CEFET/RJ

CBPF

Deep Learning Applications in Astronomy

Clécio R. Bom

clearnightsrthebest.com

debom@cbpf.br

February 12, 2018

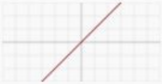
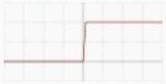
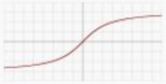
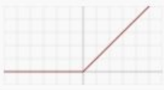
Why Conv layers are so cool?
Why it took so long to be in 'fashion'?

It's all about the way they are connected, locally.

Sparsity, specialization of certain areas, redundancy.

Optimizing millions of parameters requires nice computing resources...

Activation Functions

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Do not take advantage of Neural Nets for non linearities estimation. Useful in regression problems since is unbounded.

Hard to train, derivative vanishes.

Easily differentiable. In the last layers can be associated with probability.

Similar results as in sigmoid activations in intermediate layers. However, is numerically faster.

Tentative to avoid the vanishing of the ReLu derivative.

Adapted from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Why Sigmoid for classification?

Consider two classes, $y \in \{0, 1\}$.

The conditional probability of class $P(y|z(x))$ where $z = \omega^T h(x) + b$ the output of a set of neurons with x inputs.

Why Sigmoid for classification?

the unnormalized log probability can be written as

$$log\ \widehat{P}(y=1|z)=z \quad \text{(neurons “on”)}$$
$$log\ \widehat{P}(y=0|z)=0 \quad \text{(neurons “off”)}$$

$$\widehat{P}(y=1|z)=exp(z)$$

$$\widehat{P}(y=0|z)=exp(0)=1$$

Why Sigmoid for classification?

The Normalized version:

$$P(y=1|z) = \frac{exp(z)}{1+exp(z)}$$

$$P(y=0|z) = \frac{1}{1+exp(z)} \quad .$$

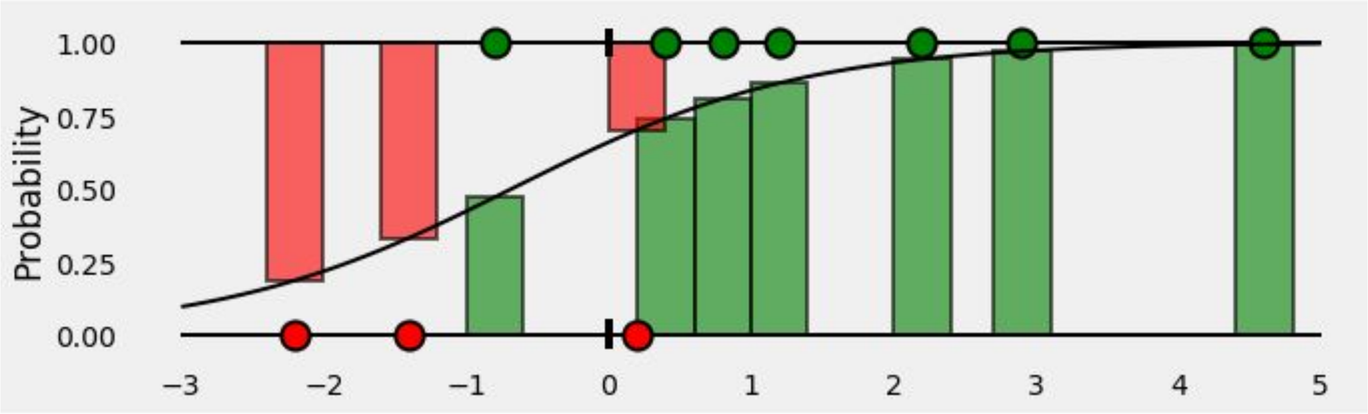
This is

$$P(y=1|z) = \frac{exp(z)}{1+exp(z)} = \frac{1}{\frac{exp(z)+1}{exp(z)}} = \frac{1}{1+exp(-z)} = \sigma(z)$$

$$P(y=0|z) = \sigma(-z) \, .$$

Some Loss intuition...

Consider the binary classification of red and greens. The True class probability of a set of z points is:

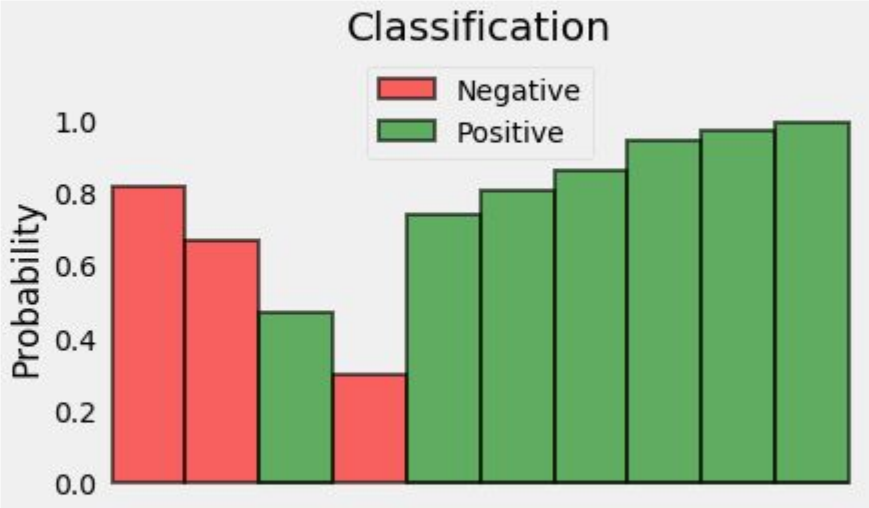


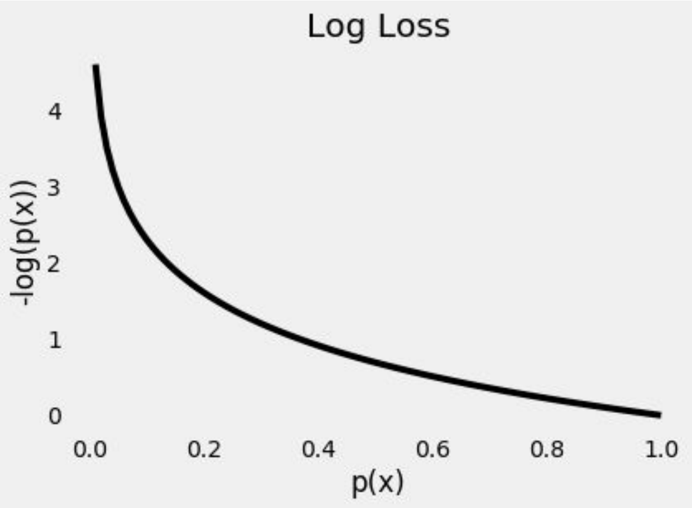
$$P(y = 0|z) = \sigma(-z)$$

$$P(y = 1|z) = \sigma(z)$$

Example adapted from : https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

Some Loss intuition...





If the predicted probability of the **true class** gets **closer to zero**, the **-log(p(x)) increases exponentially**.

Some Loss intuition…

That is the behaviour we may require in a loss function like Cross- Entropy

$$H(p,q) = -\sum_i p_i log q_i = -y log\hat{y} - (1-y) log(1-\hat{y})$$

Where p is true probability distribution of the true labels and q is the predicted probability of the predicted labels.

In our binary green-red example, for a green point (y=1), it adds log($\hat{y}$) to the loss, that is, the log predicted probability of it being green. For a red point (y=0) it adds log(1-$\hat{y}$), that is, the predicted log probability of it.

$$H(p,q) \geq H(p,p)$$

Cross-Entropy V.S. L2 (RMSE)

Penalizes small probabilities like p=0.1 and q=0.05 in contrast to p=0.80 and q=0.81 so fractional error are important.

If q=0 log(q) diverges .

Also RMSE is usually be more suitable in regression problems.

For the L2 Loss:

$$\frac{\partial L}{\partial \omega_i} \sim \frac{1}{N}(\omega_i o_i - y_i)(o_i)$$

So if one has a classification problem we would expect the desired output oi to be 1 or 0 as the it approaches this results the derivative would be very small and, therefore, the update in the weights.

What Metrics in my deep learning model is for?

Common Classification Metrics

Binary Accuracy: binary_accuracy, acc

```
K.mean(K.equal(y_true, K.round(y_pred)))
```

Categorical Accuracy: categorical_accuracy,

```
K.mean(K.equal(K.argmax(y_true, axis=-1), K.argmax(y_pred, axis=-1)))
```

Common Regression Metrics

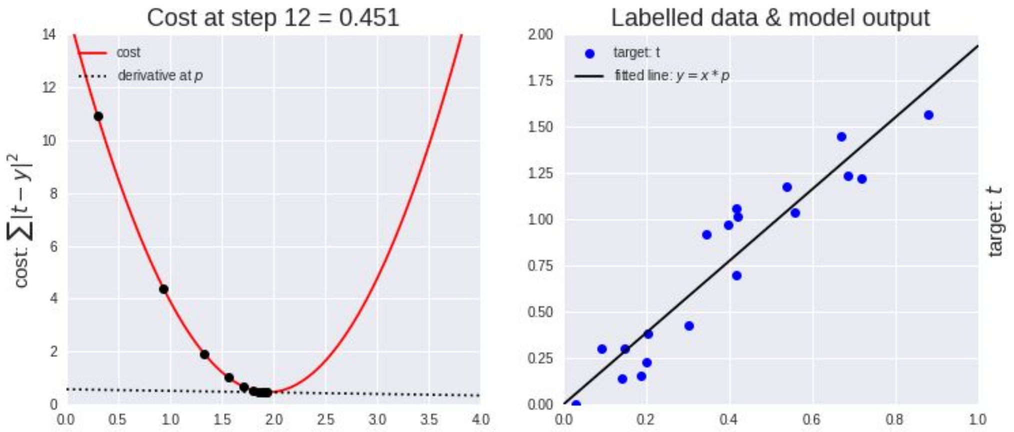
Mean Squared Error: mean_squared_error, MSE or mse
Mean Absolute Error: mean_absolute_error, MAE, mae

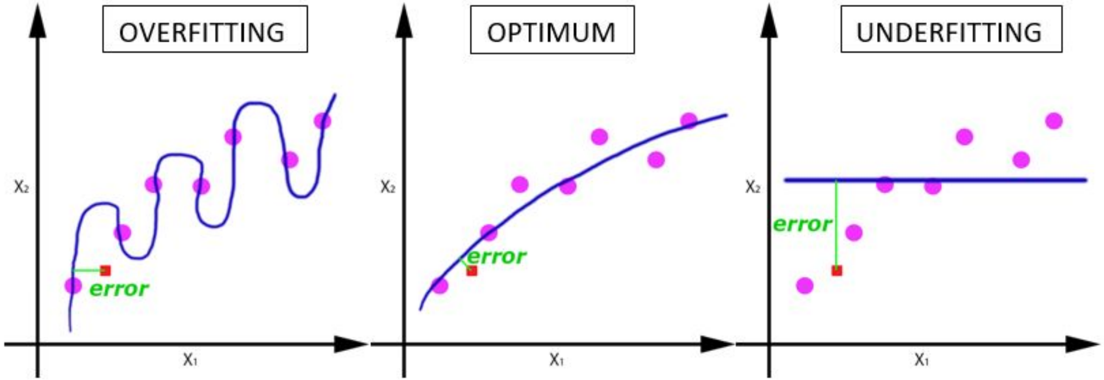
Common issues ...

Vanishing gradients—In case of deep networks, for any activation function, *abs(dW)* will get smaller and smaller as we go backwards with every layer during back propagation. The earlier layers are the slowest to train in such a case.

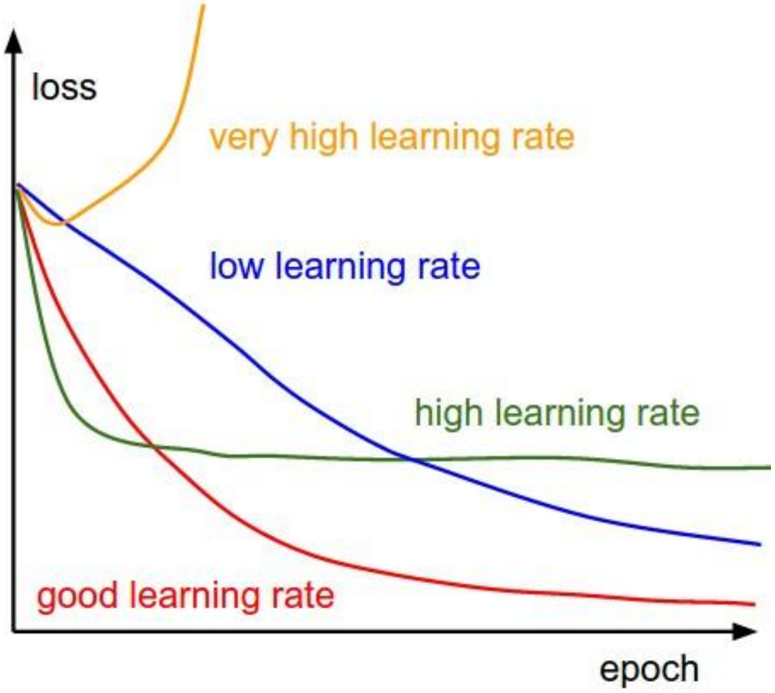
Exploding gradients—This is the exact opposite of vanishing gradients. When these weights are multiplied along the layers, they cause a large change in the cost. Thus, the gradients are also going to be large. This means that the changes in W, by W—α * dW, will be in huge steps, the downward moment will increase.
This may result in oscillating around the minima or even overshooting the optimum again and again and the model will never learn!

Most Important Piece of Advice against overfitting : Keep it Simple!



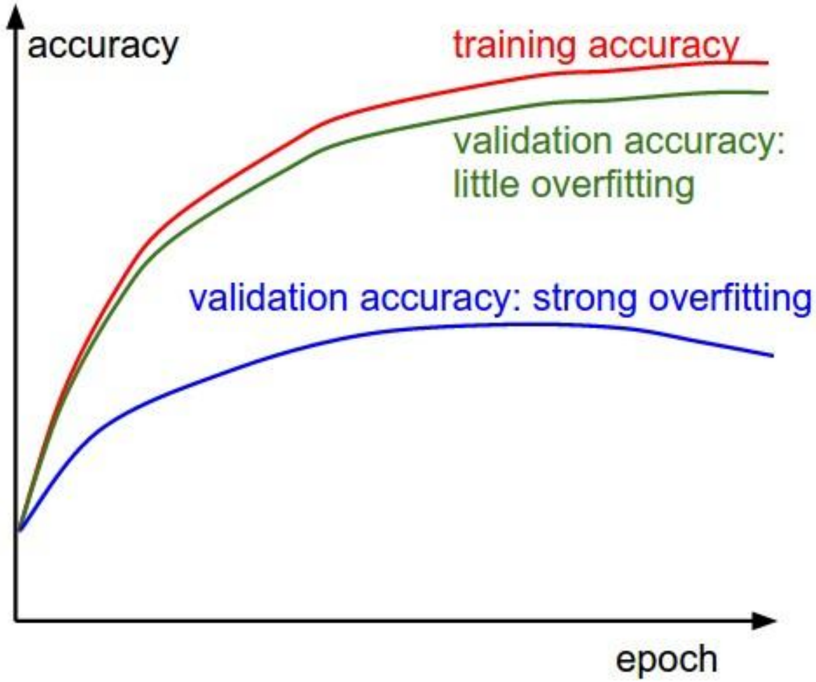


Some intuitions on your LOSS



Source:

Overfitting



Source:

Evaluating your model

True Positive (TP): • Reality: A wolf threatened. • Shepherd said: "Wolf." • Outcome: Shepherd is a hero.	**False Positive (FP):** • Reality: No wolf threatened. • Shepherd said: "Wolf." • Outcome: Villagers are angry at shepherd for waking them up.
False Negative (FN): • Reality: A wolf threatened. • Shepherd said: "No wolf." • Outcome: The wolf ate all the sheep.	**True Negative (TN):** • Reality: No wolf threatened. • Shepherd said: "No wolf." • Outcome: Everyone is fine.

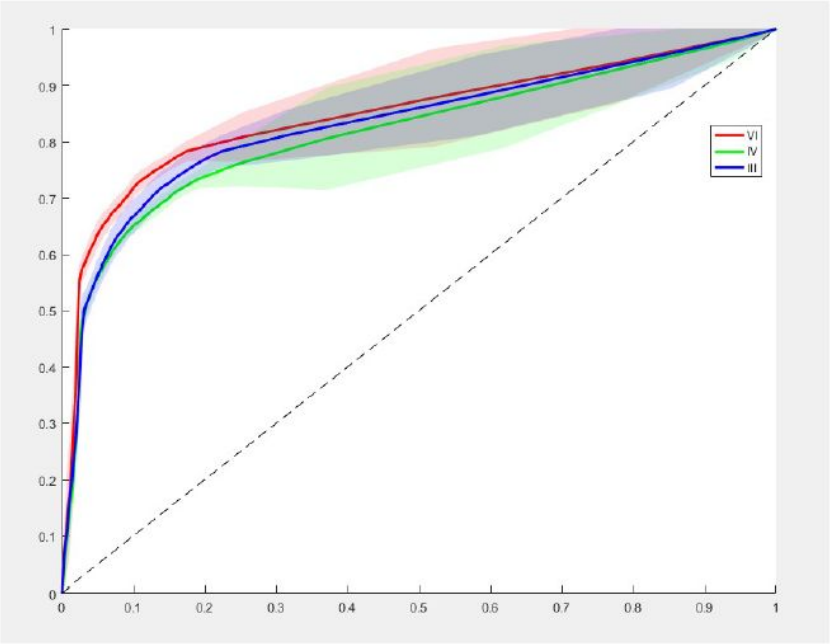
Source: https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative

Receiver Operating Characteristic

True positive = correctly identified
False positive = incorrectly identified
True negative = correctly rejected
False negative = incorrectly rejected

E.g. 3 Classes

Class 1 vs classes 2&3
Class 2 vs classes 1&3
Class 3 vs classes 1&2

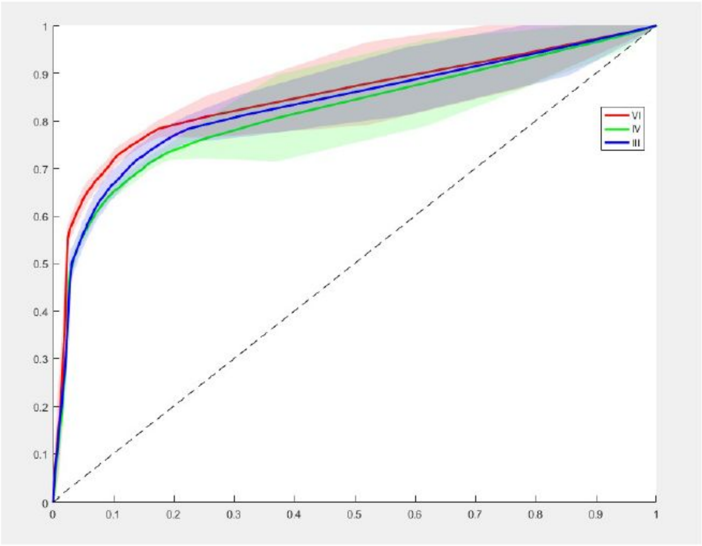


Cross- validation – K-fold validation

That is the gold standard!

Shuffle the dataset randomly.
Split the dataset into k sets
For each k set:
- Take the group as a hold out or test data set
- Take the remaining groups as a training data set
- Fit a model on the training set and evaluate it on the test set
- Save the evaluation scores in the test set

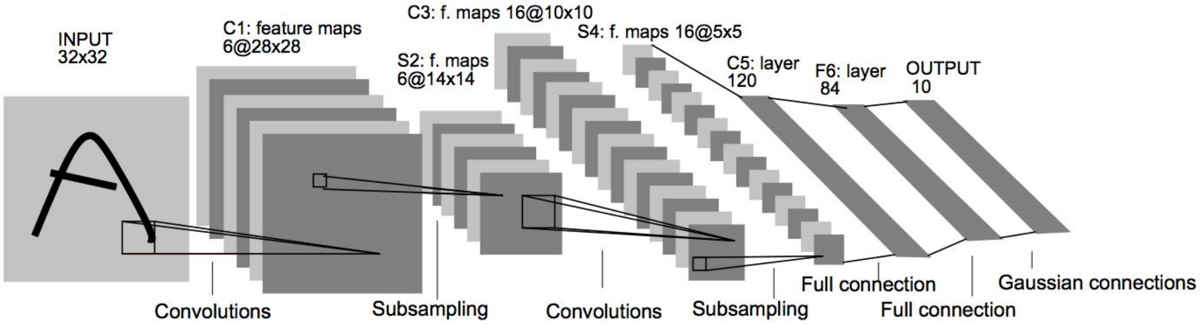
- Summarize the results by defining average (or median) and std on each threshold.



A little bit of historical Nets …

LeNet-5

Yann Lecun's LeNet-5 model was developed in 1998 to identify handwritten digits for zip code recognition in the postal service. This pioneering model largely introduced the convolutional neural network as we know it today.

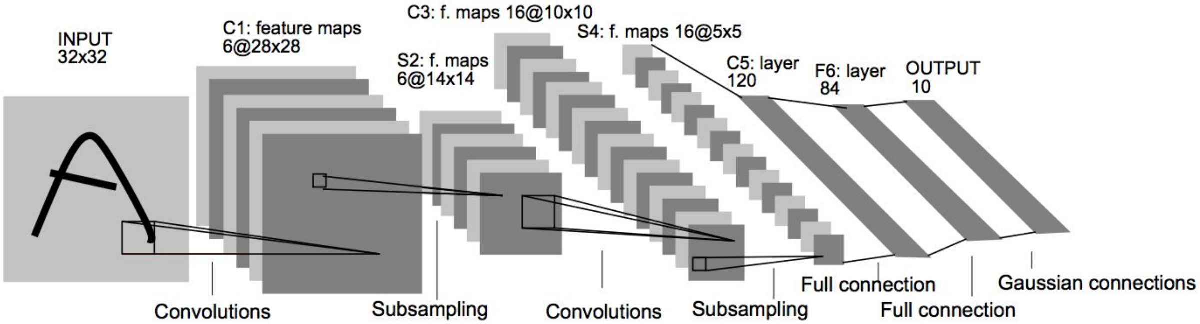


LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

A little bit of historical Nets …

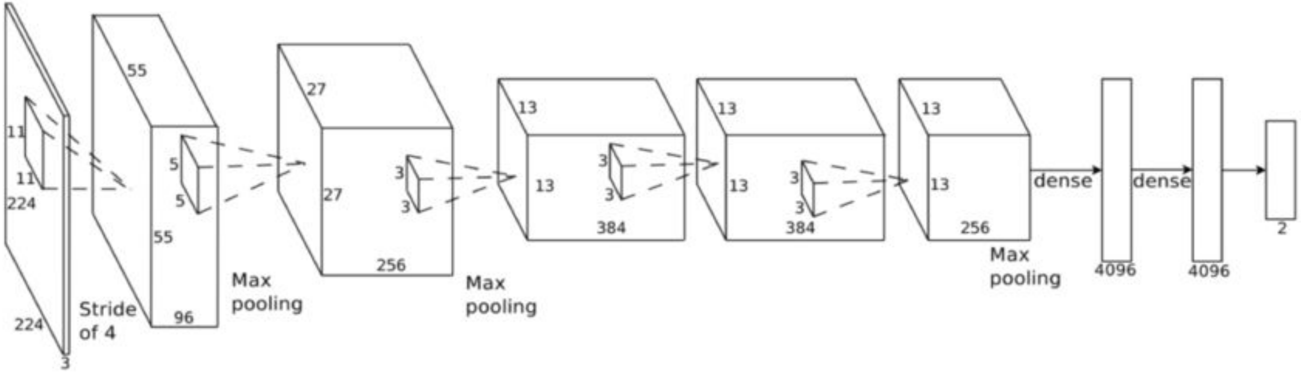
The subsampling layers use a form of average pooling.
Parameters: 60,000



LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

A little bit of historical Nets …

AlexNet was developed by Alex Krizhevsky et al. in 2012 to compete in the ImageNet competition. The general architecture is quite similar to LeNet-5, although this model is considerably larger. The success of this model (which took first place in the 2012 ImageNet competition) convinced a lot of the computer vision community to take a serious look at deep learning for computer vision tasks.

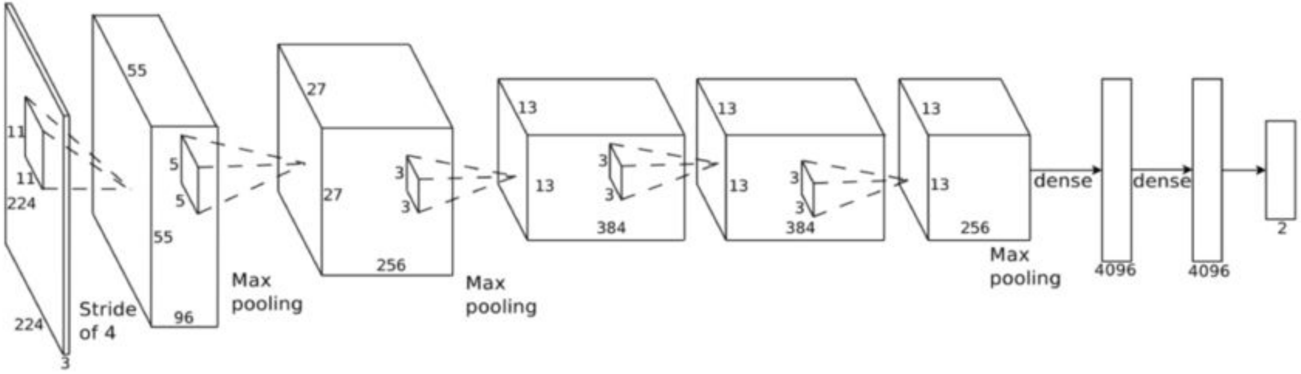


doi:10.1145/3065386

A little bit of historical Nets ...

AlexNet → 60 Million parameters!!!!

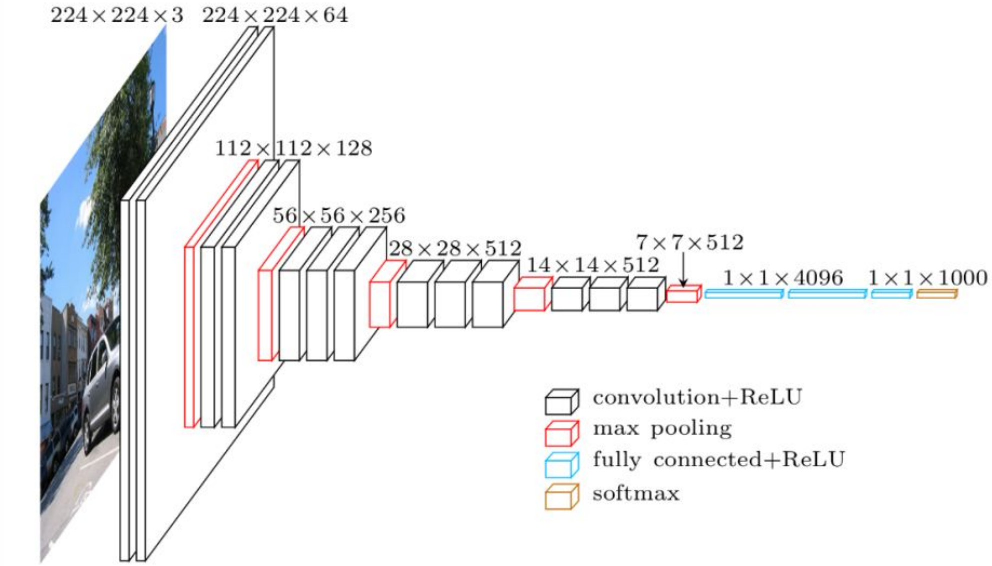
650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three globally-connected layers with a final 1000-way softmax. It was trained on two NVIDIA GPUs for about a week.



doi:10.1145/3065386

A little bit of historical Nets …

VGG-16 (2014)→ 138 Million parameters!!!!

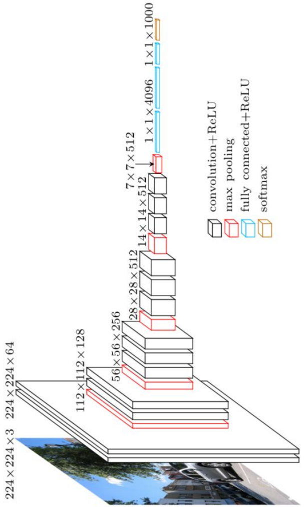


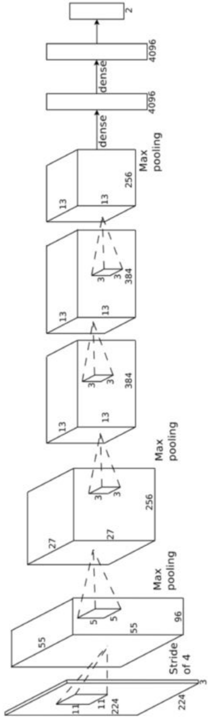
https://arxiv.org/abs/1409.1556

What can we do with such BFT (Best-Fitted and Trained?) models today?

Clustering!!!!! They do know how to extract features.
Unsupervised learning
Transfer Learning

Ben Kenobi - Old and Wise
lots of optimized parameters





Going Deeper with Convolutions - AKA - Inception paper

Bigger size typically means a larger number of parameters, which makes the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited. (…)The other drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep vision network, if two convolutional layers are chained, any uniform increase in the number of their filters results in a **quadratic increase of computation**. If the added capacity is used inefficiently (for example, **if most weights end up to be close to zero), then much of the computation is wasted**.(…).

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

storage.googleapis.com/pub-tools-public-publication-data/pdf/43022.pdf

Going Deeper with Convolutions - AKA - Inception paper

A fundamental way of solving both of these issues would be **to introduce sparsity and replace the fully connected layers by the sparse ones, even inside the convolutions.** Besides mimicking biological systems, this would also have the advantage of firmer theoretical underpinnings due to the groundbreaking work of Arora et al. [2]. **Their main result states that if the probability distribution of the dataset is representable by a large, very sparse deep neural network, then the optimal network topology can be constructed layer after layer by analyzing the correlation statistics of the preceding layer activations and clustering neurons with highly correlated outputs.**

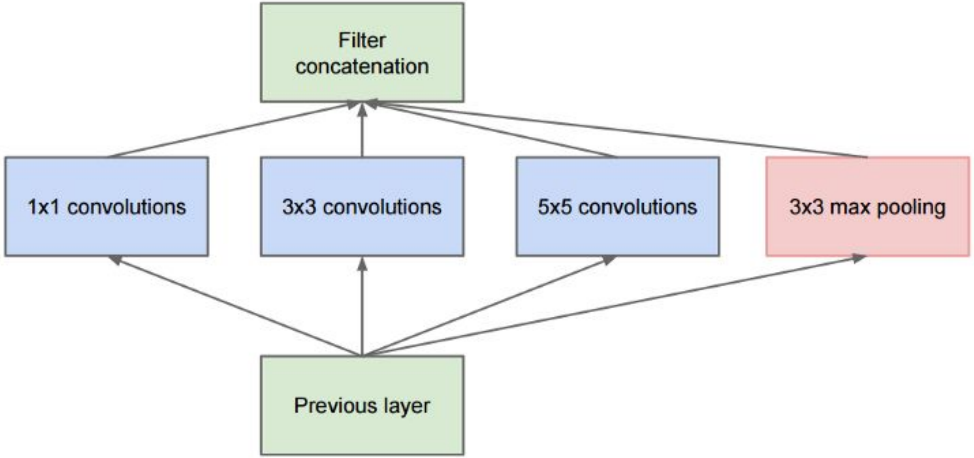
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

storage.googleapis.com/pub-tools-public-publication-data/pdf/43022.pdf

The Main (Inception) Idea...

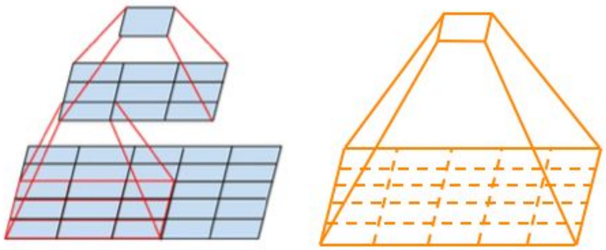
GoogLeNet (2014)

The idea is that you don't need to know in advance if it was better to do, for example, a 3×3 then a 5×5. Instead, just do all the convolutions and let the model pick what's best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions.



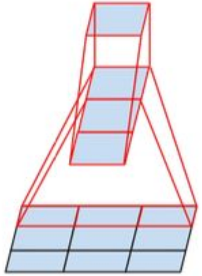
The Main (Inception) Idea...

GoogLeNet (2014)



two successive
3x3 convolutions

5x5 convolution

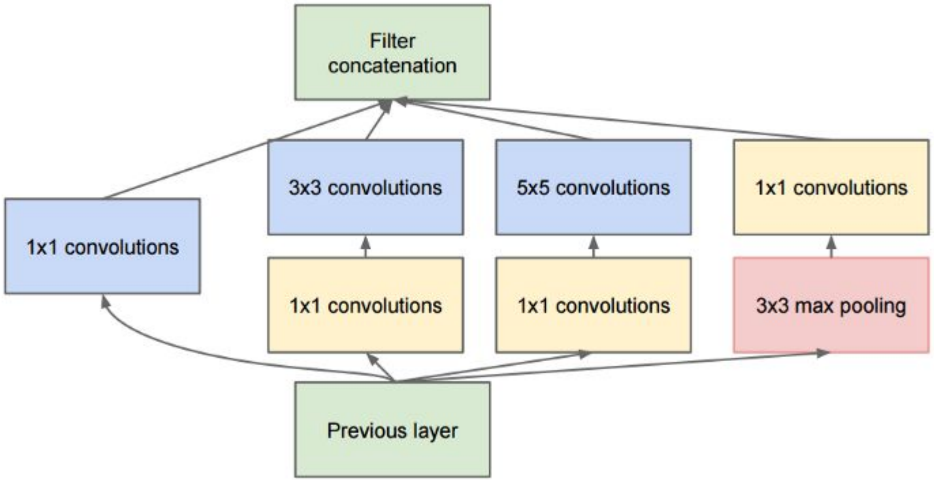


3x3 convolutions could be further deconstructed into successive 3x1 and 1x3 convolutions.

Generalizing this insight, we can more efficiently compute an n×n convolution as a 1×n convolution followed by a n×1 convolution..

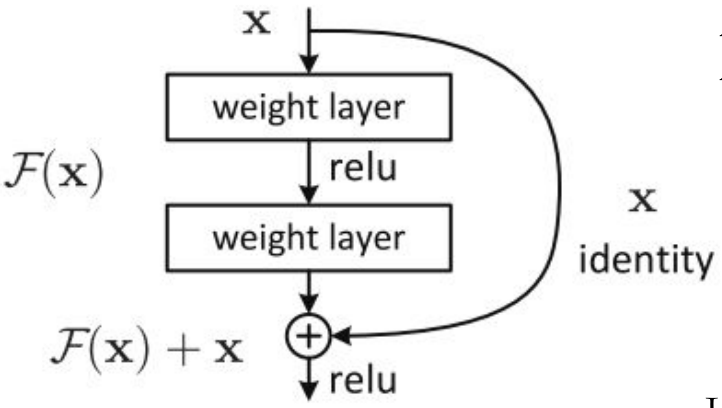
https://www.jeremyjordan.me/convnet-architectur

What to choose?



We still have millions of parameters to fit!!!! We still need some ideas to prevent overfitting

ResNet Block



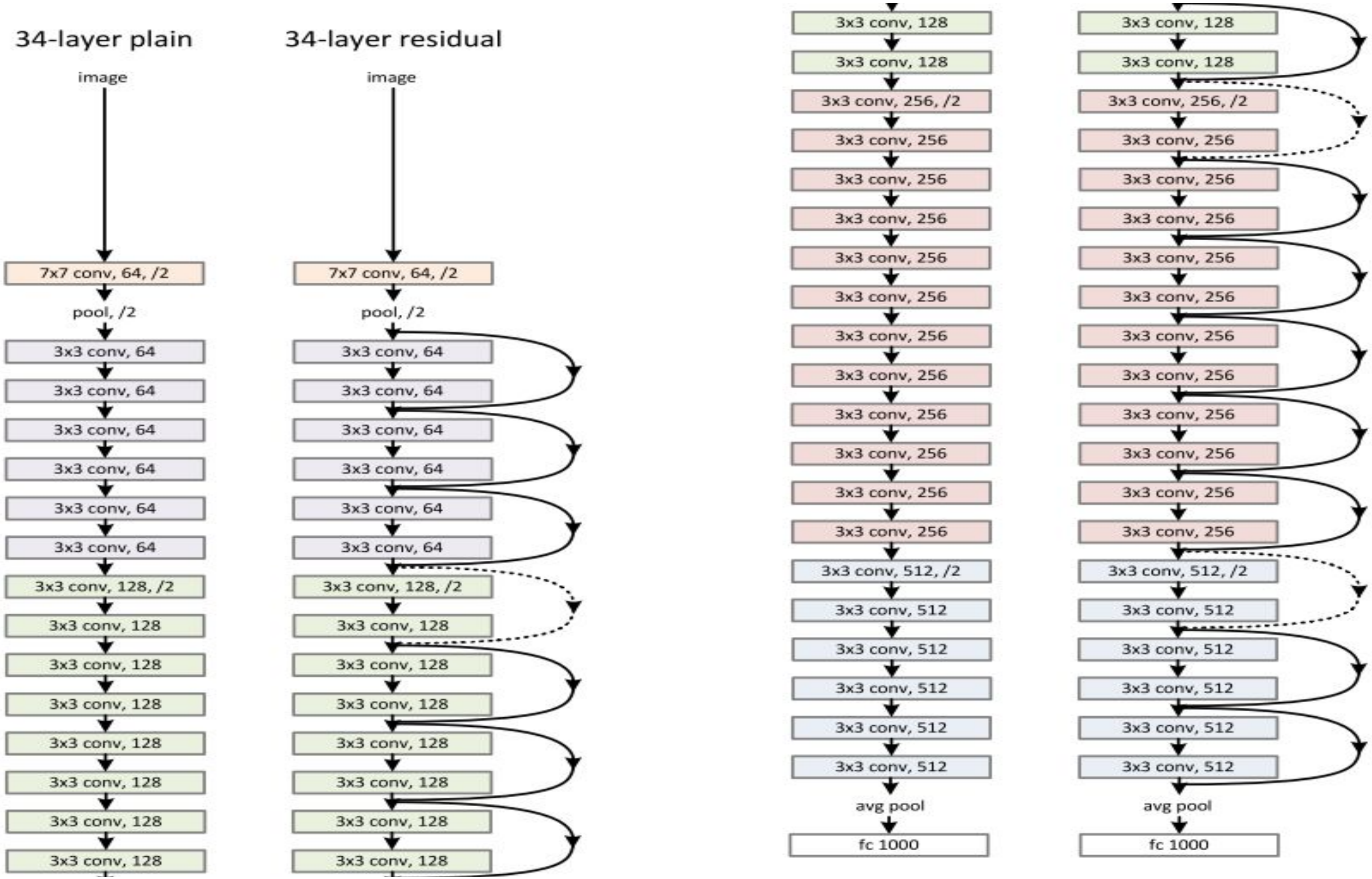
The author's hypothesis is that it is easy to optimize the residual mapping function F(x) than to optimize the original, unreferenced mapping .

H(x) = F(x)+x

If the identity mapping is optimal, We can easily push the residuals to zero (F(x) = 0) than to fit an identity mapping (x, input=output) by a stack of non-linear layers.

It also put a new light on the vanishing gradient problem…

Residual Neural Networks



Residual Neural Networks

- Won 1st place in the ILSVRC 2015 classification competition with top-5 error rate of 3.57% (An ensemble model)
- Won the 1st place in ILSVRC and COCO 2015 competition in ImageNet Detection, ImageNet localization, Coco detection and Coco segmentation.
- Replacing VGG-16 layers in Faster R-CNN with ResNet-101. They observed a relative improvements of 28%
- Efficiently trained networks with 100 layers and 1000 layers also.
- ResNet Network Converges faster compared to plain counterpart of it.
- Identity vs Projection shortcuts. Very small incremental gains using projection shortcuts in all the layers. So all ResNet blocks use only Identity shortcuts with Projections shortcuts used only when the dimensions changes.

How the other Nets were in ILSVRC 2015

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

What people are doing with this?

nature
International journal of science

Letter | Published: 30 August 2017

Fast automated analysis of strong gravitational lenses with convolutional neural networks

Yashar D. Hezaveh, Laurence Perreault Levasseur & Philip J. Marshall

Machine and Deep Learning Applied to Galaxy Morphology - A Complete Classification Catalog

P. H. Barchi,[1★] R. R. de Carvalho,[2] R. R. Rosa,[1] R. Sautter,[1] M. S
B. A. D. Marques,[4] E. Clua[4]
[1] *Lab for Computing and Applied Mathematics, National Institute for Space Research (INPE), S. J. dos Cam*
[2] *Astrophysics Division, National Institute for Space Research (INPE), S. J. dos Campos, 12245-970, Brazil*
[3] *Physics Department, Brandeis University, 415 South St, Waltham, MA 02453, USA*
[4] *Computing Institute, Federal Fluminense University (UFF), Niterói, 24220-900, Brazil*

MNRAS 000, 1–12 (2018) Preprint 30 July 2018 Compiled using MNRAS LATEX style file v3.0

Galaxy Morphology Classification with Deep Convolutional Neural Networks

Jia-Ming Dai,[1,2 ★] Jizhou Tong[1]
[1] *National Space Science Center, Chinese Academy of Sciences, Beijing 100190,China*
[2] *University of Chinese Academy of Sciences, Beijing 100049, China*

Machine and Deep Learning Applied to Galaxy Morphology - A Complete Classification Catalog

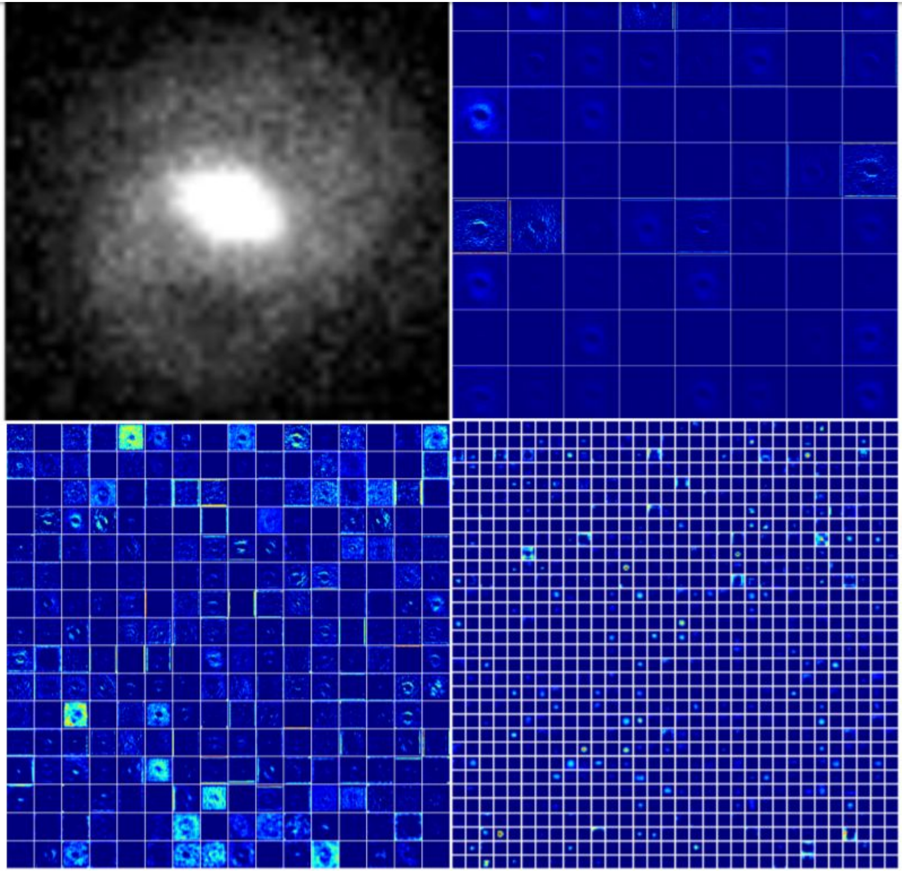


Figure 11. Example of convolutions applied to a galaxy. In the top left, the input image of a galaxy in r-band; in the top right tl output of the first convolution performed. Below, on the left, the ouput of the first Inception Module; on the right, the output of the last

- The samples are images in r-band in the SDSS-DR7 redshift range $0.03 < z < 0.1$, Petrosian magnitude in r-band brighter than 17.78 (spectroscopic magnitude limit)
- They tested a **ResNet** and **Inception**

$$OA = \frac{TP + TN}{TP + TN + FP + FN}$$

- K is the area of the galaxy's Petrosian ellipse divided by the area of the Full Width at Half Maximum (FWHM).

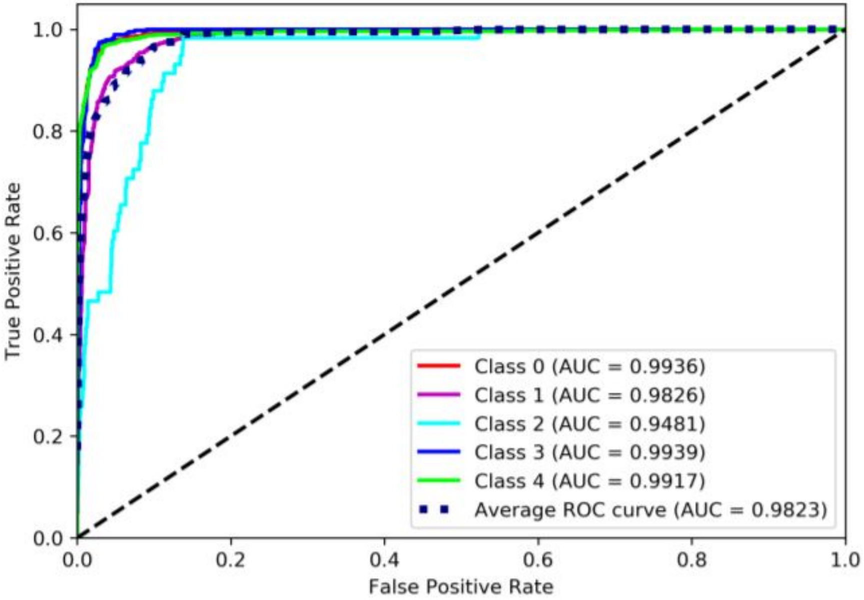
	$K \geq 5$			
	DT	**SVM**	**MLP**	**CNN**
11 classes	49.3	48.8	49.4	63.0
9 classes	60.9	63.2	63.0	70.2
7 classes	63.0	62.5	63.3	72.2
3 classes	71.9	71.2	71.2	80.8

Galaxy Morphology Classification with Deep Convolutional Neural Networks

Jia-Ming Dai,[1,2] ⋆ Jizhou Tong[1]

Figure 1. Example galaxy images from the dataset. Each row represents a class. From top to bottom, their Galaxy Zoo 2 labels are: completely round smooth, in-between smooth, cigar-shaped smooth, edge-on and spiral. They are referred to as 0, 1, 2, 3 and 4.

- The galaxy images in this study are drawn from Galaxy Zoo-the Galaxy Challenge 1 , which contain 61578 JPG color galaxy images with probabilities that each galaxy is classified into different morphologies.
- The authors tested a **ResNet**.



nature
International journal of science

Letter | Published: 30 August 2017

Fast automated analysis of strong gravitational lenses with convolutional neural networks

Yashar D. Hezaveh, Laurence Perreault Levasseur & Philip J. Marshall

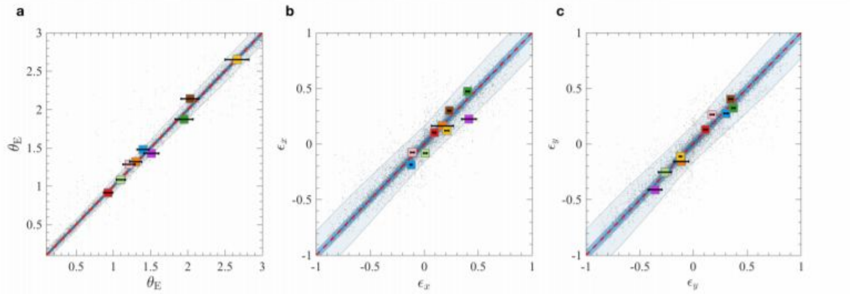


Figure 1: Comparison of parameters estimated using neural networks (on the y-axis) with their true values (x-axis). From left to right, the panels correspond to the Einstein radius and the $x-$ and $y-$ components of complex ellipticity. The shaded blue areas represent the 68, and 95% intervals of the recovered parameters on a test set that the network has not been trained on. The small gray dots show the parameters of all 10,000 test samples. The colored data points and their error bars (95% confidence) correspond to real *HST* images of gravitational lenses in SL2S sample, with the true parameters set to previously published values[17].

- This is a regression problem. The fundamental question is: Can we derive Strong lensing parameters without actually make the inverse modelling?
- The authors tested a **Inception-v411, AlexNet12, Overfeat13**
- The dataset is composed by Galaxy Zoo Challenge HST-like Images where they removed the lens galaxy using an ICA.

Network	θ_E	ϵ_x	ϵ_y	x	y
Network 1 (Inception)	0.03	0.04	0.05	0.06	0.06
Network 2 (AlexNet)	0.03	0.04	0.04	0.05	0.06
Network 3 (Overfeat)	0.04	0.05	0.05	0.06	0.06
Network 4	0.03	0.05	0.06	0.05	0.05
Combined Network	0.02	0.04	0.04	0.04	0.04

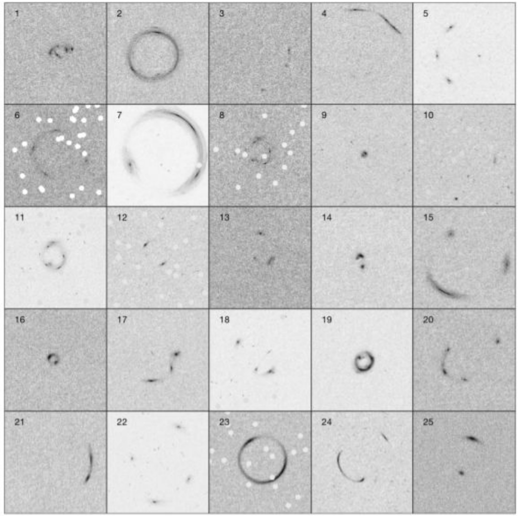
Table 1: **Errors of the individual and combined networks.** The columns present the 68% errors for the Einstein radius, θ_E, the two components of complex ellipticity (ϵ_x, ϵ_y) and the coordinates of the lensing galaxy (x, y). The angular parameters (θ_E, x, and y) are given in units of arc-seconds.

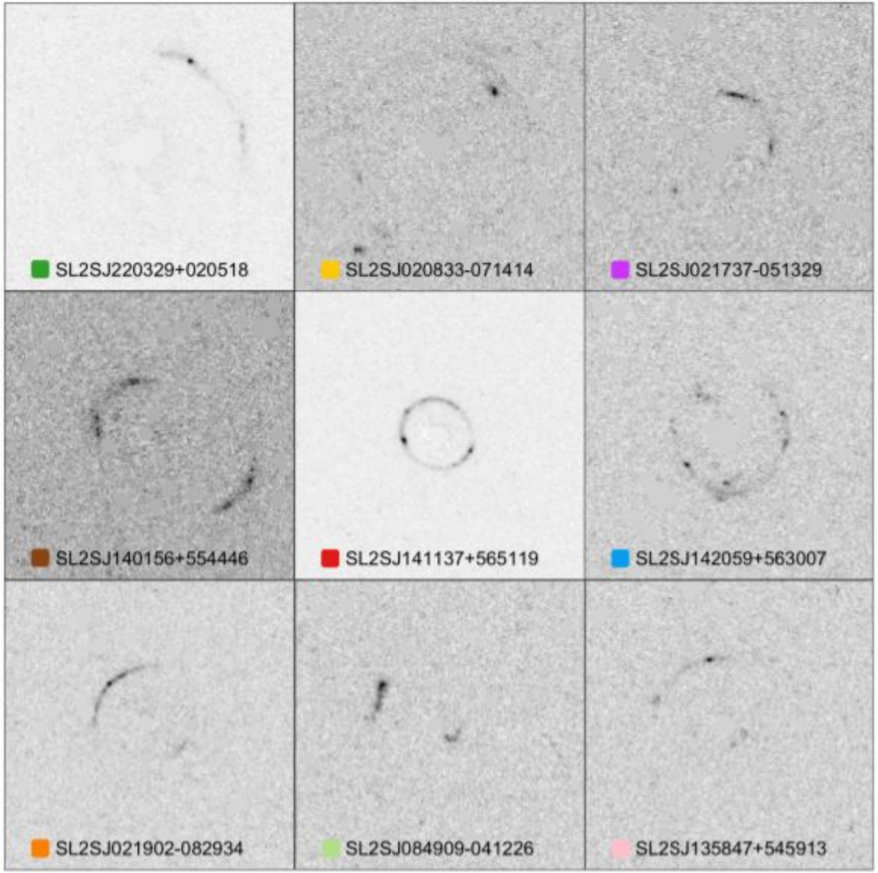
nature
International journal of science

Letter | Published: 30 August 2017

Fast automated analysis of strong gravitational lenses with convolutional neural networks

Yashar D. Hezaveh, Laurence Perreault Levasseur & Philip J. Marshall





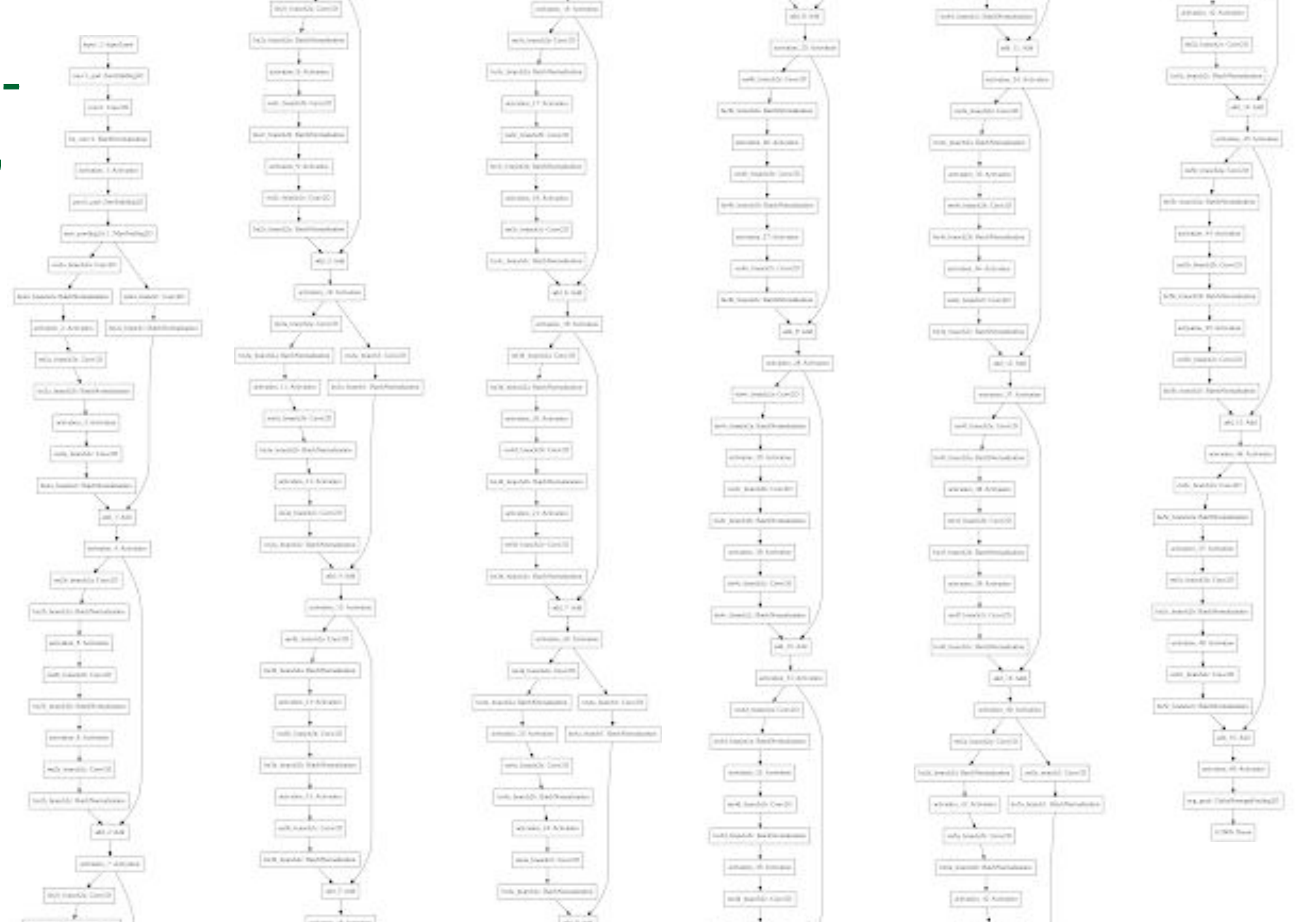
Hubble Space Telescope images of nine strongly lensed galaxies from the SL2S survey.

Example 02 - Resnet50 SL finder

Let’s say is a bit more complex than the previous...

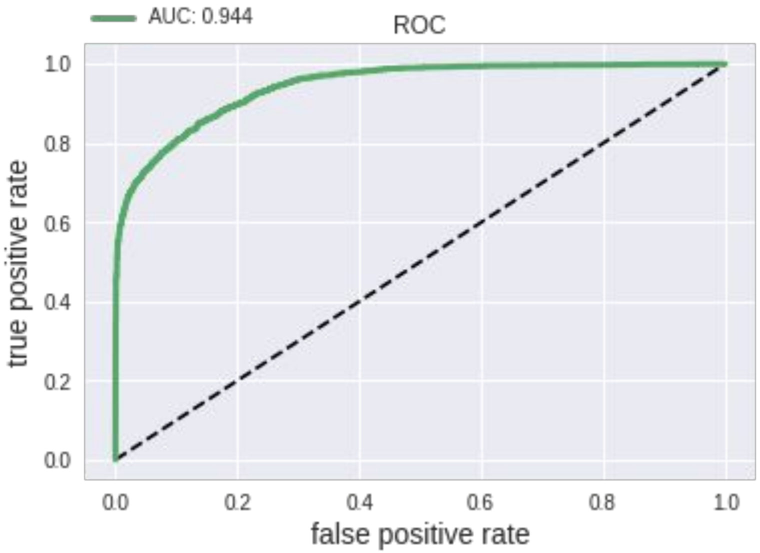
Total params: 23,591,810
Trainable params: 23,538,690
Non-trainable params: 53,120

On this version the K-fold is implemented



Example 02 - Resnet50 SL finder

On this version the K-fold is implemented



CEFET/RJ

CBPF

Deep Learning Applications in Astronomy

Clécio R. Bom

clearnightsrthebest.com

debom@cbpf.br

February 12, 2018