Les réseaux de neurones artificiels

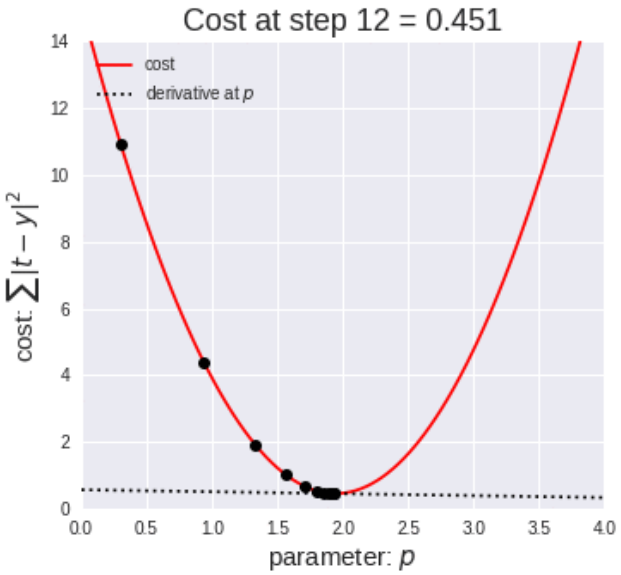
Hatem Ghorbel et Stefano Carrino
INF3
Haute Ecole Arc Ingénierie

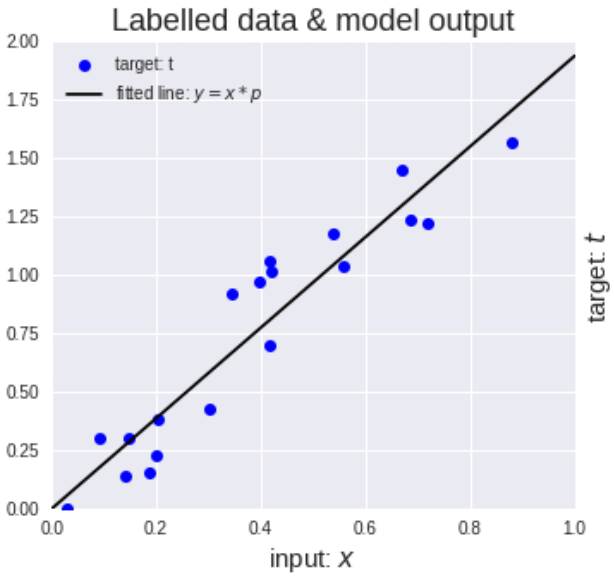
Sommaire

Neural network fondamental notions

- Epochs & Batch & Iterations
  - Batch and mini-batches
- Loss functions
- Optimizers
- Regularization
- Convolutional Neural Networks
- Transfer learning

Gradient descent (recall)



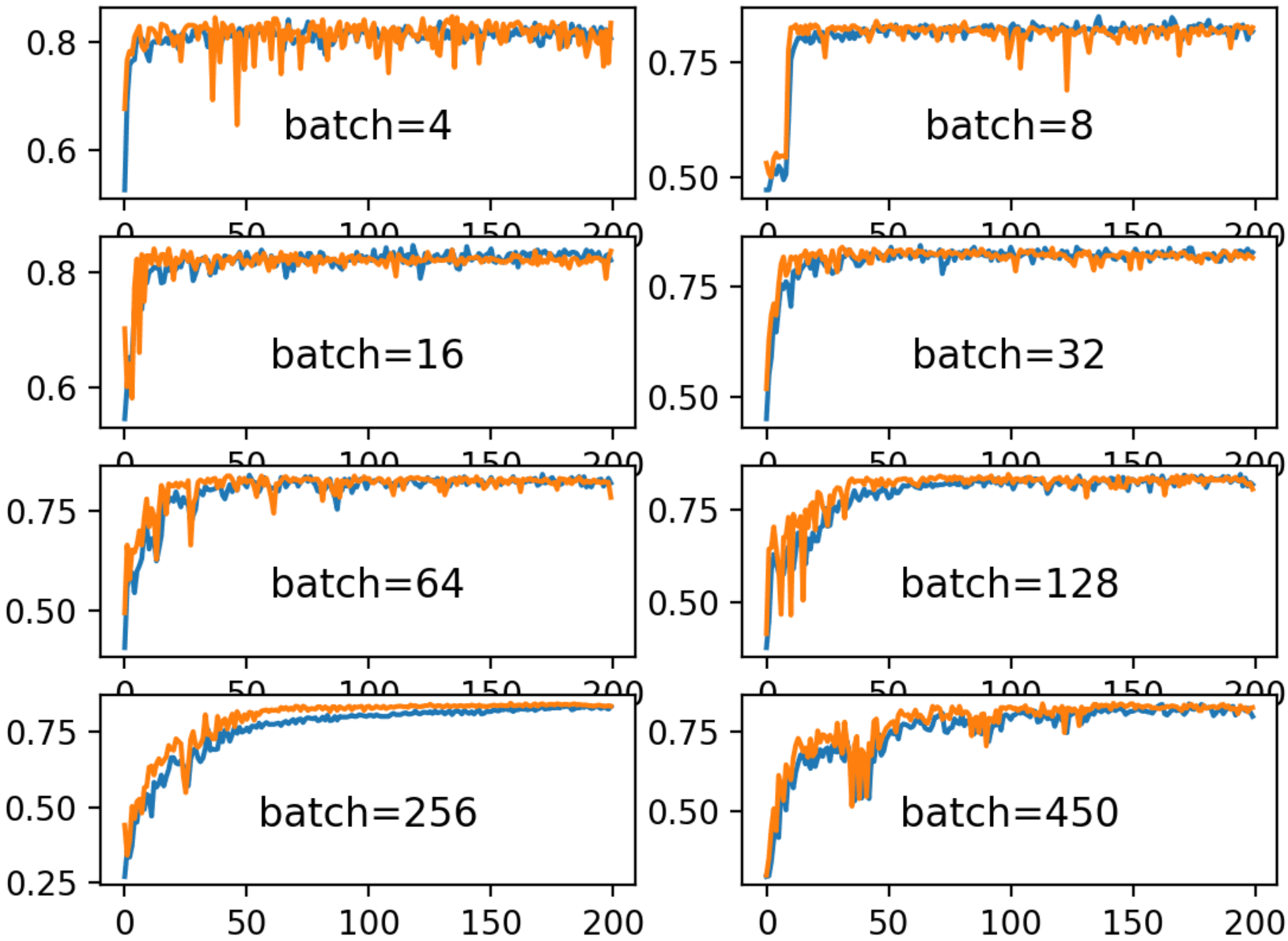


Epochs Vs Batch Vs Iteration

- One **epoch** is when an **entire** dataset is passed forward and backward through the neural network only **once.**

- The **batch size** is a hyperparameter that defines the number of samples to work through before updating the internal model parameters.
  - **Batch Gradient Descent**: Batch Size = Size of Training Set
  - **Stochastic Gradient Descent**: Batch Size = 1
  - **Mini-Batch Gradient Descent**: 1 < Batch Size < Size of Training Set

- **Iterations** is the number of batches needed to complete one epoch.


batch=4
batch=8
batch=16
batch=32
batch=64
batch=128
batch=256
batch=450
0.8
0.6
0.75
0.50
0.25
0
50
100
150
200

Loss functions

- In the context of an optimization algorithm, the function used to evaluate a candidate solution is referred to as the **objective function**.
  - We may seek to maximize or minimize the objective function, meaning that we are searching for a candidate solution that has the highest or lowest score respectively.

- With neural networks, we seek to minimize the error. As such, the objective function is often referred to as a cost function or a **loss function** and the value calculated by the loss function is referred to as simply "***loss***".

- Many loss functions exists (you can create your own), how to choose?

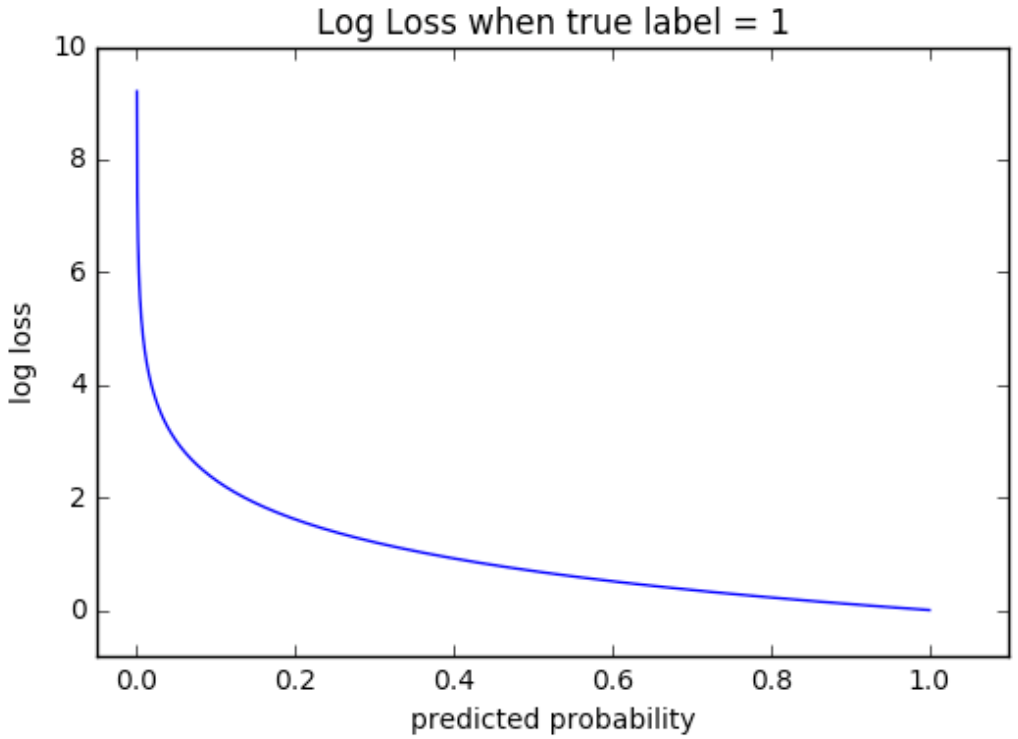
Loss functions

- **Regression Problem**
  - A problem where you predict a real-value quantity.
  - **Output Layer Configuration**: One node with a linear activation unit.
  - **Loss Function**: Mean Squared Error (MSE).

- **Binary Classification Problem**
  - A problem where you classify an example as belonging to one of two classes.
  - The problem is framed as predicting the likelihood of an example belonging to class one, e.g. the class that you assign the integer value 1, whereas the other class is assigned the value 0.
  - **Output Layer Configuration**: One node with a sigmoid activation unit.
  - **Loss Function**: Cross-Entropy, also referred to as Logarithmic loss.

- **Multi-Class Classification Problem**
  - A problem where you classify an example as belonging to one of more than two classes.
  - The problem is framed as predicting the likelihood of an example belonging to each class.
  - **Output Layer Configuration**: One node for each class using the softmax activation function.
  - **Loss Function**: Cross-Entropy, also referred to as Logarithmic loss.

Ex. Cross-entropy loss



More details: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.htm

Optimizers

- (Recall) weights update: $\Delta W_{iL} = \alpha \cdot \delta_L \cdot x_i$
- **Optimizers** tie together the loss function and model parameters by updating the model in response to the output of the loss function.

- Idea 1: adaptive learning rate (α)
- Idea 1.1: take into account past gradients to smooth out the update

- Examples
  - Mini-Batch Gradient Descent
  - Stocastic Gradient Descent (SGD)
  - Momentum
  - Adaptive Moment Estimation (ADAM)

More optimizers and details: https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html

Regularization

- In machine learning, **regularization** is way to prevent over-fitting.
- Typically, regularization reduces over-fitting by **adding a penalty to the loss function**.
- By adding this penalty, the model is trained such that it does not learn interdependent set of features weights
- Discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

Regularization

- Example: ridge regression

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

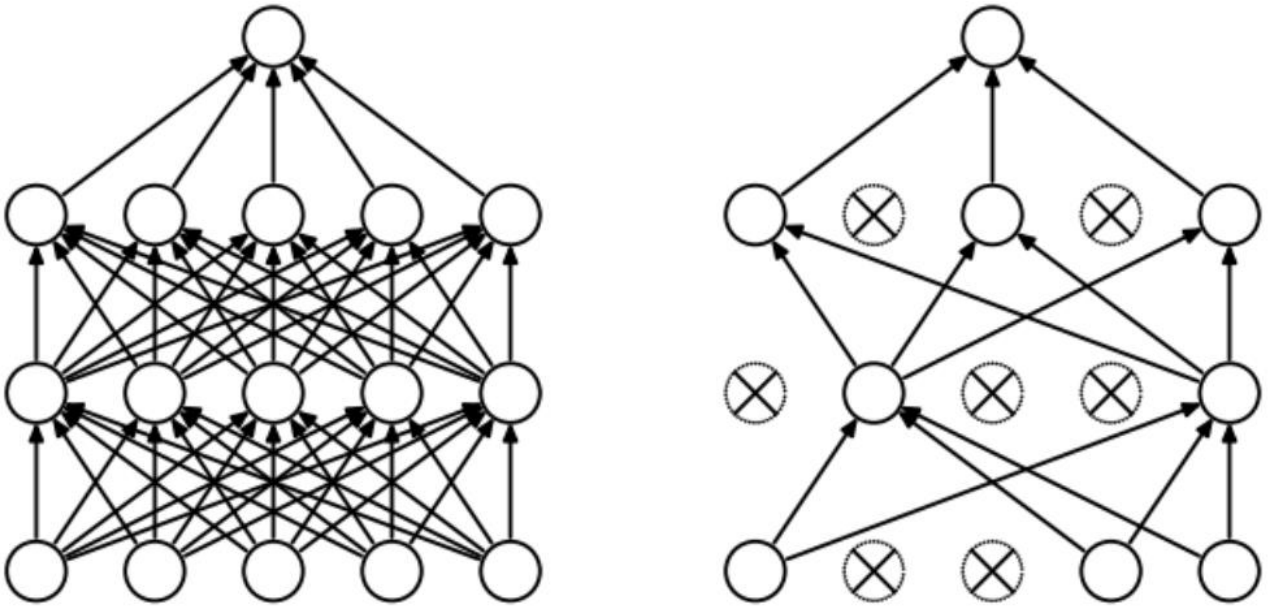
Residual Sum of Squares Loss function:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Regularization

- Dropout

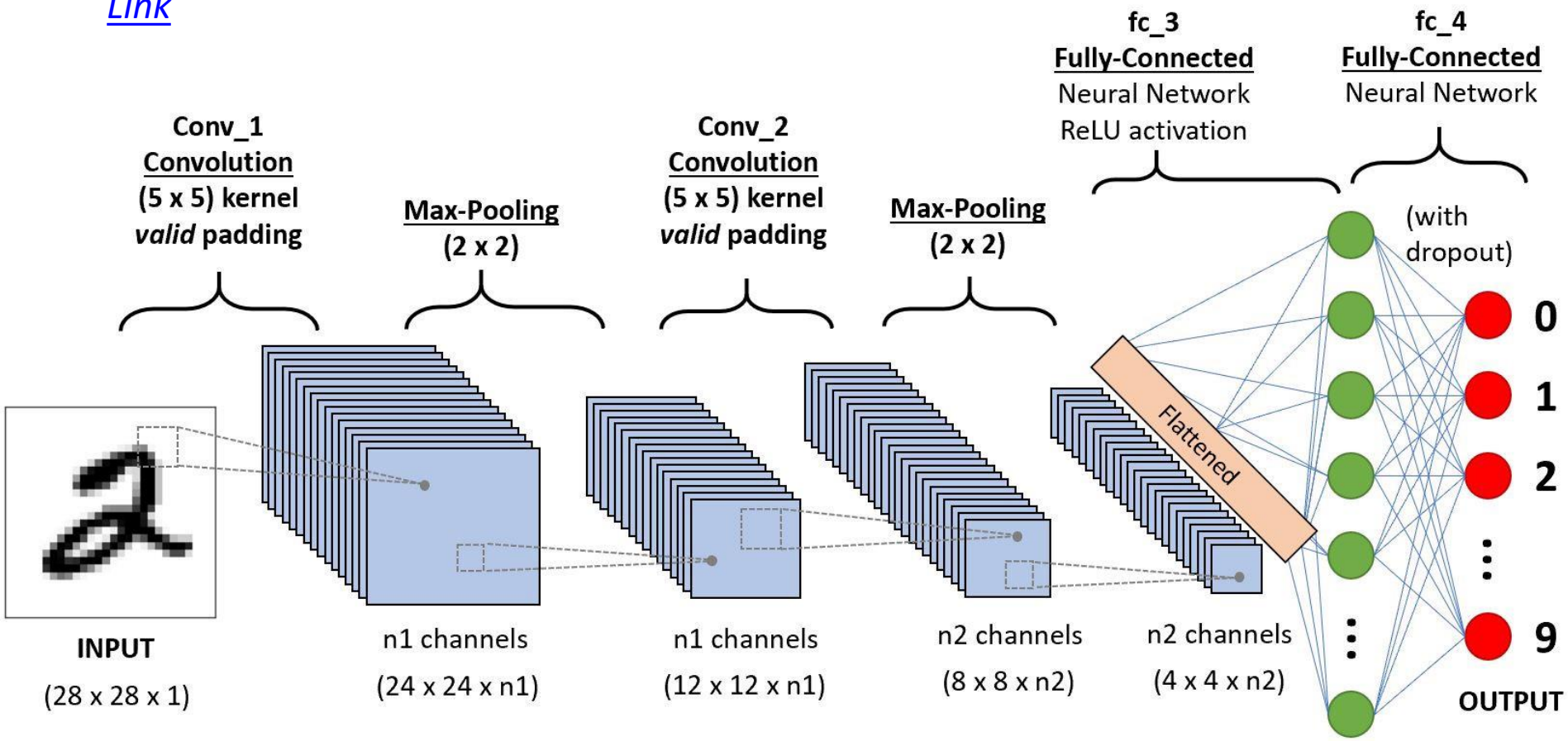


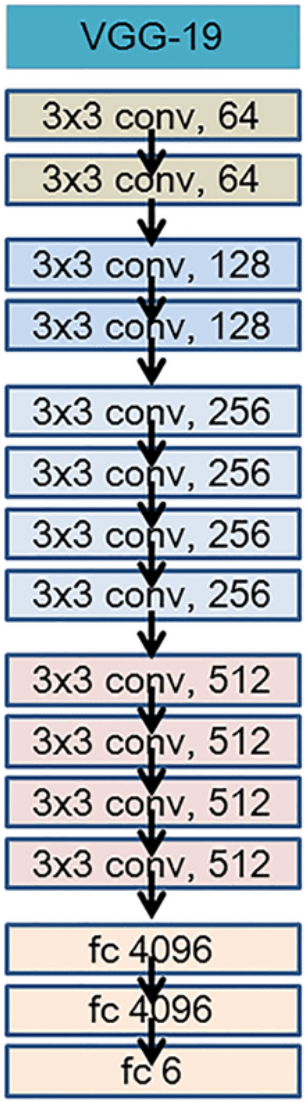
(a) Standard Neural Net

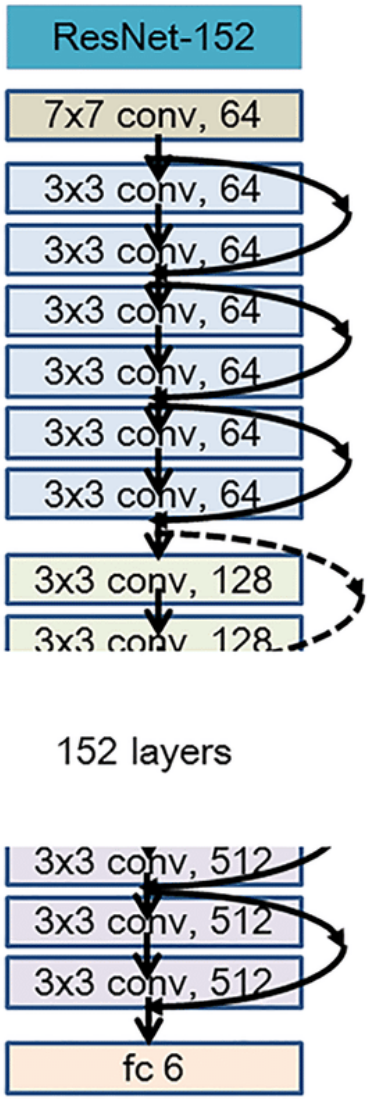
(b) After applying dropout.

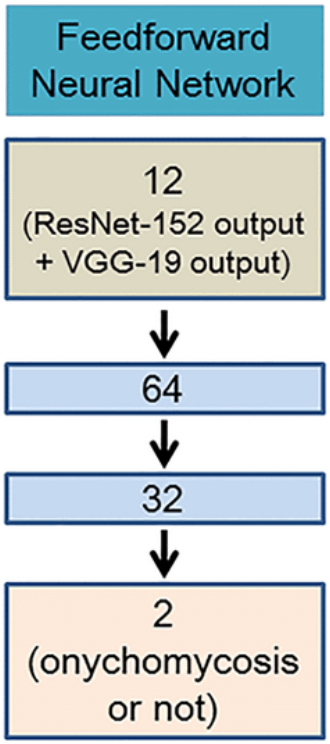
Working with images
Convolutional Neural Networks

Link

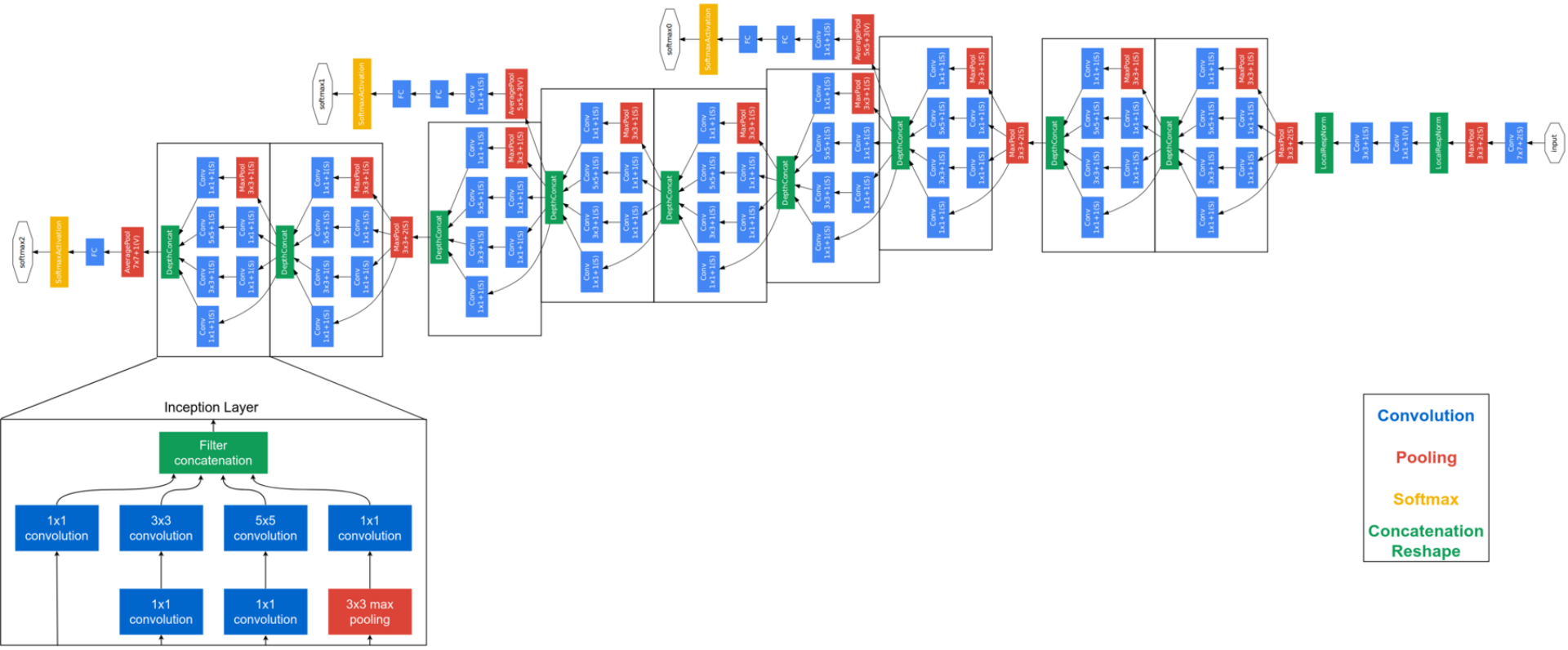



VGG-19
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
fc 4096
fc 4096
fc 6


ResNet-152
7x7 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128
3x3 conv, 128
152 layers
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
fc 6


Feedforward Neural Network
12
(ResNet-152 output + VGG-19 output)
64
32
2
(onychomycosis or not)

GoogLeNet

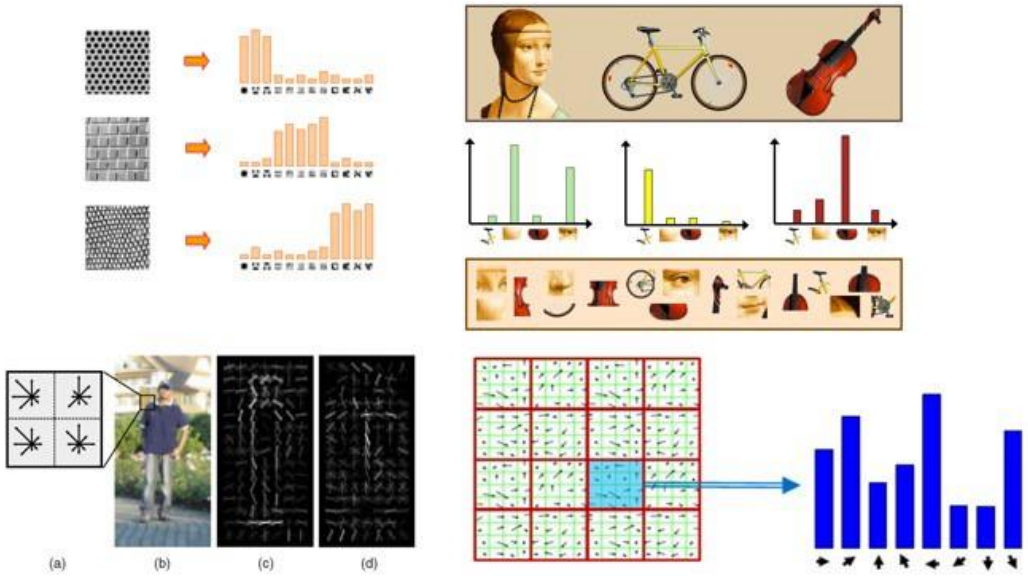

Inception Layer
Filter concatenation
1x1 convolution
3x3 convolution
5x5 convolution
1x1 convolution
1x1 convolution
1x1 convolution
3x3 max pooling
Convolution
Pooling
Softmax
Concatenation
Reshape

Transfer learning

- **Transfer learning** is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Context

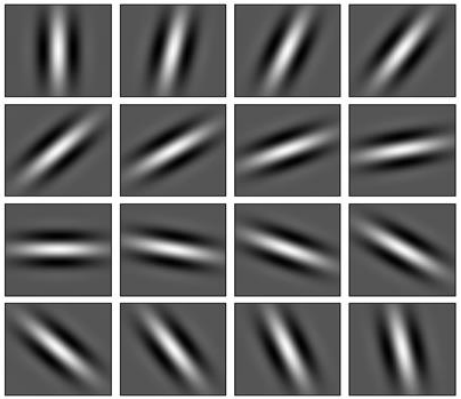
- Traditional image recognition with AI
  - Images have too much information => need for feature extraction



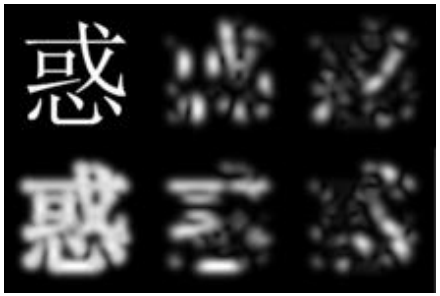
Context

- Traditional image recognition with AI
  - Images have too much information => need for feature extraction

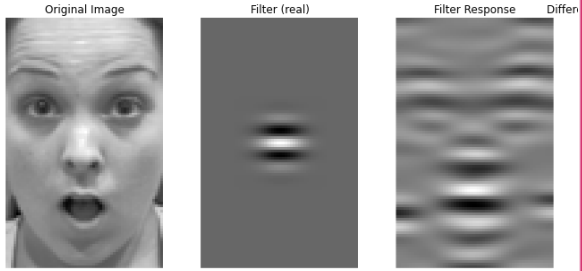
Gabor filters



https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97



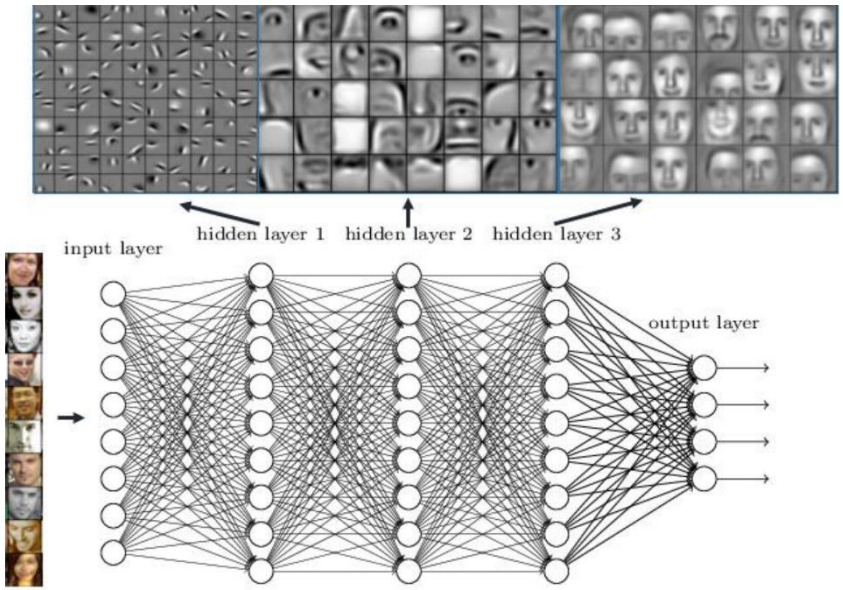
https://en.wikipedia.org/wiki/Gabor_filter#/media/File:Gabor-ocr.png

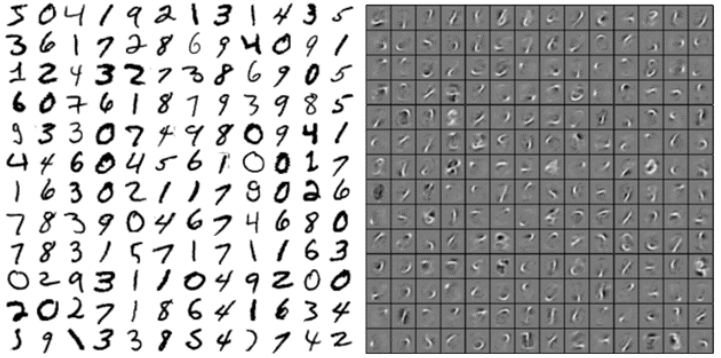


http://ttsuchi.github.io/2015/08/26/gaborfilters.html

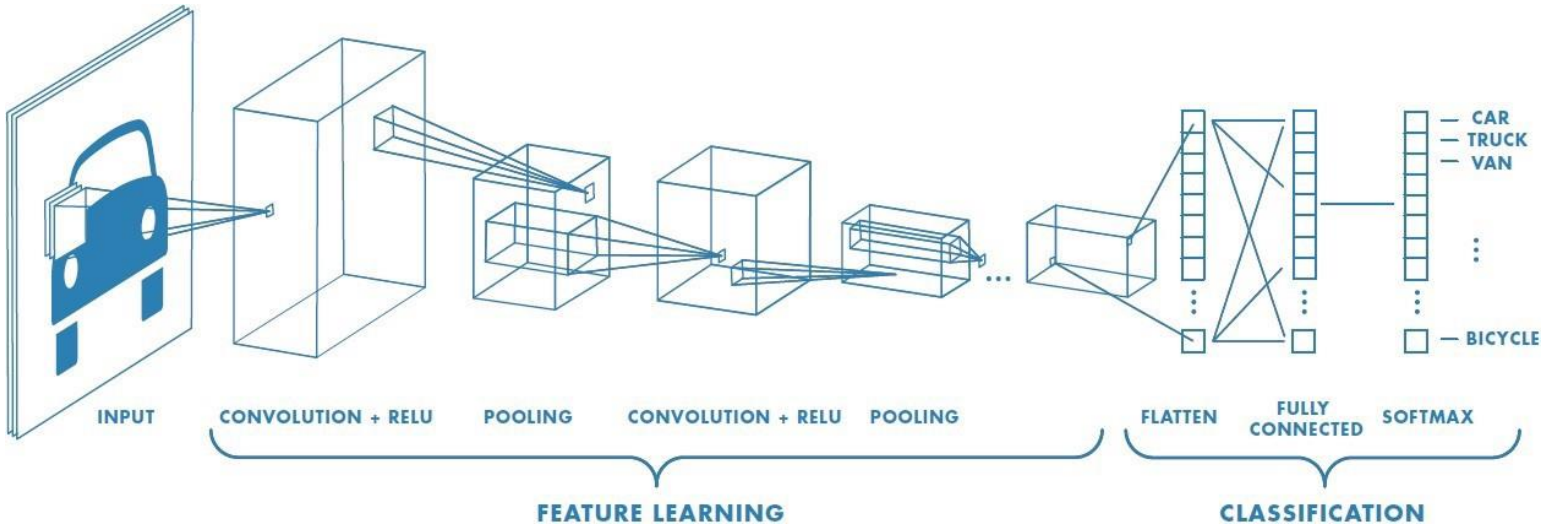
Context

- Deep learning image recognition with AI
  - No feature extraction => use raw data





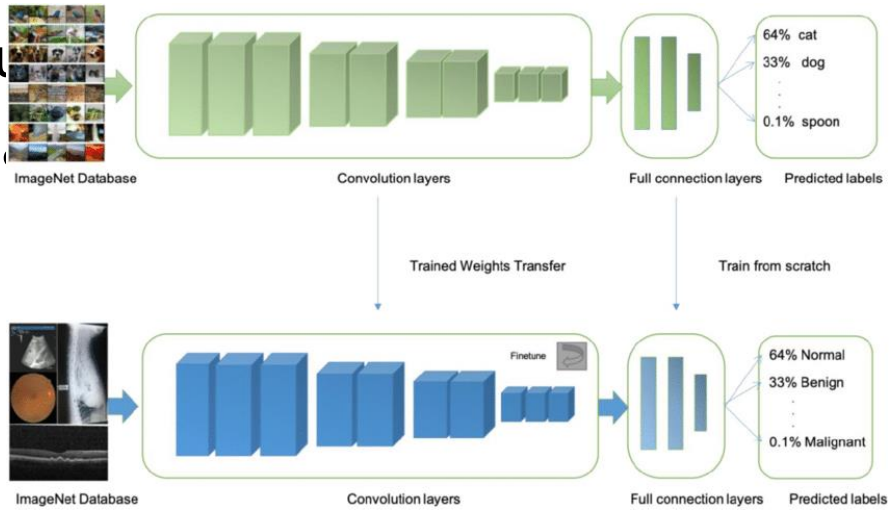
Transfer learning

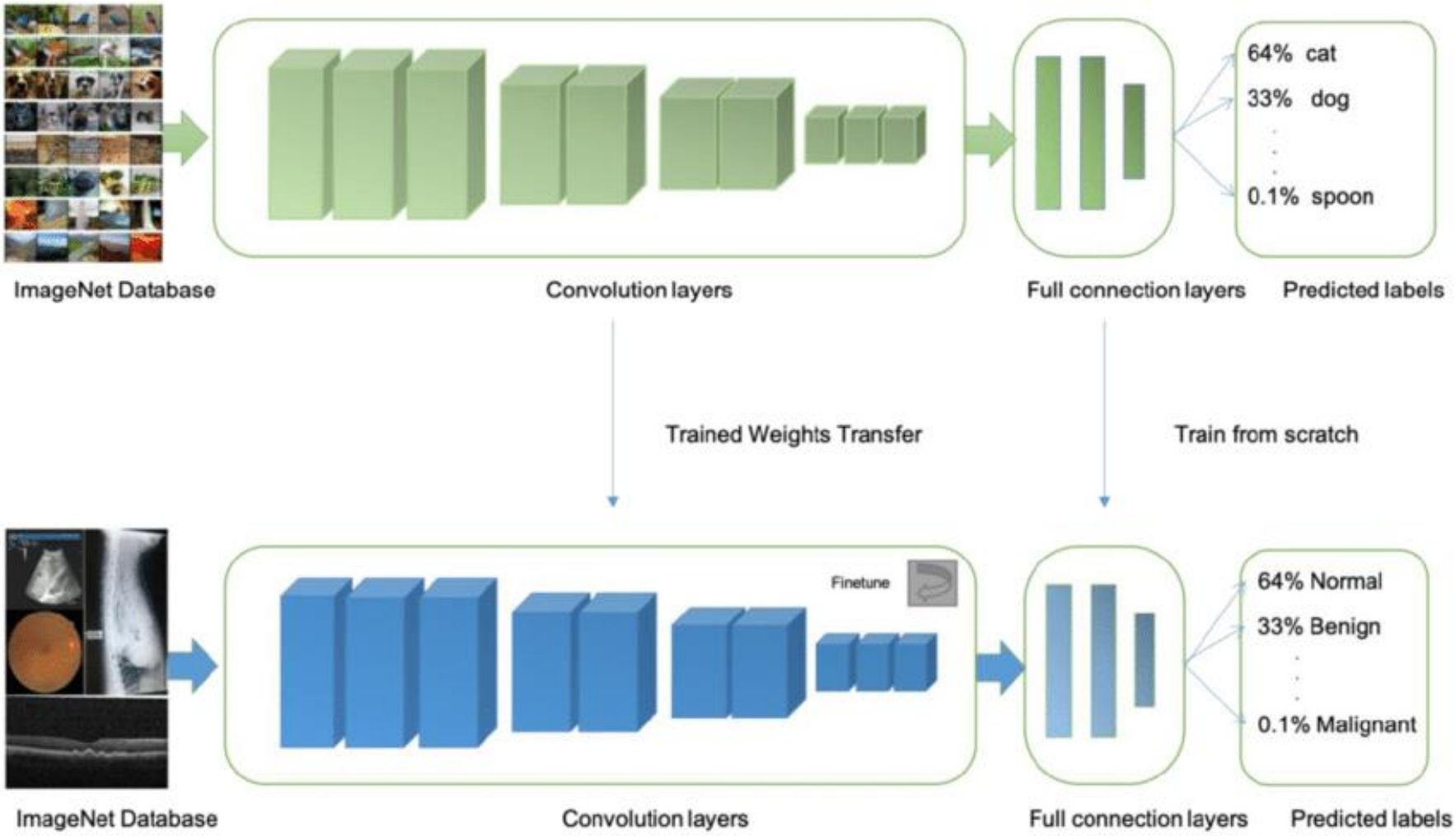


- **Transfer learning** is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Transfer learning

- 2 Typical approach:
  - **fixed feature extractor**
  - Train with generic, abou
  - Retrain with new, speci
  - Add fine tuning layers




64% cat
33% dog
0.1% spoon
ImageNet Database
Convolution layers
Full connection layers
Predicted labels
Trained Weights Transfer
Train from scratch
Finetune
64% Normal
33% Benign
0.1% Malignant
ImageNet Database
Convolution layers
Full connection layers
Predicted labels

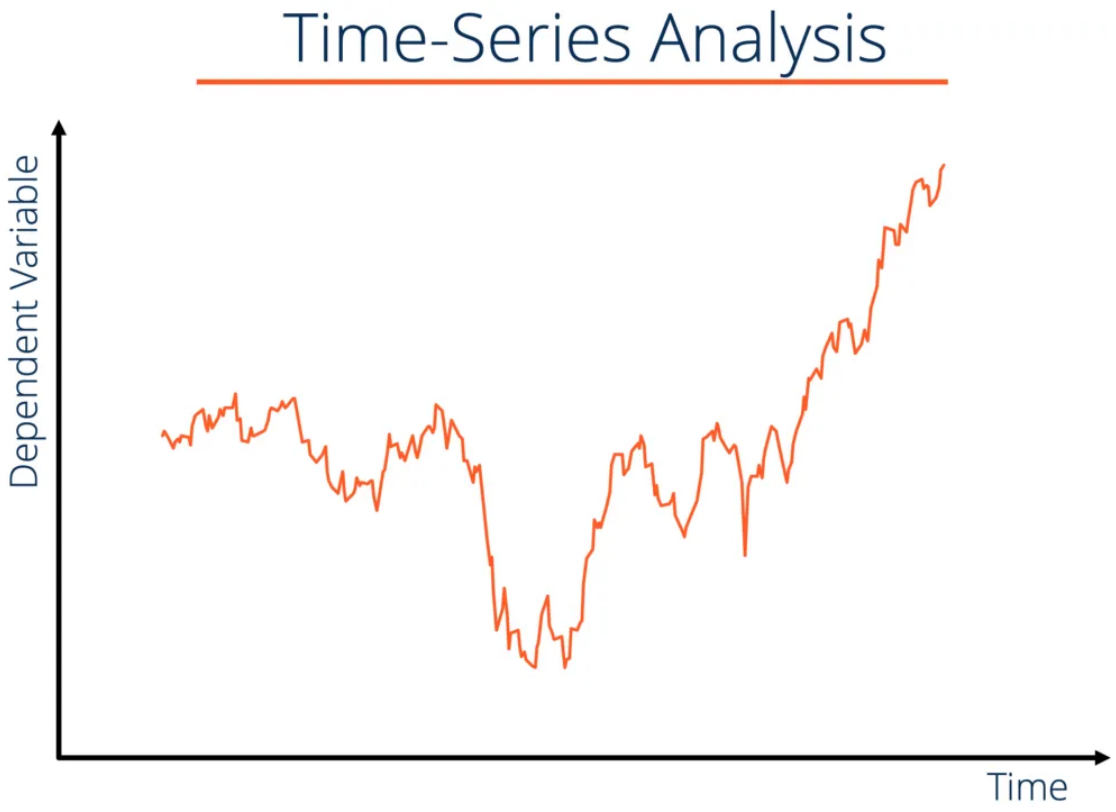
Transfer learning

- Nice tutorial (keras): https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

Going deeper…

- Recurrent Neural Networks
  - LSTM, GRU, …
- Attention mechanism
  - Transformer architecture
- Generative Adversarial Networks (GAN)
- Autoencoders
- …

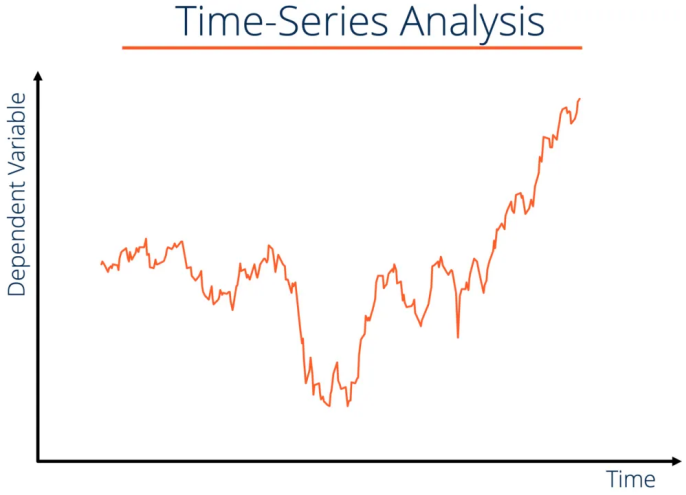
Working with time series



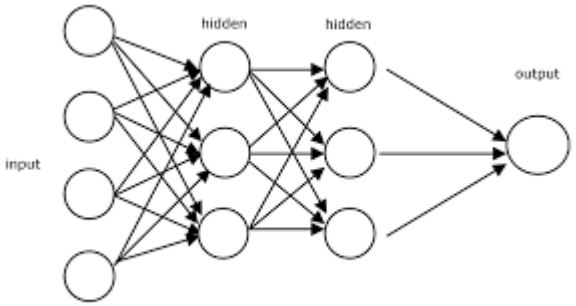
Methodology

- Dealing with time series is an adavanced topic in ML
- Here we will deal with a specific approach:
  - Transform a time serie into a "normal" input vector

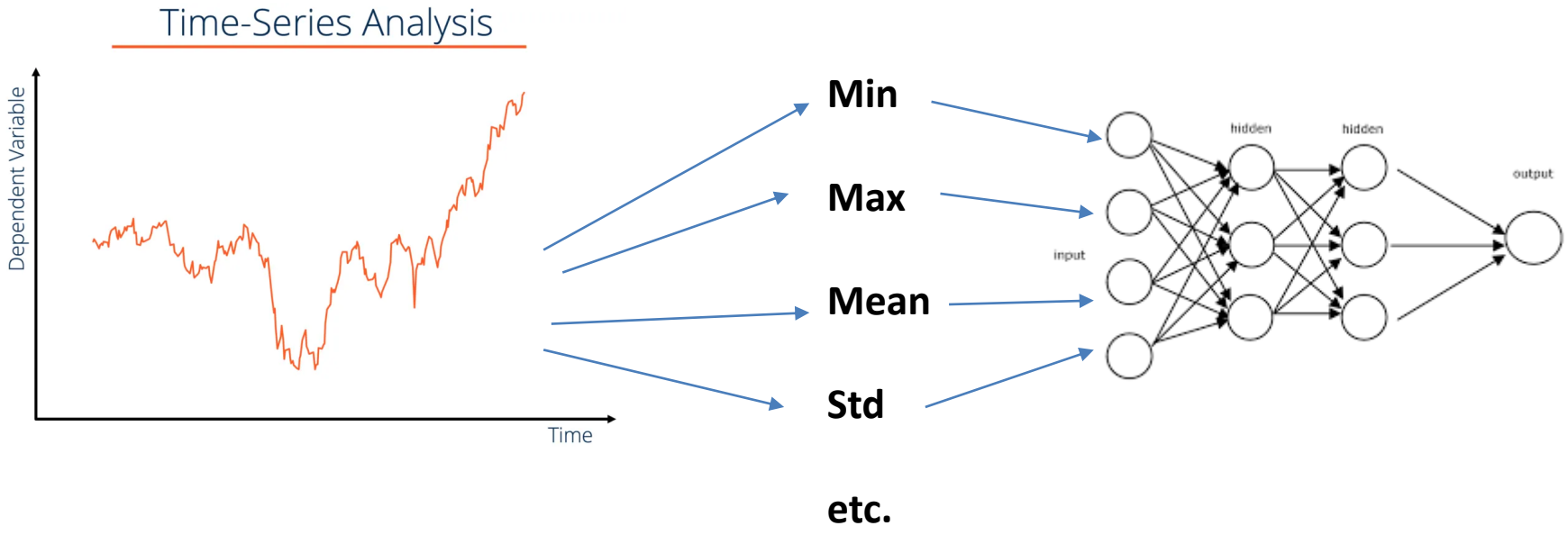
"Problems" with time series



?

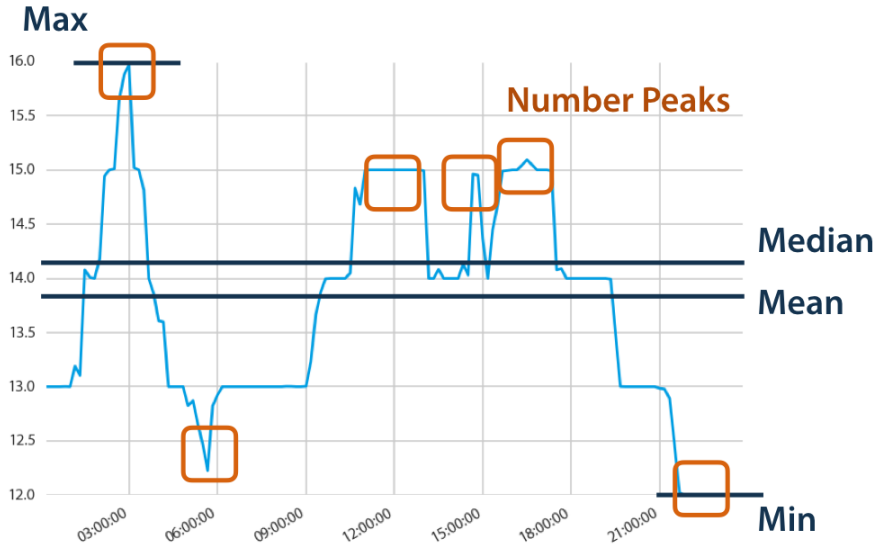


"Problems" with time series: Feature Extraction



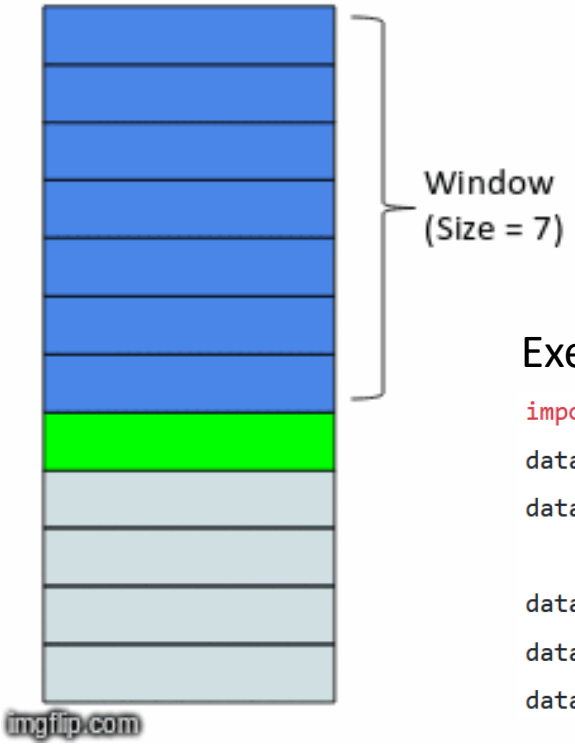
"Problems" with time series: Feature Extraction

- Other statistical features: **tsfresh** (1200+)
  - https://github.com/blue-yonder/tsfresh
- Demo/quickstart:
  - https://tsfresh.readthedocs.io/en/latest/text/quick_start.html




```
tsfresh.extract_relevant_features()
```

Feature Extraction: Rolling Window



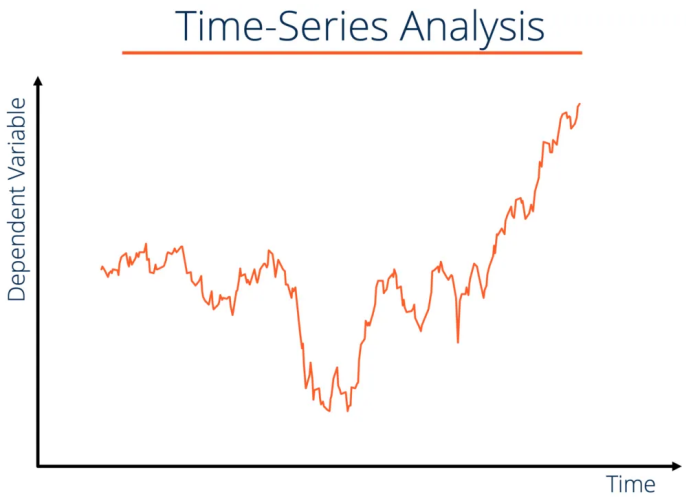
https://pandas.pydata.org/

Exemple mean (with pandas)

```
import pandas as pd
data = pd.read_csv('Train_SU63ISt.csv')
data['Datetime'] = pd.to_datetime(data['Datetime'],format='%d-%m-%Y %H:%M')

data['rolling_mean'] = data['Count'].rolling(window=7).mean()
data = data[['Datetime', 'rolling_mean', 'Count']]
data.head(10)
```

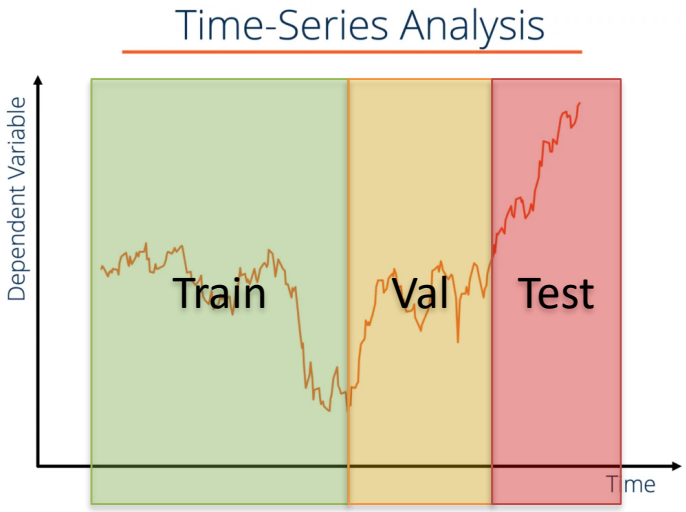
Train Validation Test split – How to



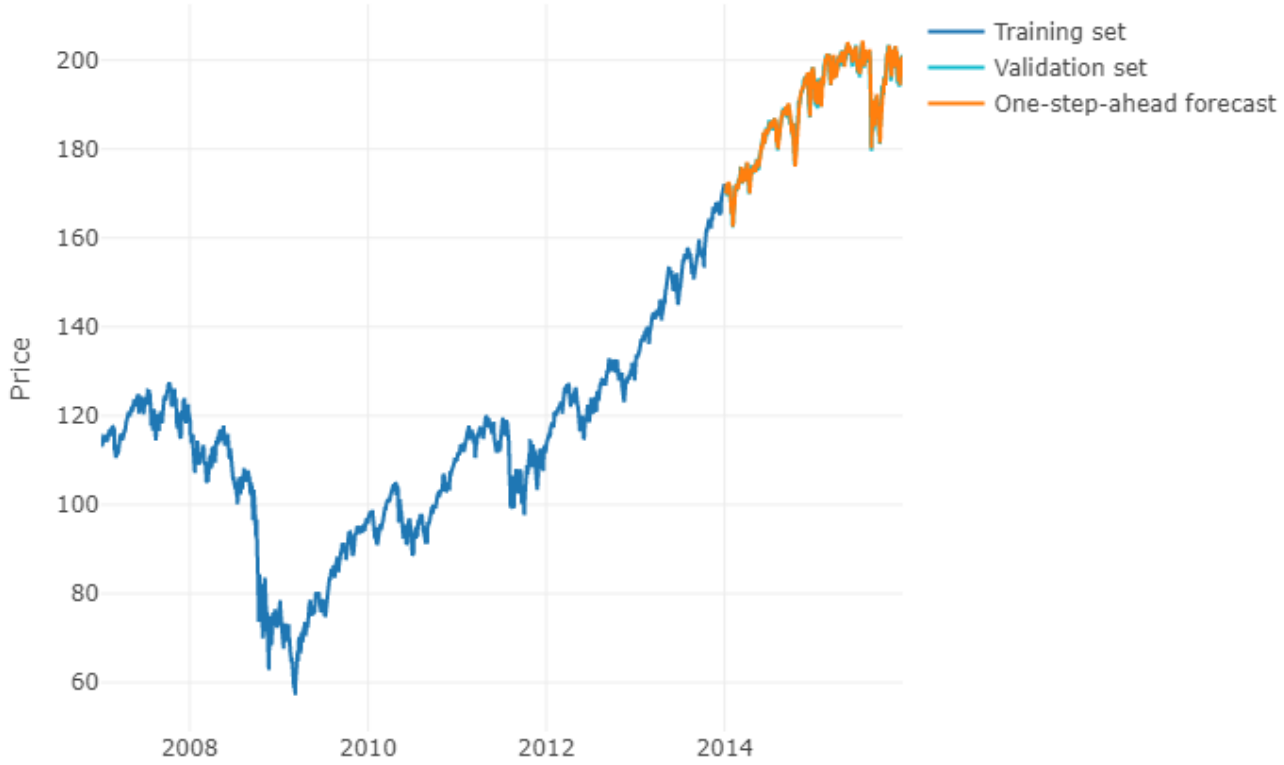
?

Train Validation Test split – How to

DON'T SHUFFLE!

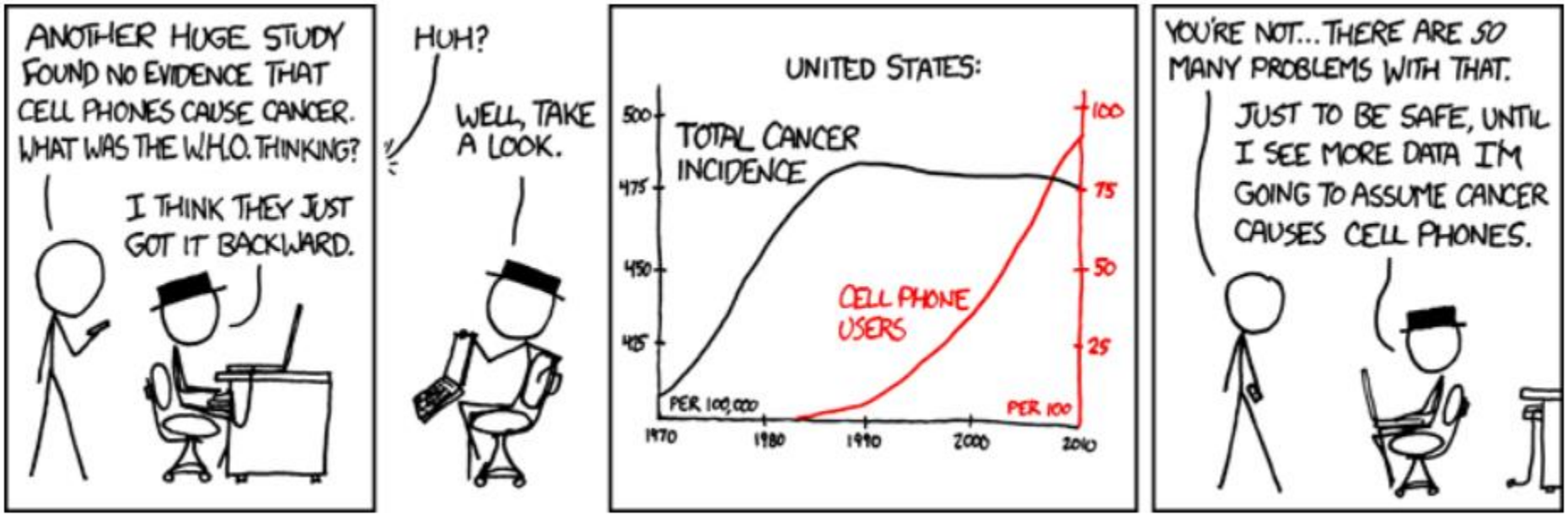


Other challenges: Forecast



More details: link

Other challenges: Causality & Correlation



(Source: https://xkcd.com/925/)