1. Какой корректный вариант для инициализации модели нейронной сети?

a) model = sequential

b) model = sequential()

c) model = Sequential

d) model = Sequential()

2. Какой тип модели глубокого обучения вы должны использовать, чтобы различать разных животных в наборе изображений?

a) Классификация

b) Регрессия

c) Любой из них

d) Нет правильного ответа

3. В нейронной сети, используемой для распознавания изображений, каково правильное описание функции узлов/слоев?

a) Узлы в первом слое ищут простые шаблоны, в то время как узлы в последних слоях ищут сложные шаблоны.

b) Узлы в первом слое ищут сложные шаблоны, в то время как узлы в последних слоях ищут простые шаблоны.

с) Нет правильного ответа

4. weights, input\_data, preds и error уже были загружены/вычислены. Как и в обратном распространении, рассчитайте наклон весов с помощью функции потерь, mean\_squared\_error.

slope = ????

print(slope)

[-84 -56 -70]

a) 2 \* error

b) 2 \* input\_data

c) input\_data \* error

d) 2 \* input\_data \* error

5. Создайте выходной слой для классификационной модели с 3 выходными классами.

model = Sequential()

model.add(Dense(100, activation='relu', input\_shape=(n\_cols,)))

model.add(Dense(????, activation=????))

a) classes = 3

b) softmax

c) 'softmax'

d) 3 classes

e) 'relu'

f) 3

6. Какое понятие, используемое в нейронных сетях, можно теоретически описать в следующем предложении:

Стоя на темном поле, желая достичь самой низкой точки, не падая, вы чувствуете склон и делаете небольшие шаги вниз по склону, пока не достигнете точки, которая находится вверх со всех сторон.

a) градиентный спучк

b) функция активации

c) прямое распространение

7. Создайте первый слой модели со 100 узлами и ReLU в качестве функции активации.

model = Sequential()

model.add(Dense(100, ????='relu', ????=(n\_cols,)))

a) columns

b) data

c) activation\_function

d) function

e) input\_shape

f) activation

8. Каково правильное описание использования обратного распространения?

a) Обратное распространение минимизирует количество узлов в модели через обучение.

b) Обратное распространение принимает сумму весов, умноженных на входные данные.

c) Обратное распространение обновляет веса, чтобы максимизировать разницу между прогнозом и целью.

d) Обратное распространение обновляет веса, чтобы разница между прогнозом и целью была сведена к минимуму.

9. Заполните код, чтобы составить классификационную модель и включить статистику точности.

Примечание: Функция add\_layers добавляет первые 2 слоя в модель.

model = Sequential()

add\_layers(model)

model.compile(????='adam',

????='categorical\_crossentropy',

????=['accuracy'])

a) metric

b) loss

c) loss\_fct

d) name

e) metrics

f) optimizer

10. Какой пакет содержит функцию Dense().

a) keras.models

b) keras.layers

c) keras.meural

11. Какой слой может создать выходной слой регрессионной модели?

a) model.add(Dense(10, activation='relu'))

b) model.add(Dense(1, activation='relu'))

c) model.add(Dense(10))

d) model.add(Dense(1))

12. Сделайте новый прогноз из обновленных весов (weights\_up). Переменные weights скорость обучения lr, error и gradient доступны локально.

weights\_up = ???? - ???? \* ????

preds\_up = (weights\_up\*input\_data).sum()

a) weights

b) error

c) gradient

d) lr

1 –

2 –

3 –

4 –

5 –

6 –

7 –

8 –

9 –

10 –

11 –

12 –