Temproal Ensembling

Π Model

提出了一种约束假设，对于加入扰动的两个同种无标签input会输出一致的output。

$$min(f(x) - f^*(x^*)) \tag{1}$$

Algorithm

Algorithm 1 Π-model pseudocode.

Require: x_i = training stimuli
Require: L = set of training input indices with known labels
Require: y_i = labels for labeled inputs $i \in L$
Require: $w(t)$ = unsupervised weight ramp-up function
Require: $f_\theta(x)$ = stochastic neural network with trainable parameters θ
Require: $g(x)$ = stochastic input augmentation function
  for t in $[1, num\_epochs]$ **do**
    for each minibatch B **do**
      $z_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$ ▷ evaluate network outputs for augmented inputs
      $\tilde{z}_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$ ▷ again, with different dropout and augmentation
      $loss \leftarrow -\frac{1}{|B|} \sum_{i \in (B \cap L)} \log z_i[y_i]$ ▷ supervised loss component
        $+ w(t) \frac{1}{C|B|} \sum_{i \in B} ||z_i - \tilde{z}_i||^2$ ▷ unsupervised loss component
      update θ using, e.g., ADAM ▷ update network parameters
    end for
  end for
  return θ

$w(t)$逐渐增大

Π Model with Temproal Ensembling

Algorithm

Algorithm 2 Temporal ensembling pseudocode. Note that the updates of Z and $\tilde{z}$ could equally well be done inside the minibatch loop; in this pseudocode they occur between epochs for clarity.

Require: x_i = training stimuli
Require: L = set of training input indices with known labels
Require: y_i = labels for labeled inputs $i \in L$
Require: α = ensembling momentum, $0 \leq \alpha < 1$
Require: $w(t)$ = unsupervised weight ramp-up function
Require: $f_\theta(x)$ = stochastic neural network with trainable parameters θ
Require: $g(x)$ = stochastic input augmentation function
  $Z \leftarrow \mathbf{0}_{[N \times C]}$ ▷ initialize ensemble predictions
  $\tilde{z} \leftarrow \mathbf{0}_{[N \times C]}$ ▷ initialize target vectors
  for t in $[1, num\_epochs]$ **do**
    for each minibatch B **do**
      $z_{i \in B} \leftarrow f_\theta(g(x_{i \in B}, t))$ ▷ evaluate network outputs for augmented inputs
      $loss \leftarrow -\frac{1}{|B|}\sum_{i \in (B \cap L)} \log z_i[y_i]$ ▷ supervised loss component
        $+ w(t)\frac{1}{C|B|}\sum_{i \in B} ||z_i - \tilde{z}_i||^2$ ▷ unsupervised loss component
      update θ using, e.g., Adam ▷ update network parameters
    end for
    $Z \leftarrow \alpha Z + (1 - \alpha) z$ ▷ accumulate ensemble predictions
    $\tilde{z} \leftarrow Z / (1 - \alpha^t)$ ▷ construct target vectors by bias correction
  end for
  return θ

Advantage

1. 每个epoch只更新一次$\tilde{z}$，相当于节省一半时间
2. $\tilde{z}$包含噪声更少