Decoupled Neural Interfaces using Synthetic Gradients

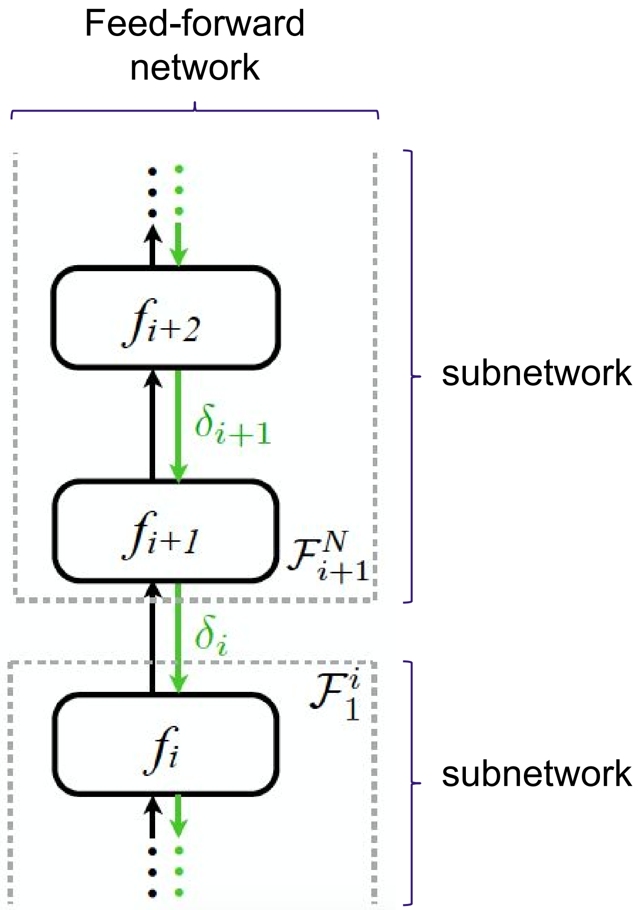
Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero,
Oriol Vinyals, Alex Graves, Koray Kavukcuoglu

BOSTON UNIVERSITY
MACHINE INTELLIGENCE
COMMUNITY

Justin Chen
Nov. 6, 2017

Locking

- **Locking** - Wait for all dependent computations before parameter update at current module
- **Forward Locking** - Wait for entire forward pass
- **Update Locking** - Wait for forward pass and compute cost function
- **Backwards Locking** - Wait for forward pass and dependent backwards pass



MIC

Backpropagation Expanded

$$\frac{\partial L}{\partial \theta_i} = f_{BProp}((h_i, x_i, y_i, \theta_i), (h_{i+1}, x_{i+1}, y_{i+1}, \theta_{i+1}), \ldots)\frac{\partial h_i}{\partial \theta_i} \simeq \hat{f}_{BProp}(h_i)\frac{\partial h_i}{\partial \theta_i}$$

- **Goal**: Remove **all locking**
- **Main contribution**: Backward unlocking for infinite time with RNNs
- Expanding BP exposes dependencies
- **Approximate gradient of activations** w.r.t. parameters of **adjacent layers** (modules depending on how you divide up the computation graph)
- Update **Synthetic Gradient Model** with True Gradients

M I C

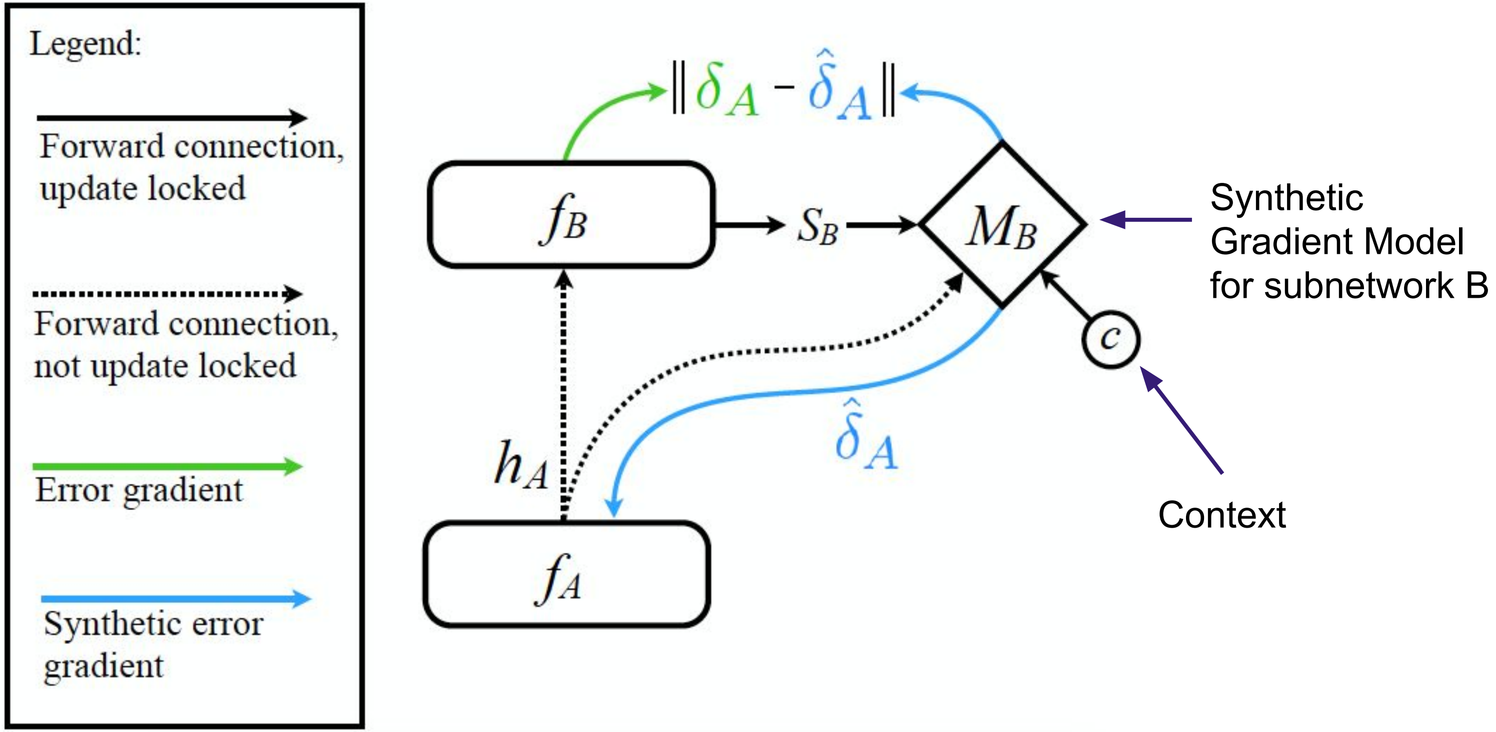
Synthetic Gradient Model

$$\hat{\delta}_A = M_B(h_A, s_B, c)$$

$\hat{\delta}_A$	Synthetic gradient **for previous layer/module**	s_B	Activation of **next layer/module**, which we refer to as module B
M_B	Synthetic Gradient Model	c	Context information - additional information such as label information if it's available
h_A	Activation of **previous layer/module**, which we refer to as module A	δ_A	**True gradient** from backpropagation

M I C

Decoupled Neural Interface (DNI)



Updating DNI

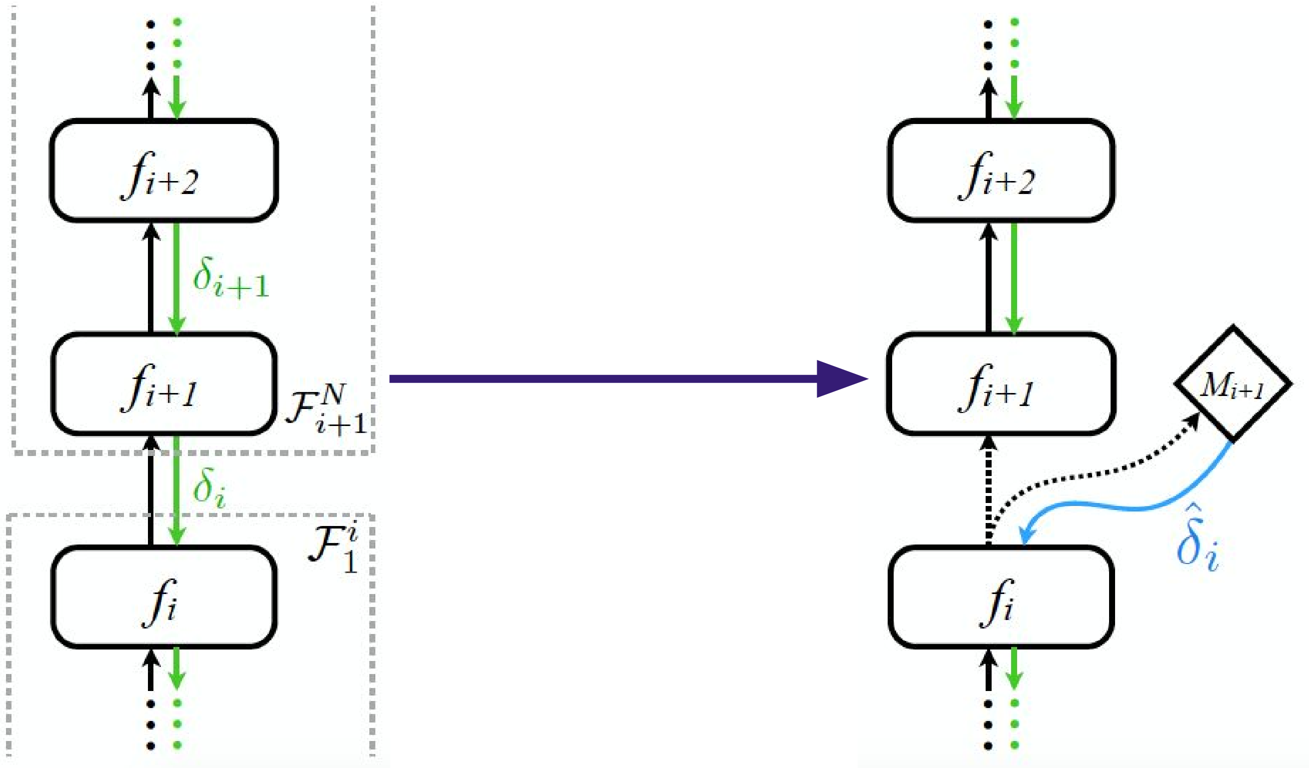
- DNI are update locked
- Trained to minimize L_2
- Can incorporate context information if available

$$L_{\delta_i} = d(\hat{\delta}_i, \delta_i)$$

$$\hat{\delta}_i = M_{i+1}(h_i, c)$$

MIC

Unlocked Subnetworks



M I C

Expanding Backpropagation

$$\theta_i \leftarrow \theta_i - \alpha\delta_i \frac{\partial h_i}{\partial \theta_i}; \delta_i = \frac{\partial L}{\partial h_i}$$

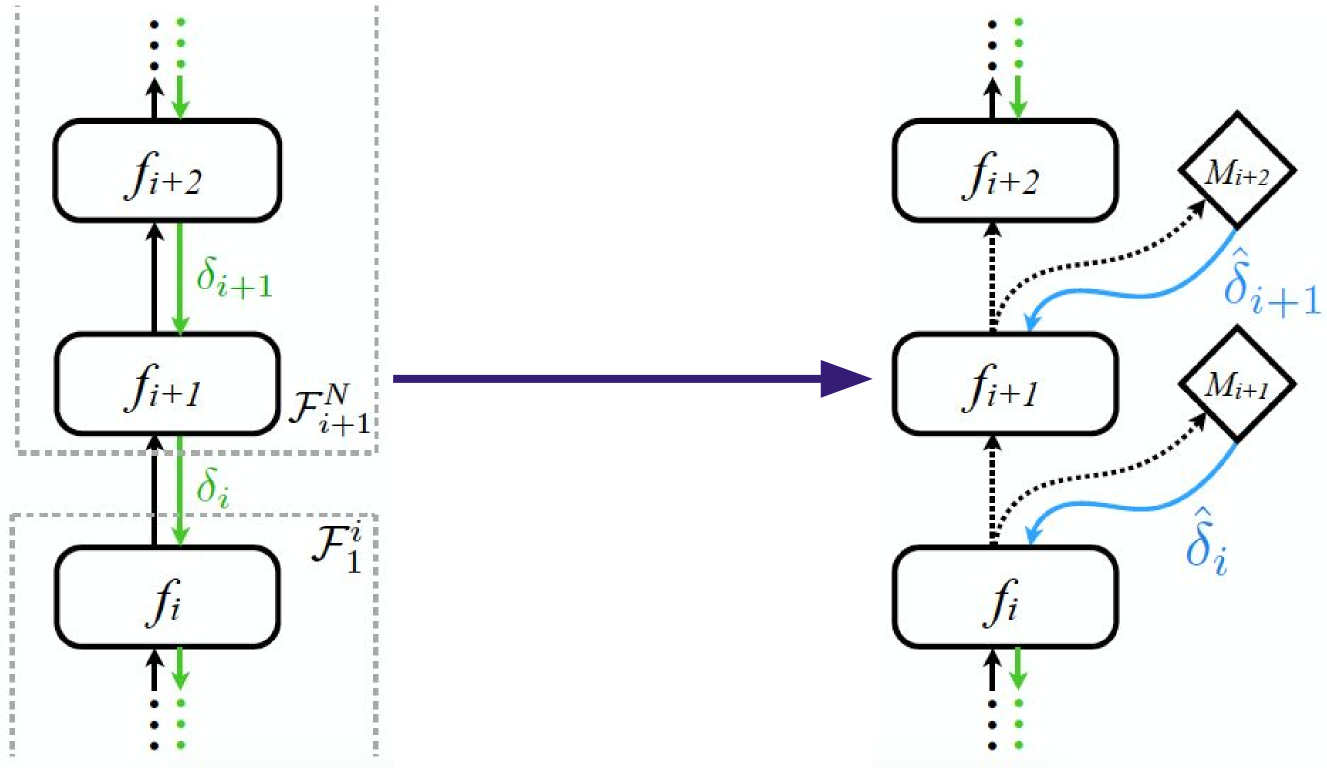
Parameters

Learning Rate

Local Gradient

True Gradient

Multiple DNI Feedforward Network



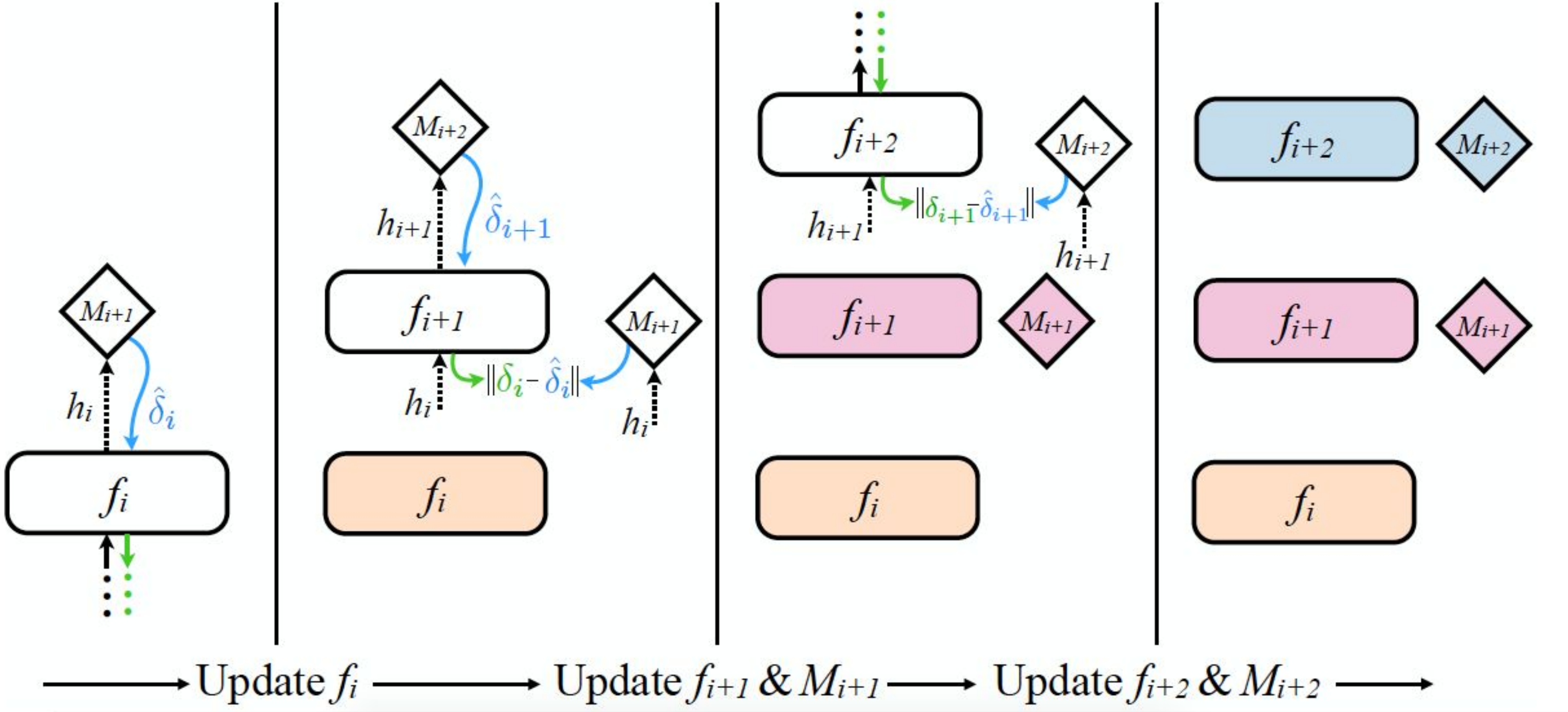
Multiple DNI Update Rule

$$\theta_n \leftarrow \theta_n - \alpha \hat{\delta}_i \frac{\partial h_i}{\partial \theta_n}, n \in \{1, ..., i\}$$

Parameters Learning Rate Synthetic Gradient Local Gradient Layers

M I C

Updating Multiple DNI Feedforward Networks



M I C

Expanding BP for RNN

$$\theta - \alpha \sum_{\tau=t}^{\infty} \frac{\partial L_\tau}{\partial \theta}$$

Parameters

Learning Rate

Sum over infinite timesteps

Gradient at timestep τ

M I C

Expanding BP for RNN

$$= \theta - \alpha \left(\underbrace{\sum_{\tau=t}^{t+T} \frac{\partial L_\tau}{\partial \theta}}_{\text{Unroll for only T timesteps}} + \underbrace{\left(\sum_{\tau=T+1}^{\infty} \frac{\partial L_\tau}{\partial h_T} \right) \frac{\partial h_T}{\partial \theta}}_{\text{Timesteps after T (future timesteps)}} \right)$$

Unroll for only
T timesteps

Timesteps after T
(future timesteps)

M I C

Expanding BP for RNN

$$= \theta - \alpha \left(\sum_{\tau=t}^{t+T} \frac{\partial L_\tau}{\partial \theta} + \underbrace{\delta_T}_{\text{Summation of true gradients over infinite timesteps after timestep T}} \frac{\partial h_T}{\partial \theta} \right)$$

- Calculating infinite timesteps is intractable
- Typically ignore timesteps after T by multiplying future gradients by 0

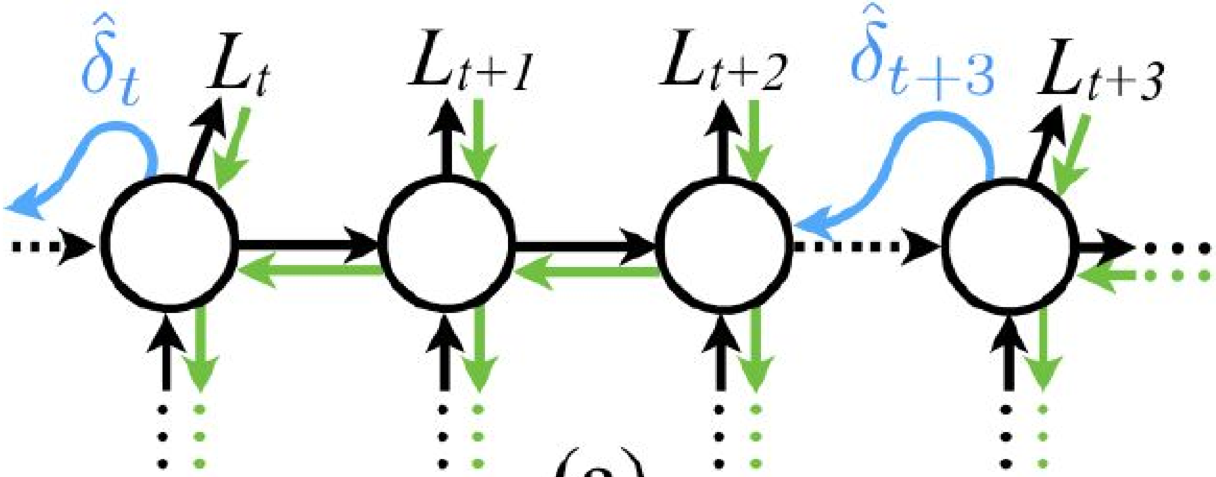
Breaking RNN Time Boundaries

$$\theta - \alpha \left(\sum_{\tau=t}^{t+T} \frac{\partial L_\tau}{\partial \theta} + \underbrace{\hat{\delta}_T}_{\text{Approximate with DNI}} \frac{\partial h_T}{\partial \theta} \right)$$

- Equivalent to unrolling for infinite timesteps with infinite subnetworks
- DNI allows RNN to asynchronously communicate with future self

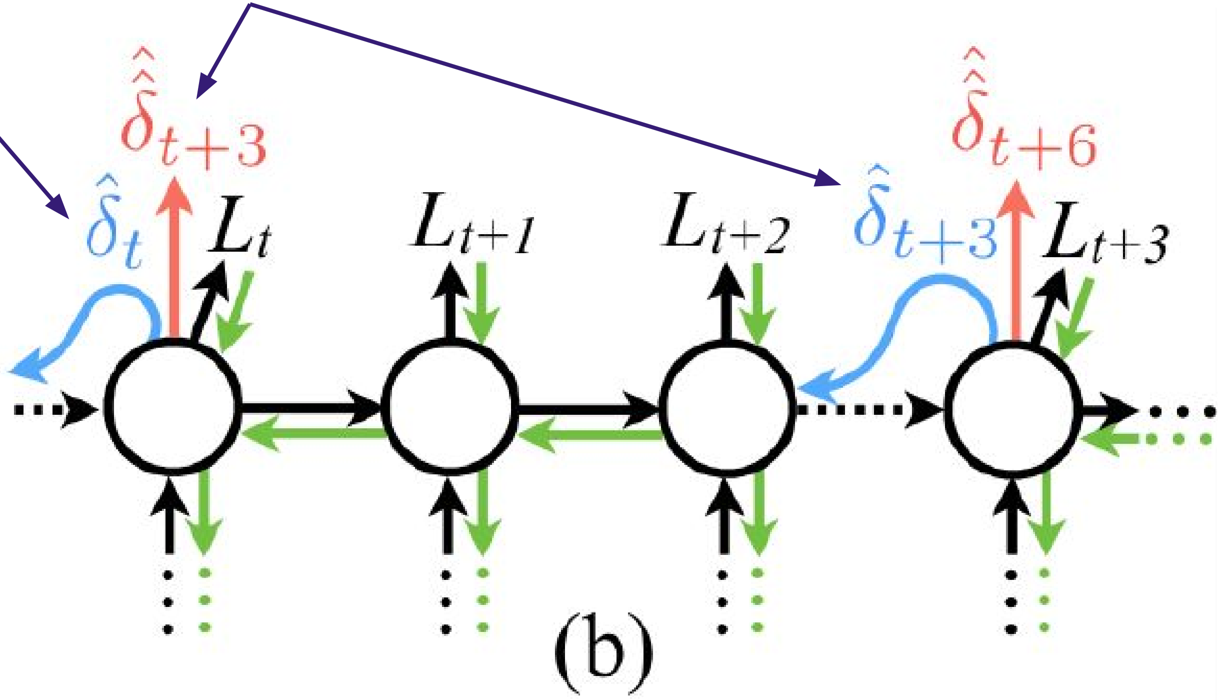
M I C

Breaking RNN Time Boundaries

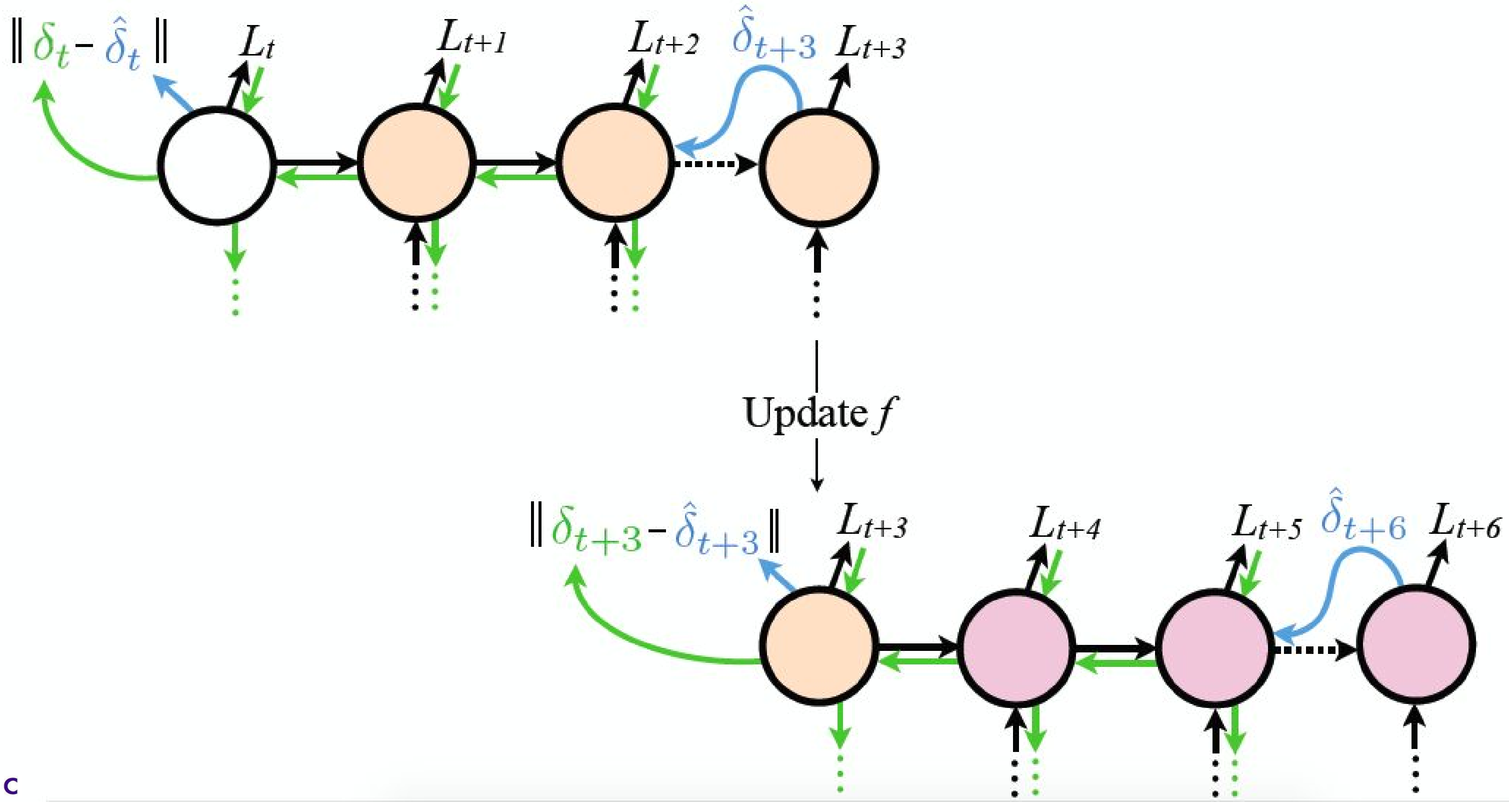


(a)

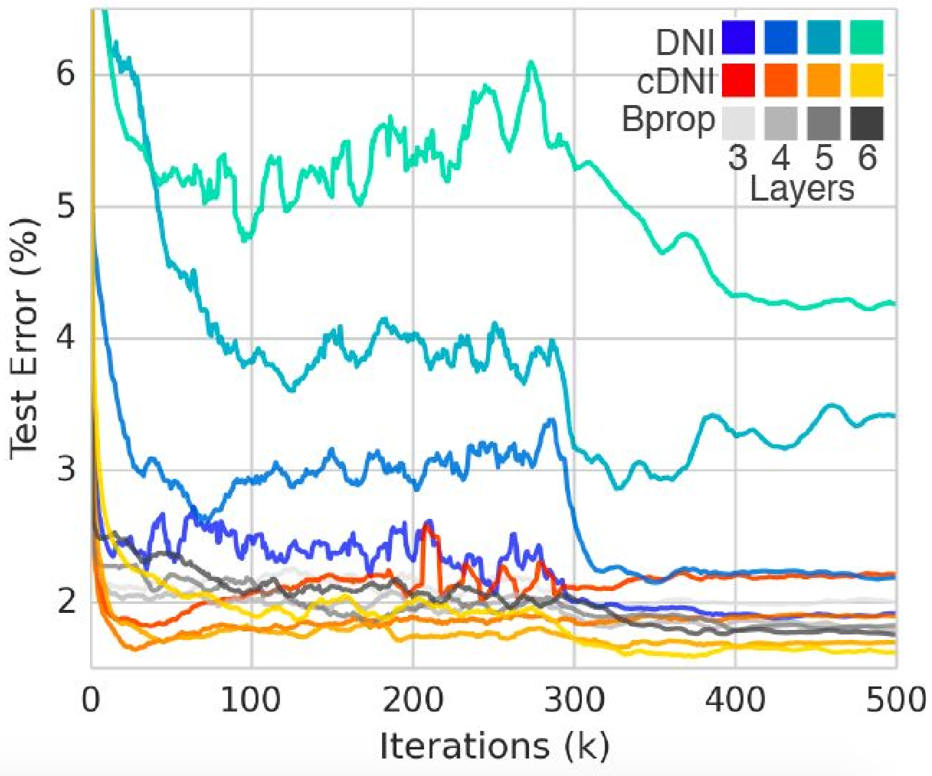
Short-term & Long-term Synthetic Gradients



Updating Multiple DNI Recurrent Networks



DNI between every layer in FCN

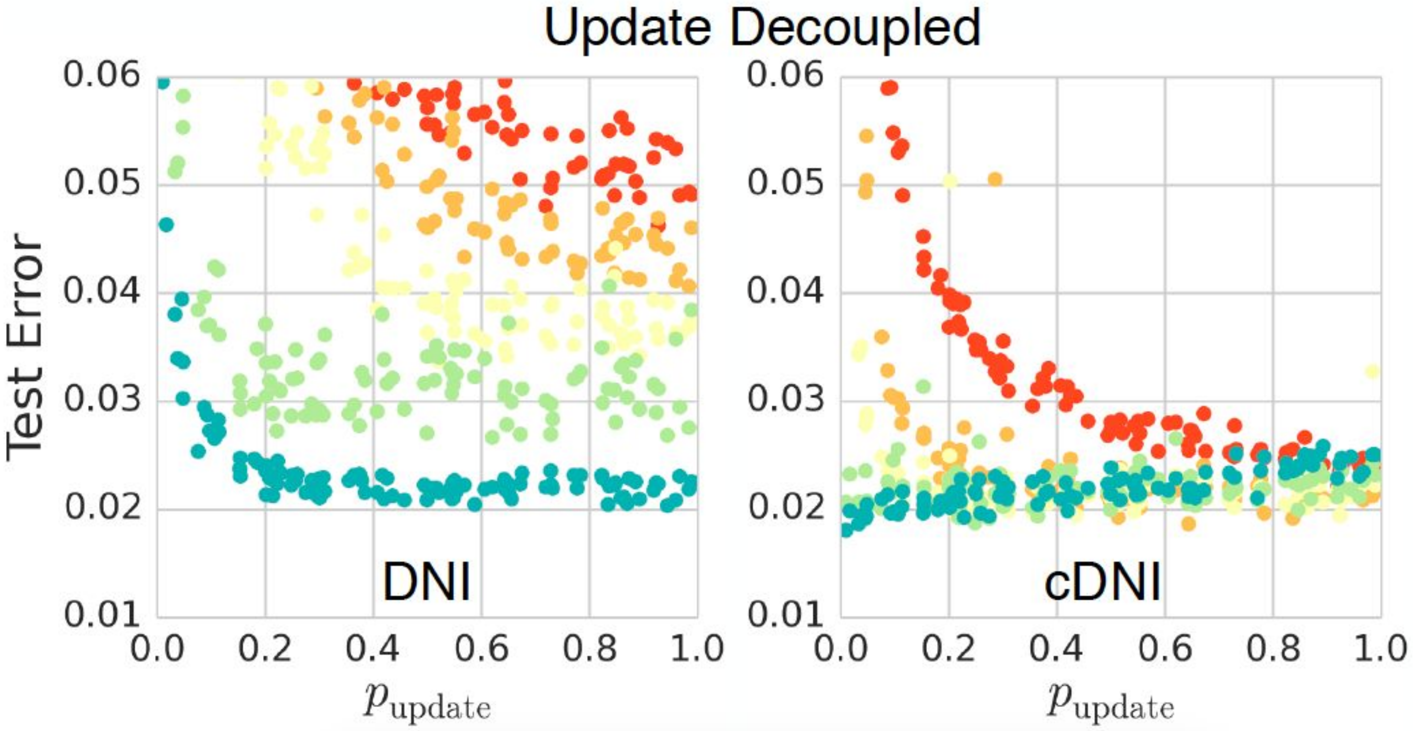


M I C

DNI between every layer in FCN and CNN

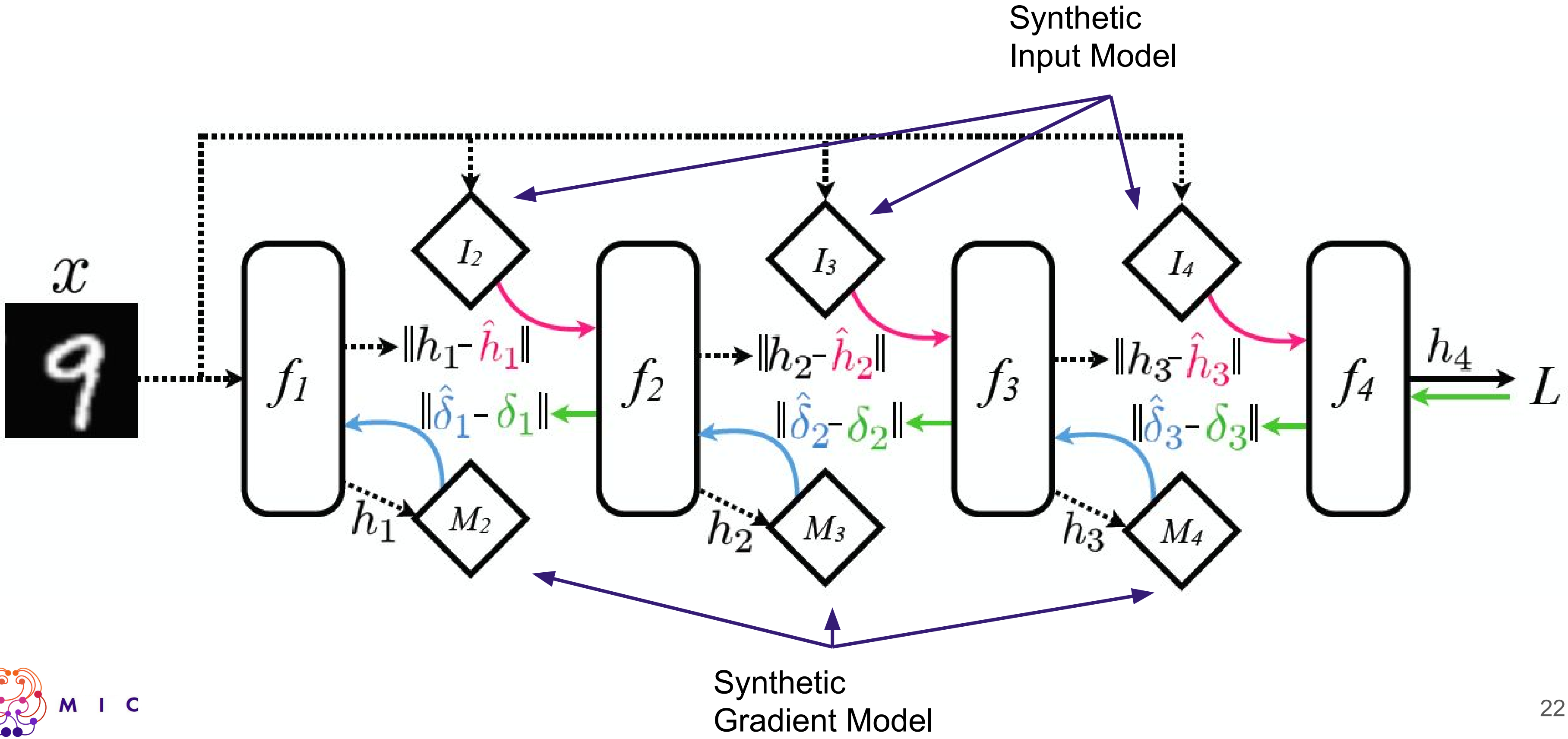
	Layers	MNIST (% Error) No Bprop	Bprop	DNI	cDNI	CIFAR-10 (% Error) No Bprop	Bprop	DNI	cDNI
FCN	3	9.3	2.0	1.9	2.2	54.9	43.5	42.5	48.5
	4	12.6	1.8	2.2	1.9	57.2	43.0	45.0	45.1
	5	16.2	1.8	3.4	1.7	59.6	41.7	46.9	43.5
	6	21.4	1.8	4.3	1.6	61.9	42.0	49.7	46.8
CNN	3	0.9	0.8	0.9	1.0	28.7	17.9	19.5	19.0
	4	2.8	0.6	0.7	0.8	38.1	15.7	19.5	16.4

20% Chance Backwards Unlocking



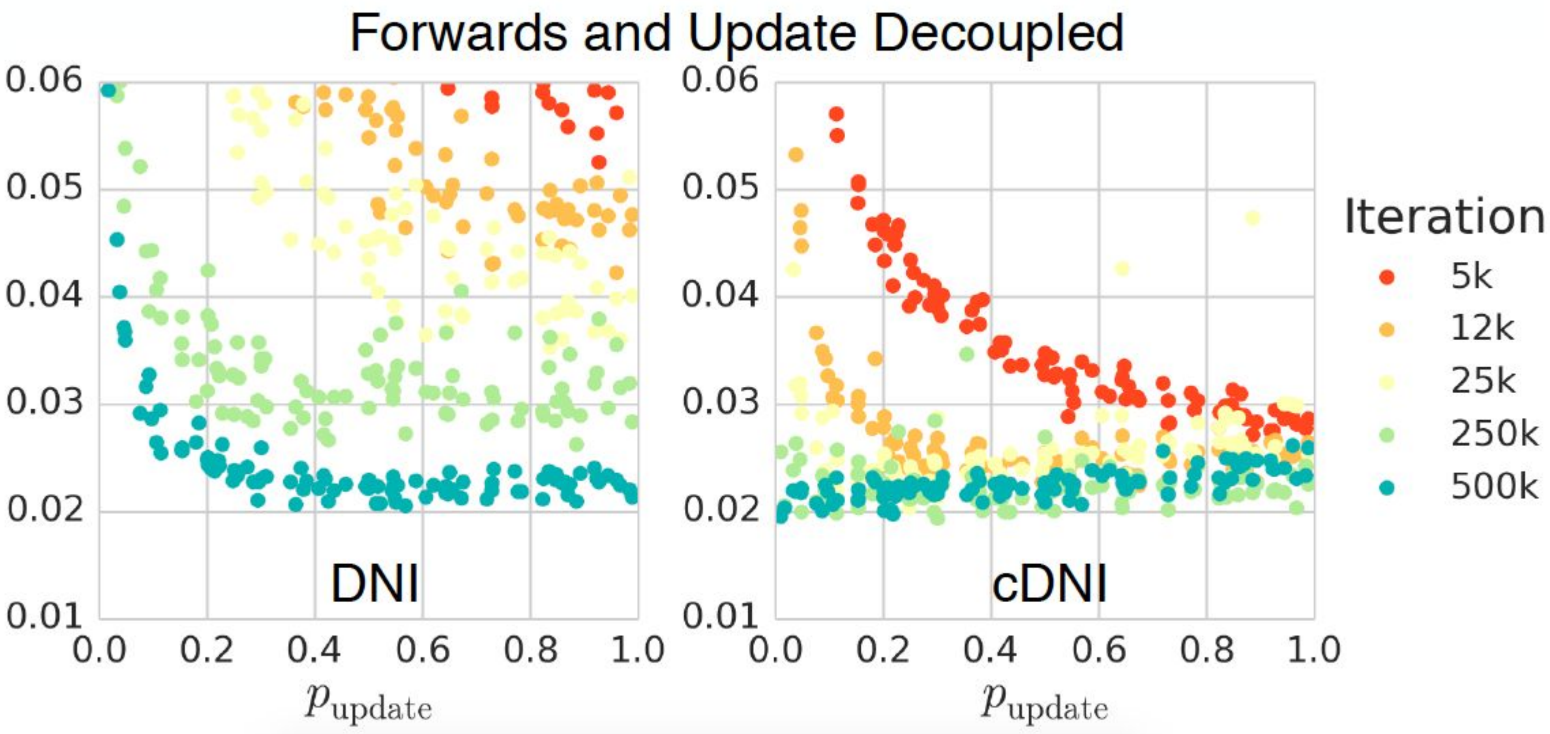
M I C

Completely Unlocked Network



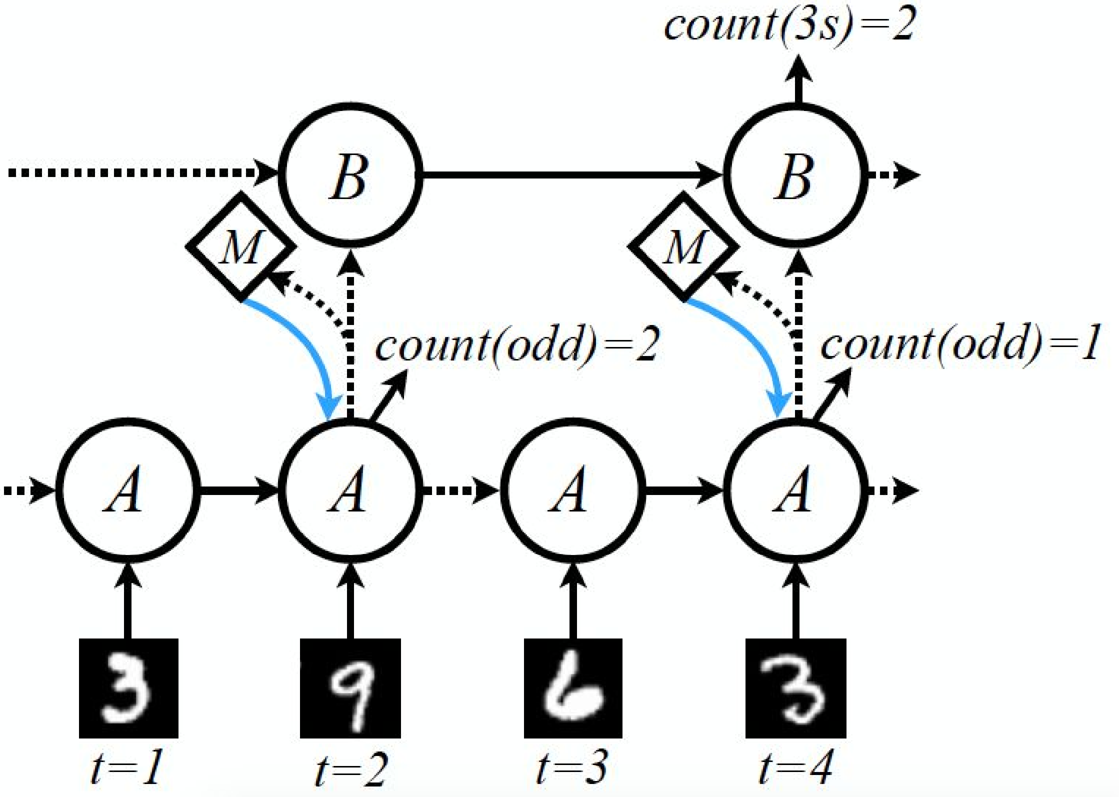
M I C

Forwards Unlocking

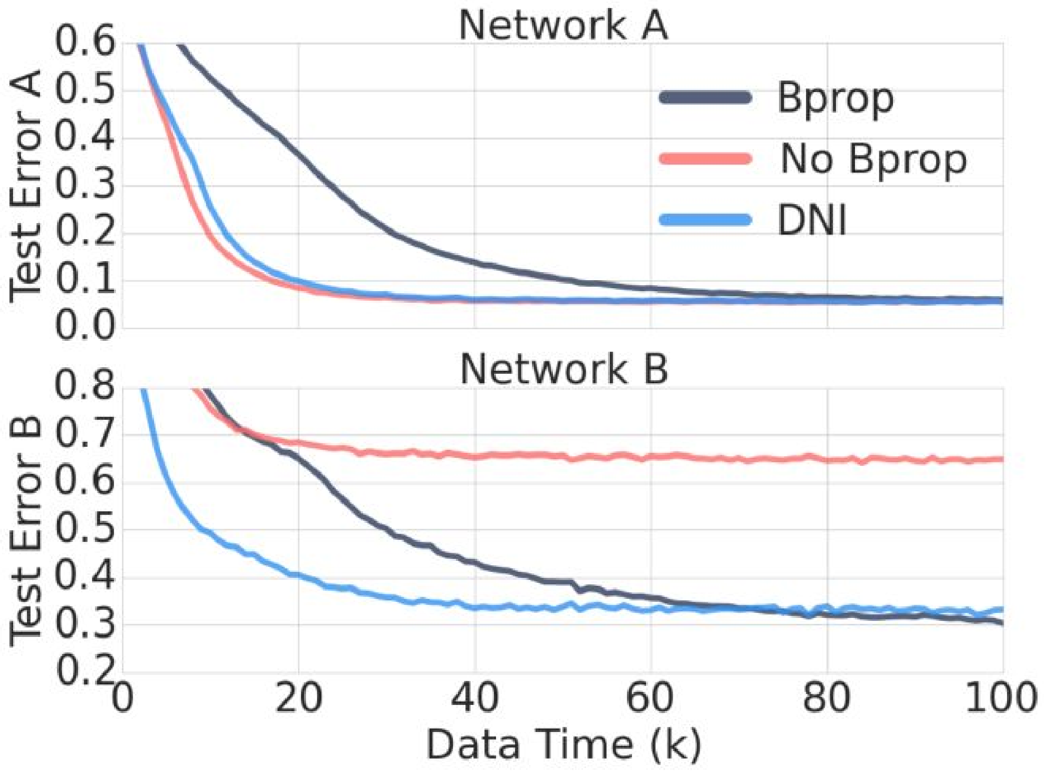


M I C

Dynamic Computation Graphs and Ensembles



Two RNNs communicating with DNI



- One RNN trained to count number of 3s seen
- Second RNN trained to count number odd numbers seen

References and Further Reading

1. Jaderberg, Max, et al. "**Decoupled neural interfaces using synthetic gradients**." arXiv preprint arXiv:1608.05343 (2016).
2. Czarnecki, Wojciech Marian, et al. "**Understanding Synthetic Gradients and Decoupled Neural Interfaces**." arXiv preprint arXiv:1703.00522 (2017).
3. Miyato, Takeru, et al. "**Synthetic Gradient Methods with Virtual Forward-Backward Networks**." (2017).
4. Czarnecki, Wojciech Marian, et al. "**Sobolev Training for Neural Networks**." arXiv preprint arXiv:1706.04859 (2017).