# Learning from Data
# Lecture 5: Neural Networks II
# Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Introduce how relates to previous week's lecture
- Mention not my slides

# Lecture overview

- Neural Networks
  - Background
  - Intuitions
  - Technicalities
- Deep Learning
  - Overview and examples

*Many slides based on the Coursera ML slides by Andrew Ng*





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Introduce how relates to previous week's lecture
- Mention not my slides

# Background

- Attempts to mimic the brain
- Widely used in the '80s and '90s
- Recent resurgence in popularity





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Mimicry  cartoon like, it's not clear to what degree artificial neural network mirror brain function
- Earliest origins (40s)
- Why resurgence? Successes, pre-training, big data, more computation

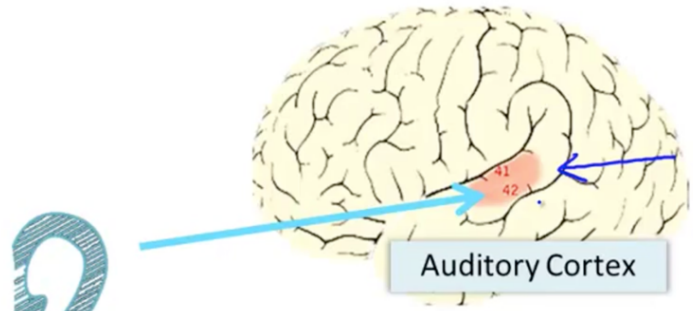
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# The Power of the Brain

## The "one learning algorithm" hypothesis




Auditory cortex learns to see

[Roe et al., 1992]

- Introduce that brain can do all kinds of things, on a single piece of hardware
  Two options:
  it has a lot of different programs for different tasks, or has one very good learning algorithm.
  The last one is most widely believed, e.g. look at these experiments.
- Animal experiments, naturally

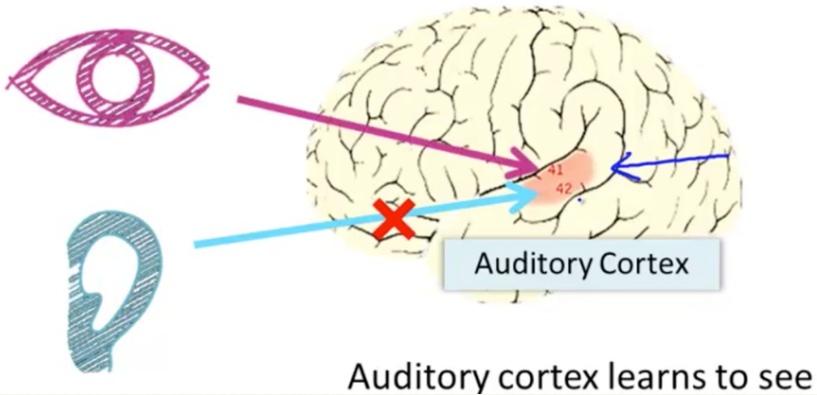
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# The Power of the Brain

## The "one learning algorithm" hypothesis




[Roe et al., 1992]

- Introduce that brain can do all kinds of things, on a single piece of hardware
  Two options:
  it has a lot of different programs for different tasks, or has one very good learning algorithm.
  The last one is most widely believed, e.g. look at these experiments.
- Animal experiments, naturally

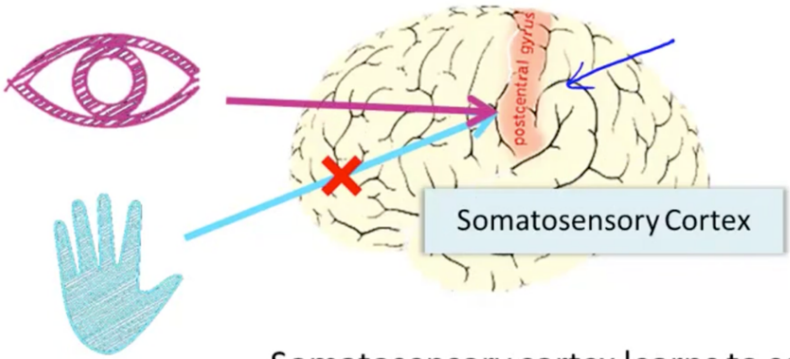
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# The Power of the Brain

**The "one learning algorithm" hypothesis**




[Metin & Frost, 1989]

- Neural rewiring experiments
- No need for thousands of programs  Implement the universal algorithm

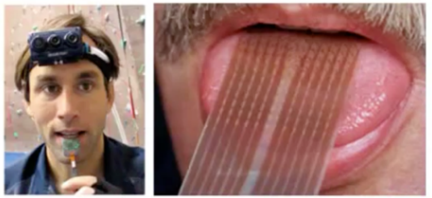
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

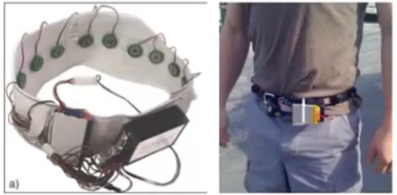
# The Power of the Brain

**Sensor representations in the brain**



Seeing with your tongue

Human echolocation (sonar)



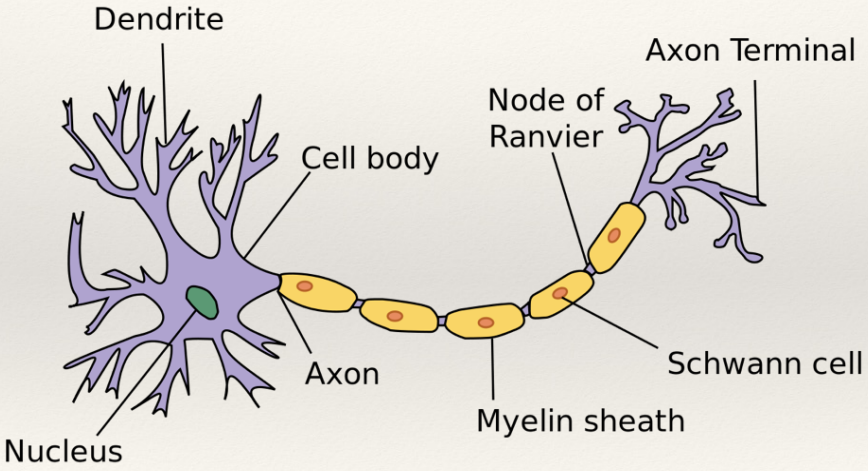
Haptic belt: Direction sense

Implanting a 3rd eye

[BrainPort; Welsh & Blasch, 1997; Nagel et al., 2005; Constantine-Paton & Law, 2009]

- Brain port - grayscale low resolution camera, input via tongue
- Echolocation - clap hands, get echo, 'seeing' via ears
- Haptic belt - Like birds sensing north, give humans direction sense
- Take-away: Any sensor plugged into the brain will be processed and learned
- What is the brains learning algorithm?

# Biological basis








Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

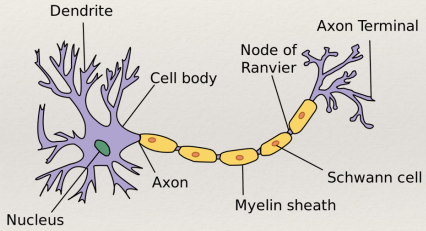
Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Go through parts
- Simulation of neurons
- Dendrites are inputs, Axons are outputs
- This hardware powers all human thought, senses, muscles etc.

# Biological basis

- Biologically inspired
- … but not very realistic








Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Several things not taken into account:
- discrete firing (bio) vs. value outputs (artificial)
- Not clear how good the approximation is
- More of a cartoon representation

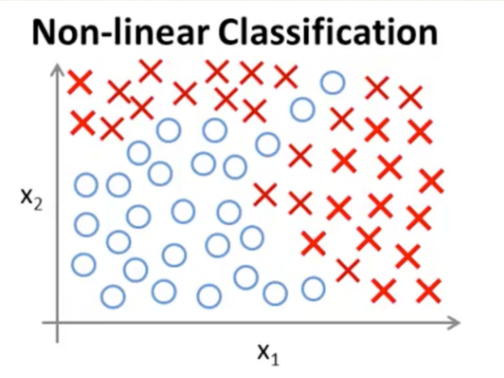
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

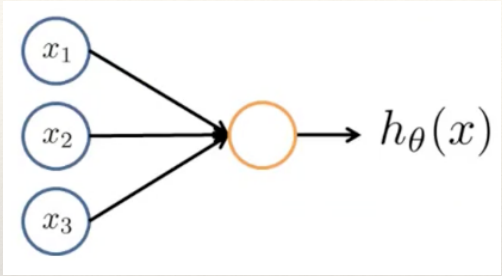
12th of December 2016

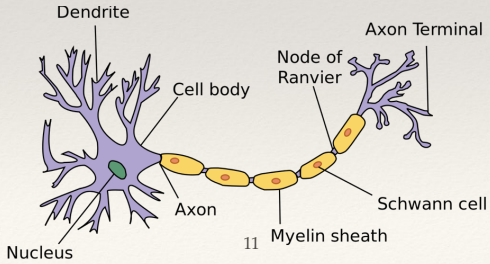
# Why Neural Networks?

- Non-linearity
- Power
  - Can represent *any* function



# Neuron model








Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

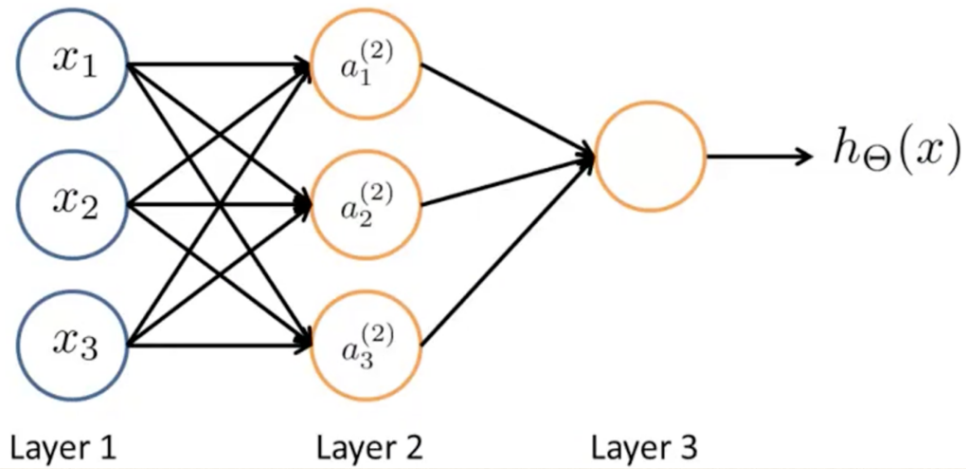
- Go through parts, point out equivalents
- Input, computation, output
- h_theta(x) - Some activation function (more later)
- Weights on arrows

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

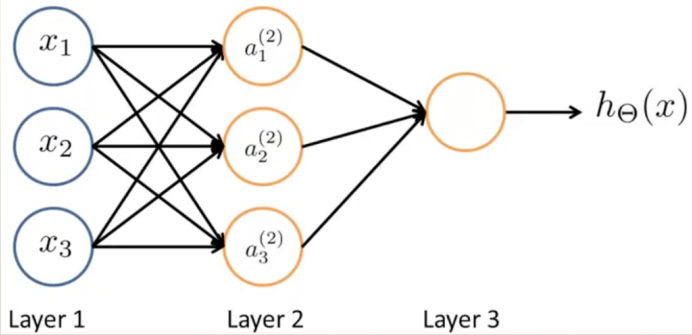
12th of December 2016

# Neural Network - Representation




- Show how this relates to the basic perceptron
- Layer 1 : Input Layer
- Layer 2: Hidden layer - not great terminology, can be many. Hidden, becaus not 'visible' in the training data.
- Layer 3: Output Layer
- Multilayer perceptron - bad terminology.
  Not several layers of perceptrons, but many perceptrons organised in layers

# Neural Network - Representation




- $a_i^{(j)}$ = activation of unit $i$ in layer $j$
- $\Theta$ = all activation weights
- $\Theta^{(j)}$ = weights for mapping from layer $j$ to $j + i$





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

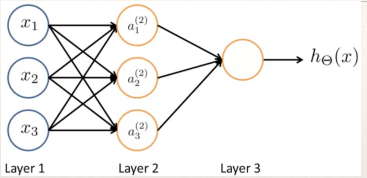
- Mention discrepancy with last week's terminology first
- Computational steps
- Terminology:
  - Activation: value computed based on act. func. and output by a unit
  - Convention: 1-indexing (in these slides)

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# Forward propagation




- $a_i^{(j)}$ = activation of unit $i$ in layer $j$
- $\Theta$ = all weights
- $\Theta^{(j)}$ = weights for mapping from layer $j$ to $j+i$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$



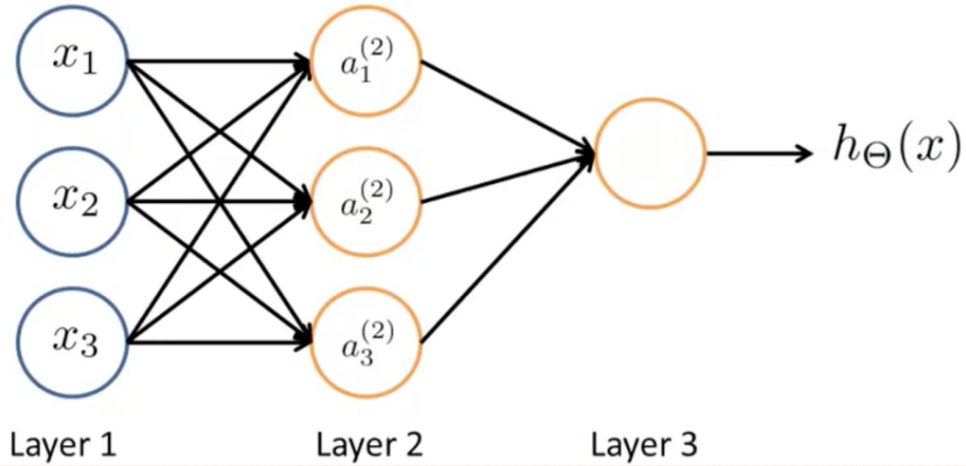
- Getting from inputs to outputs (forward propagation), i.e. the classification step, not the training/updating step (backprop)
- Weights are given
- Step through computation, quickly
- More on this soon!

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# Automatic feature learning



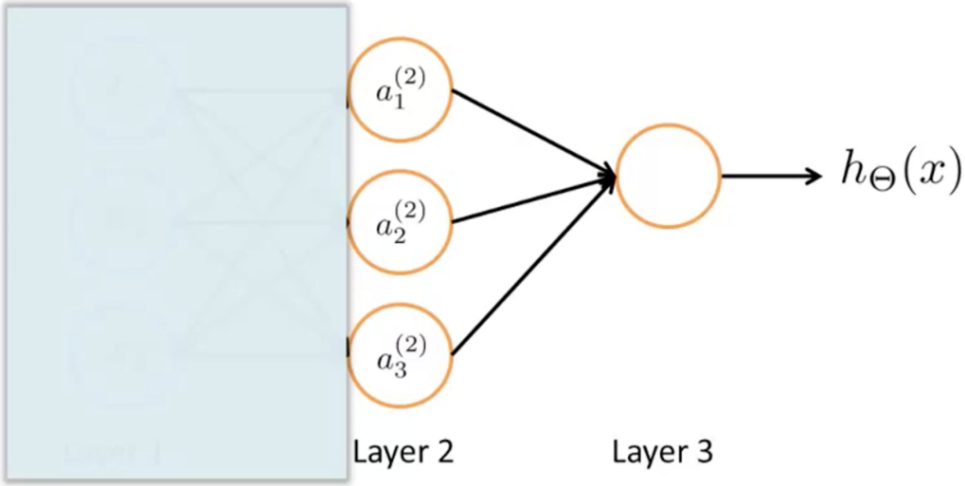

- From previous slide, simplified view in next

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# Automatic feature learning



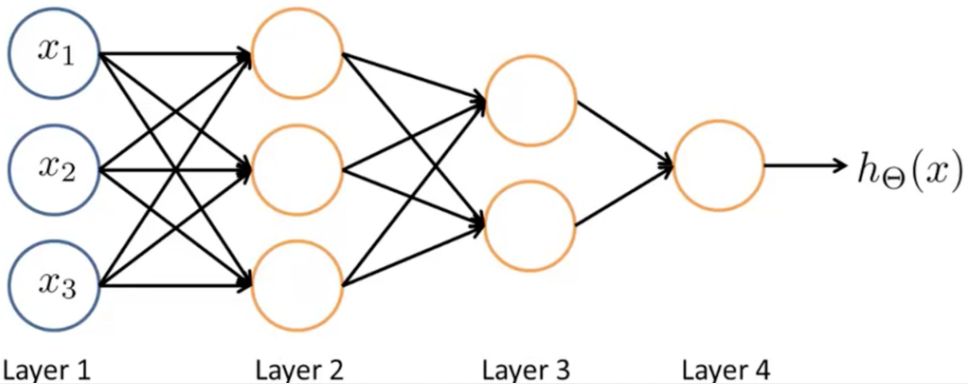

- Same as perceptron
- Uses features a1, a2, a3 instead of x1, x2, x3
- a1-3 are learned (through weights)!
- Explain intuition: we learn features automatically, rather than pre-define them

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

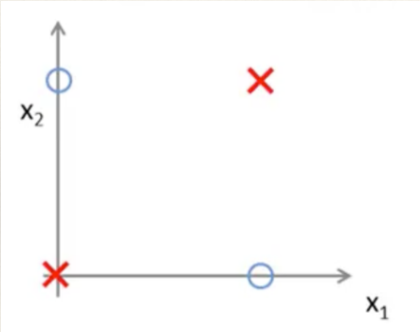
12th of December 2016

# Network Architectures




- More complex features functions, decision boundaries
- 2 hidden layers (anything not input or output)
- May seem abstract —detailed example next!

# Intuitions








Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

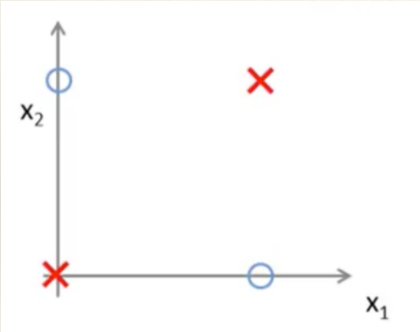
- More complex features functions, decision boundaries
- 2 hidden layers (anything not input or output)
- May seem abstract —detailed example next!

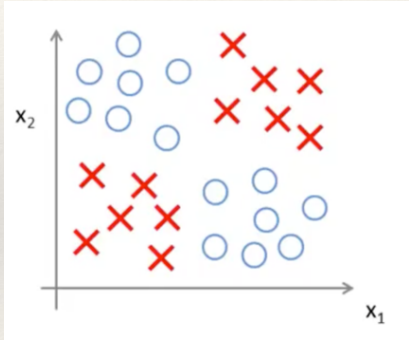
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

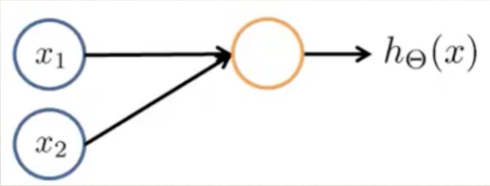
# Intuitions - XNOR

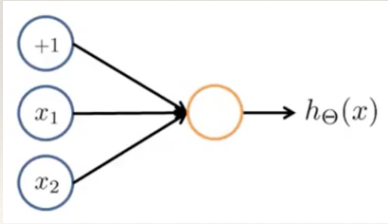



- Explain logical functions
- XNOR example not (x1 XOR x2)
- True if x1,x2 are both equally valued - otherwise false
- Boolean features
- Using perceptrons, we can do simple logical functions. Putting perceptrons together, we get neural network, which combines simple logical functions into more complex logical functions.

# Intuitions - AND function




| $x_1$ | $x_2$ | $h_\Theta(x)$ |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |



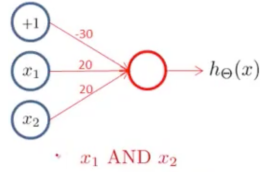


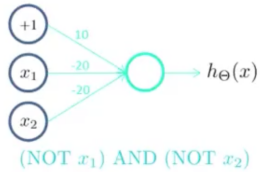
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

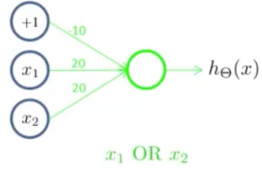
Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

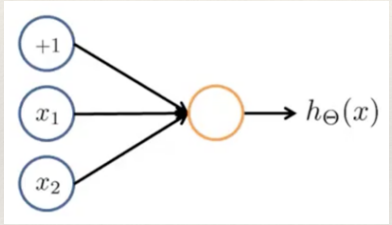
- Truth table recap
- show AND on whiteboard
- Bias unit - easier to train
- Draw sigmoid activation!!!
- Calculations clear?



# Intuitions - OR function




| $x_1$ | $x_2$ | $h_\Theta(x)$ |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |



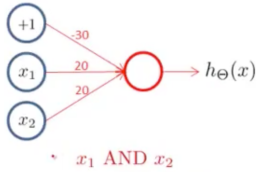


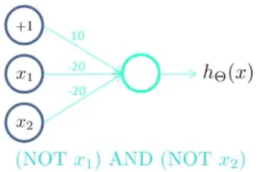
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

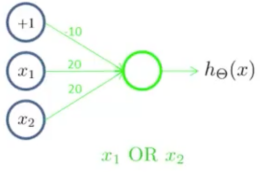
Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

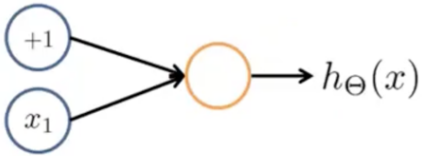
- show OR on whiteboard
- calculations clear?



# Intuitions - NOT function




| $x_1$ | $h_\Theta(x)$ |
|---|---|
| 0 | |
| 1 | |



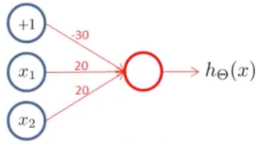


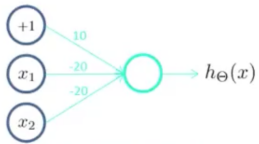
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

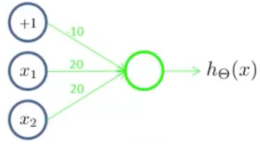
Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- show OR on whiteboard
- calculations clear?



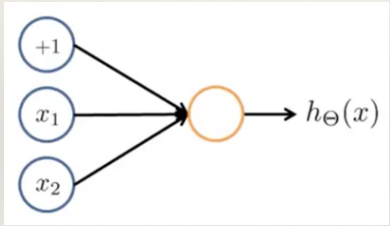



Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

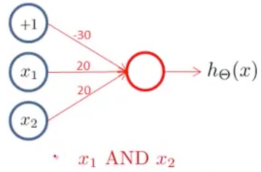
12th of December 2016

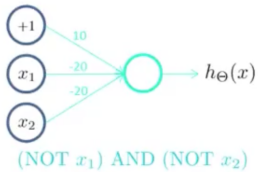
# Intuitions - NOT + AND

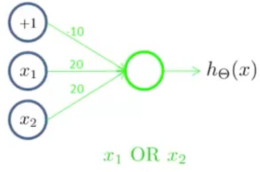



| $x_1$ | $x_2$ | $h_\Theta(x)$ |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

- show NOT on whiteboard
- Idea: large negative weight
- How would you make not x1 and not x2?



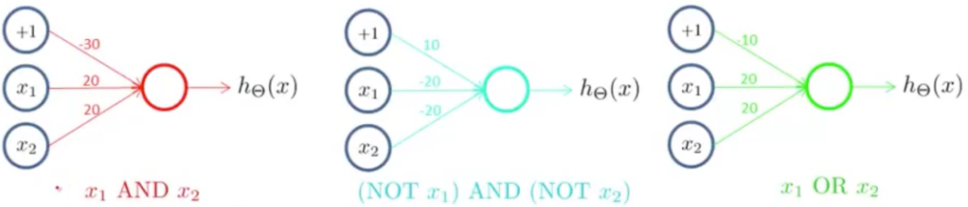



Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

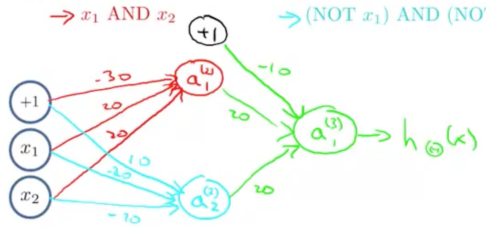
# Putting it together: XNOR





| $x_1$ | $x_2$ | $a_1^{(2)}$ | $a_2^{(2)}$ | $h_\Theta(x)$ |
|---|---|---|---|---|
| 0 | 0 | | | |
| 0 | 1 | | | |
| 1 | 0 | | | |
| 1 | 1 | | | |

- Combination of the three
- First neural network!
- Work through this example...
- Questions:
  - How should the truth table be?
  - What do we want to know (one neuron for both true, one for both false)?
  - And then (or, etc...)?
- More complex feature space!

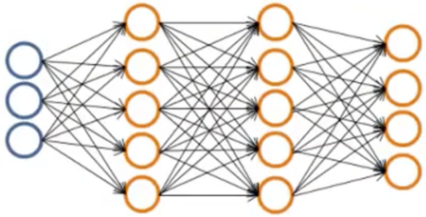



| $x_1$ | $x_2$ | $a_1^{(2)}$ | $a_2^{(2)}$ | $h_\Theta(x)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411



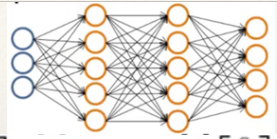
# Multi-class classification

$h_\Theta(x) \in \mathbb{R}^4$

- Previous, one-class classification.
- One-vs-all - each neuron represents a class
- One-hot representation of labels

# Multi-class representation



Want $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, etc.

when pedestrian when car when motorcycle

Training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

$y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

pedestrian car motorcycle truck





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Deep networks (many layers)
- Output layer = number of classes
- How about two classes?

# Break!

Afterwards:

- Learning weights (backpropagation)
- Deep learning and its prerequisites



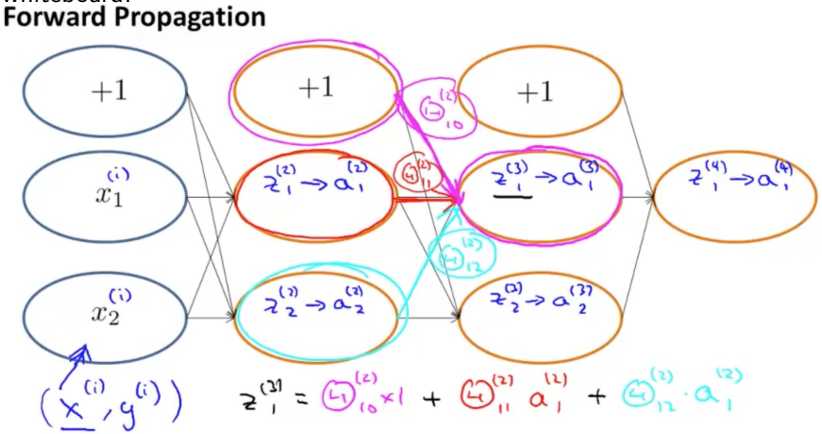


Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- In break, prepare empty neural network for forward and backward prop on whiteboard!



# Backprop algorithm

- Algorithm to 'learn weights'
  - Actually, to calculate the derivative of the 'error' of a neural network…
- Intuition: calculate how *wrong* each neuron in each layer is



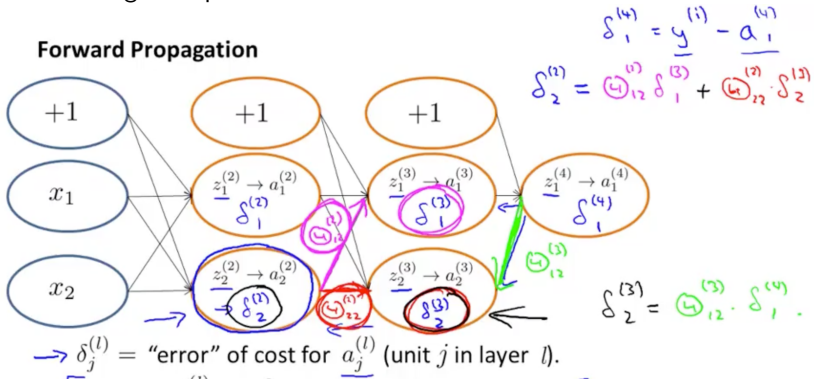


Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Backpropagation / Backprop
- Intuition: calculate the error of each neuron in each layer
- One training example



Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

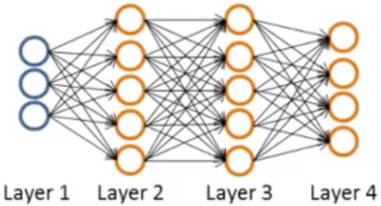
12th of December 2016

# Before backprop: forward prop

Given one training example $(x, y)$:

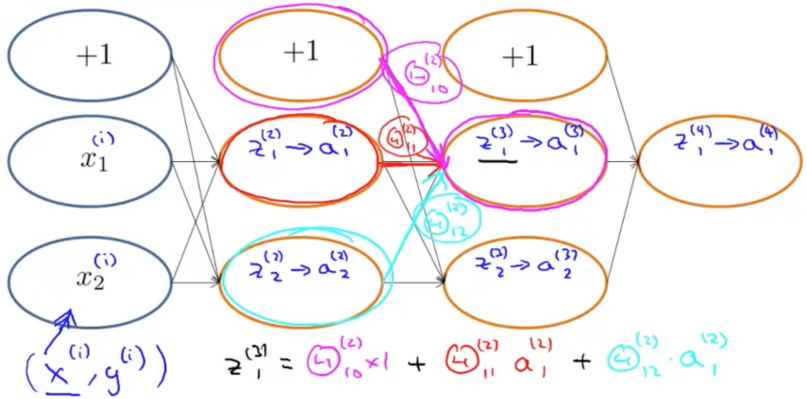
Forward propagation:

$$a^{(1)} = x$$
$$z^{(2)} = \Theta^{(1)} a^{(1)}$$
$$a^{(2)} = g(z^{(2)}) \quad (\text{add } a_0^{(2)})$$
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$a^{(3)} = g(z^{(3)}) \quad (\text{add } a_0^{(3)})$$
$$z^{(4)} = \Theta^{(3)} a^{(3)}$$
$$a^{(4)} = h_\Theta(x) = g(z^{(4)})$$






- Step by step through this **Forward Propagation**

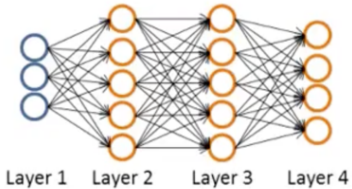


# ‘Error’ as compared to gold labels

Intuition: $\delta_j^{(l)}$ = “error” of node $j$ in layer $l$.

For each output unit (layer L = 4)

$$\delta_j^{(4)} = a_j^{(4)} - y_j$$








Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- The name backpropagation
- Begin with output

# Backprop algorithm

Calculate the error of each unit in each layer

1. Perform forward propagation to calculate activations
2. Using gold labels, calculate $\delta^{L}$
3. Calculate $\delta^{L-1}$ etc.



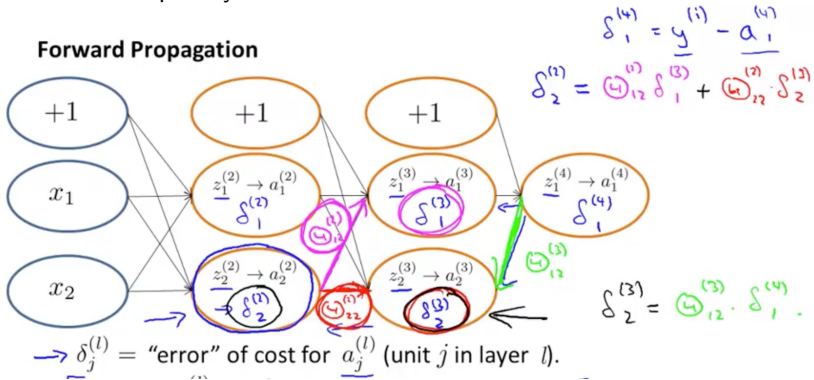


Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- The name backpropagation → Begin with output
- Quite complicated proof. Chain rule, if familiar with calculus
- No error for input layer



# Backprop – Step by step

- Weights update after each step
- A step is either:
  - One training example
  - One 'batch' of multiple training examples





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Challenging algorithm to understand.
- Okay if BP seems like a magical black box.
- Helpful to understand it, watch videos.
- Cost as difference between predicted value and actual value
- From next slide: talk about training properties

# Weight initialisation

- Zero initialisation?
- Initialise all weights to 1?
- Random initialisation?





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Some types of initialisation
- Zero/one all weights will be similar forever - all units are identical
- More sophisticated methods exist

# Dropout

- Weights are randomly 'forgotten'
  - Probabilities per layer
- Increases the robustness of a neural network
- A form of regularisation





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Tie back into overfitting / underfitting

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

# Activation functions

| Name | Plot | Equation | Derivative | Monotonic | $f(x) \approx x$ when $x \approx 0$ | Range |
|---|---|---|---|---|---|---|
| Identity |  | $f(x) = x$ | $f'(x) = 1$ | Yes | Yes | $(-\infty, \infty)$ |
| Binary step |  | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$ | Yes | No | $\{0, 1\}$ |
| Logistic (a.k.a Soft step) |  | $f(x) = \frac{1}{1+e^{-x}}$ | $f'(x) = f(x)(1 - f(x))$ | Yes | No | $(0, 1)$ |
| TanH |  | $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$ | $f'(x) = 1 - f(x)^2$ | Yes | Yes | $(-1, 1)$ |
| ArcTan |  | $f(x) = \tan^{-1}(x)$ | $f'(x) = \frac{1}{x^2+1}$ | Yes | Yes | $(-\frac{\pi}{2}, \frac{\pi}{2})$ |
| Rectified Linear |  | $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$ | Yes | No | $[0, \infty)$ |
| SoftPlus |  | $f(x) = \log_e(1 + e^x)$ | $f'(x) = \frac{1}{1+e^{-x}}$ | Yes | No | $(0, \infty)$ |
| Bent identity |  | $f(x) = \frac{\sqrt{x^2+1}-1}{2} + x$ | $f'(x) = \frac{x}{2\sqrt{x^2+1}} + 1$ | Yes | Yes | $(-\infty, \infty)$ |
| SoftExponential |  | $f(\alpha, x) = \begin{cases} -\frac{\log_e(1-\alpha(x+\alpha))}{\alpha} & \text{for } \alpha < 0 \\ x & \text{for } \alpha = 0 \\ \frac{e^{\alpha x}-1}{\alpha} + \alpha & \text{for } \alpha > 0 \end{cases}$ | $f'(\alpha, x) = \begin{cases} \frac{1}{1-\alpha(\alpha+x)} & \text{for } \alpha < 0 \\ e^{\alpha x} & \text{for } \alpha \geq 0 \end{cases}$ | Yes | Yes iff $\alpha \approx 0$ | $(-\infty, \infty)$ |
| Sinusoid |  | $f(x) = \sin(x)$ | $f'(x) = \cos(x)$ | No | Yes | $[-1, 1]$ |
| Sinc |  | $f(x) = \begin{cases} 1 & \text{for } x = 0 \\ \frac{\sin(x)}{x} & \text{for } x \neq 0 \end{cases}$ | $f'(x) = \begin{cases} 0 & \text{for } x = 0 \\ \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2} & \text{for } x \neq 0 \end{cases}$ | No | No | $[\approx -.217234, 1]$ |
| Gaussian |  | $f(x) = e^{-x^2}$ | $f'(x) = -2xe^{-x^2}$ | No | No | $(0, 1]$ |

https://en.wikipedia.org/wiki/Activation_function



- Map from the weighted input to output (activation)

# Activation functions

Traditional:

- Sigmoid, Tangent

Modern (Deep Learning!):

- Rectifier (aka. ReLU — Rectified Linear Unit)
- Maxout
  - Learns its own activation function per layer!





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Advantages of ReLU
- Show ReLU on whiteboard (with func. (max(0,x))
- Understand what that means?
- Skip MaxOut
- Next: Deep Learning

"I've worked all my life in Machine Learning, and I've never seen one algorithm knock over benchmarks like Deep Learning"

*Andrew Ng*





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Advantages of ReLU
- Show ReLU on whiteboard (with func. (max(0,x))
- Understand what that means?
- Skip MaxOut
- Next: Deep Learning

# Deep Learning

- Resurgence of Neural Network popularity!
- Catchy name…
- But what is it?





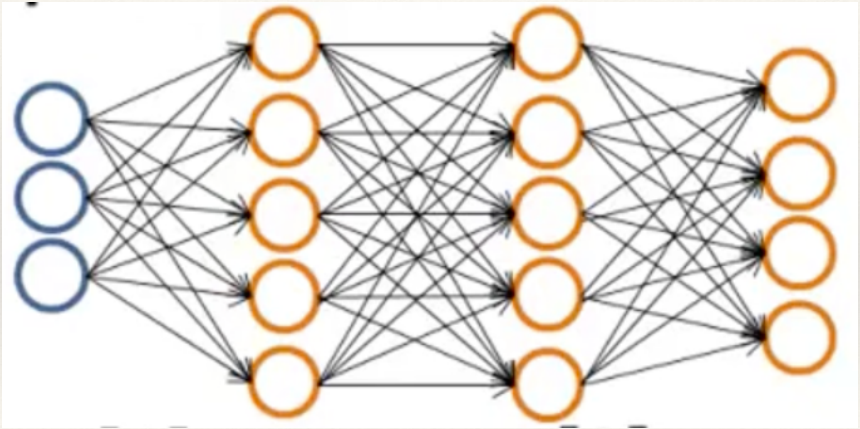
Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

•

# Deep Learning









Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411



- Really, neural networks with more than 1 hidden layer.
- Disappointing...

# Deep Learning

Why now?

- More computing power
- Better ways of training neural nets
  - Dropout
  - Rectifiers (ReLU)





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

•

# Deep Learning

Why?

- Each layer is another level of abstraction
- Much more complex problems can be handled
  - Image recognition
  - Facial recognition
  - From images to text





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Facial recognition example

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

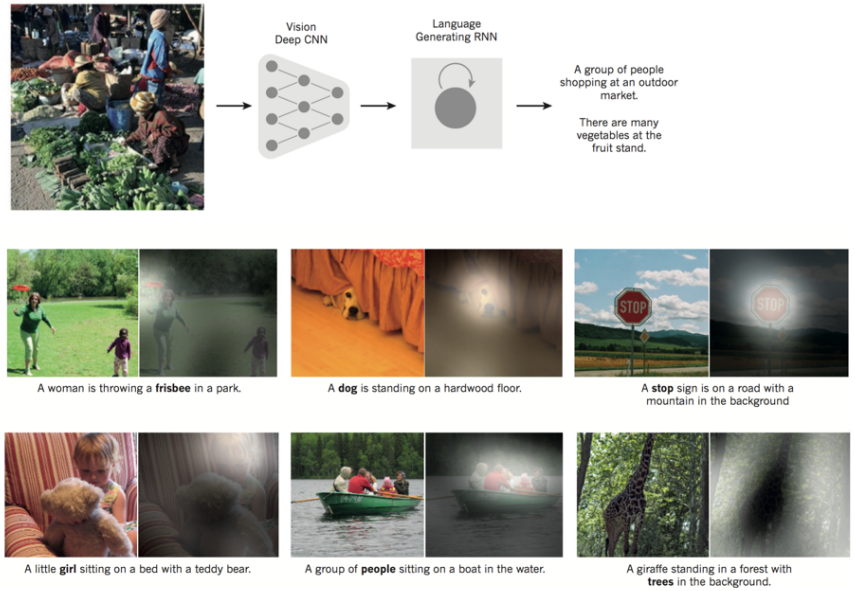



**Figure 3 | From image to text.** Captions generated by a recurrent neural network (RNN) taking, as extra input, the representation extracted by a deep convolution neural network (CNN) from a test image, with the RNN trained to 'translate' high-level representations of images into captions (top). Reproduced with permission from ref. 102. When the RNN is given the ability to focus its attention on a different location in the input image (middle and bottom; the lighter patches were given more attention) as it generates each word (bold), we found[86] that it exploits this to achieve better 'translation' of images into captions.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

- Recommended reading (on Nestor)
- Top three Neural Network researchers

# Putting it together

- Picking architecture:
  - Number of layers, units per layer
  - Activation function
  - Input units?
  - Output units?





Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

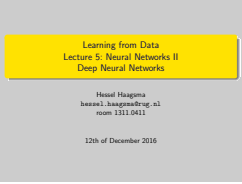
- Default: 1 hidden layer, unless complex problem
- Either:
  - Same no. units in every layer.
  - Fewer units in deeper layers - concentrate information
- General: Talked about classification, also applicable to regression

# Assignment preparation

- Install Keras (http://keras.io/#installation)
  - May not be possible on Windows — Use LWP computers
- Neural Network videos by Andrew Ng on Coursera
  - https://www.coursera.org/learn/machine-learning/home/week/4 (especially 'Model Representation')
  - https://www.coursera.org/learn/machine-learning/home/week/5 (especially Backprop)








- Especially forward prop and backprop
- Useful to hear this more than once
- Covers many of the same topics as the lecture
- Links don't work, but look on YouTube
- Theory based on what heard today
- Practice using the Keras toolkit
- Experiment with neural networks

Learning from Data
Lecture 5: Neural Networks II
Deep Neural Networks

Hessel Haagsma
hessel.haagsma@rug.nl
room 1311.0411

12th of December 2016

- Especially forward prop and backprop
- Useful to hear this more than once
- Covers many of the same topics as the lecture
- Links don't work, but look on YouTube
- Theory based on what heard today
- Practice using the Keras toolkit
- Experiment with neural networks