

# Вариационные Автоенкодеры с мультиномиальным Prior'ом.

Денис Мазур

Ноябрь 2018

## Аннотация

Метод вариационных автоэнкодеров является, вероятно, самым эффективным среди методов глубинного обучения без учителя...

## 1 Вступление

Так как прочтение данной работы не предполагает ознакомленность с методами машинного обучения, будет дан поверхностный разбор некоторых ключевых принципов и методов, необходимых для понимания вариационных автоэнкодеров.

От читателя предполагается понимание математического анализа, линейной алгебры и теории вероятности.

## 2 Обучение нейронных сетей

Для базового понимания автоэнкодеров не требуется понимание принципа работы нейронных сетей, по этой причине объяснение устройства нейронных сетей было решено опустить и уделить больше внимания принципу их обучения. Для интересующихся, литература на тему нейронных сетей будет приложена в библиографии.

Для простоты устройсто нейронной сети можно редуцировать до некоторой функции, имеющей параметры  $W$ , принимающей значения  $X$  и выдающей значения  $\hat{X}$ . Физический смысл  $X$  - данные, подаваемые на вход сети,  $\hat{X}$  - выход нейронной сети. Каждому элементу  $x \in X$  соответствует элемент  $y \in Y$ , и задачей обучения нейронной сети является подобрать параметры  $W$  нейронной сети так, чтобы они минимально или не отличались от  $Y$ . Также можно сформулировать задачу следующим образом:

$$d(f(x_n), y_n) \rightarrow \min$$

Также, можно поставить задачу как поиск оптимальных параметров  $W$ :

$$\operatorname{argmin}_W d(f(x_n), y_n)$$

Где  $f(x)$  - нейронная сеть, а  $d(x, y)$  определенная нами функция расстояния (отличия) между выходом нейронной сети и настоящим ответом, называемая функцией потерь. Важно понимать следующие свойства функций  $f(x)$  и  $d(x, y)$ , они обе дифференцируемые и определены на всех  $X$ , а это значит, что мы к параметрам нейронной сети  $W$  можем применять любые методы оптимизации функций, например, градиентный метод.

## 2.1 Градиентный метод

Градиентный метод является методом решения задач вида  $\min_{a \in \mathbb{R}^n} g(x)$ . Алгоритм градиентного метода начинается с выбора параметров  $a$ , это может делать как и из какого либо представления того какими примерно эти параметры должны быть, так и случайным образом. Далее идет так называемый "шаг" градиентного метода. Из параметров  $a$  вычитается значение градиента функции  $g(a)$  в точке текущего значения  $a$ :

$$a_{n+1} = a_n - \nabla g(a_n)$$

Чтобы не "перескочить" через минимум функции во время итерации метода, градиент  $\nabla f(x)$  обычно умножается на какую-то константу  $\alpha$ :

$$a_{n+1} = a_n - \alpha \nabla g(a_n)$$

## 2.2 Градиентный метод в обучении нейронных сетей

Теперь о том как градиентный метод может применяться в обучении нейронных сетей.

Функцию потерь нейронной сети  $d(f(x_n), y_n)$  можно записать в виде,  $d(f(x_n, W), y_n)$ , до этого, для простоты, записи параметры нейронной сети  $W$  не записывались как аргумент, хотя, очевидно, им являются. Поясню, что  $X$  и  $Y$  являются константными значениями, так как являются выборкой наших данных и известны нам заранее. Таким образом, единственный аргумент, который мы можем изменять, минимизируя функцию потерь -  $W$ .

На практике, оптимизировать функцию потерь градиентным спуском "в лоб" не получится. У этого есть несколько причин, одна из главных - риск переобучения модели (ситуация при которой модель показывает идеальных результат на обучающей выборке, но плохой на валидации). Но помимо этого, выборка данных чаще всего слишком велика и не помещается в память компьютера, по этой причине на каждой итерации градиентного метода выбирается случайная подвыборка тренировочных данных и над ней совершается шаг градиентного метода. Данный метод называется стохастическим градиентным спуском.