راهنمای جامع و عمیق

نظریه یادگیری آماری، مدل‌های احتمالی، کمی‌سازی عدم قطعیت، مقاومت و تاب‌آوری در یادگیری ماشین

از شهود و تعریف تا فرمول، معیار ارزیابی، الگوریتم و مثال‌های کاربردی

نسخه آموزشی فارسی - سطح پیشرفته / آمادگی پژوهش دکتری

راهنمای جامع مفاهیم بنیادی یادگیری ماشین قابل اعتماد

مقدمه: این جزوه دقیقاً چه چیزی را توضیح می‌دهد؟

پنج اصطلاحی که در این جزوه بررسی می‌شوند، اغلب در مقالات یادگیری ماشین کنار هم دیده می‌شوند، اما معنای آن‌ها یکسان نیست. «نظریه یادگیری آماری» درباره این است که چرا و تحت چه شرایطی یادگیری از نمونه محدود به داده‌های ندیده تعمیم پیدا می‌کند. «مدل‌های احتمالی» زبان ریاضی برای نمایش عدم قطعیت و ساخت پیش‌بینی‌های احتمالی هستند. «کمی‌سازی عدم قطعیت» می‌پرسد چگونه میزان ناآگاهی یا نویز موجود در یک پیش‌بینی را بسنجیم. «مقاومت» بررسی می‌کند آیا مدل در برابر اغتشاش، حمله یا تغییر توزیع هنوز خوب کار می‌کند. «تاب‌آوری» یک گام سیستم‌محورتر است: اگر مدل یا سامانه شکست خورد، آیا شکست را تشخیص می‌دهد، سازگار می‌شود و به وضعیت قابل قبول بازمی‌گردد؟

هدف این متن فقط ارائه تعریف نیست. برای هر مفهوم چهار سطح ارائه می‌شود: شهود، صورت‌بندی ریاضی، روش‌های عملی و دام‌های تفسیری. در بخش‌های مختلف از مثال‌های طبقه‌بندی تصویر، تشخیص پزشکی، سامانه‌های حسگری و یادگیری آنلاین استفاده می‌شود. هر جا لازم باشد، تفاوت میان «اعتماد مدل» و «احتمال واقعی»، میان «تغییر توزیع» و «نمونه خارج از توزیع»، و میان «مقاوم بودن» و «بازیابی پس از شکست» صریحاً جدا می‌شود.

نقشه ذهنی کل جزوه

Learning Theory می‌گوید چه چیزی را می‌توان از داده آموخت؛ Probabilistic Modeling می‌گوید عدم قطعیت را چگونه نمایش دهیم؛ UQ می‌گوید آن عدم قطعیت را چگونه اندازه‌گیری و ارزیابی کنیم؛ Robustness می‌گوید عملکرد در برابر اغتشاش چگونه حفظ شود؛ Resilience می‌گوید اگر حفظ نشد، سامانه چگونه شکست را مدیریت و بازیابی کند.

پیش‌نیاز ریاضی پیشنهادی

- احتمال و آمار: متغیر تصادفی، امید ریاضی، واریانس، احتمال شرطی و قانون بیز.
- جبر خطی: بردار، ماتریس، نرم‌ها، مقادیر ویژه در حد مفهومی.
- بهینه‌سازی: گرادیان، تابع هدف، کمینه‌سازی تجربی، قیود.
- یادگیری ماشین: classification، loss، train/validation/test و regression.

راهنمای جامع مفاهیم بنیادی یادگیری ماشین قابل اعتماد

فصل 1 - صورت‌بندی بنیادی مسئله یادگیری

1.1 از داده تا تابع پیش‌بینی

در یادگیری نظارت‌شده، یک فضای ورودی X و یک فضای خروجی Y داریم. نمونه‌ها به شکل زوج‌های (x,y) مشاهده می‌شوند. x می‌تواند تصویر، سیگنال، بردار ویژگی یا متن باشد و y می‌تواند کلاس، مقدار پیوسته یا ساختار پیچیده باشد. فرض نظری متداول این است که داده‌ها از یک توزیع ناشناخته P(X,Y) تولید می‌شوند. الگوریتم یادگیری، تنها یک نمونه محدود S={(x_i,y_i)} را می‌بیند و باید تابعی f را انتخاب کند که روی نمونه‌های آینده نیز خطای کمی داشته باشد.

$$S = \{(x_i, y_i)\}_{i=1}^n \sim P(X,Y)$$

نکته ظریف این است که هدف ما کمینه‌کردن خطا روی مجموعه آموزش نیست. هدف واقعی کمینه‌کردن «ریسک جمعیتی» یا Expected Risk است: متوسط خطا روی تمام نمونه‌هایی که ممکن است از توزیع واقعی ظاهر شوند. چون P ناشناخته است، این کمیت مستقیماً قابل محاسبه نیست. بنابراین از ریسک تجربی روی داده آموزش به‌عنوان تقریب استفاده می‌کنیم.

$$R(f) = E_{(X,Y)\sim P}[\ L(f(X),Y)\ ]$$
$$R_hat_S(f) = (1/n) * sum_{i=1}^n L(f(x_i), y_i)$$

1.2 **Loss** دقیقاً چیست؟

تابع زیان مشخص می‌کند «بد بودن» یک پیش‌بینی چگونه عددی شود. در رگرسیون، squared error به خطاهای بزرگ وزن بیشتری می‌دهد. در طبقه‌بندی احتمالی، cross-entropy یا negative log-likelihood مدل را تشویق می‌کند به کلاس صحیح احتمال بالا اختصاص دهد. انتخاب loss بخشی از تعریف مسئله است؛ بنابراین دو مدل با معماری یکسان ولی loss متفاوت ممکن است رفتار آماری و مقاومتی کاملاً متفاوتی پیدا کنند.

نوع مسئله	زیان رایج	نکته
رگرسیون	Squared Error	به خطاهای بزرگ حساس است و با فرض نویز گاوسی سازگار است.
رگرسیون مقاوم	Absolute / Huber	در برابر outlierها کمتر از MSE آسیب می‌بیند.

نکته	زیان رایج	نوع مسئله
با برآورد likelihood برای توزیع categorical مرتبط است.	Cross-Entropy	طبقه‌بندی
نسخه دودویی negative log-likelihood است.	Logistic Loss	طبقه‌بندی دودویی
در SVM و مسائل تفکیک حاشیه‌ای ظاهر می‌شود.	Margin Loss	رتبه‌بندی/فاصله

1.3 فرض **i.i.d**. و اینکه چرا مهم است

بخش بزرگی از نظریه کلاسیک فرض می‌کند نمونه‌ها مستقل و هم‌توزیع (independent and identically distributed) هستند. «هم‌توزیع» یعنی train و test از یک قانون آماری مشترک آمده‌اند؛ «استقلال» یعنی دانستن یک نمونه، نمونه بعدی را تعیین نمی‌کند. در بسیاری از داده‌های واقعی این فرض دقیق نیست: داده‌های زمانی هم‌بستگی دارند، داده‌های بیمارستان‌ها تفاوت دامنه دارند و داده‌های آنلاین در طول زمان drift می‌کنند. دقیقاً از همین‌جا Robustness و Resilience اهمیت پیدا می‌کنند: آن‌ها می‌پرسند اگر فرض کلاسیک شکسته شد چه می‌شود.

یک تفکیک کلیدی

Generalization کلاسیک عمدتاً درباره عملکرد روی نمونه جدید از «همان توزیع» است. Robustness در برابر perturbation یا shift و Resilience در برابر failure-and-recovery، مسئله را از این محدوده فراتر می‌برند.

1.4 فضای فرضیه و ظرفیت مدل

الگوریتم از میان مجموعه‌ای از توابع ممکن H یا Hypothesis Class یک تابع را انتخاب می‌کند. برای رگرسیون خطی، H مجموعه همه توابع خطی با پارامترهای مجاز است. برای شبکه عصبی، H بسیار بزرگ‌تر است. هرچه H انعطاف‌پذیرتر باشد، امکان fit کردن الگوهای پیچیده بیشتر می‌شود، اما امکان fit کردن نویز نیز افزایش می‌یابد. Statistical Learning Theory تلاش می‌کند رابطه میان اندازه داده، پیچیدگی H و شکاف بین empirical risk و true risk را کمی کند.

فصل 2 - Statistical Learning Theory: تعمیم، ERM و بیش‌برازش

2.1 Empirical Risk Minimization

اصل ERM می‌گوید تابعی را انتخاب کن که ریسک تجربی را در فضای فرضیه کمینه کند. تقریباً تمام آموزش استاندارد شبکه‌های عصبی را می‌توان نوعی ERM یا regularized ERM دید: با SGD پارامترهایی را پیدا می‌کنیم که میانگین loss روی mini-batchها را کاهش دهند.

f_hat = argmin_{f in H} R_hat_S(f)

ERM به خودی خود تضمین نمی‌کند مدل تعمیم خوبی داشته باشد. اگر H آن‌قدر بزرگ باشد که بتواند هر برچسب تصادفی را حفظ کند، ممکن است R_hat نزدیک صفر شود اما R روی جمعیت بالا بماند. بنابراین سؤال نظری اصلی این است: تحت چه شرایطی R_hat یک تقریب قابل اعتماد برای R است؟

2.2 Generalization Gap

Generalization Gap = R(f) - R_hat_S(f)

شکاف تعمیم را نباید با test error یکی دانست. test error یک برآورد تجربی از R است. شکاف تعمیم اختلاف میان خطای واقعی جمعیت و خطای آموزش است. در عمل، train-test gap یک نشانه است، اما به اندازه test set، leakage، انتخاب مدل و تعداد دفعات نگاه‌کردن به test وابسته است.

2.3 Underfitting و Overfitting

Underfitting زمانی است که مدل حتی ساختار اصلی داده آموزش را نمی‌گیرد؛ معمولاً training error نیز بالا است. Overfitting زمانی است که مدل جزئیات خاص نمونه آموزش را جذب می‌کند و عملکرد خارج از نمونه افت می‌کند. این دو مفهوم فقط به اندازه شبکه وابسته نیستند. آموزش طولانی، داده کم، label noise، انتخاب ویژگی نامناسب و tuning مکرر روی validation همگی می‌توانند overfitting ایجاد کنند.

وضعیت	Training Error	Validation/Test Error	تفسیر
Underfitting	بالا	بالا	مدل/ویژگی/بهینه‌سازی ناکافی

وضعیت	Training Error	Validation/Test Error	تفسیر
Fit مناسب	پایین	پایین و نزدیک به train	تعمیم مناسب در توزیع مشابه
Overfitting	خیلی پایین	به‌طور معنادار بالاتر	مدل داده آموزش را بیش از حد دنبال کرده
Shift	ممکن است پایین	بالا	لزومی ندارد overfitting باشد؛ test ممکن است از توزیع دیگری آمده باشد

2.4 نقش Validation و Test

Validation set برای تصمیم‌های مدل‌سازی استفاده می‌شود: hyperparameter، stopping، architecture و threshold. Test set باید تا حد امکان تنها در انتهای فرایند برای برآورد unbiasedتر عملکرد استفاده شود. اگر صد بار روی test تصمیم بگیریم، test عملاً به validation تبدیل می‌شود و برآورد خوش‌بینانه خواهد شد. در پژوهش جدی، nested cross-validation یا یک external test set می‌تواند این خطر را کاهش دهد.

2.5 No Free Lunch و اهمیت فرض‌ها

یادگیری بدون فرض ممکن نیست. اگر هیچ ساختاری درباره رابطه X و Y، smoothness، sparsity، invariance یا شکل توزیع نپذیریم، از نمونه‌های محدود نمی‌توان درباره نقاط ندیده نتیجه گرفت. «No Free Lunch» به‌طور شهودی یادآوری می‌کند که الگوریتمی که روی همه مسائل ممکن برتر باشد وجود ندارد. موفقیت هر روش ناشی از inductive bias مناسب آن برای یک خانواده مسئله است.

فصل 3 - ظرفیت مدل، **VC Dimension**، **PAC** و کران‌های تعمیم

3.1 **Concentration**: چرا میانگین نمونه‌ای به میانگین واقعی نزدیک می‌شود؟

اگر یک تابع ثابت f داشته باشیم و loss محدود باشد، قضایای تمرکز مانند Hoeffding می‌گویند احتمال اینکه میانگین تجربی به‌طور زیادی از امید واقعی فاصله بگیرد با افزایش n به‌سرعت کم می‌شود. این پایه شهودی بسیاری از generalization boundها است.

P(|R_hat(f)-R(f)| >= epsilon) <= 2 exp(-2 n epsilon^2)

اما در یادگیری، f قبل از دیدن داده ثابت نیست؛ الگوریتم بعد از دیدن S یک f مناسب انتخاب می‌کند. بنابراین باید نه فقط یک تابع، بلکه تمام توابعی را که الگوریتم ممکن است انتخاب کند کنترل کنیم. اینجاست که پیچیدگی کلاس H وارد می‌شود.

3.2 **Finite Hypothesis Class** و **Union Bound**

اگر H محدود و شامل |H| تابع باشد، می‌توان برای همه آن‌ها به‌طور همزمان یک کران ساخت. هزینه این آزادی انتخاب معمولاً به صورت |log|H ظاهر می‌شود. پیام اصلی: برای اطمینان یکسان، کلاس بزرگ‌تر داده بیشتری می‌طلبد.

R(f) <= R_hat(f) + sqrt((log |H| + log(2/delta))/(2n))

این فرمول نمونه‌ای آموزشی است: ثابت‌ها بسته به قضیه متفاوت‌اند، اما ساختار مهم است. خطای واقعی تقریباً از «خطای تجربی + جریمه پیچیدگی» تشکیل می‌شود. Regularization نیز از نظر مفهومی همین معامله را پیاده می‌کند.

3.3 **PAC Learning**

PAC مخفف Probably Approximately Correct است. به زبان ساده، می‌خواهیم با احتمال حداقل 1-delta، مدلی پیدا کنیم که خطای آن حداکثر epsilon با بهترین یا هدف مطلوب فاصله داشته باشد. epsilon دقت مورد نظر و delta احتمال شکست تضمین است. PAC یک زبان برای بیان sample complexity فراهم می‌کند: چند نمونه لازم است تا به چنین تضمینی برسیم؟

3.4 VC Dimension

VC Dimension معیاری از ظرفیت یک کلاس طبقه‌بندی دودویی است. یک مجموعه نقاط را «shatter» می‌کنیم اگر برای هر برچسب‌گذاری ممکن روی آن نقاط، تابعی در H وجود داشته باشد که آن برچسب‌گذاری را دقیقاً بازتولید کند. VC dimension بزرگ‌ترین تعداد نقاطی است که H می‌تواند shatter کند. این معیار تعداد پارامترها نیست، هرچند معمولاً با ظرفیت مدل ارتباط دارد.

مثال کلاسیک: آستانه روی خط حقیقی VC dimension برابر 1 دارد. بازه‌ها روی خط VC dimension برابر 2 دارند. جداکننده‌های خطی در R^d در شرایط استاندارد VC dimension تقریباً d+1 دارند. پیام مهم این است که ظرفیت باید نسبت به n سنجیده شود؛ مدل پرظرفیت لزوماً بد نیست، اما برای کنترل خطای تعمیم نیازمند داده، regularization یا inductive bias مناسب است.

3.5 Rademacher Complexity

Rademacher Complexity معیاری data-dependentتر از ظرفیت است. ایده این است: اگر برچسب‌های تصادفی +1/-1 روی نمونه بگذاریم، کلاس H تا چه حد می‌تواند این نویز تصادفی را fit کند؟ اگر به‌راحتی بتواند، ظرفیت مؤثر بالاست. این معیار برای کلاس‌های تابع واقعی و تحلیل‌های مدرن‌تر مفید است.

3.6 چرا کران‌های نظری شبکه‌های عمیق اغلب شُل هستند؟

شبکه‌های عمیق میلیون‌ها پارامتر دارند و می‌توانند labelهای تصادفی را حفظ کنند، اما در عمل با SGD و ساختار داده اغلب تعمیم خوبی دارند. کران‌های کلاسیک بر اساس بدترین حالت می‌توانند بسیار محافظه‌کار باشند. نظریه مدرن به stability، norm-based bounds، margin، implicit regularization و compression نیز توجه می‌کند. بنابراین Statistical Learning Theory مجموعه‌ای از ابزارهاست، نه یک فرمول واحد که رفتار همه شبکه‌های عصبی را کامل توضیح دهد.

دام رایج

VC dimension یا تعداد پارامتر را نباید مستقیماً به «احتمال overfitting» تبدیل کرد. ظرفیت تنها یک جزء است؛ optimizer، داده، margin، augmentation، noise و structure نیز اثر دارند.

فصل 4 - Regularization، Bias-Variance و انتخاب مدل

4.1 Bias-Variance در رگرسیون

برای squared-error regression و تحت فروض کلاسیک، خطای مورد انتظار را می‌توان به نویز غیرقابل حذف، مربع bias و variance تجزیه کرد. Bias یعنی پیش‌بینی متوسط مدل تا چه حد از تابع واقعی فاصله دارد. Variance یعنی مدل با تغییر نمونه آموزش چقدر تغییر می‌کند.

Expected Test Error = Noise + Bias^2 + Variance

مدل ساده معمولاً bias بیشتر و variance کمتر دارد. مدل بسیار منعطف می‌تواند bias را کم کند اما به نمونه آموزش حساس‌تر شود. با این حال، در deep learning پدیده‌هایی مانند double descent نشان می‌دهند تصویر ساده U-shaped همیشه کافی نیست. پس Bias-Variance یک لنز تحلیلی است، نه قانون مکانیکی برای هر معماری.

4.2 Regularization صریح و ضمنی

Regularization صریح تابع هدف را تغییر می‌دهد؛ مثلاً L2 وزن‌های بزرگ را جریمه می‌کند. L1 می‌تواند sparsity ایجاد کند. Dropout مسیرهای تصادفی را حذف می‌کند و نوعی ensemble تقریبی/نویز ساختاری ایجاد می‌کند. Early stopping نیز با جلوگیری از ادامه fit شدن جزئیات آموزش، نقش regularizer دارد.

Objective(theta) = R_hat(theta) + lambda * Omega(theta)

Regularization ضمنی از خود الگوریتم بهینه‌سازی یا معماری ناشی می‌شود. SGD، initialization، convolutional weight sharing و data augmentation همگی انتخاب‌های خاصی را در فضای عظیم راه‌حل‌ها ترجیح می‌دهند. این ترجیح‌ها inductive bias می‌سازند.

4.3 Cross-Validation

در k-fold cross-validation داده به k بخش تقسیم می‌شود و مدل k بار آموزش می‌بیند؛ هر بار یک fold برای validation کنار گذاشته می‌شود. این روش برای داده کم مفید است، ولی اگر داده گروهی یا زمانی باشد، random k-fold می‌تواند leakage ایجاد کند. برای داده بیمارمحور باید split در سطح بیمار انجام شود؛ برای داده زمانی باید ترتیب زمان رعایت شود.

4.4 **Calibration** با **Accuracy** فرق دارد

یک مدل می‌تواند accuracy بالا اما probabilities بدکالیبره داشته باشد. اگر در بین نمونه‌هایی که مدل confidence حدود 0.8 دارد فقط 0.55 واقعاً صحیح باشند، مدل overconfident است. Calibration در فصل UQ مهم خواهد شد، زیرا «احتمال خروجی softmax» به خودی خود معیار قابل اعتماد uncertainty نیست.

فصل 5 - مبانی Probabilistic Modeling

5.1 چرا مدل احتمالی؟

دنیای واقعی قطعی نیست. حتی اگر همه متغیرهای مهم را بدانیم، اندازه‌گیری‌ها نویز دارند؛ برخی متغیرها پنهان‌اند؛ و گاهی یک ورودی با چند خروجی محتمل سازگار است. مدل احتمالی به جای نگاشت سخت x→y، یک توزیع شرطی p(y|x) یا توزیع مشترک p(x,y) را مدل می‌کند. این تمایز برای تصمیم‌گیری، ریسک و uncertainty حیاتی است.

5.2 احتمال شرطی و قانون بیز

p(A|B) = p(A,B) / p(B)

p(theta|D) = p(D|theta) p(theta) / p(D)

در استنباط بیزی، prior باور پیش از داده درباره پارامتر theta است، likelihood سازگاری داده با theta را اندازه می‌گیرد و posterior باور به‌روزشده پس از مشاهده D است. evidence یا marginal likelihood نقش نرمال‌سازی دارد و در model comparison نیز مهم است.

5.3 Likelihood با Probability of Parameters فرق دارد

در مدل frequentist، theta ثابت ولی ناشناخته است و likelihood تابعی از theta با داده ثابت است. گفتن «احتمال theta برابر 0.7 است» در این چارچوب معنای بیزی ندارد. در Bayesian inference، theta خودش متغیر تصادفی با posterior distribution است. این تفاوت فلسفی روی UQ اثر مستقیم دارد.

5.4 Maximum Likelihood و MAP

theta_MLE = argmax_theta p(D|theta)

theta_MAP = argmax_theta p(D|theta) p(theta)

MLE فقط likelihood را بیشینه می‌کند. MAP prior را نیز وارد می‌کند. بسیاری از regularizerهای رایج را می‌توان به‌عنوان prior تفسیر کرد: L2 تقریباً با prior گاوسی روی وزن‌ها و L1 با prior لاپلاس مرتبط است. این تفسیر به ما نشان می‌دهد regularization صرفاً «ترفند جلوگیری از overfitting» نیست؛ می‌تواند بیان ترجیح پیشین باشد.

Posterior Predictive 5.5

p(y*|x*,D) = integral p(y*|x*,theta) p(theta|D) dtheta

در پیش‌بینی بیزی کامل، پارامتر را روی یک مقدار نقطه‌ای قفل نمی‌کنیم. خروجی را روی تمام مقادیر ممکن theta میانگین می‌گیریم، با وزنی که posterior به آن‌ها داده است. این میانگین‌گیری پارامتری یکی از سرچشمه‌های اصلی epistemic uncertainty است.

6.1 Generative در برابر Discriminative

مدل discriminative مستقیماً p(y|x) یا decision boundary را می‌آموزد. Logistic regression و بسیاری از classifierهای neural این دسته‌اند. مدل generative توزیع داده، مثلاً p(x,y) یا p(x|y)p(y)، را مدل می‌کند. Naive Bayes، Gaussian Mixture و بسیاری از مدل‌های latent-variable generative هستند. Generative بودن لزوماً به معنی تولید تصویر نیست؛ مفهوم بنیادی‌تر، مدل‌سازی توزیع مشترک یا مکانیزم تولید داده است.

6.2 Bayesian Networks و Conditional Independence

شبکه بیزی یک گراف جهت‌دار بدون دور است که وابستگی‌های شرطی میان متغیرها را نشان می‌دهد. مزیت اصلی آن factorization است: توزیع مشترک بزرگ به حاصل‌ضرب توزیع‌های شرطی محلی شکسته می‌شود. این ساختار برای causality، reasoning تحت فروض بیشتر، و ادغام دانش دامنه مفید است.

p(x_1,...,x_d) = product_i p(x_i | Parents(x_i))

6.3 Gaussian Mixture Models

GMM فرض می‌کند داده از ترکیب چند توزیع گاوسی آمده است. یک متغیر پنهان z مشخص می‌کند هر نمونه به کدام مؤلفه تعلق دارد. الگوریتم EM بین تخمین مسئولیت‌های احتمالی مؤلفه‌ها (E-step) و به‌روزرسانی پارامترها (M-step) رفت‌وبرگشت می‌کند. GMM نمونه خوبی برای فهم latent variables و soft assignment است.

6.4 Hidden Markov Models

HMM برای دنباله‌ها است. یک حالت پنهان z_t در طول زمان طبق transition probability تغییر می‌کند و هر حالت مشاهده x_t را تولید می‌کند. Markov assumption می‌گوید حالت آینده، با دانستن حالت فعلی، به گذشته دور وابسته نیست. HMMها هنوز برای فهم sequence likelihood، smoothing، filtering و state estimation بسیار آموزشی‌اند.

6.5 Gaussian Processes

Gaussian Process به جای قرار دادن prior روی پارامترهای یک تابع خاص، prior روی خود تابع‌ها تعریف می‌کند. برای هر مجموعه محدود از ورودی‌ها، مقادیر تابع توزیع گاوسی مشترک دارند. kernel تعیین می‌کند نقاط نزدیک یا مشابه چگونه هم‌بسته‌اند. GP به‌طور طبیعی mean و predictive variance می‌دهد و برای داده کم و مدل‌سازی uncertainty بسیار جذاب است، هرچند مقیاس‌پذیری کلاسیک آن محدودیت دارد.

$$f(x) \sim GP(m(x), k(x,x'))$$

نکته

Probabilistic model به معنی «مدل خوب uncertainty» نیست. اگر مدل آماری misspecified باشد، likelihood اشتباه باشد یا posterior approximation ضعیف باشد، probabilityهای خروجی می‌توانند گمراه‌کننده باشند.

فصل 7 - **Uncertainty**: دقیقاً چه چیزی نامطمئن است؟

7.1 Predictive Uncertainty

وقتی مدل برای y* توزیع پیش‌بینی می‌دهد، پراکندگی این توزیع predictive uncertainty است. این عدم قطعیت می‌تواند از دو منبع مفهومی ناشی شود: عدم قطعیت ذاتی داده و عدم قطعیت ناشی از ناآگاهی مدل. این تفکیک کامل و یکتا در همه مدل‌ها نیست، اما در بسیاری از چارچوب‌ها مفید است.

7.2 Aleatoric Uncertainty

Aleatoric uncertainty از تصادفی بودن یا ابهام ذاتی مشاهده می‌آید. مثال: یک تصویر پزشکی کم‌کیفیت که حتی متخصصان درباره مرز ضایعه اختلاف دارند. با جمع‌آوری داده بیشتر از همان نوع، این عدم قطعیت الزاماً از بین نمی‌رود. می‌توان آن را homoscedastic فرض کرد (تقریباً ثابت) یا heteroscedastic (وابسته به x). در رگرسیون heteroscedastic، شبکه می‌تواند هم mean و هم variance را پیش‌بینی کند.

y | x ~ Normal(mu_theta(x), sigma_theta(x)^2)

7.3 Epistemic Uncertainty

Epistemic uncertainty از ناآگاهی مدل درباره تابع یا پارامترها می‌آید؛ مثلاً ناحیه‌ای از فضای ورودی که داده آموزشی کمی دارد. در اصل با داده مناسب بیشتر باید کاهش یابد. Bayesian posterior over parameters، deep ensembles و برخی روش‌های تقریب بیزی برای نمایش این نوع uncertainty استفاده می‌شوند.

7.4 Model Uncertainty، Data Uncertainty و Distributional Uncertainty

در ادبیات نام‌گذاری یکدست نیست. برخی مقالات distributional uncertainty را برای ناشناخته بودن خود توزیع یا OOD بودن ورودی جدا می‌کنند. بنابراین همیشه باید تعریف نویسنده را بخوانید. استفاده از برچسب aleatoric/epistemic بدون مشخص‌کردن مدل و منبع variability ممکن است مبهم باشد.

7.5 Confidence با Uncertainty یکی نیست

بالاترین softmax probability اغلب «confidence» نامیده می‌شود، اما شبکه می‌تواند روی ورودی OOD نیز softmax بسیار تیز تولید کند. Softmax فقط scoreها را نرمال می‌کند؛ نمی‌داند آیا ورودی شبیه training distribution است یا نه. برای استفاده ایمن، OOD detection، calibration و ارزیابی uncertainty لازم‌اند.

8.1 Predictive Entropy

$$H[p(y|x)] = - \sum_c p_c \log p_c$$

Entropy میزان پراکندگی توزیع کلاس‌ها را می‌سنجد. اگر یک کلاس احتمال نزدیک 1 داشته باشد entropy کم است؛ اگر احتمال‌ها نزدیک یکنواخت باشند entropy زیاد است. اما entropy مجموع انواع uncertainty را مخلوط می‌کند و از calibration بد نیز تأثیر می‌گیرد.

8.2 Mutual Information در Bayesian Classification

در مدل‌های بیزی، می‌توان predictive entropy را به اجزایی مرتبط با expected data uncertainty و disagreement میان پارامترها تفکیک کرد. Mutual information بین prediction و model parameters اغلب به‌عنوان شاخص epistemic uncertainty استفاده می‌شود: اگر مدل‌های posterior روی جواب اختلاف داشته باشند MI بالا می‌رود.

8.3 Proper Scoring Rules

برای ارزیابی probabilistic predictions بهتر است از معیارهایی استفاده شود که مدل را به گزارش احتمال صادقانه تشویق می‌کنند. Negative Log-Likelihood و Brier Score proper scoring rule هستند. Accuracy فقط درست/غلط را می‌بیند و کیفیت probabilityها را نادیده می‌گیرد.

$$\text{Brier} = (1/N) * \sum_i \sum_c (p_{ic} - y_{ic})^2$$

8.4 Calibration و ECE

Expected Calibration Error بازه confidence را به binها تقسیم می‌کند و اختلاف میان average confidence و empirical accuracy را وزن‌دار میانگین می‌گیرد. ECE شهودی است اما به تعداد bin حساس است و تمام شکل خطای calibration را نمی‌بیند. Reliability diagram مکمل مفیدی است.

8.5 Selective Prediction و Abstention

یک سیستم ایمن مجبور نیست روی همه نمونه‌ها جواب بدهد. در selective classification، مدل می‌تواند روی نمونه‌های نامطمئن abstain کند یا آن‌ها را به انسان ارجاع دهد. سپس trade-off میان coverage و risk

بررسی می‌شود. اگر با کاهش risk، coverage روی نمونه‌های پذیرفته‌شده سریع افت کند، uncertainty ranking مفید بوده است.

$$\text{Coverage} = P(\text{model accepts } x)$$
$$\text{Selective Risk} = E[\text{Loss} \mid \text{accepted}]$$

8.6 Prediction Intervals

در رگرسیون، یک point estimate کافی نیست. interval باید محدوده محتمل y را ارائه دهد. دو ویژگی کلیدی coverage و width هستند: interval خیلی پهن coverage بالا ولی ارزش عملی کم دارد؛ interval باریک اگر coverage واقعی پایین باشد خطرناک است. روش‌های conformal prediction در ادامه، coverage توزیعی را تحت فروض exchangeability هدف می‌گیرند.

فصل 9 - روش‌های عملی Uncertainty Quantification

9.1 Deep Ensembles

در Deep Ensemble چند شبکه با initialization، ترتیب داده یا bootstrap متفاوت مستقل آموزش داده می‌شوند. میانگین probabilityها prediction نهایی است و disagreement میان اعضا سیگنالی از epistemic uncertainty می‌دهد. این روش ساده ولی بسیار رقابتی است. نقطه ضعف اصلی هزینه چندبرابری train و inference است. همچنین اگر همه مدل‌ها inductive bias مشترک و failure mode مشابه داشته باشند، ensemble ممکن است به‌طور جمعی اشتباه و مطمئن باشد.

p(y|x,D) approx (1/M) * sum_{m=1}^M p(y|x, theta_m)

9.2 Monte Carlo Dropout

در dropout، MC Dropout هنگام inference نیز فعال می‌ماند و چند forward pass انجام می‌شود. تغییر خروجی‌ها به‌عنوان تقریب توزیع predictive استفاده می‌شود. مزیت آن سادگی است، اما کیفیت تقریب بیزی به معماری، dropout rate و فرض‌های نظری وابسته است و همیشه به اندازه ensemble قوی نیست.

9.3 Bayesian Neural Networks

در BNN به جای وزن ثابت، روی وزن‌ها توزیع قرار می‌دهیم و posterior آن‌ها را با داده به‌روزرسانی می‌کنیم. چون posterior دقیق در شبکه عمیق دشوار است، روش‌هایی مانند MCMC، variational inference در مقیاس‌های کوچک‌تر و Laplace approximation استفاده می‌شوند. دشواری اصلی علاوه بر هزینه محاسباتی، حساسیت uncertainty به prior و approximation است.

9.4 Laplace Approximation

Laplace approximation posterior را نزدیک یک mode با توزیع گاوسی تقریب می‌زند. Hessian یا تقریب آن curvature تابع log posterior را تعیین می‌کند. این روش می‌تواند روی مدل از قبل آموزش‌دیده اجرا شود و نسبتاً ارزان‌تر از BNN کامل باشد، اما چندمودی بودن posterior را از دست می‌دهد.

Conformal Prediction 9.5

Conformal Prediction متفاوت است: لازم نیست کل posterior را مدل کنیم. از calibration set و یک nonconformity score استفاده می‌کنیم تا prediction set یا interval بسازیم. تحت exchangeability، می‌توان coverage marginal با سطح تقریباً alpha-1 تضمین کرد. این تضمین درباره «میانگین نمونه‌های آینده» است، نه الزاماً coverage شرطی دقیق برای هر x یا هر subgroup.

P(Y_new in C_alpha(X_new)) >= 1 - alpha

در classification، خروجی به جای یک کلاس می‌تواند مجموعه‌ای از کلاس‌ها باشد. اگر مدل نامطمئن است، مجموعه بزرگ‌تر می‌شود. در regression، interval تولید می‌شود. Conformal بسیار جذاب است چون model-agnostic است، اما distribution shift می‌تواند تضمین استاندارد را بشکند.

Quantile Regression 9.6

Quantile regression مستقیماً quantileهای شرطی را با pinball loss می‌آموزد. با پیش‌بینی quantile پایین و بالا می‌توان interval ساخت. این روش به‌خصوص برای heteroscedastic noise مفید است، ولی interval خام quantile regression تضمین finite-sample coverage ندارد مگر با calibration تکمیلی مانند conformalization.

روش	نوع سیگنال	مزیت	محدودیت
Deep Ensemble	اختلاف بین مدل‌ها	قوی و ساده	هزینه بالا
MC Dropout	تغییر بین forward passها	پیاده‌سازی آسان	تقریب محدود
BNN	posterior روی وزن‌ها	چارچوب احتمالی صریح	محاسبات و prior حساس
Conformal	prediction set/interval	coverage توزیعی با فرض exchangeability	تضمین شرطی کامل نیست؛ shift مشکل‌ساز است

روش	نوع سیگنال	مزیت	محدودیت
Quantile Regression	شرطی quantile	خوب برای noise ناهمسان	نیازمند calibration برای coverage معتبر

فصل 10 - Robustness: مقاومت در یادگیری ماشین دقیقاً چیست؟

10.1 تعریف عملی

Robustness یعنی عملکرد یا رفتار مطلوب مدل نسبت به یک خانواده مشخص از تغییرات ورودی، داده، محیط یا مدل به اندازه کافی پایدار بماند. عبارت «مدل robust است» بدون مشخص‌کردن نسبت به چه تهدید یا چه shift ناقص است. مدل ممکن است در برابر Gaussian noise مقاوم باشد اما در برابر adversarial perturbation شکننده؛ یا روی corruption تصویری خوب عمل کند ولی هنگام انتقال بین بیمارستان‌ها افت شدید داشته باشد.

10.2 Threat Model

تحلیل robustness باید threat model داشته باشد: چه چیزی می‌تواند تغییر کند؟ مهاجم وجود دارد یا تغییر طبیعی است؟ اندازه تغییر چگونه محدود می‌شود؟ آیا مهاجم به مدل دسترسی white-box دارد؟ هدف untargeted است یا targeted؟ بدون این تعریف، ادعای robust بودن قابل سنجش نیست.

10.3 انواع رایج Robustness

نوع	نمونه تغییر	سؤال اصلی
Noise Robustness	نویز سنسور، compression ،blur	آیا عملکرد تحت اغتشاش طبیعی حفظ می‌شود؟
Adversarial Robustness	perturbation هدفمند و کوچک	آیا مهاجم می‌تواند با تغییر محدود خروجی را عوض کند؟
Distribution Robustness	تغییر جمعیت/دامنه	آیا مدل تحت P_test متفاوت هنوز معتبر است؟
Missingness Robustness	ویژگی یا modality مفقود	آیا خرابی یا نبود داده مدیریت می‌شود؟

سؤال اصلی	نمونه تغییر	نوع
آیا تغییر اجرای مدل خروجی را مختل می‌کند؟	quantization، خطای محاسبه	Parameter/Hardware Robustness

Local vs Global Robustness 10.4

Local robustness رفتار مدل در همسایگی یک ورودی خاص را بررسی می‌کند؛ مثلاً همه perturbationهای با ||delta||_p <= epsilon باید کلاس را ثابت نگه دارند. Global robustness خواص گسترده‌تری روی دامنه بزرگی از ورودی‌ها می‌خواهد و اثبات آن بسیار دشوارتر است.

For all delta with ||delta||_p <= epsilon: f(x+delta) = f(x)

Robust Accuracy 10.5

در clean accuracy، adversarial evaluation کافی نیست. Robust accuracy درصد نمونه‌هایی است که تحت حمله تعریف‌شده همچنان درست طبقه‌بندی می‌شوند. اگر attack ضعیف باشد، robust accuracy مصنوعی بالا می‌رود؛ بنابراین ارزیابی باید attack قوی، parameter tuning مناسب و ترجیحاً چند حمله را شامل شود.

فصل 11 - Adversarial Robustness

11.1 Adversarial Example چیست؟

Adversarial example ورودی‌ای است که عمداً طوری تغییر کرده تا مدل اشتباه کند، در حالی که تغییر طبق معیار threat model کوچک یا مجاز است. «کوچک» معمولاً با Lp norm تعریف می‌شود، اما شباهت ادراکی انسان الزاماً با Lp norm هم‌ارز نیست. بنابراین انتخاب constraint بخشی از تعریف امنیتی است.

11.2 FGSM

Fast Gradient Sign Method از گرادیان loss نسبت به ورودی استفاده می‌کند و در جهت افزایش loss حرکت می‌کند. این روش یک حمله تک‌مرحله‌ای است و بیشتر برای شهود و آموزش اولیه مفید است.

$$x_adv = x + epsilon * sign(grad_x\ L(theta,x,y))$$

11.3 PGD

Projected Gradient Descent/Ascent در مسئله حمله چند مرحله گرادیانی انجام می‌دهد و بعد از هر مرحله perturbation را به مجموعه مجاز projection می‌کند. PGD معمولاً حمله بسیار قوی‌تری از FGSM است و در adversarial training استاندارد استفاده می‌شود.

$$delta_\{t+1\} = Proj_Delta(delta_t + alpha * sign(grad_delta\ L(theta,x+delta_t,y)))$$

11.4 Adversarial Training به‌عنوان Min-Max Optimization

$$min_theta\ E_\{(x,y)\}\ [\ max_\{delta\ in\ Delta\}\ L(theta,\ x+delta,\ y)\ ]$$

حلقه داخلی بدترین perturbation مجاز را پیدا می‌کند و حلقه خارجی پارامترها را طوری تغییر می‌دهد که loss بدترین حالت کاهش یابد. این یک نمونه از robust optimization است. هزینه آن آموزش سنگین‌تر و گاهی trade-off با clean accuracy است. همچنین robustness معمولاً به همان threat model آموزش وابسته می‌ماند.

11.5 Certified Robustness

Empirical attack نشان می‌دهد «این حمله نتوانست مدل را بشکند»، نه اینکه هیچ حمله‌ای وجود ندارد. Certified robustness تلاش می‌کند برای شعاع یا مجموعه مشخص اثبات کند خروجی تغییر نمی‌کند. روش‌ها

شامل convex relaxation، interval bound propagation و randomized smoothing هستند. certification معمولاً تضمین قوی‌تر ولی شعاع یا مقیاس محدودتری دارد.

11.6 Gradient Masking: امنیت جعلی

گاهی دفاعی باعث می‌شود گرادیان حمله خراب یا غیرقابل استفاده شود، بدون اینکه decision boundary واقعاً مقاوم شده باشد. حمله ساده شکست می‌خورد و پژوهشگر تصور می‌کند robustness بالا است. ارزیابی با attackهای gradient-free، adaptive و sanity checkها برای کشف این خطا ضروری است.

فصل 12 - Distribution Shift و OOD

12.1 Covariate Shift

در covariate shift، توزیع ورودی تغییر می‌کند ولی رابطه شرطی p(y|x) ثابت فرض می‌شود. مثال: فراوانی انواع بیماران بین دو بیمارستان فرق دارد، اما اگر ویژگی‌های کامل x را بدانیم رابطه بیماری با x یکسان است. در این حالت importance weighting بر اساس p_test(x)/p_train(x) می‌تواند از نظر نظری کمک کند، هرچند برآورد این نسبت دشوار است.

12.2 Label Shift

در label shift، p(y) تغییر می‌کند ولی p(x|y) ثابت فرض می‌شود. مثال: شیوع بیماری در جمعیت جدید تغییر می‌کند ولی الگوی تصویر مشروط به بیماری یکسان است. روش‌های correction می‌توانند prior classها را دوباره تنظیم کنند.

12.3 Concept Shift

در concept shift، خود p(y|x) تغییر می‌کند. این حالت سخت‌تر است، چون mapping هدف تغییر کرده است. مثال: تعریف برچسب، پروتکل تشخیص یا رفتار محیط در طول زمان عوض شود. صرف reweighting ورودی کافی نیست؛ مدل ممکن است نیازمند adaptation یا relearning باشد.

12.4 OOD Detection

نمونه Out-of-Distribution ورودی‌ای است که از توزیع عملیاتی مورد انتظار متفاوت است. OOD یک مفهوم نسبی است: باید بگوییم in-distribution چیست. تشخیص OOD می‌تواند با energy scores، ensemble disagreement، density/representation methods یا مدل‌های اختصاصی انجام شود. هیچ score واحدی برای تمام shiftها بهترین نیست.

12.5 Domain Adaptation و Domain Generalization

Domain Adaptation هنگام آموزش به داده‌ای از target domain دسترسی دارد (برچسب‌دار یا بدون برچسب). Domain Generalization می‌خواهد بدون دیدن target domain، از چند source domain الگویی بیاموزد که به domain جدید تعمیم یابد. این دو را نباید یکی دانست.

12.6 Distributionally Robust Optimization

DRO به جای کمینه‌کردن risk روی یک توزیع تجربی، بدترین risk را در یک مجموعه از توزیع‌های نزدیک یا plausible کمینه می‌کند. انتخاب ambiguity set تعیین‌کننده است. اگر مجموعه بیش از حد بزرگ باشد مدل محافظه‌کار می‌شود؛ اگر بد تعریف شود، robustness مورد نیاز را پوشش نمی‌دهد.

min_theta sup_{Q in U(P_hat)} E_Q[L(theta;X,Y)]

فصل 13 - Resilience: تاب‌آوری فراتر از مقاومت

13.1 تعریف سیستم‌محور

Resilience در این جزوه یعنی توانایی یک سامانه یادگیری برای ادامه خدمت قابل قبول یا بازیابی آن پس از degradation، disturbance یا failure. تفاوت اصلی با robustness این است که robustness غالباً می‌پرسد «آیا عملکرد در حضور اختلال حفظ می‌شود؟» اما resilience می‌پرسد «اگر اختلال از ظرفیت مقاومت عبور کرد و failure رخ داد، آیا سامانه آن را تشخیص می‌دهد، پاسخ می‌دهد، یاد می‌گیرد و بازیابی می‌شود؟»

13.2 چرخه Detect - Diagnose - Adapt - Recover

1. Detect: کشف کنیم که داده، عملکرد یا uncertainty از محدوده طبیعی خارج شده است.
2. Diagnose: مشخص کنیم نوع شکست چیست؛ sensor failure، distribution shift، label drift، attack یا model degradation.
3. Adapt: اقدام اصلاحی انجام دهیم؛ تغییر threshold، retraining، online update، domain adaptation یا انتخاب مدل جایگزین.
4. Recover: به سطح مشخصی از performance یا service برگردیم و پایداری پس از بازیابی را بسنجیم.
5. Learn: اطلاعات failure را در monitoring rule، replay buffer، memory یا data pipeline نگه داریم تا شکست مشابه بهتر مدیریت شود.

13.3 Robustness می‌تواند بخشی از Resilience باشد

یک سیستم resilient بهتر است ابتدا تا حدی robust باشد، زیرا نمی‌خواهیم با هر perturbation کوچک وارد حالت بحران شود. اما robustness کامل ناممکن یا بسیار پرهزینه است. resilience یک لایه بعدی فراهم می‌کند: fallback، abstain، degrade gracefully و recovery. بنابراین می‌توان resilience را یک property چرخه عمر سامانه دانست نه فقط یک property نقطه‌ای classifier.

13.4 Graceful Degradation

شکست صفر و یک نیست. سامانه خوب ممکن است هنگام خراب‌شدن یک modality به جای سقوط کامل، به مدل تک‌مودال ضعیف‌تر سوییچ کند؛ یا هنگام uncertainty بالا به جای تصمیم خودکار، پرونده را به انسان ارجاع دهد. این کاهش کنترل‌شده عملکرد، graceful degradation نامیده می‌شود و جزء مهم تاب‌آوری عملی است.

13.5 Fallback و Safe State

در کاربردهای ایمنی‌حساس، recovery همیشه به معنی «یادگیری فوری» نیست. گاهی امن‌ترین پاسخ توقف adaptation، برگشت به نسخه verified، فعال‌سازی rule-based fallback یا درخواست دخالت انسان است. Online learning بدون guardrail می‌تواند failure را تشدید کند.

فصل 14 - Concept Drift، Online Learning و Continual Adaptation

14.1 Batch vs Online Learning

در batch learning فرض می‌کنیم مجموعه داده تقریباً ثابت است و مدل دوره‌ای آموزش می‌بیند. در online learning نمونه‌ها یا batchهای کوچک به‌ترتیب زمان می‌آیند و مدل به‌روزرسانی می‌شود. این setting برای resilience مناسب است، اما خطرات جدیدی دارد: catastrophic forgetting، feedback loop، drift و poisoning.

14.2 Regret

در نظریه online learning، regret عملکرد تجمعی الگوریتم را با بهترین تصمیم ثابت در hindsight مقایسه می‌کند. sublinear regret یعنی متوسط regret در طول زمان به صفر میل می‌کند. Regret تضمین دقیقاً معادل resilience نیست، اما زبان ریاضی قدرتمندی برای سازگاری متوالی است.

```
Regret_T = sum_{t=1}^T L_t(theta_t) - min_theta sum_{t=1}^T L_t(theta)
```

14.3 Concept Drift Detection

Drift detectorها می‌توانند روی feature statistics، residual، error rate یا representation distribution نظارت کنند. detector خوب باید تعادل false alarm و detection delay را مدیریت کند. اگر بیش از حد حساس باشد دائماً retrain می‌کند؛ اگر کم‌حساس باشد degradation دیر کشف می‌شود.

14.4 Continual Learning و Catastrophic Forgetting

در continual learning، مدل وظایف یا توزیع‌های جدید را به‌تدریج یاد می‌گیرد. مشکل معروف catastrophic forgetting است: بهبود روی داده جدید باعث از دست رفتن عملکرد روی گذشته می‌شود. راهکارها شامل dynamic architectures، regularization-based methods، replay و parameter isolation هستند. سامانه resilient باید نه فقط «یاد بگیرد» بلکه بررسی کند adaptation چه چیزی را خراب کرده است.

14.5 Feedback Loops

در سیستم deployed، prediction می‌تواند داده آینده را تغییر دهد. مثلاً مدل پیشنهاددهنده چیزی را نمایش می‌دهد و سپس رفتار کاربر به همان نمایش وابسته می‌شود. اگر مدل بر اساس outcomeهای حاصل از تصمیم خودش retrain شود، distribution به‌صورت endogeneous تغییر می‌کند. monitoring ساده train-vs-test دیگر کافی نیست.

فصل 15 - چگونه Resilience را اندازه بگیریم؟

15.1 فقط Accuracy قبل و بعد کافی نیست

تاب‌آوری یک property زمانی است. دو سیستم ممکن است پس از 24 ساعت به accuracy یکسان برسند، اما یکی در پنج دقیقه failure را تشخیص داده و دیگری ده ساعت خروجی خطرناک داده باشد. بنابراین باید trajectory عملکرد را در طول disturbance و recovery اندازه‌گیری کرد.

معیار	تعریف مفهومی	چرا مهم است
Detection Delay	فاصله شروع اختلال تا کشف	failure چقدر بدون کنترل ادامه می‌یابد؟
Performance Drop	افت بیشینه یا متوسط	شدت آسیب اولیه چیست؟
Time to Recovery	زمان برگشت به سطح هدف	سامانه چقدر سریع بازیابی می‌شود؟
Recovery Level	سطح عملکرد پس از بازیابی	آیا واقعاً به baseline برگشته؟
Area of Degradation	مساحت افت عملکرد در زمان	شدت × مدت failure را یکجا می‌سنجد
False Alarm Rate	هشدار بدون failure	هزینه monitoring بی‌دلیل
Adaptation Stability	نوسان پس از update	آیا recovery پایدار است یا ناپایدار؟

15.2 Resilience Curve

یک نمودار مفید، performance یا utility در برابر زمان است. قبل از disturbance یک baseline داریم؛ سپس افت رخ می‌دهد؛ detection و adaptation شروع می‌شوند؛ و سامانه به سطحی بازیابی می‌شود. مساحت زیر baseline در دوره بحران می‌تواند یک معیار یکپارچه خسارت باشد. این نوع تعریف از ادبیات resilient systems می‌آید و برای ML نیز قابل اقتباس است.

15.3 Quality of Service و Constraintها

در کاربردهای واقعی، هدف صرفاً بیشینه‌کردن accuracy نیست. ممکن است constraint داشته باشیم: latency زیر 100 ms، false negative کمتر از مقدار معین، coverage حداقل مشخص یا safety violation تقریباً صفر. resilience باید نسبت به این service-level objectiveها تعریف شود. اگر accuracy بازیابی شود ولی latency یا calibration خراب بماند، سیستم لزوماً resilient نیست.

15.4 Formal Guarantee در برابر Empirical Evidence

Empirical resilience یعنی در سناریوهای آزمایشی تعریف‌شده recovery مشاهده شده است. Formal guarantee نیازمند تعریف ریاضی assumptions، property و اثبات/verification است. هیچ تضمینی خارج از فرض‌هایش معنا ندارد. مثلاً تضمین برای perturbation L2 با شعاع مشخص درباره concept drift آزاد چیزی نمی‌گوید.

فصل 16 - تفاوت دقیق Safety، Reliability، Resilience، Robustness و Trustworthiness

اصطلاح	سؤال مرکزی	نمونه
Robustness	آیا در برابر تغییر تعریف‌شده عملکرد حفظ می‌شود؟	کلاس تصویر با نویز/حمله تغییر نکند
Resilience	اگر اختلال باعث degradation شد، آیا detect/adapt/recover می‌کنیم؟	پس از drift مدل هشدار دهد و با کنترل بازیابی شود
Reliability	آیا سیستم در طول زمان به‌طور پایدار وظیفه را درست انجام می‌دهد؟	نرخ failure پایین در عملیات طولانی
Safety	آیا ریسک آسیب غیرقابل قبول کنترل شده است؟	جلوگیری از تصمیم خودکار هنگام uncertainty بالا
Security	آیا در برابر عامل مخرب و تهدید امنیتی محافظت شده است؟	مقاومت در برابر poisoning یا model extraction
Trustworthiness	مفهوم چتری شامل قابلیت اعتماد، ایمنی، شفافیت، privacy، fairness و ...	ارزیابی جامع سامانه در زمینه استفاده

16.1 Robustness بدون Resilience

فرض کنید classifier تا 20% blur تقریباً بدون افت کار می‌کند، اما وقتی sensor calibration عوض شود ناگهان accuracy از 95 به 50 درصد می‌رسد و هیچ monitoring ندارد. این مدل نسبت به blur robust است ولی سامانه نسبت به تغییر سنسور resilient نیست.

16.2 Resilience بدون Robustness کافی

سامانه‌ای را تصور کنید که با کوچک‌ترین نویز performance افت می‌کند ولی سریع detector آن را می‌فهمد و fallback فعال می‌کند. این سامانه ممکن است تا حدی resilient باشد، اما هزینه عملیاتی بالایی دارد چون robustness اولیه ضعیف است. طراحی خوب معمولاً هر دو را ترکیب می‌کند.

16.3 UQ چگونه به هر دو کمک می‌کند؟

Uncertainty می‌تواند نقش sensor داخلی برای failure داشته باشد. اگر epistemic uncertainty روی ناحیه جدید بالا رود، سامانه می‌تواند abstain یا adaptation را فعال کند. اما UQ به خودی خود robustness ایجاد نمی‌کند: مدلی که uncertainty خود را درست می‌داند هنوز ممکن است accuracy پایین داشته باشد. همچنین uncertainty score اگر under shift کالیبره نباشد detector قابل اعتمادی نیست.

فرمول ذهنی مفید

Robustness = مقاومت قبل از شکست؛ Resilience = مدیریت و بازیابی پس از عبور از ظرفیت مقاومت. UQ = دانستن/اندازه‌گیری میزان اطمینان؛ Learning Theory = چارچوب فهم تعمیم؛ Probabilistic Modeling = زبان نمایش تصادفی بودن و باور.

فصل 17 - Case Study 1: سامانه تشخیص پزشکی تصویری

17.1 سناریو

فرض کنید یک مدل segmentation برای MRI روی داده بیمارستان A آموزش دیده است. در deployment، بیمارستان B از scanner و protocol متفاوت استفاده می‌کند. برخی تصاویر motion artifact دارند و گاهی توالی تصویربرداری ناقص است. هدف فقط Dice بالا روی test داخلی نیست؛ باید رفتار تحت این تغییرها فهمیده شود.

17.2 از دید Statistical Learning Theory

اگر train و test داخلی i.i.d باشند، generalization estimate فقط همان distribution را پوشش می‌دهد. group split باید در سطح بیمار باشد تا slice leakage رخ ندهد. ظرفیت مدل، regularization و sample size روی in-domain generalization اثر دارند.

17.3 از دید Probabilistic Modeling و UQ

مدل می‌تواند برای هر voxel probability بدهد، اما probability خام الزاماً calibrated نیست. ensemble یا MC dropout می‌تواند uncertainty map بسازد. نواحی با ambiguity آناتومیک aleatory uncertainty بالا دارند؛ anatomy یا scanner ناشناخته می‌تواند epistemic uncertainty را افزایش دهد. calibration باید هم in-domain و هم external domain بررسی شود.

17.4 از دید Robustness

corruption benchmark می‌تواند intensity shift، blur، noise و motion artifact را شبیه‌سازی کند. Domain robustness با external test روی بیمارستان B بررسی می‌شود. اگر threat adversarial مطرح باشد، باید attack مشخص تعریف شود؛ adversarial robustness با scanner shift یک مسئله واحد نیست.

17.5 از دید Resilience

سیستم deployed باید drift در scanner statistics یا uncertainty را تشخیص دهد. اگر confidence معتبر پایین آمد، می‌تواند case را برای review انسانی علامت بزند. در drift پایدار، adaptation با داده تأییدشده انجام شود. پس از update باید regression test روی hospital A انجام شود تا catastrophic forgetting یا افت subgroup رخ نداده باشد. rollback باید امکان‌پذیر باشد.

17.6 شاخص‌های ارزیابی پیشنهادی

- Dice/IoU و surface metrics برای segmentation accuracy.
- NLL/Brier و reliability curve برای calibration.
- Risk-Coverage curve برای ارجاع موارد نامطمئن.
- Performance under corruption و external domain.
- Drift detection delay و false alarm rate.
- Recovery time و post-adaptation regression performance.

فصل 18 - **Case Study 2**: بینایی ماشین در محیط پویا

یک detector عابر پیاده روی داده روز و هوای خشک آموزش داده شده است. در عمل شب، باران، مه و دوربین‌های مختلف ظاهر می‌شوند. گاهی لنز کثیف می‌شود و frameها drop می‌شوند. مهاجم نیز ممکن است علامت یا الگوی فیزیکی ایجاد کند.

18.1 تفکیک **failure mode**ها

Failure mode	طبقه مفهومی	پاسخ مناسب
مه و باران	Natural corruption / shift	augmentation، robust training، domain testing
دوربین جدید	Domain shift	calibration، adaptation، external validation
frame drop	System/data availability failure	temporal fallback، sensor fusion
patch مخرب	Adversarial/security	threat-model specific defense
کلاس شیء ناشناخته	OOD	detection/abstention/open-set handling

18.2 چرا یک عدد **accuracy** گمراه‌کننده است

ممکن است میانگین accuracy روی dataset خوب باشد ولی خطاها دقیقاً در شرایط بحرانی متراکم شوند. بنابراین evaluation باید condition-stratified باشد: روز/شب، باران/خشک، فاصله، occlusion، نوع دوربین. Robustness profile مهم‌تر از average score است.

18.3 **Resilient pipeline**

1. Monitoring کیفیت تصویر و سلامت sensor.
2. OOD/uncertainty gating برای شرایط ناشناخته.
3. Sensor fusion و fallback در خرابی camera.
4. Safe operational envelope: خارج از شرایط تعریف‌شده، کاهش سرعت/انتقال کنترل.

5. Logging failure cases برای data curation.

6. Offline retraining + validation + staged deployment به جای update کورکورانه.

«**Softmax 0.99** یعنی مدل 99% مطمئن است.»

نه لزوماً. softmax score می‌تواند miscalibrated باشد و روی OOD نیز بالا باشد. برای تعبیر احتمالی باید calibration و validity در دامنه هدف بررسی شود.

«اگر **test accuracy** بالا است، مدل **robust** است.»

خیر. test معمولاً یک توزیع مشخص را می‌سنجد. robustness نیازمند perturbation/shift مشخص و ارزیابی تحت آن است.

«**Resilience** همان **robustness** است.»

خیر. robustness معمولاً جلوگیری/تحمل degradation است؛ resilience علاوه بر آن detection، adaptation و recovery پس از failure را در بر می‌گیرد.

«**Bayesian** بودن یعنی **uncertainty** درست است.»

خیر. likelihood، prior، مدل و posterior approximation می‌توانند نامناسب باشند. Bayesian framework زبان منسجم می‌دهد، نه مصونیت از misspecification.

«**Aleatoric** همیشه ثابت و **Epistemic** همیشه با داده بیشتر صفر می‌شود.»

این ساده‌سازی است. نوع uncertainty به مدل، مشاهده و تعریف مسئله وابسته است. داده بیشتر فقط اگر informative و از ناحیه مناسب باشد epistemic uncertainty را کاهش می‌دهد.

«**Conformal prediction** به هر نمونه **coverage** شرطی تضمین می‌دهد.»

نسخه استاندارد معمولاً marginal coverage تحت exchangeability می‌دهد، نه coverage شرطی دقیق برای هر x یا subgroup.

«**Adversarial robustness** یعنی **robustness** در دنیای واقعی.»

فقط نسبت به threat model مشخص. مقاومت L_inf کوچک لزوماً به domain shift، weather، blur یا حمله فیزیکی تعمیم نمی‌یابد.

«**Regularization** فقط برای کاهش **overfitting** است.»

می‌تواند بیان smoothness، sparsity، prior، inductive bias یا constraint عملی باشد. اثر آن گسترده‌تر از train-test gap است.

20.1 مرحله 1: **Operational Distribution** را تعریف کنید

باید مشخص شود سیستم کجا، برای چه جمعیتی، با چه sensor/protocol و تحت چه محدوده‌ای استفاده می‌شود. بدون این تعریف، OOD، in-distribution و shift مبهم‌اند.

20.2 مرحله 2: **Failure Taxonomy** بسازید

failure modeها را فهرست کنید: rare، adversarial input، domain shift، missing data، noise، hardware error، label change، class و human interaction. هر failure mode ابزار دفاع و معیار ارزیابی متفاوتی دارد.

20.3 مرحله 3: **Baseline + Calibration**

قبل از روش پیچیده، baseline ساده و reproducible بسازید. accuracy/utility و calibration را جدا گزارش کنید. confidence threshold را روی validation تنظیم کنید، نه test.

20.4 مرحله 4: **Stress Testing**

stress test باید severity ladder داشته باشد: تغییر ملایم تا شدید. فقط یک شدت، شکل منحنی degradation را نشان نمی‌دهد. برای shiftهای واقعی external dataset ارزش بیشتری از corruption مصنوعی دارد؛ هر دو می‌توانند مکمل باشند.

20.5 مرحله 5: **UQ** و **Abstention**

یک uncertainty method انتخاب و با calibration، proper scoring rules و risk-coverage ارزیابی کنید. صرفاً histogram uncertainty کافی نیست. بررسی کنید آیا نمونه‌های خطادار واقعاً uncertainty بالاتری دارند.

20.6 مرحله 6: **Recovery Mechanism**

مشخص کنید بعد از detector چه اتفاقی می‌افتد. retraining، reweighting، model selection، human escalation یا rollback؟ detector بدون action policy فقط هشداردهنده است، نه resilience کامل mechanism.

20.7 مرحله 7: **Post-Adaptation Verification**

بعد از adaptation باید هم target improvement و هم collateral damage سنجیده شود: latency، calibration drift، fairness drift، forgetting و safety constraints. نسخه قبلی باید برای rollback حفظ شود.

20.8 مرحله 8: گزارش عدم قطعیت ادعاها

در مقاله یا گزارش، دقیق بگویید «robust نسبت به چه چیزی»، «coverage تحت چه فرضی»، «external validation روی چه دامنه‌ای» و «resilience با چه failure scenario و recovery criterion». واژه‌های بزرگ بدون operational definition ارزش علمی کمی دارند.

پیوست A - واژگان و فرمول‌های کلیدی

نماد/اصطلاح	معنا
X	فضای ورودی
Y	فضای خروجی/برچسب
P(X,Y)	توزیع ناشناخته داده
S	نمونه آموزشی محدود
H	فضای فرضیه/مدل
L	تابع زیان
R(f)	ریسک واقعی/جمعیتی
R_hat(f)	ریسک تجربی
epsilon	دقت/شعاع خطا بسته به زمینه
delta	احتمال شکست تضمین در PAC/bounds
theta	پارامترهای مدل
p(theta|D)	posterior
p(y|x,D)	posterior predictive

A.1 Empirical Risk و Regularized Risk

R_hat(theta) = (1/n) sum_i L(f_theta(x_i), y_i)

J(theta) = R_hat(theta) + lambda Omega(theta)

A.2 Bayesian Update

Posterior = Likelihood * Prior / Evidence

A.3 Predictive Entropy

H(p) = - sum_c p_c log p_c

A.4 Adversarial Objective

```
min_theta E[max_{delta in Delta} L(theta,x+delta,y)]
```

A.5 Online Regret

```
Regret_T = cumulative loss of learner - loss of best comparator
```

A.6 Conformal Coverage

```
P(Y_new in C_alpha(X_new)) >= 1-alpha  (under exchangeability)
```

پیوست B - فرهنگ اصطلاحات کوتاه

Generalization

عملکرد روی داده ندیده از توزیع مورد نظر.

Overfitting

fit شدن بیش از حد به ویژگی‌های خاص نمونه آموزش.

Inductive Bias

فرض/ترجیحی که الگوریتم برای انتخاب میان راه‌حل‌ها دارد.

Calibration

همخوانی confidence گزارش‌شده با فراوانی واقعی صحت.

OOD

ورودی خارج از توزیع عملیاتی تعریف‌شده.

Distribution Shift

تفاوت آماری میان محیط آموزش و استفاده.

Aleatoric Uncertainty

عدم قطعیت مرتبط با تصادفی بودن/ابهام داده.

Epistemic Uncertainty

عدم قطعیت ناشی از ناآگاهی مدل.

Robustness

پایداری رفتار در برابر تغییر تعریف‌شده.

Resilience

تشخیص، مدیریت و بازیابی پس از اختلال/شکست.

Abstention

خودداری مدل از تصمیم در موارد نامطمئن.

Drift

تغییر تدریجی یا ناگهانی توزیع در طول زمان.

DRO

بهینه‌سازی برای بدترین توزیع در مجموعه‌ای از توزیع‌های plausible.

Certified Robustness

تضمین ریاضی عدم تغییر رفتار در محدوده تعریف‌شده.

Proper Scoring Rule

معیار احتمالی که گزارش probability صادقانه را بهینه می‌کند.

منابع پیشنهادی برای مطالعه عمیق‌تر

[1] V. N. Vapnik. Statistical Learning Theory. Wiley, 1998.

[2] S. Shalev-Shwartz and S. Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.

[3] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[4] K. P. Murphy. Probabilistic Machine Learning: An Introduction. MIT Press, 2022.

[5] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.

[6] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. ICLR, 2015.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. ICLR, 2018.

[8] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. ICML, 2016.

[9] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. NeurIPS, 2017.

[10] D. Hendrycks and T. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. ICLR, 2019.

[11] Y. Ovadia et al. Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. NeurIPS, 2019.

[12] A. N. Angelopoulos and S. Bates. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. Foundations and Trends in Machine Learning, 2023.

[13] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. ICML, 2017.

جمع‌بندی نهایی

اگر بخواهیم کل جزوه را در پنج جمله فشرده کنیم: Statistical Learning Theory توضیح می‌دهد چرا minimization روی نمونه محدود گاهی به عملکرد خوب روی جمعیت منجر می‌شود و ظرفیت مدل چه نقشی دارد. Probabilistic Modeling عدم قطعیت را در زبان توزیع، prior، likelihood و posterior بیان می‌کند. Uncertainty Quantification تلاش می‌کند میزان اطمینان پیش‌بینی را به‌صورت قابل ارزیابی گزارش کند. Robustness می‌سنجد مدل در برابر خانواده تعریف‌شده‌ای از تغییرات تا چه حد عملکرد خود را حفظ می‌کند. Resilience از سطح مدل فراتر می‌رود و چرخه failure detection، adaptation، graceful degradation، fallback و recovery را در طول زمان بررسی می‌کند.