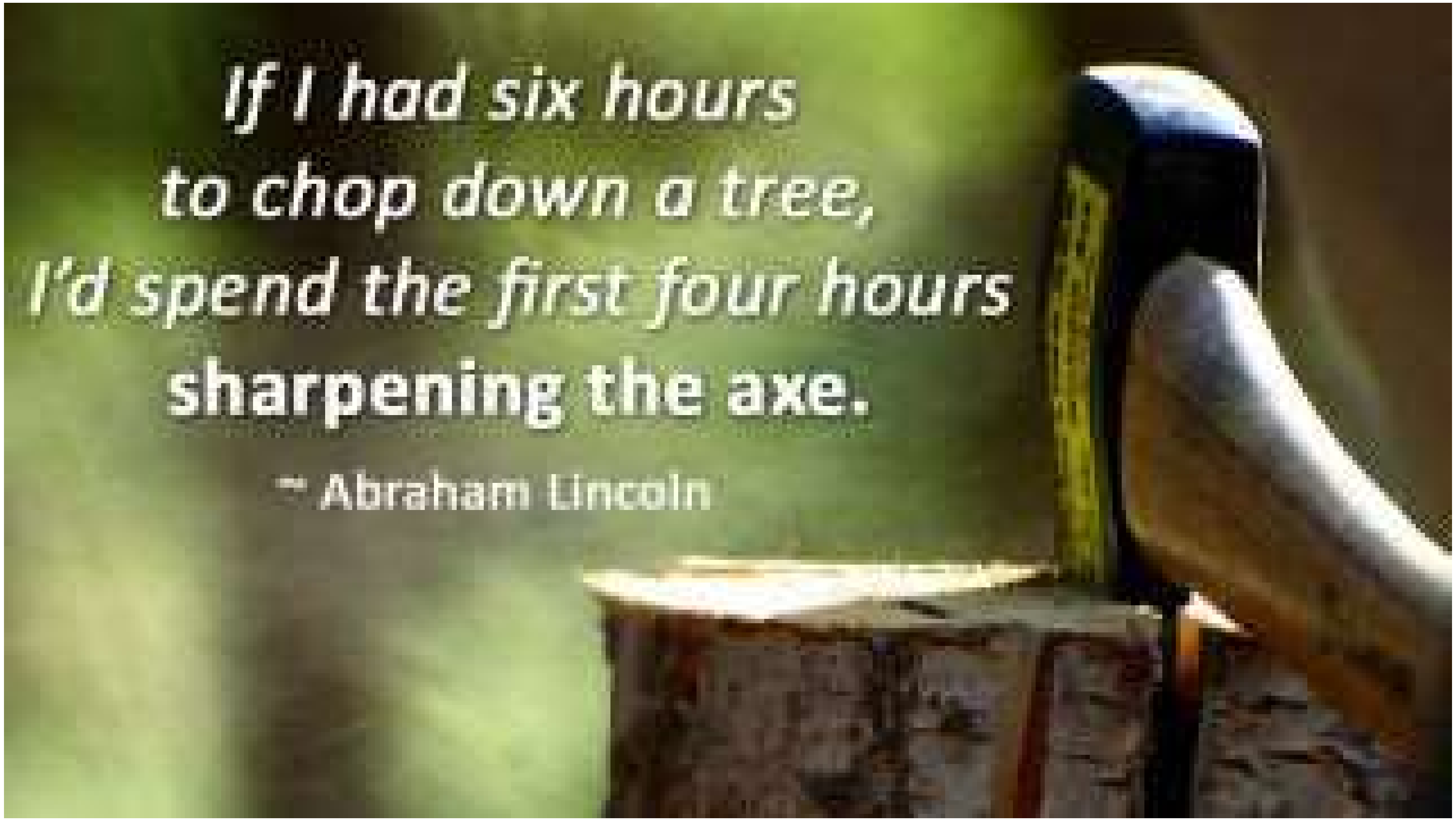
Kaggle competition kickoff

David Fredman | Sonat Consulting
Bergen Machine Learning Meetup 2017-05-04

Twitter: @davidfredman
Linkedin: https://www.linkedin.com/in/davidfredman/

Introduction

- Data Scientist & Senior Consultant at Sonat Consulting (www.sonat.no) in Bergen
- I work on machine learning strategy, architecture and development in banking, industry (maintenance, optimization), health and biotech
- Ph.D. in Bioinformatics: focus on data analytics, algorithms & method development from Karolinska Institutet, Sweden
- Recovering academic: Worked as a researcher, research leader & consultant at three universities in three countries

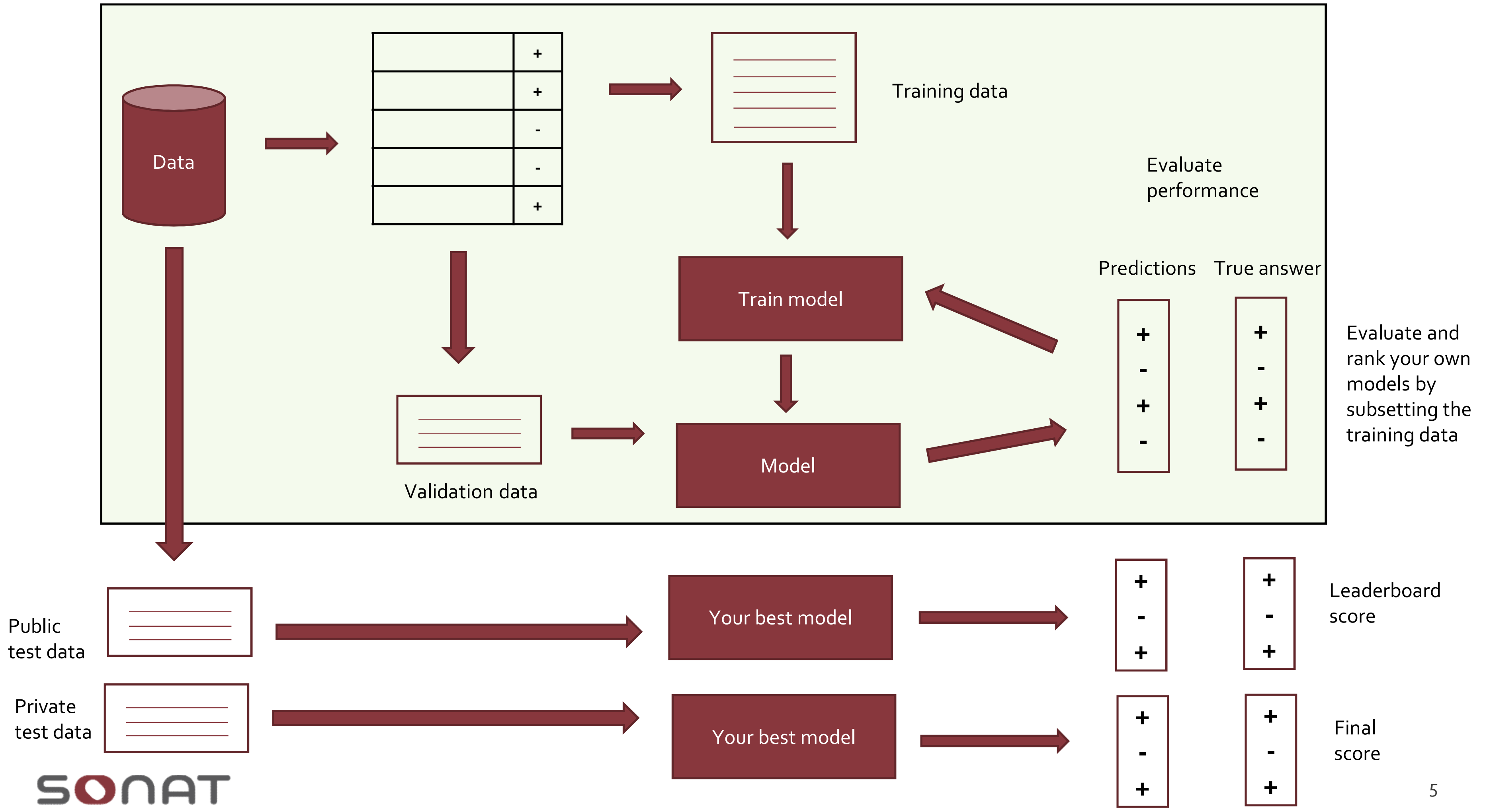

If I had six hours
to chop down a tree,
I'd spend the first four hours
sharpening the axe.
~ Abraham Lincoln

Why Kaggle

- **Open and (reasonably) clean data**
- **Practice and use new methods**
  - **Cross validation**
  - **Feature engineering**
  - **New algorithms / techniques**
- **Clear goal**
- **Instant feedback on how you did on a problem; Measurable performance and improvement**
- **Sharing of solutions / ideas / code – learn from others**
- **Support for team formation**
- The joy of mastery :) || Solving an important problem for the common good || Glory, fame & fortune

SONAT

Kaggle Evaluation



The performance of your model depends on

- ✓ **Data**
  - ➢ Preprocessing of data
  - ➢ Missing or noisy data
  - ➢ The class distribution
  - ➢ "Feature engineering"
- ✓ **The learning algorithm**
  - ✓ Which algorithm are you using
  - ✓ Model complexity
- ✓ **How we evaluate the performance**
  - ✓ Treatment of training, validation and testset
    - ✓ Your validation set should be similar to the Kaggle test set
    - ✓ You should not overtrain on the Kaggle public test set
  - ✓ Cost for misclasifying
  - ✓ Which evaluation metric is used

Tutorial exercise: Predict Titanic survival

- Small, interpretable data

Practice:
- Data exploration
- Feature engineering
- Handling missing values

Random Forest

«Bagging»-technique

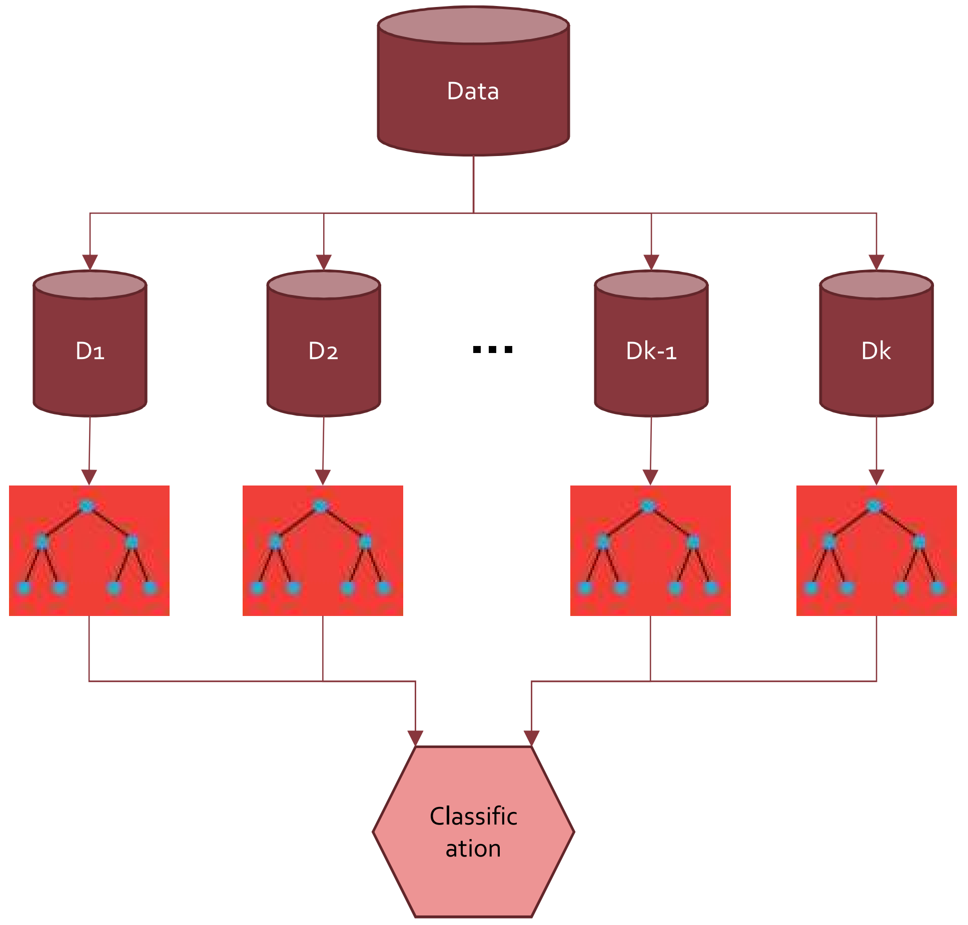
1: Split the data set randomly

2. Train a decision tree on each dataset

- Use a random subset of variables in each tree
- Split as many times as possible

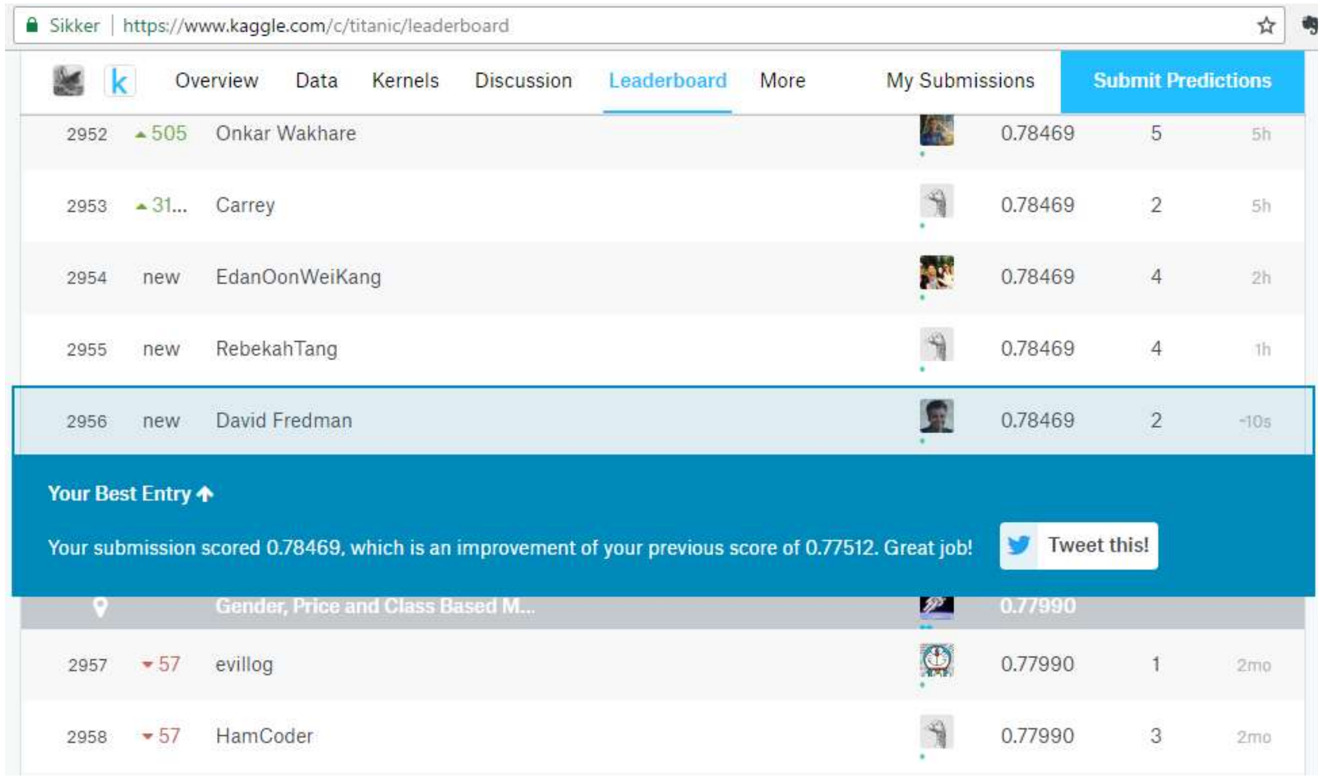
3. Combine classification from each tree

- Each tree votes for classification



Titteltekst

Benchmark: get into the top 50%



2952	▲505	Onkar Wakhare	0.78469	5	5h
2953	▲31...	Carrey	0.78469	2	5h
2954	new	EdanOonWeiKang	0.78469	4	2h
2955	new	RebekahTang	0.78469	4	1h
2956	new	David Fredman	0.78469	2	~10s

Your Best Entry ↑

Your submission scored 0.78469, which is an improvement of your previous score of 0.77512. Great job!

		Gender, Price and Class Based M...	0.77990		
2957	▼57	evillog	0.77990	1	2mo
2958	▼57	HamCoder	0.77990	3	2mo

[R Notebook demo]

Titanic: things to try

- Data exploration and interpretation:
  - what are the most informative variables?

- Feature engineering:
  - Grouping of values (rare titles)
  - Family grouping (names, sib flag)
  - Extract location on ship from cabin nr
  - ...

- Handling missing values
  - Different types of imputation

- Evaluation
  - Cross-validation as an estimate of performance on unseen data
  - Compare imputation vs algorithm that accepts missing data
  - Compare different algorithms

Dealing with missing values

In preferred order:

1. Use an algorithm/implementation that handles NA
   e.g. AzureML or h2o random forest
2. Replace NA with a new value (impute)
3. Exclude examples with NA (if few)
4. Exclude variables with NA

Titteltekst

B: Image classification with deep learning

Transfer learning

Transfer learning is a technique that generalizes models trained for one setting or task to other settings or tasks

Example: image recognition

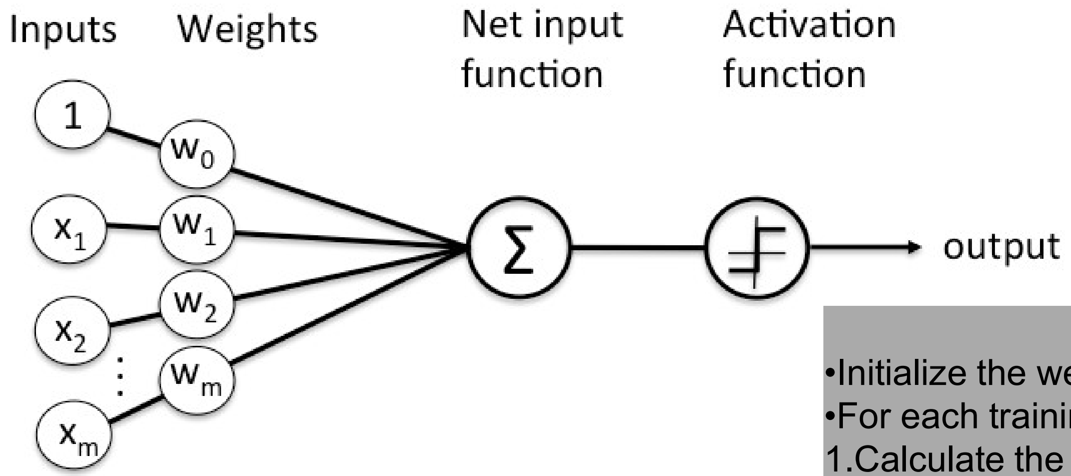
Let's suppose you want to create an algorithm which is able to differentiate a breast from a lung cancer based on medical images (such as MRI or pathology images).

Deep learning algorithms have superior performance on image recognition and should be the best choice, but would require 100k to 1M images to train it from scratch in order to achieve a good accuracy.

For this type of medical problem, most datasets that physicians and researchers can build are typically around only a few hundred images, maximum!

Solution: Transfer learning
Train the network on detecting other types of images first, then repurpose to recognize medical images.

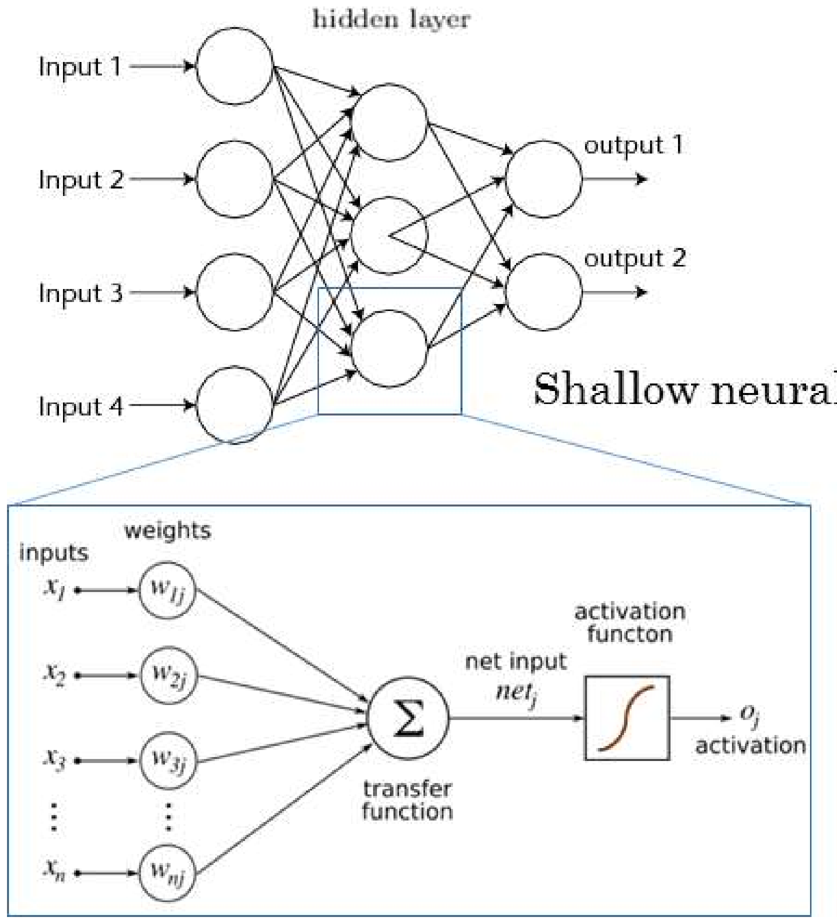
Deep Learning Quick Intro - The Perceptron



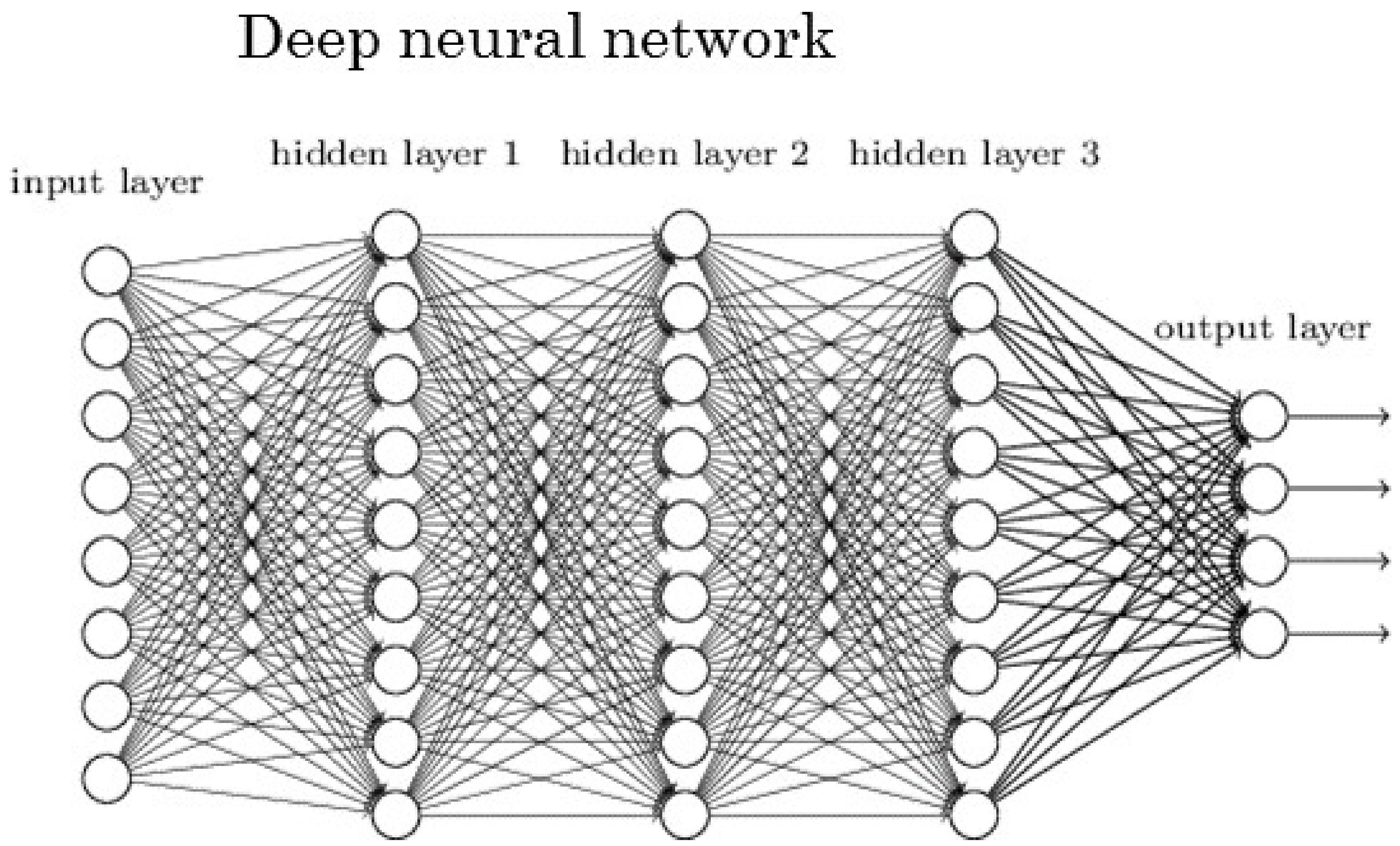
Schematic of Rosenblatt's perceptron.

- Initialize the weights to 0 or small random numbers.
- For each training sample x(i)x(i) :

1. Calculate the *output* value.
2. Update the weights.



Shallow neural network


Deep neural network
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

Deep neural networks learn hierarchical feature representations

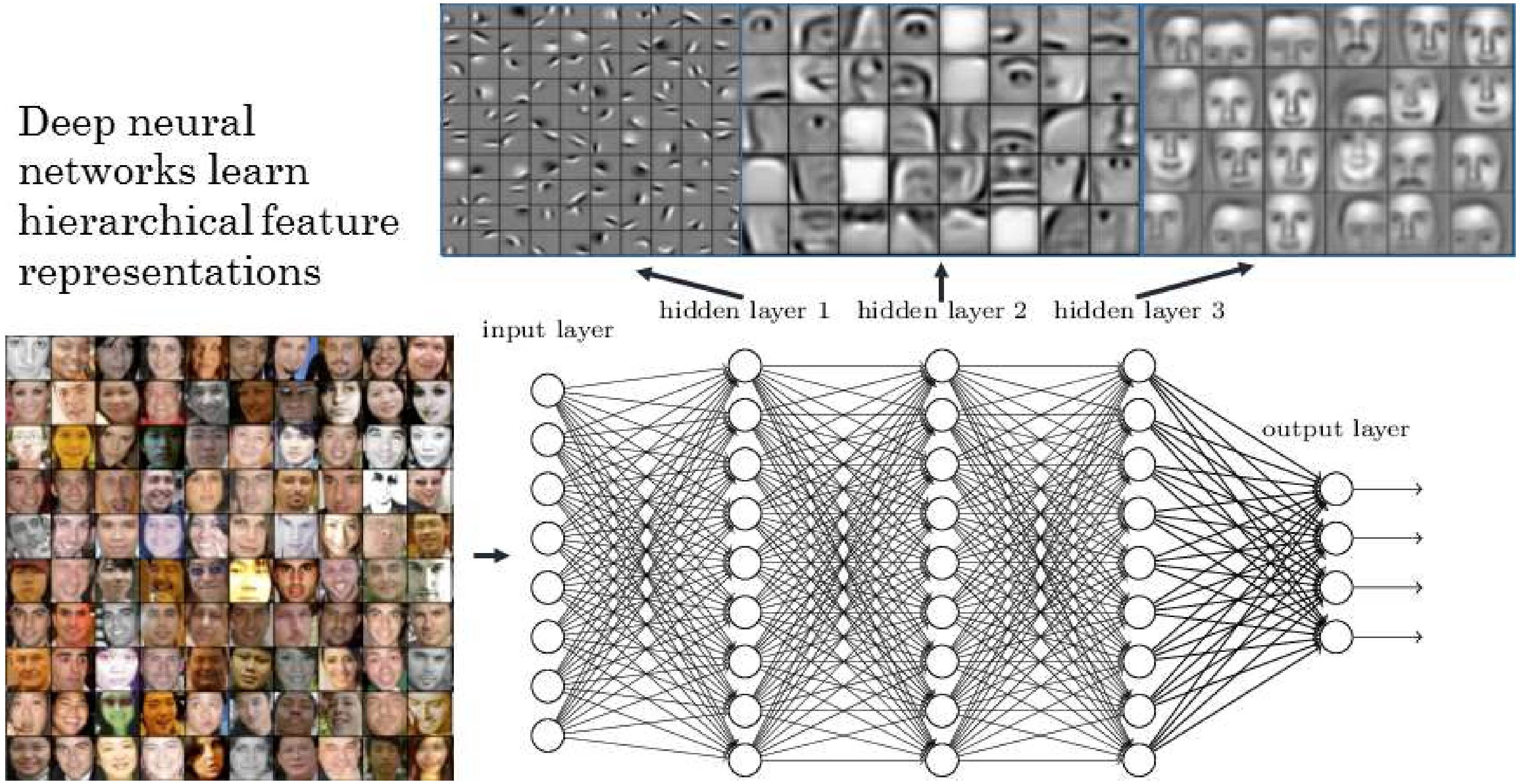
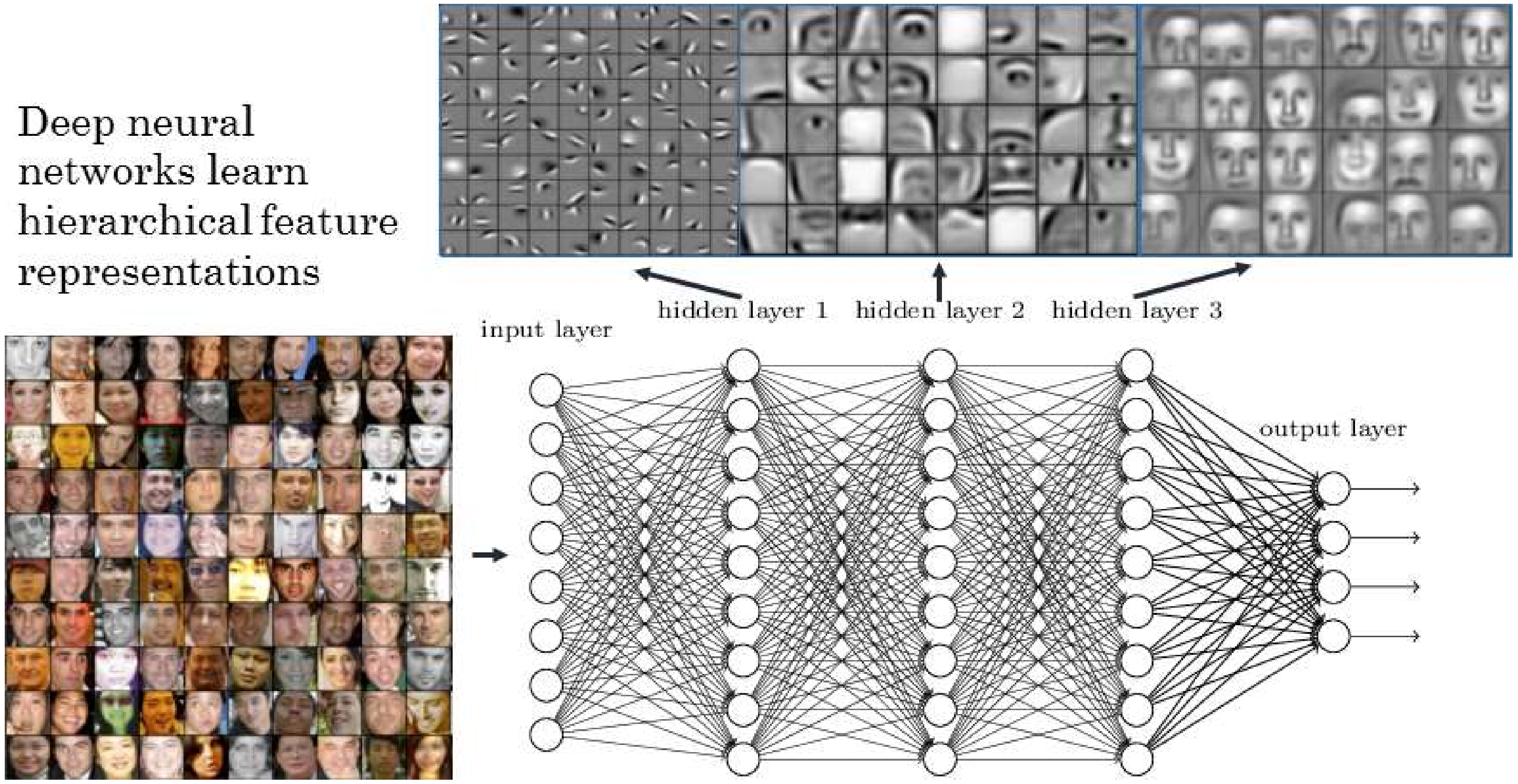


Image:	Pixel ⟶ edge ⟶ texton ⟶ motif ⟶ part ⟶object
Text:	Character ⟶ word ⟶ word group ⟶ clause ⟶ sentence
Speech:	Sample ⟶ spectral band ⟶ sound ⟶...⟶ phone ⟶ phoneme ⟶ word

Deep neural networks learn hierarchical feature representations



Transfer learning: reuse things we learned on one problem, and apply to a new similar problem.
How: Replace last (or 2nd last, 3rd last) layer with a new output layer with the classes of interest, re-train (update) weights with available examples, predict

Advantages of multi-level feature learning

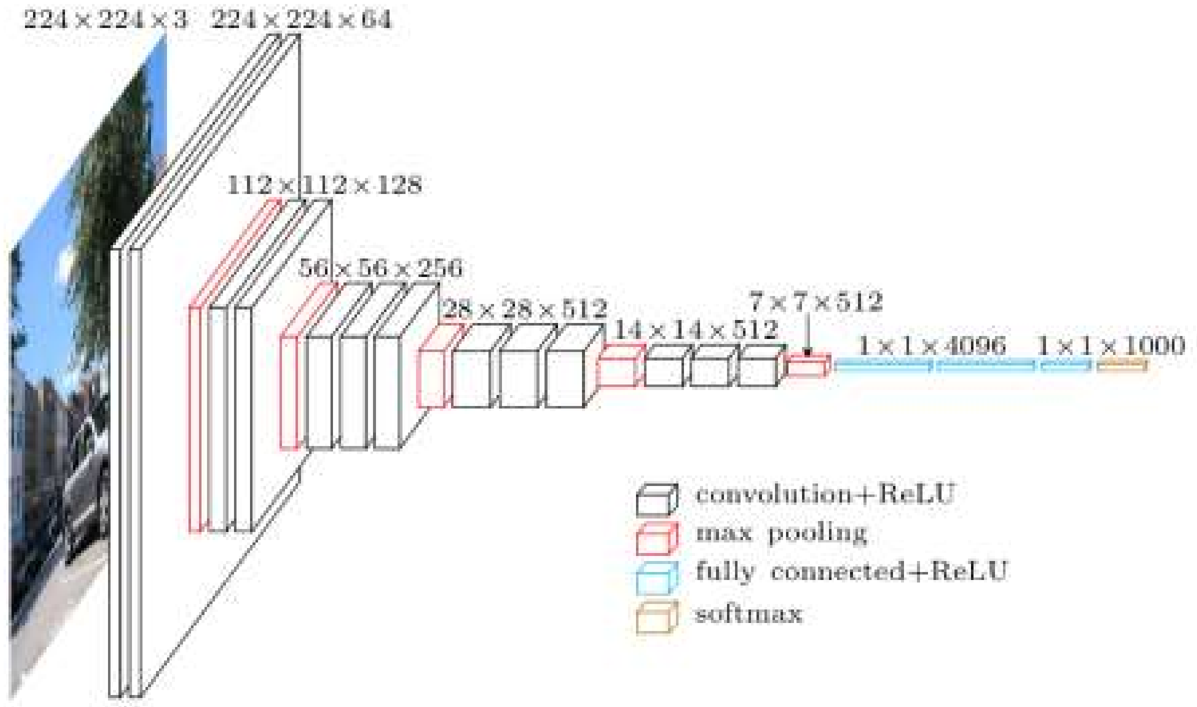
1. It can learn high-level features which are more robust against data variation than features at low-levels
2. it offers hierarchal parameter sharing that holds great expressive power
3. the feature learning can be conducted without any labelled data and so is cheap
4. with a little supervised training (fine-tuning), the learned models can be well adapted to specific tasks

Deep Visualization Toolbox

- https://github.com/yosinski/deep-visualization-toolbox
- http://yosinski.com/deepvis#toolbox

Titteltekst

VGG structure



Titteltekst

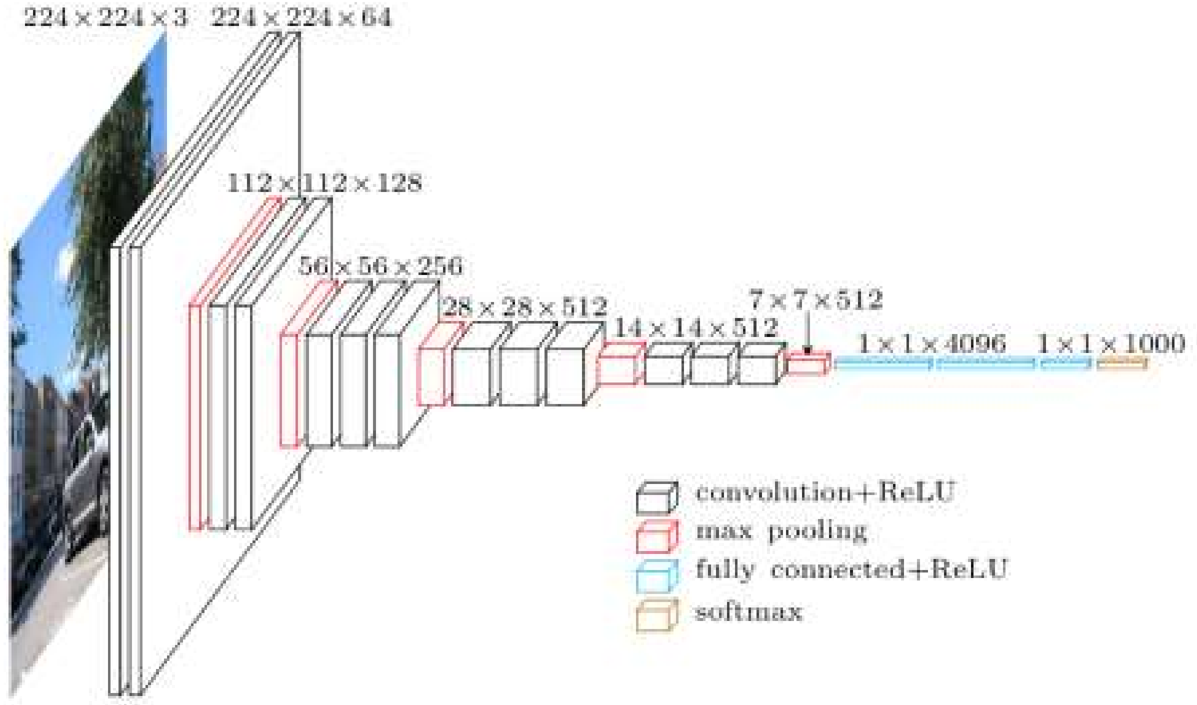
Convolution

http://setosa.io/ev/image-kernels/

Excel convolution & max pooling example

Titteltekst

VGG structure



Softmax

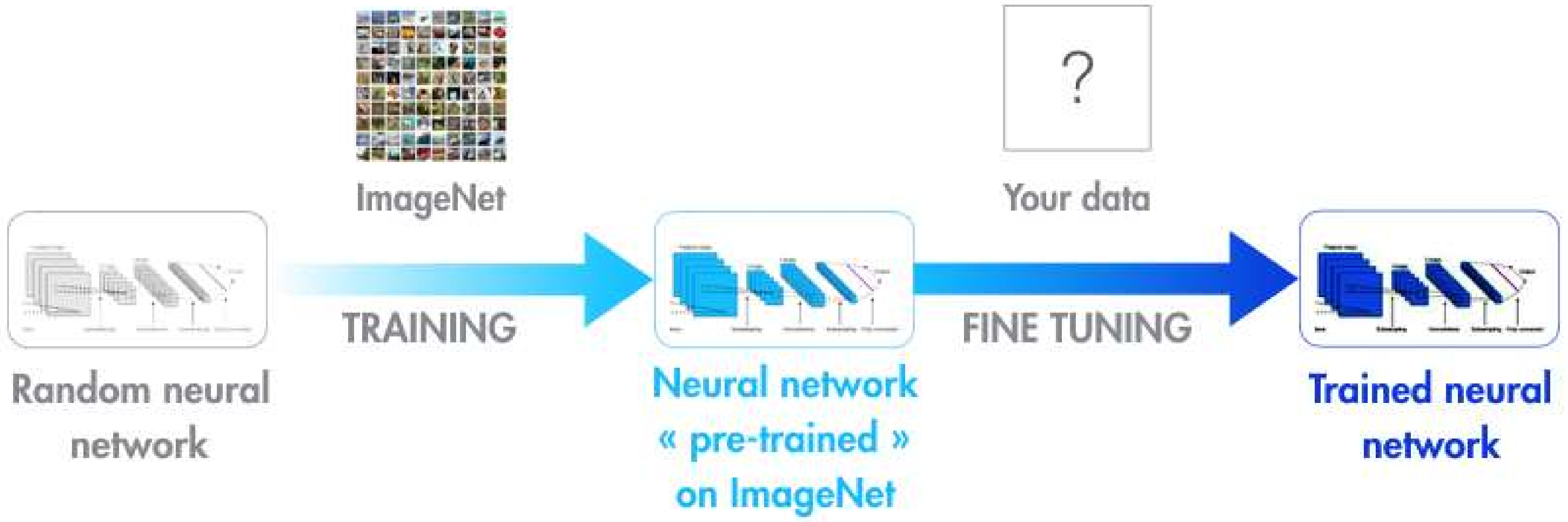
- Excel example (entropy_example.xls)

Convolution

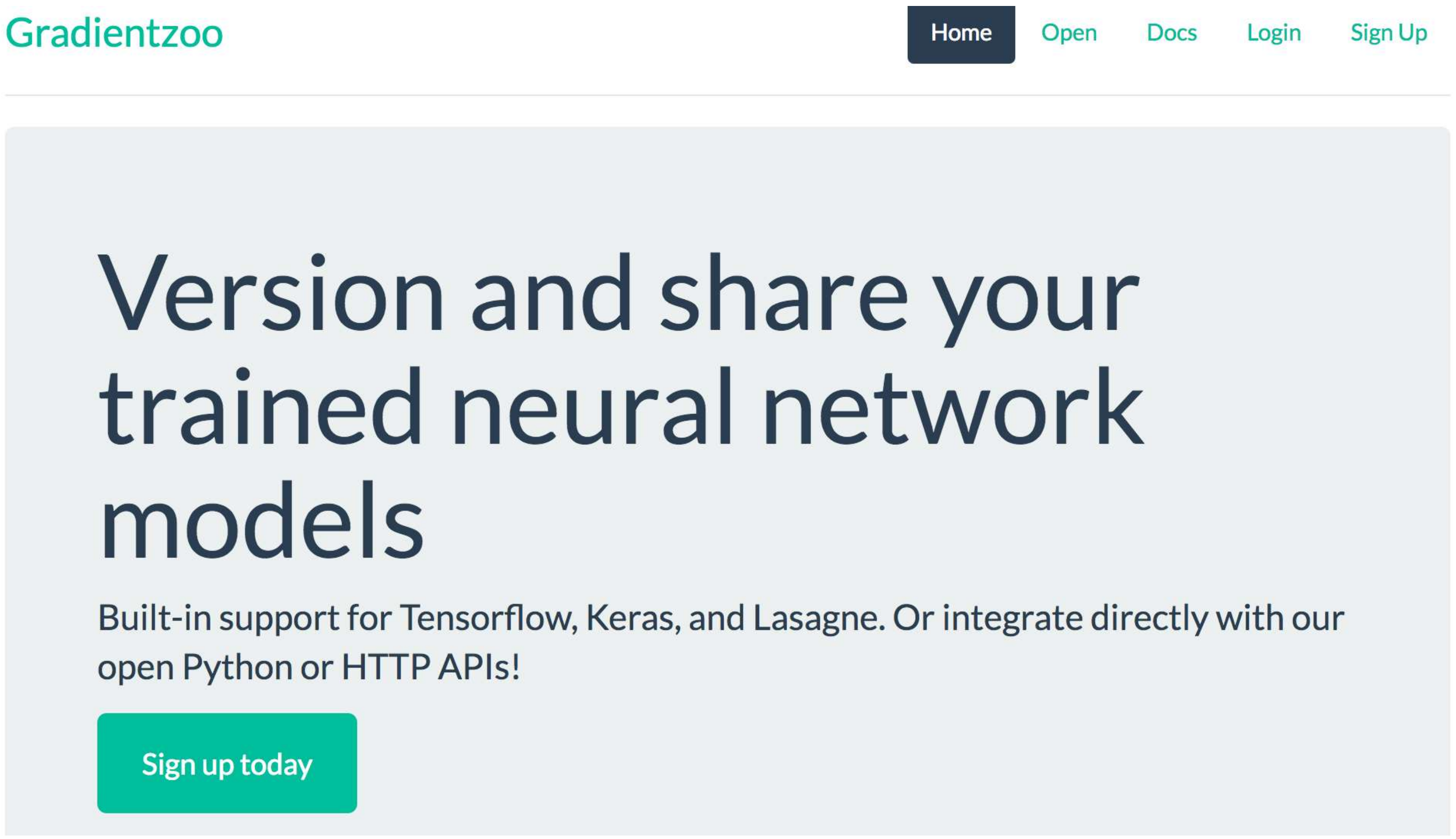
http://setosa.io/ev/image-kernels/

Excel convolution & max pooling example

TRANSFER LEARNING WITH WARM RESTART



Sharing trained networks



Version and share your trained neural network models

Built-in support for Tensorflow, Keras, and Lasagne. Or integrate directly with our open Python or HTTP APIs!

Cats vs Dogs – image classification example

[Transfer learning example – Jupyter Notebook]

Things to try

Speed of training:
- Test on a sample (for tuning of parameters, dropout rate, learning rate)
- Batch normalization ("normalization" of activation outputs - faster training). Also use this on the input layer! (instead of manual normalization). Note – if you add this to VGG, the scaling and shifting parameters must be initiated to be exactly the standard deviation and mean of the inputs, otherwise it will throw the gradient descent off on the first round.

Over/underfitting.

- Image transformations (to reduce overfitting, and boost training data variance):
  - Scan parameters for individual types of augumentation (rotation, shift, hue, scaling, flip, etc), then combine them
- Add or adjust dropout (lower if underfitting). Maybe regularization (but dropout should be enough)
- If overfitting: reduce architecture complexity
- Batch normalization (less reliance on a few nodes and reduced overfitting)
- Resnet (leaner network, faster to train)

Pseudo-labelling: once you have decent validation rate, try predicting on any additional unlabeled data and include these in the training data (circular, but effective!). Rule of thumb: 25-30% of your training batches "pseudo-labelled"

Kaggle scoring optimization: clip probabilities (avoid close to 0, 1) due to log-loss scoring

Things that are probably not good ideas:
- Converting to grayscale (information loss)
- Downscaling (but necessary with standard CNN)

We're hiring

SONAT