# Hierarchical Variational Recurrent Autoencoder with Top-Down prediction

Kosuke Miyoshi (2,3,4), Naoya Arakawa (2,4), Hiroshi Yamakawa (1,2,4)
The University of Tokyo(1) The Whole Brain Architecture Initiative(2),
Narrative Nights Inc.(3), Dwango Artificial Intelligence Laboratory(4)

## Abstract

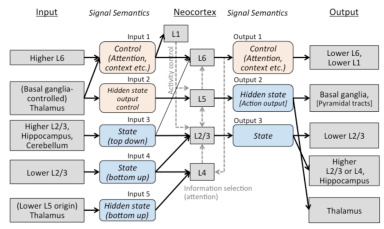
**A framework that defines functions and interface semantics of the cortical micro-circuit was previously proposed as The Coritical Master Algorithm Framework (Yamakawa 2017), and a deep learning model that supports this framework is demanded.**
**We chose the VRNN network model (Variational Recurrent Neural Network) to add a hierarchical feature. We implemented time series prediction with top-down signal, and found the representation in the lower layers became sparsely disentangled, so that the fundamental factors in the sensor input were extracted. We also discuss the ability of functional differentiation with HVRNN.**

## Cortical Master Algorithm Framework

A framework that defines the functions and interface semantics of cortical micro-circuits was previously proposed as the Cortical Master Algorithm Framework (MAF).

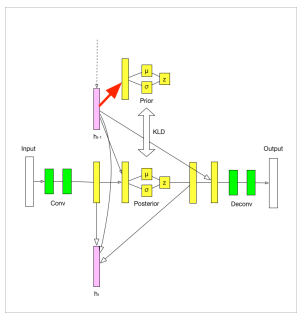
MAF requrires functional and structural prerequisites as below.

- Dimension reduction
- Unsupervised learning
- Time series prediction
- Disentanglement
- Generative model
- Sparsity
- Internal state
- Hierarchy
- Prediction with top-down signal
- Attention with control signal

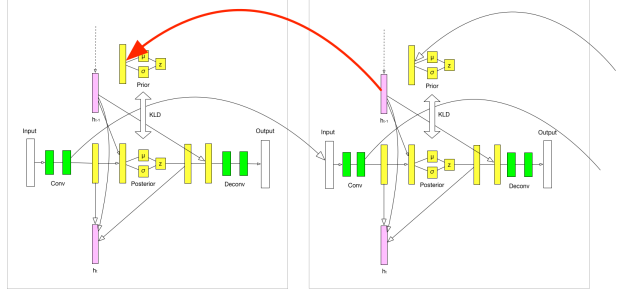



## Hierarchical VRNN (HVRNN)

### VRNN




### HVRNN (Proposed model)

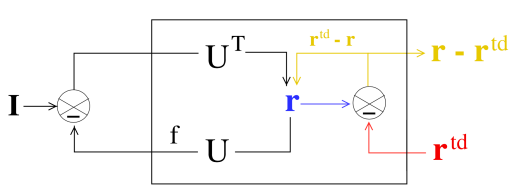



VRNN (Variational Recurrent Neural Netowrk) is a deep learning model that is capable of dimension reduction, unsupervised learning, time series prediction, generative model, and internal state. VRNN has the structure in which time series prediction is processed with latent variable **z**. Latent variable **z**'s approximated posterior distribution $\mathbf{q(z_t|x_{\leq t}, z_{<t})}$ is predicted by the prior distribution $\mathbf{p(z_t|x_{<t}, z_{<t})}$.

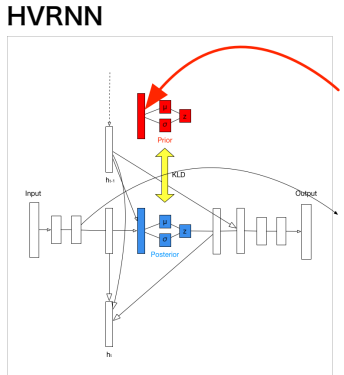
By following the MAF requirements, we propose a Hierarchical VRNN model (HVRNN) that adds hierarchy to the VRNN. In HVRNN we implemented time series prediction in lower layers with top-down signals.

## Predictive Coding

**Predictive Coding (Rao & Ballad, 1999)**




**HVRNN**



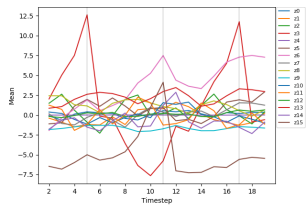

The top-down prediction in HVRNN can be seen as the predictive coding (Rao 1999). Predictive coding calculates the difference between the representation in the lower layer (**r**) and the top-down prediction ($\mathbf{r_{td}}$), and sends the prediction error ($\mathbf{r_{td}-r}$, $\mathbf{r-r_{td}}$) to both lower and higher layers to minimize the error.

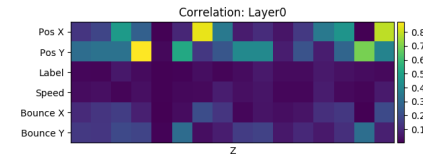
HVRNN doesn't have these explicit prediction error signal feedback. However, through the back-propagation training, it contains exactly the same error feedback by minimizing KL divergence between the posterior and the prior.

## Experiments

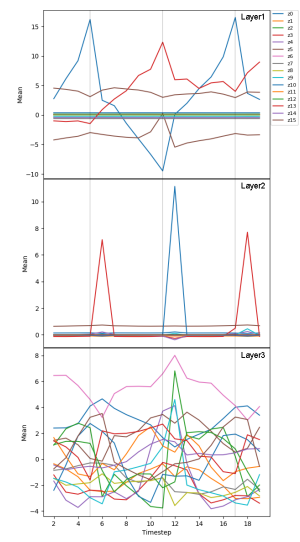
To examine HVRNN, we trained both VRNN and HVRNN using our own dataset following the Moving MNIST format with unsupervised training. One sequence consists of a 20 step image time series (64x64 pixels).

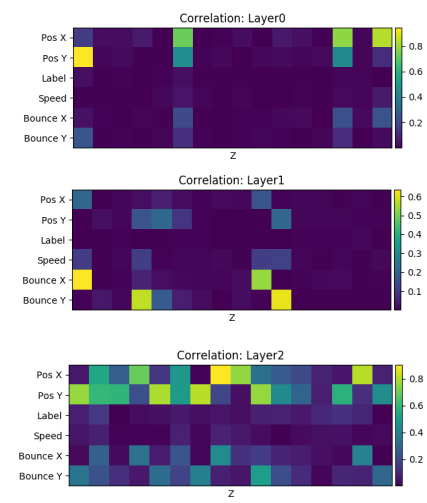
### VRNN




### HVRNN



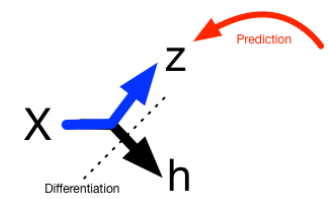




The graphs on the left show show the mean of latent variable **z**'s posterior during 20 time steps. Figures on the right show the correlation between each dimension of **z** and the factors that the input image data comprises.

In VRNN some dimensions of latent variable correspond to the movement of a digit, and in HVRNN this correspondence becomes more apparent, and they are more sparsely disentangled in the first layer. In HVRNN we can clearly see that in the second layer **z** corresponds to the timing of the bounces of the digit at the top and bottom walls with precise one frame delay after the bounce timing.

## Functional Differentiation

VRNN has the structure like the conditional VAE which is conditioned on RNN hidden state **h**. Conditional VAE tries to find factors of input **X** that are not included in the condition **h** and extract them as latent **z**. HVRNN's latent **z** is predicted by the higher layer, so we assume that the RNN state **h** in the lower layer is differentiated from the higher layer.




To confirm this, we suppressed the RNN state **h** in each layer of HVRNN. When **h** in the layer1 was suppressed, the shape of the object became corrupt while the position was kept. When there was no top-down prediction, both shape and position became corrupt. This result implies the functional differentiation in the HVRNN.