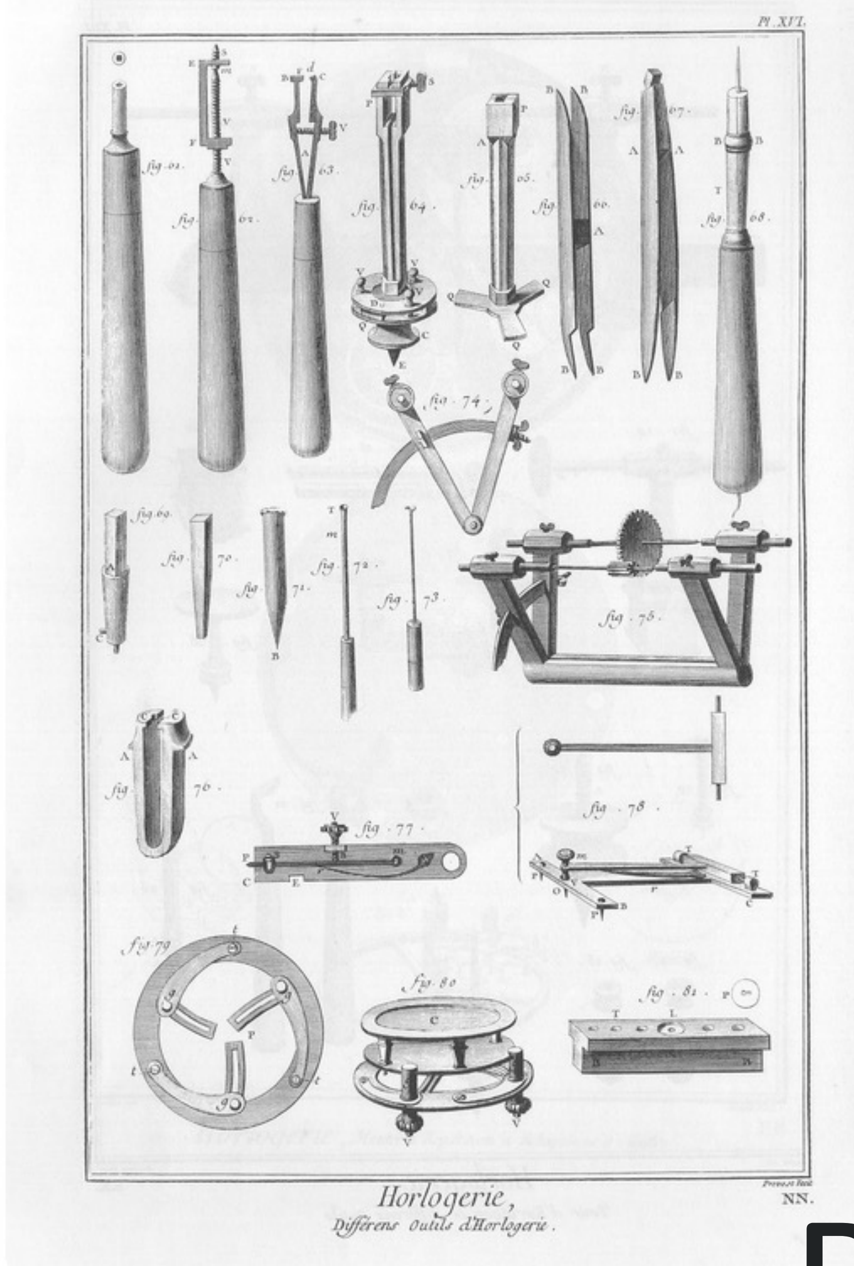
ML Fundamentals

Instituto Balseiro
23/09/2022

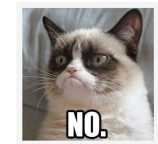


Lecture 11
Best practices

Announcements

- Hoy es la última clase 💪 vean el discussions!
- Guía 04 online, recomentamos hacerla 👉
- Clase 10 está demorada por glitch del audio
- Antes del café hablamos del P2
- El colectivo negro
- Notas del P1 en breve
  - Varios deben todavía el **error de test**

P1 – avoiding some pitfalls

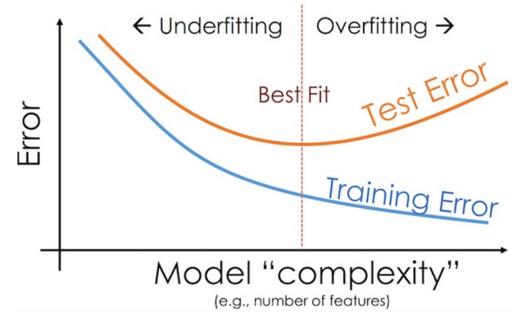
- Not comparing traning and testing errors 

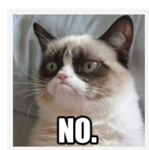
sklearn.model_selection.GridSearchCV

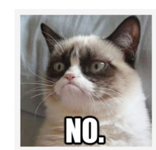
class `sklearn.model_selection.GridSearchCV`(*estimator, param_grid, *, scoring=None, n_jobs=None, refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False*) [source]

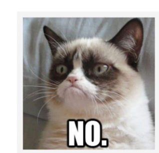
return_train_score : *bool, default=False*

If `False`, the `cv_results_` attribute will not include training scores. Computing training scores is used to get insights on how different parameter settings impact the overfitting/underfitting trade-off. However computing the scores on the training set can be computationally expensive and is not strictly required to select the parameters that yield the best generalization performance.



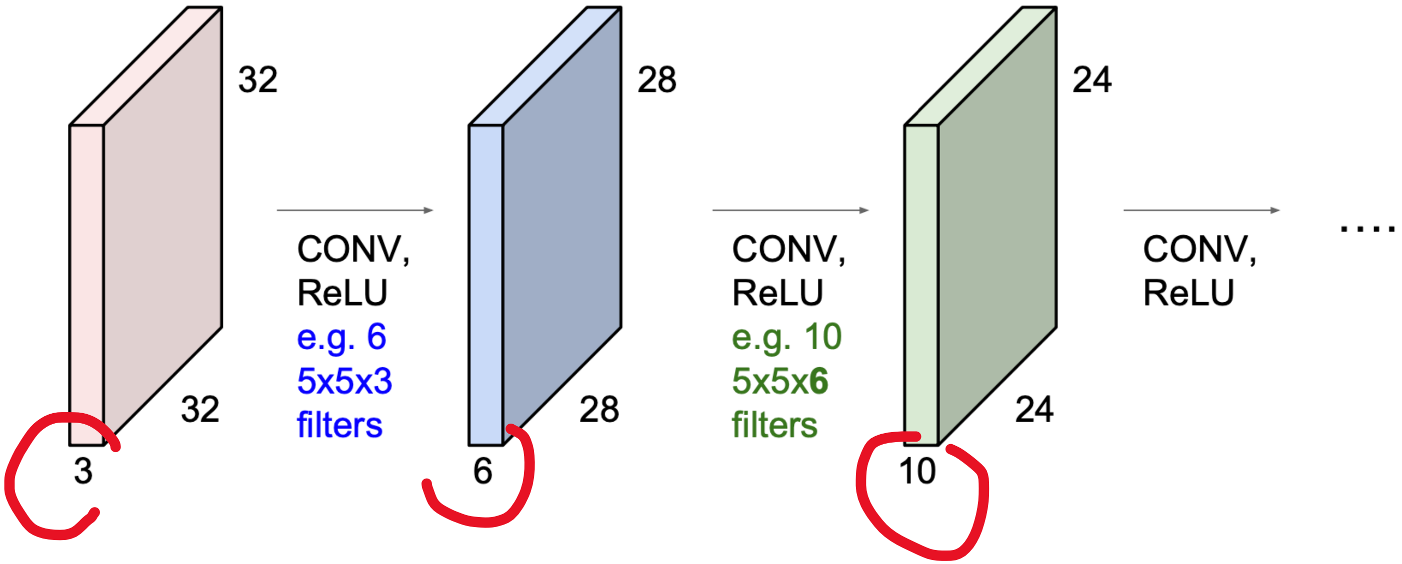
- Use test to train 

- Trust accuracy on imbalanced data 

- Trust correlations among one-hot features 

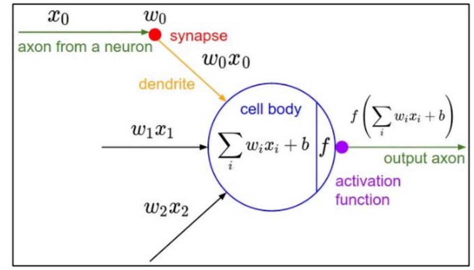
La pregunta de Pablo

Las capas sucesivas no deberían tener la misma profundidad? Las capas siempre aumentan su profundidad?

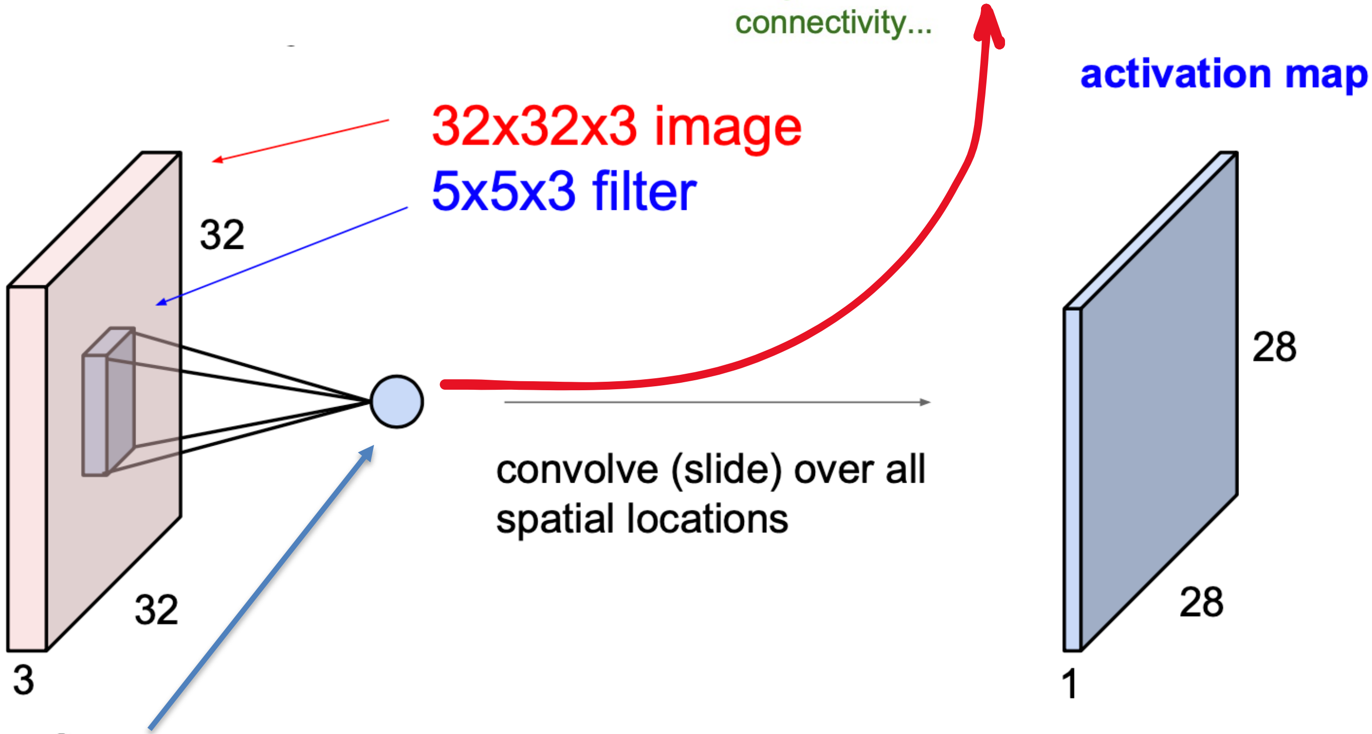


Convolution layer

Image

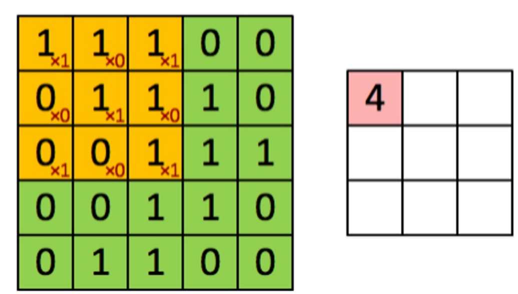


It's just a neuron with local connectivity...



1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

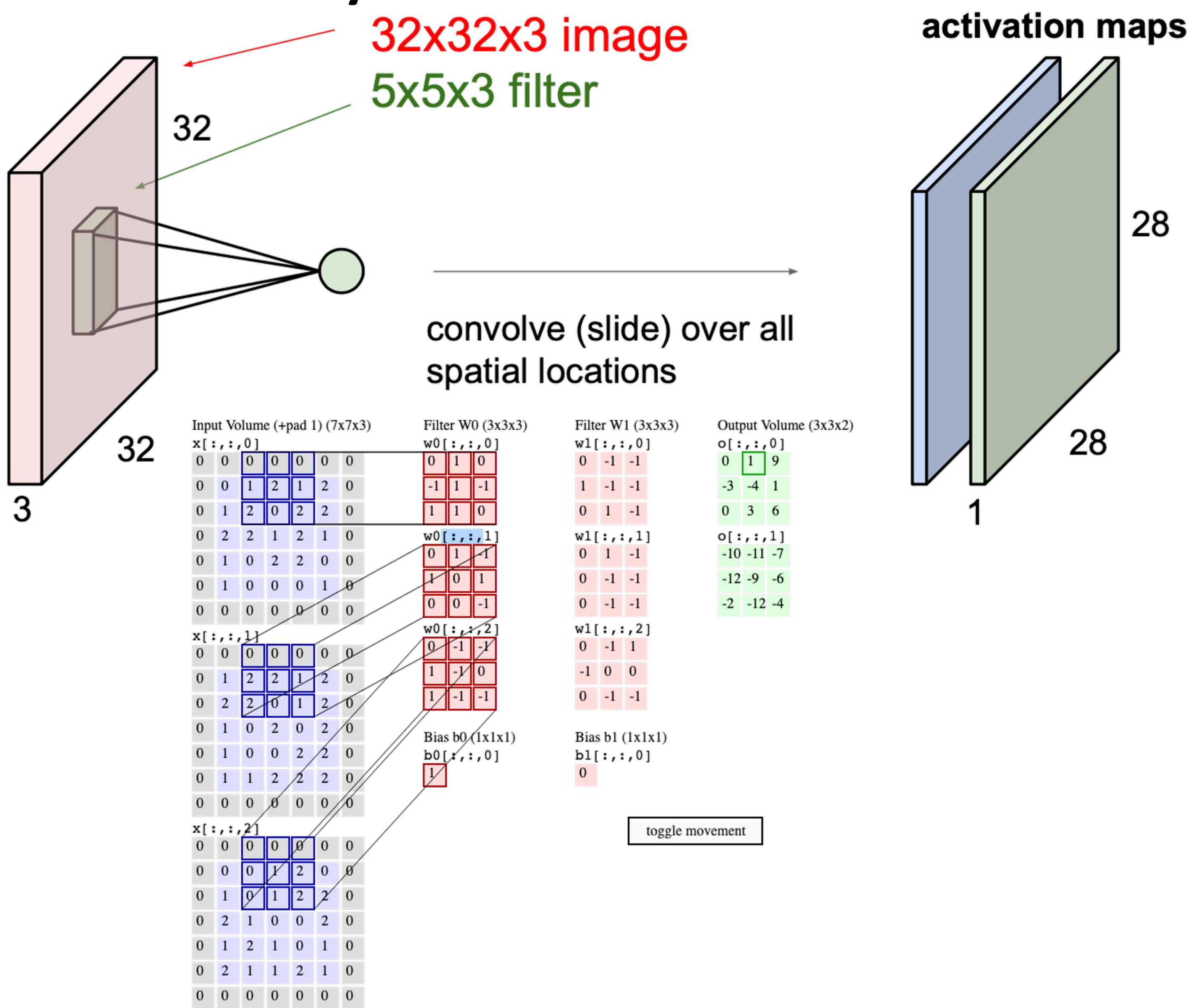
$$w^T x + b$$



CS231n 2016 - Prof.Fei-FeiLi & Andrej Karpathy & Justin Johnson

Luis G. Moyano - Fundamentos de ML Instituto Bakseiro

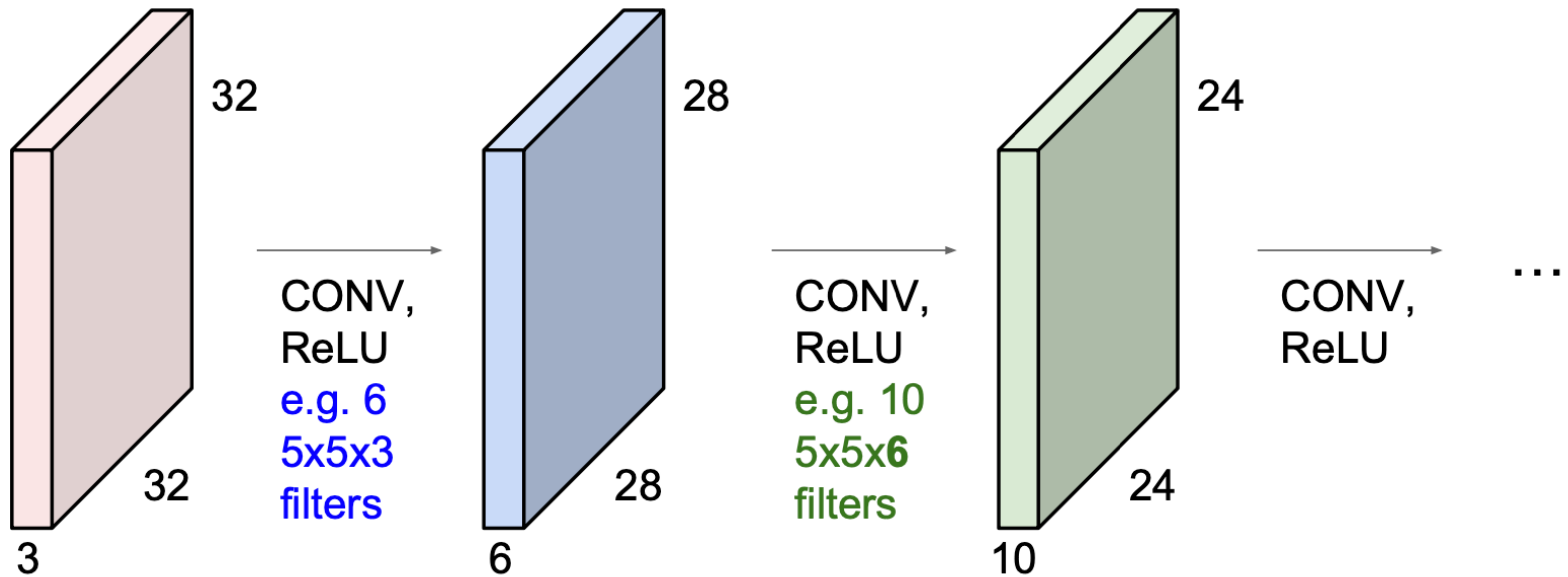
Convolution layer

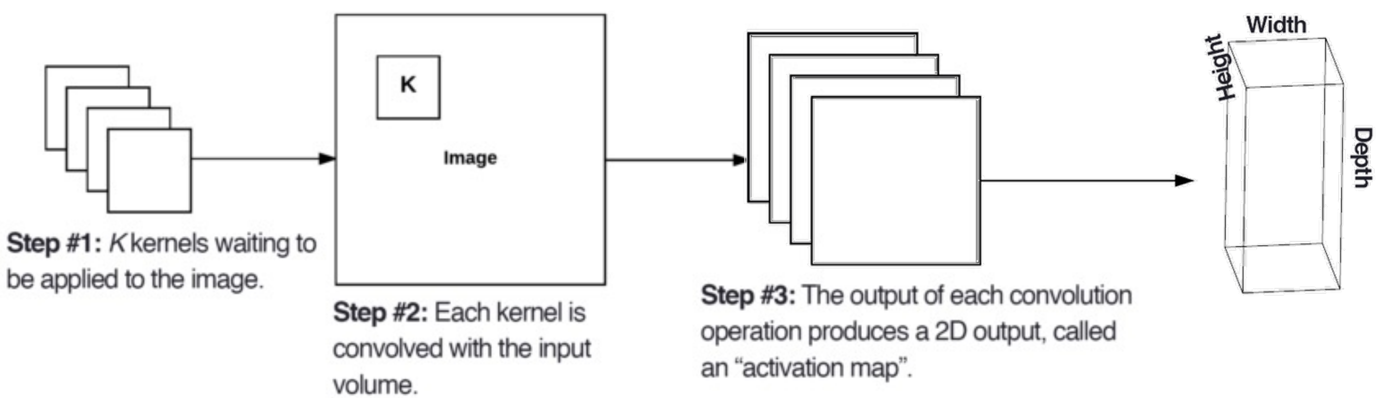


Luis G. Moyano - Fundamentos de ML Instituto Bakseiro

Convnet

ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



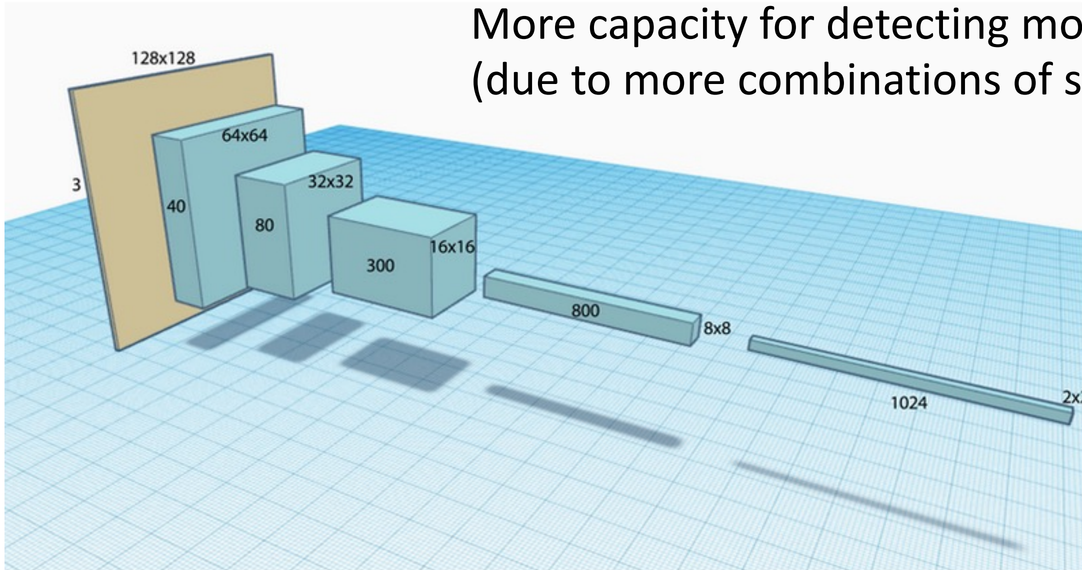


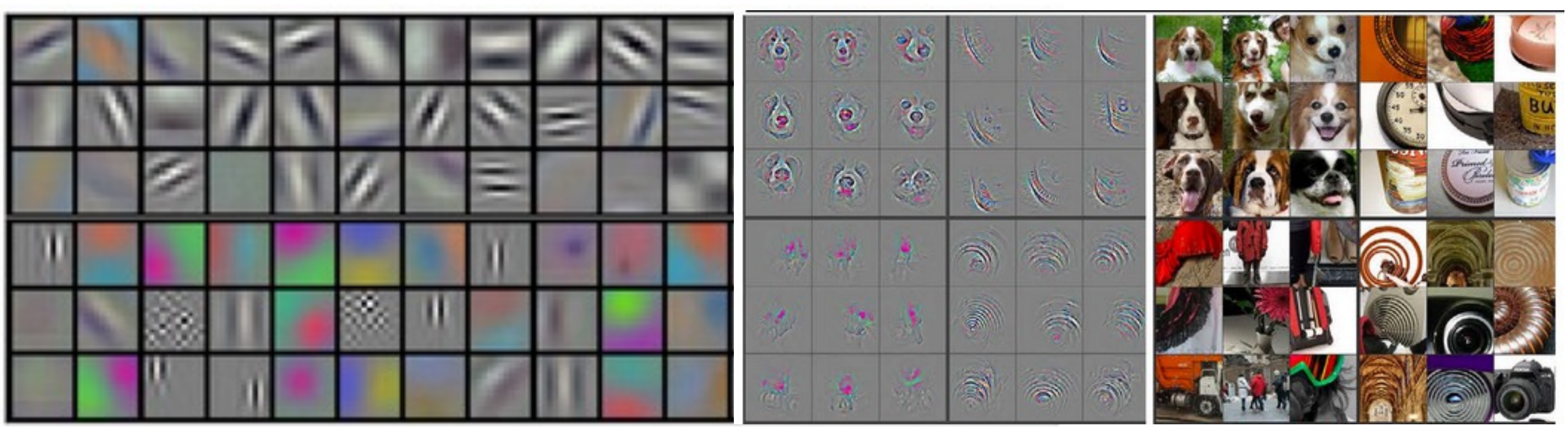
CS231n 2016 - Prof.Fei-FeiLi & Andrej Karpathy & Justin Johnson

Zieler & Fergus 2013

Its common for the number of kernels to grow

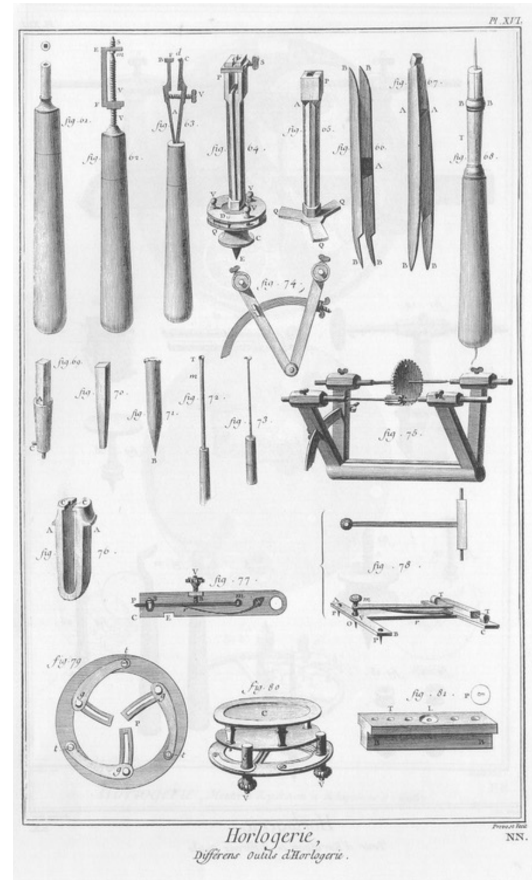
More capacity for detecting more complex patterns
(due to more combinations of simpler ones)



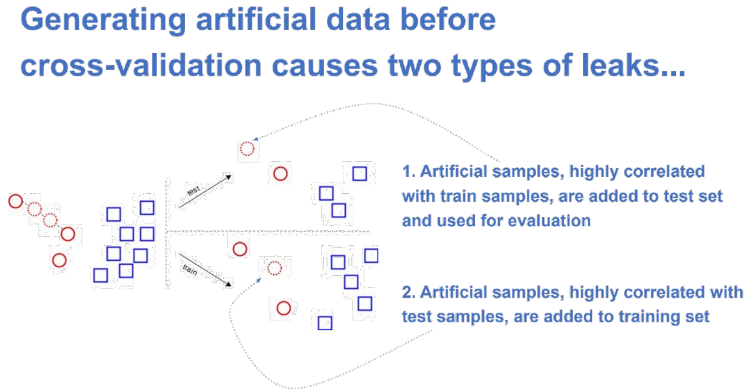


ML Fundamentals – Lecture 11

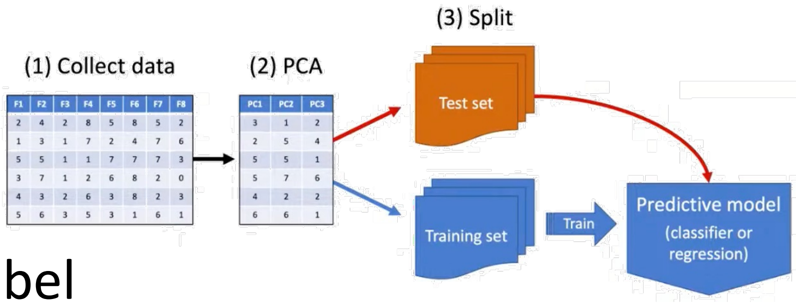
- Best practices
  - Always be skeptical
  - Data leakage
  - Hastie's CV done right
  - Karpathy's workflow
    - EDA + dumb baselines
    - Overfit & regularize
    - Tune & squeeze



General tips



- Always be skeptical of results that seem too good to be true.
- Applying ML (or DL) when is not the best for the job
- Insufficient data
  - beware of unbalanced data



- Data leakage:
  - feature leakage, be label or proxy label
    - e.g. MonthlySalary when predicting YearlySalary
  - training example leakage
    - time series, group splitting

Tip: use sklearn Pipelines (to assemble several steps that can be cross-validated together)

Check 2022 – Kapoor - Leakage and the Reproducibility Crisis in ML-based Science

The Wrong and Right Way to Do CV

2013 Hastie - section 7.10.2

1. Screen the features in the data: find a subset of “good” features that show fairly strong (univariate) correlation with the class labels
2. Using just this subset of predictors, build a multivariate classifier.
3. Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

The Wrong and Right Way to Do CV

2013 Hastie - section 7.10.2

1. Screen the features **in the data**: find a subset of “good” features that show fairly strong (univariate) correlation with the class labels
2. Using just this subset of predictors, build a multivariate classifier.
3. Use cross-validation to estimate the unknown tuning parameters and to estimate the prediction error of the final model.

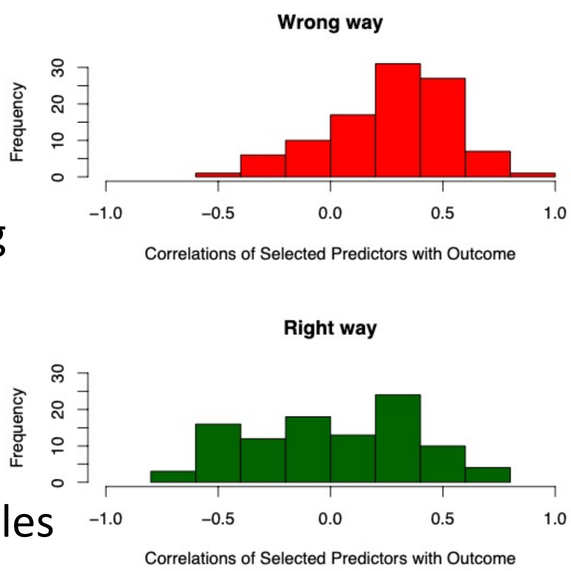
Here is the **correct way** to carry out cross-validation in this example:

1. Divide the samples into K cross-validation folds (groups) at random, and, for each fold k = 1,2,...,K:

a) Find a subset of “good” predictors that show fairly strong (uni- variate) correlation with the class labels, using all of the samples except those in fold k.

(b) Using just this subset of predictors, build a multivariate classi- fier, using all of the samples except those in fold k.

(c) Use the classifier to predict the class labels for the samples in fold k.



How to avoid machine learning pitfalls: a guide for academic researchers

[2022 – Lones - How to avoid machine learning pitfalls: a guide for academic]

[Sloppy Use of Machine Learning Is Causing a 'Reproducibility Crisis' in Science]

A Recipe for Training Neural Networks (2019)

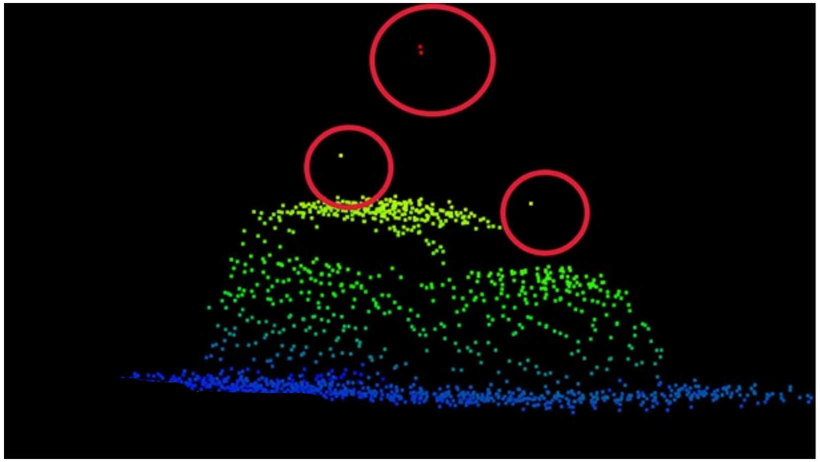
Karpathy's tips

1. Become one with the data
2. Set up the end-to-end training/evaluation skeleton + get dumb baselines
3. Overfit
4. Regularize
5. Tune
6. Squeeze out the juice

A "fast and furious" approach to training neural networks does not work and only leads to suffering.

Become one with the data

- Begin by not touching any neural net code
- Thoroughly inspect your data. Look for:
  - patterns, distributions
  - duplicates
  - mistakes
  - imbalance
  - bias
  - noise, variation
  - outliers (which in turn might point to bugs)



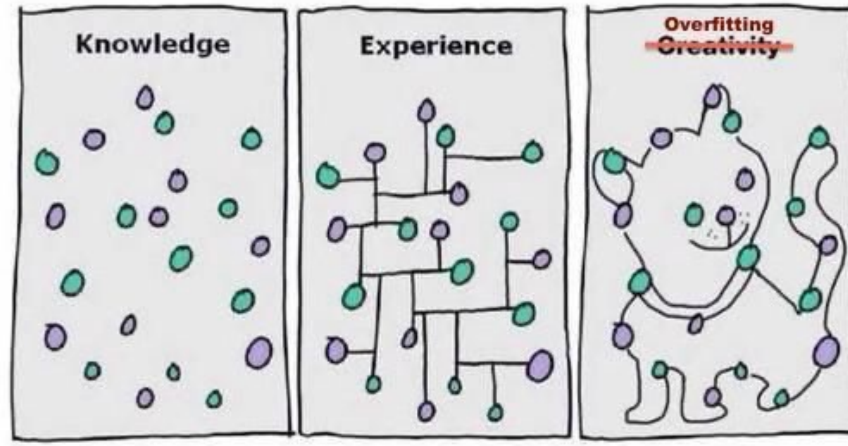
Set up an end-to-end ML skeleton + get dumb baselines

- pick some simple model e.g. a linear classifier, or a very tiny ConvNet: train/val/test + tweak
- get human baseline
- input-indepent baseline (zeros)

- fix random seed
- simplify
- verify loss at init
- initialize well: choose bias such as to facilitate convergence
- overfit one batch and check if loss is zero
- (see article for more)

Overfit

If no low error rate with any model, it may indicate some issues, bugs, or misconfiguration.

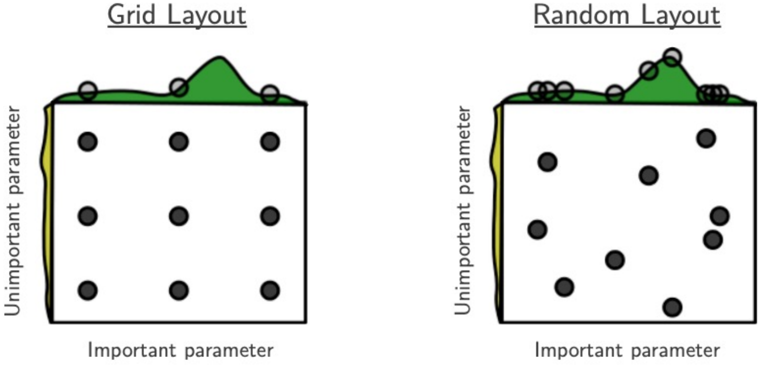


- Don't be a hero: find the most related paper and copy paste their simplest architecture
- Adam (optimizer) is safe. Start with learning rate of 3e-4.
- Complexify only one at a time
- Do not trust learning rate decay defaults (data size/epoch dependant).
  – may be disable and use constant learning rate

Regularize & Tune

Gain some validation accuracy by giving up some of the training accuracy.

- get more data
- data augment
- creative augmentation (e.g. simulation)
- pretrain
- smaller input dimensionality
- smaller model size
- decrease the batch size
- add dropout
- weight decay (l1, l2, KL regularizations)
- early stopping
- random over grid search
- other hyper-parameter optimization (Bayesian)



2012 – Bergstra- Random Search for Hyper-Parameter Optimization

Squeeze out the juice

- ensembles: "Model ensembles are a pretty much guaranteed way to gain 2% of accuracy on anything."
- leave it training: "One time I accidentally left a model training during the winter break and when I got back in January it was SOTA (“state of the art”)."

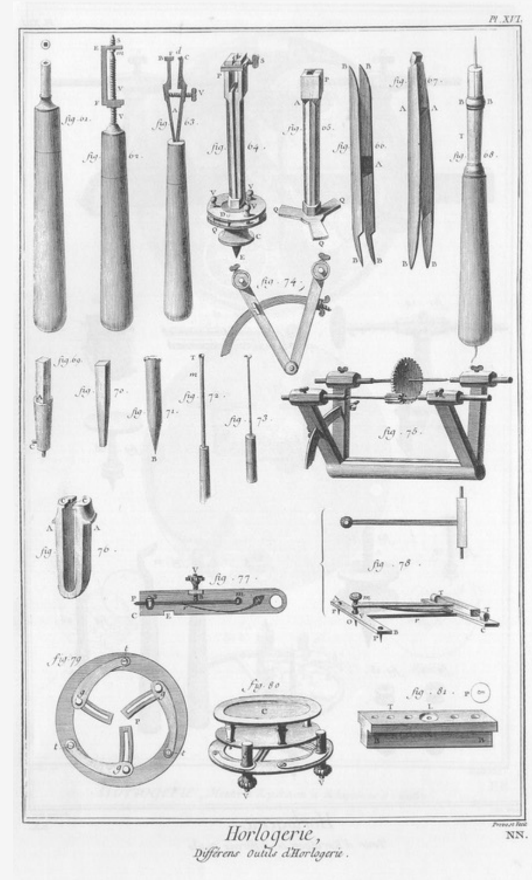
Announcements

Sobre el P2:

- No vamos a pedir póster, por impracticable.
- Vamos a pedir **ejercicio resuelto + video corto.**
- Después, **examen oral** corto, sobre lo hecho.
- Van a tener 7 días, y es un 'hard' deadline.
  - Prevengan bloopers logísticos 💥
  - Se puede hacer resubmission en Classroom, úsenlo

ML Fundamentals – Lecture 11

- Best practices
  - Always be skeptical
  - Data leakage
  - Hastie's CV done right
  - Karpathy's workflow
    - EDA + dumb baselines
    - Overfit & regularize
    - Tune & squeeze



Next:

P2 Final report & exam

Coming up:

EDGE IMPULSE

TINY ML