Neural networks

Training CRFs - pairwise log-factor gradient

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

PARAMETER GRADIENTS

Topics: loss gradient at pairwise log-factor and parameters

- Partial derivative for log-factor:

$$\frac{\partial - \log p(\mathbf{y}|\mathbf{X})}{\partial a_p(y'_k, y'_{k+1})} \quad = \quad -(1_{y_k = y'_k, y_{k+1} = y'_{k+1}} - p(y'_k, y'_{k+1}|\mathbf{X}))$$

- Partial derivative of log-factor parameters:

$$\frac{\partial - \log p(\mathbf{y}|\mathbf{X})}{\partial V_{y'_k, y'_{k+1}}} \quad = \quad \sum_{k=1}^{K-1} -(1_{y_k = y'_k, y_{k+1} = y'_{k+1}} - p(y'_k, y'_{k+1}|\mathbf{X}))$$

- Gradient of log-factor parameters

$$\begin{aligned} \nabla_{\mathbf{V}} - \log p(\mathbf{y}|\mathbf{X}) \quad &= \quad \sum_{k=1}^{K-1} -(\mathbf{e}(y_k)\ \mathbf{e}(y_{k+1})^{\top} - \overbrace{\mathbf{p}(y_k, y_{k+1}|\mathbf{X})}^{\text{matrix of all pairwise marginal probabilities}}) \\ &= - \left(\underbrace{\mathrm{freq}(y_k, y_{k+1})}_{\text{matrix of all pairwise label frequencies}} - \sum_{k=1}^{K-1} \mathbf{p}(y_k, y_{k+1}|\mathbf{X}) \right) \end{aligned}$$

REGULARIZATION

Topics: regularization

- For regularization, we can use the same regularizers as for a non-sequential neural network
  - add a regularizing term for all connection matrices
  - do not regularize the bias vectors
- We could scale λ by the sequence size
- With the loss and regularization gradients, we have all the ingredients to perform stochastic gradient descent