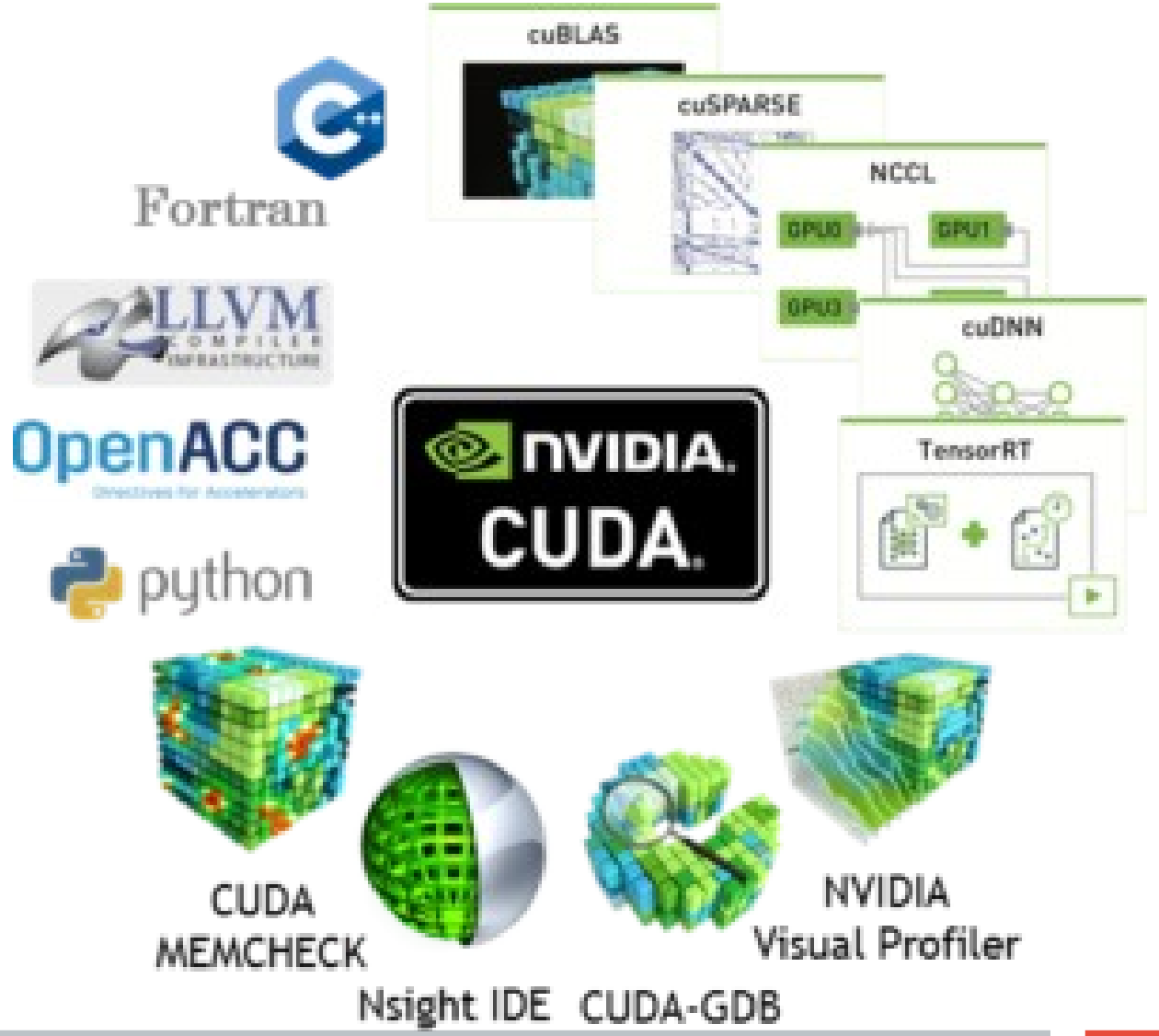
Нейросети на практике

Nvidia - CUDA


cuBLAS
cuSPARSE
NCCL
GPU0
GPU1
GPU3
cuDNN
TensorRT
Fortran
LLVM
COMPILER
INFRASTRUCTURE
OpenACC
python
NVIDIA
CUDA
CUDA
MEMCHECK
Nsight IDE
CUDA-GDB
Debugger
NVIDIA
Visual Profiler

Главное – процесс обучения.

Смотрим на графики тренировки:

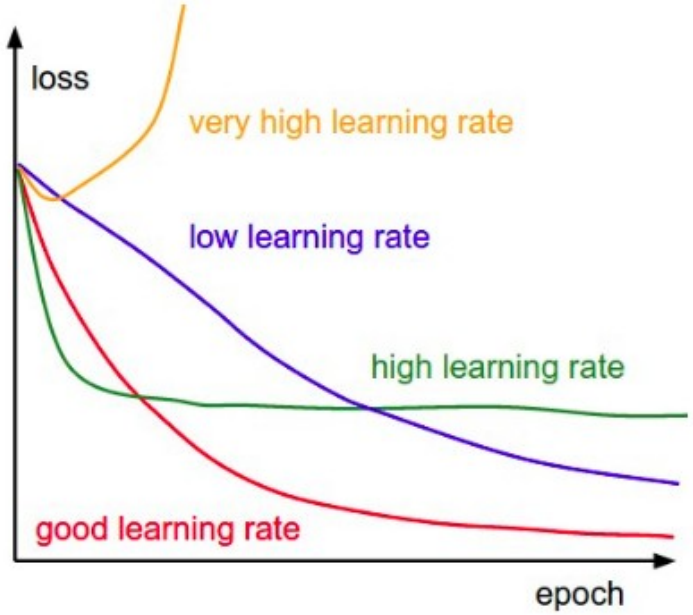
Val loss и train loss – близки друг другу и уменьшаются медленно → underfitting.

Следует начать с этого:

Preprocessing и Optimizers(Adam).

Скорость обучения(Learning rate)

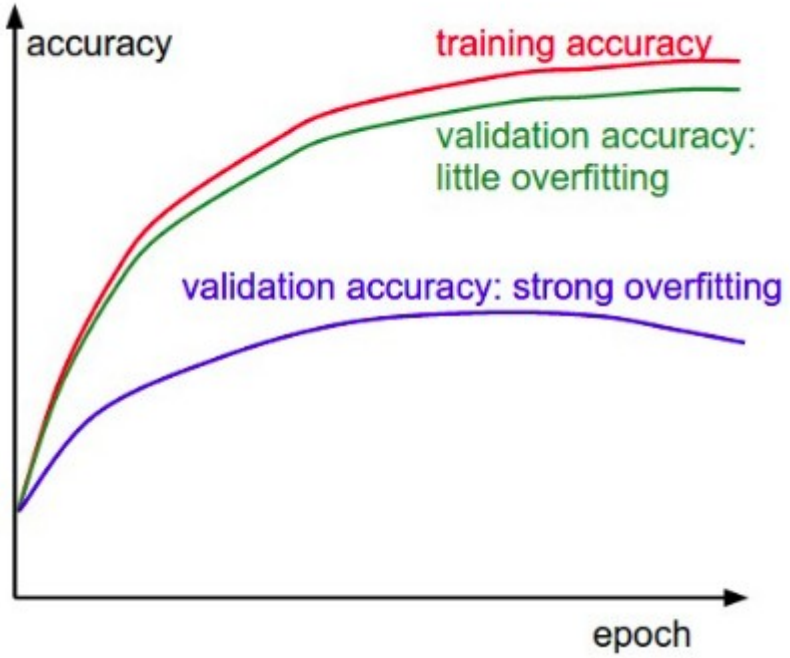
Annealing




```
torch.optim.lr_scheduler.
```

https://cs231n.github.io/neural-networks-3/

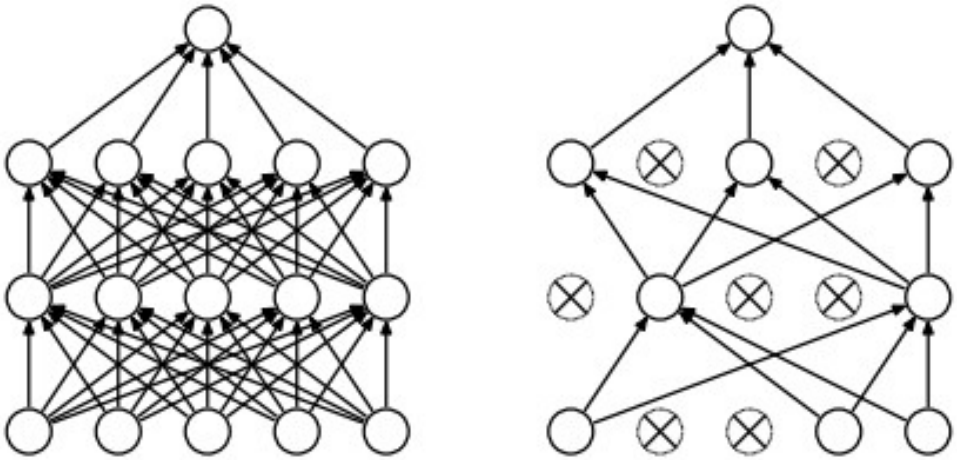
Overfitting на практике



Решение:
1. Регуляризация – L2;

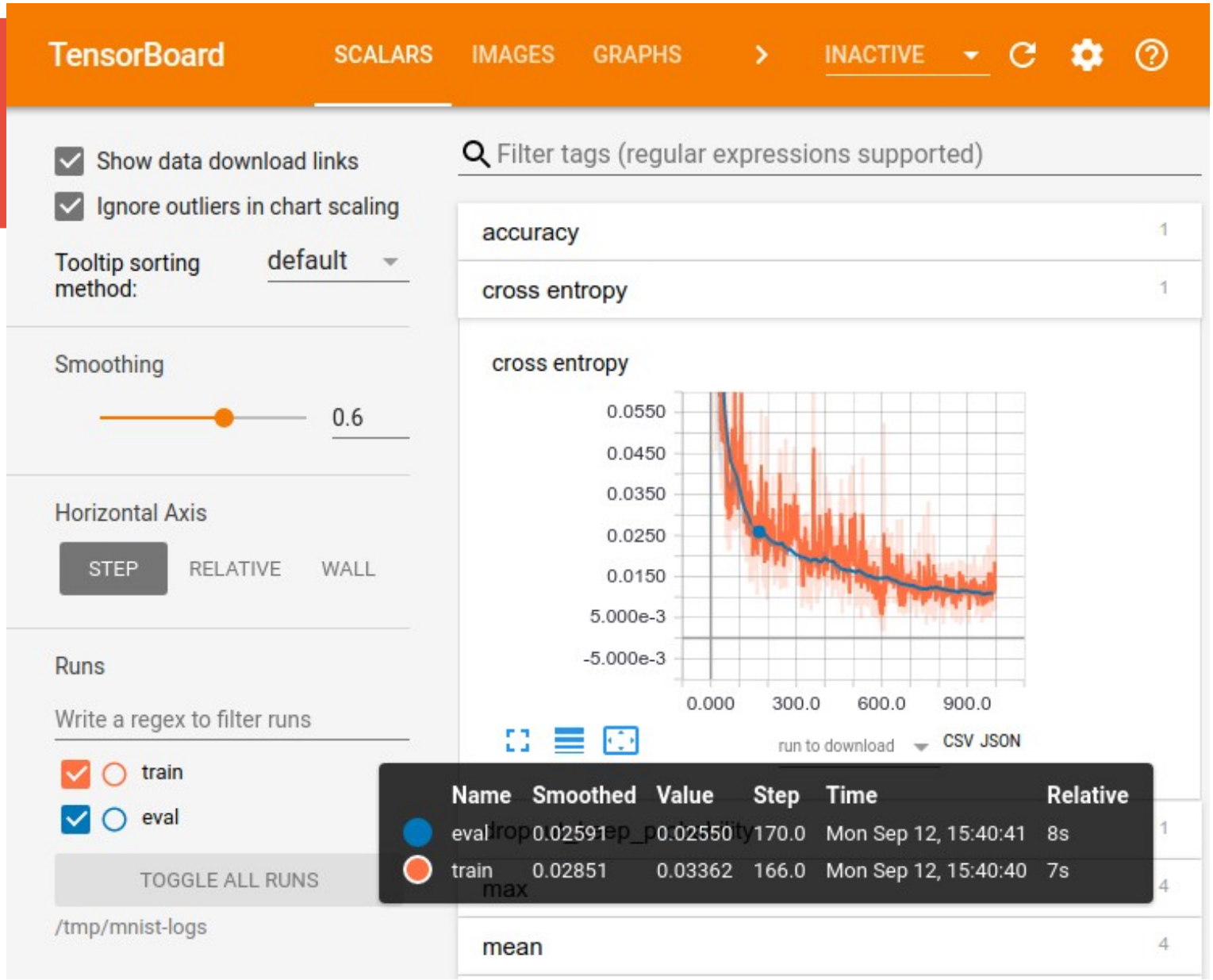
https://cs231n.github.io/neural-networks-3/

2. Dropout – статья (https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf)



(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.



Информация тут- https://habr.com/ru/post/349338/

Batch Normalization

- Ускоряет и стабилизирует тренировку
- Регуляризирует
- Не так важна инициализация

! Реализация – слой сети (В PyTorch, например)

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

https://arxiv.org/abs/1502.03167

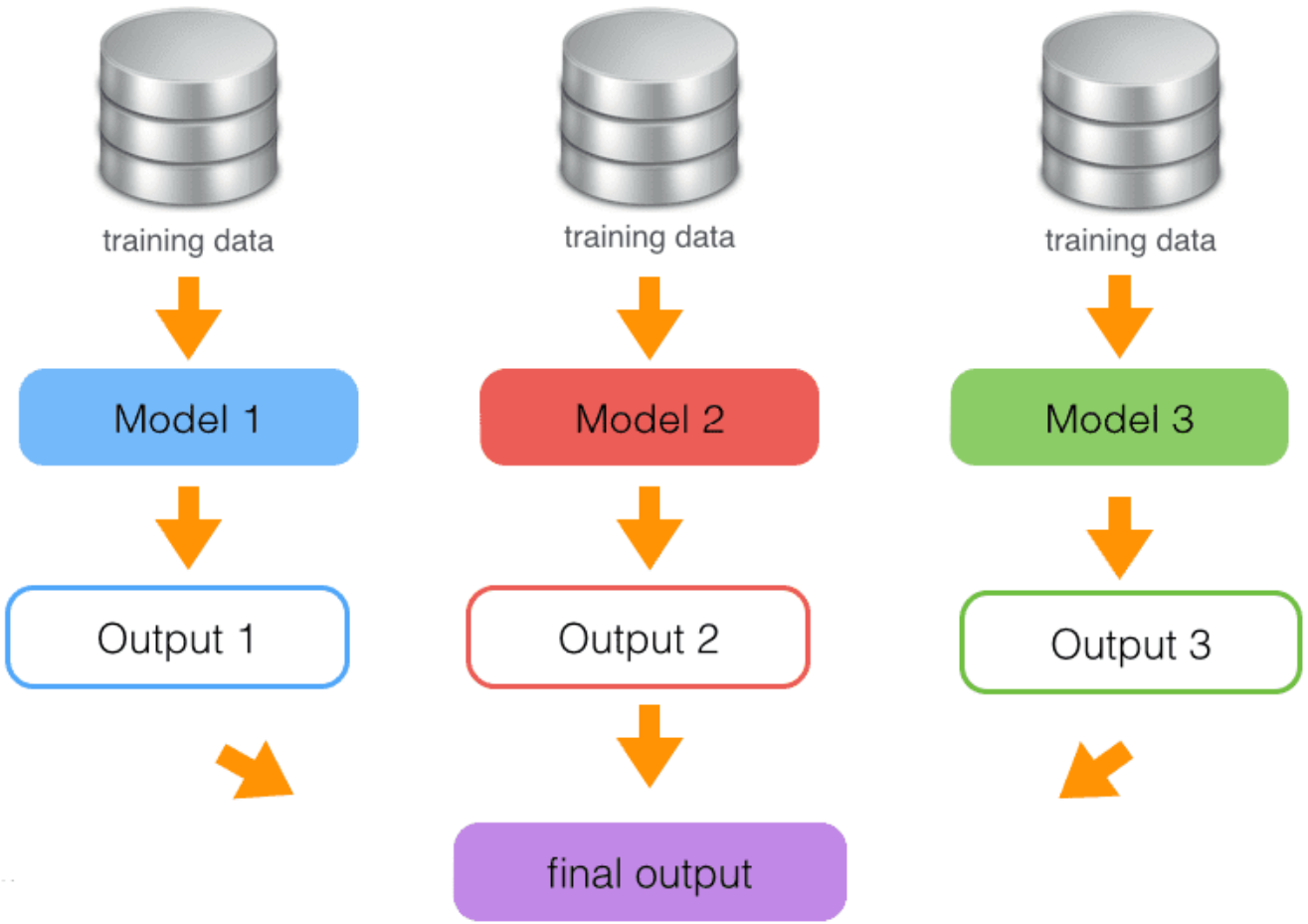
Почитать тут и здесь

Подбор гиперпараметров

1. learning rate;
2. batch size;
3. коэффицент регуляризации;
4. .....

Random Search

Ансамбль моделей



Практические советы

Шаги перед решением какой-либо задачи:

1. Найти наиболее подходящий датасет или есть разобраться какимм способами будете его составлять.

2. Поиск статей, описывающих решение вашей или близкой проблемы.

3. Написать код алгоритма, используя самые базовые алгоритмы/подходы.

4. Подбор гиперпараметров = серия экспериментов.

5. Сделать выводы и улучшать!