Beginner's guide to neural networks

Tom Charnock

Resources

Overview
neuralnetworksanddeeplearning.com

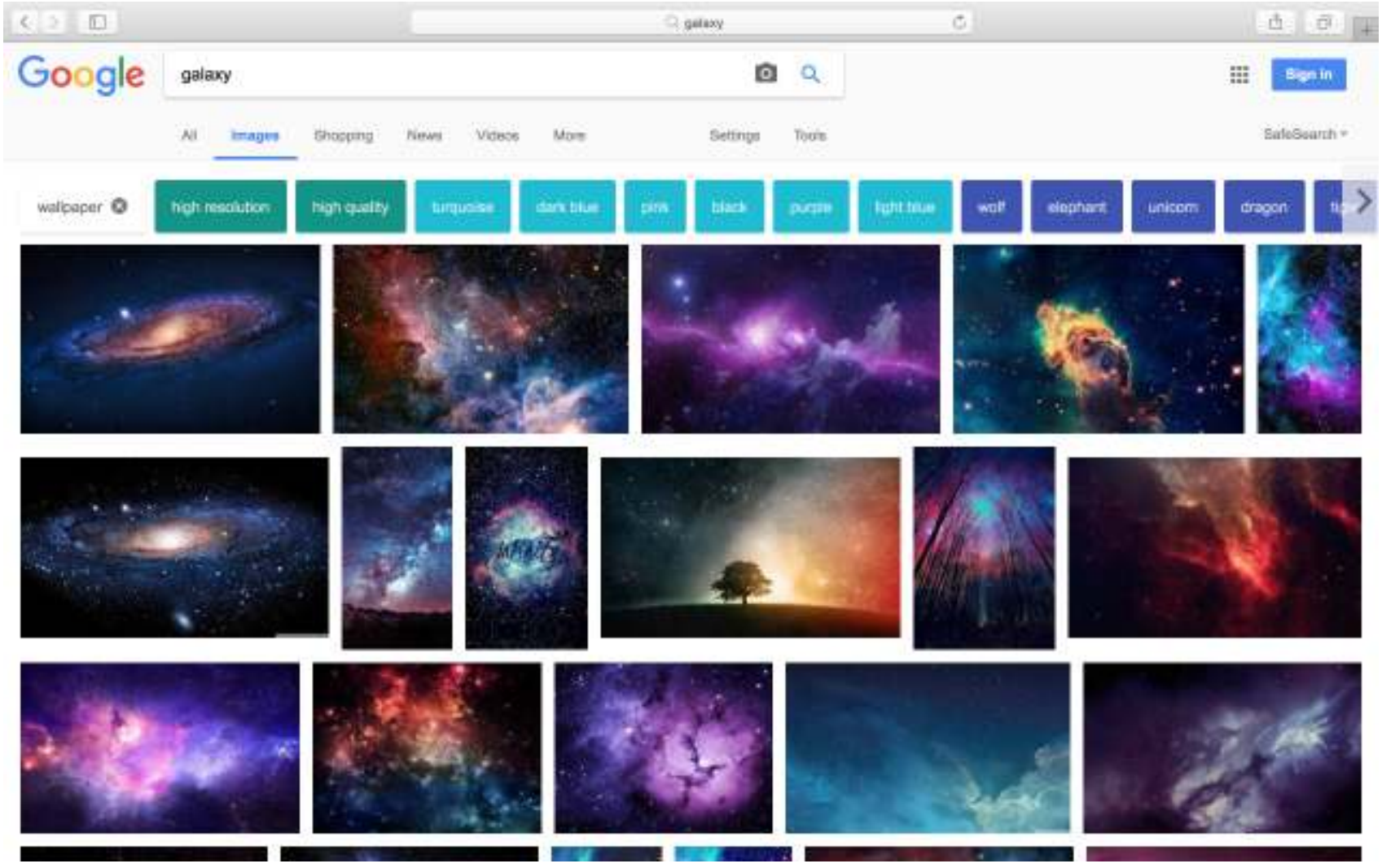
Code examples
keras.io

Convolutional neural networks
cs231n.github.io

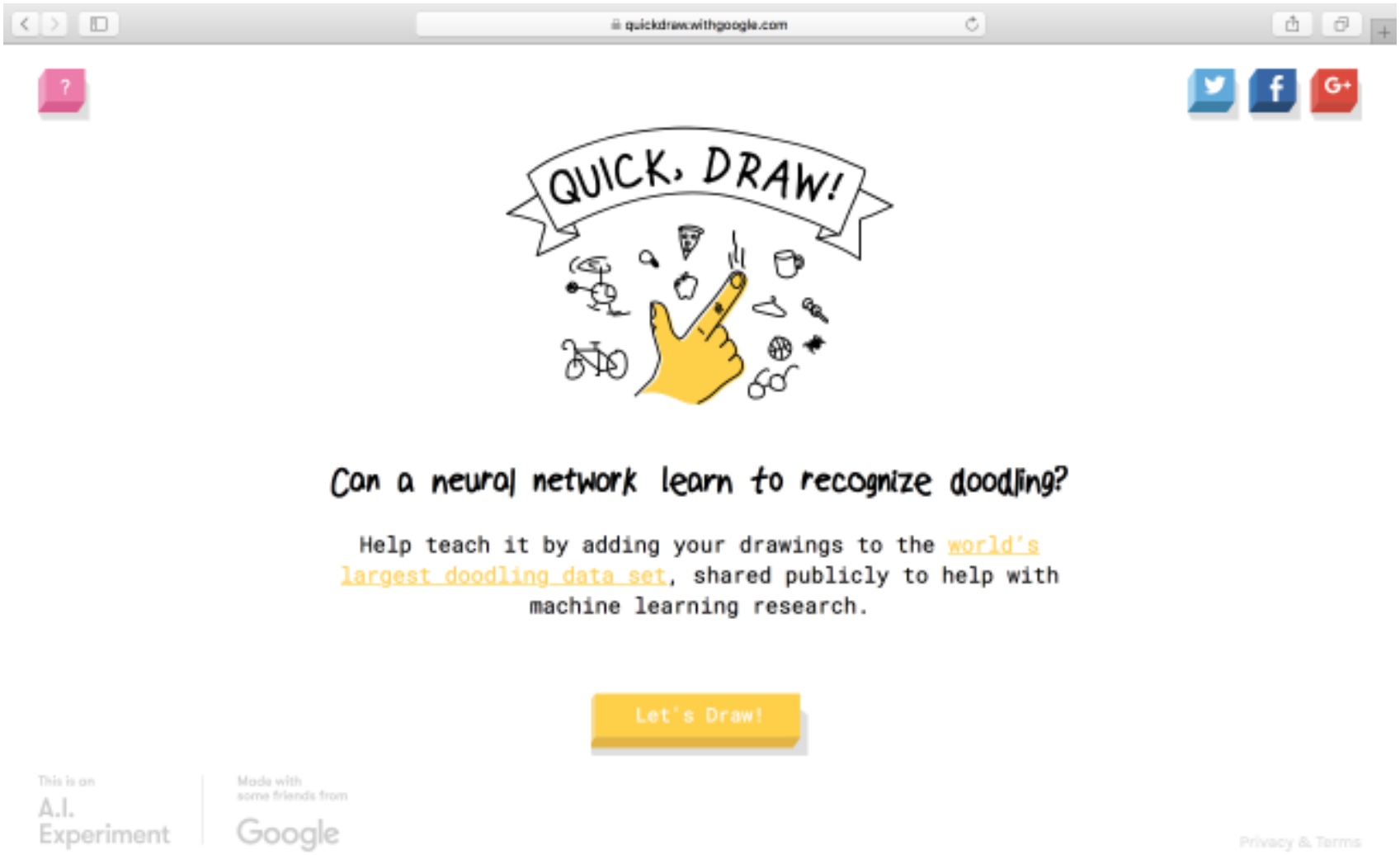
Interesting places to checkout

Google

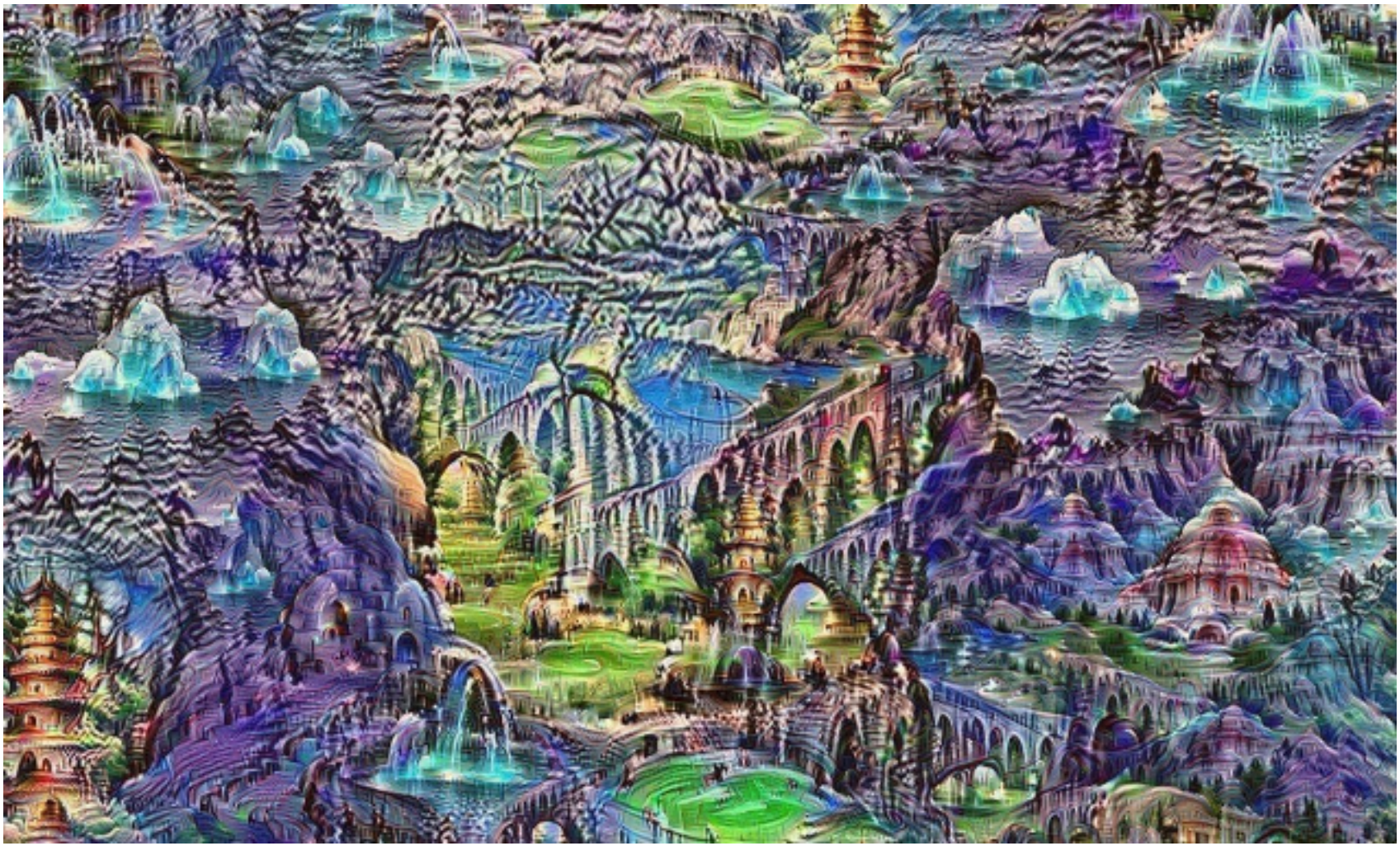
Interesting places to checkout - images.google.com



Interesting places to checkout - quickdraw.withgoogle.com



Interesting things to checkout - Inceptionism



Languages

Python	Not Python
TensorFlow	
Theano	R
Keras	Java
Lasagne	MATLAB
Caffe	

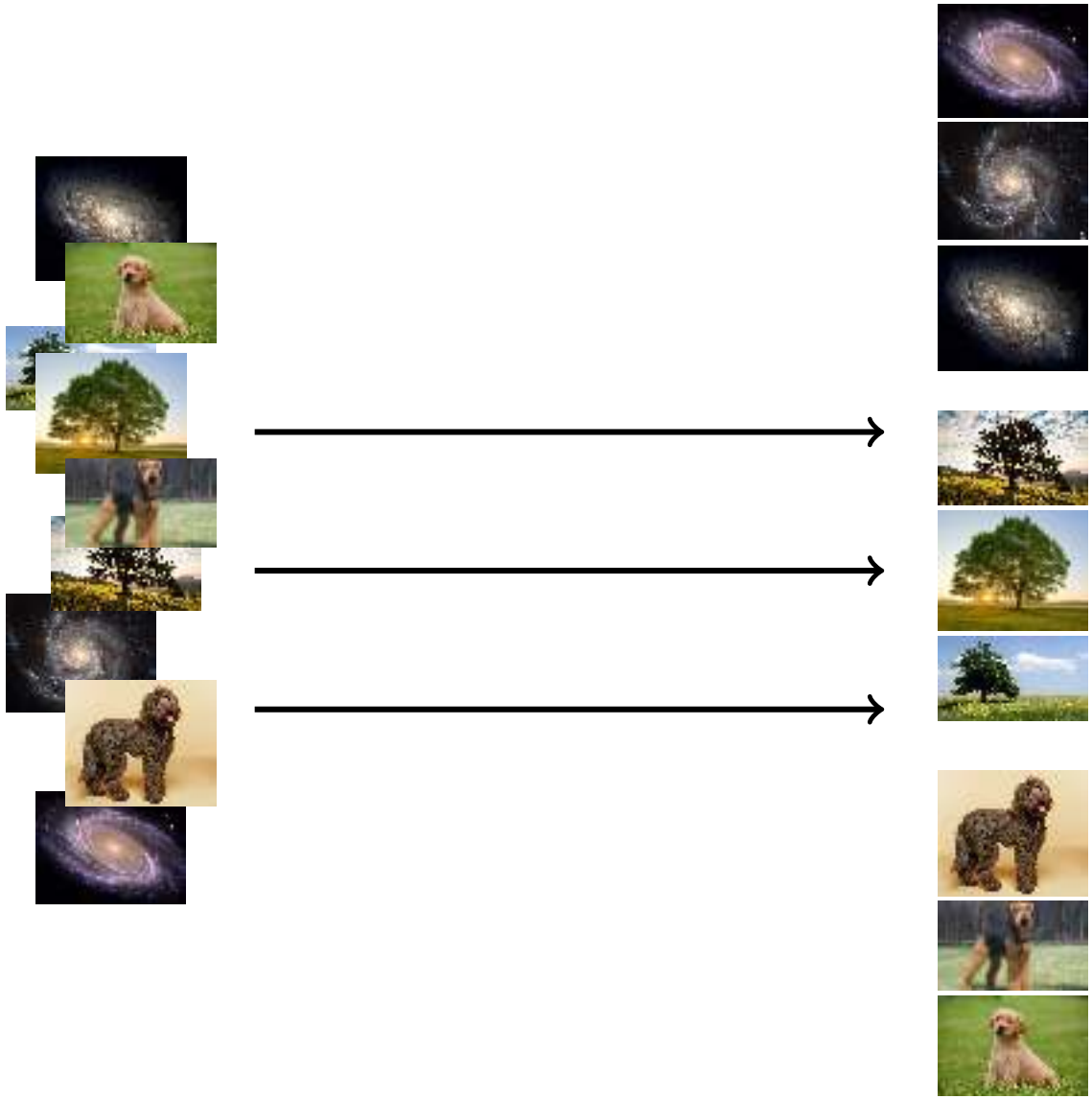
Supervised machine learning

→ Galaxy

→ Dog

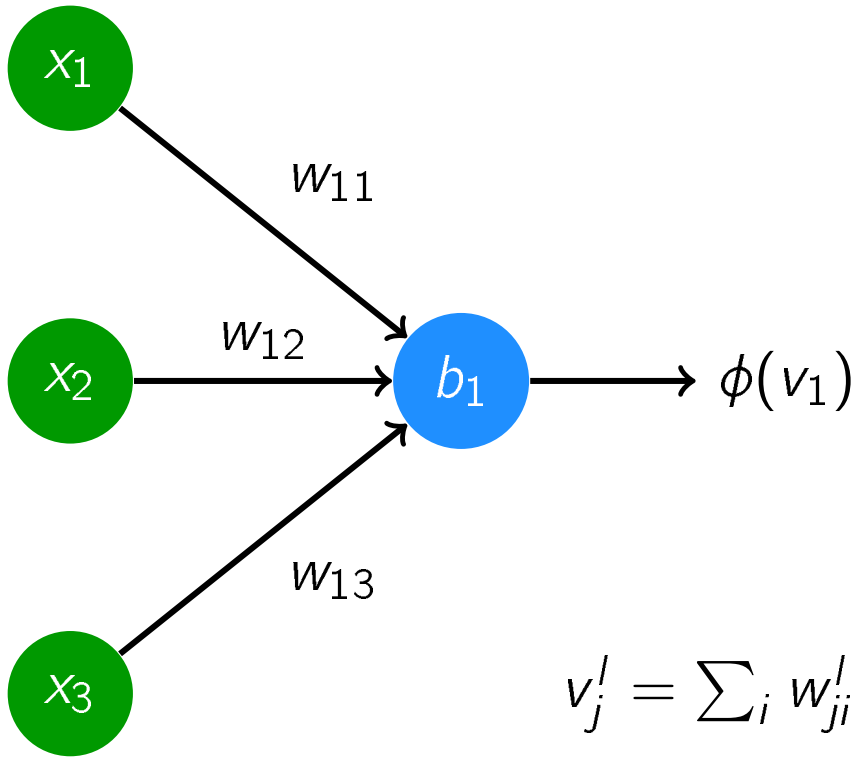
→ Tree

Unsupervised machine learning



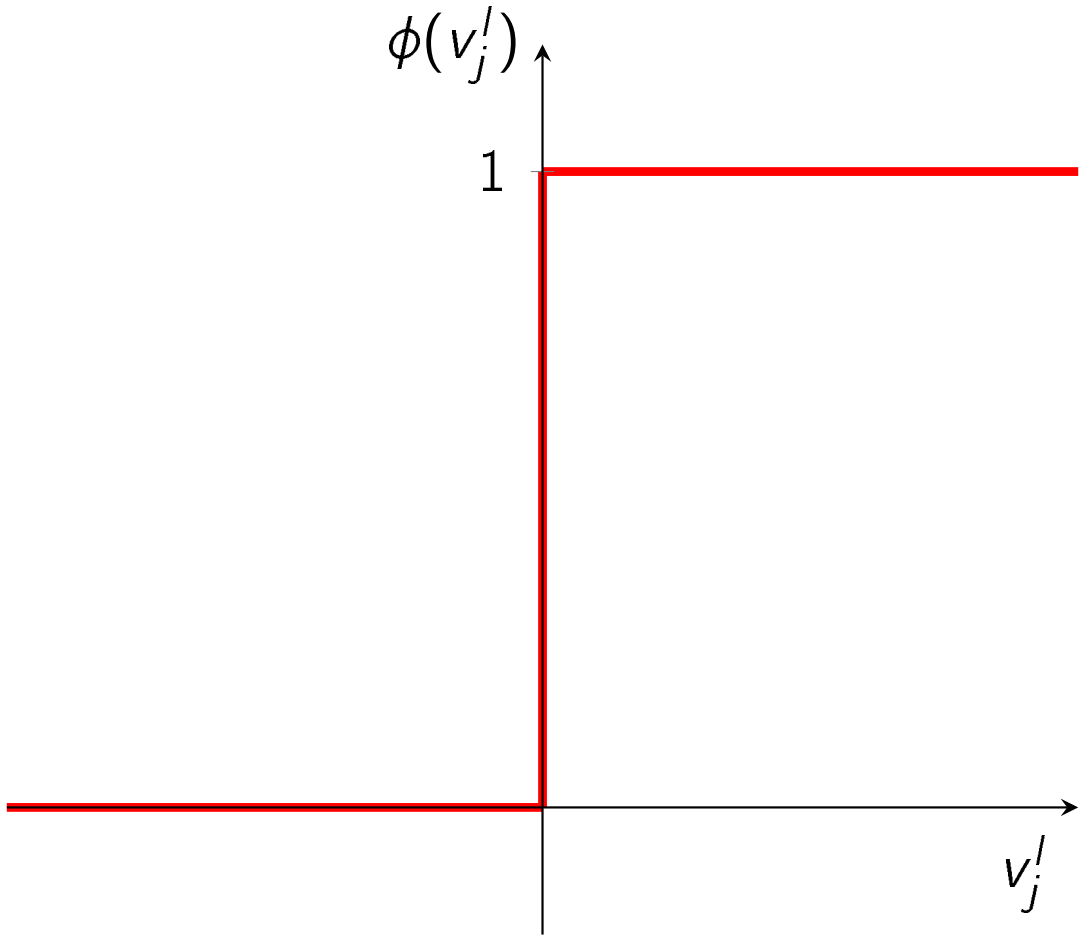
Reinforcement machine learning

Perceptrons

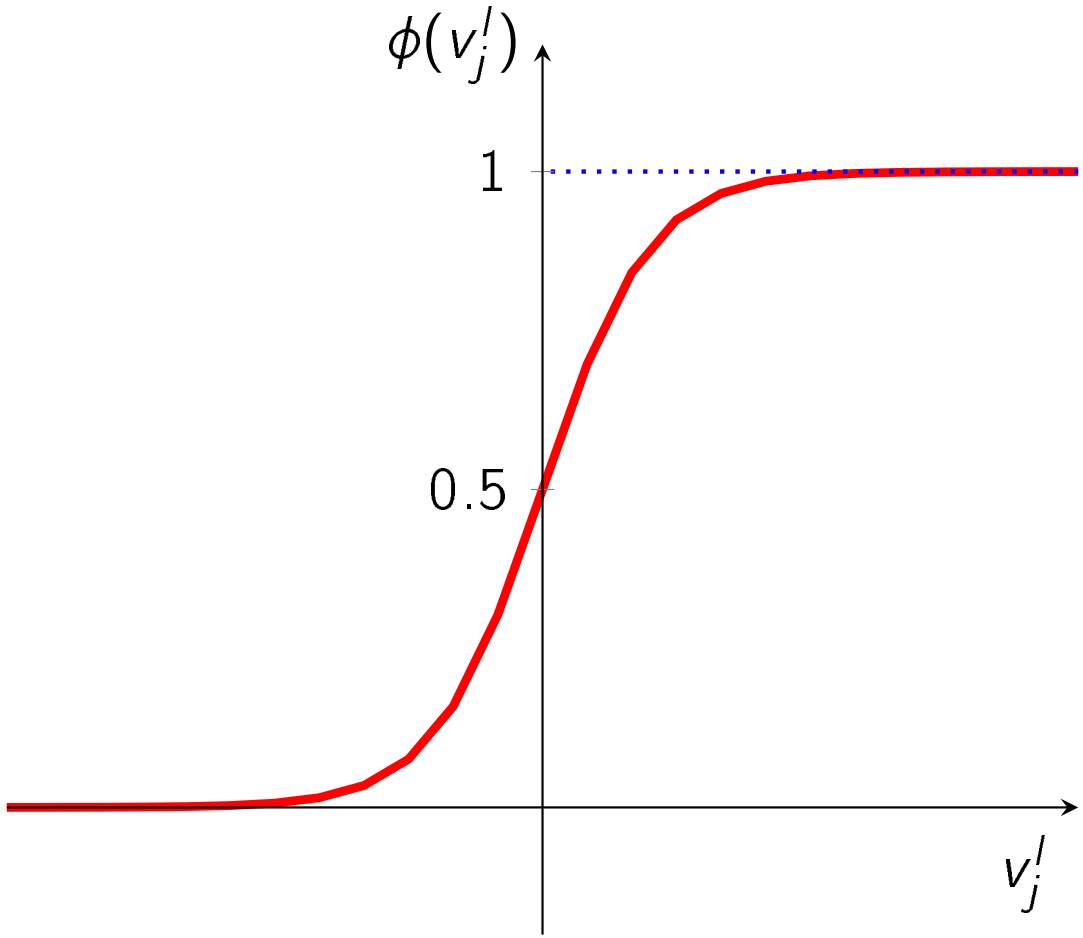


$$v_j^l = \sum_i w_{ji}^l x_i^l + b_j^l$$

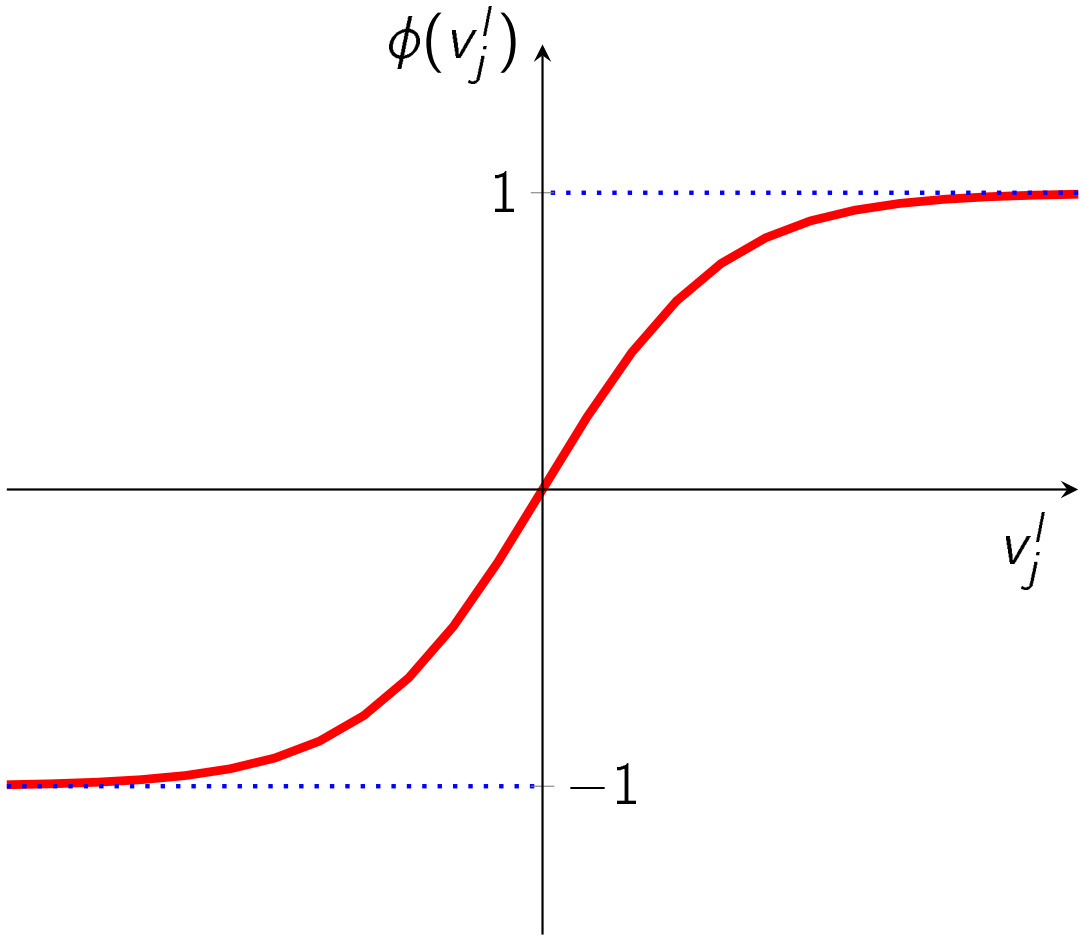
Activation



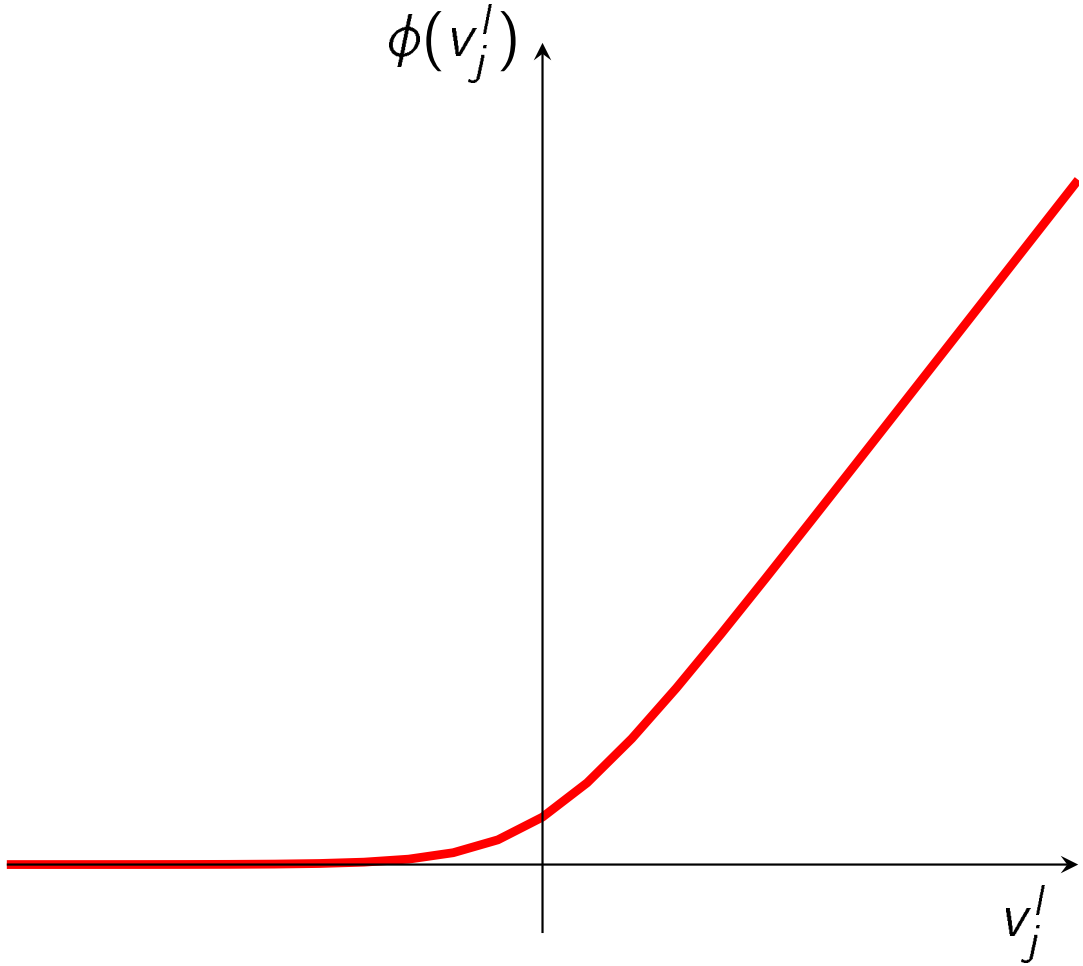
Activation



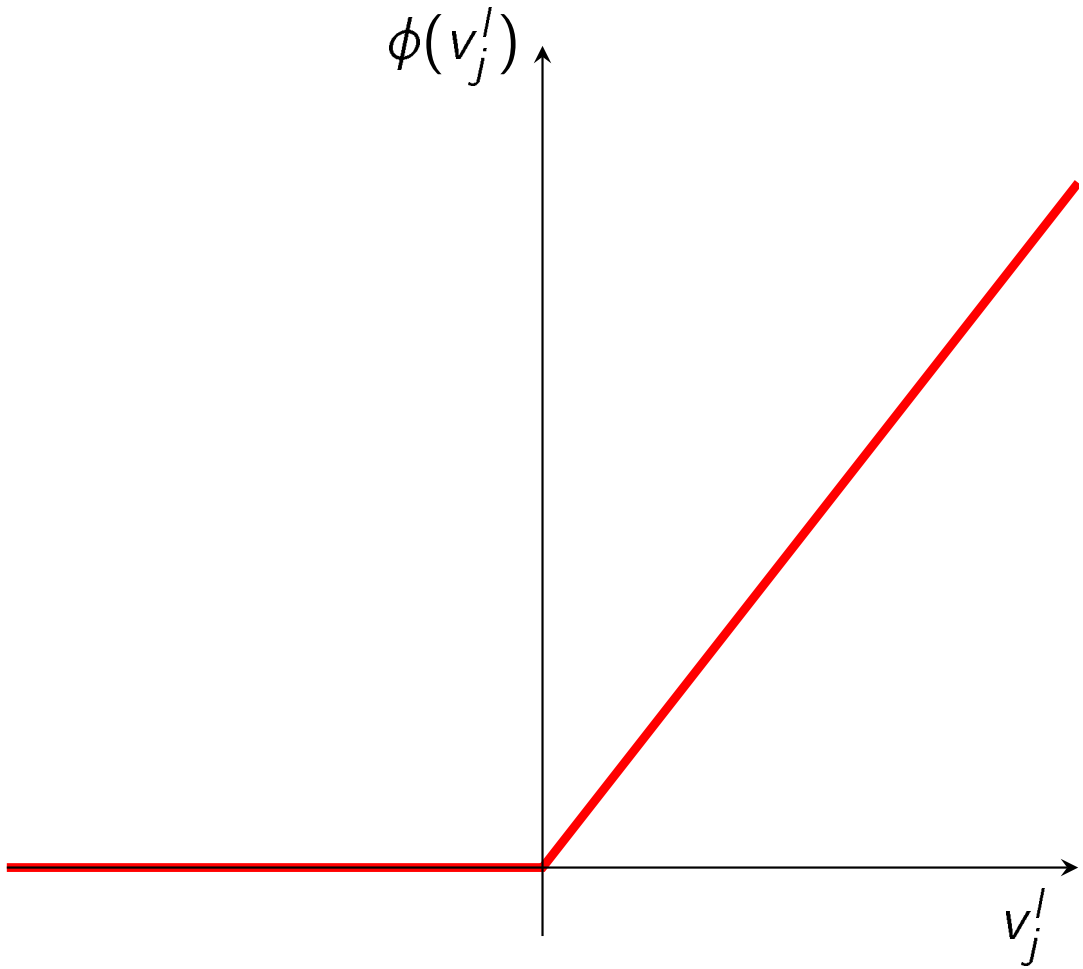
Activation



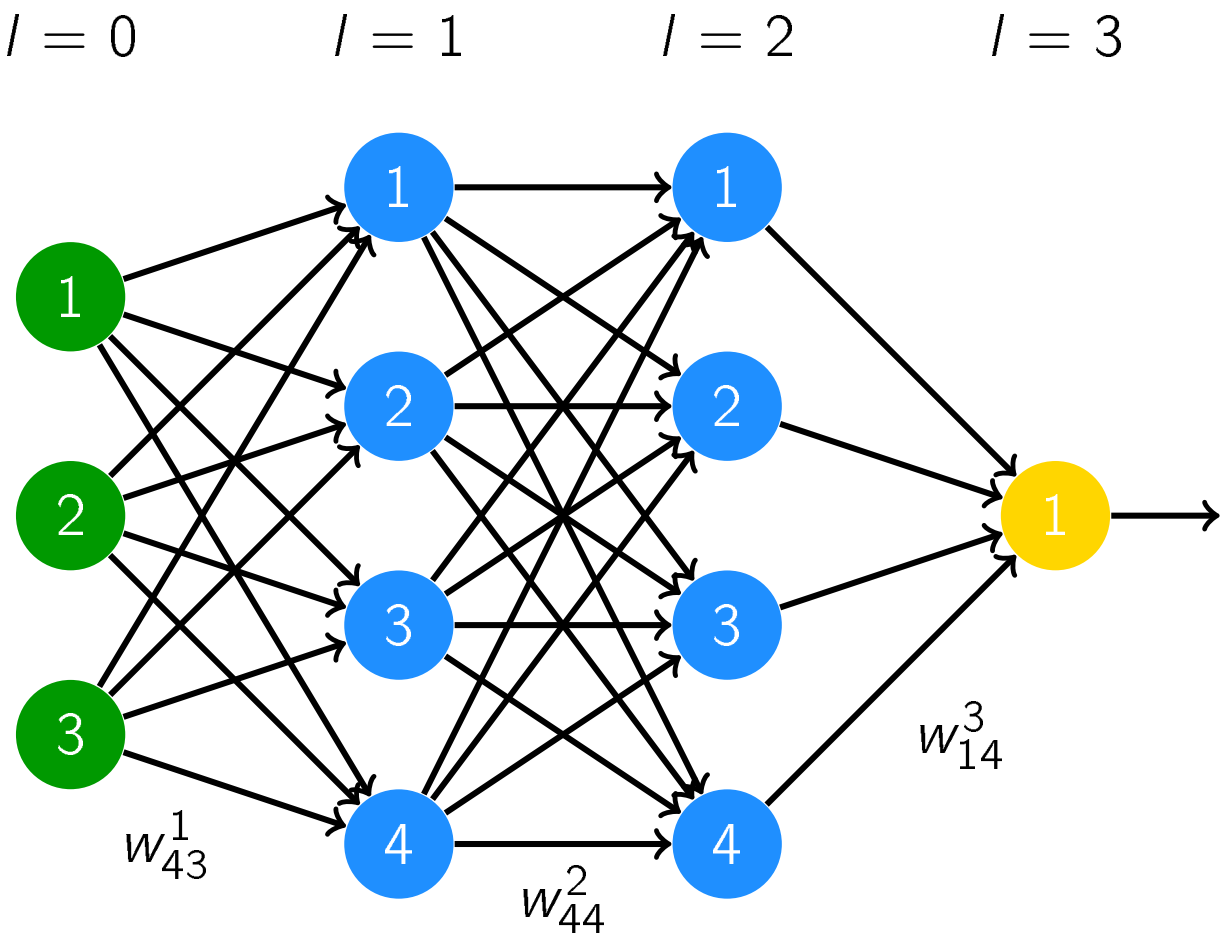
Activation



Activation

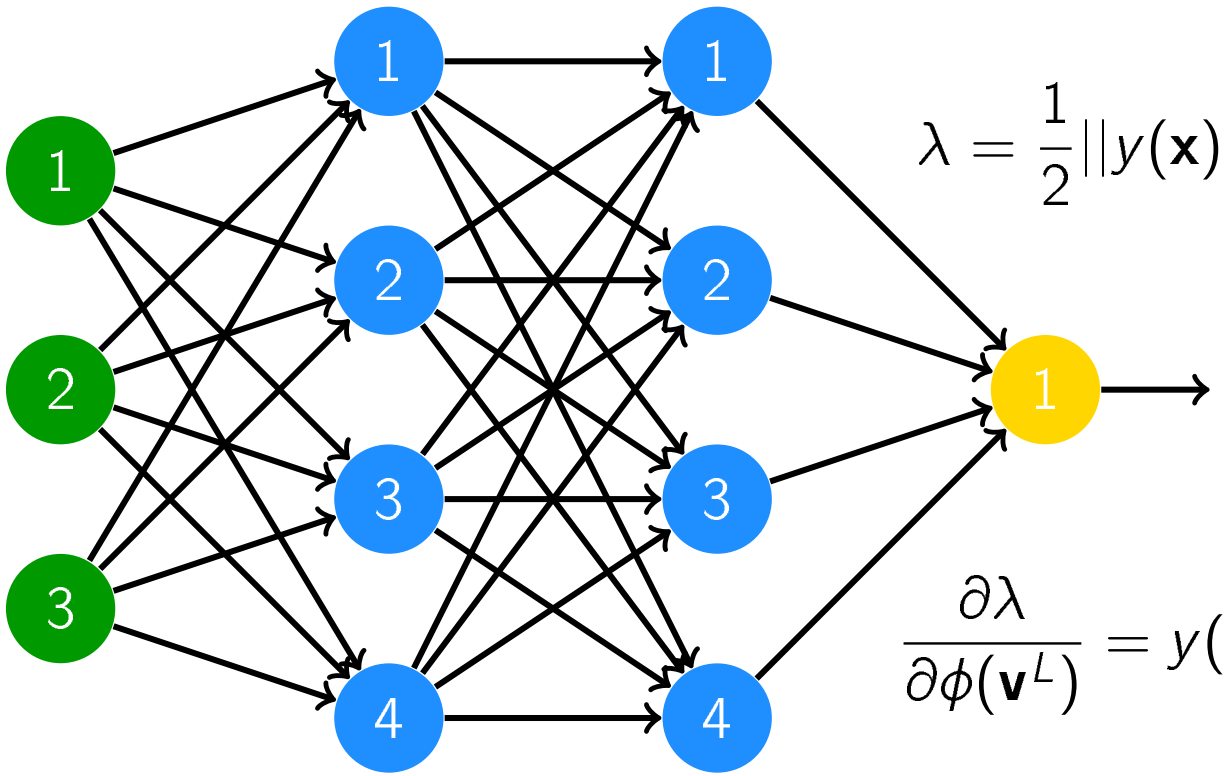


Multilayer perceptrons



$\mathbf{x} = \phi(\mathbf{v}^0)$ $\phi(\mathbf{v}^1)$ $\phi(\mathbf{v}^2)$ $\phi(\mathbf{v}^3)$

Back propagation

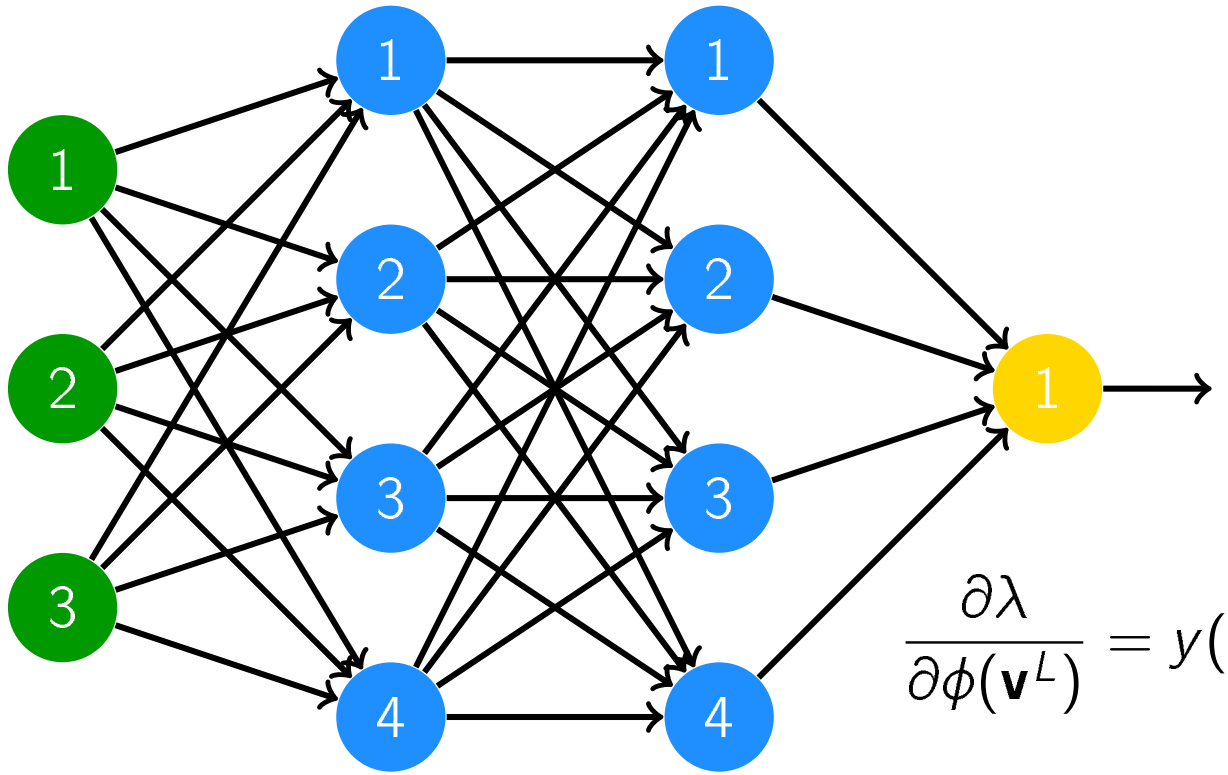


$$\lambda = \frac{1}{2}||y(\mathbf{x}) - \phi(\mathbf{v}^L)||^2$$

$$\frac{\partial \lambda}{\partial \phi(\mathbf{v}^L)} = y(\mathbf{x}) - \phi(\mathbf{v}^L)$$

Back propagation

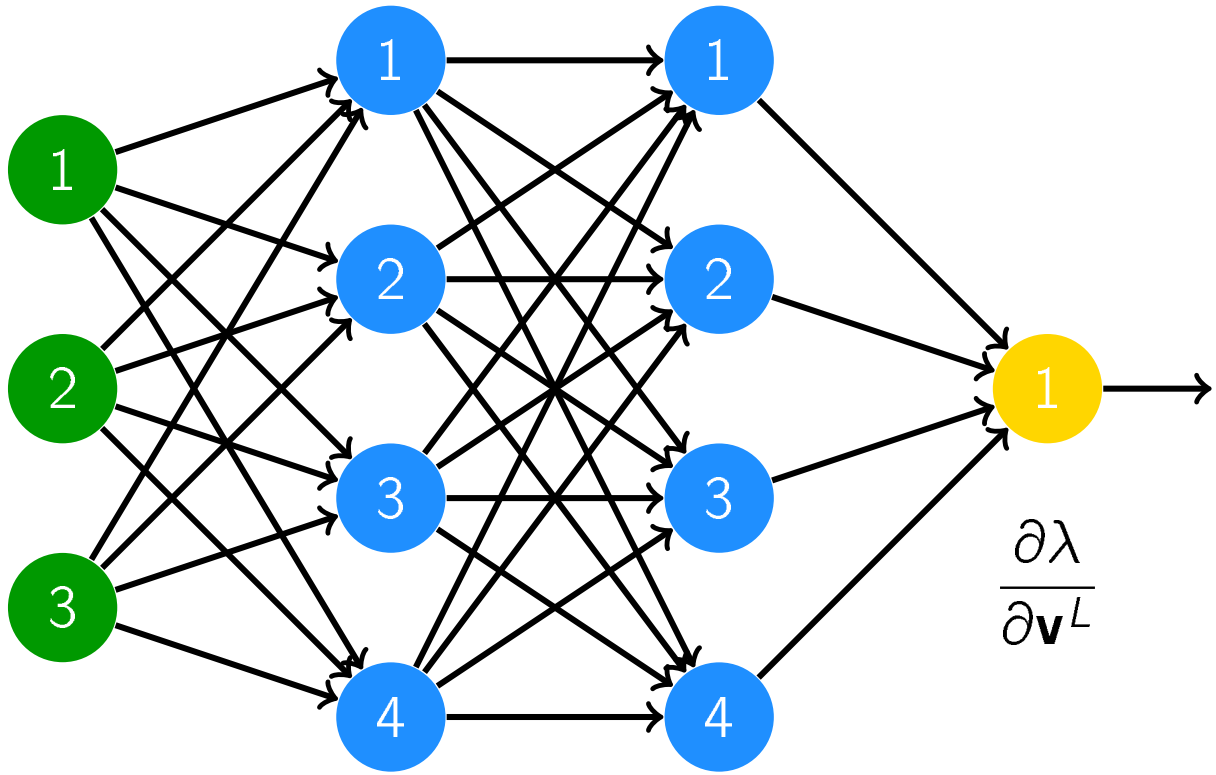
$$\frac{\partial \lambda}{\partial \mathbf{v}^L} = \frac{\partial \lambda}{\partial \phi(\mathbf{v}^L)} \odot \frac{\partial \phi(\mathbf{v}^L)}{\partial \mathbf{v}^L}$$



$$\frac{\partial \lambda}{\partial \phi(\mathbf{v}^L)} = y(\mathbf{x}) - \phi(\mathbf{v}^L)$$

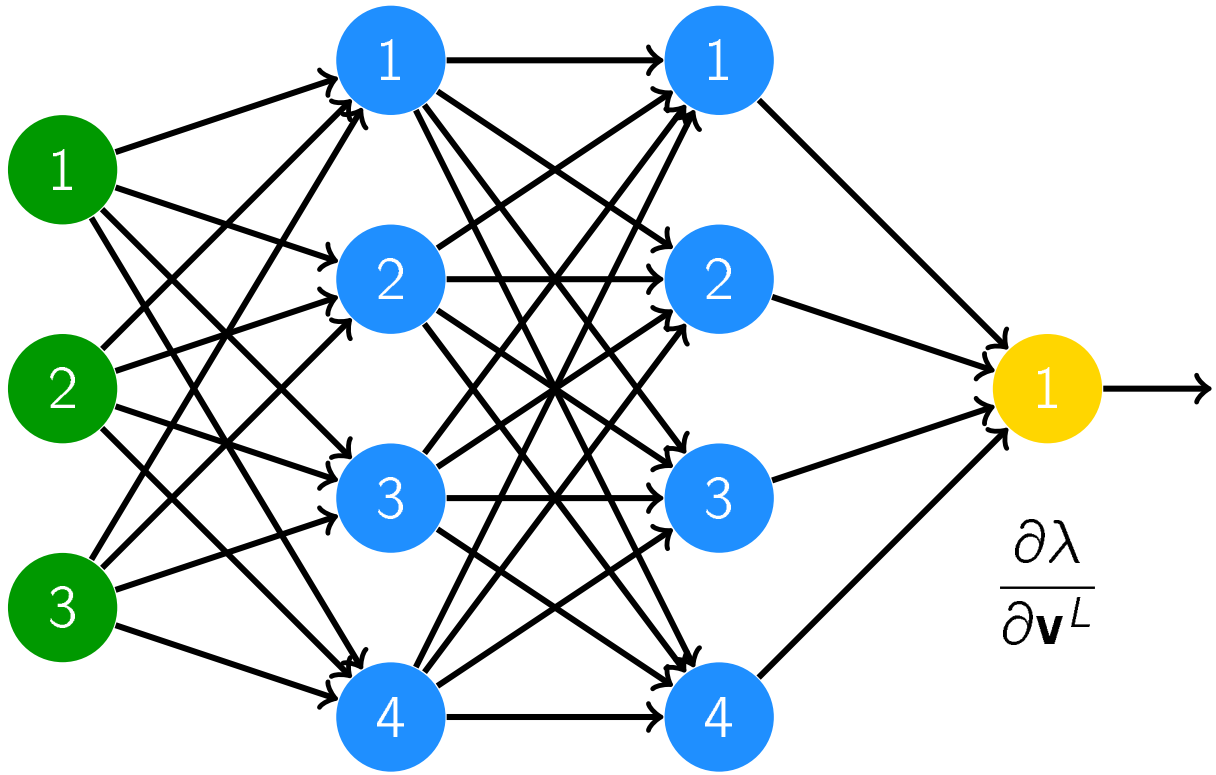
Back propagation

$$\frac{\partial \lambda}{\partial \mathbf{v}^L} = \frac{\partial \lambda}{\partial \phi(\mathbf{v}^L)} \odot \frac{\partial \phi(\mathbf{v}^L)}{\partial \mathbf{v}^L}$$



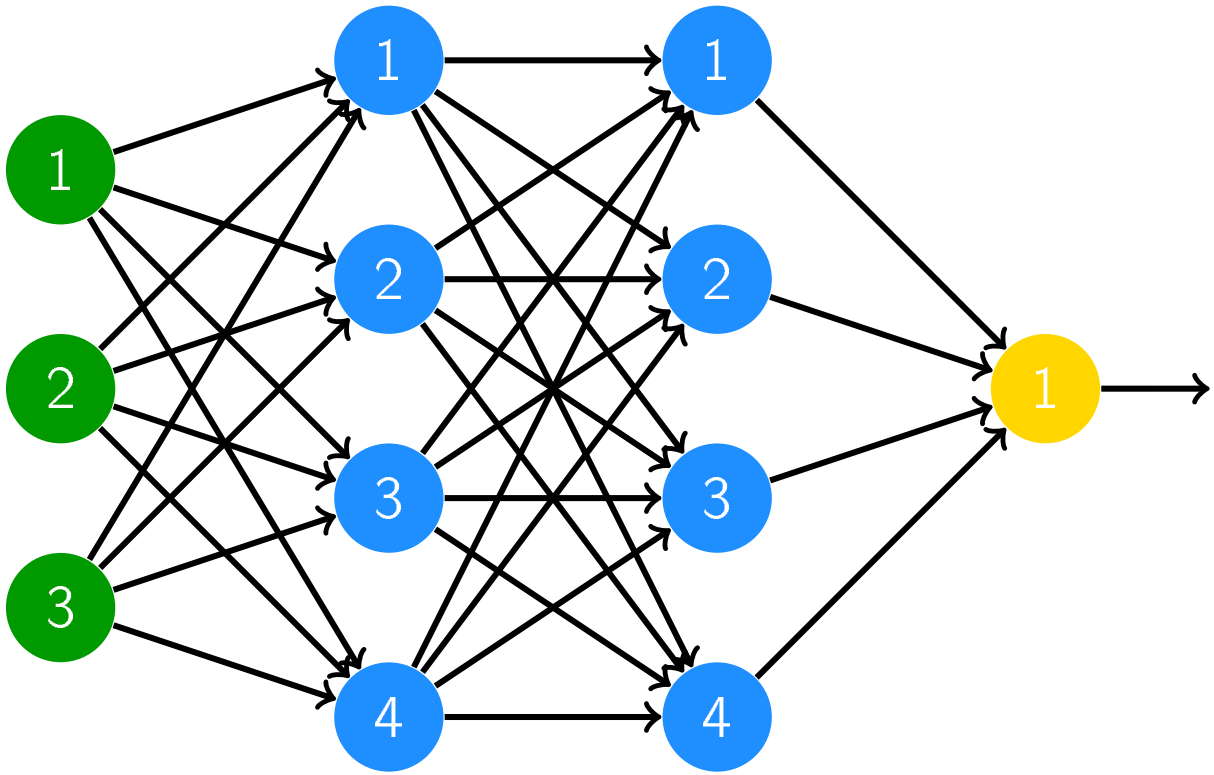
Back propagation

$$\frac{\partial \lambda}{\partial \mathbf{v}^{L-1}} = \left(\mathbf{w}^{L}\right)^{T} \frac{\partial \lambda}{\partial \mathbf{v}^{L}} \odot \frac{\partial \phi(\mathbf{v}^{L-1})}{\partial \mathbf{v}^{L-1}}$$



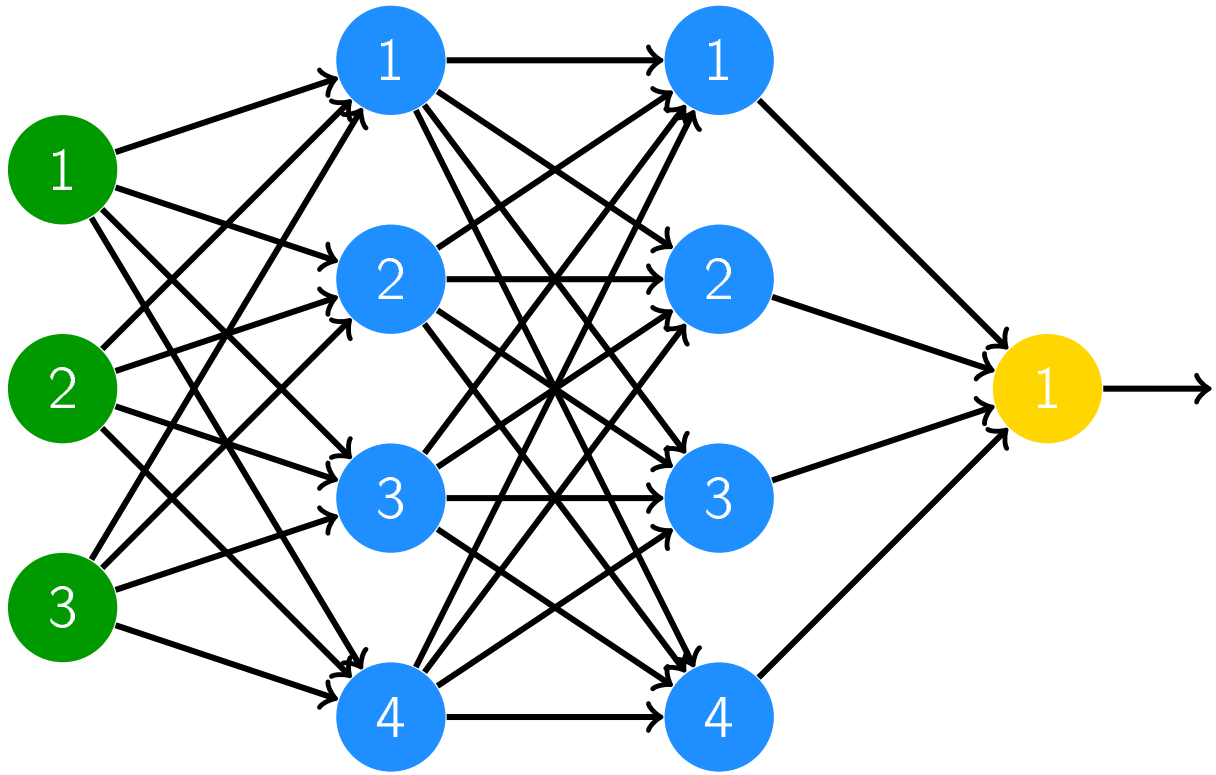
Back propagation

$$\frac{\partial \lambda}{\partial \mathbf{v}^l} = \left(\mathbf{w}^{l+1}\right)^T \frac{\partial \lambda}{\partial \mathbf{v}^{l+1}} \odot \frac{\partial \phi(\mathbf{v}^l)}{\partial \mathbf{v}^l}$$



Back propagation

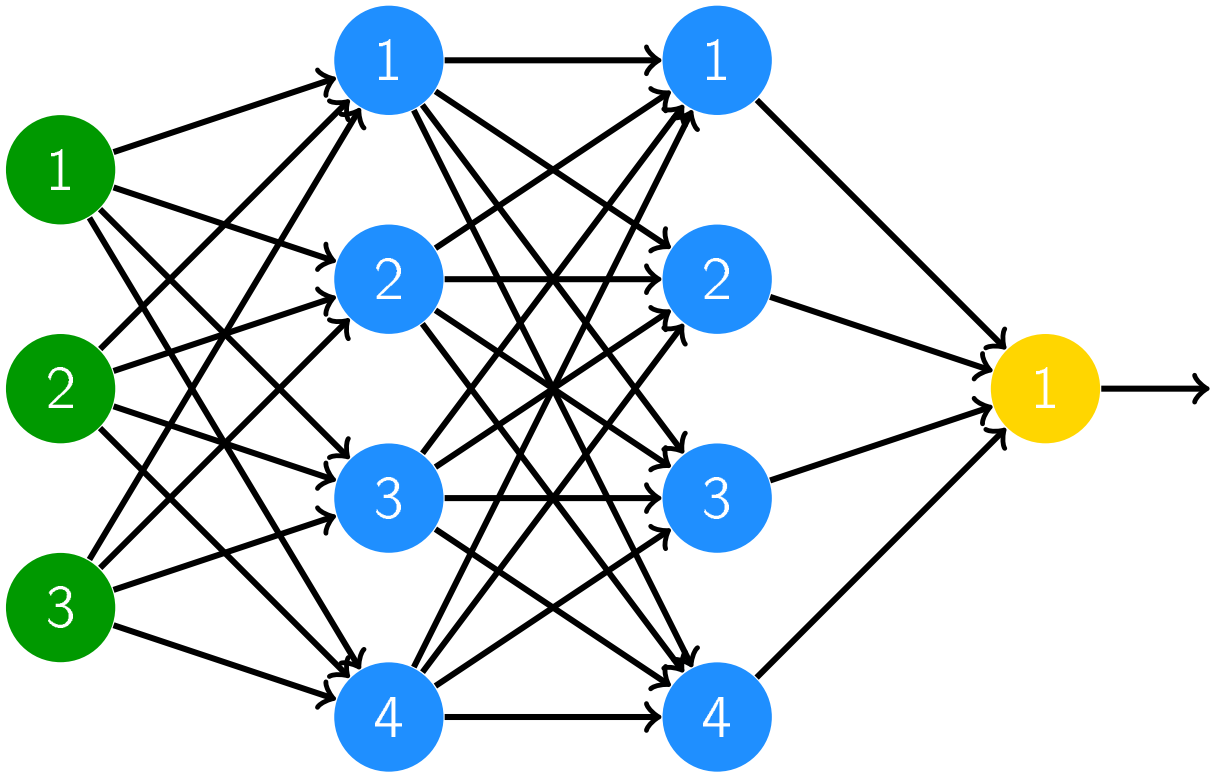
$$\frac{\partial \lambda}{\partial \mathbf{w}^l} = \phi\left(\mathbf{v}^{l-1}\right) \frac{\partial \lambda}{\partial \mathbf{v}^l} \quad \text{and} \quad \frac{\partial \lambda}{\partial \mathbf{b}^l} = \frac{\partial \lambda}{\partial \mathbf{v}^l}$$



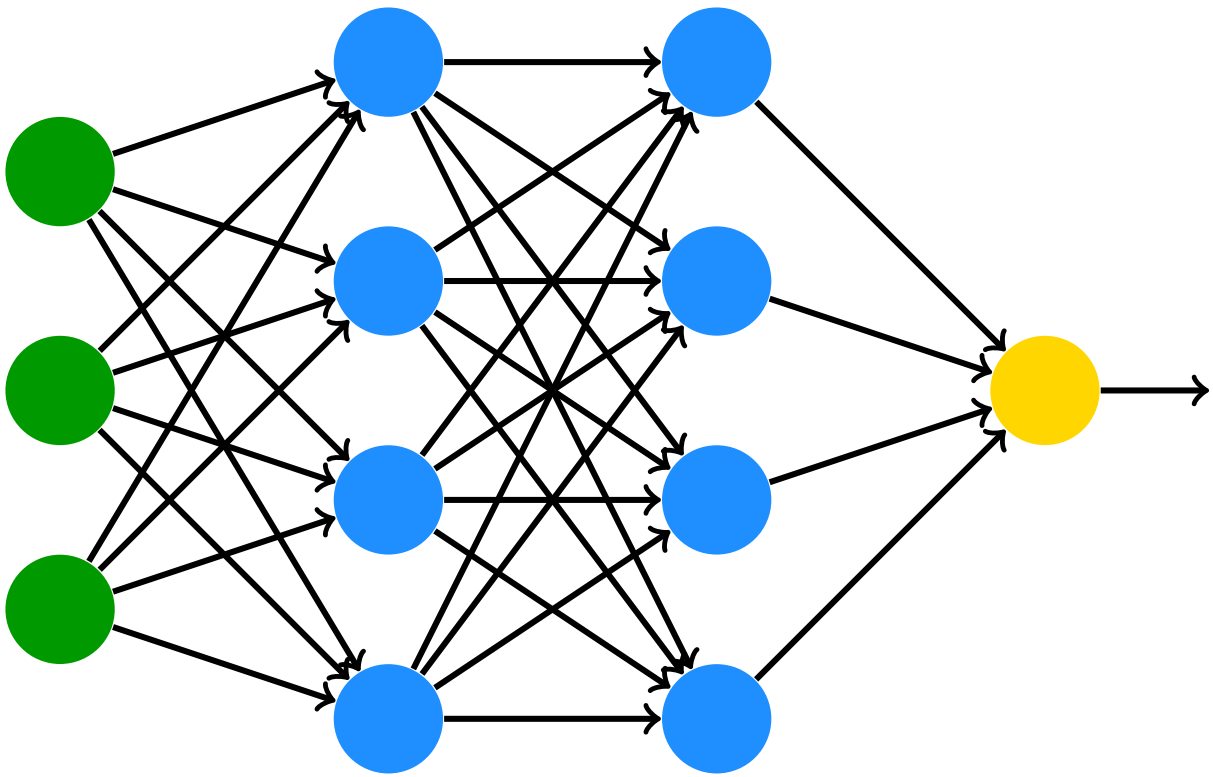
$$\frac{\partial \lambda}{\partial \mathbf{v}^1} \qquad \frac{\partial \lambda}{\partial \mathbf{v}^2} \qquad \frac{\partial \lambda}{\partial \mathbf{v}^L}$$

Back propagation

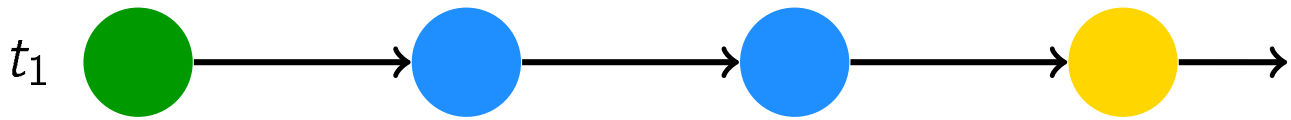
$$\mathbf{w}^l \rightarrow \mathbf{w} - \eta \frac{\partial \lambda}{\partial \mathbf{w}^l} \quad \text{and} \quad \mathbf{b}^l \rightarrow \mathbf{b}^l - \eta \frac{\partial \lambda}{\partial \mathbf{b}^l}$$



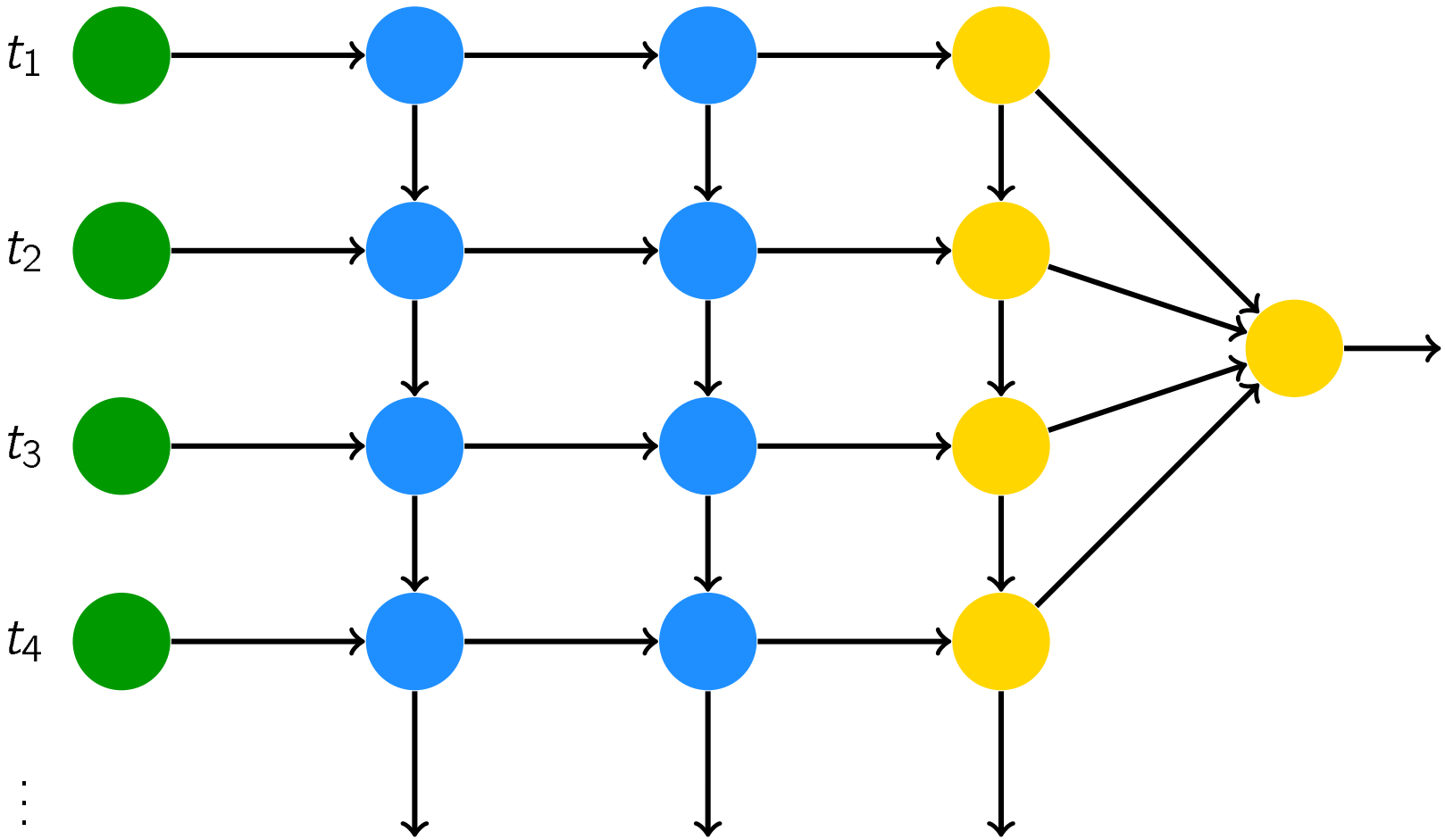
Artificial neural network



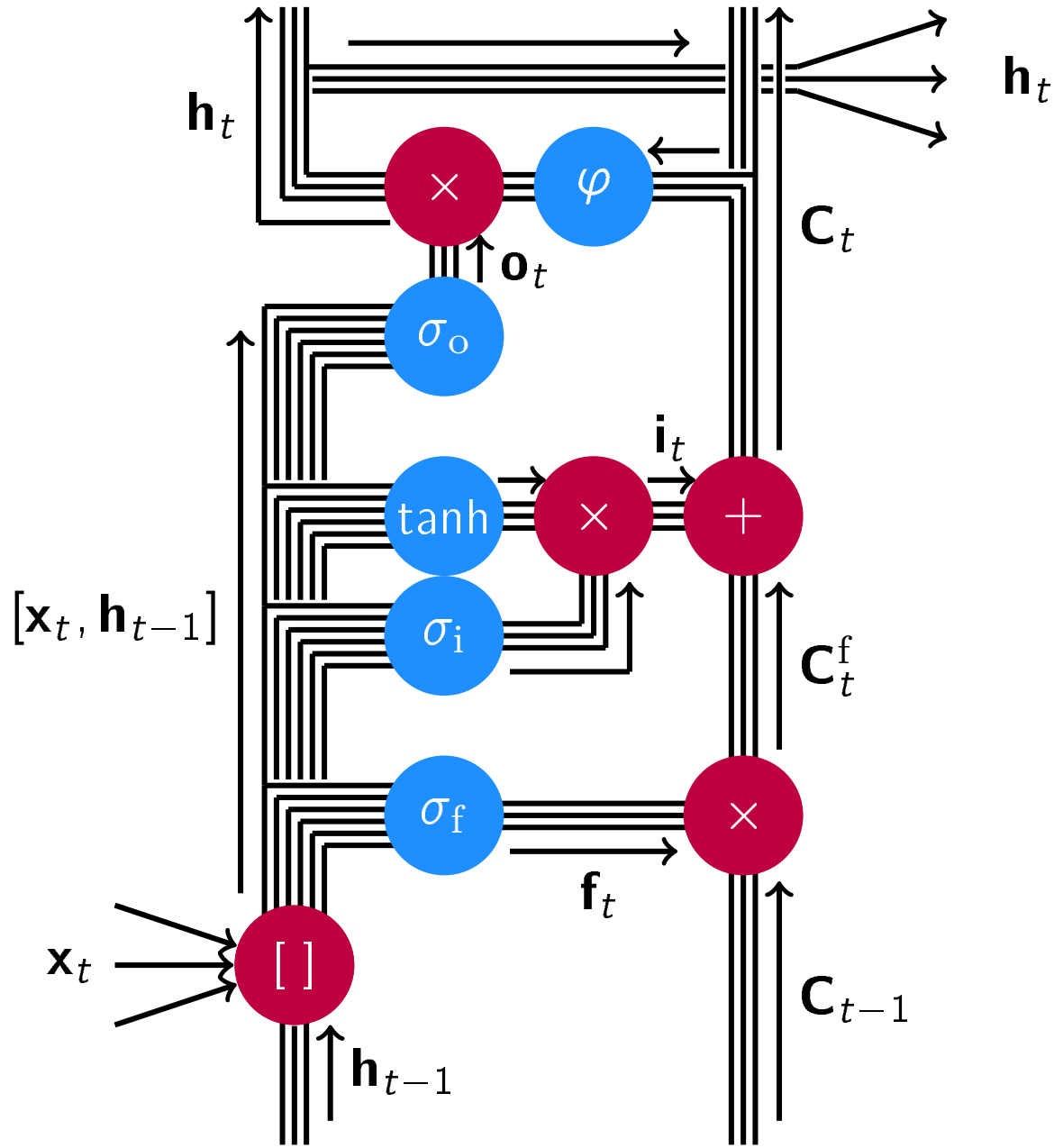
Recurrent neural networks



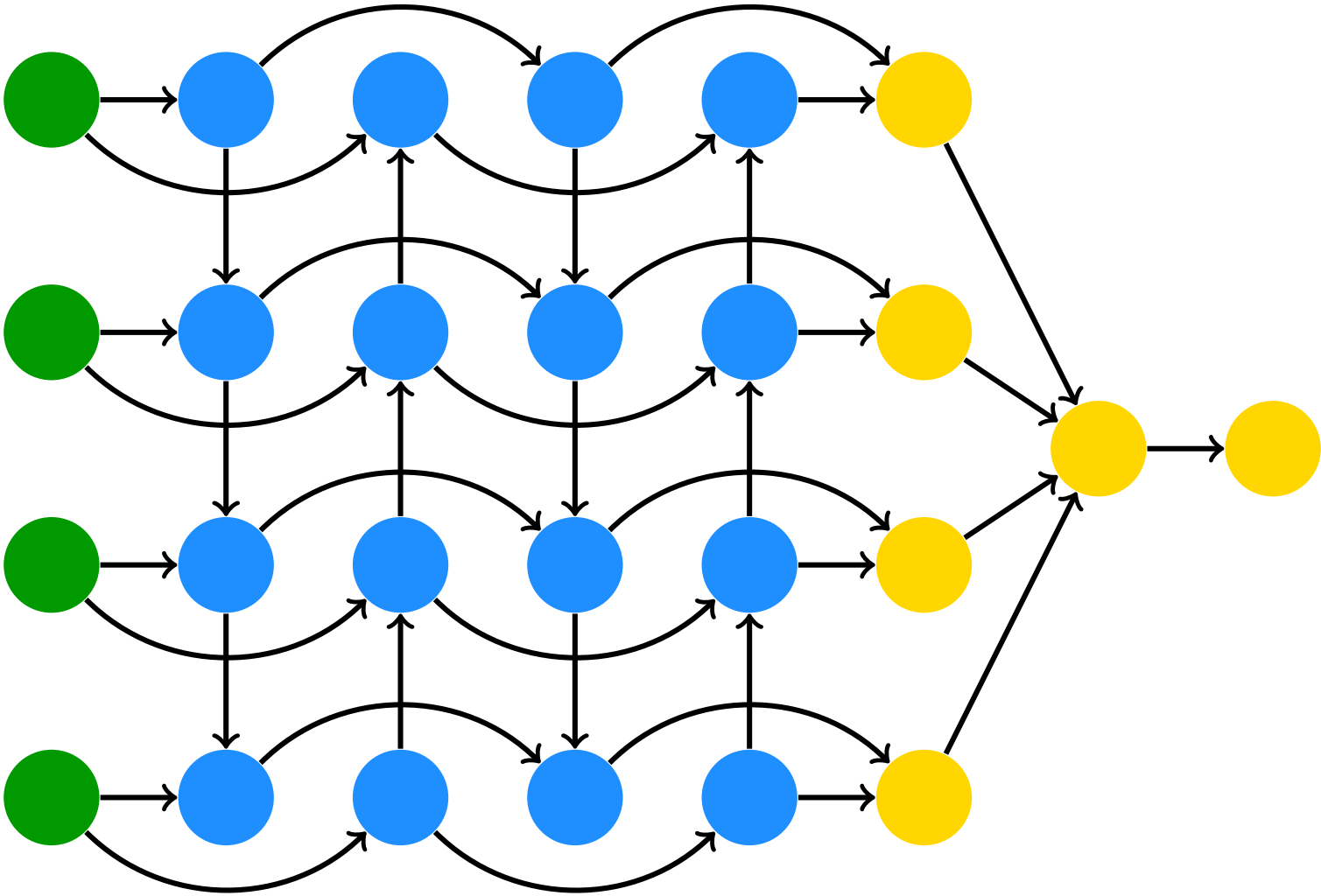
Recurrent neural network



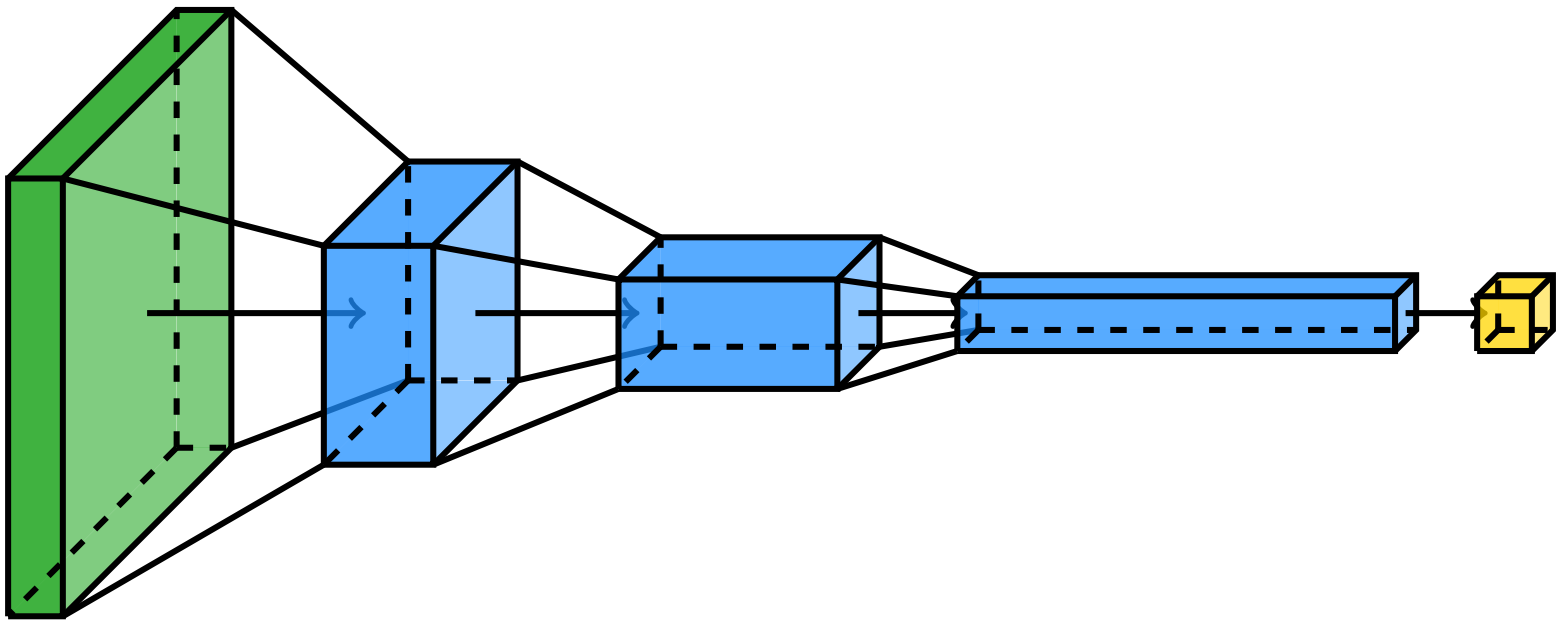
LSTM


$\mathbf{h}_t$
$\mathbf{h}_t$
$\times$
φ
$\mathbf{C}_t$
$\mathbf{o}_t$
σ_o
$\mathbf{i}_t$
tanh
$\times$
$+$
$[\mathbf{x}_t, \mathbf{h}_{t-1}]$
σ_i
$\mathbf{C}_t^\mathrm{f}$
σ_f
$\times$
$\mathbf{f}_t$
$\mathbf{x}_t$
[ ]
$\mathbf{C}_{t-1}$
$\mathbf{h}_{t-1}$

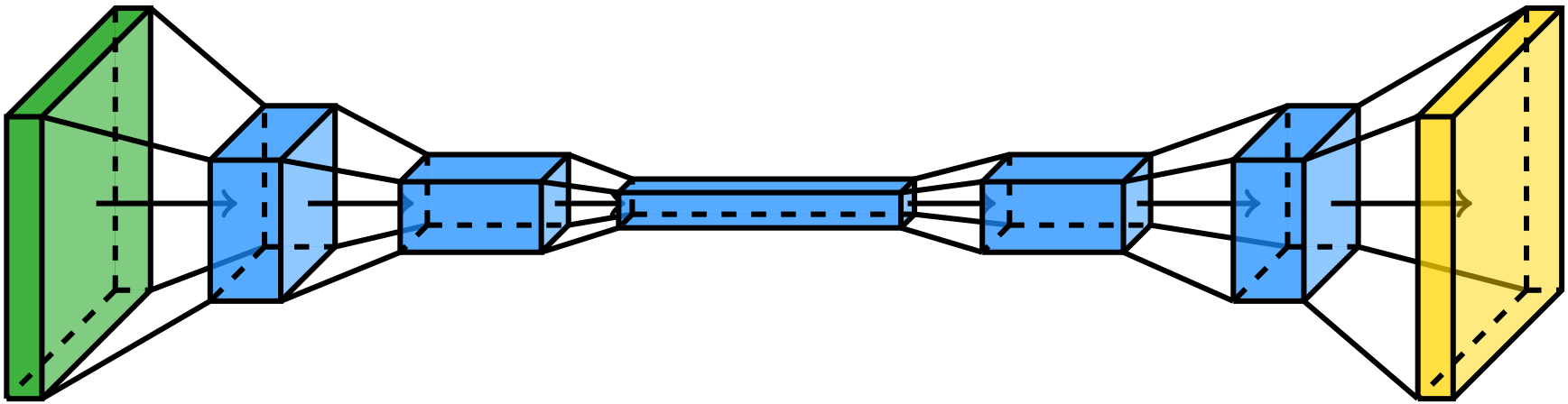
Recurrent neural network



Convolutional neural network

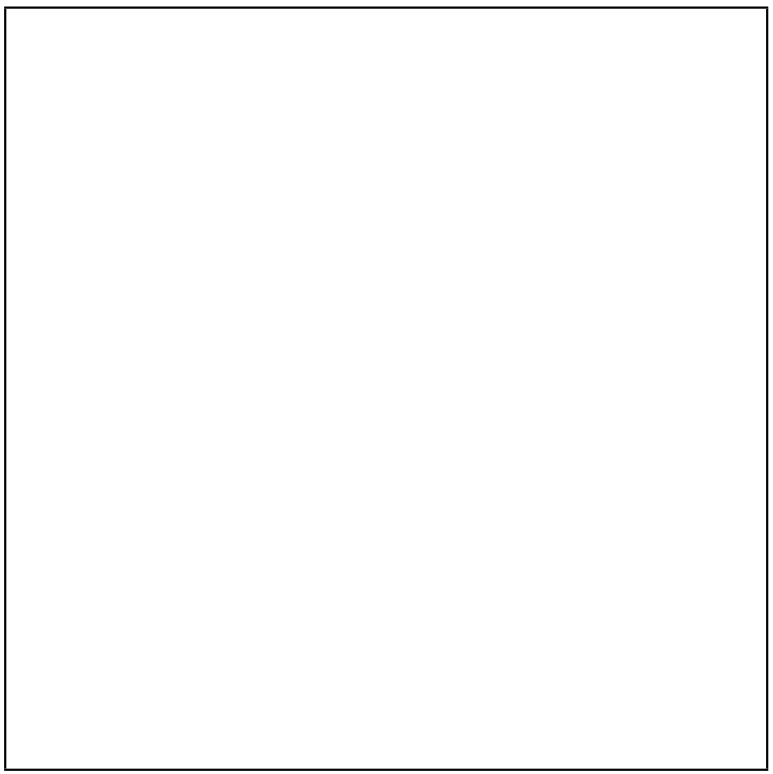


Autoencoder



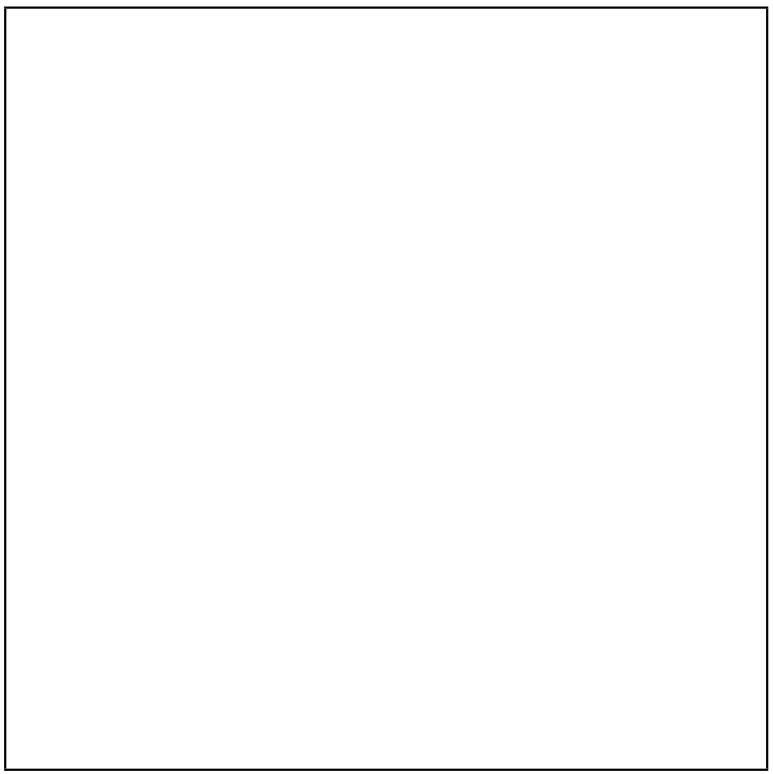
Deep convolutional neural networks

Deep convolutional neural networks



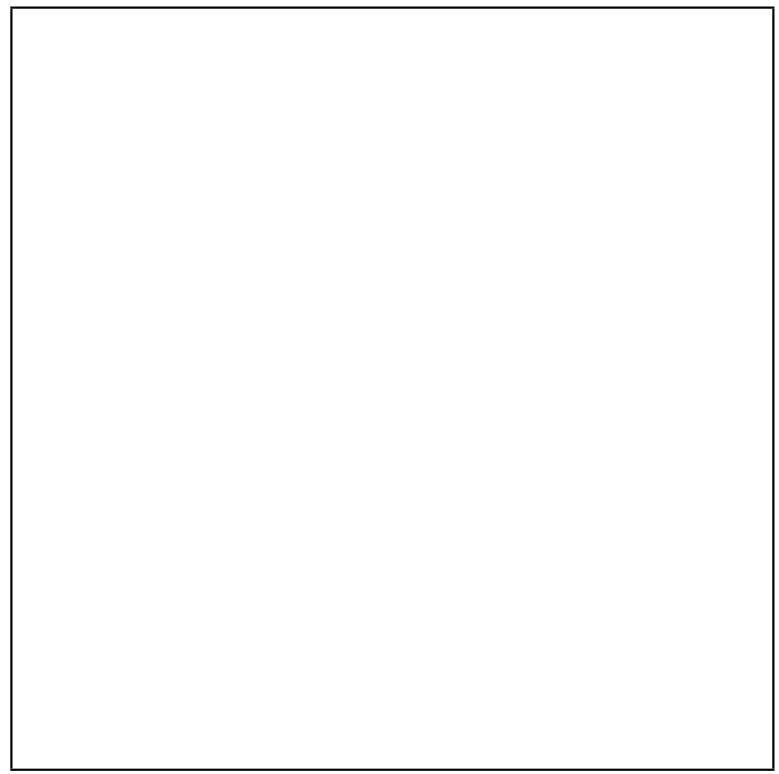
Deep convolutional neural networks

Deep convolutional neural networks



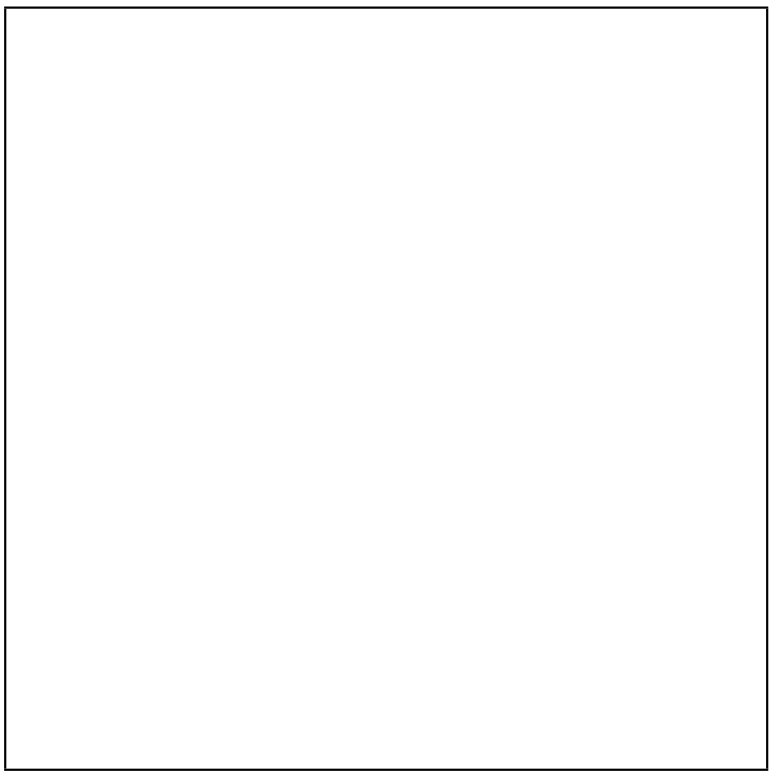
Deep convolutional neural networks

Deep convolutional neural networks

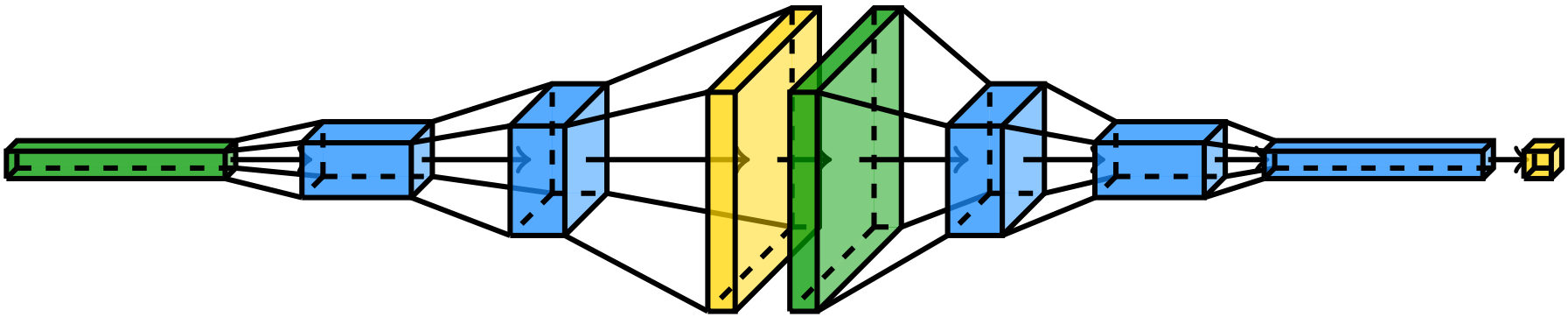


Deep convolutional neural networks

Deep convolutional neural networks



Deep convolutional neural networks



Resources

Overview
neuralnetworksanddeeplearning.com

Code examples
keras.io

Convolutional neural networks
cs231n.github.io