است.

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}(\mathbf{t}_q - \mathbf{a}_q)^T(\mathbf{t}_q - \mathbf{a}_q)$$

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}\sum_{i=1}^{S^M}(t_{i,q} - a_{i,q})^2$$

Minkowski error

$$F(\mathbf{x}) = \frac{1}{QS^M}\sum_{q=1}^{Q}\sum_{i=1}^{S^M}|t_{i,q} - a_{i,q}|^K$$

Cross-Entropy

به‌همراه تابع انتقال «سافت‌مکس» در لایه‌ی آخر

$$F(\mathbf{x}) = -\sum_{q=1}^{Q}\sum_{i=1}^{S^M} t_{i,q}\ln\frac{a_{i,q}}{t_{i,q}}$$

Committees of Networks

- Restart training at 5 to 10 different initial conditions to be sure to reach a global minimum.
- You can also train several different networks with different initial conditions and different divisions of the data into training and validation sets. This produces a committee of networks.
- Take the average of the committee outputs to produce a more accurate fit than any of the individual networks.
- For pattern recognition problems, you can take a vote of the committee of networks to produce a more accurate classification.

سلسله آموزش‌های چندگانه و کمیته‌های شبکه‌ها

MULTIPLE TRAINING RUNS AND COMMITTEES OF NETWORKS

- Restart training at 5 to 10 different initial conditions to be sure to reach a global minimum.
  - آموزش را با ۵ الی ۱۰ شرایط آغازین مجدداً شروع کنید تا مطمئن شوید که به یک می‌نیمم سراسری رسیده‌اید.
- You can also train several different networks with different initial conditions and different divisions of the data into training and validation sets. This produces a committee of networks.
  - می‌توانید شبکه‌های متعدد متفاوتی را با شرایط آغازین متفاوت و تقسیم‌بندی‌های متفاوت از داده‌ها به مجموعه‌های آموزشی و اعتبارسنجی، آموزش بدهید. این یک کمیته از شبکه‌های عصبی را ایجاد می‌کند.
- Take the average of the committee outputs to produce a more accurate fit than any of the individual networks.
  - متوسط خروجی‌های کمیته را محاسبه کنید تا برازش دقیق‌تری نسبت به هر یک از شبکه‌های انفرادی ایجاد شود.
- For pattern recognition problems, you can take a vote of the committee of networks to produce a more accurate classification.
  - برای مسائل بازشناسی الگو، می‌توانید از کمیته‌ی شبکه‌ها رأی‌گیری کنید تا طبقه‌بندی دقیق‌تری تولید شود.

شبکه‌های عصبی مصنوعی

موضوعات مطرح در آموزش عملی

۳

گام‌های پس از آموزش

گام‌های آموزش شبکه

گام‌های پسا–آموزش

POST-TRAINING STEPS

گام‌های آموزش شبکه

۱	۲	۳
گام‌های پیشا–آموزش *Pre-Training Steps*	گام‌های هنگام آموزش *Training Steps*	**گام‌های پسا–آموزش** ***Post-Training Steps***
○ انتخاب داده‌ها ○ پیش‌پردازش داده‌ها ○ انتخاب معماری شبکه	○ مقداردهی آغازین وزن‌ها ○ انتخاب الگوریتم آموزش ○ تعیین ضابطه‌ی توقف ○ انتخاب تابع کارآیی ○ اجرای چند سلسله‌ی یادگیری	○ تحلیل کارآیی ○ تحلیل بیش‌برازش / برون‌یابی ○ تحلیل حساسیت

Post-Training Analysis

- Fitting
- Pattern Recognition
- Clustering
- Prediction

تحلیل‌های پسا-آموزش

برای انواع مسئله‌ها

POST-TRAINING ANALYSIS

برازش	طبقه‌بندی	خوشه‌بندی	پیش‌بینی
Fitting	*Classification*	*Clustering*	*Prediction*

Fitting

Regression Analysis (Outputs vs Targets)

$$a_q = mt_q + c + \varepsilon_q$$

$$\hat{m} = \frac{\sum_{q=1}^{Q} (t_q - \dot{t})(a_q - \bar{a})}{\sum_{q=1}^{Q} (t_q - \dot{t})^2}$$

$$\dot{t} = \sum_{q=1}^{Q} t_q$$

$$\hat{c} = \bar{a}\hat{m} - \bar{t}$$

$$\bar{a} = \sum_{q=1}^{Q} a_q$$

R Value ($-1<R<1$)

$$R = \frac{\sum_{q=1}^{Q} (t_q - \dot{t})(a_q - \bar{a})}{(Q-1)s_t s_a}$$

$$s_t = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q} (t_q - \dot{t})}$$

$$s_a = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q} (a_q - \bar{a})}$$

تحلیل‌های پسا–آموزش

برای مسائل برازش

FITTING

پیش‌بینی *Prediction*	خوشه‌بندی *Clustering*	طبقه‌بندی *Classification*	**برازش** *Fitting*

Regression Analysis (Outputs vs Targets)

$$a_q = mt_q + c + \varepsilon_q$$

$$\hat{c} = \bar{a}\hat{m} - \bar{t}$$

$$\hat{m} = \frac{\sum_{q=1}^{Q} (t_q - \dot{t})(a_q - \bar{a})}{\sum_{q=1}^{Q} (t_q - \dot{t})^2}$$

$$\dot{t} = \sum_{q=1}^{Q} t_q$$

$$\bar{a} = \sum_{q=1}^{Q} a_q$$

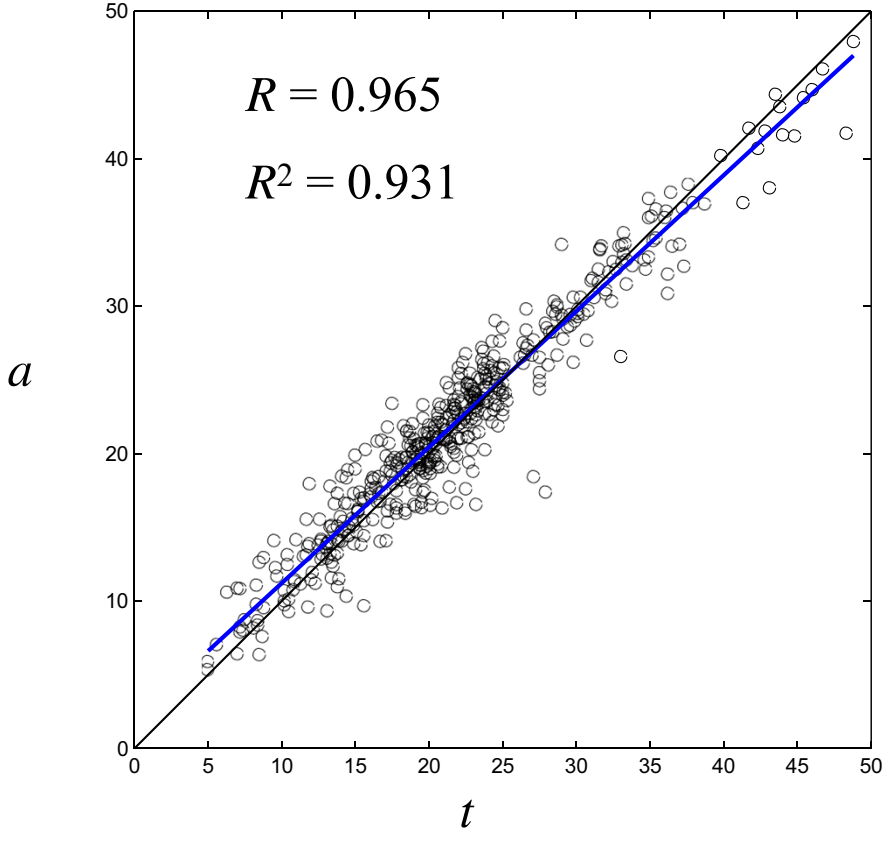
R Value ($-1<R<1$)

$$R = \frac{\sum_{q=1}^{Q} (t_q - \dot{t})(a_q - \bar{a})}{(Q-1)s_t s_a}$$

$$s_t = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q} (t_q - \dot{t})}$$

$$s_a = \sqrt{\frac{1}{Q-1}\sum_{q=1}^{Q} (a_q - \bar{a})}$$

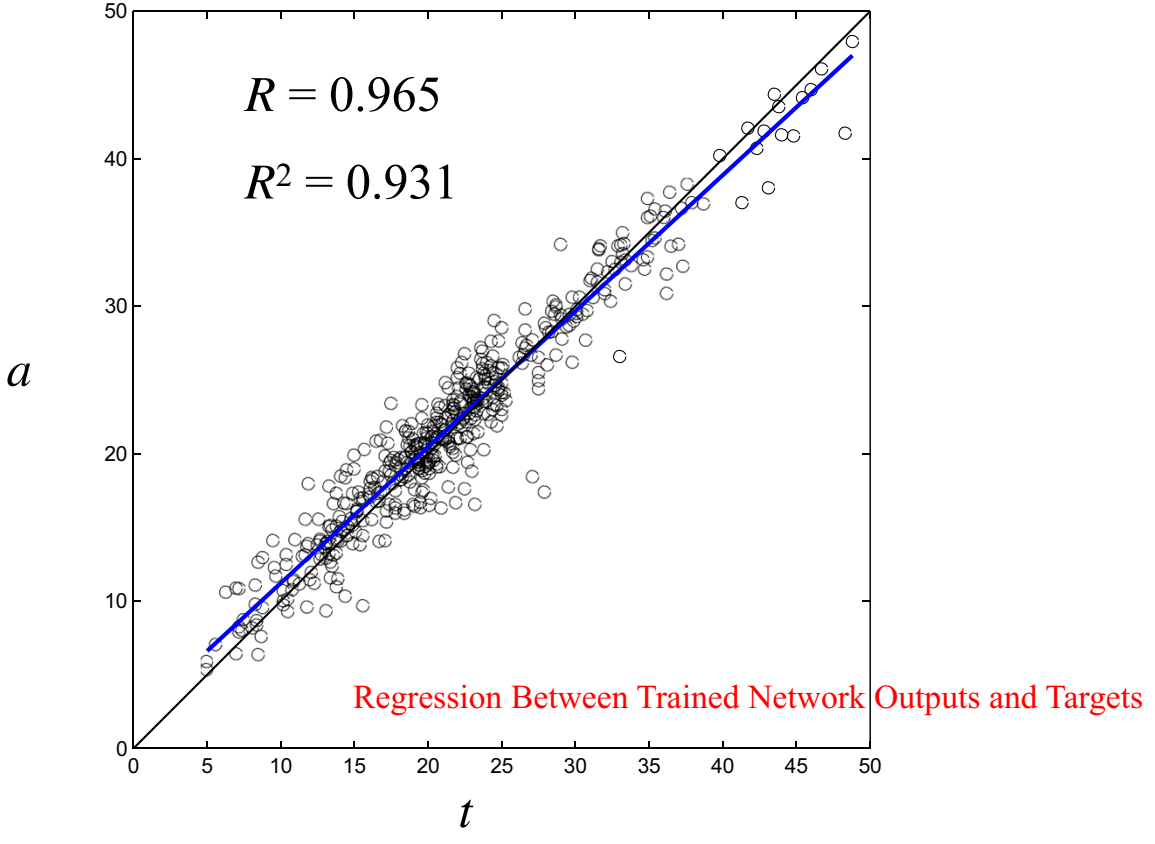
Sample Regression Plot



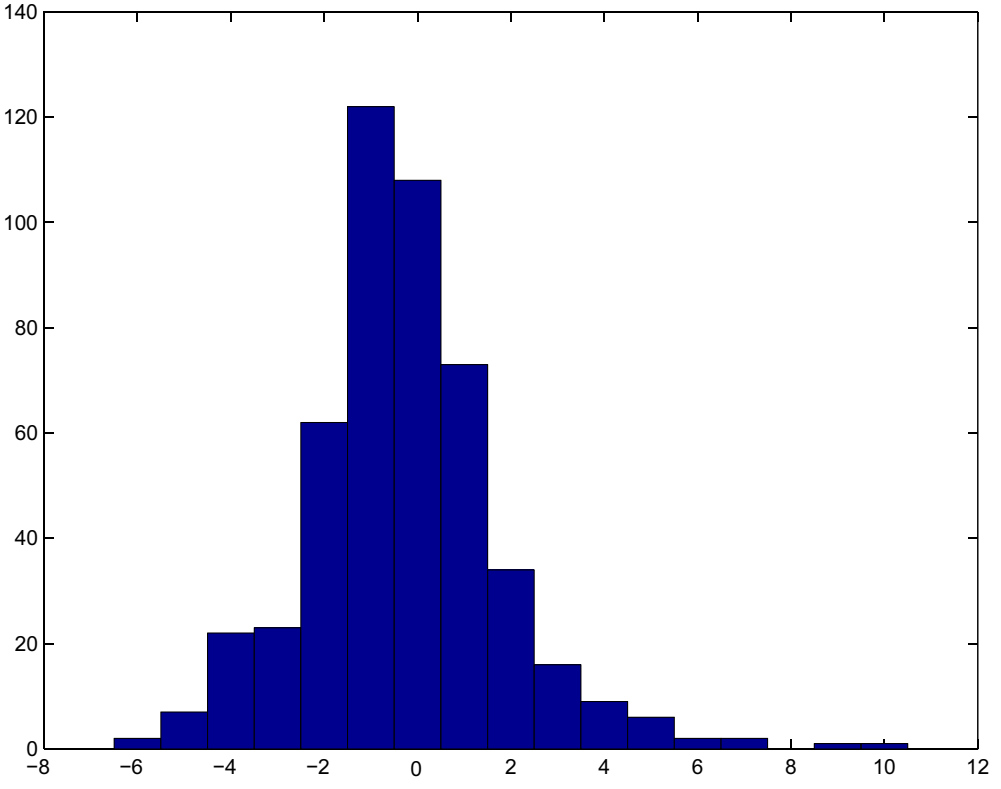
تحلیل‌های پسا–آموزش

برای مسائل برازش: یک نمودار رگرسیون نمونه

SAMPLE REGRESSION PLOT



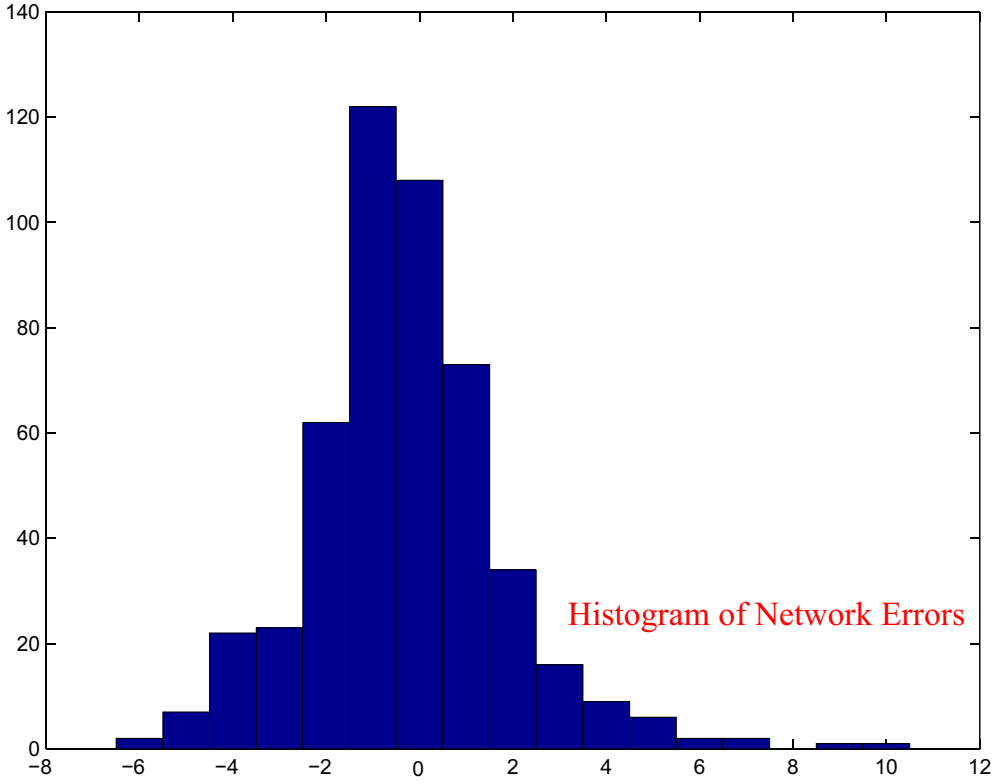
Error Histogram



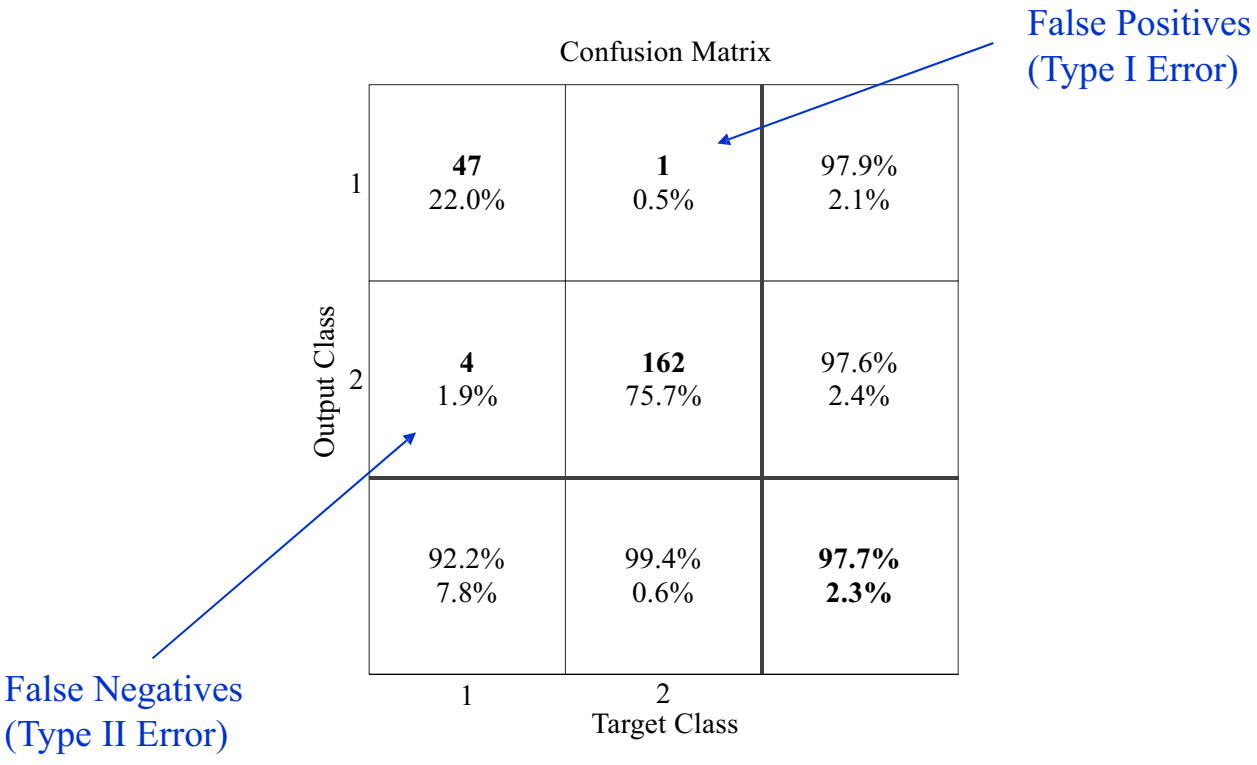
تحلیل‌های پسا–آموزش

برای مسائل برازش: یک نمودار هیستوگرام خطا

ERROR HISTOGRAM



Pattern Recognition

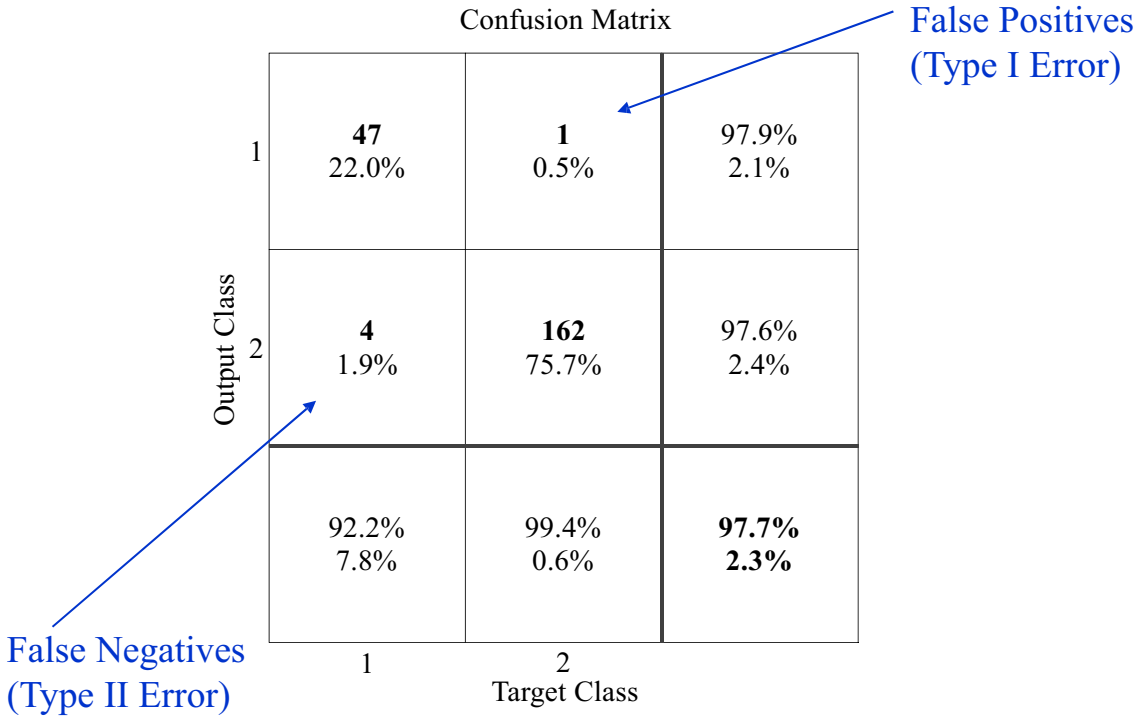


تحلیل‌های پسا–آموزش

برای مسائل طبقه‌بندی (بازشناسی الگو)

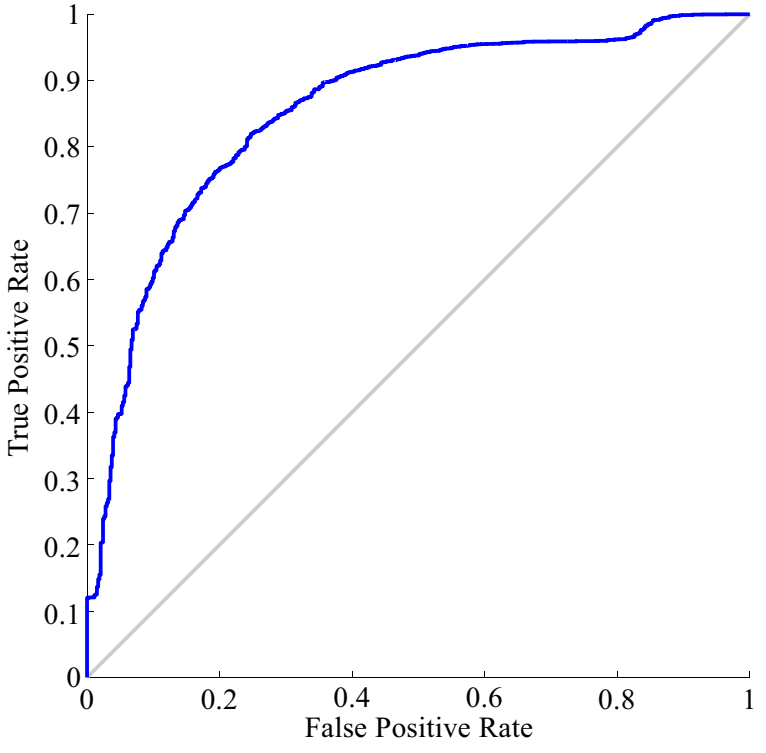
PATTERN RECOGNITION

پیش‌بینی *Prediction* | خوشه‌بندی *Clustering* | **طبقه‌بندی** *Classification* | برازش *Fitting*



Pattern Recognition

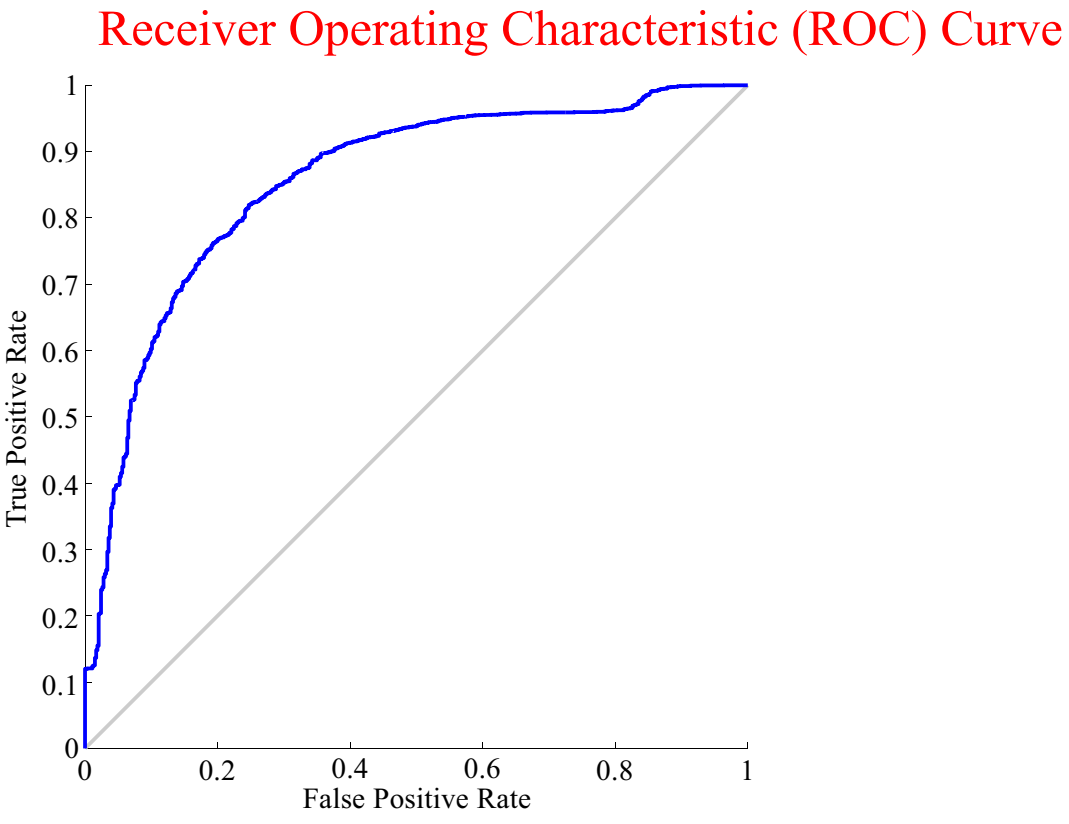
Receiver Operating Characteristic (ROC) Curve



تحلیل‌های پسا-آموزش

برای مسائل طبقه‌بندی (بازشناسی الگو): منحنی مشخصه‌ی عملکرد گیرنده

RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE



Clustering (SOM)

- Quantization Error. The average distance between each input vector and the closest prototype vector.
- Topographic Error. The proportion of all input vectors for which the closest prototype vector and the next closest prototype vector are not neighbors in the feature map topology.
- Distortion

$$E_d = \sum_{q=1}^{Q} \sum_{i=1}^{S} h_{ic_q} \left\| {}_i\mathbf{w} - \mathbf{p}_q \right\|^2$$

Prototype closest to the input vector.

$$c_q = \arg\min_j \{ \left\| {}_j\mathbf{w} - \mathbf{p}_q \right\| \}$$

$h_{i,j}$ = neighborhood function

$$h_{ij} = \exp\left(\frac{-\left\| {}_i\mathbf{w} - {}_j\mathbf{w} \right\|^2}{2d^2} \right)$$

تحلیل‌های پسا–آموزش

برای مسائل خوشه‌بندی

CLUSTERING (SOM)

پیش‌بینی Prediction	**خوشه‌بندی** *Clustering*	طبقه‌بندی Classification	برازش Fitting

- خطای چندی‌سازی. فاصله‌ی متوسط بین هر بردار ورودی و نزدیک‌ترین بردار پروتوتایپ
  - Quantization Error. The average distance between each input vector and the closest prototype vector.
- خطای توپوگرافیک. نسبت همه‌ی بردارهای ورودی که برای آنها نزدیک‌ترین بردار پروتوتایپ و نزدیک‌ترین بردار پروتوتایپ بعدی، در توپولوژی نقشه‌ی ویژگی همسایه نیستند.
  - Topographic Error. The proportion of all input vectors for which the closest prototype vector and the next closest prototype vector are not neighbors in the feature map topology.
- معیار اعوجاج
  - Distortion Measure

Distortion Measure

$$E_d = \sum_{q=1}^{Q} \sum_{i=1}^{S} h_{ic_q} \|_i\mathbf{w} - \mathbf{p}_q\|^2$$

Prototype closest to the input vector.

$$c_q = \arg\min_j \{ \|_j\mathbf{w} - \mathbf{p}_q\| \}$$

$$h_{i,j} = \text{neighborhood function}$$

$$h_{ij} = \exp\left(\frac{-\|_i\mathbf{w} - {}_j\mathbf{w}\|^2}{2d^2} \right)$$

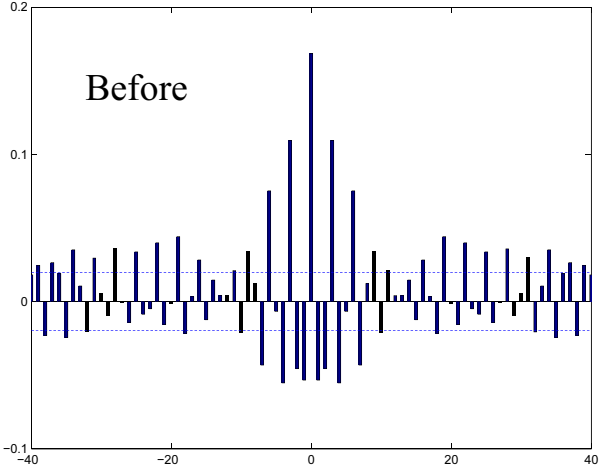
Prediction

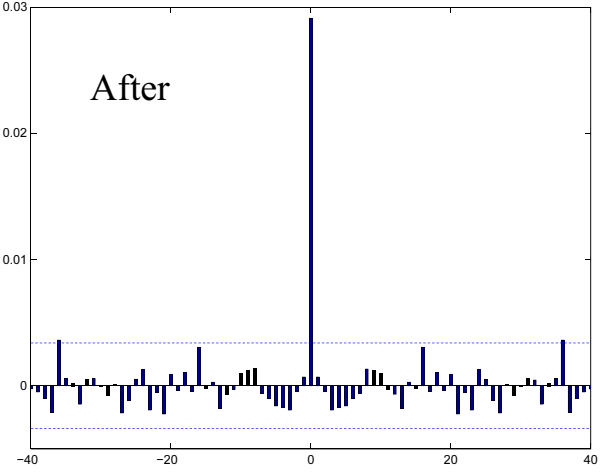
Autocorrelation Function of Prediction Errors.

$$R_e(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} e(t)e(t+\tau)$$

Confidence Intervals.

$$-\frac{2R_e(0)}{\sqrt{Q}} < R_e(\tau) < \frac{2R_e(0)}{\sqrt{Q}}$$





تحلیل‌های پسا–آموزش

برای مسائل پیش‌بینی

PREDICTION

پیش‌بینی *Prediction*	خوشه‌بندی *Clustering*	طبقه‌بندی *Classification*	برازش *Fitting*

Autocorrelation Function of Prediction Errors.

برای تعیین میزان همبستگی با زمان

$$R_e(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} e(t)e(t+\tau)$$

Confidence Intervals. (95%)

$$-\frac{2R_e(0)}{\sqrt{Q}} < R_e(\tau) < \frac{2R_e(0)}{\sqrt{Q}}$$

(۱) خطای‌های پیش‌بینی نباید با زمان همبستگی داشته باشد.

* همبستگی در خطاهای پیش‌بینی، می‌تواند نشان‌دهنده‌ی آن باشد که طول خطوط TDL در شبکه باید افزایش یابد.

تحلیل‌های پسا–آموزش

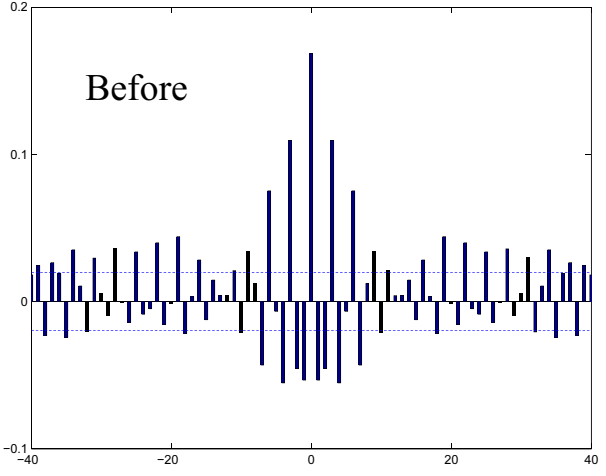
برای مسائل پیش‌بینی

PREDICTION

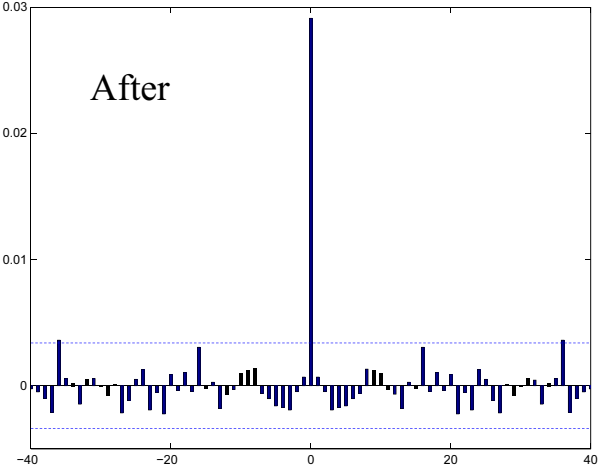
$$R_e(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} e(t)e(t+\tau)$$

Autocorrelation Function of Prediction Errors.

$R_e(\tau)$
for Inadequately Trained Network



$R_e(\tau)$
for Successfully Trained Network



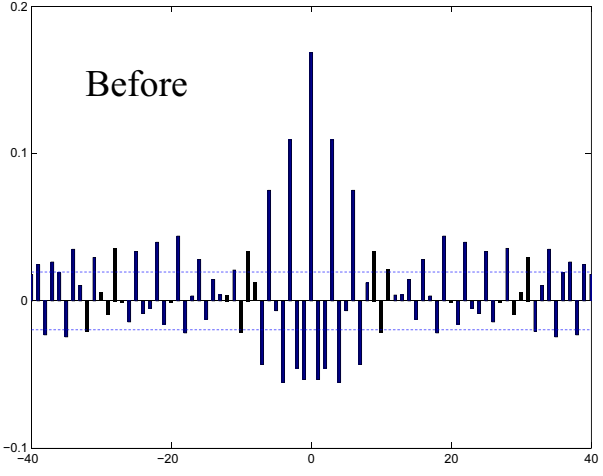
Prediction

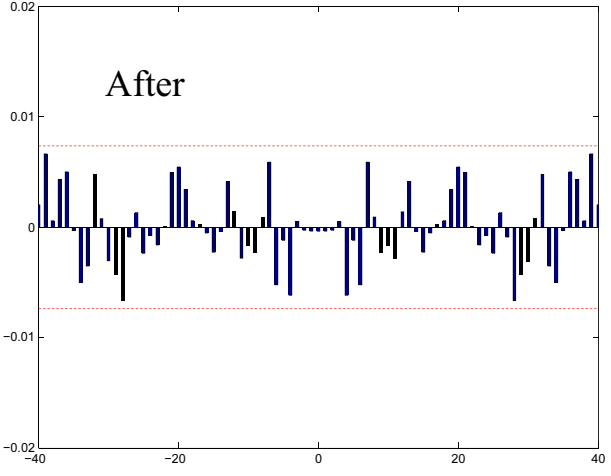
Cross-correlation Between Prediction Errors and Input.

$$R_{pe}(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} p(t)\,e(t+\tau)$$

Confidence Intervals.

$$-\frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}} < R_{pe}(\tau) < \frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}}$$





تحلیل‌های پسا–آموزش

برای مسائل پیش‌بینی

PREDICTION

پیش‌بینی *Prediction*	خوشه‌بندی *Clustering*	طبقه‌بندی *Classification*	برازش *Fitting*

Cross-correlation Between Prediction Errors and Input.

برای تعیین میزان همبستگی با دنباله‌ی ورودی

$$R_{pe}(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} p(t)\,e(t+\tau)$$

Confidence Intervals. (95%)

$$-\frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}} < R_{pe}(\tau) < \frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}}$$

(۲) خطای‌های پیش‌بینی نباید با دنباله‌ی ورودی همبستگی داشته باشد.

تحلیل‌های پسا–آموزش

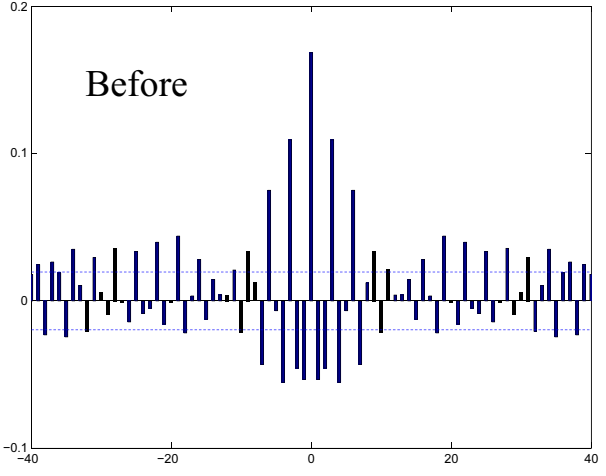
برای مسائل پیش‌بینی

PREDICTION

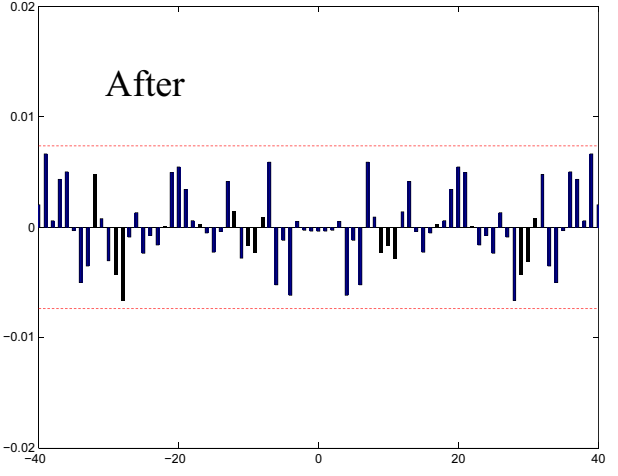
$$R_{pe}(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} p(t)\,e(t+\tau)$$

Cross-correlation Between Prediction Errors and Input.

$R_{pe}(\tau)$
for Inadequately Trained Network



$R_{pe}(\tau)$
for Successfully Trained Network



Overfitting and Extrapolation

If, after a network has been trained, the test set performance is not adequate, then there are usually four possible causes:

- the network has reached a local minimum,
- the network does not have enough neurons to fit the data,
- the network is overfitting, or
- the network is extrapolating.

بیش‌برازش و برون‌یابی

OVERFITTING AND EXTRAPOLATION

اگر پس از اینکه یک شبکه آموزش داده شد، کارآیی مجموعه‌ی آزمایشی کافی نبود، آن‌گاه معمولاً چهار علت ممکن می‌تواند وجود داشته باشد:

If, after a network has been trained, the test set performance is not adequate, then there are usually four possible causes:

- شبکه به یک می‌نیمم محلی رسیده است.
  - the network has reached a local minimum,
- شبکه تعداد نرون کافی برای برازش داده‌ها ندارد
  - the network does not have enough neurons to fit the data,
- شبکه دچار بیش‌برازش شده، یا
  - the network is overfitting, or
- شبکه برون‌یابی کرده است.
  - the network is extrapolating.

Diagnosing Problems

- The local minimum problem can almost always be overcome by retraining the network with five to ten random sets of initial weights.
- If the validation error is much larger than the training error, then overfitting has probably occurred.
- If the validation, training and test errors are all similar in size, but the errors are too large, then the network is not powerful enough to fit the data. Add neurons.
- If the validation and training errors are similar in size, but the test errors are significantly larger, then the network may be extrapolating.
- If training, validation and test errors are similar, and the errors are small enough, then we can put the multilayer network to use.

بیش‌برازش و برون‌یابی

تشخیص مشکل در مسئله‌ها

DIAGNOSING PROBLEMS

- The local minimum problem can almost always be overcome by retraining the network with five to ten random sets of initial weights.
- مشکل می‌نیمم محلی تقریباً همیشه می‌تواند از طریق آموزش مجدد شبکه با ۵ الی ۱۰ مجموعه‌ی تصادفی از وزن‌های آغازین حل شود.

- If the validation error is much larger than the training error, then overfitting has probably occurred.
- اگر خطای اعتبارسنجی بسیار بزرگ‌تر از خطای آموزش بود، آن‌گاه احتمالاً بیش‌برازش اتفاق افتاده است.

- If the validation, training and test errors are all similar in size, but the errors are too large, then the network is not powerful enough to fit the data. Add neurons.
- اگر خطاهای اعتبارسنجی، آموزشی و آزمایشی همگی از نظر اندازه مشابه بودند، اما این خطاها بزرگ بسیار بودند، آن‌گاه شبکه برای برازش داده‌ها به‌اندازه‌ی کافی قدرت‌مند نیست. به آن نرون اضافه کنید.

- If the validation and training errors are similar in size, but the test errors are significantly larger, then the network may be extrapolating.
- اگر خطاهای اعتبارسنجی و آموزشی از نظر اندازه مشابه بودند، اما خطاهای آزمایشی به‌طور چشمگیری بزرگ‌تر بودند، آن‌گاه شبکه ممکن است برون‌یابی کرده باشد.

- If training, validation and test errors are similar, and the errors are small enough, then we can put the multilayer network to use.
- اگر خطاهای آموزشی، اعتبارسنجی و آزمایشی همگی از نظر اندازه مشابه بودند و این خطاها به‌اندازه‌ی کافی کوچک بودند، آن‌گاه می‌توانیم از این شبکه استفاده کنیم.

Novelty Detection

- To detect extrapolation, train a companion competitive network to cluster the input vectors of the training set.
- When an input is applied to the multilayer network, the same input is applied to the companion competitive network.
- When the distance of the input vector to the nearest prototype vector of the competitive network is larger than the distance from the prototype to the most distant member of its cluster of inputs in the training set, we can suspect extrapolation.

بیش‌برازش و برون‌یابی

آشکارسازی تازگی

NOVELTY DETECTION

- To detect extrapolation, train a companion competitive network to cluster the input vectors of the training set.
- When an input is applied to the multilayer network, the same input is applied to the companion competitive network.
- When the distance of the input vector to the nearest prototype vector of the competitive network is larger than the distance from the prototype to the most distant member of its cluster of inputs in the training set, we can suspect extrapolation.

- برای آشکارسازی برون‌یابی، یک شبکه‌ی رقابتی را در کنار شبکه‌ی اصلی آموزش دهید تا بردارهای ورودی مجموعه‌ی آموزشی را خوشه‌بندی کند.
- وقتی یک ورودی به شبکه‌ی چندلایه اعمال می‌شود، همان ورودی به شبکه‌ی رقابتی همراه آن نیز اعمال می‌شود.
- وقتی فاصله‌ی بردار ورودی تا نزدیک‌ترین بردار پروتوتایپ از شبکه‌ی رقابتی، بزرگ‌تر از فاصله‌ی بردار پروتوتایپ به دورترین عضو از خوشه‌ی آن در ورودی‌های واقع در مجموعه‌ی آموزشی شد، آن‌گاه می‌توانیم به برون‌یابی مشکوک شویم.

Sensitivity Analysis

Check for important inputs.

$$s_i^m \equiv \frac{\partial F}{\partial n_i^m}$$

$$\frac{\partial \hat{F}}{\partial p_j} = \sum_{i=1}^{S^1} \frac{\partial \hat{F}}{\partial n_i^1} \times \frac{\partial n_i^1}{\partial p_j} = \sum_{i=1}^{S^1} s_i^1 \times \frac{\partial n_i^1}{\partial p_j}$$

$$n_i^1 = \sum_{j=1}^{R} w_{i,j}^1 p_j + b_i^1$$

$$\frac{\partial \hat{F}}{\partial p_j} = \sum_{i=1}^{S^1} \frac{\partial \hat{F}}{\partial n_i^1} \times \frac{\partial n_i^1}{\partial p_j} = \sum_{i=1}^{S^1} s_i^1 \times w_{i,j}^1$$

$$\frac{\partial \hat{F}}{\partial \mathbf{p}} = (\mathbf{W}^1)^T \mathbf{s}^1$$

تحلیل حساسیت

SENSITIVITY ANALYSIS

Check for important inputs.

$$s_i^m \equiv \frac{\partial F}{\partial n_i^m}$$

$$\frac{\partial \hat{F}}{\partial p_j} = \sum_{i=1}^{S^1} \frac{\partial \hat{F}}{\partial n_i^1} \times \frac{\partial n_i^1}{\partial p_j} = \sum_{i=1}^{S^1} s_i^1 \times \frac{\partial n_i^1}{\partial p_j}$$

$$n_i^1 = \sum_{j=1}^{R} w_{i,j}^1 p_j + b_i^1$$

$$\frac{\partial \hat{F}}{\partial p_j} = \sum_{i=1}^{S^1} \frac{\partial \hat{F}}{\partial n_i^1} \times \frac{\partial n_i^1}{\partial p_j} = \sum_{i=1}^{S^1} s_i^1 \times w_{i,j}^1$$

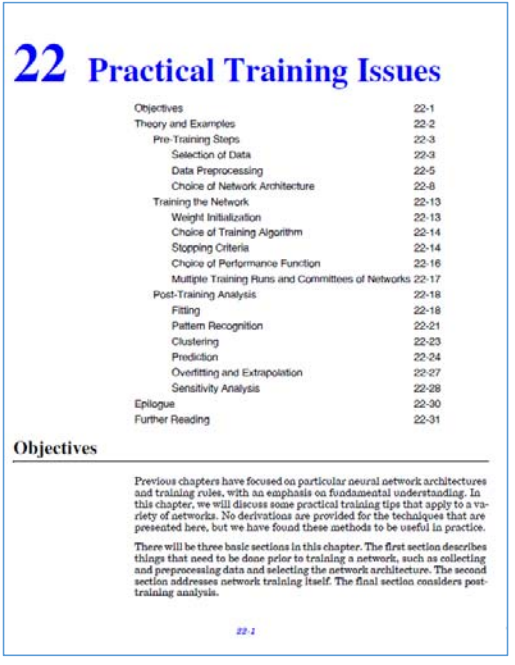
$$\frac{\partial \hat{F}}{\partial \mathbf{p}} = (\mathbf{W}^1)^T \mathbf{s}^1$$

موضوعات مطرح در آموزش عملی

۴

منابع

منبع اصلی



22 Practical Training Issues

Objectives

Previous chapters have focused on particular neural network architectures and training rules, with an emphasis on fundamental understanding. In this chapter, we will discuss some practical training tips that apply to a variety of networks. No derivations are provided for the techniques that are presented here, but we have found these methods to be useful in practice.

There will be three basic sections in this chapter. The first section describes things that need to be done prior to training a network, such as collecting and preprocessing data and selecting the network architecture. The second section addresses network training itself. The final section considers post-training analysis.

22-1

Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus,
Neural Network Design,
2nd Edition, Martin Hagan, 2014.
Chapter 22

Online version can be downloaded from: http://hagan.okstate.edu/nnd.html