Input Switched Affine Networks: An RNN Architecture Designed for Interpretability(ICML 2017)

Jakob N.Foerster, Justin Gilmer, Jascha Sohl-Dickstein, Jan Chorowski, David Sussillo

Nayeong Kim

Machine Learning Group, Department of Computer Science and Engineering, POSTECH, 77-Cheongam-ro, Nam-gu, Pohang-si 37673, Gyungsangbuk-do, Republic of Korea

kimnay@postech.ac.kr

What is interpretability in machine learning?

Interpretability

: Ability to explain or to present in understandable terms to a human.

- Artificial neural networks are not interpretable because human can understand configuration of hidden state.

Why interpretability? Incompleteness

The need for interpretability stems from an incompleteness in the problem formalization, creating a fundamental barrier to optimization and evaluation *(Been Kim)*.

- **Safety** - For complex tasks, the end-to-end system is almost never completely testable.
- **Scientific understanding** - The human's goal is to gain knowledge. We do not have a complete way of stating what knowledge is.
- **Ethics** - The human may want to guard against certain kinds of discrimination, and their notion of fairness may be too abstract to be completely encoded into the system
- **Mismatched objectives** - The agent's algorithm may be optimizing an incomplete objective— that is, a proxy function for the ultimate goal.
- **Multi-objective trade-offs** - Two well-defined desiderata in ML systems may compete with each other, such as privacy and prediction quality or privacy and nondiscrimination.

Interpreting neural network

What we can do with interpretable neural network.

- Can explain process of the solution is derived with interpreting latent variables.
- Can modify model after finding errors.
- It is possible to introduce artificial models to end-to-end problems, such as medical diagnosis, where errors are fatal.

Approaches to interpreting neural network

1. Train the network as normal, and then apply analysis techniques after training.
2. Build a neural network where interpretability is an explicit design constraint.

Input Switched Affine Networks (ISAN)

RNN without any explicit nonlinearities, but with input dependent recurrent weights.

- ISAN Model is defined as

$$\mathrm{h_t} = W_{x_t} h_{t-1} + b_{x_t}$$

$\mathrm{W_x}$, $\mathrm{b_x}$: transition matrix and a bias vector for a specific input x

$\mathrm{x_t}$: the input at time t

- The probabilities are computed as

$$\mathrm{p(x_{t+1})} = softmax(l_t)$$

$$\mathrm{l_t} = \mathrm{W}_{ro} h_t + b_{ro}$$

W_{ro}, b_{ro} : the readout weights and biases

l_t : the logit vector.

Input Switched Affine Networks (ISAN)

- ISAN Model is defined as

$$\mathrm{h_t} = W_{x_t} h_{t-1} + b_{x_t}$$

$\mathrm{W_x}$, $\mathrm{b_x}$: transition matrix and a bias vector for a specific input x

$\mathrm{x_t}$: the input at time t

EX) Input string: ((((((((([[[[[[ ))) aaaaaa]]]]]]]]] ( ) ( ) ( ) (

possible input : {‘(’, ‘)’, ‘[’, ‘]’, ‘a’}

$$\mathrm{W_x} = \{W_(, W_), W_[, W_], W_a\}$$

$$x_1 = '(', \qquad W_{x_1} = W_($$

Input Switched Affine Networks (ISAN)

With ISAN, we can analyze which factors were important in the past for determining the current character prediction.

$$h_t = \Sigma_{s=0}^{T}\left(\Pi_{s'=s+1}^{t} W_{X_{s'}}\right) b_{X_s}$$

$$l_t = b_{ro} + \Sigma_{s=0}^{t} \kappa_s^t$$

$$\kappa_s^t = W_{ro}\left(\Pi_{s'=s+1}^{t} W_{X_{s'}}\right) b_{X_s}$$

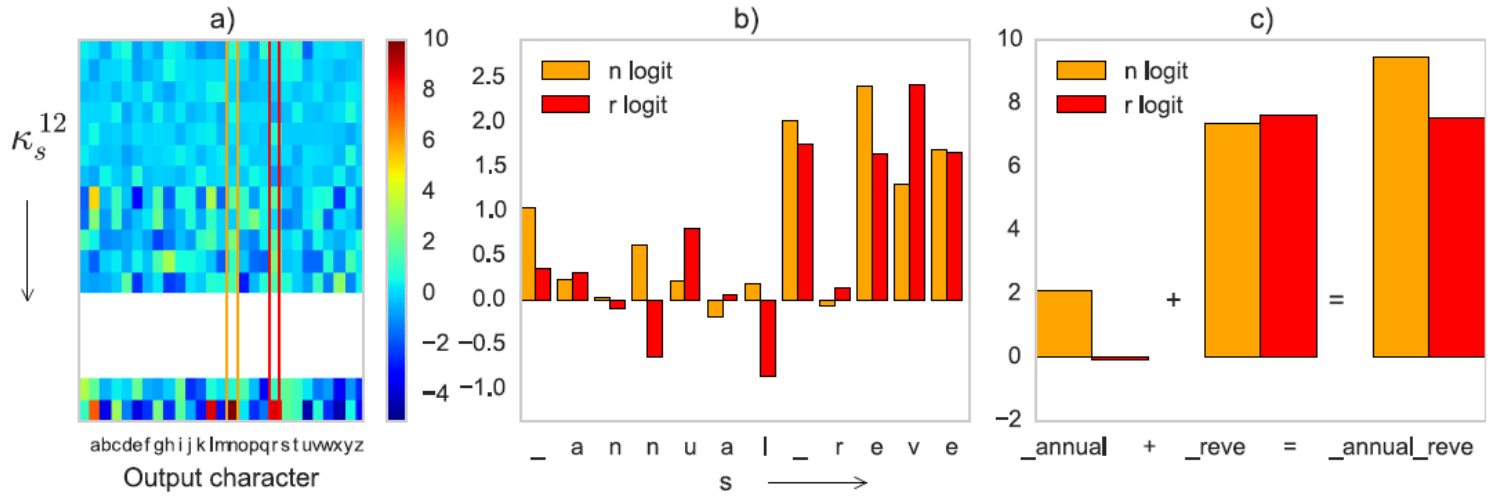
- κ_s^t is the contribution from time step s to the logits at time step t, and $\kappa_t^t = b_{X_t}$

  Ex) $\kappa_{'q'}^t$ refers to the contribution from the character 'q' in a string.

Example

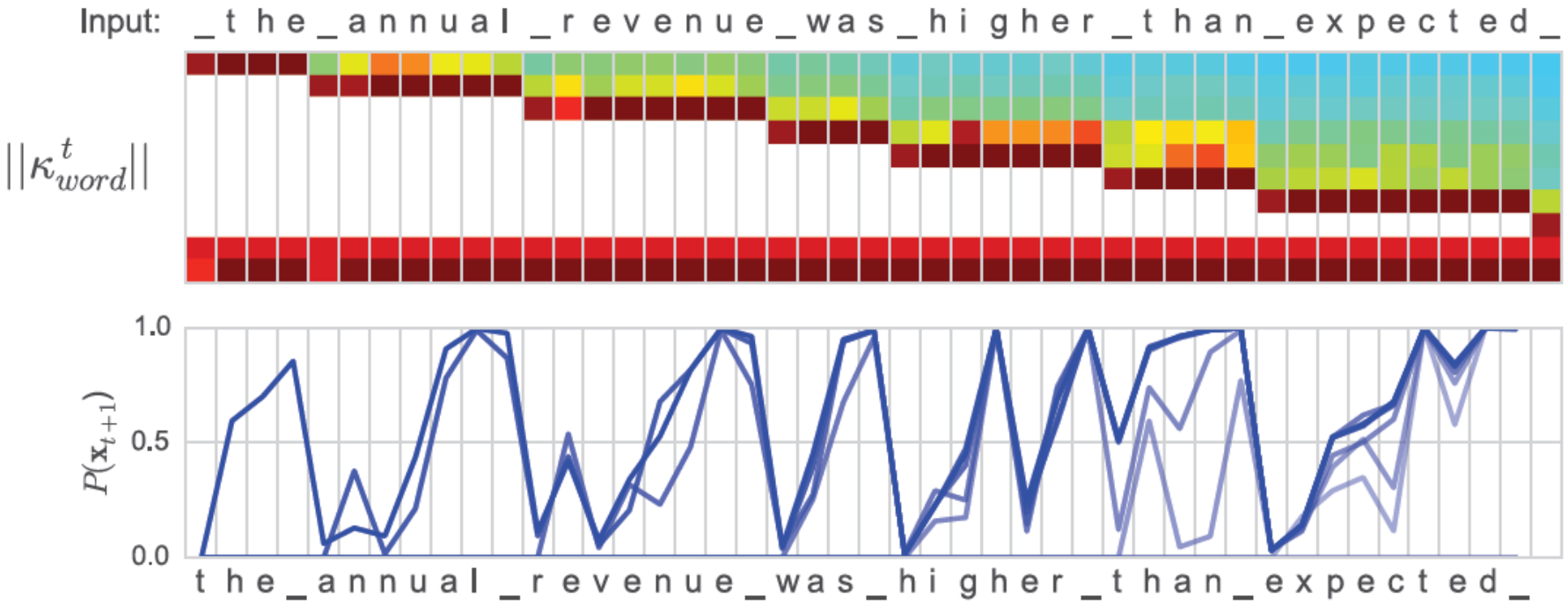
A detailed view of how past characters contribute to the logits predicting the next character.

- There are two competing options for the next letter in the word stem ‘reve’: either ‘revenue’ or ‘reverse’.
  - ✓ Without the contributions from ‘_annual’ the most likely decoding of the character after the second ‘e’ is ‘r’ (to form ‘reverse’),
  - ✓ While the contributions from ‘_annual’ tip the balance in favor of ‘n’, decoding to ‘revenue’.



Example

A detailed view of how past characters contribute to the logits predicting the next character.



Interpreting hidden state - Change of basis

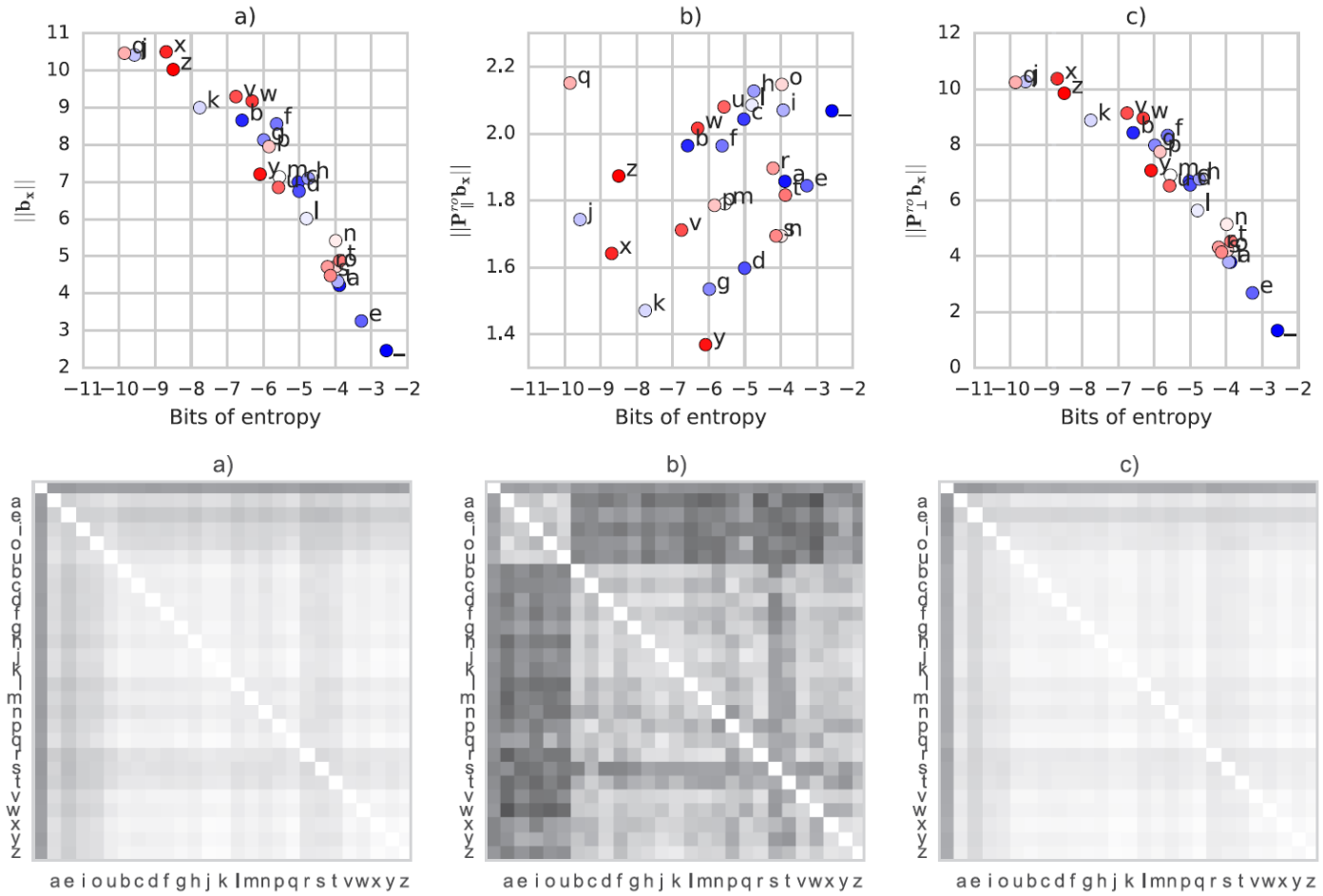
Change of basis

: Divide the latent space into a subspace $P_{||}^{ro}$ spanned by the rows of the readout matrix W_{ro}, and its orthogonal complement $P_{\perp}^{ro}$, which is computational space.

- Change of basis is not possible for other RNN architectures, since the action of the nonlinearity depends on the choice of basis.
- The readout dimension is determined by the number of types of input data.

  Ex) possible input : {‘(’, ‘)’, ‘[’, ‘]’, ‘a’}, readout dimension : $\mathbb{R}^5$

Interpreting hidden state - Change of basis



- The data correlation of the data is maintained at the computational dimension.
- This implies a connection between information and distance in the computational subspace of state space.

Example – Analysis of a parentheses counting task

- Analyze the task of counting the nesting levels of multiple parentheses types.
  - inputs : one-hot encoding of the different opening and closing parentheses (e.g. '(', ')', '[', ']') as well as a noise character (e.g. 'a').

    '((((((((([[[[[[ ))) aaaaaa]]]]]]]]] ( ) ( ) ( ) ('

  - outputs : one-hot encoding of the nesting level between (0-5) one set of counts for each parenthesis type.

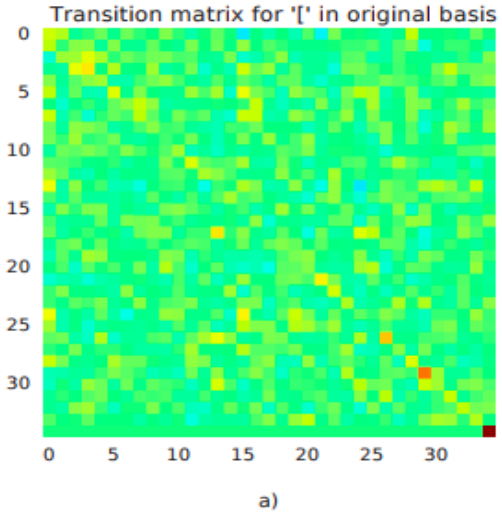
- Hidden state of ISAN can be projected into a readout space using change of basis and expressed in an interpretable form.

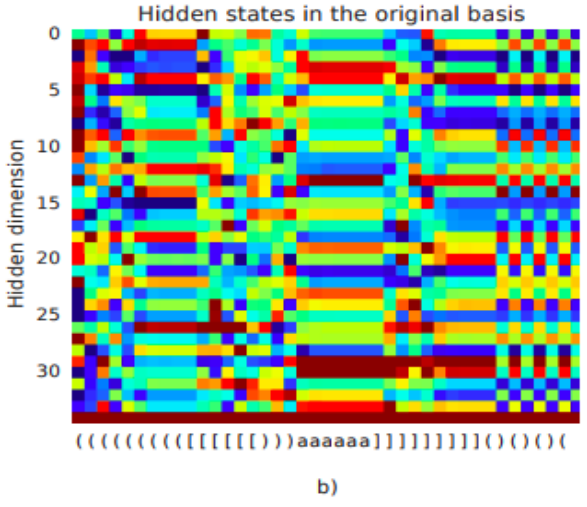
$$W' = ||P_{\perp}^{ro} W|| = \begin{bmatrix} W^{rr} & W^{rc} \\ W^{cr} & W^{cc} \end{bmatrix}$$

$$h' = ||P_{\perp}^{ro} h|| = \begin{bmatrix} h^{r} \\ h^{c} \end{bmatrix}$$

Example – Analysis of a parentheses counting task

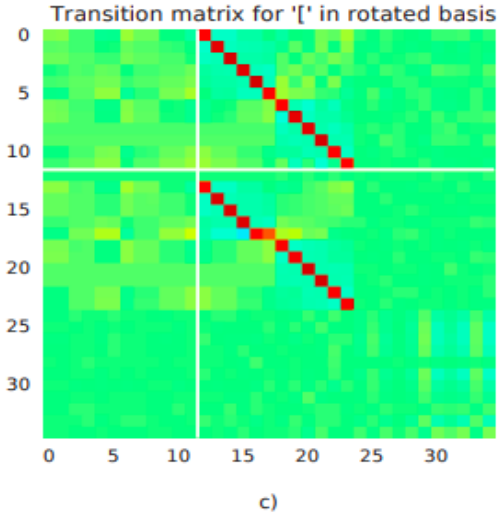
W

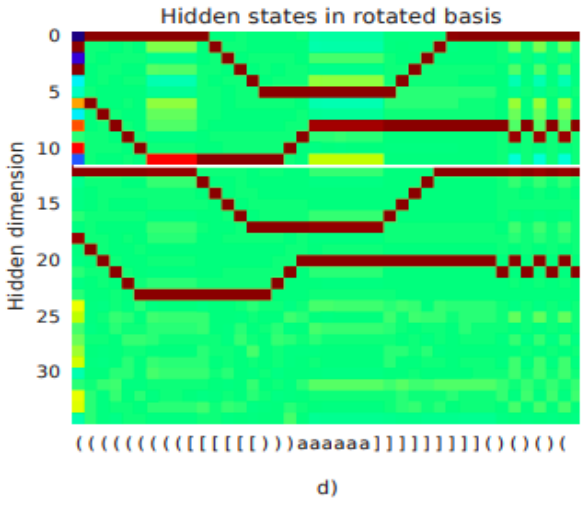




h

$W' = ||P_{\perp}^{ro} W|| = \begin{bmatrix} W^{rr} & W^{rc} \\ W^{cr} & W^{cc} \end{bmatrix}$

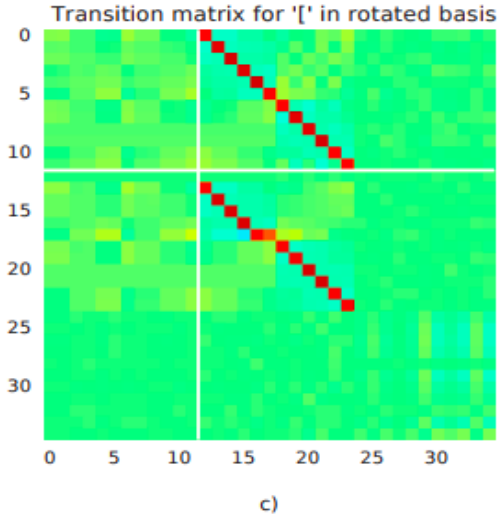




$h' = ||P_{\perp}^{ro} h|| = \begin{bmatrix} h^r \\ h^c \end{bmatrix}$

- visualization of the dynamics of an ISAN for the two parentheses counting task with 1 time lag (count either ’()’ or ’[]’ nesting levels with a one-step readout delay).

Example – Analysis of a parentheses counting task



$$W'_{[} = ||P_{\perp}^{ro} W_{[}|| = \begin{bmatrix} W_{[}^{rr} & W_{[}^{rc} \\ W_{[}^{cr} & W_{[}^{cc} \end{bmatrix}$$

$$h'_{t+1} = W'_t h'_t = \begin{bmatrix} W_{[}^{rr} h_t^r + W_{[}^{rc} h_t^c \\ W_{[}^{cr} h_t^r + W_{[}^{cc} h_t^c \end{bmatrix}$$

i) $W_{[}^{rr} = W_{[}^{cr} = 0$

: h_t^r has no influence on h_{t+1}.

$$h'_{t+1} = W'_t h'_t = \begin{bmatrix} 0 \cdot h_t^r + W_{[}^{rc} h_t^c \\ 0 \cdot h_t^r + W_{[}^{cc} h_t^c \end{bmatrix}$$

ii) i) AND $W_{[}^{rc} = W_{[}^{cc}$

: $h_t^r = h_{t-1}^c$

$$h'_{t+1} = W'_t h'_t = \begin{bmatrix} 0 \cdot h_t^r + W_{[}^{rc} h_t^c \\ 0 \cdot h_t^r + W_{[}^{rc} h_t^c \end{bmatrix}$$

Example – Analysis of a parentheses counting task