

Machine Learning

Neural Networks, part 2; and the beginning

Jeff Abrahamson

Cours sur l'année, 2017–2018

Backpropagation

Backpropagation

Assign small random weights

while *not converged* **do**

 Feed forward to find values at each node

 Backpropagate to correct weights:

Algorithm `algo()`

 Compute partial derivatives

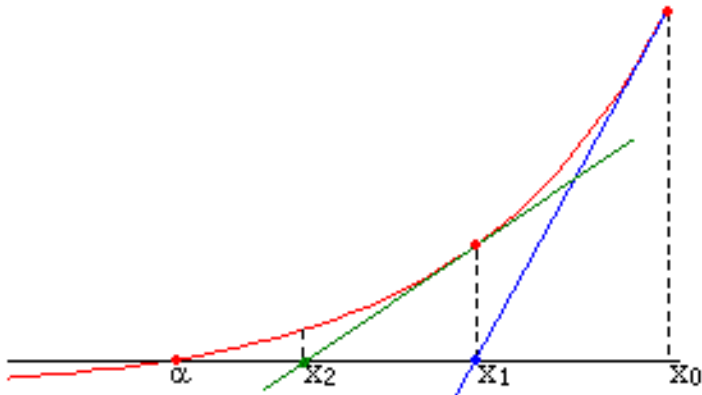
 Use partial derivatives to compute gradient

 Use gradient to update weights (classic gradient descent)

end

Backpropagation

Newton's Method



Backpropagation

Newton's Method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{f(x_n)}{\frac{df}{dx}(x_n)}$$

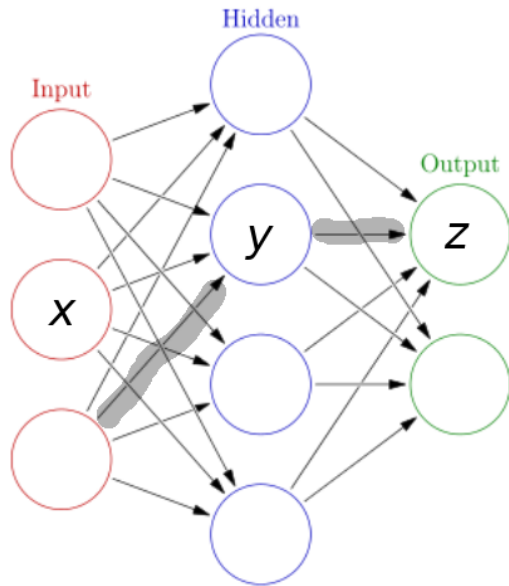
Backpropagation

Newton's Method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{f(x_n)}{\frac{df}{dx}(x_n)}$$

or

$$y = \frac{df}{dx}(x_n)(x - x_n) + f(x_n)$$



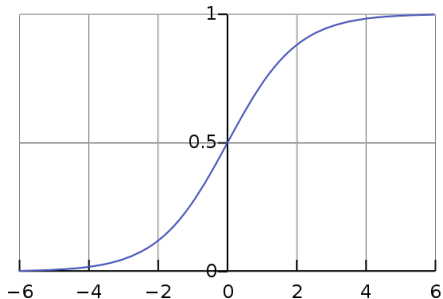
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \frac{\partial y}{\partial x}$$

Perceptron

Activation function

How the neuron decides when to fire.

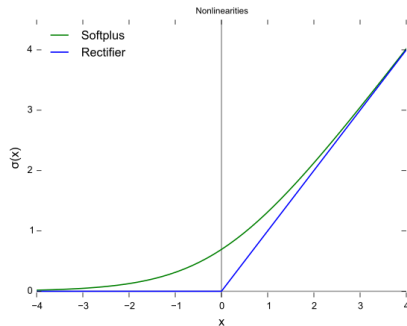
Vocabulary



Sigmoid

*By Qef (talk) - Created from scratch with gnuplot, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=4310325>*

Vocabulary

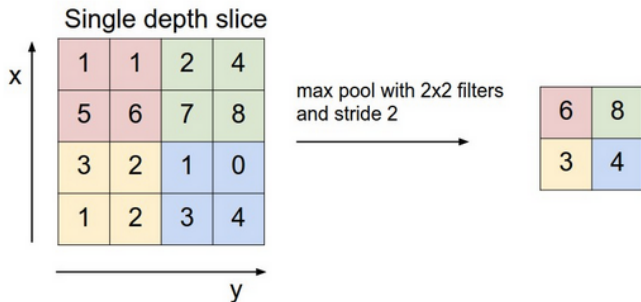


ReLU and Softmax

Rectified Linear Unit

Wikimedia Commons

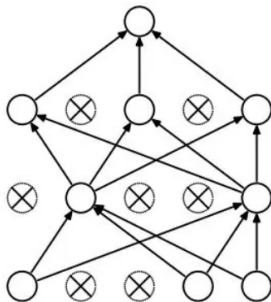
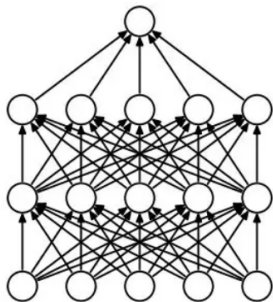
Vocabulary



Pooling

max pooling
downsampling

Vocabulary



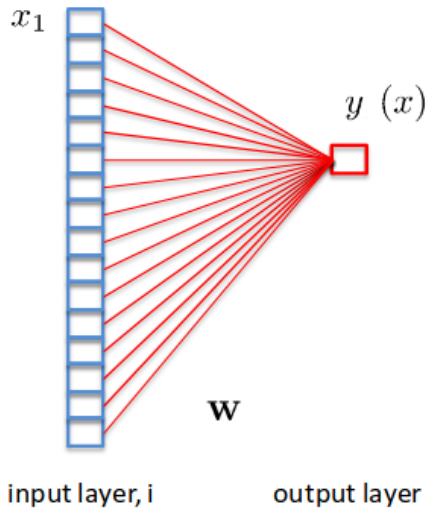
Dropout

Multilayer Perceptron (MLP)

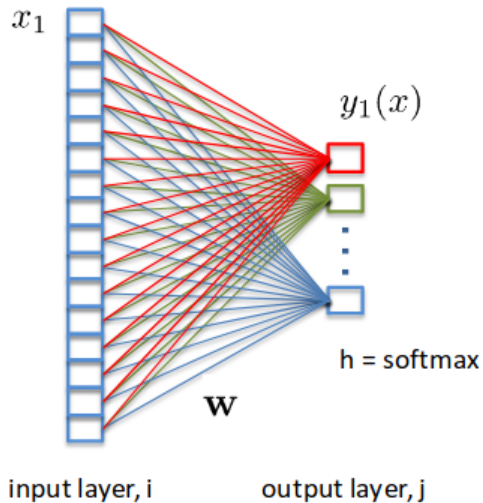
MLP

- ① Multiple layers
- ② First layer is linear
- ③ Later layers use non-linear activation functions (typically sigmoid)
- ④ Feedforward
- ⑤ Can find non-linear separators

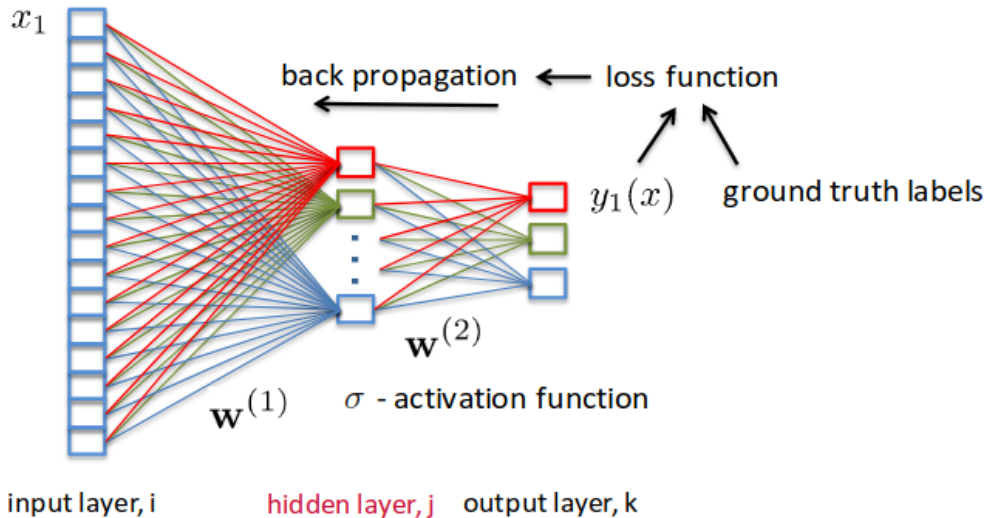
MLP



MLP



MLP



Example: MNIST

Example: simple classification tasks

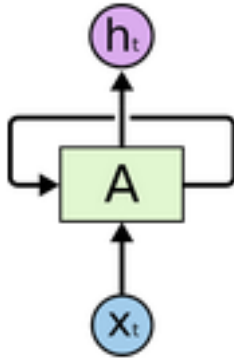
Example: time series

Recurrent Neural Networks

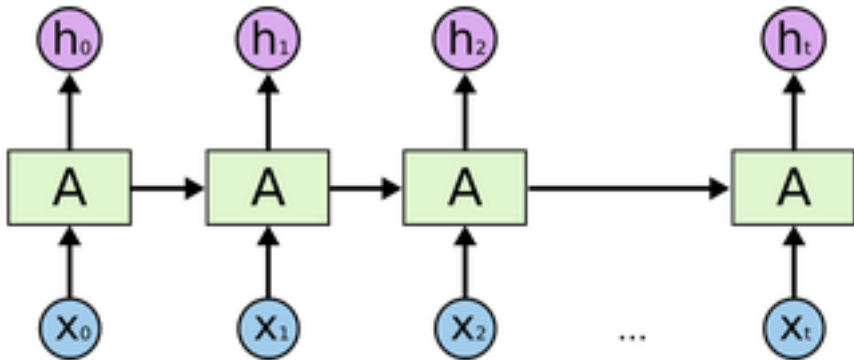
loops

loops

so also memory



<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>



<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

ConvNets

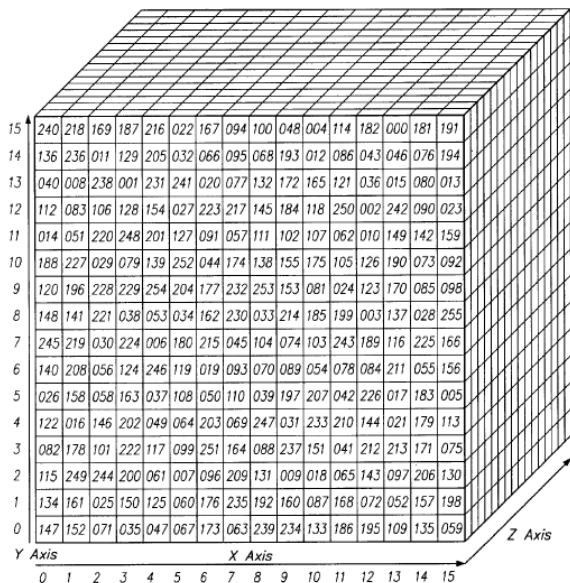
Tensors

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Tensors

$$\begin{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} & \begin{pmatrix} 3 \\ 4 \end{pmatrix} & \begin{pmatrix} 5 \\ 6 \end{pmatrix} \\ \begin{pmatrix} 7 \\ 8 \end{pmatrix} & \begin{pmatrix} 9 \\ 10 \end{pmatrix} & \begin{pmatrix} 11 \\ 12 \end{pmatrix} \end{pmatrix}$$

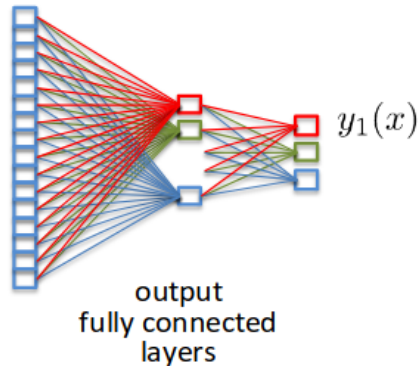
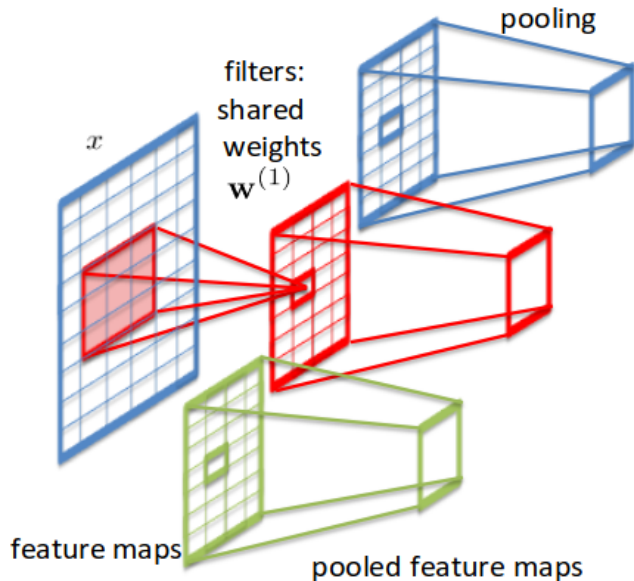
Tensors



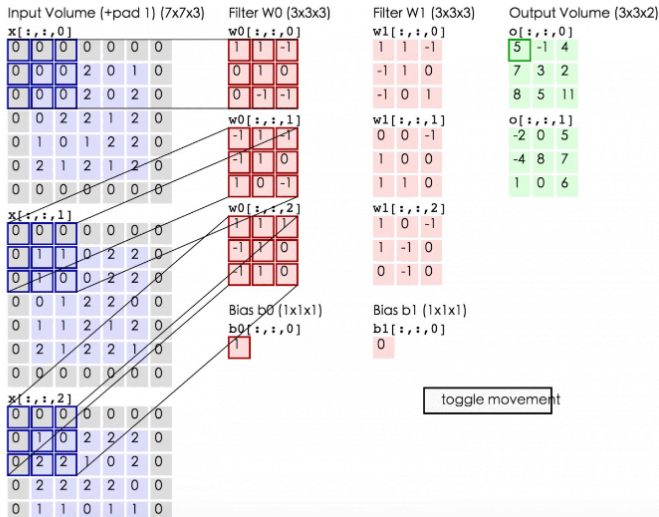
Convolutional Neural Networks

The neurons are convolutions

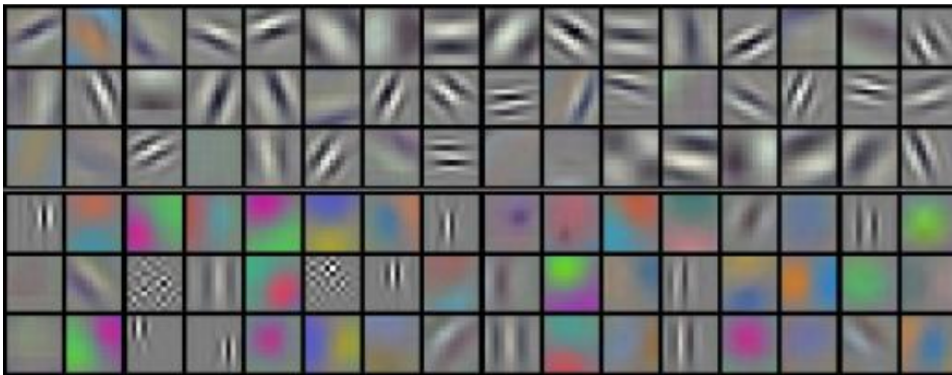
ConvNets



Convolutional Layers



ConvNets



Example: images

Transfer Learning

- Pre-trained models
- Only retrain the classifier at the end

Example: PlacesVGG (2015)

PlacesVGG (Places2 2015)

https://static.turi.com/models/.../places_vgg_16-1.0.tar.gz

This model is trained with VGG-16 architecture, on the Places2 dataset. The Places2 dataset contains 8 million images of 400 different scene categories. More details about the dataset can be found at <http://places2.csail.mit.edu/>

https://turi.com/products/create/docs/graphlab.mxnet.pretrained_image_model.html

The Beginning

Perspectives

- Data science is iterative
- Start simple, get better

Perspectives

- Data science is iterative
- Start simple, get better

Sometimes no business case to do more.

Risks

- Nothing is guaranteed, but competitors are innovating and experimenting
- Examples from past projects can help, but often leads to “this is different”

Success

Think early about how to measure success.

Success

Think early about how to measure success.

The definition of success will evolve.

questions?