

1. Для чего может добавляться шум к весам в ходе обучения?

Добавление шума к весам способствуют тому, что выход нейросети является плавной функцией входа или его веса, соответственно. Этот метод хорошо себя проявляет в рекуррентных сетях.

2. Каким образом Вы получаете конечный ответ из ансамбля?

Суммируются прогнозы всех архитектур, а затем эта сумма делится на их количество (метод усреднения).

```
def ensemble(models, test_x, test_y):  
    preds = np.array(models[0].predict(test_x))  
    models[0].evaluate(test_x, test_y, verbose=2)  
    for i in range(1, len(models)):  
        preds = preds + np.array(models[i].predict(test_x))  
        models[i].evaluate(test_x, test_y, verbose=2)  
    print(preds / len(models))  
    print(accuracy_score(test_y, (preds / len(models)).round(), normalize=False) / 100)
```

Таким образом мы получаем прогнозы для ансамбля сетей.

3. Какие методы надо определить в пользовательском callback, чтобы он срабатывал в начале обучения и конце каждой эпохи?

Чтобы custom callback срабатывал в начале обучения, в нем надо определить метод `on_train_begin(self, logs=None)`.

Чтобы custom callback срабатывал в конце каждой эпохи, в нем надо определить метод `on_epoch_end(self, epoch, logs = None)`.