



ОНЛАЙН-ОБРАЗОВАНИЕ

Рекуррентные сети. Обзор.

Профессор, конечно, лопух, но аппаратура при нем.

Артур Кадулин
Преподаватель

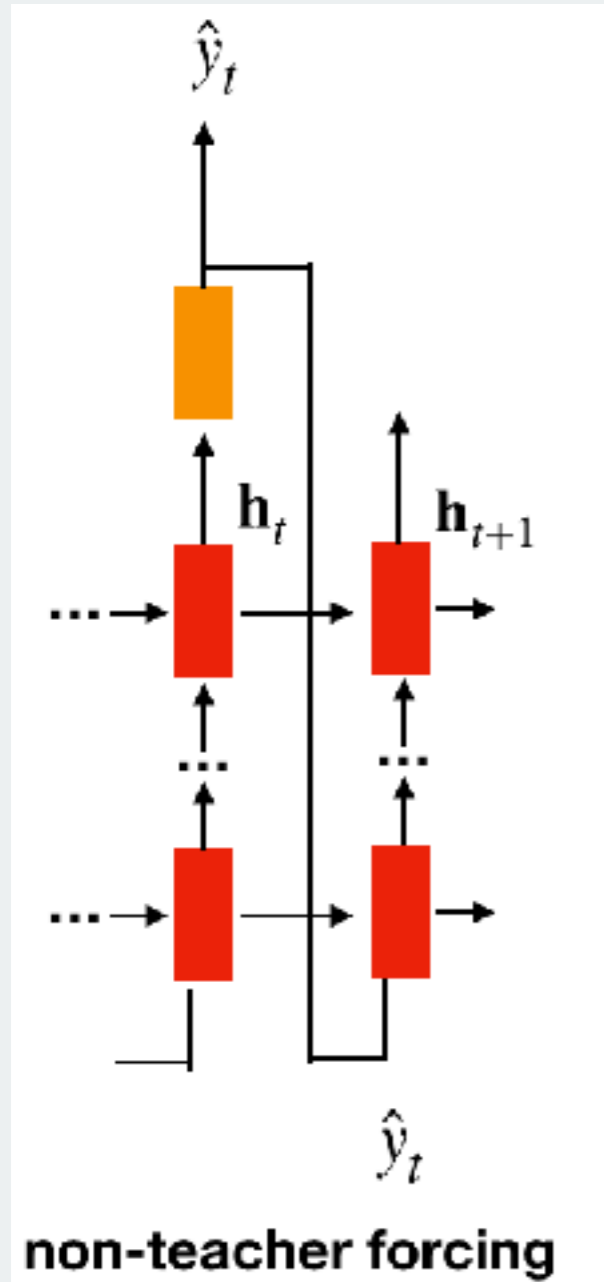


План на сегодня

1. **Учителю виднее!**
2. Обзор
3. Профессор против Учителя
4. Gumbel-softmax trick



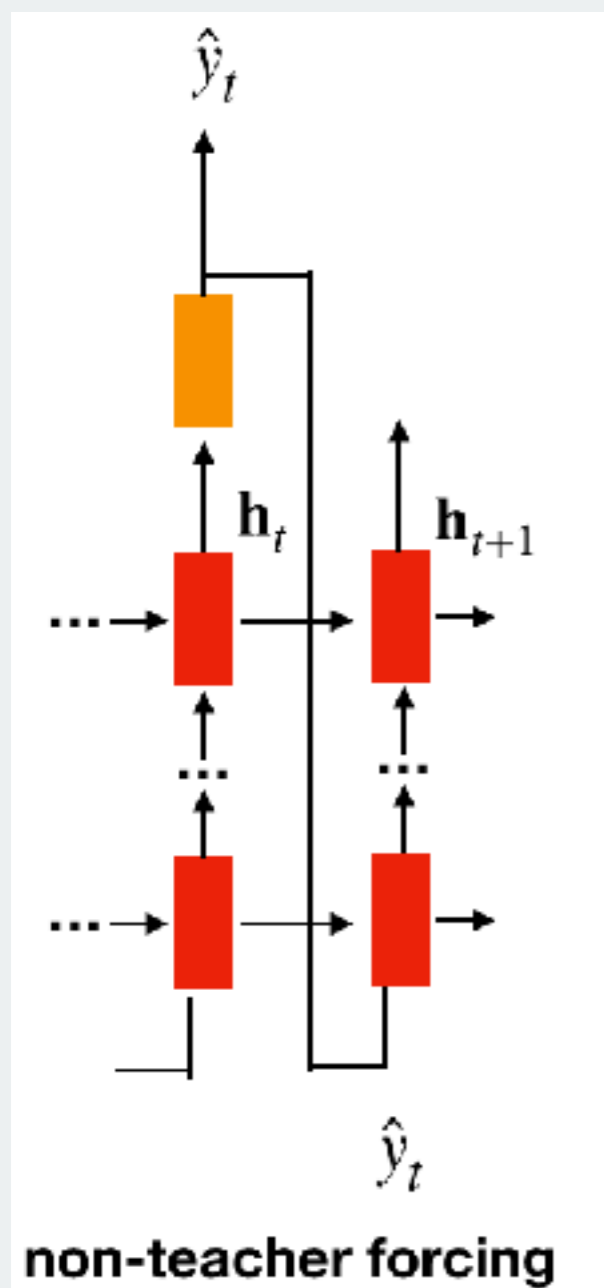
Teacher Forcing



При обучении рекуррентной сети вход очередного шага является выходом предыдущего шага.



Teacher Forcing

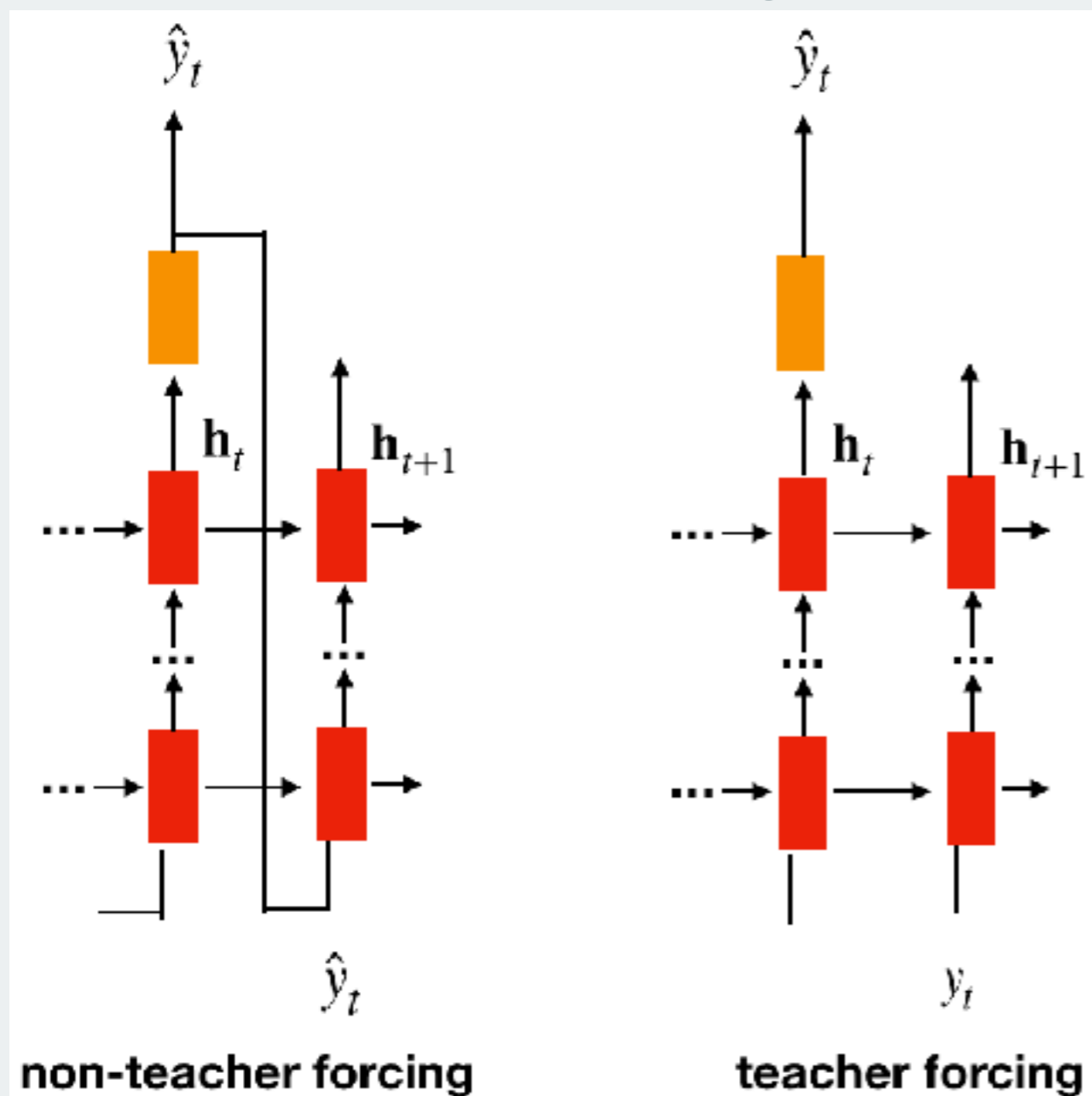


При обучении рекуррентной сети вход очередного шага является выходом предыдущего шага.

Однако если сеть ошиблась уже в первом символе, дальше ошибка будет только расти.



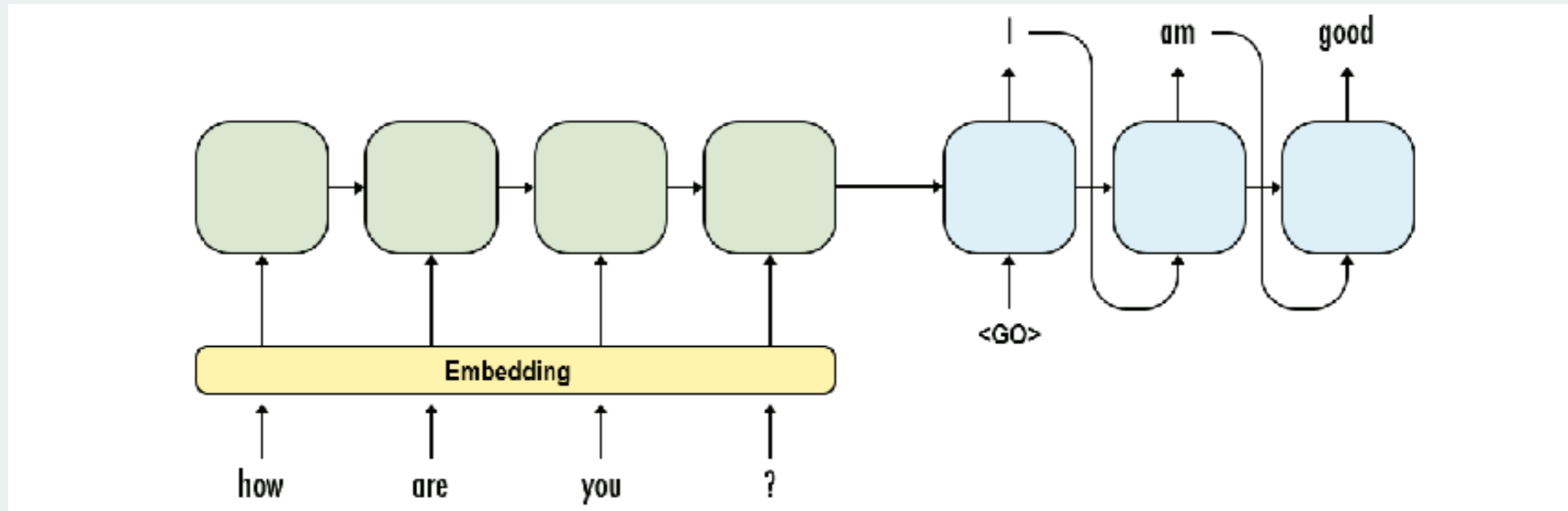
Teacher Forcing



Поэтому при обучении генеративных рекуррентных моделей часто используют технику Teacher Forcing. Если мы знаем какой должна быть последовательность на выходе, то на очередном шаге будем подавать «правильный» символ вместо сгенерированного.



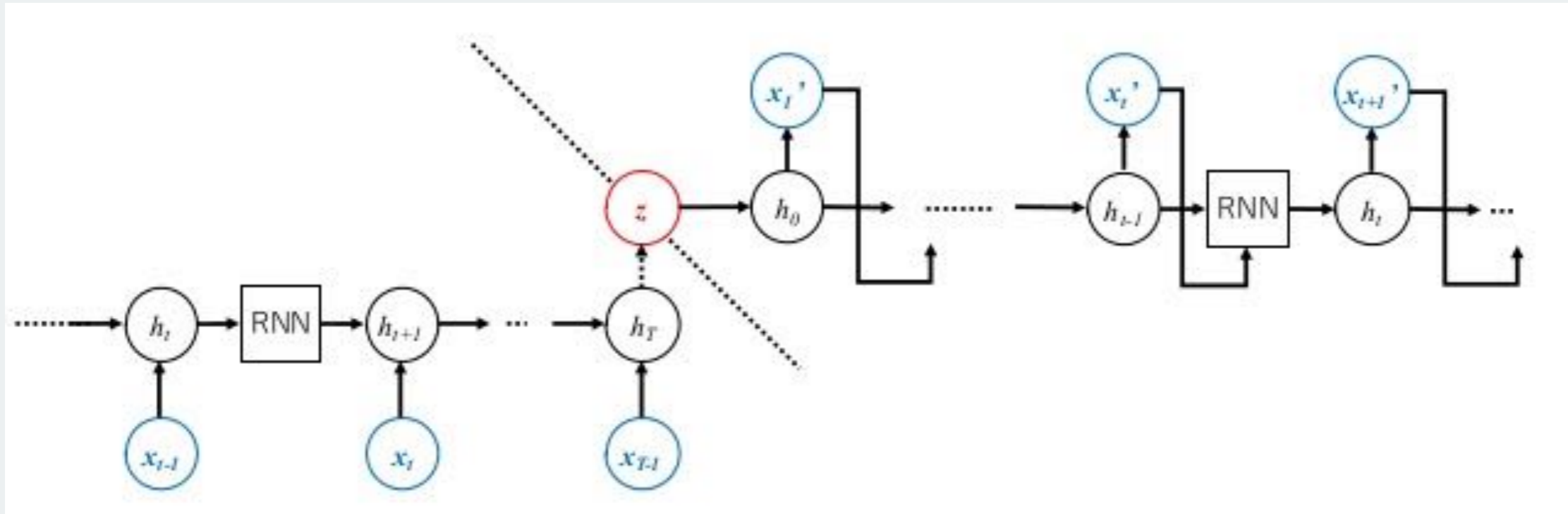
Инференс



При обучении модели используется подход Teacher Forcing, однако во время использования мы генерируем ответную последовательность подавая на вход очередного шага то что на самом деле было сгенерировано сетью на предыдущем шаге (Free Running).



VRAE



Как не трудно догадаться, VRAE — это Seq2Seq с репараметризацией и KL-дивергенцией на латентном слое. Или просто VAE с рекуррентными Энкодером и Декодером. Аналогичным образом можно получить связательный автокодировщик и условные версии этих моделей.

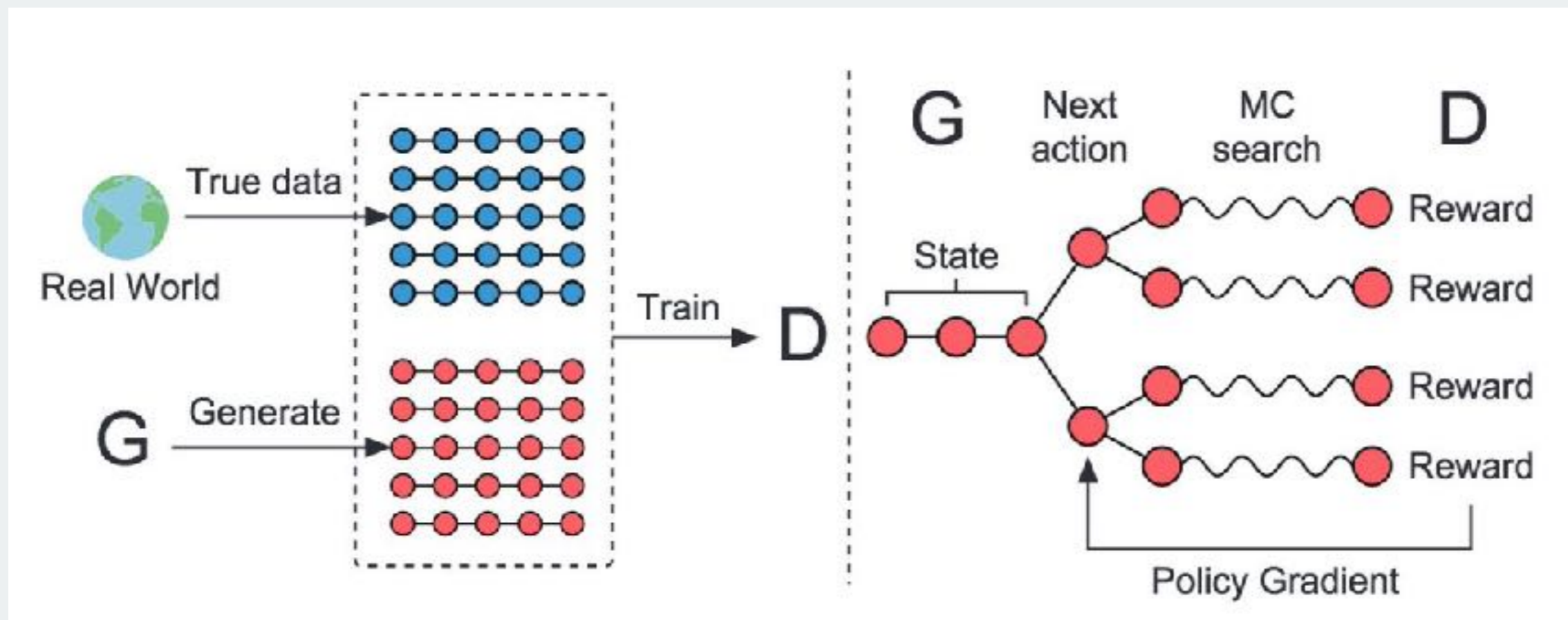


План на сегодня

1. Учителю виднее!
- 2. Обзор**
3. Профессор против Учителя



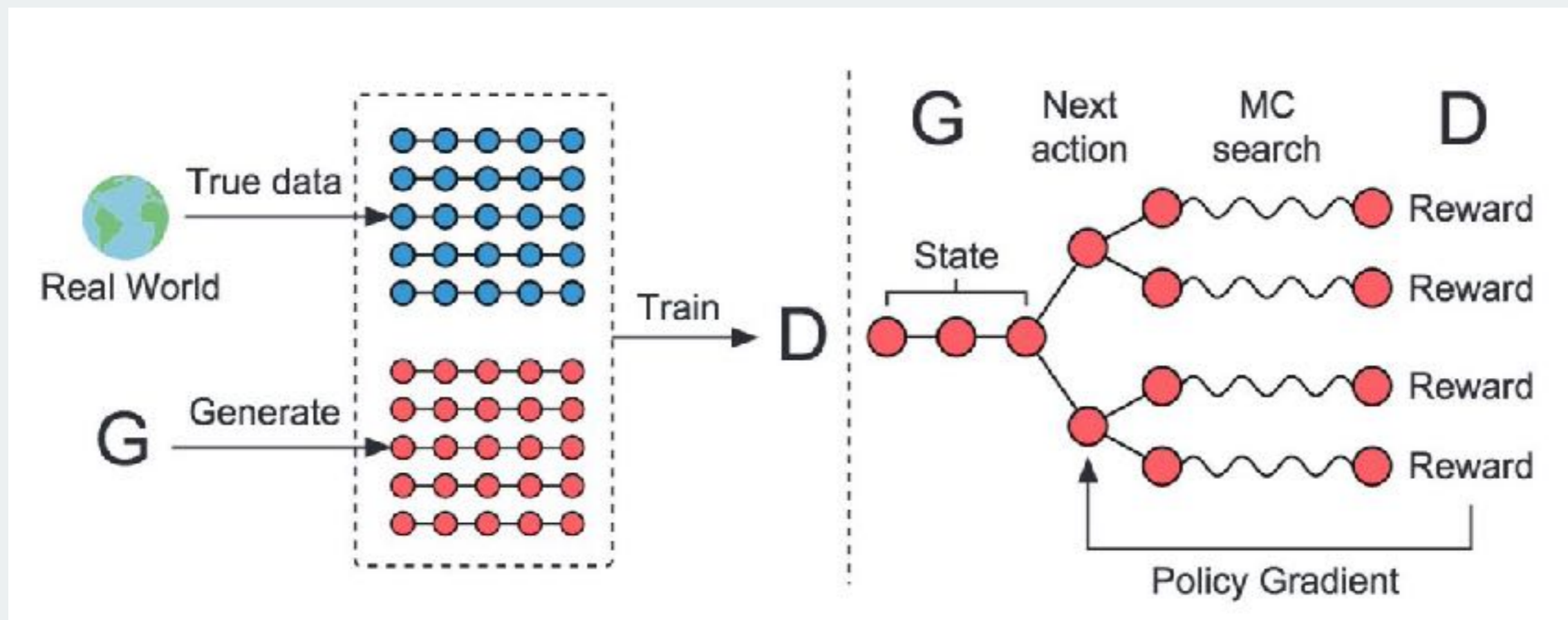
SeqGAN



Так как мы не можем пропустить градиент напрямую сквозь дискретную последовательность, мы используем RL!



SeqGAN



Для очередного шага генерации мы оцениваем «эффективность» каждого из возможных «действий» и используем ее как ошибку.



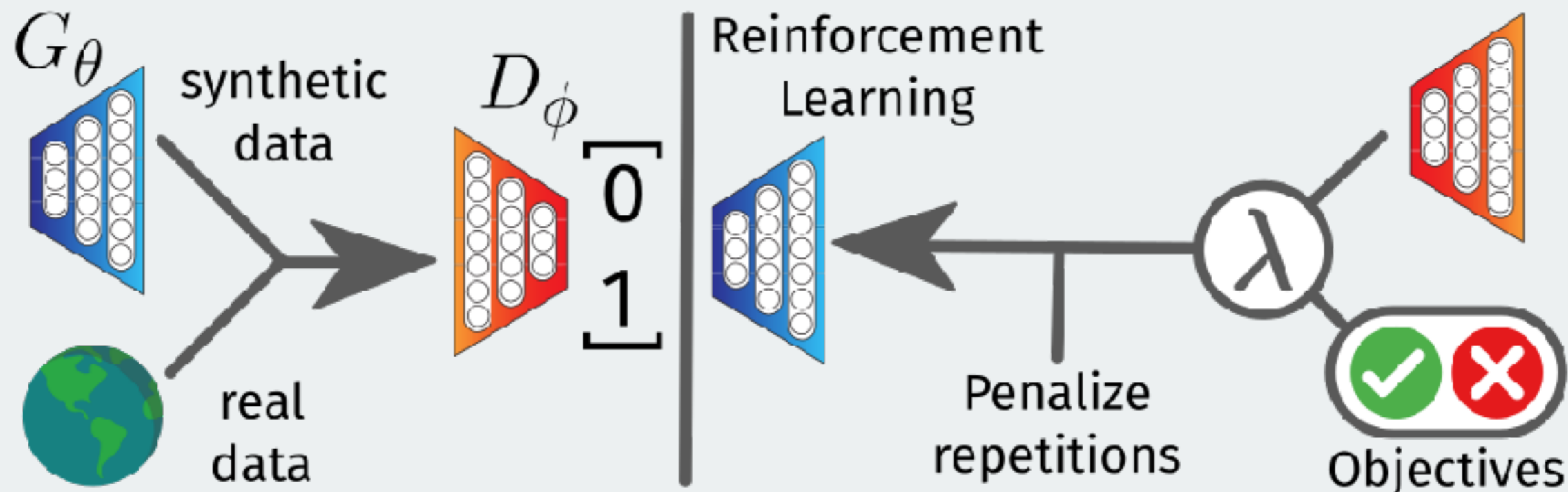
SeqGAN

Реализации: <https://github.com/topics/seqgan>

Авторская реализация: <https://github.com/LantaoYu/SeqGAN>



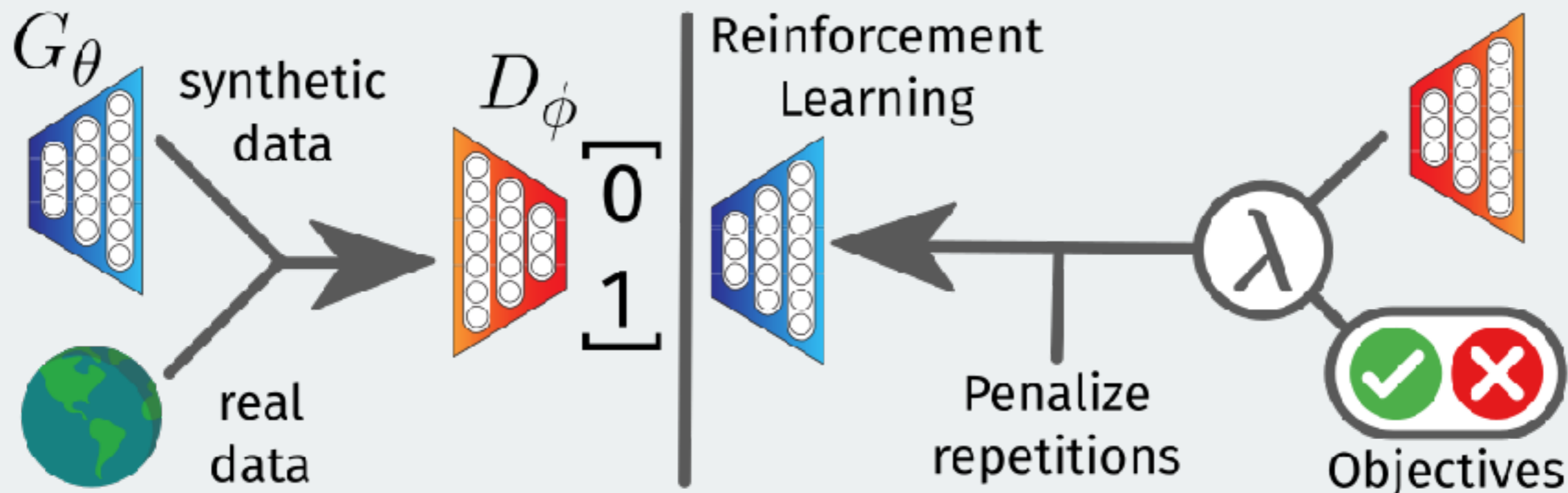
Objective-Reinforced GANs



Раз уж мы считаем ошибку как мат.ожидание награды по free run'ам, то почему бы не добавить в нее дополнительных не дифференцируемых слагаемых?



Objective-Reinforced GANs



В качестве Objectives могут выступать оценки от стороннего софта или ассесоров, валидность или разнообразие объектов в сгенерированном батче и т.д.

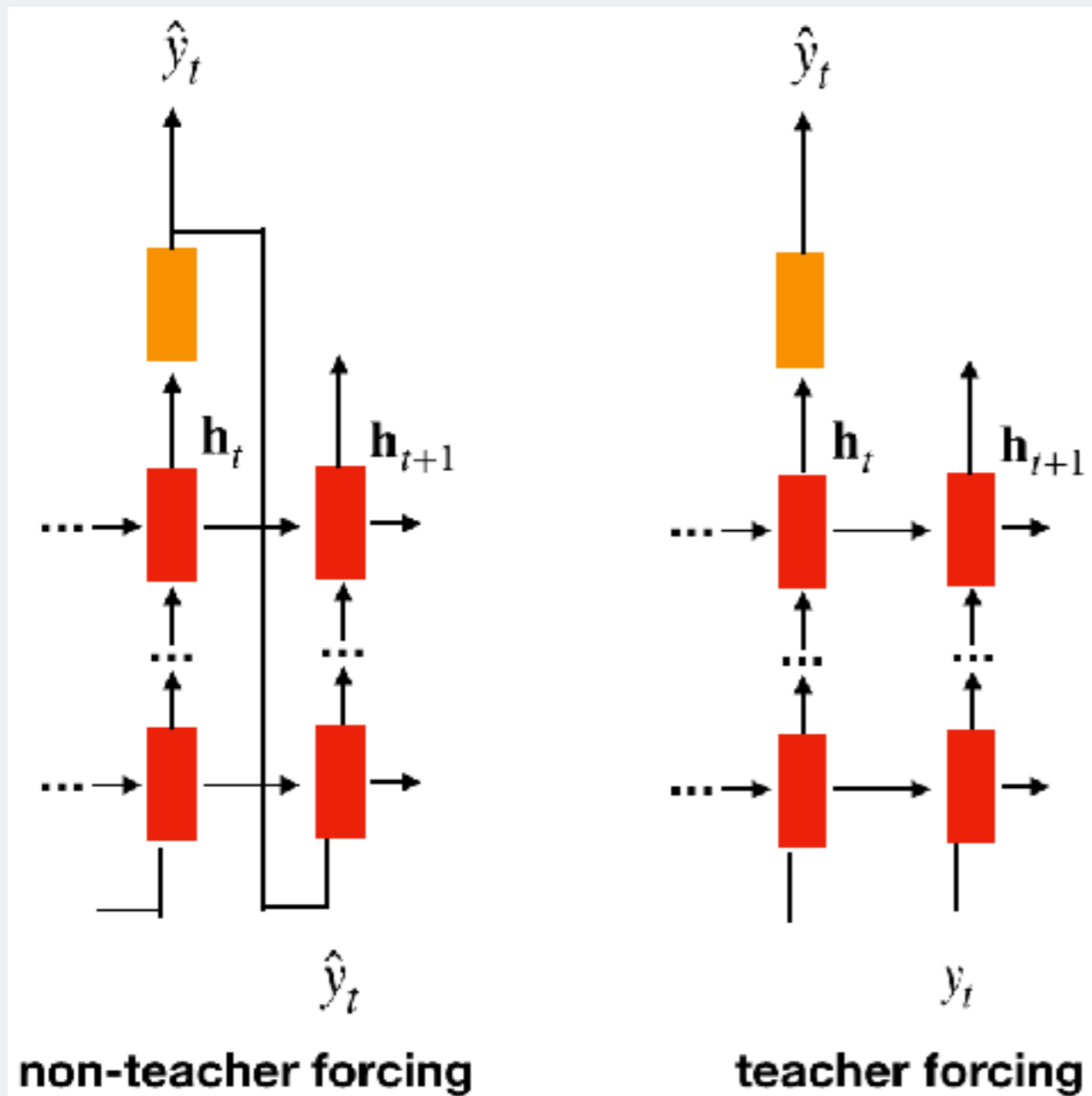


План на сегодня

1. Учителю виднее!
2. Обзор
- 3. Профессор против Учителя**



Professor Forcing

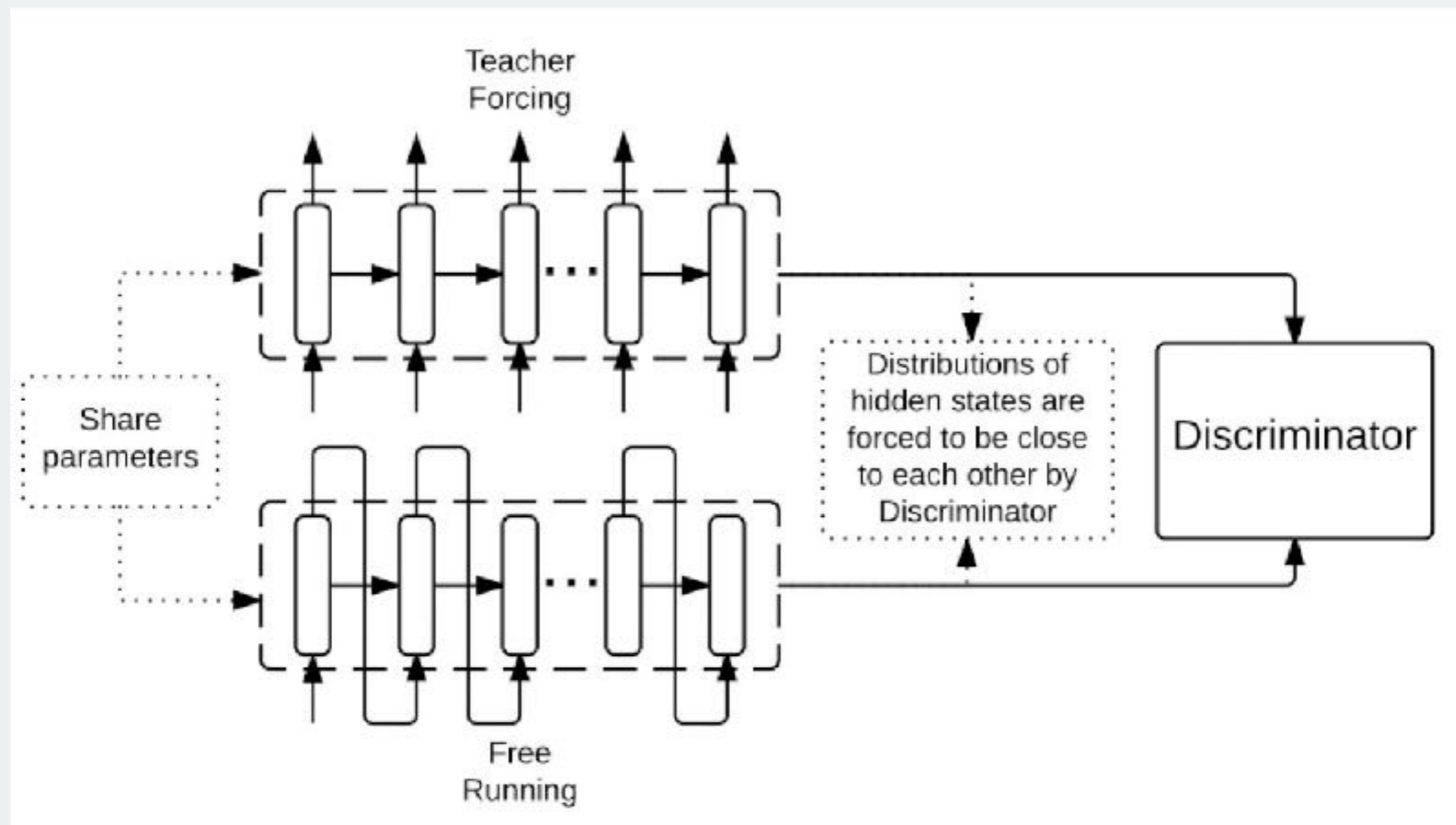


Одна из проблем Teacher Forcing заключается в том, что состояния ячеек во время free running могут сильно отличаться от состояний при использовании teacher forcing. Т.е. во время генерации новых примеров мы работаем в другой области пространства.

Как это исправить?



Professor Forcing



Gumbel-trick

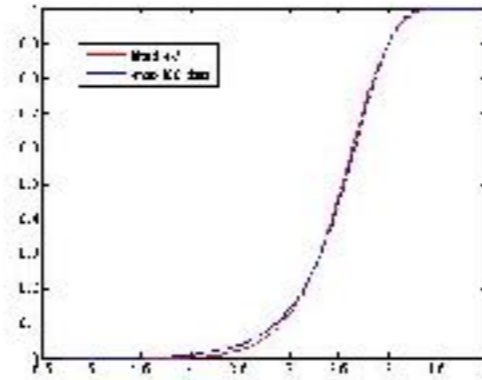
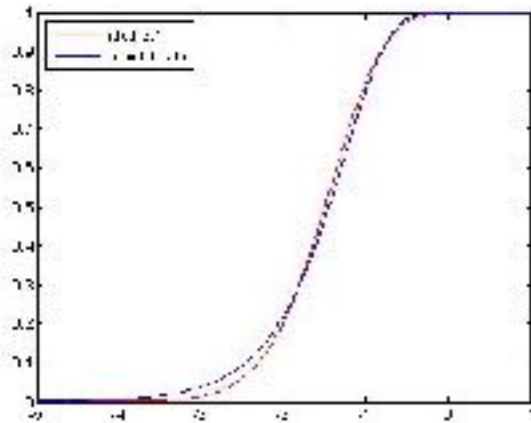
Gumbel distribution

- PDF $= \frac{1}{\beta} \exp(-z - e^{-z})$, $z = \frac{x - \mu}{\beta}$ CDF $= \exp(-e^{-z})$

- Mean, median, mode and variance

Mean $= \mu + \beta\gamma$ median $= \mu - \beta \ln(\ln(2))$ mode $= \mu$

Variance $= \frac{\pi^2}{6} \beta^2$ Euler-Mascheroni constant $\gamma \approx 0.5772$



Распределение Гумбеля используется для оценки вероятности того, что максимальная статистика из выборки достигнет определенного максимума

Это полезно при прогнозировании «экстремальных» событий

<https://slideplayer.com/slide/6411959/>



Gumbel-trick

Reparameterization trick

Как мы поступаем, если в нейросети узлы или слои - случайные?



Gumbel-trick

Reparameterization trick

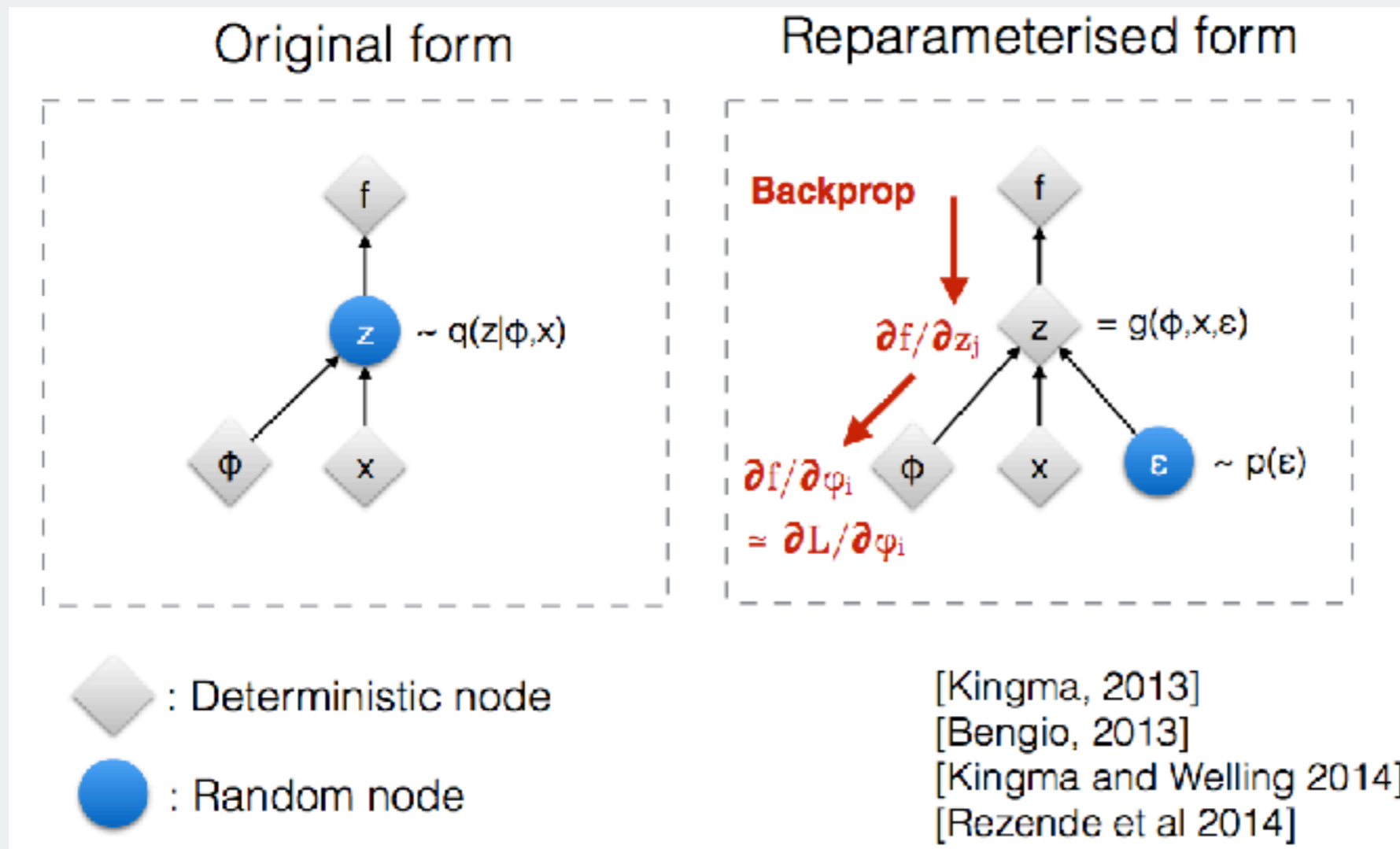
Как мы поступаем, если в нейросети узлы или слои - случайные?

- > Мы не можем вычислять градиент на этих узлах
- > Поэтому вычисляем градиент для μ , Σ распределения в этом узле (или другие параметры)



Gumbel-trick

Reparameterization trick



Gumbel-trick

Reparameterization trick

Непрерывное распределение: ✓
Дискретное распределение: ???



Gumbel-trick

Reparameterization trick

Непрерывное распределение: ✓

Дискретное распределение: ???

Почему нельзя использовать reparameterization trick с дискретным распределением?



Gumbel-trick

Reparameterization trick

Непрерывное распределение: ✓

Дискретное распределение: ???

Почему нельзя использовать reparameterization trick с дискретным распределением?

Нельзя получить (*нетривиальную*) дифференцируемую функцию, которая отображает непрерывное множество в дискретное! Следовательно, мы не сможем считать градиент для reparameterization trick'a



Gumbel-trick

Замечательное свойство распределения Гумбеля

> Вход: дискретное распределение a_i с заданными вероятностями классов i (активации какого-то слоя)

Сгенерируем семплы из распределения так:

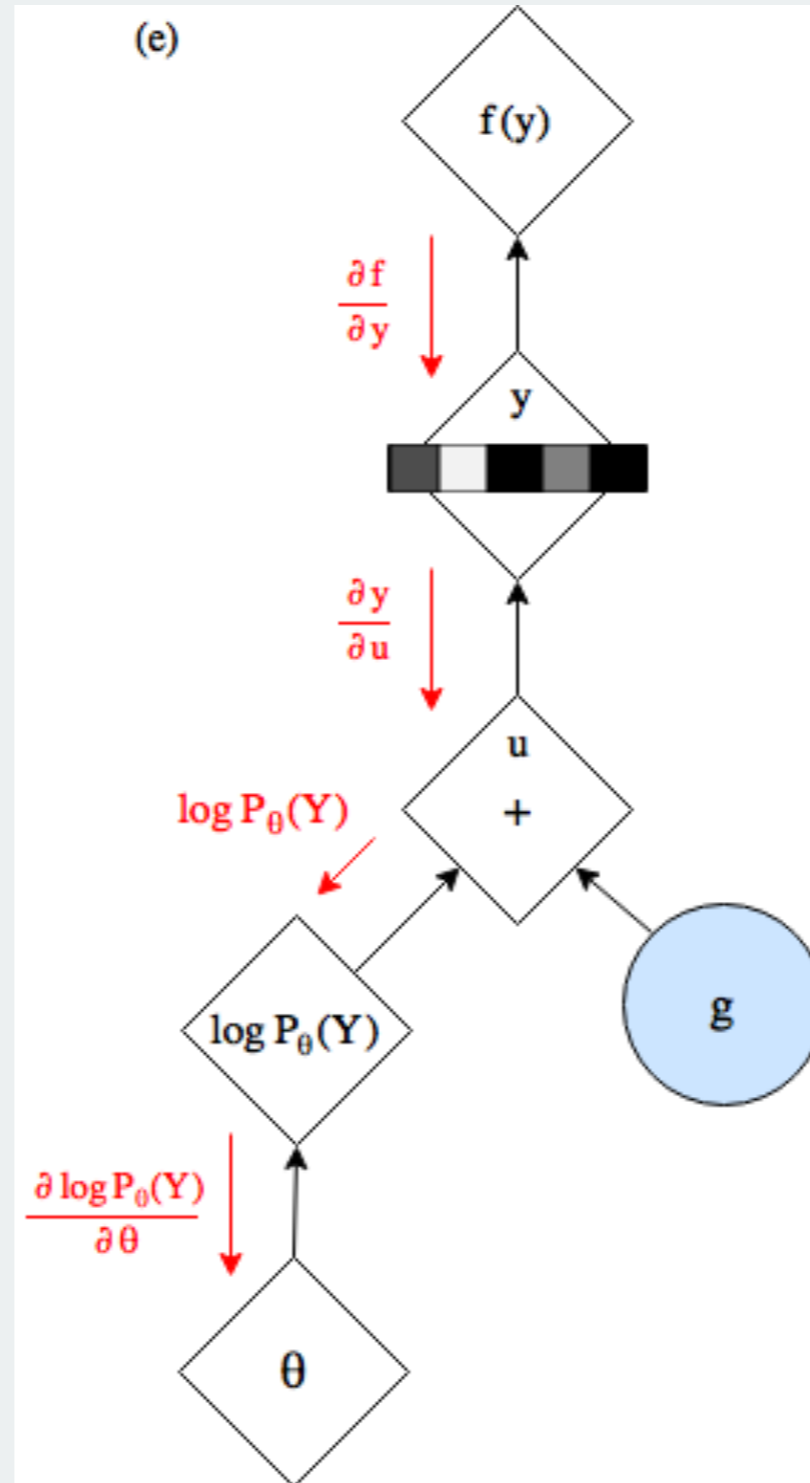
$$z = \text{one-hot}(\arg \max_i [g_i + \log(a_i)])$$

Если сложить значения, взятые из стандартного распределения Гумбеля, и логиты (значения узла, прогнозирующие вероятность i -го класса), а затем вернуть индекс максимального значения, мы получим значение, распределенное как наше оригинальное



Gumbel-trick

Схема трюка



Gumbel-trick

Замечательное свойство распределения Гумбеля

$$z = \text{one-hot}(\arg \max_i [g_i + \log(a_i)])$$

Проблемы?



Gumbel-trick

Замечательное свойство распределения Гумбеля

$$z = \text{one-hot}(\arg \max_i [g_i + \log(a_i)])$$

Проблемы?

> arg-max недифференцируем

Решения?



Gumbel-trick

Замечательное свойство распределения Гумбеля

$$z = \text{one-hot}(\arg \max_i [g_i + \log(a_i)])$$

Проблемы?

> arg-max недифференцируем

Решения?

Использовать softmax для того, чтобы приближенно моделировать arg-max + one-hot.

Подробности в коде!





Спасибо
за внимание!