

Принципы работы нейросети на примере распознавания символов

Автор:

Виноградов Всеволод Алексеевич

Научный руководитель:

Быстров Алекснадр Александрович

Введение

Искусственные нейронные сети активно развиваются и применяются в различных областях. Но многие считают их сложными для понимания. Я объясню принцип работы нейросетей на примере распознавания рукописных цифр.

Постановка задачи

Задача распознавания рукописных цифр заключается в преобразовании изображения с помощью «чёрного ящика» из 10 нейронов (функций вероятности), в вероятность соответствия каждой цифре от 0 до 9. Затем выбирается цифра с наибольшей вероятностью.

Работа с функциями вероятности

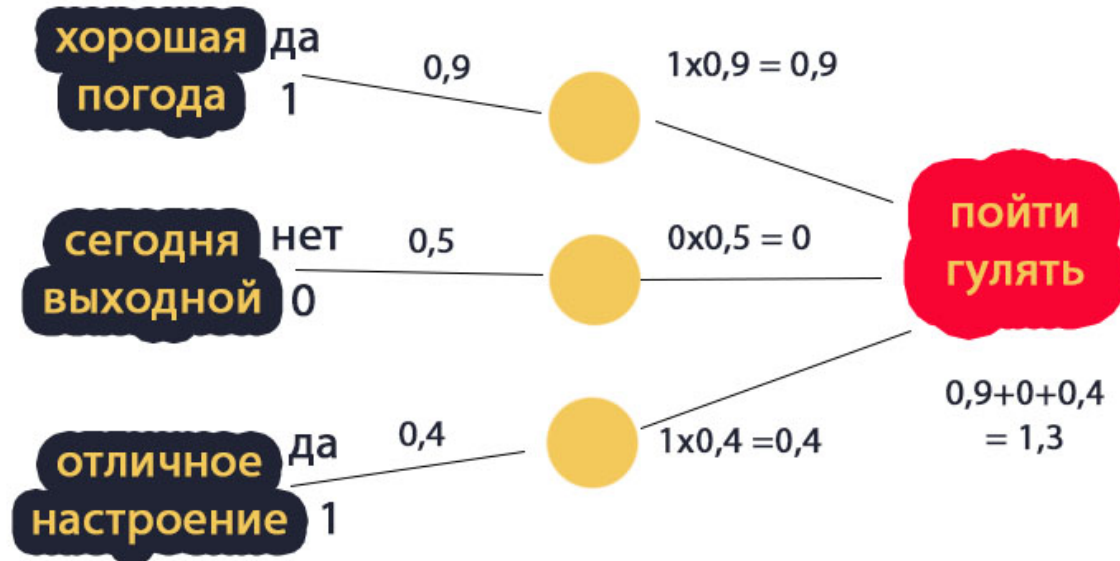
Эти нейроны (обозначим $P(\mathbf{X}, \boldsymbol{\theta})$) зависят от вектора весов $\boldsymbol{\theta}$ и вектора входов \mathbf{X} .

Изначально $\boldsymbol{\theta}$ можно заполнить случайно. А $P(\mathbf{X}, \boldsymbol{\theta})$ давайте зададим линейно:

$$P(\mathbf{X}, \boldsymbol{\theta}) = P(\mathbf{X}, \boldsymbol{\theta}) = \theta_0 + X_1 * \theta_1 + X_2 * \theta_2 + \dots + X_n * \theta_n$$

Здесь θ_0 — сдвиг (первоначальная вероятность) нейрона.

Пример



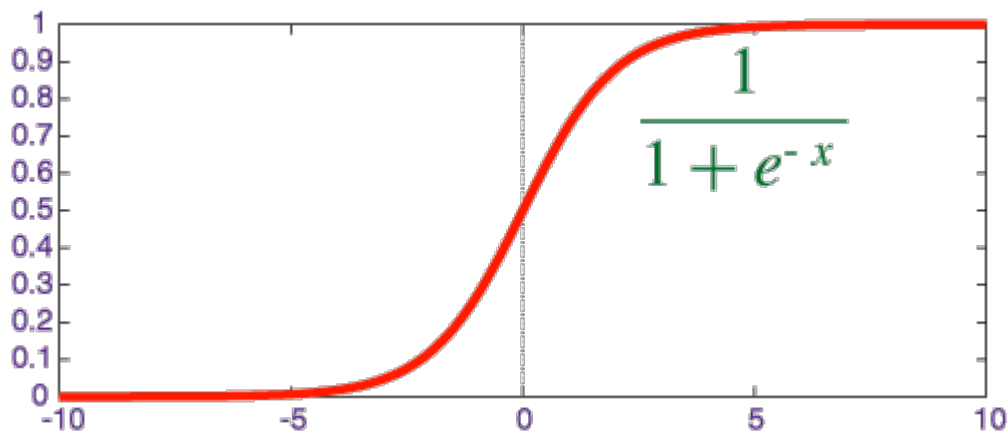
Как видно
нейрон
заданный так
способен
находить
вероятность

прогулки, однако $1,3 > 1$ и не
похожа на вероятность.

Функция активации

Чтобы внести значения нейронов в промежутки используют функции активации, нам подходит логистический сигмоид $\sigma(x)$.

$$\sigma(1,3)=0,785$$



Функция отклонений

Обучение – нахождение весов θ .

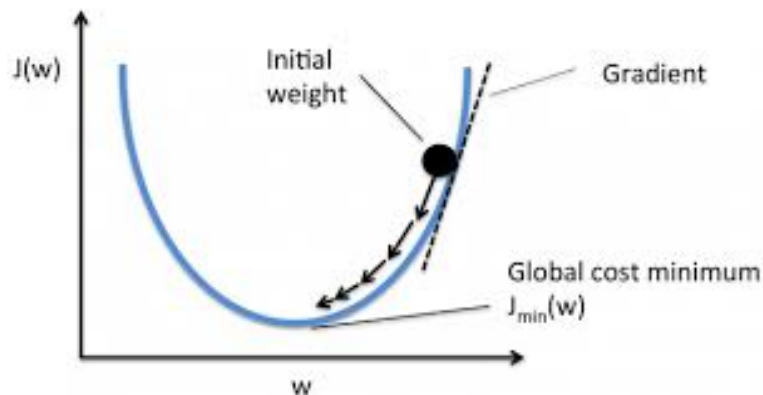
Чтобы обучить нейросеть надо сделать отклонения ответов нейросети от верных минимальным.

Сумма квадратов отклонений:

$$E(Y, \hat{Y}) = (Y_1 - \hat{Y}_1)^2 + (Y_2 - \hat{Y}_2)^2 + \dots + (Y_n - \hat{Y}_n)^2$$

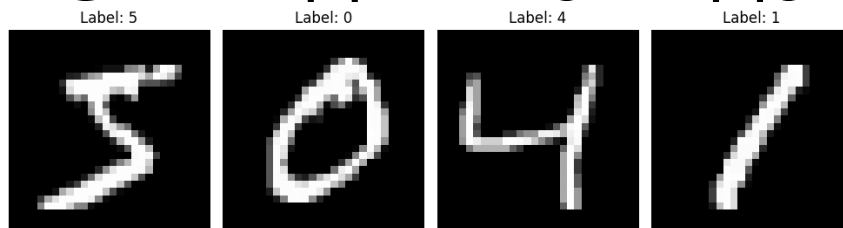
Оптимизация функции отклонений

Для минимизации функций при градиентном спуске, мы от каждого веса отнимаем скорость роста функции при его росте на шаг обучения.



Данные для обучения

Обучающие данные взяты из набора MNIST. Он содержит 60 000 изображений размером 28 на 28 пикселей, представленных числами от 0 до 255, для обучения и 10 000 изображений для валидации.

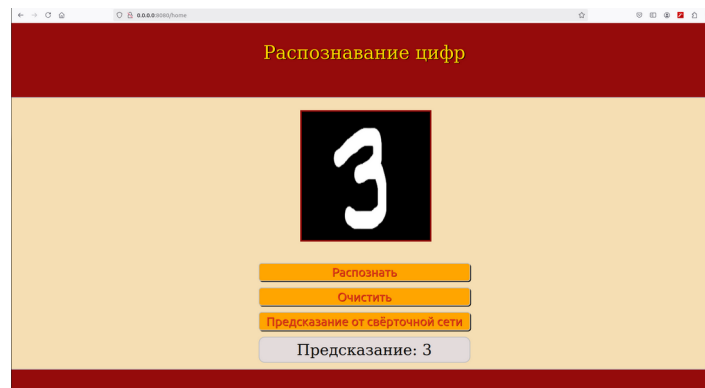
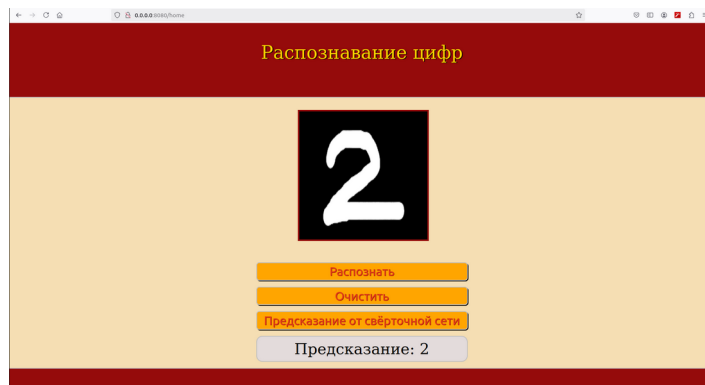


Создание нейросети на keras

1. Распаковываем MNIST.
2. Делим пиксели на 255.
3. Создаём модель из одного слоя с `activation="sigmoid"`.
4. Компилируем с `optimizer="sgd"`, `loss="mean_squared_method"`.
5. Обучаем по распакованному.

Испытание

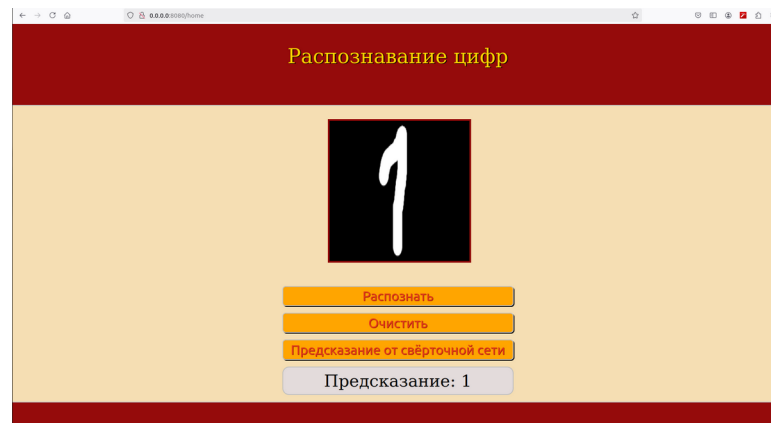
```
Epoch 1/30
1875/1875 — 2s 1ms/step - accuracy: 0.2274 - loss: 0.1338 - val_accuracy: 0.5357 - val_loss: 0.0821
Epoch 2/30
1875/1875 — 2s 854us/step - accuracy: 0.5729 - loss: 0.0788 - val_accuracy: 0.6338 - val_loss: 0.0703
Epoch 3/30
1875/1875 — 2s 873us/step - accuracy: 0.6502 - loss: 0.0679 - val_accuracy: 0.6822 - val_loss: 0.0621
Epoch 4/30
1875/1875 — 2s 964us/step - accuracy: 0.6949 - loss: 0.0608 - val_accuracy: 0.7223 - val_loss: 0.0565
Epoch 5/30
1875/1875 — 2s 1ms/step - accuracy: 0.7305 - loss: 0.0556 - val_accuracy: 0.7528 - val_loss: 0.0524
Epoch 6/30
1875/1875 — 2s 947us/step - accuracy: 0.7604 - loss: 0.0518 - val_accuracy: 0.7795 - val_loss: 0.0492
Epoch 7/30
1875/1875 — 2s 901us/step - accuracy: 0.7867 - loss: 0.0488 - val_accuracy: 0.8019 - val_loss: 0.0466
Epoch 8/30
1875/1875 — 2s 884us/step - accuracy: 0.8027 - loss: 0.0463 - val_accuracy: 0.8185 - val_loss: 0.0445
Epoch 9/30
1875/1875 — 2s 876us/step - accuracy: 0.8166 - loss: 0.0442 - val_accuracy: 0.8283 - val_loss: 0.0427
Epoch 10/30
1875/1875 — 2s 889us/step - accuracy: 0.8220 - loss: 0.0427 - val_accuracy: 0.8341 - val_loss: 0.0411
Epoch 11/30
1875/1875 — 2s 880us/step - accuracy: 0.8296 - loss: 0.0413 - val_accuracy: 0.8392 - val_loss: 0.0397
Epoch 12/30
1875/1875 — 2s 1ms/step - accuracy: 0.8333 - loss: 0.0399 - val_accuracy: 0.8431 - val_loss: 0.0385
Epoch 13/30
1875/1875 — 2s 891us/step - accuracy: 0.8356 - loss: 0.0391 - val_accuracy: 0.8463 - val_loss: 0.0375
Epoch 14/30
1875/1875 — 2s 902us/step - accuracy: 0.8426 - loss: 0.0376 - val_accuracy: 0.8490 - val_loss: 0.0366
Epoch 15/30
1875/1875 — 2s 950us/step - accuracy: 0.8446 - loss: 0.0369 - val_accuracy: 0.8529 - val_loss: 0.0358
Epoch 16/30
1875/1875 — 2s 937us/step - accuracy: 0.8462 - loss: 0.0361 - val_accuracy: 0.8553 - val_loss: 0.0350
Epoch 17/30
1875/1875 — 2s 951us/step - accuracy: 0.8484 - loss: 0.0354 - val_accuracy: 0.8577 - val_loss: 0.0343
Epoch 18/30
1875/1875 — 2s 946us/step - accuracy: 0.8533 - loss: 0.0345 - val_accuracy: 0.8609 - val_loss: 0.0337
Epoch 19/30
1875/1875 — 2s 922us/step - accuracy: 0.8527 - loss: 0.0342 - val_accuracy: 0.8624 - val_loss: 0.0332
Epoch 20/30
1875/1875 — 2s 933us/step - accuracy: 0.8541 - loss: 0.0336 - val_accuracy: 0.8638 - val_loss: 0.0326
```



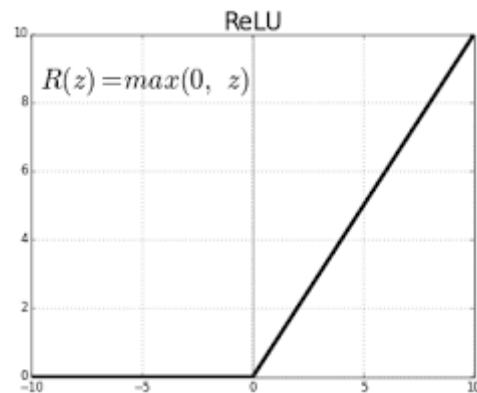
На 30 шаге точность – 86%.

Добавление слоёв

```
Epoch 1/30
1875/1875 — 3s 2ms/step - accuracy: 0.1678 - loss: 0.1304 - val_accuracy: 0.3311 - val_loss: 0.0875
Epoch 2/30
1875/1875 — 3s 1ms/step - accuracy: 0.3760 - loss: 0.0859 - val_accuracy: 0.4604 - val_loss: 0.0793
Epoch 3/30
1875/1875 — 3s 2ms/step - accuracy: 0.4700 - loss: 0.0770 - val_accuracy: 0.5281 - val_loss: 0.0699
Epoch 4/30
1875/1875 — 3s 1ms/step - accuracy: 0.5524 - loss: 0.0681 - val_accuracy: 0.6438 - val_loss: 0.0610
Epoch 5/30
1875/1875 — 3s 1ms/step - accuracy: 0.6723 - loss: 0.0592 - val_accuracy: 0.7594 - val_loss: 0.0520
Epoch 6/30
1875/1875 — 3s 2ms/step - accuracy: 0.7671 - loss: 0.0507 - val_accuracy: 0.8054 - val_loss: 0.0444
Epoch 7/30
1875/1875 — 3s 1ms/step - accuracy: 0.8088 - loss: 0.0437 - val_accuracy: 0.8293 - val_loss: 0.0388
Epoch 8/30
1875/1875 — 3s 1ms/step - accuracy: 0.8285 - loss: 0.0386 - val_accuracy: 0.8462 - val_loss: 0.0348
Epoch 9/30
1875/1875 — 3s 1ms/step - accuracy: 0.8412 - loss: 0.0348 - val_accuracy: 0.8592 - val_loss: 0.0317
Epoch 10/30
1875/1875 — 3s 1ms/step - accuracy: 0.8517 - loss: 0.0322 - val_accuracy: 0.8670 - val_loss: 0.0293
Epoch 11/30
1875/1875 — 3s 1ms/step - accuracy: 0.8621 - loss: 0.0297 - val_accuracy: 0.8715 - val_loss: 0.0274
Epoch 12/30
1875/1875 — 3s 2ms/step - accuracy: 0.8675 - loss: 0.0279 - val_accuracy: 0.8771 - val_loss: 0.0258
Epoch 13/30
1875/1875 — 3s 1ms/step - accuracy: 0.8733 - loss: 0.0263 - val_accuracy: 0.8820 - val_loss: 0.0245
Epoch 14/30
1875/1875 — 3s 1ms/step - accuracy: 0.8778 - loss: 0.0249 - val_accuracy: 0.8867 - val_loss: 0.0234
Epoch 15/30
1875/1875 — 3s 2ms/step - accuracy: 0.8793 - loss: 0.0241 - val_accuracy: 0.8886 - val_loss: 0.0225
Epoch 16/30
1875/1875 — 3s 2ms/step - accuracy: 0.8841 - loss: 0.0232 - val_accuracy: 0.8918 - val_loss: 0.0216
Epoch 17/30
1875/1875 — 3s 1ms/step - accuracy: 0.8862 - loss: 0.0224 - val_accuracy: 0.8942 - val_loss: 0.0209
Epoch 18/30
1875/1875 — 3s 2ms/step - accuracy: 0.8867 - loss: 0.0219 - val_accuracy: 0.8977 - val_loss: 0.0203
Epoch 19/30
1875/1875 — 3s 2ms/step - accuracy: 0.8922 - loss: 0.0208 - val_accuracy: 0.8994 - val_loss: 0.0197
Epoch 20/30
1875/1875 — 3s 1ms/step - accuracy: 0.8926 - loss: 0.0205 - val_accuracy: 0.9007 - val_loss: 0.0192
```

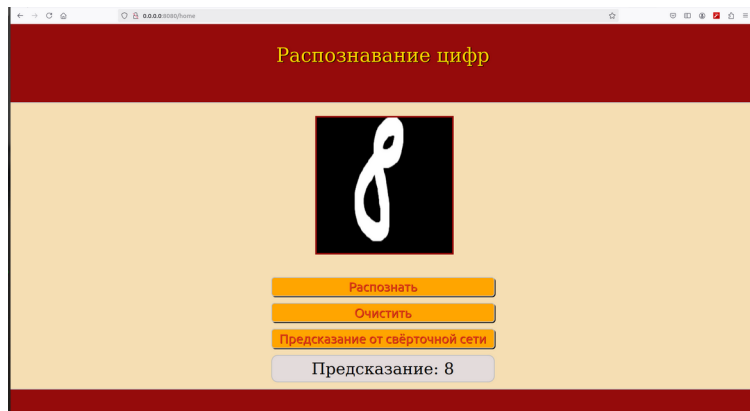
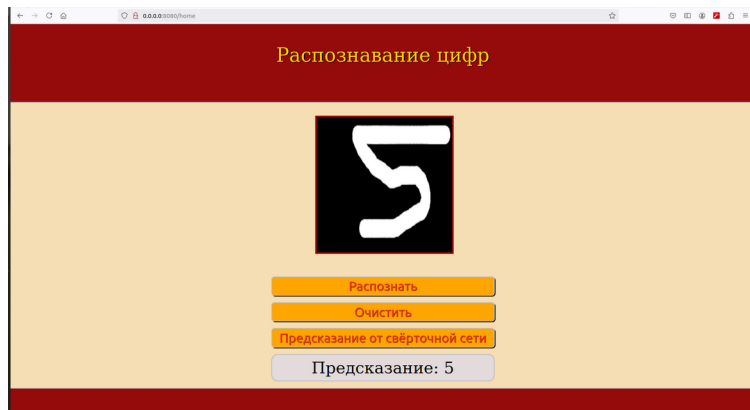


Точность
МОЖНО
повысить до
91%, добавив 2
слоя с RELU
функцией
активации.



Кросс-энтропия

```
Epoch 1/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.7169 - loss: 1.0543 - val_accuracy: 0.9048 - val_loss: 0.3237
Epoch 2/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9117 - loss: 0.3100 - val_accuracy: 0.9257 - val_loss: 0.2582
Epoch 3/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9274 - loss: 0.2551 - val_accuracy: 0.9351 - val_loss: 0.2188
Epoch 4/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9385 - loss: 0.2140 - val_accuracy: 0.9435 - val_loss: 0.1915
Epoch 5/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9466 - loss: 0.1849 - val_accuracy: 0.9484 - val_loss: 0.1746
Epoch 6/30
1875/1875 ————— 3s 1ms/step - accuracy: 0.9524 - loss: 0.1657 - val_accuracy: 0.9522 - val_loss: 0.1613
Epoch 7/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9577 - loss: 0.1519 - val_accuracy: 0.9574 - val_loss: 0.1410
Epoch 8/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9635 - loss: 0.1291 - val_accuracy: 0.9605 - val_loss: 0.1339
Epoch 9/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9659 - loss: 0.1191 - val_accuracy: 0.9618 - val_loss: 0.1249
Epoch 10/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9701 - loss: 0.1104 - val_accuracy: 0.9643 - val_loss: 0.1169
Epoch 11/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9704 - loss: 0.1056 - val_accuracy: 0.9662 - val_loss: 0.1110
Epoch 12/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9732 - loss: 0.0944 - val_accuracy: 0.9676 - val_loss: 0.1061
Epoch 13/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9762 - loss: 0.0857 - val_accuracy: 0.9674 - val_loss: 0.1045
Epoch 14/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9770 - loss: 0.0828 - val_accuracy: 0.9692 - val_loss: 0.0991
Epoch 15/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9779 - loss: 0.0790 - val_accuracy: 0.9712 - val_loss: 0.0964
Epoch 16/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9793 - loss: 0.0731 - val_accuracy: 0.9719 - val_loss: 0.0939
Epoch 17/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9808 - loss: 0.0686 - val_accuracy: 0.9726 - val_loss: 0.0906
Epoch 18/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9820 - loss: 0.0645 - val_accuracy: 0.9726 - val_loss: 0.0898
Epoch 19/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9834 - loss: 0.0618 - val_accuracy: 0.9734 - val_loss: 0.0893
Epoch 20/30
1875/1875 ————— 3s 2ms/step - accuracy: 0.9844 - loss: 0.0580 - val_accuracy: 0.9730 - val_loss: 0.0864
```



Точность – 99.2%

Быстрее, точнее, сложнее

Всю презентацию мы использовали самую простую модель нейросети под названием перцептрон, но есть ещё много более сложных. Свёрточные – для распознавания изображений, рекуррентные – для генерации текста, глубокие – в сложных задачах, автокодировщики, генеративные и т.д.

Спасибо за внимание!