Convolutional Neural Nets

Oliver Dürr

Datalab-Lunch Seminar Series

Winterthur, 17 December 2014

History of CNN

Zurich University of Applied Sciences

zh aw

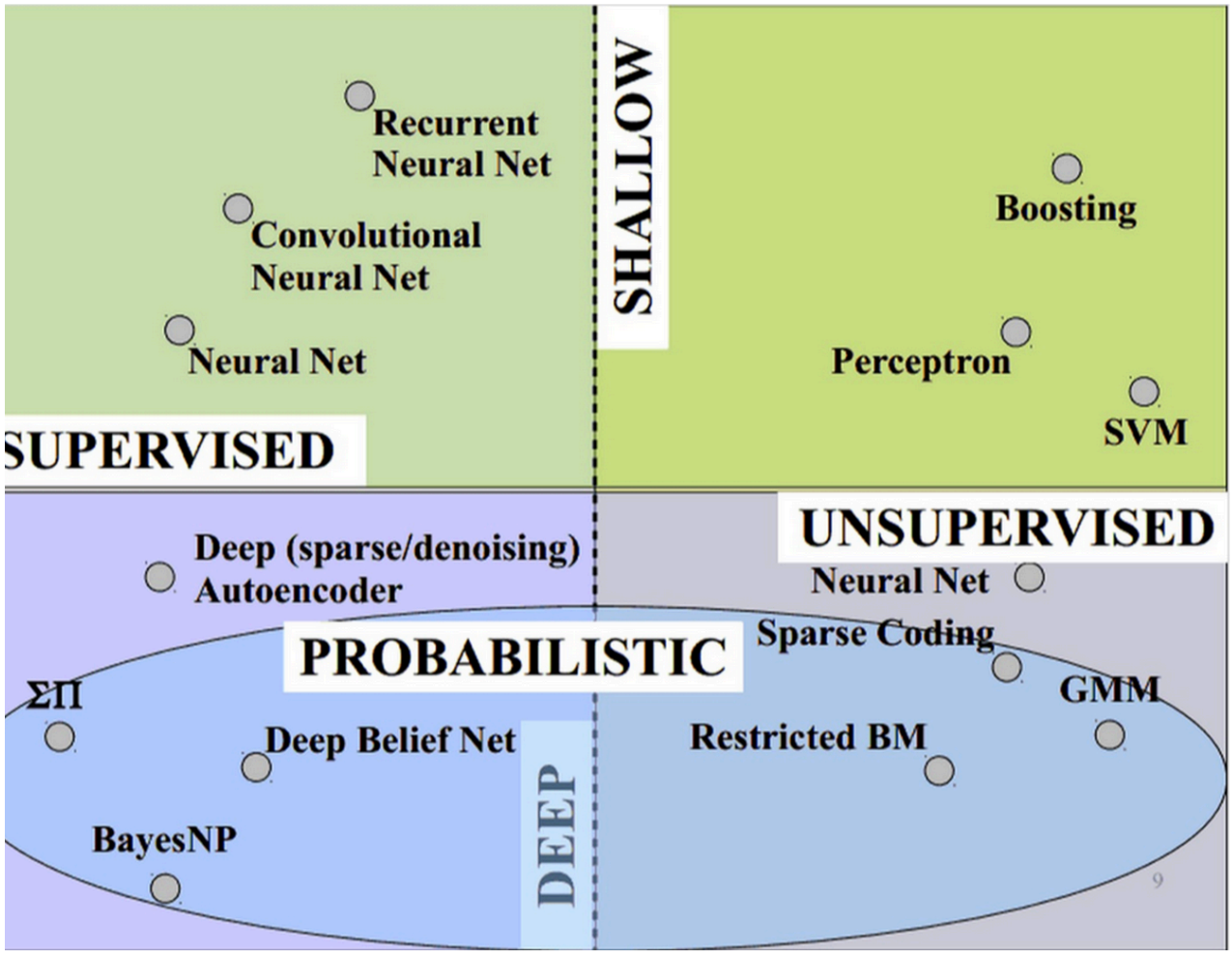
- 1980 Kunihiko Fukushima introduction
- 1998 Le Cunn (Backpropagation)
- Schmidt Huber Group many successfull implementations of CNN
  - Many Contests won
    - 2011&2014 MINST Handwritten Dataset
    - 201X CIFAR10 dataset
    - 201X Chinese Handwritten Charater
    - 2011 German Traffic Signs
    - See http://people.idsia.ch/~ciresan/
- Also other Applications besides vision
- Deep Face (2014)
  - Use partly a CNN

A map of deep learning

Zurich University of Applied Sciences

zh aw

What is Deep Learning?



Vast space of models!

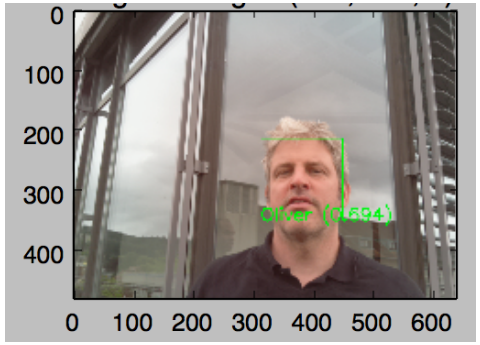
Caffe models are loss-drive
- supervised
- unsupervised

slide credit Marc'aurelio Ranza
CVPR '14 tutorial.

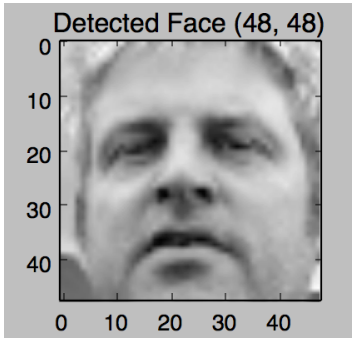
Toy example (Face Detection on a Raspberry Pi)

Zurich University of Applied Sciences

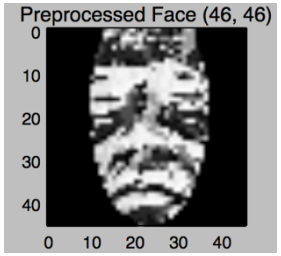
zh aw



Detection
(Viola Jones)



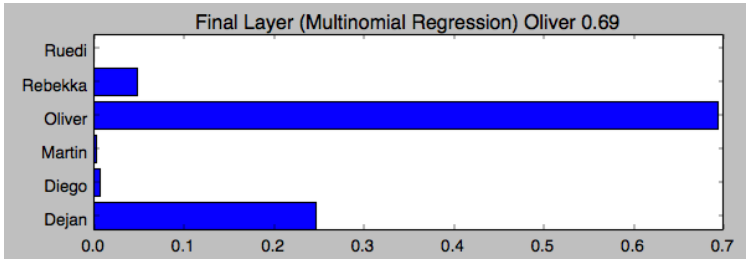
No Alignment



Preprocessed image
LBP-Operator & „Ellipse Masking“
Extreme Simple and Fast

Convolutional Neural Nets

Topic of this talk



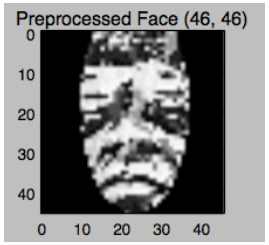
Result
mostly me
(usually better)

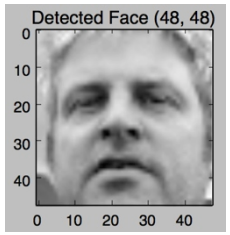
Overview of the CNN

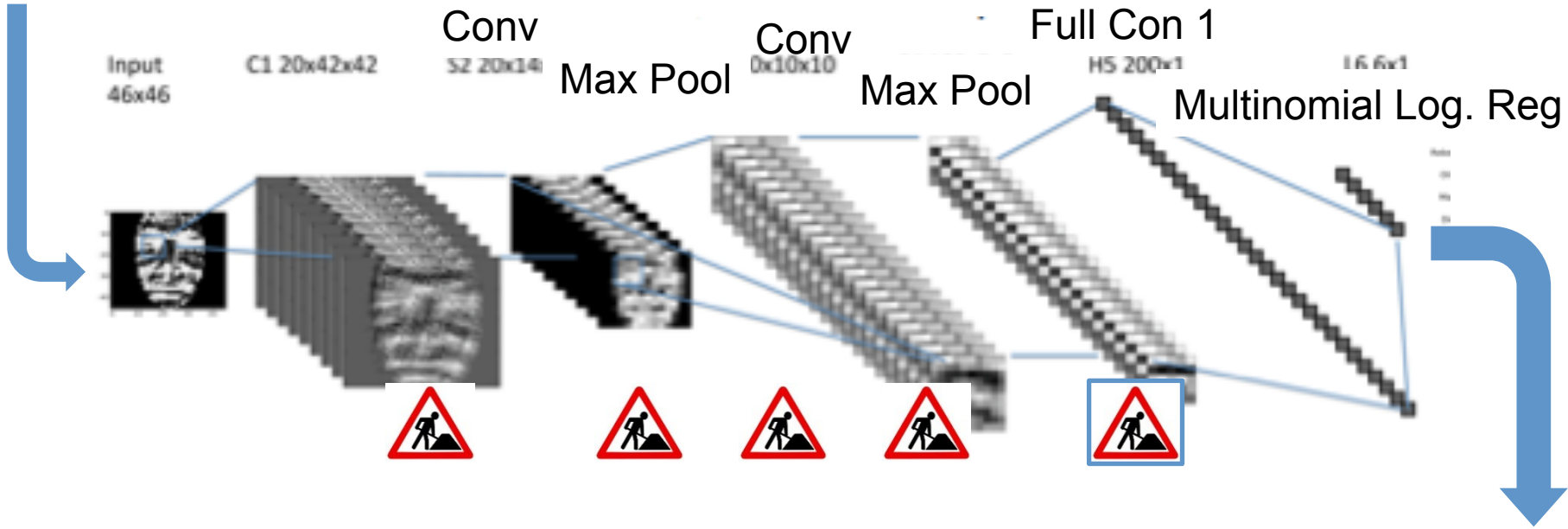
Zurich University of Applied Sciences

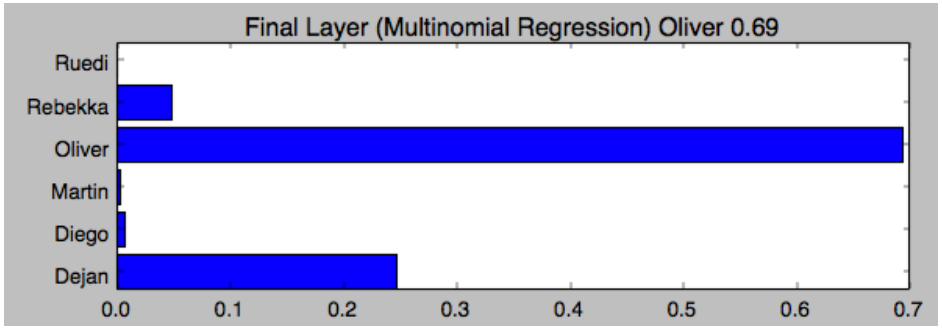
zh aw

Preproccessed Image



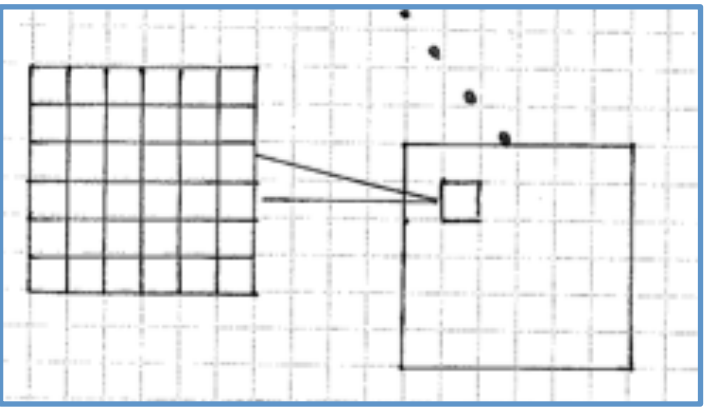




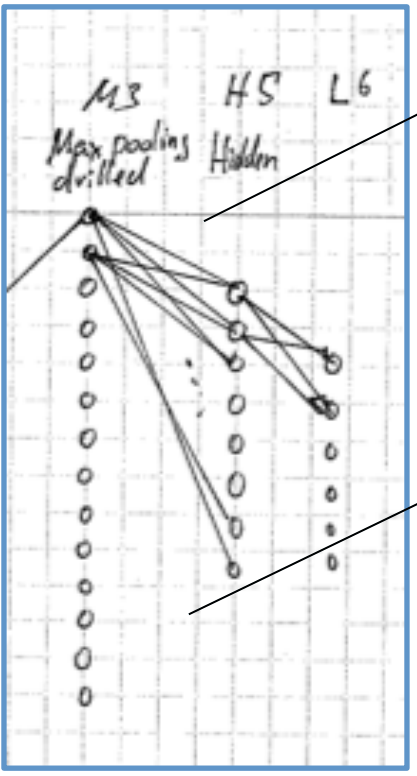


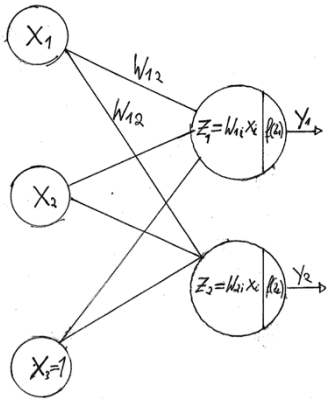
Lets look at the building blocks...

The individual building blocks



Convolution and (Maxpooling)

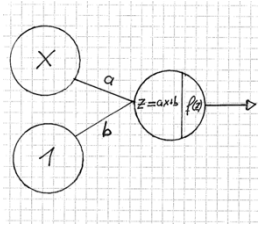




Multinominal Regression
(if last layer)

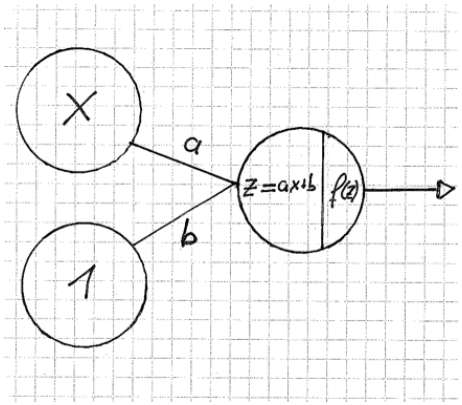
Hidden Layer / Fully Connected
(if somewhere inbetween)

We start with....



1-D Logistic Regression

Zurich University
of Applied Sciences



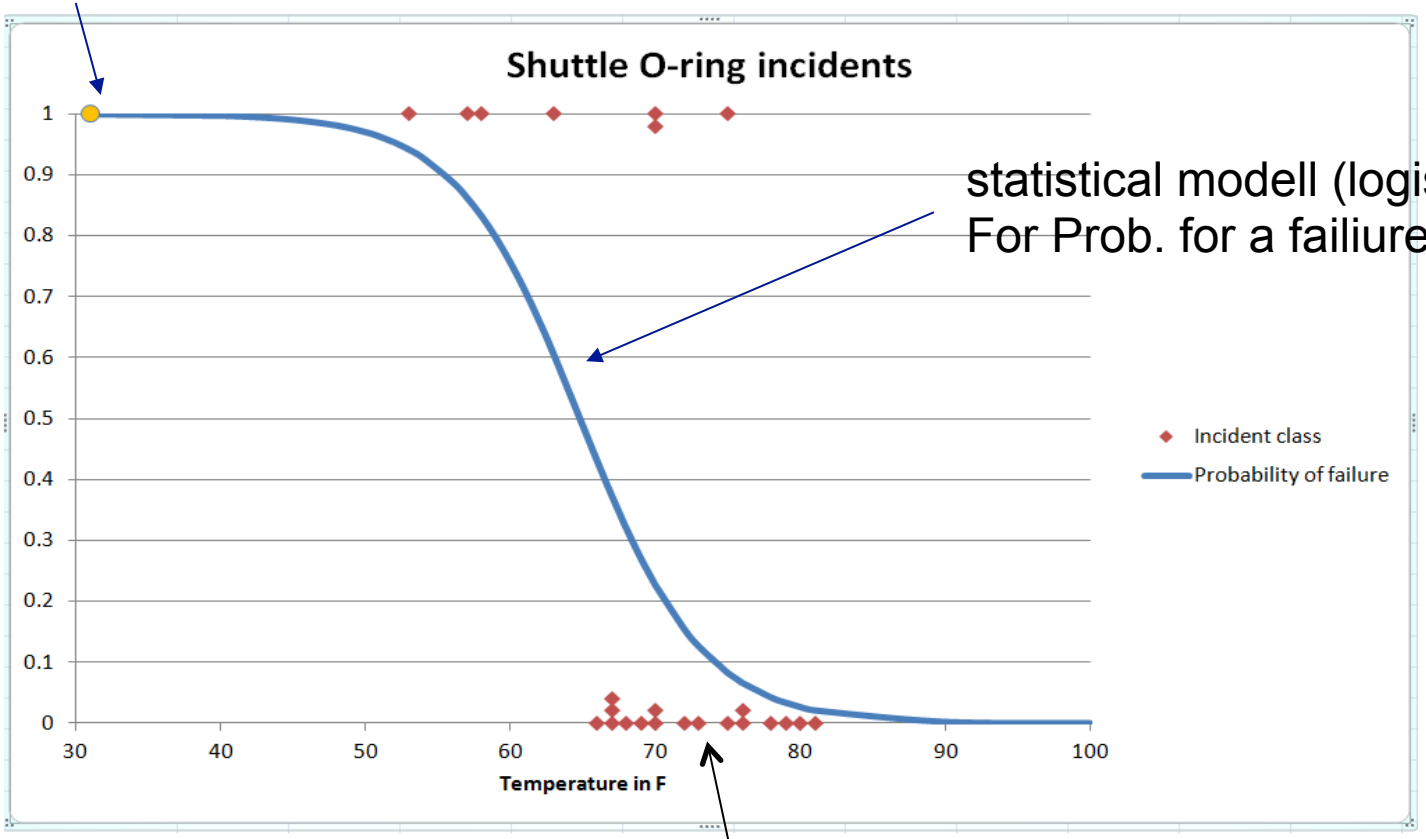
The Building Blocks: Simple Logistic Regression The mother of all nets

Logistic Regression

Predict if O-Ring is broken depending on temperature

Challenger launch @31 F

Prob. of a failure=0.9997



statistical modell (logistic regression)
For Prob. for a failiure (Y=1)

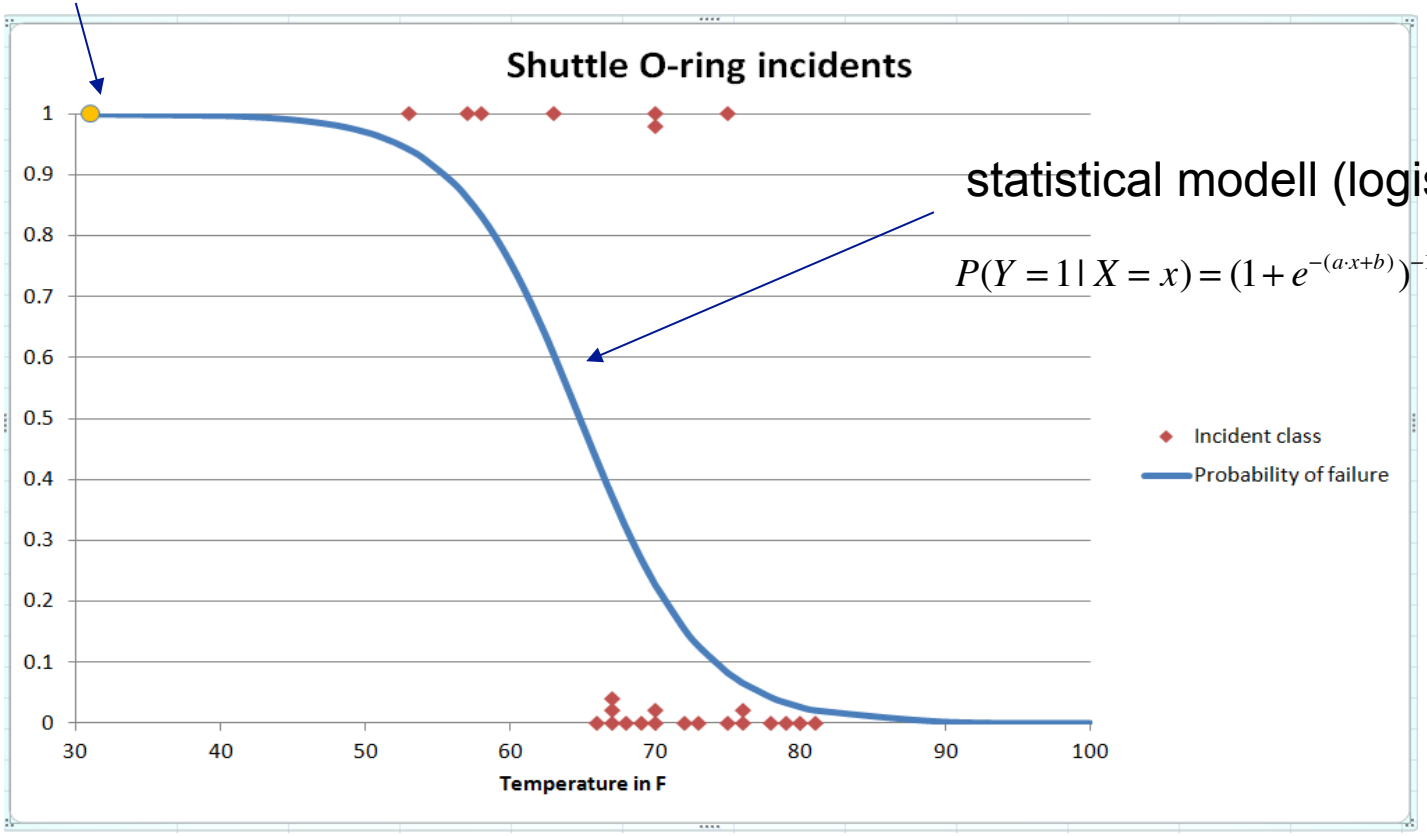
No failiure Y=0

Logistic Regression

Predict if O-Ring is broken depending on temperature

Challenger launch @31 F

Prob. of a failure=0.9997



statistical modell (logistic regression)

$$P(Y=1 \mid X=x) = (1+e^{-(a \cdot x+b)})^{-1} = (1+e^{-z})^{-1} = f(x)$$

$$z \in]-\infty,\infty[\rightarrow [0,1]$$

How do we determine the parameters θ of the model? M(θ)

Maximum Likelihood
(one of the most beautiful ideas in statistics)

Zurich University of Applied Sciences

zh aw

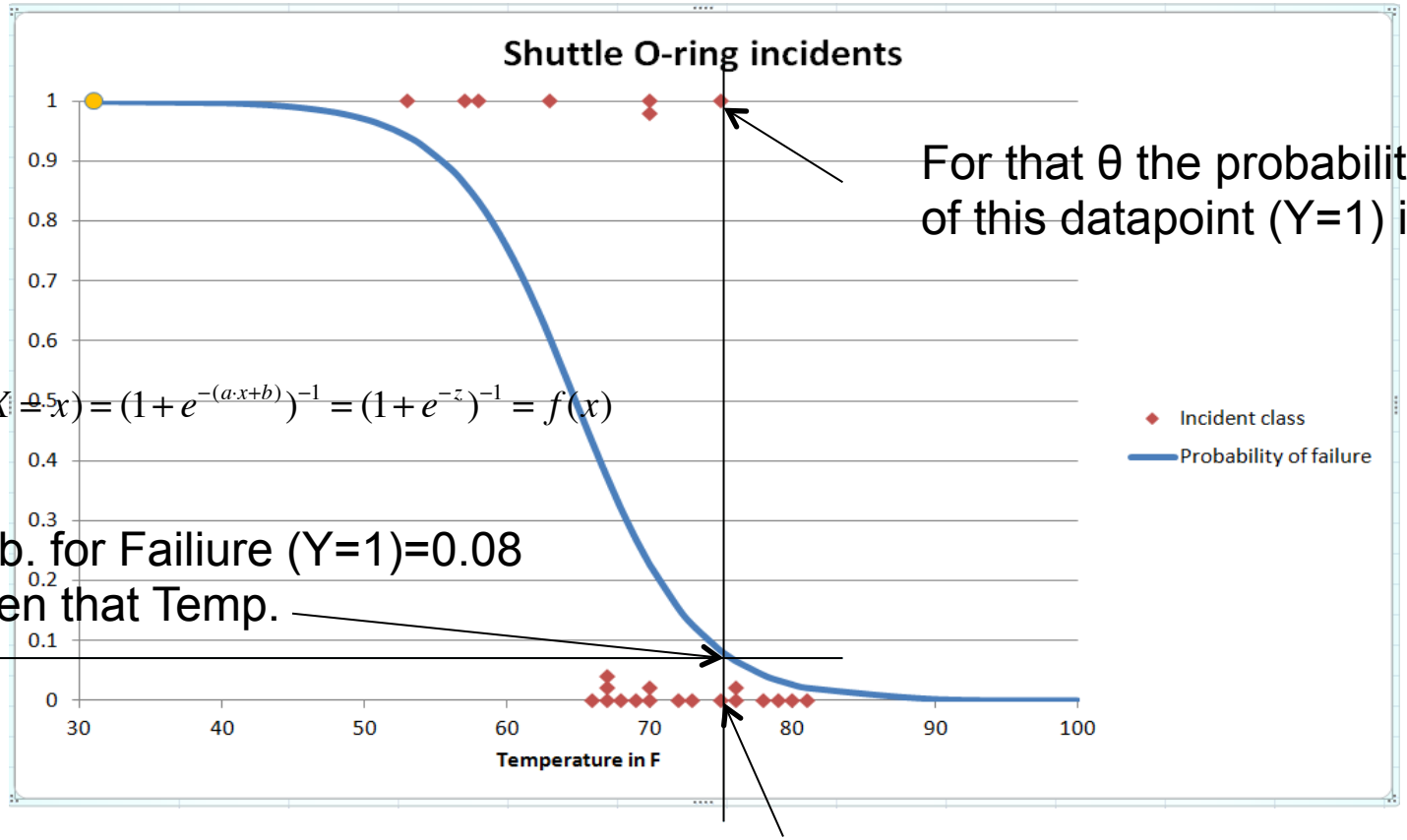
Prob. often kown

$M(\theta) \longrightarrow \text{Data}$

Tune the parameter(s) θ
so that (observed) data is most likely

What‘s the probability of the data for log. regression...

Maximum Likelihood for log. Regression



$$P(Y=1 \mid X=x) = (1+e^{-(a \cdot x+b)})^{-1} = (1+e^{-z})^{-1} = f(x)$$

For that θ the probability of this datapoint (Y=1) is p_1=0.08

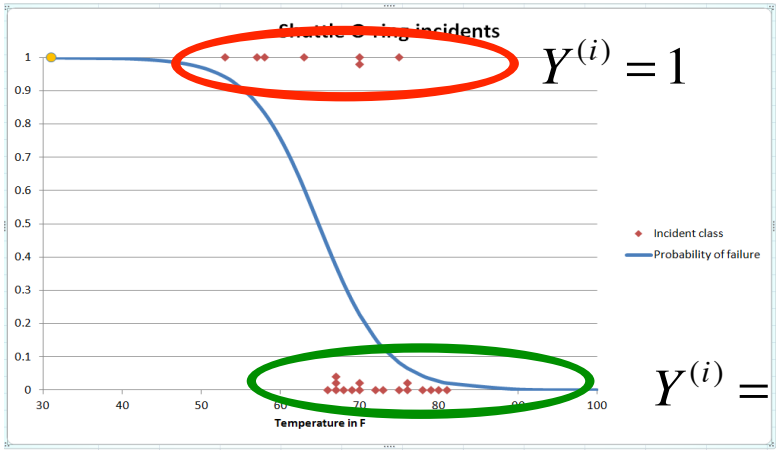
Prob. for Failiure (Y=1)=0.08
Given that Temp.

For that θ the probability of this datapoint (Y=0) is 1 - 0.08 = 92%

Prob. of all data points is the product of the individual data points. (if iid).

How to fit such a model?

Zurich University of Applied Sciences



$Y^{(i)} = 1$

$Y^{(i)} = 0$

Training Data i = 1...N

$X^{(i)}, Y^{(i)}$

$$p_1(X) = P(Y = 1 \mid X) = (1 + e^{-(a \cdot x + b)})^{-1} = (1 + e^{-z})^{-1} = f(x)$$

Probability to find Y=1 for a given values X (single data point) and a,b

$p_0(X) = 1 - p_1(X)$ Probability to find Y=0 for a given value X (single data point)

Likelihood (probability+ of the Trainingset given the parameters)

$$L(a,b) = \prod_{i \in All\ ones} p_1(x^{(i)}) * \prod_{i \notin All\ Zeros} p_0(x^{(j)})$$

Let‘s Maximize this Probability

Maximizing the Likelihood

Zurich University of Applied Sciences

zh aw

Likelihood (prob of a given Trainingset) want to maximized wrt. parameters

$$L(a,b) = \prod_{i \in All\ ones} p_1(x^{(i)}) \quad * \quad \prod_{i \notin All\ Zeros} p_0(x^{(j)})$$

Taking log (maximum of log is at same position)

$$-nJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in All\ ones} \log(p_1(x^{(i)})) \quad * \quad \sum_{i \in All\ zeros} \log(p_0(x^{(i)})) = \sum_{i \in All\ Training} y_i \log(p_1(x^{(i)})) + (1-y_i)\log(p_0(x^{(i)}))$$

Gradient Descent for Minimum of J

$$\theta_i' \leftarrow \theta_i' - \alpha \frac{\partial J(\theta)}{\partial \theta_i}$$

Takeing the Gradient effectively with backpropagation (or using Theano).

In Theano

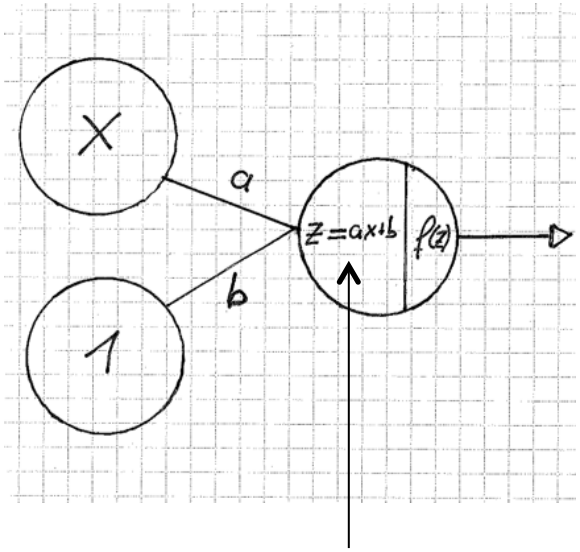
```
p_1 = 1 / (1 + T.exp(-T.dot(x, w) - b))        # p_1 = p(y=1 | x)
like = y * T.log(p_1) + (1-y) * T.log(1-p_1)   # Log likelihood
cost = -like.mean()
gw, gb = T.grad(cost, [w, b])
```

Logistic Regression in the neural net speak

Zurich University of Applied Sciences

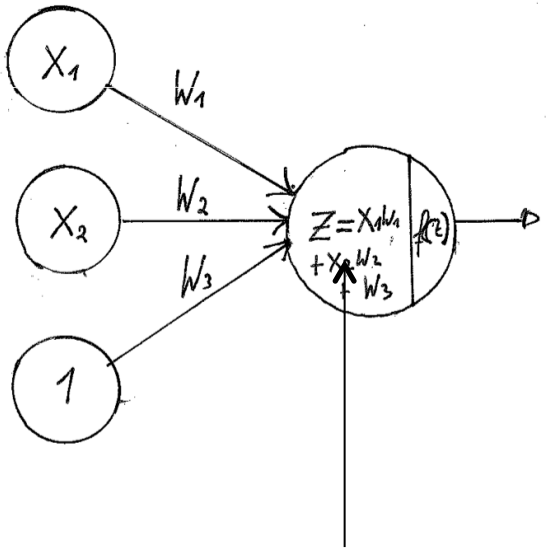
zh aw

1-D log Regression



$$z = ax + b$$

N-D log Regression



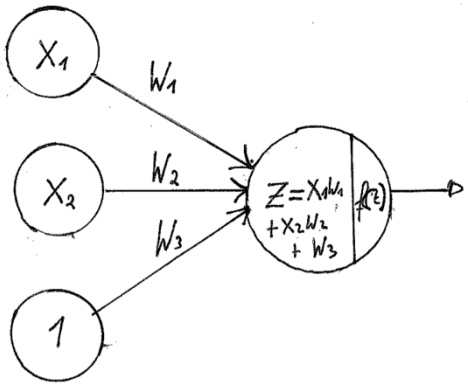
$$z = x_1 W_1 + x_2 W_2 + W_3 = \theta^T x$$

$$P(Y = 1 \mid X = x) = [1 + \exp(-\theta^T x)]^{-1} = \frac{\exp(\theta^T x)}{1 + \exp(\theta^T x)}$$

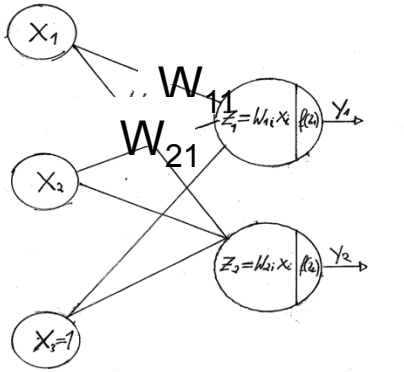
Multinomial Regression (Definition)

Zurich University of Applied Sciences

Logistic Regression
N-Inputs 1-Output



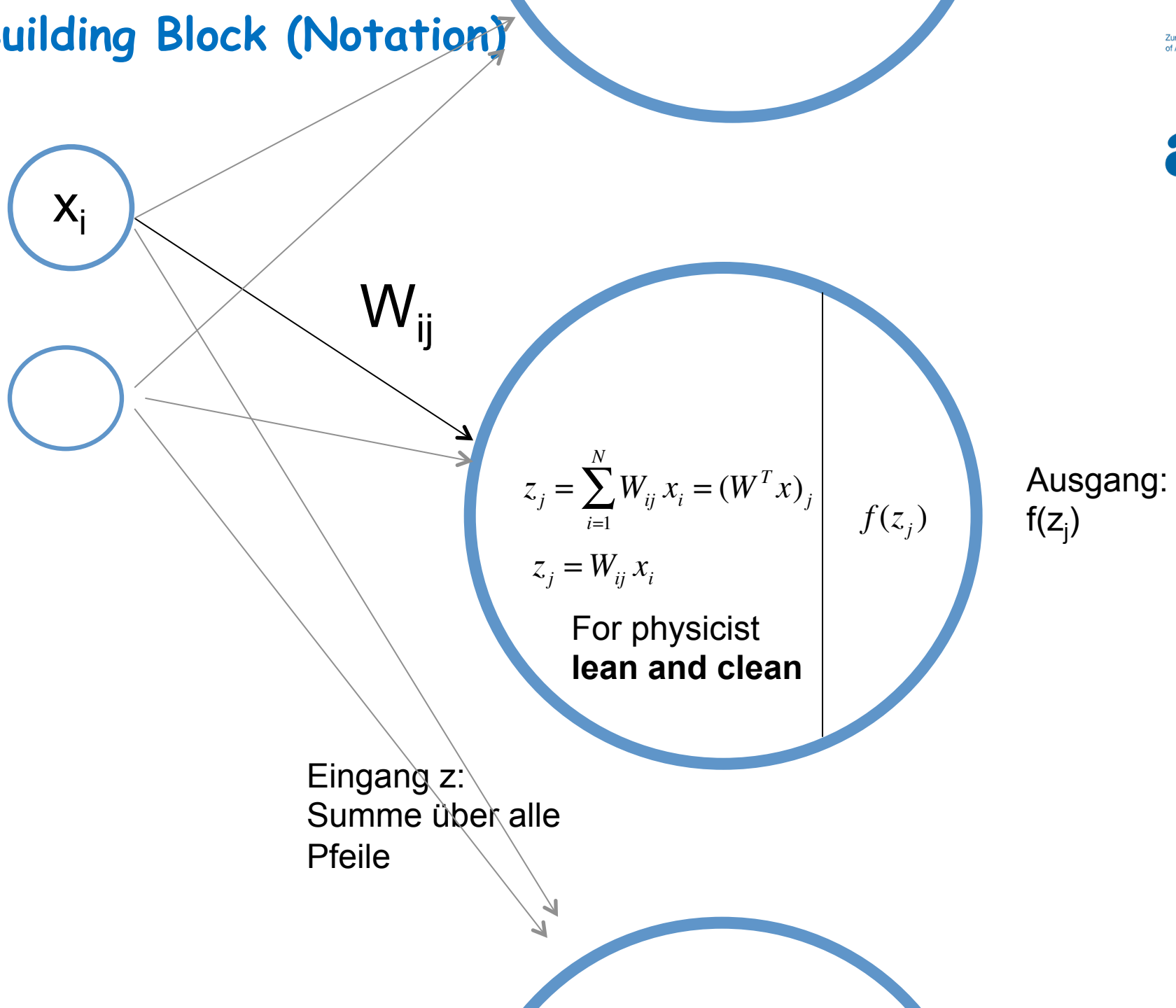
Mulitinomial Logistic Regression
N-Inputs M-Outputs



Mutlinominal Regression aka

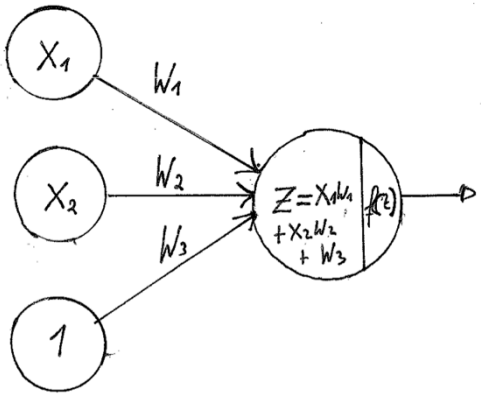
- multiclass LR, **softmax regression**, multinomial logit, maximum entropy (MaxEnt)

Building Block (Notation)



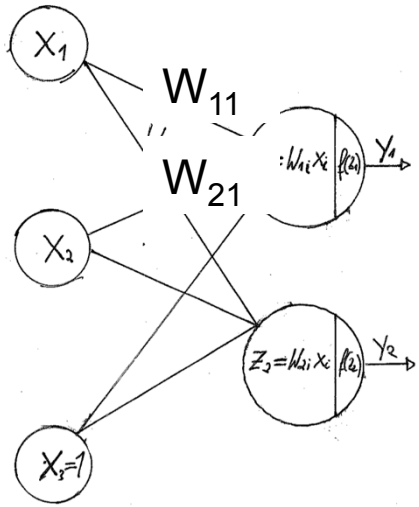
Multinominal Regression (Likelihood)

Zurich University of Applied Sciences



Binary Case

$$P(Y=1 \mid X=x) = \frac{\exp(W_i x_i)}{1+\exp(W_i x_i)}$$



More than one class

Normalisation

$$\sum_{i=1} p_i = 1$$

$$p_1 = P(Y_1 = 1 \mid X = x) \propto \exp(W_{i1} x_i) \qquad p_1 = \frac{\exp(W_{i1} x_i)}{\sum_j \exp(W_{ij} x_i)}$$

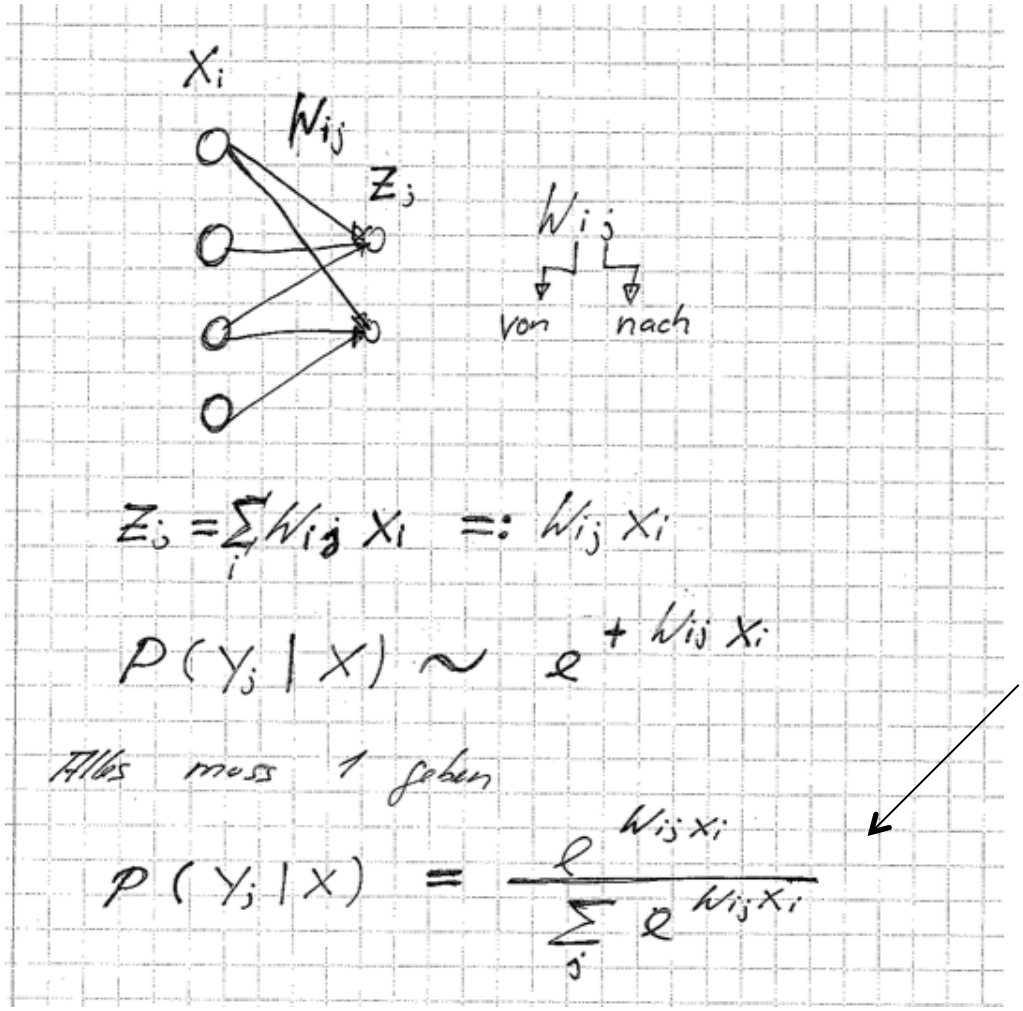
$$p_2 = P(Y_2 = 1 \mid X = x) \propto \exp(W_{i2} x_i)$$

Function to Maximize (prob.)

$$-nJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in y_j=1} \log(p_1(x^{(i)})) + \sum_{i \in y_j=2} \log(p_2(x^{(i)})) + \ldots + \sum_{i \in y_j=N} \log(p_N(x^{(i)}))$$

Sum over all Trainingexamples with Y=1 Y=2 Y=N

Multinominal Regression (Likelihood)



Binary Case

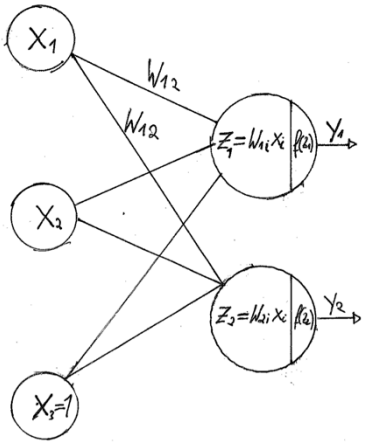
$$P(Y=1 \mid X=x) = [1+\exp(-W^T x)]^{-1} = \frac{\exp(W^T x)}{1+\exp(W^T x)}$$

Generalisation to more than two cases

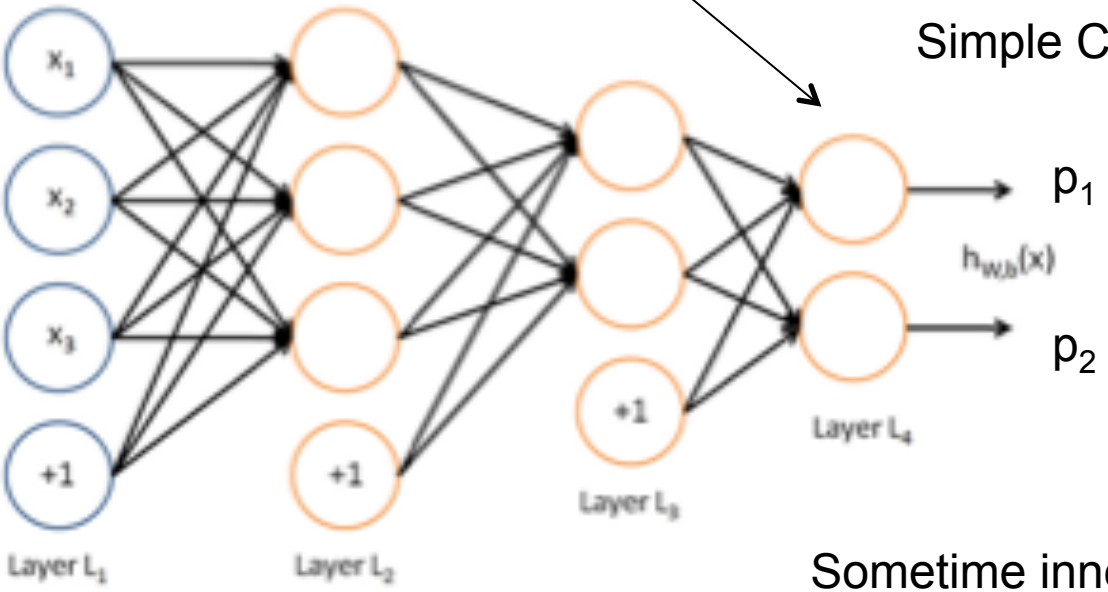
Function to Maximize (prob.)

$$-nJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in y_j=1} \log(p_1(x^{(i)})) + \sum_{i \in y_j=2} \log(p_2(x^{(i)})) + \ldots + \sum_{i \in y_j=N} \log(p_N(x^{(i)}))$$

More than one layer: Full Neural Network



$$X_j^{(\ell)} = f\Big(\underbrace{W_{j'j}^{(\ell-1)} f\big(W_{j''j'}^{(\ell-2)} f\big(W_{j'''j''}^{(\ell-3)} f(\cdots f(W_{i\,j''''} X_i^{(1)}))\big)\big)}_{Z_j^{(\ell-1)}}\Big)$$



Simple Chaining

Sometime inner layers have different activation functions (than Softmax) i.e. tanh.

Fitting Multi. Logistic Regression Likelihod

Zurich University of Applied Sciences

zh aw

Function to Maximize

$$-nJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in y_j = 1} \log(p_1(x^{(i)})) \quad + \sum_{i \in y_j = 2} \log(p_2(x^{(i)})) + \ldots + \sum_{i \in y_j = N} \log(p_N(x^{(i)}))$$

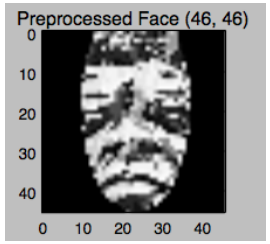
Same prodecure as before

$$\theta_i' \leftarrow \theta_i' - \alpha \frac{\partial J(\theta)}{\partial \theta_i}$$

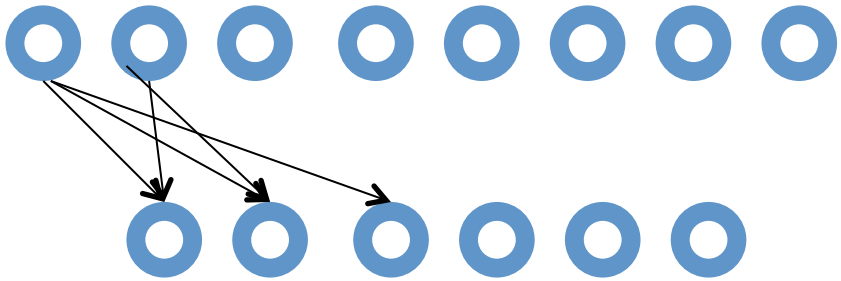
Example Output

Zurich University of Applied Sciences

zh aw

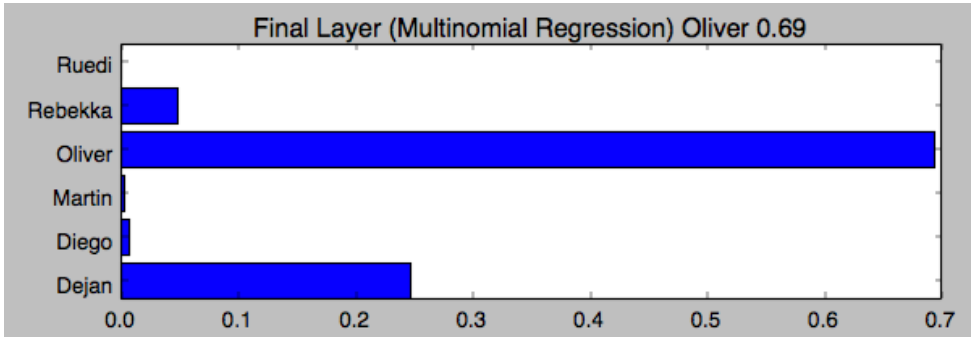


Still as Some magic



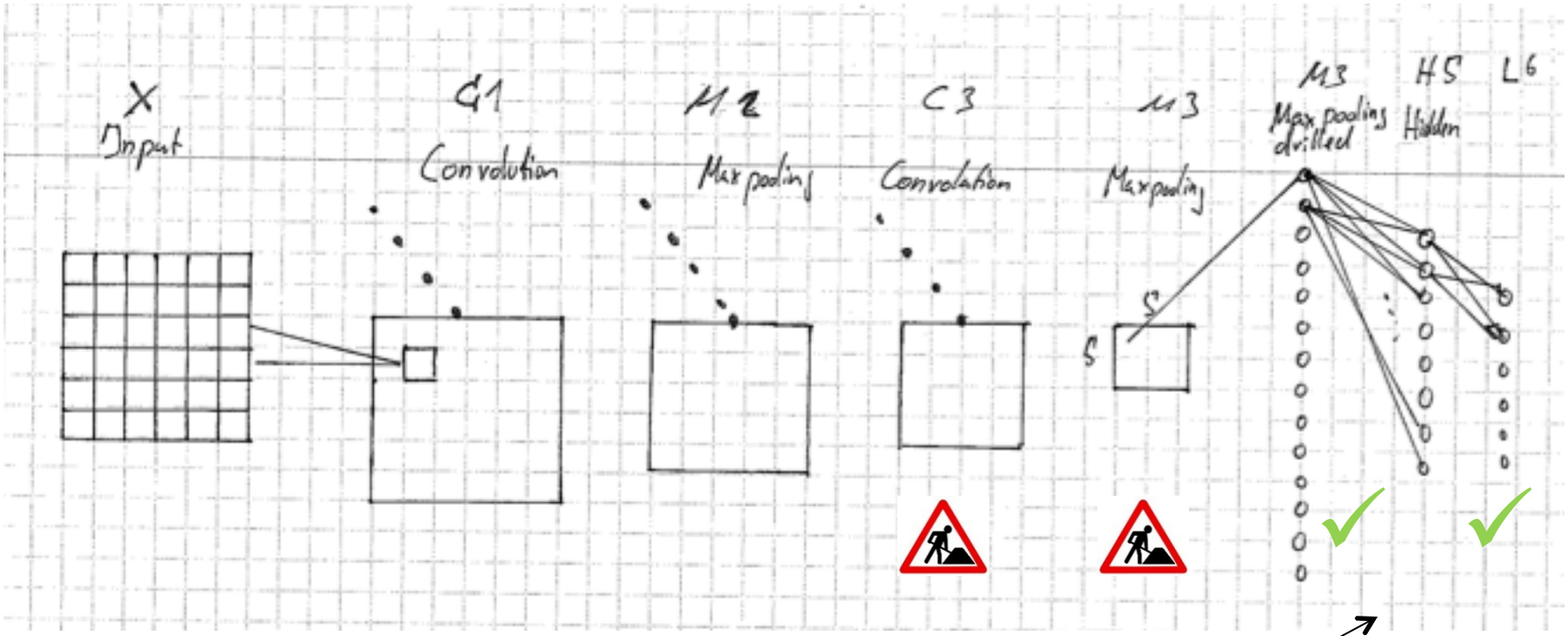
Input of Logistic Regression (200)

Output of Logistic Regression 6



$$P(Y_j \mid X) \sim e^{+ b_{ij} x_i}$$

Looking at the individual parts



Hidden Layers and Mult. Regression

Zurich University
of Applied Sciences

zh
aw

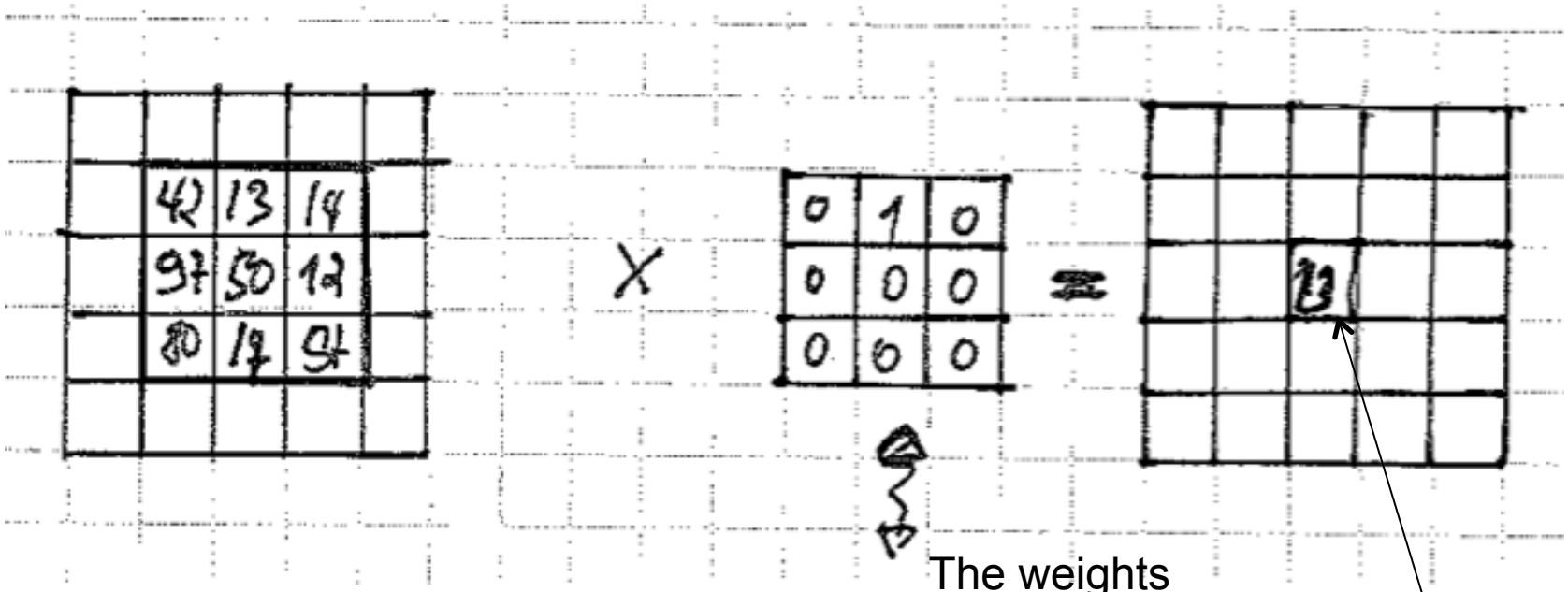
The convolutional part

The convolutional layer
Ingredient I: Convolution

Zurich University of Applied Sciences

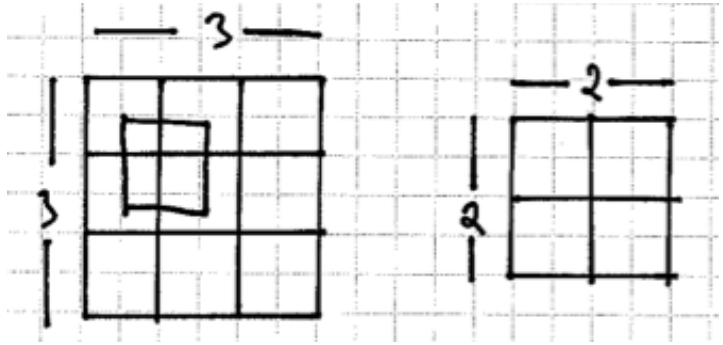
zh aw

What is convolution?



The weights W_{ij} here a.k.a. Kernels

That‘s a 13

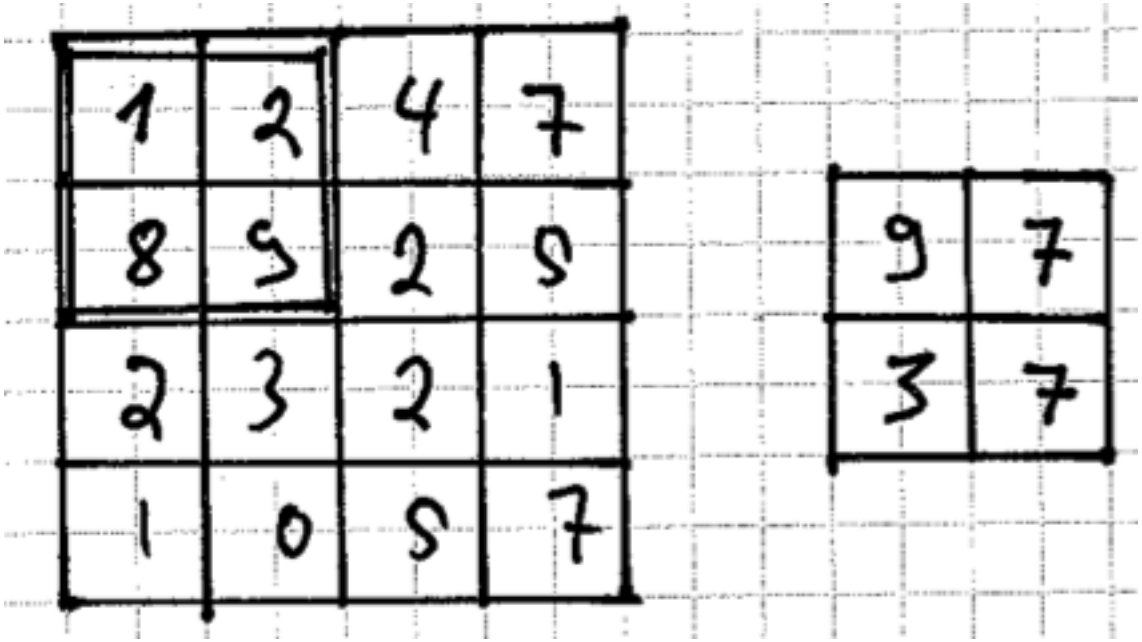


Dimension get‘s smaller

The convolutional layer
Ingredient II: Max-Pooling

Zurich University of Applied Sciences

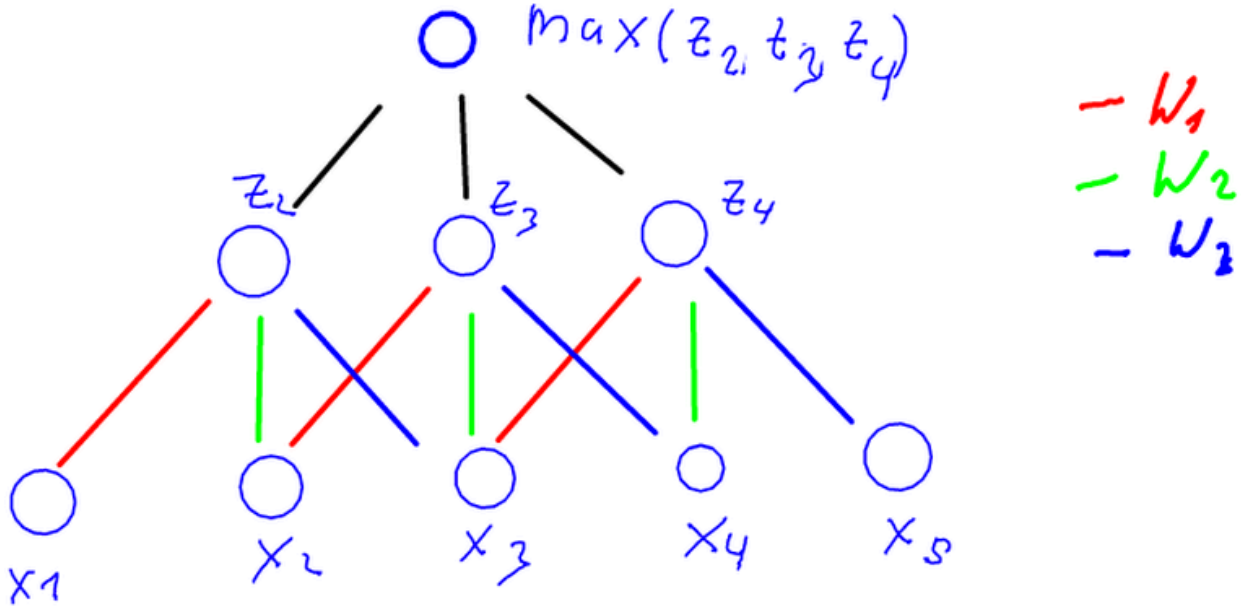
zh aw



Simply join e.g. 2x2 adjacent pixels in one.

Idee behind: Sparse Connectivity & Weight Sharing

Zurich University of Applied Sciences

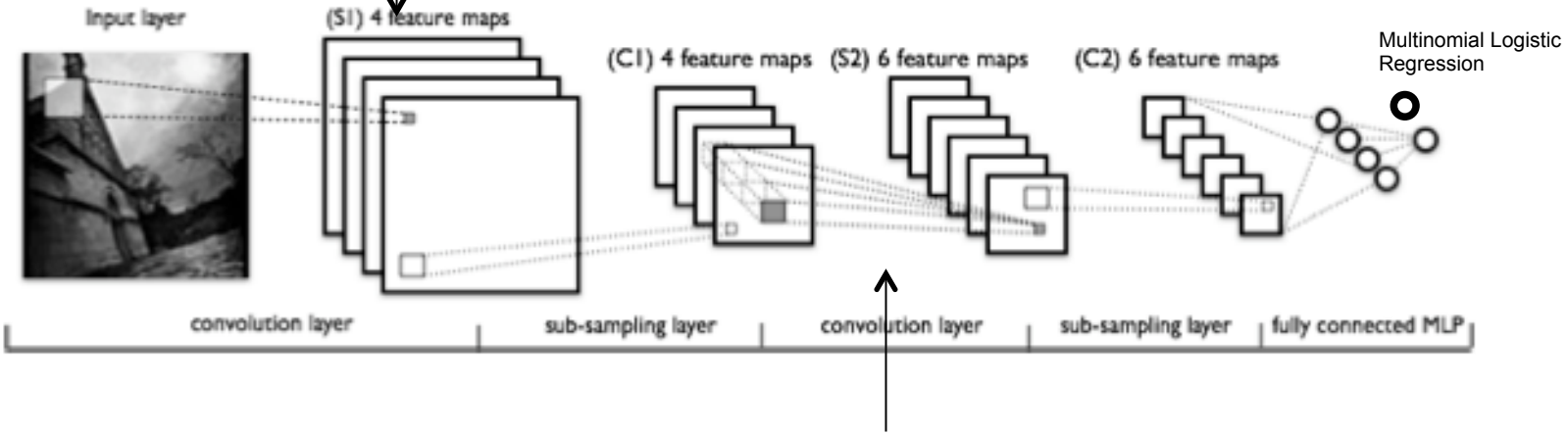


Weight sharing and reduction.

Now lets play bob the builder (Faces Simple)

The Network as used in the Raspberry Pi experiment.

More than one kernel



Averaged from the all the other kernels („fully connected“)

Now lets play bob the builder (DeepFace)

Preprocessing (not part of this talk)
2 Phase Alignment

The Deep Face Network

Convolutional / Pooling Layers

Locally Conected

Fully connected

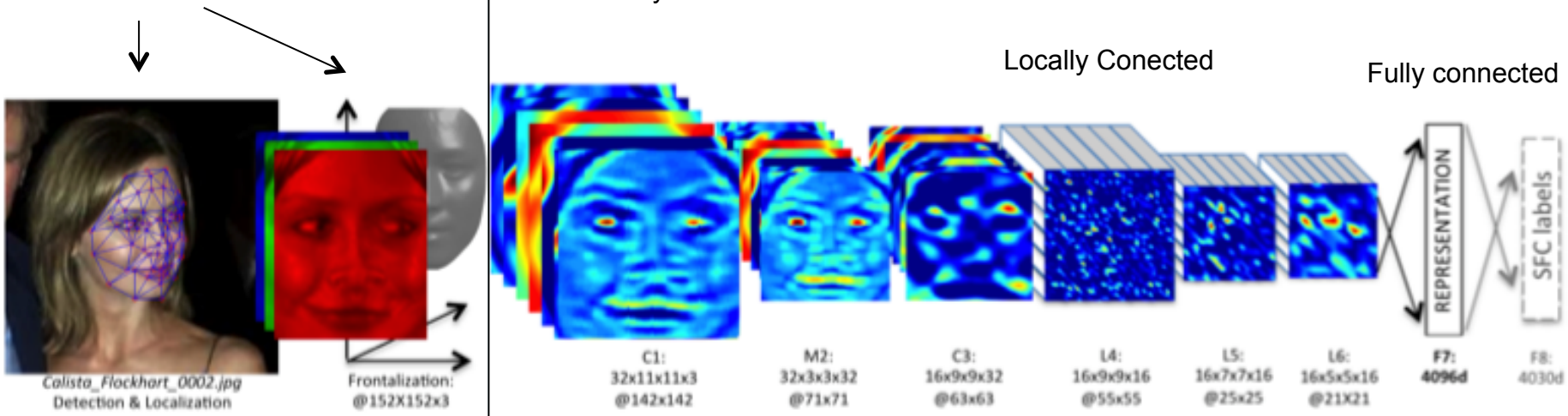
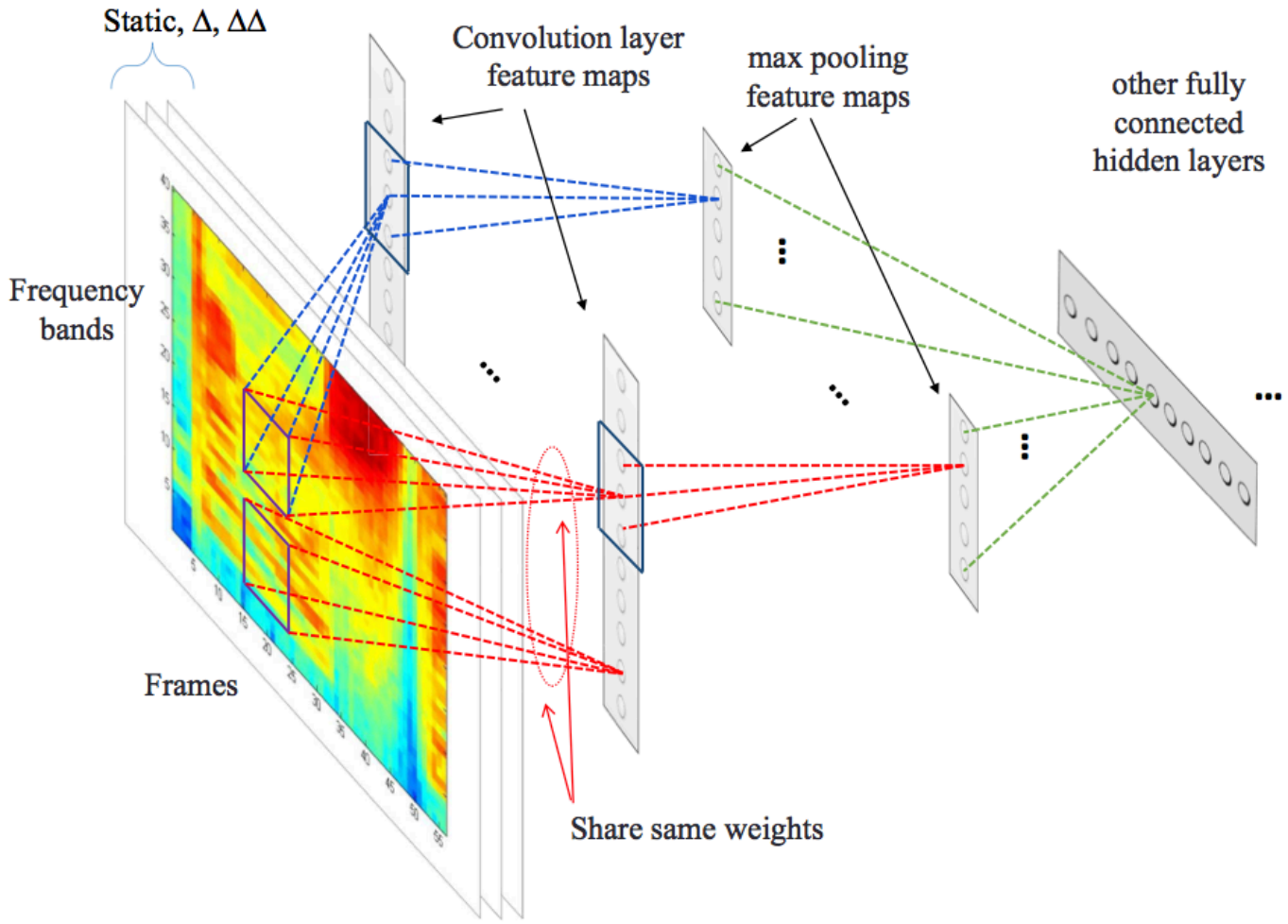


Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Now lets play bob the builder (Audio)



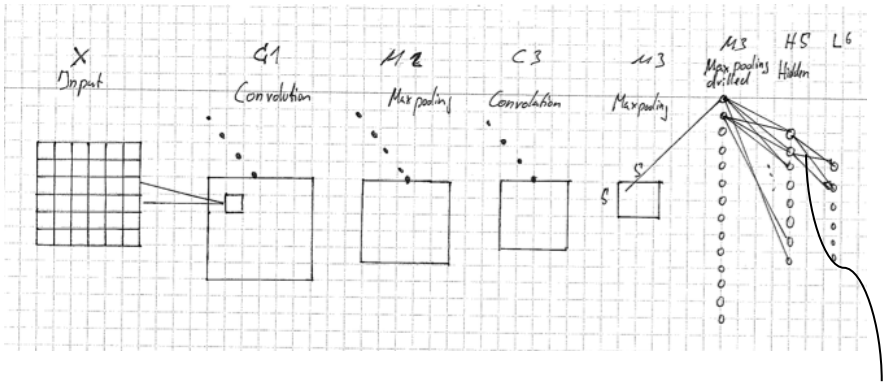
From Hamid et all
Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition

Lerning

Zurich University of Applied Sciences

zh aw

Testimages
x



Input from second but last layer

$$-nJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in y_j = 1} \log(p_1(x^{(i)})) \quad + \sum_{i \in y_j = 2} \log(p_2(x^{(i)})) + ... + \sum_{i \in y_j = N} \log(p_N(x^{(i)}))$$

$$\theta_i' \leftarrow \theta_i' - \alpha \frac{\partial J(\theta)}{\partial \theta_i}$$

Complicated function taking an Image X and returning p_i i=1,...,Number of People

The example

Zurich University of Applied Sciences

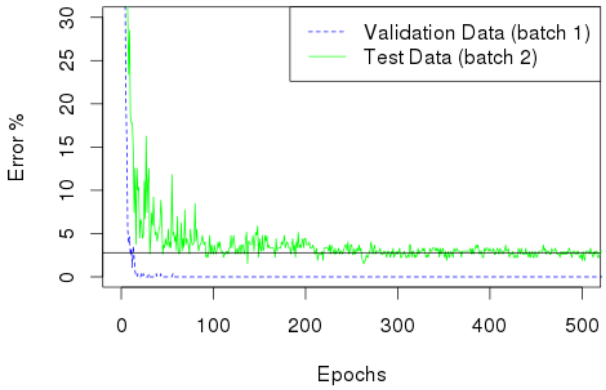
zh

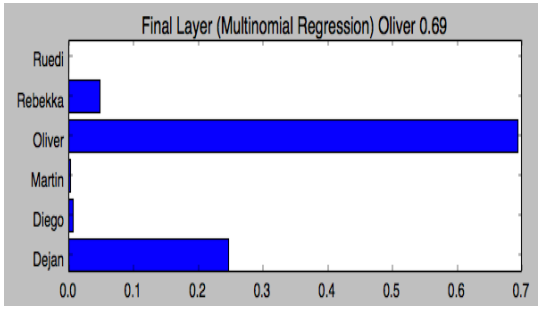
Batch 1 ~ 300 images

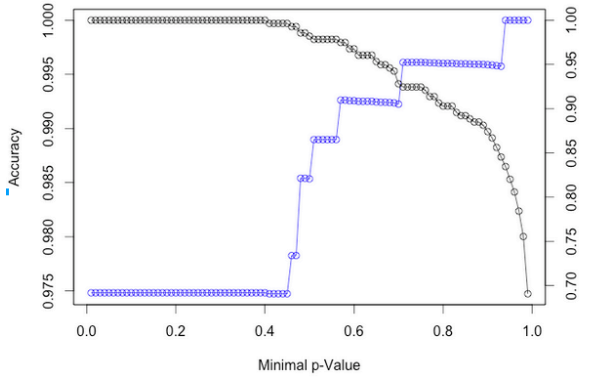
Training
Deformed image

Validation
On original images

Batch 2 ~ 310 images
Testing





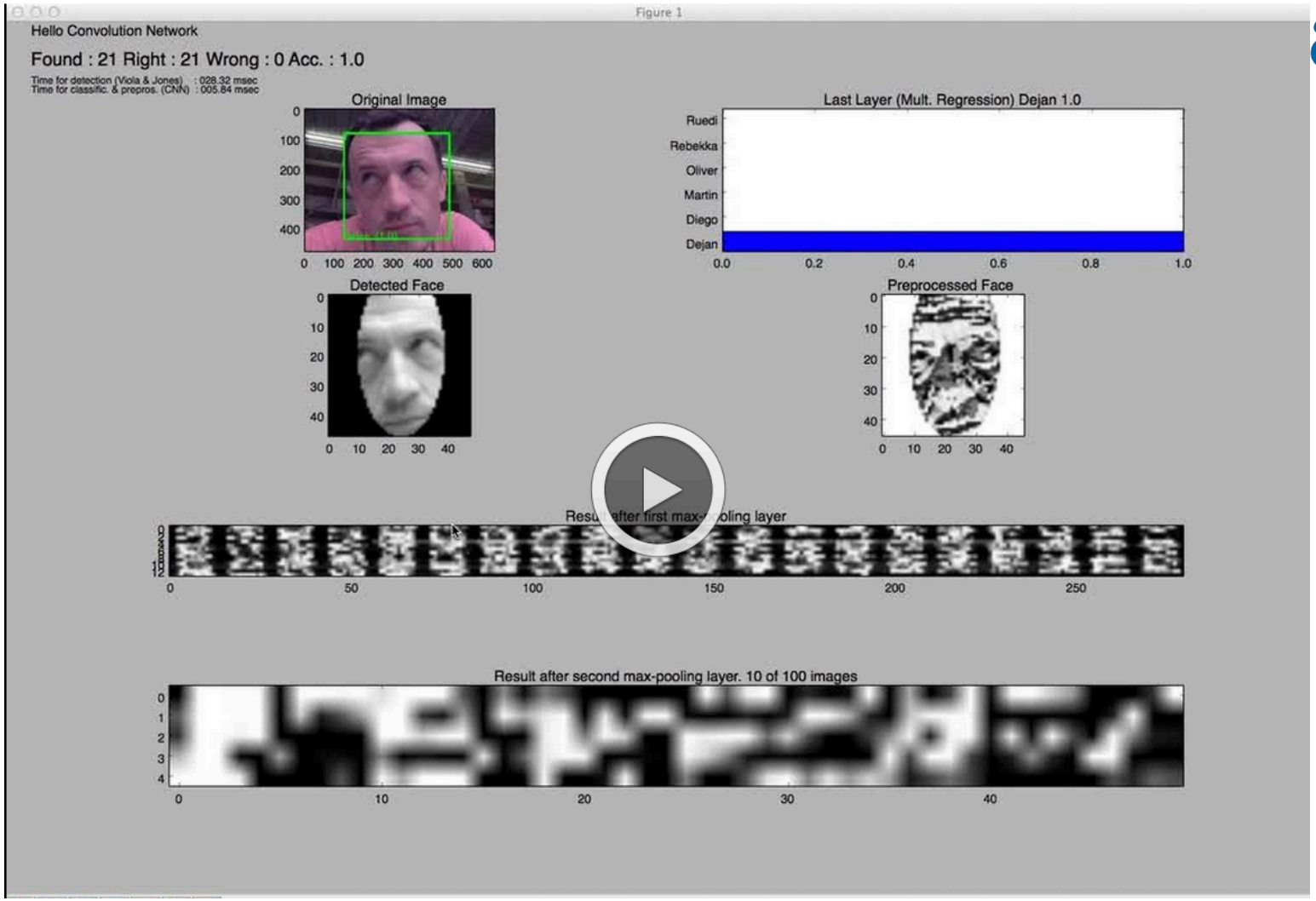


On Raspberry Pi Model B
Feature extraction and classifcation in approx 100 msec (600 msec openCV)

Let's watch a movie

Zurich University of Applied Sciences

zh aw



https://www.youtube.com/watch?v=ol1eJa-UWNU&feature=youtu.be

Software for CNN

Zurich University of Applied Sciences

zh aw

- Theano used so far
  - Code at https://github.com/Oliver4242/dl-playground/tree/master/python/FaceRec

- Other DL Software
  - NVIDEA cuDNN
  - Caffe (http://caffe.berkeleyvision.org/)
    - Integrated cuDNN
    - Pure C++ / CUDA architecture for deep learning
    - command line, Python, MATLAB interfaces
    - Link1

Thanks