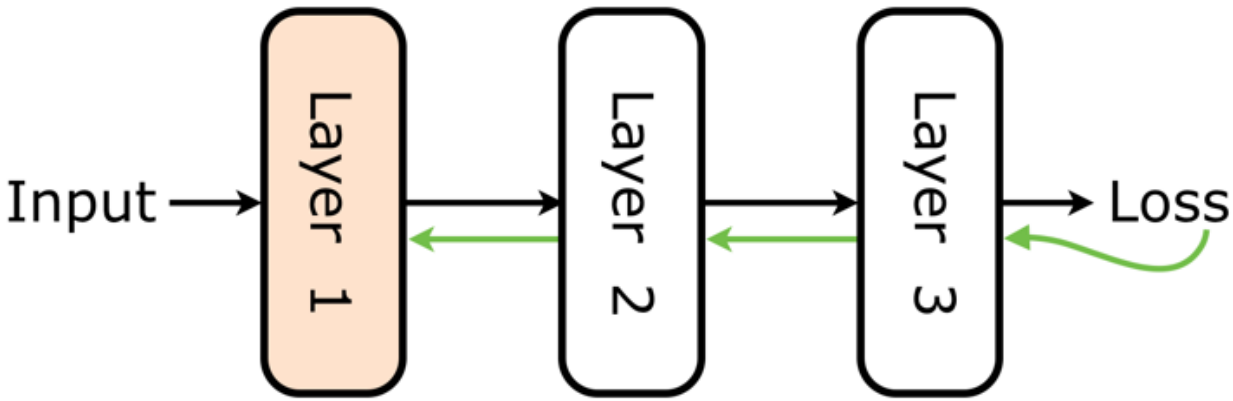
Синтетические градиенты

Иван Гущенко-Чеверда

Национальный Исследовательский университет
"Высшая школа экономики"

24 января, 2017

Обратное распространение ошибки



Вычислительные ограничения

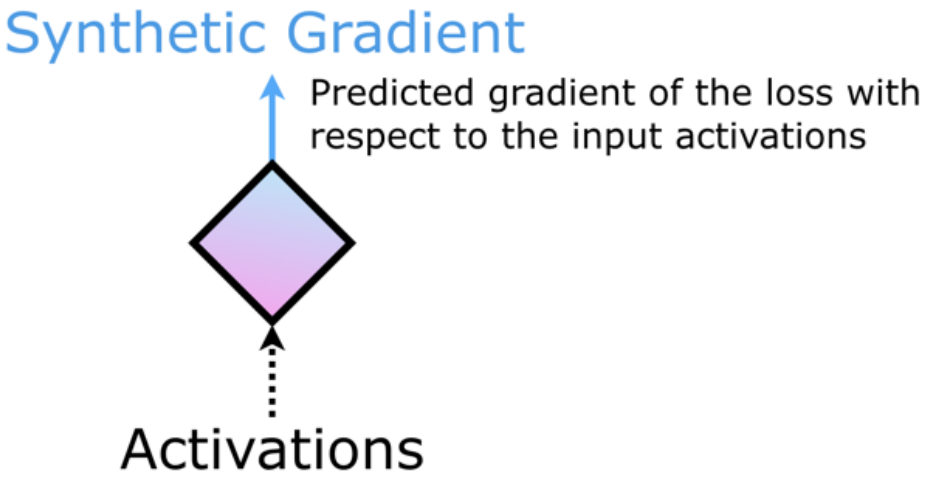
- **Forward Locking** - для получения выхода k-го слоя нужно получить выход предыдущих.
- **Update Locking** - для обновления весовых коэфициентов на k-м слое необходимо получить выход всех слоев после k-го
- **Backwards Locking** - выход на всех слоях должен быть получен, а так же должен быть получен градиент ошибки на всех слоях после k-го

Посыл

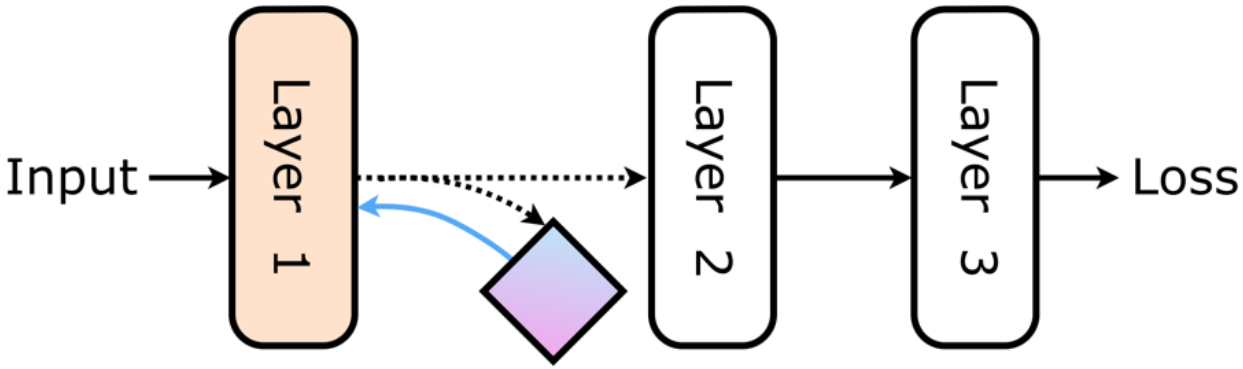
Все эти блокировки усложняют конструирование распределенных систем для обработки нейронных сетей, так как они требуют синхронной работы.

Синтетические градиенты

Следующая модель позволяет строить граф вычислений, в котором нет Update и Backwards Locking. Основная ее идея видна на изображении.

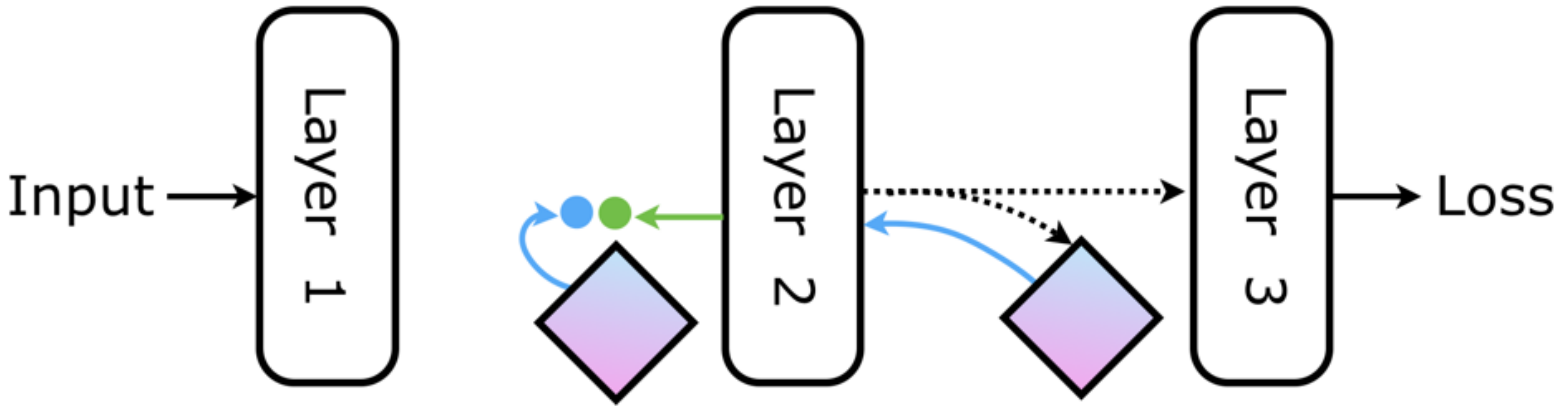


Описание работы



Строится модель, которая учится предсказывать градиент по активациям слоя. Оказывается, что с этим хорошо справляются нейросети с 0-3 скрытыми слоями. То есть даже линейная модель предсказывает градиент достаточно хорошо, чтобы при использовании такой архитектуры можно обучаться.

Описание работы



Так происходит обучение нашей модели. К ней на вход приходит либо градиент, распространенный с предыдущего слоя, полученный другой такой же моделью, либо реальные ошибки backpropagation.

Качество модели

	Layers	MNIST (% Error)				CIFAR-10 (% Error)			
		No Bprop	Bprop	DNI	cDNI	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

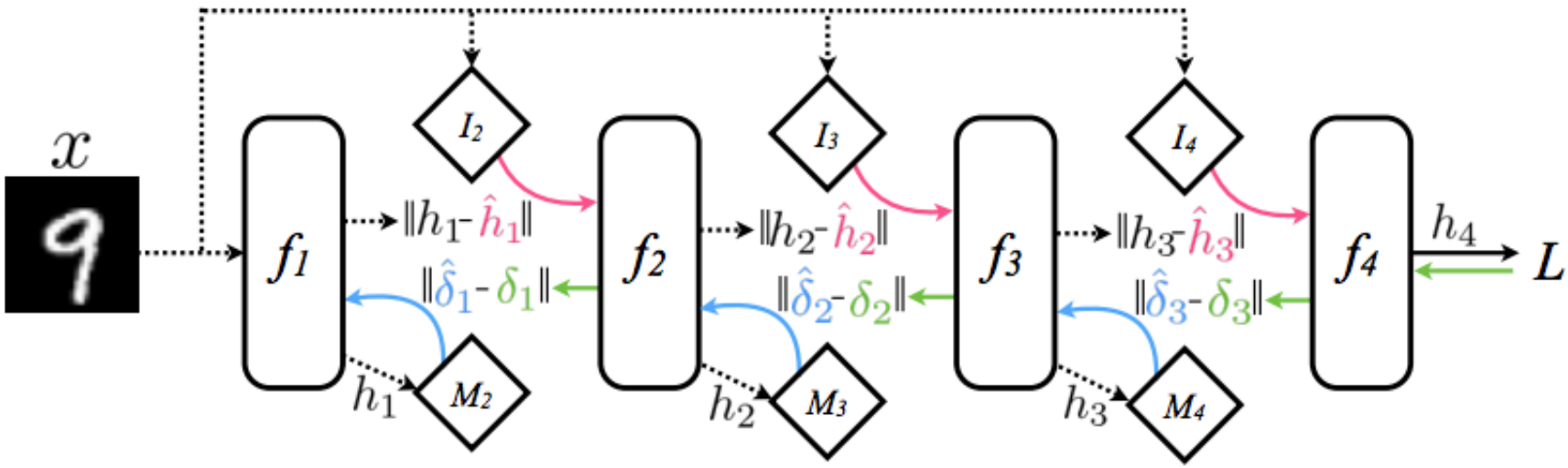
Test Error (%)

cDNI

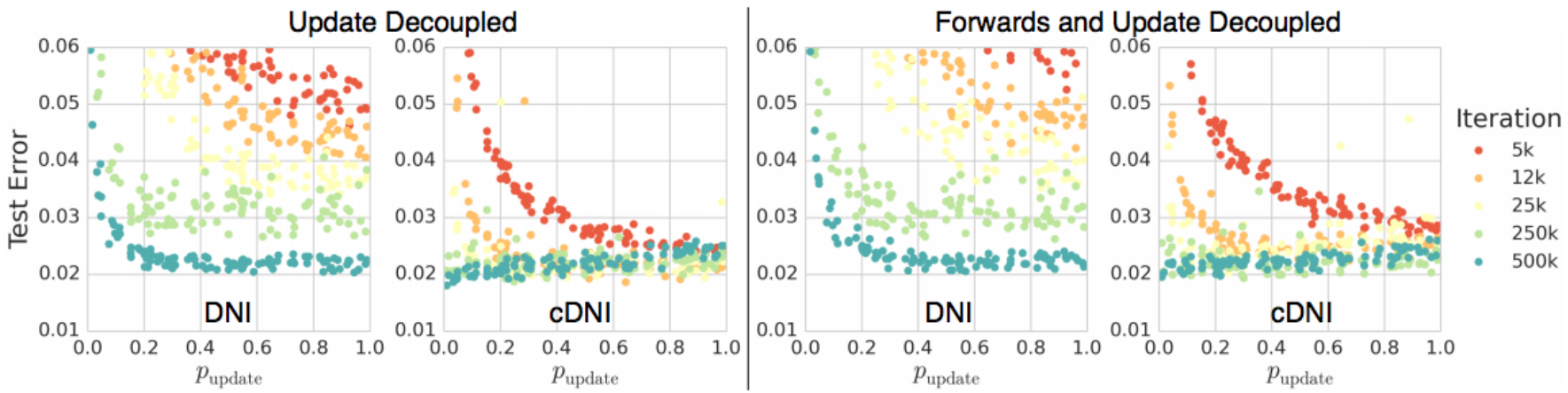
К каждому синтетическому градиенту присоединяется дополнительно реальная метка класса, закодированная one-hot кодированием. То есть, для рассмотренных выше датасетов, вход синтетического градиента расширяется на 10 бинарных переменных

Complete Unlock

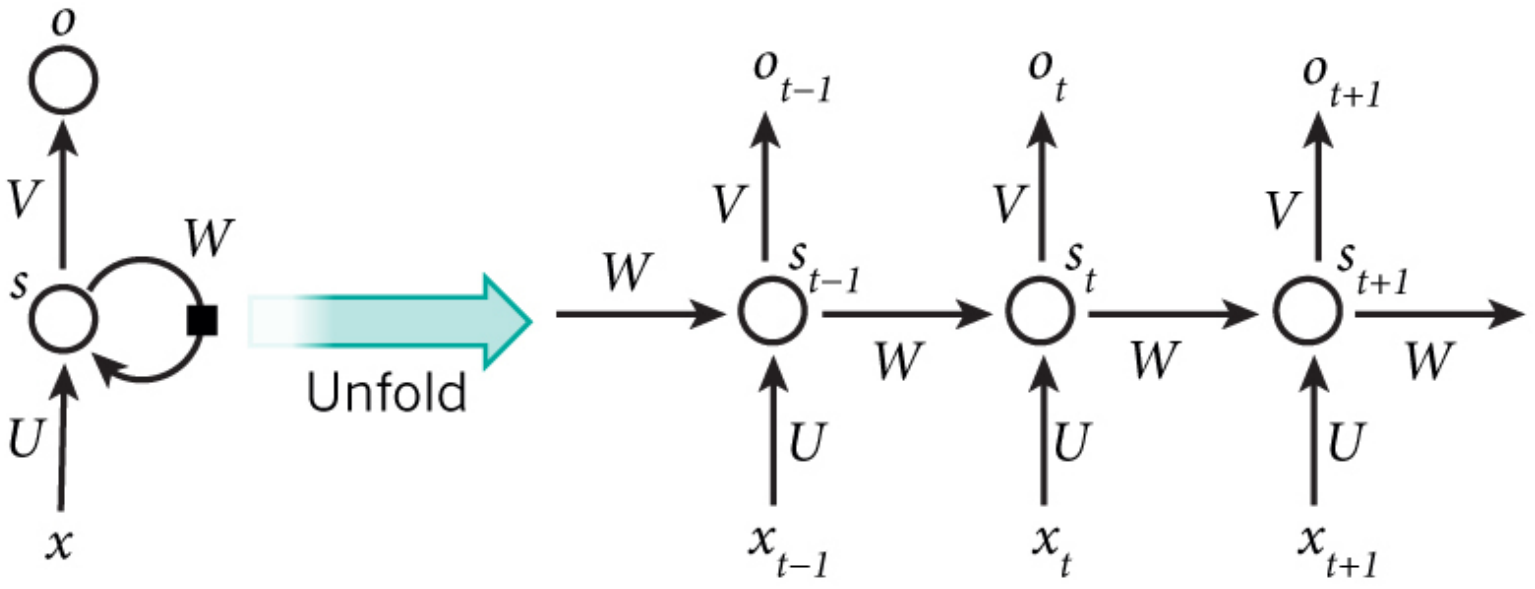
Чтобы избавиться от **Forward Locking** можно использовать модель синтетического входа (аналогично синтетическим градиентам)



Тестирование



Архитектура рекуррентной сети

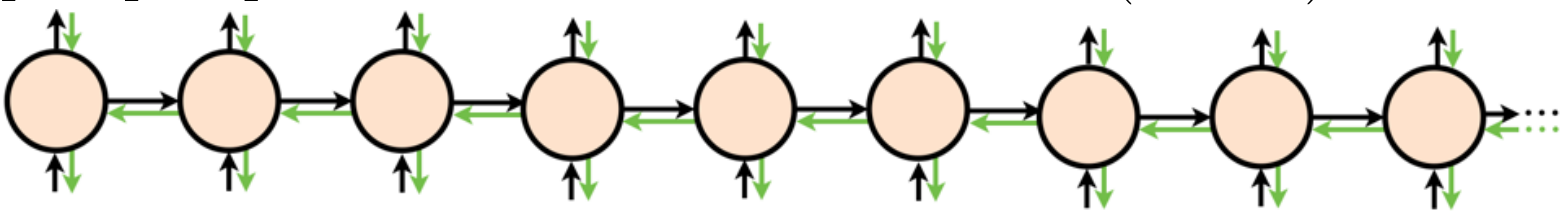


$$s_t = Ux_t + Ws_{t-1}$$

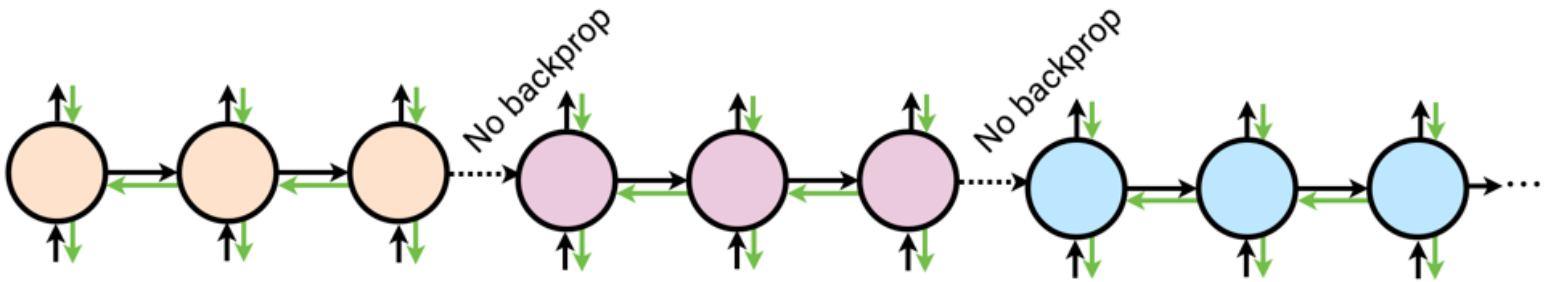
$$o_t = Vs_t$$

Архитектура рекуррентной сети

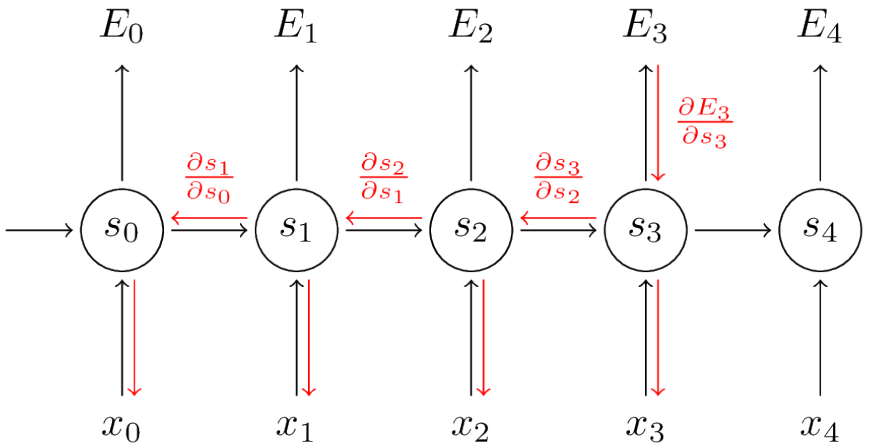
Теоретическая RNN выглядит так и обучается обратным распространением ошибки по всей цепи (BPTT).



В реальности используются же лишь последние k-шагов. Обучение методом усеченного обратного распространения. (truncated BPTT).



Обучение рекуррентной сети



Функция потерь определяется, как

$$E(y, o) = \sum_{i=k}^{k+3} \bar{E}(o_i, y_i)$$

где $\bar{E}$ – функция потерь для одного элемента.

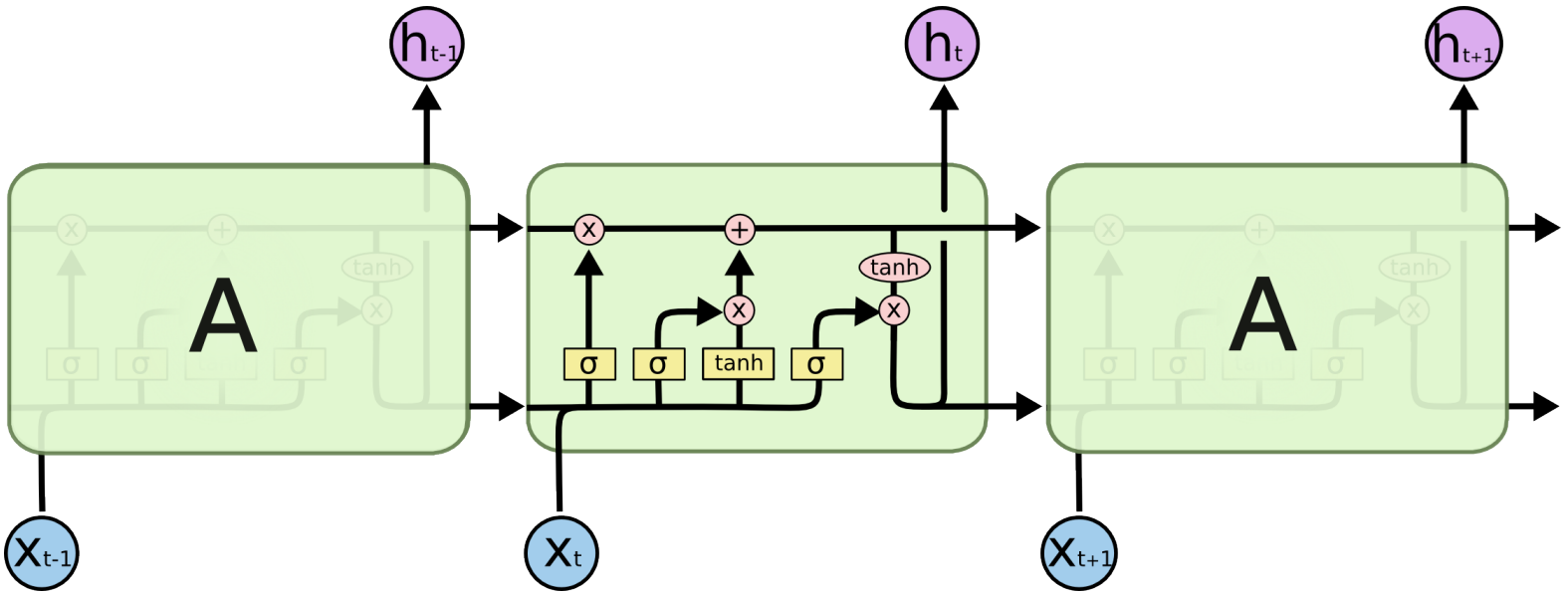
Обучение рекуррентной сети

Обучение каждого отдельного куска сети происходит обратным распространением ошибки. Так как для ошибка хорошо дифференцируется по каждой из матриц U, W, V. Дифференциал $\bar{E}(o_i, y_i)$ по W, например будет равен

$$\frac{\partial E_{k+3}}{\partial W} = \sum_{i=k}^{k+3} \frac{\partial E_{k+3}}{\partial o_{k+3}} \frac{\partial o_{k+3}}{\partial s_{k+3}} \frac{\partial s_{k+3}}{\partial s_i} \frac{\partial s_i}{\partial W}$$

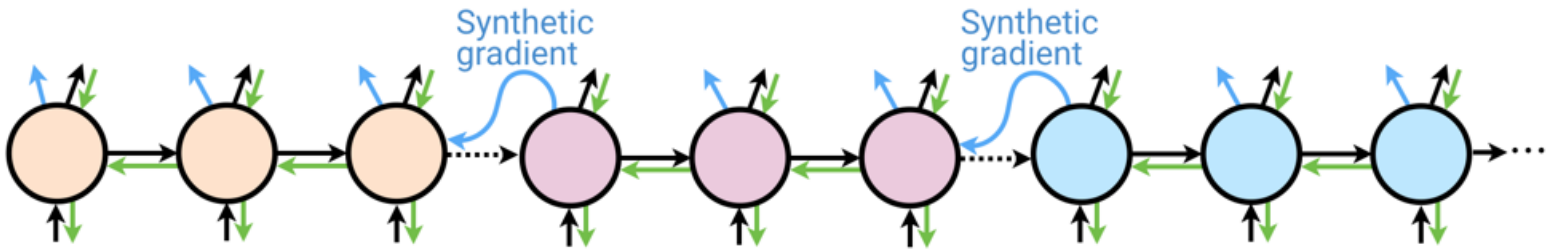
Для матриц V и U он выписывается аналогично.

LSTM core



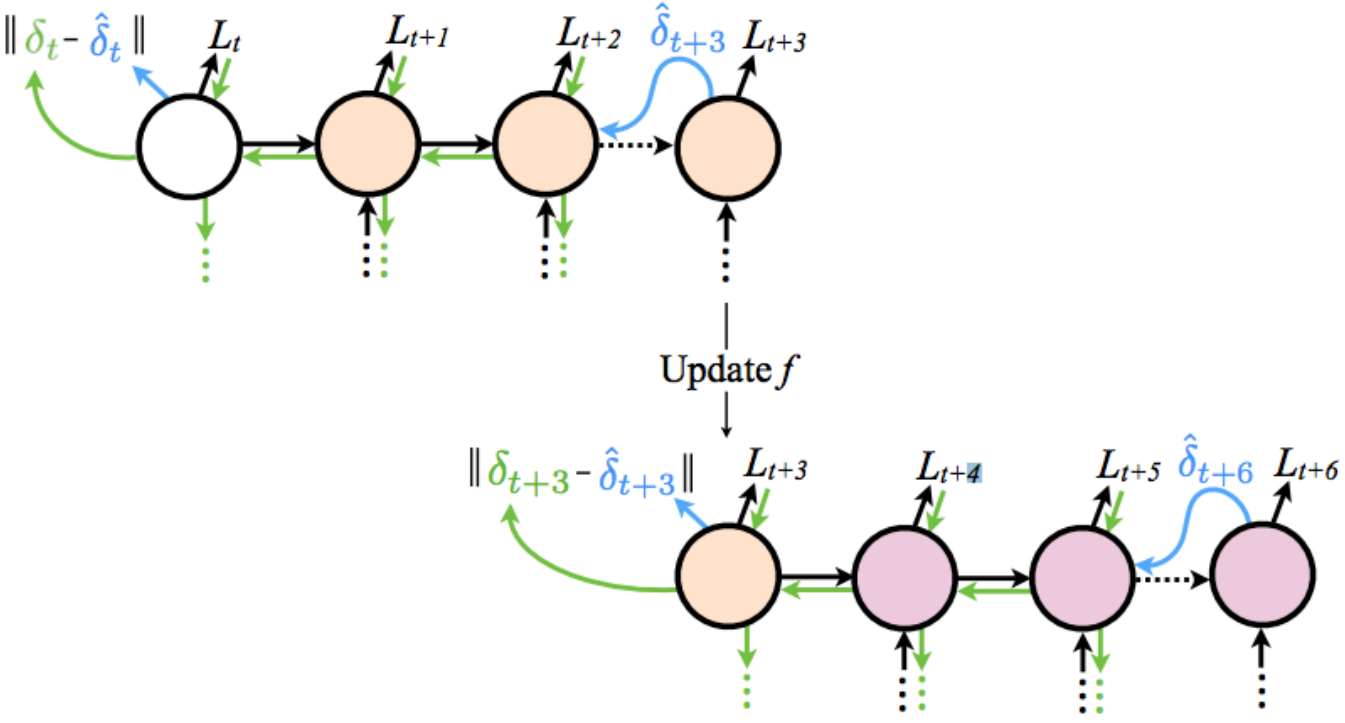
Синтетические градиенты в рекуррентной сети

Основная идея состоит в том, чтобы предсказывать ошибку последующей сети с помощью синтетического градиента.

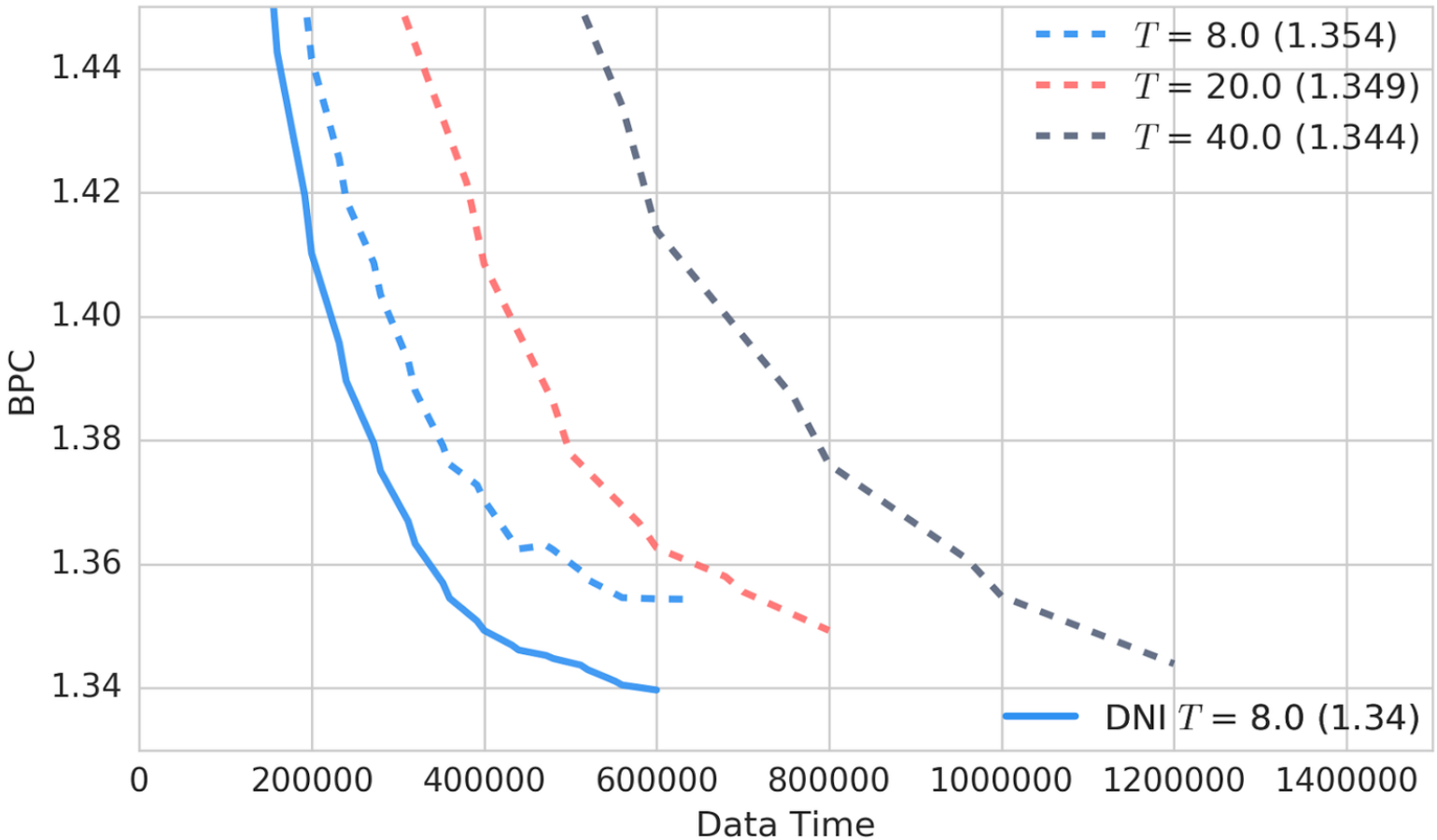


Это позволяет не ограничиваться лишь текущим куском сети, незначительно увеличивая требуемую память/время на вычисление ошибки и градиентов.

Синтетические градиенты в рекуррентной сети



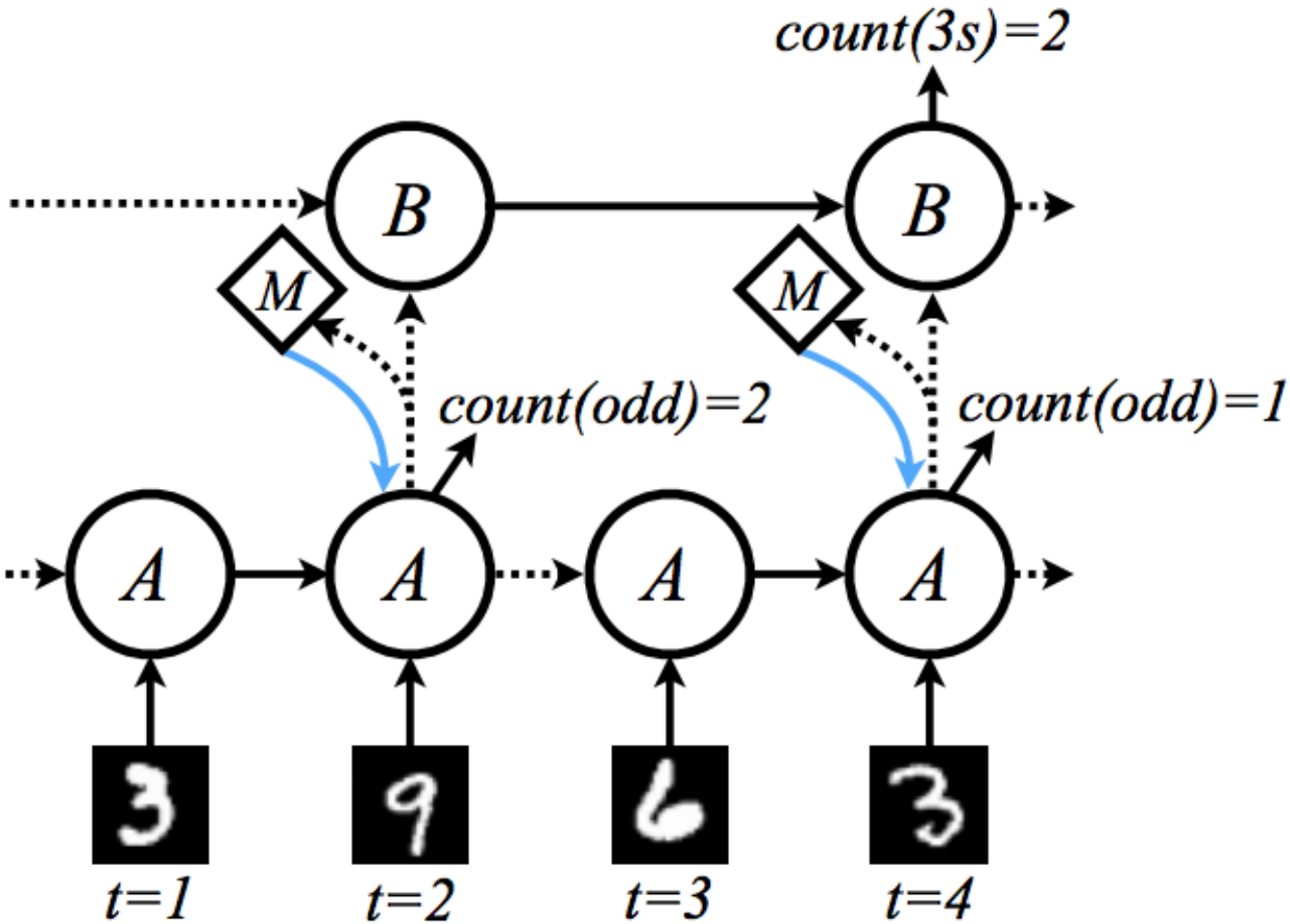
Тестирование



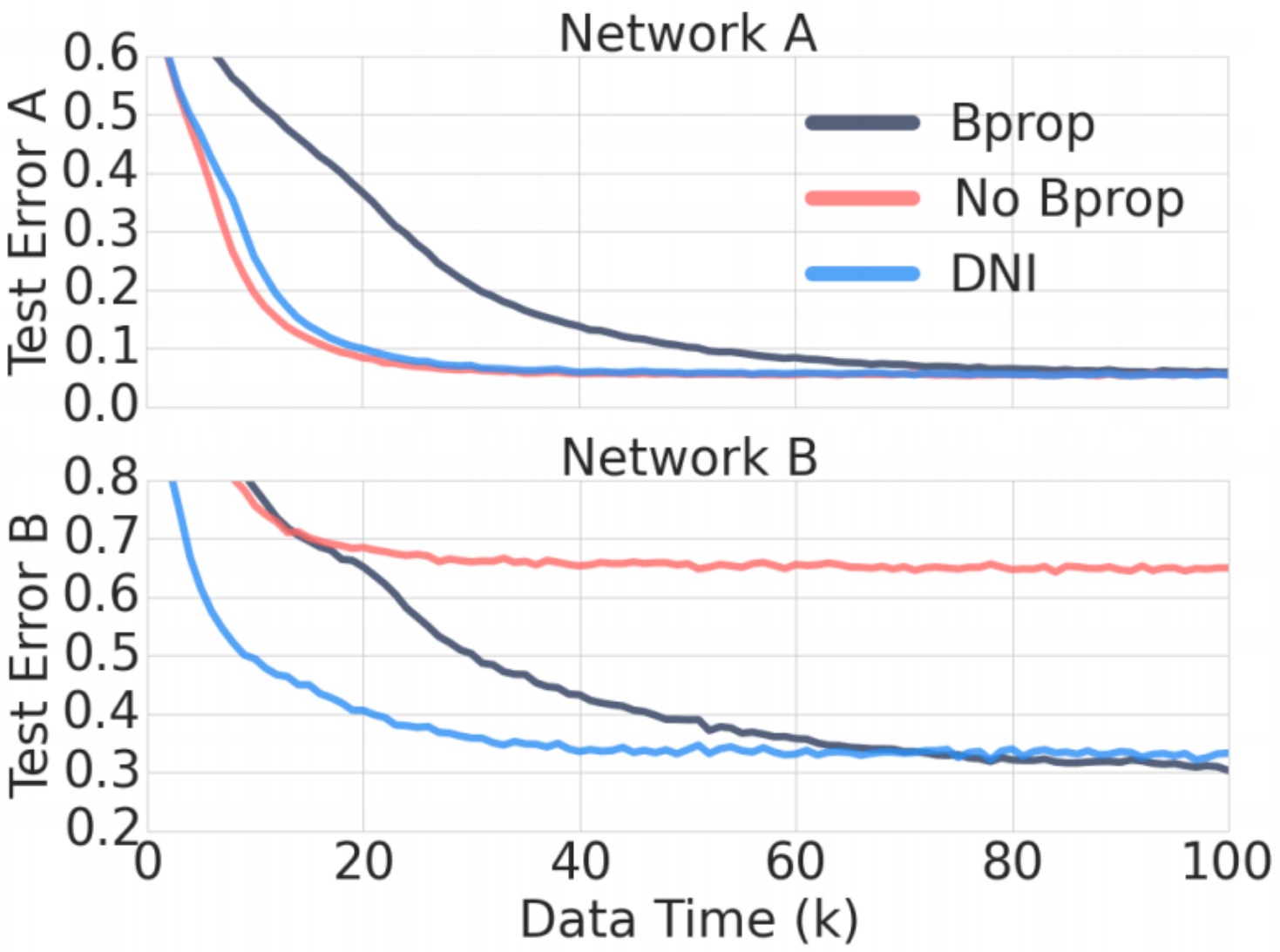
Тестирование

- Датасет: Penn Treebank
- Задача: предсказание следующего символа
- BPS: bits-per-character $-\log_2 P(X_{t+1}|y_t)$ усредненный по всему тексту. Здесь y_t - вход в сеть на t-м шаге. X_{t+1} - предсказание на $t+1$ шаге.
- Data Time: Время(на машине с одним GPU) и количество данных из датасета.

Пример использования в системе из нескольких нейронных сетей



Сравнение эффективности



Статьи на тему

- Decoupled Neural Interfaces using Synthetic Gradients (https://arxiv.org/abs/1608.05343)
- Generating Sequences With Recurrent Neural Networks (https://arxiv.org/abs/1308.0850)