Machine Learning

Lecture 13

Intro to Neural Networks

Anna Kuzina

Learning Outcomes

After this lecture you should know:

- What is an artificial neuron, perceptron and artificial neural network
- What is a dense layer
- Which activation function are used
- How to train neural networks

Recap Discriminative models

Data:

$\{x_1, \ldots, x_n\}$ - observed variables

$\{y_1, \ldots, y_N\}$ - unobserved / target variables

Model with unknown parameters θ:

$$p(y \mid x, \theta)$$

Recap Linear Regression

Data:

$\{x_1, \ldots, x_n\}$ - observed variables

$\{y_1, \ldots, y_N\}$ - unobserved / target variables

Model with unknown parameters θ:

$$p(y \mid x, \theta)$$

Recap Logistic Regression

Data:

$\{x_1, \ldots, x_n\}$ - observed variables

$\{y_1, \ldots, y_N\}$ - unobserved / target variables

Model with unknown parameters θ:

$$p(y \mid x, \theta)$$

Recap Discriminative models

Linear Regression:

$$p(y \mid x, \theta) = \mathcal{N}(y \mid w^T x, \sigma^2)$$

Logistic Regression:

$$p(y \mid x, \theta) = \mathcal{B}e(y \mid \text{sigm}(w^T x))$$

Can we parametrise distributions in other ways?

Recap Discriminative models

Linear Regression:

$$p(y \mid x, \theta) = \mathcal{N}(y \mid w^T x, \sigma^2)$$

Logistic Regression:

$$p(y \mid x, \theta) = \mathcal{B}e(y \mid \text{sigm}(w^T x))$$

Can we parametrise distributions in other ways?

Regression:

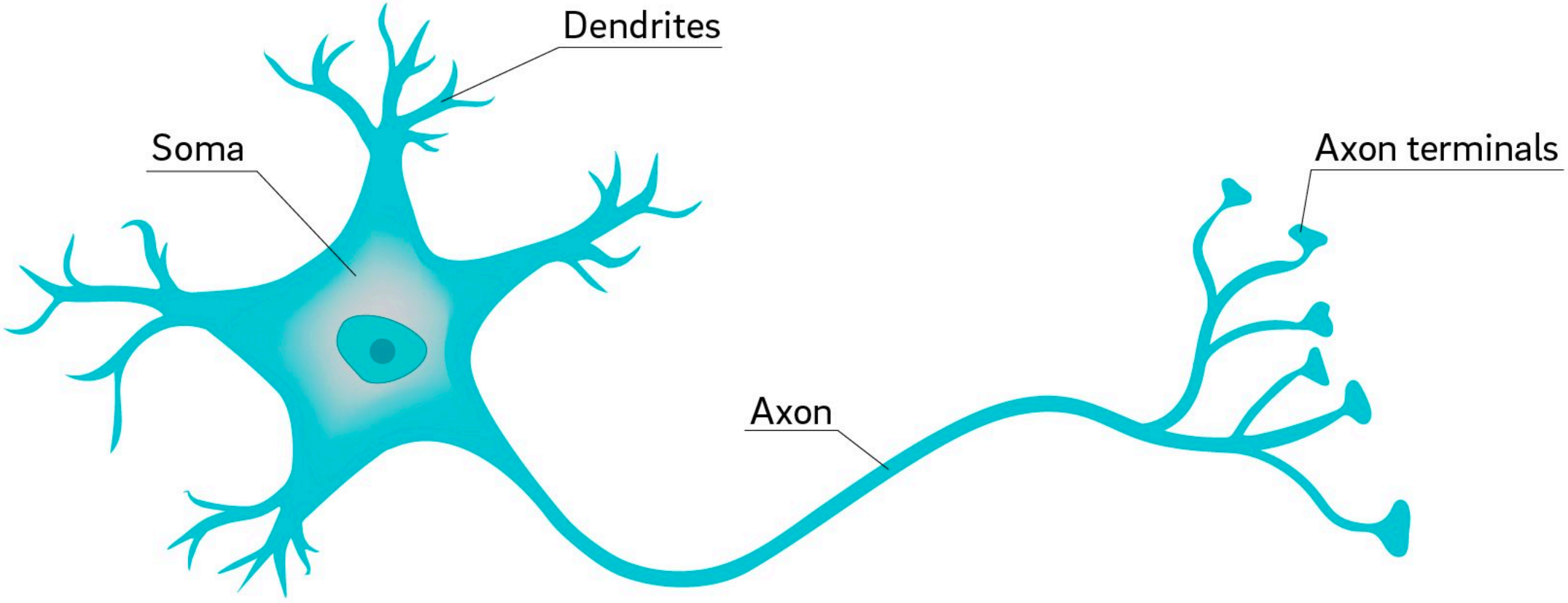
$$p(y \mid x, \theta) = \mathcal{N}(y \mid f(x), \sigma^2)$$

Binary Classification:

$$p(y \mid x, \theta) = \mathcal{B}e(y \mid f(x))$$

But we need to pay attention to the constraints!

Artificial Neuron: Inspiration

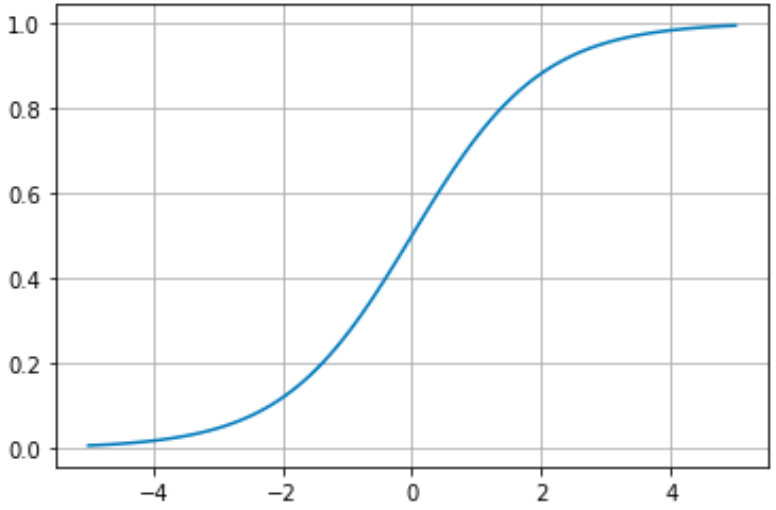


Artificial Neuron: Perceptron

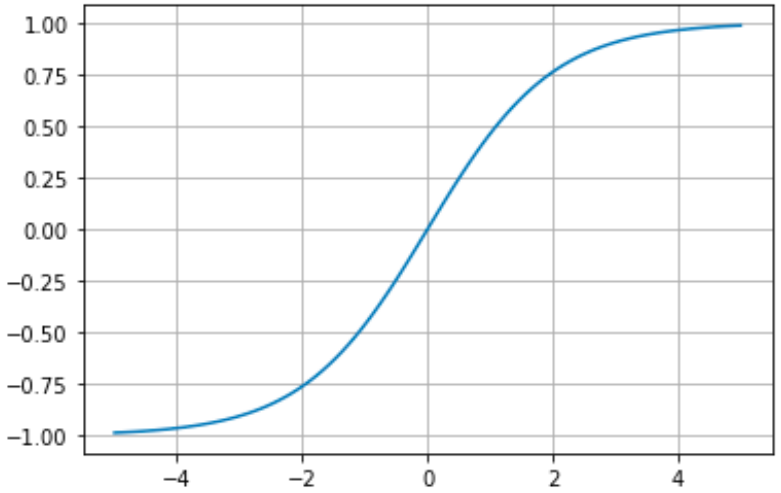
Artificial Neuron: Logistic Regression

Artificial Neuron: Activation Functions

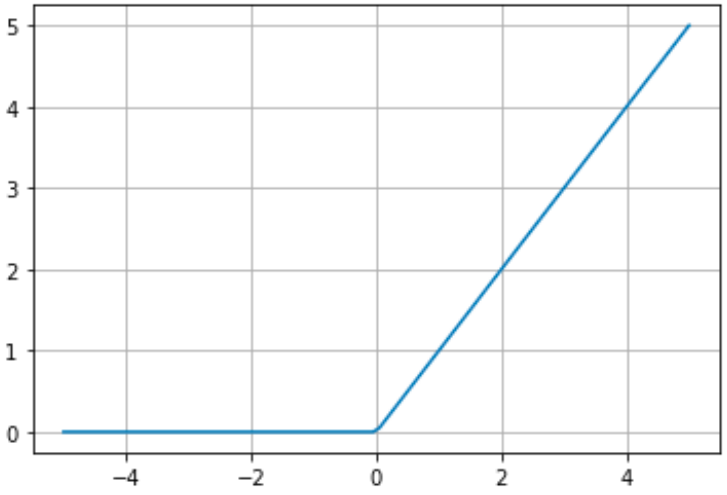
sigmoid



tanh



ReLU



Artificial Neural Network (sceme)

Artificial Neural Network (formula)

Artificial Neural Network

Layers in Artificial Neural Network

How to train?

SGD recap

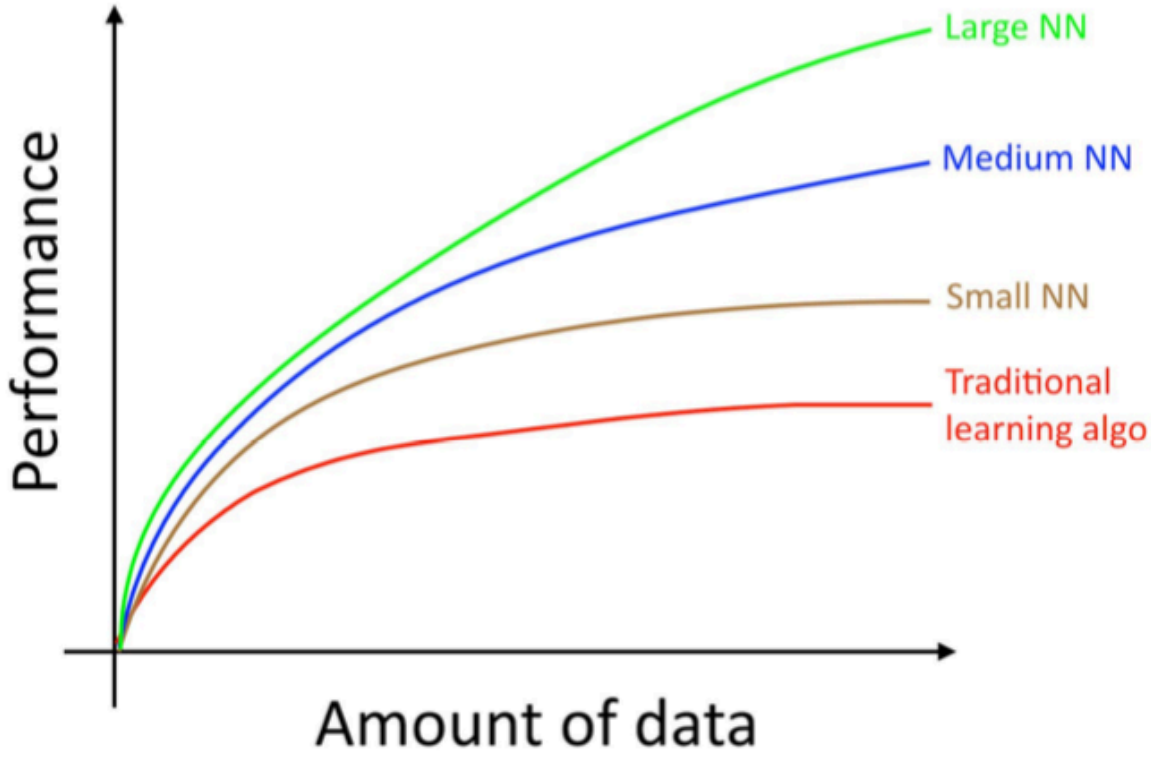
Chain Rule Recap

Chain Rule in NN

Backpropagation

Deep Neural Networks

- In theory can approximate any arbitrary function
- Benefit from large datasets
- Can be implemented fast on GPU



Tasks

- Supervised: regression and classification
- Representation Learning
- Image / Text generation (next lecture)

Limitations and Challenges

- Vanishing Gradient
  Other activation functions, BatchNorm, ResNet

- Overfitting
  Regularization, Data Augmentation, Dropout

- Requires Large Dataset
  Transfer Learning