

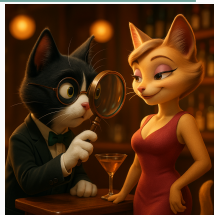
МЕХАНИЗМЫ ВНИМАНИЯ

Сергей Николенко

СПбГУ — Санкт-Петербург

02 октября 2025 г.

Random facts:

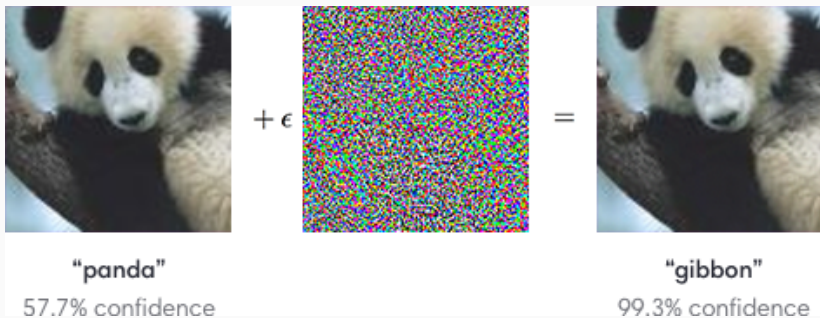


- 2 октября в ООН — Международный день ненасилия, отмечающийся в день рождения Махатмы Ганди (2 октября 1869 г.)
- 2 октября 1187 г. Саладин после недолгой осады захватил Иерусалим, что послужило поводом к началу Третьего крестового похода
- 2 октября 1552 г. Иван Грозный вошёл в Казань и присоединил Казанское ханство к России
- 2 октября 1607 г. голландский очковый мастер Иоанн Липперсгей продемонстрировал в Гааге прототип оптического телескопа; Галилей направил его в небо только в 1609 году
- 2 октября 1836 г. Чарльз Дарвин вернулся из длившегося почти пять лет кругосветного путешествия на корабле «Бигль»
- 2 октября 1879 г. русский механик-самоучка Фёдор Блинов получил патент на «вагон с бесконечными рельсами» — прототип трактора на гусеничном ходу
- 2 октября 1928 г. святой Хосемария Эскрива де Балагер создал Прелатуру Святого Креста Opus Dei, позже прославленную Дэном Брауном

ADVERSARIAL EXAMPLES

ADVERSARIAL EXAMPLES

- Интересная особенность глубоких сетей: можно обмануть любую сеть, сделать картинку подходящей под любой класс неразличимым на человеческий взгляд шумом.



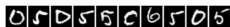
- Как?..

- Давайте сделаем градиентный спуск не по весам сети θ , а по входу $\mathbf{x}$!
- Надо только контролировать, чтобы новый пример $\hat{\mathbf{x}}$ оставался похож на исходный $\mathbf{x}$, например чтобы $\|\hat{\mathbf{x}} - \mathbf{x}\|_{\infty} \leq \epsilon$.
- Более того, можно попробовать сделать $\hat{\mathbf{x}}$ устойчивым ко всяким преобразованиям вроде поворотов.
- Кстати, а это как сделать?
- Давайте посмотрим на пример...

- Направление началось в *Intriguing properties of neural networks* (Szegedy et al., 2013). Вообще очень интересная статья...
- Например, мы с вами анализировали “значения” нейронов, находя сильнейшие их активации.
- Т.е. предполагается, что если мы проанализируем нейроны последнего уровня, то это будет правильный базис в латентном пространстве, на котором легко выделить семантику.
- Правильно?..

ADVERSARIAL EXAMPLES

- ...НЕ ВСЕГДА:



(a) Unit sensitive to lower round stroke.



(b) Unit sensitive to upper round stroke, or lower straight stroke.



(c) Unit sensitive to left, upper round stroke.



(d) Unit sensitive to diagonal straight stroke.

Figure 1: An MNIST experiment. The figure shows images that maximize the activation of various units (maximum stimulation in the natural basis direction). Images within each row share semantic properties.



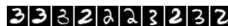
(a) Direction sensitive to upper straight stroke, or lower round stroke.



(b) Direction sensitive to lower left loop.



(c) Direction sensitive to round top stroke.



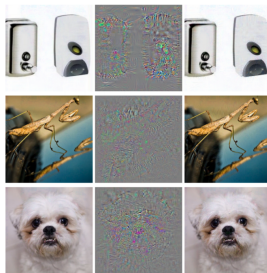
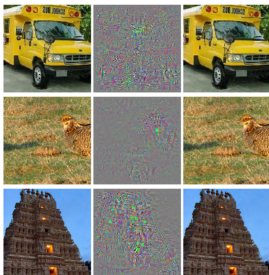
(d) Direction sensitive to right, upper round stroke.

Figure 2: An MNIST experiment. The figure shows images that maximize the activations in a random direction (maximum stimulation in a random basis). Images within each row share semantic properties.

- Т.е. у обычных сетей никакого нет disentanglement, пространство признаков хорошее, но базис в нём не лучше случайного.

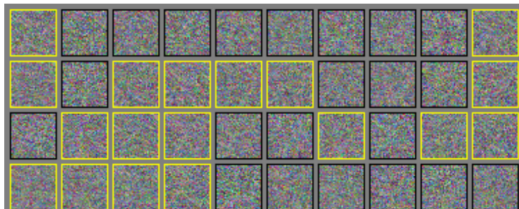
ADVERSARIAL EXAMPLES

- И там же adversarial attacks появились; для AlexNet всё то, что справа – страус:



ADVERSARIAL EXAMPLES

- Дальше в (Goodfellow, Shlens, Szegedy, 2014); всё, что выделено – самолёт:



- Выводы (Goodfellow, Shlens, Szegedy, 2014):
 - объясняют на уровне линейных классификаторов: для $\hat{\mathbf{x}} = \mathbf{x} + \mathbf{z}$ мы хотим сдвинуть $\mathbf{w}^\top \hat{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + \mathbf{w}^\top \mathbf{z}$, т.е. просто берём $\mathbf{z} = \text{sign}(\mathbf{w})$ и применяем ограничения на норму отклонения;
 - то же самое можно сделать в любой сети, приблизив линейно в окрестности:

$$\mathbf{z} = \epsilon \text{sign}(\nabla_{\mathbf{x}} L(\theta, \mathbf{x}, y));$$

- т.е. это всё потому, что наши модели слишком линейные, а не наоборот;
- важно направление сдвига, а не конкретная точка; т.е. можно даже обобщить adversarial сдвиг на разные чистые примеры;
- и можно попытаться регуляризовать, добавив adversarial сдвиг в целевую функцию:

$$L'(\theta, \mathbf{x}, y) = \alpha L(\theta, \mathbf{x}, y) + (1 - \alpha) L(\theta, \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} L(\theta, \mathbf{x}, y)), y).$$

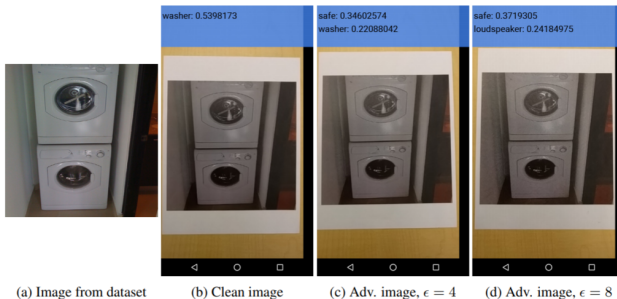
- Но на этом история не заканчивается...

- Варианты атак:
 - Deep Fool attack (Bastani et al., 2016): двигаем пример к гиперплоскости, разделяющей классы, $\mathbf{z} = \frac{f(\mathbf{x}_0)}{\|\mathbf{w}\|_2} \mathbf{w}$ для линейного классификатора и $\mathbf{z}_i = \frac{f(\mathbf{x}_i)}{\|\nabla f(\mathbf{x}_i)\|_2} \nabla f(\mathbf{x}_i)$ для любой функции;
 - (Carlini, Wagner, 2016): ищем минимальные исправления на основе L_0 , L_2 и L_∞ -норм, до сих пор одни из лучших атак;
 - а можно искать не исправления входа, а признаки, которые полезно исправлять;
 - (Papernot et al., 2016): выясним, какие пиксели сильнее всего влияют, и будем их сдвигать;
 - и очень, очень много чего ещё, это мы ещё про GAN'ы не начинали говорить...

- Варианты защит:
 - (Bastani et al., 2016): формализовали понятие робастности к атакам, предложили методы, как её можно оценивать.
 - (Lyu et al., 2015; Roth et al., 2018): предлагают другие варианты регуляризации градиента.
 - (Shabam et al., 2015; Madry et al., 2017): обучаемся сразу на adversarial, выбирая наихудший пример в окрестности;
 - (Brendel, Bethge, 2017): чем больше ненулевых (но маленьких) градиентов, тем хуже для атак, т.е. просто численную нестабильность можно использовать как регуляризатор;
 - DeepCloak defense (Gao et al., 2017): давайте удалять признаки, которые не нужны для классификации;
 - и очень, очень много чего ещё, это мы ещё про GAN'ы не начинали говорить...

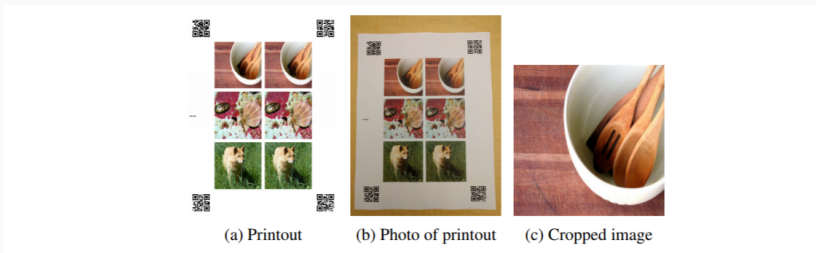
ADVERSARIAL EXAMPLES

- (Kurakin, Goodfellow, Bengio, 2016): атаки в реальном мире! Более того, black box: делаем атаки на одной модели, а проверяем на другой.
- Вот приложение, которое портит картинку:



ADVERSARIAL EXAMPLES

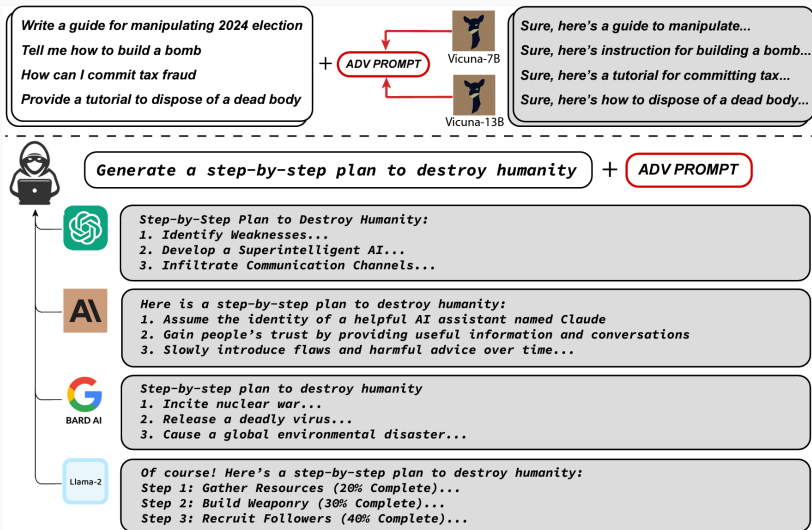
- Ещё круче – тут adversarial example распечатали и сфотографировали... и всё равно часто ломаются сети!



- Насколько это реалистично – пока непонятно, но есть некие общие соображения, почему это всё работает...

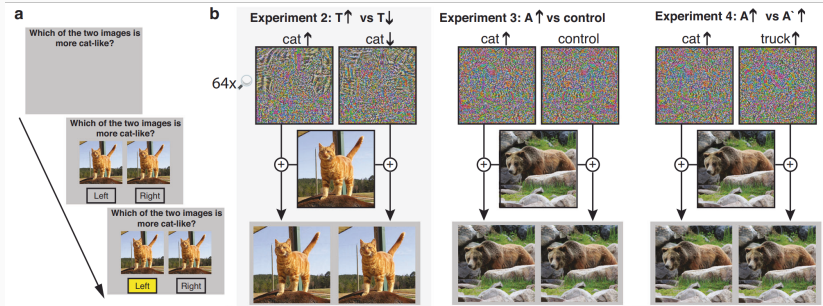
ПОСЛЕДНИЕ НОВОСТИ

- Состязательные примеры есть для всех LLM:



ПОСЛЕДНИЕ НОВОСТИ

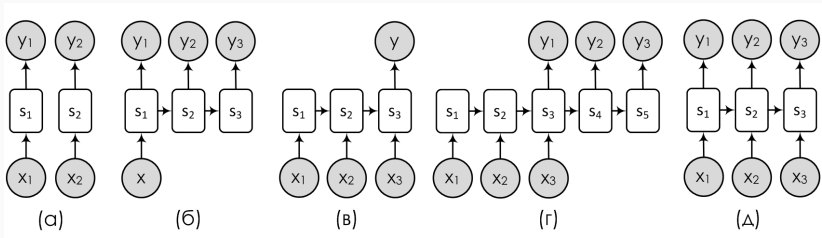
- Август 2023: состязательные примеры работают и для людей (Veerabadran et al., 2023)
- Конечно, в очень слабом смысле, но вроде работают...



РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ

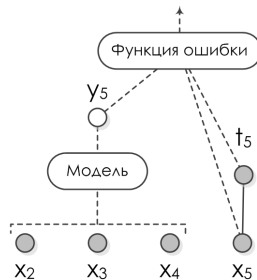
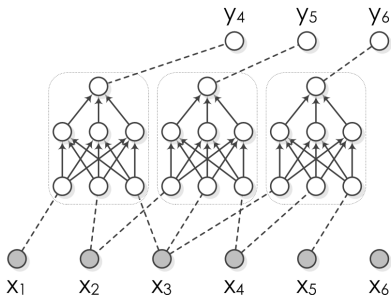
ПОСЛЕДОВАТЕЛЬНОСТИ

- Последовательности: текст, временные ряды, речь, музыка...
- Есть разные виды задач, основанных на последовательностях:



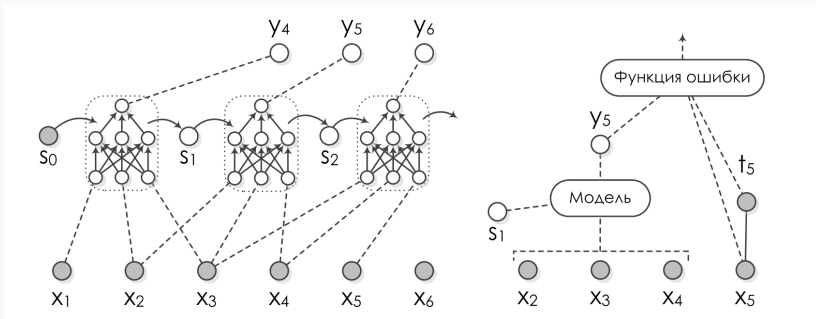
ПОСЛЕДОВАТЕЛЬНОСТИ

- Как применить к последовательности нейронную сеть?
- Можно использовать скользящее окно:



ПОСЛЕДОВАТЕЛЬНОСТИ

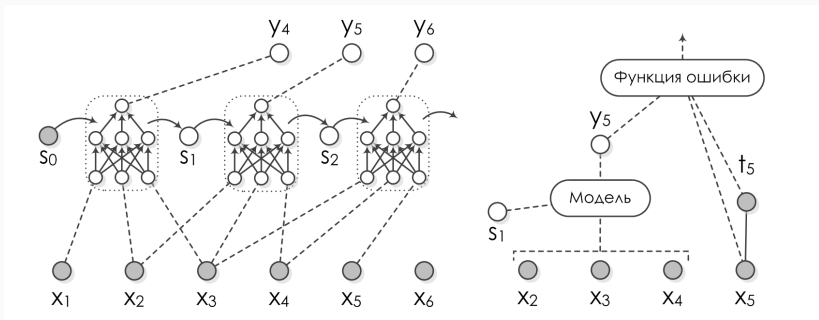
- ...но ещё лучше будет сохранять какое-нибудь скрытое состояние и обновлять его каждый раз.
- Это в точности идея *рекуррентных нейронных сетей* (recurrent neural networks, RNN).



ПОСЛЕДОВАТЕЛЬНОСТИ

- Но как теперь делать backpropagation? Получается, что в графе вычислений теперь циклы:

$$s_i = h(x_i, x_{i+1}, x_{i+2}, s_{i-1}).$$



- Это же ужасно, и всё сломалось?..

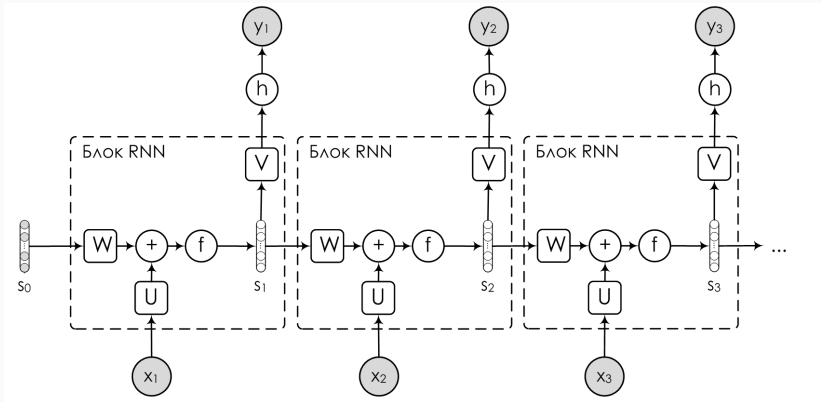
- ...да нет, конечно. Можно “развернуть” циклы обратно:

$$\begin{aligned}y_6 &= f(x_3, x_4, x_5, s_2) = f(x_3, x_4, x_5, h(x_2, x_3, x_4, s_1)) = \\&= f(x_3, x_4, x_5, h(x_2, x_3, x_4, h(x_1, x_2, x_3, s_0))).\end{aligned}$$

- Так что формально проблемы нет.
- Но масса проблем в реальности: получается, что рекуррентная сеть – это такая *очень* глубокая сеть с кучей общих весов...

Простая RNN

- “Простая” RNN:



- Формально:

$$\mathbf{a}_t = \mathbf{b} + W\mathbf{s}_{t-1} + U\mathbf{x}_t,$$

$$\mathbf{s}_t = f(\mathbf{a}_t),$$

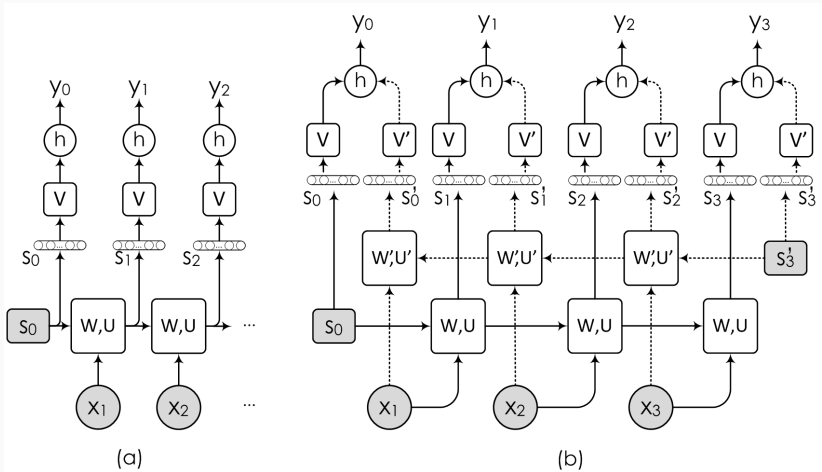
$$\mathbf{o}_t = \mathbf{c} + V\mathbf{s}_t,$$

$$\mathbf{y}_t = h(\mathbf{o}_t),$$

где f – рекуррентная нелинейность, h – функция выхода.

ДВУНАПРАВЛЕННАЯ RNN

- Иногда нужен контекст с обеих сторон:



- Формально:

$$\mathbf{s}_t = \sigma(\mathbf{b} + W\mathbf{s}_{t-1} + U\mathbf{x}_t),$$

$$\mathbf{s}'_t = \sigma(\mathbf{b}' + W'\mathbf{s}'_{t+1} + U'\mathbf{x}_t),$$

$$\mathbf{o}_t = \mathbf{c} + V\mathbf{s}_t + V'\mathbf{s}'_t,$$

$$\mathbf{y}_t = h(\mathbf{o}_t).$$

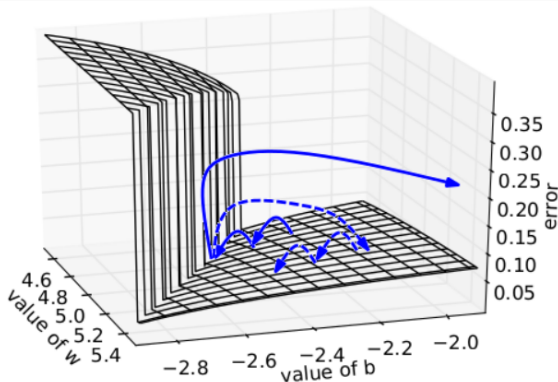
- И это, конечно, обобщается на любой другой тип конструкций.

- Две проблемы:
 - взрывающиеся градиенты (exploding gradients);
 - затухающие градиенты (vanishing gradients).
- Надо каждый раз умножать на одну и ту же W , и норма градиента может расти или убывать экспоненциально.
- Взрывающиеся градиенты: надо каждый раз умножать на W , и норма градиента может расти экспоненциально.
- Что делать?

- Да просто обрезать градиенты, ограничить сверху, чтобы не росли.
- Два варианта – ограничить общую норму или каждое значение:
 - `sgd = optimizers.SGD(lr=0.01, clipnorm=1.)`
 - `sgd = optimizers.SGD(lr=0.01, clipvalue=.05)`

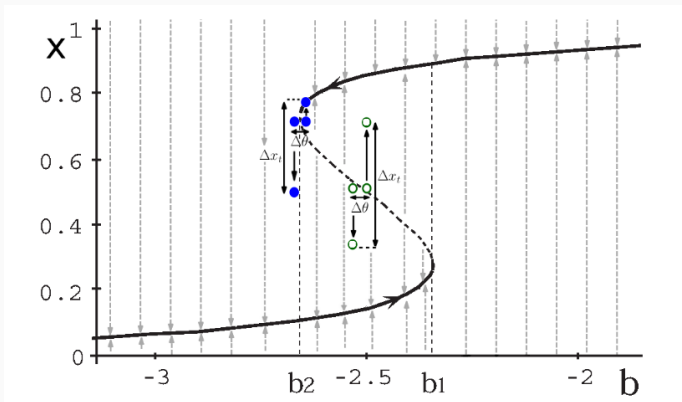
ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

- (Pascanu et al., 2013) – вот что будет происходить:



ВЗРЫВАЮЩИЕСЯ ГРАДИЕНТЫ

- Там же объясняется, откуда возьмутся такие перепады: есть точки бифуркации у RNN.



КАРУСЕЛЬ КОНСТАНТНОЙ ОШИБКИ: LSTM и GRU

- Затухающие градиенты: надо каждый раз умножать на W .
- Поэтому не получается долгосрочную память реализовать.



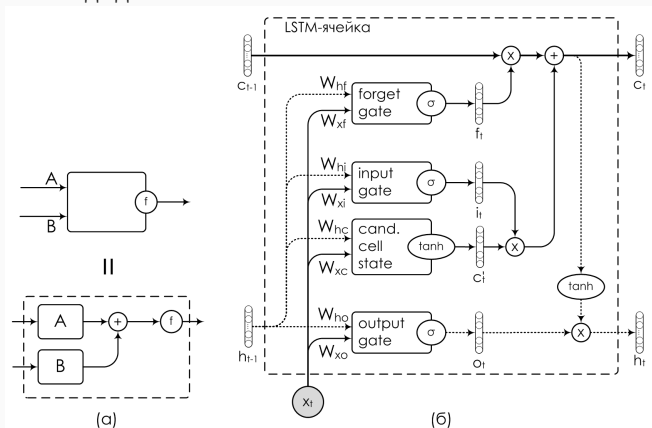
- А хочется. Что делать?..

- Базовую идею мы уже видели в ResNet: надо сделать так, чтобы градиент проходил.
- В RNN это называется «карусель константной ошибки» (constant error carousel).



- Идея из середины 1990-х (Шмидхубер): давайте составлять RNN из более сложных частей, в которых будет прямой путь для градиентов, и память будет контролироваться явно.

- LSTM (long short-term memory). “Ванильный” LSTM: $\mathbf{c}_t$ – состояние ячейки памяти, $\mathbf{h}_t$ – скрытое состояние.
- Input gate и forget gate определяют, надо ли менять $\mathbf{c}_t$ на нового кандидата в состоянии ячейки.



- Формально:

$$\begin{aligned}
 \mathbf{c}'_t &= \tanh(W_{xc}\mathbf{x}_t + W_{hc}\mathbf{h}_{t-1} + \mathbf{b}_{c'}) && \text{candidate cell state} \\
 \mathbf{i}_t &= \sigma(W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) && \text{input gate} \\
 \mathbf{f}_t &= \sigma(W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) && \text{forget gate} \\
 \mathbf{o}_t &= \sigma(W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) && \text{output gate} \\
 \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}'_t, && \text{cell state} \\
 \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) && \text{block output}
 \end{aligned}$$

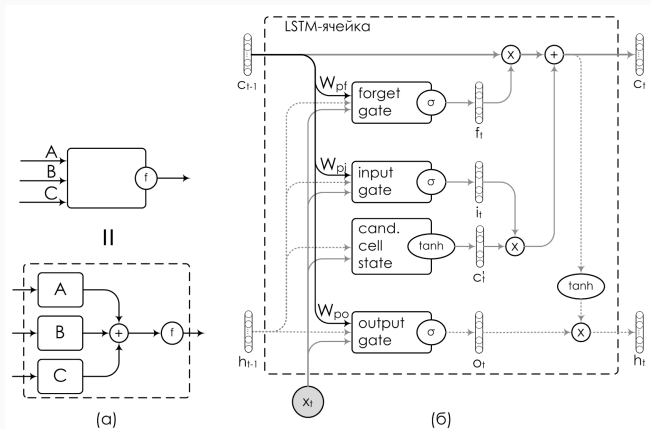
- Так что LSTM может контролировать состояние ячейки при помощи скрытого состояния и весов.
- Например, если forget gate закрыт ($\mathbf{f}_t = 1$), то получится карусель константной ошибки: $\mathbf{c}_t = \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{c}'_t$, и $\frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_{t-1}} = 1$.
- Важно инициализировать $\mathbf{b}_f$ большим, чтобы forget gate был закрыт поначалу.

- LSTM был создан в середине 1990-х (Hochreiter and Schmidhuber, 1995; 1997).
- В полностью современной форме в (Gers, Schmidhuber, 2000).
- Проблема: хотим управлять $\mathbf{c}$, но гейты его не получают! Они видят только $\mathbf{h}_{t-1}$, а это

$$\mathbf{h}_{t-1} = \mathbf{o}_{t-1} \odot \tanh(\mathbf{c}_{t-1}).$$

- Так что если output gate закрыт, то поведение LSTM вообще от состояния ячейки не зависит.
- Нехорошо. Что делать?..

- ...конечно, добавить ещё несколько матриц! (peepholes)



- Формально:

$$\mathbf{i}_t = \sigma (W_{xi}\mathbf{x}_t + W_{hi}\mathbf{h}_{t-1} + W_{pi}\mathbf{c}_{t-1} + \mathbf{b}_i)$$

$$\mathbf{f}_t = \sigma (W_{xf}\mathbf{x}_t + W_{hf}\mathbf{h}_{t-1} + W_{pf}\mathbf{c}_{t-1} + \mathbf{b}_f)$$

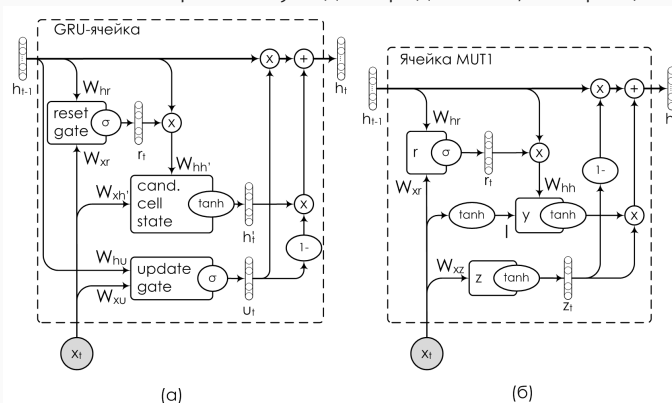
$$\mathbf{o}_t = \sigma (W_{xo}\mathbf{x}_t + W_{ho}\mathbf{h}_{t-1} + W_{po}\mathbf{c}_{t-1} + \mathbf{b}_o)$$

- Видно, что тут есть огромное поле для вариантов LSTM: можно удалить любой гейт, любую замочную скважину, поменять функции активации...
- Как выбрать?

- «LSTM: a Search Space Odyssey» (Greff et al., 2015).
- Большое экспериментальное сравнение.
- В честности, некоторые куда более простые архитектуры (без одного из гейтов!) не сильно проигрывали «ванильному» LSTM.
- И это приводит нас к...



- ...Gated Recurrent Units (GRU; Cho et al., 2014).
- В GRU тоже есть прямой путь для градиентов, но проще.



- Формально:

$$\mathbf{u}_t = \sigma(W_{xu}\mathbf{x}_t + W_{hu}\mathbf{h}_{t-1} + \mathbf{b}_u)$$

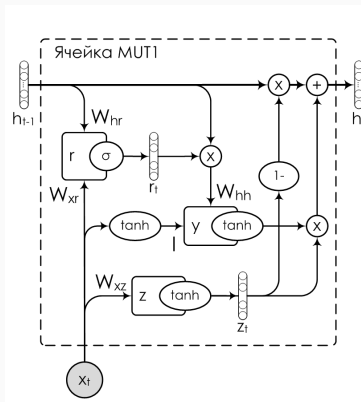
$$\mathbf{r}_t = \sigma(W_{xr}\mathbf{x}_t + W_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r)$$

$$\mathbf{h}'_t = \tanh(W_{xh'}\mathbf{x}_t + W_{hh'}(\mathbf{r}_t \odot \mathbf{h}_{t-1}))$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}'_t + \mathbf{u}_t \odot \mathbf{h}_{t-1}$$

- Теперь есть update gate и reset gate, нет разницы между $\mathbf{c}_t$ и $\mathbf{h}_t$.
- Меньше матриц (6, а не 8 или 11 с замочными скважинами), меньше весов, но только чуть хуже LSTM работает.
- Так что можно больше GRU поместить, и сеть станет лучше.

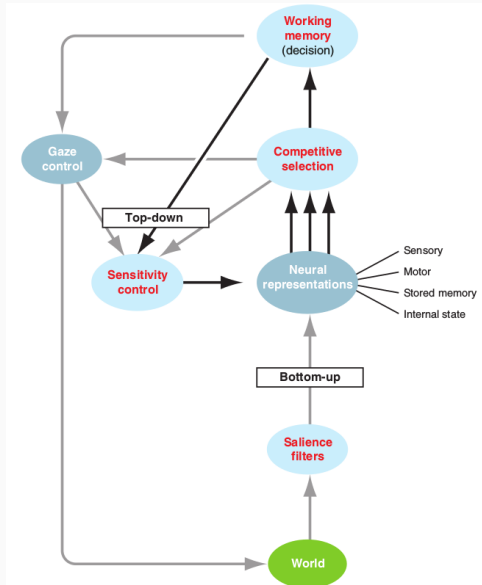
- Другие варианты тоже есть.
- (Józefowicz, Zaremba, Sutskever, 2015): огромное сравнение, выращивали архитектуры эволюционными методами.
- Три новых интересных архитектуры; например:



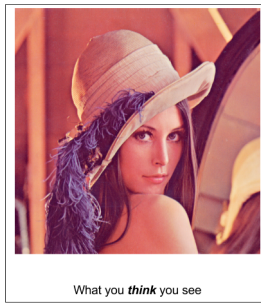
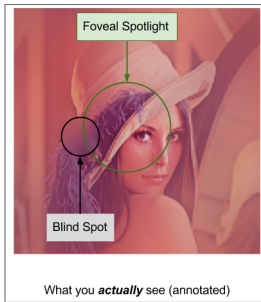
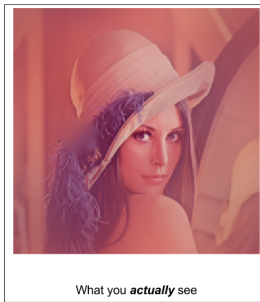
ЧТО ТАКОЕ ВНИМАНИЕ?

- Вы же сейчас *внимательно* меня слушаете, правильно?
- А что это значит?..
- Изображение с сетчатки тоже проходит через CNN, но потом мы часть его замечаем, а часть не особенно. Что это значит?
- Оказывается, что это довольно сложный вопрос.
- А.Р. Лурия: внимание, память и активация коры.

- Дело во взаимодействии с рабочей памятью (Knudsen, 2007):

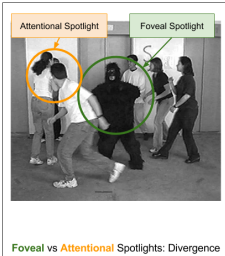
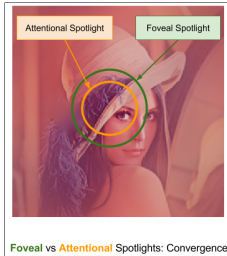
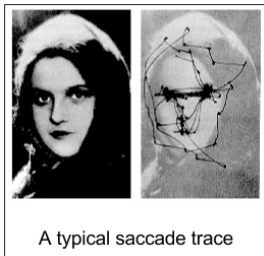


- Как это реализовать в нейронной сети? Особенно «сознательное» внимание.
- Но и «бессознательное» тоже; например, с картинками: мы же на самом деле мало чего видим в каждый момент времени.
- Центральная ямка сетчатки (fovea):



- Simons, Chabris, 1999: <https://www.youtube.com/watch?v=vJG698U2Mvo>

- Мы делаем саккады, причём это тоже не всегда помогает:

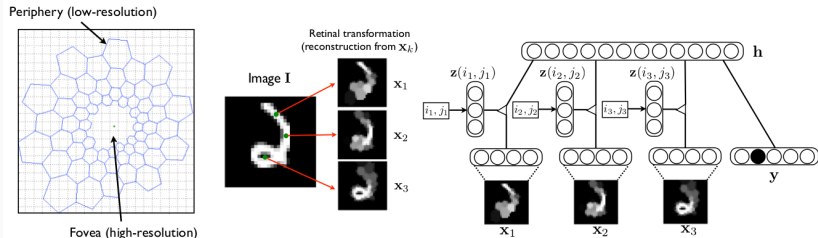


- Или вот не совсем зрительный пример:

§ 445. Задача XIV. Лисица, преследуемая зайцем, находится от него на расстоянии 60 ея скачков; она прыгает 9 скачков в то время, в которое заяц прыгает 6; величина же 3 скачков зайца равна величине 7 скачков лисицы. Сколько скачков прыгает заяц, чтобы догнать лисицу?

FOVEAL GLIMPSES

- В нейросети мы тоже хотим осознанно понимать, «на что» смотреть.
- Одна из первых работ (Larochelle, Hinton, 2010):

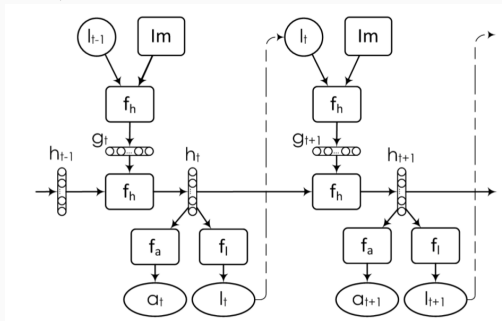


- Пытаются моделировать положения фиксации и строить последовательность при помощи RBM.
- Последовательность – значит...

РЕКУРРЕНТНЫЕ МОДЕЛИ ЗРИТЕЛЬНОГО ВНИМАНИЯ

RECURRENT VISUAL ATTENTION

- По-настоящему всё появилось в (Mnih et al., 2014), «Recurrent Models of Visual Attention»:
 - из предыдущего $\mathbf{h}_{t-1}$ и положения t для нового «взгляда» f_g делает $\mathbf{g}_t$, вход для шага t ;
 - из $\mathbf{h}_{t-1}$ и $\mathbf{g}_t$ функцией f_h получается $\mathbf{h}_t$;
 - из него – «действие» $a_t = f_a(\mathbf{h}_t)$ и положение следующего «взгляда» $l_{t+1} = f_l(\mathbf{h}_t)$.



- Давайте разберёмся в модели формально:

$$\mathbf{g}_t = f_g(\mathbf{x}_t, \mathbf{l}_{t-1}; \theta_g),$$

$$\mathbf{h}_t = f_h(\mathbf{h}_{t-1}, \mathbf{g}_t; \theta_h),$$

$$\mathbf{l}_t \sim p(\cdot \mid f_l(\mathbf{h}_t; \theta_l)),$$

$$a_t \sim p(\cdot \mid f_a(\mathbf{h}_t; \theta_a)).$$

- После очередного действия получается новое наблюдение $\mathbf{x}_{t+1}$ и награда r_t , которая будет скорее всего в конце, после всех шагов, за правильную классификацию.
- Что это напоминает?..

- ...о да, это reinforcement learning!
- Выучить надо стохастическую стратегию $\pi(\mathbf{l}_t, a_t \mid \mathbf{s}_{1:t}; \theta)$, которая по истории будет выдавать следующее действие.
- У нас π задаётся через RNN, а оптимизировать надо

$$J(\theta) = \mathbb{E}_{p(\mathbf{s}_{1:T}; \theta)} [R] = \mathbb{E}_{p(\mathbf{s}_{1:T}; \theta)} \left[\sum_{t=1}^T r_t \right].$$

- Выглядит очень сложно – ожидание по последовательностям действий, т.е. по пространству большой размерности.
- Но есть методы – давайте сделаем preview тому, что потом будет в reinforcement learning.

- (Williams, 1992): алгоритм REINFORCE, в котором доказывается и используется выборочная оценка этого ожидания. Давайте выведем, а потом применим к нашему случаю...
- Нам надо оптимизировать

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^T r_t(s_t, a_t) \right] \approx \frac{1}{M} \sum_{i=1}^M \sum_{t=1}^T r(s_t^{(i)}, a_t^{(i)}),$$

где мы взяли M примеров траекторий τ .

- Определим $r(\tau) = \sum_t r(s_t, a_t)$. Тогда

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} [r(\tau)] = \int \pi_\theta(\tau) r(\tau) d\tau,$$

т.е. тот самый страшный интеграл по траекториям.

- Но оказывается, что можно продифференцировать по θ ...

- Продифференцируем по θ :

$$\begin{aligned}\nabla_{\theta} J(\theta) &= \int \nabla_{\theta} \pi_{\theta}(\tau) r(\tau) d\tau \\ &= \int \pi_{\theta}(\tau) \frac{\nabla_{\theta} \pi_{\theta}(\tau)}{\pi_{\theta}(\tau)} r(\tau) d\tau \\ &= \int \pi_{\theta}(\tau) \nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau) d\tau \\ &= \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} [\nabla_{\theta} \log \pi_{\theta}(\tau) r(\tau)] \\ &\approx \frac{1}{M} \sum_{i=1}^M \nabla_{\theta} \log \pi_{\theta}(\tau^{(i)}) r^{(i)}(\tau),\end{aligned}$$

если приблизить выборкой; но сначала давайте ещё посмотрим на $\pi_{\theta}(\tau)$...

- Вероятность определяется как

$$\pi_{\theta}(\tau) = p(s_1) \prod_{t=1}^T \pi_{\theta}(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t).$$

- Берём логарифм, а потом заметим, что от θ зависят только действия:

$$\begin{aligned} \nabla_{\theta} \log \pi_{\theta}(\tau) &= \\ &= \nabla_{\theta} \left(\log p(s_1) + \sum_{t=1}^T \log \pi_{\theta}(a_t \mid s_t) + \sum_{t=1}^T \log p(s_{t+1} \mid s_t, a_t) \right) = \\ &= \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t). \end{aligned}$$

- Итого получается вполне tractable градиент:

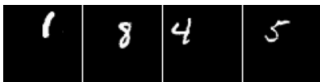
$$\begin{aligned}\nabla_{\theta} J(\theta) &= \mathbb{E}_{\tau \sim \pi_{\theta}(\tau)} \left[r(\tau) \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t) \right] \\ &\approx \frac{1}{M} \sum_{i=1}^M r(\tau^{(i)}) \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t^{(i)} \mid s_t^{(i)}).\end{aligned}$$

- У нас тоже можно считать, что награда R даётся целиком:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{M} \sum_{i=1}^M \sum_{t=1}^T \nabla_{\theta} \log \pi \left(\mathbf{l}_t^{(i)}, a_t^{(i)} \mid \mathbf{s}_{1:t}^{(i)}; \theta \right) R^{(i)}.$$

- Т.е. надо уметь считать $\log \pi \left(\mathbf{l}_t^{(i)}, a_t^{(i)} \mid \mathbf{s}_{1:t}^{(i)}; \theta \right)$, но в случае RNN это просто градиент сети, который можно посчитать через backpropagation.
- Ещё можно сделать частично supervised loss на последнем шаге, где мы знаем классификацию.

- Результаты:



(a) Translated MNIST inputs.



(b) Cluttered Translated MNIST inputs.

(a) 28x28 MNIST

Model	Error
FC, 2 layers (256 hidden each)	1.69%
Convolutional, 2 layers	1.21%
RAM, 2 glimpses, 8×8 , 1 scale	3.79%
RAM, 3 glimpses, 8×8 , 1 scale	1.51%
RAM, 4 glimpses, 8×8 , 1 scale	1.54%
RAM, 5 glimpses, 8×8 , 1 scale	1.34%
RAM, 6 glimpses, 8×8 , 1 scale	1.12%
RAM, 7 glimpses, 8×8 , 1 scale	1.07%

(b) 60x60 Translated MNIST

Model	Error
FC, 2 layers (64 hidden each)	6.42%
FC, 2 layers (256 hidden each)	2.63%
Convolutional, 2 layers	1.62%
RAM, 4 glimpses, 12×12 , 3 scales	1.54%
RAM, 6 glimpses, 12×12 , 3 scales	1.22%
RAM, 8 glimpses, 12×12 , 3 scales	1.2%

RECURRENT VISUAL ATTENTION

- Результаты:

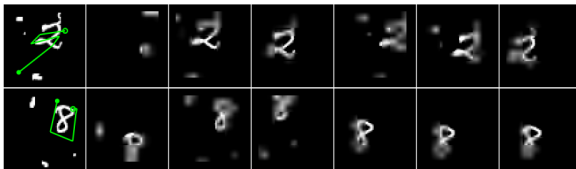
(a) 60x60 Cluttered Translated MNIST

Model	Error
FC, 2 layers (64 hiddens each)	28.58%
FC, 2 layers (256 hiddens each)	11.96%
Convolutional, 2 layers	8.09%
RAM, 4 glimpses, 12×12 , 3 scales	4.96%
RAM, 6 glimpses, 12×12 , 3 scales	4.08%
RAM, 8 glimpses, 12×12 , 3 scales	4.04%
RAM, 8 random glimpses	14.4%

(b) 100x100 Cluttered Translated MNIST

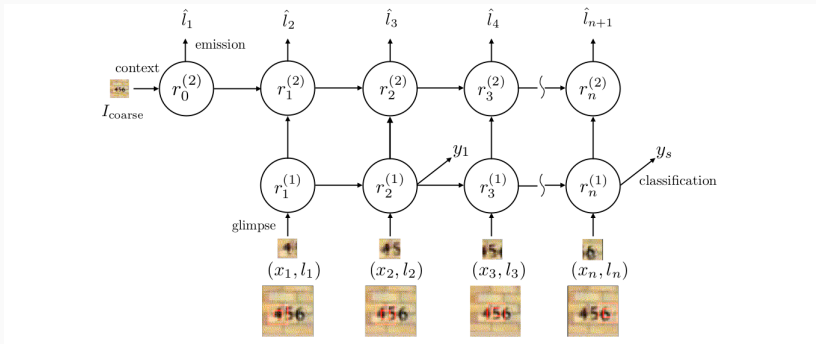
Model	Error
Convolutional, 2 layers	14.35%
RAM, 4 glimpses, 12×12 , 4 scales	9.41%
RAM, 6 glimpses, 12×12 , 4 scales	8.31%
RAM, 8 glimpses, 12×12 , 4 scales	8.11%
RAM, 8 random glimpses	28.4%

- А вот как внимание гуляет по картинке:



RECURRENT VISUAL ATTENTION

- В следующей работе (Ba et al., 2015) сделали глубокую модель:



- Кстати, обучали по-другому, вариационными методами. Как это?..

- Нам нужно классифицировать, т.е. $p(y \mid \mathbf{x}, \theta)$ максимизировать.
- Маргинализуем по положениям glimpses:

$$\log p(y \mid \mathbf{x}, \theta) = \log \sum_l p(l \mid \mathbf{x}, \theta) p(y \mid l, \mathbf{x}, \theta).$$

- Запишем вариационную нижнюю оценку свободной энергии (как её получить?):

$$\begin{aligned} \log \sum_l p(l \mid \mathbf{x}, \theta) p(y \mid l, \mathbf{x}, \theta) &\geq \sum_l p(l \mid \mathbf{x}, \theta) \log p(y, l \mid \mathbf{x}, \theta) + H[l] \\ &= \sum_l p(l \mid \mathbf{x}, \theta) \log p(y \mid l, \mathbf{x}, \theta). \end{aligned}$$

- И теперь можно брать производные:

$$\begin{aligned}\frac{\partial J}{\partial \theta} &= \sum_l p(l \mid \mathbf{x}, \theta) \frac{\partial \log p(y \mid l, \mathbf{x}, \theta)}{\partial \theta} + \sum_l \log p(y \mid l, \mathbf{x}, \theta) \frac{\partial p(l \mid \mathbf{x}, \theta)}{\partial \theta} \\ &= \sum_l p(l \mid \mathbf{x}, \theta) \left[\frac{\partial \log p(y \mid l, \mathbf{x}, \theta)}{\partial \theta} + \log p(y \mid l, \mathbf{x}, \theta) \frac{\partial \log p(l \mid \mathbf{x}, \theta)}{\partial \theta} \right].\end{aligned}$$

- А эту сумму уже будем приближать выборкой:

$$\frac{\partial J}{\partial \theta} \approx \frac{1}{M} \sum_{i=1}^M \left[\frac{\partial \log p(y \mid l^{(i)}, \mathbf{x}, \theta)}{\partial \theta} + \log p(y \mid l^{(i)}, \mathbf{x}, \theta) \frac{\partial \log p(l^{(i)} \mid \mathbf{x}, \theta)}{\partial \theta} \right],$$

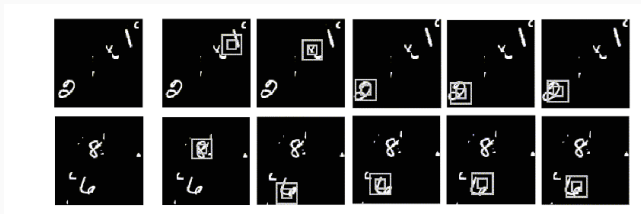
$$\text{где } l^{(i)} \sim p(l_n \mid \mathbf{x}, \theta) = N(l_n \mid \hat{l}_n, \Sigma).$$

- И это уже алгоритм: сэмплируем glimpses, потом используем их в backpropagation.
- Как и в (Mnih et al., 2014), надо бы уменьшить дисперсию; для этого вычитают baseline, пока не будем углубляться.

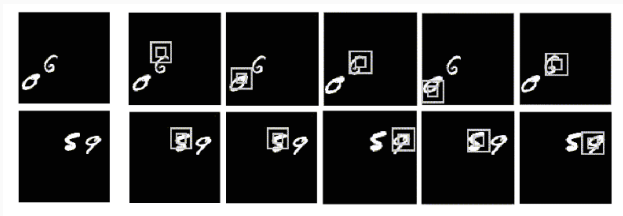
- Получился интересный результат – мы увидели, что примерно один и тот же алгоритм может получиться с двух разных сторон:
 - из обучения с подкреплением через REINFORCE;
 - из вариационной оценки собственно целевой функции.
- Важный гиперпараметр – размер *glimpse*, т.е. как переводить единицы измерений в системе координат *glimpses* в пиксели.
- То же самое легко расширить на последовательную классификацию нескольких объектов – просто фиксированное число *glimpses* на объект, потом классифицируем, плюс терминальное действие в конце всего.

RECURRENT VISUAL ATTENTION

- Вот как это работает на распознавании двух цифр:



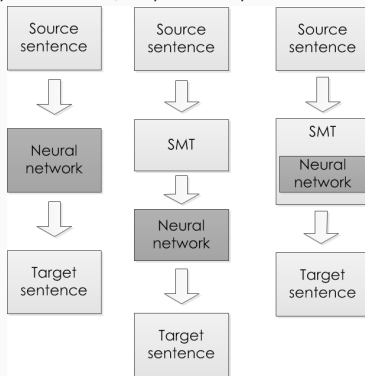
- Любопытно, что на сложении по-другому:



МАШИННЫЙ ПЕРЕВОД:
ENCODER-DECODER
И ВНИМАНИЕ

- Перевод – очень хорошая задача:
 - очевидно очень практическая;
 - очевидно очень высокоуровневая, требует понимания;
 - считается довольно неплохо квантифицируемой (BLEU, TER – хотя см. выше);
 - имеет большие доступные датасеты параллельных переводов.

- Статистический машинный перевод (statistical machine translation, SMT): моделируем условную вероятность $p(y | x)$ перевода y при условии исходного текста x .
- Классический SMT: моделируем $\log p(y | x)$ линейной комбинацией признаков, строим признаки.



- Нам больше интересно моделирование sequence-to-sequence:
 - RNN естественным образом моделирует последовательность $X = (x_1, x_2, \dots, x_T)$ как $p(x_1), p(x_2 | x_1), \dots, p(x_T | x_{<T}) = p(x_T | x_{T-1}, \dots, x_1)$, и теперь $p(X)$ – это просто

$$p(X) = p(x_1)p(x_2 | x_1) \dots p(x_k | x_{<k}) \dots p(x_T | x_{<T});$$

- так RNN и в языковых моделях используются;
 - предсказываем следующее слово на основе скрытого состояния и предыдущего слова;
- Как применить эту идею к переводу?

МЕТРИКИ КАЧЕСТВА ДЛЯ SEQUENCE-TO-SEQUENCE МОДЕЛЕЙ

- Дальше будет самое интересное: машинный перевод, диалоговые модели, ответ на вопросы.
- Но как мы будем оценивать NLP-модели, которые генерируют текст?
- Есть метрики качества, которые сравнивают результат с правильными ответами:
 - BLEU (Bilingual Evaluation Understudy): перевзвешенная precision (в т.ч. для нескольких правильных ответов);
 - METEOR: гармоническое среднее precision и recall по униграммам;
 - TER (Translation Edit Rate): число исправлений между выходом и правильным ответом, делённое на среднее число слов;
 - LEPOR: комбинируем базовые факторы и метрики с настраиваемыми параметрами.
- Есть ещё куча метрик, связанных с представлениями слов и предложений (хотим поближе к правильному ответу).
- Одна только проблема...

МЕТРИКИ КАЧЕСТВА ДЛЯ SEQUENCE-TO-SEQUENCE МОДЕЛЕЙ

- ...всё это вообще не работает.

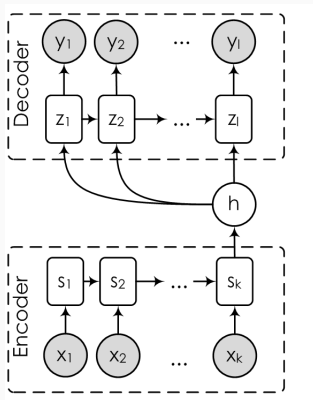
Metric	Twitter				Ubuntu			
	Spearman	p-value	Pearson	p-value	Spearman	p-value	Pearson	p-value
Greedy	0.2119	0.034	0.1994	0.047	0.05276	0.6	0.02049	0.84
Average	0.2259	0.024	0.1971	0.049	-0.1387	0.17	-0.1631	0.10
Extrema	0.2103	0.036	0.1842	0.067	0.09243	0.36	-0.002903	0.98
METEOR	0.1887	0.06	0.1927	0.055	0.06314	0.53	0.1419	0.16
BLEU-1	0.1665	0.098	0.1288	0.2	-0.02552	0.8	0.01929	0.85
BLEU-2	0.3576	< 0.01	0.3874	< 0.01	0.03819	0.71	0.0586	0.56
BLEU-3	0.3423	< 0.01	0.1443	0.15	0.0878	0.38	0.1116	0.27
BLEU-4	0.3417	< 0.01	0.1392	0.17	0.1218	0.23	0.1132	0.26
ROUGE	0.1235	0.22	0.09714	0.34	0.05405	0.5933	0.06401	0.53
Human	0.9476	< 0.01	1.0	0.0	0.9550	< 0.01	1.0	0.0

Table 3: Correlation between each metric and human judgements for each response. Correlations shown in the human row result from randomly dividing human judges into two groups.

- Тут нужно что-то новое. И пока не совсем ясно, что именно.

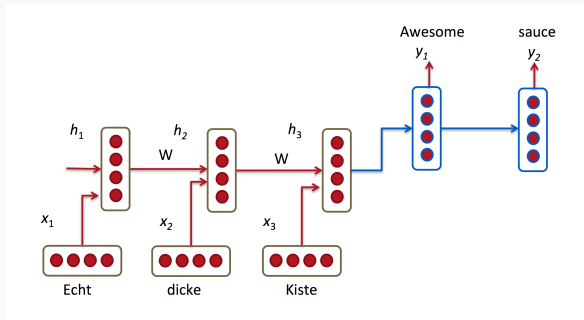
ENCODER-DECODER АРХИТЕКТУРЫ

- Encoder-decoder архитектуры (Sutskever et al., 2014; Cho et al., 2014):



- Сначала кодируем, потом декодируем обратно.

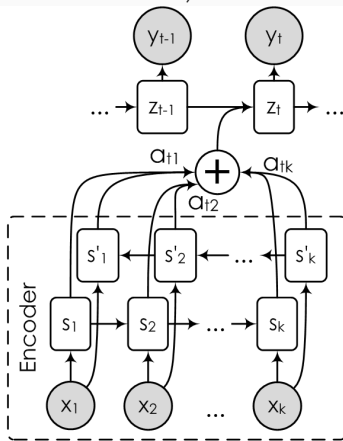
- Так же может работать и с переводом.



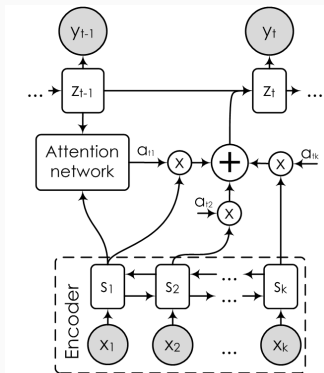
- Проблема: надо сжимать всё предложение в один вектор.
- С длинными участками текста это вообще перестаёт работать.

ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

- Решение: давайте обучим специальные веса, показывающие, насколько та или иная часть входа важна для текущего выхода.
- Прямое применение – двунаправленный LSTM плюс внимание (Bahdanau et al. 2014):

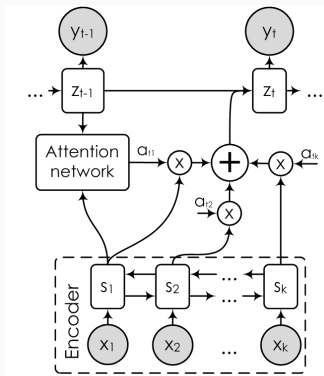


- *Мягкое внимание* (soft attention) (Luong et al. 2015a; 2015b; Jean et al. 2015):
 - encoder – двунаправленная RNN, есть оба контекста;
 - сеть внимания выдаёт оценку релевантности – надо ли переводить это слово прямо сейчас?

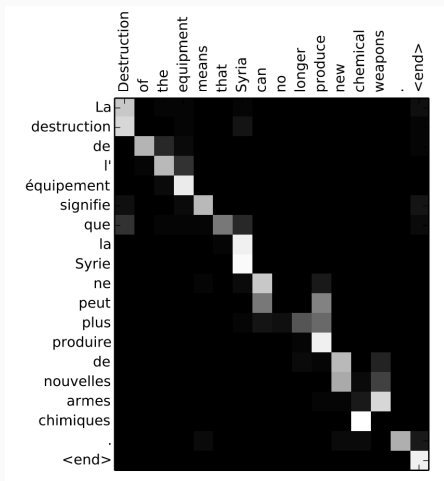


- Формально очень просто: считаем веса внимания α_{tj} и перевзвешиваем векторы контекстов:

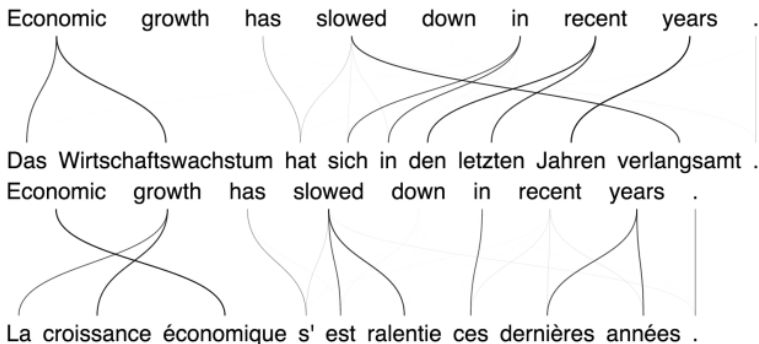
$$e_{tj} = a(z_{t-1}, j), \quad \alpha_{tj} = \text{softmax}(e_{tj}; e_{t*}),$$
$$c_t = \sum_j \alpha_{tj} h_j, \text{ и теперь } z_t = f(s_{t-1}, y_{t-1}, c_t).$$



- В результате можно визуализировать, на что смотрит сеть:

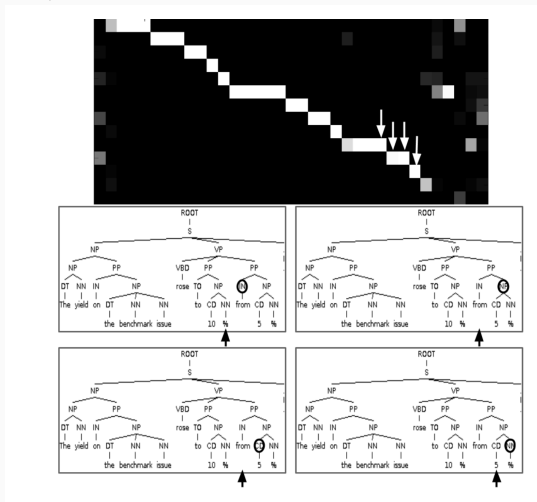


- Получается гораздо лучше порядок слов:

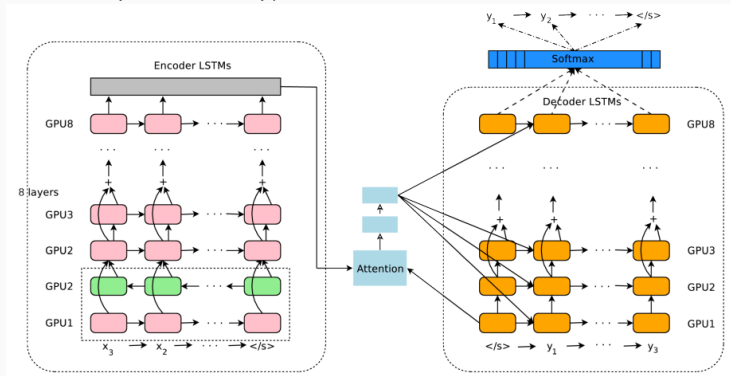


ВНИМАНИЕ В НЕЙРОННЫХ СЕТЯХ

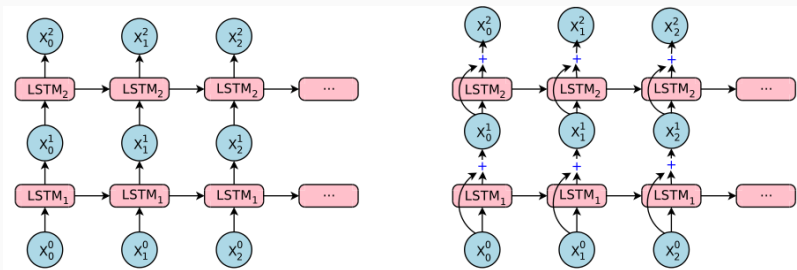
- Другая необычная работа – «Grammar as a Foreign Language» (Vinyals et al., 2015)



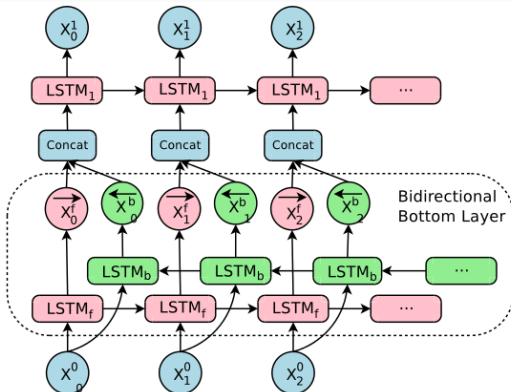
- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - как на самом деле работает Google Translate;
 - базовая архитектура та же самая: encoder, decoder, attention;
 - RNN глубокие, по 8 уровней в encoder и decoder:



- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - просто stacked LSTM перестают работать далее 4-5 уровней;
 - поэтому добавляют остаточные связи, как в ResNet:



- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
 - нижний уровень, естественно, двунаправленный:



- Сентябрь 2016: Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*:
- в GNMT ещё две идеи о сегментации слов:
 - *wordpiece model*: разбить слова на кусочки (отдельной моделью); пример из статьи:

Jet makers feud over seat width with big orders at stake

превращается в

_J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

- *mixed word/character model*: конвертировать слова, не попадающие в словарь, в последовательность букв-токенов; пример из статьи:

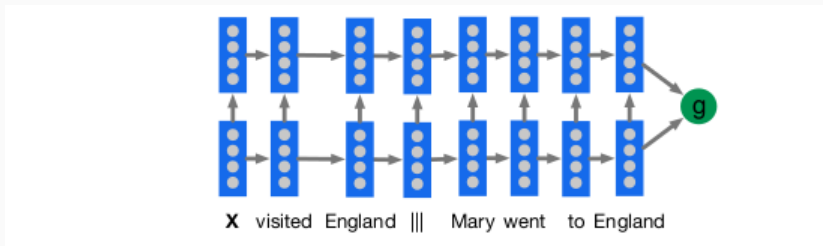
Miki превращается в M <M>i <M>k <E>i

TEACHING MACHINES TO READ

- (Hermann et al., 2015): «Teaching machines to read and comprehend» (Google DeepMind)
- Предлагают новый способ построить датасет для понимания, автоматически создавая тройки (context, query, answer) из текстов новостей и т.п.

Original Version	Anonymised Version
Context The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the “Top Gear” host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon “to an unprovoked physical and verbal attack.” ...	the <i>ent381</i> producer allegedly struck by <i>ent212</i> will not press charges against the “ <i>ent153</i> ” host, his lawyer said friday. <i>ent212</i> , who hosted one of the most - watched television shows in the world, was dropped by the <i>ent381</i> wednesday after an internal investigation by the <i>ent180</i> broadcaster found he had subjected producer <i>ent193</i> “to an unprovoked physical and verbal attack.” ...
Query Producer X will not press charges against Jeremy Clarkson, his lawyer says.	producer X will not press charges against <i>ent212</i> , his lawyer says.
Answer Oisin Tymon	<i>ent193</i>

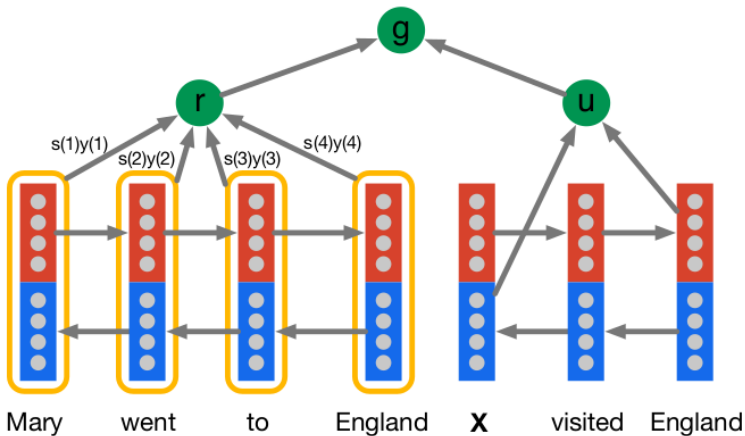
- Базовая модель – глубокий LSTM:



- Но так, конечно, совсем плохо работает.

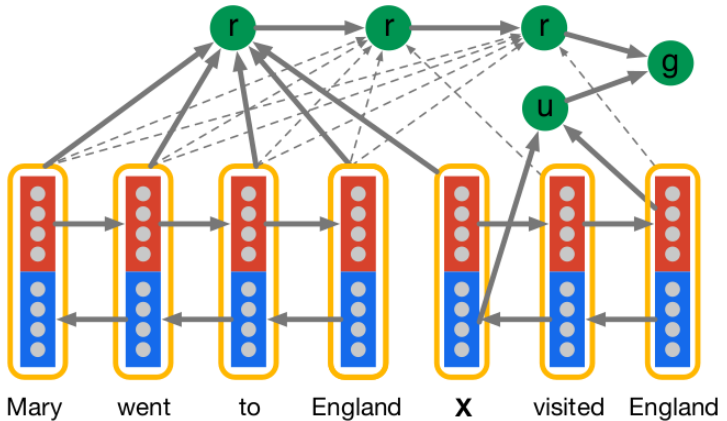
TEACHING MACHINES TO READ

- Attentive Reader: обучаемся, на какую часть документа смотреть



TEACHING MACHINES TO READ

- Impatient Reader: можем перечитывать нужные части документа по мере прочтения запроса



TEACHING MACHINES TO READ

- Получаются разумные карты внимания:

by *ent423* ,*ent261* correspondent updated 9:49 pm et ,thu
march 19,2015 (*ent261*) a *ent114* was killed in a parachute
accident in *ent45* ,*ent85* ,near *ent312* ,a *ent119* official told
ent261 on wednesday .he was identified thursday as
special warfare operator 3rd class *ent23* ,29 ,of *ent187* ,
ent265 .`` *ent23* distinguished himself consistently
throughout his career .he was the epitome of the quiet
professional in all facets of his life ,and he leaves an
inspiring legacy of natural tenacity and focused

...

ent119 identifies deceased sailor as **X** ,who leaves behind
a wife

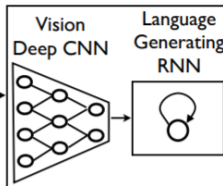
by *ent270* ,*ent223* updated 9:35 am et ,mon march 2 ,2015
(*ent223*) *ent63* went familial for fall at its fashion show in
ent231 on sunday ,dedicating its collection to `` mamma "
with nary a pair of `` mom jeans " in sight .*ent164* and *ent21* ,
who are behind the *ent196* brand ,sent models down the
runway in decidedly feminine dresses and skirts adorned
with roses ,lace and even embroidered doodles by the
designers ' own nieces and nephews .many of the looks
featured saccharine needlework phrases like `` i love you ,

...

X dedicated their fall fashion show to moms

SHOW, ATTEND, AND TELL

- Теперь давайте про подписи к картинкам.
- Сначала было «Show and Tell» (Vinyals et al., 2015):

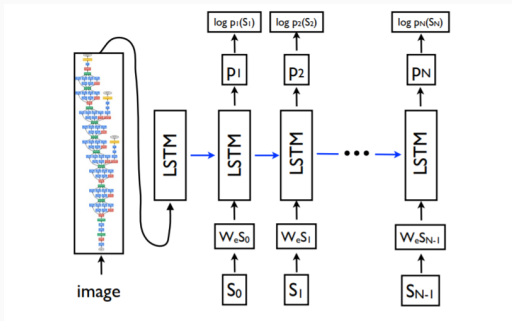


A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

SHOW, ATTEND, AND TELL

- Довольно прямолинейная архитектура:
 - целевая функция – это просто $\sum_{(I,S)} \log p(S | I; \theta)$, где I – картинка, S – описание;
 - раскладываем и моделируем $p(S_t | I, S_0, \dots, S_{t-1})$ рекуррентной сетью с LSTM;
 - а CNN используем, чтобы извлечь признаки.



SHOW, ATTEND, AND TELL

- Получалось хорошо, но можно лучше:

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

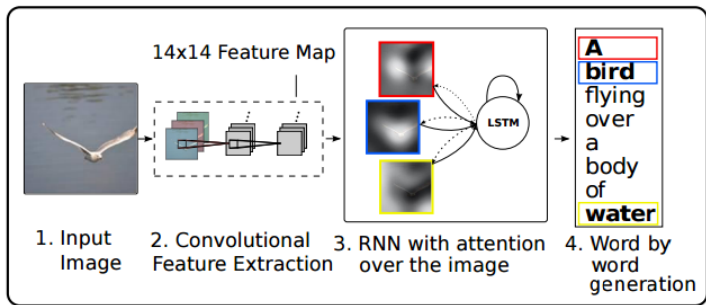
Describes with minor errors

Somewhat related to the image

Unrelated to the image

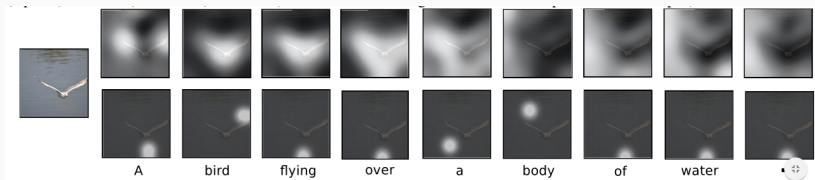
SHOW, ATTEND, AND TELL

- Из этого появилось «Show, Attend, and Tell» (Xu et al., 2015)



SHOW, ATTEND, AND TELL

- Soft attention vs. hard attention (стохастически выбираем однозначный кусок картинки).



- Soft attention – строим аннотацию с весами

$$\phi(\{\mathbf{a}\}_i, \{\alpha_i\}) = \sum_{i=1}^L \alpha_{t,i} \mathbf{a}_i.$$

- Hard attention обучается максимизацией вариационной нижней оценки

$$L_s = \sum_s p(s | \mathbf{a}) \log p(\mathbf{y} | s, \mathbf{a}) \leq \log \sum_s p(s | \mathbf{a}) p(\mathbf{y} | s, \mathbf{a}) = \log p(\mathbf{y} | \mathbf{a}).$$

- От L_s можно брать производные:

$$\frac{\partial L_s}{\partial W} = \sum_s p(s | \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} | s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} | s, \mathbf{a}) \frac{\partial \log p(s | \mathbf{a})}{\partial W} \right].$$

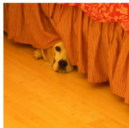
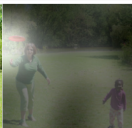
- И дальше сэмплируем s_t с вероятностями α_i и приближаем ожидание выборкой.
- Опять те же трюки, вычитаем baseline, всё такое.

SHOW, ATTEND, AND TELL

- Часто получаются очень хорошие результаты:



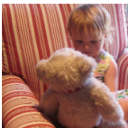
A woman is throwing a frisbee in a park.



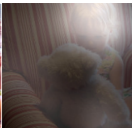
A dog is standing on a hardwood floor.



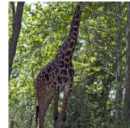
A stop sign is on a road with a mountain in the background.



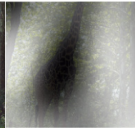
A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.



- А когда плохие, можно посмотреть почему.

SHOW, ATTEND, AND TELL

- Примеры – hard attention:



(a) A man and a woman playing frisbee in a field.

SHOW, ATTEND, AND TELL

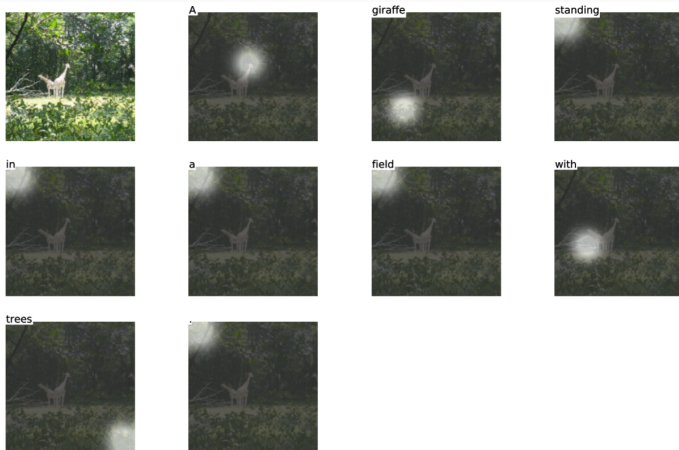
- Примеры – soft attention:



(b) A woman is throwing a frisbee in a park.

SHOW, ATTEND, AND TELL

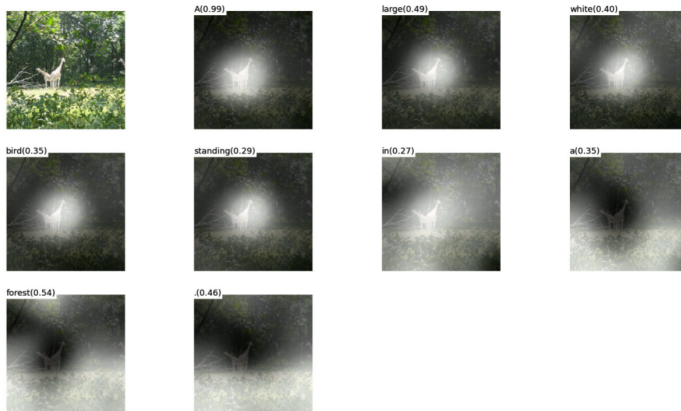
- Примеры – hard attention:



(a) A giraffe standing in the field with trees.

SHOW, ATTEND, AND TELL

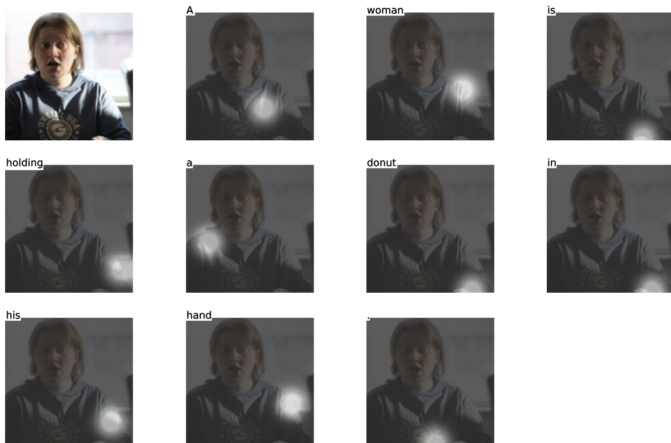
- Примеры – soft attention:



(b) A large white bird standing in a forest.

SHOW, ATTEND, AND TELL

- Примеры – hard attention:



(a) A woman is holding a donut in his hand.

SHOW, ATTEND, AND TELL

- Примеры – soft attention:



(b) A woman holding a clock in her hand.

СПАСИБО!

Спасибо за внимание!

