Recitation 3

Neural CRF - preventing numerical instability

Chaitanya Ahuja

Carnegie Mellon University

Table of contents

Introduction

Neural CRF

- Neural CRFs and CRFs are practically the same

Neural CRF

- Neural CRFs and CRFs are practically the same
- except emission function is estimated by a neural network (more precisely an RNN)

Neural CRF

- Neural CRFs and CRFs are practically the same
- except emission function is estimated by a neural network (more precisely an RNN)
    - $\eta(x_i|y_i) = \text{LSTM}(y_i, \mathbf{x}, i)$

Neural CRF

- Neural CRFs and CRFs are practically the same
- except emission function is estimated by a neural network (more precisely an RNN)
  - $\eta(x_i|y_i) = \text{LSTM}(y_i, \mathbf{x}, i)$
- We have a transition matrix $\mathbf{A}$ which captures the score of transitioning from one tag to another

Neural CRF

- Neural CRFs and CRFs are practically the same
- except emission function is estimated by a neural network (more precisely an RNN)
    - $\eta(x_i|y_i) = \text{LSTM}(y_i, \mathbf{x}, i)$
- We have a transition matrix **A** which captures the score of transitioning from one tag to another
    - $\gamma(y_i|y_{i-1}) = \mathbf{A}_{y_i, y_{i-1}}$

Cost Function

Cost Function

$$\max_{w} p(\mathbf{y}|\mathbf{x}; w)$$

Cost Function

$$\max_{w} p(\mathbf{y}|\mathbf{x}; w)$$

$$\equiv \min_{w} \left[- \log p(\mathbf{y}|\mathbf{x}; w)\right]$$

Cost Function

$$
\begin{aligned}
& \max_{w} p(\mathbf{y}|\mathbf{x}; w) \\
\equiv & \min_{w} \left[- \log p(\mathbf{y}|\mathbf{x}; w)\right] \\
\equiv & \min_{w} \log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right] - \log \left[\psi(\mathbf{y}; w)\right]
\end{aligned}
$$

Cost Function

$$
\begin{aligned}
& \max_w p(\mathbf{y}|\mathbf{x}; w) \\
\equiv\ & \min_w \left[-\log p(\mathbf{y}|\mathbf{x}; w)\right] \\
\equiv\ & \min_w \log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] - \log\left[\psi(\mathbf{y}; w)\right]
\end{aligned}
$$

given,

$$
p(\mathbf{y}|\mathbf{x}; w) = \frac{\psi(\mathbf{y}; w)}{\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)}
$$

Cost Function

$$
\begin{aligned}
&\max_{w} p(\mathbf{y}|\mathbf{x}; w) \\
&\equiv \min_{w} \left[-\log p(\mathbf{y}|\mathbf{x}; w)\right] \\
&\equiv \min_{w} \log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] - \log\left[\psi(\mathbf{y}; w)\right]
\end{aligned}
$$

given,

$$
p(\mathbf{y}|\mathbf{x}; w) = \frac{\psi(\mathbf{y}; w)}{\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)}
$$

where,

$$
\psi(\mathbf{y}; w) = \psi(\mathbf{y}) = \prod_{i=1}^{n} \psi(y_i, y_{i-1}, x_i)
$$

Cost Function

$$
\begin{aligned}
&\max_w p(\mathbf{y}|\mathbf{x}; w) \\
\equiv &\min_w \left[- \log p(\mathbf{y}|\mathbf{x}; w)\right] \\
\equiv &\min_w \log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] - \log \left[\psi(\mathbf{y}; w)\right]
\end{aligned}
$$

given,

$$
p(\mathbf{y}|\mathbf{x}; w) = \frac{\psi(\mathbf{y}; w)}{\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)}
$$

where,

$$
\psi(\mathbf{y}; w) = \psi(\mathbf{y}) = \prod_{i=1}^{n} \psi(y_i, y_{i-1}, x_i)
$$

$$
\psi(y_i, y_{i-1}, x_i) = \exp(\Phi(y_i, y_{i-1}, x_i)) = \exp(\gamma(y_i|y_{i-1})\eta(x_i|y_i))
$$

Cost Function

$$\equiv \min_{w} \underbrace{\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right]}_{\text{Forward Algorithm}} \underbrace{- \log \left[\psi(\mathbf{y}; w) \right]}_{\text{Ground Truth Labels and Tokens}} \tag{1}$$

Cost Function

$$\equiv \min_{w} \underbrace{\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right]}_{\text{Forward Algorithm}} \underbrace{- \log \left[\psi(\mathbf{y}; w) \right]}_{\text{Ground Truth Labels and Tokens}} \tag{1}$$

- Why should we use log?
  - To prevent underflow as we are maximising the likelihood of a probability distribution

Cost Function

$$\equiv \min_{w} \underbrace{\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right]}_{\text{Forward Algorithm}} \underbrace{- \log\left[\psi(\mathbf{y}; w)\right]}_{\text{Ground Truth Labels and Tokens}} \quad (1)$$

- Why should we use log?
    - To prevent underflow as we are maximising the likelihood of a probability distribution

Forward Algorithm

Forward Algorithm

$$\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right]$$

Forward Algorithm

$$\log\left[\sum_{\forall \mathbf{y}'}\psi(\mathbf{y}'; w)\right]$$

$$= \log\left[\sum_{\forall \mathbf{y}'_n}\sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_1}\psi(y'_1, y'_0, x_1)\psi(y'_2, y'_1, x_2)\dots\psi(y'_{n+1}, y'_n, x_{n+1})\right]$$

Forward Algorithm

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right]$$

$$= \log\left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_1} \underbrace{\psi(y'_1, y'_0, x_1)}_{\alpha_1(y'_1)} \psi(y'_2, y'_1, x_2) \dots \psi(y'_{n+1}, y'_n, x_{n+1})\right]$$

Forward Algorithm

$$\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right]$$

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_1} \underbrace{\psi(y'_1, y'_0, x_1)}_{\alpha_1(y'_1)} \psi(y'_2, y'_1, x_2) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_1} \alpha_1(y'_1) \psi(y'_2, y'_1, x_2) \psi(y'_3, y'_2, x_2) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

$$\vdots$$

Forward Algorithm

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right]$$

$$= \log\left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_1} \underbrace{\psi(y'_1, y'_0, x_1)}_{\alpha_1(y'_1)} \psi(y'_2, y'_1, x_2) \dots \psi(y'_{n+1}, y'_n, x_{n+1})\right]$$

$$= \log\left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \underbrace{\sum_{\forall \mathbf{y}'_1} \alpha_1(y'_1)\psi(y'_2, y'_1, x_2)}_{\alpha_2(y'_2)} \psi(y'_3, y'_2, x_2) \dots \psi(y'_{n+1}, y'_n, x_{n+1})\right]$$

$$\vdots$$

Forward Algorithm (contd.)

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \sum_{\forall \mathbf{y}'_2} \alpha_2(y'_2) \psi(y'_3, y'_2, x_3) \psi(y'_3, y'_2, x_3) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

Forward Algorithm (contd.)

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \underbrace{\sum_{\forall \mathbf{y}'_2} \alpha_2(y'_2)\psi(y'_3, y'_2, x_3)}_{\alpha_3(y'_3)} \psi(y'_3, y'_2, x_3) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

Forward Algorithm (contd.)

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \underbrace{\sum_{\forall \mathbf{y}'_2} \alpha_2(y'_2)\psi(y'_3, y'_2, x_3)}_{\alpha_3(y'_3)} \psi(y'_3, y'_2, x_3) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

$$\vdots$$

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \alpha_n(y'_n)\psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

Forward Algorithm (contd.)

$$= \log \left[\sum_{\forall \mathbf{y}'_n} \sum_{\forall \mathbf{y}'_{n-1}} \dots \underbrace{\sum_{\forall \mathbf{y}'_2} \alpha_2(y'_2)\psi(y'_3, y'_2, x_3)}_{\alpha_3(y'_3)} \psi(y'_3, y'_2, x_3) \dots \psi(y'_{n+1}, y'_n, x_{n+1}) \right]$$

$$\vdots$$

$$= \log \left[\underbrace{\sum_{\forall \mathbf{y}'_n} \alpha_n(y'_n)\psi(y'_{n+1}, y'_n, x_{n+1})}_{\alpha_n(y'_n)} \right]$$

Forward Algorithm (contd.)

Hence,

$$\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right] = \log \left[\alpha_n(y'_n) \right]$$

Forward Algorithm (contd.)

Hence,

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] = \log\left[\alpha_n(y'_n)\right]$$

where,

$$\log\left[\alpha_i(y'_{i-1})\right] = \log\left[\sum_{\forall y'_i} \alpha_{i-1}(y'_{i-1})\psi(y'_i, y'_{i-1}, x_i)\right]$$

Forward Algorithm (contd.)

Hence,

$$\log \left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w) \right] = \log \left[\alpha_n(y'_n) \right]$$

where,

$$\begin{aligned} \log \left[\alpha_i(y'_{i-1}) \right] &= \log \left[\sum_{\forall y'_i} \alpha_{i-1}(y'_{i-1}) \psi(y'_i, y'_{i-1}, x_i) \right] \\ &= \log \left[\sum_{\forall y'_{i-1}} \exp^{\log \alpha_{i-1}(y'_{i-1})} \exp^{\Phi(y'_i, y'_{i-1}, x_i)} \right] \end{aligned}$$

Forward Algorithm (contd.)

Hence,

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] = \log\left[\alpha_n(y'_n)\right]$$

where,

$$\begin{aligned}\log\left[\alpha_i(y'_{i-1})\right] &= \log\left[\sum_{\forall y'_i} \alpha_{i-1}(y'_{i-1})\psi(y'_i, y'_{i-1}, x_i)\right] \\ &= \log\left[\sum_{\forall y'_{i-1}} \exp^{\log \alpha_{i-1}(y'_{i-1})} \exp^{\Phi(y'_i, y'_{i-1}, x_i)}\right]\end{aligned}$$

Forward Algorithm (contd.)

Hence,

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] = \log\left[\alpha_n(y_n')\right]$$

where,

$$\begin{aligned}
\log\left[\alpha_i(y_{i-1}')\right] &= \log\left[\sum_{\forall y_i'} \alpha_{i-1}(y_{i-1}')\psi(y_i', y_{i-1}', x_i)\right] \\
&= \log\left[\sum_{\forall y_{i-1}'} \exp^{\log \alpha_{i-1}(y_{i-1}')} \exp^{\Phi(y_i', y_{i-1}', x_i)}\right] \\
&= \log\left[\sum_{\forall y_{i-1}'} \exp^{\zeta(y_i', y_{i-1}', x_i)}\right]
\end{aligned}$$

Forward Algorithm (contd.)

Hence,

$$\log\left[\sum_{\forall \mathbf{y}'} \psi(\mathbf{y}'; w)\right] = \log\left[\alpha_n(y'_n)\right]$$

where,

$$\begin{aligned}
\log\left[\alpha_i(y'_{i-1})\right] &= \log\left[\sum_{\forall y'_i} \alpha_{i-1}(y'_{i-1})\psi(y'_i, y'_{i-1}, x_i)\right] \\
&= \log\left[\sum_{\forall y'_{i-1}} \exp^{\log \alpha_{i-1}(y'_{i-1})} \exp^{\Phi(y'_i, y'_{i-1}, x_i)}\right] \\
&= \log\left[\sum_{\forall y'_{i-1}} \exp^{\zeta(y'_i, y'_{i-1}, x_i)}\right]
\end{aligned}$$

We need a good way to estimate **Log Sum of Exponents**

Numerical Stability

Log Sum of Exponents or Sloppy Log

Consider,

$$\log(\exp^{a} + \exp^{b})$$

Log Sum of Exponents or Sloppy Log

Consider,

$$\log(\exp^a + \exp^b)$$

If a or b have very low negative values, their exponents could be very small causing underflow.

Log Sum of Exponents or Sloppy Log

Consider,

$$\log(\exp^a + \exp^b)$$

If a or b have very low negative values, their exponents could be very small causing underflow.

But, if we re-write the equation as:

Log Sum of Exponents or Sloppy Log

Consider,

$$\log(\exp^{a} + \exp^{b})$$

If a or b have very low negative values, their exponents could be very small causing underflow.
But, if we re-write the equation as:

$$\max(a, b) + \log(\exp^{a-\max(a,b)} + \exp^{b-\max(a,b)})$$

Log Sum of Exponents or Sloppy Log

Consider,

$$\log(\exp^a + \exp^b)$$

If a or b have very low negative values, their exponents could be very small causing underflow.
But, if we re-write the equation as:

$$\underbrace{\max(a,b)}_{\text{No } \textbf{log} \text{ or } \textbf{exp}} + \log(\exp^{\overbrace{a-\max(a,b)}^{\leq 0}} + \exp^{\overbrace{b-\max(a,b)}^{\leq 0}})$$

Sloopy log for a vector

Given,

$$\log \left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})} \right]$$

Sloopy log for a vector

Given,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right]$$

Let,

$$\zeta_m = \max_{y_i'} \zeta(y_{i+1}', y_i', x_{i+1})$$

Sloopy log for a vector

Given,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right]$$

Let,

$$\zeta_m = \max_{y_i'} \zeta(y_{i+1}', y_i', x_{i+1})$$

Hence,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right] = \zeta_m + \log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1}) - \zeta_m}\right]$$

Sloopy log for a vector

Given,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right]$$

Let,

$$\zeta_m = \max_{y_i'} \zeta(y_{i+1}', y_i', x_{i+1})$$

Hence,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right] = \zeta_m + \log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1}) - \zeta_m}\right]$$

This formulation is vectorizable.

Sloopy log for a vector

Given,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right]$$

Let,

$$\zeta_m = \max_{y_i'} \zeta(y_{i+1}', y_i', x_{i+1})$$

Hence,

$$\log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1})}\right] = \zeta_m + \log\left[\sum_{\forall y_i'} \exp^{\zeta(y_{i+1}', y_i', x_{i+1}) - \zeta_m}\right]$$

This formulation is vectorizable.

> *Running* **for** *loops where vector operations are possible is blasphemy.*
>
> — *Chaitanya A. 2018*

Semi Rings

It might be useful to think of log as a different semi-ring.

$$a \bigoplus b = \log(\exp^a + \exp^b)$$

$$a \bigotimes b = \log\left(\exp^a \exp^b\right) = a + b$$

Summary

Summary

- Minimizing for a negative log-likelihood cost function prevents underflow due to small values of probability.

Summary

- Minimizing for a negative log-likelihood cost function prevents underflow due to small values of probability.
- Use **Sloppy log** to prevent underflow.

Summary

- Minimizing for a negative log-likelihood cost function prevents underflow due to small values of probability.
- Use **Sloppy log** to prevent underflow.
- **Sloppy log** is vectorizable.

Questions ?