HybridGR4J Model Pseudocode

Data:

Meteorological sequence for a gauged catchment: $[P^{(t)}, T^{(t)}, D^{(t)}, ...], t = 1, ..., T$

Observed runoff sequence: q_{obs_t}

Result:

parameters: $\theta^{\langle G \rangle} = \{X_1, X_2, X_3\}$; $\theta^{\langle N \rangle} = \{W_{uh1}, W_{uh2}\}$

Simulated runoff sequence: $\hat{q}_{sim_t}$

Stage 1: Model Initialization

1. Define PBM-RNN layer: $\hat{q}_{sim_t} = PBMRNN(P^{(t)}, T^{(t)}, D^{(t)}; \theta^{\langle G \rangle})$
2. Initialize $\theta^{\langle G \rangle}: X_1 \in [1,\ 2000];\ X_2 \in [-20,\ 20];\ X_3 \in [1,\ 300]$
3. Compute potential evapotranspiration: $PET^{(t)} = Hamon\left(T^{(t)}, D^{(t)}\right)$
4. Build input vector: $x_t = \left[P^{(t)},\ T^{(t)},\ PET^{(t)}\right]$
5. Initialize neural network parameters $\theta^{\langle N \rangle}$

Stage 2: Physical Modeling and Routing using PBM-RNN layer

For $t = 1$ to b:

1. Calculate $P_n^{(t)}$ and $E_n^{(t)}$ using Equation (1) and (2)
2. Calculate $P_s^{(t)}$ and $E_s^{(t)}$ using Equation (3) and (4)
3. Update $S^{(t)}$ in the production store using Equation (5)
4. Calculate $Perc^{(t)}$ using Equation (6)
5. Update $S^{(t)}$ using Equation (7)
6. Calculate $P_r^{(t)}$ using Equation (8)
7. Define unit hydrographs:

   $UH_1 = \exp\left(-i/5\right),\ i = 0...W_{uh1}$

   $UH_2 = \exp\left(-i/10\right),\ i = 0...W_{uh2}$

   Normalize UH_1, UH_2
8. Routing by unit hydrographs convolution:

   $$Q_9^{(t)} = 0.9 \times \sum_{i=1}^{W_{uh1}} UH_{1i} \times P_r^{(t-i+1)}$$

   $$Q_1^{(t)} = 0.1 \times \sum_{i=1}^{W_{uh2}} UH_{2i} \times P_r^{(t-i+1)}$$
9. Calculate $F^{(t)}$ using Equation (15)
10. Update $R^{(t)}$ in the routing store using Equation (16)
11. Calculate $P_r^{(t)}$ and $R^{(t)}$ using Equation (17) and (18)
12. Calculate total runoff $\hat{q}_{sim_t}$ using Equation(19) and (20)

Stage 3: Correction of initial simulated runoff using ConvNet

1.Build input features: $C_{input} = \left[P,\ T,\ PET, \ldots,\ \hat{q}_{sim}\right]$

2.ConvNet structure: $q_{sim} = ELU(W_2\ ELU(W_1 \cdot C_{input} + b_1) + b_2)$

Stage 4: HybridGR4J Model Training

Set hyper-parameters: Learning rate(η), Epochs(N_{epoch}), Batch size(b)

for epoch = 1, ..., N_{epoch} do

for batch = 1, ..., b do

1.Forward propagate:

$$\hat{q}_{sim} = PBMRNN\left(x;\ \theta^{<G>}, \theta^{<N>}\right)$$

$$C_{input} = \left[P,\ T,\ PET, \ldots,\ \hat{q}_{sim}\right]$$

$$q_{sim} = ConvNet(C_{input}; W_1, W_2, b_1, b_2)$$

2.Compute loss: $\mathcal{L} = 1 - NSE\left(q_{sim},\ q_{obs}\right)$

3.Backward propagate:

a) Compute gradients: $\nabla\theta^{<G>}$ and $\nabla\theta^{<N>}$

b) Update parameters:

$$\theta^{<G>} \leftarrow \theta^{<G>} - \eta\nabla\theta^{<G>}\mathcal{L}$$

$$\theta^{<N>} \leftarrow \theta^{<N>} - \eta\nabla\theta^{<N>}\mathcal{L}$$

end

end

Stage 5: Evaluation and output

1. Simulate runoff for the testing period: $q_{sim} = \text{HybridGR4J}\left(x\right)$
2. Evaluate model performance using NSE metric