

Квиз №3

* Indicates required question

1. Email *

2. Ваше имя и фамилия *

3. Какие из этих слоев можно увидеть и в полносвязной, и в сверточной, и в рекуррентной нейросети? * 1 point

Check all that apply.

- ☐ nn.Linear
- ☐ nn.Dropout
- ☐ nn.Conv1d
- ☐ nn.MaxPool1d
- ☐ nn.RNN

4. Чем однослойная нейросеть отличается от многослойной? *

1 point

Mark only one oval.

- ☐ Существенных различий нет
- ☐ Однослойная нейросеть не будет содержать нелинейность
- ☐ Однослойная нейросеть не будет работать
- ☐ Однослойная нейросеть может содержать больше параметров

5. Что представляет из себя **скрытое состояние** рекуррентной нейросети? * 1 point

Mark only one oval.

- ☐ Вектор, содержащий информацию, передающуюся от одного шага обучения к другому
- ☐ Простое число, являющееся "памятью" нейросети
- ☐ Набор нейронов, обрабатывающий информацию, приходящую из входного слоя, и передающий ее на выходной слой

6. Как относятся друг к другу обратное распространение ошибки (backpropagation) и градиентный спуск? * 1 point

Mark only one oval.

- ☐ Градиентный спуск происходит во время обратного распространения ошибки
- ☐ Обратное распространение ошибки является частью градиентного спуска
- ☐ Эти понятия не связаны
- ☐ Эти понятия означают одно и то же

7. Какие утверждения об **обучении базовых моделей** (пре-тренине) верны? * 1 point

Check all that apply.

- ☐ Это вычислительно недорогой этап, позволяющий модели запоминать простейшие инструкции
- ☐ Этот этап включает в себя обучение модели на огромных объемах данных
- ☐ На этом этапе модель обучается предсказывать следующий токен в последовательности
- ☐ Этот этап рекомендуется повторять при дообучении модели

8. Что за нейросеть представлена на картинке? *

1 point

```
1 class Model(nn.Module):
2     def __init__(self, vocab_size, embedding_size):
3         super().__init__()
4         self.hidden_layer = nn.Embedding(vocab_size, embedding_size)
5         self.output_layer = nn.Linear(embedding_size, vocab_size, bias=False)
6
7     def forward(self, input):
8         hidden = self.hidden_layer(input)
9         output_probabilities = self.output_layer(hidden)
10        return output_probabilities
```

Mark only one oval.

- ☐ Это сверточная нейросеть для классификации изображений
- ☐ Это рекуррентная нейросеть для перевода
- ☐ Это полносвязная нейросеть для обучения эмбедингов

9. Где в данном примере ошибка? *

1 point

```
1 for step in range(100):
2     # выбираем случайный пример для обучения на этом этапе
3     ind_ = random.randint(0, inputs.size()[0]-1)
4     current_input = inputs[ind_]
5     current_target = targets[ind_]
6
7     outputs = model(current_input)
8
9     current_loss = loss(outputs, current_target.to(torch.long))
10
11    optimizer.step()
12    optimizer.zero_grad()
```

Mark only one oval.

- ☐ Не хватает одного этапа backward pass
- ☐ Выполнены лишние действия с оптимизатором
- ☐ Видимых ошибок нет
- ☐ Входные данные неправильно передаются в модель

10. Где в данном примере ошибка? *

2 points

```
1 class SampleNN(nn.Module):
2     def __init__(self, input_dim, hidden_dim, output_dim):
3         super(SampleNN, self).__init__()
4         self.fc1 = nn.Linear(input_dim, hidden_dim)
5         self.fc2 = nn.Linear(hidden_dim, output_dim)
6         self.relu1 = nn.ReLU()
7
8     def forward(self, x):
9         x = self.fc1(x)
10        x = self.relu1(x)
11        x = self.fc2(x)
12        return x
```

Mark only one oval.

- ☐ Ошибки нет, если далее будет подобрана правильная лосс-функция
- ☐ Функция активации использована слишком рано
- ☐ Недостаточно скрытых слоёв
- ☐ Ошибки нет в любом случае

11. В чем преимущества использования механизма внимания? *

1 point

Check all that apply.

- ☐ Интерпретируемость
- ☐ Возможность на каждом шаге генерации посмотреть на всё входное предложение и определить наиболее важные на данный момент элементы
- ☐ Работа с текстами любой длины без риска "забывания" моделью начала текста
- ☐ Возможность применения teacher forcing
- ☐ Возможность не пользоваться жадным поиском лучшего слова на каждом этапе генерации, а отслеживать ограниченное количество возможных выходных последовательностей на каждом шаге

12. Сопоставьте семейство трансформеров и задачу, которую ими наиболее удобно решать:

* 3 points

Mark only one oval per row.

	Анализ тональности	Генерация диалога	Автоматическое реферирование
BERT	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
GPT	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
BART	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. Отметьте все верные утверждения о нейросетях-трансформерах: *

2 points

Check all that apply.

- ☐ Эncoder и декодер обмениваются информацией только в одном месте
- ☐ Вычисления внутри трансформера могут происходить только последовательно
- ☐ На каждом шаге модели известно, на какой позиции в предложении находится слово
- ☐ Self-attention декодера может "смотреть вперед", но только на этапе обучения
- ☐ Модель не может состоять только из энкодера или только из декодера

14. С какими сложностями вы можете столкнуться при обучении в парадигме RLHF?

* 1 point

Check all that apply.

- ☐ Понадобится очень много аннотированных данных
- ☐ Модель-агент может научиться обманывать модель вознаграждения
- ☐ Обучение модели вознаграждения очень дорогое
- ☐ Модель обучится просто имитировать "идеальные" ответы

15. Зачем нужны специальные токены начала и конца предложения в sequence-to-sequence-нейросетях? * 1 point

Mark only one oval.

- ☐ Чтобы знать, когда начинать и заканчивать генерацию
- ☐ С их помощью осуществляется padding
- ☐ Для инициализации скрытого состояния декодера
- ☐ Чтобы знать, когда обрезать слишком длинное предложение

16. Какие нейросети обычно применяют для энкодеров и декодеров в seq2seq-нейросетях? * 1 point

Check all that apply.

- ☐ Обычные RNN
- ☐ LSTM
- ☐ Полносвязные нейросети
- ☐ Сверточные нейросети

This content is neither created nor endorsed by Google.

Google Forms