

Combining
Multiple Learners

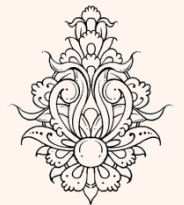
یادگیری ماشین
(۰۱-۸۰۵-۱۱-۱۳)
فصل هفدهم



دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
پاییز ۱۳۹۴
احمد محمودی ازناوه

فهرست مطالب

- ترکیب الگوریتم‌های یادگیری
- انتخاب یادگیرنده
- شیوه‌های ترکیب یادگیرنده‌ها
 - رأی‌گیری
 - Bagging
 - Boosting
 - Stacking
 - Mixture of Experts
 - Cascading



پیش‌گفتار

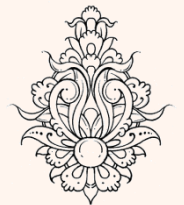
- الگوریتم‌های آموزش متفاوتی مطرح شده‌اند که در نظر گرفتن hyperparameter-های مختلف بر عملکرد آن‌ها مؤثر است.

- مثلاً برای دسته‌بندی می‌توان از دسته‌بندی پارامتری و یا شبکه‌ی عصبی (mlp) استفاده کرد.

- در صورت استفاده از شبکه‌ی عصبی، تعداد لایه‌های مخفی بر عملکرد کلی و تعداد گره‌های هر لایه بر کارایی روش اثرگذار خواهد بود.

- «هیچ الگوریتم یادگیری وجود ندارد که در همه‌ی زمینه‌ها بهینه باشد.»

No Free Lunch Theorem

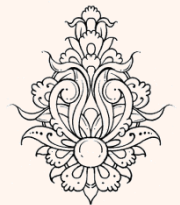


بهترین الگوریتم یادگیری

- یک راه، انتخاب بهترین الگوریتم یادگیری بر پایه‌ی یک «مجموعه‌ی اعتبار» است.

- هر الگوریتم یادگیری محدود به مدل خاصی است؛ در واقع «بایاس استقرا»ی خاصی دارد که در صورتی که مفروضات برای داده‌ها معتبر نباشد، موجب ایجاد خطا می‌شود.
- یادگیری یک مسأله‌ی ill-posed است، با در اختیار داشتن مجموعه داده‌های محدود الگوریتم‌های مختلف به پاسخ‌های متفاوتی می‌رسند.
- در صورت تنظیم پارامترهای مدل برای یک مجموعه‌ی آموزشی، (fine-tuned) ممکن است مدل برای برخی داده‌ها مناسب نباشد و مدل **دیگری** برای آن‌ها پاسخ بهتری داشته باشد.

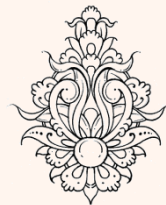
- با ترکیب چند الگوریتم یادگیری می‌توان کارایی بهتری به دست آورد.



ترکیب الگوریتم‌های یادگیری (ادامه...)

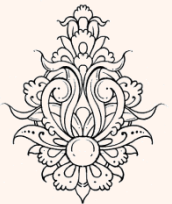
Base-learner

- بدین ترتیب چندین «یادگیرنده‌ی پایه» برای به دست آوردن کارایی بهتر با هم ترکیب می‌شوند.
 - چگونه یادگیرنده‌های پایه‌ای انتخاب شوند که عملکرد یکدیگر را پوشش دهند؟
 - چگونه فروجی یادگیرنده‌های مختلف برای به دست آوردن بهترین نتیجه با هم ترکیب شوند؟
- ترکیب یادگیرنده‌های یکسان سودی ندارد، گاهی ممکن است این کار تنها منجر به افزایش هزینه‌ی محاسباتی شود.



انتخاب یادگیرنده

- استفاده از **الگوریتم** های یادگیری متفاوت:
 - هر الگوریتم یادگیری مفروضات خاصی دارد. با انتخاب تنها یک الگوریتم روی فرضیات آن مسأله تاکید شده و سایر موارد مخفول می ماند. به عنوان مثال ترکیب روش های پارامتری و ناپارامتری
- استفاده از **hyperparameter** - های متفاوت:
 - برای یک شیوهی یادگیری پارامترهای متفاوتی در نظر گرفت که در کارایی نهایی مؤثرند: تعداد گره ها در لایه ی مخفی شبکه ی عصبی، k در k -nn، حد آستانه ی خطا در دریافت تصمیم، تابع کرنل در ماشین های بردار پشتیبان، استفاده از ماتریس کواریانس یکسان/متفاوت در روش های پارامتری، انتخاب حالت اولیه در روش های تکرار شونده (نزول گرادیان)
 - با آموزش با hyperparameter های متفاوت و میانگین گیری ساده واریانس و در نتیجه فضای کلی کاهش می یابد.



انتخاب یادگیرنده (ادامه...)

- استفاده از **نمایش متفاوت داده‌ها** برای هر یادگیرنده:

– به جای ترکیب داده‌های مختلف، می‌توان بردارهای فمبصه‌ی کوچک‌تری برای یادگیرنده‌های متفاوت در نظر گرفت.

- به عنوان مثال در شناسایی صوت؛ استفاده‌ی جداگانه ویدئوی حرکت لب‌ها و سیگنال صوتی

sensor fusion

- استفاده از کلمات کلیدی در کنار تصویر در «بازیابی تصویر»

Multi-view learning

- در صورتی که داده‌ها دارای نمایش واحدی باشند، باز هم می‌توان از چنین شیوه‌ای بهره گرفت، باعث می‌شود هر یادگیرنده روی زیرفضای خاصی از داده تمرکز کند، مانند

random subspace
(attribute bagging)

random forest



انتخاب یادگیرنده (ادامه...)

• استفاده از **مجموعه‌های آموزشی متفاوت**:

– با توجه به تاثیر داده‌هایی که برای آموزش به کار می‌روند، رویکرد دیگر استفاده از مجموعه‌های آموزشی متفاوت است.

– یک راه انتخاب تصادفی داده‌های آموزشی برای هر یادگیرنده است.

– راه دیگر استفاده از یک یادگیرنده برای داده‌های است که در یادگیرنده(ها)ی مورد استفاده منجر به ارائی پاسخ مطلوب نشده‌اند.

– هر یادگیرنده به نواحی خاصی از داده‌ها به صورت محلی اختصاص یابد.

– می‌توان کار اصلی را به چند کار جزیی‌تر تقسیم نمود.

bagging

boosting

cascading

mixture of experts

error correcting output code(ECOC)

یادگیرنده‌ها صرفاً به خاطر کارایی مورد استفاده قرار نمی‌گیرند، سادگی و تفاوت آن‌ها اهمیت دارد. (Diversity vs. Accuracy)



شیوه‌های ترکیب یادگیرنده‌ها

Multiexpert combination

- یادگیرنده پس از آموزش به صورت موازی داده‌ی ورودی را دریافت و تصمیم نهایی بر اساس ترکیب تصمیم‌ها انجام می‌شود.

– رأی‌گیری، ترکیب خبره‌ها و تصمیم پیشته‌وار (stacked generalization)

Multistage combination

- هر یادگیرنده تنها در صورتی مورد استفاده (آموزش) قرار می‌گیرد که یادگیرنده‌های قبلی توفیق نسبی در انجام کار خویش نداشته‌اند.

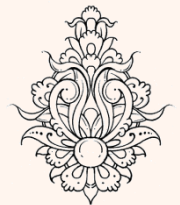
– یادگیرنده‌ها بر اساس پیچیدگی مرتب می‌شوند.

– اتصال سری (cascading)

- در صورتی که L یادگیرنده مفروض باشد که تصمیم یادگیرنده‌ی j -ام (\mathcal{M}_j) با $d_j(x)$ نشان داده شود، تصمیم نهایی به صورت زیر نمایش داده می‌شود:

$$y = f(d_1, d_2, \dots, d_L | \Phi)$$

combining function



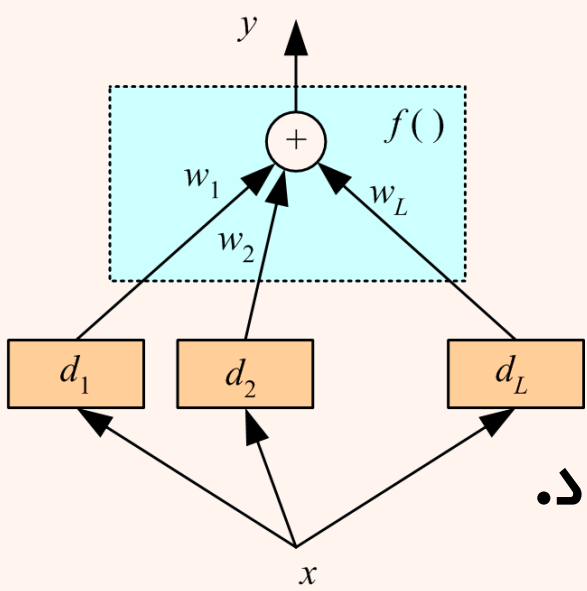
• ترکیب خطی فروجی یادگیرنده ها:

$$y = \sum_{j=1}^L w_j d_j$$

$$w_j \geq 0 \text{ and } \sum_{j=1}^L w_j = 1$$

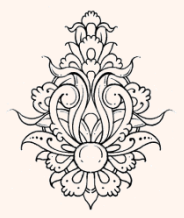
ensembles -linear opinion pools

• برای دسته بندی هر یادگیرنده به هر کلاس یک رأی اختصاص می دهد:



$$y_i = \sum_{j=1}^L w_j d_{ji}$$

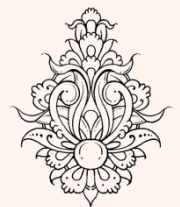
• وزن هر یادگیرنده را می توان از روی صحت عملکرد آن تعیین کرد.



سایر شیوه‌های جمع‌بندی

Rule	Fusion function $f(\cdot)$
Sum	$y_i = \frac{1}{L} \sum_{j=1}^L d_{ji}$
Weighted sum	$y_i = \sum_j w_j d_{ji}, w_j \geq 0, \sum_j w_j = 1$
Median	$y_i = \text{median}_j d_{ji}$
Minimum	$y_i = \min_j d_{ji}$
Maximum	$y_i = \max_j d_{ji}$
Product	$y_i = \prod_j d_{ji}$

	C_1	C_2	C_3
d_1	0.2	0.5	0.3
d_2	0.0	0.6	0.4
d_3	0.4	0.4	0.2
Sum	0.2	0.5	0.3
Median	0.2	0.5	0.4
Minimum	0.0	0.4	0.2
Maximum	0.4	0.6	0.4
Product	0.0	0.12	0.032



$$y = \sum_{j=1}^L w_j d_j$$

رای گیری (ادامه...)

- می توان به رای گیری از دریچه ی قانون Bayes نگریم:

$$w_j \equiv P(\mathcal{M}_j) \quad d_{ji} \equiv P(C_i | x, \mathcal{M}_j)$$

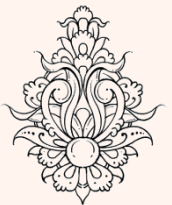
$$P(C_i | x) = \sum_{\text{all models } \mathcal{M}_j} P(C_i | x, \mathcal{M}_j) P(\mathcal{M}_j)$$

- در صورت که رای گیری یک میان گیری ساده فرض شود و یادگیرنده ها مستقل در نظر گرفته شوند:

$$E[y] = E\left[\sum_j \frac{1}{L} d_j\right] = \frac{1}{L} L \cdot E[d_j] = E[d_j]$$

$$\text{Var}(y) = \text{Var}\left(\sum_j \frac{1}{L} d_j\right) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L^2} L \cdot \text{Var}(d_j) = \frac{1}{L} \text{Var}(d_j)$$

واریانس و در نتیجه خطا کاهش می یابد



رای گیری (ادامه...)

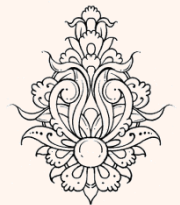
- در صورتی یادگیرنده‌ها مستقل نباشند:

$$\text{Var}(y) = \frac{1}{L^2} \text{Var}\left(\sum_j d_j\right) = \frac{1}{L^2} \left[\sum_j \text{Var}(d_j) + 2 \sum_j \sum_{i < j} \text{Cov}(d_i, d_j) \right]$$

- در صورتی که همبستگی مثبت بین یادگیرنده‌ها وجود داشته باشد، میزان واریانس و در نتیجه خطا افزایش خواهد یافت.

- در صورت همبستگی منفی بین یادگیرنده‌ها میزان واریانس کمتر هم خواهد شد.

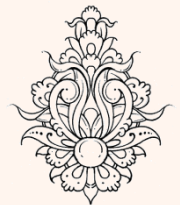
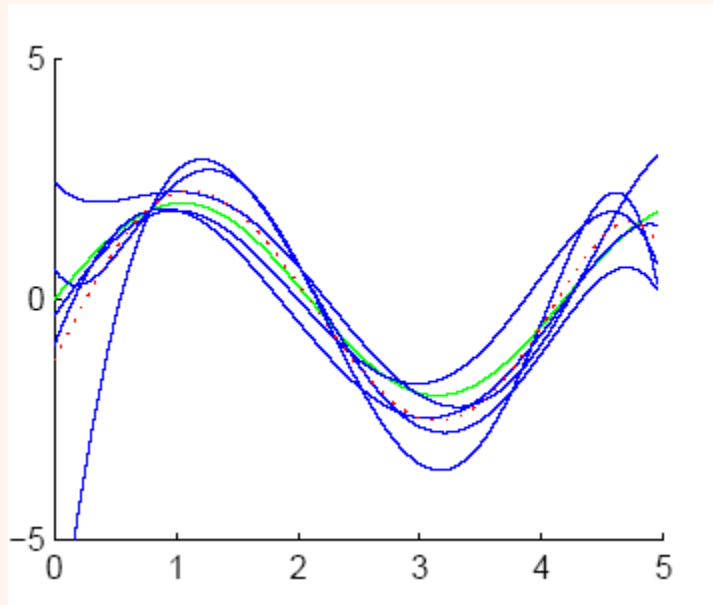
– مشروط به این که بایاس افزایش نیابد! (؟)



رای‌گیری (ادامه...)

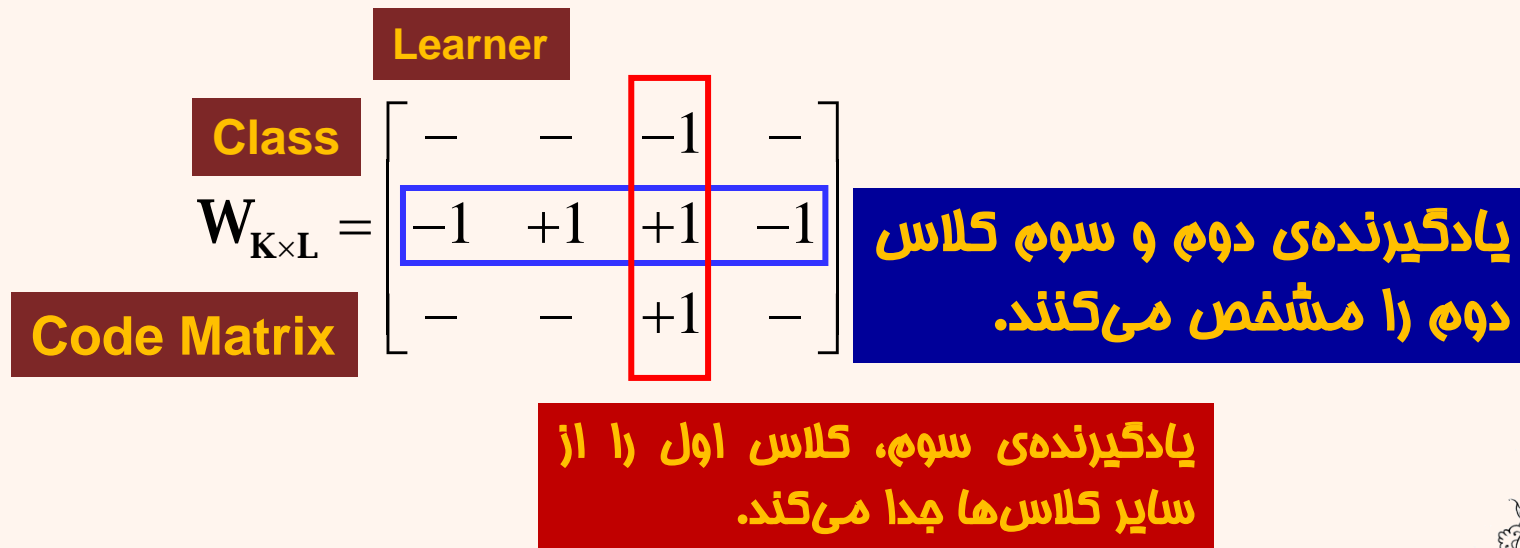
- می‌توان هر یادگیرنده را به صورت تابع درست به همراه نویز تصور کرد. در نتیجه رای‌گیری به نوعی فرآیند کاهش نویز خواهد بود.

– استفاده از مدل‌هایی با بایاس کم و واریانس بالا

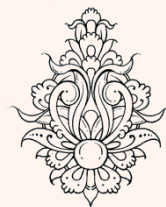


Error-Correcting Output Codes(ECOC)

- در این شیوه یک دسته‌بندی پیچیده، به یک سری دسته‌بندی ساده‌تر تقسیم می‌شوند.



- در این حالت مجموعه‌ی آموزشی برای هر یادگیرنده به دو دسته تقسیم می‌شود.



Error-Correcting Output Codes(ECOC)

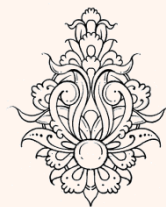
- در حالتی که برای هر کلاس یک یادگیرنده در نظر گرفته شود، در صورت بروز خطا در یکی از یادگیرنده‌ها پاسخ نهایی اشتباه خواهد بود.
- برای رفع این مشکل می‌توان تعداد یادگیرنده‌ها را افزایش داد ($L > K$).
 - فاصله‌ی همینگ بین سطرها افزایش می‌یابد.
 - یک راه استفاده از یک دسته‌بند به ازای هر دو کلاس است.

One per class ($L=K$)

$$W = \begin{bmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{bmatrix}$$

Pairwise $L=K(K-1)/2$

$$W = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$



Error-Correcting Output Codes(ECOC)

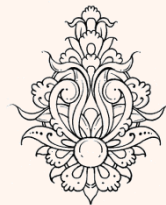
- برای تعداد کلاس (K) بالا جداسازی دوطرفه و شدنی نیست.
- می‌توان حالت‌هایی که بیشترین فاصله‌ی همینگ را دارند انتخاب نمود.

Full code $L=2^{(K-1)}-1$

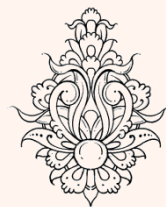
$$W = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & +1 & +1 & +1 & +1 \\ -1 & +1 & +1 & -1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 \end{bmatrix}$$

$$y_i = \sum_{j=1}^L w_j d_{ji}$$

- نوعی فرآیند رأی‌گیری است که وزن هر یادگیرنده به کلاس نیز بستگی دارد.
- این رأی‌گیری را می‌توان به صورت soft هم انجام داد.

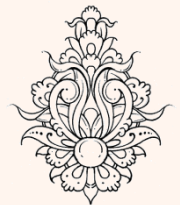


- در این شیوه، یادگیرنده‌های یکسان از مجموعه‌های آموزشی متفاوت استفاده می‌کنند.
 - برای تولید مجموعه‌های آموزشی متفاوت از bootstrap استفاده می‌شود.
 - از یک مجموعه‌ی آموزشی به طول N ، N داده به صورت تصادفی و همراه با جایگزینی استخراج می‌شود. فرآیند فوق L بار تکرار می‌شود.
 - از رأی‌گیری (میانگین‌گیری) برای ترکیب نتایج استفاده می‌شود.
- پایداری الگوریتم یادگیری را افزایش می‌دهد. (کاهش واریانس)
 - یک الگوریتم «ناپایدار» است، در صورتی که تغییر کوچکی در مجموعه‌ی آموزشی منجر به تغییر کلی یادگیرنده شود.



Boosting

- در این شیوه به جای انتخاب تصادفی داده‌های آموزشی، مجموعه‌ی آموزشی برای یک یادگیرنده بر اساس اشتباهات یادگیرنده(ها)ی قبلی شکل می‌گیرد.
- در روش اولیه از ترکیب سه یادگیرنده‌ی ضعیف یک یادگیرنده‌ی قوی‌تر ایجاد می‌شود.
- یادگیری:
 - مجموعه‌ی آموزشی (X) به سه بخش افراز می‌شود (X_1, X_2, X_3)
 - از X_1 برای آموزش d_1 استفاده می‌شود.
 - از تمام نمونه‌های اشتباه تشخیص داده شده X_1 توسط d_1 و X_2 برای آموزش d_2 استفاده می‌شود.
 - به همین ترتیب برای یادگیرنده‌ی سوم

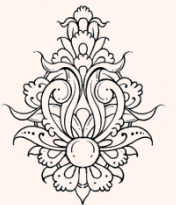


• آزمون:

– در این مرحله نتیجه‌ی اعمال ورودی به یادگیرنده‌ی اول و دوم بررسی می‌شود. در صورت یکسان بودن نتیجه، تصمیم دو یادگیرنده‌ی اول پذیرفته می‌شود وگرنه رأی یادگیرنده‌ی سوم تعیین کننده خواهد بود.

• از معایب این روش نیاز به تعداد داده‌های آموزشی بالاست.

• در روش AdaBoost از یک مجموعه‌ی آموزشی آموزشی بارها استفاده می‌شود.



AdaBoost.M1 (Adaptive Boosting)

- در این روش تعداد یادگیرنده‌های پایه محدود نیست.
- در این شیوه احتمال انتخاب داده براساس خطای یادگیرنده‌های قبلی تعیین می‌شود.
- بعد از تکمیل آموزش، از رأی‌گیری ساده استفاده می‌کند.
- در صورتی که یادگیرنده‌ها پیچیده انتخاب شوند، در مراحل بعدی تنها نویز باقی خواهد ماند.
 - به عنوان مثال در حالتی که از درخت تصمیم استفاده می‌شود، برای عمق درخت محدودیت در نظر گرفته می‌شود.
- نشان داده شده است که این شیوه منجر به افزایش ماشیه جداسازی می‌شود.

Decision stumps



AdaBoost.M1 (Adaptive Boosting)

Training:

For all $\{x^t, r^t\}_{t=1}^N \in \mathcal{X}$, initialize $p_1^t = 1/N$

For all base-learners $j = 1, \dots, L$

Randomly draw \mathcal{X}_j from \mathcal{X} with probabilities p_j^t

Train d_j using \mathcal{X}_j

For each (x^t, r^t) , calculate $y_j^t \leftarrow d_j(x^t)$

Calculate error rate: $\epsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t)$

If $\epsilon_j > 1/2$, then $L \leftarrow j - 1$; stop

$\beta_j \leftarrow \epsilon_j / (1 - \epsilon_j)$

For each (x^t, r^t) , decrease probabilities if correct:

If $y_j^t = r^t$ $p_{j+1}^t \leftarrow \beta_j p_j^t$ Else $p_{j+1}^t \leftarrow p_j^t$

Normalize probabilities:

$Z_j \leftarrow \sum_t p_{j+1}^t$; $p_{j+1}^t \leftarrow p_{j+1}^t / Z_j$

Testing:

Given x , calculate $d_j(x), j = 1, \dots, L$

Calculate class outputs, $i = 1, \dots, K$:

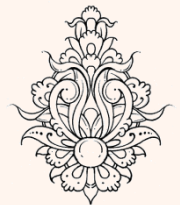
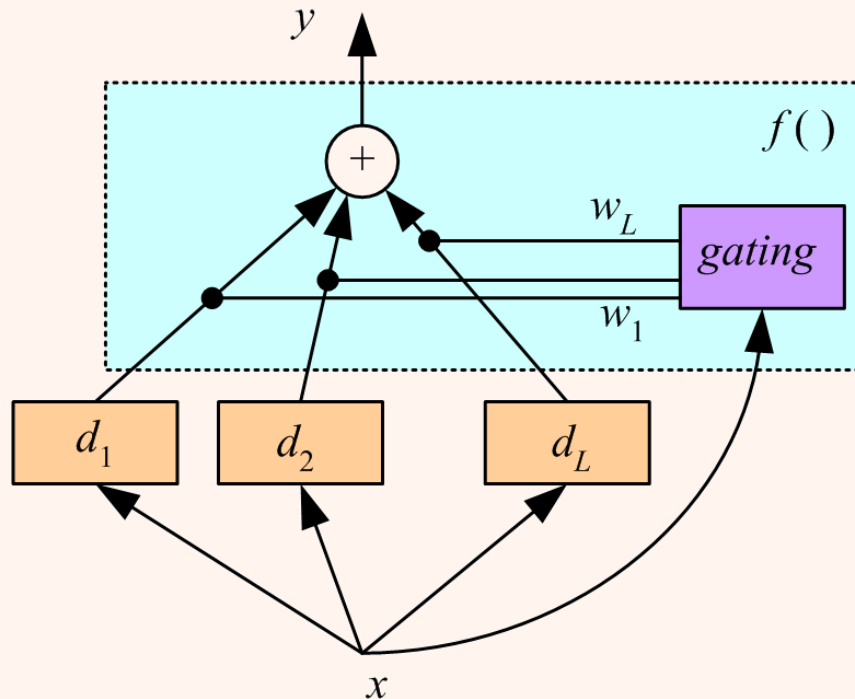
$$y_i = \sum_{j=1}^L \left(\log \frac{1}{\beta_j} \right) d_{ji}(x)$$



- مانند رأی‌گیری است با این تفاوت که وزن آرا به ورودی بستگی دارد.

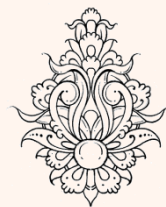
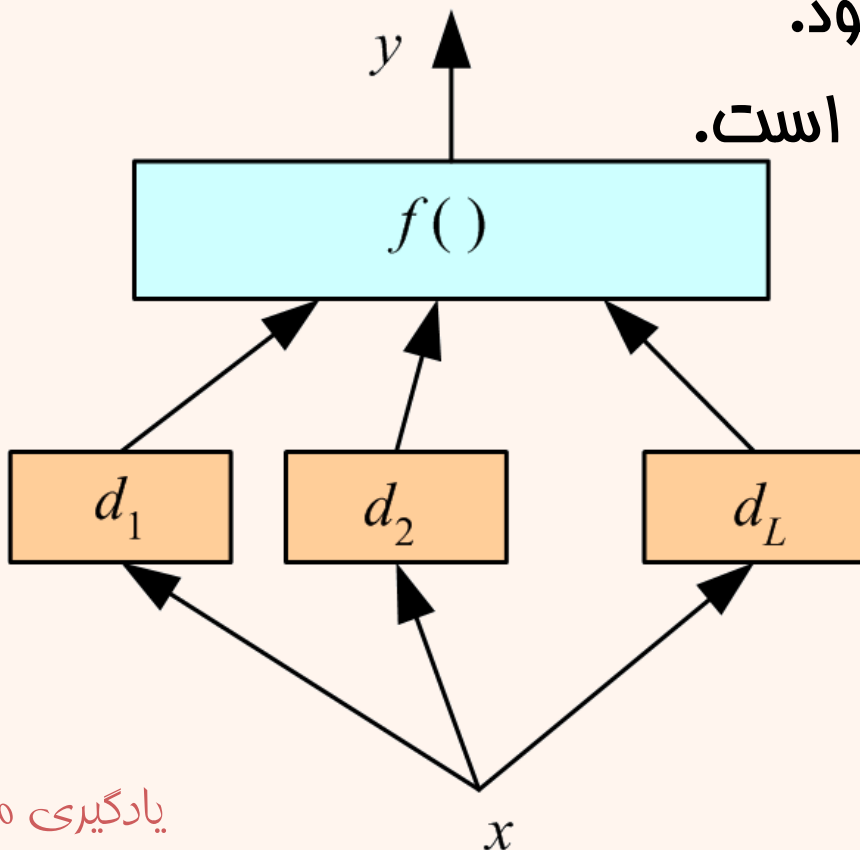
– مانند الگوریتم‌های رقابتی

$$y = \sum_{j=1}^L w_j(x) d_j$$



Stacking

- در این شیوه، نمونه‌ی ترکیب نتایج نیز توسط الگوریتم‌های یادگیری انجام می‌شود.
- برای این منظور باید از مجموعه‌ای جدا از مجموعه‌ی آموزشی استفاده شود.
- هدف کاهش بایاس است.



Cascading

