

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра дифференциальных уравнений и системного анализа

БАЙЕСОВСКИЕ НЕЙРОННЫЕ СЕТИ

Курсовая работа

Афанасенко Григория Сергеевича  
студента 2-го курса  
специальности 1-31 03 09  
«Компьютерная математика  
и системный анализ»

Научный руководитель:  
ст. преподаватель А. Э. Малевич

Минск, 2024

# ОГЛАВЛЕНИЕ

<b>1</b>	<b>Виды нейронных сетей.</b>	<b>3</b>
1.1	Детерминированные нейронные сети. . . . .	3
1.2	Байесовские нейронные сети. . . . .	5
1.2.1	Вероятностные графы вычислений. . . . .	5
1.2.2	Байесовские нейронные сети. . . . .	6

# ГЛАВА 1

## Виды нейронных сетей.

### 1.1 Детерминированные нейронные сети.

Сначала напомним, что такое обычные(детерминированные) нейронные сети и как они обучаются.

Основная задача обычных искусственных нейронных сетей(*ANN*) в том, чтобы аппроксимировать некоторую зависимость выхода  $y$  от входа  $x$ :  $y = \Phi(x)$ . Зависимость  $\Phi(x)$  аппроксимируем через композицию последовательных преобразований.

Для простоты будем рассматривать обычные *полносвязные* сети со входом  $x$ , скрытыми(промежуточными) состояниями слоёв  $\mathbf{h}_i$ , функциями активации  $a_i(\cdot)$  и выходом  $y$ :

$$\begin{aligned}\mathbf{h}_0 &= \mathbf{x} \\ \mathbf{h}_i &= a_i(\mathbf{W}_i \cdot \mathbf{h}_{i-1} + \mathbf{b}_i), i = \overline{1...n} \\ \mathbf{h}_n &= \hat{y} \\ L &= \mathcal{L}(\hat{y}, y),\end{aligned}$$

где  $\mathcal{L}(\cdot, \cdot)$  - функция ошибки.

Обозначим параметры модели на  $i$ -ом слое  $\boldsymbol{\theta}_i = (\mathbf{W}_i, \mathbf{b}_i)$ , а параметры всей модели через  $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i : i = \overline{1...n}\}$ . Чаще всего нейронные сети принято рассматривать, как вычислительный граф/граф вычислений. Такой подход удобен с инженерной точки зрения, поскольку позволяет воспользоваться инструментом автоматического дифференцирования, и используется во всех современных фреймворках: PyTorch, TensorFlow и прочие. Граф вычислений является ациклическим ориентированным графом, составленным из вершин-переменных и вершин-операций(Рисунок 1.1).

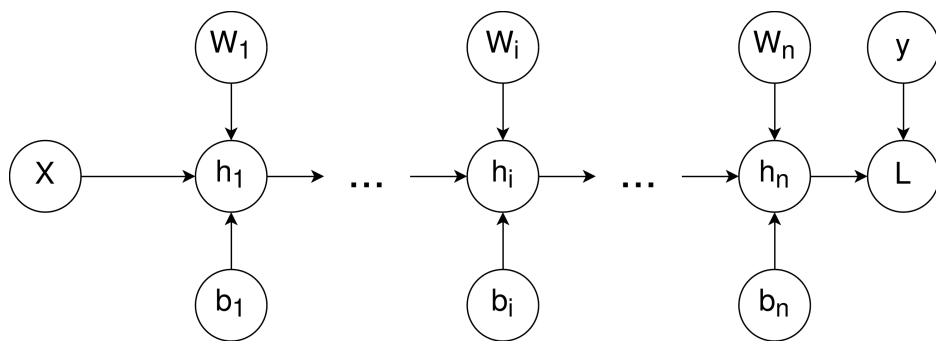


Рисунок 1.1 Полносвязная сеть в виде графа вычислений

Далее будем называть модели, основанные на графах вычислений, — графовыми моделями. Графы вычислений могут разных типов: статическими/динамическими, детерминированными/вероятностными и т.д. Для обучения/настройки параметров детерминированных графовых моделей используется метод *обратного распространения ошибки (back propagation)*, который широко используется в современном мире. Вкратце напомним алгоритм:

После прямого выполнения графа (*forward pass*), то есть в соответствии с направлениями рёбер на выходе мы получаем  $L$ -значение функции ошибки, которые в зависимости от задач мы хотим либо минимизировать, либо максимизировать. Для этого мы пользуемся градиентными методами оптимизации, что требует вычисление градиентов  $\frac{dL}{dW_i}, \frac{dL}{db_i}$  по нашим параметрам модели, где  $i = \overline{1, n}$ . В общем случае это трудная задача, однако в случае детерминированных графовых моделей мы можем использовать цепное правило (*chain rule*) для того, чтобы последовательно проталкивать градиенты, начиная с концевой вершины, содержащей  $L$ .

Например, для подсчёта градиентов  $\frac{dL}{dW_i}, \frac{dL}{db_i}$  мы представим его в виде

$$\frac{dL}{dW_i} = \frac{dL}{dh_n} \cdot \frac{dh_n}{dW_i}$$

$$\frac{dL}{db_i} = \frac{dL}{dh_n} \cdot \frac{dh_n}{db_i}$$

Аналогично для всех остальных параметров модели мы будем проталкивать накопленный с концевой вершины градиент до соответствующих вершин и с помощью этого градиента высчитывать градиент по параметрам модели. Схему работы алгоритма обратного распространения ошибки можно увидеть на Рису-

НОК 1.2.

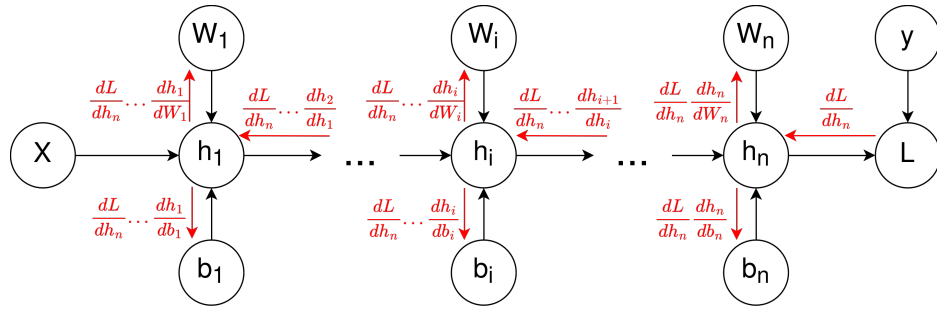


Рисунок 1.2 Обратное распространение ошибки по графу вычислений детерминированной полносвязной сети

Однако детерминированные нейронные сети обладают несколькими проблемами:

- Переобучение.
- Низкая интерпретируемость.
- Завышенная/заниженная уверенность модели в предсказаниях, даже если они неверные.
- Низкий уровень откалиброванности модели.

Указанные проблемы попытаемся решить с помощью байесовского подхода к нейронным сетям, который рассмотрим далее.

## 1.2 Байесовские нейронные сети.

### 1.2.1 Вероятностные графы вычислений.

Перед тем, как приступить к байесовским нейронным сетям, рассмотрим *вероятностные графы вычислений*, на которых основаны байесовские сети. В литературе также часто вместо названия *вероятностные графы вычислений* встречается *вероятностные графические модели*. Второе название является более общим, в то время как первое более специфично именно для байесовских нейронных сетей. Такие графы вычислений широко используются и известны достаточно давно. Они лежат в основе, например, Марковских цепей, которые

ранее активно использовались в различных задачах машинного предсказания, распознавания образов и т.п.

Основная мотивация в использовании вероятностного подхода состоит в том, что в реальном мире мы чаще имеем дело с неопределённостью в данных и знаниях и не можем детерминированно описать все приходящие переменные для решения задачи. Для решения проблем с неопределённостью можно попробовать собрать большие объёмы данных для того, чтобы попытаться "понять" эту неопределённость. С другой стороны мы можем использовать байесовский подход, который напрямую оперирует с неопределённостью.

Рассмотрим структуру *вероятностных графовых моделей*. В отличие от детерминированных моделей в граф добавляются вершины со случайными переменными. Таким образом в нашем совместно существуют детерминированные вершины и случайные (Рисунок 1.3). Стоит отметить, что после вступления в контакт детерминированных переменных и случайных весь дальнейший результат будет случайным.

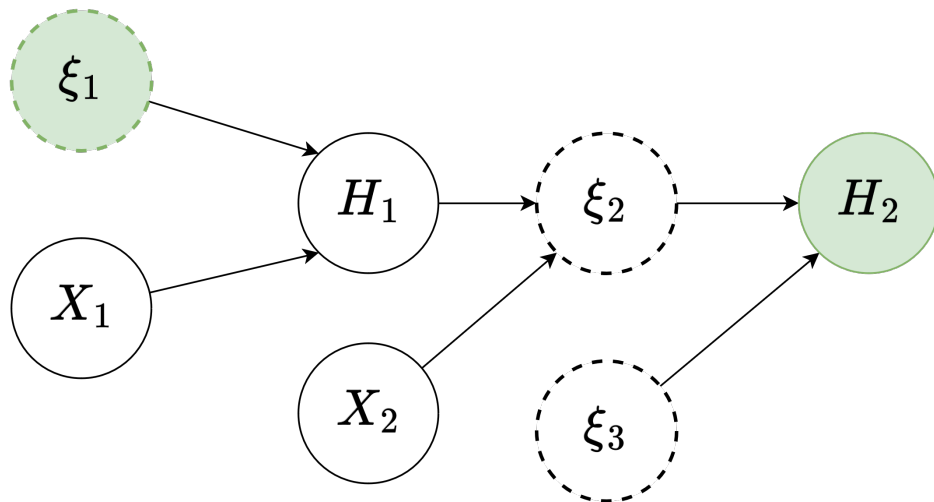


Рисунок 1.3 Вероятностная графическая модель. Здесь круги с пунктирной границей являются сэмплируемыми случайными величинами. Зелёным цветом обозначены наблюдаемые случайные переменные.

### 1.2.2 Байесовские нейронные сети.