

(a) آشکار ساز در مرحله ای: شامل در مرحله اول استخراج proposal region، سپس classification، bounding box است. (R-CNN)
 - تک مرحله: در تک مرحله احتمال هر کلاس و bounding box را برای فینس box تعیین می‌کنند (val) و مقایسه می‌کنند. معمولاً در مرحله ای از تک مرحله گنجانده تر است. معمولاً در مرحله ای accuracy بیشتری نسبت به تک مرحله مخصوصاً در detect های بی‌دانه دارند.
 در حوزه هکریال با توجه به سرعت بودن مدل‌های تک مرحله ای، به هنگام نیاز به تصمیم گیری در real time رایج است از این‌ها استفاده شود. اما در مناطقی که دقت بیشتری مورد نیاز است مثل آنالیز تصاویر از الگوریتم‌های دو مرحله ای استفاده کرد. مثلاً به هنگام کمک به یک پزشک در پیدا کردن نواحی مورد نظر در اتاق عمل می‌توان از روش تک مرحله ای اما به هنگام شناسایی نواحی مورد نیاز در مطب که نیاز به سرعت زیاد نیست از مدل‌های ۲ مرحله ای استفاده کرد.

(b) الگوریتم کلی R-CNN به شرح زیر است:

- 1- بایک الگوریتم مبتنی بر search، مورد 2000 Region Proposal استخراج می‌شود.
- 2- هر ناحیه استخراج شده به یک اندازه ثابت درآمده و به یک شبکه کانفولوشنی pre-train یا ResNet وارد می‌شود.
- 3- با استفاده از خروجی‌های شبکه در هر یک از نقاط classification loss و bounding box (L2 loss) دارد.

Fast R-CNN: در R-CNN هر Region Proposal نیز به یک forward pass شبکه عصبی دارد که باعث کندی الگوریتم می‌شود. Fast R-CNN با استفاده از convolutional features بین تمام ناحیه‌ها به یک بار بدون کپی‌برداری شده است.
 - به جای پردازش هر Region Proposal جداگانه، Fast R-CNN با ROI pooling و feature map، ROI pooling به یک اندازه ثابت با استفاده از max pooling محاسبه می‌کند.
 Faster R-CNN: این روش یک شبکه integrated برای تولید region proposal ها معرفی می‌کند که مستقیماً از روی feature map ها استخراج می‌شوند.
 - این روش قابلیت آموزش end-to-end را به مدل می‌دهد.

(a) دلایل نوسان بودن saliency map ها:

- 1) ناپیوستگی در گره‌های بی‌اعتبار است؛ به علت استفاده از ReLU activation function های ناپیوسته.

$$\text{smooth Grad}(m) := \frac{1}{n} \int_0^1 \frac{\partial f(x^*)}{\partial x^*} dx^*, \quad x^* = x + N(0, \sigma^2)$$
- 2) saturation: $f(x)$ یک ویژگی ممکن است تا حدی global و قدری داشته باشد و درivate محلی نداشته باشد.

- Int Grad (x) := $\frac{\partial f(x^*)}{\partial x^*}$, $x^* = \alpha n$, $0 < \alpha < 1$ رنج: truthful بودن saliency map ها:
 ویژگی‌ها: به صورت تصادفی در تصویر scatter شده اند و در نحوه تصمیم گیری شبکه نقش مهمی دارند.

(b) selectivity evaluation صرف ارزیابی کلی از روی attribution-based در تفسیر پذیری مدل هاست. attribution-based صرفاً به این معنی است که هر بخشی از ورودی چه در تصمیم گیری مدل تأثیر داشته اند. ارزیابی selectivity بر اینست که attribution مشخص کننده اهمیت هر feature یا خاصه در ورودی است، تمرکز دارد. Pixel-flipping: حذف ویژگی های مهم مشخص کرده توسط مدل تفسیر پذیری و مشاهده آن تغییرکاری مدل در Prediction (تغییر ارزیابی) را انجام می دهد.

شبهه که:

```

pcurve = []
for p in argsort(-R) do ← مرتب کردن ویژگی ها بر اساس اهمیت به ترتیب نزولی
    n ← n - d * p ← حذف ویژگی از تصویر ورودی
    pcurve.append(f(x))
return pcurve

```

- در نمودار مرتب سول بعد از حذف Pixel های Perturb شده از تصویر ورودی و محدودی می بینیم score خفیف است.

(3)

(a) distributed learning دنیا (centrally ذخیره شدن است) مطلقاً به خاطر سرعت بیشتر آموزش دنیا به سرعت یکسانیت و تعارض بین worker ها یعنی هرگز. FL دنیا به سرعت ذاتی توزیع شده و به سرعت محلی که به سرور است و در نهایت دنیا iid است، هم چنین است. Imbalance باید.

(b) در MAML هدف یافتن مدلی است که در تمام task ها به خوبی در مورد انتظار عمل کند، در عوض فرقی ندارد به سیستم رسیدن task جدید باید initialization مناسب به نقطه چند step از Gradient descent، مدل به خوبی عمل کند. در نتیجه با توجه به هر user ها انجام می گیرد GD بر روی تابع هزینه خودش داریم:

$$\min_{w \in \mathbb{R}^d} F(w) := \frac{1}{n} \sum_{i=1}^n f_i(w - \alpha \nabla f_i(w)) \quad (1)$$

برای حل این عبارت، FedAvg در هر راند، سرور کس از user ها را انتخاب می کند و مدل کنونی را به هر کدام از آنها ارسال می کند و هر user به اساس تابع هزینه خود مدل را Update می کند و سپس به سرور ارسال می کند. در نهایت سرور با average کردن مدل Global را آپدیت می کند. در FedAvg از مفاهیم مشابه استفاده می شود. عبارت یک سی ترانه به سرعت $F_i(w) := f_i(w - \alpha \nabla f_i(w))$ به نویسنده شود. (از هر کاربر) جهت محاسبه هر local function ها داریم:

$$\nabla F_i(w) = (I - \alpha \nabla^2 f_i(w)) \nabla f_i(w - \alpha \nabla f_i(w)).$$

مکعب $\nabla^3 f_i(w)$ در هر راند هزینه بالاتر به محسنت است از batch ای از دیتا به اساس گزینش p_i بزرگترین unbiased $\tilde{\nabla}^2 f_i(w, D^i)$ استفاده می شود. در نتیجه داریم:

$$\tilde{\nabla}^2 f_i(w, D^i) = \frac{1}{|D^i|} \sum_{(x,y) \in D^i} \nabla^2 f_i(w; x, y)$$

به طرقت به ماتریس سین $\nabla^2 f_i(w)$ می براند $\tilde{\nabla}^2 f_i(w, D^i)$ تخمین زده شود.

در مرحله k ام از روش، مکتب FedAvg، سرور مدل global (w_k) را به کسری از $user$ ها ارسال می کند. هر $user$ $i \in A_k$ به اعمال حتم SGD محلی به نوبت به F_i محسنت می کند. در نتیجه به روز رسانی محلی $\{w_{k+1,t}^i\}_{t=0}^T$ که $w_k = w_{k+1,0}^i$ برای $t \geq 1$ کوئید می شود داریم:

$$w_{k+1,t}^i = w_{k+1,t-1}^i - \underset{\text{local LR}}{\beta} \tilde{\nabla} F_i(w_{k+1,t-1}^i)$$

توجه شود که در هر مرحله از SGD در $\tilde{\nabla} F_i(w_{k+1,t-1}^i)$ به جای مستقل $D_t^i, D_{t-1}^i, D_{t-2}^i$ استفاده می شود. در نتیجه داریم:

$$\tilde{\nabla} F_i(w_{k+1,t-1}^i) = (I - \alpha \tilde{\nabla}^2 f_i(w_{k+1,t-1}^i, D_t^i)) \tilde{\nabla} f_i(w_{k+1,t-1}^i, D_t^i) - \alpha \tilde{\nabla} f_i(w_{k+1,t-1}^i, D_t^i), D_{t-1}^i)$$

* $\tilde{\nabla} F_i(w_{k+1,t-1}^i)$ تخمین biased از $\nabla F_i(w_{k+1,t-1}^i)$ هست به دلیل وجود SGD داخل SGD دیگر.

پس از ازنس به روز رسانی محلی $w_{k+1,t}$ ، $user$ آنرا را به سرور ارسال می کند، سرور به Global averaging مدل را update می کند.

- ماتریس Hessian در update rule per-FedAvg ظاهر می شود که در بالا علت به وجه آمدن توضیح داده شده است.