UPPSALA
UNIVERSITET

Deep Learning

Niklas Wahlström

Department of Information Technology, Uppsala University, Sweden

September 21, 2017

UPPSALA UNIVERSITET

Deep Learning: Caption generation

Generate caption automatically from images

Xu, K., Lei Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R. Richard S. Zemel, R. S., and Bengio, Y. **Show, attend and tell: neural image caption generation with visual attention**. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, July, 2015.

Deep learning: AI Go player

An AI defeated a human professional for the first time in the game of Go

Silver, D. et al. **Mastering the game of Go with deep neural networks and tree search**, *Nature*, Vol 529, 484–489 (2016)

Outline

1. What is a neural network (NN)?
2. Why do deep neural networks work so well?
    a) Why neural networks?
    b) Why deep?
3. Some comment, pointers and summary

Outline

1. **What is a neural network (NN)?**
2. Why do deep neural networks work so well?
   a) Why neural networks?
   b) Why deep?
3. Some comment, pointers and summary

UPPSALA UNIVERSITET

Skin cancer – background

One recent result on the use of deep learning in medicine - Detecting skin cancer (February 2017)

Andre Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. and Thrun, S. **Dermatologist-level classification of skin cancer with deep neural networks**. *Nature*, 542, 115–118, February, 2017.

Skin cancer – background

One recent result on the use of deep learning in medicine - Detecting skin cancer (February 2017)

Andre Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. and Thrun, S. **Dermatologist-level classification of skin cancer with deep neural networks**. *Nature*, 542, 115–118, February, 2017.

Some background figures (from the US) on skin cancer:

- Melanomas represents less than 5% of all skin cancers, **but** accounts for 75% of all skin-cancer-related deaths.
- Early detection absolutely critical. Estimated 5-year survival rate for melanoma: Over 99% if detected in its earlier stages and 14% is detected in its later stages.

Skin cancer – taxonomy used

Skin cancer – task

Image copyright Nature (doi:10.1038/nature21056)

Skin cancer – solution (ultrabrief)

Start from a neural network trained on 1.28 million images (**transfer learning**).

Make minor modifications to this model, specializing to present situation.

Skin cancer – solution (ultrabrief)

Start from a neural network trained on 1.28 million images (**transfer learning**).

Make minor modifications to this model, specializing to present situation.

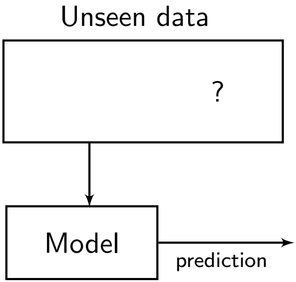
Learn new model parameters using $129\,450$ clinical images (~ 100 times more images than any previous study).

Skin cancer – solution (ultrabrief)

Start from a neural network trained on 1.28 million images (**transfer learning**).

Make minor modifications to this model, specializing to present situation.

Learn new model parameters using $129\,450$ clinical images (~ 100 times more images than any previous study).



Skin cancer – indication of the results

$$\text{sensitivity} = \frac{\text{true positive}}{\text{positive}} \qquad \text{specificity} = \frac{\text{true negative}}{\text{negative}}$$

Skin cancer – indication of the results

$$\text{sensitivity} = \frac{\text{true positive}}{\text{positive}} \qquad \text{specificity} = \frac{\text{true negative}}{\text{negative}}$$

Constructing an NN for regression

A **neural network (NN)** is a nonlinear function $\hat{\mathbf{y}} = \mathbf{g}_{\boldsymbol{\theta}}(\boldsymbol{\varphi})$ from an input variable $\boldsymbol{\varphi}$ to an output variable $\hat{\mathbf{y}}$ parameterized by $\boldsymbol{\theta}$.

Linear regression models the relationship between a continuous target variable $\hat{y}$ and an input variable $\boldsymbol{\varphi}$,

$$\hat{y} = \sum_{i=1}^{n} \varphi_i \theta_i + \theta_0 = \boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{\theta},$$

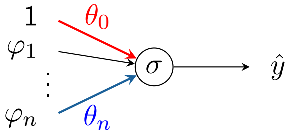
where $\boldsymbol{\theta}$ is the parameters composed by the "weights" θ_i and the offset ("bias") term θ_0,

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 & \theta_1 & \theta_2 & \cdots & \theta_n \end{pmatrix}^{\mathsf{T}},$$
$$\boldsymbol{\varphi} = \begin{pmatrix} 1 & \varphi_1 & \varphi_2 & \cdots & \varphi_n \end{pmatrix}^{\mathsf{T}}.$$

Generalized linear regression

We can generalize this by introducing nonlinear transformations of the predictor $\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}$,

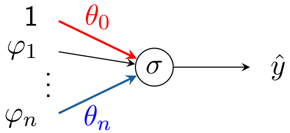
$$\hat{y} = \sigma(\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}).$$



Generalized linear regression

We can generalize this by introducing nonlinear transformations of the predictor $\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}$,

$$\hat{y} = \sigma(\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}).$$

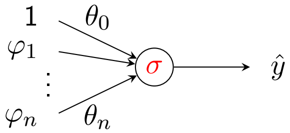


We call $\sigma(x)$ the *activation function*. Two common choices are:

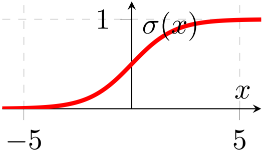
Generalized linear regression

We can generalize this by introducing nonlinear transformations of the predictor $\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}$,

$$\hat{y} = \sigma(\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}).$$



We call $\sigma(x)$ the *activation function*. Two common choices are:

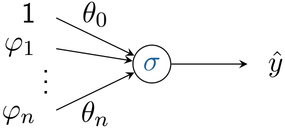


Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$

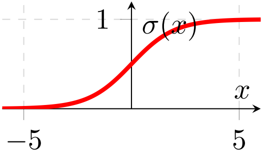
Generalized linear regression

We can generalize this by introducing nonlinear transformations of the predictor $\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{\theta}$,

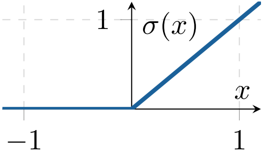
$$\hat{y} = \sigma(\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{\theta}).$$



We call $\sigma(x)$ the *activation function*. Two common choices are:



Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$



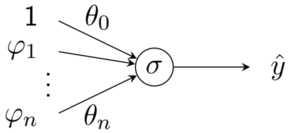
ReLU: $\sigma(x) = \max(0, x)$

Let us consider an example of a **feed-forward NN**, indicating that the information flows from the input to the output layer.

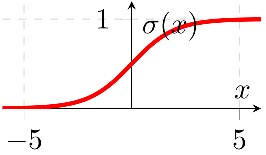
Generalized linear regression

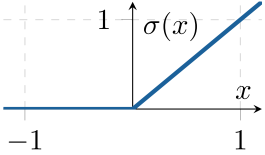
We can generalize this by introducing nonlinear transformations of the predictor $\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{\theta}$,

$$\hat{y} = \sigma(\boldsymbol{\varphi}^{\mathsf{T}}\boldsymbol{\theta}).$$



We call $\sigma(x)$ the *activation function*. Two common choices are:





Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$ ReLU: $\sigma(x) = \max(0, x)$

Let us consider an example of a **feed-forward NN**, indicating that the information flows from the input to the output layer.

UPPSALA UNIVERSITET

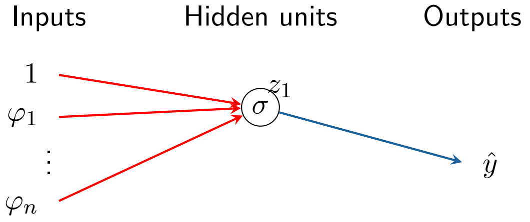
Neural network - construction

A NN is a sequential construction of several linear regression models.

Inputs	Hidden units	Outputs
1		
φ_1		
$\vdots$		
φ_n		

Neural network - construction

A NN is a sequential construction of several linear regression models.



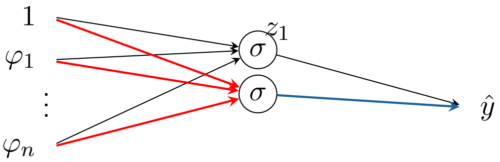
$$z_1 = \sigma\left(\theta_{01}^{(1)} + \sum_{j=1}^{n} \theta_{j1}^{(1)} \varphi_j\right)$$

$$y = \theta_1^{(2)} z_1$$

Neural network - construction

A NN is a sequential construction of several linear regression models.

Inputs Hidden units Outputs



$$z_1 = \sigma\left(\theta_{01}^{(1)} + \sum\nolimits_{j=1}^{n} \theta_{j1}^{(1)} \varphi_j\right)$$

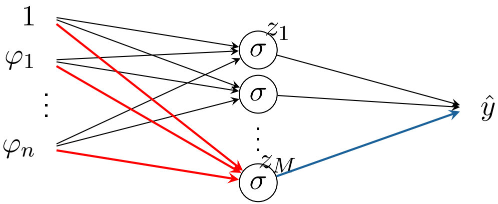
$$z_2 = \sigma\left(\theta_{02}^{(1)} + \sum\nolimits_{j=1}^{n} \theta_{j2}^{(1)} \varphi_j\right)$$

$$y = \sum_{m=1}^{2} \theta_m^{(2)} z_m$$

Neural network - construction

A NN is a sequential construction of several linear regression models.

Inputs Hidden units Outputs



$$z_1 = \sigma\left(\theta^{(1)}_{01} + \sum\nolimits_{j=1}^{n} \theta^{(1)}_{j1}\varphi_j\right)$$

$$z_2 = \sigma\left(\theta^{(1)}_{02} + \sum\nolimits_{j=1}^{n} \theta^{(1)}_{j2}\varphi_j\right)$$

$$\vdots$$

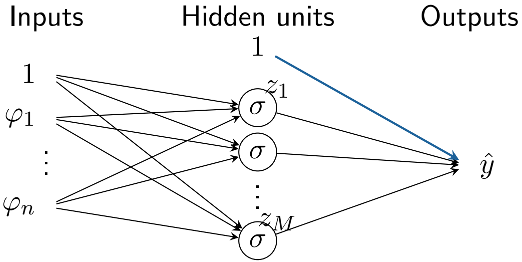
$$z_M = \sigma\left(\theta^{(1)}_{0M} + \sum\nolimits_{j=1}^{n} \theta^{(1)}_{jM}\varphi_j\right)$$

$$y = \sum_{m=1}^{M} \theta^{(2)}_{m} z_m$$

UPPSALA UNIVERSITET

Neural network - construction

A NN is a sequential construction of several linear regression models.



$$z_1 = \sigma\left(\theta_{01}^{(1)} + \sum\nolimits_{j=1}^{n} \theta_{j1}^{(1)} \varphi_j\right)$$

$$z_2 = \sigma\left(\theta_{02}^{(1)} + \sum\nolimits_{j=1}^{n} \theta_{j2}^{(1)} \varphi_j\right)$$

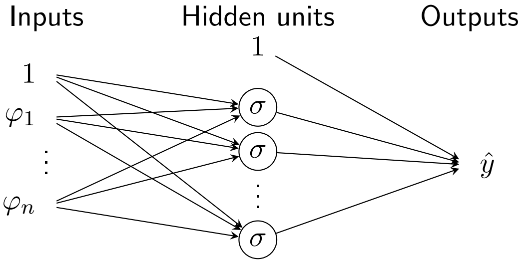
$$\vdots$$

$$z_M = \sigma\left(\theta_{0M}^{(1)} + \sum\nolimits_{j=1}^{n} \theta_{jM}^{(1)} \varphi_j\right)$$

$$y = \theta_0^{(2)} + \sum_{m=1}^{M} \theta_m^{(2)} z_m$$

Neural network - construction

A NN is a sequential construction of several linear regression models.



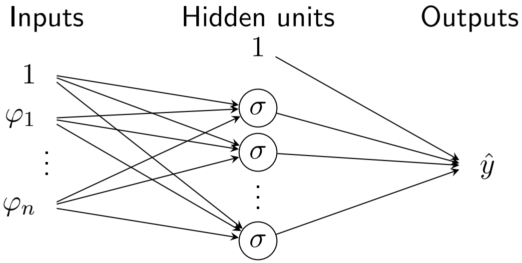
$$\mathbf{z} = \sigma(W_1^{\mathsf{T}}\boldsymbol{\varphi} + \mathbf{b}_1^{\mathsf{T}}) \qquad \hat{y} = W_2^{\mathsf{T}}\mathbf{z} + \mathbf{b}_2^{\mathsf{T}}$$

$$\mathbf{b}_1 = \begin{bmatrix} \theta_{01}^{(1)} & \dots & \theta_{0M}^{(1)} \end{bmatrix} \qquad \mathbf{b}_2 = \begin{bmatrix} \theta_0^{(1)} \end{bmatrix}$$

$$W_1 = \begin{bmatrix} \theta_{01}^{(1)} & \dots & \theta_{0M}^{(1)} \\ \vdots & \dots & \vdots \\ \theta_{n1}^{(1)} & \dots & \theta_{nM}^{(1)} \end{bmatrix} \qquad W_2 = \begin{bmatrix} \theta_0^{(2)} \\ \vdots \\ \theta_M^{(2)} \end{bmatrix}$$

Neural network - construction

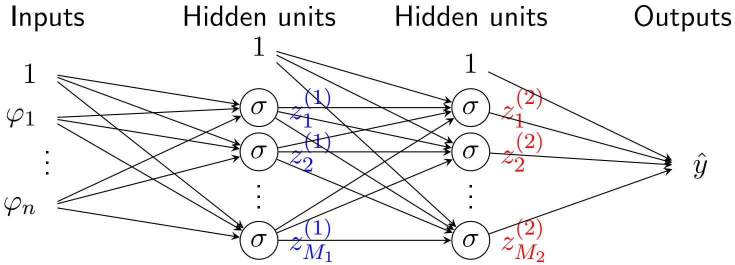
A NN is a sequential construction of several several linear regression models.



$$\mathbf{z} = \sigma(W_1^\mathsf{T}\boldsymbol{\varphi} + \mathbf{b}_1^\mathsf{T})$$
$$y = W_2^\mathsf{T}\mathbf{Z} + \mathbf{b}_2^\mathsf{T}$$

Neural network - construction

A NN is a sequential construction of several linear regression models.



$$\mathbf{z}^{(1)} = \sigma(W_1^{\mathsf{T}}\boldsymbol{\varphi} + \mathbf{b}_1^{\mathsf{T}})$$
$$\mathbf{z}^{(2)} = \sigma(W_2^{\mathsf{T}}\mathbf{z}^{(1)} + \mathbf{b}_2^{\mathsf{T}})$$
$$y = W_3^{\mathsf{T}}\mathbf{z}^{(2)} + \mathbf{b}_3^{\mathsf{T}}$$

The model learns better using a deep network (several layers) instead of a wide and shallow network.

UPPSALA UNIVERSITET

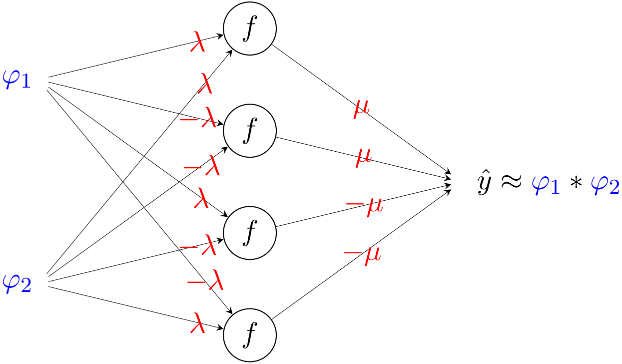
Outline

1. What is a neural network (NN)?
2. **Why do deep neural networks work so well?**
   a) **Why neural networks?**
   b) Why deep?
3. Some comment, pointers and summary

Why neural networks?

Continuous multiplication gate

A neural network with only four hidden units can model multiplication of two numbers arbitrarily well.



If we choose $\mu = \frac{1}{4\lambda^2 f''(0)}$ then $\hat{y} \to \varphi_1 * \varphi_2$ when $\lambda \to 0$.

Henry W. Lin and Max Tegmark. (2016) **Why does deep and cheap learning work so well?**, *arXiv*

A regression example

Input: $\mathbf{u} \in \mathbb{R}^{1000}$
Output: $y \in \mathbb{R}$
Task: Model a quadratic relationship between y and $\mathbf{u}$

A regression example

Input: $\mathbf{u} \in \mathbb{R}^{1000}$
Output: $y \in \mathbb{R}$
Task: Model a quadratic relationship between y and $\mathbf{u}$

Linear regression

$$\hat{y} = u_1 u_1 \theta_{1,1} + u_1 u_2 \theta_{1,2} + \cdots + u_{1000} u_{1000} \theta_{1000,1000} = \boldsymbol{\varphi}^\mathsf{T} \boldsymbol{\theta}$$

where

$$\boldsymbol{\varphi} = \begin{bmatrix} u_1 u_1 & u_1 u_2 & \dots & u_{1000} u_{1000} \end{bmatrix}^\mathsf{T}$$
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1000,1000} \end{bmatrix}^\mathsf{T}$$

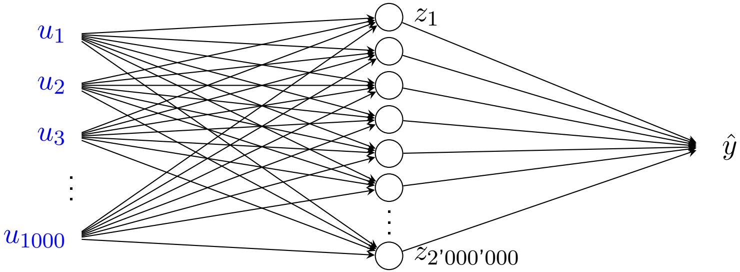
Requires $\approx \frac{1'000 * 1'000}{2} = 500'000$ parameters!

A regression example

Input: $\mathbf{u} \in \mathbb{R}^{1000}$
Output: $y \in \mathbb{R}$
Task: Model a quadratic relationship between y and $\mathbf{u}$

Neural network

To model all products with a neural network we would need $4 * 500'000 = 2 * 10^6$ hidden units and hence 2 billion parameters...



$1000 * (2 * 10^6) \quad + \quad 2 * 10^6 \approx 2 * 10^9$ param.

A regression example (cont.)

Input: $\mathbf{u} \in \mathbb{R}^{1000}$
Output: $y \in \mathbb{R}$
Task: Model a quadratic relationship between y and $\mathbf{u}$

Assume that only 10 of the regressors $u_i u_j$ are of importance

Linear regression

$$\hat{y} = u_1 u_1 \theta_{1,1} + u_1 u_2 \theta_{1,2} + \cdots + u_{1000} u_{1000} \theta_{1000,1000} = \boldsymbol{\varphi}^\mathsf{T} \boldsymbol{\theta}$$

where

$$\boldsymbol{\varphi} = \begin{bmatrix} u_1 u_1 & u_1 u_2 & \ldots & u_{1000} u_{1000} \end{bmatrix}^\mathsf{T}$$
$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \ldots & \theta_{1000,1000} \end{bmatrix}^\mathsf{T}$$

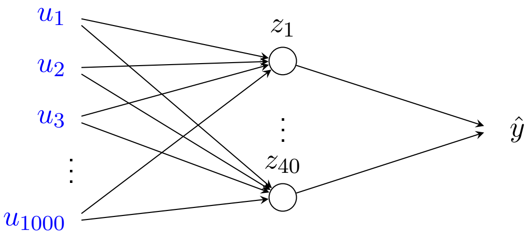
You probably want to regularize, but 500'000 parameters are still required!

A regression example (cont.)

Input: $\mathbf{u} \in \mathbb{R}^{1000}$
Output: $y \in \mathbb{R}$
Task: Model a quadratic relationship between y and $\mathbf{u}$

Assume that only 10 of the regressors $u_i u_j$ are of importance

Neural network
To model 10 products with a neural network we would need 4*10 hidden units, i.e. leading to only $\approx$ 40'000 parameters!



1000*40 $+$ 40 $=$ 40'040

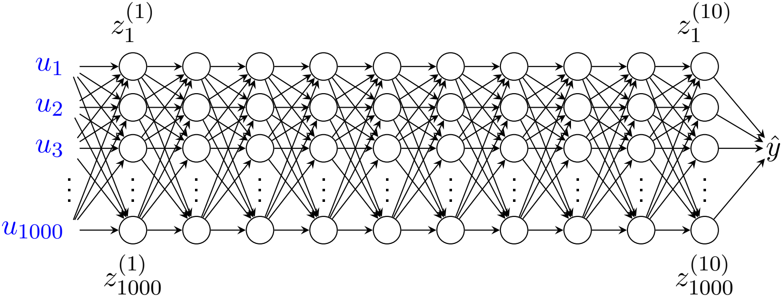
UPPSALA UNIVERSITET

Outline

1. What is a neural network (NN)?
2. **Why do deep neural networks work so well?**
   a) Why neural networks?
   b) Why deep?
3. Some comment, pointers and summary

Why deep? - A regression example

- Consider the same example. Now we want a model with complexity corresponding to polynomials of degree 1'000.
- Keep 250 products in each layer $\Rightarrow 250*4=1'000$ hidden units.



$$\underbrace{10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^3}_{\approx 10^7 \text{ parameters}}$$

Linear regression would require $\approx \frac{1000^{1000}}{1000!}$ parameters to model such a relationship...

Why deep? - Image classification

Example: Image classification

Input: pixels of an **image**

Output: **object identity**

Each hidden layer extracts increasingly abstract features.

Zeiler, M. D. and Fergus, R. **Visualizing and understanding convolutional networks** *Computer Vision - ECCV* (2014).

Deep neural networks

Deep learning methods allow a machine to make use of raw data to automatically discover the representations (abstractions) that are necessary to solve a particular task.

Deep neural networks

Deep learning methods allow a machine to make use of raw data to automatically discover the representations (abstractions) that are necessary to solve a particular task.

It is accomplished by using **multiple levels of representation**. Each level transforms the representation at the previous level into a new and more abstract representation,

$$\mathbf{z}^{(l+1)} = \mathbf{f}\left(W^{(l+1)}\mathbf{z}^{(l)} + \mathbf{b}^{(l+1)}\right),$$

starting from the input (raw data) $\mathbf{z}^{(0)} = \mathbf{u}$.

Deep neural networks

Deep learning methods allow a machine to make use of raw data to automatically discover the representations (abstractions) that are necessary to solve a particular task.

It is accomplished by using **multiple levels of representation**. Each level transforms the representation at the previous level into a new and more abstract representation,

$$\mathbf{z}^{(l+1)} = \mathbf{f}\left(W^{(l+1)}\mathbf{z}^{(l)} + \mathbf{b}^{(l+1)}\right),$$

starting from the input (raw data) $\mathbf{z}^{(0)} = \mathbf{u}$.

Key aspect: The layers are **not** designed by human engineers, they are generated from (typically lots of) data using a learning procedure and lots of computations.

Outline

1. What is a neural network (NN)?
2. Why do deep neural networks work so well?
   a) Why neural networks?
   b) Why deep?
3. **Some comment, pointers and summary**

Some comments - Why now?

Neural networks have been around for more than fifty years. Why have they become so popular now (again)?

To solve really interesting problems you need:

1. Efficient learning **algorithms**
2. Efficient computational **hardware**
3. A lot of labeled **data**!

These three factors have not been fulfilled to a satisfactory level until the last 5-10 years.

Some pointers

A book has recently been written

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *MIT Press*, 2016

http://www.deeplearningbook.org/

Some pointers

A book has recently been written

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *MIT Press*, 2016

http://www.deeplearningbook.org/

A well written introduction:

LeCun, Y., Bengio, Y., and Hinton, G. (2015) **Deep learning**, *Nature*, 521(7553), 436–444.

Some pointers

A book has recently been written

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *MIT Press*, 2016

http://www.deeplearningbook.org/

A well written introduction:

LeCun, Y., Bengio, Y., and Hinton, G. (2015) **Deep learning**, *Nature*, 521(7553), 436–444.

Details about the multiplication gate

Henry W. Lin and Max Tegmark. (2016) **Why does deep and cheap learning work so well?**, *arXiv*

UPPSALA UNIVERSITET

Some pointers

A book has recently been written

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *MIT Press*, 2016

http://www.deeplearningbook.org/

A well written introduction:

LeCun, Y., Bengio, Y., and Hinton, G. (2015) **Deep learning**, *Nature*, 521(7553), 436–444.

Details about the multiplication gate

Henry W. Lin and Max Tegmark. (2016) **Why does deep and cheap learning work so well?**, *arXiv*

You will also find more material than you can possibly want here

http://deeplearning.net/

Summary

A **neural network (NN)** is a nonlinear function $\mathbf{y} = \mathrm{g}_{\boldsymbol{\theta}}(\mathbf{u})$ from an input variable $\mathbf{u}$ to an output variable $\mathbf{y}$ parameterized by $\boldsymbol{\theta}$.

Summary

A **neural network (NN)** is a nonlinear function $\mathbf{y} = \mathrm{g}_{\boldsymbol{\theta}}(\mathbf{u})$ from an input variable $\mathbf{u}$ to an output variable $\mathbf{y}$ parameterized by $\boldsymbol{\theta}$.

We can think of an NN as a sequential/recursive construction of several generalized linear regressions.

Summary

A **neural network (NN)** is a nonlinear function $\mathbf{y} = \mathrm{g}_{\boldsymbol{\theta}}(\mathbf{u})$ from an input variable $\mathbf{u}$ to an output variable $\mathbf{y}$ parameterized by $\boldsymbol{\theta}$.

We can think of an NN as a sequential/recursive construction of several generalized linear regressions.

Deep learning refers to learning NNs with several hidden layers. Allows for data-driven models that automatically learns rep. of data (features) with multiple layers of abstraction.

UPPSALA UNIVERSITET

Summary

A **neural network (NN)** is a nonlinear function $\mathbf{y} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{u})$ from an input variable $\mathbf{u}$ to an output variable $\mathbf{y}$ parameterized by $\boldsymbol{\theta}$.

We can think of an NN as a sequential/recursive construction of several generalized linear regressions.

Deep learning refers to learning NNs with several hidden layers. Allows for data-driven models that automatically learns rep. of data (features) with multiple layers of abstraction.

A deep NN is very **parameter efficient** when modelling high-dimensional, complex data.

Thank you!