Towards physics-based machine learning for land surface modeling:
The case of land-atmosphere interactions.

Andrew Bennett
andrbenn@email.arizona.edu
March 29, 2022
The main idea:
Put the neural network *inside* of the hydrologic model!
We reframe things in terms of energy fluxes

\[ R_n = H + LE + G \]
We gathered data from 60 FluxNet sites, totalling over 500 site-years of half-hourly data.

We used the SUMMA hydrologic modeling framework for all of our configurations.
Model configurations

SUMMA-x (single timestep)

Inputs
- Parameters
- Meteorologic forcing data

Compute Fluxes
- Surface runoff
- Turbulent heat flux (see panel b for options)
- Snowmelt

Update States
- Canopy temperature
- Canopy storage
- Soil storage

Update time step
Model output
Model configurations

SUMMA-x (single timestep)

Inputs
- Parameters
- Meteorologic forcing data

Compute Fluxes
- Surface runoff
- Turbulent heat flux (see panel b for options)
- Snowmelt

Update States
- Canopy temperature
- Canopy storage
- Soil storage

Update time step

Model output

Turbulent heat flux options

SA

Parameters → Canopy LE
Met. Forcings → Ground LE
SUMMA Fluxes → Canopy H
SUMMA States → Ground H
Model configurations

SUMMA-x (single timestep)

Inputs
- Parameters
- Meteorologic forcing data

Compute Fluxes
- Surface runoff
- Turbulent heat flux (see panel b for options)
- Snowmelt

Update States
- Canopy temperature
- Canopy storage
- Soil storage

Update time step

Model output

Turbulent heat flux options
- Parameters
- Met. Forcings
- SUMMA Fluxes
- SUMMA States

SA
- Canopy LE
- Ground LE
- Canopy H
- Ground H

NN1W
- Parameters
- Met. Forcings
- Bulk LE
- Bulk H
Model configurations

SUMMA-x (single timestep)

Inputs
- Parameters
- Meteorologic forcing data

Compute Fluxes
- Surface runoff
- Turbulent heat flux
  (see panel b for options)
- Snowmelt

Update States
- Canopy temperature
- Canopy storage
- Soil storage

Model output

Update time step

Turbulent heat flux options

SA
- Parameters
- Met. Forcings
- SUMMA Fluxes
- SUMMA States
- Canopy LE
- Ground LE
- Canopy H
- Ground H

NN1W
- Parameters
- Met. Forcings
- SUMMA States
- Bulk LE
- Bulk H

NN2W
- Parameters
- Met. Forcings
- SUMMA States
- Bulk LE
- Bulk H
Both neural network parameterizations outperformed the standalone model, for both latent and sensible heat
Inclusion of soil states in **NN2W** improves long-term water balance over **NN1W**

1-1 line is “perfect”
One of the major shortcomings is a mismatch between process fidelity and the observed data for training.
The land surface is heterogeneous!

Flux towers measure bulk fluxes

But we want to model the various components

Without fancy techniques supervised machine learning can only learn bulk fluxes from observations then
So, we’ve got tradeoffs

<table>
<thead>
<tr>
<th>Superior performance</th>
<th>Process based model</th>
<th>Machine learned model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>checkmark</td>
</tr>
<tr>
<td>Process fidelity</td>
<td>checkmark</td>
<td>x</td>
</tr>
</tbody>
</table>
Why don't we have both?
So, we’ve got tradeoffs

<table>
<thead>
<tr>
<th>Superior performance</th>
<th>Process based model</th>
<th>Machine learned model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

Process fidelity

✓ ✓
Why don’t “physics” based models perform well?

These bulk transfer equations are very common in hydrologic and land surface modeling:

- Andreadis et al., 2009
- Bonan, 1991
- Inclan and Forkel, 1995
- Sellers et al., 1986
- Mahat et al., 2013
- Clark et al., 2015
- …

Sensible heat fluxes:

\[ Q_{h}^{\text{veg}} = -\rho_{\text{air}} c_p C_h^{\text{veg}} (T^{\text{veg}} - T^{\text{cas}}) \]

\[ Q_{h}^{\text{sfc}} = -\rho_{\text{air}} c_p C_h^{\text{sfc}} (T^{\text{sfc}} - T^{\text{cas}}) \]

Latent heat fluxes:

\[ Q_{\text{evap}}^{\text{veg}} = -\frac{L_{\text{vap}} \rho_{\text{air}} \varepsilon}{P_{\text{air}}} C_{\text{evap}}^{\text{veg}} \left[ e_{\text{sat}} (T^{\text{veg}}) - e^{\text{cas}} \right] \]

\[ Q_{\text{trans}}^{\text{veg}} = -\frac{L_{\text{vap}} \rho_{\text{air}} \varepsilon}{P_{\text{air}}} C_{\text{trans}}^{\text{veg}} \left[ e_{\text{sat}} (T^{\text{veg}}) - e^{\text{cas}} \right] \]

\[ Q_{l}^{\text{sfc}} = -\frac{L_{\text{vap}} \rho_{\text{air}} \varepsilon}{P_{\text{air}}} C_{\text{water}}^{\text{sfc}} \left[ \phi_{\text{hum}} e_{\text{sat}} (T^{\text{sfc}}) - e^{\text{cas}} \right] \]
Why don’t “physics” based models perform well?

These consist of three main parts

1. Constants & parameters
2. Temperature or moisture gradients
3. Conductance terms
Why don’t “physics” based models perform well?

These consist of three main parts

1. Constants & parameters
2. Temperature or moisture gradients
3. Conductance terms
Why don’t “physics” based models perform well?

These consist of three main parts

1. Constants & parameters
2. Temperature or moisture gradients
3. Conductance terms

I’m going to argue these are either:

1. Pretty well known
2. Parts of other processes
Why don’t “physics” based models perform well?

These consist of three main parts

1. Constants & parameters
2. Temperature or moisture gradients
3. Conductance terms

And likewise, this is where the model uncertainty really is...
The hybrid neural network architecture

This work is heavily inspired by:
The hybrid neural network architecture

Technical note: This “layer” has no trainable weights - it just encodes the equations in a differentiable form.

We’re still able to outperform a calibrated PBHM using the same bulk transfer equations.*

*other pure ML based approaches outperform this but that’s asking a different question
Walnut Gulch near Tombstone, AZ (US-Whs) shows ground component is largest
Blodgett Forest near Sacramento, CA (US-Blo) shows vegetation components are largest
Mixed forest near Vielsalm, Belgium shows a larger mixture between components
The overall partitioning matches physical intuition to a first order

Note: values can be <0 and >1 because condensation exists
The overall partitioning matches physical intuition to a first order, mostly

Note: values can be $<0$ and $>1$ because condensation exists
We can also truncate the network to analyze the conductances!
Comparing conductances shows the network learns nonlinear behavior.
Conductances are not 1-1 with heat fluxes at Blodgett, CA
Wrapup and future work

We’ve quantified that a large amount of predictive performance is due to conductance terms.

Need methods/data to better constrain the partitioning, particularly at sites with human interventions, like croplands.

Coupling to the PBHM is still incomplete, but needed to analyze the effects on the full water cycle.

Lots of validation work is still needed to build confidence.
Thanks!

Questions?