# Synthetic Gradient Methods with Virtual Forward-Backward Networks

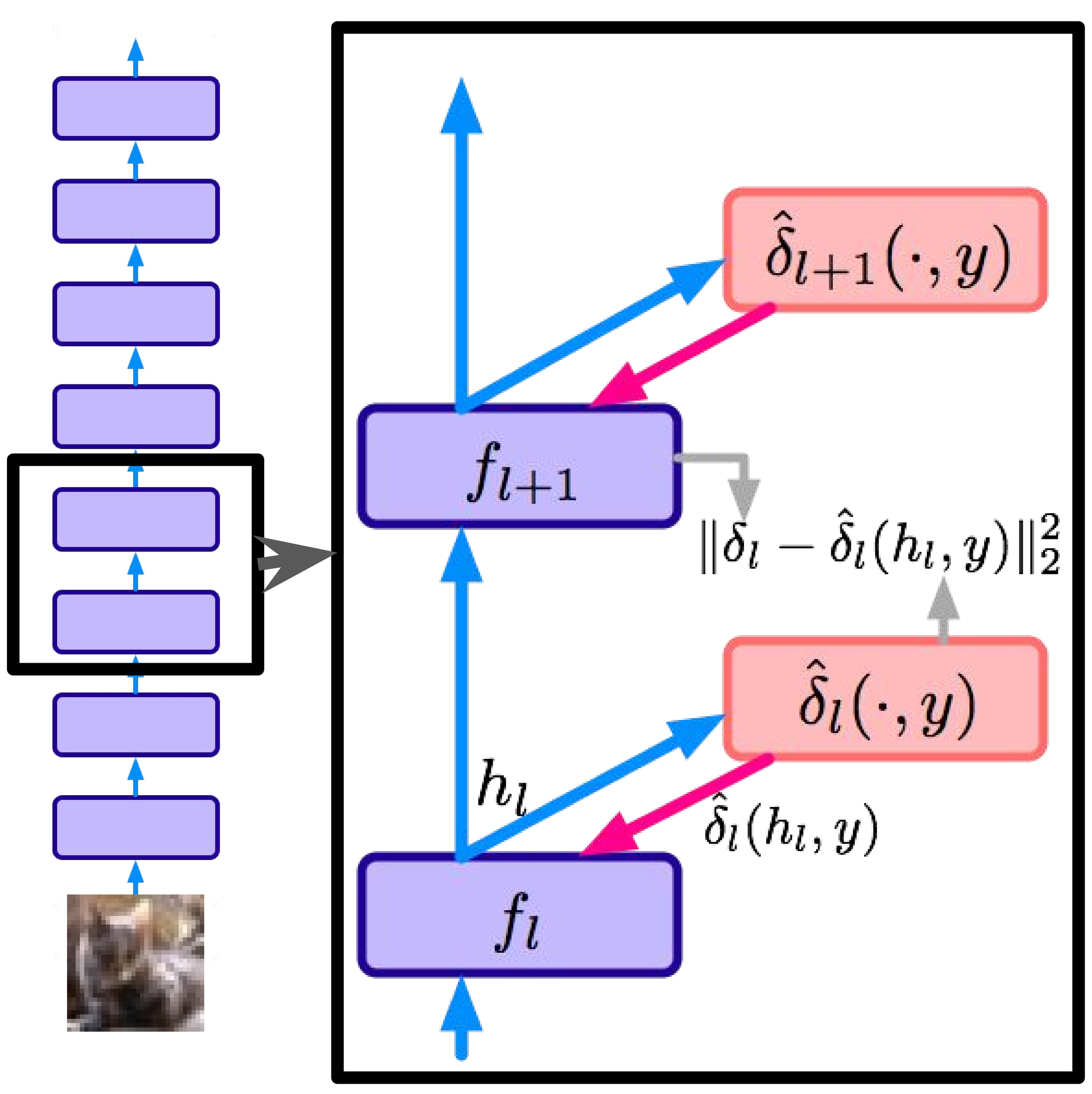

**Takeru Miyato[1,2], Daisuke Okanohara[1], Shin-ichi Maeda[3,(*)], Masanori Koyama[4]**

{miyato, hillbig, ichi}@preferred.jp, koyama.masanori@gmail.com

1.Preferred Networks, Inc. 2.ATR cognitive mechanisms laboratories
3.Kyoto University 4.Ritsumeikan University
(*). Shin-ichi Maeda currently belongs to Preferred Networks

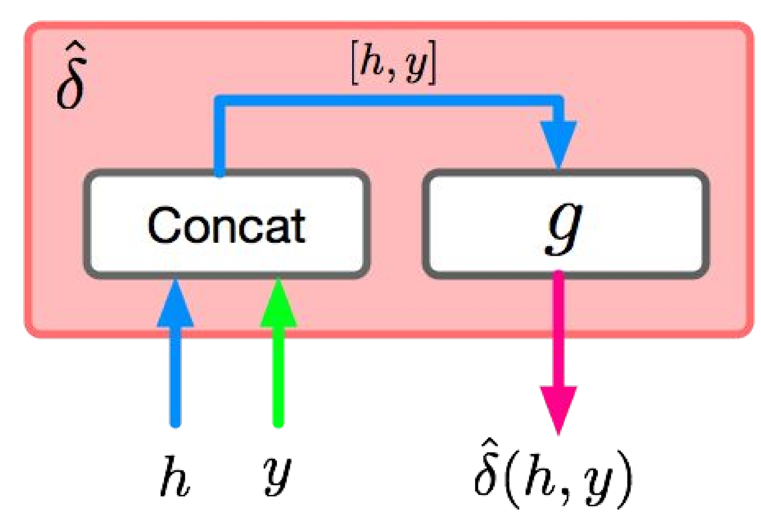

## Synthetic gradient for decoupling neural networks (Jaderberg, et al. 2016)

In synthetic gradient method, the parametric model predicts the gradients coming from the top layer, i.e. synthesizing gradients.
*-The updates of each layer are free from forward and backward locking.*



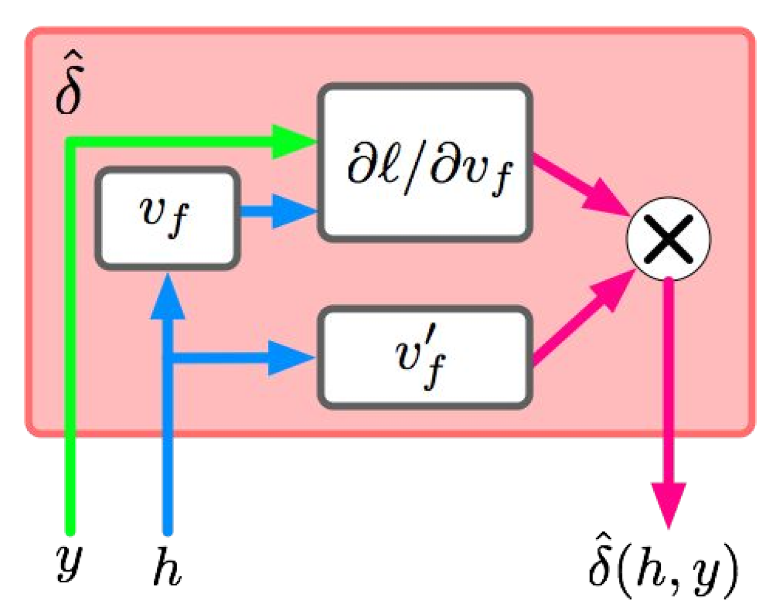

- The gradient of loss w.r.t. the weight $W_l$ of the $l$-th layer:
$$\frac{\partial \ell}{\partial W^l} = \frac{\partial \ell}{\partial h_l}\frac{\partial h_l}{\partial W^l} = \delta_l \frac{\partial h_l}{\partial W^l}$$
where $h_l$ is hidden layer of the network.
- Approximate gradient $\delta_l$ from hidden layer $h_l$ and context vector $c$ (label) with a parametric model (gradient synthesizer) to be trained:
$$\delta_l \approx \hat{\delta}_l(h_l, c)$$
- Train the gradient synthesizer by minimizing L2 loss between synthetic gradient and true gradient:
$$e_l(\hat{\delta}_l) := \|\delta_l - \hat{\delta}_l\|_2^2$$

## (This work) Proposed model of gradient synthesizer




(a) The model used in the original paper. (Jaderberg's model)

- For the gradient synthesizer in the original paper (Jaderberg eta al. 2016), they used the label information $y$ for the contextual information $c$, and used a model that takes the concatenated vector $[h, y]$ as the input:
$$\hat{\delta}(h, y) := g([h, y])$$
  - However, Jaderberg's model for gradient synthesizer has little relation to ***the gradient derived from the objectivefunction of the target task***.




(b) Our proposed model (VFBN)

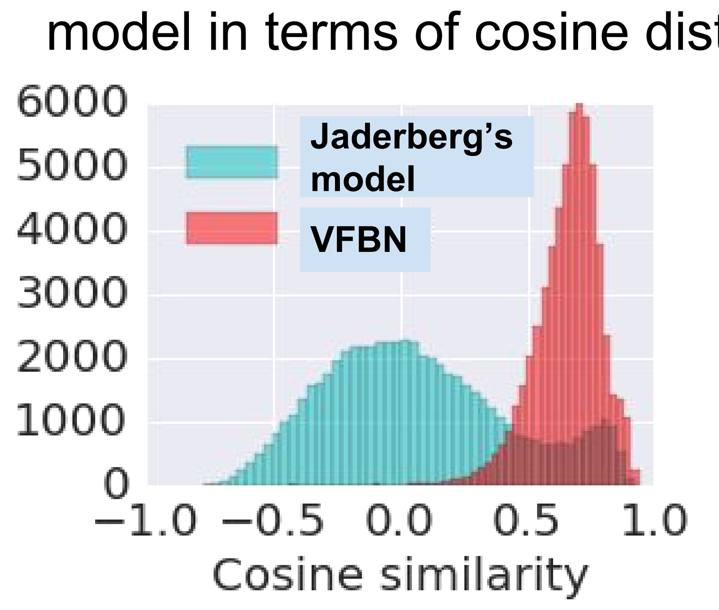
- We introduce *virtual forward-backward networks* (VBFN). VBFN is a model that produces synthetic gradient with a function which is analogous in its structure to the one derived from the objective function of the target task.

- The derivative of the original tasks can be represented by:
$$\delta_l(h, y) := \frac{\partial \ell(y, fwd(h))}{\partial h} = bwd(h) \times (\partial_{fwd(h)} \ell(y, fwd(h)))$$
*$fwd$ : forward function after $h$ that is derived from $p(Y|x, \theta)$,*
*$bwd$ : the derivative of $fwd$ w.r.t. $h$.*

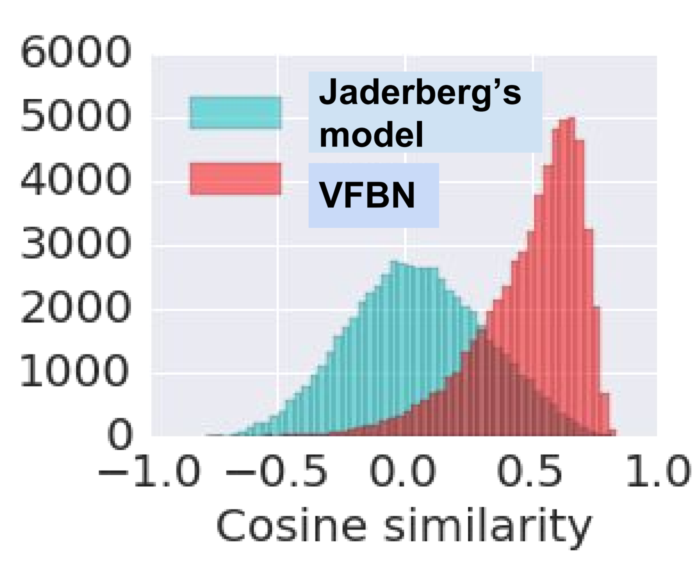
- Replacing the fwd above with ***virtual approximator*** $v_f(h; \Phi)$ , we get our **VFBN** gradient synthesizer:
$$\hat{\delta}_l(h, y)_{\mathrm{VFBN}} := \frac{\partial \ell(y, v_f(h; \phi))}{\partial h} = v'_f(h; \phi) \times \partial_{v_f(h)} \ell(y, v_f(h; \phi))$$
  - (e.g) For softmax classification on $h$, the VFBN should be:
$$\hat{\delta}(h, y) := W_v^{\mathrm{T}}(y - softmax(W_v h))$$

## Experiments: Cosine similarity on the true gradient and synthetic gradient

- VFBN improved the quality of synthetic gradients over the original model in terms of cosine distance.




(a) on MLP




(b) on CNN

## Experiments: CIFAR-10 classification with ResNet-110

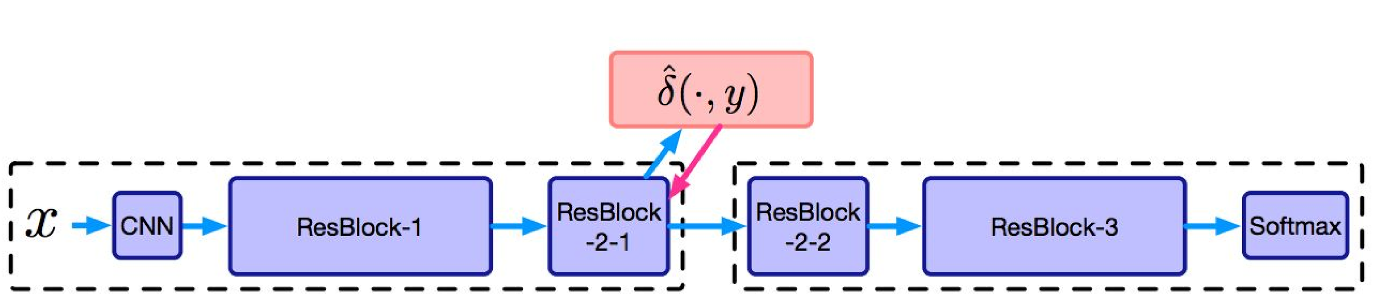



Figure 2: Decoupling of ResNet-110.

| | Test error (%) |
|---|---|
| BackProp | 5.15 |
| Bottom half with back prop | 5.76 |
| Subnetwork-wise supervised loss | 5.71 |
| (Synthetic Gradient models) | |
| Jaderberg's small-ResNet | 20.45 |
| Jaderberg's Linear | 15.56 |
| VFBN (ours, $\alpha_{gs} = 0$) | 5.73 |
| VFBN (ours, $\alpha_{gs} = 1e\text{-}3$) | 5.51 |

Table 2: Test error rates on CIFAR-10

(The test error is calculated by averaging over 3 different random seeds )

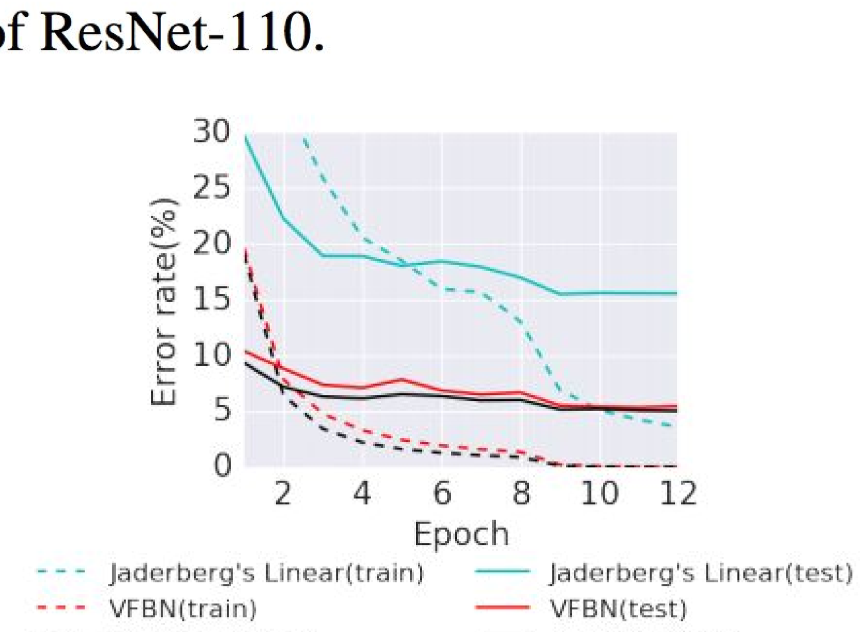



Figure 3: Learning curves on CIFAR-10

- We applied VFBN on decoupling ResNet-110 into 2 subnetworks, and used 4-layered (2-ResNet modules) CNN as VFBN.
  - The learning curve of the Jaderberg's model fall significantly behind the BP, while our VFBN keeps its pace with the BP throughout.
  - The performance with VFBN is ***5.51*** % error rate, which is better than the baseline such as half-ResNet (5.76%) and subnetwork-wise supervised loss learning (5.71%), but worse than standard BackProp.

## References


Jaderberg, Max and Czarnecki, Wojciech Marian and Osindero, Simon and Vinyals, Oriol and Graves, Alex and Kavukcuoglu, Koray. *Decoupled neural interfaces using synthetic gradients.* arXiv preprint, 2016