M. Rautenhaus and P. Austin

Neural Network Satellite Retrievals of Nocturnal Stratocumulus Cloud Properties

CERES Victoria
November 16, 2007
Outline/Objectives

• Determine $\tau/\text{reff}$ for a nocturnal stratocumulus case with ship tracks and pockets of open cells (DYCOMS II July 11, 2001).

Test Scene: DYCOMS-II RF02, July 11, 2001

- nocturnal flights
- horizontal flight circles at cloud top and bottom
- five hour time lag between satellite overpass and in-situ measurements
Adiabatic/constant N model compared to DYCOMS sounding

- droplet concentration [cm⁻³]
- LWC [g/m³]
- height [km]
- temperature [K]
- particle radius [µm]

spec. humidity [g/kg]
Lookup Table: MODIS Ch. 20, 31, 32

Cloud top temperature = 285 K, cloud top pressure = 900 hPa.
Retrievals: Cloud Top Effective Radius

![Image of histogram and map showing cloud top effective radius in micrometers with ship tracks indicated.]
Retrievals: Cloud Top Temperature

- Graph showing retrieved cloud top temperature [K] with values ranging from 283 to 286 K.
- Color map indicating cloud top temperature [K] with a gradient from 280 to 290 K.
- In-situ temperature [K] graph with values ranging from 283 to 286 K.
Retrievals: Optical Thickness
Standard retrieval:

- Given a forward radiative transfer model $y(x)$ that maps atmospheric properties

$$x = \{ \tau, r_{\text{eff}}, \text{lwp}, T_{\text{cld}}, \text{overlying atmosphere} \ldots \}$$  \hspace{1cm} (1)

onto radiances (targets) $t$

$$t = (I_{3.7}, I_{11}, I_{12}, \ldots) = y(x)$$  \hspace{1cm} (2)

- Find cloud properties $x_*$ for a radiance measurement $t_*$ that minimizes a cost function:

$$E(x) = \sum_{i=1}^{d} (y(x_i) - t_i)^2.$$  \hspace{1cm} (3)
Inversion with Neural Networks

![Diagram of a neural network with inputs, biases, and outputs connecting through layers](image)

- **Inputs (radiances)**
- **Outputs (Cloud properties)**
- **Bias nodes**
- **Hidden layers (z_i)**
- Weights (w_i,j)

Rautenhaus and Austin :: Neural Network Satellite Retrievals of Nocturnal Stratocumulus Cloud Properties
Bayesian neural net:

- Given a training dataset $D$ consisting of TOA radiances $\mathbf{x}$ and cloud targets $\mathbf{t} = \{\tau, r_{\text{eff}}, T_{\text{cl}}\}$, find the underlying generator for the LUT, $y$, by choosing a set of network weights $\mathbf{w}$ that map $\mathbf{x}$ to $\mathbf{t}$:

$$y_k = \tilde{g} \left( \sum_{j=0}^{M} w_{j,k}^{(2)} \times g \left( \sum_{i=0}^{d} w_{i,j}^{(1)} x_i \right) \right). \quad (4)$$

- The set of optimal weights, $\mathbf{w}_*$, is the one that minimizes the cost function.

$$E = \frac{1}{2} \sum_{n=1}^{N} \{y(x^n; \mathbf{w}) - t^n\}^2. \quad (5)$$
What about Bayes?

There’s no guarantee that the optimal set of weights $w_*$ is the one that gives the best physical representation of the generator $y$. There is some probability distribution for the weights given the training set, given by Bayes theorem:

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}.$$  \hspace{1cm} (6)

which can be reduced to the product of two Gaussians:

$$p(w|D) \propto \exp \left( -\epsilon_D(w) - \epsilon_W(w) \right),$$  \hspace{1cm} (7)

where $\epsilon_D$ and $\epsilon_W$ are data and weights error functions.
Distribution of W11 for long and short training times
The Hessian and the Jacobian

- Working with $p(w|D)$: where do we get this PDF?
- Second order Taylor series expansion:

$$
\epsilon(w) = \epsilon(w^*) + b^T \cdot \Delta w + \frac{1}{2} \Delta w^T \cdot \tilde{H} \cdot \Delta w,
$$

where $\Delta w = w - w^*$. $b$ denotes the gradient of $E$ at $w^*$,

$$
b = \nabla \epsilon(w)|_{w=w^*} = 0,
$$

and the Hessian is given by

$$
\tilde{H} = \nabla \nabla \epsilon(w)|_{w=w^*}
$$

so that

$$
p(w|D) = \frac{1}{Z} \exp \left( -\frac{1}{2} \Delta w^T \cdot \tilde{H} \cdot \Delta w \right).$$
Distribution of the Jacobian for the July 11 scene

\[ J_{ij} = \frac{\partial y_j}{\partial x_i} \]
Spatial Distribution of Jacobian

\[ \frac{\partial r_{\text{eff}}}{\partial l_{3.7}} \]

Larger droplets, higher sensitivity

Smaller droplets, lower sensitivity
Network Architecture

- Jacobian point estimate - compare dependences with “known” values

- Example: brightness temperatures or brightness temperature differences as inputs?

\[ \approx 2.5 \, \mu m / K \]

<table>
<thead>
<tr>
<th>( \partial r_{eff} )</th>
<th>( \frac{[\mu m/K]}{} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \partial BT(3.7) )</td>
<td>( 0.22 \pm 4.72 )</td>
</tr>
<tr>
<td>( \partial BT(11) )</td>
<td>( -20.16 \pm 20.92 )</td>
</tr>
<tr>
<td>( \partial BT(12) )</td>
<td>( 19.11 \pm 21.67 )</td>
</tr>
<tr>
<td>( \partial T_{sfc} )</td>
<td>( 1.18 \pm 3.37 )</td>
</tr>
</tbody>
</table>
Scene Jacobian

<table>
<thead>
<tr>
<th></th>
<th>$\partial r_{eff}$ / $[\mu m/K]$</th>
<th>$\partial T$ / $[K/K]$</th>
<th>$\partial \tau$ / $[K^{-1}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\partial B T(3.7)$</td>
<td>$1.47 \pm 1.32$</td>
<td>$-0.22 \pm 0.56$</td>
<td>$-0.94 \pm 1.53$</td>
</tr>
<tr>
<td>$\partial B T(11)$</td>
<td>$-1.23 \pm 1.37$</td>
<td>$0.56 \pm 0.80$</td>
<td>$1.07 \pm 1.69$</td>
</tr>
<tr>
<td>$\partial B T D(11-12)$</td>
<td>$-7.75 \pm 6.11$</td>
<td>$-2.39 \pm 2.32$</td>
<td>$-5.24 \pm 7.86$</td>
</tr>
<tr>
<td>$\partial T_{sfc}$</td>
<td>$-0.16 \pm 0.54$</td>
<td>$0.62 \pm 0.43$</td>
<td>$-0.04 \pm 0.82$</td>
</tr>
</tbody>
</table>

- Use mean Jacobian to estimate
  
  1) average dependences
  2) ill-conditioning of the problem
  3) sensitivities to inputs
  4) importance of inputs
### normalised mean Jacobian - importance of inputs

<table>
<thead>
<tr>
<th></th>
<th>$\partial r_{eff}$ / [µm]</th>
<th>$\partial T$ / [K]</th>
<th>$\partial \tau$ / [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\partial BT(3.7)$</td>
<td>2.36</td>
<td>-0.35</td>
<td>-1.51</td>
</tr>
<tr>
<td>$\partial BT(11)$</td>
<td>-1.82</td>
<td>0.91</td>
<td>1.74</td>
</tr>
<tr>
<td>$\partial BTD(11-12)$</td>
<td>-1.34</td>
<td>-0.42</td>
<td>-0.92</td>
</tr>
<tr>
<td>$\partial T_{sfc}$</td>
<td>-0.20</td>
<td>0.78</td>
<td>-0.04</td>
</tr>
</tbody>
</table>
Summary

• Neural net able to retrieve reff, Tcld, tau for nocturnal stratocumulus case

• Bayesian approach uses the Hessian of the error function to estimate weight distribution, distribution of the network sensitivities (Jacobian)

• More work to do on generating a robust network (network ensembles), improving the retrieval via the prior weight distribution, tracking diurnal changes.