Neural networks

Training CRFs - loss function

LINEAR CHAIN CRF

Topics: reminder of notation

- Then we have:

$$p(\mathbf{y}|\mathbf{X}) = \exp\left(\sum_{k=1}^{K} a_u(y_k) + \sum_{k=1}^{K-1} a_p(y_k, y_{k+1})\right) / Z(\mathbf{X})$$

where

$$Z(\mathbf{X}) = \sum_{y'_1} \sum_{y'_2} \cdots \sum_{y'_K} \exp\left(\sum_{k=1}^{K} a_u(y'_k) + \sum_{k=1}^{K-1} a_p(y'_k, y'_{k+1})\right)$$

- Two types of (log-)factors:
  - unary: $a_u(y_k) = a^{(L+1,0)}(\mathbf{x}_k)_{y_k} +$
    $1_{k>1}\ a^{(L+1,-1)}(\mathbf{x}_{k-1})_{y_k} +$
    $1_{k<K}\ a^{(L+1,+1)}(\mathbf{x}_{k+1})_{y_k}$
  - pairwise: $a_p(y_k, y_{k+1}) = 1_{1\leq k<K}\ V_{y_k, y_{k+1}}$

MACHINE LEARNING

Topics: empirical risk minimization, regularization

- Empirical risk minimization
  - framework to design learning algorithms

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_t l(\mathbf{f}(\mathbf{X}^{(t)}; \boldsymbol{\theta}), \mathbf{y}^{(t)}) + \lambda \Omega(\boldsymbol{\theta})$$

  - $l(\mathbf{f}(\mathbf{X}^{(t)}; \boldsymbol{\theta}), \mathbf{y}^{(t)})$ is a loss function
  - $\Omega(\boldsymbol{\theta})$ is a regularizer (penalizes certain values of $\boldsymbol{\theta}$)
- Learning is cast as optimization
  - ideally, we'd optimize classification error, but it's not smooth
  - loss function is a surrogate for what we truly should optimize

MACHINE LEARNING

Topics: stochastic gradient descent (SGD)

- Algorithm that performs updates after each example
  - initialize $\boldsymbol{\theta}$
  - for N iterations
    - for each training example $(\mathbf{X}^{(t)}, \mathbf{y}^{(t)})$
      - ✓ $\Delta = -\nabla_{\boldsymbol{\theta}} l(\mathbf{f}(\mathbf{X}^{(t)}; \boldsymbol{\theta}), \mathbf{y}^{(t)}) - \lambda \nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta})$
      - ✓ $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\, \Delta$

  } training epoch = iteration over **all** examples

- To apply this algorithm to a CRF, we need
  - the loss function $l(\mathbf{f}(\mathbf{X}^{(t)}; \boldsymbol{\theta}), \mathbf{y}^{(t)})$
  - a procedure to compute the parameter gradients $\nabla_{\boldsymbol{\theta}} l(\mathbf{f}(\mathbf{X}^{(t)}; \boldsymbol{\theta}), \mathbf{y}^{(t)})$
  - the regularizer $\Omega(\boldsymbol{\theta})$ (and the gradient $\nabla_{\boldsymbol{\theta}} \Omega(\boldsymbol{\theta})$)
  - initialization method

LOSS FUNCTION

Topics: loss function for sequential classification with CRF

- CRF estimates $p(\mathbf{y}|\mathbf{X})$
  - we could maximize the probabilities of $\mathbf{y}^{(t)}$ given $\mathbf{X}^{(t)}$ in the training set

- To frame as minimization, we minimize the negative log-likelihood

$$l(\mathbf{f}(\mathbf{X}), \mathbf{y}) = -\log p(\mathbf{y}|\mathbf{X})$$

  - unlike for non-sequential classification, we never explicitly compute the value of $p(\mathbf{y}|\mathbf{X})$ for all values of $\mathbf{y}$