

**1 Постройте нейронную сеть (берем простую линейную сеть, которую разбирали на уроке: меняем число слоев, число нейронов, типы активации, тип оптимизатора) на датасет `from sklearn.datasets import load_boston`.**

Model: "sequential\_47"

Layer (type)	Output Shape	Param #
dense_133 (Dense)	(None, 64)	896
dropout_23 (Dropout)	(None, 64)	0
dense_134 (Dense)	(None, 32)	2,080
dense_135 (Dense)	(None, 4)	132

Total params: 3,108 (12.14 KB)

Trainable params: 3,108 (12.14 KB)

Non-trainable params: 0 (0.00 B)

```
Epoch 197/200
9/9 0s 10ms/step - accuracy: 0.9069 - loss: 0.2279 - val_accuracy: 0.6761 - val_loss: 0.6703
Epoch 198/200
9/9 0s 12ms/step - accuracy: 0.9242 - loss: 0.1876 - val_accuracy: 0.6761 - val_loss: 0.6773
Epoch 199/200
9/9 0s 10ms/step - accuracy: 0.9233 - loss: 0.2089 - val_accuracy: 0.6901 - val_loss: 0.6773
Epoch 200/200
9/9 0s 10ms/step - accuracy: 0.9179 - loss: 0.1894 - val_accuracy: 0.6901 - val_loss: 0.6882
5/5 - 0s - 9ms/step - accuracy: 0.8092 - loss: 0.6248
Test acc = 0.8092 %
5/5 0s 14ms/step
Acc = 0.8092 %
```

**2 Измените функцию потерь и метрику для этой задачи. Постройте 10-15 вариантов и сведите результаты их работы в таблицу. Опишите, какого результата вы добились от нейросети? Что помогло вам улучшить ее точность?**

Model: "sequential\_47"

Layer (type)	Output Shape	Param #
dense_133 (Dense)	(None, 64)	896
dropout_23 (Dropout)	(None, 64)	0
dense_134 (Dense)	(None, 32)	2,080
dense_135 (Dense)	(None, 4)	132

Total params: 3,108 (12.14 KB)

Trainable params: 3,108 (12.14 KB)

Non-trainable params: 0 (0.00 B)

Функции потерь	categorical_crossentropy	kullback_leibler_divergence	poisson
Метрики			
Accuracy	(0.8092) 0.8092	(0.7895) 0.7895	(0.7895) 0.7895
Precision	(0.7682) 0.7682	(0.8092) 0.8092	(0.7815) 0.7763
Recall	(0.7829) 0.8026	(0.7829) 0.7829	(0.8026) 0.8092
AUC	(0.9515) 0.7763	(0.9551) 0.8026	(0.9521) 0.8092
mse	(0.0771) 0.7829	(0.08) 0.7895	(0.0712) 0.7961

В основном все результаты точности нейронной сети очень близки, в диапазоне от 0.77 до 0.81. Разница метрик между функциями потерь тоже небольшая, либо ее практически нет.

Самое главное, что помогло улучшить работу нейронной сети, это:

- изменение количества до 4 с разницей в значениях (0, 20, 25, 35)
- изменение количества нейронов в слоях до 64 и 32
- изменения значения слоя Dropout до 0.2
- изменение количества эпох до 200

**3 Поработайте с документацией TensorFlow 2. Найти 2-3 полезные команды TensorFlow, не разобранные на уроке (полезные для Вас)**

`tf.keras.utils.get_file()` — загружает файл с URL и кеширует его в локальном каталоге (удобно для датасетов/ресурсов).

`tf.keras.backend.clear_session()` — освобождает ресурсы Keras (графы, модели) — полезно при многократном создании/обучении моделей в одном ноутбуке (избегает утечек памяти и конфликтов имён слоёв)