Learning without forgetting

Saturday, December 10, 2022 6:28 PM

$\theta_s \rightarrow$ shared weights (CNN)

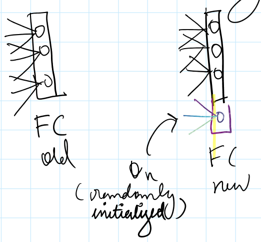
$\theta_o \rightarrow$ old weights (pre-trained)

$\theta_n \rightarrow$ new weights (fine tuning)

Method

1) Record responses y_o on each new task image from the original network for outputs on the old tasks defined by θ_s & θ_o

$y_o \rightarrow$ set of class probabilities for each image.



number of new parameters = no. of new classes × no. of nodes in last shared layer

2) Train model to minimize loss for all tasks and Regularization R ~~for~~ using SGD

$R \rightarrow$ weight decay of 0.0005

3) First freeze θ_s and θ_o & train θ_n to convergence. (warm up)

4) Jointly train all weights θ_s, θ_o, θ_n until convergence (joint optimize step)

* Warm up $\rightarrow$ enhances fine tuning's old task performance (not crucial)

Loss encourages predictions $\hat{y}_n$ to be consistent with the gnd-truths y_n

Multinomial logistic loss

$$\mathcal{L}_{new}(y_n, \hat{y}_n) = -y_n \cdot \log \hat{y}_n$$

$\hat{y}_n$ = softmax output

$y_n \rightarrow$ one hot encoded gnd truth vector

multilabel classification

multilabel classification

$\Downarrow$

true false predictions
for each label

for original task → output prob
use knowledge distillation loss. { as close as possible to output of original network

$$\mathcal{L}_{old}(y_o, \hat{y}_o) = -H(y_o', \hat{y}_o')$$

$$= -\sum_{i=1}^{l} y_o'^{(i)} \log \hat{y}_o'^{(i)}$$

where $l \Rightarrow$ no. of labels

$y_o'^{(i)} \rightarrow$ recorded prob
$\hat{y}_o'^{(i)} \rightarrow$ current prob } Modified version of $y_o^{(i)}$ & $\hat{y}_o^{(i)}$

* Modified CE loss which increases the weight for smaller probabilities.

$$y_o'^{(i)} = \frac{\left(y_o^{(i)}\right)^{1/T}}{\sum_j \left(y_o^{(j)}\right)^{1/T}} \qquad \hat{y}_o'^{(i)} = \frac{\left(\hat{y}_o^{(i)}\right)^{1/T}}{\sum_j \left(\hat{y}_o^{(j)}\right)^{1/T}}$$

$T > 1$ → increases the weight of the smaller of logits.

$T = 2$ is used (found by grid search)

$\lambda_o \rightarrow 1$ { loss balance weight
larger values favour old task performance over new task
[1 = balance]

* Instead of storing old task data [Joint training] we use recorded response [Y_o] instead.

[Joint training] we use recorded responses [Y_o] instead.

This is used for joint optimization of θ_s

Algorithm

start:

$\theta_s \rightarrow$ shared params

$\theta_o \rightarrow$ old task params

$X_n, Y_n \rightarrow$ training data & ground truth of the new task.

Initialize

$$Y_o \leftarrow CNN(X_n, \theta_s, \theta_o)$$

$$\theta_n \leftarrow RandInit(|\theta_n|)$$

Train

$$\hat{Y}_o = CNN(X_n, \hat{\theta}_s, \hat{\theta}_o) \text{ // old task}$$

$$\hat{Y}_n = CNN(X_n, \hat{\theta}_s, \hat{\theta}_n) \text{ // new task}$$

$$\theta_s^*, \theta_o^*, \theta_n^* \leftarrow \underset{(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n)}{\operatorname{argmin}} \left(\lambda_o L_{old}(Y_o, \hat{Y}_o) + L_{new}(Y_n, \hat{Y}_n) + R(\hat{\theta}_s, \hat{\theta}_o, \hat{\theta}_n) \right)$$