

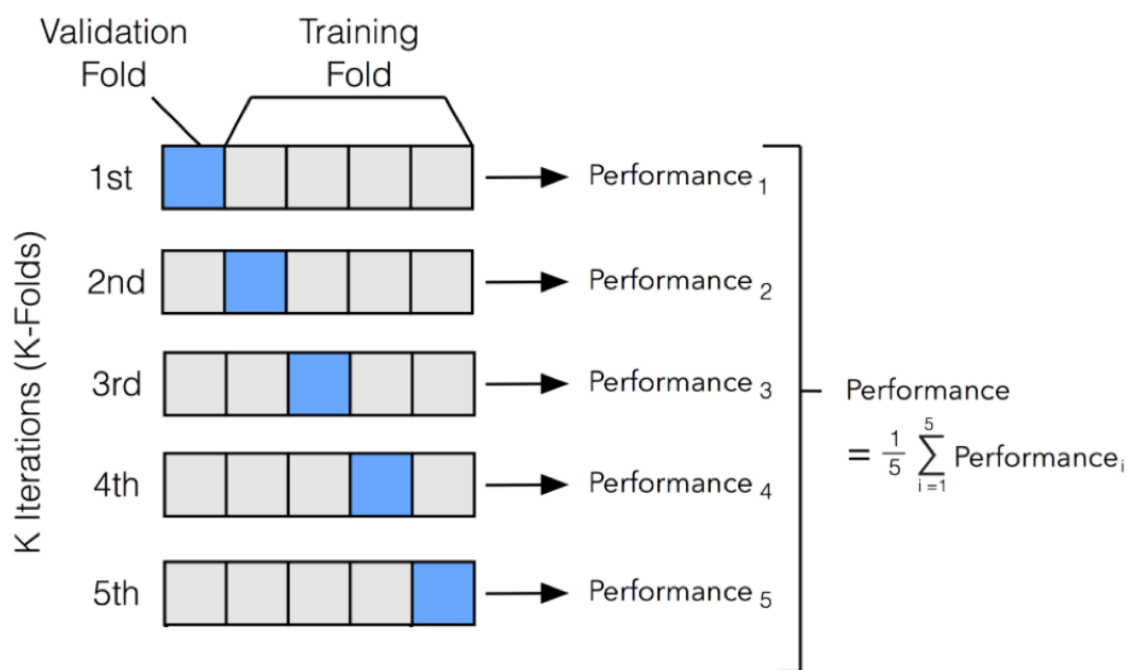


## سؤال ۱.

• (آ)

— Cross Validation: یک روش نمونه‌گیری مجدد برای ارزیابی مدل‌های یادگیری ماشین با داده‌های نمونه‌ای محدود است. در این روش یک پارامتر به  $k$  وجود دارد که تعداد دسته‌هایی که داده باید به آن تقسیم شود را تعیین می‌کند.<sup>۱</sup> به دلیل قابلیت درک ساده آن، یک روش محبوب است. نحوه کلی کارکرد آن به صورت زیر است:

۱. مجموعه‌ی داده‌ها را به‌طور تصادفی درهم می‌کنیم.
۲. مجموعه داده‌ها را به  $k$  بخش تقسیم می‌کنیم.
۳. برای هر گروه عملیات زیر را انجام می‌دهیم: آن گروه را به عنوان مجموعه تست در نظر می‌گیریم. باقی گروه‌ها را به برای داده‌های یادگیری استفاده می‌کنیم. یک مدل را بر روی داده‌های یادگیری تنظیم کرده و سپس آن را روی داده‌ها تست، ارزیابی می‌کنیم. نمره ارزیابی را در جایی ذخیره می‌کنیم و مدل را دور می‌ریزیم.
۴. قدرت مدل را بر اساس امتیازهایی که بر روی داده‌های نمونه به دست آورده خلاصه کرده و گزارش می‌کنیم.



شکل ۱: نحوه عمل‌کرد Cross Validation

<sup>۱</sup> به همین خاطر به آن k-fold cross validation نیز می‌گویند.

– Data Augmentation: به فرآیند افزایش میزان و تنوع داده‌ها می‌گویند. در این روش داده‌ی جدیدی جمع‌آوری نمی‌شود؛ بلکه داده‌های موجود را با عملیات‌هایی تبدیل به داده‌ی جدید می‌کنیم. علت این کار این است که معمولا به مجموعه داده‌های بزرگی (مخصوصا در شبکه‌های عصبی) نیاز داریم و جمع‌آوری داده‌های بزرگ در بسیاری از موارد برای ما امکان‌پذیر نمی‌باشد. عملیات‌های رایجی<sup>۲</sup> که معمولا در این روش استفاده می‌شوند، عبارتند از:

Rotation \*  
Shearing \*  
Zooming \*  
Cropping \*  
Flipping \*  
... و \*

– Data Vanishing: این مشکل وابسته به انتخاب تابع‌های activation است. بسیاری از تابع‌های رایج activation (مانند تانژانت و سیگموید) ورودی خود را به‌طور غیرخطی به محدوده خروجی بسیار کوچکی تبدیل می‌کنند. برای مثال اگر تابع سیگموید، اعداد حقیقی را روی بازه‌ی بسیار کوچک  $[0, 1]$  نگاشت کند، این اتفاق خواهد افتاد. در نتیجه تابع در بسیاری از موارد flat است و یک محدوده‌ی بسیار بزرگ (اعداد حقیقی) به یک بازه‌ی خیلی خیلی کوچکی نگاشت شده‌اند و تغییرات بسیار کوچکی دارند، بنابراین گرادینت بسیار کوچک خواهد بود.<sup>۳</sup> یکی از راه‌حل‌های گریز از این مشکل استفاده از تابع‌های activation دیگر نظیر ReLU است که مشتق‌های کوچک ندارد.

– Dropout: شبکه‌های عصبی عمیق با تعداد زیادی از پارامترها، در سیستم‌های یادگیری ماشین بسیار قدرت‌مند هستند. با این حال overfitting یک مشکل بسیار جدی در این شبکه‌ها است. Dropout یک تکنیک برای حل کردن این مشکل است. ایده‌ی اصلی آن این است که به‌طور تصادفی در طول آموزش unit‌ها و روابطشان را حذف کنیم. این کار باعث می‌شود که unit‌ها خودشان را بسیار تطبیق ندهند تا باعث overfitting شوند).

• (ب) این کار خطاست. زیرا عموما داده‌ی تست یک بخش از داده‌ای است که می‌خواهیم برای بررسی مدل نهایی و ارزیابی عمل‌کرد، از آن استفاده کنیم. اگر از این داده‌ها برای بهبود و انتخاب hyperparameterها استفاده کنیم، این‌گونه به مدل شانس دیدن داده‌های تست را می‌دهیم و یک bias با توجه به داده‌های تست به‌وجود می‌آید؛<sup>۴</sup> بنابراین امکان ارزیابی تابع به دلیل دیدن داده‌هایی که نباید دیده می‌شدند، از دست می‌رود.

ساده‌ترین راه برای رفع این خطا تقسیم داده به سه بخش یادگیری، اعتبارسنجی<sup>۵</sup> و تست است. البته روش‌های دیگر و بهتری نظیر Cross Validation که در قسمت اول این سوال توضیح دادیم وجود دارد.

• (ج)

<sup>۲</sup> این عملیات، عملیات رایج روی داده‌های تصویری هستند.

<sup>۳</sup> اگر این اتفاق در لایه‌های اولیه اتفاق بیفتند، وضعیت بسیار بدتر می‌شود.

<sup>۴</sup> این کار لزوما باعث overfit نمی‌شود؛ در حالی که مجموعه داده‌های ما بسیار بزرگ باشد.  
<sup>۵</sup> validation