دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

درس هوش مصنوعی

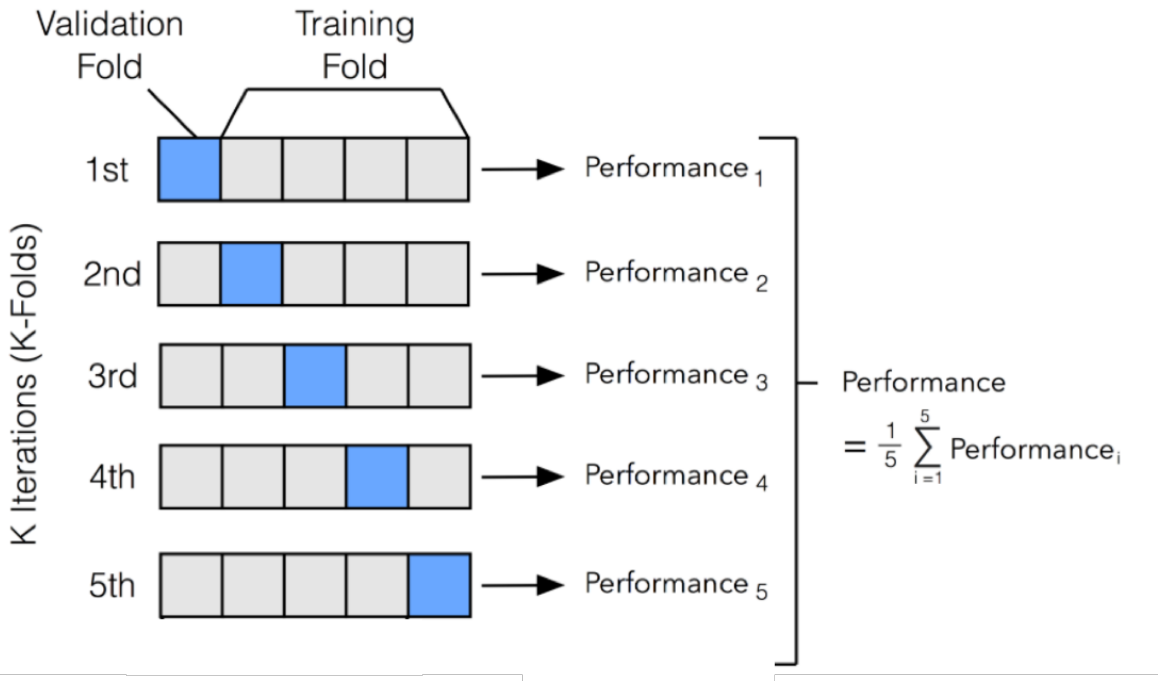
استاد محمدحسین رهبان

سؤال ۱. مفاهیم یادگیری ماشین

- آ)

– Cross Validation: یک روش نمونه‌گیری مجدد برای ارزیابی مدل‌های یادگیری ماشین با داده‌های نمونه‌ی محدود است. در این روش یک پارامتر به k وجود دارد که تعداد دسته‌هایی که داده باید به آن تقسیم شود را تعیین می‌کند.[۱] به دلیل قابلیت درک ساده آن، یک روش محبوب است. نحوه‌ی 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 داده‌هیا 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 دست‌آورده خلاصه کرده و گزارش می‌کنیم.



شکل ۱: نحوه عمل‌کرد Cross Validation

[۱] به همین‌خاطر 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 مجموعه‌داده‌های بزرگی (مخصوصا در شبکه‌های عصبی) نیاز داریم و جمع‌آوری داده‌های بزرگ در بسیاری از موارد برای ما امکان‌پذیر نمی‌باشد. عملیات‌های رایجی [۲] 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 کوچک نگاشت شده‌اند و تغییرات بسیار کوچکی دارند، بنابراین گرادیان بسیار کوچک خواهد بود.[۳] 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؛ [۴] 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[۵] و تست است. البته روش‌های دیگر و بهتری نظیر Cross Validation که در قسمت اول این سوال توضیح دادیم وجود دارد.

- ج) از لحاظ علمی معتبر نیست؛ زیرا اصلا نباید از داده‌های تست جز برای ارزیابی عمل‌کرد مدل آن‌هم تنها برای یک‌بار استفاده کرد و مدل باید طوری طراحی شود که برای هر مجموعه داده‌ی تستی به‌طور قابل قبول کار کند[۶].

---

[۲] این عملیات، عملیات رایج روی داده‌های تصویری هستند.
[۳] اگر این اتفاق در لایه‌های اولیه اتفاق بیفتند، وضعیت بسیار بدتر می‌شود.
[۴] این کار لزوما باعث overfit نمی‌شود؛ در حالتی که مجموعه داده‌های ما بسیار بزرگ باشد.
[۵] validation
[۶] یعنی قابلیت تعمیم یا Generalization را داشته باشد.

سؤال ۲. گرادیان کاهشی

- آ) برای به دست آوردن مقدار بهینه‌ی w باید از رابطه‌ی داده شده به نسبت w مشتق بگیریم.

$$J(w) = \Sigma_{i=1}^{n}(y^{(i)} - w^T x^{(i)})^2$$
$$\implies (Y - X^T w)^T (Y - X^T w)$$
$$\implies Y^T T + W^T X X^T W - 2Y^T X^T W$$

$$\frac{dJ}{dw} = 0 \implies w = (XX^T)^{-1} XY \qquad (۱)$$

همان‌طور که در رابطه (۱) می‌بینیم، برای این که این رابطه، تعریف‌شده باشد، باید عبارت XX^T معکوس‌پذیر باشد.

هر چه تعداد داده‌ها بیش‌تر باشد، مدل بهتر fit می‌شود و بنابراین یادگیری با دقت بالاتری انجام می‌شود؛ از طرفی هر چه تعداد ویژگی‌ها بیش‌تر شود، امکان overfit شدن نیز بیش‌تر می‌شود. برای حل کردن این مشکل باید تابع نرمال [۷] شود.

- ب) پیدا کردن رابطه‌ی بین ویژگی‌ها کار دشواری است (انتخاب ویژگی‌ها[۸] و رابطه‌ی آن‌ها با مدل). به دلیل نرمال کردن تابع، باید محدودیت‌هایی روی ابرپارامترها[۹] و ویژگی‌ها قرار دهیم. به همین خاطر این کار هزینه‌بری است.

- ج) اگر جمله‌ی منظم‌ساز را اضافه کنیم رابطه به شکل زیر در خواهد آمد:

$$J(w) = \Sigma_{i=1}^{n}(y^{(i)} - w^T x^{(i)})^2 + \lambda||w||^2 = (Y - X^T W)(Y - X^T W) + \lambda w^t w$$

$$\frac{dJ}{dw} = 0 \implies -2XX^T w + 2XY - 2\lambda w = 0$$

---

[۷] normalize
[۸] selection feature
[۹] hyper-parameters

سؤال ۳. شبکه‌های عصبی

- آ) برای این‌که بتواند خروجی صعودی شبکه را تشخیص دهد، باید به صورت زیر طراحی شود:

$$h_1 = x_1 w_{11}^{(1)} + x_2 w_{12}^{(1)} + x_3 w_{13}^{(1)} + x_4 w_{14}^{(1)} + b_{11} \to h_1 = f(x_2 - x_1)$$

$$h_2 = x_1 w_{21}^{(1)} + x_2 w_{22}^{(1)} + x_3 w_{23}^{(1)} + x_4 w_{24}^{(1)} + b_{21} \to h_2 = f(x_3 - x_2)$$

$$h_3 = x_1 w_{31}^{(1)} + x_2 w_{32}^{(1)} + x_3 w_{33}^{(1)} + x_4 w_{34}^{(1)} + b_{31} \to h_3 = f(x_4 - x_3)$$

$$\implies w^{(1)} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$y = h_1 w_{11}^{(2)} + h_2 w_{12}^{(2)} + h_3 w_{13}^{(2)} \to y = f(h_1 + h_2 + h_3 - 2)$$

$$\implies w^{(2)} = \begin{pmatrix} 1 & 1 & 1 & -2 \end{pmatrix}$$

- ب) خیر؛ ساخت آن با یک لایه نهان امکان‌پذیر نیست، زیرا:

از آن‌جایی که معادله‌ی خط در هر ناحیه محور مختصاتی متفاوت است، بنابراین بایدفضای مسئله را به ۴ ناحیه تقسیم کنیم. در هر ناحیه شرط‌ها را می‌توان به صورت AND تعدادی پارامتر (معادله خط‌های مربوط به آن ناحیه) بنویسیم. در نهایت باید بتوان بین این ۴ بخش، یک OR بگیریم که این کار نیازمند یک لایه‌ی نهان دیگر است.