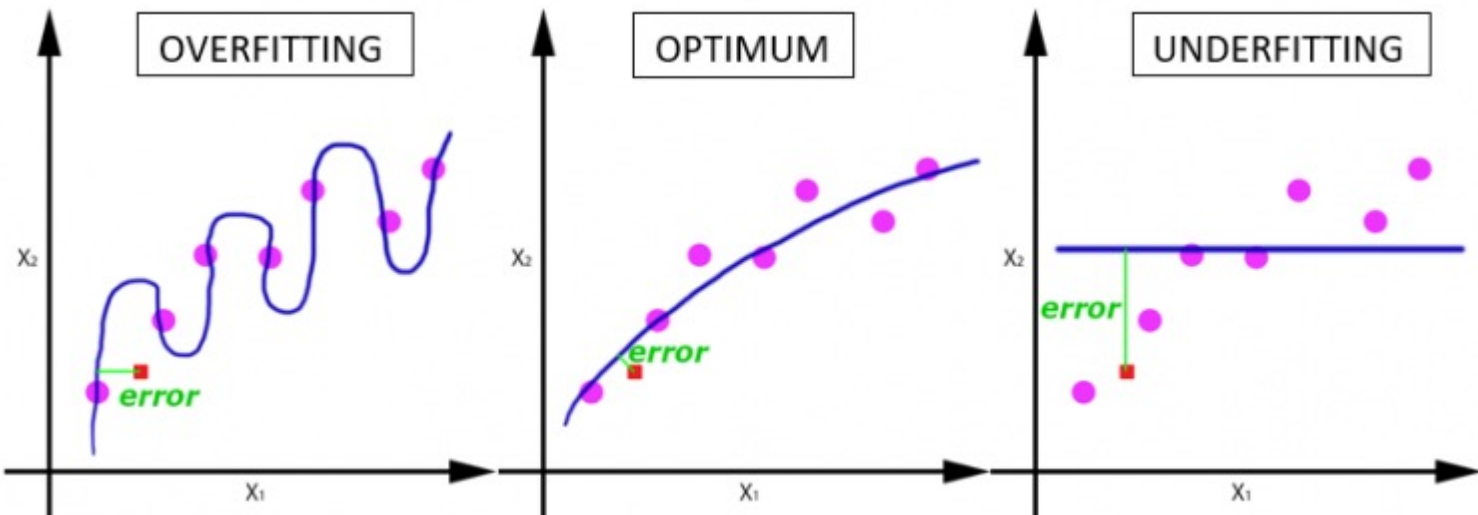
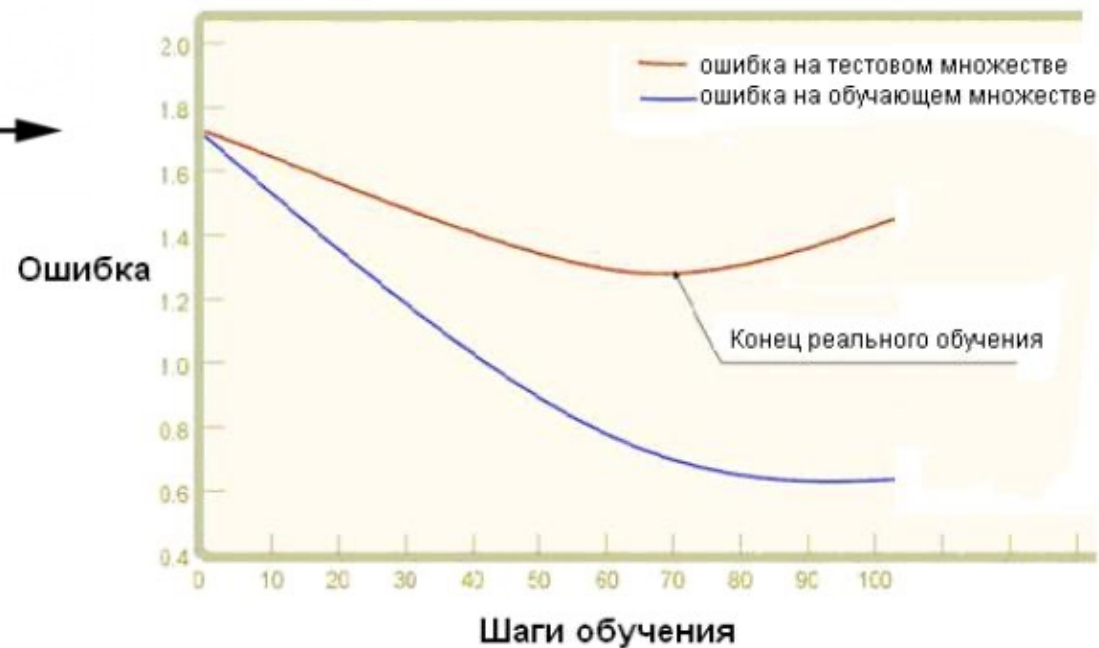
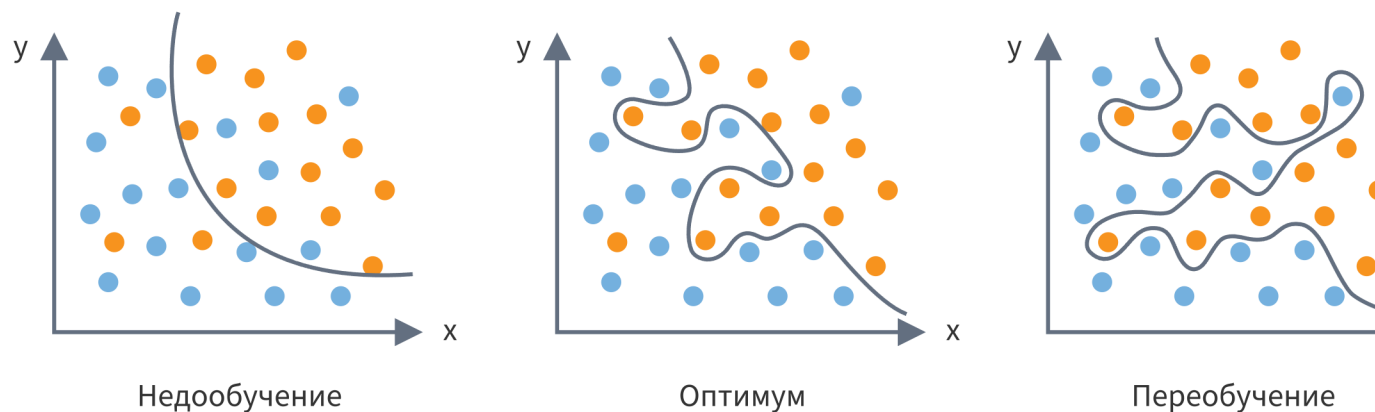
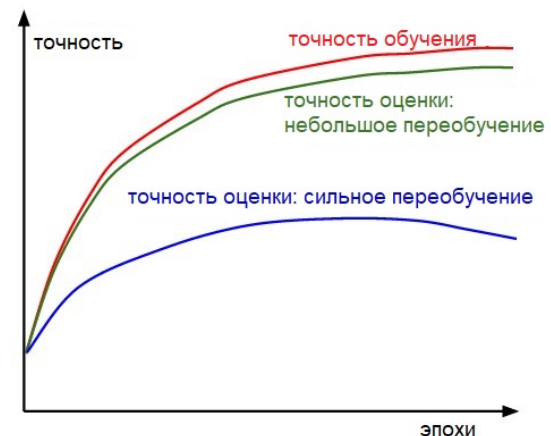


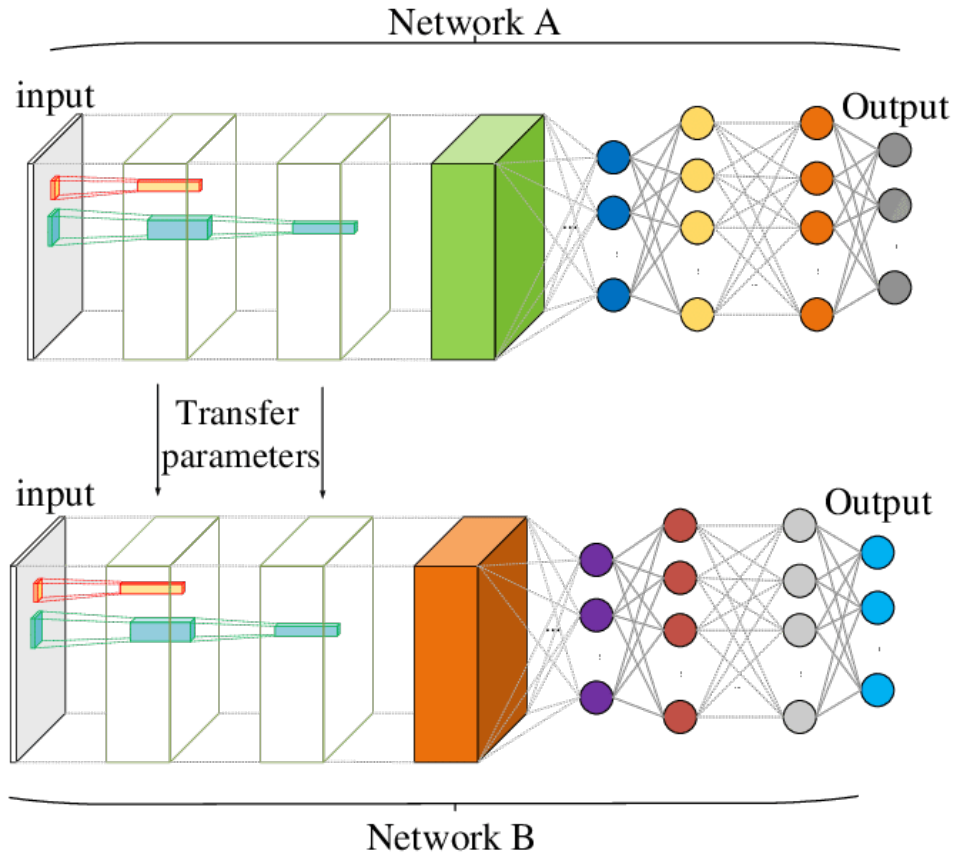
Точность, переобучение



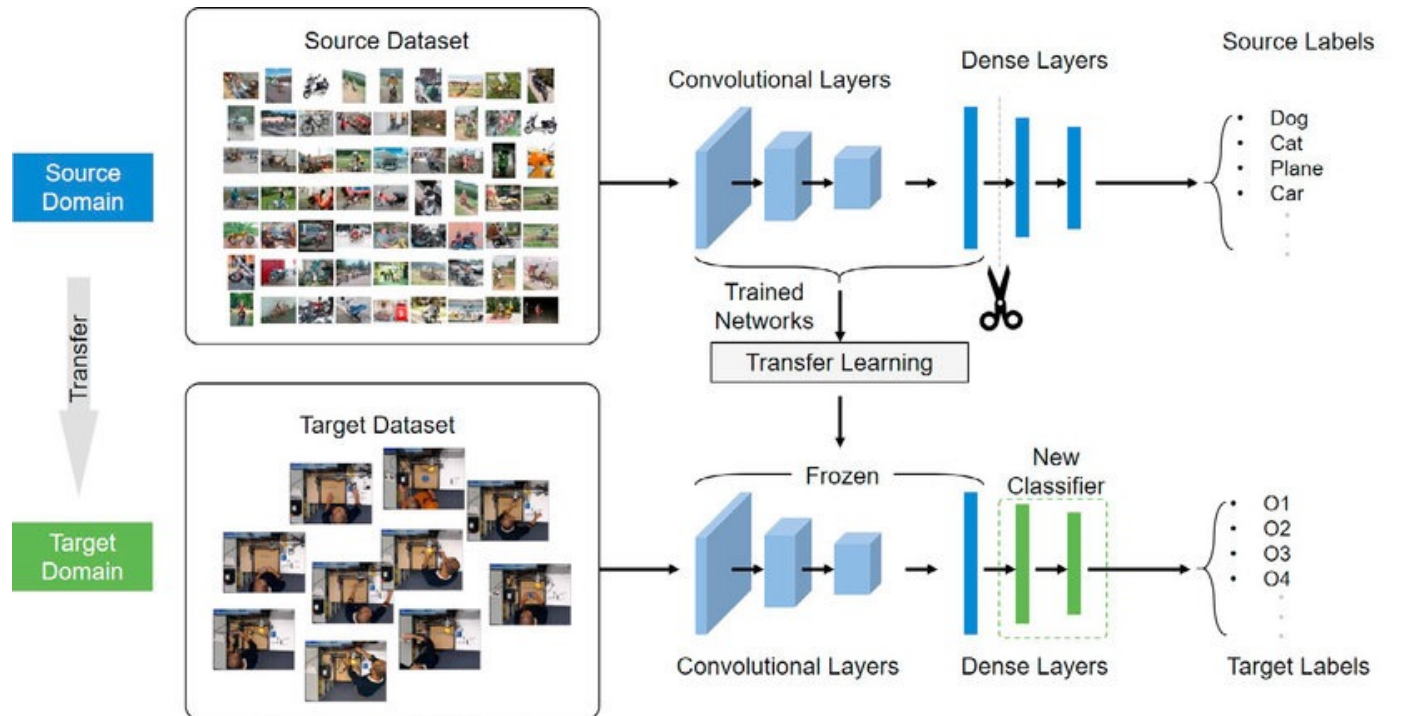
- Переобучение при долгом обучении слишком сложной модели



Дообучение - transfer learning



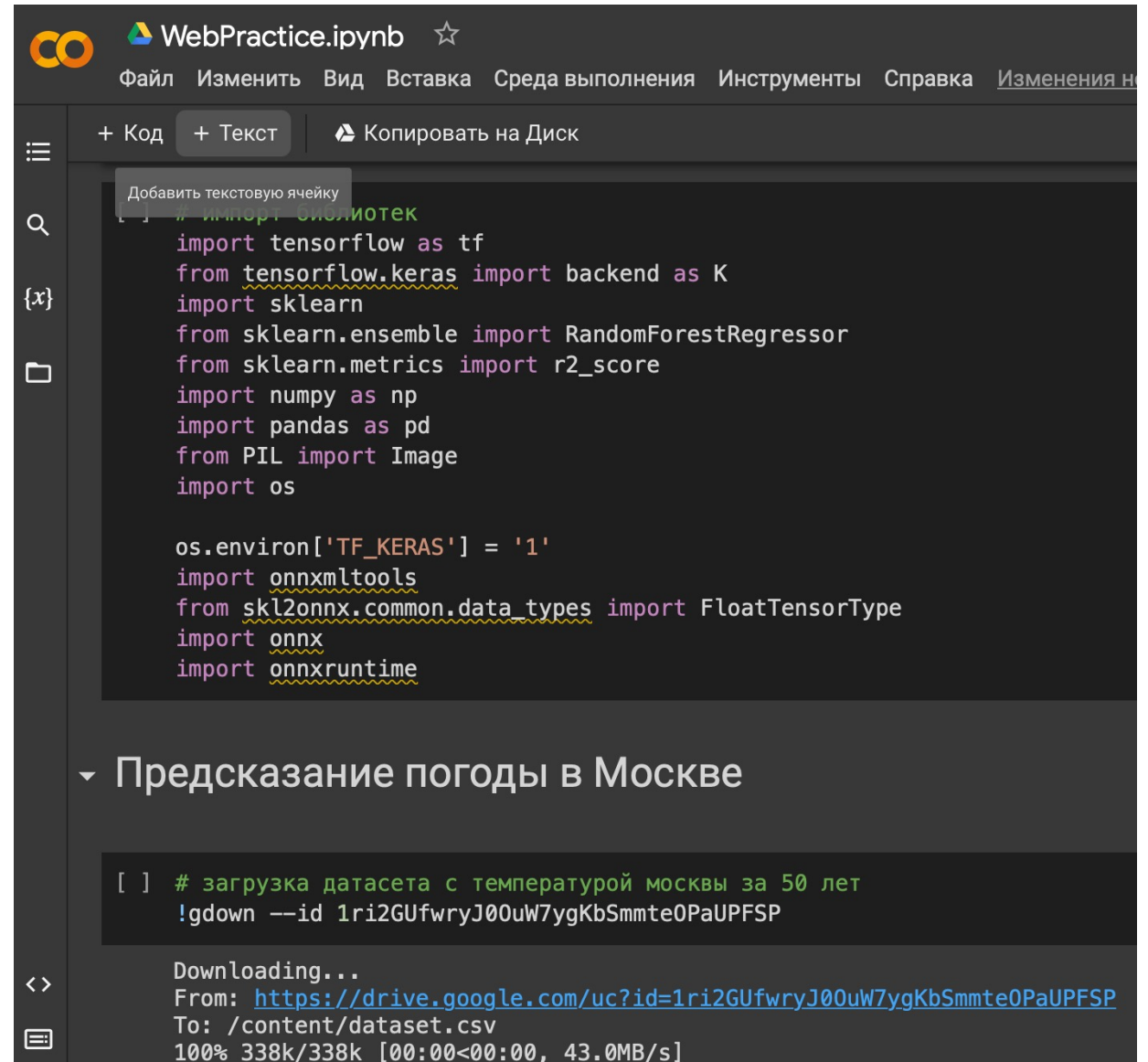
- Использование готовых моделей
- Для близких задач на новых класса можно применить transfer learning



Google Colab

- Простой пример работы с библиотекой ONNX

<https://github.com/iu5git/Deep-learning/blob/main/notebooks/WebPractice.ipynb>



```
WebPractice.ipynb ☆
Файл  Изменить  Вид  Вставка  Среда выполнения  Инструменты  Справка  Изменения н
+ Код  + Текст  Копировать на Диск
Добавить текстовую ячейку
[ ] # импорт библиотек
import tensorflow as tf
from tensorflow.keras import backend as K
import sklearn
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
import numpy as np
import pandas as pd
from PIL import Image
import os

os.environ['TF_KERAS'] = '1'
import onnxmltools
from skl2onnx.common.data_types import FloatTensorType
import onnx
import onnxruntime

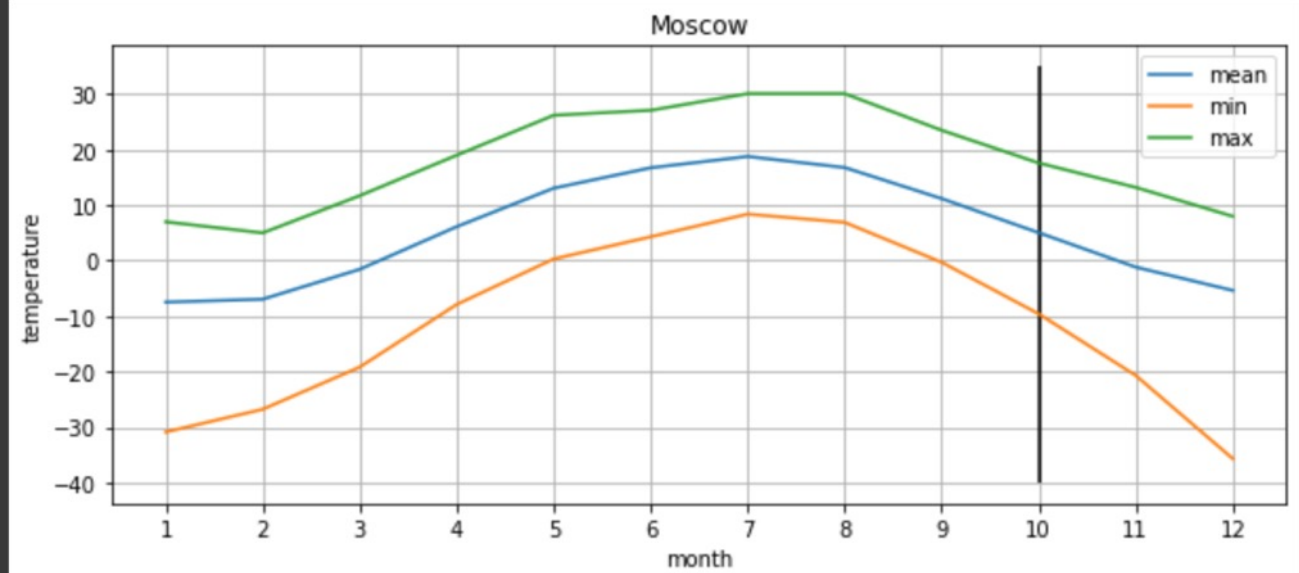
▼ Предсказание погоды в Москве

[ ] # загрузка датасета с температурой москвы за 50 лет
!gdown --id 1ri2GUfwryJ00uW7ygKbSmmte0PaUPFSP

Downloading...
From: https://drive.google.com/uc?id=1ri2GUfwryJ00uW7ygKbSmmte0PaUPFSP
To: /content/dataset.csv
100% 338k/338k [00:00<00:00, 43.0MB/s]
```

Google Colab

```
# визуализация данных
ax = df.groupby('month').\
    Temperature.\
    agg(['mean', 'min', 'max']).\
    plot(figsize=(10, 4))
ax.set_xticks([i for i in range(1, 13)])
ax.set_ylabel('temperature')
ax.set_title('Moscow')
ax.vlines(10, -40, 35)
ax.grid()
```



ONNX

- Использование модели ONNX в Python

```
def predictImageData(modelName, filePath):  
    img = Image.open(filePath).convert("RGB")  
    img = np.asarray(img.resize((32, 32), Image.ANTIALIAS))  
    sess = onnxruntime.InferenceSession(r'C:\DZ1\media\models\cifar100.onnx') #<-здесь требуется  
    outputOFModel = np.argmax(sess.run(None, {'input': np.asarray([img]).astype(np.float32)}))  
    score = imageClassList[str(outputOFModel)]  
    return score
```

ONNX

- Использование модели ONNX в JS

```
let reader = new FileReader();
reader.onloadend = async function () {
  onnxSess = new any InferenceSession();
  await onnxSess.loadModel(reader.result);
  const img = new Float32Array(32 * 32 * 3);
  img.fill(1);
  const input = new onnx.Tensor(img, "float32", [1, 32, 32, 3]);
  const output = (await onnxSess.run([input])).get("output").data;
```