UPPSALA UNIVERSITET

Deep Learning

Niklas Wahlström

Department of Information Technology, Uppsala University, Sweden

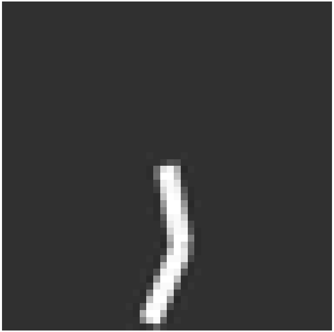
September 23, 2016

UPPSALA UNIVERSITET

Deep Learning: A recent example

First steps towards an autonomous system that learns by itself from raw pixel data.

Trial: 3 Frame: 94

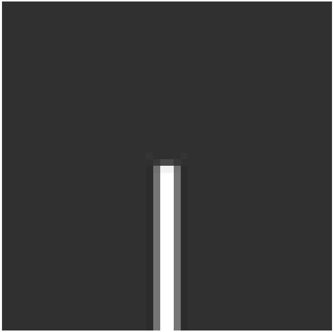


J.-A. M. Assael, N. Wahlström, T. B. Schön, and M. P. Deisenroth. **Data-Efficient Learning of Feedback Policies from Image Pixels using Deep Dynamical Models**. In *Deep Reinforc. Learning WS at the Conference on Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec. 2015.

Deep Learning: A recent example

First steps towards an autonomous system that learns by itself from raw pixel data.

Trial: 1 Frame: 1



J.-A. M. Assael, N. Wahlström, T. B. Schön, and M. P. Deisenroth. **Data-Efficient Learning of Feedback Policies from Image Pixels using Deep Dynamical Models**. In *Deep Reinforc. Learning WS at the Conference on Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec. 2015.

Deep Learning: Another recent example

Generate caption automatically from images

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

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.

A few examples where it failed

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.

Deep learning: On more recent example

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)

UPPSALA UNIVERSITET

Outline

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

Constructing an NN for regression

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

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

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

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 $\boldsymbol{\varphi}^\mathsf{T}\boldsymbol{\theta}$,

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

Generalized linear regression

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

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

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

NN for regression – an example

1. Form m_1 linear combinations of the input $\boldsymbol{\varphi} \in \mathbb{R}^n$

$$a_j^{(1)} = \sum_{i=1}^{n} \theta_{ji}^{(1)} \varphi_i + \varphi_{j0}^{(1)}, \qquad j = 1, \ldots, m_1.$$

NN for regression – an example

1. Form m_1 linear combinations of the input $\boldsymbol{\varphi} \in \mathbb{R}^n$

$$a_j^{(1)} = \sum_{i=1}^{n} \theta_{ji}^{(1)} \varphi_i + \varphi_{j0}^{(1)}, \qquad j = 1, \dots, m_1.$$

2. Apply a nonlinear transformation

$$z_j = f\left(a_j^{(1)}\right), \qquad j = 1, \dots, m_1.$$

NN for regression – an example

1. Form m_1 linear combinations of the input $\boldsymbol{\varphi} \in \mathbb{R}^n$

$$a_j^{(1)} = \sum_{i=1}^{n} \theta_{ji}^{(1)} \varphi_i + \varphi_{j0}^{(1)}, \qquad j = 1, \ldots, m_1.$$

2. Apply a nonlinear transformation

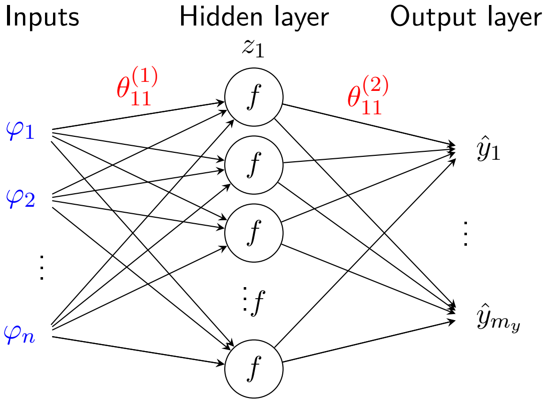
$$z_j = f\left(a_j^{(1)}\right), \qquad j = 1, \ldots, m_1.$$

3. Form m_y linear combinations of $\mathbf{z} \in \mathbb{R}^{m_1}$

$$y_k = \sum_{j=1}^{m_y} \theta_{kj}^{(2)} z_j + \theta_{k0}^{(2)}, \qquad k = 1, \ldots, m_y.$$

NN for regression – an example

$$\hat{y}_k(\theta) = \sum_{j=1}^{m_1} \theta_{kj}^{(2)} f\left(\sum_{i=1}^{n} \theta_{ji}^{(1)} \varphi_i + \theta_{j0}^{(1)}\right) + \theta_{k0}^{(2)}$$



UPPSALA UNIVERSITET

Multi-layer neural networks

We can think of the neural network as a sequential/recursive construction of several generalized linear regressions.

Each layer in a multi-layer NN is modelled as

$$\mathbf{z}^{(l+1)} = \mathbf{f}\left(\Theta^{(l+1)}\mathbf{z}^{(l)} + \theta_0^{(l+1)}\right),$$

starting with the input $\mathbf{z}^{(0)} = \boldsymbol{\varphi}$. (The nonlinearity operates element-wise.)

UPPSALA UNIVERSITET

Multi-layer neural networks

We can think of the neural network as a sequential/recursive construction of several generalized linear regressions.

Each layer in a multi-layer NN is modelled as

$$\mathbf{z}^{(l+1)} = \mathbf{f}\left(\Theta^{(l+1)}\mathbf{z}^{(l)} + \theta_0^{(l+1)}\right),$$

starting with the input $\mathbf{z}^{(0)} = \varphi$. (The nonlinearity operates element-wise.)

The scalar nonlinear function $f(\cdot)$ is what makes the neural network nonlinear. Common functions are $f(z) = 1/(1 + e^{-z})$, $f(z) = \tanh(z)$ and $f(z) = \max(0, z)$.

The so-called **rectified linear unit (ReLU)** $f(z) = \max(0, z)$ is heavily used for deep architectures.

UPPSALA UNIVERSITET

Outline

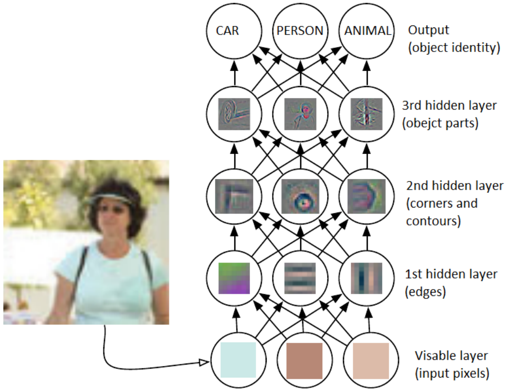
1. Introduction via three recent applications
2. What is a neural network (NN)?
3. **Why do deep neural networks work so well?**
   a) Why neural networks?
   b) Why deep?
4. Some comment, pointers and summary

Why do deep neural networks work so well?

Example: Image classification

Input: pixels of an **image**
Output: **object identity**

- 1 megapixel (black/white) $\Rightarrow$ $2^{1'000'000}$ possible images!
- A **deep neural network** can solve this with a few million parameters!



How can deep neural networks work so well?

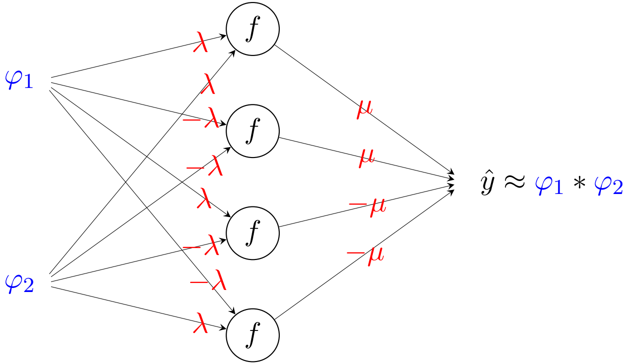
Outline

1. Introduction via three recent applications
2. What is a neural network (NN)?
3. **Why do deep neural networks work so well?**
   a) **Why neural networks?**
   b) Why deep?
4. 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

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

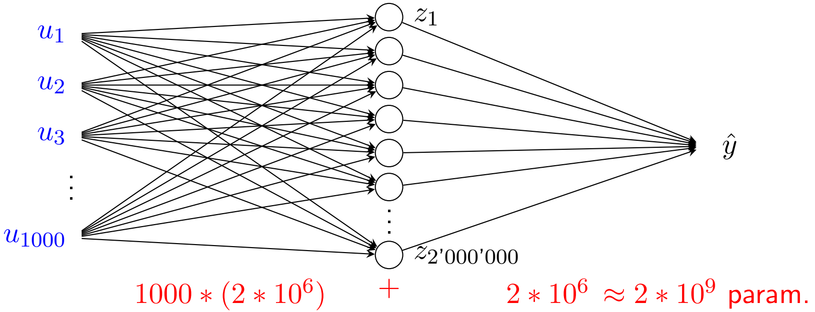
Requires $\approx \frac{1'000 * 1'000}{2} = 500'000$ parameters!

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...



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

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

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

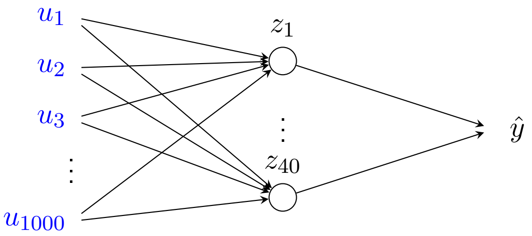
UPPSALA UNIVERSITET

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!



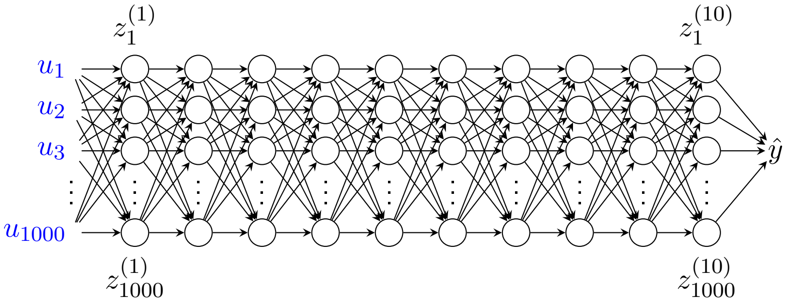
1000*40 $+$ 40 $=$ 40'040

Outline

1. Introduction via three recent applications
2. What is a neural network (NN)?
3. **Why do deep neural networks work so well?**
   a) Why neural networks?
   b) **Why deep?**
4. 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 ⇒ $250*4=1'000$ hidden units.



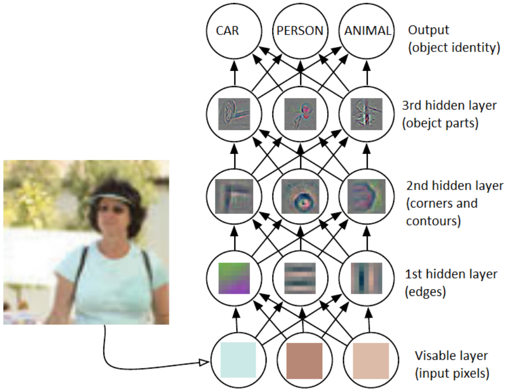
$$\underbrace{10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^6+10^3}_{\approx 10^7 \text{ parameters}}$$

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. Introduction via three recent applications
2. What is a neural network (NN)?
3. Why do deep neural networks work so well?
   a) Why neural networks?
   b) Why deep?
4. **Some comment, pointers and summary**

UPPSALA UNIVERSITET

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 is being written at the moment

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *Book in preparation for MIT Press*,

http://www.deeplearningbook.org/

UPPSALA UNIVERSITET

Some pointers

A book is being written at the moment

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *Book in preparation for MIT Press*,

http://www.deeplearningbook.org/

A well written and timely introduction:

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

Some pointers

A book is being written at the moment

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *Book in preparation for MIT Press*,

http://www.deeplearningbook.org/

A well written and timely 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*

Some pointers

A book is being written at the moment

I. Goodfellow, Y. Bengio and A. Courville **Deep learning** *Book in preparation for MIT Press*,

http://www.deeplearningbook.org/

A well written and timely 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.

UPPSALA UNIVERSITET

Thank you!