Вопрос 1

Верно

Баллов: 1,00 из 1,00

Фиксирование seed не гарантирует, что код будет одинаково выполняться на разных компьютерах. Но на одной и той же машине вы будете получать одинаковые результаты, перезапуская один и тот же скрипт.

Например, функция random.randint(start, end) отдает случайное целое число в диапазоне от start, end (включительно). Запуская скрипт, состоящий из вызова этой функции, вы будете получать разные ответы.

В этом задании вам нужно подобрать seed, чтобы функция random.randint(0, 10) выдала число 5

PS: надо понимать, что фиксирование random seed не приведет к тому, что повторный вызов random.randint(0, 10) **внутри** того же скрипта снова даст 5.

Для примера:

Ввод	Результат
`anything`	`5`

Ответ: (штрафной режим: 0 %)

Сброс ответа

```
1 import random
2 random.seed(9000)
3 print(random.randint(0, 10))
```

Прошли все тесты! ✔

Верно

Баллы за эту попытку: 1,00/1,00.

Вопрос Инфо

Вопрос 2

Выполнен

Баллов: 1,00 из 1,00

В нашем примере мы хотим предсказать, к какому из трех классов относится конкретная бутылка.

Отметьте верные утверждения:

Выберите один или несколько ответов:

- [ ] a. Для решения данной задачи нужно не менее 5 нейронов в первом скрытом слое
- [x] b. Функция SoftMax может выдать одинаковые вероятности для разных классов
- [ ] c. Мы должны зафиксировать инициализацию весов сети для того, чтобы решить задачу классификации
- [x] d. Сумма выходов функции SoftMax равна 1 (с точностью до машинного нуля)
- [ ] e. Сумма выходов функции SoftMax равна количеству предсказываемых классов

Ваш ответ верный.

Вопрос 3

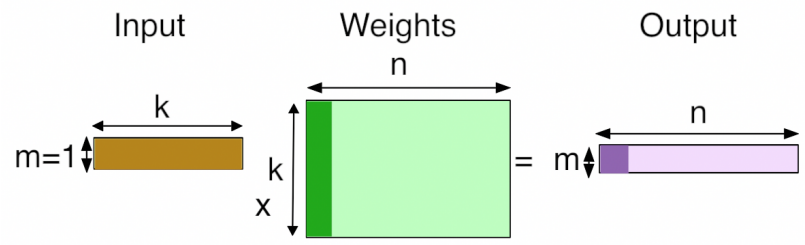
Неверно

Баллов: 0,00 из 1,00

Наша нейросеть – это сложная функция от входов, параметризованная весами и смещениями (bias). Давайте разберемся, как выглядит ее аналитический вид:

Возьмем первый полносвязный слой (torch.nn.Linear). Он принимает на вход вектор одного объекта $\vec{x}$ из k чисел. В нашем случае у объекта 2 признака, значит k=2. Сам слой содержит n_hidden_neurons. У каждого нейрона этого слоя k весов (по одному весу на вход) и по одному значению bias. Каждый нейрон отдает 1 число: таким образом, выход слоя – это вектор из n чисел.

Функцию полносвязного слоя можно представить в виде матричного перемножения $\vec{x}$ и матрицы весов всех нейронов слоя, где один нейрон отвечает одному столбцу в матрице:



При этом нужно не забыть про bias. Один из способов его учесть: расширить $\vec{x}$, добавив к нему 1 в качестве последнего значения. В таком случае, bias будет просто одним из весов, который всегда умножается на 1. На картинке выше значения bias'ов будут находится в последней строчке матрицы весов.

Получается, формула для вычисления выходов из первого слоя выглядит следующим образом:

$$\begin{bmatrix} z_1 & z_2 & z_3 & \dots & z_n \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_k & 1 \end{bmatrix} \begin{bmatrix} w_1^1 & w_1^2 & w_1^3 & \dots & w_1^n \\ w_2^1 & w_2^2 & w_2^3 & \dots & w_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_k^1 & w_k^2 & w_k^3 & \dots & w_k^n \\ b^1 & b^2 & b^3 & \dots & b^n \end{bmatrix}$$

Кроме того, мы можем посчитать выходы сразу для нескольких объектов, если вместо одного элемента $\vec{x}$ пододим сразу матрицу из m элементов, где каждая строчка будет отвечать за конкретный вектор входа:

$$\begin{bmatrix} z_1^1 & z_1^2 & z_1^3 & \dots & z_1^n \\ z_2^1 & z_2^2 & z_2^3 & \dots & z_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_m^1 & z_m^2 & z_m^3 & \dots & z_m^n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & \dots & x_1^k & 1 \\ x_2^1 & x_2^2 & x_2^3 & \dots & x_2^k & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_m^1 & x_m^2 & x_m^3 & \dots & x_m^k & 1 \end{bmatrix} \begin{bmatrix} w_1^1 & w_1^2 & w_1^3 & \dots & w_1^n \\ w_2^1 & w_2^2 & w_2^3 & \dots & w_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_k^1 & w_k^2 & w_k^3 & \dots & w_k^n \\ b^1 & b^2 & b^3 & \dots & b^n \end{bmatrix}$$

Выходы слоя мы передадим в функцию $\sigma (z)={\frac {1}{1+e^{-z}}}$, которая будет применена для каждого элемента результирующей матрицы z_i^j. Далее эти выходы можно передать в следующий слой, аналогично тому, как мы сделали с матрицей X. К матрице выходов из последнего слоя можно применить функцию Softmax. Softmax применится построчно (в видео мы для этого выставили параметр dim=1) и преобразует выходы из нейронов в "вероятности".

Запишите в одну строчку формулу функции inference для сети WineNet.

Первый слой с активацией можно записать как Sigmoid(X*W1+b1). Заметьте, что здесь вектор b1 мы применяем иначе, но результат от этого не меняется.

В формуле не должно быть пробелов. Допустимые элементы формулы:

X W1 W2 W3 b1 b2 b3 Sigmoid Softmax + *) (

Ответ: (штрафной режим: 0 %)

```
Sigmoid(X*W1+b1)
```

	Comment
✖	Попробуйте ещё раз.

Ваш код должен пройти все тесты, чтобы заработать какие-либо оценки. Пробуйте снова.

Неверно

Баллы за эту попытку: 0,00/1,00.

Вопрос Инфо

Вопрос 4

Выполнен

Баллов: 1,00 из 1,00

В нашем примере мы делим датасет на батчи (batch).

Отметьте правильные утверждения:

Выберите один или несколько ответов:

- [x] a. Полезно перед каждой эпохой перемешивать порядок объектов
- [x] b. В нашем случае гарантируется, что при обучении нейронная сеть увидит весь обучающий датасет за одну эпоху
- [x] c. Одна из причин использовать батчи – экономия оперативной и видео- памяти
- [ ] d. Батч содержит данные как из обучающей, так и из тестовой выборки
- [ ] e. За одну эпоху обучение проходит на одном батче

Ваш ответ верный.