Neural networks

Training CRFs - loss function
LINEAR CHAIN CRF

**Topics:** reminder of notation

- Then we have:
  \[
p(y|X) = \exp \left( \sum_{k=1}^{K} a_u(y_k) + \sum_{k=1}^{K-1} a_p(y_k, y_{k+1}) \right) / Z(X)
  \]
  where
  \[
  Z(X) = \sum_{y_1'} \sum_{y_2'} \cdots \sum_{y_K'} \exp \left( \sum_{k=1}^{K} a_u(y_k') + \sum_{k=1}^{K-1} a_p(y_k', y_{k+1}') \right)
  \]
- Two types of (log-)factors:
  - unary: \( a_u(y_k) = a^{(L+1,0)}(x_k) y_k + \)
    \[
    1_{k>1} a^{(L+1,-1)}(x_{k-1}) y_k + \\
    1_{k<K} a^{(L+1,+1)}(x_{k+1}) y_k
    \]
  - 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
    \[
    \arg\min_{\theta} \frac{1}{T} \sum_{t} l(f(X^{(t)}; \theta), y^{(t)}) + \lambda \Omega(\theta)
    \]
    - $l(f(X^{(t)}; \theta), y^{(t)})$ is a loss function
    - $\Omega(\theta)$ is a regularizer (penalizes certain values of $\theta$)

- 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 \( \theta \)
  - for N iterations
    - for each training example \((X^{(t)}, y^{(t)})\)
      - \( \Delta = -\nabla_{\theta} l(f(X^{(t)}; \theta), y^{(t)}) - \lambda \nabla_{\theta} \Omega(\theta) \)
      - \( \theta \leftarrow \theta + \alpha \Delta \)

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

Topics: loss function for sequential classification with CRF

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

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

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

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